

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de fin de cycle
en vue de l'obtention du diplôme de master professionnel en Informatique
Option : Administration et sécurité des réseaux informatique

Thème

Conception et réalisation d'une application réseau pour la gestion du restaurant cas d'étude : hôtel CHREA

Présenté par :
✓ *M^{lle}* Brachouche Souhila.

Proposé et encadré par :
✓ M. OUZEGGANE Redouane.

Devant le jury composé de :

✓ Président : M^r. ABBACHE Bournane.
✓ Examineur 1 : M^r. NAFI Mohammed.

Promotion 2012/2013

**** Remerciements ****

Je remercie DIEU tout puissant de mon avoir donné la force, la santé, le courage et la patience de pouvoir accomplir ce travail.

Je tiens à remercier vivement mon promoteur M^r OUZEGGANE Redouane d'avoir accepté de me guider tout au long du travail.

Je tiens aussi à remercier également tous les membres de jury pour avoir accepté d'évaluer mon travail.

Mon remerciement va également à tout le personnel de l'hôtel CHREA pour leur hospitalité.

Un grand merci à toute ma famille surtout mes parents pour leurs encouragement et leurs suivi avec patience du déroulement de mon projet.

Mes sincères remerciements s'adressent aussi à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

* * * * *Dédicaces* * * * *

Je dédie ce modeste travail :

À la mémoire de mon père .

A ma chère mère pour ses sacrifices depuis qu'elle m'a mis au monde.

À ma sœur Lynda et mon frère Farid.

À mon adorable nièce Hanane.

À mes grands-mères que dieu les bénies.

À mes oncles, tantes ,cousins et cousines.

À mes amis(es) ceux avec j 'ai parcouru le long chemin avec tant de peine et de joie.

Mlle Brachouche Souhila

TABLE DES MATIÈRES

Table des Matières	i
Liste des abréviations	iv
Liste des Figures	v
Liste des Tableaux	vii
Introduction générale	1
1 Cadre générale	3
1.1 Introduction	3
1.2 Présentation de l'organisme d'accueil	3
1.2.1 Présentation de l'hôtel Chréa	3
1.2.2 La structure d'hôtel	4
1.2.2.1 Le lieu de réception	4
1.2.2.2 Les chambres	4
1.2.2.3 Le restaurant	4
1.2.2.4 La salle polyvalente	4
1.2.3 Description du cadre d'étude	4
1.2.3.1 Description du restaurant	4
1.2.3.2 Organigramme du restaurant	5
1.2.3.3 Le personnel du restaurant et ses activités	5
1.2.3.4 Problématique	6
1.2.3.5 Objectif de l'étude	6
1.3 Présentation de la méthodologie de conception UML	6
1.3.1 Définition	6

1.3.2	Les diagramme d’UML	6
1.3.2.1	Les diagramme structurels	7
1.3.2.2	Les diagramme de comportement	8
1.3.3	Présentation du processus de développement UP	8
1.3.4	Définition	9
1.3.5	Structure d’UP	9
1.3.5.1	Les phases du processus	10
1.3.5.2	Les activités du processus	11
1.4	Conclusion	12
2	Capture des besoins	13
2.1	Introduction	13
2.2	Etude préliminaire	13
2.2.1	identification des acteurs	13
2.2.2	identification des messages	14
2.2.3	Diagramme de contexte	16
2.3	Capture des besoins fonctionnels	17
2.3.1	Identification des cas d’utilisation	17
2.3.2	Représentation et description des diagrammes des cas d’utilisation	17
2.3.2.1	Diagramme de cas d’utilisation associé à l’utilisateur	18
2.3.2.2	Diagramme de cas d’utilisation associé au directeur du restaurant	21
2.3.2.3	Diagramme de cas d’utilisation associé au serveur	29
2.4	Capture des besoins non fonctionnels	30
2.5	Conclusion	30
3	Analyse et conception	31
3.1	Introduction	31
3.2	Analyse	31
3.2.1	Présentation du diagramme de séquence	32
3.2.1.1	Eléments de base d’un diagramme de séquence	32
3.2.1.2	Elaboration du digramme de séquence	33
3.3	Conception	44
3.3.1	Présentation de diagramme de classe	44
3.3.1.1	Eléments de base d’un diagramme de classe	44
3.3.1.2	Règle de passage	45
3.3.1.3	Elaboration du diagramme de classe	47
3.3.1.4	Dictionnaire de données	47
3.3.1.5	Passage au modèle relationnel	49
3.3.1.6	Elaboration du modèle relationnel	50

3.3.2	Présenatation du diagramme de déploiement	51
3.3.2.1	Eléments de base d'un diagramme de déploiement	51
3.3.2.2	Elaboration de diagramme de déploiement	52
3.4	Conclusion	52
4	Réalisation	53
4.1	Introduction	53
4.2	Les outils d'implémentation d'une base de données	53
4.2.1	Présentation du MySQL-Server	53
4.2.2	Présentaion du phpMyAdmin	53
4.3	Les outils de développement de l'application	54
4.3.1	Présentation du langage de programmation java	54
4.3.2	Présentation de l'environnement Eclipse	54
4.4	Représenation des interface de l'application	55
4.4.1	Interface authentification	55
4.4.2	Interface Principale	56
4.4.3	Interface gestion de table	56
4.4.4	Interface Editer une commande	57
4.4.5	Interface Plan de table	58
4.5	Conclusion	58
	Conclusion générale	59
	Bibliographie	59

LISTE DES ABRÉVIATIONS

EDI	Environnement de Développement Intégré
HTML	Hyper Text Markup Language
IBM	International Business Machine
IDE	Integrated Development Environment
IHM	Interface Homme Machine
MySQL	My Structured Query Language
PHP	Hyper Text PreProcessor
SD	Sequence Diagram
SGBD	Système de Gestion de Base de Données
SQL	Structured Query Language
UML	Unified Modeling Language
UP	Unified Process

LISTE DES FIGURES

1.1	Organigramme du restaurant.	5
1.2	Schéma descriptif de La structure d'UML.	7
2.1	la relation de généralisation entre les acteurs du système.	14
2.2	Diagramme de contexte.	16
2.3	Diagramme de cas d'utilisation associé à l'utilisateur.	18
2.4	Diagramme de cas d'utilisation associé au directeur du restaurant.	21
2.5	Diagramme de cas d'utilisation au serveur.	29
3.1	Diagramme de séquence du cas d'utilisation s'authentifier.	34
3.2	Diagramme de séquence du cas d'utilisation changer le mot de passe	35
3.3	Diagramme de séquence du cas d'utilisation gérer une salle.	36
3.4	Diagramme de séquence du cas d'utilisation gérer une table.	37
3.5	Diagramme de séquence du cas d'utilisation gérer une catégorie de consommation .	38
3.6	Diagramme de séquence du cas d'utilisation gérer une consommations.	39
3.7	Diagramme de séquence du cas d'utilisation élaborer un plan des tables.	40
3.8	Diagramme de séquence du cas d'utilisation gérer un utilisateur	41
3.9	Diagramme de séquence du cas d'utilisation gérer un rôle.	42
3.10	Diagramme de séquence du cas d'utilisation éditer une commande	43
3.11	Diagramme de classe.	47
3.12	Présentation graphique d'un noeud.	51
3.13	Présentation graphique d'un artéfact.	51
3.14	Présentation du chemin de communication entre des noeuds.	52
3.15	Diagramme de déploiement de l'application à réaliser.	52
4.1	Interface authentification.	55
4.2	Interface changer le mot de passe.	55

4.3	Interface principale.	56
4.4	Interface de la gestion de table.	56
4.5	Interface Editer une commande.	57
4.6	Interface Plan de table.	58

LISTE DES TABLEAUX

1.1	La liste du personnel du restaurant	5
1.2	Description des phases du processus UP	10
1.3	Description activités du processus UP	11
2.1	Les acteurs du système.	14
2.2	Les messages émis et reçus entre les acteurs et le système.	15
2.3	Les cas d'utilisation du système.	17
2.4	Le formalisme de description des cas d'utilisation.	18
2.5	Description du cas d'utilisation s'authentifier.	19
2.6	Description du cas d'utilisation changer le mot de passe.	20
2.7	Description du cas d'utilisation un rôle.	22
2.8	Description du cas d'utilisation gérer un utilisateur.	23
2.9	Description du cas d'utilisation gérer une salle	24
2.10	Description du cas d'utilisation gérer une table.	25
2.11	Description du cas d'utilisation gérer une catégorie de consommation.	26
2.12	Description du cas d'utilisation gérer une consommation.	27
2.13	Description du cas d'utilisation élaborer un plan des tables.	28
2.14	Description du cas d'utilisation éditer une commande.	29
3.1	Dictionnaire de données.	48

INTRODUCTION GÉNÉRALE

L'informatique étant une science de traitement automatique de données qui s'avère bénéfique dans tous les domaines qu'ils soient scientifiques ou professionnels, privés et/ou publics. En observant les grandes entreprises dans le monde, on se rend vite compte qu'elles réalisent des travaux complexes en des fractions de temps très réduit à l'aide des machines, ce qui leur coûterait des journées manuellement.

Vu la complexité des activités d'une entreprise, l'outil informatique est un élément qui lui facilite une bonne gestion pour une meilleure prise des décisions dans le but d'orienter la politique générale de celle-ci.

La gestion étant un ensemble d'activités, de décisions qui se déroulent dans une entreprise ou, plus généralement, dans une organisation (administration, association, groupe,...) nécessite un suivi délicat qui pourra permettre aux responsables de celle-ci d'appliquer un contrôle efficace, de bien traiter les informations comptables, financières.

À l'issue de nos observations lors de nos pré-enquêtes, nous avons constaté que le restaurant du l'hôtel CHREA connaît actuellement assez de difficultés liées à son mode de gestion manuelle entraînant le gaspillage du temps lors du traitement des opérations des clients au restaurant, tel que les erreurs dans les commandes des clients, les retards dans le service des clients à cause des allers-retours inutile en cuisine qui fait retarder les serveurs, la lenteur dans l'élaboration des rapports journalier, mensuel et annuel.

Pour répondre aux problèmes de notre étude, nous pensons que la mise en place d'un système de gestion informatisée du restaurant de l'hôtel CHREA permettrait d'améliorer son système de gestion.

L'objectif de notre projet est de concevoir et réaliser un système d'information fiable et de qualité concrétisé sous la forme d'une application trois tiers avec une base de données gérés par le serveur MySQL.

Afin de mener à bien notre travail nous avons fragmenté notre processus de développement en quatre étapes qui se caractérisent sous quatre chapitres dans ce mémoire :

Le premier chapitre nommé " Cadre général " est consacré à la description de l'organisme d'accueil et ses différentes fonctions ainsi les différentes méthodes et langage de conception qui existent ou nous mettrons l'accent sur UML ainsi que le processus de développement adopté.

Le deuxième chapitre est voué à la spécification des exigences fonctionnelles et non fonctionnelles de notre système.

Le troisième chapitre intitulé analyse et conception est consacrée à la réalisation d'un modèle dynamique modélisé par le diagramme de séquence pour la représentation des interactions selon un point de vue temporel et le séquençement des opérations, ainsi qu'un modèle statique représenté par le diagramme de classe et objet qui nous permettra d'avoir un aperçu sur la base de données et un diagramme de déploiement décrivant la disposition physique des ressources matérielles et logicielle qui composent le système.

Le quatrième chapitre est dédié à la réalisation où nous exposerons les techniques d'implémentation que nous avons mises en oeuvre et nous donnerons un aperçu du résultat final à travers quelques IHM (interface homme machine).

CHAPITRE 1

CADRE GÉNÉRALE

1.1 Introduction

Dans le présent chapitre, nous entamons la présentation de notre cas d'étude. Tout d'abord, nous présentons le cadre de notre travail avec la définition de la problématique et nous traçons ensuite les objectifs de notre projet.

Et enfin, nous allons définir quelques généralités portant sur la méthode et outils mettant en évidence la réalisation de notre projet. Nous allons commencer par présenter le langage de modélisation unifié UML (Unified Modeling Language), définir la démarche générique du processus de développements logiciel qui l'accompagne.

1.2 Présentation de l'organisme d'accueil

1.2.1 Présentation de l'hôtel Chréa

L'hôtel est situé en plein centre urbain de la ville de Bejaïa à proximité de la maison de la culture et carrefour Aamirouche, sur le grand boulevard Krim Belkacem. Inauguré le 29 Mars 2000 par le wali de Bejaia, est opérationnel depuis le 02 avril 2000, géré par un encadrement spécialisé ainsi qu'un personnel d'exploitation professionnel hautement qualifié, et donc en mesure de satisfaire une clientèle des plus en plus exigeantes.

1.2.2 La structure d'hôtel

1.2.2.1 Le lieu de réception

L'hôtel dispose d'un lieu de transition certes, mais beaucoup plus conçu comme un salon de détente, la clientèle y est systématiquement prise en charge en toute diligence dans un contexte architectural agréable et confortable par une équipe aussi dévouée qu'attentive.

1.2.2.2 Les chambres

Il dispose de 34 chambres, 10 suites juniors, une suite sénior, un appartement. Toutes les chambres sont dotées d'un excellent niveau de confort et de commodités avec des équipements de qualité assurant ainsi la sensation de bien être et de détente totale/mobilier de haut standing, télévision avec réception satellite, chauffage et climatisation, téléphone et internet.

1.2.2.3 Le restaurant

Il est de type gastronomique qui a une capacité d'accueil pour 120 couverts et offre une gamme culinaire riche et varié, et le chef est constamment attentif à toute suggestion et remarque.

1.2.2.4 La salle polyvalente

Il Dispose d'une grande salle. Elle est réservée pour les séminaires, les conférences et les colloques.

L'hôtel dispose aussi des autres commodités comme :

- ✓ Une pizzeria.
- ✓ Une agence de location de véhicule.
- ✓ Une agence de voyage.
- ✓ Une galerie de marchande.
- ✓ Une salle des fêtes.
- ✓ Un salon de coiffure pour dames

1.2.3 Description du cadre d'étude

Nous consacrons notre étude sur le restaurant de l'hôtel :

1.2.3.1 Description du restaurant

Le restauration est considéré parmi les services important de l'hôtel, il est de type gastronomique, il propose à ses clients soit des menu à prix fixe sans ou avec choix, il dispose de 25

table, avec la capacité de 120 couverts, et le moyen utilisé pour effectuer une réservation de table par les clients soit par téléphone ou par fax. Le restaurant est ouvert 7 jours sur 7 et les heures d'ouverture de midi jusqu'à 14h30 et le soir de 19h jusqu'à 22h30.

1.2.3.2 Organigramme du restaurant

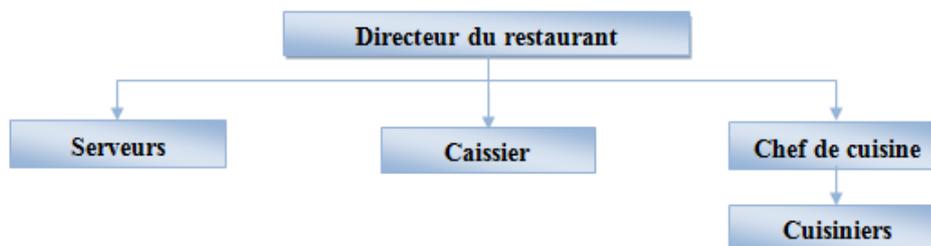


FIGURE 1.1 – Organigramme du restaurant.

