



N° Réf :.....

Centre Universitaire de Mila

Institut des Sciences et de Technologie / Département de mathématique et informatique

**Mémoire préparé en vue de l'obtention de diplôme de licence
en: - filière : informatique général**

Thème

***La conception et la réalisation d'un site web
dynamique pour réservation en ligne
dans un hôtel***

Préparé par :

***GUENDOZ BILAL.
ABID MOHAMMED.
BOUNNECHE ZAKARYA.***

Encadre par : ***HETTAB ABDELKAMEL.***

Année universitaire : 2013/2014

Sommaire

Introduction Générale

Chapitre 1

l'internet et les applications web

1.1 Introduction.....	1
1.2.Internet.....	1
1.2.1 Qu'est-ce qu'un internet.....	1
1.2.1.1 Définition Internet.....	1
1.2.2 Les protocole internet.....	1
1.2.2.1 Le protocole TCP	1
1.2.2.2 Le Protocole IP.....	2
1.2.2.3 Le protocole http.....	2
1.2.3 Qu'est-ce qu'un web.....	2
1.2.4 Qu'est-ce qu'un site web.....	2
1.2.4.1 Site web statique	2
1.2.4.2 Site web dynamique.....	3
1.2.5 Page web.....	4
1.2.5.1 Page Web statique.....	4
1.2.5.1 Page Web statique.....	4
1.2.6 Navigateur.....	5
1.2.7 URL	5
1.2.8 LES SERVICES DE L'INTERNET.....	5
1.2.8.1 TELNET.....	5
1.2.8.2 Les news forums.....	6
1.2.8.3 FTP (File Transfer Protocol).....	6
1.2.8.4 Wais (Wide area Information Service).....	6
1.2.8.5 Courrier électronique.....	6
1.2.8.6 DNS.....	6
1.2.10 architectures client/serveur.....	7
1.2.10.1 Définition	7
1.2.10.2 Avantage de l'architecteur client-serveur	7
1.2.10.3 Inconvénient de l'architecteur client-serveur	7

1.3 conclusion	8
----------------------	---

Chapitre 2

l'UML et le processus unifié UP

2.1. Introduction.....	9
2.2. UML.....	9
2.2.1. Définition UML.....	9
2.2.2. Histoire d'UML	9
2.2.3. Les principes de la modélisation.....	10
2.2.4. Les objectifs d'UML.....	10
2.2.5. UML et les domaines d'utilisation.....	10
2.2.6. Les trois éléments de base en UML.....	10
2.2.7. UML est une unification des méthodes.....	10
2.2.8. UML n'est pas une méthode.....	11
2.2.9. Avantages et inconvénients d'UML	11
2.2.9.1. Les avantages d'UML.....	11
2.2.9.2. Les inconvénients d'UML.....	11
2.2.10. Les diagrammes de l'UML	11
2.2.10.1. Diagrammes des cas d'utilisation	11
2.2.10.2. Diagramme de classes	12
2.2.10.3. Diagramme d'objet	12
2.2.10.4. Diagrammes de composants.....	12
2.2.10.5. Diagrammes de déploiement.....	12
2.2.10.6. Diagrammes comportementaux (dynamiques).....	13
2.2.10.6.1. Diagrammes de séquence	13
2.2.10.6.2. Diagramme des activités.....	13
2.2.10.6.3. Diagramme de collaboration.....	13
2.2.10.6.4. Diagramme états-transitions.....	14
2.3. Le processus unifié.....	14
2.3.1. Définition	14
2.3.1.1. Piloté par les cas d'utilisation.....	14
2.3.1.2. Centré sur l'architecture.....	14
2.3.1.3. Itératif et incrémental	14
2.3.2. Les phases d'UP.....	15
2.3.2.1. Phase d'identification des besoins et spécification des fonctionnalités.....	15

2.3.2.1.1. Identification et représentation des besoins.....	15
2.3.2.1.2. Spécification détaillée des besoins.....	15
2.3.2.1.3. Maquette de l'IHM de l'application.....	16
2.3.2.2. Phase d'analyse.....	16
2.3.2.2.1. Analyse du domaine : Modèle du domaine.....	16
2.3.2.2.2. Diagramme de classes participantes.....	17
2.3.2.2.2.1. Les classe de dialogues	17
2.3.2.2.2.2. Les classe de contrôles	17
2.3.2.2.2.3. Les classes entités.....	17
2.3.2.3. Phase de conception.....	18
2.3.2.3.1. Diagramme d'interaction.....	18
2.3.2.3.2. Diagramme de classes de conception.....	18
2.3.2.4. Phase d'implémentation.....	19
2.4. Conclusion.....	19

Chapitre 3

Identification des besoins

3.1 Introduction.....	20
3.2 Identification des cas d'utilisation	20
3.3 Diagramme de cas d'utilisation.....	21
3.4 Description des cas d'utilisation (fiche descriptive).....	22
3.4.1 Cas d'utilisation « s'authentifier »	22
3.4.2 Cas d'utilisation « Réserver une chambre »	22
3.4.3 Cas d'utilisation « Annuler la réservation »	23
3.4.4 Cas d'utilisation « Modifier la réservation ».....	23
3.4.5 Cas d'utilisation « Enregistrer réservation »	23
3.4.6 Cas d'utilisation « modifier les tarifs »	24
3.4.7 Cas d'utilisation « consulter la liste des réservation en cours ».....	24
3.4.8 Cas d'utilisation « Supprimer les réservations consommées ».....	24
3.4.9 Cas d'utilisation « consulter le site ».....	25
3.5 Les diagrammes de séquence système.....	25
3.5.1 S'authentifier.....	25
3.5.2 Réserver une chambre.....	26
3.5.3 Annuler la réservation.....	27
3.5.4 Modifier la réservation	28

3.5.5 Enregistrer la réservation	28
3.5.6 Modifier les tarifs.....	29
3.5.7 Consulter les listes des réservations en cours.....	29
3.5.8 Supprimer les réservations consommées	30
3.5.9 Consulter le site.....	30
3.6 Conclusion	31

Chapitre 4

l'analyse du domaine

4.1 Introduction.....	32
4.2 analyse du domaine.....	32
4.3 Identification des concepts de domaine	32
4.4 Diagramme de domaine	33
4.5 Les diagrammes de classes participantes.....	34
4.5.1 S'authentifier	34
4.5.2 Réserver une chambre	35
4.5.3 Annuler la réservation.....	36
4.5.4 Modifier la réservation.....	37
4.5.5 Enregistrer la réservation.....	38
4.5.6 Modifier les tarifs	39
4.5.7 Consulter la liste des réservations en cours.....	39
4.5.8 Supprimer les réservations consommées.....	40
4.6 Les diagrammes d'activités de navigation	41
4.6.1 S'authentifier	41
4.6.2 Réserver une chambre	42
4.6.3 Annuler la réservation	43
4.6.4 Modifier la réservation.....	44
4.6.5 Enregistrer la réservation.....	45
4.6.6 modifier les tarifs.....	46
4.6.7 Consulter la liste des réservations en cours.....	47
4.6.8 Supprimer la réservation.....	48
4.7 Conclusion.....	48

Chapitre 5

phase de conception

5.1 introduction.....	49
5.2 Les diagrammes d'interaction.....	49
5.2.1 S'authentifier	49
5.2.2 Réserver chambre	50
5.2.3 Annuler réservation	51
5.2.4 Modifier la réservation.....	52
5.2.5 Enregistrer réservation.....	53
5.2.6 Modifier les tarifs	54
5.2.7 Consulter la liste des réservations en cours.....	55
5.2.8 Supprimer les réservation consommées.....	56
5.3 Les diagrammes de classes de conception	57
5.3.1 S'authentifier.....	57
5.3.2 réserver une chambre.....	58
5.3.3 annuler la réservation.....	59
5.3.4 modifier la réservation.....	60
5.3.5 enregistrer la réservation.....	61
5.3.6 modifier les tarifs.....	62
5.3.7 consulter la liste des réservations en cours.....	63
5.3.8 supprimer les réservations consommées	64
5.4 Diagramme de classe.....	65
5.5 Conclusion	66

Chapitre 6

l'implémentation

6.1. Introduction.....	67
6.2.1. Langage HTML	67
6.2.2. Définition.....	67
6.2.3. Principe du langage HTML	68
6.2.4. La structure générale d'un document HTML	68
6.3. JAVA SCRIPT	69
6.4. Bases de données et MySQL	69
6.4.1. Structure d'une base de données	69
6.4.2 Base de données MySQL	69
6.4.3 Base de données et applications web dynamiques	70

6.4.4. Passage du diagramme de classe au modèle relationnel.....	70
6.4.4.1 Transformation des entités /classes	70
6.4.4.2 Transformation des associations	70
6.4.5. Le serveur de bases de données MySQL	71
6.4.6. Pourquoi choisir MySQL	71
6.4.7. Organisation générale de MySQL	72
6.5. Langage PHP	72
6.5.1. Origine du langage PHP	72
6.5.2. Avantages du langage PHP	73
6.6. Easy PHP	73
6.7. DREAMWEAVER CS5.....	74
6.8. Représentation de l'interface de site	75
6.9. Conclusion	76
Conclusion Générale	

Liste des figures

Chapitre 1

L'internet et les applications web

Figure 1.1 : d'un site web statique.....	3
Figure 1.2 : fonctionnement d'un site web dynamique.....	4
Figure 1.3 : composant d'une ULR.....	5

Chapitre 2

UML ET le processus unifié UP

Figure 2.1 : Histoire d'UML	9
Figure 2.2 : processus itératif et incrémental	15
Figure 2.3 : Les besoins sont modélisés par un diagramme de cas d'utilisation	15
Figure 2.4 : Les diagrammes de séquence système illustrent la description textuelle des cas d'utilisation.....	15
Figure 2.5 : Une maquette d'IHM facilite les discussions avec les futurs Utilisateurs.....	16
Figure 2.6 : La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classe.....	17
Figure 2.7 : Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation, Le modèle du domaine et les diagrammes de conception logicielle.....	18
Figure 2.8 : Diagramme de classes de conception	19

Chapitre 3

Identification des besoins

Figure 3.1: Diagramme de cas d'utilisation	21
Figure 3.2: Diagramme de séquence système cas d'utilisation « s'authentifier ».....	25
Figure 3.3: Diagramme de séquence système cas d'utilisation « Réserver une chambre ».....	26
Figure 3.4: Diagramme de séquence système cas d'utilisation « Annuler la réservation ».....	27
Figure 3.5: Diagramme de séquence système cas d'utilisation « Modifier la réservation ».....	28
Figure 3.6: Diagramme de séquence système cas d'utilisation «Enregistrer la réservation».....	28
Figure 3.7: Diagramme de séquence système cas d'utilisation «Modifier les tarifs».....	29
Figure 3.8: Diagramme de séquence système cas d'utilisation «Consulter les listes des réservations en cours».....	29
Figure 3.9: Diagramme de séquence système cas d'utilisation «Supprimer les réservations consommées».....	30
Figure 3.10: Diagramme de séquence système cas d'utilisation « Consulter le site ».....	30

Chapitre 4

Analyse du domaine

Figure 4.1 diagramme de domaine.....	33
Figure 4.2 diagramme de classe participantes (s'authentifier).....	34
Figure 4.3 diagramme de classe participantes (réserver une chambre).....	35
Figure 4.4 diagramme de classe participantes (annuler la réservation).....	36
Figure 4.5 diagramme de classe participantes (modifier la réservation).....	37
Figure 4.6 diagramme de classe participantes (enregistrer la réservation).....	38

Figure 4.7 diagramme de classe participantes (modifier les tarifs).....	39
Figure 4.8 diagramme de classe participantes (consulter la liste des réservations en cours).....	39
Figure 4.9 diagramme de classe participantes (supprimer les réservations consommées).....	40
Figure 4.10 diagramme d'activités de navigation (s'authentifier).....	41
Figure 4.11 diagramme d'activités de navigation (réserver une chambre).....	42
Figure 4.12 diagramme d'activités de navigation (annuler la réservation).....	43
Figure 4.13 diagramme d'activités de navigation (modifier la réservation).....	44
Figure 4.14 diagramme d'activités de navigation (enregistrer la réservation).....	45
Figure 4.15 diagramme d'activités de navigation (modifier les tarifs).....	46
Figure 4.16 diagramme d'activités de navigation (consulter la liste des réservations en cours).....	47
Figure 4.17 diagramme d'activités de navigation (supprimer les réservations consommées).....	48

Chapitre 5

Phase de conception

Figure 5.1 diagramme d'interaction (s'authentifier).....	49
Figure 5.2 diagramme d'interaction (réserver une chambre).....	50
Figure 5.3 diagramme d'interaction (annuler la réservation).....	51
Figure 5.4 diagramme d'interaction (modifier la réservation).....	52
Figure 5.5 diagramme d'interaction (enregistrer la réservation).....	53
Figure 5.6 diagramme d'interaction (modifier les tarifs).....	54
Figure 5.7 diagramme d'interaction (consulter la liste des réservations en cours).....	55
Figure 5.8 diagramme d'interaction (supprimer des réservations consommées).....	56
Figure 5.9 diagramme de classe de conception (s'authentifier).....	57
Figure 5.10 diagramme de classe de conception (réserver une chambre).....	58
Figure 5.11 diagramme de classe de conception (annuler la réservation).....	59
Figure 5.12 diagramme de classe de conception (modifier la réservation).....	60
Figure 5.13 diagramme de classe de conception (enregistrer la réservation).....	61
Figure 5.14 diagramme de classe de conception (modifier les tarifs).....	62
Figure 5.15 diagramme de classe de conception (consulter la liste des réservations en cours).....	63
Figure 5.16 diagramme de classe de conception (supprimer les réservations consommées).....	64
Figure 5.17 diagramme de classe.....	65

Chapitre 6

L'implémentation

Figure 6.1 : Langage HTML.....	67
Figure 6.2 : organisation de MySQL.....	72
Figure 6.3: Easy PHP.....	74
Figure 6.4: Dreamweaver CS5.....	74
Figure 6.5 : Représentation de l'interface de site.....	75

Résumer

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.

Chaque création nécessite une modélisation avec un langage universel bien spécifié tel qu'UML la réalisation quant à elle nécessite des outils de développements bien adaptés au contexte de l'application. Pour les bases de données, l'utilisation d'un SGBD tel que MySQL, ORACLE, PostgreSQL, etc. est indispensable.

Notre travail consiste à concevoir un site Web dynamique pour la gestion de réservation dans un hôtel, ce dernier offre des services à l'utilisateur avec un processus de réservation facile et il donne du temps et de l'effort, aussi rendre plus facile pour l'administration d'hôtel de communiquer rapidement et efficacement avec les clients.

Pour elle, nous avons utilisé les langages de programmation suivants:

- PHP: largement utilisé sur Internet, permettra de rendre le site dynamique
- MYSQL : Ce langage permet la préparation d'une base de données d'utilisateurs.
- HTML: permet de créer un site web statique.

Finalement, le diplôme que nous venons d'obtenir sera pour nous une ouverture sur le domaine de recherche en informatique, et une expérience afin de faire part à d'autres stagiaires de notre domaine.

Summery

Data processing reached an extraordinary technological development in various fields (data-processing networks, databases, the Web, etc). This evolution is necessary to cure the problems encountered in the current life.

Each creation requires a modeling with a universal language good specified such as UML, the realization as for it requires development tools adapted well to the context of the application. For the databases, the use of a DBMS such as MySQL, ORACLE, PostgreSQL, etc is essential.

This project is to develop a dynamic website to manage reservation in a hotel, and enjoy the services offered to the user with an easy booking process and it gives time and effort, also make it easier for administration to communicate quickly and effectively with clients.

For her, we used the following programming languages:

- PHP: widely used on the Internet, will make the site dynamic
- MYSQL: This language allows the preparation of a database of user data.
- HTML: allows you to create a static website.

Finally, the degree just to get us to be an opening in the field of research in computer science and experience to share with other students in our area.

التكنولوجية المذهلة () البيانات
وهذا نواجهها في وقتنا الحاضر .
يتطلب UML عالمية ولتحقيق يتطلب تطوير لسياق التطبيق
البيانات .

البيانات الخلية

ان هذا المشروع يتمثل في اعداد موقع الكتروني ديناميكي خاص بالحجز في الفندق و الاستفادة من الخدمات المتاحة حيث يسهل عملية الحجز و يوفر عليه الوقت و الجهد كما يسهل ايضا على ادارة الفندق التواصل المباشر و السريع مع الزبائن .
لأجل ذلك استعملنا لغات البرمجة الآتية :

PHP

MYSQL هذه اللغة تسمح بإعداد قاعدة بيانات للمستخدمين

HTML مل هذه اللغة لإعداد مواقع ثابتة

في الاخير هذا العمل مكننا من اكتساب معارف هامة في مجال المعلوماتية و حقق تطلعاتنا .

Introduction Générale

Aujourd'hui, Internet offre donc un grand éventail de possibilités, qui sont de plus en plus importantes de jour en jour avec entre autre des connections haut débit comme le Câble ou l'ADSL. On trouve de tout sur le net : de la musique, des images, des logiciels, des jeux, des informations ...

Dans notre projet nous allons essayer de réaliser un site web dynamique pour la gestion d'un hôtel pour faire une communication entre le visiteur et l'administration d'hôtel et pour faciliter la réservation dans un hôtel en créance sur l'évolution technologie d'internet. Cette solution résumer le temps et l'effort.

Pour la réalisation de ce site nous allons choisir l'outil de modélisation UML (langage de modélisation unifier) parmi plusieurs méthodes car il est facile et de langages de programmation PHP, HTML.

Notre site contient les personnes suivantes:

- Le Visiteur: le personne qui visiter le site et les services disponible.
- Le Client: le personne qui gérer leur réservation (réserver une chambre, modifier la réservation, annuler la réservation....etc.).
- L'Agent d'administratif : le personne qui gérer le site et les réservations.

Organisation du mémoire :

Notre mémoire organise 6 chapitres :

- **Chapitre 01:** dans ce chapitre nous allons présenter la définition d'internet et leur protocole et les applications web.
- **Chapitre 02:** dans ce chapitre nous allons définie l'outil de conception (UML).
- **Chapitre 03:** ce chapitre représenter l'analyse de tout les besoins d'utilisateurs (diagramme de cas d'utilisation, les fiches descriptives, diagramme de séquence système).
- **Chapitre 04:** dans ce chapitre nous allons élaborer le modèle de domaine on suite les diagrammes de classe participante et enfin les diagrammes d'activité de navigation.
- **Chapitre 05 :** ce chapitre représenté la conception du site en utilisant les processus de développement UP (diagramme d'interaction, diagramme de conception, diagramme de classe).
- **Chapitre 06 :** ca chapitre consacré à la phase d'implémentation en utilisant les outils de développement suivant : PHP, HTML, EasyPHP, MYSQL, DreamWeaver CS5.5.

Le mémoire se termine par une conclusion générale.



PARTIE 1

*ETAT DE
L'ART UML
ET LES
APPLICATION
WEB*





CHAPITRE 1

*L'INTERNET
ET
LES
APPLICATION
WEB*



1.1 Introduction :

S'il est un phénomène qui a marqué le monde de l'informatique ces dernières années, c'est bien celui d'Internet. Ce réseau mondial créé en 1969 par l'armée américaine, puis utilisé par les chercheurs et les scientifiques, a connu une croissance phénoménale après du grand public avec l'introduction du World Wide Web en 1989.

Le Web, c'est l'application client/serveur la plus largement déployée au monde. Ce dernier permet de publier simplement des informations riches, des mises en forme et pouvant même, par la suite contenir des documents multimédia.

Cependant il est nécessaire d'avoir un minimum de notion sur la technologie Internet, l'architecture client/serveur et le Web.

1.2 Internet :

1.2.1 Qu'est-ce qu'un internet :

1.2.1.1 Définition Internet :

Est un système d'interconnexion de machines qui constitue un réseau informatique mondial, utilisant un ensemble standardisé de protocoles de transfert de données. C'est un réseau de réseaux, sans centre névralgique, composé de millions de réseaux aussi bien publics que privés, universitaires, commerciaux et gouvernementaux. Internet transporte un large spectre d'information et permet l'élaboration d'applications et de services variés comme le courrier électronique, la messagerie instantanée et le World Wide Web. [1]

Le concept a été inventé pendant la guerre froide, par l'agence américaine ARPA. Il permettait de sécuriser les transmissions informatiques contre les attaques nucléaires. [2]

1.2.2 Les protocole internet :

1.2.2.1 Le protocole TCP :

TCP (qui signifie Transmission Control Protocol, soit en français: Protocole de Contrôle de Transmission) est un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle (c'est-à-dire le protocole IP). Lorsque les données sont fournies au protocole IP, celui-ci les encapsule dans des datagrammes IP, en fixant le champ protocole à 6 (Pour savoir que le protocole en amont est TCP...). TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission. [3]

1.2.2.2 Le Protocole IP:

IP est le protocole spécifique à Internet, qui se charge de transmettre les données sous forme de paquets. L'envoi de ces paquets est réalisé en fonction des adresses de réseaux ou de sous-réseaux qu'ils contiennent. * Mise à jour du 22/01/2003 * IP a été inventé par Vinton Cerf et Bob Kahn en 1973 dans le cadre d'un projet de recherche de la Défense américaine (DARPA) : il s'agissait de trouver des technologies permettant de relier des réseaux transportant des paquets de données.[4]

1. 2.2.3 Le protocole http :

HTTP est un protocole de la couche application. il peut fonctionner sur n'importe quelle connexion fiable, dans les faits on utilise le protocole TCP comme couche de transport.

Les clients http les plus connus sont les navigateurs Web permettant à un utilisateur d'accéder à un serveur contenant les données. Il existe aussi des systèmes pour récupérer automatiquement le contenu d'un site tel que les aspirateurs de site Web ou les robots d'indexation.

Ces client se connectent à des serveurs HTTP tels qu'apache HTTP server ou internet [5]

1.2.3 Qu'est-ce qu'un web :

Le World Wide Web est constitué de plusieurs millions de pages Web du monde entier. Une page Web est un fichier informatique unique écrit en HTML et peut contenir de nombreux éléments multimédias tels du texte, des images, des sons, et des animations et peut être reliée à d'autres pages Web. L'aspect des liaisons du Web est à l'origine de son nom, étant donné qu'une représentation graphique des liaisons entre les fichiers du monde entier ressemble à une toile d'araignée (web en anglais). Pour visionner des pages Web, vous avez besoin d'un navigateur Web. [6]

1.2.4 Qu'est-ce qu'un site web :

Un site Web est un groupement de pages Web créé par une organisation ou par un particulier. Les sites Web sont généralement dotés d'une page principale appelée "page d'accueil" qui permet de grouper toutes les principales parties du site [6]

1.2.4.1 Site web statique :

Un site statique est un site qui représente des pages figées, reliées de manière définie les unes aux autres. Il s'agit des sites tels qu'ils existaient au début de l'Internet.. Il s'agit en gros d'un "prospectus" sur Internet, que l'on peut juste consulter. La figure suivante résume le format général des sites Web statiques. [7]

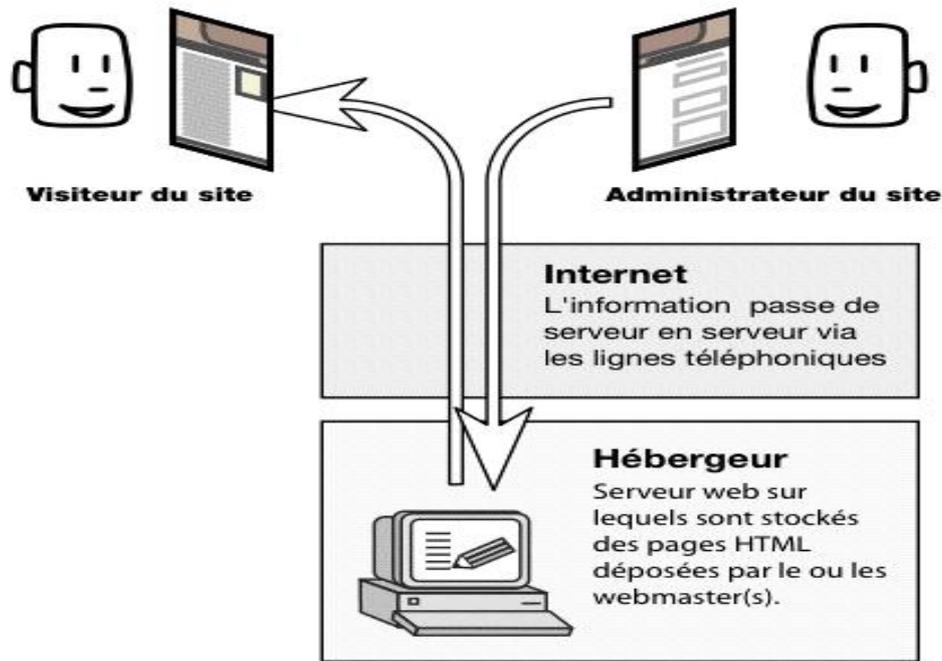


Figure 1.1 : d'un site web statique

1.2.4.2 Site web dynamique :

Site Web dont les pages HTML se construisent lors de sa consultation par un internaute en sollicitant des bases de données filtrées par des outils logiciels de mise en forme. La plupart des sites marchands sont des sites dynamiques. La mise à jour des sites dynamiques tant sur le fond que sur la forme est facilitée. Le site dynamique permet de plus d'intégrer des fonctions de personnalisation. Cette architecture peut en revanche se révéler coûteuse à mettre en place, délicate à piloter notamment au niveau de l'hébergement et peu performante au niveau du référencement. Par opposition, un site statique, est constitué de pages HTML "en dur", créées une fois pour toutes.[8]

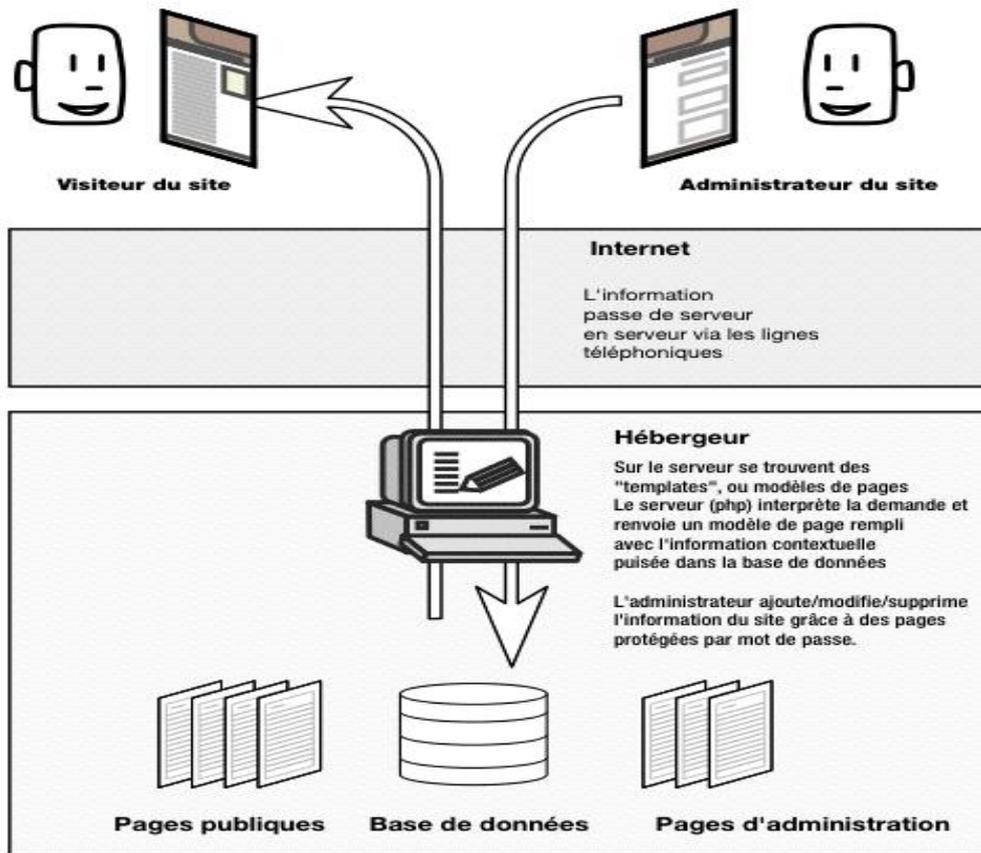


Figure 1.2 : fonctionnement d'un site web dynamique

1.2.5 Page web :

La page Web est l'unité de consultation du World Wide Web. Ce terme a une signification pratique ; il n'a pas de définition technique formelle. C'est un document informatique qui peut contenir du texte, des images, des formulaires à remplir et divers autres éléments multimédias et interactifs. [9]

1.2.5.1 Page Web statique :

Une page web statique est une page web dont le contenu ne varie pas en fonction des caractéristiques de la demande, c'est-à-dire qu'à un moment donné tous les internautes qui demandent la page reçoivent le même contenu. [10]

1.2.5.1 Page Web statique :

Une page web dynamique est une page web générée à la demande, par opposition à une page web statique. Le contenu d'une page web dynamique peut donc varier en fonction d'informations (heure, nom de l'utilisateur, formulaire rempli par l'utilisateur, etc.) qui ne sont connues qu'au moment de sa consultation. À l'inverse, le contenu d'une page web statique est a priori identique à chaque consultation. La figure suivante résume le format général des sites Web dynamique. [11]

1.2.6 Navigateur :

Un navigateur Internet est un logiciel informatique qui permet à un utilisateur d'accéder à l'Internet. En anglais on parle de Web Browser. Les plus connus sont Internet Explorer, Safari, Firefox... Au cœur du logiciel, il y a un moteur de rendu visuel qui interprète l'HTML contenu dans une page Internet. [12]

1.2.7 URL

Une URL (*Uniform Resource Locator*) est un format de nommage universel pour désigner une ressource sur Internet. Il s'agit d'une chaîne de caractères ASCII imprimables. [13]

Un URL se compose de plusieurs parties aux quelles une application peut accéder .Ces différentes parties sont décrite dans la figure suivante :

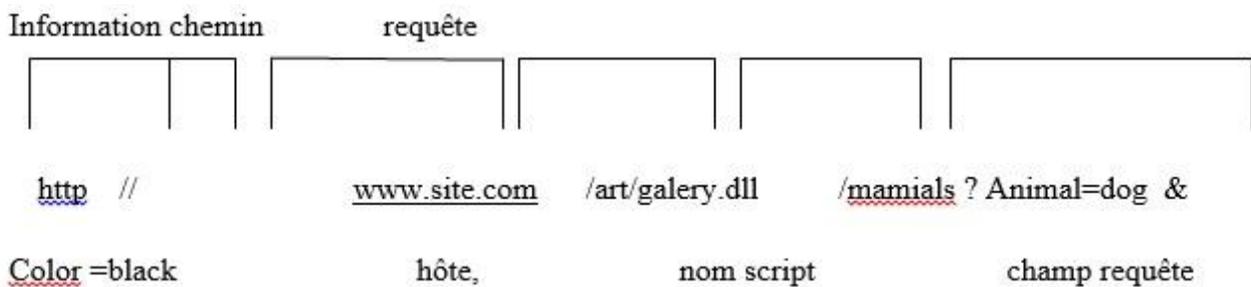


Figure 1.3 : composant d'une ULR

- La première partie (qui n'appartient pas techniquement à URL) identifie les protocoles HTTP. Cette partie peut spécifier d'autre protocole comme HTTPS (Secure HTTP, FTP, GOPHER, TELNET WAIS). La partie hôte identifie la machine qui exécute le serveur web et l'application serveur web.
- La partie nom script spécifie le nom de l'application serveur web. Il s'agit de l'application à laquelle le serveur web transmet les messages. à la suite du nom de script apparait chemin.
- Ce dernier identifie la destination du message dans l'application serveur web. Les valeurs du chemin peuvent faire référence à des messages spécifiques ou à tout autre mécanisme utilisés par l'application serveur web pour diviser le traitement des messages entrants
- La partie requête contient un ensemble de valeurs nommées ces valeurs et leurs noms sont définis l'application serveur web. [14]

1.2.8 Les services de l'internet :

1.2.8.1 TELNET :

Le protocole Telnet permet de se connecter sur un serveur distant à l'aide d'un identifiant et d'un mot de passe. L'adresse du service sera du type telnet://hôte. Il est utilisé pour exécuter des programmes à distance (par ex. bases de données, catalogues de bibliothèque, programme nécessitant une grande vitesse de calcul). [15]

1.2.8.2 Les news forums :

Usenet (les news, newsgroup ou groupes de nouvelles) est un service de conférences publiques accessibles à un nombre illimité de personnes. Pour y participer, il faut posséder un logiciel de messagerie spécialisé ou un navigateur. [15].

