



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMARI DE TIZI OUZOU
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT INFORMATIQUE

Mémoire

EN VUE DE L'OBTENTION D'UN MASTER EN INFORMATIQUE

Thème

*Gestion d'un ensemble de parkings
automobiles*

Dirigé par :

Mr OUAMRANE

Réalisé par :

**Mlle AMNACHE Ahlem
Mr OUADI Athmane**

2010/2011

Remerciements :

D 'abord nous remercions le bon dieu de nous avoir donné santé, courage et foi pour réaliser ce travail avec volonté.

*N ous tenons à exprimer notre profonde gratitude à notre promoteur **M OUAMRANE***

*N os plus vifs remerciements vont aussi à tout le personnel de l'entreprise **SERIUS NET**, plus particulièrement à **M FELLAHI** qui nous a généreusement aidés durant notre stage au sein de l'entreprise.*

Que les membres du jury trouvent ici nos plus vifs remerciements pour avoir aimablement accepté de juger notre travail.

U n grand merci aussi à tous ceux qui, de près ou de loin ont contribué à ce que ce modeste travail voit le jour. E n particulier nos chères familles et nos amis(es).

AHLEM ET ATHMANE

Sommaire

I. Introduction Générale	01
---------------------------------------	----

Chapitre I : Généralités

Partie 1 : Généralisation sur les parkings

Introduction.....	02
II. Définition.....	02
III. Types de parkings.....	02
III.1. Les parkings de surface.....	02
III.2. La zone bleue.....	02
III.3. Les parking fermés ou souterrains.....	02
III.4. Les parkings à étages.....	02
III.5. Les parking-relais.....	03
IV. Pourquoi louer une place de parking.....	03
V. Gestion des parkings pour amélioration de leurs utilisations.....	03
Conclusion.....	03

Partie 2 : Généralisation sur les réseaux

I.1.Introduction.....	04
I.2.Les réseaux informatiques.....	04
I.2.1.Intérêt des réseaux.....	04
I.2.2.Classification des réseaux.....	04
I.2.2.1.les réseaux filaires.....	05
I.2.2.1.1.En fonction de la taille.....	05
I.2.2.1.2.En fonction de la topologie.....	05
I.2.2.2.les réseaux sans fils.....	09
I.2.2.2.1.Réseau personnel ou « Bluetooth ».....	09
I.2.2.2.2.Réseau local sans fils.....	09
I.2.2.2.3.Réseau métropolitains sans fils.....	10
I.2.2.2.4. Réseau Wan sans fils.....	10
I.2.2.2.5. Réseau ad hoc.....	11
I.3. Les modèles de références.....	11
I.3.1. Le modèle OSI.....	11
I.3.2. Le modèle TCP/IP.....	12
I.3.3. Architecture client/serveur.....	13
I.3.3.1.Définition.....	13
I.3.3.2.Les notions de base.....	13
I.3.3.3.Fonctionnement d'un système Client/Serveur.....	14
I.3.3.4.Les apports et les contraintes de l'architecture Client/serveur.....	14
I.3.3.5.Les différentes architectures client/serveur.....	16
I.3.3.5.1. Architecture à deux niveaux.....	16
I.3.3.5.2.Architecture à trois niveaux.....	16
I.3.3.5.3.Architecture multi niveaux.....	16
Conclusion.....	17

Partie 3 : Généralisation sur l'Internet

I. Définition.....	18
--------------------	----

II. Les principaux services offerts par Internet.....	18
III. Intranet et extranet.....	19
III.1. Intranet.....	19
III.2. Extranet.....	20
Conclusion.....	20

Partie 4 : Généralisation sur le Web

I. Introduction.....	21
II. Historique.....	21
III. Définition du Web.....	21
IV. Pourquoi un tel succès du web.....	22
V. Principaux termes du Web.....	22
VI. Site statique et site dynamique.....	24
VI.1. Site statique.....	24
VI.1.1. Définition.....	24
VI.1.2. Fonctionnement d'un site statique.....	24
VI.2. Site dynamique.....	25
VI.2.1. Définition.....	25
VI.2.2. Fonctionnement d'un site dynamique.....	25
VII. Les langages de développement des sites web.....	26
VII.1. Côté client.....	26
VII.2. Côté serveur.....	27
Conclusion.....	28

Chapitre II : Analyse et Conception

I. Introduction.....	29
II. Objectif du projet.....	29
III. Analyse.....	30
III.1. Définition des besoins.....	30
III.1.1. Les besoins non fonctionnels.....	30
III.1.2. Les acteurs et les besoins fonctionnels.....	30
III.2. Identification et structuration des cas d'utilisation.....	31
III.2.1. Définition.....	31
III.2.2. Structuration des cas d'utilisation.....	31
III.2.3. Description des scénarios.....	33
III.2.4. Diagramme de cas d'utilisation général.....	37
IV. Conception.....	38
IV.1. Elaboration des diagrammes de séquence.....	38
IV.2. Diagrammes d'activité pour quelques cas d'utilisation.....	42
IV.2.1. Diagramme d'activité pour le cas d'utilisation « Identification».....	42
IV.2.2. Diagramme d'activité pour le cas d'utilisation « Réserver une place».....	43
IV.2.3. Diagramme d'activité pour le cas d'utilisation « Publier des tarifs».....	44
IV.3. Diagrammes de classe généraux.....	45
V. Mapping Objet-Relationnel.....	49
V.1. Principes d'optimisation et de transformation.....	49
Conclusion.....	51

Chapitre III : Réalisation de l'application

I. Introduction.....	52
II. Environnement de développement.....	52
II.1. WAMP.....	52

III. Les outils utilisés.....	53
III.1. Les langages utilisés.....	53
III.2. Les logiciels.....	54
IV. Implémentation de la base de données.....	56
V. Stockage des données.....	57
VI. Présentation de quelques interfaces.....	60
Conclusion.....	64
Conclusion Générale.....	65
IV. Annexe.....	66
Références Bibliographie.....	73

On entend souvent dire que l'Internet est aussi important pour la société que l'ordinateur personnel, le téléphone ou même, autrefois, la presse d'imprimerie. Et pourtant, il est difficile d'expliquer à quelqu'un qui n'a jamais utilisé le réseau son pouvoir de changer les vies, de créer de nouvelles activités économiques ou de rendre pas mal de tâches très aisées. Pour ceux qui utilisent déjà l'Internet, qui ont franchi le cap des premières frustrations inhérentes à l'apprentissage d'une nouvelle technologie, l'Internet a tous ces pouvoirs, et bien d'autres encore.

[WB.INIW]

La circulation décongestionnée, les problèmes de stationnement, la perte de temps pour rechercher un parking, l'apparition de la notion de commerce électronique, et les avantages d'Internet et les réseaux informatiques ont permis d'améliorer les services fournis à la clientèle par les systèmes de gestion de parkings automobiles, en étendant la portée géographique des services et en offrant une nouvelle source de recettes potentielle.

Le travail que *Serius Net* nous a confié consiste en la conception et la réalisation d'un système de gestion de parkings automobiles afin de fournir la possibilité de consulter les réservations disponibles, s'exprimer librement en laissant des commentaires, et il offre, essentiellement, aux abonnés des places disponibles, et donne aux clients du site le choix de réserver une place dans un parking pendant leurs navigations.

Pour une bonne organisation de notre projet, nous l'avons structuré comme suit :

- * **Chapitre I** : « Généralités » ; qui donne un aperçu sur les parkings et un survol sur les réseaux, internet et le web.
- * **Chapitre II** : « Analyse & conception » ; décrit la conception de notre application.
- * **Chapitre III** : « Réalisation » ; consacré à la réalisation et l'implémentation de l'application et son fonctionnement.

Introduction :

Qu'est ce qu'un parking ? Qu'elles sont les différents types de parkings existants ? Pourquoi louer une place dans un parking ? ... cette partie aura pour objectif de répondre à ces questions on donnant un petit aperçu sur les parkings.

II. Définition :

Un **parking**, ou parc de stationnement (expression recommandée par la Commission générale de terminologie et de néologie en France), ou en Amérique du nord terrain de stationnement ou simplement stationnement, est un espace ou un bâtiment spécifiquement aménagé pour le stationnement des véhicules. On en trouve le plus souvent à côté des bâtiments publics (gare, aéroport..), des lieux de travail, des centres commerciaux ou devant les grandes surfaces pour accueillir les usagers.

Le symbole signalétique presque universel signifiant parking est un **P** majuscule blanc sur un disque ou un carré bleu.

III .Types de parkings :

III.1. Les parkings de surface :

Se situe de plain pied, à l'extérieur, sur l'espace public ou privé. Ce type de parking comprend le stationnement en voirie (places le long d'une rue, d'un quai, etc.) et les espaces dégagés à cette fin entre des bâtiments, ou établis sur des anciens champs, des anciens terrains vagues, etc.

III.2. La zone bleue :

La zone bleue est une zone de stationnement réglementé, limité dans la durée. Il ne s'agit pas d'un parking à proprement parler, mais d'une zone où le stationnement en voirie est autorisé. Peuvent s'y garer les automobilistes qui possèdent un disque de stationnement. Cet accessoire se compose d'une pochette cartonnée ou plastifiée contenant un disque cartonné, agrafé à la pochette en son centre afin de pouvoir tourner. Sur le disque figurent d'une part des heures d'arrivée possibles, d'autre part la durée autorisée et les heures de départ correspondantes, en sachant que la durée de stationnement en zone bleue est généralement limitée à 1h30.

III.3. Les parking fermés ou souterrains :

En ville ou sous les aéroports, sous les bâtiments de certaines zones d'activité, souvent sur plusieurs niveaux, ils permettent d'économiser le foncier. Chaque niveau s'apparente à un parking classique. Des rampes permettent de passer d'un niveau à l'autre. Des ascenseurs ou des escaliers permettent aux occupants des véhicules, une fois ceux-ci garés, de remonter à la surface.

Dans plusieurs pays, les parkings souterrains sont désormais obligatoires pour toutes les constructions d'immeubles dans certains zonages urbains, avec des prescriptions en matière d'aération, lutte contre l'incendie, sorties de secours, etc.

III.4. Les parkings à étages :

Un parking à étages est un bâtiment construit en extérieur. Son mode de fonctionnement ressemble beaucoup à celui du parking souterrain à la différence qu'il ne demande pas de lourds travaux de creusement. Là aussi des rampes permettent de monter ou de descendre les étages en voiture, et il y a des ascenseurs et des escaliers pour les piétons.

III.5. Les parking-relais :

Les parkings-relais sont en effet positionnés à proximité immédiate de gares routières ou ferroviaires, de terminus de métro... La logique de cette implantation est de favoriser l'inter modularité (la mixité des moyens de transports) afin de désengorger la circulation dans les centres urbains.

IV. Pourquoi louer une place de parking ?

- **Gain des temps :** Fini les pertes de temps à chercher une place de parking, vous êtes à proximité de l'endroit où vous voulez vous rendre. Vous êtes détendus et moins stressés.
- **Economie :** Fini l'angoisse de trouver son véhicule couvert de PV ou pire encore, à la fourrière. Louer un parking privé est plus économique que d'utiliser des solutions de stationnement classiques.
- **Voiture en sécurité :** Votre voiture est mieux protégée dans un parking privé qu'à l'extérieur, et elle a moins de risques de se faire abimer (griffes, rétros cassés, vol..).
- **Un geste pour la planète :** Réduisez vos émissions de gaz à effets de serre et de polluants.
- **Une circulation décongestionnée.**

V. Gestion des parkings pour amélioration de leurs utilisations :

Les automobilistes ont besoin d'informations en continu concernant les disponibilités, les lieux et tarifs. Un guide d'accès de la ville fournira toutes ces informations. Ce genre de guide est en général développé par des agences publiques spécialisées.

Les informations sur ces parkings fournies aux automobilistes comprennent des cartes, signalisations, kiosque, brochures, GPS et bien d'autres systèmes de communication électroniques. Ces outils peuvent améliorer la commodité, la sécurité, ainsi que proposer d'autres solutions de parking à moindre coût. Les zones industrielles ou résidentielles mettent en place des stratégies adaptées afin d'améliorer la circulation pendant les heures de pointe.

Conclusion :

Nous venons de définir dans ce chapitre la notion de parking, et la nécessité de louer une place dans ce dernier, et avec l'informatisation des systèmes de gestion de parkings nous allons présenter dans le chapitre suivant les diverses fonctionnalités des réseaux informatiques.

I.1 Introduction : [ING info 2009]

Vu l'évolution rapide de l'information et l'intérêt croissant de vouloir gagner en temps, de conserver les données, de limiter le nombre d'employés et pas mal d'autres raisons, ont poussé petites, moyennes et grandes entreprises à chercher des solutions informatiques capables de répondre à leurs besoins. Les réseaux informatiques sont devenus très vaste aujourd'hui, ils sont employés dans toutes les entreprises privés ou étatiques. Les réseaux couvrent complètement la planète grâce à divers équipements et l'apparition de l'Internet et le Web offrent des services énormes.

Ce chapitre aura pour objectif de présenter quelques notions sur les réseaux informatiques en premier lieu puis donnera un aperçu sur l'Internet en général et sur le web en particulier.

I.2 Les réseaux informatiques :

Définition

Le terme « **réseau** » définit un ensemble d'entités interconnectées les unes avec les autres. Un réseau permet ainsi de faire circuler des éléments entre chacune de ces entités selon des règles bien définies.

Un réseau informatique : ensemble d'ordinateurs (ou périphériques) interconnectés entre eux grâce à des lignes physiques ou sans fils et échangeant des informations sous forme de données numériques (des valeurs binaires, c'est à dire codées sous forme de signaux pouvant prendre des valeurs : 0 et 1).

I.2.1 Intérêt des réseaux

Les réseaux permettent :

- ü Le partage de ressources (fichiers, applications ou matériels, connexion à internet, etc.)
- ü La communication entre personnes (courrier électronique, discussion en direct, etc.)
- ü La communication entre processus (entre des ordinateurs industriels par exemple)
- ü La garantie de l'unicité et de l'universalité de l'accès à l'information (bases de données en réseau)

Les réseaux permettent aussi de standardiser les applications, on parle généralement de groupware pour qualifier les outils permettant à plusieurs personnes de travailler en réseau. Par exemple la messagerie électronique et les agendas de groupe permettent de communiquer plus efficacement et plus rapidement.

I.2.2 Classification des réseaux : [WB.APINF]

La classification des réseaux peut être faite selon plusieurs critères, la plus évidente est celle qui considère le critère distance ou taille, la classification peut être aussi en fonction de la topologie:

I.2.2.1. les réseaux filaires :

I.2.2.1.1 En fonction de la taille :

On compte généralement quatre catégories de réseaux informatiques différenciés par la distance qui séparent les points de réseau :

- **PAN** (Personal Area Network) :

C'est les systèmes de communication qui permettent une connexion à quelques mètres de l'utilisateur. Les technologies qui permettent ce type de liaison sont : Bluetooth, USB, infrarouge.

Le PAN concerne la connectivité entre les PC, les téléphones portables, les assistants numériques, les imprimantes, ainsi que d'autres terminaux pouvant fournir notamment des liaisons sans fils pour la synchronisation des données, l'accès à internet et l'impression.

- **LAN** (Local Area Network) :

Système de communication permettant de relier une centaine d'ordinateurs et de périphériques, il peut s'étendre de quelques mètres à quelques kilomètres et correspond au réseau d'une entreprise. Il peut se développer sur plusieurs bâtiments et permet de satisfaire tous les besoins internes de cette entreprise.

- **MAN** (Metropolitan Area Network) :

Un réseau métropolitain interconnecte plusieurs lieux situés dans une même ville, par exemple les différents sites d'une université ou d'une administration, chacun possédant son propre réseau local. En résumé c'est le résultat 'interconnexion d'un ensemble de réseaux locaux.

- **WAN** (Wide Area Network)

C'est un réseau étendu qui permet de communiquer à l'échelle d'un pays, ou de la planète entière, les infrastructures physiques pouvant être terrestres ou spatiales à l'aide de satellites de télécommunications. Les entreprises privées ont leurs propres réseaux étendus pour relier les bureaux éloignés ou utilisent internet pour la connectivité sachant que l'internet est le plus grand WAN.

I.2.2.1.2. En fonction de la topologie : [WB.CCM]

L'arrangement physique, c'est-à-dire la configuration spatiale du réseau est appelée **topologie physique**. On distingue généralement les topologies suivantes :

- **Topologie en bus :**

Une topologie en bus est l'organisation la plus simple d'un réseau. En effet, dans une topologie en bus tous les ordinateurs sont reliés à une même ligne de transmission par l'intermédiaire de câble, généralement coaxial. Le mot « bus » désigne la ligne physique qui relie les machines du réseau.



Figure I.1 : topologie en bus.

§ **Avantage :**

- Facile à mettre en œuvre et à étendre.
- Utilisable pour des réseaux temporaires (installation facile).
- Présente l'un des coûts de mise en réseau le plus bas.

§ **Inconvénients :**

- Longueur du câble et nombre de stations limité.
- Un câble coupé peut interrompre le réseau.
- Les coûts de maintenance peuvent être importants à long terme.
- Les performances se dégradent avec l'ajout de stations.
- Faible sécurité des données transitant sur le réseau (toutes les stations connectées au bus peuvent lire toutes les données transmises sur le bus).
- Un virus sur le réseau peut affecter toutes les stations (mais pas plus qu'avec une topologie en anneau).
- Il faut utiliser un bouchon pour les extrémités du bus (la boucle doit être fermée).

- **Topologie en étoile :**

Les stations du réseau communiquent bi-directionnellement avec le serveur. C'est une liaison point à point entre le serveur et chaque machine. Deux stations peuvent échanger des données à condition d'en passer par le serveur.



Figure I.2 : topologie en étoile.

§ **Avantage :**

- Ajout facile de nœuds.
- Localisation facile des pannes.
- Le débranchement d'une connexion ne paralyse pas le reste du réseau.
- Simplicité éventuelle des équipements au niveau des nœuds : c'est le concentrateur qui est intelligent.

§ **Inconvénients :**

- Plus onéreux qu'un réseau à topologie en bus (achat du concentrateur et d'autant de câbles que de nœuds).
- Si le concentrateur est défectueux, tout le réseau est en panne.

• **Topologie en anneau :**

Dans ce type de topologie, les ordinateurs sont situés sur une boucle et communiquent chacun à leur tour.

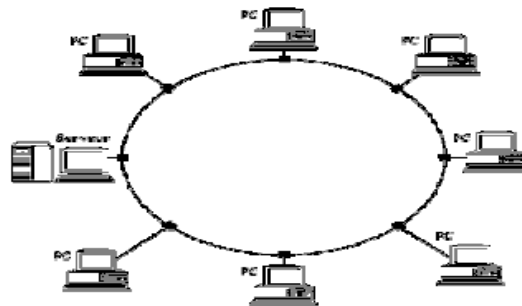


Figure I.3 : topologie en anneau.

En réalité, dans une topologie anneau, les ordinateurs ne sont pas reliés en boucle, mais sont reliés à un répartiteur (appelé *MAU*, *Multistation Access Unit*) qui va gérer la communication entre les ordinateurs qui lui sont reliés en impartissant à chacun d'entre-eux un temps de parole.

§ **Avantage :**

- Minimisation de la quantité de câble nécessaire.
- Simplicité du protocole, en évitant la gestion des collisions.

§ **Inconvénient :**

- Le retrait ou la panne d'une entité active, paralyse le trafic du réseau.
- Difficulté de planification et d'insertion des stations.

• **Topologie en arbre :**

Aussi connu sous le nom de *topologie hiérarchique*, le réseau est divisé en niveaux. Le sommet, le haut niveau, est connectée à plusieurs nœuds de niveau inférieur, dans la hiérarchie. Ces nœuds peuvent être eux-mêmes connectés à plusieurs nœuds de niveau inférieur. Le tout dessine alors un arbre, ou une arborescence.

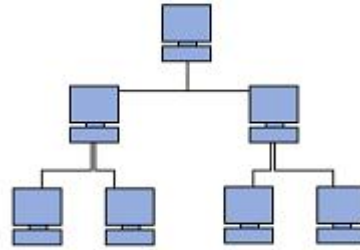


Figure I.4: topologie en arbre.