1.2.3.3 Le personnel du restaurant et ses activités

Le tableau suivant montre les différents types du personnel du restaurant ainsi que leurs rôles respectifs.

Personnels	Rôle
Directeur du restaurant	<ul style="list-style-type: none"> - Il organise le travail du personnel ; - Il participe à la l'élaboration des plats inscrit sur la carte ou sur le menu avec le chef de cuisine ; - Il est responsable des ventes. - Il veille à la qualité de service.
Serveurs	<ul style="list-style-type: none"> - Ils dressent les tables ; - Ils prennent les commandes des clients ; - Ils servent les clients et encaisse leurs règlement ; - Ils nettoient les tables et les salles ; - Ils débarassent les tables.
Caissier	<ul style="list-style-type: none"> - IL se charge de la caisse ; - Il prépare les factures des clients ;
Chef de cuisine	<ul style="list-style-type: none"> - Il donne les instructions et des ordres aux personnels de cuisine ; - Il se charge de la préparation de la liste des plats qui seront sur le menu ; - Il prépare les plats les plus complexe.
Cuisiniers	<ul style="list-style-type: none"> - Ils préparent les plats.

TABLE 1.1 – La liste du personnel du restaurant

1.2.3.4 Problématique

Après avoir entamé notre étude au sein de l'hôtel " CHREA" et effectué une interview avec le responsable concerné, nous avons pu recenser les problèmes suivants :

L'utilisation de carnet de bons et le stylo pour prendre les commandes, ces méthodes ne vont satisfaire totalement les services en salle de restaurant et nous allons constater qu'il y aura beaucoup de dysfonctionnements :

- Un chef qui n'arrive à relire les commandes.
- des erreurs des préparation des plats(mauvaise écriture sur le bon de commande).
- des allers retours inutiles en cuisine(absence des serveurs auprès des clients).

1.2.3.5 Objectif de l'étude

L'objectif principal de l'hôtel" CHREA " est d'automatiser la gestion de son restaurant afin de simplifier le pilotage des activités de chaque personnel du restaurant pour cela la solution informatique a pour objectif de :

- Automatiser certaines procédures tel que : saisie des bons de commande.
- Optimiser le temps.
- Organisation de l'espace de travail.

1.3 Présentation de la méthodologie de conception UML

1.3.1 Définition

UML(Unified Modeling Language) est un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, concevoir des solutions et communiquer des points de vue.

UMLunifie à la fois les notations et les concepts orientés objet, et également les notations nécessaires aux différentes activités d'un processus de développement [5].

1.3.2 Les diagramme d'UML

UML2comporte treize diagrammes qui sont utilisés dans la description d'un système, ces diagrammes sont regroupés dans deux grands ensembles :

Les diagrammes structurels : ils sont au nombre de six et ils représentent l'aspect statique d'un système (classes, objets, composants,etc).

Les diagrammes de comportement : ils sont au nombre de sept et ils représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs.

Ces différents diagrammes sont représentés dans le schéma suivant :

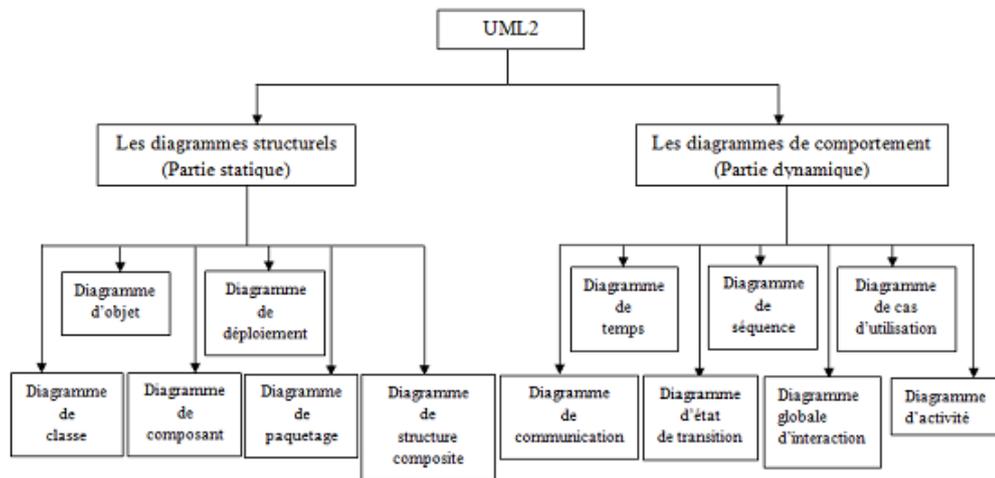


FIGURE 1.2 – Schéma descriptif de La structure d'UML.

1.3.2.1 Les diagramme structurels

Les six diagrammes structurels se présentent comme suit[5] :

- **Diagramme de classes** :Il montre les briques de base statiques : Classes, associations, interfaces, attributs, opérations, généralisations, etc.
- **Diagramme d'objets** :Il montre les instances des éléments structurels et leurs liens à l'exécution.
- **Diagramme de packages** :Il montre l'organisation logique du modèle et les relations entre packages.
- **Diagramme de structure composite** :Il montre l'organisation interne d'un élément statique complexe.
- **Diagramme de composants** :Il montre des structures complexes, avec leurs interfaces fournies et requises.

- **Diagramme de déploiement** : Il montre le déploiement physique des " artefacts " sur les ressources matérielles .

1.3.2.2 Les diagramme de comportement

Les sept diagrammes de comportement se présentent comme suit[5] :

- **Diagramme de cas d'utilisation** : Il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.
- **Diagramme de séquence** : : Il montre la séquence verticale des messages passés entre objets au sein d'une interaction
- **Diagramme de communication** : : Il montre la séquence verticale des messages passés entre objets au sein d'une interaction
- **Diagramme d'états** : Il montre les différents états et transitions possibles des objets d'une classe.
- **Diagramme de temps** : Il fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.
- **Diagramme d'activité** : Il montre l'enchaînement des actions et décisions au sein d'une activité.
- **Diagramme globale d'interaction** : Il fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.

Pour l'analyse et la conception de notre projet, nous utiliserons les diagrammes suivants : diagramme de cas d'utilisation, de séquence, de classes et de déploiement.

1.3.3 Présentation du processus de développement UP

Pour effectuer la phase de modélisation nous avons opté pour le langage d'analyse et de conception UML2 qui nous offre la possibilité de manipuler plusieurs diagrammes pour décrire les différentes fonctionnalités du système sous ses multiples aspects statiques soient-ils ou dynamique.

Cependant on ne peut exploiter les diagrammes qu'offre UML sans adopter un processus de développement unifié qui viendra régir et encadrer la mise en IJuvre du système depuis la

capture des besoins jusqu'à sa livraison et le processus qu'on va utiliser est le UP.

1.3.4 Définition

Le Processus Unifié UP (Unified Process) est un processus de développement logiciel, associé à UML, met en oeuvre les principes suivants [5] :

Processus Itératif et incrémental : le projet est découpé en itérations de courte durée qui aident à mieux suivre l'avancement global de celui-ci. A la fin de chaque itération, une partie exécutable du système final est produite de façon incrémentale.

Processus Centré sur l'architecture : tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitée. Cette architecture qu'elle soit fonctionnelle, logique, matérielle ou autre doit être modélisée en UML et pas seulement documentée en texte.

Processus Conduit par les cas d'utilisation : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation permettent d'exprimer les interactions du système avec les utilisateurs, donc de capturer les besoins. Il sert aussi à montrer comment ces derniers constituent un vecteur structurant pour le développement et les tests du système.

Processus Piloté par les risques : l'analyse des risques doit être présente à tous les niveaux du développement d'un système. Il est important de bien évaluer ces risques afin d'aider à la bonne prise de décision. Du fait de l'application du processus itératif, le processus unifié contribue à la diminution des risques au fur et à mesure du déroulement des itérations successives.

1.3.5 Structure d'UP

UP gère le processus de développement par deux axes :

L'axe vertical : représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l'aspect statique du processus qui s'exprime en termes de composants, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.

L'axe horizontal : représente le temps et montre le déroulement du cycle de vie du processus ; cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en terme

de cycles, de phases, d'itérations et de jalons.

1.3.5.1 Les phases du processus

Le processus up est divisé en quatre phases [5] :

Les phases	Description
Inception(Lancement)	définir la "vision" du projet, sa portée, sa faisabilité, son business case, afin de pouvoir décider au mieux de sa poursuite ou son arrêt.
Élaboration	<ul style="list-style-type: none">- identifier et décrire la majeure partie des besoins des utilisateurs ;- construire l'architecture de base du système ;- lever les risques majeurs du projet.
Construction	<ul style="list-style-type: none">- concevoir et implémenter l'ensemble des éléments opérationnels (autres que l'architecture de base).
Transition	<ul style="list-style-type: none">- construire une version finale ;- tester le produit en vue de livraison.

TABLE 1.2 – Description des phases du processus UP

1.3.5.2 Les activités du processus

Les activités menées à l'intérieur des quatre phases sont les suivantes [3] :

Les activités	Description
Expression des besoins	<ul style="list-style-type: none"> -comprendre le contexte du système en produisant un modèle du domaine ; -Appréhender les besoins fonctionnels qui conduisent à des modèles de cas d'utilisation ; -Appréhender les besoins non fonctionnels et livrer une liste des exigences supplémentaires.
Analyse	<ul style="list-style-type: none"> - permet une formalisation du système de développer en réponse à l'expression des besoins formulée.
Conception	<ul style="list-style-type: none"> - prendre en compte les choix d'architecture technique retenus pour le développement et l'exploitation du système ; -permet d'étendre la représentation des diagrammes effectuée au niveau de l'analyse en intégrant les aspects techniques plus proche de préoccupation physique.
Tests	<ul style="list-style-type: none"> - vérifier la bonne implémentation de toutes les exigences(fonctionnels et techniques ; - vérifier le fonctionnement correct des interactions entre les objets ; - vérifier la bonne intégration de tous les composants dans le logiciel.

TABLE 1.3 – Description activités du processus UP

1.4 Conclusion

Nous avons vu, en détail, dans ce chapitre la structure et les différents services de l'organisme d'accueil " Hôtel CHREA", ainsi illustré la problématique et défini les objectifs du ce projet b.

Et enfin, nous avons présenté la méthode et le processus de développement adopté pour réaliser ce projet.

CHAPITRE 2

CAPTURE DES BESOINS

2.1 Introduction

Cette première étape, nous permettra de délimiter le périmètre de notre projet, elle consiste à effectuer un premier repérage des besoins fonctionnels et non fonctionnels du système, ainsi que l'identification des acteurs qui interagissent avec le système et les messages échangés entre eux.

2.2 Etude préliminaire

Dans cette partie nous effectuerons une étude préliminaire des principaux acteurs du système, et définir le rôle de chacun.

2.2.1 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. Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données [1].

Nous allons présenter dans le tableau ci-dessous les acteurs de notre système :

Les acteurs	Fonctionnalités
Utilisateur	est un acteur générique qui doit s'authentifier pour accéder au système, ainsi il peut changer son mot de passe.
Directeur du restaurant	est un acteur qui peut définir les tables et les salles du restaurant, la consommation et ses catégories, les utilisateurs du système et leurs rôles. Il peut aussi élaborer un plan des tables pour chaque salle du restaurant.
Serveur	- est un acteur qui prend les commandes des clients et l'addition.

TABLE 2.1 – Les acteurs du système.

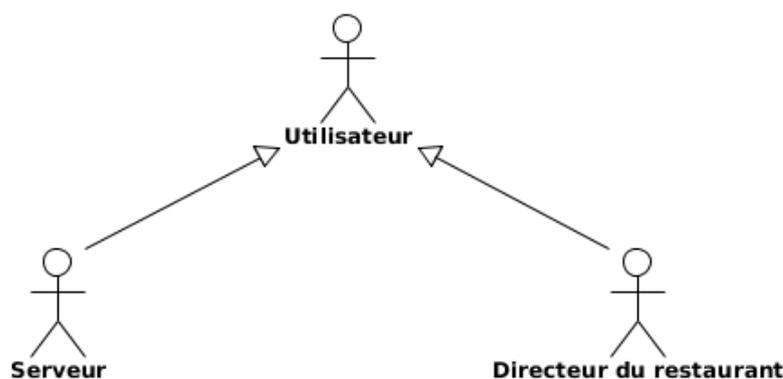


FIGURE 2.1 – la relation de généralisation entre les acteurs du système.

2.2.2 identification des messages

Définition : Un message représente la spécification d'une communication unidirectionnel entre objets qui transporte de l'information avec l'intention, de déclencher une activité chez le récepteur [2].

Les différents acteurs de notre système, peuvent échanger des messages qu'ils soient émis ou reçus. En effet, les messages émis représentent les demandes qu'un acteur effectue tandis que les messages reçus représentent la réponse du système à une demande donnée.

Le tableau ci-dessous représente les messages circulant entre les acteurs et le système soit des messages émis ou des messages reçus :

acteurs	messages émis	messages reçus
Utilisateur	M1-Demande de s'authentifier. M2-Demande de changer le mot de passe.	m1-accès au système. m2-confirmation.
Directeur du restaurant	M3-Demande de gestion des salles(ajouter, supprimer, modifier). M4-Demande de gestion des tables(ajouter, supprimer, modifier). M5-Demande de gestion des catégories de consommation(ajouter, supprimer, modifier). M6-Demande de gestion des consommations(ajouter, supprimer, modifier). M7-Demande de gestion des utilisateurs(ajouter, supprimer, modifier). M8-Demande de gestion des rôles(ajouter, supprimer, modifier). M9-Demande d'élaboration un plan des tables.	m3- la liste des salle/confirmation. m4- la liste des tables/confirmation. m5- la liste des catégories/confirmation. m6- la liste des consommation/confirmation. m7- la liste des utilisateur/confirmation. m8- la liste des rôles/confirmation. m9- confirmation de création.
Serveur	M10-Demande d'affichage le plan des tables. M11-Demande de création d'une commande.	m10- affichage du plan. m11- confirmation de la création et envoie de la commande.

TABLE 2.2 – Les messages émis et reçus entre les acteurs et le système.

2.2.3 Diagramme de contexte

Un diagramme de contexte dynamique représente les messages échangés entre le système (le système est une boîte noire) et les acteurs.

