

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida  
USDB.

Faculté des sciences.  
Département informatique.

**Mémoire pour l'obtention  
d'un diplôme d'ingénieur d'état en informatique.**  
Option : SI

Sujet :

**CONCEPTION ET REALISATION  
D'UN SITE WEB DE COMMERCE  
ELECTRONIQUE B2C**

**Présenté par :** GAMOUDA Hamza  
DERBAL Billel

**Promotrice :** M<sup>lle</sup> S.EL Ghers

**Organisme d'accueil :** Concessionnaire de voiture Meklati (Citroën)

**Soutenue le:** date soutenance, devant le jury composé de :

Nom président du jury, grade, organisme

**Président**

Nom examinateur 1, grade, organisme

**Examineur**

Nom examinateur 2, grade, organisme

**Examineur**

- 2005/2006-

# Remerciement

*Avant tout nous remercions le dieu tout puissant qui nous a donnée la force du parvenir au bout de nos peines.*

*Nous tenons à adresser nos sincères remerciements à notre promotrice M<sup>lle</sup> S. EL GHERS qui a su nous conseiller par ses critiques constructives et nous guider dans notre modeste travail.*

*Nous remercions aussi les membres du département informatique, spécialement le Chef de département Mr Massied qui nous a assuré l'environnement adéquat afin de réaliser notre travail.*

*Nos sentiments de profonde gratitude vont à nos professeurs qui tout au long des cinq années d'études nous ont transmit leur savoir sans réserve.*

*Nos remerciements vont aussi à tous ceux et celles qui ont participé de près ou de loin à l'élaboration du présent travail.*

*Enfin, nous tenons à remercier tous nos amis et collègues pour leur soutien moral tout au long de la préparation de ce mémoire.*

# *Dédicace*

Je dédie ce modeste travail tout d'abord :

- A mes chers parents
- A mes frères et mes soeurs
- A mes Cousins et Cousines
- A toute ma FAMILLE grand et petit
- A ma Promotrice.
- A mon Binôme Hamza et sa FAMILLE
- A tous mes amies, spécialement « **RAFIK** », aissa, samir, yahia, sofiane, abelhak, brahim, taleb, rougi, farid, flita.

*Billel*

## Dédicace

C'est avec un grand plaisir que je dédie ce modeste travail, fruit de mes études en exprimant ma profonde gratitude à tous mes proches particulièrement :

A mes précieux parents pour leur amour, affection et compréhension.  
Ma mère pour l'éducation qu'elle m'a inculquée et toutes les peines qu'elle s'est donnée pour ma réussite.  
Mon père pour sa patience, ses encouragements et ses sacrifices.

A mes frères Lyes, Sid Ali et Billel.

A ma très chère soeur Amina.

A mon binôme Billel.

A toute ma famille et mes amis : Sofiane, Yahia, Rougi, Bourac, Oussama, Hamza, Amine, Biou, Mahfoud et Houria.

A mes trois chers amis, O.Salem, B.Rahim, E.Brahim.

A la mémoire de notre ami Nasr-eddine

A tous mes amis et collègues de la promotion sortante 2006

Hamza

# SOMMAIRE

INTRODUCTION GENERALE :.....	1
<b>Chapitre I : généralités sur Internet, le Web et L'XML</b>	
I. INTRODUCTION :.....	3
II.1 INTERNET : .....	3
II.1.1 Historique : .....	3
II.1.2 Fonctionnement d'Internet : .....	5
II.1.3 Les protocoles :.....	5
II.1.3.1 Les couches TCP/IP:.....	6
II.1.4 Les services d'Internet :.....	8
II.2 LE WORLD WIDE WEB:.....	8
II.2.1 Définition : .....	8
II.2.2 Historique : .....	8
II.2.3 Les caractéristiques techniques du Web :.....	9
II.2.3.1 Un protocole : HTTP .....	9
II.2.3.2 Un langage de description : HTML.....	11
II.2.3.3 Un localisateur universel de ressource : l'URL.....	11
II.2.3.4 Un browser (navigateur) : .....	12
II.2.3.5 Un serveur Web : .....	13
II.2.4 Les noms de domaine :.....	13
II.2.5 Site web : .....	13
II.2.6 Le fonctionnement du Web :.....	13
III- Le langage XML : .....	14
III.1- Origine et objectif de XML : .....	15
III.2- L'intérêt d'XML : .....	15
III.2- Définition d'XML :.....	15
III.4- Règles d'XML :.....	16
III.5- Avantages et inconvénients d'XML : .....	17
IV- CONCLUSION.....	18

## Chapitre II : LE COMMERCE ELECTRONIQUE

I- Introduction : .....	19
II- LE COMMERCE ELECTRONIQUE : .....	19
II.1- Définitions du commerce électronique : .....	19
II.2- Historique :.....	20
II.3- Avantages du commerce électronique : .....	20
II.4- Les acteurs du commerce électronique :.....	21
II.5- Les différents types du e-commerce : .....	22
II.6- Les fonctionnalités d'un site e-commerce:.....	24
III- Services de paiement électronique : .....	24
a\ Le crédit : .....	24
b\ Le débit : .....	24

c\ Le porte-monnaie électronique : .....	25
IV- Les conditions nécessaires pour le commerce électronique .....	25
V- Les risques dans le commerce électronique:.....	27
VI- Communications sécurisées:.....	27
VI.1- Cryptographie: .....	27
VI.2- Fonctionnement de la cryptographie par clé publique : .....	28
VI.3- Certificats : .....	28
VI.4- Les protocoles de sécurité : .....	28
a\ SSH: .....	28
b\ PGP: .....	29
c\ SSL:.....	29
VII- Conclusion : .....	30

### Chapitre III : Applications Web : Architecture et langages de programmation

I- Introduction.....	31
II.1- ARCHITECTURE DES APPLICATIONS WEB.....	31
II.1.1 Application Web.....	31
II.1.2 Architecture client-serveur.....	31
a\ Fonctionnement .....	32
b\ Caractéristiques.....	32
II.1.3 Les différents types d'architecture Client/Serveur.....	33
II.1.3.1 Présentation de l'architecture à 2 niveaux.....	33
II.1.3.2 Présentation de l'architecture à 3 niveaux.....	33
II.1.3.3 Présentation de l'architecture multi niveaux.....	34
II.1.4 Avantages et inconvénients de l'architecture client/serveur.....	35
II.2- LES LANGAGES DE PROGRAMMATION (SCRIPTS) .....	36
II.2.1- Langages coté serveur.....	36
II.2.2- Langages coté client.....	36
II.2.3- Les Scripts utilisés.....	38
II.2.4 ASP : définitions et fonctionnalités.....	39
II.2.5 Variables, constantes et types de données.....	42
II.2.6 ASP et le service de messagerie (Mail) .....	42
II.3- ASP ET L'INTERFAÇAGE AVEC LES BASES DE DONNEES.....	43
II.3.1 ADO (ActiveX Data Objects) .....	43
II.3.2 L'API ODBC.....	43
II.3.3 L'API OLE DB.....	43
II.3.4 Le modèle objet ADO.....	44
II.3.4.1 Le Système DSN (DataSource Name) .....	45
II.3.4.2 Le DSN-Less (Sans DSN) .....	45
II.4- ASP ET LA COMMUNICATION CLIENT-SERVEUR.....	46
II.4.1 Les formulaires.....	46
II.4.2 Les variables d'environnements.....	47
II.5- ASP ET LA GESTION DES TRANSACTIONS.....	48
III- CONCLUSION.....	48