§ Avantage :

- Les mêmes avantages que la topologie en étoile.
- Peu réunir plusieurs sous réseaux en étoile.

§ Inconvénient :

- Plus d'utilisateurs, moins de performance.
- Si un poste tombe en panne tous les sous-réseaux inférieurs de ce poste seront exclus du réseau.

• **Topologie maillé :** [WB.WIKIV]

Une topologie maillée, est une évolution de la topologie en étoile, elle correspond à plusieurs liaisons point à point. Une unité réseau peut avoir (1, N) connexions point à point vers plusieurs autres unités. Chaque terminal est relié à tous les autres. L'inconvénient est le nombre de liaisons nécessaires qui devient très élevé.

Cette topologie se rencontre dans les grands réseaux de distribution (Exemple : Internet). L'information peut parcourir le réseau suivant des itinéraires divers, sous le contrôle de puissants superviseurs de réseau, ou grâce à des méthodes de routage réparties. En cas de rupture d'un lien, l'information peut quand même être acheminée.

Elle existe aussi dans le cas de couverture Wifi. On parle alors bien souvent de topologie mesh mais ne concerne que les routeurs Wifi.

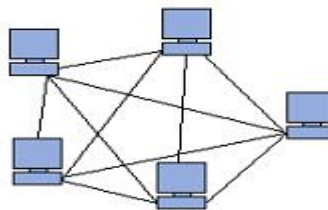


Figure I.5: topologie maillée.

§ Avantage :

- Cette méthode garantit le transfert des données en cas de panne d'un nœud.
- liaison directe entre les machines.

§ Inconvénient :

- Elle est complexe à mettre en œuvre et ne peut pas être utilisée dans les réseaux locaux.
- Nombreuses connexions dues au fait que le nombre de liaisons nécessaire devient très élevé.
- Le coût s'avère cher en cas d'augmentation du nombre de postes connectés ou de la distance.

I.2.2.2. les réseaux sans fils : [DJE/09]

Définition :

Se sont des réseaux filaires dans lesquels au moins une liaison filaire (câbles, fibre optique) est remplacée par une liaison radio, permettant ainsi la mobilité de l'ordinateur concernée par cette liaison.

I.2.2.2.1. Réseau personnel ou « Bluetooth » :

C'est généralement un réseau d'interconnexion des différents composants d'un ordinateur à l'aide d'une liaison radio très courte distance. Ce système permet donc de connecter un ordinateur et ses périphériques (clavier, souris, imprimantes, camera numérique scanner, casque, téléphone portable, ...) en les plaçant dans la zone de couverture radio du système (**FigI.6**).

Ce réseau utilise le modèle Maître/Esclave. C'est l'ordinateur (Maître) qui gère la communication avec les différents périphériques (Esclave).

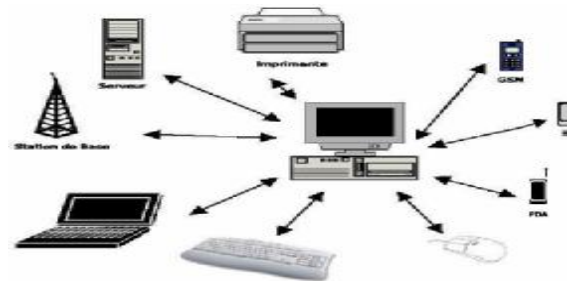


Figure I.6: Réseau Bluetooth

I.2.2.2.2. Réseau local sans fils :

Généralement appelés Wireless Local Networks. Dans ce cas chaque ordinateur est doté d'une antenne radio grâce à laquelle il peut communiquer avec les autres ordinateurs via une antenne commune appelée Station de base (point d'accès) leur donnant ainsi accès au réseau filaire (**Fig I.7**). Toutefois, si les ordinateurs sont très proches l'un de l'autre, ils peuvent communiquer entre eux en mode Peer to Peer (poste à poste).



Figure I.7: réseau local sans fils.

§ **Remarque :** la technologie la plus connue dans ce type de réseau est Wifi qui offre des débits de l'ordre de 54 Mbps sur une portée de plusieurs centaines de mètres

I.2.2.2.3. Réseau métropolitains sans fils :

Appelés aussi Wireless Métropolitain Area Networks ou Boucle Locale Radio. La technologie la plus connue est le WIMAX qui offre des débits de l'ordre 70Mbps pour une portée de plusieurs kilomètres.



Figure I.8 : Réseau métropolitain sans fils.

I.2.2.2.4. Réseau Wan sans fils :

Un bon exemple de ce type de réseau est le téléphone mobile. Il est généralement constitué d'un réseau de transport longue distance filaire auquel accèdent les clients (ordinateurs ou téléphones mobiles) via des antennes stations de base fixes (**Fig I.9**).

De plus, les clients peuvent être mobiles. En d'autres termes, durant une session de travail, un client peut sortir de la zone de couverture de l'antenne d'une station de base (Hand off) puis passer à la zone de couverture d'une autre station de base.

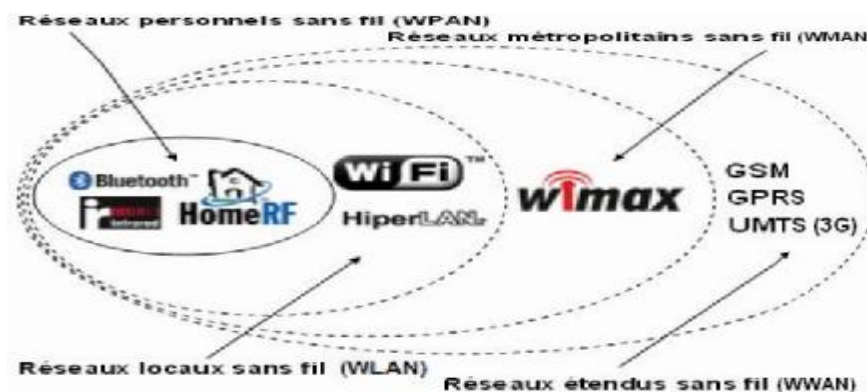


Figure I.9 : Réseau sans fils.

§ **Remarque** : les principales technologies sont :

- Global System for Mobile communication (GSM).
- General Packet Radio Service (GPRS).
- Universal Mobile Telecommunication System (UMTS).

I.2.2.2.5. Réseau ad hoc :

Ces réseaux sont caractérisés par le fait que chaque ordinateur le constituant est lui-même un équipement de communication (Routeur) du réseau en plus d'être un Hôte. Ces ordinateurs, en tant que Routeur, sont équipés d'antennes leur permettant d'effectuer des liaisons radio avec tous les ordinateurs situés dans leur voisinage, formant ainsi un réseau sans fils. De plus ces ordinateurs sont mobiles, ce qui induit une structure dynamique des réseaux entièrement sans fils et mobiles.

I.3 Les modèles de références :

I.3.1. Le modèle OSI :

Définition : [LAI 02]

Un modèle d'architecture pour les protocoles de communication a été développé par l'ISO (International Standards Organization) entre 1977 et 1984. Ce modèle sert souvent de référence pour décrire la structure et le fonctionnement des protocoles de communication.

Les constituants de ce modèle sont si largement employés qu'il est difficile de parler de réseaux sans y faire référence.

- **Description du modèle :**

Le modèle normalisé connu sous le nom de modèle OSI (Open Systems Interconnection ou Interconnexion de systèmes ouverts) repose sur l'empilement de 7 couches pouvant communiquer verticalement entre elles (**Fig I.10**).

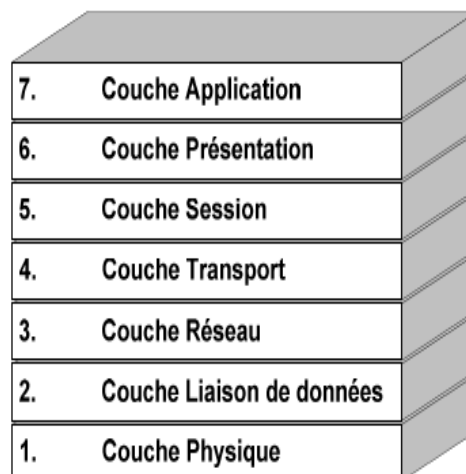


Figure I.10 : le modèle OSI en 7 couches.

- **La couche physique** : Cette couche offre les services de l'interface entre l'équipement de traitement informatique (ordinateur ou terminal) et le support physique de transmission. L'unité de transfert gérée par cette couche est l'information élémentaire binaire.

- **La couche liaison de données** : Pour qu'une succession de bits prenne sens, cette couche définit des conventions pour délimiter les caractères ou des groupes de caractères. Elle a, donc pour but d'acheminer sur une voie physique toutes les informations qui lui sont transmises par la couche supérieure. L'unité de traitement à ce niveau est appelée trame de données.
- **La couche réseau** : Cette couche permet de gérer le sous-réseau, la façon dont les paquets (ensemble de trames) sont acheminés de la source au destinataire. La couche réseau a de ce fait pour rôle essentiel d'assurer les fonctions de routage.
- **La couche transport** : Elle s'occupe de l'acheminement des informations (messages) de bout en bout via le réseau (entre expéditeur et destinataire) de façon indépendante du type de réseau.
- **La couche session** : Cette couche effectue la mise en relation de deux applications et fournit les protocoles nécessaires à la synchronisation des échanges, leur arrêt et leur reprise. Elle permet d'attribuer un numéro d'identification à l'échange d'une suite de messages entre applications.
- **La couche présentation** : Cette couche s'occupe de la représentation des données, elle gère tout ce qui est transformation de données, formatage, conversion et syntaxe dans un langage de telle sorte que les deux clients qui dialoguent se comprennent.
- **La couche application** : Cette couche définit la façon dont les applications accèdent au réseau, elle fournit une fenêtre à travers laquelle une application obtient un accès à l'ensemble des services des couches inférieures : transfert de fichiers, partage de ressources...

I.3.2. Le modèle TCP/IP : [WB.USSK]

Dans les années 70, DOD (Department Of Defense), devant le foisonnement de machines utilisant des protocoles de communication différents et incompatibles, décide de définir sa propre architecture. L'architecture TCP/IP est à la source du réseau Internet de nombreux réseaux privés. Cette architecture comme celle d'OSI est répartie en couches.

TCP/IP désigne communément une architecture réseau, mais cet acronyme désigne en fait deux protocoles étroitement liés :

- Un protocole de transport, TCP (Transmission Control Protocole) qu'on utilise par-dessus un protocole réseau.
- IP (Internet Protocol).

Ce qu'on entend par modèle TCP/IP, c'est en fait une architecture réseau en quatre couches dans lesquelles les protocoles TCP et IP jouent un rôle prédominant, car ils en constituent l'implémentation la plus courante.

- **La couche accès réseaux** : Cette couche concerne la connexion physique, est directement liée au réseau utilisé (Ethernet, réseau à jeton, etc.) Cette couche peut être considérée comme la fusion des couches *Liaison de données* et *physique* du modèle OSI. A ce niveau on parle de trames d'information.
- **La couche Inter-Réseau** : Cette couche est responsable de l'adressage logique du réseau, de l'acheminement de l'information d'un nœud du réseau à un autre. Les unités logiques d'information véhiculées par cette couche sont appelées datagramme.

- **La couche transport** : Cette couche parfois appelée couche hôte ou service provider layer ou l'on trouve deux protocoles TCP et UDP, est responsable du service de transmission fiable de données. Le terme segment est utilisé pour désigner les paquets d'informations.
- **La couche application** : Cette couche regroupe un ensemble d'applications liées aux réseaux TCP/IP.

Cette couche regroupe les trois couches hautes du modèle OSI (Application, Présentation et Session).

I.3.3. Architecture client/serveur : [BEBA/09]

I.3.3.1 Définition :

L'architecture client/serveur est une architecture informatique qui désigne un mode de communication entre plusieurs unités fonctionnelles qui émettent des requêtes et d'autres unités qui traitent ces requêtes.

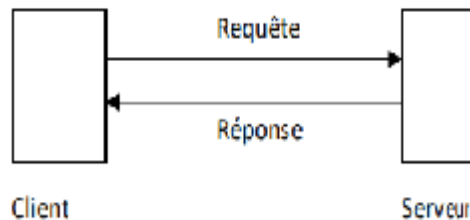


Figure I.11 : Le dialogue client/serveur

I.3.3.2 Les notions de base :

§ Le serveur :

On appelle logiciel serveur un programme qui offre un service sur le réseau. Le serveur accepte des requêtes, les traite et renvoie le résultat au demandeur. Le terme serveur s'applique à la machine sur lequel s'exécute le logiciel serveur.

§ Le client :

On appelle logiciel client un programme qui **utilise le service** offert par un serveur. Le client envoie une requête et reçoit la réponse. Le client peut-être raccordé par une liaison temporaire.

§ Réponse :

C'est un message transmis par un serveur à un client suite à l'exécution d'une opération contenant les paramètres de l'opération.

§ Requête :

C'est un message transmis par un client à un serveur décrivant l'opération à exécuter.

§ Le middleware :

Ensemble des services logiciels construits au-dessus d'un protocole de transport afin de permettre l'échange de requêtes et des réponses associées entre client et serveur de manière transparente.

I.3.3.3 Fonctionnement d'un système Client/Serveur :

Un système client/serveur fonctionne selon le schéma suivant :

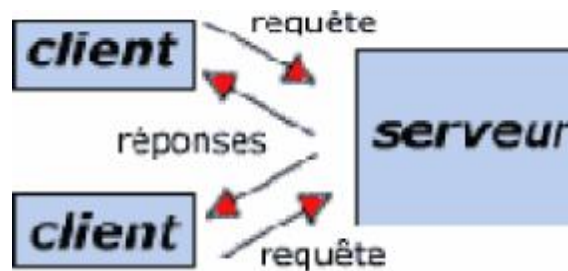


Figure I.12 : Système client/serveur

- ü Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur.
- ü Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port.

I.3.3.4 Les apports et les contraintes de l'architecture Client/serveur :

a) Les apports :

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

§ Des ressources centralisées :

Étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction.

§ Une meilleure sécurité :

Car le nombre de points d'entrée permettant l'accès aux données est moins important.

§ Une administration au niveau serveur :

Les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.

§ Un réseau évolutif :

Grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure.

§ Capacité de traitement :

Le modèle client/serveur s'appuie sur le progrès réalisé dans le monde de microprocesseurs. Les techniques qu'il met en œuvre nécessitent des capacités de traitement importantes que les microprocesseurs d'aujourd'hui ont capables de fournir.

§ Standardisation :

En s'appuyant sur les normes standards, le modèle client/serveur assure la portabilité des applications entre systèmes distincts, et garanti leur inter portabilité.

§ Capacité de mémorisation :

Dans une architecture client/serveur, les serveurs doivent gérer simultanément les contextes applicatifs de plusieurs postes clients. Cette architecture sera d'autant mieux adaptée à l'environnement transactionnel lourd où les serveurs disposent de grandes capacités mémoires.

§ Productivité du développeur :

L'environnement client/serveur facilite le processus de développement applicatif. Cela est du notamment à la séparation fonctionnelle qui constitue le principe de base du modèle client/serveur.

§ Portabilité et interopérabilité :

Le modèle client/serveur assure le partage d'une même application sur différentes plates formes, échanger des informations et faire interagir plusieurs applications dans des environnements hétérogènes.

b) Les contraintes :

L'architecture client/serveur a tout de même quelques lacunes parmi lesquelles :

§ Un coût élevé :

Dû à la technicité du serveur.

§ Un maillon faible :

Le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui ! Heureusement, le serveur a une grande tolérance aux pannes (notamment grâce au système RAID).

§ Complexité de mise en œuvre :

Une complexité qui se manifeste notamment lors du découpage des traitements entre partie client et partie serveur.

I.3.3.5 Les différentes architectures client/serveur : [WB.CCM]

I.3.3.5.1 Architecture à deux niveaux :

L'architecture à deux niveaux (aussi appelée architecture 2-tiers, tiers signifie rangé en anglais) caractérise des systèmes client/serveur pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources.

Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service.

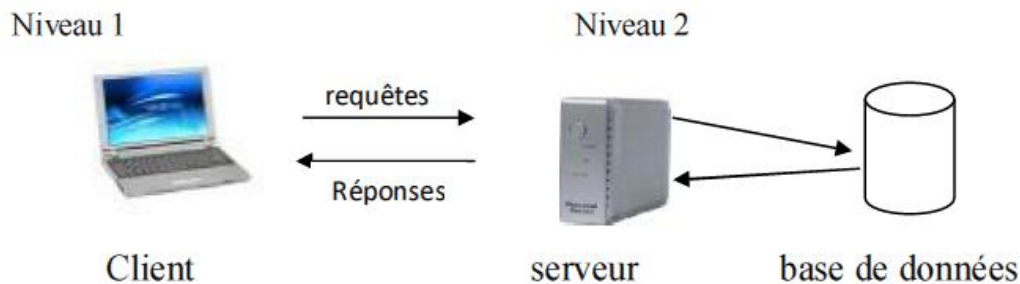


Figure I.13 : architecture client/serveur à deux niveaux.

I.3.3.5.2 Architecture à trois niveaux :

Dans l'architecture à trois niveaux (appelé architecture à 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

- ü **Un client** : l'ordinateur demande des ressources.
- ü **Le serveur d'application** (appelé également middleware) : le serveur chargée de fournir la ressource mais faisant appel à un autre serveur.
- ü **Le serveur de données** fournissant au serveur d'application dont il a besoin.

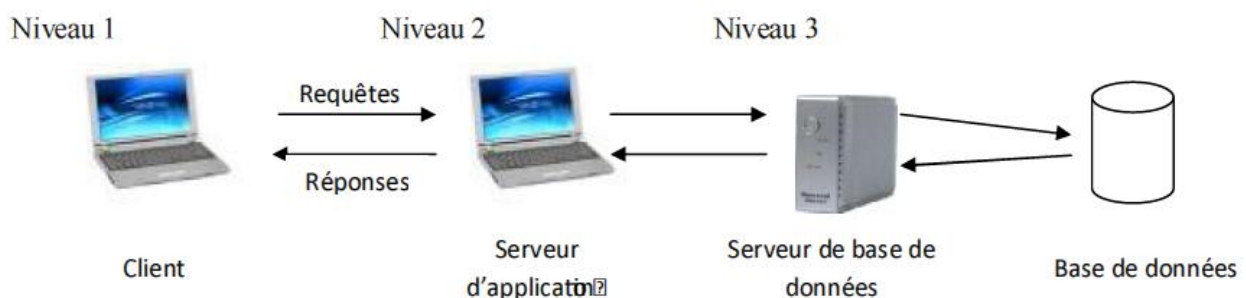


Figure I.14 : architecture client/serveur à trois niveaux.

I.3.3.5.3 Architecture multi niveaux :

Dans l'architecture à trois niveaux, chaque serveur (niveau 2 et 3) effectue une tâche (un service) spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à n niveaux.

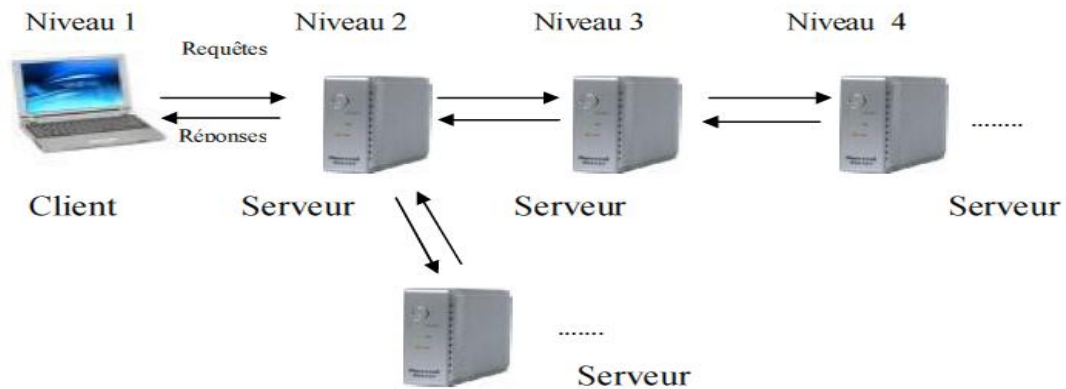


Figure I.15 : Architecture client/serveur multi niveaux.