Dans la figure ci-dessous, nous illustrerons le diagramme du contexte de notre système :

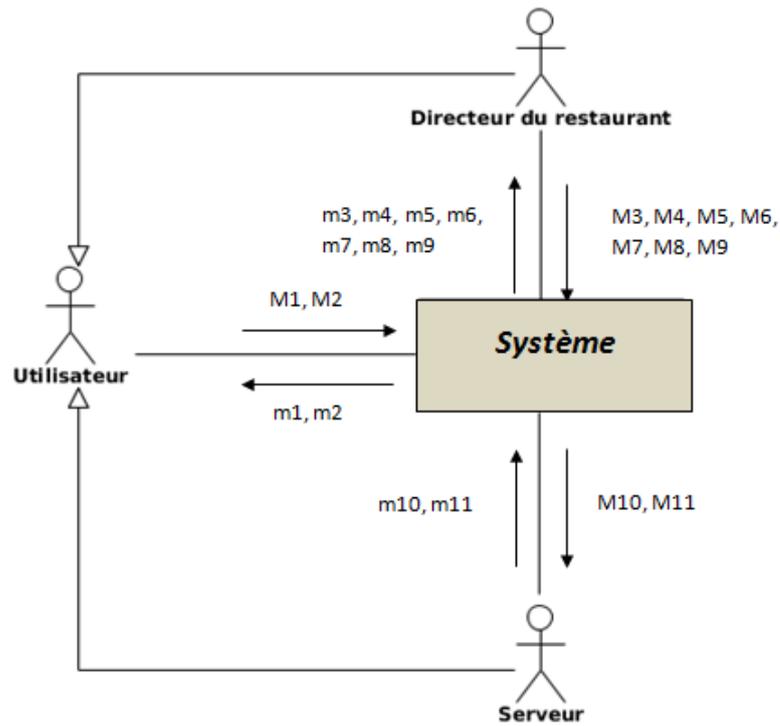


FIGURE 2.2 – Diagramme de contexte.

2.3 Capture des besoins fonctionnels

Cette phase formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire, elle va nous permettre de préciser l'étude du contexte fonctionnel du système, en décrivant les différentes façons qu'auront les acteurs d'utiliser le futur système [2].

2.3.1 Identification des cas d'utilisation

Définition : un cas d'utilisation permet de décrire l'interaction entre les acteurs et le système. La description de l'interaction est réalisée suivant le point de vue de l'utilisateur [3].

L'identification des cas d'utilisation, nous donne un aperçu des fonctionnalités futures que doit implémenter le système.

Dans le tableau ci-dessous, nous illustrons les cas d'utilisation de notre système :

Acteurs	Cas d'utilisation
Utilisateur	Authentification. Changer le mot de passe.
Directeur du restaurant	Gérer les catégories de consommation. Gérer les consommations. Gérer les salles. Gérer les tables. Elaborer un plan des tables. Gérer les rôles. Gérer les utilisateurs.
Serveur	Editer une commande.

TABLE 2.3 – Les cas d'utilisation du système.

2.3.2 Représentation et description des diagrammes des cas d'utilisation

Définition : Un diagramme de cas d'utilisation est un formalisme permettant de modéliser le fonctionnement d'un système par un découpage en fonctionnalités. Il illustre de plus la nature des interactions avec ces fonctionnalités offertes à titre de services à des acteurs externes au système. Chaque fonctionnalité est appelée un cas d'utilisation [4].

Formalisme de description des cas d'utilisation : Afin de décrire les interactions entre les cas d'utilisation, nous présentons ces derniers de façon textuelle. Cependant, cette

description prend souvent une forme rédigée qui convient mieux à la communication avec les utilisateurs. Des règles de structurations doivent être appliquées pour en faciliter l'expression, la compréhension et la cohérence.

Le tableau ci-dessous représente le formalisme choisi pour décrire les cas d'utilisation du système à réaliser :

Sommaire d'identification	
Titre :	Nom du cas d'utilisation
But :	Objectif du cas d'utilisation
Acteur :	Acteurs participants au cas d'utilisation
Description des enchaînements	
Pré conditions	
Condition qui doit être remplie avant le début du cas d'utilisation	
SCENARIO NOMINAL	
Il s'agit du scénario principal qui doit se dérouler sans incident et qui permet d'aboutir au résultat souhaité	
ENCHAÎNEMENT ALTERNATIF	
Les autres scénarios, secondaires ou correspondants à la résolution d'anomalies.	

TABLE 2.4 – Le formalisme de description des cas d'utilisation.

Dans ce qui suit, nous décrivons le diagramme détaillé des cas d'utilisation associés à chacun des acteurs du système à réaliser, ainsi la description textuelle de ce dernier :

2.3.2.1 Diagramme de cas d'utilisation associé à l'utilisateur

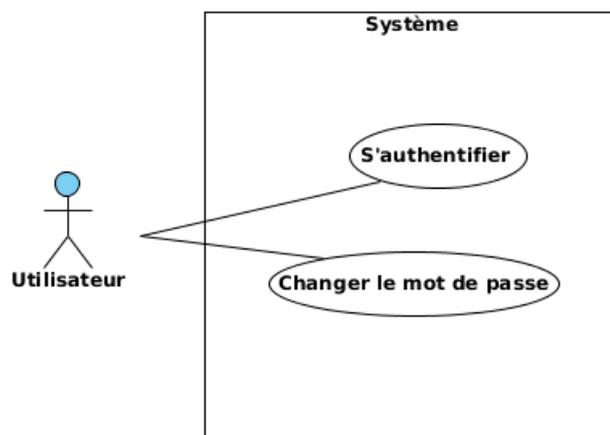


FIGURE 2.3 – Diagramme de cas d'utilisation associé à l'utilisateur.

Cas d'utilisation S'authentifier

Sommaire d'identification	
Titre :	S'authentifier
But :	Vérification de l'identité des utilisateurs
Acteur :	Utilisateur
Description des enchaînements	
Pré conditions	
Aucune	
SCENARIO NOMINAL	
<ol style="list-style-type: none"> 1. Le système affiche le formulaire de l'authentification, 2. L'utilisateur saisit son login et son mot de passe, 3. Le système vérifie la conformité des informations fournies E1 4. Le système donne l'accès à l'interface correspondante. 	
ENCHAÎNEMENT ALTERNATIF	
E1 : Erreur de mot de passe et/ou le login. <ol style="list-style-type: none"> 1- Le système affiche un message d'erreur, 2- Le système reprend de 2. 	

TABLE 2.5 – Description du cas d'utilisation s'authentifier.

Cas d'utilisation Changer le mot de passe

Sommaire d'identification	
Titre :	Changer le mot de passe
But :	Permet de changer son mot de passe pour accéder au système
Acteur :	utilisateur
Description des enchainements	
Pré conditions	
Aucune	
SCENARIO NOMINAL	
<ol style="list-style-type: none"> 1. Le système affiche le formulaire de l'authentification, 2. L'utilisateur demande de changer son mot de passe, 3. Le système affiche le formulaire, 4. L'utilisateur saisit les données, 5. Le système vérifie E1, 6. L'utilisateur valide, 7. Le système affiche un message de confirmation. 	
ENCHAINEMENT ALTERNATIF	
E1 : Les champs sont vides et/ou l'ancien mot de passe est incorrect. <ol style="list-style-type: none"> 1-Le système affiche un message d'erreur, 2-Le système reprend de 4. 	

TABLE 2.6 – Description du cas d'utilisation changer le mot de passe.

2.3.2.2 Diagramme de cas d'utilisation associé au directeur du restaurant

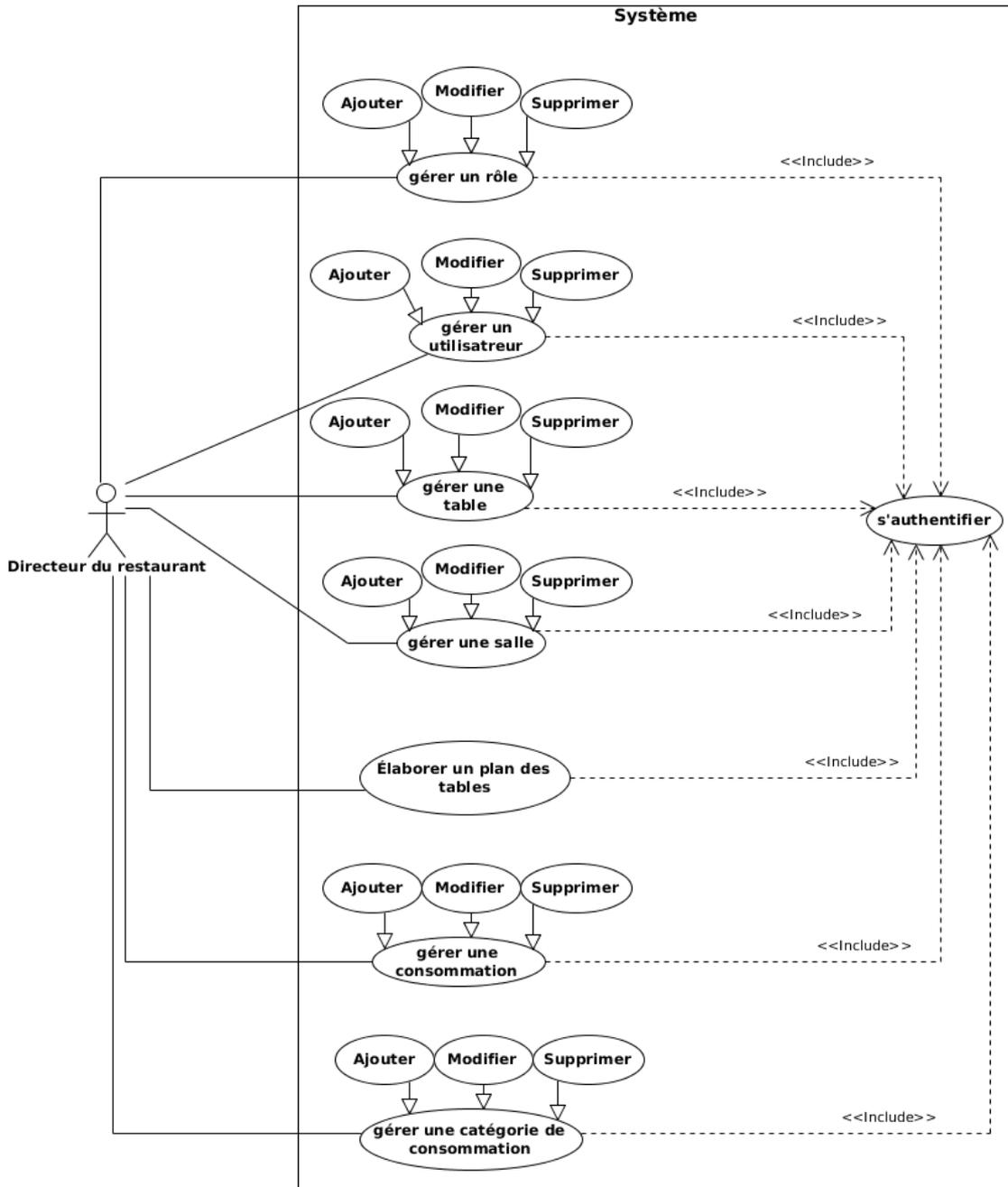


FIGURE 2.4 – Diagramme de cas d'utilisation associé au directeur du restaurant.

Cas d'utilisation Gérer un rôle

Sommaire d'identification	
Titre :	Gérer un rôle
But	Le directeur du restaurant peut ajouter ou modifier ou supprimer les rôles du système et le système met à jour la base de données.
Acteur	Directeur du restaurant
Description des enchainements	
Pré conditions	
Le directeur du restaurant doit s'authentifier.	
SCENARIO NOMINAL	
<ol style="list-style-type: none"> 1. Authentification, 2. Accès au système, 3. le directeur du restaurant demande la gestion des rôles, 4. Le système affiche l'interface, <ol style="list-style-type: none"> 4.1. Le directeur demande de créer un rôle, <ol style="list-style-type: none"> 4.1.1. Le système affiche le formulaire, 4.1.2. Le directeur saisit les données puis valide, 4.1.3. Le système affiche un message de confirmation, 4.2. Le directeur sélectionne un rôle et demande de le modifier, <ol style="list-style-type: none"> 4.2.1. Le système affiche le formulaire, 4.2.2. Le directeur ressaisi les données à modifier puis il valide, 4.2.3. Le Système affiche le rôle modifié, 4.3. Le directeur sélectionne un rôle et demande de le supprimer, <ol style="list-style-type: none"> 4.3.1. Le directeur valide, 4.3.2. Le système affiche un message de confirmation. 	
ENCHAINEMENT ALTERNATIF	
Aucun	

TABLE 2.7 – Description du cas d'utilisation un rôle.

Cas d'utilisation Gérer un utilisateur

Sommaire d'identification	
Titre :	Gérer un utilisateur
But	Le directeur du restaurant peut ajouter ou modifier ou supprimer les utilisateurs du système et le système met à jour la base de données.
Acteur	Directeur du restaurant
Description des enchainements	
Pré conditions	
Le directeur du restaurant doit s'authentifier.	
SCENARIO NOMINAL	
<ol style="list-style-type: none"> 1. Authentification, 2. Accès au système, 3. le directeur du restaurant demande la gestion d'un utilisateur, 4. Le système affiche l'interface, <ol style="list-style-type: none"> 4.1. Le directeur demande de créer un utilisateur, <ol style="list-style-type: none"> 4.1.1. Le système affiche le formulaire, 4.1.2. Le directeur saisit les données puis valide, 4.1.3. Le système affiche un message de confirmation, 4.2. Le directeur sélectionne un utilisateur et demande de le modifier, <ol style="list-style-type: none"> 4.2.1. Le système affiche le formulaire, 4.2.2. Le directeur ressaisi les données à modifier puis il valide, 4.2.3. Le Système affiche l'utilisateur modifié, 4.3. Le directeur sélectionne un utilisateur et demande de le supprimer, <ol style="list-style-type: none"> 4.3.1. Le directeur valide, 4.3.2. Le système affiche un message de confirmation. 	
ENCHAINEMENT ALTERNATIF	
Aucun	

TABLE 2.8 – Description du cas d'utilisation gérer un utilisateur.

Cas d'utilisation gérer une salle

Sommaire d'identification	
Titre :	Gérer une salle
But	Le directeur du restaurant peut ajouter ou modifier ou supprimer les salles du système et le système met à jour la base de données.
Acteur	Directeur du restaurant
Description des enchainements	
Pré conditions	
Le directeur du restaurant doit s'authentifier.	
SCENARIO NOMINAL	
<ol style="list-style-type: none"> 1. Authentification, 2. Accès au système, 3. le directeur du restaurant demande la gestion des salles, 4. Le système affiche l'interface, <ol style="list-style-type: none"> 4.1. Le directeur demande de créer une salle, <ol style="list-style-type: none"> 4.1.1. Le système affiche le formulaire, 4.1.2. Le directeur saisit les données puis valide, 4.1.3. Le système affiche un message de confirmation, 4.2. Le directeur sélectionne une salle et demande de la modifier, <ol style="list-style-type: none"> 4.2.1. Le système affiche le formulaire, 4.2.2. Le directeur ressaisi les données à modifier puis il valide, 4.2.3. Le Système affiche la salle modifiée, 4.3. Le directeur sélectionne une salle et demande de la supprimée, <ol style="list-style-type: none"> 4.3.1. Le directeur valide, 4.3.2. Le système affiche un message de confirmation. 	
ENCHAINEMENT ALTERNATIF	
Aucun	

TABLE 2.9 – Description du cas d'utilisation gérer une salle .