### Chapitre IV : La Conception

I- Introduction.....	49
II- l'analyse des besoins.....	49

II.1-Présentation du site e-commerce.....	50
III- Conception du site e-commerce.....	51
III.1- Identification des acteurs.....	51
III.2- Identification des cas d'utilisation et les diagrammes de séquence.....	53
III.2.1- Représentation générale des cas d'utilisations.....	51
III.3- DIAGRAMME DES CLASSES.....	62
III.4- DIAGRAMMES D'ACTIVITES.....	64
III.5- DIAGRAMMES DE COLLABORATION.....	69
III.6-DIAGRAMME DE DEPLOIEMENT.....	71
III.7- Conception de la base de données.....	72
III.7.1- Règles de passage du model objet au modèle relationnel.....	72
1) Représentation des classes d'objets en tables.....	72
2) Représentation des associations d'objets en tables.....	72
3) Représentation de la généralisation de l'héritage simple.....	72
III.7.2- Traduction du modèle Objet En base de Données relationnelle.....	72
IV- CONCLUSION.....	73

## Chapitre V : Implémentation du site web

I-INTRODUCTION.....	75
II- CONFIGURATION LOGICIELLE.....	75
III - DEVELOPPEMENT A TROIS NIVEAUX .....	76
III-1 Niveau présentation .....	77
III-2 Niveau Métier .....	78
III-3 Niveau base de données .....	78
III-4 L'objet CitroenCommerce .....	78
III-5 Conception du modèle .....	79
IV- ENCHAINEMENT DES PAGES WEB DU SITE .....	80
V- LES INTERFACES DU SYSTEME .....	81
V.1- COTE CLIENT (VISITEUR).....	81
VI- CONSULTATION DES RAYONS.....	82
VII- SECURITE DE PAIEMENT : (LE PROTOCOLE SSL).....	88
VIII- UTILISATION L'XML POUR L'ECHANGE DE DONNEES .....	90
VIII.1- COTE ADMINISTRATEUR .....	91
IX- CONCLUSION.....	92
Conclusion générale & perspective .....	93
Référence	
Annexe A,B,C	

## **LISTE DES FIGURES**

### **Chapitre I**

Figure 1.1: Principe de fonctionnement d'Internet: .....	5
Figure 1.2 : Syntaxe d'une requête HTTP: .....	10
Figure 1.3 : Syntaxe d'une réponse HTTP: .....	11
Figure 1.4: Principe de fonctionnement du Web: .....	14
Figure 1.5: différence entre un document HTML et un document XML:.....	17

### **Chapitre II**

Figure 2.1: Marché Business to Business: .....	22
Figure 2.2: Marché Business to Consumer: .....	23
Figure 2.3 : Schéma de paiement électronique par carte de crédit:.....	25
Figure 2.4 : Fonctionnement de SSL:.....	29

### **Chapitre III**

Figure 3.1 : fonctionnement d'un système client/ serveur : .....	32
Figure 3.2 : Architecture Client/Serveur à 2 niveaux: .....	33
Figure 3.3 : Architecture Client/Serveur à 3 niveaux: .....	34
Figure 3.4 : Architecture Client/Serveur multi niveaux:.....	35
Figure 3.5 : Le modèle objet de l'ASP:.....	41
Figure 3.6 : Les fournisseurs d'accès OLEDB:.....	44
Figure 3.7 : Objets et Collections du modèle objet ADO: .....	44

### **Chapitre IV**

Figure 4.1 : l'architecture à 3-tiers du site Web de commerce électronique:.....	51
Figure 4.2 : Diagramme de cas d'utilisation: .....	53
Figure 4.3 : Diagramme de séquence - recherche produit -:.....	54
Figure 4.4 : Diagramme de séquence < Exception (produit non trouvé) >:.....	55
Figure 4.5 : Diagramme de séquence - gérer panier -: .....	56
Figure 4.6 : Diagramme de séquence - effectuer une commande -:.....	58
Figure 4.7 : Diagramme de séquence - imprimer un devis -: .....	59
Figure 4.8 : Diagramme de classes : .....	63
Figure 4.9 : Diagramme d'activité - Recherche produit -: .....	65
Figure 4.10 : Diagramme d'activité - Gérer panier -:.....	66
Figure 4.11 : Diagramme d'activité - Effectuer commande -: .....	67
Figure 4.12 : Diagramme d'activité - Maintenir le catalogue -:.....	68
Figure 4.13 : Diagramme de collaboration - recherche produit -: .....	69
Figure 4.14 : Diagramme de collaboration - Gérer panier -: .....	70
Figure 4.15 : Diagramme de collaboration - effectuer commande -: .....	70
Figure 4.16 : Diagramme de déploiement:.....	71

## Chapitre V

Figure 5.1 : les différents niveau d'une application distribuée orienté web .....	77
Figure 5.2 : Structure hiérarchique du Modèle Objet CitroenCommerce .....	79
Figure 5.3 : Enchaînement des pages Web: .....	80
Figure 5.4 : La page d'accueil: .....	81
Figure 5.5 : Le formulaire de recherche .....	82
Figure 5.6 : Schéma de structure des rayons: .....	82
Figure 5.7 : rayons des véhicules selon catégories.....	84
Figure 5.8 : rayon des véhicules touristique:.....	85
Figure 5.9 : Page Web qui Affiche le contenu du panier: .....	85
Figure 5.10 : Formulaire pour passer une commande:.....	86
Figure 5.11 : La commande finale: .....	86
Figure 5.12 : mode de paiement .....	87
Figure 5.13 : exemple de commande au format PDF.....	87
Figure 5.14 : formulaire de carte de crédit.....	88
Figure 5.15: La page de connexion à la session d'administration: .....	91
Figure 5.16 : La page d'ajout d'un véhicule: .....	92

## **Introduction générale :**

Les avancées réalisées ces dernières années dans les domaines de la micro-informatique, des logiciels et des technologies de l'information sont en train de changer nos vies d'une façon que nous n'aurions jamais pu imaginer il y a vingt cinq ans. Ces avancées ont permis l'apparition de nouveaux moyens de communication qui permettent d'échanger des données informatisées et d'effectuer des transactions commerciales uniquement par l'intermédiaire d'un ordinateur, ce qui transforme considérablement de nombreux aspects de notre organisation économique et sociale.

Avec l'apparition de l'Internet, nous pouvons communiquer à toute heure du jour et de la nuit avec d'autres internautes. Nous pouvons consulter tout type d'information, connaître la météo, les programmes télévision, les informations de n'importe quel pays afin de rechercher des livres ou des produits. Bien sûr, pour acheter un produit à distance, il était déjà possible de le faire grâce à des catalogues papier sur lesquels on choisissait l'objet voulu puis on le commandait par téléphone ou par courrier, mais pour obtenir ce catalogue, il fallait déjà connaître l'entreprise étrangère qui le produisait et le lui commander. On n'était pas vraiment en mesure d'avoir accès à tous les produits de tous les magasins du monde entier et de pouvoir les commander instantanément pour les recevoir quelques jours plus tard. Aujourd'hui, tout cela est possible, il est même possible d'acheter des produits, s'ils sont numérisés (applications, logiciel....etc.), directement via le web et de les recevoir immédiatement sur son ordinateur.