Conclusion :

Nous venons de voir dans cette partie quelques concepts de base des technologies de l'information et de la communication à savoir les réseaux et leurs architectures, la notion de client/serveur ainsi que les différents protocoles utilisés pour cette communication.

Le plus grand réseau planétaire de cette interconnexion est **Internet** qui fera l'objet de notre prochaine partie.

I. Définition : [LIV 08]

Le mot Internet vient du mot anglais « *International Network* ».c'est un réseau international d'ordinateurs, qui communiquent entre eux grâce à un protocole d'échange de données standard (TCP/IP).Les différents ordinateurs branches au réseau communiquent entre eux, et la démarche est transparente pour l'utilisateur.

II. Les principaux services offerts par Internet :

L'internet offre de multiservices tels que :

- **Le www(world wide web) :**

Constituent une véritable bibliothèque virtuelle d'où un explorateur peut télécharger un très grand nombre de documents et de fichiers multimédia et des pages écrite en HTML (HyperText Markup Langage).Une des forces des réseaux est qu'il donne accès a tous les services d'Internet ou en partie, il utilise le protocole http (HyperText Transfert Protocol) pour l'échange d'informations entre le logiciel client c'est-à-dire le navigateur et le logiciel serveur.

- **Le chat :**

Discuter en temps réel avec une ou plusieurs personne au même temps par envoie de texte écrit.

- **Le transfert de fichiers :**

FTP (File Transfert Protocol) c'est le service d'échange de fichiers, il permet de déposer des fichiers sur une machine distante (fonctionnant sur des systèmes et des architectures divers), mais aussi de télécharger des fichiers sur sa machine.

- **La visioconférence :**

Discuter avec le son et l'image avec d'autre interlocuteur en temps réel.

- **Le forum Usenet :**

C'est la participation via Internet à des groupes d'étude et des réflexions sur des sujets et intérêts communs.

- **E-mail :**

C'est l'échange de courrier via des boites aux lettres électronique entre ordinateurs reliés à internet.

- **Le WAIS (Wide Area Informatique Système) :**

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.

- **Multimédia :**

Ensemble des techniques et des produits qui présentent l'information sous forme combinée de sons, d'images, d'animations et de vidéos.

Le domaine de l'édition multimédia est aujourd'hui en grande expansion, grâce à la popularisation d'internet et à la multiplication des CD-ROM à fonction pédagogique ou ludique.

III. Intranet et extranet : [LIV 08]

III.1. Intranet :

Ø Définition :

Un **intranet** est un ensemble de services internet (par exemple un serveur web) internes à un réseau local, c'est-à-dire accessibles uniquement à partir des postes d'un réseau local, ou bien d'un ensemble de réseaux bien définis, et invisibles (ou inaccessibles) de l'extérieur. Il consiste à utiliser les standards client-serveur de l'internet (en utilisant les protocoles TCP/IP), comme par exemple l'utilisation de navigateurs internet (client basé sur le protocole HTTP) et des serveurs web (protocole HTTP), pour réaliser un information interne à une organisation ou une entreprise.

Un intranet repose généralement sur une architecture à trois niveaux, composée :

- de clients (navigateur internet généralement).
- d'un ou plusieurs serveurs d'application (middleware): un serveur web permettant d'interpréter des scripts CGI, PHP, ASP ou autres, et les traduire en requêtes SQL afin d'interroger une base de données.
- d'un serveur de bases de données.

De cette façon, les machines clientes gèrent l'interface graphique, tandis que les différents serveurs manipulent les données. Le réseau permet de véhiculer les requêtes et les réponses entre clients et serveurs.

Un intranet possède naturellement plusieurs clients (les ordinateurs du réseau local) et peut aussi être composé de plusieurs serveurs. Une grande entreprise peut par exemple posséder un serveur web pour chaque service afin de fournir un intranet composé d'un serveur web fédérateur liant les différents serveurs gérés par chaque service.

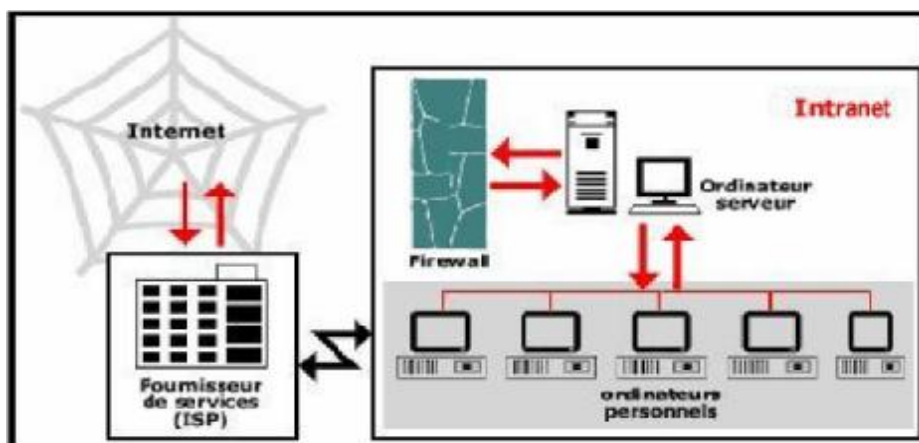


Figure I.16 : Un intranet.

III.2.Extranet :

Ø Définition :

Un **extranet** est une extension du système d'information de l'entreprise à des partenaires situés au-delà du réseau.

L'accès à l'extranet doit être sécurisé dans la mesure où cela offre un accès au système d'information à des personnes situées en dehors de l'entreprise.

Il peut s'agir soit d'une authentification simple (authentification par nom d'utilisateur et mot de passe) ou d'une authentification forte (authentification à l'aide d'un certificat). Il est conseillé d'utiliser HTTPS pour toutes les pages web consultées depuis l'extérieur afin de sécuriser le transport des requêtes et des réponses HTTP et d'éviter notamment la circulation du mot de passe en clair sur le réseau.

Un extranet n'est donc ni un intranet, ni un site internet. Il s'agit d'un système supplémentaire offrant par exemple aux clients d'une entreprise, à ses partenaires ou à des filiales, un accès privilégié à certaines ressources informatiques de l'entreprise par l'intermédiaire d'une interface Web.

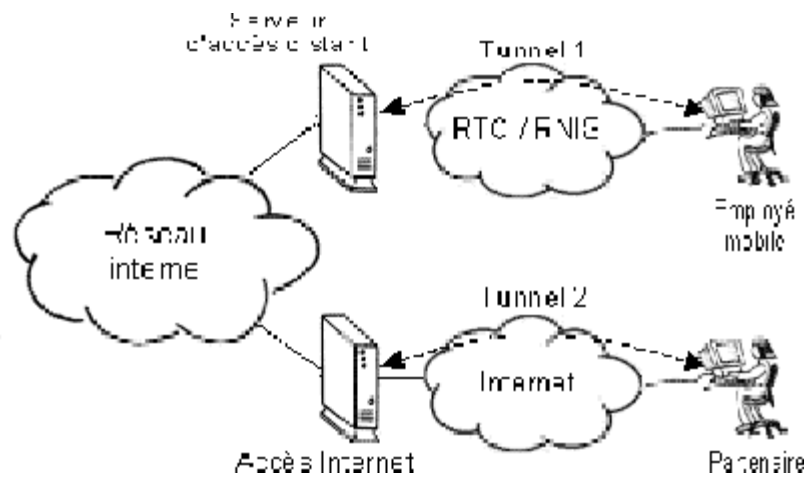


Figure I.17 : Un extranet.

Conclusion :

Nous venons de donner dans cette partie une description de ce qui constitue l'**Internet** et nous avons présenté ses différents services, dont le web auquel nous consacrerons la partie suivante.

I. Introduction : [ING télécom-2000]

Le développement rapide du Web a créé de ressources d'informations nouvelles et importantes, de ce fait toutes les sociétés ont voulu avoir leur « vitrine » sur Internet. Le moyen le plus simple et le plus rapide était de créer un site Web qui en général présente des informations de base sur la société, le groupe ou l'association : activité, adresse, quelques produits et même les tarifs. Toutes ces sociétés ont voulu faire leur publicité.

II. Historique :

Le Web a été inventé en 1989 par TIM BERNERS-LEE et ROBERT CAILLIAU pour le centre européen de recherche nucléaire (CERN) (Conseil Européen pour la Recherche Nucléaire), en Suisse.

Le Web, WWW est aujourd'hui le service le plus connu et le plus utilisé dans l'Internet. Son succès est dû à plusieurs facteurs : l'appareil implicite des logiciels de navigation, la présentation conviviale et multimédia des documents mis à disposition des internautes et la puissance du système en lui-même. Il donne accès à un nombre incalculable de documents sur des thèmes les plus variés, on trouve tout ce qu'on cherche sur Internet.

III. Définition du Web :

Le World Wide Web, communément appelé le Web, parfois la toile, littéralement la « toile d'araignée mondiale », est un système hypertexte public fonctionnant sur Internet et qui permet de consulter, avec un navigateur, des pages mises en ligne dans des sites. L'image de la toile vient des hyperliens qui lient les pages Web entre elles.

Le Web n'est qu'une des applications d'Internet, avec le courrier électronique, la messagerie instantanée, ... etc.

Il a été inventé plusieurs années après Internet, mais c'est le web qui a rendu les médias grand public attentifs à Internet. Depuis, il est fréquemment confondu avec Internet, en particulier le mot toile est souvent utilisé de manière très ambiguë.

La navigation dans le monde web se fait par l'intermédiaire d'un programme appelé navigateur (browser). Chaque document (ou page) est identifié par une adresse précisée de la forme :

http://www.site.dom/répertoire/nom_page.html

Http : protocole utilisé.

www.site.dom : nom du serveur.

Répertoire : répertoire.

Nom page.html : fichier.

IV. Pourquoi un tel succès du web :

- Ø La navigation hypertexte étendue : les liens peuvent s'appliquer à des documents hébergés sur d'autres sites au sens géographique du terme.
- Ø Support multimédia : images fixes, animées, sons, vidéos, etc.
- Ø Multi-plateforme : les codages utilisés permettent d'utiliser toutes les machines existantes.
- Ø L'unité de visualisation du contenu (navigateur).
- Ø Les liens hypertextes.
- Ø Les systèmes d'adresses URL.
- Ø Intégration de tous les services qui existent (messagerie, FTP, news, etc.).
- Ø Gratuité des protocoles : Sont offerts au domaine public comme « http ».

V. Principaux termes du Web :

- Ø **Page Web** : les pages Web représentent l'ensemble des informations et des connaissances proposées au visiteur d'un site Web. Plus généralement, le contenu des pages est un ensemble de documents, de composants multimédias (texte, son, vidéo, etc.) ou de données diverses. La technologie Internet permet de disposer de ce contenu en ligne ou de le télécharger.

- **Page web statique** :

Ce sont des pages web écrites en HTML invariantes, proposé à l'avance. Autrement dit, à chaque fois qu'un client affiche une page, cette dernière se présente de la même manière tant que le code HTML n'a pas été mis à jour par l'auteur.

Les fichiers du code HTML ne sont pas en mesure d'interagir avec le serveur. Si par exemple, on veut récupérer des informations liées à l'utilisateur pour les stocker dans une base de données, il va falloir mettre en œuvre d'autres outils capables de supporter ce type d'interaction. La mise à jour des pages HTML est très difficile et doit s'effectuer via un éditeur, de nombreux fichiers sont liés, la cohérence des liens est difficile à gérer.

- **Page web dynamique** :

Les lourdeurs de la consultation et la mise à jour des pages web statiques ainsi que le faible volume d'informations qu'elles contiennent ont rapidement conduit les concepteurs à proposer des « sites web dynamiques ». Dans ce cas, les pages web ne sont pas fabriquées à l'avance.

Cette composante peut recouvrir des aspects tels que la diffusion d'informations évaluant dans le temps ou la récupération des renseignements saisis par l'utilisateur pour décider de déclencher telle ou telle autre action.

Les auteurs chargés de la conception de telles pages web doivent faire appel à des ressources de programmation dépassant le cadre du HTML, qui connaît quelques limitations dont la principale est de proposer des pages statiques. Ces sources de programmation peuvent

être des langages de scripts, les outils d'interfaçage web/base de données ou autres. Voici une liste forcément incomplète des fonctionnalités dynamiques de base :

- Û Pages d'accueil personnalisable.
- Û Accès à des bases de données.
- Û Transactions commerciales.
- Û Formulaires de renseignements.
- Û Services d'inscription.
- Û Annonces journalières.

- Ø **Site Web** : un site Web est un ensemble de pages Web liées entre elles par une structure cohérente, publiées par un propriétaire (une entreprise, une administration, une association, un particulier...) et qui sont hébergées sur un serveur particulier.
- Ø **Page d'accueil** : c'est la page d'introduction d'un site Web qu'on peut aussi appeler « page de couverture ». Elle est très importante car elle détermine la première impression de l'internaute.
- Ø **Site Internet** : c'est un ensemble de données (texte, image, son, vidéo, etc....) hébergées sur un serveur ayant une adresse IP et accessible aux internautes. Il existe différents types de sites selon leurs contenus et la nature des services qu'ils offrent. Citons les sites FTP, Web, etc.
- Ø **Serveur Internet** : c'est l'ensemble des ressources matérielles (espace disque, connexion Internet, contenu du site, etc.) et logicielles (les programmes) qui servent les clients d'où émanent les demandes de services.
- Ø **URL (Uniform Resource Locator)** : c'est une adresse universelle d'une page Web. Elle a le format suivant :
Protocole utilisé://nom du serveur/chemin d'accès au fichier
Exemple : <http://WWW.vacance.fr/index.html>
Une URL s'apparente à un nom de serveur et une information sur le type de protocole de réseau à utiliser pour atteindre la ressource.
- Ø **HTML (Hypertexte Mark up Language)** : les pages Web (ayant l'extension html) sont écrites en HTML. C'est un langage de balisage hypertextuel permettant d'inclure outre du texte, des graphiques, son, vidéo et des pointeurs vers d'autres pages du même site ou d'un site distant.
- Ø **Hypertexte** : c'est une forme nom séquentielle non narrative d'organisation et d'accès à l'information représentée par un réseau de liens associatifs qui forment une structure d'hypertexte.
- Ø **Hypermédia** : le terme hypermédia désigne un hypertexte dont on insiste sur la nature multimédia de son contenu informationnel.

- Ø **Hyper document** : il désigne un ensemble de documents multimédia ayant une organisation et un fonctionnement hypertextuel.
- Ø **Lien hypertexte** : c'est un pointeur qui pointe sur un nœud de destination, souvent il est sous forme d'une icône, symbole, petite image, un mot dans un texte, etc. il permet d'explorer le Web avec un simple clic de souris. On distingue :
 - Les liens internes qui renvoient à d'autres pages du même site.
 - Les liens externes qui renvoient à d'autres pages d'un autre site.
 - Les liens vers un e-mail qui permettent à l'internaute d'envoyer un e-mail directement depuis la même page.
 - liens de téléchargement qui permettent de télécharger des documents depuis la même page.
- Ø **Navigateur** : c'est un logiciel qui offre une interface d'accès aux différentes informations disponibles sur le Web et l'exploitation de ce dernier, ce dernier interprète le texte et les commandes de formatage qu'il contient et affiche à l'écran la page correctement formatée.
- Ø **Moteur de recherche** : un moteur de recherche est un programme qui indexe le contenu de différentes ressources Internet, et plus particulièrement de sites Web, il permet à l'internaute de rechercher de l'information à partir d'un navigateur Web, selon différents paramètres, en se servant de mots clés, et qui permet d'avoir accès à l'information ainsi trouvée.

VI. Site statique et site dynamique :

VI.1. Site statique :

VI.1.1. Définition :

Un site statique est composé uniquement de pages html prédéfinies qui contiennent à la fois les informations qui seront lues par ses visiteurs ainsi que les instructions qui définissent leur mise en page.

Bien que les sites statiques soient théoriquement plus simples à développer et qu'ils n'imposent pas de contraintes techniques particulières lors du choix de la solution d'hébergement, ils sont de plus en plus souvent remplacés par des solutions dites dynamiques.

VI.1.2. Fonctionnement d'un site statique :

L'administrateur du site compose avec un éditeur HTML des pages web stockées sur le serveur web. Celui-ci renvoie ces pages à la demande du visiteur. Par la suite ces pages ne pourront être modifiées que via un éditeur HTML, par l'administrateur. Le contenu de ces pages est fixe comme un fichier Word, et n'est pas modifié par le serveur. Le site est donc dit « statique » car son contenu ne change que par une intervention humaine et non pas des fonctions automatiques opérées par le serveur, comme l'illustre la figure suivante :



Figure I.18 : fonctionnement d'un site statique.

- **Avantage :**

- Ø Simple à créer (quoique, les menus d'accès et retour à un niveau supérieur d'arborescence ou à l'accueil deviennent vite des éléments durs à gérer).
- Ø Rapides à afficher (sous réserve de rester raisonnable quant au total d'une page donnée, à savoir environ 100 ko).
- Ø Simple à héberger.

- **Inconvénient :**

- Ø Lourdes à gérer dès qu'ils dépassent quelques pages.
- Ø Difficiles à gérer à plusieurs personnes.
- Ø Ces sites deviennent vite totalement statiques, sans mise à jour.

VI.2. Site dynamique :

VI.2.1. Définition :

La particularité des sites dynamiques est les pages lues par ses visiteurs n'existent pas physiquement. Elles sont fabriquées en temps réel par un ensemble de logiciels ou scripts résidant sur le serveur sur lequel il est hébergé.

VI.2.2. Fonctionnement d'un site dynamique :

L'administrateur du site et le visiteur utilisent le même outil : le navigateur web. Mais les pages avec lesquelles travaille l'administrateur ne sont pas les mêmes que celle que le visiteur voit : il encode dans des pages d'administration (aussi appelées : formulaire), qui nourrissent la base de données.

De son côté, le visiteur visualise des pages qui font appel au contenu de la base de données.

C'est le serveur web qui s'occupe de récupérer le contenu des formulaires que remplit l'administrateur, et de renvoyer cette information dans les vues par le visiteur, comme l'illustre la figure suivante :



Figure I.19 : fonctionnement d'un site dynamique.

- **Avantage :**

- Ø Moins de publication de pages web sur le site donc moins de risques d'incohérence.
- Ø Plus grande automatisation de la mise à jour.
- Ø Possibilité de séparer contenu et forme.
- Ø Plus d'interactivité et de personnalisation.

- **Inconvénient :**

- Ø Nécessite des connaissances préalables en programmation.
- Ø Nécessite de connaissances préalables en bases de données.
- Ø Traitement des requêtes plus long.

VII. Les langages de développement des sites web : [AKCH/08]

Les langages utilisés pour le web peuvent être classés en deux grandes catégories selon qu'ils soient exécutés sur la machine cliente ou sur le serveur.

VII.1. Côté client :

- **HTML :**

HTML est un langage dit «marquage» chargé de formaliser un document avec des balises de formatage indiquant la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents.

Il permet notamment la lecture de document sur Internet à partir de machines différentes grâce au protocole http, permettant d'accéder via le réseau à des documents repérés par une adresse unique, appelée URL.

- **CSS :**

Les feuilles de style en cascade, également appelées CSS sont apparues avec le HTML4 et DHTML. Ce sont des technologies récentes fonctionnant sur les derniers navigateurs avec des machines plus puissantes.

Aujourd'hui, les feuilles de style sont très répandues, elles permettent de simplifier le formatage d'un texte et d'attribuer rapidement des propriétés Java Script à des objets.

- **XML** : (eXtensible Markup Language) : est en quelque sorte un langage HTML amélioré permettant de définir de nouvelles balises. Il s'agit effectivement d'un langage permettant de mettre en forme des documents grâce à des balises.
Contrairement à HTML, qui est à considérer comme un langage défini et figé (avec un nombre de balises limité), XML est considéré comme un métalangage permettant de définir d'autres langages, c'est-à-dire définir de nouvelles balises permettant de décrire la présentation d'un texte.
- **Applets** : se sont de petits programmes exécutables, intégrée aux documents HTML, de sorte qu'ils peuvent être directement exécutés par la machine virtuelle Java intégrée du navigateur, ceci après avoir été transféré du serveur web.
- **JavaScript** : aujourd'hui c'est le moyen le plus courant qu'utilisent les créateurs de pages HTML pour animer celles-ci. Ils introduisent dans ces pages des lignes de codes qui sont repérées par des balises spécifiques. Ces lignes de code font parties de la page HTML, et sont donc envoyées au navigateur. Ce dernier, interprète le langage HTML pour afficher la page et interprète le code JavaScript pour les animer.