Cas d'utilisation Gérer une table

Sommaire d'identification	
Titre :	Gérer une table
But	Le directeur du restaurant peut ajouter ou modifier ou supprimer les tables du système et le système met à jour la base de données.
Acteur	Directeur du restaurant
Description des enchainements	
Pré conditions	
Le directeur du restaurant doit s'authentifier.	
SCENARIO NOMINAL	
<ol style="list-style-type: none"> 1. Authentification, 2. Accès au système, 3. le directeur du restaurant demande la gestion des tables, 4. Le système affiche l'interface, <ol style="list-style-type: none"> 4.1. Le directeur demande de créer une table, <ol style="list-style-type: none"> 4.1.1. Le système affiche le formulaire, 4.1.2. Le directeur saisit les données puis valide, 4.1.3. Le système affiche un message de confirmation, 4.2. Le directeur sélectionne une table et demande de la modifier, <ol style="list-style-type: none"> 4.2.1. Le système affiche le formulaire, 4.2.2. Le directeur ressaisi les données à modifier puis il valide, 4.2.3. Le Système affiche la table modifiée, 4.3. Le directeur sélectionne une table et demande de la supprimée <ol style="list-style-type: none"> 4.3.1. Le directeur valide, 4.3.2. Le système affiche un message de confirmation. 	
ENCHAINEMENT ALTERNATIF	
Aucun	

TABLE 2.10 – Description du cas d'utilisation gérer une table.

Cas d'utilisation Gérer une catégorie de consommation

Sommaire d'identification	
Titre :	Gérer une catégorie de consommation
But	Le directeur du restaurant peut ajouter ou modifier ou supprimer les catégories de consommation du système et le système met à jour la base de données.
Acteur	Directeur du restaurant
Description des enchainements	
Pré conditions	
Le directeur du restaurant doit s'authentifier.	
SCENARIO NOMINAL	
<ol style="list-style-type: none"> 1. Authentification, 2. Accès au système, 3. le directeur du restaurant demande la gestion des catégories de consommation 4. Le système affiche l'interface, <ol style="list-style-type: none"> 4.1. Le directeur demande de créer une catégorie de consommation, <ol style="list-style-type: none"> 4.1.1. Le système affiche le formulaire, 4.1.2. Le directeur saisit les données puis valide, 4.1.3. Le système affiche un message de confirmation, 4.2. Le directeur sélectionne une catégorie et demande de la modifier, <ol style="list-style-type: none"> 4.2.1. Le système affiche le formulaire, 4.2.2. Le directeur ressaisi les données à modifier puis il valide, 4.2.3. Le Système affiche la catégorie de consommation modifiée, 4.3. Le directeur sélectionne une catégorie de consommation et demande de la supprimée <ol style="list-style-type: none"> 4.3.1. Le directeur valide, 4.3.2. Le système affiche un message de confirmation. 	
ENCHAINEMENT ALTERNATIF	
Aucun	

TABLE 2.11 – Description du cas d'utilisation gérer une catégorie de consommation.

Cas d'utilisation Gérer une consommation

Sommaire d'identification	
Titre :	Gérer une consommation
But	Le directeur du restaurant peut ajouter ou modifier ou supprimer les consommations du système et le système met à jour la base de données.
Acteur	Directeur du restaurant
Description des enchainements	
Pré conditions	
Le directeur du restaurant doit s'authentifier.	
SCENARIO NOMINAL	
<ol style="list-style-type: none"> 1. Authentification, 2. Accès au système, 3. le directeur du restaurant demande la gestion des consommations, 4. Le système affiche l'interface, <ol style="list-style-type: none"> 4.1. Le directeur demande de créer consommation, <ol style="list-style-type: none"> 4.1.1. Le système affiche le formulaire, 4.1.2. Le directeur saisit les données puis valide, 4.1.3. Le système affiche un message de confirmation, 4.2. Le directeur sélectionne une consommation et demande de la modifier <ol style="list-style-type: none"> 4.2.1. Le système affiche le formulaire, 4.2.2. Le directeur ressaisi les données à modifier puis il valide, 4.2.3. Le Système affiche la consommation modifiée, 4.3. Le directeur sélectionne une consommation et demande de la supprimée <ol style="list-style-type: none"> 4.3.1. Le directeur valide, 4.3.2. Le système affiche un message de confirmation. 	
ENCHAINEMENT ALTERNATIF	
Aucun	

TABLE 2.12 – Description du cas d'utilisation gérer une consommation.

Cas d'utilisation Elaborer un plan des tables

Sommaire d'identification	
Titre :	Elaborer un plan des tables
But	Le directeur du restaurant peut créer un plan des tables de chaque salle du restaurant
Acteur	Directeur du restaurant
Description des enchainements	
Pré conditions	
Le directeur du restaurant doit s'authentifier.	
SCENARIO NOMINAL	
<ol style="list-style-type: none"> 1. Authentification, 2. Accès au système, 3. le directeur du restaurant demande de créer un plan des tables, 4. Le système affiche l'interface, 5. Sélectionner une salle, 6. Sélectionner et insérer les éléments de la salle, 7. Le système confirme la création. 	
ENCHAINEMENT ALTERNATIF	
Aucun	

TABLE 2.13 – Description du cas d'utilisation élaborer un plan des tables.

2.3.2.3 Diagramme de cas d'utilisation associé au serveur

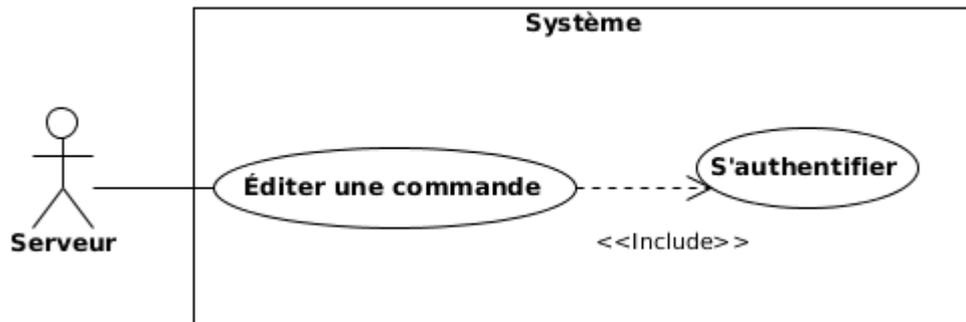


FIGURE 2.5 – Diagramme de cas d'utilisation au serveur.

Cas d'utilisation éditer une commande

Sommaire d'identification	
Titre :	Editer une commande
But	Le serveur peut créer une commande et la lancer
Acteur	serveur
Description des enchainements	
Pré conditions	
Le serveur doit s'authentifier.	
SCENARIO NOMINAL	
<ol style="list-style-type: none"> 1. Authentification, 2. Accès au système, 3. le serveur demande de créer une commande, 4. Le système affiche l'interface, 5. Sélectionner une salle et une table, 6. Sélectionner et insérer les éléments de la commande, 7. Valider et lancer, 8. Le système confirme et lance la commande 	
ENCHAINEMENT ALTERNATIF	
Aucun	

TABLE 2.14 – Description du cas d'utilisation éditer une commande.

2.4 Capture des besoins non fonctionnels

Les besoins non-fonctionnels sont des contraintes qui agissent sur le système, pour l'application que nous allons réaliser, les besoins non fonctionnels sont les suivant :

a) L'ergonomie des interfaces :

-Le futur logiciel doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.

b) Sécurité :

-Le système doit être capable de certifier la sécurité des données : par l'authentification de chaque utilisateur qui veut avoir l'accès au système.

c) Capacité :

-Le serveur d'application doit avoir une grande capacité de traitement qui lui permette de traités plusieurs requête des utilisateurs en même temps Afin de garantir un temps de réponse minimum

-Le système doit avoir une bande passante importante afin que les commandes des clients soit traitées à temps réel pour garantir une rapidité de service.

d) Rapidité et Fiabilité :

-L'accès à la base de données doit être souple et rapide.

-L'application doit être toujours fonctionnelle.

2.5 Conclusion

Dans ce chapitre nous avons spécifié les exigences des utilisateurs du système et cela à travers l'élaboration d'une étude préalable liée à l'existant, ce que nous a permis de distinguer les multiples cas d'utilisation, de cerner et de définir les périmètres du système à réaliser.

Le prochain chapitre sera consacré à l'analyse et la conception du système.

CHAPITRE 3

ANALYSE ET CONCEPTION

3.1 Introduction

Dans le cycle de vie d'un système informatique, il existe deux étapes primordiales qui sont l'analyse et la conception.

En effectuant l'analyse nous examinerons les données existantes, relative à l'étude préalable faite précédemment et nous étudierons les traitements à effectuer et cela en ayant recours à quelque techniques de modélisation.

Ainsi que la définition du futur système informatique et les différentes notions qui nous permettrons la conception, l'écriture et sa mise au point, qui seront traitées dans la conception.

3.2 Analyse

La phase d'analyse a pour objectif de décrire de manière précise, concise, correcte et compréhensible les besoins et les exigences du client. Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structures sous une forme qui facilite la compréhension (scénario). L'activité d'analyse est la spécification de ce que devra faire le système à réaliser, en déterminant ses éléments et leurs interactions en les représentant au moyen d'un diagramme de séquence.

3.2.1 Présentation du diagramme de séquence

L'objectif du diagramme de séquence est de représenter les interactions entre les objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés [3].

3.2.1.1 Eléments de base d'un diagramme de séquence

Lors de la représentation d'un diagramme de séquence, plusieurs concepts sont utilisés :

a) **Formalisme d'un diagramme de séquence**

Un diagramme de séquence est représenté globalement dans un grand rectangle avec indication du nom du diagramme en haut à gauche (figure III.1). Il est à noter que "sd" est l'abréviation de "Sequence Diagram" [3].

b) **Objet**

Un objet est une instance d'une classe et il est représenté par un rectangle dans le quel le nom de la classe est écrit et souligné [6].

c) **Ligne de vie**

Une ligne de vie représente l'ensemble des opérations exécutées par un objet, elle se présente par un rectangle qui est accroché à une ligne verticale pointillée [3].

d) **Message**

Un message est un élément de communication unidirectionnel entre objets qui déclenche une activité dans l'objet destinataire. la réception d'un message provoque un événement dans l'objet récepteur. La flèche pointillée représente un retour au sens UML. Cela signifie que le message en question est le résultat direct du message précédent [5].

Dans un diagramme de séquence, deux types de message peuvent être distingués [3] :

Message asynchrone

L'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. Un message asynchrone se représente par une flèche avec une extrémité non pleine.

Message synchrone

L'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. Un message synchrone se représente par une flèche en trait pleine et à l'extrémité pleine. Ce message peut être suivi d'une réponse qui se représente par une flèche en pointillé.

d) **Fragment d'interaction**

Un fragment d'interaction correspond à un ensemble d'interaction auquel on applique un opérateur. Un fragment combiné se représente globalement comme un diagramme de séquence avec indication dans le coin à gauche du nom de l'opérateur.

Il existe treize opérateurs définis dans la notation UML 2.0 : alt, opt, loop, par, strict/weak, break, ignore/consider, critical, negative, assertion et ref [3].

Nous décrivons dans ce qui suit les opérateurs que nous allons utiliser pour l'analyse de notre projet : alt, opt, loop et ref [3].

a) **Opérateur alt** :correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il permet aussi d'utiliser les clauses de type sinon et se représente dans un fragment possédant au moins deux parties séparées par des pointillés.

b) **Opérateur loop** :un fragment combiné de type loop peut s'exécuter plusieurs fois, et la condition de garde explicite l'itération.

b) **Opérateur opt** :correspond à une instruction de test sans alternative (sinon). Il se représente dans un fragment possédant une seule partie.

d) **Opérateur ref** :permet d'appeler une séquence d'interactions décrite par ailleurs constituant ainsi une sorte de sous-diagramme de séquence.

3.2.1.2 Elaboration du digramme de séquence

En se basant sur les éléments décrits précédemment, nous présentons les diagrammes de séquence des cas d'utilisations de l'application à réaliser.

a) Diagramme de séquence du cas d'utilisation s'authentifier

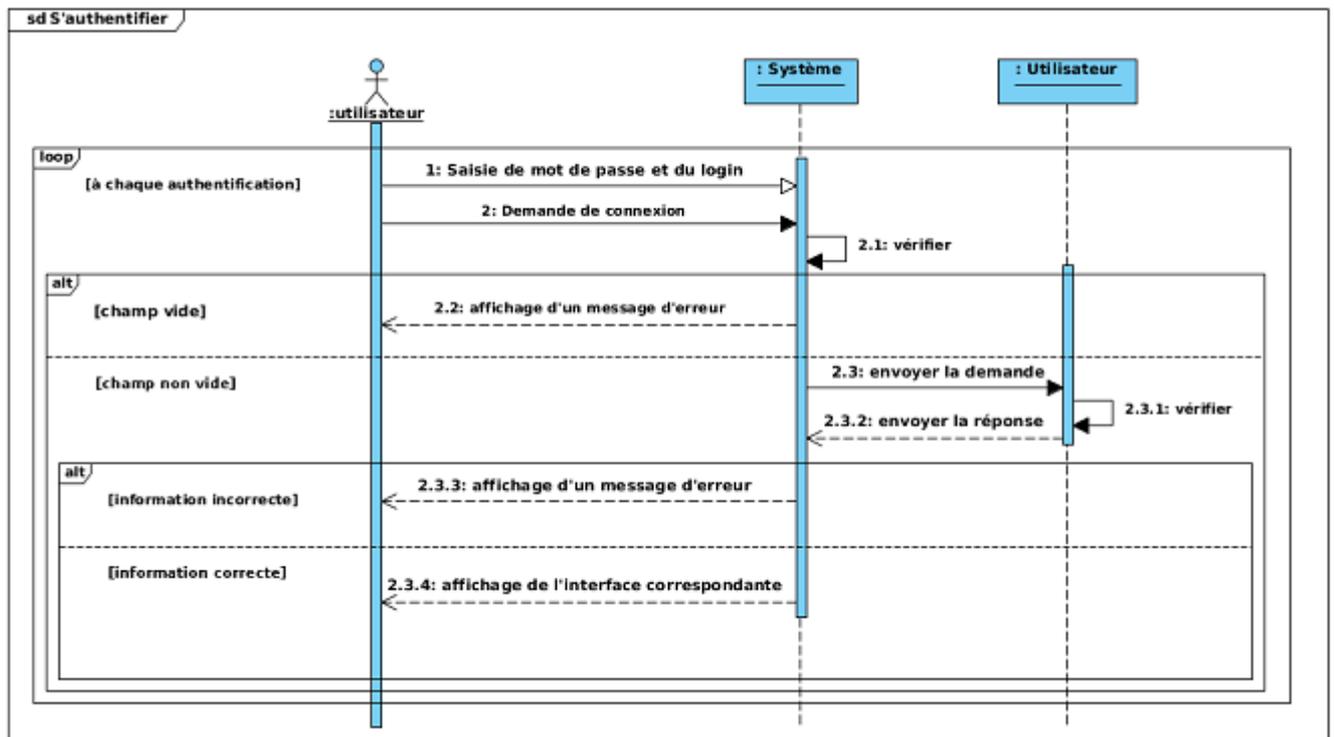


FIGURE 3.1 – Diagramme de séquence du cas d'utilisation s'authentifier.