Les transactions commerciales (négociations, transactions, décisions) peuvent aujourd'hui être réalisées dans un cadre temporel et géographique qui aurait paru inconcevable il y a quelques années. On peut aujourd'hui avoir accès à des sources d'informations gigantesques, inépuisables et constamment renouvelées.

Le commerce électronique est désormais une donnée importante des activités économiques mondiales. En effet, il présente un secteur d'avenir qui offre aux citoyens et aux entreprises des perspectives en termes de compétitivité, de croissance et d'emploi. Cependant, L'intérêt porté par les entreprises au commerce sur Internet s'explique par les possibilités que ce nouveau canal de distribution offre en termes de services aux clients et de réduction des coûts. À cet égard, les attentes des entreprises sont de pouvoir offrir des services de meilleure qualité et plus de commodité aux clients lors de la réalisation de la transaction.

Les faiblesses du réseau Internet en matière de sécurité et en garantie de service le rendent inadapté à la conduite de transactions commerciales sérieuses. Néanmoins, toutes les

organisations, des petites aux grandes entreprises et administrations, pensent qu'Internet est une tendance majeure et irréversible, imposée par le développement économique et technologique.

Ce travail s'inscrit dans ce cadre, il consiste à réaliser un site web dynamique de commerce électronique pour le compte du concessionnaire des voitures Meklati (Citroën).

Le présent mémoire est structuré comme suit :

- Le premier chapitre est consacré au réseau Internet, son architecture, son fonctionnement ainsi que les protocoles et services qui font partie, et le langage XML qui permettant de décrire, à l'aide de balises personnalisées, la structure de données ou document de tous types.
- Le deuxième chapitre est réservé au commerce électronique objet de notre travail, on va vu de pré les différents aspects du commerce électronique en présentant ses différents types, ses fonctionnalités, ses avantages et ces risques, ainsi que la partie sécurité.
- Le troisième chapitre vise à donner un aperçu général sur l'architecture des sites Web dynamiques, l'architecture client/serveur à deux et trois niveaux et enfin les langages de programmation utilisés pour développer les sites Web dynamiques.
- Le quatrième chapitre sera consacré à l'analyse qui aura pour but la définition des besoins des utilisateurs, et la conception quant à elle définira l'architecture générale de l'application. UML sera utilisé comme langage de modélisation.
- Et enfin, le cinquième chapitre est réservé à l'implémentation de notre application. Nous évoquerons aussi dans ce chapitre quelques illustrations de navigation au sein de notre site Web dynamique.

# Chapitre I

## Chapitre I: Généralités sur Internet, le Web et L'XML

### I. Introduction :

Ce premier chapitre est partagé en deux parties, la première donne une vue générale sur les technologies du web (ou toile), la <toile virtuelle> formée par différents documents liés entre eux par des liens, c'est une énorme archive vivante composée d'une myriade de sites web proposant des pages web contenant du texte mis en forme, des images, des sons, des vidéos, etc, le web représente un des usages possibles offerts par le réseau Internet et constitue la principale façon de naviguer parmi des documents en ligne. La deuxième partie est consacrée pour le langage XML *eXtensible Markup Language*, qui est un format de balisage des données utilisé par de nombreuses applications, notamment les applications web.

### II.1 INTERNET :

Internet est le plus grand réseau (global) du monde. Il connecte des ordinateurs répartis sur tout le globe et permet l'échange de données numériques (textes, images, sons, vidéo, programmes informatiques) d'un ordinateur à l'autre suivant des protocoles précis. Il est possible d'y connecter un ordinateur de n'importe quelle partie du monde.

Le terme "Internet" vient du latin "inter" (entre) et de l'anglais "net" (réseau), et peut être interprété comme "réseau intermédiaire". Internet connecte des réseaux isolés, de manière à ce que l'échange de données et la communication électronique au niveau mondial devienne possible.

Les ordinateurs du monde entier sont connectés soit par des lignes téléphoniques, soit par des fibres optiques, des câbles à forte bande passante, des ondes radio dirigées ou des satellites [W3O 06]

#### II.1.1 Historique :

- **1969** : Création de l'ancêtre d'Internet, appelé **ARPANET**, sous l'influence du département de la défense américain. Les quatre premiers nœuds du réseau sont les suivants:
  1. Université de Californie, Los Angeles (UCLA)
  2. Stanford Research Institute, Menlo Park (SRI)
  3. Université de Californie, Santa Barbara (UCSB)
  4. Université d'Utah

Le premier paquet de données passe en octobre de UCLA au SRI. Il s'en suit le premier plantage du système, lors de l'introduction de la lettre G de LOGIN.

- **1972** : Introduction du service e-mail sur ARPANET. Apparaît alors le fameux caractère @, qui prend la signification "à" ("at").
- **1973** : L'"University College of London" en Angleterre et le "Royal Radar Establishment" en Norvège sont les premières connexions internationales à ARPANET, qui compte alors déjà plus de 2000 utilisateurs.
- **1974** : Première spécification du protocole TCP.
- **1975** : Apparition des premières listes de diffusion (mailing-lists).
- **1979** : Spécification et apparition des premiers news groups, qui étaient alors modérés.
- **1982** : Les protocoles TCP et IP sont établis comme standard pour ARPANET, et constituent la première définition d'Internet comme un groupe de réseaux qui utilisent spécifiquement ces protocoles.
- **1984** : Introduction du système de noms de domaines (DNS). Plus de 1'000 ordinateurs sont connectés.
- **1987** : Plus de 10'000 ordinateurs connectés.
- **1988** : Apparition du premier virus informatique sur Internet. 6'000 des 60'000 ordinateurs connectés sont infectés.
- **1989** : Plus de 100'000 ordinateurs connectés.
- **1991** : Invention du World Wide Web (WWW) au CERN (Suisse). L'inventeur est Tim Berners-Lee.
- **1992** : Plus de 1 million d'ordinateurs connectés.
- **2002** : plus de 36 millions de sites web. 600 millions d'utilisateurs. **[PAS IRD]**

### II.1.2 Fonctionnement d'Internet :

Internet fonctionne sur la base de protocoles dont le plus connu est TCP-IP. Ce protocole, en donnant une identité à chaque ordinateur (une adresse électronique), organise la relation entre les différentes machines sur le Net.