VII.2. Côté serveur :

- **CGI** (Common Gateway Interface): un script CGI est un programme exécuté par le serveur web, permettant d'envoyer au navigateur de l'internaute un code HTML crée automatiquement par le serveur (basé par exemple sur une autre application, telle qu'un système de gestion de base de données, d'où le nom passerelle). Un des principaux intérêts de l'utilisation de CGI est la possibilité de fournir des pages dynamique, c'est-à-dire des pages personnalisées selon un choix ou une saisie de l'utilisateur.
- **Servlet** : est constitué par une classe Java compilée. Techniquement cette solution ressemble à celle des API sauf que le développement est beaucoup plus facile.
- **ASP (Active Server Pages)** : est un standard mis par Microsoft en 1996 permettant de développer des applications web interactives, c'est-à-dire dont le contenu est dynamique. Ainsi une page web ASP (fichier repérable par l'extension .ASP) aura un contenu pouvant être différent selon certains paramètres tandis qu'une page web « classique » (dont l'extension est .html) affichera continuellement la même information. La programmation en langage de script est aisée et réalisée.

- **PHP (HyperPreProcessor)** : est un langage de script orienté objet. Principalement utilisé pour produire des pages web dynamiques via un serveur http, il est intégré le code HTML et qui interprété côté serveur. La syntaxe du PHP est un mélange de langage C et Perl.
- **ASP.NET** : Technologie développée par Microsoft sur base du framework.NET.l'évolution est importante par rapport à ASP et les possibilités deviennent quasi infinies. ASPI.NET a le plus d'avantages, par rapport à d'autres technologies, de bien séparer le contenu de la page web, sa mise en forme et la programmation des aspects dynamiques (travail en Code Behind).

Conclusion :

Dans cette partie, nous avons vu quelques généralités sur le plus populaire de l'Internet qui est le **WEB** en introduisant les deux types de sites existants à savoir, les sites statiques et les sites dynamiques, par la suite nous avons donné une vue globale sur ces différentes technologies du coté Client et u coté Serveur.

Les progrès réalisés dans les technologies de l'information nous ont fait entrer dans une nouvelle ère, celle de l'informatique ouverte au grand public.

Au cours de ce chapitre, nous avons présenté certains concepts des nouvelles technologies de l'information et de la communication, à citer : les réseaux informatiques, l'Internet et le web, l'architecture client/serveur... Qui nous aiderons à mieux comprendre notre mode d'opération et les notions fondamentales pour mener à bien notre application.

I. Introduction :

Avant toute réalisation d'une application informatique, il convient de suivre une démarche méthodologique et rigoureuse pour planifier et concevoir l'application, en mettant en évidence tous les objectifs tracés pour la bonne élaboration du projet souhaité.

Dans notre cas nous avons opté pour la modélisation orientée objet avec le langage UML qu'avec son extension pour le Web fournit une notation pour exprimer les composants de technologie web du système dans le modèle.

Comme UML n'impose pas de méthode de travail particulière et peut être intégré à n'importe quel processus de développement logiciel, la démarche de modélisation choisie pour la conception de notre application est la suivante :

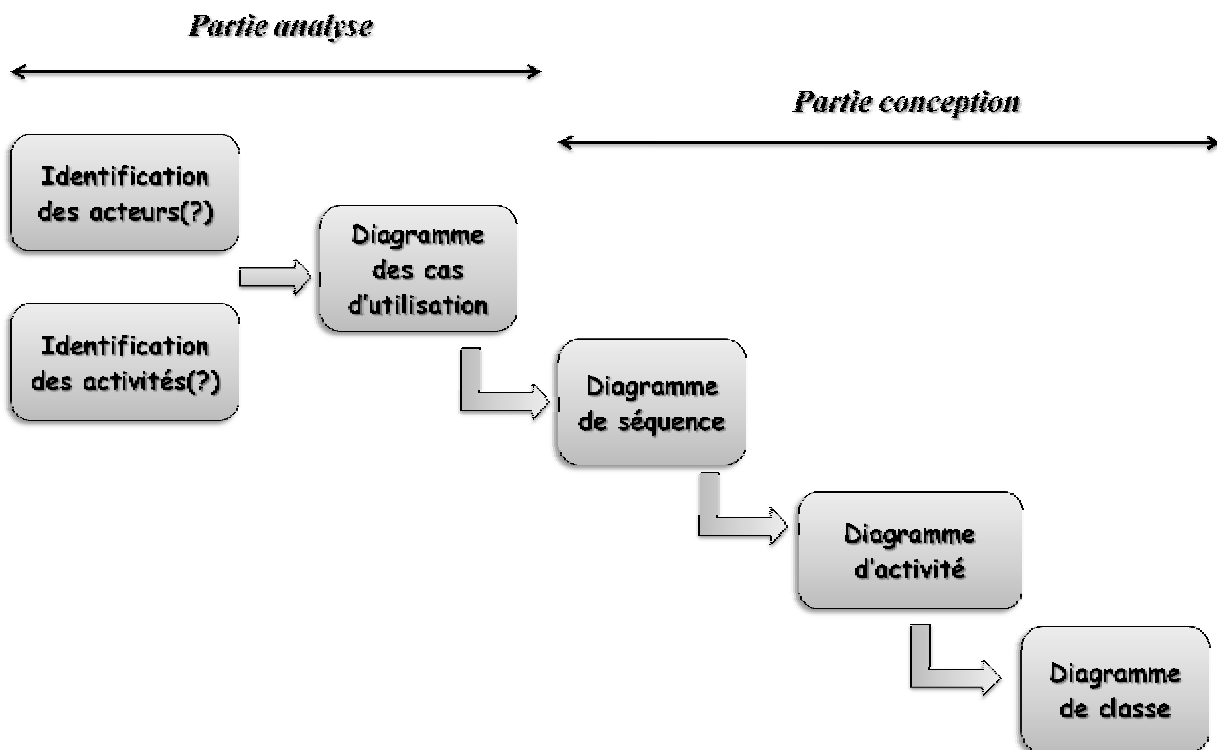


Figure II.1 Démarche de modélisation de l'application.

II. Objectif du projet :

En utilisant un serveur web nous mettrons en place une application qui permettra de gérer un ensemble de parkings automobiles, n'importe quel individu peut naviguer dans l'application mais réserver une place que s'il est client ou abonné au site.

III. Analyse :

Cette activité commence par la mise en évidence des différents acteurs intervenants dans le système cible ainsi que leurs besoins. Ensuite, la phase conception donnera la modélisation des objectifs à atteindre en s'appuyant sur les résultats de la phase analyse.

III.1 Définition des besoins :

III.1.1 Les besoins non fonctionnels :

a) Les besoins de performance :

Décrivent les performances d'exécution du système, généralement en termes de temps de réponse. Le temps de chargement des pages web ne devra pas dépasser quelques secondes.

b) Les besoins de disponibilité/fiabilité :

Concernent le niveau de disponibilité qui doit être explicitement défini pour toute application critique.

Notre système doit être opérationnel 24h/24h et 7j/7j.

c) Les besoins de sécurité :

Notre système garantira que l'accès aux espaces appropriés pour chaque acteur sera sécurisé par l'authentification par un login et un mot de passe.

d) Les besoins matériels :

Définissent souvent les configurations matérielles minimales nécessaires au fonctionnement du système.

Notre application utilisera des technologies un peu avancées utilisant du multimédia. Donc une configuration standard du côté client comme : un processeur Pentium 4 (2.0 GHz) avec 160 MB de mémoire, une carte graphique SVGA, une carte son, un écran avec une résolution de 800x600 pixels, une palette de couleurs 16 à 32 bits,...etc., sera acceptable, et une configuration plus performante côté serveur.

III.1.2 Les acteurs et les besoins fonctionnels :

On emploie le terme d'acteur pour décrire un rôle générique d'utilisateur et réserve celui d'utilisateur pour des instances des personnes qui utilisent le système. Un utilisateur peut très bien jouer le rôle de plusieurs acteurs du système. Le nom qui identifie un acteur doit représenter le type de rôle qu'il joue dans le système.

Notre application regroupe l'ensemble des acteurs suivants :

Ø Visiteur :

N'importe quel individu peut naviguer dans l'application, devenir client, s'abonner au site et contacter l'administrateur.

Ø Client :

Est toute personne voulons réserver une place dans un parking, après avoir confirmé son inscription on lui donne les droits d'accès (login, mot de passe),il devient membre.

Ø Abonné :

Est toute personne voulons réserver une place dans un parking, après avoir confirmé son inscription on lui donne les droits d'accès (login, mot de passe), il devient membre.

Remarque : A la différence du client, une place est toujours disponible pour l'abonné.

Ø Employé :

C'est la personne chargé de gérer, autrement dit mettre à jour les entrées sorties dans un parking.

Ø Administrateur :

L'administrateur est la personne chargée d'administrer le site donc il a pour fonction après l'identification:

- Gestion des utilisateurs (clients, abonnés).
- Gérer la base de données.
- Gestion des réservations.
- Gérer la messagerie.

III.2 Identification et structuration des cas d'utilisation :**III.2.1 Définition :**

Un cas d'utilisation est la description d'une séquence d'actions exécutées par un système pour conduire un résultat qui peut être constaté par un acteur particulier .Il sert à structurer les éléments comportementaux d'un modèle et est réalisé par une collaboration.

III.2.2 Structuration des cas d'utilisation :

Le tableau suivant résume les différents cas d'utilisation assurée par chaque acteur du système.

Acteur	Tâches
Visiteur	T 01- Se connecter au site T 02- Naviguer dans le site T 03- Devenir client T 04-S'abonner T 05- Contacter l'administrateur
Client	T 01 - T 02 - T 06 -S'authentifier T 07 -Accéder à l'espace Client T 08 -Réserver une place T 09 -Modifier une réservation T 10 -Annuler une réservation T 04 - T 05 - T 11-Se déconnecter
Abonné	T 01 - T 02 - T 06 - T 12 -Accéder à l'espace Abonné T 08 - T 09 - T 10 - T 05 - T 11 -
Administrateur	T 01 - T 02 - T 06 - T 13 -Accéder à l'espace Administrateur T 14 -Publier des tarifs T 11 - T 15 -Gérer les membres T 16 -Gérer les parkings T 17 -Consulté les réservations T 18 -Gérer la messagerie

Employé	T 01 - T 02 - T 06 - T 19 - Accéder à l'espace Employé T 20 – Vérifier les réservations T 21 - Gérer les places T 11 -
----------------	--

Figure II.2 : Structuration des cas d'utilisation.

III.2.3 Description des scénarios :

• **Un scénario :**

Représente une succession particulière d'enchaînement s'exécutant du début à la fin du cas d'utilisation, un enchaînement étant l'unité de description de séquences d'actions, un cas d'utilisation contient en général un scénario nominal et plusieurs scénarios alternatifs (qui se terminent d'une façon normale) ou d'erreurs (qui se terminent en échec).

Après avoir identifié les cas d'utilisation de l'application, le tableau suivant mettra en évidence les scénarios appropriés :

Acteurs	Tâches	Scénarios
Visiteur	T 01- Se connecter au site	S01-Taper l'URL.
	T 02- Naviguer dans le site	S02-Consulter les différents liens du site.
	T 03- Devenir client	S03-Sélectionner le lien devenir client. S04-Remplir le formulaire puis valider.
	T 04-S'abonner	S05-Sélectionner le lien s'abonner. S06-Remplir le formulaire puis valider.

	T 05- Contacter l'administrateur	S07-Cliquer sur le lien «contacter l'administrateur». S08-Saisir le message dans l'espace afficher et envoyer.
Client	T 01-	S01-
	T02-	S02-
	T06- S'authentifier	S09-Accéder au formulaire d'authentification S10-Saisir les coordonnées (login mot de passe)
	T07- Accéder a l'espace client	S11-Valider le formulaire d'authentification
	T08- Réserve une place	S12-Sélectionner le lien « réservation » S13-remplir le formulaire S14-valider la réservation
	T09- Modifié une réservation	S12- S15-Accéder à la réservation qu'on veut modifier S16-faire la modification S14-
	T10- Annulé une réservation	S12- S17-Sélectionner la réservation à supprimer S18-cliquer sur le lien « annuler réservation »
	T04-	S05- S06-
	T05-	S07- S08-
	T11- Se déconnecter	S12-Sélectionner le lien « Déconnecter »
Abonné	T01-	S01-
	T02-	S02-

	T06-	S09- S10-
	T12- Accéder à l'espace abonné	S11-
	T08-	S12- S13- S14-
	T09-	S12- S15- S16- S14-
	T10-	S12- S17- S18-
	T05-	S07- S08-
	T11-	S12-
Administrateur	T 01-	S01-
	T 02-	S02-
	T 06 -	S09- S10-
	T 13 -Accéder à l'espace administrateur	S11-
	T 14 -Publier des tarifs	S19-Sélectionner le lien « publier des tarifs» S20-Accéder à la page publier des tarifs S21-Ajouter le tarif S22-Enregistrer le tarif
	T 11 -	S12-
	T 15 - Gérer les membres	S23-Visualiser la liste des membres S24-Modifier les informations sur un membre S25-Ajouter un membre S26-Supprimer un membre

	T 16 - Gérer les parkings	S27-Visualiser la liste des parkings S28-Modifier les informations d'un parking S29-Ajouter un parking S30-Supprimer un parking
	T 17 - Consulter les réservations	S31-Sélectionner le lien réservation S32-visiter les consultations
	T 19 - Gérer la messagerie	S33-Sélectionner le lien « message » S34-Consulter les messages en cliquant sur le lien « lire » S35-Répondre aux messages en cliquant sur le lien « Répondre » S36-Supprimer le message en cliquant sur le lien « Supprimer »
Employé	T 01 -	S01-
	T 06 -	S09- S10-
	T 20 - Accéder à l'espace employé	S11-
	T 21 - Vérifier les réservations	S37-Sélectionner le lien « réservation » S38-Remplir le formulaire et valider.
	T 22 - Gérer les places	S39-Visualiser la liste des places S40-Occuper des places S41-Libérer des places
	T 11 -	S12-

Figure II.3 : Description des scénarios.

III.2.4 Diagramme de cas d'utilisation général :

Le diagramme de cas d'utilisation permet de présenter les acteurs du système, ses grandes fonctions ainsi que les liens existant entre ces fonctions.

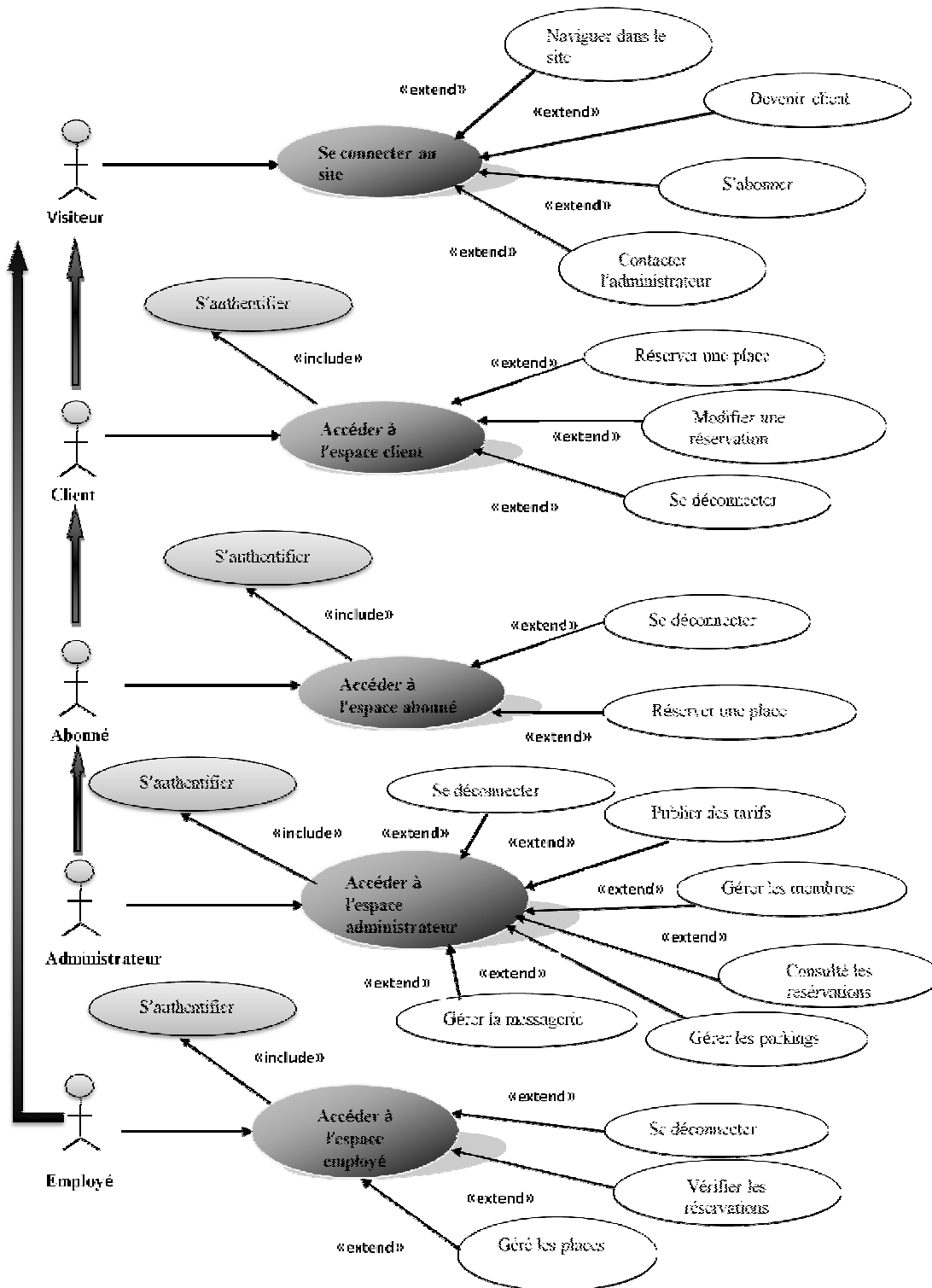


Figure II.4 : Diagramme de cas d'utilisation général.

IV. Conception :

Après avoir déterminé les cas d'utilisations et spécifier les besoins de l'application, nous allons passer à la phase de conception du projet, qui consiste à affiner les spécifications d'après leurs contenus.

Notre étude se base toujours sur le formalisme UML qui fournit un ensemble de diagrammes permettant d'affiner le modèle d'analyse jusqu'à l'obtention d'une nouvelle vue du modèle qui sera réalisable sous forme logicielle.

IV.1 Elaboration des diagrammes de séquence :

Les schémas des diagrammes de séquence permettent de détailler tous les scénarios possibles d'un cas d'utilisation en précisant les objets du système et les acteurs impliqués dans le scénario. Nous allons présenter les diagrammes de séquence pour les cas d'utilisation suivants :

- Ü Identification.
- Ü Réserver une place.
- Ü Publier des réservations.