1.2.8.3 FTP (File Transfer Protocole)

Ce protocole permet de se connecter temporairement à un serveur FTP distant pour ramener ou déposer des fichiers. Pour le téléchargement des fichiers, un simple navigateur suffit. L'adresse du service sera du type ftp://hôte. [15]

1.2.8.4 Wais (Wide area Information Service)

Wide Area Information Servers, ou Service, ou System ou Searching, aussi connu sous l'acronyme WAIS, est un système informatique client-serveur permettant à un logiciel client WAIS de se connecter à un serveur répertoriant des bases de données documentaires distribuées sur un réseau informatique et de lancer une recherche dans le texte des documents. Les recherches peuvent s'effectuer par mots clés ou en langage courant. La réponse est donnée sous la forme d'une liste de documents consultables triés par ordre de pertinence. [16]

1.2.8.5 Courrier électronique

Le courriel permet aux internautes de s'échanger des documents, photos, liens, textes, etc....

Les messages (mails) qui sont envoyés et reçus, le sont grâce à deux protocoles ; SMTP (*Simple Mail Transfert Protocole*) et POP (*Post Office Protocole*). [17]

➤ **SMTP**

(Simple mail Transfer Protocol) est le protocole de transport et de remise du courrier d'un serveur à l'autre sur internet, il se charge en outre de terminer le meilleur itinéraire pour le courrier.

➤ **POP**

(Post office Protocol) est le protocole chargé de la gestion et de la remise du message email entre un serveur et un client. [17]

1.2.8.6 DNS :

Le Domain Name System (ou **DNS**, système de noms de domaine) est un service permettant de traduire un nom de domaine en informations de plusieurs types qui y sont associées, notamment en adresses IP de la machine portant ce nom. À la demande de la DARPA, Jon Postel et Paul Mockapetris ont conçu le « *Domain Name System* » en 1983 et en écrivirent la première réalisation.[18]

1.2.10 architectures client/serveur

1.2.10.1 Définition

L'environnement **client-serveur** désigne un mode de communication à travers un réseau entre plusieurs programmes ou logiciels : l'un, qualifié de client, envoie des requêtes ; l'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent. Par extension, le client désigne également l'ordinateur sur lequel est exécuté le logiciel client, et le serveur, l'ordinateur sur lequel est exécuté le logiciel serveur.

1.2.10.2 Avantage de l'architecteur client-serveur :

- Toutes les données sont centralisées sur un seul serveur, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels.
- Les technologies supportant l'architecture client-serveur sont plus matures que les autres.
- La complexité du traitement et la puissance de calculs sont à la charge du ou des serveurs, les utilisateurs utilisant simplement un client léger sur un ordinateur terminal qui peut être simplifié au maximum.
- Recherche d'information : les serveurs étant centralisés, cette architecture est particulièrement adaptée et véloce pour retrouver et comparer de vaste quantité d'informations (moteur de recherche sur le Web), ce qui semble être rédhibitoire pour le P2P beaucoup plus lent, à l'image de Free net.

1.2.10.3 Inconvénient de l'architecteur client-serveur :

- Si trop de clients veulent communiquer avec le serveur au même moment, ce dernier risque de ne pas supporter la charge (alors que les réseaux pair-à-pair fonctionnent mieux en ajoutant de nouveaux participants).
- Si le serveur n'est plus disponible, plus aucun des clients ne fonctionne (le réseau pair-à-pair continue à fonctionner, même si plusieurs participants quittent le réseau).
- Les coûts de mise en place et de maintenance peuvent être élevés.
- En aucun cas les clients ne peuvent communiquer entre eux, entraînant une asymétrie de l'information au profit des serveurs. [19]

1.3 Conclusion:

Dans ce chapitre nous avons fait une visites très rapide des technologies qui sont a la base de première hypertexte sur le web client/serveur .cet ensemble de technologie relativement simple permet de construire de manière très efficace des web interconnectes qui reflètent la nature distribuée de l'information. Les applications web reposant sur l'architecture client/serveur qui constitue une étape important sans l'évolution des systèmes informatiques.

Dans ce qui suit nous présentant la notation UML.



CHAPITRE 2

L'UML

ET

L'APPLICATION

WEB



2.1. Introduction

Pour programmer une application, il ne convient pas de se lancer tête baissée dans

L'écriture du code : il faut d'abord organiser ses idées, les documenter, puis organiser la réalisation en définissant les modules et étapes de la réalisation. C'est cette démarche antérieure à l'écriture que l'on appelle modélisation. Pour modéliser notre application nous avons choisi le langage UML (Unified modeling language). Dans ce chapitre nous allons présenter les concepts de base du langage de modélisation UML, ainsi que le processus (UP) que nous allons utiliser pour modéliser notre application.

2.2. UML

2.2.1. Définition UML

UML est la forme contractée d'Unified Modeling Language qui peut se traduire en français par le langage unifié pour la Modélisation. Il fournit les fondements pour spécifier, construire, visualiser et décrire les éléments d'un modèle [29]. Il se base sur une sémantique précise et sur une notation Graphique expressive. C'est un moyen d'exprimer des modèles en faisant abstraction de leur implémentation, c'est à dire le modèle fourni par UML est valable pour N'importe quel langage de programmation.

2.2.2. Histoire d'UML

L'unification a progressé par étapes. En 1995, Booch et Rumbaugh (et quelques autres) se sont mis d'accord pour construire une méthode unifiée, Unified Method 0.8 ; en 1996, Jacobson les a rejoints pour produire UML 0.9 (notez le remplacement du mot méthode par le mot langage, plus modeste). Les acteurs les plus importants dans le monde du logiciel s'associent alors à l'effort (IBM, Microsoft, Oracle, DEC, HP, Rational, Unisys etc.) et UML 1.0 est soumis à l'OMG5. L'OMG adopte en novembre 1997 UML 1.1 comme langage de modélisation des systèmes d'information à objets. La version d'UML en cours en 2008 est UML 2.2 et les travaux d'amélioration se poursuivent. [22]

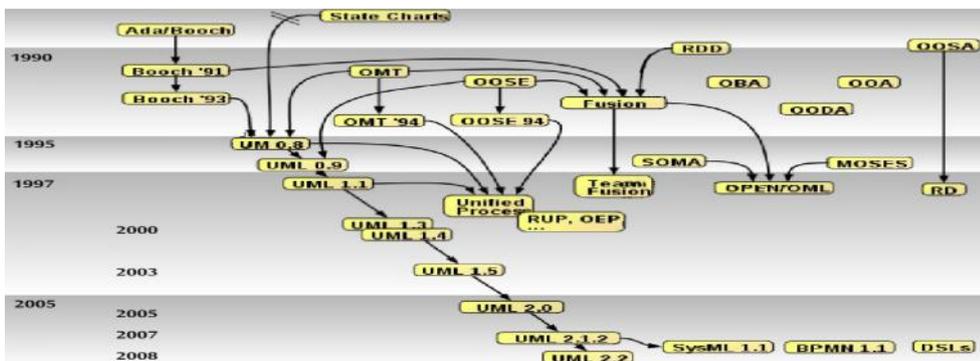


Figure 2.1 : Histoire d'UML [22]

2.2.3. Les principes de la modélisation

- 1) Le modèle doit être connecté au monde réel
- 2) Un modèle peut être exprimé avec différents niveaux de précision
- 3) Un simple modèle n'est pas suffisant, il y a plusieurs façons de voir un système. (Plan de masse, vue de face, de côté, ..., plan des niveaux, plan électrique, plan de plomberie, plan des calculs de construction). [26]

2.2.4. Les objectifs d'UML : sert à

- Décomposer le processus de développement.
- Mettre en relation les experts métiers et les analystes.
- Coordonner les équipes d'analyse et de conception.
- Séparer l'analyse de la réalisation.
- Prendre en compte l'évolution de l'analyse et du développement.
- Migrer facilement vers une architecture objet d'un point de vue statique et dynamique. [27]

2.2.5. UML et les domaines d'utilisation

Systemes d'information des entreprises

Les Banques et les services financiers

Télécommunications Transport Défense et aérospatiale Scientifique

Applications distribuées par le WEB [26]

2.2.6. Les trois éléments de base en UML

1) les blocs de base pour construire

_ Les entités utilisées (Entités structurelles ; Entités de comportement ; Entités de regroupement ; Entité d'annotation)

_ La notion de relation les diagrammes

2) les règles à observer pour utiliser ces blocs de base : Règles sémantiques règles de présentation³)

les mécanismes communs : spécification présentation extension des modèles [26]

2.2.7. UML est une unification des méthodes : UML est né de la fusion des trois méthodes

Ont le plus influencé la modélisation objet au milieu des années 90 :

- OMT de James Rumbaugh (General Electric) fournit une représentation graphique

Des aspects statique, dynamique et fonctionnel d'un système.

- OOD de Grady Booch, définie pour le Département of Défense, introduit le

Concept de paquetage (package).

- OOSE d'Ivar Jacobson (Ericsson) fonde l'analyse sur la description des besoins

Des utilisateurs (cas d'utilisation, ou use cases). [29]

2.2.8. UML n'est pas une méthode : UML n'est pas une méthode ou un processus UML est développé pour permettre la modélisation des systèmes d'une manière standard et pas pour être une méthode de conception ou d'analyse, pour ça il lui manque la démarche. [29]

2.2.9. Avantages et inconvénients d'UML

2.2.9.1. Les avantages d'UML

_ UML est un langage formel et normalisé, il permet un gain de précision et un gain de stabilité. Ce qui encourage l'utilisation d'outils.

_ UML est un support de communication performant, il cadre l'analyse et il facilite la compréhension des représentations abstraites complexes. [25]

_ Langage de contraintes OCL pour spécifications précises à utiliser en complément [23]

2.2.9.2. Les inconvénients d'UML

_ La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.

_ UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue. [25]

Sémantique floue ou mal définie pour certains types de diagrammes

Lien parfois difficile entre les vues et diagrammes d'une même [23]

2.2.10. Les diagrammes de l'UML :

UML définit plusieurs sortes de diagrammes pour représenter les différents points de vue de modélisation. Un diagramme donne à l'utilisateur un moyen de visualiser et de manipuler des éléments de modélisation.

Les diagrammes de l'UML peuvent être divisés en deux groupes : ceux qui décrivent les aspects statique et ceux qui décrivent la dynamique du système. [20]

2.2.10.1. Diagrammes des cas d'utilisation :

Le principe : Besoins des utilisateurs. [24]

Le diagramme de Cas d'utilisation constitue l'apport original de la méthode OOSE à la notation UML. L'approche consiste à regarder le système à construire de l'extérieur, du point de vue de l'utilisateur et des fonctionnalités qu'il en attend.

Les cas d'utilisation représentent le dialogue entre l'acteur et le système de manière abstraite.

Le diagramme de cas d'utilisation décrit la succession des opérations réalisées par un acteur (personne qui assure l'exécution d'une activité). C'est le diagramme principal du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre. [25]

2.2.10.2. Diagramme de classes :

Le principe : Description statique des données et des traitements. [24]

Les diagrammes de classes expriment la structure statique du système en termes de classe et de relation entre elle. Un diagramme de classes est une collection d'éléments de modélisation statiques (classes, paquetages...), qui montre la structure d'un modèle.

Le diagramme de classe représente l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens, que ceux-ci représentent un emboîtement conceptuel (héritage, marqué par une flèche terminée par un triangle) ou une relation organique (agrégation, marquée par une flèche terminée par un diamant). [25]

2.2.10.3. Diagramme d'objet :

Le principe : Instances des classes. [24]

Ce type de diagramme UML montre des objets (instances de classes dans un état Particulier) et des liens (relations sémantiques) entre ces objets. Les diagrammes d'objets s'utilisent pour montrer un contexte (avant ou après une interaction entre objets par exemple). Ce type de diagramme sert essentiellement en phase exploratoire, car il possède un très haut niveau d'abstraction. Le diagramme d'objet permet d'éclairer un diagramme de classe en l'illustrant par des exemples. [25]

2.2.10.4. Diagrammes de composants

Le principe: Représentation des composants logiciels d'un système. [24]

Représente les différents constituants logiciels d'un système (architecture logicielle d'un système). Ils permettent la description de l'architecture technique, des nœuds et leurs Interconnexions. Les nœuds de l'architecture sont des serveurs, des postes de travail et des périphériques. Les composants sont alloués aux différents nœuds. [25]

2.2.10.5. Diagrammes de déploiement

Le principe : Description de l'architecture technique du système. [24]

Décrit l'architecture technique d'un système (composants matériels du système informatique et leurs connexions).

Ils sont utilisés pour regrouper des éléments de modélisation : classes, cas D'utilisation ou autres diagrammes. Ils sont, aussi, utilisés pour:

- _ obtenir une vision de plus haut niveau.
- _ mettre en évidence des éléments réutilisables.
- _ répartir le travail entre développeurs.
- _ organiser la gestion de configuration (versionning). [25]

2.2.10.6. Diagrammes comportementaux (dynamiques)

2.2.10.6.1. Diagrammes de séquence :

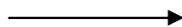
Le principe: Scénario d'un cas d'utilisation : chronologie des opérations. [24]

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur : saisir une donnée, consulter une donnée, lancer un traitement; il indique les objets que l'acteur va manipuler, et les opérations qui font passer d'un objet à l'autre. On peut représenter les mêmes opérations par un diagramme de collaboration.

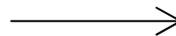
Les diagrammes de séquences et les diagrammes d'état-transitions sont les vues dynamiques les plus importantes d'UML. [25]

Les différentes communications :

- appel de procédure (*procedure call*) ou flot de contrôle imbriqué (nested flow of control) : représentés par une flèche dont la pointe est remplie (filled solid arrowhead);



- flot de contrôle non imbriqué (*flat flow of control*) : représenté par une flèche dont la pointe n'est pas remplie (*stick arrowhead*) ;



- retour de procédure (*return from procedure call*) : représenté par une flèche en pointillé dont la pointe n'est pas remplie



(Dashed arrow with stick arrowhead). [22]

2.2.10.6.2. Diagramme des activités

Le principe: Vue des enchaînements des activités d'un cas d'utilisation ou d'une opération. [24]

Un diagramme d'activité est une variante du diagramme d'états-transitions. Il s'applique à représenter les comportements internes des méthodes et des opérations des objets. C'est en fait un diagramme d'états-transitions qui s'organise par rapport aux actions.

Le diagramme d'activité n'est autre que la transcription dans UML de la représentation du processus telle qu'elle a été élaborée lors du travail qui a préparé la modélisation : il montre l'enchaînement des activités qui concourent au processus. [25]

2.2.10.6.3. Diagramme de collaboration

Le principe : Scénario d'un cas d'utilisation: activités des objets et des messages échangés. [24]

Décrit les scénarios de chaque cas d'utilisation en mettant l'accent sur l'interaction des objets et les messages échangés.

Ces diagrammes montrent les interactions entre les objets. Ils insistent particulièrement sur la structure des liens permettant de mettre les objets en collaboration, en communication. Ils représentent à la fois les objets et les messages que ceux-ci s'envoient. Les diagrammes de collaboration sont une extension des diagrammes d'objets. [25]

2.2.10.6.4. Diagramme états-transitions

Le principe : États des objets selon les événements. [24]

Montre les différents états des objets en réaction aux événements. Le diagramme Etats-Transitions représente la dynamique de chacun des objets (et donc du système entier) au moyen d'automates. Un automate est composé d'états, Modélisant une situation dans laquelle l'objet se situe à un instant précis. L'automate est composé de transitions, précisant comment on passe d'un état à un autre. Il décrit l'évolution au cours du temps d'une instance d'une classe en réponse aux interactions avec d'autres objets. [25]

2.3. Le processus unifié

2.3.1. Définition :

UP: Processus de développement proposé par J-B-R

Processus: Recensement des cas d'utilisation et Construction de l'architecture du système dès le début

Avec le principe d'itérations et incrémentations et évaluation des risques à toutes les étapes. [24]

Les traits véritablement distinctifs du processus unifié tiennent en trois expressions clés:

2.3.1.1. Piloté par les cas d'utilisation

L'objectif d'un système logiciel est de rendre service à ses utilisateurs. Pour réussir la mise au point d'un système, il importe, par conséquent, de bien comprendre les désirs et les besoins de ses futurs utilisateurs.

Un cas d'utilisation est une fonctionnalité du système produisant un résultat satisfaisant pour l'utilisateur. Les cas d'utilisation saisissent les besoins fonctionnels et leur ensemble forme le modèle des cas d'utilisation qui décrit les fonctionnalités complètes du système. [32]

2.3.1.2. Centré sur l'architecture

Le rôle de l'architecture logicielle est comparable à celle que joue l'architecte dans la construction d'un bâtiment. Le bâtiment est envisagé de différents points de vue: structure, services, conduite de chauffage, plomberie, etc. Ce regard multiple dessine une image complète du bâtiment avant le début de la construction. De la même façon, l'architecture d'un système logiciel peut être décrite comme les différentes vues du système qui doit être construit. [32]

2.3.1.3. Itératif et incrémental

Le développement d'un produit logiciel destiné à la commercialisation est une vaste opération qui peut s'étendre sur plusieurs mois, voire sur une année ou plus. Il n'est pas inutile de découper le travail en plusieurs parties qui sont autant de mini-projets (Concept systémique de système et sous-

désignent des étapes de l'enchaînement d'activités, tandis que les incréments correspondent à des stades de développement du produit. [32]

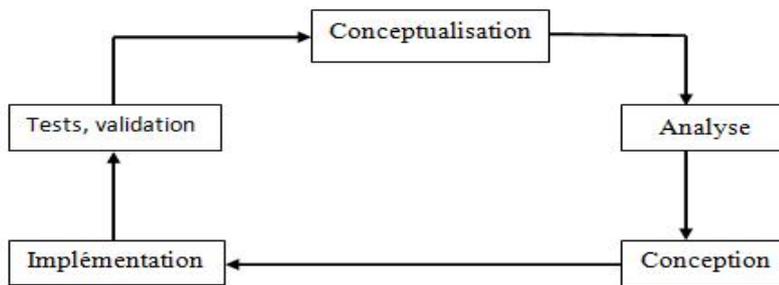


Figure 2.2 : processus itératif et incrémental [33]

2.3.2. Les phases d'UP

2.3.2.1. Phase d'identification des besoins et spécification des fonctionnalités

2.3.2.1.1. Identification et représentation des besoins

Identification et représentation des besoins à partir des informations recueillies lors des rencontres entre informaticiens et utilisateurs. [30]

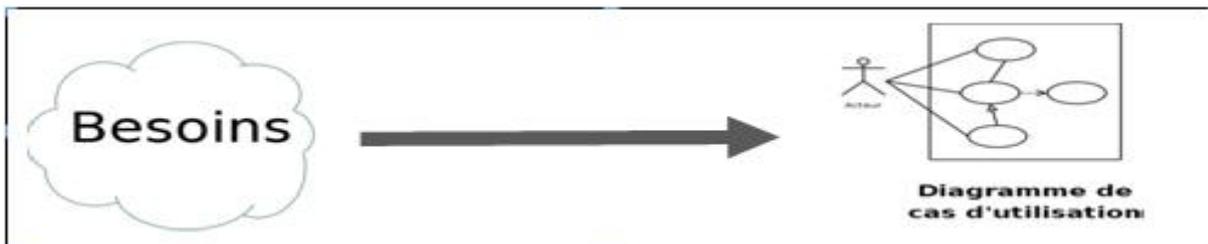


Figure 2.3 : Les besoins sont modélisés par un diagramme de cas d'utilisation [34]

2.3.2.1.2. Spécification détaillée des besoins

Dans cette étape, on cherche à détailler la description des besoins par la description textuelle des cas d'utilisation et la production de diagrammes de séquence système illustrant cette description textuelle. Cette étape amène souvent à mettre à jour le diagramme de cas d'utilisation puisque nous sommes toujours dans la spécification des besoins. [31]

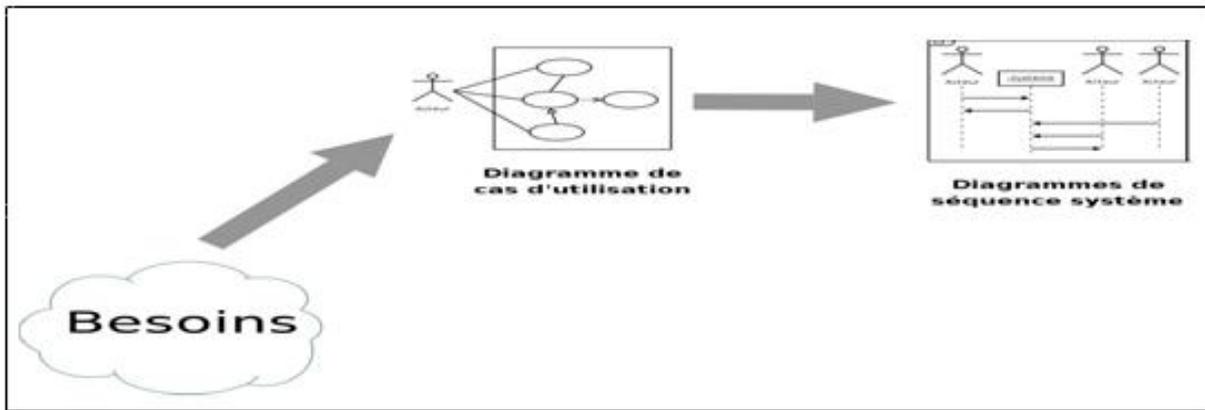


Figure 2.4 : Les diagrammes de séquence système illustrent la description Textuelle des cas d'utilisation. [34]

2.3.2.1.3. Maquette de l’IHM de l’application

Une maquette d’IHM (Interface Homme-Machine) est un produit jetable permettant aux utilisateurs d’avoir une vue concrète mais non définitive de la future interface de l’application. La maquette peut très bien consister en un ensemble de dessins produits par un logiciel de présentation ou de dessin. Par la suite, la maquette pourra intégrer des fonctionnalités de navigation permettant à l’utilisateur de tester l’enchaînement des écrans ou des menus, même si les fonctionnalités restent fictives. La maquette doit être développée rapidement afin de provoquer des retours de la part des utilisateurs. [31]

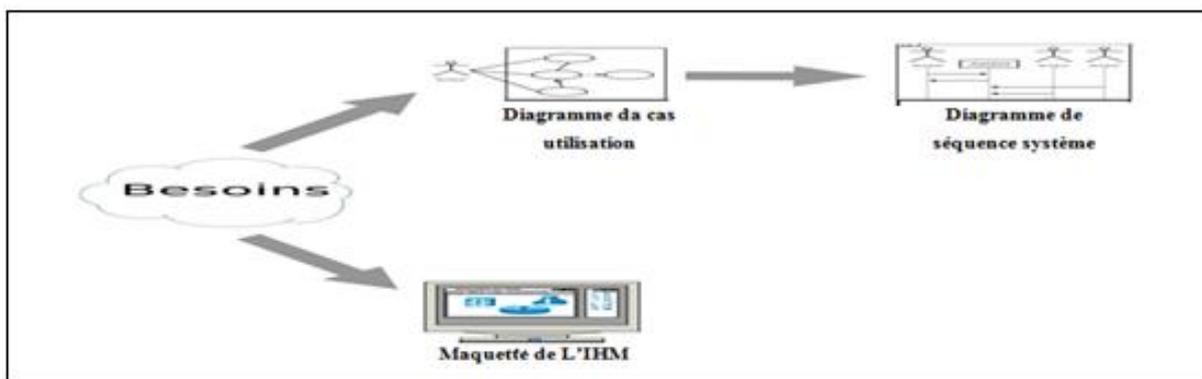


Figure 2.5 : Une maquette d’IHM facilite les discussions avec les futurs Utilisateurs [34]

2.3.2.2. Phase d’analyse

2.3.2.2.1. Analyse du domaine : Modèle du domaine

L’élaboration du modèle des classes du domaine permet d’opérer une transition vers une véritable modélisation objet. L’analyse du domaine est une étape totalement dissociée de l’analyse des besoins. Elle peut être menée avant, en parallèle ou après cette dernière. [29]

La phase d’analyse du domaine permet d’élaborer la première version du diagramme de classes appelée modèle du domaine. Ce modèle décrit les classes qui modélisent les concepts du modèle

d'application (on utilise aussi le terme de métier), c.à.d. des objets du monde réel. Les concepts du domaine peuvent être identifiés directement à partir de la connaissance du domaine ou par interview des experts métier. [31]

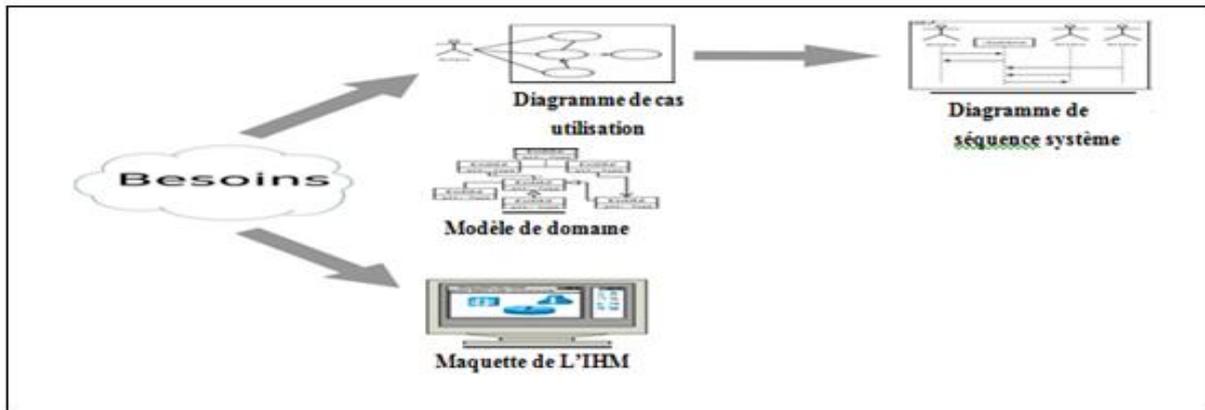


Figure 2.6: La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classe. [34]

2.3.2.2.2. Diagramme de classes participantes

Le diagramme de classes participantes est particulièrement important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation, le modèle du domaine et la maquette, et d'autre part, les diagrammes de conception logicielle que sont les diagrammes d'interaction et le diagramme de classes de conception. Le diagramme de classes participantes modélise trois types de classes d'analyse, les dialogues, les contrôles et les entités ainsi que leurs relations :

2.3.2.2.2.1. Les classes de dialogues :

Les classes qui permettant les interactions entre l'IHM et les utilisateurs sont qualifiées de dialogues. Ces classes sont directement issues de l'analyse de la maquette. En général les dialogues vivent seulement le temps du déroulement du cas d'utilisation concerné.

2.3.2.2.2.2. Les classes de contrôles :

Les classes qui modélisent la cinématique de l'application sont appelées contrôles. Elles font la jonction entre les dialogues et les classes métier en permettant aux différentes vues de l'application de manipuler des informations détenues par un ou plusieurs objets métier. Elles contiennent les règles applicatives et les isolent à la fois des dialogues et des entités.

2.3.2.2.2.3. Les classes entités:

Les classes métier, qui proviennent directement du modèle du domaine, sont qualifiées d'entités. Ces classes sont généralement persistantes, c'est-à-dire qu'elle survient à l'exécution d'un cas d'utilisation particulier et qu'elles permettant à des données et des relations d'être stockées dans des

fichiers ou des bases de données. Lors de l'implémentation, ces classes peuvent ne pas se concrétiser par des classes mais par des relations, au sens des bases de données relationnelles. [28]

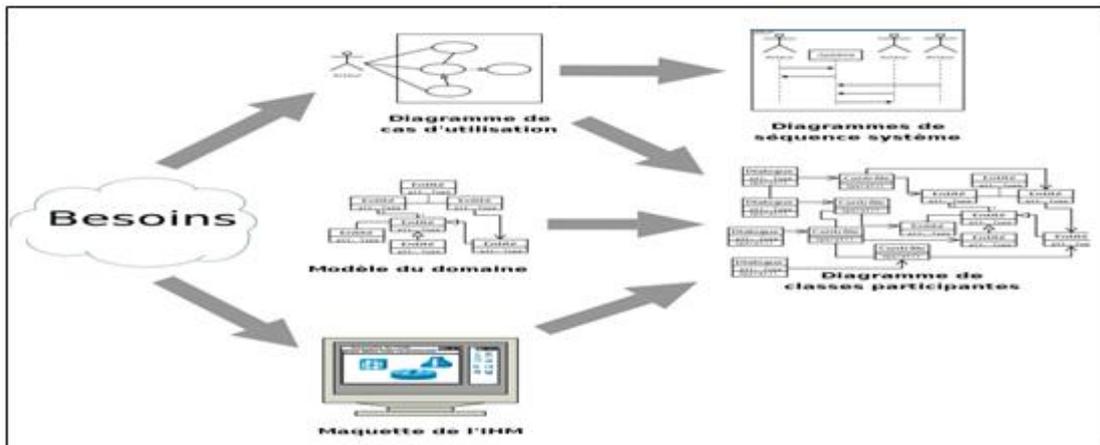


Figure 2.7 : Le diagramme de classes participantes effectue la jonction entre les cas D'utilisation, Le modèle du domaine et les diagrammes de conception logicielle [34]

2.3.2.3. Phase de conception

2.3.2.3.1. Diagramme d'interaction

Il faut attribuer précisément les responsabilités de comportement, dégagée par les diagrammes de séquence système, aux classes d'analyse du diagramme de classes participantes élaboré. Les résultats de cette réflexion sont présentés sous la forme de diagrammes d'interaction UML. Parallèlement, une première ébauche de la vue statique de conception, c'est à dire du diagramme de classes de conception, est construite et complétée. Durant cette phase, l'ébauche du diagramme de classes de conception reste indépendante des choix technologiques qui seront faits ultérieurement. [31]

2.3.2.3.2. Diagramme de classes de conception

L'objectif de cette étape est de produire le diagramme de classes qui servira pour l'implémentation. Une première ébauche du diagramme de classes de conception a déjà été élaborée en parallèle du diagramme d'interaction. Il faut maintenant le compléter en précisant les opérations privées des différentes classes. [31]

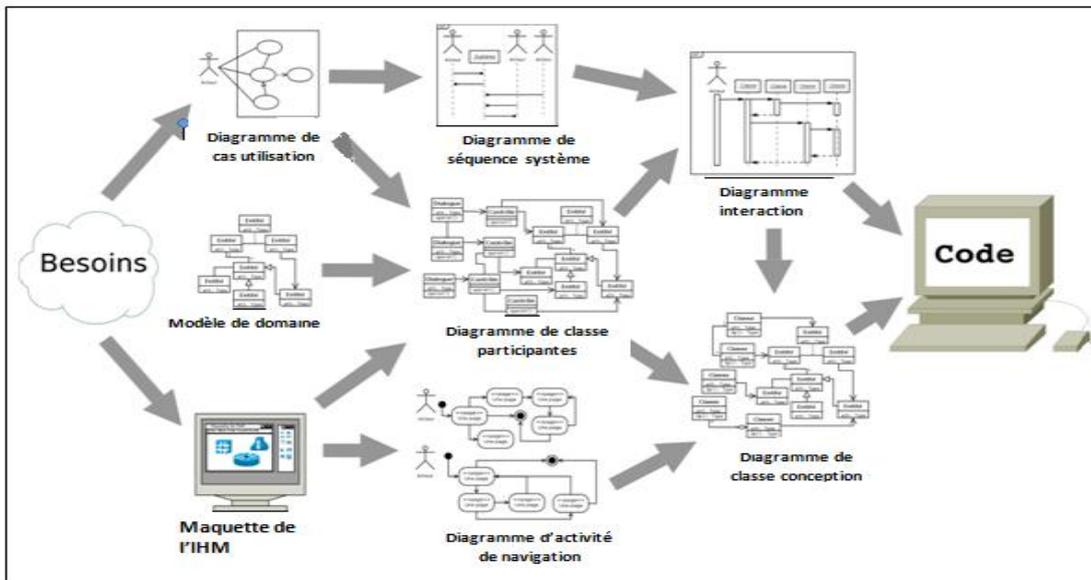


Figure 2.8: Diagramme de classes de conception [34]

2.3.2.4. Phase d'implémentation

C'est lors de cette phase que le projet est réalisé ou fabriqué, c'est-à-dire que les tâches permettant de mettre en œuvre le nouveau produit, bien ou service, sont réalisées. Dans les projets informatiques, c'est cette phase qui permet la construction du logiciel. Il faut maintenant le compléter en précisant les opérations privées des différentes classes. Il faut prendre en compte les choix techniques, comme le choix du langage de programmation, le choix des différentes bibliothèques utilisées (notamment pour l'implémentation de l'interface graphique), etc. [31]

2.4. Conclusion :

Dans ce chapitre, nous avons passé en revue le langage de modélisation UML et ses différents diagrammes.