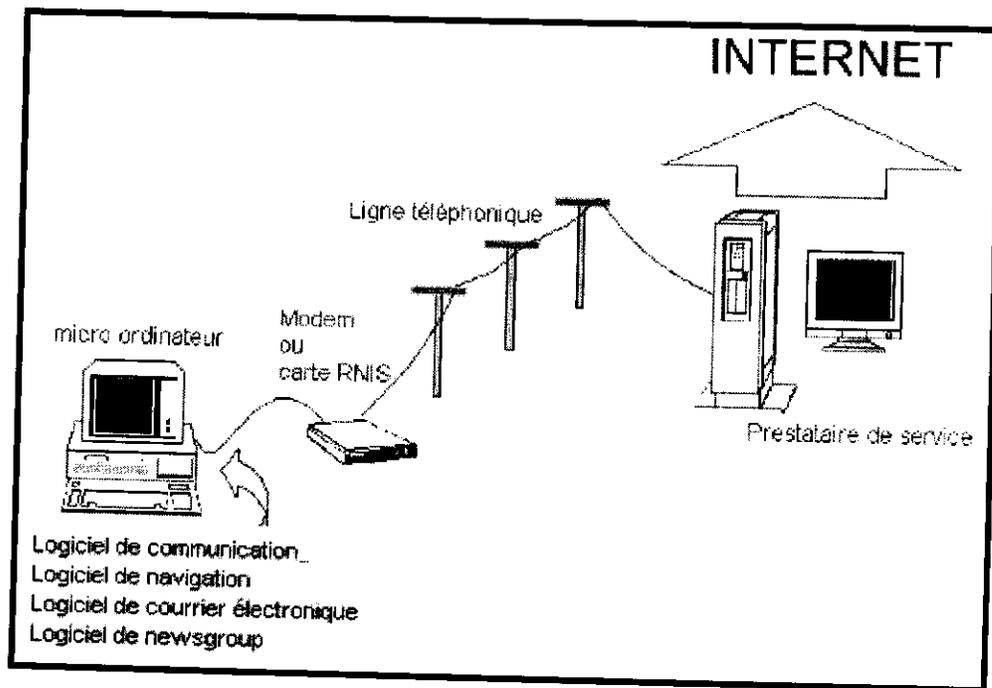


Figure 1.1: Principe de fonctionnement d'Internet

### II.1.3 Les protocoles :

« Un protocole est une méthode standard qui permet la communication entre deux machines, c'est-à-dire un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau » [DOU 01].

Sur Internet, les protocoles utilisés font partie d'une suite de protocoles liés entre eux, cette suite de protocoles s'appelle *TCP/IP*. Elle contient, entre autres les protocoles suivants:[DOU 01]

- **HTTP** : ( Hypertext Transfert Protocol) protocole permettant le transfert de fichiers (essentiellement au format HTML) localisé grâce à L'URL
- **HTTPS** : ( Hypertext Transfert Protocol Secure), protocole de transmission issu de Netscape lié à une connexion par socket sécurisée. C'est de HTTP avec une pincée de SSL (Secure Socket Layer).

- **URL** : (Uniform Resource Locatore), adresse des pages Internet exploitée par les navigateurs (Explorer, Netscape...), nous le verrons en détail dans la suite du chapitre.
- **ICMP** : (Internet Control Message Protocol), protocole qui permet de gérer les informations relatives aux erreurs des machines connectées. Il ne permet pas de corriger ces erreurs mais en fait part aux protocoles des couches voisines.
- **Telnet** : protocoles standard permettant l'interfaçage de terminaux et d'applications.
- **UDP** : (User Datagram Protocol), offre un service de transport minimal, protocole non orientés connexion, il envoie des données sans prévenir la machine réceptrice.
- **FTP** : (File Transfer Protocol), protocole qui permettant de transférer des fichiers ou de pages web entre l'ordinateur local et un ordinateur distant en utilisant le protocole TCP/IP.
- **TCP** : (Transmission Contol Protocol), permet de remettre en ordre les data-grammes en provenance du protocole IP, de vérifier le flot de données afin d'éviter une saturation du réseaux, de formater les données en segments de longueur variable afin de les remettant au protocole IP.
- **IP** : (Internet Protocol), c'est un des protocoles les plus importants d'Internet car il permet l'élaboration et le transport des data-grammes IP (les paquets de données).

#### **II.1.3.1- Les couches TCP/IP :**

Quatre couches sont suffisantes pour définir l'architecture de TCP/IP : [FRA 97]

- 4 Couche Application (Application layer).
- 3 Couche Transport (Transport layer).
- 2 Couche Internet (Internet layer).
- 1 Couché interface réseau (Network access layer).
- 0 Matériel (n'est pas une couche comprise dans le protocole).

**II.1.3.1.1 La couche Application :**

Au plus haut niveau les utilisateurs invoquent les programmes qui permettent l'accès au réseau.

Chaque programme d'application interagit avec la couche de transport pour envoyer ou recevoir des données. En fonction des caractéristiques de l'échange le programme a choisi un mode de transmission à la couche de transport.

**II.1.3.1.2 La couche transport :**

La principale tâche de la couche de transport est de fournir la communication d'un programme d'application à un autre. Cette couche peut avoir à réguler le flot de données et à assurer la fiabilité du transfert : les octets reçus doivent être identiques aux octets envoyés.

**II.1.3.1.3 La couche Internet :**

Cette couche reçoit des data-grammes en provenance de la couche réseau qu'elle doit analyser pour savoir s'ils lui sont adressés ou pas. Dans le premier cas elle doit 'décapsuler' son en-tête du data-gramme pour transmettre les données à la couche de transport et au bon protocole de cette couche, dans le deuxième cas elle les ignore.

Cette couche prend aussi en charge la communication de machine à machine. Elle accepte des requêtes venant de la couche de transport avec une identification de la machine vers laquelle le paquet doit être envoyé.

**II.1.3.1.4 La couche interface réseau :**

Le protocole dans cette couche définit le moyen pour un système de délivrer l'information à un autre système physiquement reliée. Il définit comment les data-grammes IP sont transmis. La définition de ceux-ci reste indépendante de la couche réseau, ce qui leur permet de s'adapter à chaque nouvelle technologie au fur et à mesure de leur apparition.

### II.1.4 Les services d'Internet :

Les principaux services d'Internet sont liés à la communication tels que : le Web, le courrier électronique (*e-mail*), transfert de fichiers (*ftp*).....etc. Chacun de ces services utilise un logiciel spécifique.

Il est possible de les classer en deux catégories: la première concerne les outils qui privilégient l'information tels que les outils de messagerie *Email* (Electronic mail) et les forums de discussion (*NEWS*), tandis que la seconde englobe les outils basés sur la recherche documentaire tels que *FTP* et le *Web*.

## II.2 LE WORLD WIDE WEB:

### II.2.1 Définition :

Le web est le service le plus connu et maintenant le plus utilisé de consultation d'hyper documents. C'est un ensemble infini de sites reliés entre eux, hébergés sur les réseaux d'Internet et accessible par l'utilisation d'un logiciel : le browser (ou navigateur) capable de lire le langage *HTML*.

Le World Wide Web est un système d'information ouvert, conçu spécifiquement pour simplifier l'utilisation et l'échange de documents [GUY 98]

### II.2.2 Historique :

La technologie Web a été inventée au *CERN* (Centre Européen de Recherche Nucléaire) en 1989 par Tim Berners-Lee.

L'objectif était la diffusion de l'information en physique nucléaire. L'idée essentielle était d'organiser les informations sous forme d'un hypertexte permettant les liens entre les documents.

Ainsi est née l'idée d'un vaste système d'information distribué à l'échelle mondiale sur lequel les utilisateurs peuvent naviguer depuis leurs terminaux : la toile d'araignée mondiale des documents désignée par l'acronyme *WWW* ou le diminutif *Web* (World Wide Web) [GAR 99].