a) Diagramme de séquence du cas d'utilisation changer le mot de passe

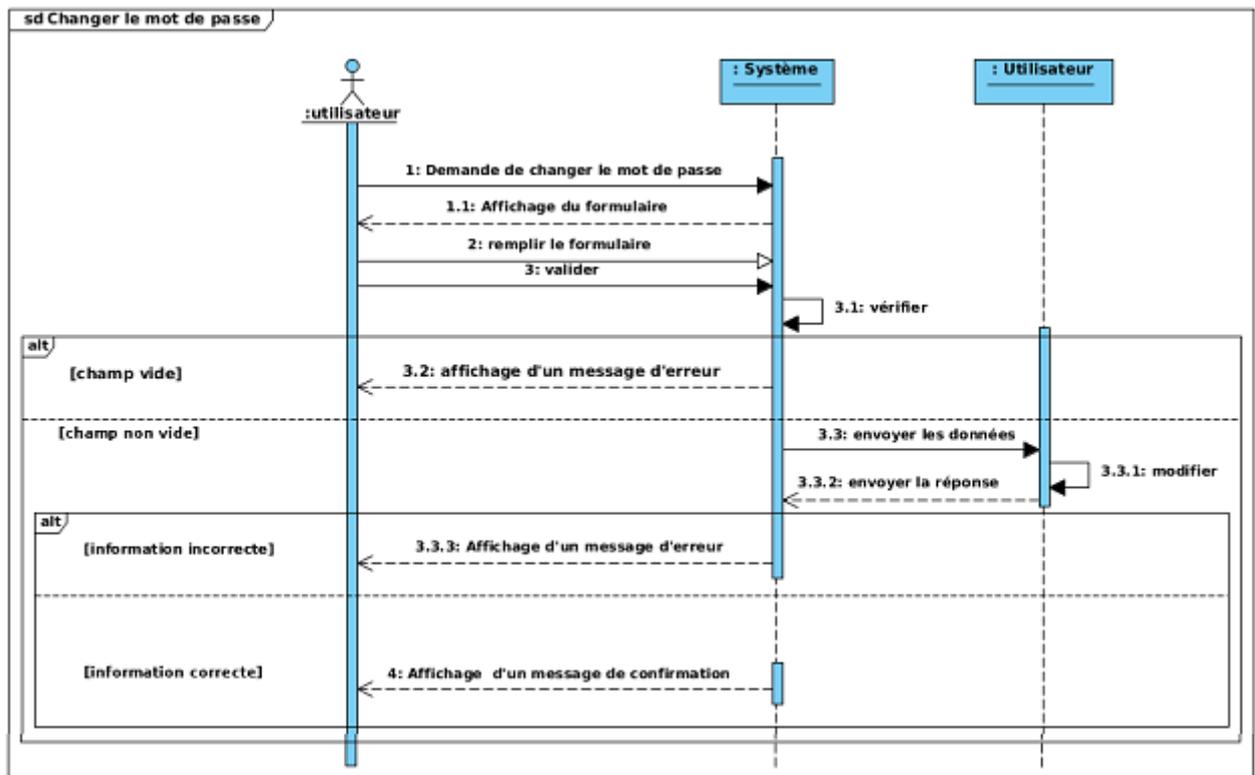


FIGURE 3.2 – Diagramme de séquence du cas d'utilisation changer le mot de passe

b) Diagramme de séquence du cas d'utilisation gérer une salle

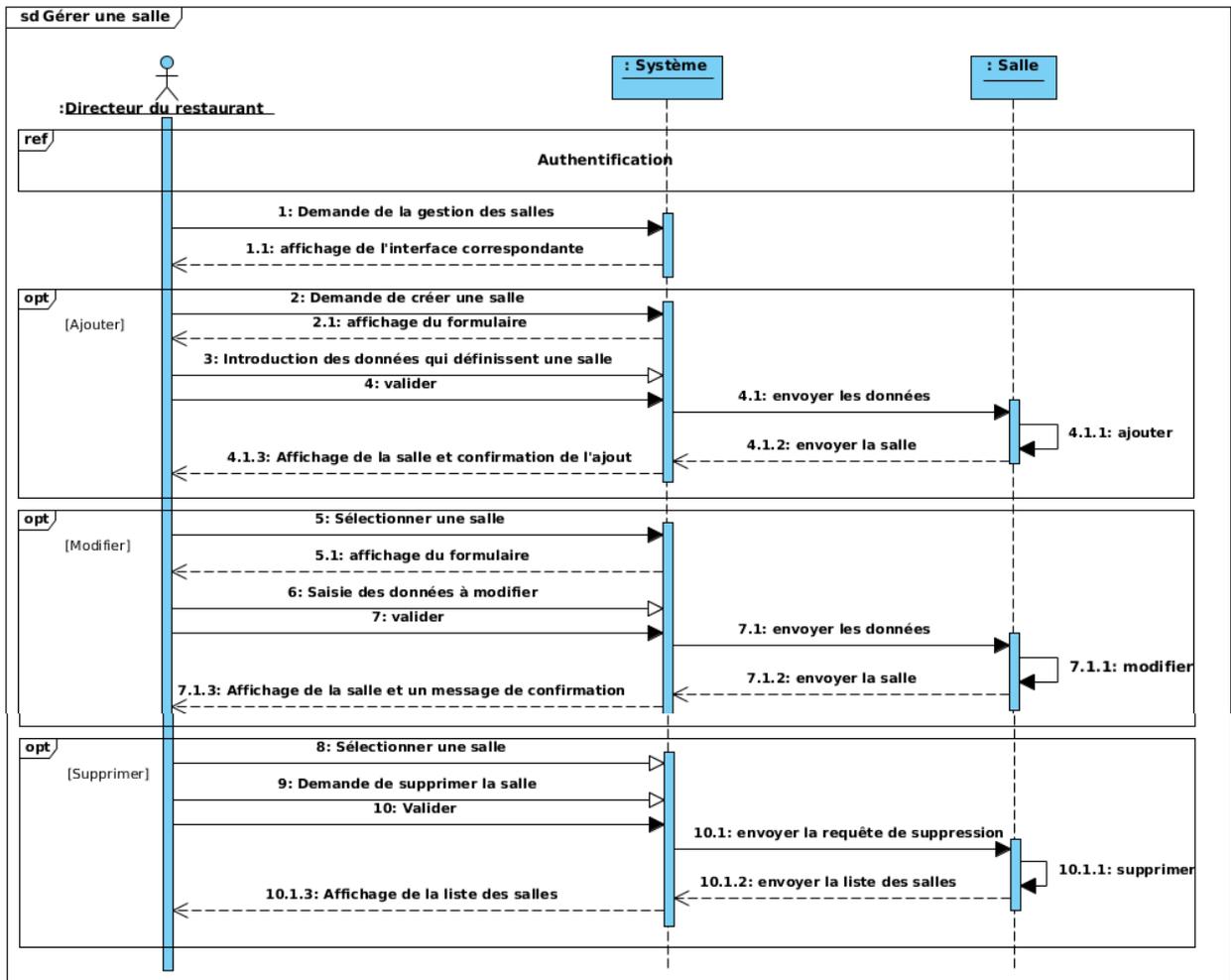


FIGURE 3.3 – Diagramme de séquence du cas d'utilisation gérer une salle.

c) Diagramme de séquence du cas d'utilisation gérer une table

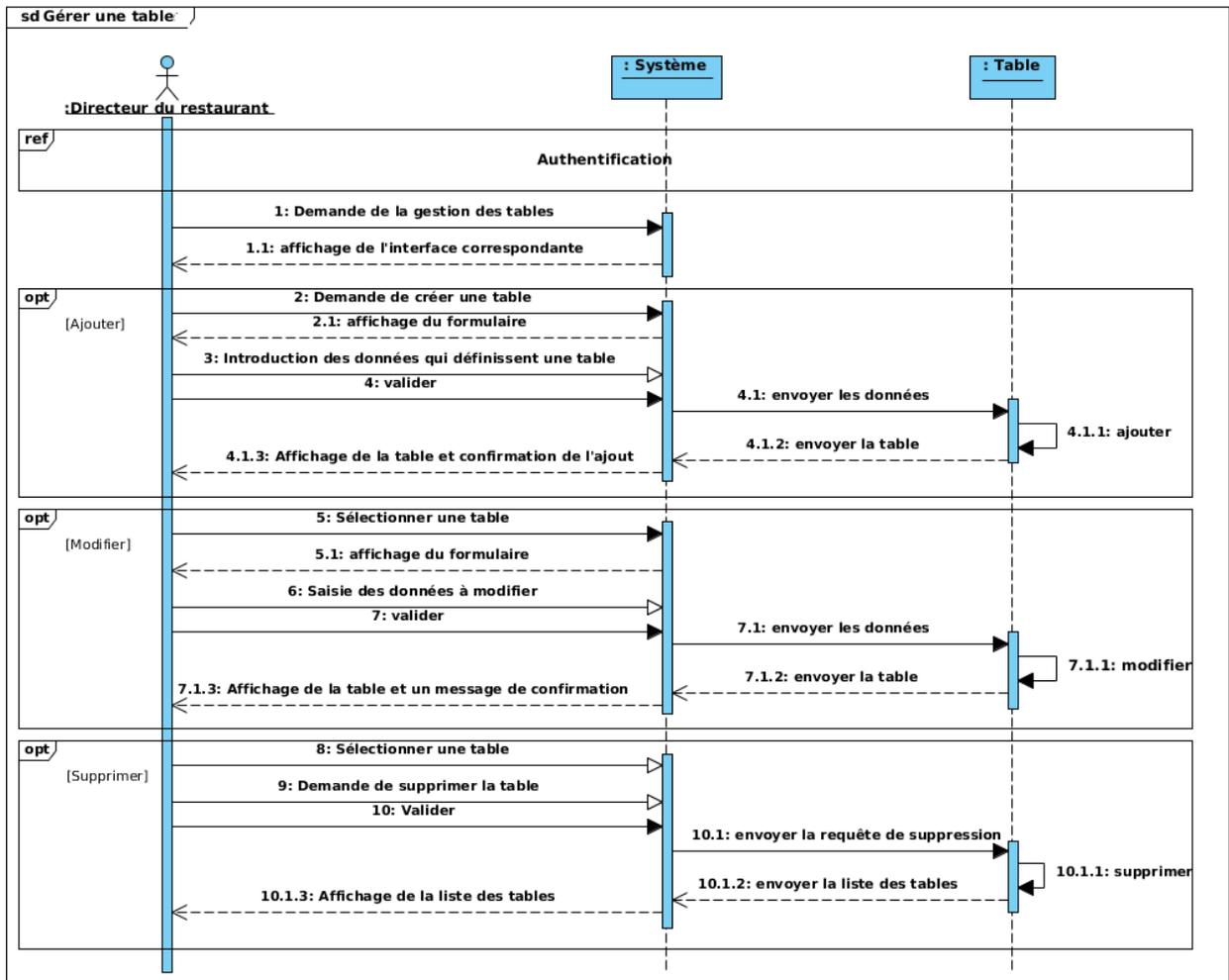


FIGURE 3.4 – Diagramme de séquence du cas d'utilisation gérer une table.

e) Diagramme de séquence du cas d'utilisation gérer une catégorie de consommation

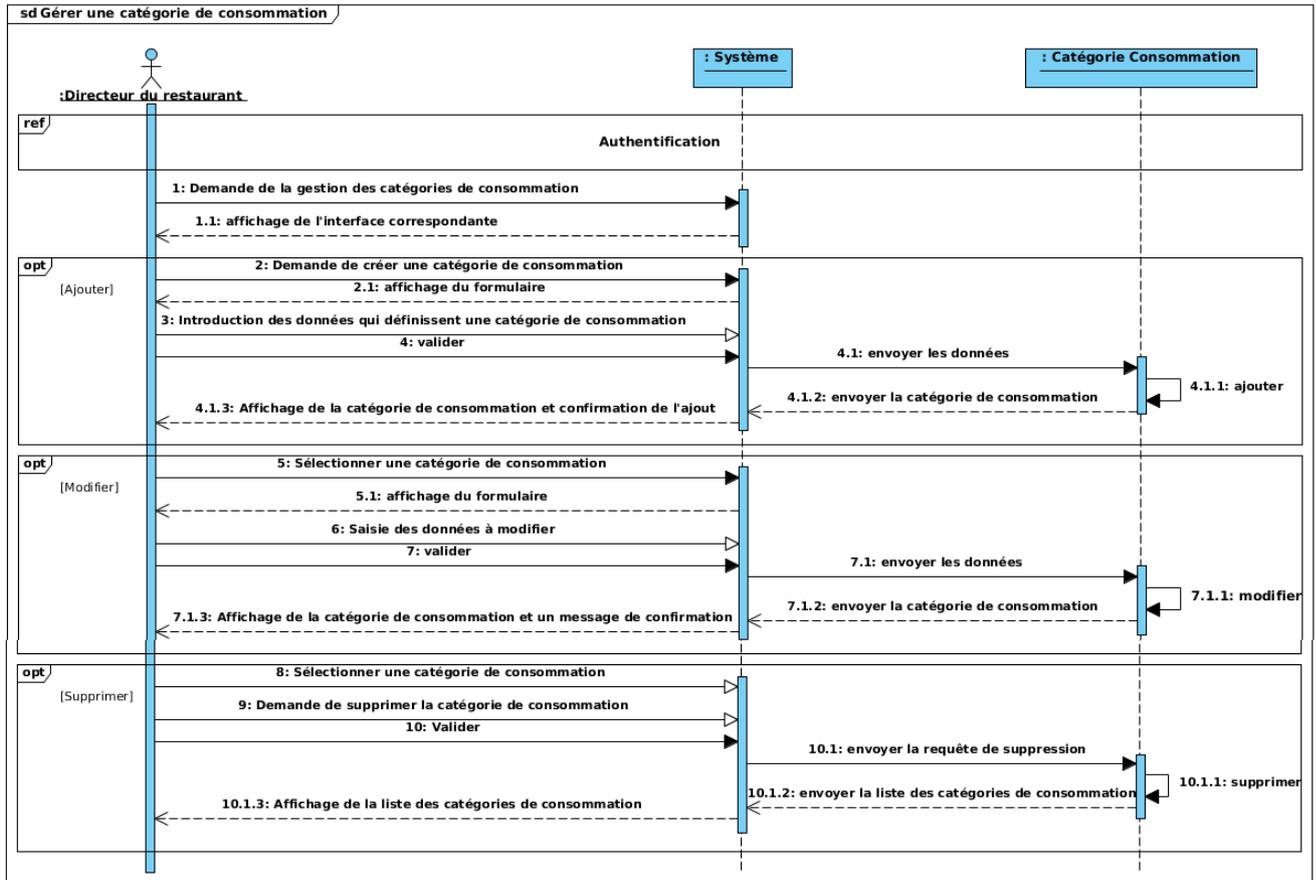


FIGURE 3.5 – Diagramme de séquence du cas d'utilisation gérer une catégorie de consommation