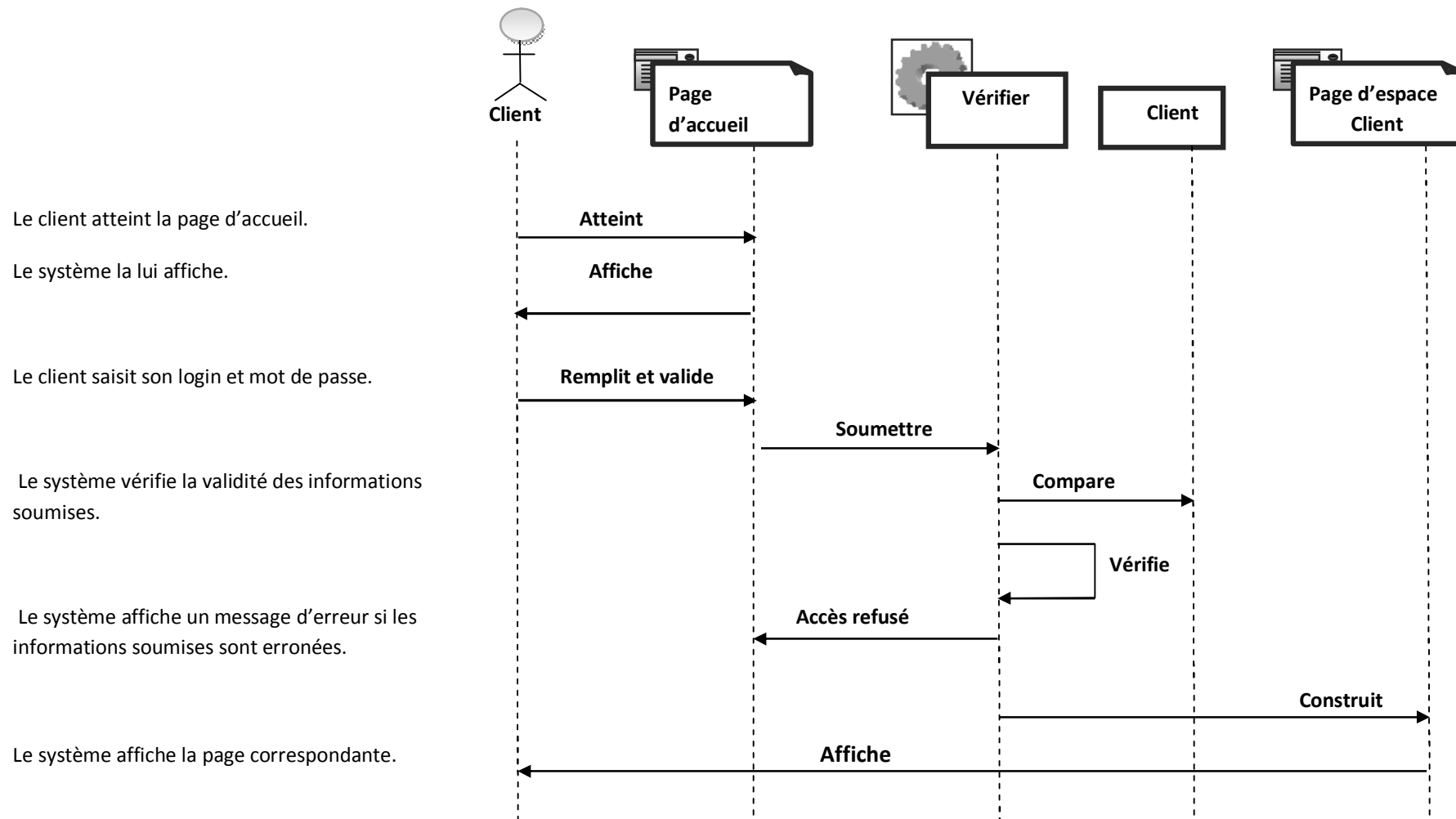


Figure II.5 : Diagramme de séquence du cas d'utilisation « Identification ».

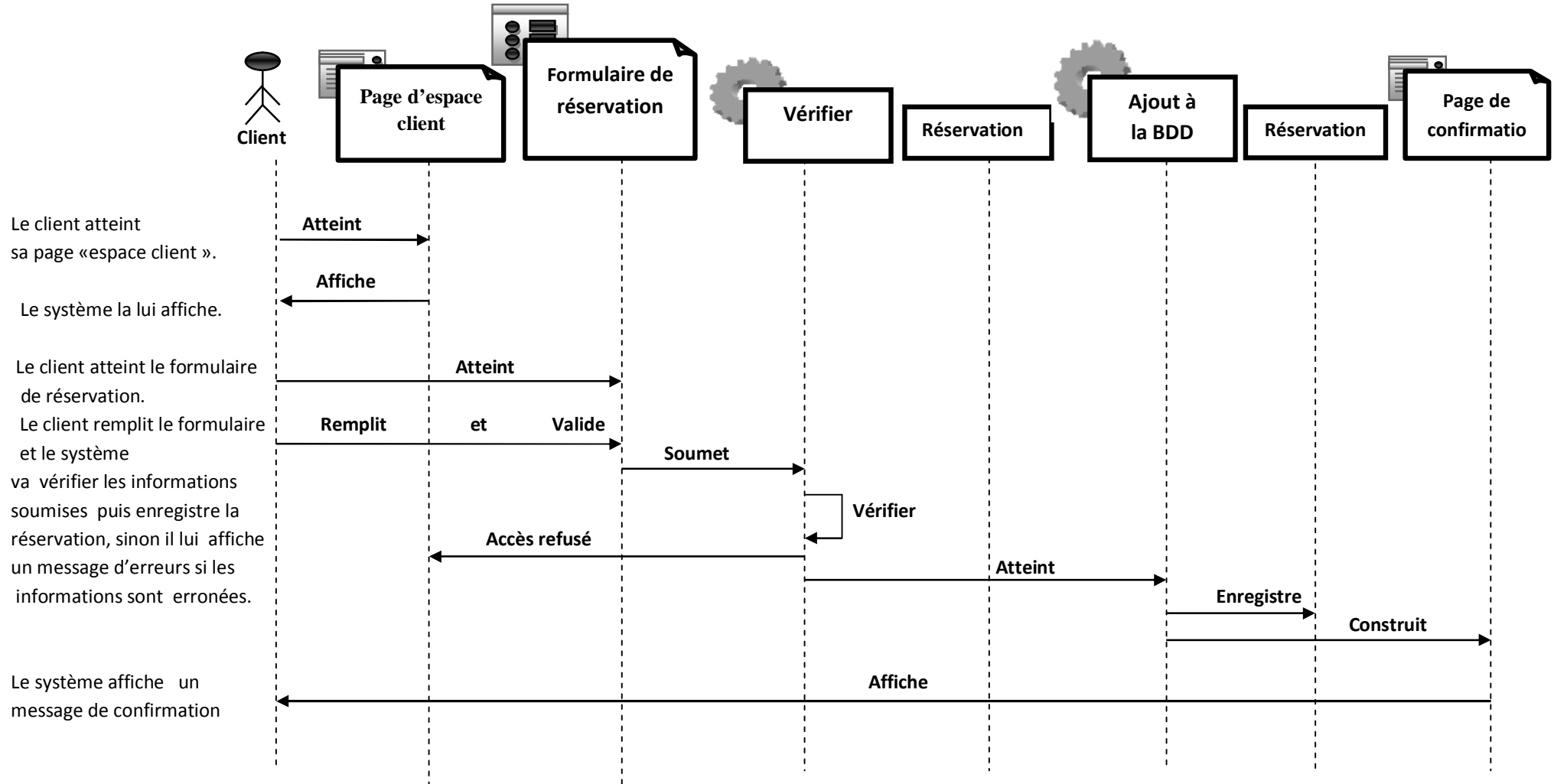


Figure II.6 : Diagramme de séquence du cas d'utilisation « Réserver une place ».

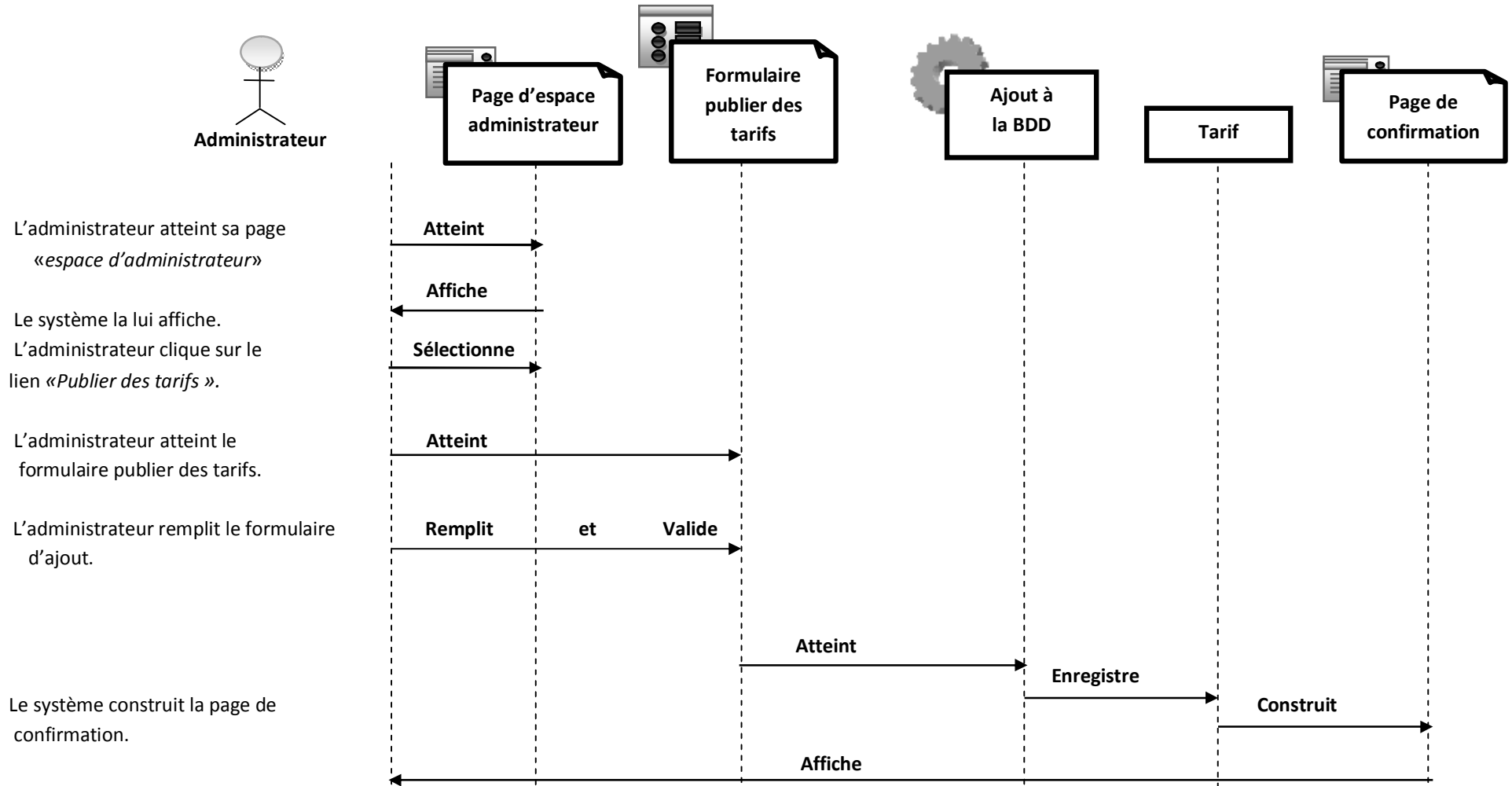


Figure II.7 : Diagramme de séquence du cas d'utilisation « Publier des tarifs ».

IV.2 Diagrammes d'activité pour quelques cas d'utilisation:

Le diagramme d'activité fait partie des diagrammes d'UML utilisé pour la modélisation des aspects dynamique des systèmes .C'est une variante des diagrammes d'états transition organisé par rapport aux actions et principalement destiné à représenter le comportement interne d'une méthode ou d'un cas d'utilisation.

IV.2.1 Diagramme d'activité pour le cas d'utilisation « Identification » :

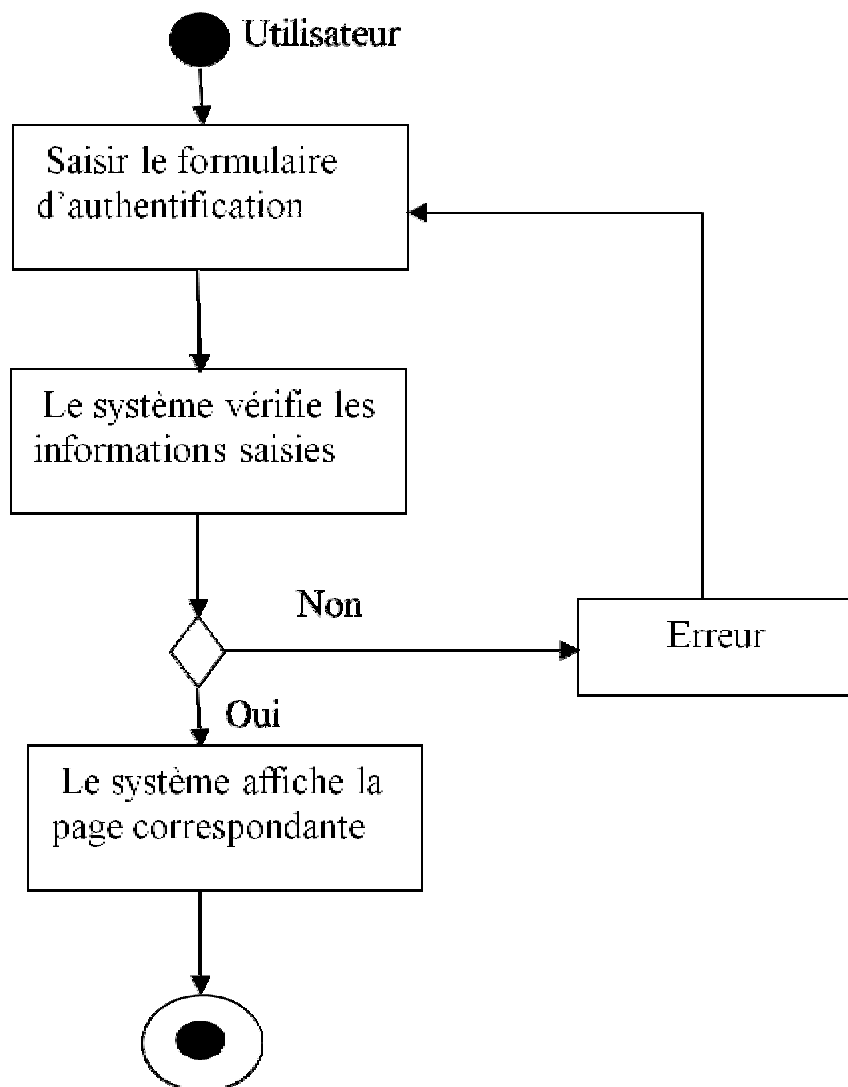


Figure II.8 : Diagramme d'activité pour le cas d'utilisation « Identification ».

IV.2.2 Diagramme d'activité pour le cas d'utilisation « Réserver une place » :

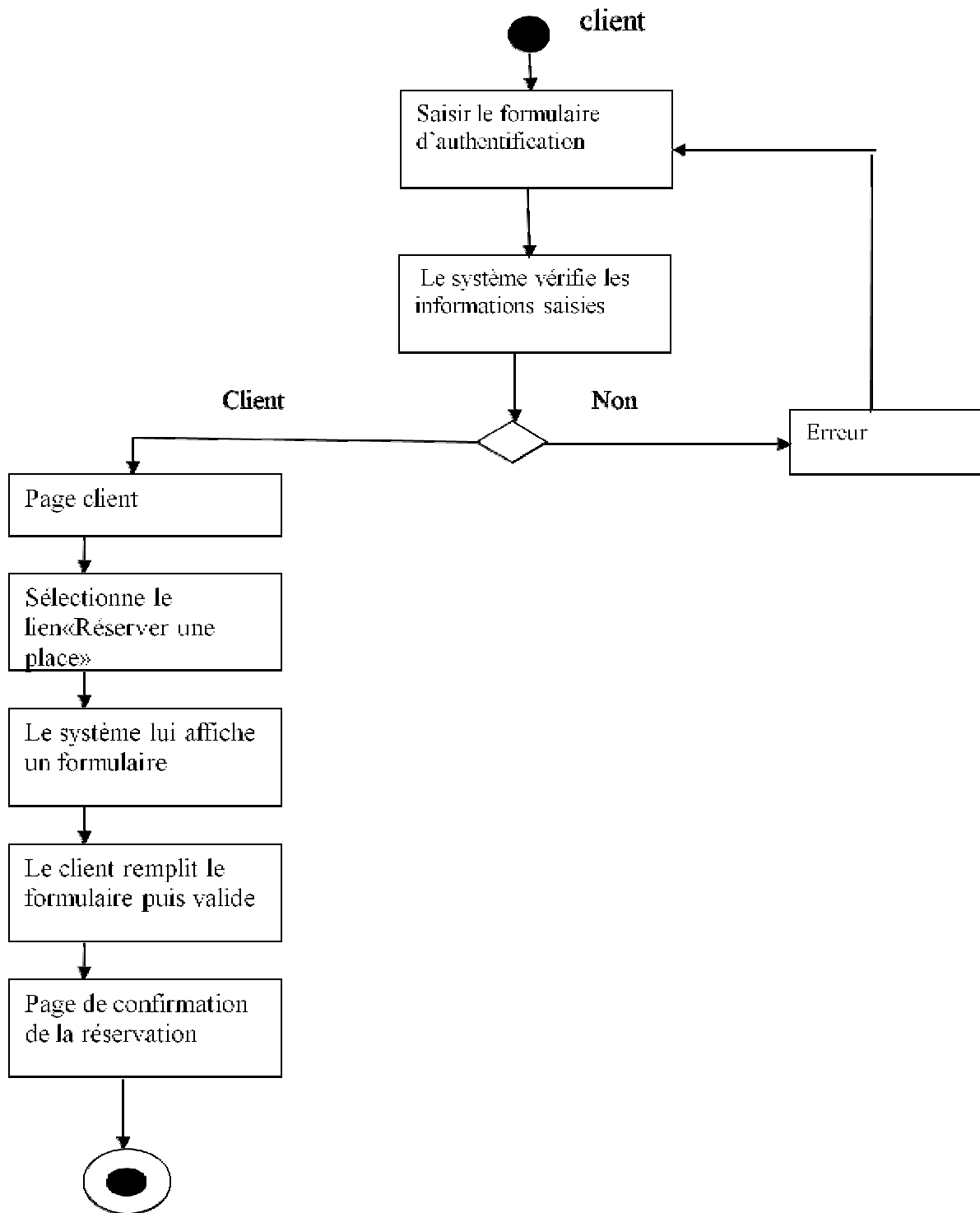


Figure II.9 :Diagramme d'activité pour le cas d'utilisation « Réserver une place».

IV.2.3 Diagramme d'activité pour le cas d'utilisation « Publier des tarifs » :

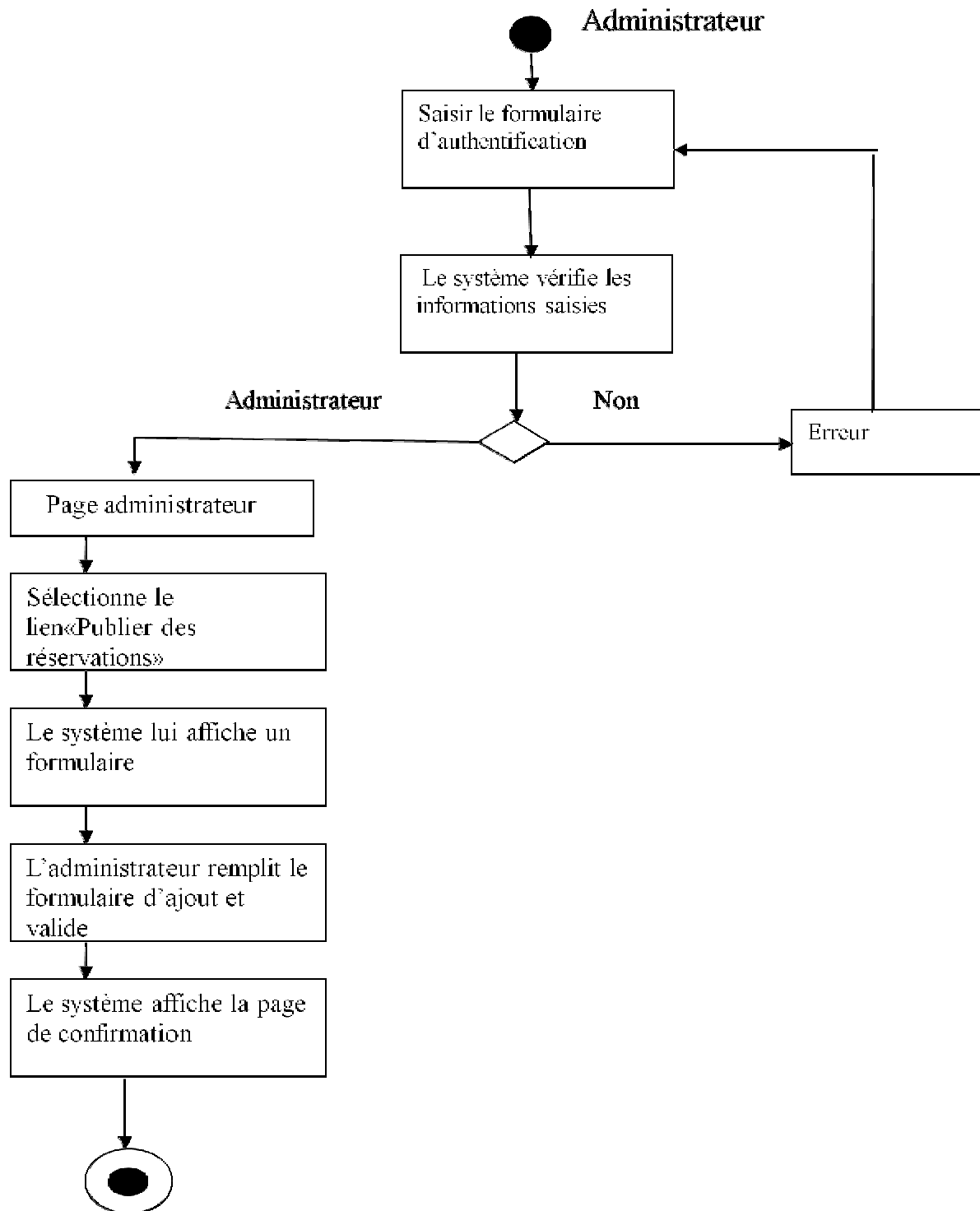


Figure II.10 :Diagramme d'activité pour le cas d'utilisation « Publier des tarifs».