## II.2.3 Les caractéristiques techniques du Web :

### II.2.3.1 Le protocole HTTP :

Comme indiqué précédemment, le protocole est le langage commun parlé par le client et le serveur pour se comprendre et mener à bien les requêtes de l'utilisateur. Le protocole de loin le plus utilisé sur le World Wide Web est le protocole HTTP (*HyperText Transfer Protocol*).

C'est un protocole développé spécialement pour le transfert de documents écrits en HTML. Une requête HTTP est envoyée par le navigateur au serveur, qui renvoie une réponse HTTP. Dans cette optique, HTTP est un protocole sans état (il ne maintient pas d'information interne au cours d'une connexion) afin de traiter avec un maximum d'efficacité des requêtes extrêmement simples mais nombreuses. [OSC 98]

#### II.2.3.1.1 Description :

Le protocole HTTP est très simple. Une transaction HTTP se décompose en effet en quatre phases :

- Connexion
- Requête
- Réponse
- Fermeture

La **figure 1.2** montre la syntaxe d'une requête HTTP. Elle comprend :

- **une ligne de requête:** c'est une ligne précisant le type de document demandé, la méthode qui doit être appliquée, et la version du protocole utilisée.
- **Les champs d'en-tête de la requête:** il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la requête et/ou le client Navigateur, système d'exploitation,....
- **Le corps de la requête:** C'est un ensemble de lignes optionnelles qui permettant par exemple un envoi de données par une commande POST lors de l'envoi de données au serveur par un formulaire.

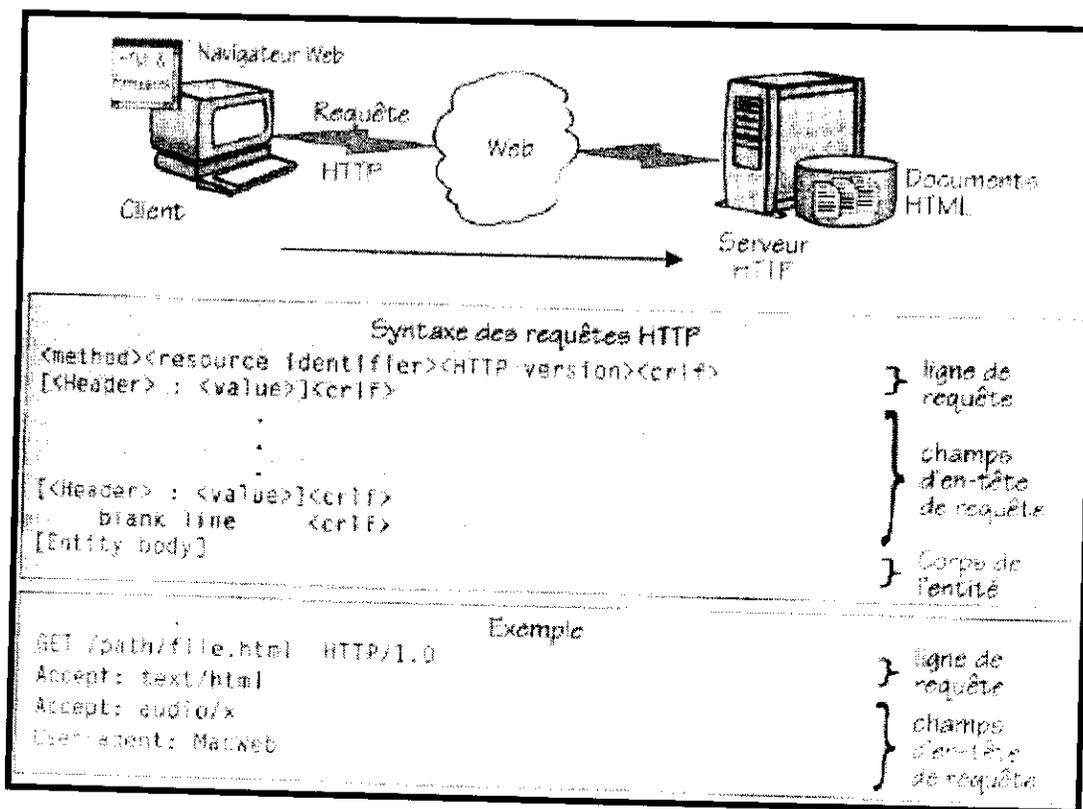


Figure 1.2 : Syntaxe d'une requête HTTP

Le schéma de la figure 1.3 présente la syntaxe d'une réponse HTTP. Elle comprend :

- **La ligne d'en-tête de réponse** : fournit au client la version de HTTP, le statut de la réponse et des explications sur le statut renvoyé.
- **Les champs d'en-tête de réponse** donnent aux client des informations qui décrivent les attributs du serveur et le document retourné.  
Il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et/ou le serveur. Chacune de ces lignes est composée d'un nom qualifiant le type d'en-tête, suivi de deux points (:) et de la valeur de l'entête.
- **Le corps de la réponse**: Il contient le document HTML demandé.