f) Diagramme de séquence du cas d'utilisation gérer une consommation

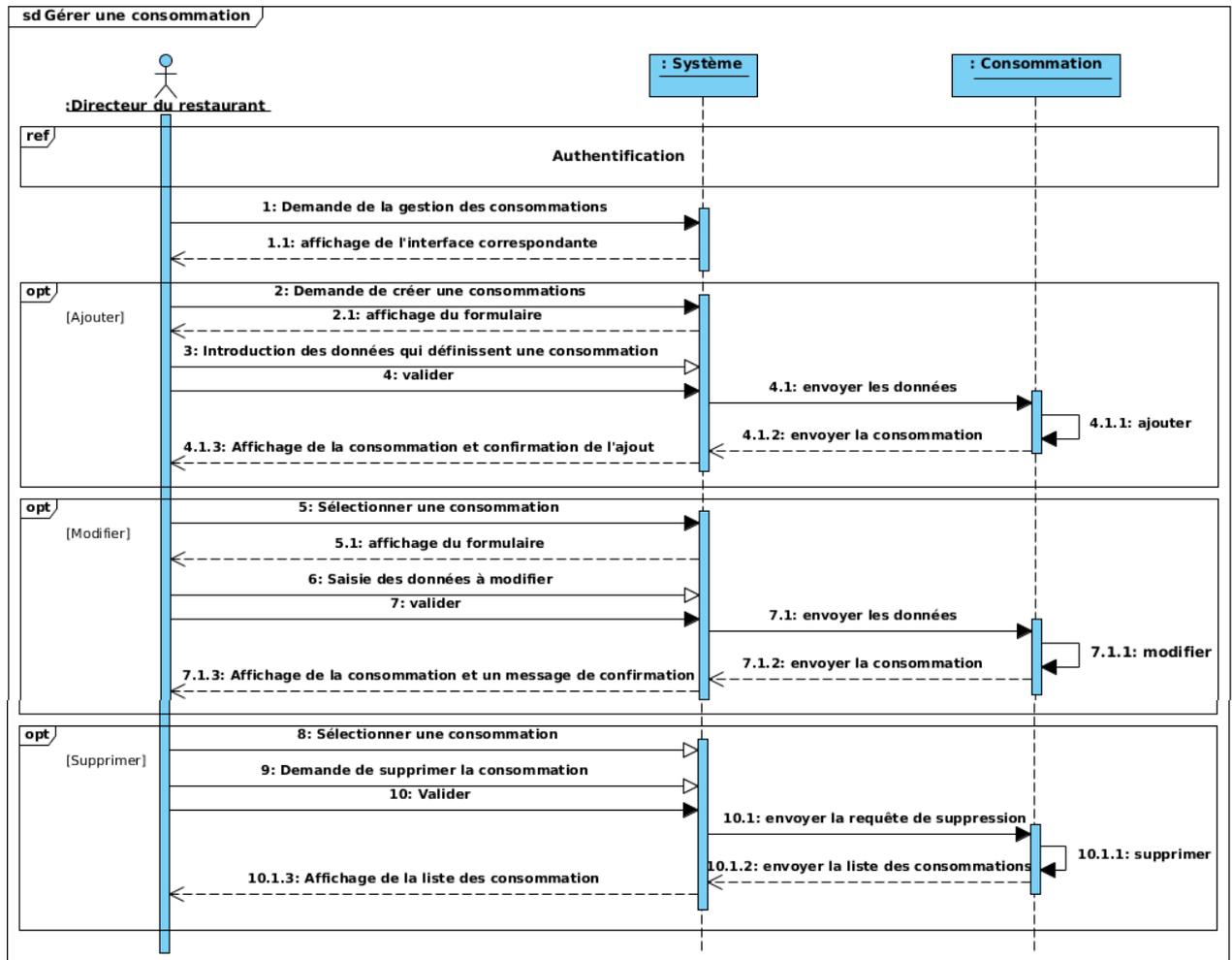


FIGURE 3.6 – Diagramme de séquence du cas d'utilisation gérer une consommations.

d) Diagramme de séquence du cas d'utilisation élaborer un plan des tables

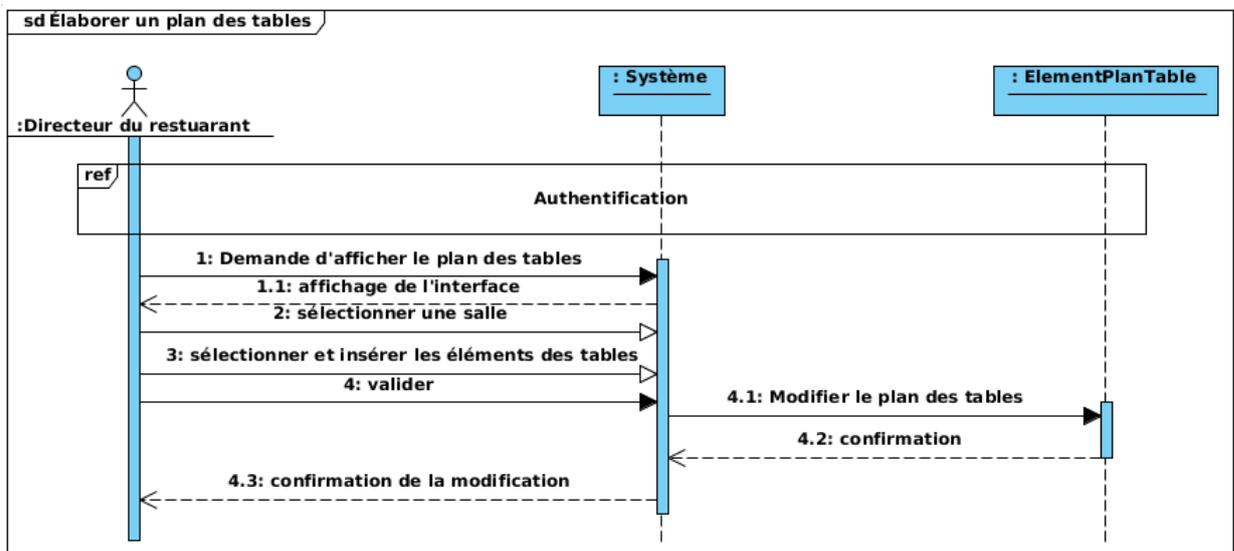


FIGURE 3.7 – Diagramme de séquence du cas d'utilisation élaborer un plan des tables.

d) Diagramme de séquence du cas d'utilisation gérer un utilisateur

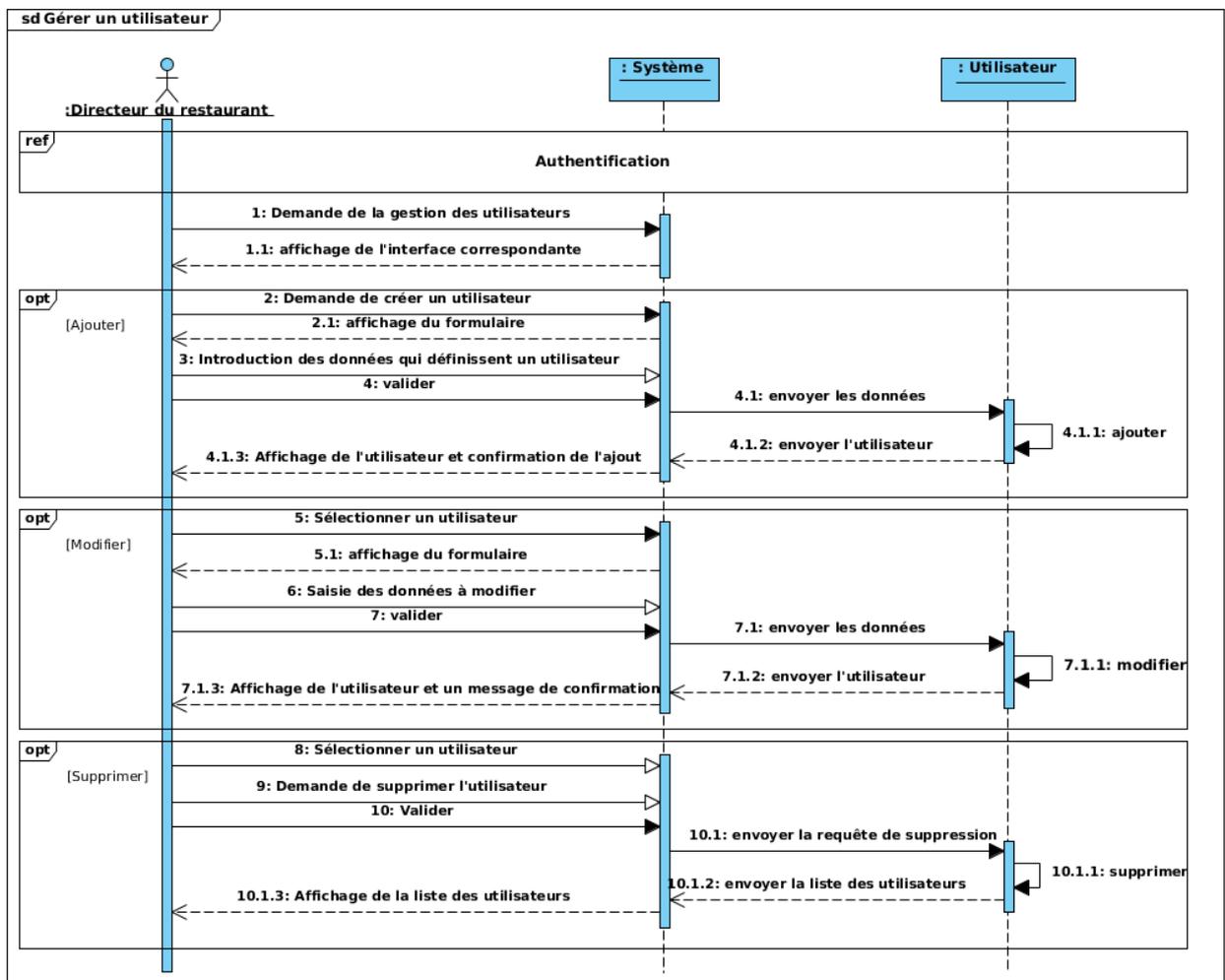


FIGURE 3.8 – Diagramme de séquence du cas d'utilisation gérer un utilisateur .

d) Diagramme de séquence du cas d'utilisation gérer un rôle

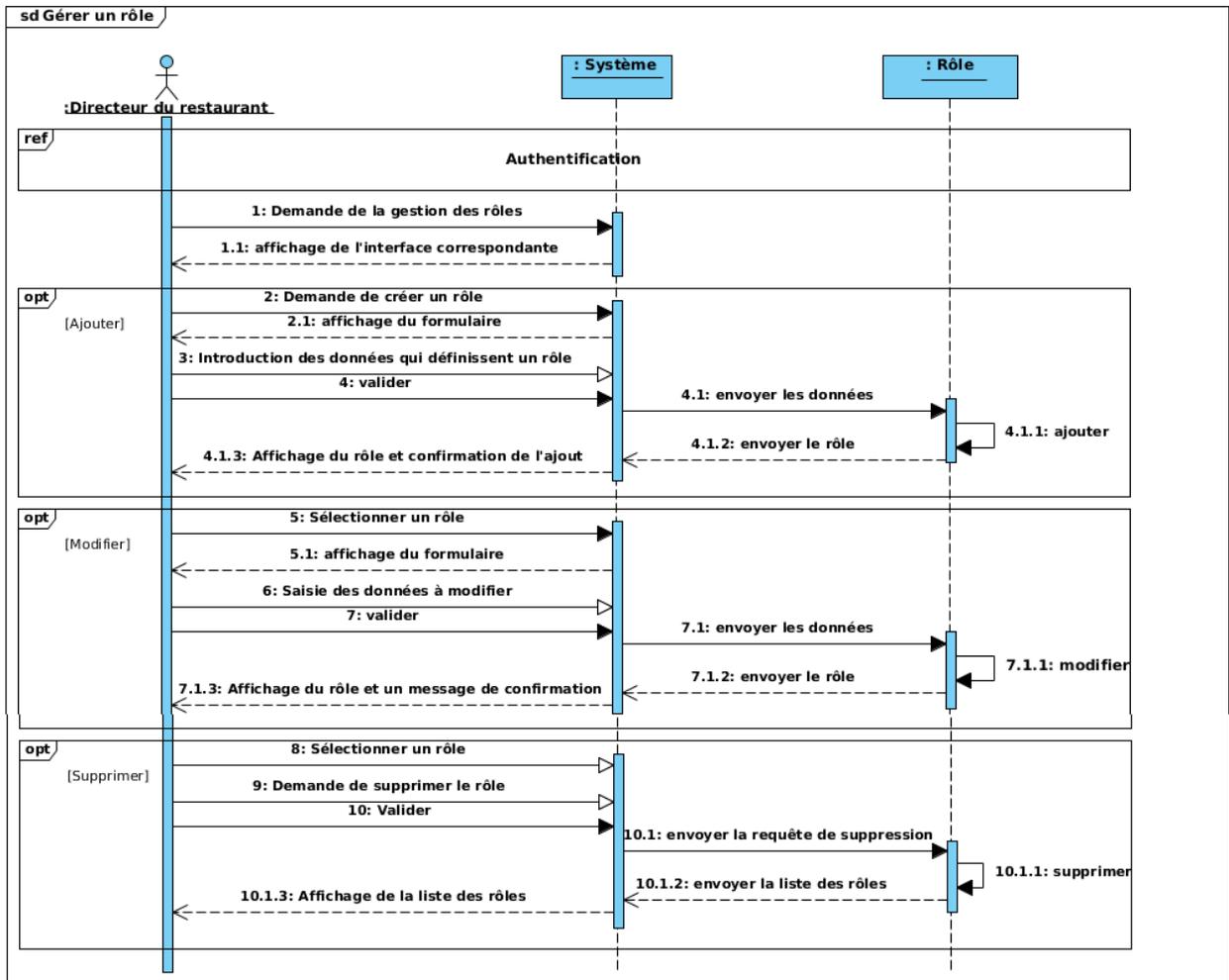


FIGURE 3.9 – Diagramme de séquence du cas d'utilisation gérer un rôle.

d) Diagramme de séquence du cas d'utilisation éditer une commande

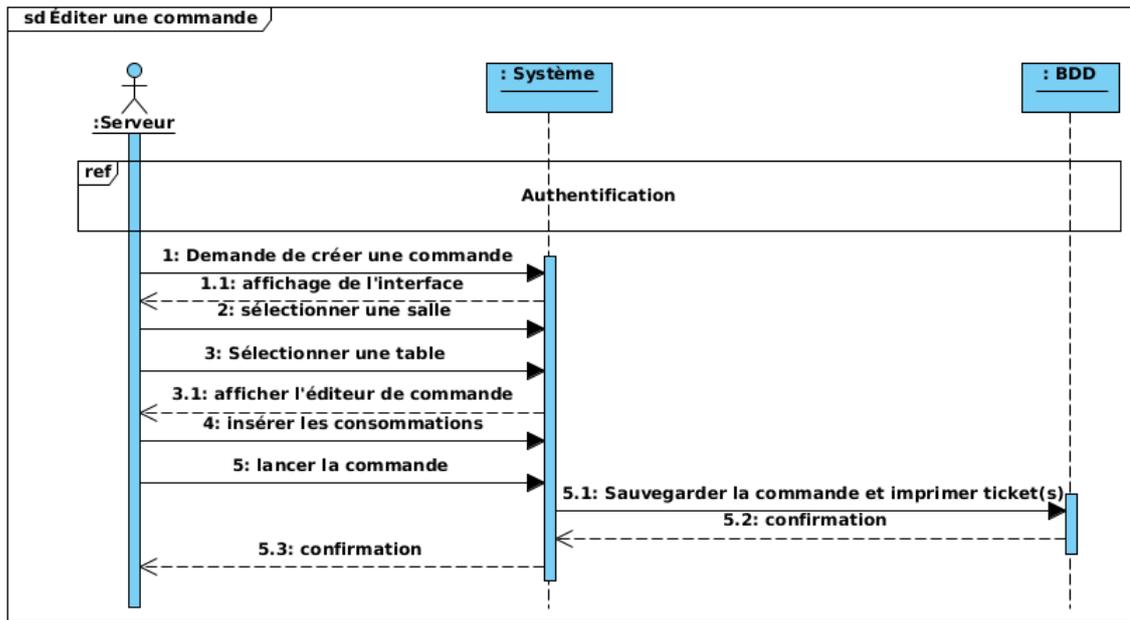


FIGURE 3.10 – Diagramme de séquence du cas d'utilisation éditer une commande .