IV.3 Diagrammes de classe généraux :

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet. Il est utilisé pour représenter une vue purement statique du système.

En voici les diagrammes de classe pour quelques cas d'utilisations :

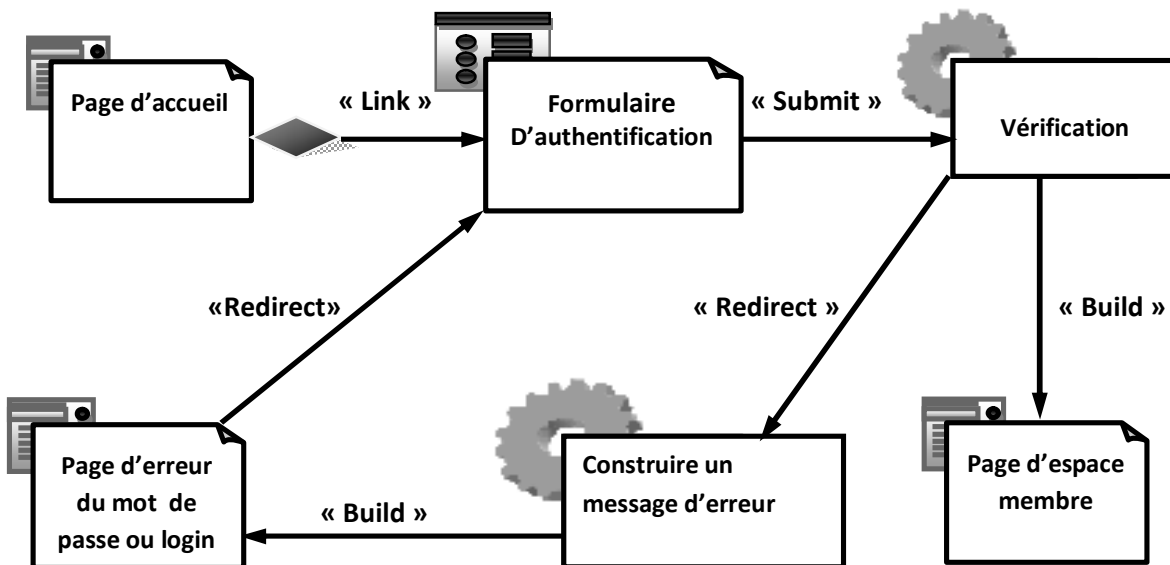


Figure II.11 : Diagramme de classe du cas d'utilisation « Identification ».

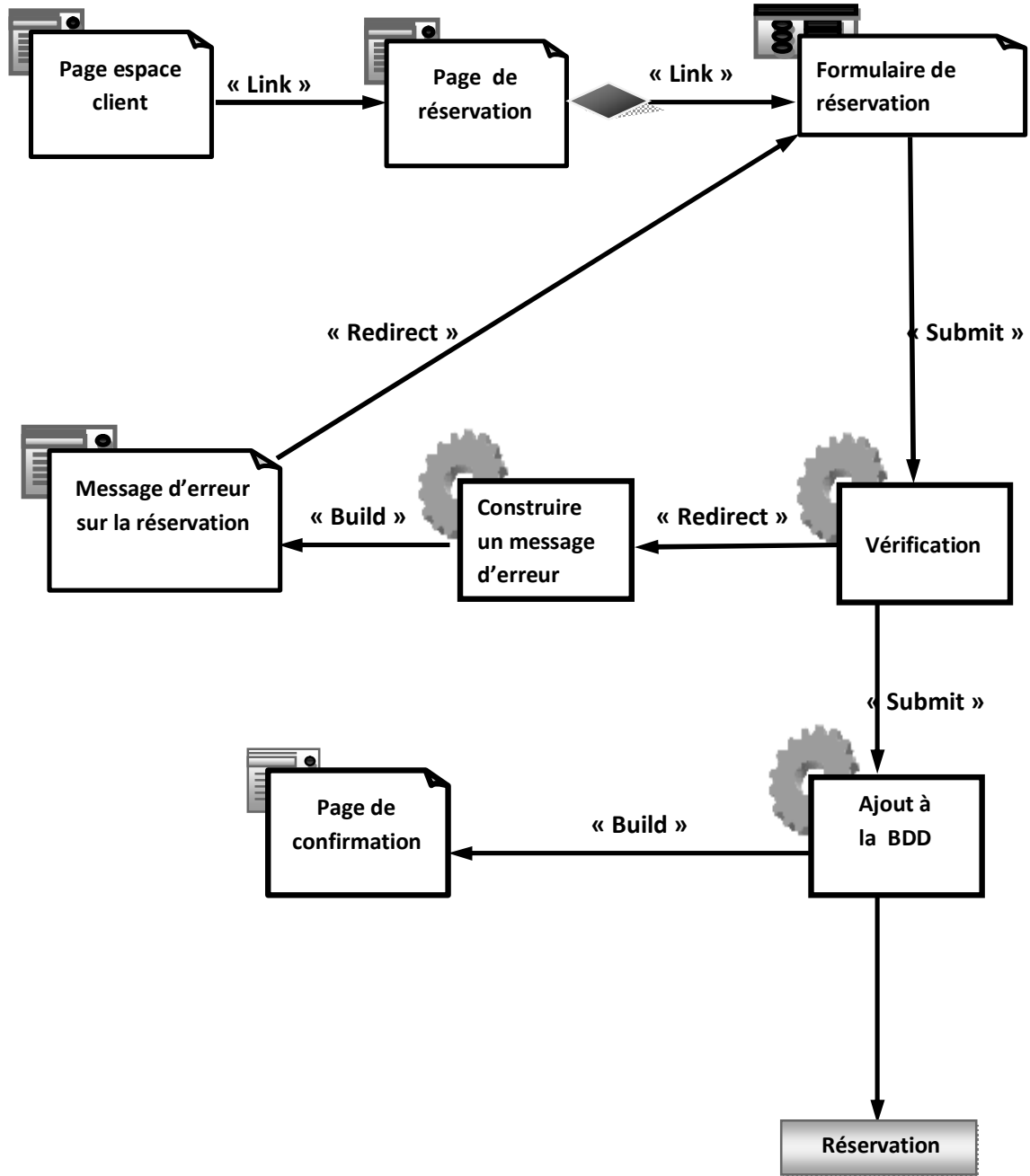


Figure II.12 : Diagramme de classe du cas d'utilisation « Réserver une place».

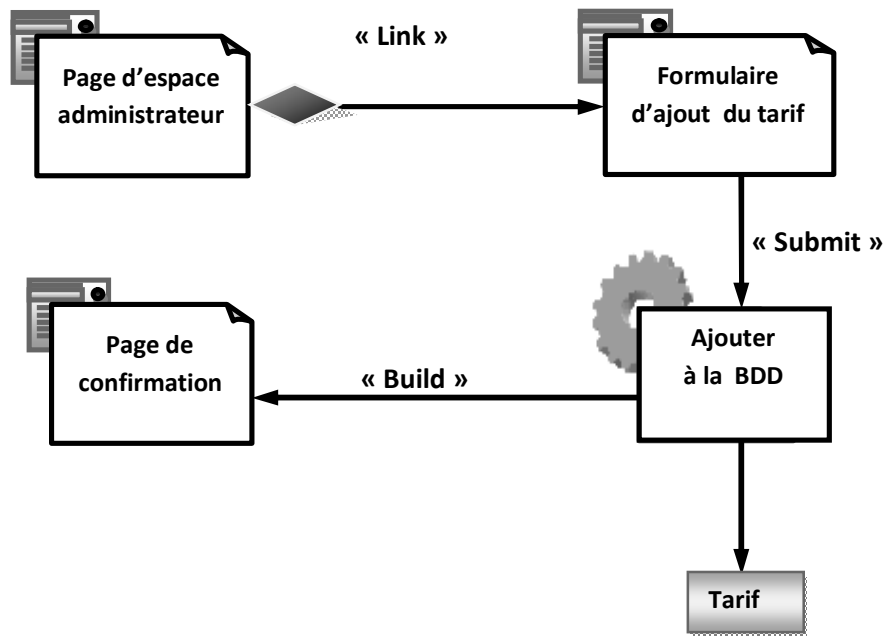


Figure II.13 : Diagramme de classe du cas d'utilisation « Publier des tarifs ».

V. Mapping Objet-Relationnel :

Les premières questions que tous les programmeurs et chefs de projets se posent avant de commencer à développer une application sont celles concernant le choix de la plateforme de développement, et celui du serveur de bases de données. Pour nous il s'agit du couple PHP/MySQL.

La tendance actuelle veut que le développement soit fait dans un langage objet et utilise une base de données relationnelle. Mais il existe des difficultés de cohabitation entre les mondes objets et relationnels et les développeurs élaborent ainsi les « *modèles relationnels* » pratiquement indépendamment aux « *modèles objets* » en l'occurrence les diagrammes de classes, ses difficultés et problèmes sont résolus grâce au concept de « *Mapping objet-relationnel* » (O/R Mapping), qui est le nom donné aux techniques d'optimisation des modèles objets (le diagramme de classes global pour notre cas) et de transformation de ces modèles en modèles relationnels.

V.1 Principes d'optimisation et de transformation (O/R) : [D1]

L'entité principale du monde objet est la « *classe* ». Celle-ci sera transformée en une « *table* », à laquelle on pourra donner le même nom que la classe. De la même manière qu'une classe est composée de plusieurs « *attributs* », une table est composée de « *champs* », tous typés.

Réservation
« date »date_res
« heure »heure_debut
« heure »heure_fin

Classe réservation

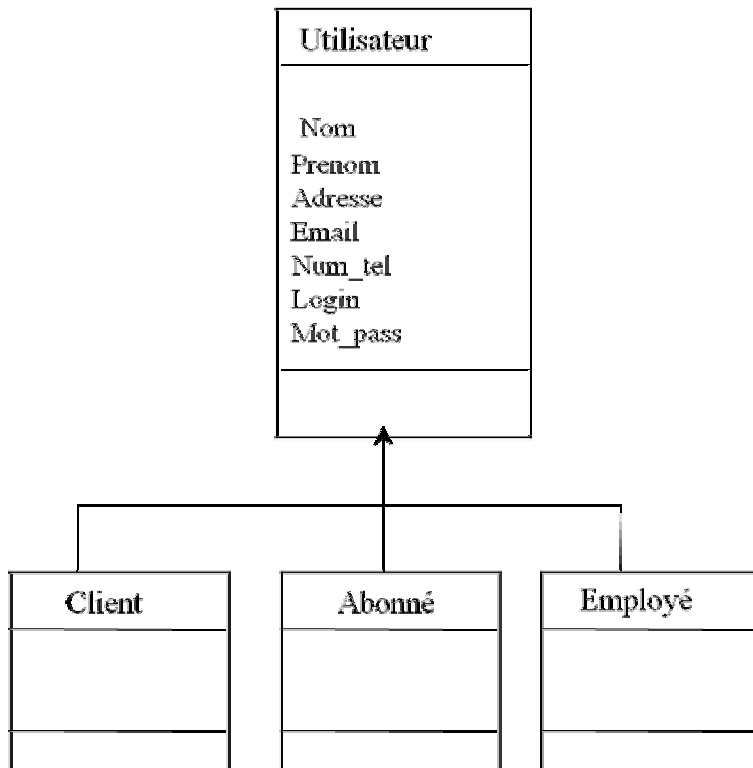
Réservation	
CP	Id_Res
	date_res
	heure_res
	heure_fin

Table correspondante

Ainsi, une « *instance* » d'une classe particulière, correspondra à un « *enregistrement* » de la table correspondante.

On peut remarquer que la table « *Réservation* » de la figure précédente contient un champ qui n'a aucun équivalent dans le modèle objet. Il s'agit de la clé **primaire** : Id_Res. Ce champ est de type GUID (General Unique Identifier), et représente de manière unique un enregistrement dans une table.

Intéressons nous maintenant à l'héritage. La méthode utilisée permet d'obtenir de très bonnes performances, Il s'agit de modéliser toute une hiérarchie de classes dans une même table, chaque classe ajoutant ses attributs propres comme de nouveaux champs. Il nous suffit alors d'ajouter un champ contenant le type de l'instance pour pouvoir charger les champs correspondants.



Exemple de diagramme de classe

Utilisateur	
CP	Id_Util
	Nom
	Prenom
	Adresse
	Email
	Num_tel
	Login
	Mot_pass
	Statut

Table correspondante

▼ **Statut** :c'est un champ pour distinguer entre « *client* » « *abonné* » et « *employé* ».

Etudions maintenant comment modéliser les relations entre les classes. En UML, il est possible de spécifier la cardinalité des relations. Voici des exemples de relations (de notre diagramme de classes) avec des cardinalités différentes :

- La relation la plus simple est la relation 1-1, cela signifie qu'un objet de type *classe-1* ne peut posséder qu'une seule instance de type*classe-2* et inversement.

Pour effectuer la correspondance en base de données, il suffit d'ajouter à la table*classe-1* un champ de type GUID, qui contiendra la valeur du champ ID d'un enregistrement de la*tableclasse-2*. Il s'agit donc d'une clé étrangère.

- La relation ***-***, cette relation représente le fait qu'un objet de *classe-1* peut posséder des références vers plusieurs objets *classe-2*, et que chaque objet *classe-1* peut être référencé par plusieurs objet *classe-2*.

Remarque : ces deux premiers cas ne figurent pas dans notre diagramme.

La relation entre *réservation* et *parking* est une relation **1-***, c'est-à-dire qu'une instance de la classe *parking* pourra posséder plusieurs *réservations*, et que chaque *réservation* peut être effectué que dans un seul *parking*.



Toute comme pour la relation 1-1, il nous faut ajouter une clé étrangère. Mais cette fois ce n'est plus la table *parking* qui va référencer les enregistrements de la table *réservation*, mais l'inverse.

La table *réservation* possède donc autant de champs que la classe *réservation* possède d'attributs plus un champ *ID* et la clé étrangère pour référencer l'objet de type *parking* qui le possède.

Voici le résultat sur la table

Réservation	
CP	Id_Res
CE	Id_Park

Conclusion :

Dans ce chapitre nous avons présenté une brève définition de l'UML puis on est passé à la modélisation du système cela en s'aidant des différents outils de modélisation u langage UML.

Après avoir analysé la problématique en profondeur et effectuer la conception de notre application on passe au chapitre suivant qui est la réalisation.

I. Introduction :

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, nous allons présenter dans ce dernier chapitre l'environnement de développement, les outils qui ont servi à la réalisation de notre application, et nous terminerons par la présentation de ses fonctionnalités à travers ses différentes interfaces.

Plusieurs solutions existent pour le développement d'une application web dynamique, nous avons opté pour la solution PHP/MySQL pour réaliser notre application.

II. Environnement de développement :

II.1. WAMP :

- Ø **Windows** : Assure l'attribution de ressources aux composants suivant.
- Ø **Apache** : Apache est le plus populaire des serveurs Internet. Il a été créé dans la volonté de développer et de maintenir un serveur http sécurisé, efficace et évolutif pour les systèmes d'exploitation modernes.

Apache est multi plate forme et gratuit, son installation est facile, rapide et son utilisation pas très compliqué. Grace à une association avec PHP apache devient un serveur web dynamique. [WB.RATT]

- Ø **MySQL** : Est édité par la société suédoise MySQL AB. De part la possibilité de l'utiliser gratuitement et aussi de part ses performances, il est très populaire pour la réalisation des sites Internet. Il est disponible pour plusieurs plates-formes dont Linux et Windows. [WB.ALD]

MySQL est un serveur de base de données relationnelle, basé sur le langage de requête SQL (Structured Query Language) qui est un langage standard pour le traitement des bases de données. Le serveur MySQL peut fonctionner en mode client/serveur. Il contrôle l'accès aux données pour assurer que plusieurs utilisateurs peuvent se servir simultanément d'une même base de données.

Le serveur MySQL offre des fonctions nombreuses et puissantes. Ses possibilités de connexions, sa rapidité et sa sécurité font de lui un serveur hautement adapté à l'internet. [ING info 2008-09]

- Ø **PHP** : c'est un langage généralisé pour la production de contenu Web dynamique. Son association avec un serveur Apache et une base de donnée MySQL forme une solution gratuite, fiable et puissante pouvant être déployé sur de nombreuses plate forme telles que Windows.

III. Les outils utilisés :

III.1 Les langages utilisés :

Ø **HTML** : (Hyper Text Markup Language) est un langage hypertexte à balises (marqueurs). Ces balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents. Cela veut dire que l'on va gérer la façon dont un texte va s'afficher au sein du navigateur.

L'html n'est pas un langage de programmation au sens classique du terme, mais est essentiellement un ensemble de règles qui indiquent à un navigateur comment afficher une page du web. Il est souvent utilisé conjointement avec des langages de programmation telle que JavaScript et des formats de présentation (feuilles de style en cascade). [WB.HRHTML] [WB.OBGM]

Ø **CSS** : Abréviations de *Cascading Style Sheets*, « feuilles de style en cascade » en français, est un langage qui permet de gérer la présentation d'une page Web. Le principe des feuilles de style consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments. Il suffit de définir par un nom un ensemble de définitions et de caractéristiques de mise en forme, et de l'appeler pour l'appliquer à un texte.

Les feuilles de style ont été mises au point afin de compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation. [WB.CCM] [WB.FSC] [WB.CCM]

Ø **PHP** : Abréviations de *PreHypertextProcessor*, préprocesseur hypertexte en français, PHP est un langage qui a été conçu en 1994 par Rasmus Lerdorf. Les premières versions (qui restèrent privées) étaient utilisées afin de savoir qui venait consulter son CV en ligne. La première version publique fut disponible au début de l'année 1995. Elle fut connue sous le nom de "*Personal Sommaire Page Tools*". Elle était composée d'un analyseur extrêmement simple qui ne reconnaissait que quelques macros spéciales et d'un petit nombre d'utilitaires couramment utilisés dans les pages web.

PHP est un langage utilisé dans des applications web pour écrire des scripts HTML. Le but premier de ce langage est de permettre aux développeurs web de concevoir rapidement des sites aux pages dynamiques. C'est un langage exécuté côté serveur, à la différence de JavaScript et html, donc le code de l'utilisateur n'apparaît pas dans la source de sa page web, seule les informations qu'il souhaite afficher seront visible par les visiteurs. [WB.DFT]

Ø **SQL** : C'est un langage de manipulation de base de données mis au point dans les années 70 par IBM. Il permet notamment :

- La manipulation des tables : création, suppression, modification de la structure des tables.