UML est un moyen d'exprimer des modèles en faisant une abstraction de leur implémentation, i.e. que le modèle fourni par UML est valable pour n'importe quel langage de programmation. Il offre une manière de représenter le système selon différentes vues complémentaires grâce aux diagrammes.

La famille des « Unified Process » constitue une trame commune pour intégrer les Meilleures pratiques de développement. Un processus UP est interactif et incrémental, Centré sur l'architecture conduit par les exigences d'utilisateurs, piloté par les risques et orienté composant.

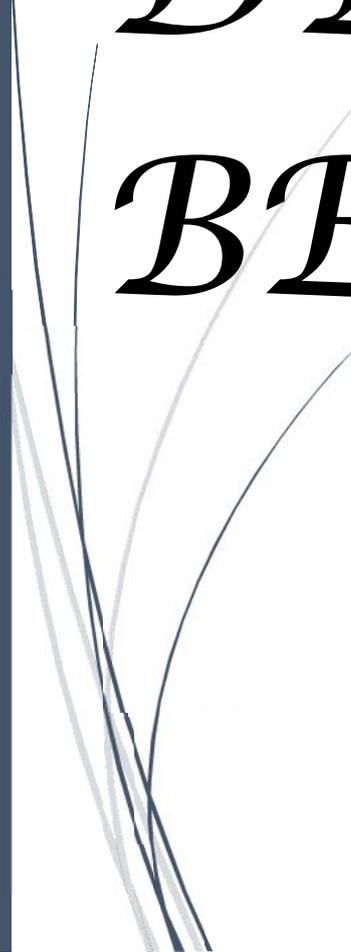
PARTIE 2

*LA
CONCEPTION
DE
L'APPLICATION*



CHAPITRE 3

*IDENTIFICATION
DES
BESOINS*



3.1 Introduction :

L'objectif de ce chapitre est de déterminer de façon détaillée et précise ce que le système devra faire, afin de répondre aux objectifs établis lors de l'étude de l'existant, tout en respectant les Contraintes établies préalablement.

3.2 Identification des cas d'utilisation :

Le tableau suivant englobe les différents CU de ce système :

N°	Cas d'utilisation	Acteur
01	Authentification	Agent administratif, client
02	Réserver une chambre	Visiteur
03	Annuler la réservation	Client, Agent administratif
04	Modifier la réservation	Client, Agent administratif
05	Enregistrer la réservation	Agent administratif
06	Modifier les tarifs	Agent administratif
07	Consulter la liste des réservations en cours	Agent administratif
08	Supprimer les réservations consommées	Agent administratif
09	Consulter le site	Visiteur

3.3 Diagramme de cas d'utilisation :

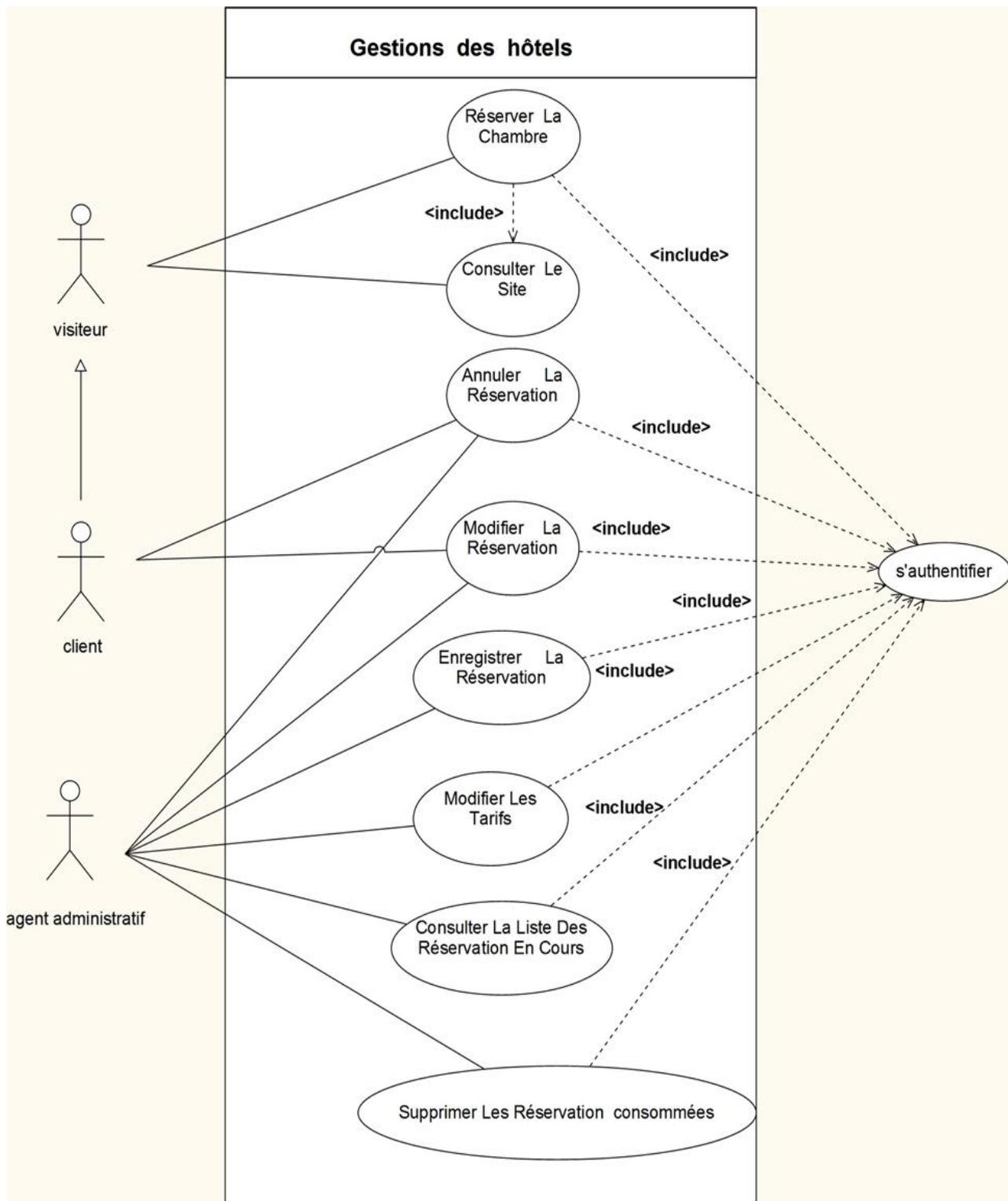


Figure 3.1: Diagramme de cas d'utilisation

3.4 Description des cas d'utilisation (fiche descriptive)

3.4.1 Cas d'utilisation « s'authentifier »

S'authentifier
Description : Permet aux acteurs (clients) de s'authentifier et d'accéder à leurs comptes respectifs.
Acteur : Client, agent d'administratif
Pré condition :
Post condition : Le système reconnaît l'utilisateur
Nominal : <ol style="list-style-type: none"> 1. Le système affiche le formulaire d'identification. 2. Le client ou l'agent d'administratif saisit l'utilisateur Name et le mot de passe. 3. Le système vérifie les données et affiche son compte.
Alternatif : <ol style="list-style-type: none"> 1. Erreurs en l'information (L'utilisateur ou le mot de passe est erroné) (retour au scénario nominal 2)
Exception : L'utilisateur quitter le système.

3.4.2 Cas d'utilisation « Réserver une chambre »

Réserver une chambre
Description : ce cas Permet aux utilisateurs de réserver une chambre
Acteurs : visiteur
Pré condition :
Post condition :
Nominal : <ol style="list-style-type: none"> 1. Le visiteur demande de consulter la liste des chambres disponibles 2. Le système affiche les chambres disponibles avec leurs informations 3. Le visiteur sélectionne une chambre 4. Le système affiche le formulaire de réservation 5. Le visiteur entre ses informations et confirme 6. Le système affiche un message de confirmation
Alternatif : <ol style="list-style-type: none"> 1. Erreurs en information (retour au scénario nominal 5) 2. Le visiteur ne respecte pas le formulaire
Exception : Le visiteur quitter le système

3.4.3 Cas d'utilisation « Annuler la réservation »

Annuler la réservation
Description : permet aux clients et l'agent administratif d'annuler la réservation
Acteur : Client, agent d'administratif
Pré condition : L'agent doit s'authentifier Réservation existe déjà
Post condition : le système afficher un message d'annulation
Nominal : <ol style="list-style-type: none"> 1. Le client ou l'agent d'administratif demandé d'annuler la réservation 2. Le système afficher le formulaire 3. Le client ou L'agent d'administratif confirmer l'annulation 4. Le système afficher un message d'annulation
Alternatif :
Exception : L'utilisateur quitter le système

3.4.4 Cas d'utilisation « Modifier la réservation »

Modifier la réservation
Description : permet aux clients de modifier la réservation
Acteur : Client, agent d'administratif
Pré condition : L'utilisateur doit s'authentifier Réservation existe déjà
Post condition : la réservation est modifiée
Nominal : <ol style="list-style-type: none"> 1. Le client ou l'agent d'administratif demander de modifier la réservation 2. Le système afficher le formulaire de la modification 3. Le client ou l'agent d'administratif remplir le formulaire et confirmer la modification 4. Le système afficher un message de modification
3. Alternatif : erreur en l'information (retour au scenario nominal 3)
Exception : L'utilisateur quitter le système

3.4.5 Cas d'utilisation « Enregistrer réservation »

Enregistrer réservation
Description :
Acteur : agent d'administratif
Pré condition : le client doit réserver une chambre L'agent doit s'authentifier
Post condition : réservation une chambre
Nominal : <ol style="list-style-type: none"> 1. L'agent d'administratif choisir d'enregistrer une réservation 2. Le système afficher le formulaire 3. L'agent remplir le formulaire et confirmer 4. Le système afficher un message d'enregistrement
Alternatif : erreur en l'information (retour au scenario nominal 3)

3.4.6 Cas d'utilisation « modifier les tarifs »

Modifier les tarifs
Description : ce cas permet à l'agent administratif de demander la modification des tarifs des chambres
Acteur : agent d'administratif
Pré condition : L'agent doit s'authentifier
Post condition : les tarifs des chambres modifiés
Nominal : <ol style="list-style-type: none"> 1. L'agent d'administratif demander de modification des tarifs 2. Le système afficher le formulaire 3. L'agent d'administratif saisir les nouveaux tarifs et les confirmer 4. Le système afficher un message de confirmation
Alternatif : erreur (retour au scenario nominal 3)
Exception :

3.4.7 Cas d'utilisation « consulter la liste des réservation en cours »

Consulter la liste des réservations en cours
Description : ce cas permet à l'agent d'administratif de consulter la liste des réservations en cours
Acteur : agent d'administratif
Pré condition : L'agent doit s'authentifier
Post condition :
Nominal : <ol style="list-style-type: none"> 1. L'agent d'administratif choisir de consulter la liste des réservations en cours 2. Le système afficher la liste
Alternatif :
Exception :

3.4.8 Cas d'utilisation « Supprimer les réservations consommées »

Supprimer les réservations consommées
Description : ce cas permet à l'agent administratif de supprimer des réservations consommées
Acteurs : agent d'administratif
Pré condition : la réservation doit consommée L'agent doit s'authentifier
Post condition : une réservation sera supprimée de la liste des réservations
Nominal : <ol style="list-style-type: none"> 1. L'agent d'administratif choisir de supprimer une réservation consommée 2. Le système afficher un formulaire de saisir 3. L'agent d'administratif confirmer la suppression 4. Le système afficher un message de confirmation
Alternatif :

Exception :

3.4.9 Cas d'utilisation « consulter le site »

Consulter le site
Description : permet aux visiteurs de consulter le site
Acteur : visiteur
Pré condition :
Post condition : le visiteur consulter le site
Nominal : 1. le visiteur consulter les informations d'hôtel et des services disponible. 2. le système afficher les informations d'hôtel et les services disponibles.
Alternatif :
Exception : Le visiteur quitter le système

3.5 Les diagrammes de séquence système :

3.5.1 S'authentifie

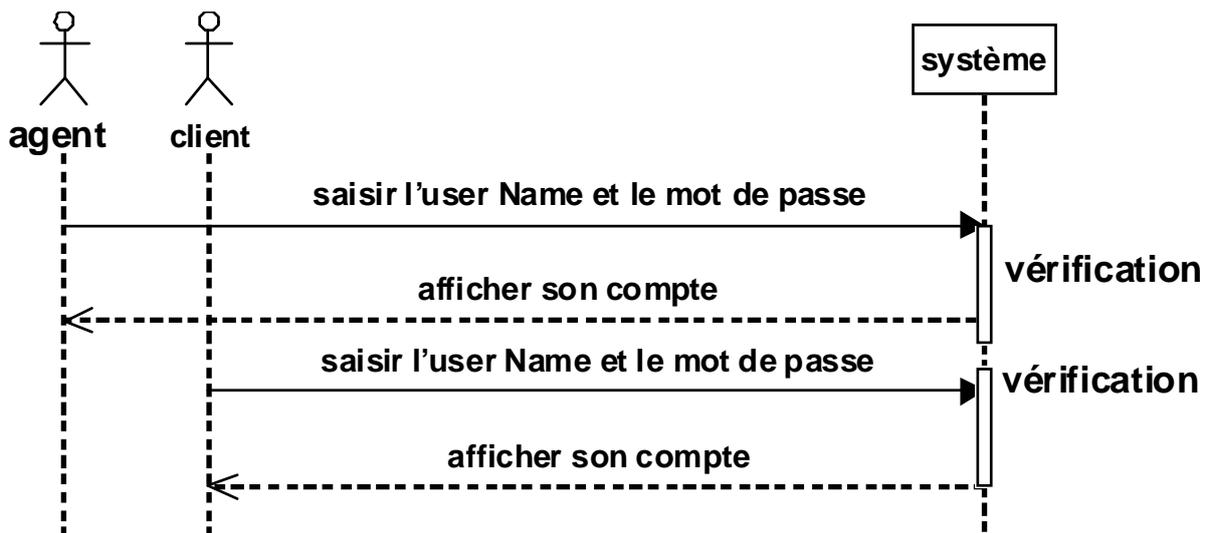


Figure 3.2: Diagramme de séquence système cas d'utilisation « s'authentifier »

3.5.2 Réserver une chambre

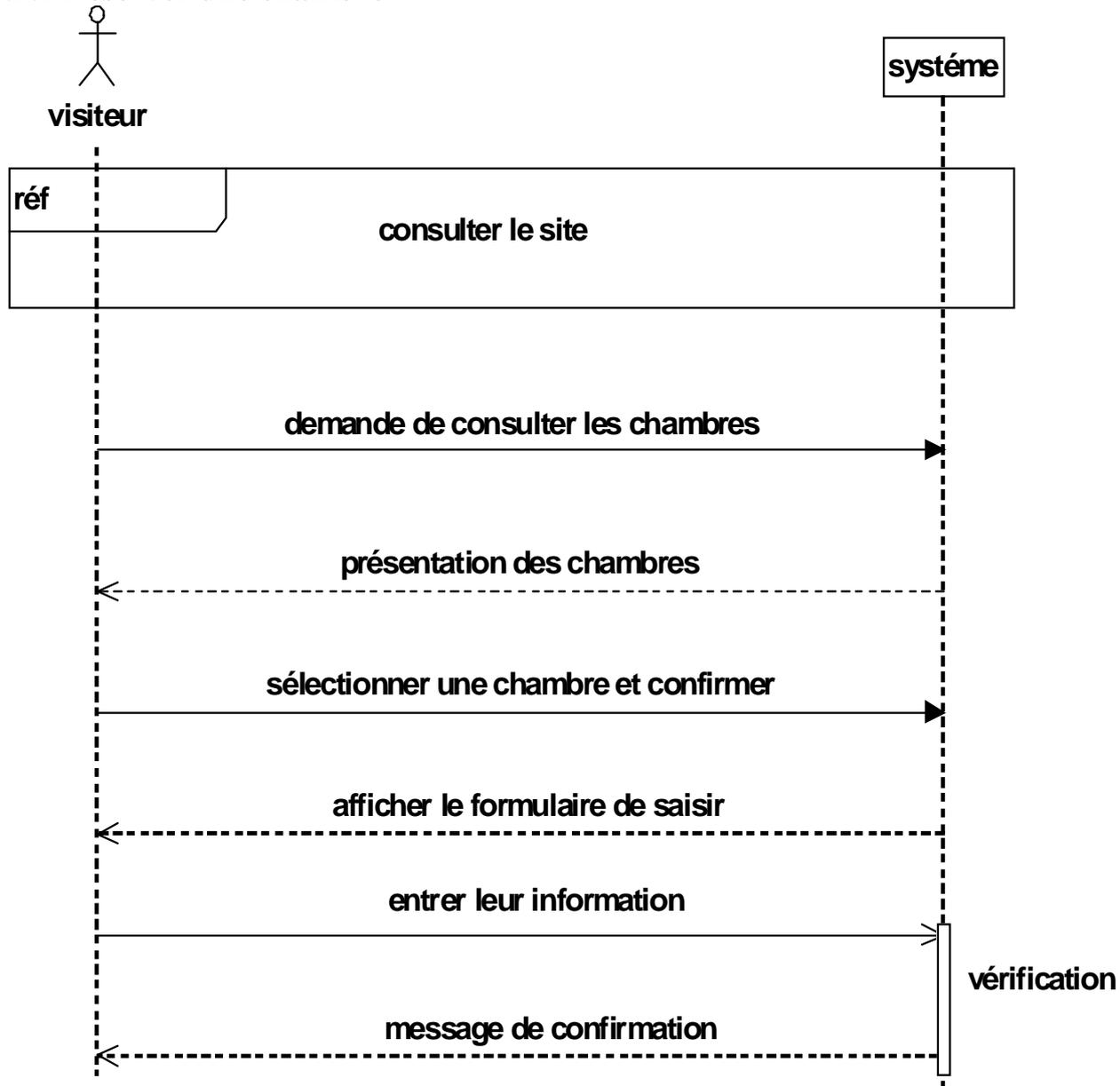


Figure 3.3: Diagramme de séquence système cas d'utilisation « Réserver une chambre »

3.5.3 Annuler la réservation :

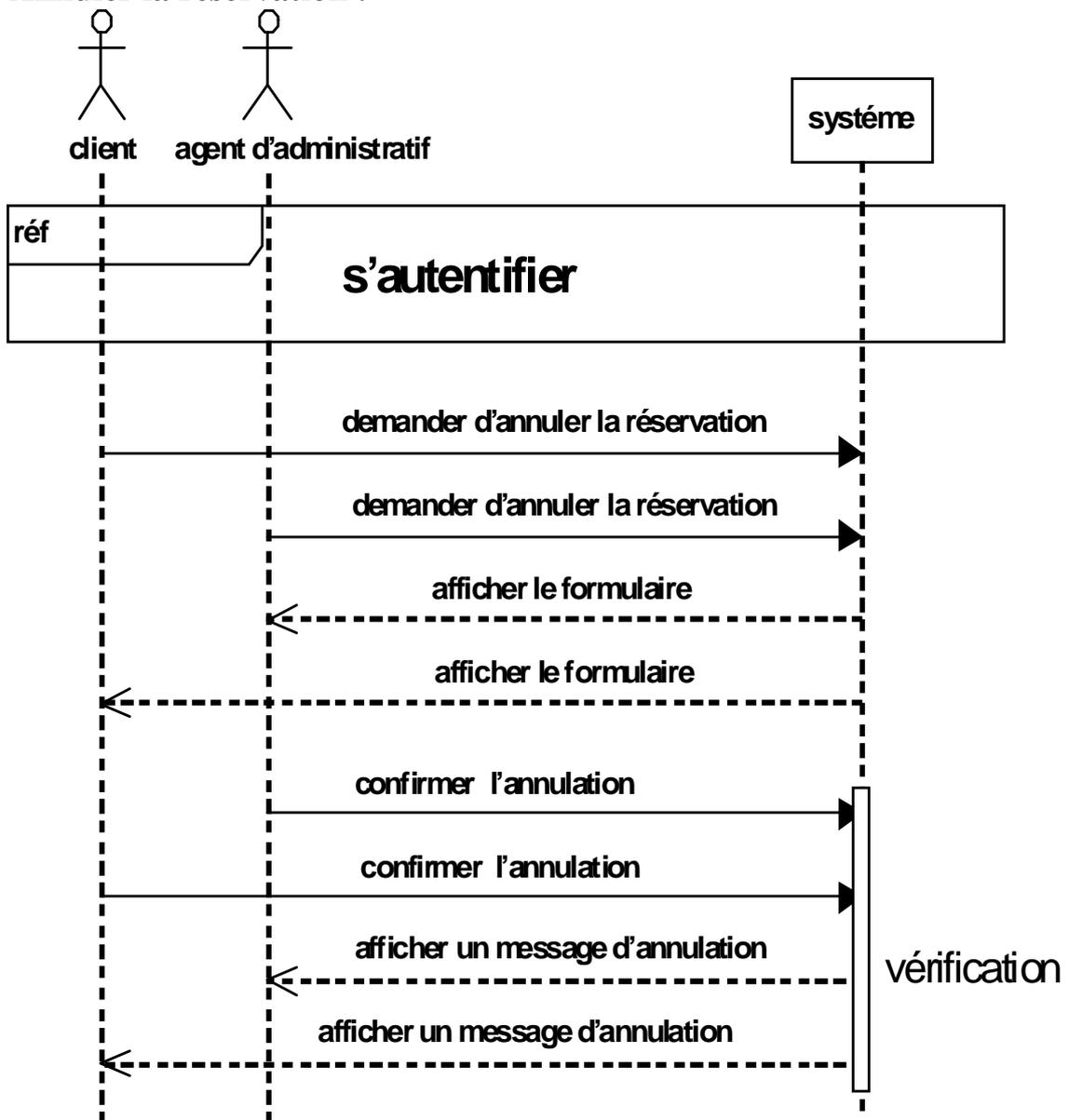


Figure 3.4: Diagramme de séquence système cas d'utilisation « Annuler la réservation »

3.5.4 Modifier la réservation :

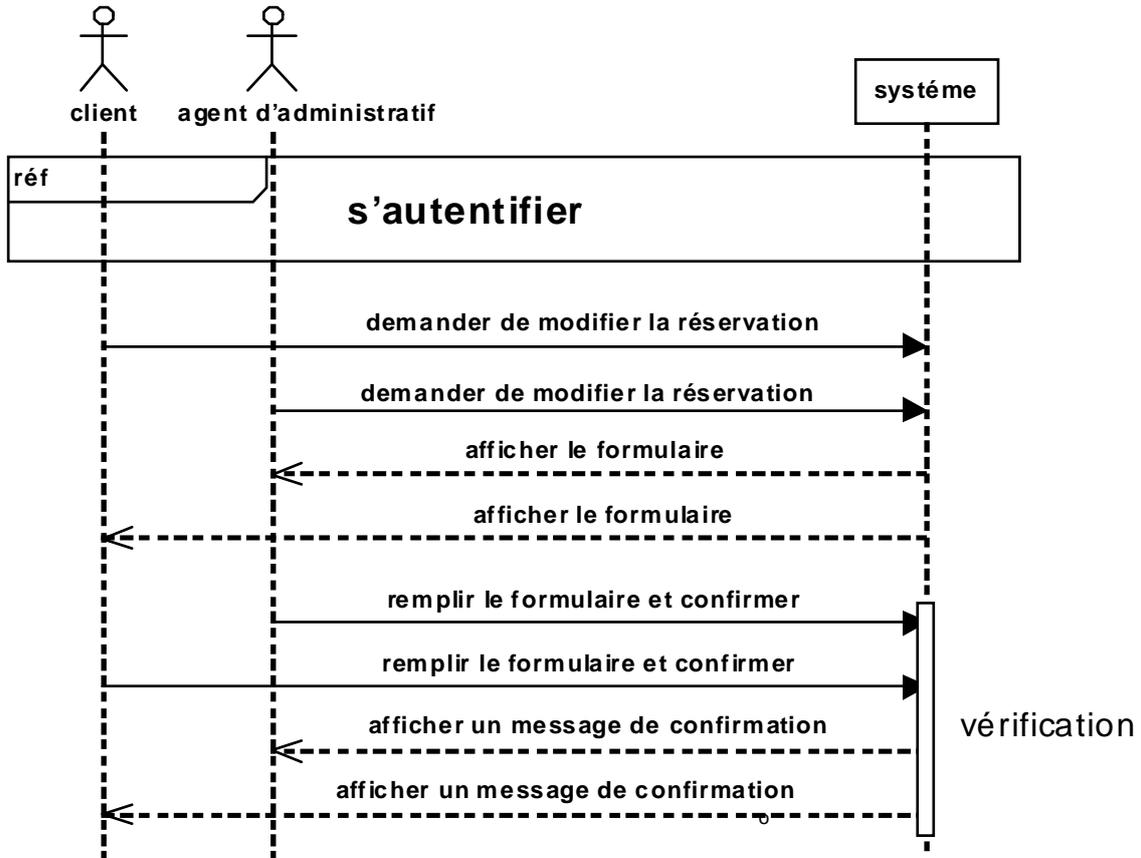


Figure 3.5: Diagramme de séquence système cas d'utilisation « Modifier la réservation »

3.5.5 Enregistrer la réservation :

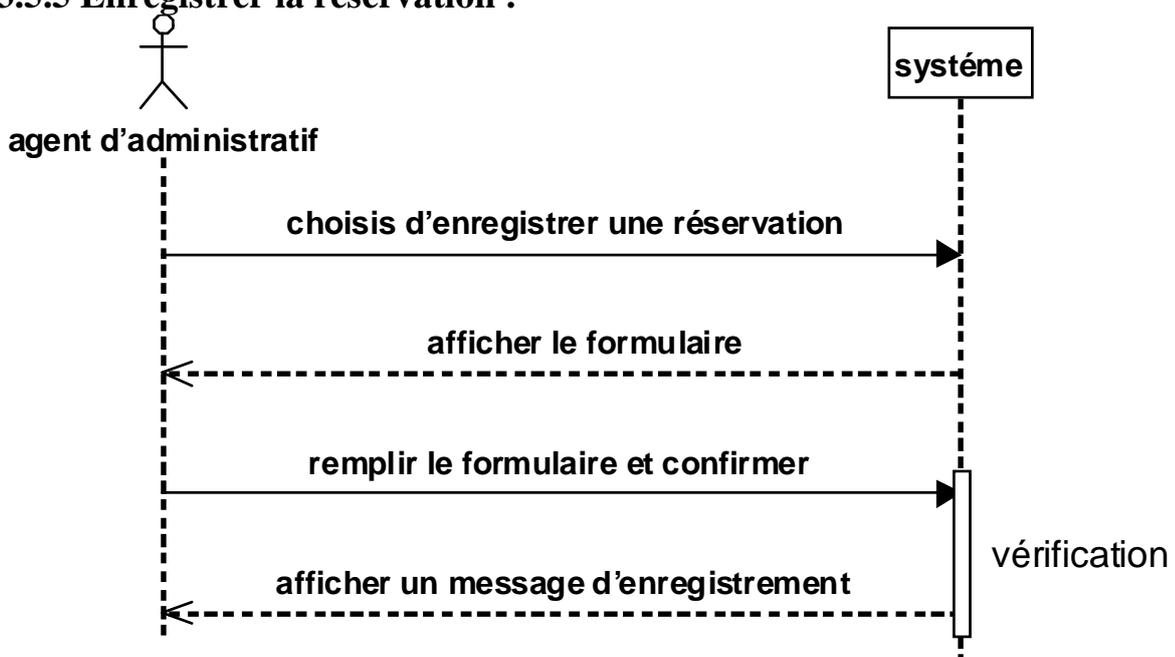


Figure 3.6: Diagramme de séquence système cas d'utilisation «Enregistrer la réservation»

Figure

3.5.6 Modifier les tarifs :

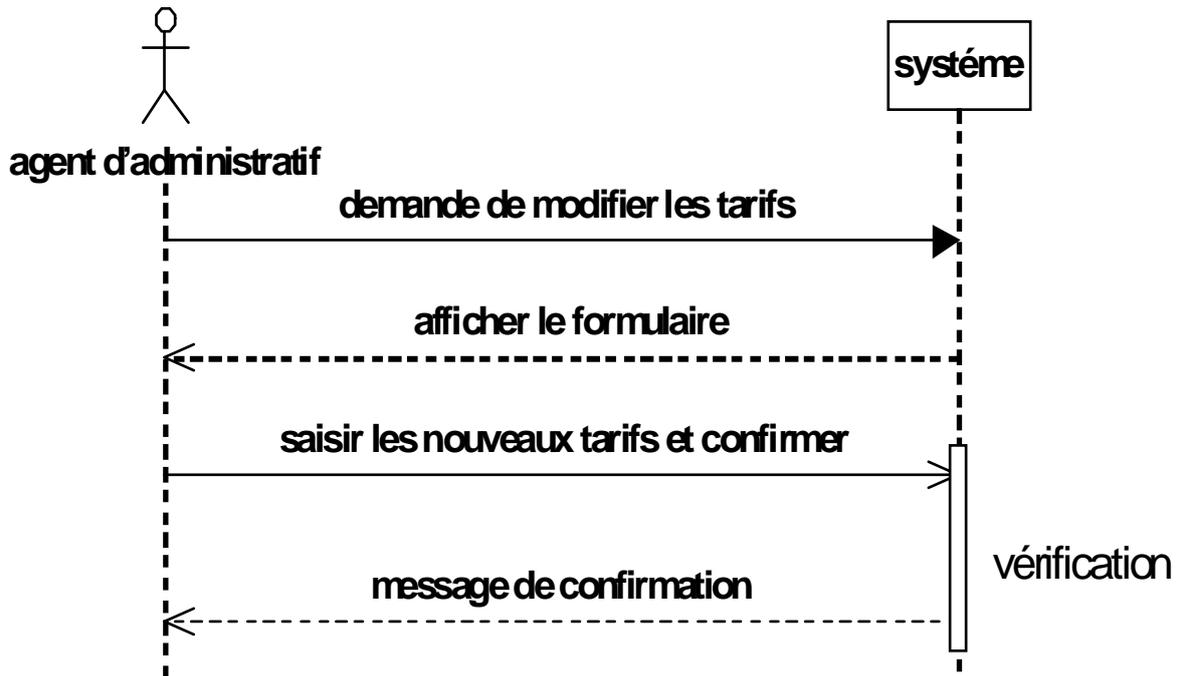


Figure 3.7: Diagramme de séquence système cas d'utilisation «Modifier les tarifs»