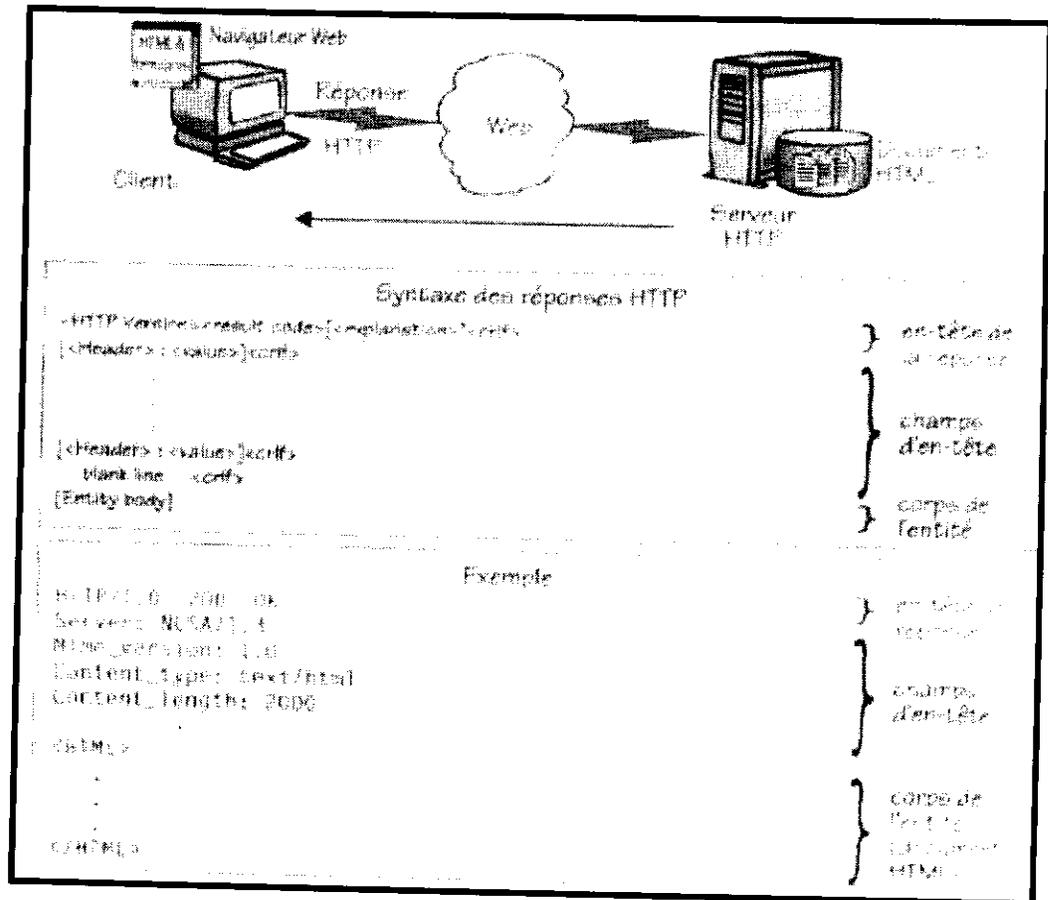


Figure 1.3 : Syntaxe d'une réponse HTTP

### II.2.3.2 Un langage de description : HTML

Les informations transmises par le serveur Web via *HTTP* sont portées au regard de l'utilisateur. La présentation et la mise en forme des documents sont codifiées par un langage nommé *HTML*.

*HTML* décrit la présentation du contenu d'une page Web [DAV 00]: les éléments à afficher sont encadrés par des balises qui précisent leurs mise en forme. Ainsi, différentes balises permettent de changer la taille des caractères, l'alignement,...etc.

Les balises sont fixes, nous citons quelques exemples :

- *<HTML>* et *</HTML>* : Début et fin du document *HTML*.
- *<BODY>* et *</BODY>* : Début et fin du corps du document *HTML*.
- *<TITLE>* et *</TITLE>* : Début et fin du titre du document *HTML*.

### II.2.3.3 Un localisateur universel de ressource : l'URL

Chaque page Web possède un nom unique permettant de l'identifier. Ce nom, le localisateur universel de ressource (*URL* : Uniforme Resource Locator), commence par la

description de la méthode d'accès utilisée, le protocole de transfert, dont dépend le format du reste de l'*URL*.

Une URL de type HTTP est structurée par exemple de la façon suivante :

*Protocole (http) : //nom\_DNS : numéro\_port / répertoire / nom\_fichier*

*Exemple: http: //www.whatis.edu.com/index.html*

Voici la signification des différents éléments :

- Protocole : le nom du protocole employé pour transmettre l'information, dans notre cas il s'agit de HTTP.
- Le nom **DNS** ( Domain Name Server : Serveur de noms utilisé pour établir la correspondance entre le nom de domaine et le numéro IP, c'est à dire traduire les noms d'hôtes en adresses Internet), commençant généralement par www et finissant par un code désignant le pays (dz, fr, etc.) ou le type d'activité de ce serveur (com, edu, org, etc.).
- Numéro\_port : le numéro de port qui permet de joindre le processus de serveur Web. On dit que le serveur Web « écoute » sur ce port. Avec le protocole HTTP, il s'agit par défaut du numéro 80, TELNET est identifié par le numéro 23 ; un numéro de port est un canal de communication virtuel entre deux applications localisées sur des machines différentes.
- Répertoire : la localisation du fichier sur le serveur physique, il peut s'agir d'un emplacement absolu par rapport à la racine de l'emplacement des fichiers HTML sur le serveur, ou d'alias.
- Nom\_fichier : le nom du fichier HTML proprement dit.

#### **II.2.3.4 Un browser (navigateur) :**

C'est un logiciel (ex : Netscape ou Internet explorer) qui permet la navigation sur le Web en allant d'une adresse de document (*URL*) à une autre. Sa fonction principale consiste à disposer des informations numériques en format *HTML* dont le résultat est une mise en page graphique adaptée à la taille de l'écran [DAV 00].

### II.2.3.5 Un serveur Web :

Un serveur Web est un processus (programme) capable de traiter les requêtes *HTTP* issues des clients Web (navigateurs) [DAV 00].

Ce processus est présent sur un serveur physique (une machine) accessible par les ordinateurs du réseau via *TCP/IP*. Ce serveur contient les informations qui peuvent être mises à disposition des utilisateurs sous forme de fichiers *HTML* localisés dans un ou plusieurs périphériques de stockage, généralement des disques durs.

### II.2.4 Les noms de domaine :

Un nom de domaine est constitué d'une suite de mots élémentaires séparés par un délimiteur : le point.

Pour un système de nom de domaine DNS, chaque mot élémentaire est appelé un label. C'est ainsi que le nom de domaine « *cs.purdue.edu* » contient trois labels *cs*, *purdue* et *edu*, le dernier label ou l'occurrence « *edu* » est dit label de niveau supérieure (top level domain). [DOU 01]

### II.2.5 Site Web:

Les pages web sont généralement organisées autour d'une page d'accueil, à l'aide des liens hypertextes. Cet ensemble cohérent de pages web est appelé site web.

Le site web (aussi appelé site Internet) est hébergé sur le serveur web et doit être connecté en permanence à Internet. [JEN 05]

### II.2.6 Le fonctionnement du Web :

Le schéma de la **figure 1.4** illustre comment fonctionne le Web en supposant que l'on veut accéder au site web:

- L'utilisateur, à l'aide du navigateur Web, contacte le serveur Web où se trouve le site en tapant l'adresse Web (*URL*).
- Le serveur Web accepte la demande, recherche l'information, et répond en envoyant un document sous la forme d'une page Web.
- Le navigateur Web interprète alors le document et l'affiche sur l'écran de l'utilisateur.

En ce qui concerne les images, les applets Java ou autres petites applications incluses dans le document, le navigateur Web les obtient à partir du serveur sur lequel ils se trouvent.

Le Web est considéré comme un système client/serveur. L'ordinateur qui demande les données joue le rôle de client, tandis que celui qui lui fournit ces données est dit serveur.