- La manipulation des bases de données : sélection, suppression et modification d'enregistrements.
 - La gestion des droits d'accès aux tables : contrôles de données et validation des modifications.
- Ø **JavaScript** : JavaScript est un langage de scripts incorporé aux balises html, permettant d'améliorer la présentation et l'interactivité des pages Web. Il est plus simple à mettre en œuvre car c'est du code que vous ajouterez à votre page écrite en Html, il est donc une extension du code Html des pages Web en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web. [WB.CCM]

Les balises annonçant un code JavaScript sont les suivantes :

```
<SCRIPT language= JavaScript >
    Mettez ici votre code
</SCRIPT>
```

III.2 Les logiciels :

- § **WampServer** : Est une plate forme de développement sous Windows. Il permet de développer des applications web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin et SQLite Manager pour gérer plus facilement les bases de données. [WB.RATT]

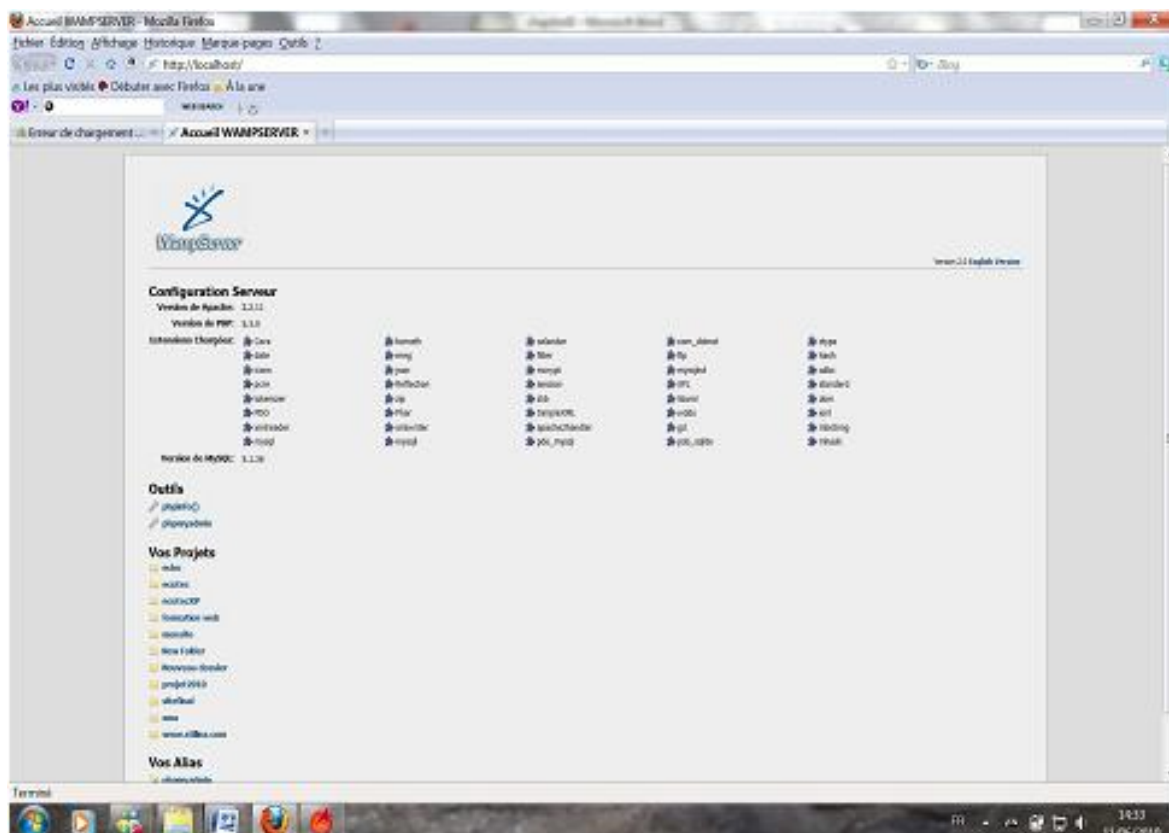


Figure III.1 : Interface de WampServer.

§ Macromedia Dreamweaver 8:

C'est un éditeur de pages web c'est-à-dire en quelques sortes un traitement de texte spécialisé dans la création de pages web. Lorsque vous utilisez ce type de logiciel, le HTML nécessaire à la création de vos pages est automatiquement généré. Néanmoins, la connaissance d'un tel logiciel ne vous dispensera pas (à terme, et pour l'utilisation de notions telles que CSS ou PHP) de l'étude du langage HTML.

Dreamweaver permet aussi d'obtenir une partie du Javascript nécessaire à quelques contrôles. Ce logiciel vous permettra de concevoir un site web de manière efficace et conviviale intégrant des pages web traditionnelles, mais aussi des styles. [WB.RATT]



Figure III.2 : L'interface Dreamweaver.

IV. Implémentation de la base de données :

Après le choix de la technologie à utiliser (dans notre cas c'est la technologie relationnelle) et en se basant sur le schéma conceptuel de données établi précédemment et après transformation et optimisation grâce aux techniques de Mapping nous obtiendrons le schéma relationnel suivant :

- **Administrateur** (id_admin, mot_pass_admin, login_admin, nom_admin, adress_admin, email_admin).
- **Utilisateur** (id_util, mot_pass_util, login_util, nom_util, prenom_util, adress_util, email_util, numtel_util, statut_util, #id_preins).
- **Employé** (id_emp, mot_pass_emp, login_emp, nom_emp, prenom_emp, adress_emp, email_emp, numtel_emp, #id_park).
- **Preinsc**(id_preinsc, mot_pass_preinsc, login_preinsc, nom_preinsc, prenom_preinsc, adress_preinsc, email_preinsc, numtel_preinsc, statut_preinsc, #id_park, #id_util).
- **Reservation** (id_res, date_res, heure_debut, heure_fin, #id_park).
- **Park** (id_park, nbre_place_t, nbre_place_abo, nbre_place_libre, adress_park, nom_park, #id_admin).
- **Contacte** (id_contact, email_contact, prenom, nom, message, heure_contacte).
- **attrib_res_util** (id_attrib, #id_util, #id_res).
- **Tarif** (id_tarif, prix, formule, #id_admin).
- **Vehicule** (num_immat, heure_entree, heure_sortie, #id_emp).

✓ **Signification :** : Clé primaire.
 # : Clé étrangère.

Le schéma relationnel étant ainsi établi, on passe à l'élaboration du modèle physique des données :

V. Stockage des données :**Table préinscription :**

Nom du champ	Type	Description	Clés
id_preinsc	Varchar(4)	Identifiant du préinscrit	Primaire
nom_preinsc	Varchar(50)	Nom du préinscrit	
prenom_preinsc	Varchar(50)	Prénom du préinscrit	
email_preinsc	Varchar(50)	E-mail du préinscrit	
adress_preinsc	Text	Adresse du préinscrit	
login_preinsc	Varchar(40)	Login d'authentification préinscription	
mot_pass_preinsc	Varchar(12)	Mot de passe préinscription	
numtel_preinsc	Int	Numéro du téléphone préinscription	
statut_preinsc	Varchar()	Statut de préinscriprion	
id_park	Varchar(4)	Identifiant du parking	Etrangère
id_util	Varchar(4)	Identifiant d'utilisateur	Etrangère

Table Utilisateur :

Nom du champ	Type	Description	Clés
id_util	Varchar(4)	Identifiant d'utilisateur	Primaire
nom_preinsc	Varchar(50)	Nom d'utilisateur	
prenom_util	Varchar(50)	Prénom d'utilisateur	
email_util	Varchar(50)	E-mail d'utilisateur	
adress_util	Text	Adresse d'utilisateur	
login_util	Varchar(40)	Login d'authentification utilisateur	
mot_pass_util	Varchar(12)	Mot de passe utilisateur	
numtel_util	Int	Numéro du téléphone utilisateur	
statut_util	Varchar()	Statut d'utilisateur	
id_preins	Varchar(4)	Identifiant du préinscription	Etrangère

Table administrateur :

Nom du champ	Type	Description	Clés
id_admin	Varchar(4)	Identifiant d'administrateur	Primaire
nom_admin	Varchar(50)	Nom d'administrateur	
email_admin	Varchar(50)	E-mail d'admanistrateur	
adress_admin	Text	Adresse d'administrateur	
login_admin	Varchar(40)	Login d'authentification administrateur	
mot_pass_admin	Varchar(12)	Mot de passe administrateur	

Table Employé :

Nom du champ	Type	Description	Clés
id_emp	Varchar(4)	Identifiant d'employé	Primaire
nom_emp	Varchar(50)	Nom employé	
prenom_emp	Varchar(50)	Prénom employé	
email_emp	Varchar(50)	E-mail employé	
adress_emp	Text	Adresse employé	
login_emp	Varchar(40)	Login d'authentification employé	
mot_pass_emp	Varchar(12)	Mot de passe employé	
numtel_emp	Int	Numéro du téléphone employé	
id_park	Varchar(4)	Identifiant du parking	Etrangère

Table Park :

Nom du champ	Type	Description	Clés
id_park	Varchar(4)	Identifiant du parking	Primaire
nbre_place_t	Int	Nombre de place totale	
nbre_place_abo	Int	Nombre de place des abonnés	
nbre_place_libre	Int	Nombre de place libre	
adress_park	Text	Adresse parking	
nom_park	Varchar(50)	Nom parking	
id_admin	Varchar(4)	Identifiant d'administrateur	Etrangère

Table Réservation :

Nom du champ	Type	Description	Clés
id_res	Varchar(4)	Identifiant de la réservation	Primaire
date_res	Date	Date e réservation	
heure_debut	Heure	Heure début de la réservation	
heure_fin	Heure	Heure fin de la réservation	
id_park	Varchar(4)	Identifiant du parking	Etrangère

Table Contacte :

Nom du champ	Type	Description	Clés
id_contact	Varchar(4)	Identifiant du contacte	Primaire
email_contact	Varchar(50)	E-mail u contacte	
message	Text	message	
heure_contacte	Heure	Heure du contacte	
nom	Varchar(50)	Nom du contacte	
prenom	Varchar(50)	Prénom du contacte	

Table attrib_res_util :

Nom du champ	Type	Description	Clés
id_attrib	Varchar(4)	Identifiant de l'attribution	Primaire
id_util	Varchar(4)	Identifiant d'utilisateur	Etrangère
id_res	Varchar(4)	Identifiant de la réservation	Etrangère

Table Tarif :

Nom du champ	Type	Description	Clés
id_tarif	Varchar(4)	Identifiant du tarif	Primaire
prix	Int	Prix u tarif	
Formule	Varchar(40)	Formule de paiement	
id_admin	Varchar(4)	Identifiant d'administrateur	Etrangère

Table Véhicule :

Nom du champ	Type	Description	Clés
num_immat	Varchar(4)	Numéro d'immatriculation	Primaire
heure_entree	Heure	Heure d'entrée en parking	
heure_sortie	Heure	Heure de sortie u parking	
id_emp	Varchar(4)	Identifiant d'employé	Etrangère

VI. Présentation de quelques interfaces :

§ Page d'accueil :

C'est la première page téléchargée et visitée par les visiteurs, c'est la page principale u site car elle donne une idée générale sur les fonctions de notre application.



Figure III.3 : Page d'accueil

§ *Page réservation :*

Cette page donne au client ou abonné la possibilité de faire une réservation dans un parking, pour y accéder, il doit cliquer sur le lien « réservation ». la page est composé d'un formulaire qui doit être remplis et validé en appuyant sur le bouton « envoyer ».

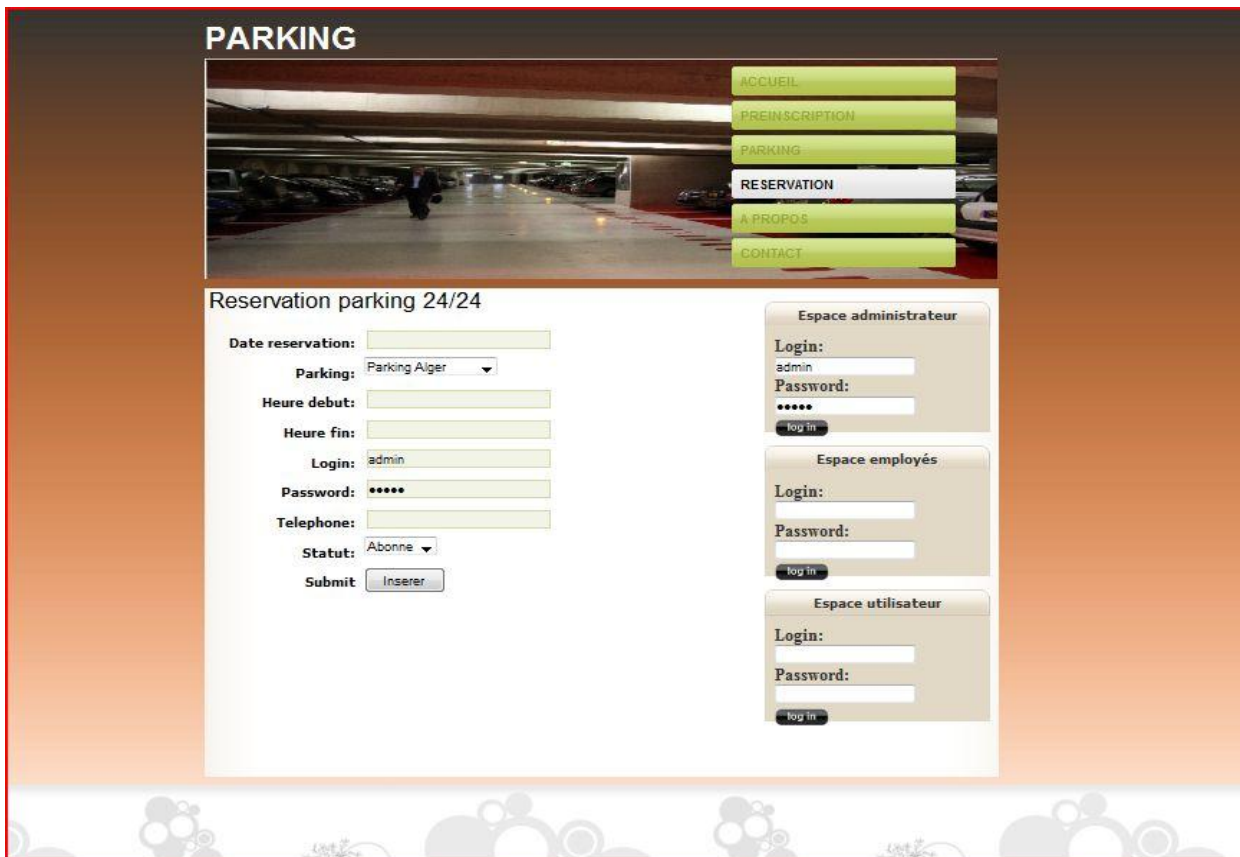
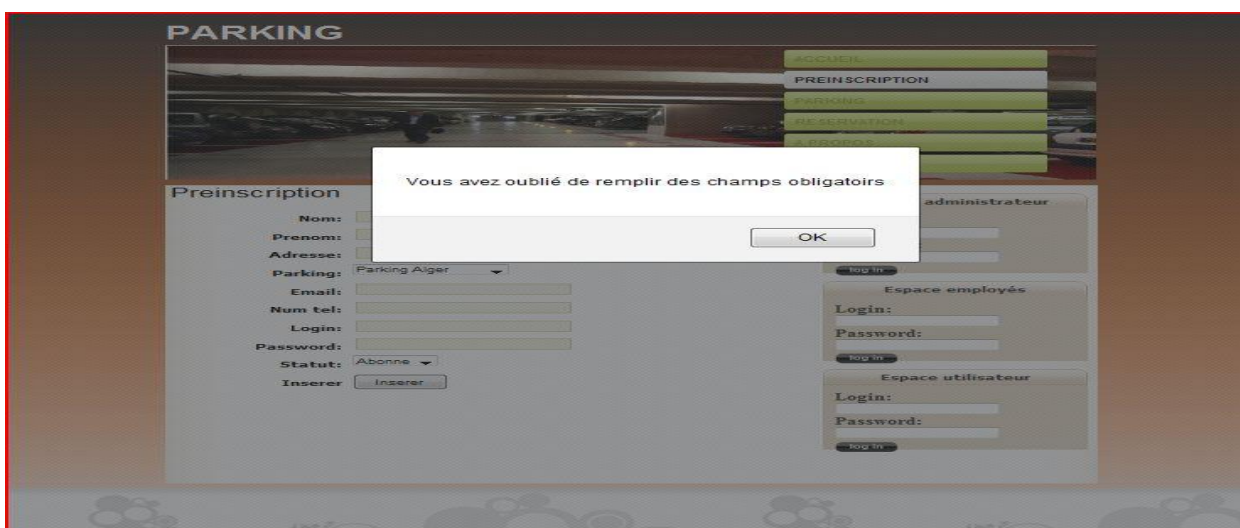


Figure III.4 : Page réservation.

En cas d'erreur un message sera afficher le voici :



§ Page préinscription :

Cette page permet au visiteur de faire une préinscription dans un parking du site.

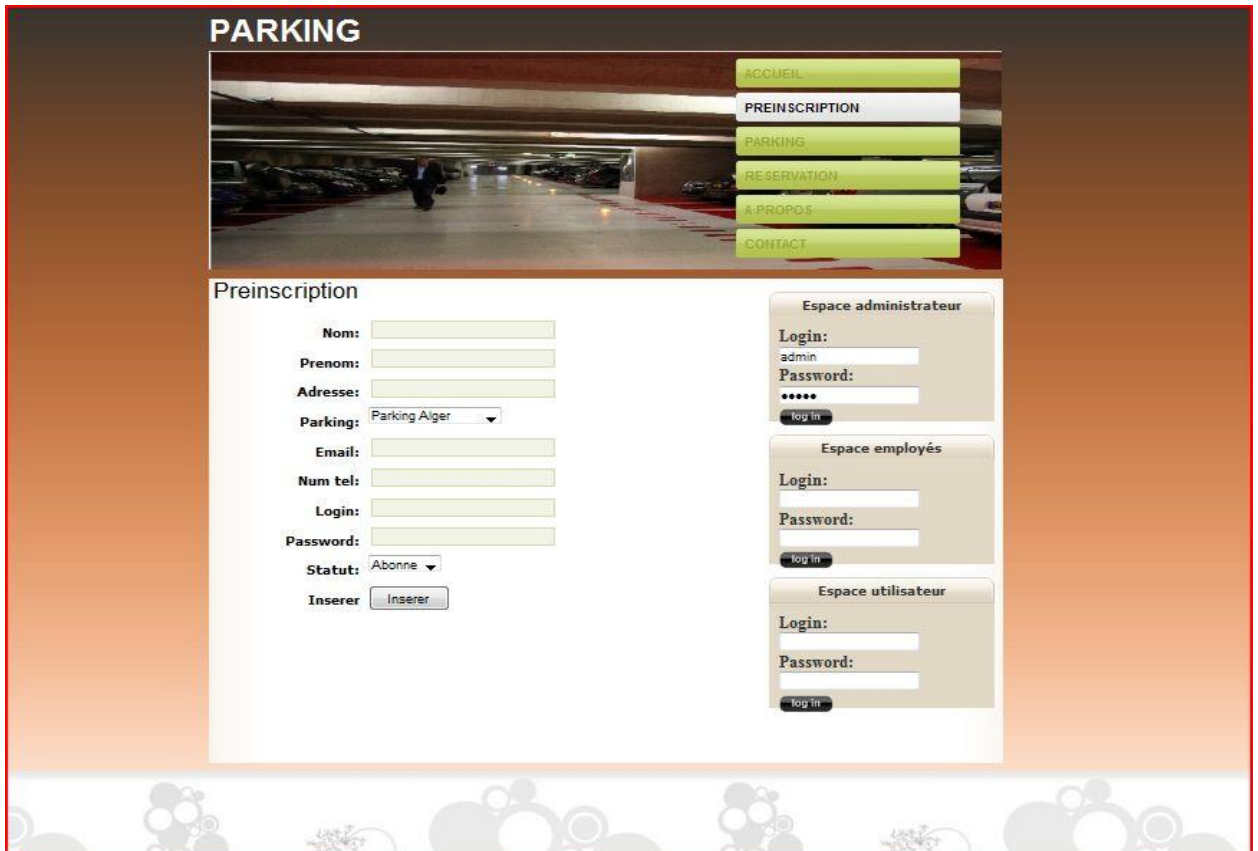
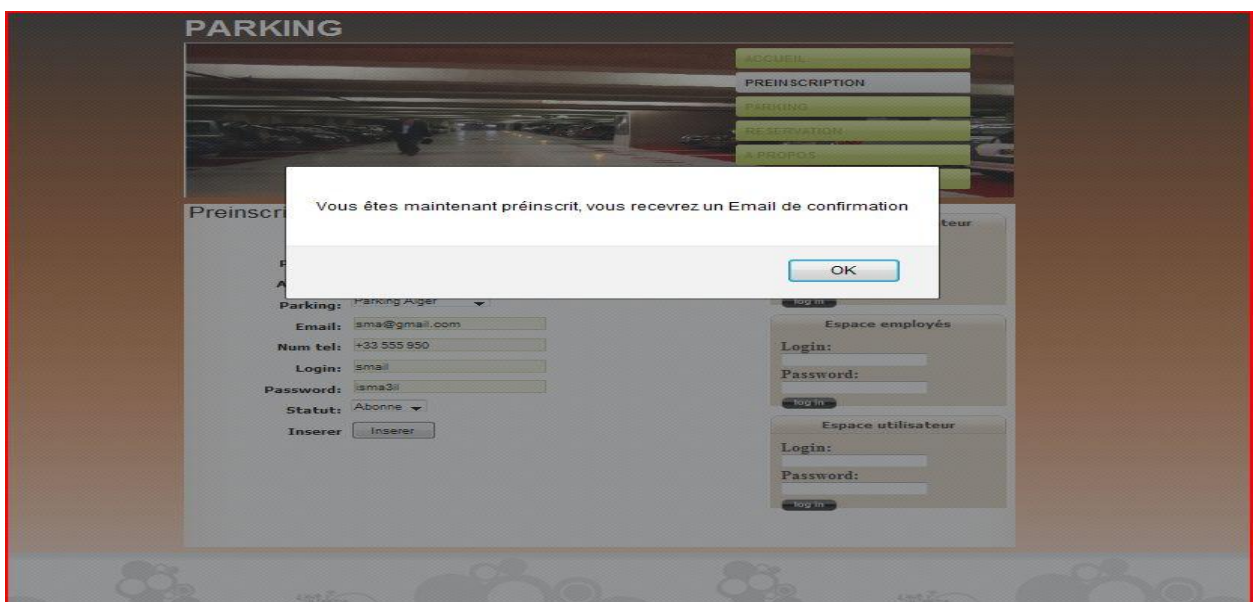


Figure III.5 : Page préinscription.