3.5.7 Consulter les listes des réservations en cours :

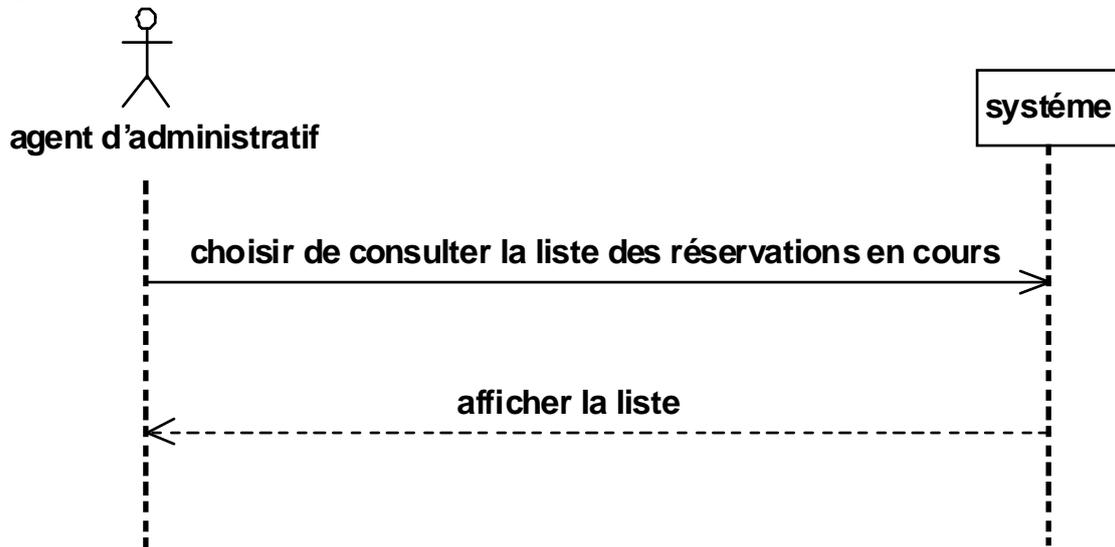


Figure 3.8: Diagramme de séquence système cas d'utilisation «Consulter les listes des réservations en cours»

3.5.8 Supprimer les réservations consommées :

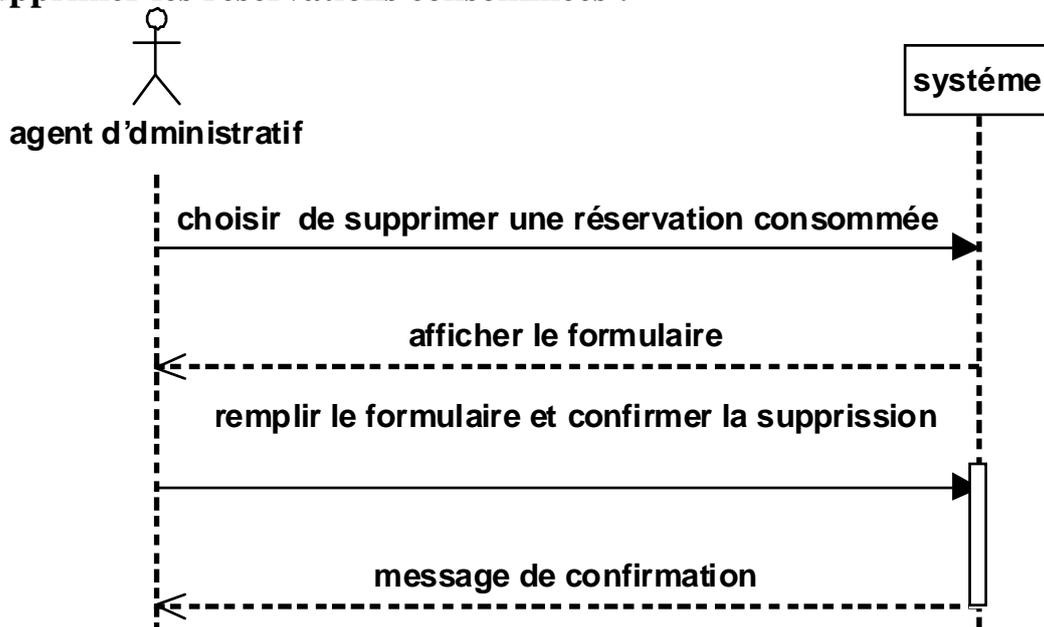


Figure 3.9: Diagramme de séquence système cas d'utilisation «Supprimer les réservations consommées»

3.5.9 Consulter le site :

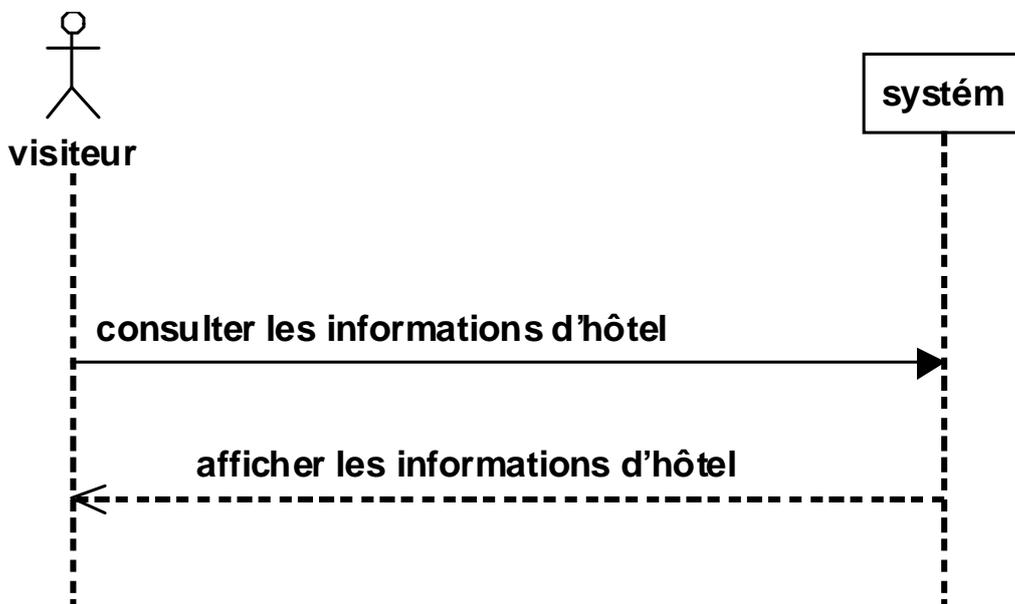


Figure 3.10: Diagramme de séquence système cas d'utilisation « Consulter le site »

3.6 Conclusion :

La phase d'identification des besoins nous a permis de modéliser le contexte du système considéré comme une boîte noire en:

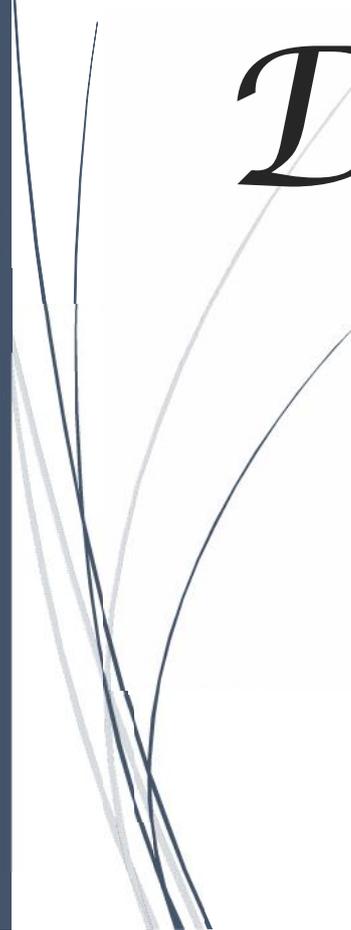
- Répertorient les interactions entre acteurs et système.
- Représentant les interactions sur le diagramme de contexte dynamique.

Ainsi l'identification de besoin nous a permis aussi de modéliser les cas d'utilisations et Les diagrammes de séquences système en prennent en considération que le système a toujours Vu comme une boîte noire. Ensuite cette phase prépare la phase d'analyse qui est l'objet de chapitre suivant.



CHAPITRE 4

*L'ANALYSE
DU
DOMAINE*



4.1 Introduction:

Dans ce chapitre nous allons élaborer du diagramme de classes qu'est le modèle de classes du domaine. Puis, nous développerons les diagrammes de classes participantes qui effectuent la jonction entre, d'une part, les cas d'utilisation, le modèle du domaine et la maquette, et d'autre part, les diagrammes de conception logicielle que sont les diagrammes de séquence et le diagramme de classes de conception, enfin, nous représenterons l'activité de navigation dans l'interface du future application en produisant des diagrammes d'activités de navigation.

4.2 analyse du domaine:

La conception objet demande principalement une description structurelle, statique, du système à réaliser, sous forme d'un ensemble de classes logicielles. Les classes candidats sont celle issues d'une analyse de domaine, effectivement ces concepts (souvent appelés objets métier) peuvent être identifié directement à partir de la connaissance du domaine.

Détaillons les différentes étapes de la démarche que nous allons suivre :

- Identifier les concepts du domaine ;
- Ajouter les associations entre classes ;
- Ajouter les attributs

4.3 Identification des concepts de domaine :

- **Visiteur:**

Réserver une chambre

Visiter le site

- **Client :**

S'authentifier

Annuler la réservation

Modifier la réservation

- **Agent administratif**

S'authentifier

Annuler la réservation

Modifier la réservation

Enregistrer la réservation

Modifier les tarifs

Consulter la liste des réservations en cours.

Supprimer les réservations consommées.

4.4 Diagramme de domaine :

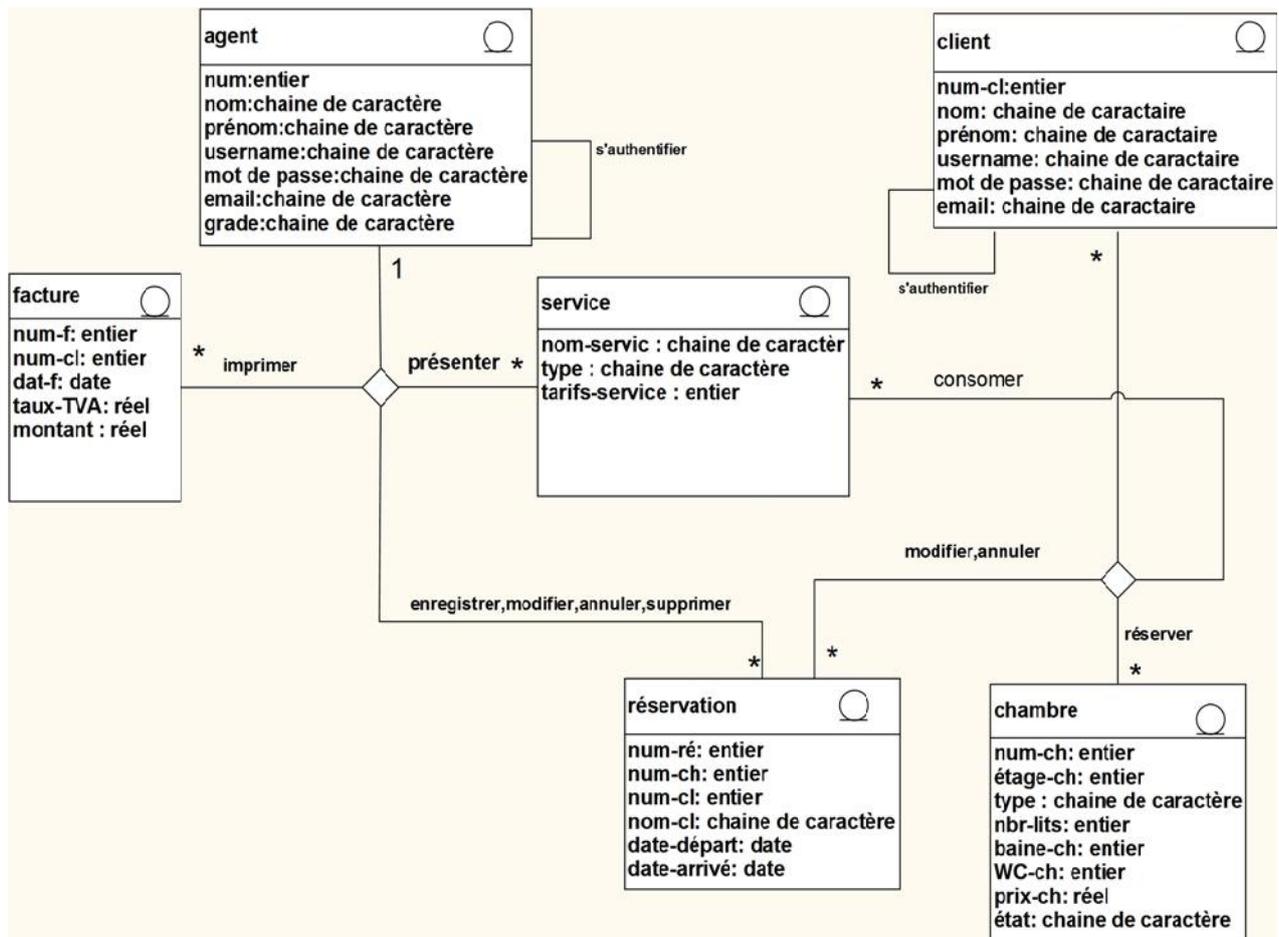


Figure 4.1 diagramme de domaine

4.5 Les diagrammes de classes participantes :

4.5.1 S'authentifier :

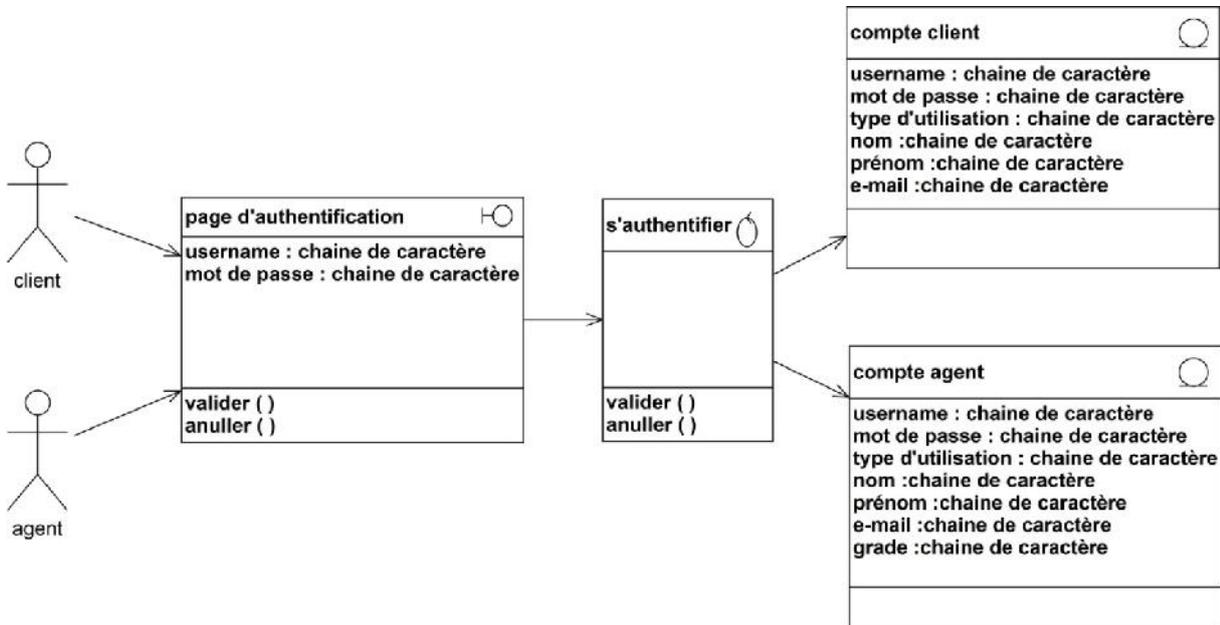


Figure 4.2 diagramme de classe participantes (s'authentifier)

4.5.2 Réserver une chambre :

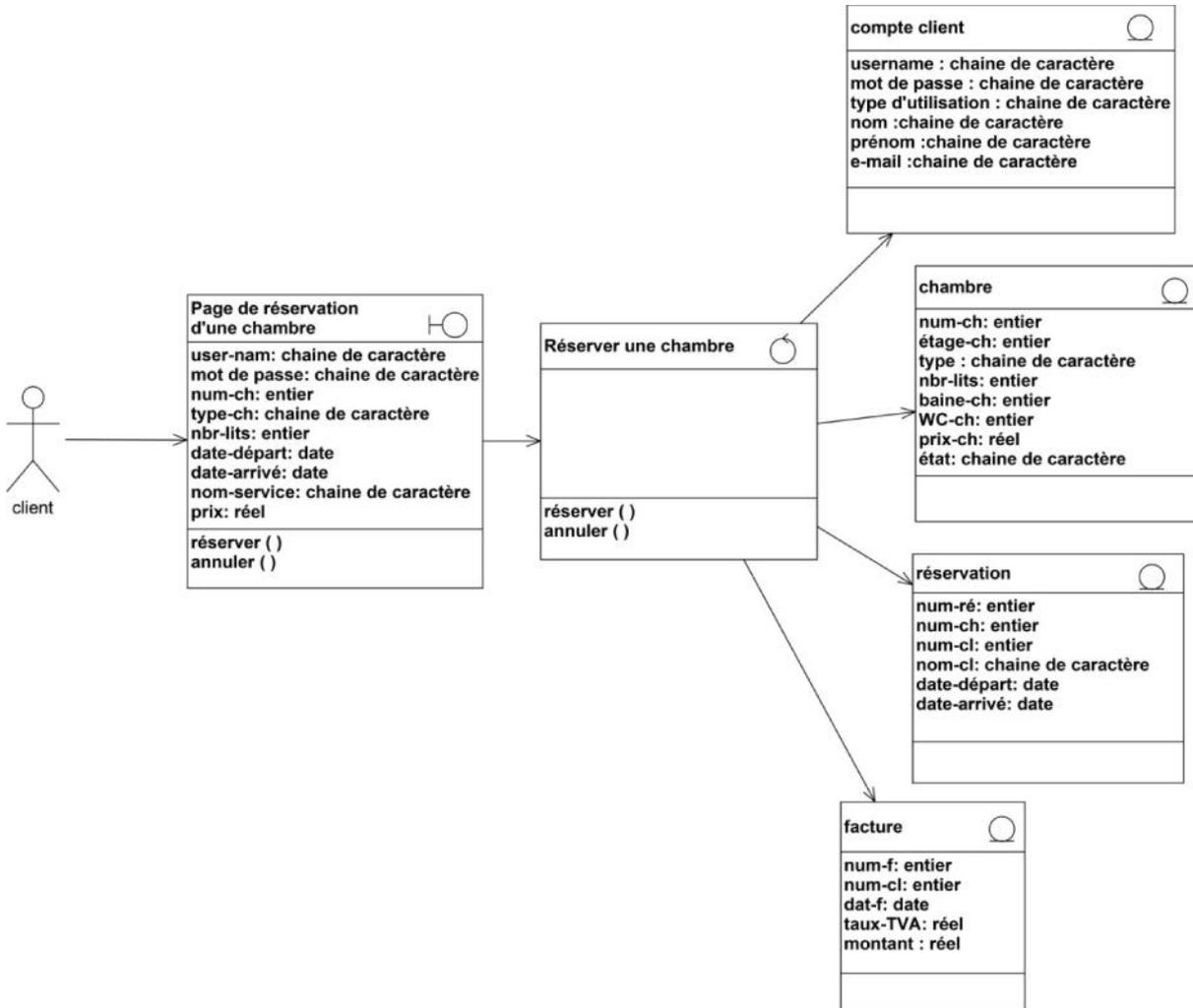


Figure 4.3 diagramme de classe participantes (réserver une chambre)

4.5.3 Annuler la réservation :

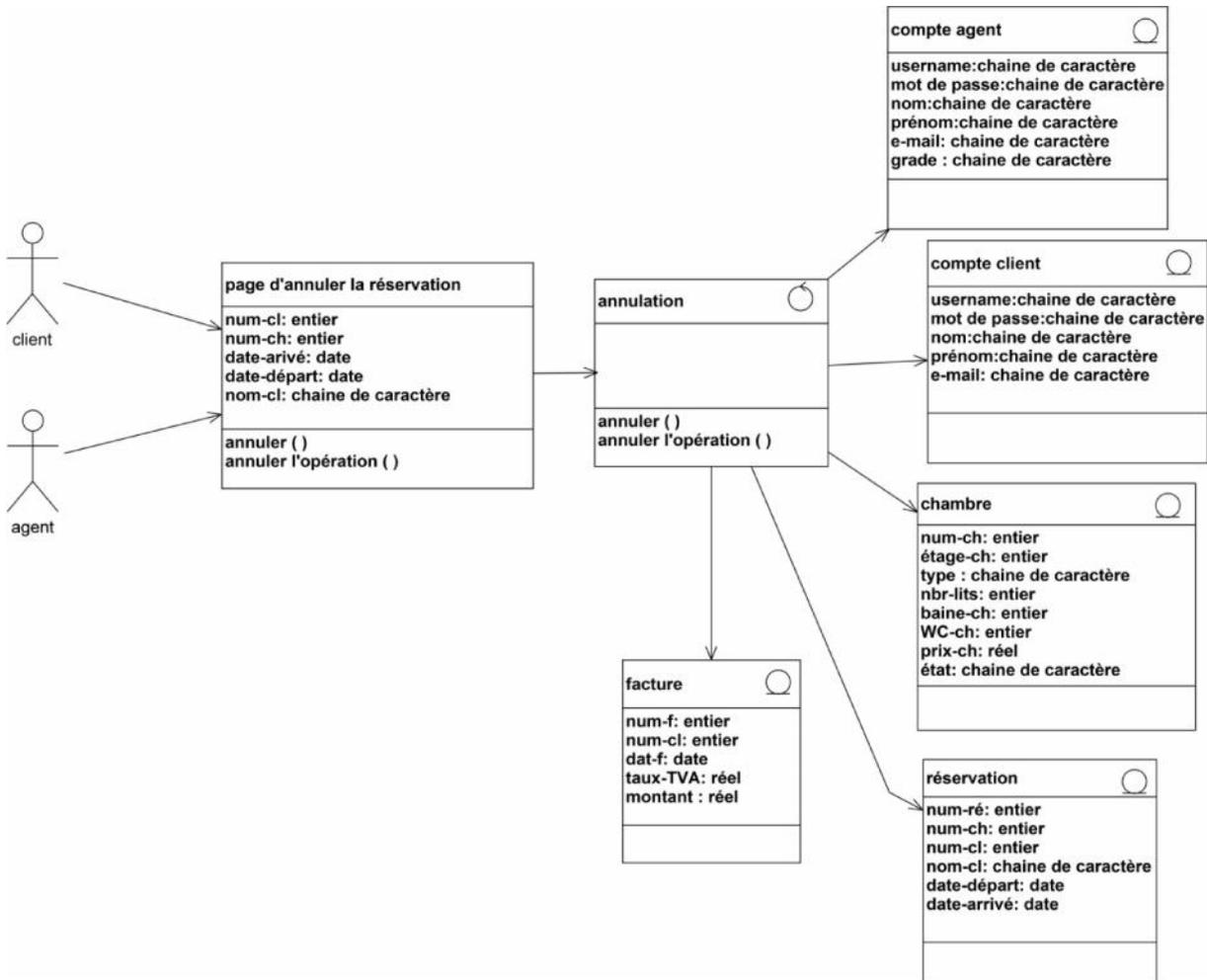


Figure 4.4 diagramme de classe participantes (annuler la réservation)

4.5.4 Modifier la réservation :

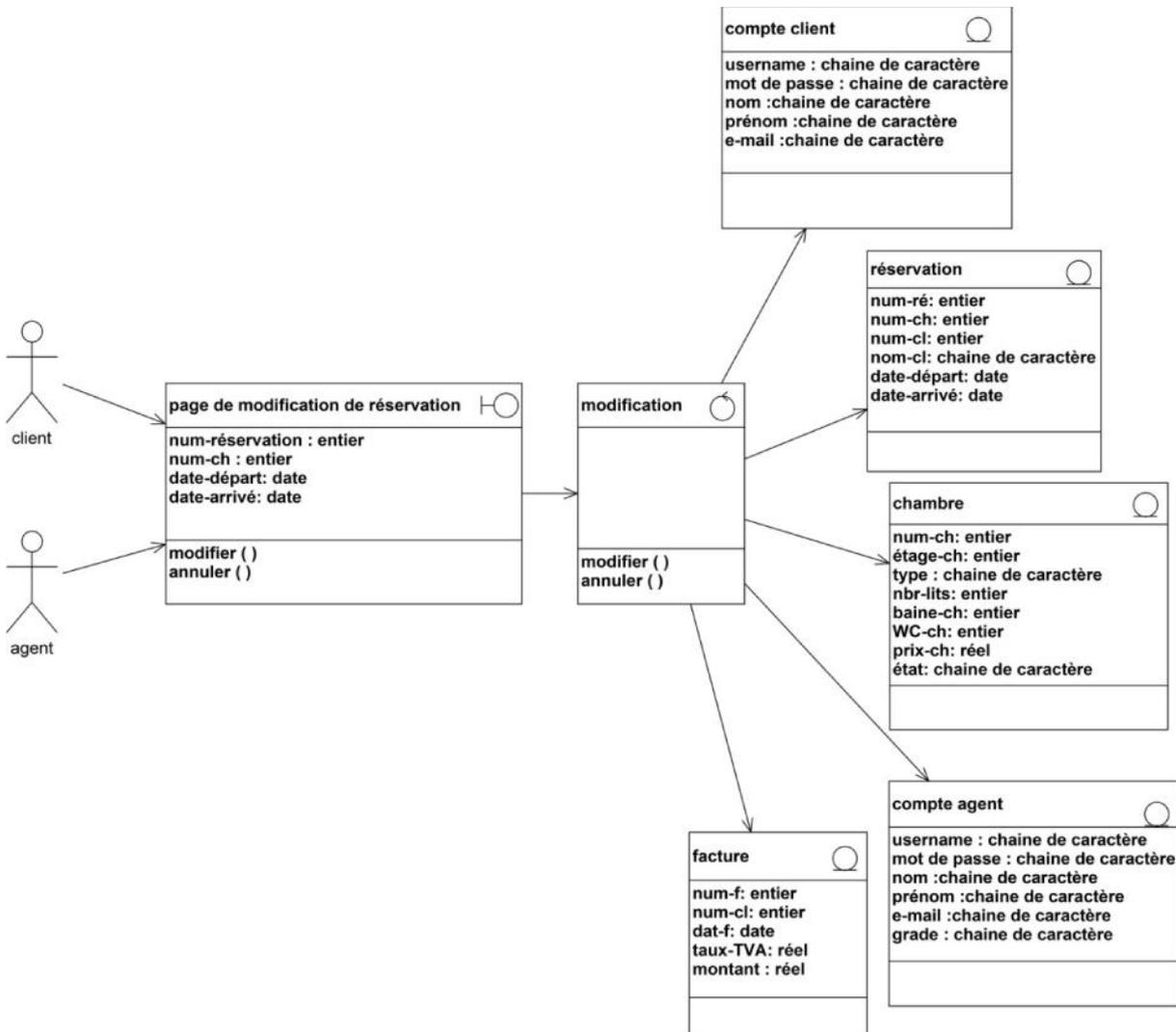


Figure 4.5 diagramme de classe participantes (modifier la réservation)

4.5.5 Enregistrer la réservation :

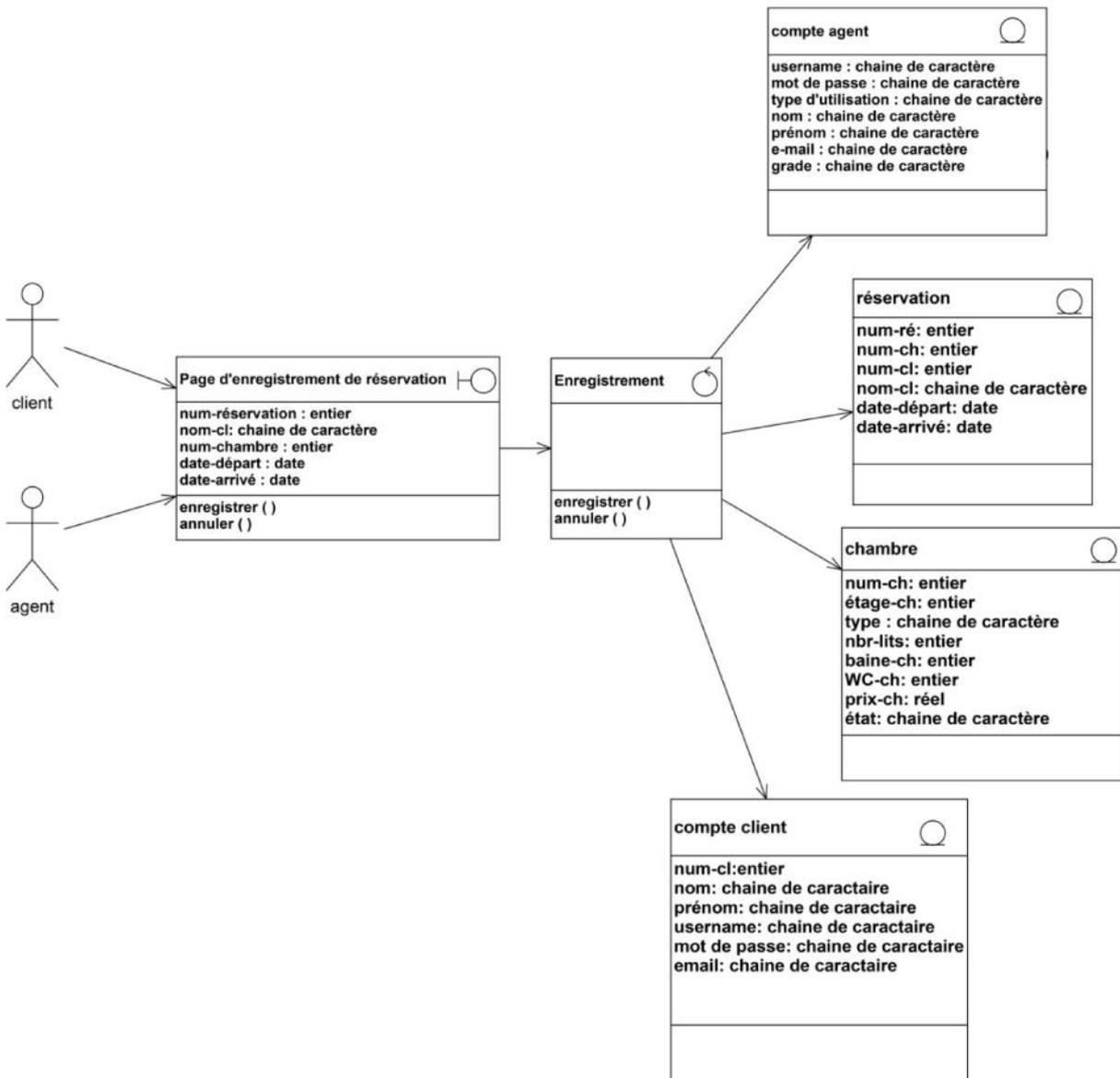


Figure 4.6 diagramme de classe participantes (enregistrer la réservation)