3.3 Conception

Dans cette partie, nous allons présenter l'aspect statique de notre système, pour cela nous aurons recours aux diagrammes de déploiement et classe offert par uml qui permettent de présenter la configuration physique et logiciel du système, et ainsi de créer un modèle objet qui permettra le passage au modèle relationnel afin de définir les tables de base de données du système.

3.3.1 Présentation de diagramme de classe

Le diagramme de classes est le point central dans un développement orienté objet. En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet ou, à un niveau de détail plus important [1].

3.3.1.1 Eléments de base d'un diagramme de classe

Lors de la représentation d'un diagramme de classe, plusieurs concepts sont utilisés :

a)Classe

Une classe représente une entité, un groupe d'objets, qui partagent des propriétés (attributs) et un comportement (opérations) communes. Une classe est instanciée pour créer un objet en particulier [5].

Une classe est représentée par un rectangle séparé en trois parties qui sont :

Nom d'une classe : doit évoquer le concept décrit par la classe.

Attributs : des données déclarées au niveau d'une classe, éventuellement typée, à laquelle chacun des objets de cette classe donne une valeur [5].

Opération : une fonction applicable aux objets d'une classe. Une opération permet de décrire le comportement d'un objet [3].

b)Relation entre classe

Une relation est une liaison entre éléments, il existe plusieurs relations parmi eux on cite [3] :

Association : une relation sémantique durable entre deux classes, qui décrit un ensemble de liens entre instances.

Classes-association : permet de décrire soit des attributs soit des opérations propres à l'association.

Agrégation : une association qui permet de représenter un lien de type " ensemble " comprenant des " éléments ". Il s'agit d'une relation entre une classe représentant le niveau " ensemble " et 1 à n classes de niveau " éléments ". **Composition** : 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 " .

Généralisation : 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 " .

Héritage permet à une sous-classe de disposer des attributs et opérations de la classe dont elle dépend.

3.3.1.2 Règle de passage

Pour élaborer les multiplicités du diagramme de classe, les règles de gestion suivantes doivent être vérifiées :

-Une consommation ne peut appartenir ou bien appartenir à une et une seule catégorie de consommation et chaque catégorie de consommation peut avoir une ou plusieurs consommations.

-Une table appartient à une et une seule salle et chaque salle peut avoir une ou plusieurs tables.

-Une consommation ne peut avoir ou avoir une et une seule photo et une photo appartient à une et une seule consommation.

-une catégorie de consommation ne peut appartenir ou bien appartenir à une et une seule catégorie de consommation et chaque catégorie de consommation peut avoir une ou plusieurs catégories de consommation.

-une catégorie de consommation ne peut avoir ou bien avoir une et une seule photo et une photo appartient à une et une seule catégorie de consommation.

-une table ne peut avoir ou bien avoir une et une seule photo et une photo appartient à une ou plusieurs tables.

-une salle est composé d'un ou plusieurs element de plan table et chaque element de plan table est un composant de la salle.

-Un utilisateur ne peut avoir ou bien avoir une et une seule photo et une photo appartient à un et un seule utilisateur.

-un rôle est affecté à un ou plusieurs utlisateurs et chaque utilisateur peut être affecter à un et un seul rôle.

-une consommation est associée à une et une seule imprimante et chaque imprimante ne peut ou bien être associer à plusieurs consommation.

-un element graphique ne peut représenter ou bien représenter une photo et chaque photo représente un et un seul élément graphique.

-une table ne peut représenter ou bien représenter un élément graphique et un élément graphique ne peut être une table.

-un element de plan table ne peut représenter ou bien représenter un ou plusieurs élément graphique et un élément graphique ne peut être un élément de plan table.

3.3.1.3 Elaboration du diagramme de classe

En se basant sur les éléments précédents, nous allons présenter le diagramme de classe de notre application :

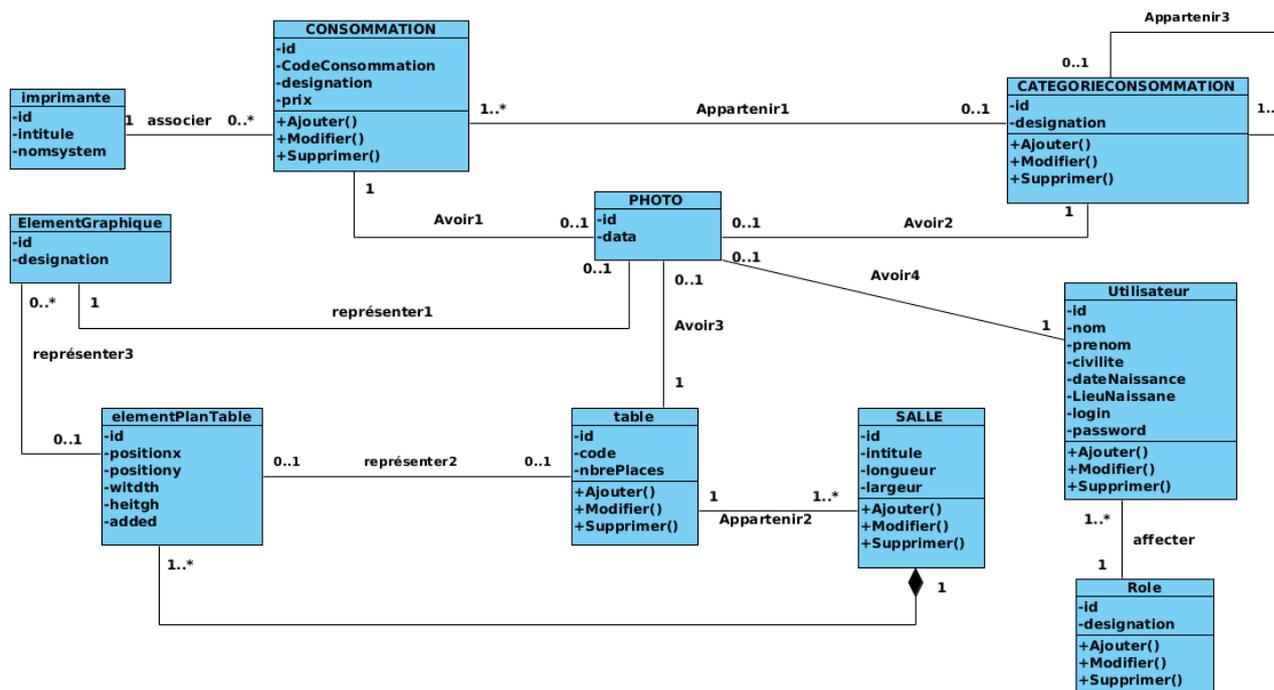


FIGURE 3.11 – Diagramme de classe.

3.3.1.4 Dictionnaire de données

Le dictionnaire de données présente la définition, le type (texte, numérique, etc) et le format (nombre de caractères, de décimales, etc) de l'ensemble des données gérées dans la base de données. Il se présente sous la forme d'un tableau.

L'objectif du dictionnaire de données est de permettre à toute personne découvrant la base de données de la reconstruire dans n'importe quel Système de gestion de Base de Données (SGBD) [w1].

Dans le tableau ci-dessous, nous illustrons le dictionnaire de données de la base de données de notre application :

<i>Nom de la classe</i>	<i>Nom d'attribut</i>	<i>Désignation de l'attribut</i>	<i>Type</i>	<i>Taille</i>
consommation	id	Identificateur de la consommation	int	11
	CodeConsommation	Code de la consommation	Varchar	50
	Designation	nom de la consommation	Varchar	100
	Prix	prix de la consommation	double	
CategorieConsommation	id	Identificateur de la catégorie de la consommation	int	11
	Designation	nom de la catégorie	varchar	50
photo	id	Identificateur de la photo	int	11
	Data	Contenu binaire de la photo	mediumblob	
table	id	Identificateur de la table	int	5
	Code	Code de la table	varchar	20
	nbrePlaces	Nombre de chaise de la table	int	11
salle	id	Identificateur de la salle	int	5
	Intitule	Nom de la salle	varchar	100
	Longueur	La longueur de la salle	double	
	Largeur	La largeur de la salle	double	
elementPlanTable	id	Id de l'élément de la salle	int	11
	Positionx	Position horizontale de l'élément dans la salle	int	11
	Positiony	Position verticale de l'élément dans la salle	int	11
	Width	La hauteur de l'élément	int	11
	Length	La largeur de l'élément	int	11
	Added	Élément ajouter au plan ou non	boolean	
ElementGraphique	id	Identificateur de l'élément	int	11
	Designation	Intitulé de l'élément	varchar	100
imprimante	id	Identificateur de l'imprimante	int	11
	Intitule	Nom significatif de l'imprimante	varchar	100
	Nomsysteme	Nom du système	varchar	100
Utilisateur	id	Identificateur de l'utilisateur	int	11
	nom	Nom de famille de l'utilisateur	varchar	50
	prenom	Prenom de l'utilisateur	varchar	50
	civilite	civilité	enum ('Mr', 'Mme')	
	dateNaissance	Date de naissance	date	
	LieuNaissance	Lieu de naissance	varchar	50
	login	Nom de l'utilisateur	varchar	50
	password	Mot de passe de l'utilisateur	varchar	50
Role	id	Identificateur du rôle	int	11
	designation	Le nom du rôle	varchar	50

TABLE 3.1 – Dictionnaire de données.

3.3.1.5 Passage au modèle relationnel

Nous avons identifié dans le diagramme de classes les objets persistants de notre système, à présent nous allons construire à partir de cette description un modèle relationnel qui nous servira de schéma structurant les données du système et il est basé sur les concepts suivants [4] :

Relation :est une table comportant des lignes et des colonnes.

Attribut :une colonne nommée de la table représentant la relation.

Tuple :est une ligne dans une table.

Clé primaire :est un ensemble minimal d'attribut qui permet d'identifier de manière unique chaque tuple dans une table.

Pour dériver un modèle relationnel à partir d'un modèle conceptuel exprimé à l'aide d'un diagramme de classe, il faut respecter certaines règles de passage et elles sont comme suit[4] :

Règle 1 : Transformation des classes

Chaque classe devient une relation. L'identifiant (respectivement les attributs) de la classe devient la clé primaire (respectivement des attributs) de la relation.

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.

Règle 3 : association réflexive un à plusieurs

L'association donne lieu à une table fille associé à la table mère dérivée de la seule entité donc il faut dériver une seule table qui contient une clé étrangère. Celle-ci est un double de sa clé primaire qui doit porter un nom différent.

Règle 4 : Association un-à-un

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

Règle 5 : Transformation de l'héritage

Il existe trois cas de représentation d'héritage :

1ère solution : elle consiste à utiliser une seule relation qui réunit les attributs de la super classe et celle des sous classes, cette solution accepte d'avoir des champs vides selon le cas de la classe choisie.

2ème solution : il faut transformer chaque sous-classe en une relation. La clé primaire de la sur-classe, migre dans la (les) relations(s) issue(s) de la (des) sous-classe(s) et devient à la fois clé primaire et clé étrangère.

3ème solution : même chose que la deuxième solution, en plus les attributs de la super classe sont dupliqués dans les relations des sous classes.

3.3.1.6 Elaboration du modèle relationnel

En appliquant les règles citées ci-dessus nous obtiendrons les tables relationnelles suivantes :

photo (id, data),

categorieConsommation (id, designation, #idPhoto, #idCategorieConsommation),

Consommation (id, CodeConsommation, designation, prix, #idPhoto, #IdCategorieConsommation, #idimprimante),

Salle (id, intitule, longueur, largeur),

table (id, code, nbrePlaces, #idPhoto, #idSalle),

elementPlanTable (id, positionx, positiony, width, length, added, #idPhoto, #idSalle, #idTable, #idelementgraphique).

ElementGraphique(id, designation, #idPhoto)

Imprimante(id, intitule, nomsystem)

Role(id, designation)

Utilisateur(id, nom, prenom, dateNaissance, LieuNaissance, civilite, login, password, #idPhoto, #idRole)

3.3.2 Présentation du diagramme de déploiement

Le diagramme de déploiement permet de décrire la disposition physique des ressources matérielles d'un système. Cette architecture comprend des noeuds correspondant aux supports physiques (ordinateurs, serveurs, routeurs, etc) ainsi que la répartition des artefacts logiciels (bibliothèques, exécutables, code source, etc) sur ces noeuds. C'est un véritable réseau constitué de noeuds et de connexions entre ces noeuds qui modélise cette architecture [3].

3.3.2.1 Eléments de base d'un diagramme de déploiement

Lors de la représentation d'un diagramme de déploiement, plusieurs concepts sont utilisés [3] :

a)Noeud : une ressource matérielle de traitement sur laquelle des artefacts seront mis en oeuvre pour l'exploitation du système. Les noeuds peuvent être interconnectés pour former un réseau d'éléments physique. Un noeud se représente par un cube comme suite :



FIGURE 3.12 – Présentation graphique d'un noeud.

b)Artéfact : est la spécification d'un élément physique qui est utilisé ou produit par le processus de développement du logiciel ou par le déploiement du système. C'est donc un élément concret comme par exemple : un fichier, un exécutable (.exe, .jar,...), une table d'une base de données ou un fichier source (.java, .c,...). Il se représente par un rectangle caractérisé par le mot-clé " artifact "comme suite :



FIGURE 3.13 – Présentation graphique d'un artéfact.

c)chemin de communication : est une association entre deux noeuds au travers de laquelle les noeuds peuvent communiquer par l'échange de messages et de signaux.



FIGURE 3.14 – Présentation du chemin de communication entre des nœuds.

3.3.2.2 Elaboration de diagramme de déploiement

En se basant sur les éléments précédents, nous allons présenter le diagramme de déploiement de notre application :

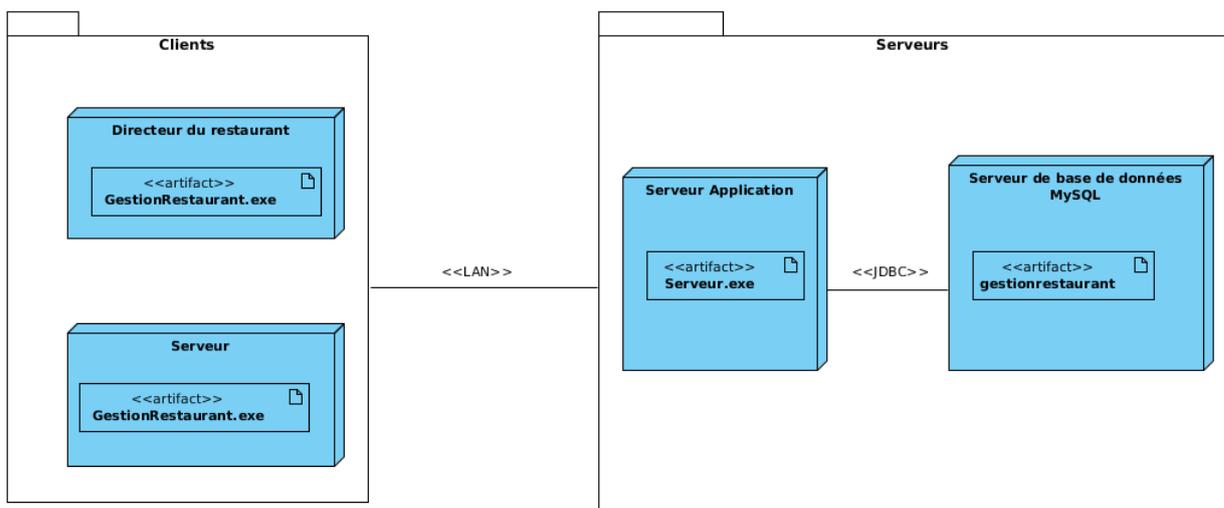


FIGURE 3.15 – Diagramme de déploiement de l'application à réaliser.