Une fois vous avez remplis le formulaire de préinscription, un message de confirmation vous sera afficher comme suit :



§ Page gestion contact :

Cette page permet au visiteur de laisser un message pour l'administrateur.

PARKING

ACCUEIL
PREINSCRIPTION
PARKING
RESERVATION
A PROPOS
CONTACT

Contacter nous

Nom:
Prenom:
Adresse:
Email:
Message:
Submit

Espace administrateur
Login: admin
Password:

Espace employés
Login:
Password:

Espace utilisateur
Login:
Password:

Figure III.6 : Interface contacter l'administrateur.

Conclusion :

Dans ce chapitre, nous avons présenté l'environnement d'implémentation et de développement de notre application, en se focalisant sur les techniques de programmation utilisées pour implémenter les pages client, les pages serveur et les animations interactives multimédias.

La description de notre application s'est faite en présentant les interfaces essentielles de notre site et en suivant des scénarios de navigations décrits schématiquement.

Le travail que l'entreprise nous a confié et que nous avons réalisé tout au long de notre stage nous a permis de découvrir le domaine de commercialisation et du commerce électronique.

Nous avons présenté une étude complète dans le sens où nous avons effectué une analyse à partir d'une problématique, définir des spécifications, concevoir et réaliser un environnement opérationnel que nous avons testé. Ce travail est aussi motivant et enrichissant, notamment par l'acquisition d'une expérience dans la conception avec le langage UML ainsi que de gagner de larges connaissances sur les langages de programmations tel que HTML, PHP, JavaScript et découvrir un ensemble de logiciels qui permet de faciliter le développement de logiciels. Et à travers ce thème nous avons eu l'occasion d'approfondir nos connaissances en matière de systèmes d'informations et des bases de données.

Cependant, malgré les diverses fonctionnalités qu'offre notre application, quelques améliorations peuvent être apportées.

En guise de perspectives :

- ✓ Réalisation d'un système pour régler les factures en ligne.
- ✓ Intégration d'un système d'information géographique.
- ✓ Intégration d'un système de reconnaissance de plaques minéralogiques ce qui permettra l'identification automatique des véhicules d'abonnés et des clients.

I. Introduction :

UML (Unified Modeling Language) est un langage de standard conçu pour l'écriture de plans d'élaboration de logiciels. Il peut être utilisé pour visualiser, spécifier, construire et documenter les artefacts d'un système à forte composante logicielle.

Dans cette annexe nous présentons la notation UML utilisée dans notre mémoire. (Pour plus de détails sur UML, consulter les livres traitant ce langage dont une liste figure en bibliographie).

II. Les briques de base d'UML :

La terminologie d'UML inclut trois sortes de briques :

- Des éléments.
- Des relations.
- Des diagrammes.

III. Eléments d'UML :

Il existe quatre types d'éléments dans UML :

- les éléments structurels,
- les éléments comportementaux,
- les éléments de regroupement,
- les éléments d'annotation.

III.1 Les éléments structurels :

Les éléments structurels sont représentés par des noms dans les modèles UML. Ce sont les parties les plus statiques d'un modèle : ils représentent des éléments conceptuels ou physiques.

III.1.1 La classe : [Re.Ar]

Elle représente un ensemble d'éléments qui partagent les mêmes attributs, les mêmes opérations, les mêmes relations et les mêmes sémantiques.

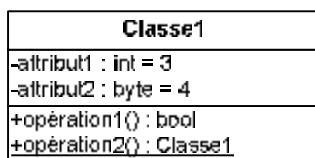


Figure A1 : classes

III.1.2 Les classes-associations :

Il est possible de représenter une association par une classe pour ajouter, par exemple, des attributs et des opérations dans l'association. Une classe de ce type, appelée **classe-associative** ou **classe-association**, possède à la fois les caractéristiques d'une classe et d'une association, et peut à ce titre participer dans d'autres relations dans le modèle.

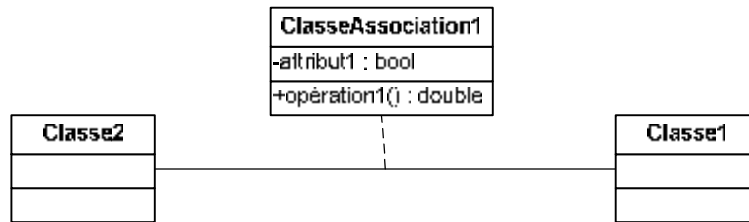
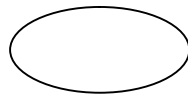


Figure A2 : classe-association.

III.1.3 Les cas d'utilisation :[UML]

Un cas d'utilisation est la description d'une séquence d'actions exécutées par un système pour conduire à un résultat qui peut être constaté par un acteur particulier. Il sert à structurer les éléments comportementaux d'un modèle et est réalisé par une collaboration comme le montre la **figure A3**, un cas d'utilisation est représenté par une ellipse en trait plein qui, en règle générale, contient seulement son nom.



Cas d'utilisation

Figure A3 : Cas d'utilisation.

III.2 Les éléments comportementaux :

Les éléments comportementaux représentent les parties dynamiques des modèles UML. Ce sont les verbes du modèle et ils représentent son comportement dans le temps et dans l'espace.

III.2.1 Les interactions :

Une interaction est un comportement qui comprend un ensemble de messages échangés au sein d'un groupe d'éléments, dans un contexte particulier, pour atteindre un but bien défini. Le comportement d'un ensemble d'objets ou celui d'une opération individuelle peut être précisé par une interaction. Cette dernière implique un certain nombre d'éléments, y compris des messages, des séquences d'actions (comportement induit par un message) et des liens (relations entre des éléments). Comme le montre la **figure A4**, un message est représenté par une ligne fléchée, qui indique le nom de son opération.



Figure A4 : Message

III.3 Les éléments de regroupement :

Les éléments de regroupements représentent les parties organisationnelles des modèles UML. Ce sont des boîtes dans lesquelles un modèle peut être décomposé. Il existe un seul type fondamental d'éléments de regroupement : le « paquetage ».

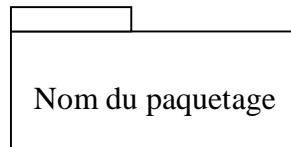


Figure A5 : Paquetage

III.4 Les éléments d'annotation :

Les éléments d'annotation représentent les parties explicatives des modèles UML. Ce sont les commentaires qui peuvent accompagner tout élément dans un modèle, à des fins de description, d'exploitation et de remarque. Il existe un type fondamental d'éléments d'annotation appelé « note » qui est simplement un symbole utilisé pour représenter les contraintes et les commentaires rattachés à un élément ou un ensemble d'éléments. Comme le montre la **figure A6**, une note est représentée par un rectangle écorné qui contient un commentaire textuel ou graphique.

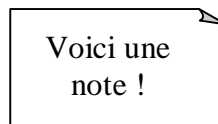


Figure A6 : Note

IV. Relations dans UML :

Il existe quatre types de relations dans UML :

- La dépendance.
- L'association.
- La généralisation.
- La réalisation.

IV.1 Dépendance :

C'est une relation sémantique entre deux éléments selon laquelle un changement apporté à l'un (élément indépendant) peut affecter la sémantique de l'autre (élément dépendant) comme le montre la **figure A7**, une dépendance est représentée par une ligne en pointillés qui peut être fléchée ; elle comprend parfois une étiquette.

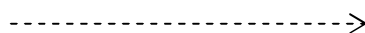


Figure A7 : Dépendance

IV.2 Association :

Une association est une relation structurelle qui décrit un ensemble de liens, un lien constituant une relation entre différents objets .L'agrégation est un type particulier d'association, qui représente une relation structurelle entre un tout et ses parties. Comme le montre la **figure A8**, une association est représentée par une ligne qui peut être fléchée ; elle comprend parfois une étiquette et souvent d'autres décorations, comme la multiplicité et les noms de rôles.

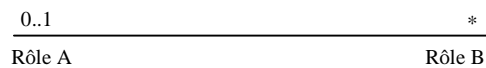


Figure A8 :Association

IV.3 Généralisation :

Une généralisation est une relation de spécialisation selon laquelle les attributs de l'élément spécialisé (l'enfant) peuvent se substituer aux attributs de l'élément généralisé (le parent).De cette manière, l'enfant partage la structure et le comportement du parent. Comme le montre la **figure A9**, une relation de généralisation est représentée par une flèche dont le trait est plein et dont la pointe creuse est dirigée vers le parent.

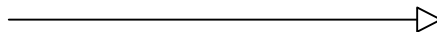


Figure A9 : Généralisation

IV.4 Réalisation :

La réalisation est une relation sémantique utilisée principalement soit pour indiquer qu'une interface est réalisée par une classe, soit pour indiquer qu'un cas d'utilisation est réalisé par une collaboration d'objets. On la représente par :

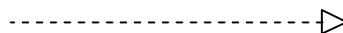


Figure A10 : Réalisation

V. Extensibilité :

UML comporte des mécanismes qui permettent d'étendre la syntaxe et la sémantique du langage.

V.1. Stéréotypes :

Les stéréotypes représentent de nouveaux éléments de modélisation, ils constituent un moyen de classer les éléments de la modélisation et facilitent l'élaboration du méta modèle d'UML.

Ils s'appliquent principalement aux classes et rendent possible l'identification d'une typologie de classe souvent nécessaire lorsqu'on manipule un grand nombre de classes. Le nom du stéréotype est indiqué entre guillemets.

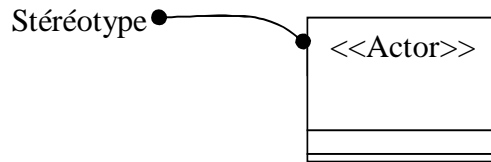


Figure .A11 : schéma représentatif d'une classe stéréotypée

V.2. Les contraintes:

Une contrainte est une note ayant une valeur sémantique particulière pour un élément de la modélisation, elle s'écrit entre accolades {}, elle peut concerner plusieurs éléments

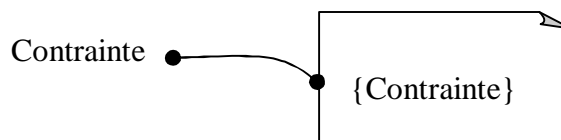


Figure .A12 : Schéma représentatif d'une contrainte

VI. Diagrammes d'UML :[UML]

Un diagramme est une représentation graphique d'un ensemble d'éléments qui constituent un système. La plupart du temps, il se présente sous la forme d'un graphe connexe où les sommets correspondent aux éléments et les arcs aux relations. Les diagrammes servent à visualiser un système sous différentes perspectives et sont donc des projections d'un système.

VI.1 Le diagramme de classes:

Le diagramme de classes est un diagramme structurel qui exprime d'une manière générale la structure statique d'un système en termes de classes, d'interfaces et de collaborations, ainsi que leurs relations.

VI.2 Le diagramme d'objets :

Le diagramme d'objets appelé aussi diagramme d'instance représente aussi une structure statique et montre des objets et des liens.

Les notations retenues pour les diagrammes d'objets sont dérivées de celles des diagrammes des classes ; les éléments qui sont des instances sont soulignés.

Pour le nom de l'objet on peut le désigner sous trois (03) formes :

- Nom de l'objet : désignation directe et explicite du nom d'objet

- Nom de l'objet : **nom de la classe**: Désignation incluant le nom de la classe.
- Nom de la classe : désignation anonyme d'un objet d'une classe donnée.

VI.3 Le diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation représente les cas d'utilisation, les acteurs et les relations entre eux.

Un acteur représente un rôle joué par une personne ou une chose qui interagit avec un système, il est représenté sous la forme de personnage et déclenche des cas d'utilisation. Il existe quatre (04) catégories d'acteurs à savoir :

- Les acteurs principaux : ce sont les personnes qui utilisent les fonctions principales du système.
- Les acteurs secondaires : ce sont des personnes qui effectuent des fonctions secondaires du système.
- Les matériels externes : Ce sont des dispositifs matériels nécessaires pour être utilisés.
- Les autres systèmes : ce sont des systèmes avec lesquels le système doit interagir. On peut schématiser l'acteur par la figure suivante :



Acteur

Figure. A13 : Représentation d'un acteur.

✓ Les relations entre cas d'utilisations : [BEBA/09]

Û Relation d'inclusion « Include » :

Une relation d'inclusion d'un cas d'utilisation "A" par rapport à un cas d'utilisation "B" signifie qu'une instance de "A" contient le comportement décrit dans "B", le cas d'utilisation "A" ne peut être utilisé seul.

Û Relation d'extension « Extend » :

Une relation d'extension d'un cas d'utilisation "A" par rapport à un cas d'utilisation "B" signifie qu'une instance de "A" peut être étendu par le comportement décrit dans "B".

Û Relation d'utilisation « Use » :

Le cas d'utilisation "A" est une généralisation de "B", si "B" est un cas particulier de "A" c'est-à-dire lorsque "A" peut être substitué par "B" pour un cas précis.

Ü Relation de généralisation (Relation entre acteurs) :

La seule relation possible entre deux acteurs est la généralisation : un acteur "A" est une généralisation d'un acteur "B" si l'acteur "A" peut être substitué par l'acteur "B". Dans ce cas, tous les cas d'utilisation accessibles à "A" le sont aussi à "B", mais l'inverse n'est pas vrai.

VI.4 Le diagramme de séquences :

Le diagramme de séquences montre les interactions entre objets selon un point de vue temporel. La représentation du contexte des objets se concentre sur l'expression des interactions.

Un objet est matérialisé par un rectangle et une barre verticale appelée ligne de vie des objets. Les objets communiquent en échangeant des messages représentés au moyen de flèches orientées, de l'émetteur du message vers le destinataire. L'ordre des messages est donné par leur position sur l'axe vertical. **[Re.Ar]**

Les autres diagrammes UML sont:

- † Le diagramme des activités qui décrit le comportement d'une opération en termes d'actions.
- † Le diagramme de collaboration qui est une représentation spatiale des objets, des liens et des interactions.
- † Le diagramme de composants qui décrit les composants physiques d'une application.
- † Le diagramme de déploiement qui décrit les composants sur les dispositifs matériels.
- † Le diagramme d'états transitions qui décrit le comportement d'une classe en terme d'états.

VII. Conclusion :

Cette annexe a proposé un survol et une présentation brève des principaux concepts de modélisation UML ainsi que leur notation.

Bibliographie

Mémoires

- ü [ING télécom-2000] : M.NEZROUK ABDERAHMANE et M.ACHACHI NAMANE, Conception d'un site web (application à la gestion d'une bibliothèque), 1999/2000, INSTITUT DES TELECOMUNICATION ABELHAFID BOUSSOUF-ORAN.
- ü [ING info 2009] : AIT TALEB et AITHEMI, Conception et réalisation d'une application web, 2008/2009, Université Mouloud Mammeri de Tizi-Ouzou.
- ü [BEBA/09] : BENCHABANE Bahia et BALHADJ Sonia, Conception et réalisation d'un portail web collaboratif, cas : la direction Coordination Groupe Finance de SONATRACH (DG), 2008/2009. Université Mouloud Mammeri de Tizi-Ouzou.
- ü [AKCH/08] : AKCHICHE et CHAHED, Conception et réalisation d'une application WEB pour la gestion du carburant, cas : Oil Field Service de SCHLUMBERGER NAG, 2007/2008, Université Mouloud Mammeri de Tizi-Ouzou.
- ü [ING info 2008-09] : HIMEUR & CHEBALLAH, Conception a réalisation d'une application web pour l'Electro- Industries, 2008/2009, Université Mouloud Mammeri.
- ü [Re.Ar] : Réseau architectures, protocoles, application.
- ü [UML] : UML outil de génie logiciel N.Aabdat, L.Mahdaoui, maison d'édition pour l'enseignement et la formation 2007.

Livres & Documents

- ü [LAI 02] : François LAISSUS, Cours d'introduction à TCP/IP, 2002.
- ü [DJE/09] : Polycopié complément de cours Communication de données et réseaux, Mr DJEMAH «4ème année Ingénieur Promotion 2008/2009».
- ü [LIV 08] : Les plus grandes réussites du web, Jessica Livingston & Judith Strauser, Edition ESF 2008.
- ü [D1] : DotNetGuru.org, Choisir un outil de Mapping objet relationnel.
- ü Document → Méthodologie des systèmes d'information – UML, fichier PDF créé par DI GALLO Frédéric.2000-2001.

Webiographie

- ü [WB.CCM] : <http://www.commentcamarche.net>
- ü [WB.WIKIV] : <http://www.wikiversity.org>
- ü [WB.USSK] : <http://www.User.skynet.be/ybet>
- ü [WB.APINF] : <http://www.aprendre-informatique.com>
- ü [WB.RATT] : <http://www.ratiatum.com>
- ü [WB.ALD] : <http://www.alaide.com>
- ü [WB.FSC] : <http://www.futura-sciences.com/fr/definition>
- ü [WB.DFT] : <http://damien.faucillon.free.fr/tutorial/PHPdef.htm>
- ü [WB.CCM] : <http://www.ccim.be/ccim328/js>
- ü [WB.HRHTM] : <http://www.henri-ruch.ch/HTML/HTML.asp>
- ü [WB.OBLG] : <http://obligement.free.fr/>
- ü [WB.INIW] : <http://www.itu.ini/newsarchive/wtd/2001/ExecutiveSummary.html>

Référence de quelques langages utilisés :

- § <http://www.php.net>
- § <http://www.mysql.com>
- § <http://www.phpmyadmin.sourceforge.net>