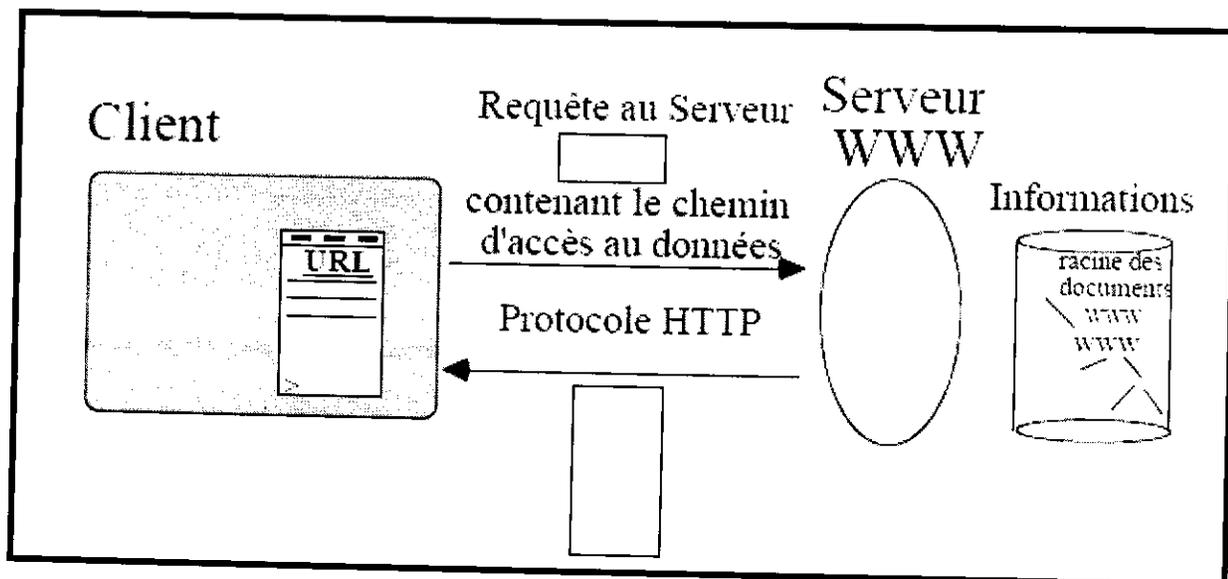


Figure 1.4: Principe de fonctionnement du Web

### III- Le langage XML :

Depuis sa première publication, en 1998, le langage **XML**, *eXtensible Markup Language*, est devenu un format de balisage des données utilisé par de nombreuses applications. Cela s'explique par le fait que **XML** est capable de baliser aussi bien un contenu *web* que des données utilisées par les applications. En fait, un document **XML** transporte les données et les informations sur les données transportées. Il a été conçu pour rendre l'utilisation de **SGML** (Standard Generalized Markup Language) sur le Web facile et directe, il simplifie le degré d'optionalité de SGML tout en permettant de développer sur le Web des types de documents créés par l'utilisateur. Avec XML, il sera facile de définir des types de documents, facile de créer et de gérer des documents définis en SGML et facile de les transmettre et de les partager sur le Web. Il permet de supprimer deux contraintes qui limitent les développements du Web :

- dépendance envers un type de document unique et non flexible (HTML).
- complexité du SGML intégral, dont la syntaxe autorise un grand nombre d'options puissantes mais difficiles à programmer.

Ainsi, dans la mesure où il permet aux sociétés et à leurs ordinateurs de communiquer plus facilement, XML est le socle d'un ensemble de nouvelles façons de communiquer au travers d'Internet.

### **III.1 Origine et objectif de XML**

XML est le résultat de la coopération d'un grand nombre d'entreprise et de chercheurs partenaires du **Word Wide Web Construm (W3C)**. Le développement de XML a commencé en 1996 et il a été approuvé en février 1998. Il a été développé par un groupe de travail (GT) XML [XML Working Group] (initialement connu sous le nom de comité d'examen éditorial SGML [SGML Editorial Review Board]). [ALA 00]

### **III.2 L'intérêt d'XML :**

XML, est un métalangage dérivé de SGML (Standard Generalized Markup Language). SGML est un métalangage particulièrement puissant, mais en contrepartie très complexe. Créé en 1986, il a été largement adopté dans certaines industries comme l'aéronautique. Malheureusement, sa complexité le rend peu adapté au Web. XML est une version simplifiée de SGML, destinée à rendre plus aisé la définition de documents et le développement d'applications les manipulant. Il omet les parties les moins utilisées de SGML, en contrepartie d'un emploi plus simple. Les objectifs recensés par les initiateurs d'XML étaient de rendre SGML applicable au Web [KNA 99]

### **III.3 Définition d'XML :**

XML est l'acronyme d'eXtensible Markup Language. Il est un sous-ensemble du langage SGML (Standard Generalized Markup Language). XML n'est pas réellement un langage, c'est un métalangage à balise, utilisé pour définir des langages de description. Il a été conçu pour être totalement compatible avec son antécédent SGML. [DEN 05]

Le langage XML possède dix règles de base qui ressortent des objectifs fixés par le W3C :

- XML doit être directement utilisable sur Internet et sans difficultés.
- XML doit pouvoir supporter une large variété d'applications.
- XML doit être compatible avec SGML.
- Il doit être facile d'écrire des programmes qui puissent traiter les documents XML.
- Le nombre des caractéristiques optionnelles d'XML doit être minimum, idéalement il doit même être nul.
- Les documents XML doivent être humainement lisibles et raisonnablement clairs.
- La construction de documents XML doit pouvoir être préparée rapidement.
- Elle doit être formelle et concise.

- Les documents XML doivent être facile à créer. [W3C 06]

#### **III.4 Règles d'XML :**

Le XML impose des règles de syntaxe très spécifiques par rapport aux autres langages de balisage:

- Il permet de définir ses propres balises et ses propres attributs. Il est donc plus flexible que HTML qui, lui, ne possède qu'un nombre limité de balise.
- Un document XML peut être validé par des règles strictes, contenues par des DTD ou des Schémas, décrivant sa structure et la hiérarchie de ses données.
- XML est un format standardisé ouvert ne nécessitant aucune licence, intégralement basé texte et qui peut être associé à n'importe quel jeu de caractère.
- XML est un document portable, il peut être lu sur n'importe quelle plate forme car c'est du texte et n'importe quel outil pouvant lire un fichier texte peut lire un document XML.
- De plus en plus d'applications utilisent le format XML ; C'est le cas de certains SGBD (Système de Gestion de Base de Données) mais aussi d'outils de bureautique comme Microsoft Office 2003 ou Sun Open Office. XML est également au cœur de la nouvelle plate-forme de développement de Microsoft.Net.
- son interopérabilité et le fait que de grands acteurs de l'informatique dont IBM, Microsoft et Sun préconisent l'utilisation de ce puissant langage. [W3C 06]
- enfin, Séparation du contenu, de la structure et de la présentation :

L'idée centrale d'XML est qu'il permet d'apporter de la valeur ajoutée si les trois aspects fondamentaux d'un document que sont son contenu, sa structure et sa présentation sont séparés pour celui qui le rédige contrairement a HTML [YAC 01]

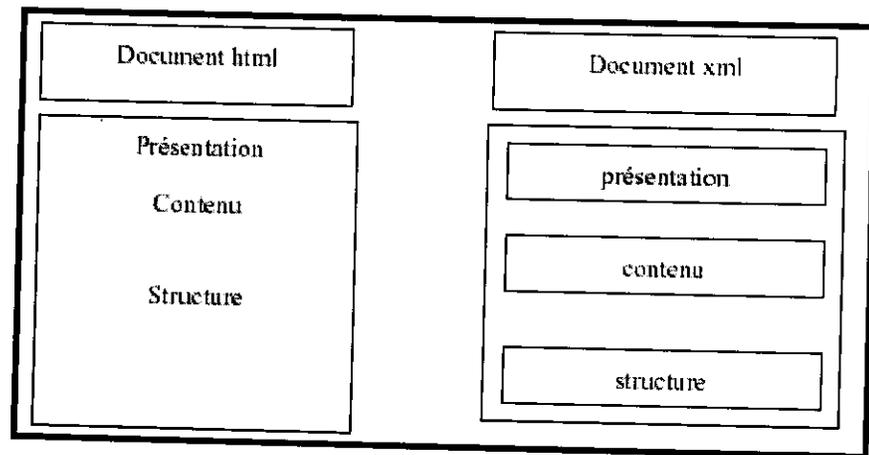


Figure 1.5 : différence entre un document HTML et un document XML.