4.5.6 Modifier les tarifs :

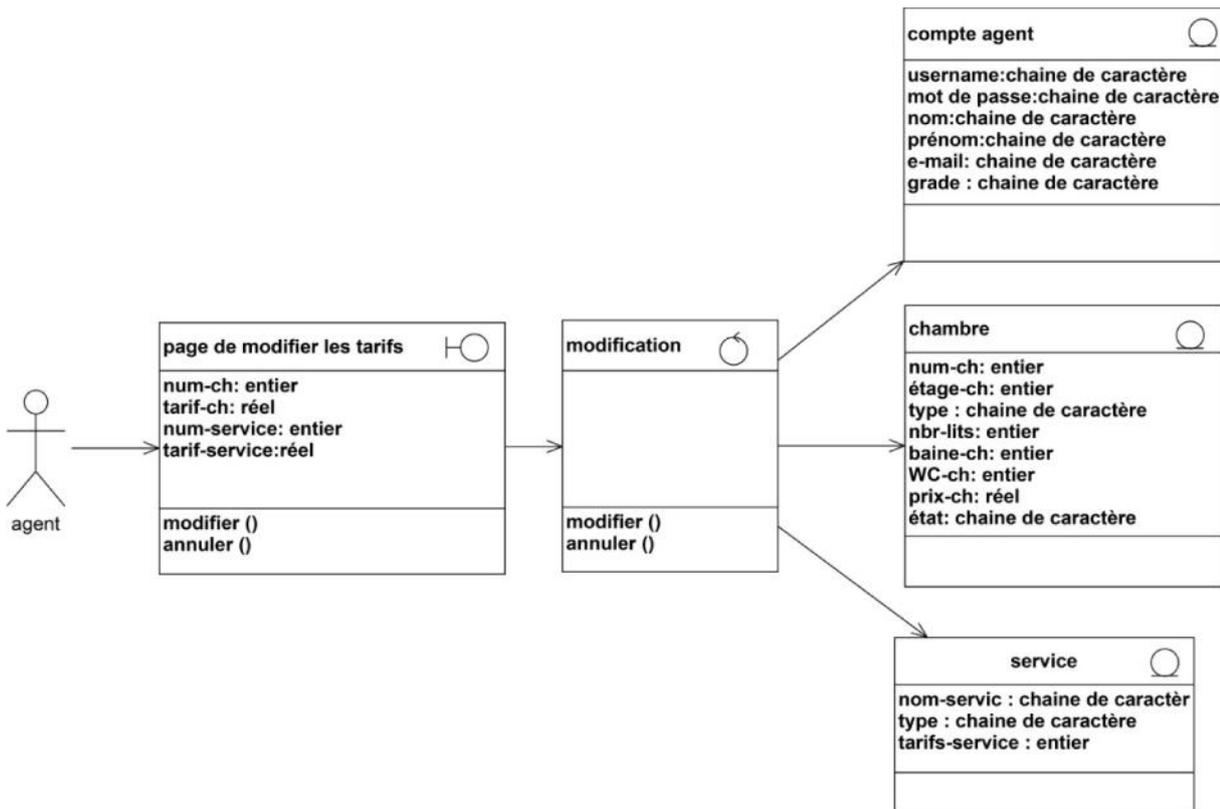


Figure 4.7 diagramme de classe participantes (modifier les tarifs)

4.5.7 Consulter la liste des réservations en cours :

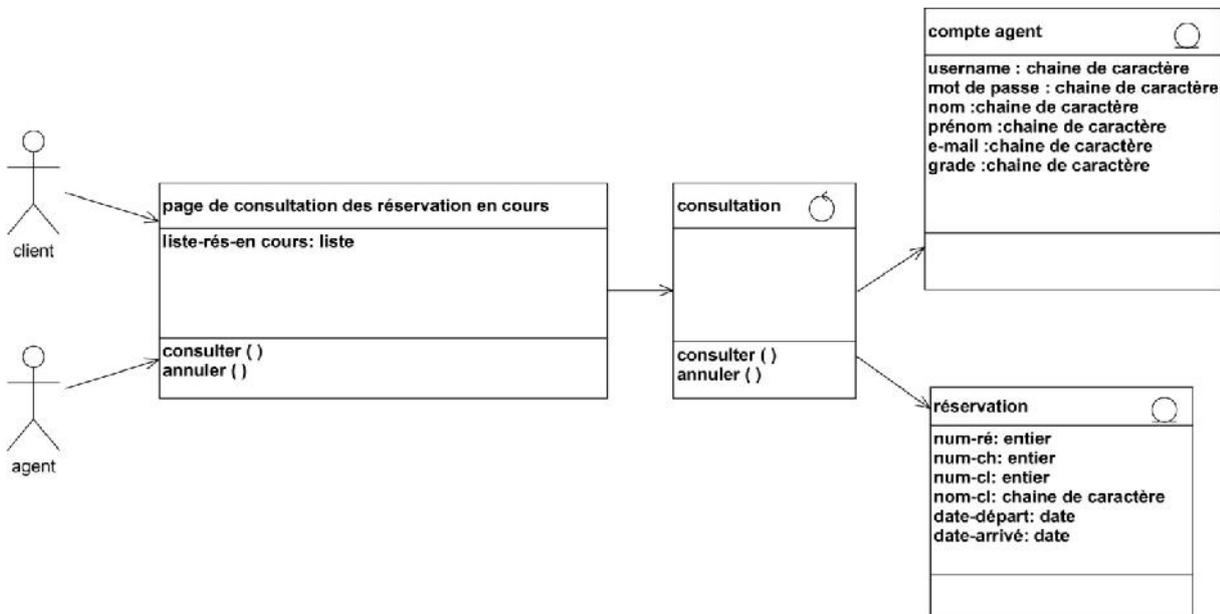


Figure 4.8 diagramme de classe participantes (consulter la liste des réservations en cours)

4.5.8 Supprimer les réservations consommées :

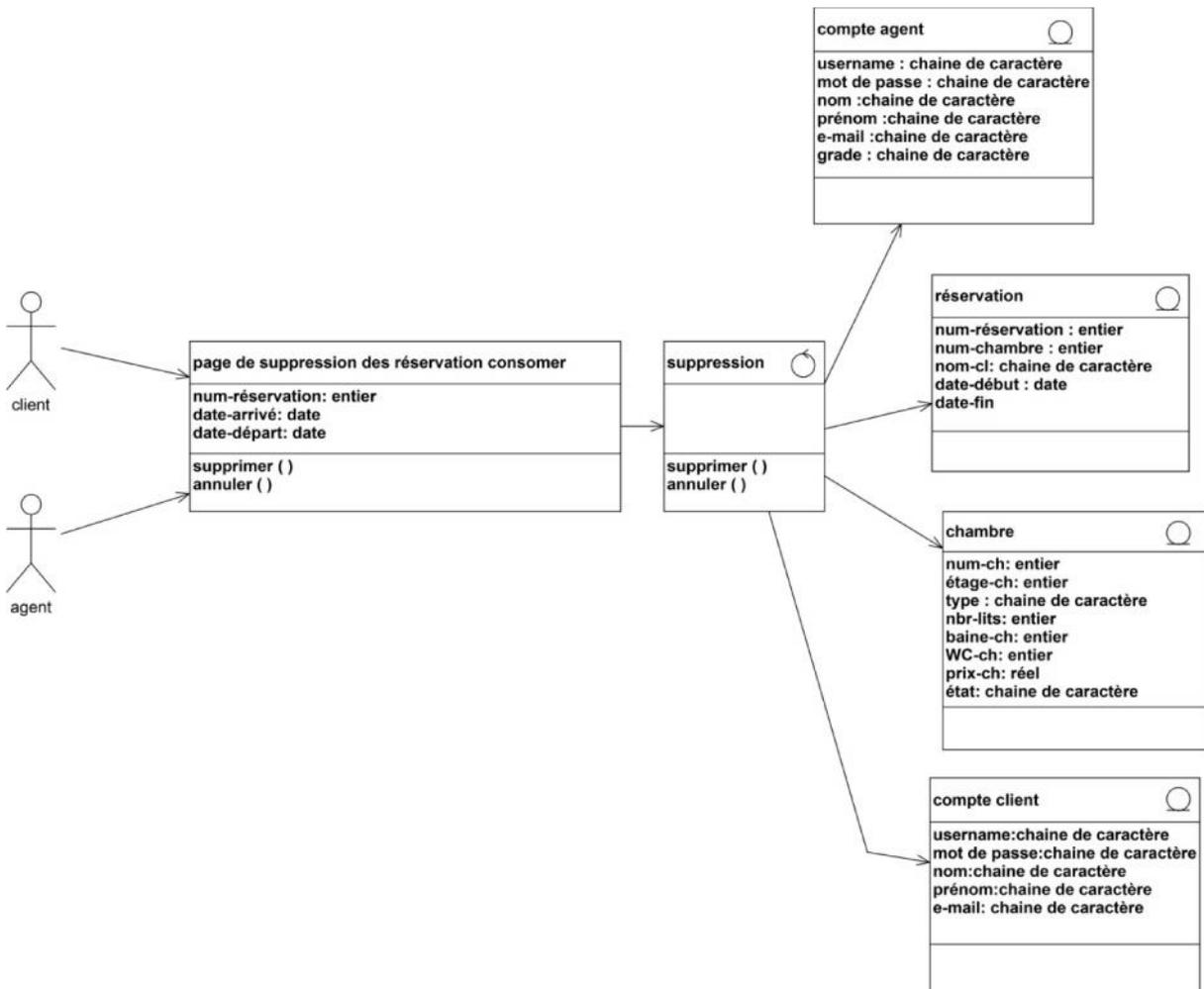


Figure 4.9 diagramme de classe participantes (supprimer les réservations consommées)

4.6 Diagrammes d'activités de navigation :

4.6.1 S'authentifier :

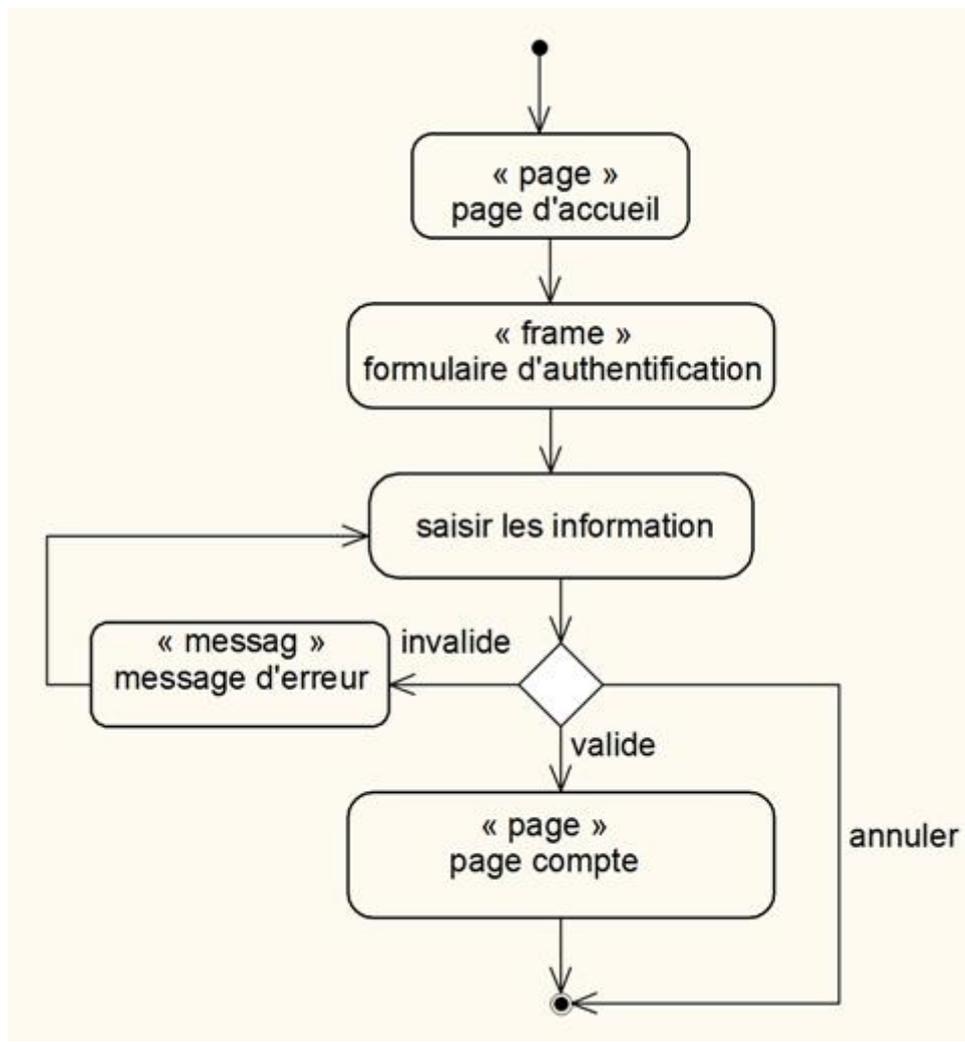


Figure 4.10 diagramme d'activités de navigation (s'authentifier)

4.6.2 Réserver une chambre :

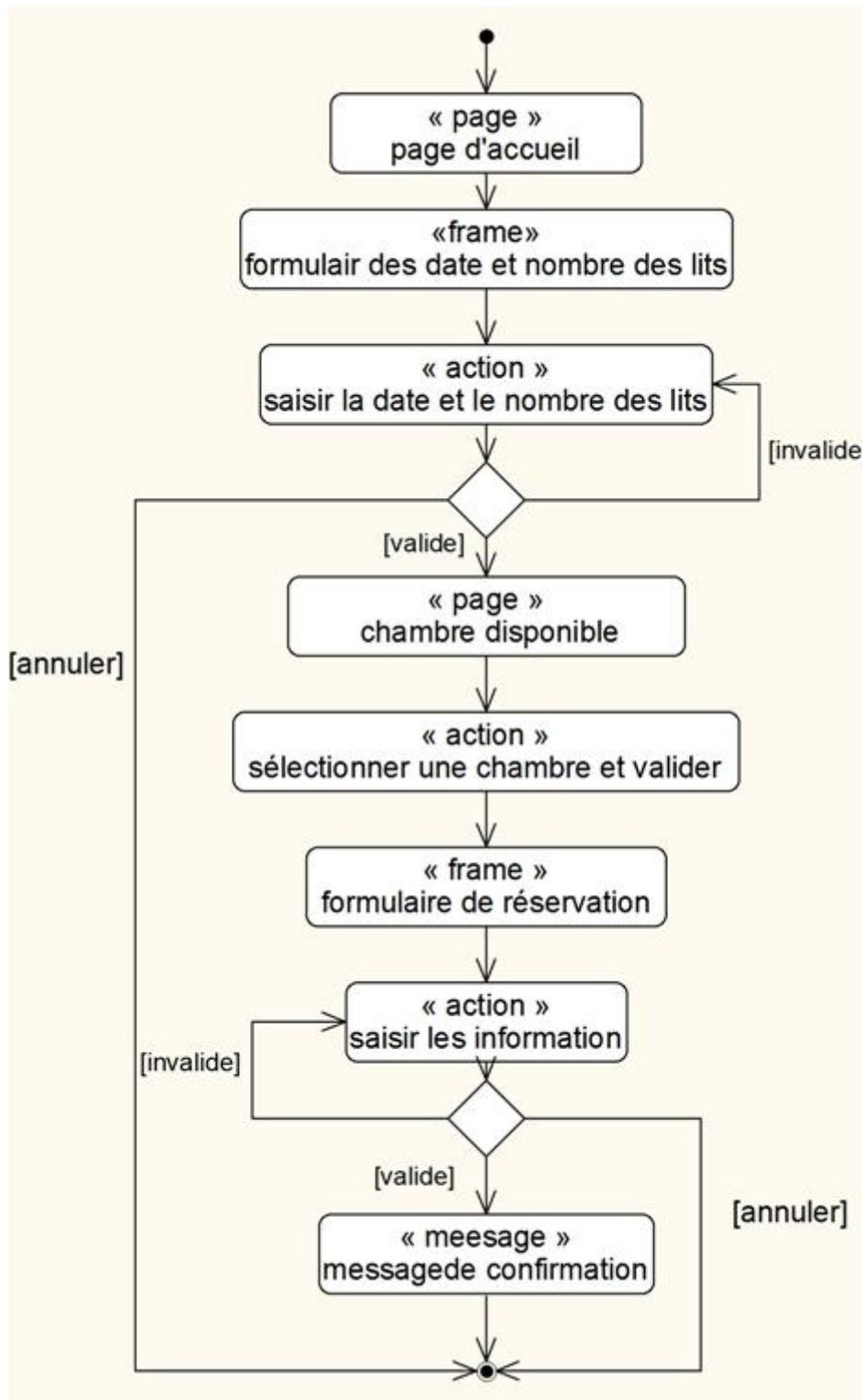


Figure 4.11 diagramme d'activités de navigation (réserver une chambre)

4.6.3 Annuler la réservation :

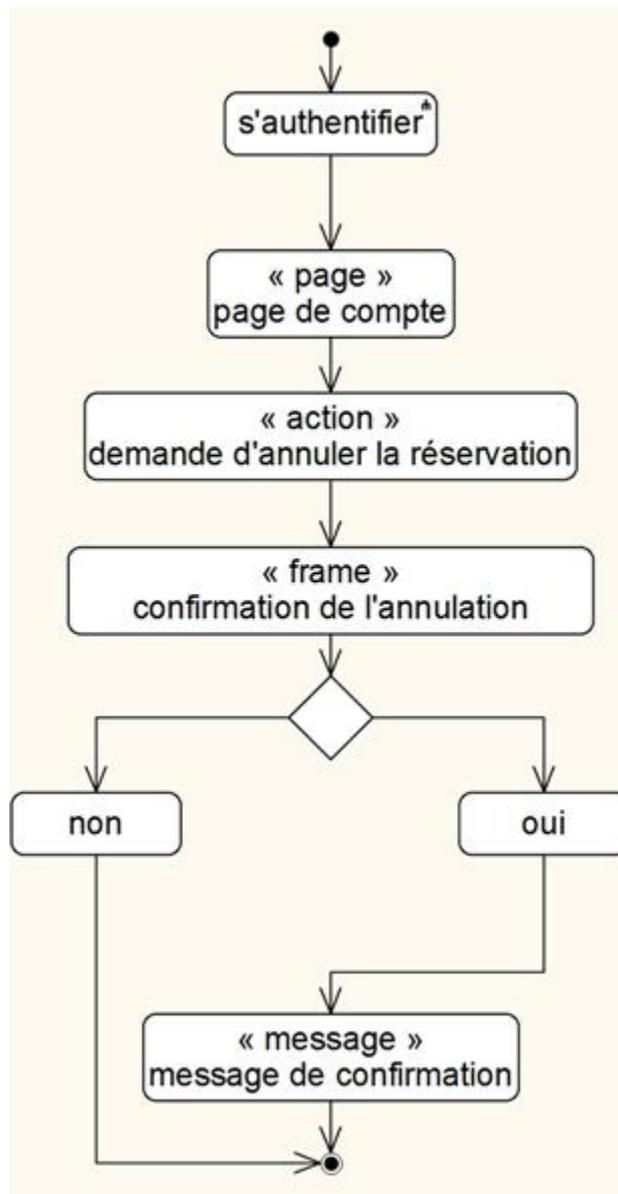


Figure 4.12 diagramme d'activités de navigation (annuler la réservation)

4.6.4 Modifier la réservation :

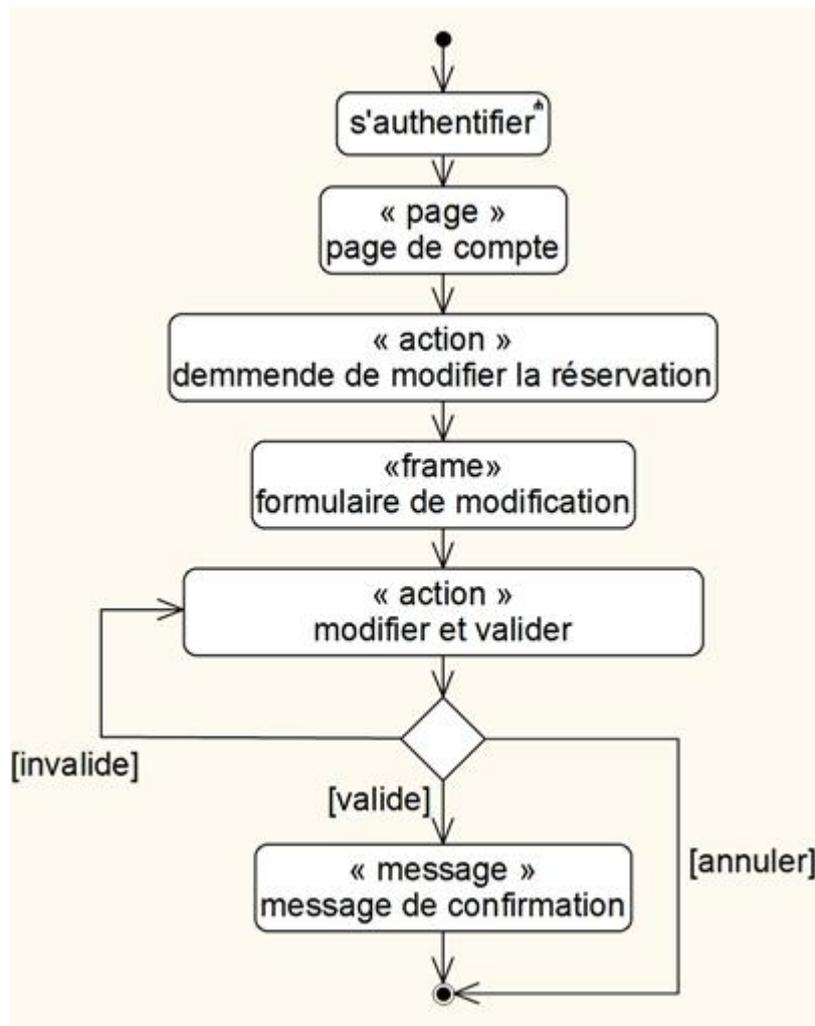


Figure 4.13 diagramme d'activités de navigation (modifier la réservation)

4.6.5 Enregistrer la réservation :

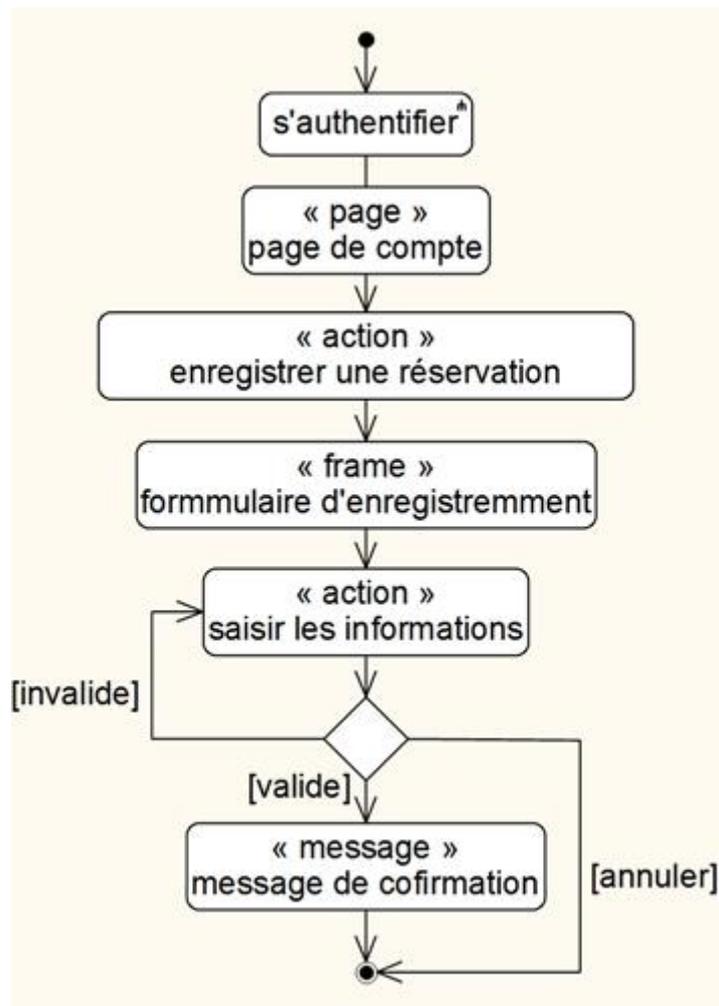


Figure 4.14 diagramme d'activités de navigation (enregistrer la réservation)

4.6.6 modifier les tarifs

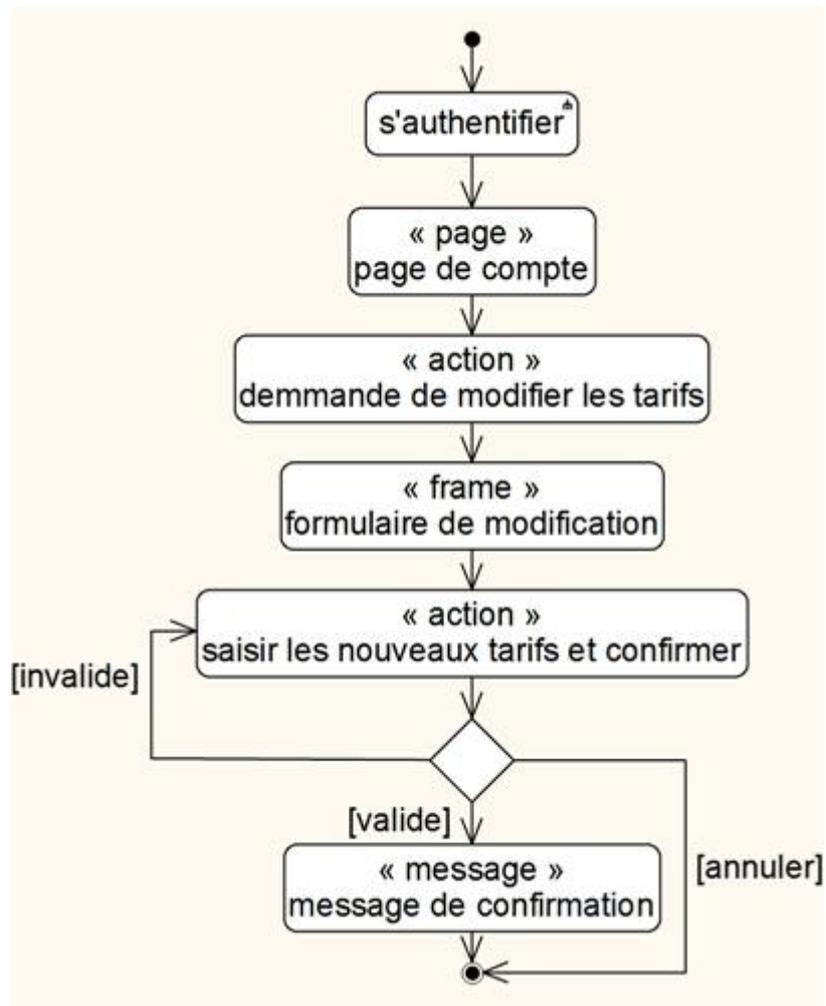


Figure 4.15 diagramme d'activités de navigation (modifier les tarifs)

4.6.7 Consulter la liste des réservations en cours :

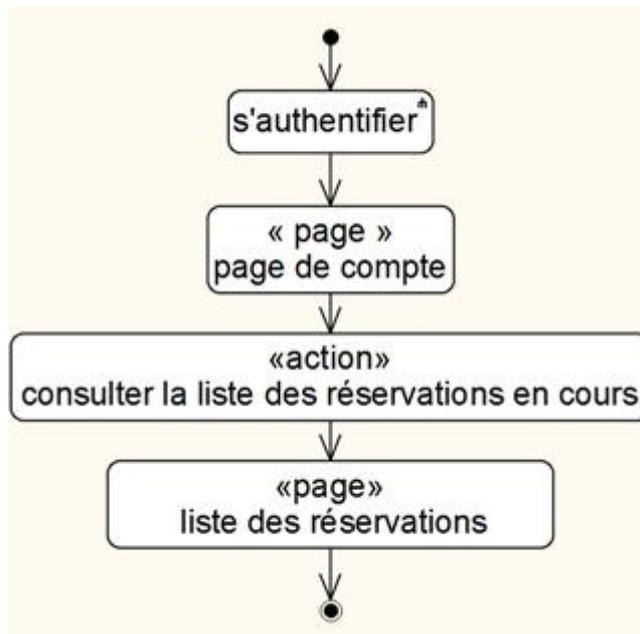


Figure 4.16 diagramme d'activités de navigation (consulter la liste des réservations en cours)

4.6.8 Supprimer la réservation consommée :

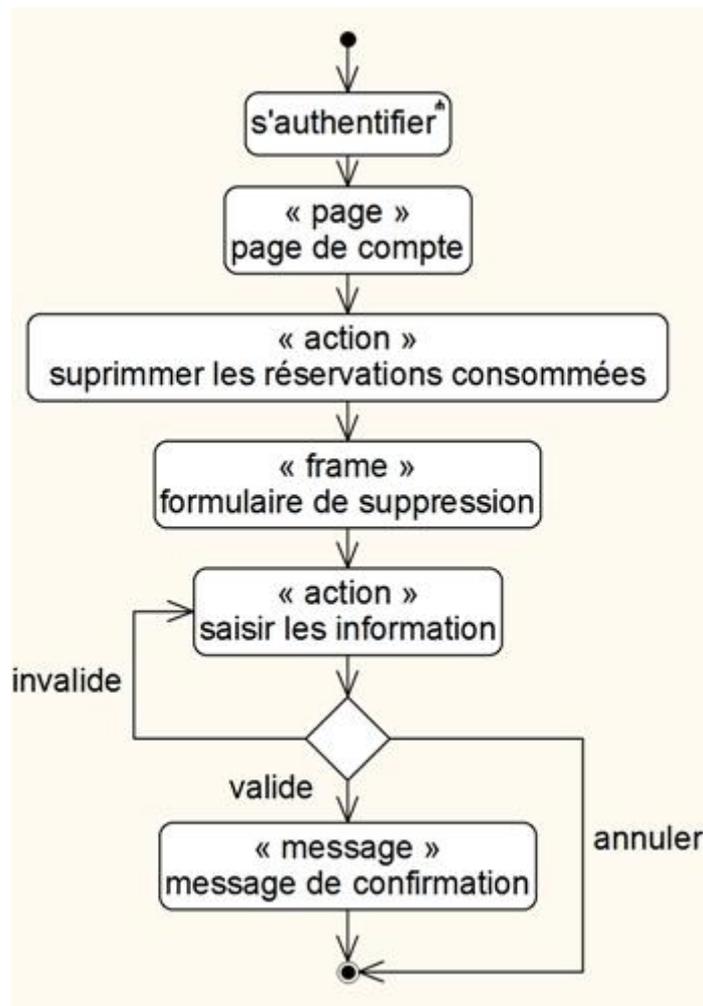


Figure 4.17 diagramme d'activités de navigation (supprimer les réservations consommées)

4.7 Conclusion:

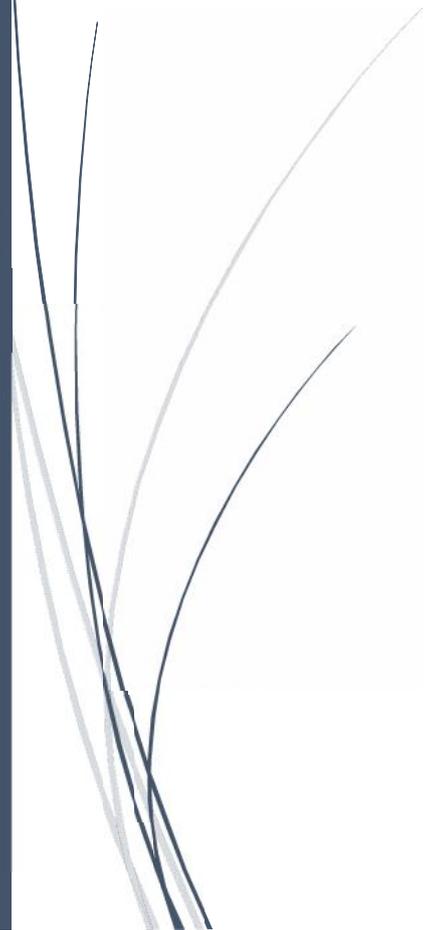
A l'issue de cette étape nous avons exprimé clairement les objectifs attendus du futur système à concevoir, ainsi que l'analyse associée à chaque cas d'utilisation et la possibilité de les réaliser dans un paradigme orienté objet, sans s'attacher à aucun outil de développement.

Nous allons dans le chapitre suivant on va faire la conception de notre système.



CHAPITRE 5

*PHASE
DE
CONCEPTION*



5.1 Introduction:

Nous allons maintenant passer à l'étude de la vue dynamique (le comportement) du système. Pour cela nous allons chercher à mettre en évidence les interactions entre les objets, ainsi que les messages échangés à travers les diagrammes d'interaction (précisément ici le diagramme de séquence).