3.4 Conclusion

Dans ce chapitre nous avons détaillé deux phases essentielles dans l'élaboration d'un système informatique qui concernent l'analyse et la conception, ou nous avons recours à différents diagrammes UML pour associer la modélisation adéquate à chacune d'elles.

Le chapitre suivant sera dédié à la réalisation où nous exposerons certains détails concernant l'implémentation du système.

CHAPITRE 4

RÉALISATION

4.1 Introduction

Dans la réalisation de notre application réseau nous avons fait appel à un ensemble d'outils et langages de développement. Dans ce qui suit nous tenons à présenter nos choix. Et enfin, nous présenterons les différentes fonctionnalités offertes par le système impliquant différents utilisateurs. Cela sera illustré par la présentation des différentes interfaces utilisateurs.

4.2 Les outils d'implémentation d'une base de données

Pour l'implémentation de la base de données de notre application, nous avons choisi comme outils : textbfle serveur MySQL, textbfle gestionnaire phpMyAdmin

4.2.1 Présentation du MySQL-Server

MySQL (My Structed Query Language) est un gestionnaire de base de données SQL. Il s'agit d'un dispositif capable de créer et de gérer une base de données par l'intermédiaire de fonction et d'instruction SQL. Ces instructions sont écrites dans un langage scripte, appelé PHP. PHP est employé à l'intérieur des pages HTML, afin de réaliser des actions/réponses vers une base de données [8].

4.2.2 Présentaion du phpMyAdmin

PhpMyAdmin est une interface conviviale qui permet de gérer très facilement une base de données, sans nécessiter une connaissance avancée des requêtes SQL. Le fait que l'interface soit développée en PHP la rend parfaitement adaptée à l'utilisation conjointe d'une base

MySQL et d'un moteur de scripts PHP.

Il permet d'effectuer différentes tâches telles que la création, la modification ou la suppression de bases de données , des tables , des champs ou des lignes, exécution des requêtes SQL , ou la gestion des utilisateurs et des permissions[9].

4.3 Les outils de développement de l'application

Pour notre application nous avons choisi les outils suivants : java, environnement Eclipse.

4.3.1 Présentation du langage de programmation java

Puisque notre travail est basé sur la méthode objet, nous avons choisi le langage java comme langage de développement.

Java est issu des langages C et C++.c'est pourquoi sa syntaxe et sa structure orienté objet ressemblent à celles de C++.

Java permet de développer des applications autonomes mais aussi, et surtout, des applications client/serveur. Il est doté en standard d'une riche bibliothèque de classes, comprenant la gestion des interfaces graphiques (fenêtre, boites de dialogue, contrôles, menus) [7].

4.3.2 Présentation de l'environnement Eclipse

Le langage java nécessite forcément un environnement de développement dans lequel on désire développer une application, pour cela nous avons choisi l'environnement Eclipse pour réaliser notre application.

Eclipse est un IDE, Integrated Development Environment (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation.

Au fur et à mesure que vous programmez, eclipse compile automatiquement le code que vous écrivez, en soulignant en rouge ou jaune les problème qu'il décèle. Il souligne en rouge les parties du programme qui ne compilent pas, et en jaune les parties qui compilent [w2].

4.4 Représentation des interface de l'application

Dans cette partie nous présenterons l'application réalisée à travers des captures d'écrans de quelques interfaces qui illustrent les principales fonctionnalités du système.

4.4.1 Interface authentification

Si l'application est lancée pour la première fois, l'interface ci-dessous se présente aux utilisateurs. Elle lui offre deux possibilités : d'accéder à son interface correspondante par l'introduction de son mot de passe et son login ou la possibilité de changer son mot de passe.



FIGURE 4.1 – Interface authentification.

→ interface "changer le mot de passe"



FIGURE 4.2 – Interface changer le mot de passe.

4.4.2 Interface Principale



FIGURE 4.3 – Interface principale.

4.4.3 Interface gestion de table

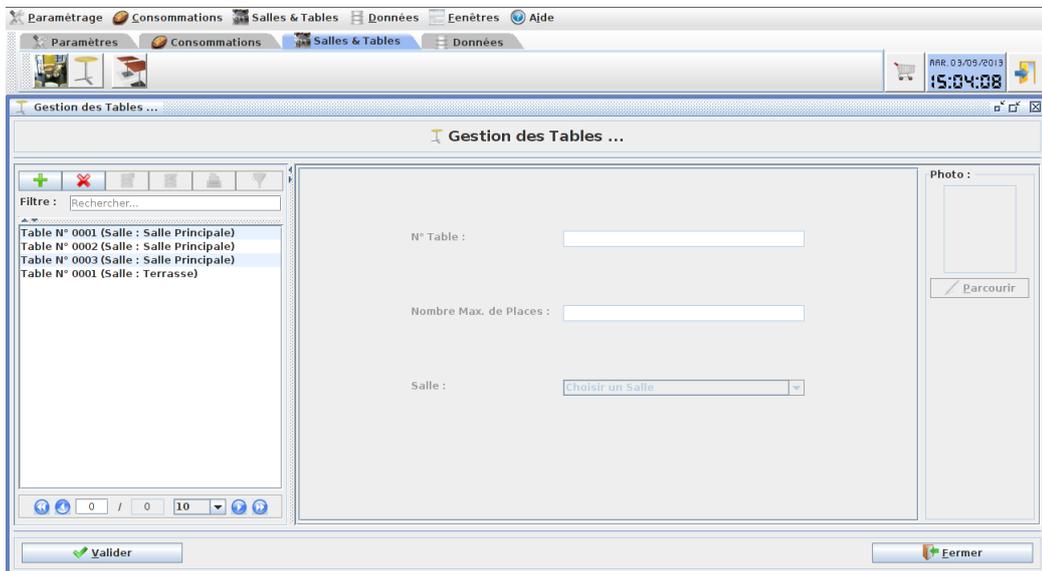


FIGURE 4.4 – Interface de la gestion de table.

4.4.4 Interface Editer une commande

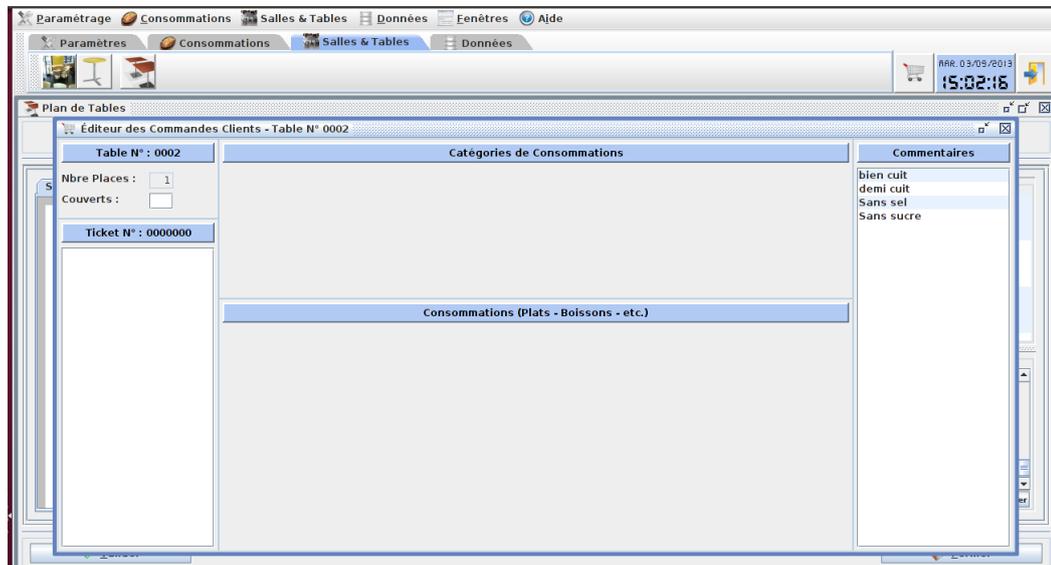


FIGURE 4.5 – Interface Editer une commande.

4.4.5 Interface Plan de table

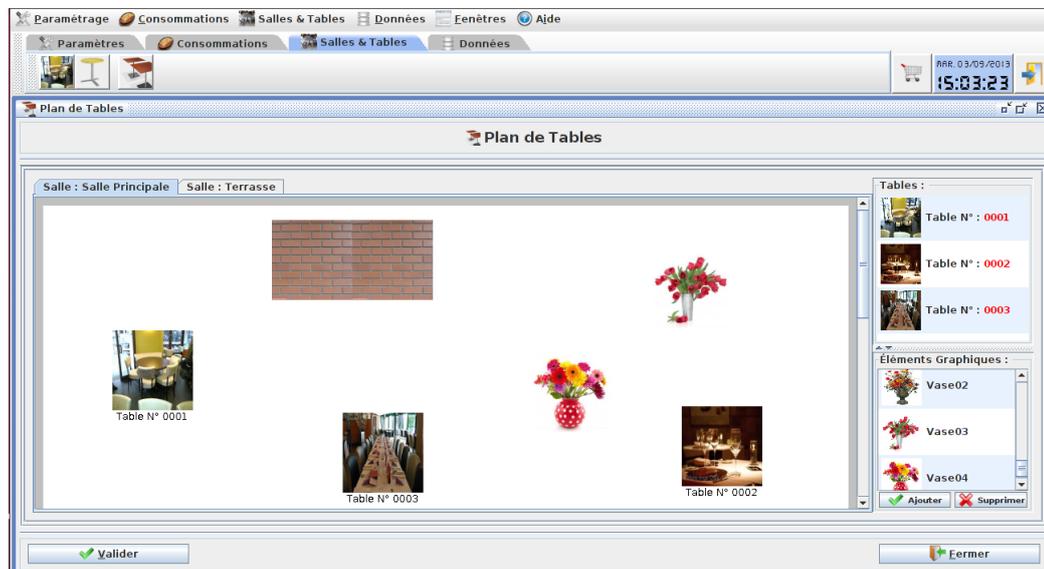


FIGURE 4.6 – Interface Plan de table.

4.5 Conclusion

La phase de réalisation est l'étape la plus importante dans le cycle de vie d'une application. Dans ce chapitre, nous avons présenté les outils qui nous ont permis la réalisation de notre application et les différentes fonctionnalités sous forme de fenêtres.

CONCLUSION GÉNÉRALE

Le travail que nous avons accompli a pour principale vocation la conception et la réalisation d'une application réseau pour la gestion du restaurant de l'hôtel CHREA.

Cette application a permis en premier lieu de répondre aux besoins des utilisateurs par la résolution des problèmes de la gestion du restaurant, qui est l'étape la plus importante dans ce travail. En deuxième lieu, elle a permis d'orienter le personnel vers une application unique. En effet l'application réalisée fournit un point d'accès unique pour les informations importantes du restaurant.

Afin d'accomplir notre travail, nous avons choisi le Processus Unifié comme méthodologie à suivre au niveau du développement et le langage UML comme langage de modélisation. Ainsi nous nous sommes servis de java comme langage de programmation, dans un environnement Eclipse et comme un gestionnaire de base de données le serveur MySQL.

BIBLIOGRAPHIE

- [1] Pascal Roque, "UML2 par la pratique, Etude de cas et exercices corrigés", EYROLLES 5e édition,2006.
- [2] Pascal Roque, Franck Vallée , "UML 2 en action, de l'analyse des besoins à la conception",EYROLLES 4e édition,2004.
- [3] Joseph Gabay, David Gabay, " UML2 Analyse et conception, Mise en IJuvre guidée avec étude de cas ", DUNOB, 2008.
- [4] Gilles Roy, "Conception de bases de données avec UML ", Presses de l'Université de Québec,1e édition, 2009.
- [5] Pascal Roques, "Les cahiers de programmeurs, UML 2 Modéliser une application web ",23e édition,EYROLLES,2007.
- [6] Xavier Blanc, Isabelle Mounier et Cédric Besse,"UML2 pour les développeurs, cours avec exercices corrigés",EYROLLES.
- [7] Laura Lemay et Charles L.Perkins,"Le programmeur, Apprenez java1.1 en 21 jours ",s et SM,1997.
- [8] Rémy Lentzner, " 300 astuces pour SQL et MYSQL ",2001.
- [9] Jean-Marie Defrance,PHP/MySQL avec Greamweaver 8,EYROLLES,2006.

Webliographie

- [w1] [http ://www.rebent.org/documents/document.pdf](http://www.rebent.org/documents/document.pdf)
- [w2] [http ://www.enseignement.polytechnique.fr/informatique/profs/Julien.Cervelle/eclipse/](http://www.enseignement.polytechnique.fr/informatique/profs/Julien.Cervelle/eclipse/)

Résumé

Aujourd'hui, l'informatique a atteint une prodigieuse évolution technologique dans différents domaines (réseaux informatiques, bases de données, le Web, etc.). Cette évolution est nécessaire pour remédier aux problèmes rencontrés dans la vie actuelle, c'est pour cela que l'hôtel CHREA a voulu informatiser son système de gestion de son restaurant.

Pour implémenter les besoins spécifiés au par avant, la modélisation est une étape clé. Nous avons utilisé le formalisme graphique proposé par UML2, ce dernier est un langage proche de l'orienté objet soutenu par un processus unifié UP, qui a son tour nous est venu en aide pour accélérer le processus de conception en assurant un certain parallélisme entre la modélisation et la réalisation proprement dite pour respecter les délais fixés.

Afin de réaliser notre application sous un environnement client-serveur, nous avons utilisé comme serveur de base de données "MySQL ". Quant à la conception des interfaces, nous avons utilisé le langage de programmation : Java.

Mots clés : UP, UML, MYSQL, Java, Eclipse, Gestion du restaurant.

Abstract

today, the computer has reached a tremendous technological developments in different fields (computer networks, databases, web, etc..). This is necessary to address the problems in the present life, that is why the hotel Chréa wanted to computerize its system for managing restaurant.

To implement the requirements specified by the forward modeling is a key step. We used the model proposed by UML2 graphical formalism, it is close to a unified process supported by UP, which in turn object-oriented language has helped us to accelerate the design process by providing a parallelism between the modeling and actual achievement to meet deadlines.

To achieve our application in a client-server environment, we have used as a database server "MySQL" database. As for interface design, we used the programming language java.

Keywords : UP, UML, MYSQL, Java, Eclipse, managing restaurant