5.2 Les diagrammes d'interaction :

5.2.1 S'authentifier :

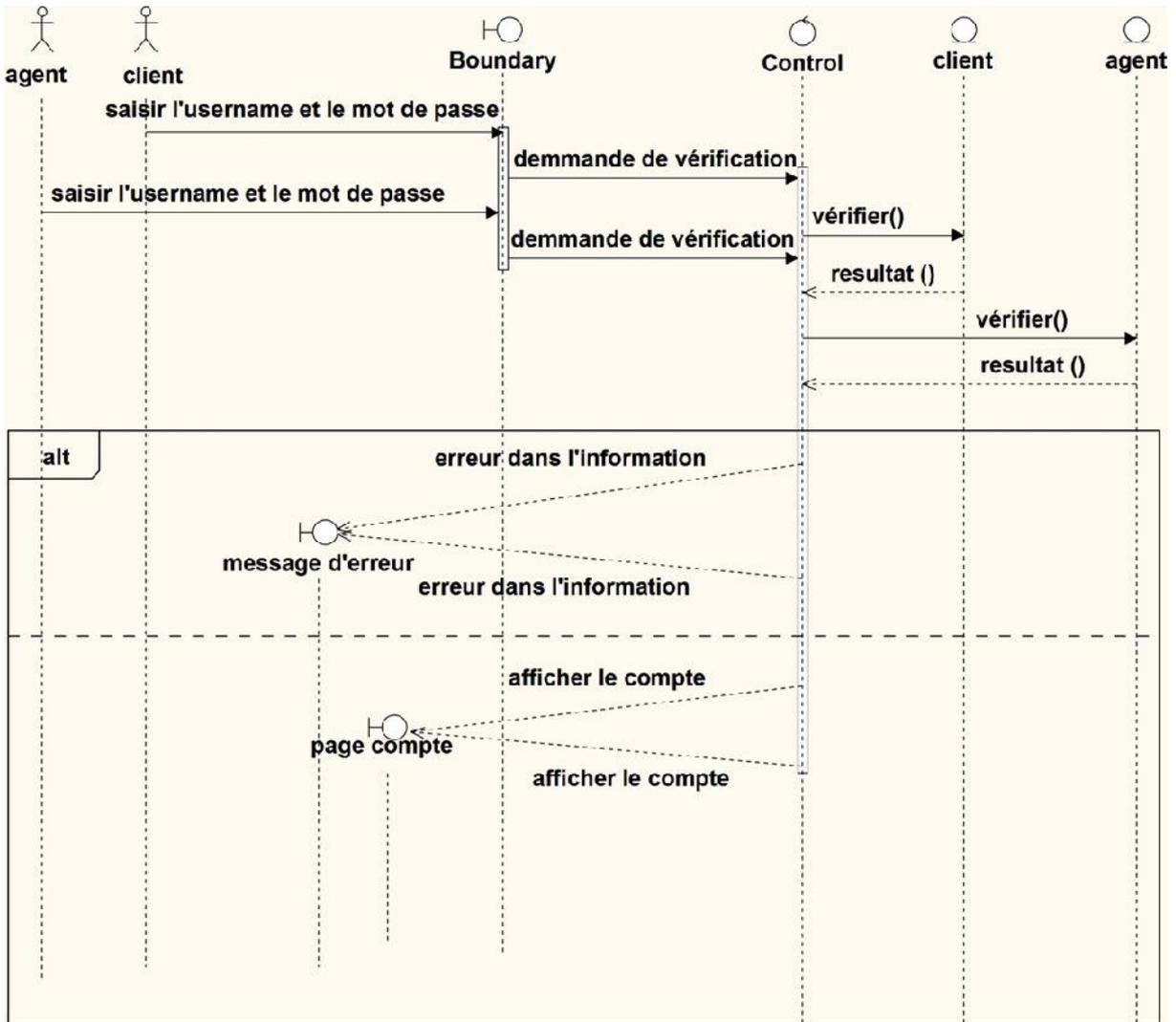


Figure 5.1 diagramme d'interaction (s'authentifier)

5.2.2 Réserver chambre :

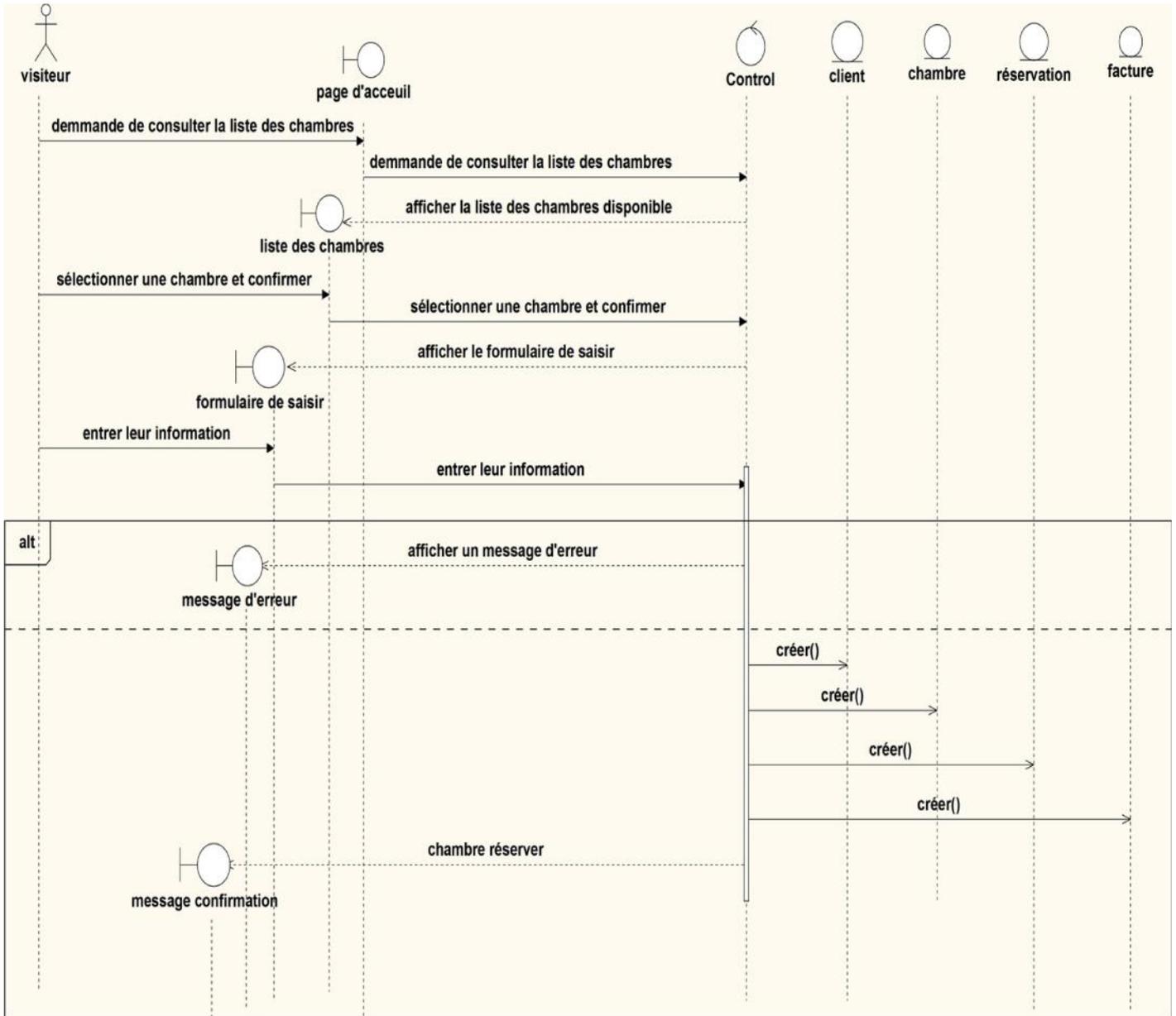


Figure 5.2 diagramme d'interaction (réserver une chambre)

5.2.3 Annuler réservation :

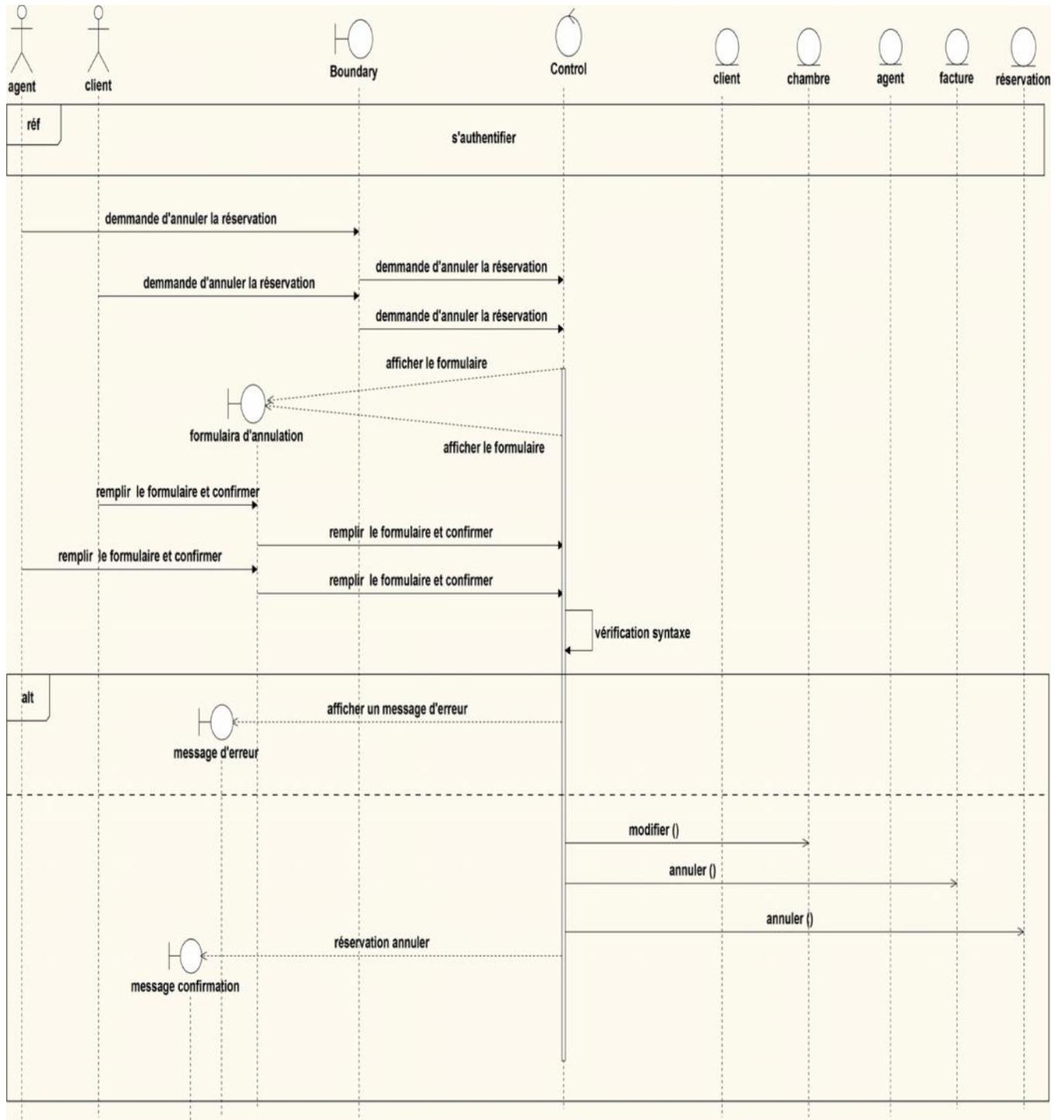


Figure 5.3 diagramme d'interaction (annuler la réservation)

5.2.4 Modifier la réservation :

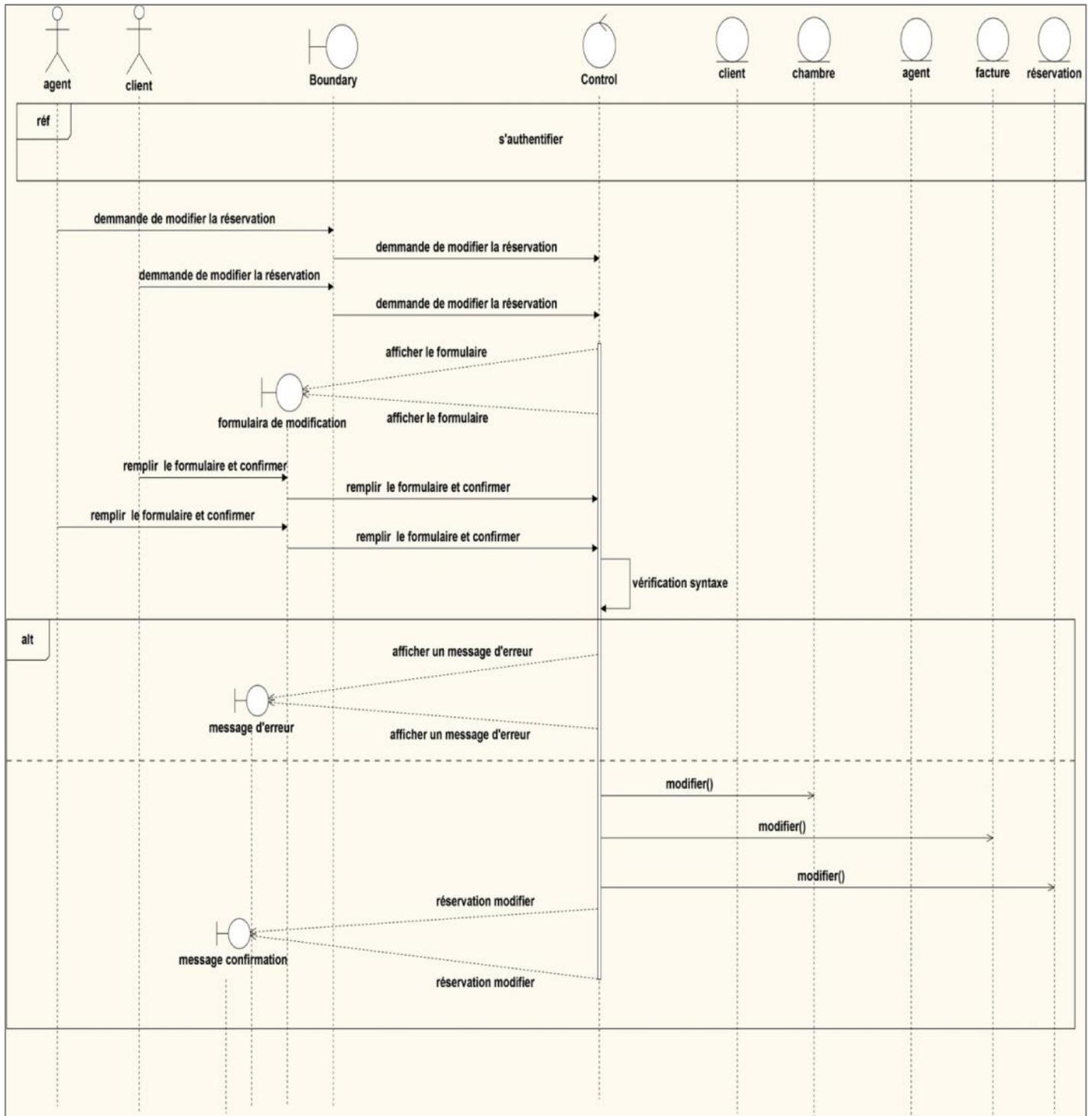


Figure 5.4 diagramme d'interaction (modifier la réservation)

5.2.5 Enregistrer réservation :

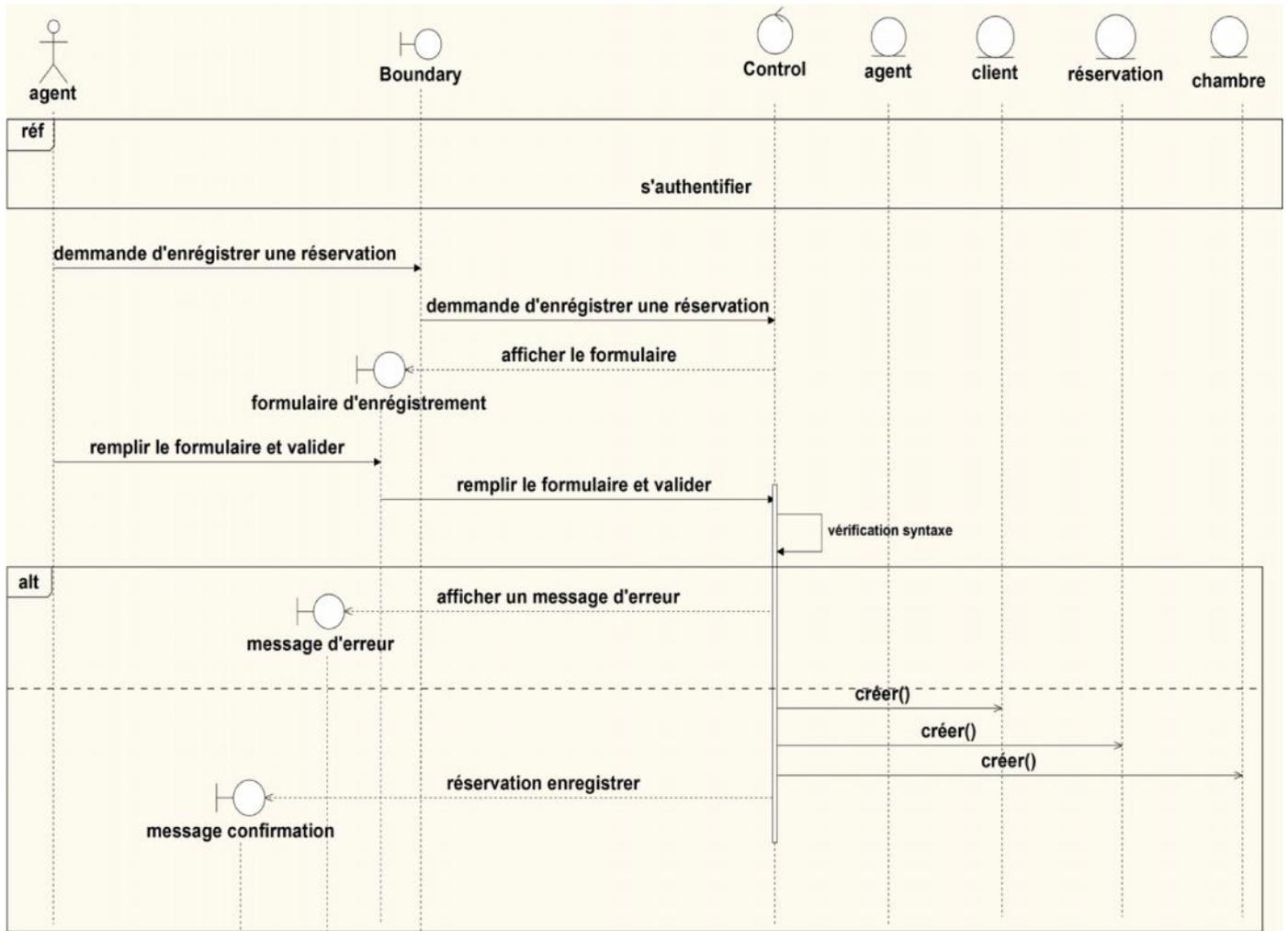


Figure 5.5 diagramme d'interaction (enregistrer la réservation)

5.2.6 Modifier les tarifs :

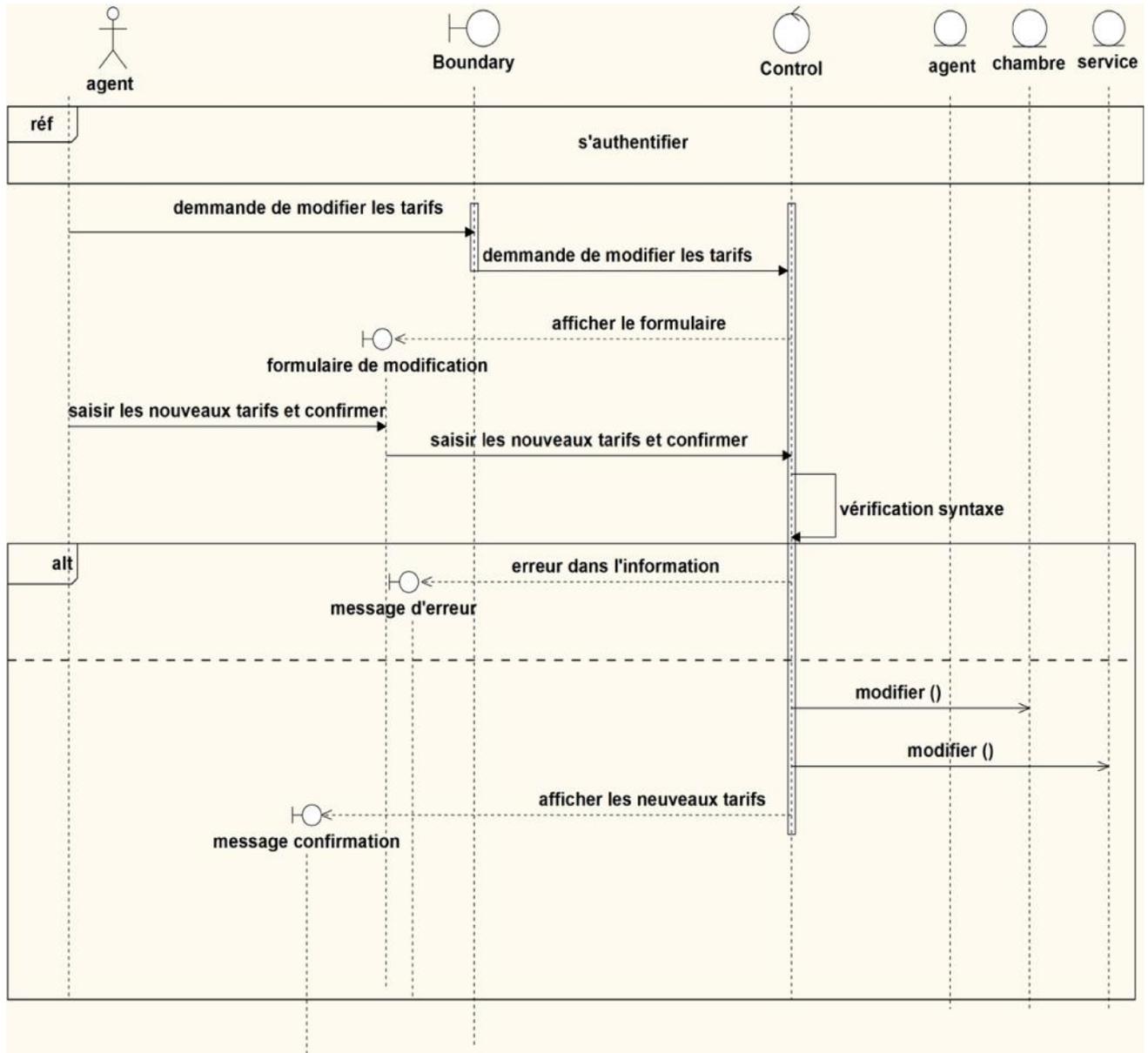


Figure 5.6 diagramme d'interaction (modifier les tarifs)

5.2.7 Consulter la liste des réservations en cours :

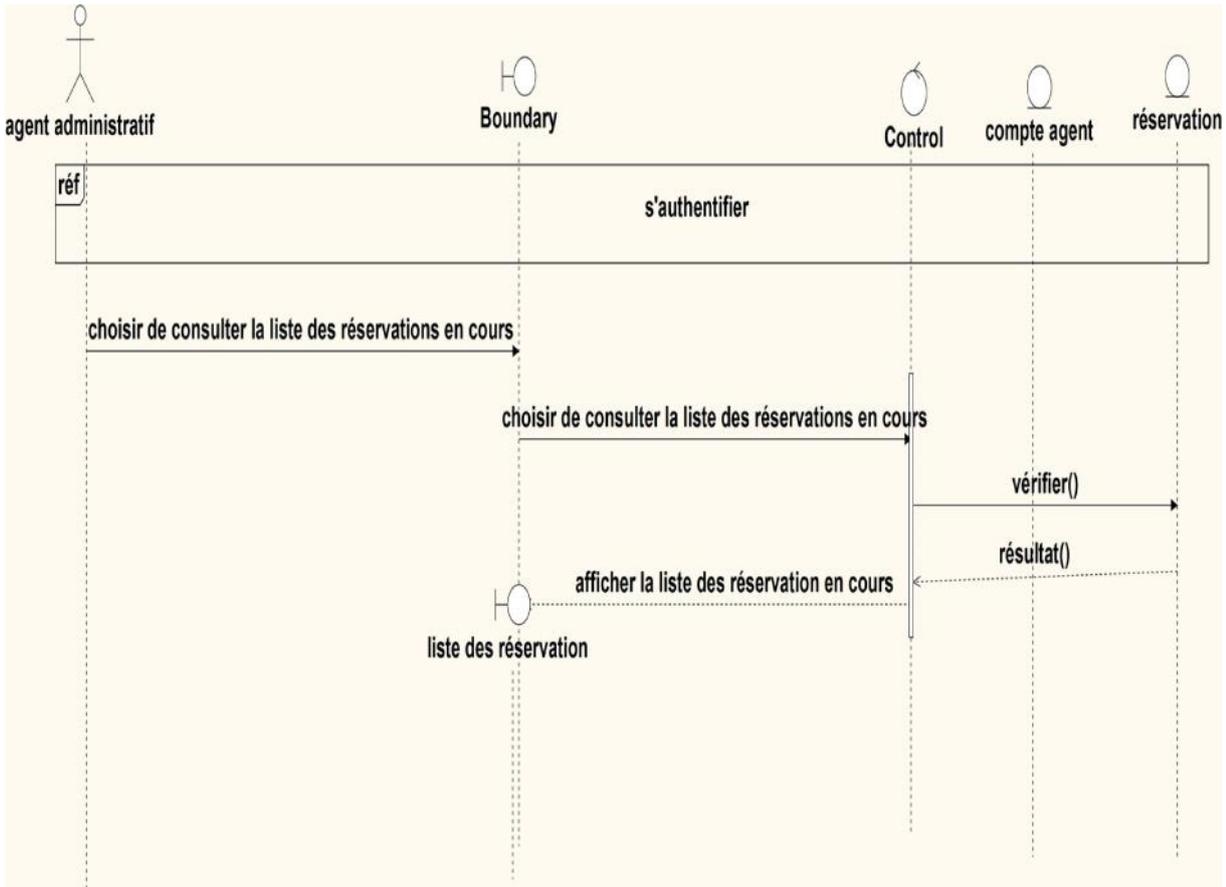


Figure 5.7 diagramme d'interaction (consulter la liste des réservations en cours)

5.2.8 Supprimer les réservation consommées :

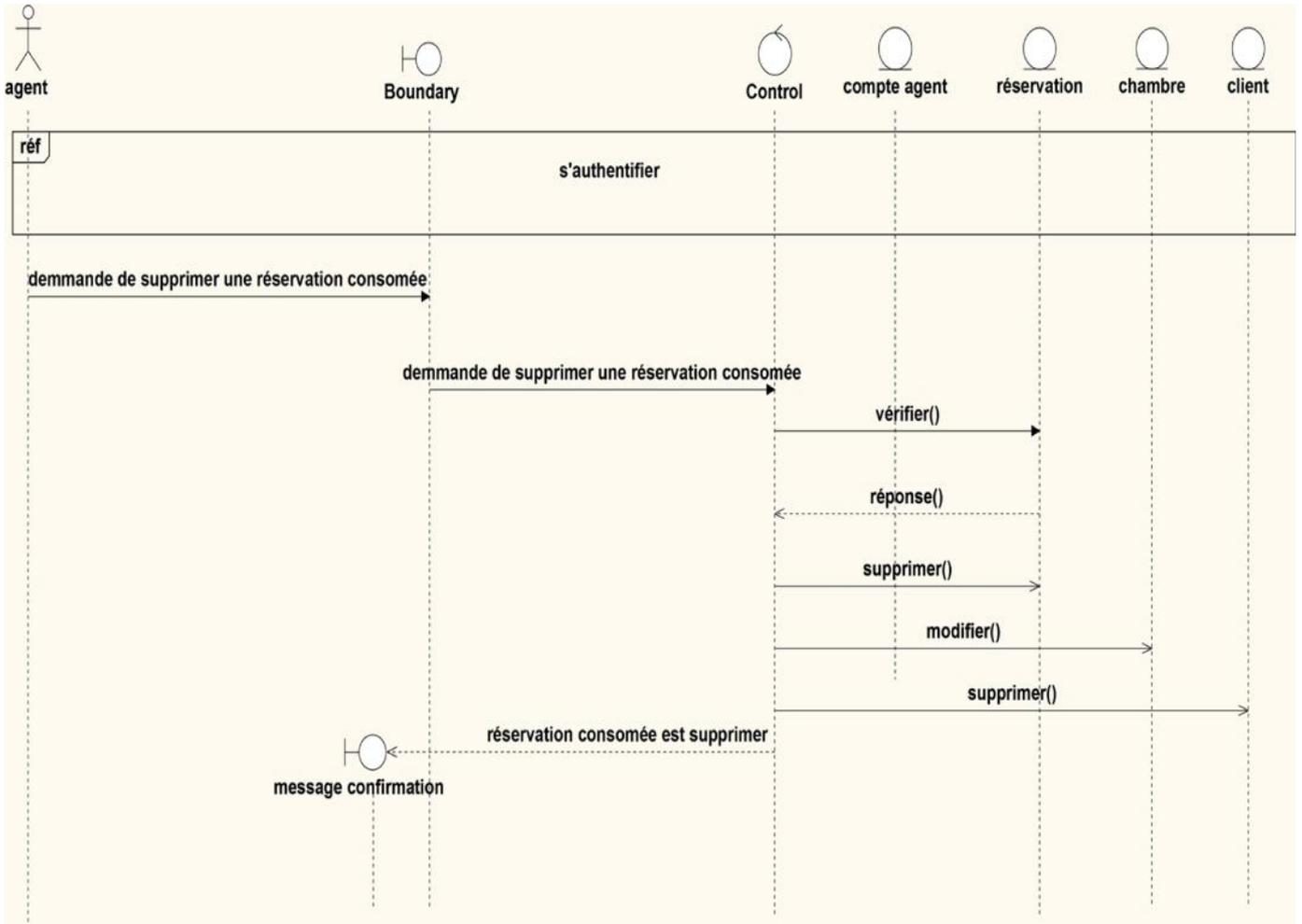


Figure 5.8 diagramme d'interaction (supprimer des réservation consommées)

5.3 Les diagrammes de classes de conception :

5.3.1 S'authentifier :

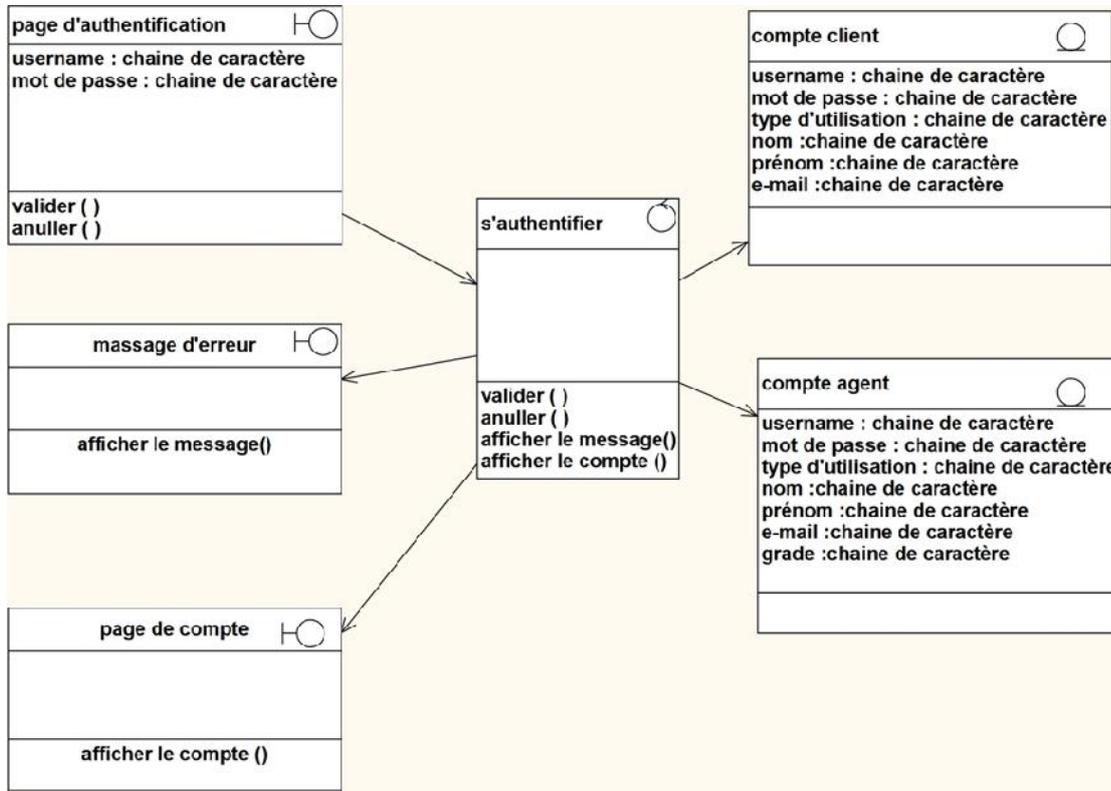


Figure 5.9 diagramme de classe de conception (s'authentifier)

5.3.2 Réserver une chambre :

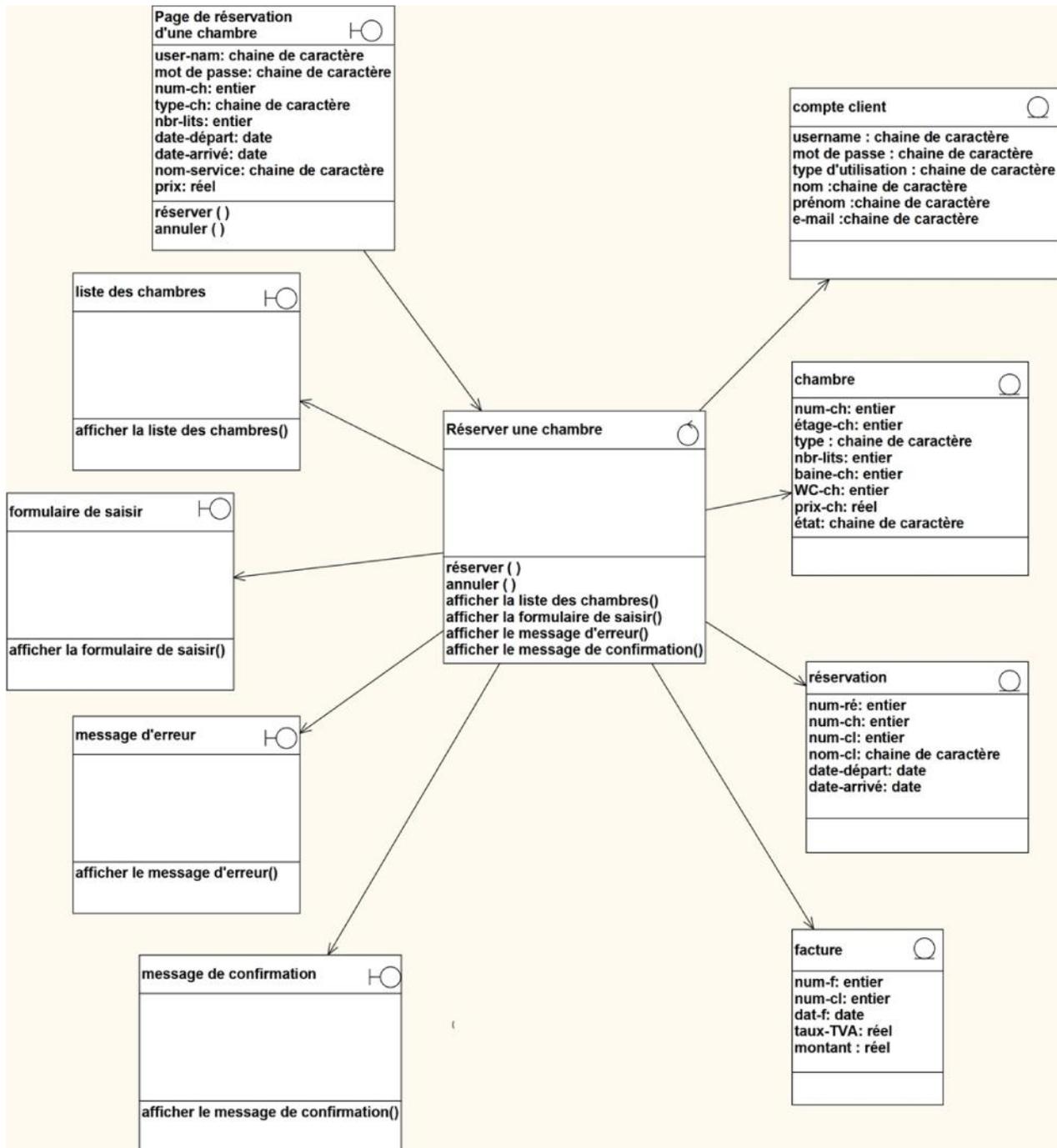


Figure 5.10 diagramme de classe de conception (réserver une chambre)

5.3.3 Annuler la réservation :

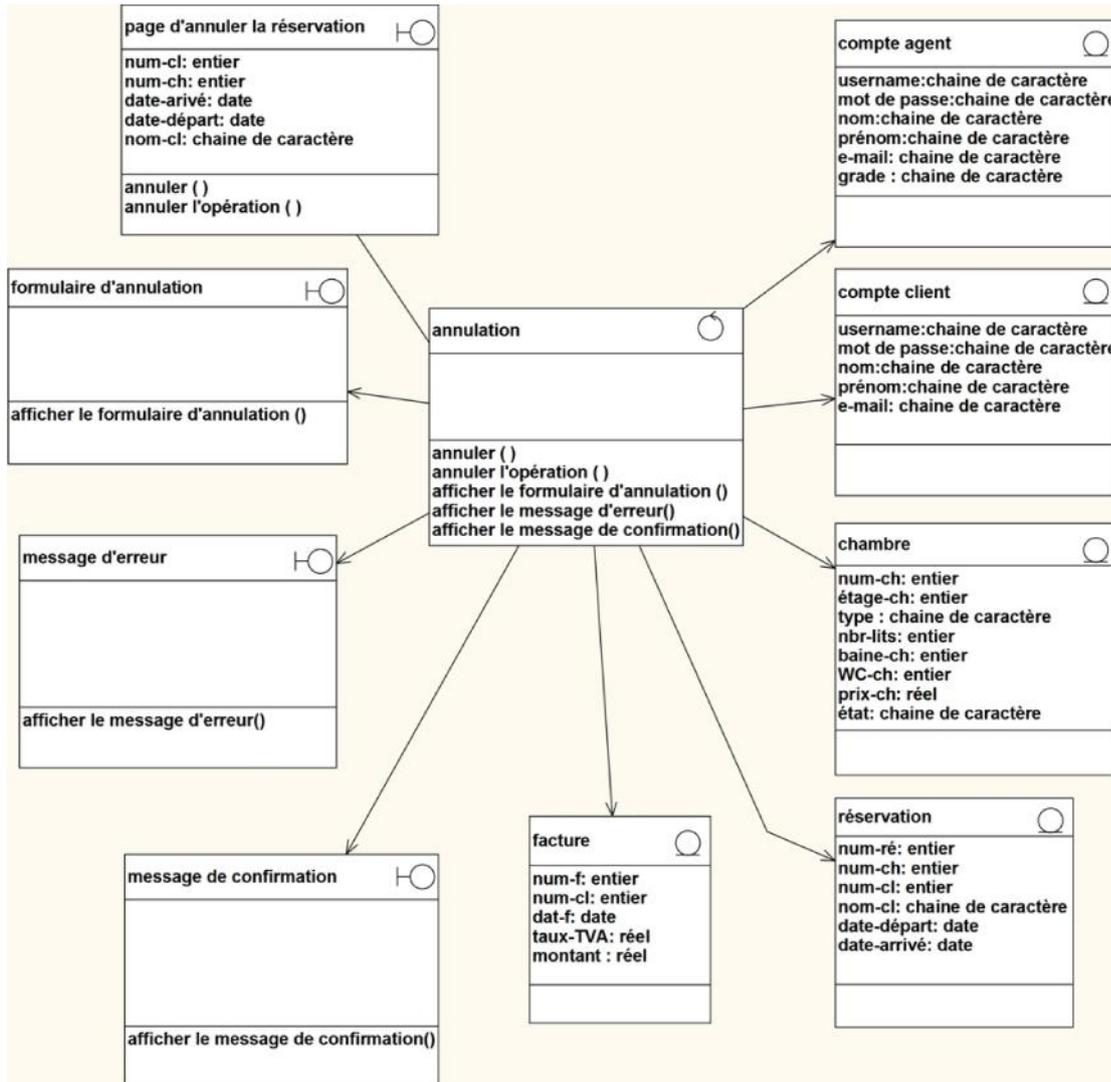


Figure 5.11 diagramme de classe de conception (annuler la réservation)

5.3.4 Modifier la réservation :

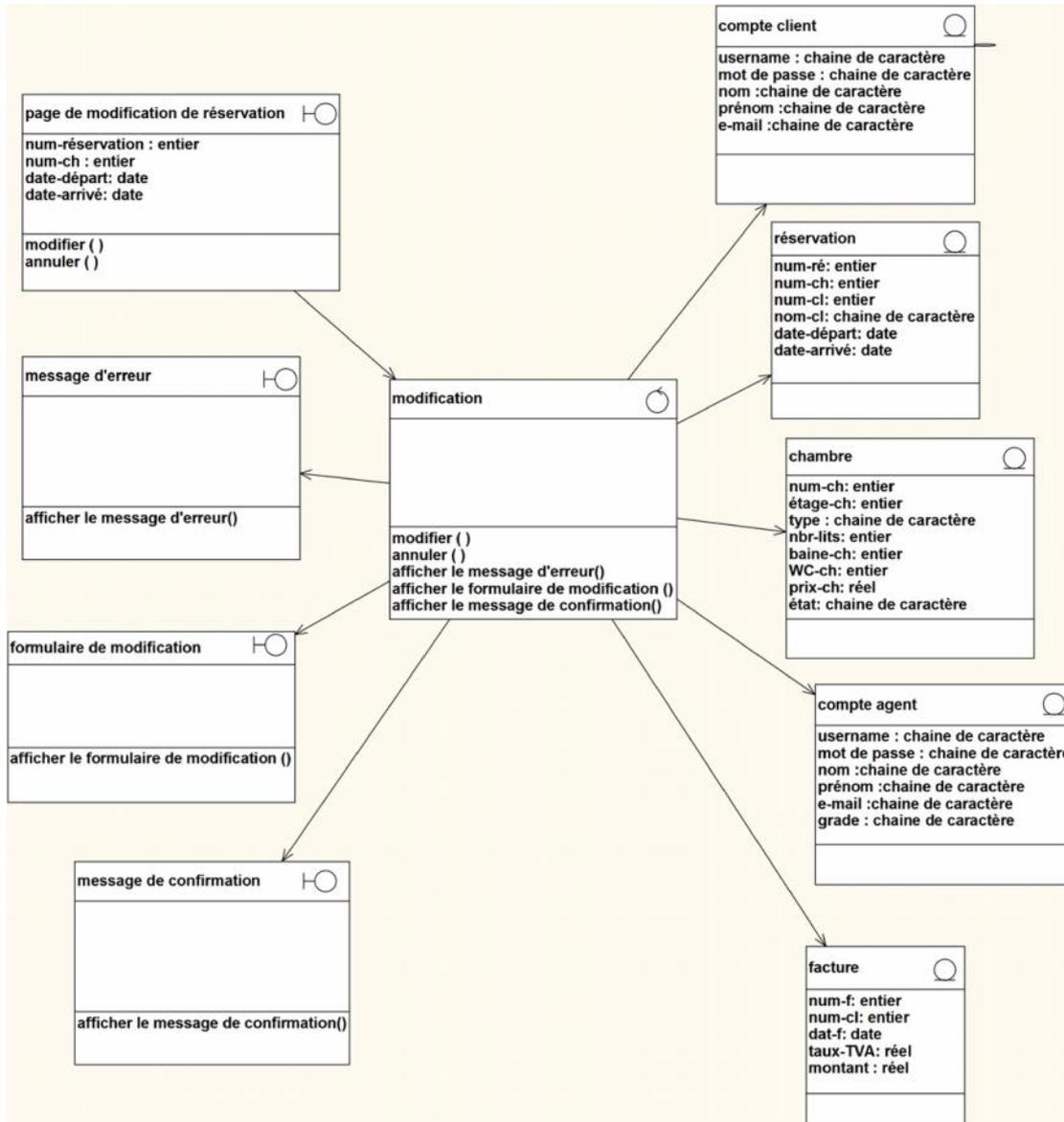


Figure 5.12 diagramme de classe de conception (modifier la réservation)

5.3.5 Enregistrer la réservation :

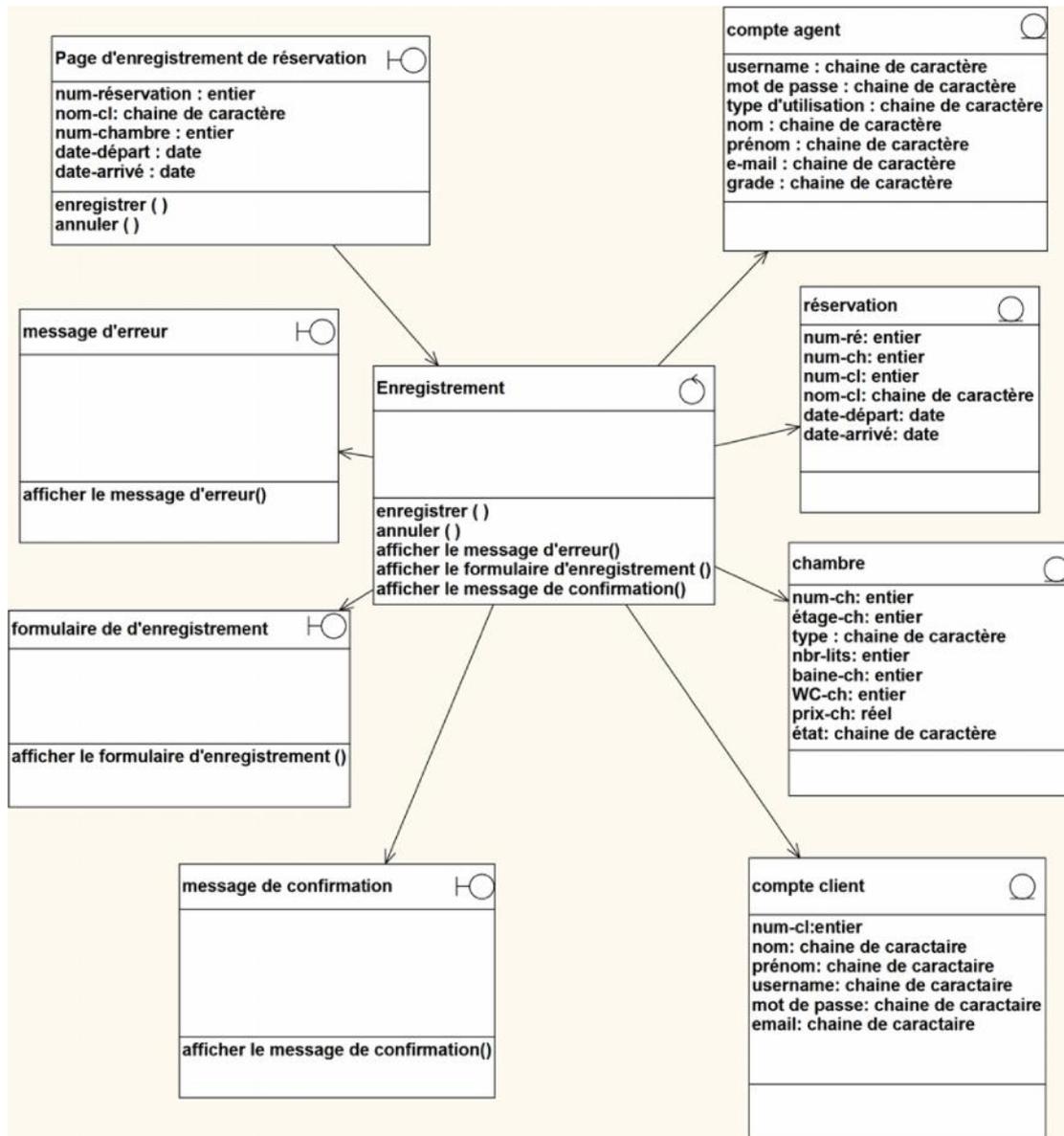


Figure 5.13 diagramme de classe de conception (enregistrer la réservation)

5.3.6 Modifier les tarifs :

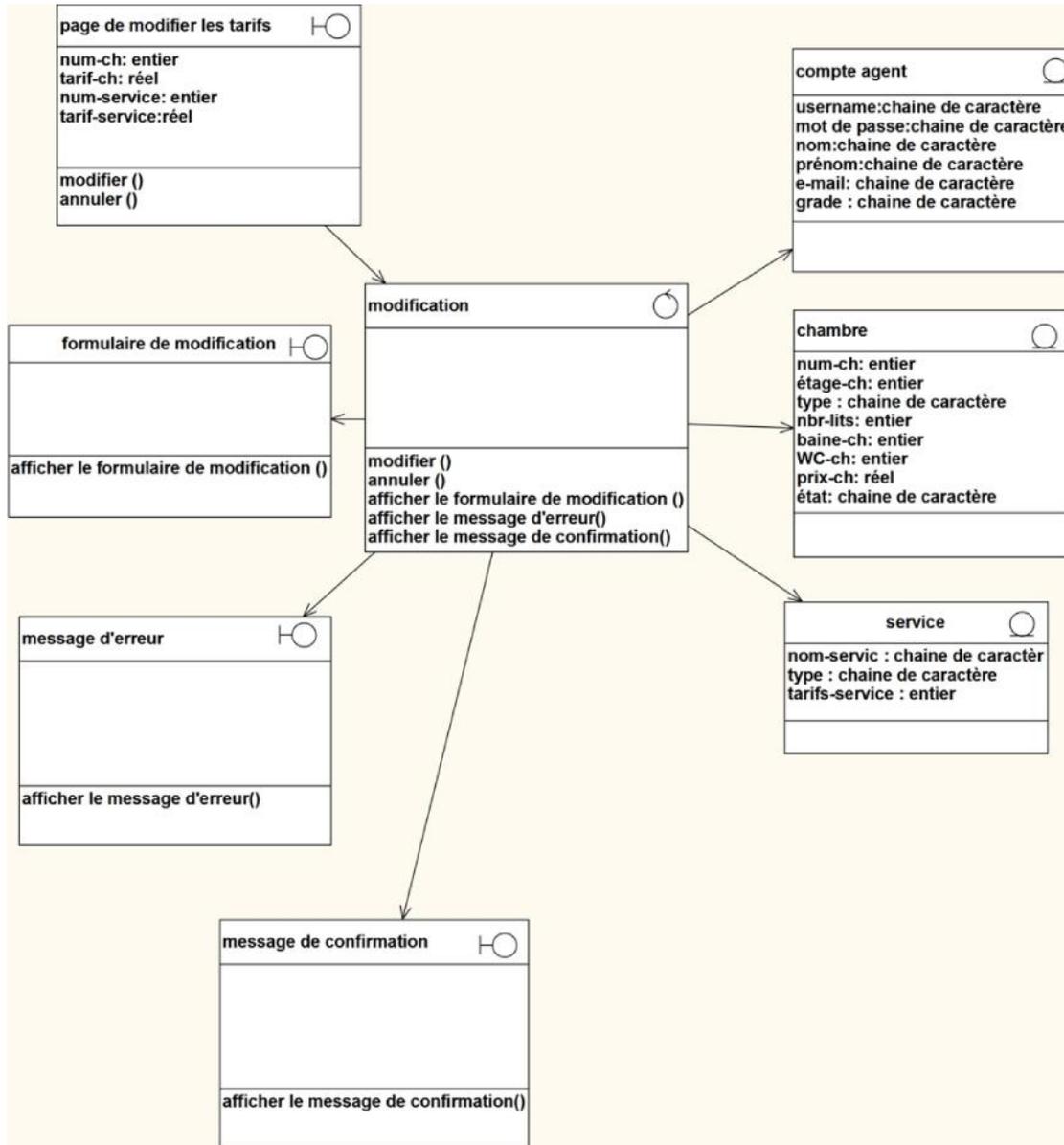


Figure 5.14 diagramme de classe de conception (modifier les tarifs)

5.3.7 Consulter la liste des réservations en cours :

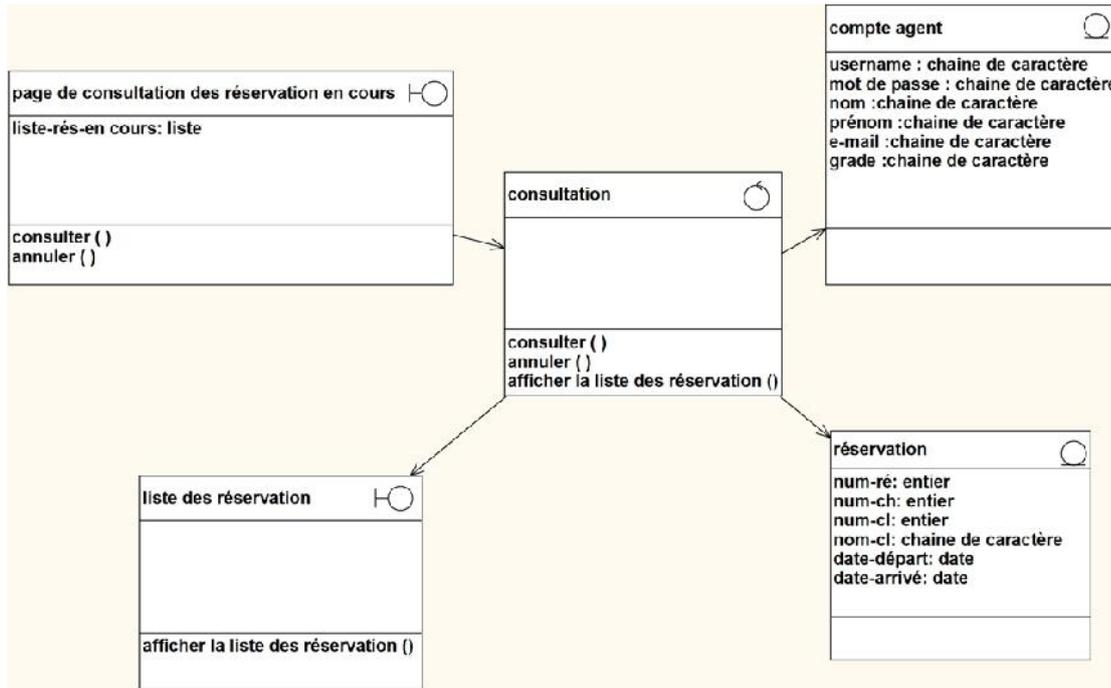


Figure 5.15 diagramme de classe de conception (consulter la liste des réservations en cours)

5.3.8 Supprimer les réservations consommées :

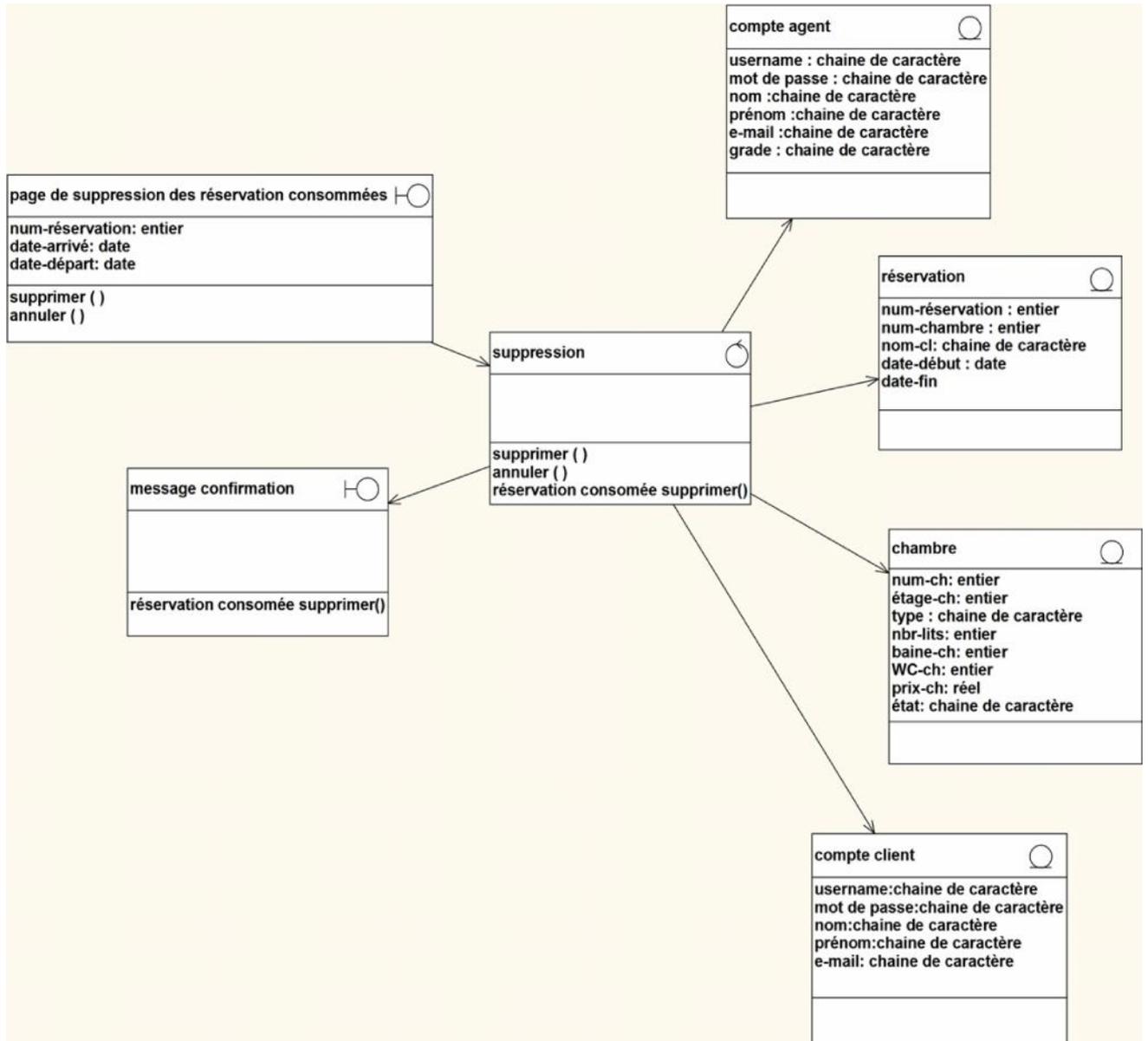


Figure 5.16 diagramme de classe de conception (supprimer les réservations consommées)

5.4 Diagramme de class :

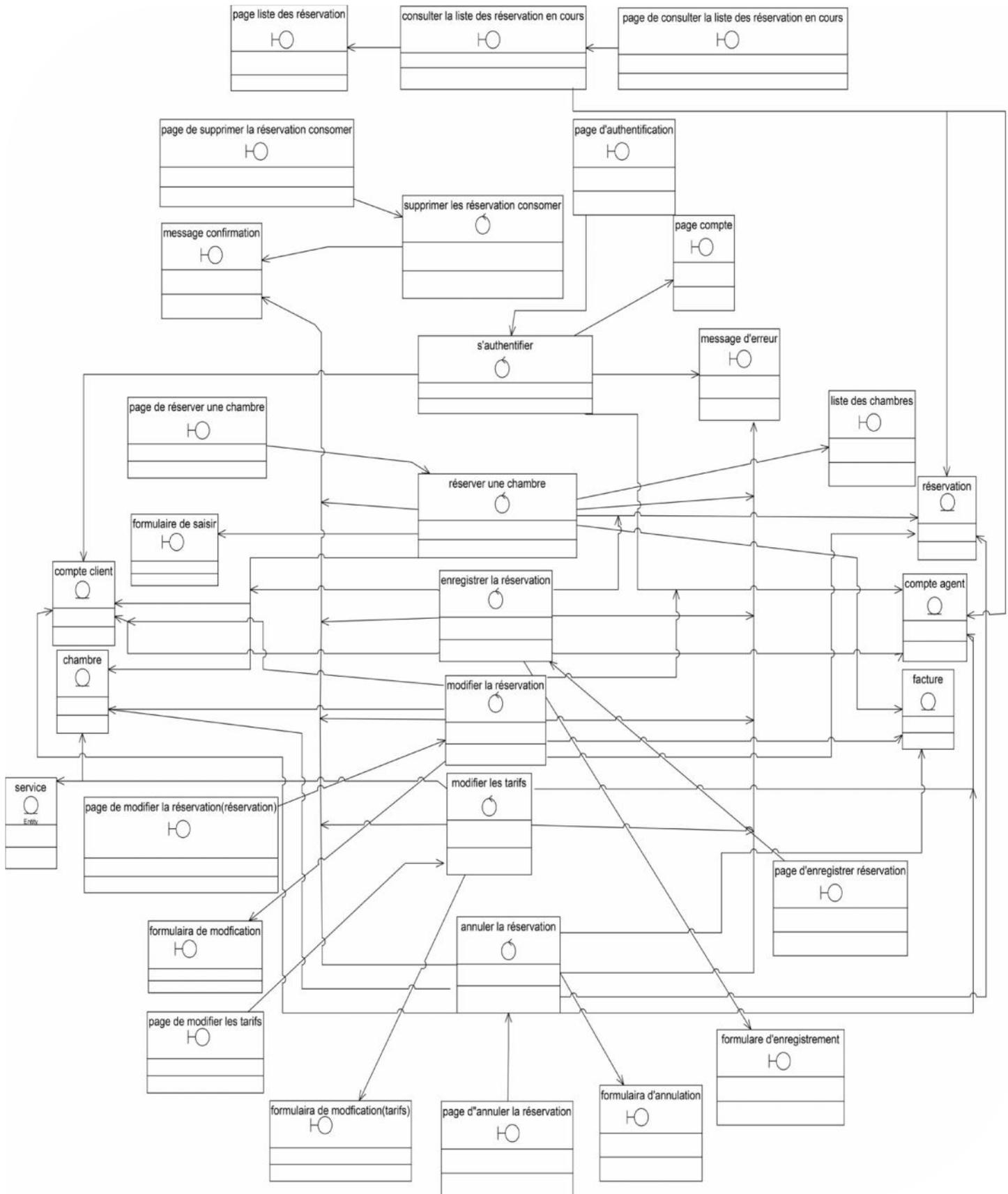


Figure 5.17 diagramme de classe de conception

5.5 Conclusion :

Dans ce chapitre nous allons élaborer les diagrammes d'interactions qui nous permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse du diagramme de classes participantes.

La phase de conception a intégré des nouvelles classes et méthodes aux modèles d'analyse et la manière de naviguer entre ces classes de conception.

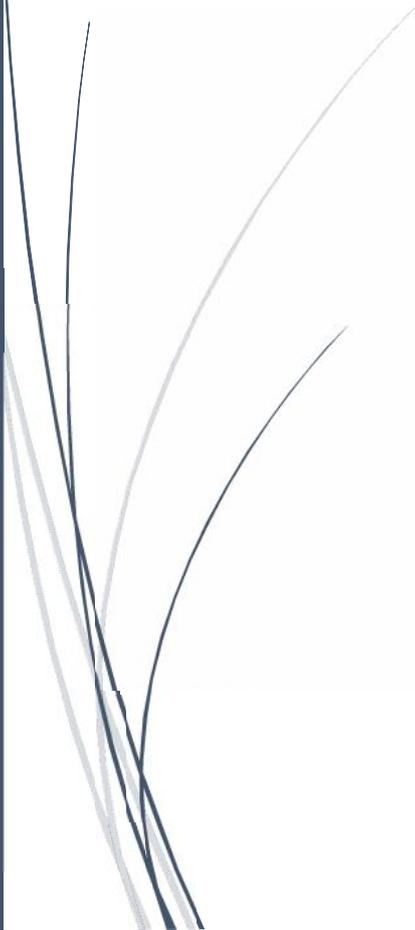
PARTIE 3

*REALISATION
DE
L'APPLICATION*



CHAPITRE 6

IMPLEMENTATION



6.1 Introduction :

Lors de la phase d'implémentation, nous allons utiliser les résultats de la phase de conception (le diagramme de classes de conception, et les diagrammes d'interactions) pour implémenter notre application dans un langage de programmation. Nous allons également préparer les données nécessaires à l'exploitation du logiciel, en convertissant les classes entités en une base de données. Pour réaliser notre application nous avons utilisé le langage **PHP** avec le serveur de bases de données **MySQL**, et langage **HTML**.

6.2 HTML

6.2.1. Langage HTML :

HTML est l'abréviation de **HyperText Markup Language**. Il a été présenté en version 1 en 1993. C'est justement à cette époque qu'est apparu **MOSAIC**, le premier navigateur **WWW**. Depuis lors, **HTML** a été largement amélioré et étendu. [33]

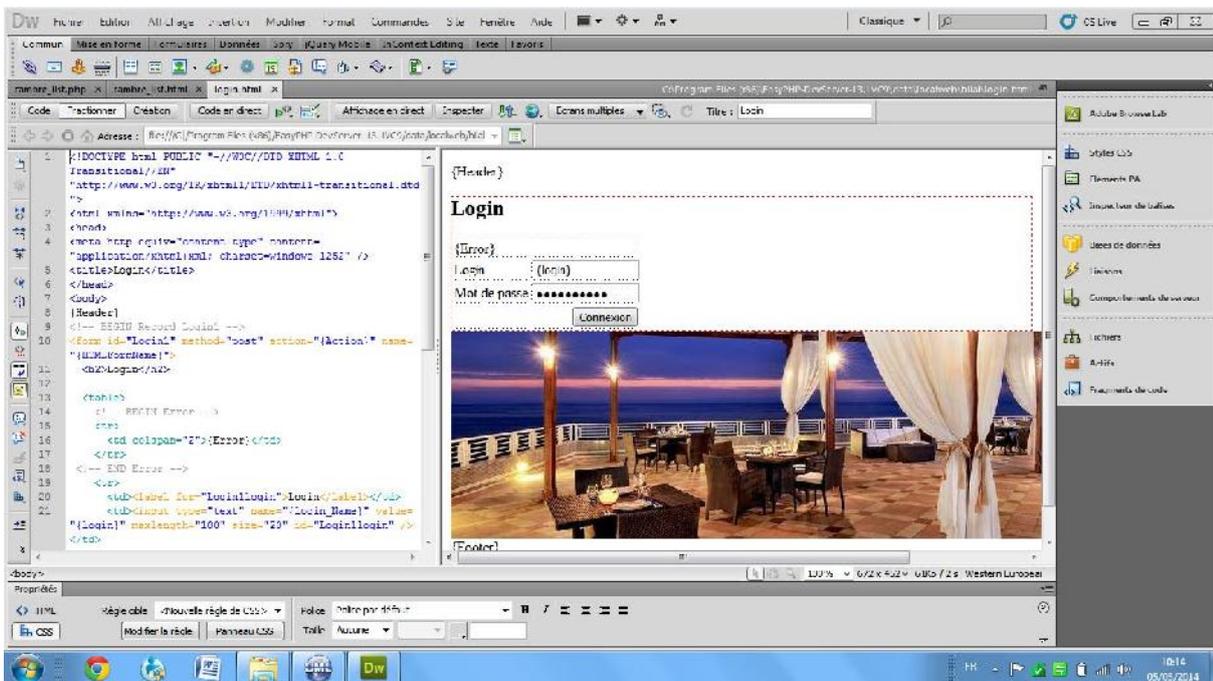


Figure 6.1 : Langage HTML

6.2.2. Définition :

HTML (**Hyper Text Markup Langage**) est langage de description de document. C'est un langage particulier issu de la norme **SGML** (**Stadardized Generalized Markup Langage**) qui définit des langages de balisage. C'est le langage qui a été construit pour décrire les documents **HyperText** sur Internet. **HTML** est normalisé par le **W3C** (consortium qui regroupe nombreuses et organisations). Etre normalisé signifie que les acteurs d'internet se sont mis d'accord pour suivre tous les mêmes règles pour écrire et traiter des documents **HyperText** sur le web [33].

6.2.3. Principe du langage HTML

Le langage **HTML** permet de créer des documents indépendants de toute plate-forme, et Donc particulièrement bien adaptés à des échanges d'information dans un environnement hétérogène comme le Web.

Il existe de nombreux éditeurs **HTML** qui facilitent grandement la programmation parmi eux, nous citons Microsoft FrontPage, mais si vous souhaitez réellement construire des pages Web fonctionnelles, vous éviterez difficilement une part de travail manuel. [33]

6.2.4. La structure générale d'un document HTML :

Les pages **HTML** ont toutes la même structure de base. Elles se composent de plusieurs balises (tags) ou instruction, placées entre les caractères < et >. La plupart d'entre elles vont de paire avec une balise de fin, servant à interrompre leur action.

La balise de fin est identique à celle de départ sauf que l'instruction proprement dite est précédée d'un slash (/).

Même si vous n'avez rien inséré, une page **HTML** comporte néanmoins quelques éléments .Il s'agit des balises qui représentent la structure de base.

Un document **HTML** est constitué de « tags » définissant la structure de la page, et de texte définissant le contenu ou le comportement (script) du document. Un « tags » est une sorte d'instructions **HTML**. Il est constitué d'un « mot clé » du langage entre deux chevrons.

La plupart des tags sont double un ouvrant et un fermant. Un tag fermant ajoute un slash (/) derrière le chevron gauche. Ainsi, un document **HTML** commence par <html> et se termine par </html>.

```
<html>
....
<head>
....
<title> </title>
</head>
<body>
....
</body>
```

Structure de bases d'un document HTML

Une page **HTML** valide est composée de deux parties : l'entête « Head » et corps « body » .

- **L'entête « Head »**

L'entête est comprise entre les tags <Head> et </Head>. S'il est présent, il apparaît nécessairement juste après l'ouverture du bloc **HTML**, et contient un nombre d'informations sur la page : adresse de base pour les liens relatifs, propriété d'index spécifications des liens, etc.

Mais les deux composantes les plus utilisées sont les tags <title> ; le titre de la page, C'est-à-dire le texte apparaissant dans la barre de titre du navigateur si la page. Occupe la fenêtre, et <meta> qui permet de fournir des informations sur la page.

L'entête est destiné à contenir des informations relatives à l'autre du document et à la nature de son contenu. Ces informations ne sont pas affichées par les navigateurs, mais peuvent être utilisées par exemple par les moteurs de recherche pour faciliter l'accès au document.

- **Corps « Body »**

Le corps de la page est compris entre les tags <body> et </body >. Tout élément se trouvant en dehors de ces tags ne sera pas affiché par le navigateur.

6.3. JAVA SCRIPT

Le JAVA SCRIPT est un langage de script incorporé dans un document HTML.

Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client. C'est-à-dire au niveau du navigateur et non du serveur web.

Ainsi le langage de JAVA SRIPT est fortement dépendant du navigateur appelant la page Web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage JAVA, avec lequel il a longtemps été confondu [34].

6.4. Bases de données et MySQL :

6.4.1. Structure d'une base de données :

Une base de données peut contenir un nombre variable de tables. Chaque table est organisée en lignes et en colonnes. L'intersection entre une ligne et une colonne correspond à l'emplacement de l'élément de donnée que l'on désire stocker ou auquel il désire accéder. Chaque colonne accepte seulement un type de données prédéfinie, par exemple : "int" pour entier, "varchar" pour un nombre variable mais limité de caractères.

[35]

6.4.2. Base de données MySQL :

L'un des caractéristiques du langage PHP est la facilité de manipuler des bases de données et spécifiquement une base de données appelée : "MySQL".

Cette base de données est conforme à l'esprit du PHP : gratuite, elle demeure d'un outil puissant susceptible de servir de base pour des projets d'envergure.

MySQL utilise une architecture multi utilisateur et multi traitement, ce qui permet aux utilisateurs de se connecter à partir de la même machine ou encore de machines distantes : une fois connecté au serveur, l'utilisateur peut sélectionner la base de données s'il possède des privilèges appropriés. [35]

6.4.3. Base de données et applications web dynamiques :

Les bases de données constituent le cœur d'application web dynamique, le serveur web obtient les informations actualisées destinées aux pages web créées à la volé grâce à l'effet de ces bases de données. [36]

6.4.4. Passage du diagramme de classe au modèle relationnel

Nous donnons ici- après quatre règles (de R1 à R4) pour traduire un schéma conceptuel entité association ou UML en un schéma relationnel équivalent.

Il existe d'autre solution de transformation mais ces règles sont les plus simples et les plus opérationnelles :

6.4.4.1 Transformation des entités /classes : la règle est simple

R1 :* Chaque entité devient une relation, identifiant de l'entité devient clé primaire pour la relation.

* chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle identifiant.

6.4.4.2 Transformation des associations :

Les règles de transformation que nous allons voir dépendent des cardinalités /multiplicités maximale des associations. Nous distinguons trois familles d'association.

A2 -Association 1.* : la règle est la suivante :

R2 : 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.

A3 -Association *.* : la règle est la suivante :

R3 : association/classe – association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des identités connecté à l'association. Chaque attribut devient clé étrangère si entité/classe connectée dont il devient une relation en vertu de la règle R1. Les attributs d'association/classe – association doivent être ajoutés à la nouvelle relation ces attribut ne sont ni clé primaire, ni clé étrangère.

A4 -Association 1.1 : la règle est la suivante :

R4 : il faut ajouter un attribut de type clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimale égale à zéro. Dans le cas de diagramme UML il faut ajouter un attribut de type 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 d'entité classe connectée à l'association. Si les deux cardinalités minimales égales à zéro, le choix est donné entre les deux relations dérivées de la R1.

Si les deux cardinalités minimales égales à un, il est préférable de fusionner les deux entités/classe en une seule. [37]

6.4.5. Le serveur de bases de données MySQL

MySQL est un SGBD (Système de Gestion de Bases de Données) relationnel performant basé sur le langage d'interrogation standardisé SQL (Structured Query Language)

MySQL est inclus dans les principales distributions de Linux. Ecrit en langage C et C++, **MySQL** a été porté sur d'autres plates-formes, dont Windows NT, OS/2 et Mac OS X.

La popularité de **MySQL** tient également aux bibliothèques qui le complètent. Les développeurs PHP bénéficient d'un ensemble complet de fonctions d'interrogation de serveurs **MySQL**. Les adeptes d'autres langages de programmation trouveraient également les outils nécessaires à l'intégration de **MySQL** dans leurs réalisations. [35]

6.4.6. Pourquoi choisir MySQL

Les alternatives sérieuses à **MySQL** existent Citons DB. **SGBD** (éprouvé, conçu par l'université de Berkeley), Postage **SQL** et de très nombreuses solutions commerciales, dominées par liégeant Américain Oracle. Pour quoi préférer **MySQL** ?

Une raison pragmatique en courage ce choix De nombreux fournisseurs d'accès Internet, et hébergeurs de sites proposent un accès gratuit à une plate-forme **PHP/MySQL**. D'autres facteurs d'ordre technique plaident en faveur de cette solution :

❖ Vitesse de traitement

MySQL dérive d'un autre SGBD relationnel nommé **MySQL**. Il s'en distingue par une vitesse de traitement accrue résultant de la réécriture de routine **ISMA** de gestion des entrées/sorties et de l'adoption d'algorithmes d'appariement tables particulièrement optimisés. Organisé autour threads, **MySQL** sait tirer parti d'une architecture multiprocesseur si le système le permet. Les performances globales de **MySQL**. Sont étroitement liées aux choix architecturaux des concepteurs de systèmes d'exploitation.

▪ Fiabilité

L'exploitation quotidienne de bases de données de plusieurs dizaines de millions d'enregistrements confirme la robustesse de **MySQL**. Des listes de diffusion active contribuent à la correction rapide d'éventuels bogues. De plus, un utilitaire nommé **MYIAMCHK** teste la cohérence des tables et les répare automatiquement si nécessaire.

▪ Compatibilité SQL

MySQL est compatible SQL, standard de fait des langages de base de données MySQL respecte la norme ANSI SQL 92.

▪ **Sécurité**

MySQL enregistre les droits des utilisateurs dans des tables internes. L'administrateur d'une base de données contrôle finement les catégories de personnes et les machines autorisées ase connecté. Les privilèges sont restreints à la consultation ou étendus à l'écriture de données, voir à la modification de la structure même de la base de données. L'authentification repose sur des mots de passe cryptés.

❖ **Limitations**

MySQL n'intègre pas de mécanisme de transaction ou de réplication des bases de données existantes. Les instructions SELECT ne peuvent être imbriquées. Des techniques alternatives comblent ces manques. Journaux de requêtes pour la copie des bases et verrous pour la simulation des transactions Ces limitations résultent d'un choix de conception : proposer un SGBD relationnel véloce au détriment des techniques les couteuses en ressources machine

6.4.7. Organisation générale de MySQL

MySQL adopte le modèle client /Serveur. Un serveur héberge un démon nommé Mysqld (Unix) ou MySQL _shareware (Windows 9x). Ce démon interface le moteur du SGBD et les requêtes des clients. [35]

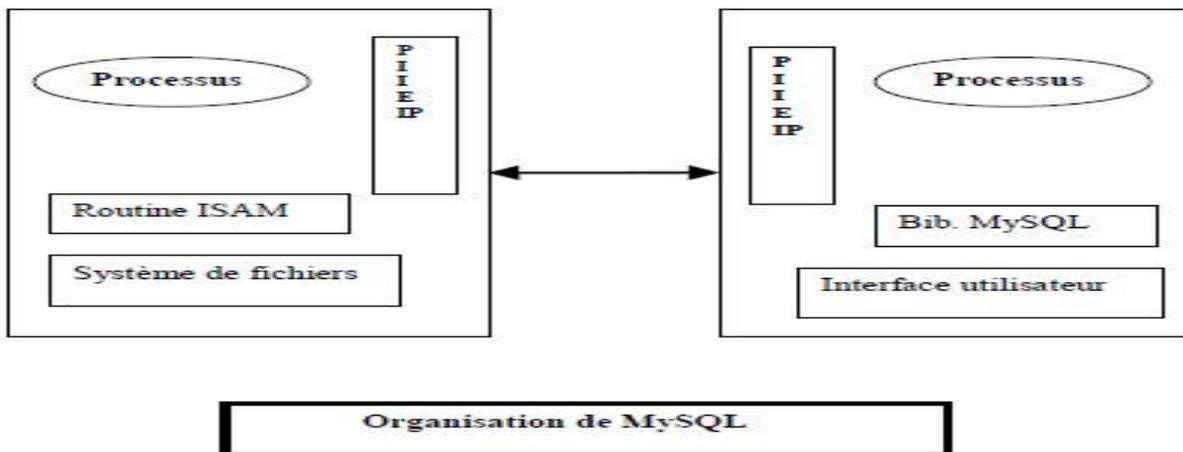


Figure 6.2 : organisation de MySQL

6.5. Langage PHP

Personnel Home page, ou même une définition récursive **PHP** HyperText préprocesseur, les significations choisies pour cet acronyme ne manquent pas **PHP** est un langage de programmation dédié à Internet, directement inclus dans les pages Web. On appelle aussi ce genre de langage, un langage de script.

Tout code **PHP** doit être inclus dans une balise < ? PHP ... ? >. Des balises « courtes » < ? > sont parfois acceptées, mais elles risquent d'entrer en conflit avec d'autres langages (comme XML) et ne sont donc pas recommandées. Comme en C le séparateur d'instructions est le point-virgule « ; ». Noter qu'une instruction « vide » marquée par un point-virgule est acceptée. [35]

6.5.1. Origine du langage PHP

Le langage PHP a vu le jour en 1994, sur le site de *Rasmus Lerdorf*. Ce dernier avait créé une page personnelle avec son Curriculum Vitae (CV), il souhaitait conserver une trace du passage de ses visiteurs et pour ce faire il inventa quelques scripts. Très vite les internautes qui découvrent le site, désirent obtenir un exemplaire du langage. C'est le début du succès pour *Rasmus* qui décide que son langage sera mis gratuitement à disposition des utilisateurs. [35]

6.5.2. Avantages du langage PHP

Les avantages de **PHP** sont sa facilité d'utilisation, sa très grande richesse fonctionnelle notamment vis-à-vis de la connexion à des bases de données. **PHP** est disponible pour énormément de plateformes compatibles avec beaucoup de serveurs web. Son utilisation se généralise très rapidement. On peut parier que **PHP** est en train de devenir un incontournable du web. [35]

6.6. Easy PHP

Contrairement à un site statique, un site dynamique nécessite la disponibilité d'une infrastructure serveur. En effet comme nous allons indiquer précédemment.

Le pack **Easy PHP** est ainsi téléchargeable, il regroupe tous les éléments indispensables à l'évaluation d'un site dynamique :

- Le serveur Web Apache.
- Le serveur de base de données **MySQL**.
- Le serveur d'application **PHP**.
- L'outil **PhpMyAdmin** permettant de gérer des bases **MySQL**.

PHP MyAdmin : 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**. Elle peut donc fonctionner directement sur le serveur Web et être accessible par le biais d'un simple navigateur. On peut ainsi créer avec la même facilité des bases de données et des tables en local comme sur le serveur distant. Avec le gestionnaire de bases de données **PhpMyAdmin**, vous pourrez rapidement :

- Créer et supprimer des données ;
- Créer copier supprimer et modifier des tables ;
- Supprimer, éditer et ajouter des champs ;

- Exécuter des requêtes SQL ;
- Créer et exploiter des sauvegardes de tables. [35]

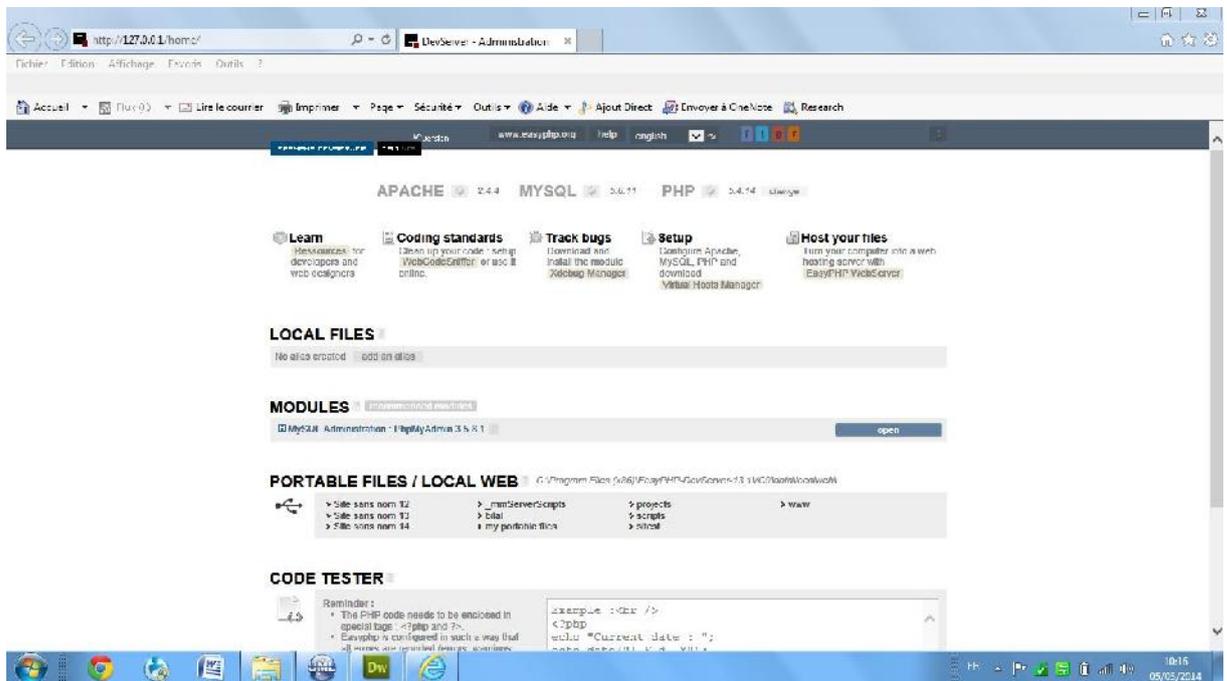


Figure 6.3 : Easy PHP

6.7. DREAMWEAVER CS5

Dreamweaver est un logiciel de mise en page et de gestion de sites internet. Il permet de construire des pages web en travaillant sur le code html et dans une interface graphique. Dreamweaver est la référence des éditeurs de page html. Il est utilisé par les webdesigners, les webmasters et toute personne impliquée dans la création de sites internet. [34]

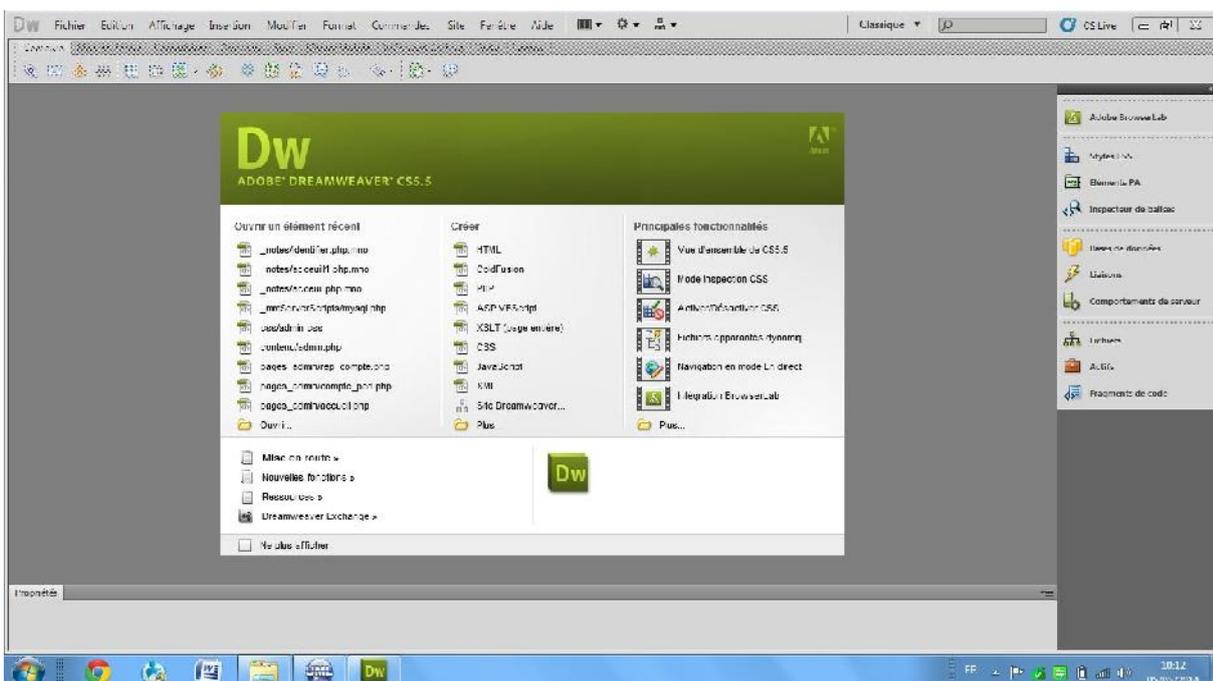


Figure 6.4 : Dreamweaver CS5.

6.8. Représentation de l'interface de site :

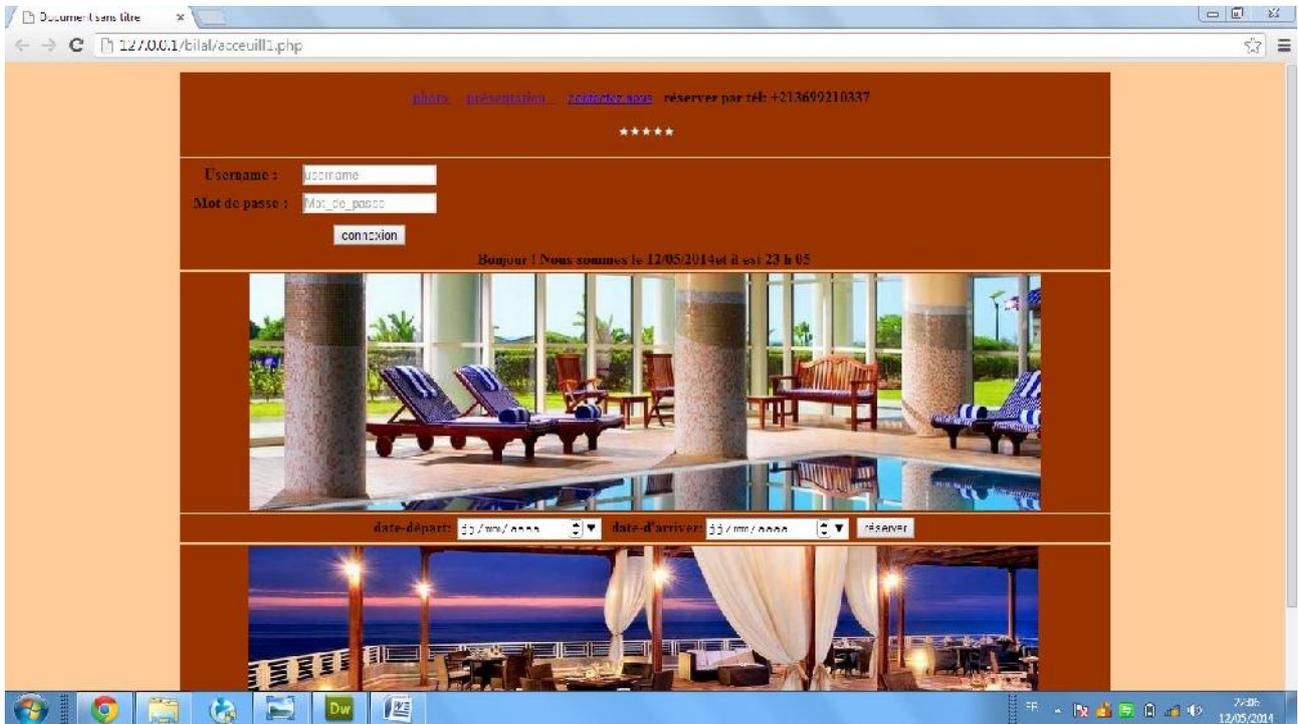


Figure 6.5 : Représentation de l'interface de site

6.9. Conclusion :

Dans ce chapitre, nous avons abordé tous les détails techniques en rapport avec le modèle d'implémentation de notre application. Dans sa première partie, ce chapitre a été consacré à la présentation des différents outils utilisés pour la réalisation de notre site. Nous avons introduit les concepts relatifs à l'environnement de développement tels que PHP, MYSQL, CSS, Dans la deuxième partie, nous avons présenté le dossier technique de notre application (fenêtre, code...). Toutes les fonctionnalités de notre outil sont accessible à travers une interface graphique permettant de faciliter les interactions entre l'utilisateur et le système.

Conclusion Générale

L'objectif de notre projet était réaliser un site web pour la gestion de réservation dans un hôtel, offrant un ensemble des services tels que l'accès à l'hôtel avec la permutation de réserver une chambre ainsi que les services disponible. Le site offre aussi un service de gérer la réservation (modifier, annuler, consulter, supprimer) après la reconnaître d'utilisateur, le forum permettant une interactivité entre les différents acteurs.

Plusieurs technologies ont été nécessaires pour la réalisation de notre projet, on citera donc le langage HTML pour la réalisation des pages statiques, le langage PHP pour la partie dynamique, SQL pour l'élaboration des requêtes d'interrogation de la base de donnée, et enfin le logiciel DreamWeaver pour l'écriture du code.

Notre site web n'est pas complet puisque on n'a pas pu respecter certaine fonction comme le mode paiement, mais Nous espérons, que notre travail a été bien détaillé. Le présent travail nous a permis d'acquérir des connaissances dans le domaine de la programmation web, et de conforter nos connaissances en conception logicielle.

REFERENCES BIBLIOGRAPHIQUE

- [1] <http://fr.wikipedia.org/wiki/Internet>
- [2] <http://pedagene.creteil.iufm.fr/internet/definit.htm>
- [3] <http://www.commentcamarche.net/contents/538-le-protocole-tcp>
- [4] http://encyclopedie.linternaute.com/definition/150/11/0/internet_protocol.shtml
- [5] http://fr.wikipedia.org/wiki/hypertext_transfer_protocol
- [6] <http://www.aira.com/aide/decouvrir/web.html>
- [7] <http://www.cardwellium.com/dynastatic.html>
- [8] http://www.journaldunet.com/encyclopedie/definition/54/41/21/site_dynamique.shtml
- [9] <http://www.adproxima.fr/glossaire-207-la-page-web.html>
- [10] http://fr.wikipedia.org/wiki/Page_web_statique
- [11] http://fr.wikipedia.org/wiki/Page_web_dynamique
- [12] <http://www.yakaferci.com/definition/navigateur-internet/19/>
- [13] <http://www.commentcamarche.net/contents/542-url>
- [14] <http://www.wikipedia.com>
- [15] <http://aeris.11vm-serv.net/cours/internet/internet.html>
- [16] http://fr.wikipedia.org/wiki/Wide_Area_Information_Servers
- [17] <http://lesservicesinternet.essca.over-blog.com/article-les-principaux-services-de-l-internet-100897674.html>
- [18] http://fr.wikipedia.org/wiki/Domain_Name_System
- [19] <http://fr.wikipedia.org/wiki/Client-serveur>
- [20] Bouali Meriem, Boulmaiz Amina, Développement d'une application client/serveur pour la gestion de la bibliothèque du centre universitaire de Mila, CENTRE UNIVERSITAIRE DE MILA, 2010/2011
- [21] Bara mohammed, Zemamouche tahar, Conception et réalisation d'un système pour la gestion clientèle au sein de SONELGAZ, CENTRE UNIVERSITAIRE DE MILA, 2011/2012.
- [22] Christine Solnon ; **Modélisation UML** ; INSA de Lyon - 3IF, 2013 - 2014
- [23] Eric Cariou ; **Introduction à UML 2** ; Master Technologies de l'Internet 1ère année Université de Pau et des Pays de l'Adour - UFR Sciences Pau – Département Informatique Eric.Cariou@univ-pau.fr
- [24] C Crochepeyre , Éléments d'UML pour le projet , (Unified Modeling Language)
- [25] Ferhat Yasser,Laouar Djaafar, Conception et réalisation d'un site web dynamique pour la vente des véhicules « L'entreprise KIV FOTON », CENTRE UNIVERSITAIRE DE MILA,2010/2011 .

- [26] Dehimi Meriem, Mimeche Besma «**Système informatique pour la gestion des assurances médicales par carte**», Université Mentouri de Constantine 2010
- [27] Djouambi Mohamed, Lemeurs Abdelbassit, Fredj Younes «**Conception et réalisation d'un site web dynamique pour le Centre universitaire de Mila**», Université Mentouri de Constantine 2009.
- [28] « **Gestion électronique des documents** », Université Mentouri de Constantine
- [29] Sadek Benhammada , Méthodes d'analyse et de conception orientée objet « Introduction au langage de modélisation UML », Centre Universitaire de Mila, 2012
- [30] Mickaël Gardoni, Cours GPA 789 – Analyse et Conception Orientées Objet
« Mise en Œuvre UML », Ecole de technologie supérieure, Université du Québec, 2009
- [31] Le processus unifié de développement logiciel-Partie I
Les concepts Présentation réalisée par P.-A. Sunier Professeur à la HE-Arcde Neuchâtel.
- [32] violaine louvet, le processus unifié appliqué au projet MOOCS, GTN MAI 2003, Orsay
- [33] Le langage HTML1.pdf.
- [34] boufenneche amine boudjadja sara LA CONCEPTION ET REALISATION DE SITE WEB DYNAMIQUE DE ACHETER ET VENTE UN IMMOBILIER EN LIGNE / Centre Universitaire de Mila 2012 / 2013
- [35]: PHP 4 MATT ZANASTRA Publié par COMPUS PRESS EDITION 2000
- [36]: Grand Livre PHP & MySQL/GA. LEIERER et R.STOLL/Traduction: Samy Boutayeb 1^{ère} édition août 2000
- [37]: Laurent AUDIBERT UML 2 Édition 2007-2008