### III.5 Avantages et inconvénients d'XML :

- **La lisibilité** : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML
- **Une structure arborescente** : permettant de modéliser la majorité des problèmes informatiques
- **Universalité et portabilité** : les différents jeux de caractères sont pris en compte.
- **Déployable** : il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme http.
- **Intégrabilité** : un document XML est utilisable par toute application pourvue d'un parser (un logiciel permettant d'analyser un code XML).
- **Extensibilité** : un document XML doit pouvoir être utilisable dans tous les domaines d'applications.
- Ainsi, XML est particulièrement adapté à l'échange de données et de documents. L'intérêt de disposer d'un format commun d'échange d'information dépend du contexte professionnel dans lequel les utilisateurs interviennent. C'est pourquoi, de nombreux formats de données issus de XML apparaissent (il en existe plus d'une centaine) :
- **OFX** : **O**pen **F**inancial **eX**change pour les échanges d'informations dans le monde financier
- **MathML** : **M**athematical **M**arkup **L**anguage permet de représenter des formules mathématiques
- **CML** : **C**hemical **M**arkup **L**anguage permet de décrire des composés chimiques
- **SMIL** : **S**ynchronized **M**ultimedia **I**ntegration **L**anguage permet de créer des présentations multimédia en synchronisant diverses sources : audio, vidéo, texte,...

Parmi les inconvénients on trouve que :

- Le principal inconvénient est le format texte lui même Les données stockées au format texte sont en général plus volumineuses que celles stockées au format binaire.
- Le tableur est mieux compris par Excel : mais les choses ne peuvent pas forcément toujours être décrites par des tableaux 2-dimensions. De plus on peut difficilement ajouter des commentaires dans les fichiers tabulaires. . [W3C 06]

#### **IV- Conclusion :**

Ce premier chapitre est consacré au réseau Internet, son architecture, son fonctionnement ainsi que les protocoles et services qui font partie, et le World Wide Web, ses caractéristiques techniques ainsi que son fonctionnement. Aussi Dans cette partie nous avons présenté L'XML, qui est un langage permettant de décrire, à l'aide de balises personnalisées, la structure de données ou document de tous types.

Toutes les applications qui importent, exportent, manipulent, stockent ou transmettent des données l'ont adopté comme standard. Cela inclut bien sûr toutes les applications liées au web (sites e-commerce, portail Internet ou Extranet...), mais aussi les bases de données, les logiciels de bureautique ou de graphisme, les logiciels de gestion, les outils de CRM (gestion de relation client) et les applications mobiles (WAP).

# Chapitre II

## **Chapitre II : LE COMMERCE ELECTRONIQUE**

### **I- Introduction :**

Les entreprises s'éloignent peu à peu du commerce traditionnel, et font appel au commerce électronique, ceci est dû aux facilités ainsi qu'au gain de ce qu'il offre. Dans ce chapitre nous allons définir le commerce électronique qui est le but de notre travail, ses différents types, avantages, etc.

### **II- LE COMMERCE ELECTRONIQUE :**

#### **II.1- Définitions du commerce électronique :**

Il n'y a pas de définition universelle du commerce électronique en raison du grand nombre de marchés et d'acteurs sur Internet et de l'évolution rapide de leurs rapports complexes.

##### **Définition 1 :**

Le fondement de la définition du commerce électronique découle de deux mots : électronique et commerce. Le terme électronique suppose que l'activité est réalisée grâce à l'utilisation de bits plutôt que d'atomes. Commerce, par ailleurs, peut s'appliquer tant dans le domaine des bits que dans celui des atomes et suppose traditionnellement une activité commerciale d'un type quelconque.

On peut ramener la définition du commerce électronique à des : [FRE 99]

**Transactions utilisant des canaux informatisés incluant le transfert de propriété ou le droit d'utilisation d'actifs tangibles ou intangibles.**

##### **Définition 2 :**

Selon l'Organisation pour la coopération économique et le développement (OCDE), "le commerce électronique désigne la conduite d'activités commerciales au moyen de technologies de pointe appartenant aux domaines des communications et de l'informatique"

Le commerce électronique (ou e-commerce) est un processus d'achat et de vente qui s'effectue sur des réseaux ouverts (l'Internet) ou fermés (réseaux privés), reliant principalement des ordinateurs.

Plus spécifiquement, le commerce électronique est une forme de commerce dans lequel les transactions financières se déroulent entièrement ou partiellement sur un réseau informatique avec un paiement et une livraison physique ou numérique [FYV 04]

## **II.2- Historique :**

Le commerce Electronique n'est pas un nouveau concert. Il a vu le jour dans les années 60 sur des réseaux privés. A l'époque, l'entreprise, fournisseurs et consommateurs ont recours à des moyens électroniques pour échange des données et conduire des transactions commerciales, l'accès au commerce électronique a été handicapé par des coûts élevés, une complexité d'installation et d'usage, des interfaces pas très engageantes (écran minitel en noire et blanc). Cette complexité était liée à la difficulté d'établissement de lignes de communication fiables et sécurisées, plus encore, à l'absence d'applications standard permettant de visualiser et de partager des informations une fois la connexion en place. [AHM 01]

De nos jours, le commerce électronique n'est plus l'apanage des grandes entreprises et des réseaux privés. L'Internet et surtout le World Wide Web, ouvrent de nouveaux horizons commerciaux aux grandes organisations, en plus d'offrir aux petites et moyennes entreprises un point d'entrée persistant.

## **II.3- Avantages du commerce électronique :**

Le commerce électronique est le moteur de la nouvelle économie, et Internet en est le principal outil. Internet est un réseau de communication qui a révolutionné la manière d'accéder aux informations, de les partager et de les utiliser.

La quantité des informations échangées et la rapidité des échanges ont considérablement augmenté. Grâce à des transferts d'informations rapides et fiables, il est possible de gagner du temps et de l'argent, ce qui a transformé les organisations en éliminant les communications sur support papier, en réduisant le coût des transactions et en diminuant le nombre de niveaux hiérarchiques.

Les avantages du commerce électronique sont les suivants: [MAL 02]

- Internet est accessible partout à bas prix.
- Le commerce électronique est accessible grâce à divers supports technologiques (ordinateurs, téléphones portables, télévisions numériques, kiosques).
- Le délai de commercialisation est réduit.
- un plus grand choix de produits.
- la commodité : le magasin est toujours ouvert dans Internet (24 heures par jour et 7 jours par semaine).
- Une baisse potentielle des prix grâce à la compression des frais généraux et à la concurrence accrue.
- Les mécanismes de paiement par carte existants peuvent être adaptés à ce type de transaction.
- un marché élargi : La solution ne connaît pas de limite géographique.
- Les intermédiaires peuvent être éliminés de la chaîne des prestations.
- Les stocks peuvent être réduits au minimum ou éliminés grâce à des procédures de production à flux tendus.

#### **II.4- Les acteurs du commerce électronique :**

- **Le client** : ce qu'il recherche, c'est un moyen de paiement simple, un accès facile à des sites commerciaux, une sécurité d'emploi du canal d'achat qui doit être un gage de confiance dans le système.
- **Le commerçant** : ce qu'il recherche, c'est un moyen de facturation qui lui garantisse d'être payé, un canal de distribution commercial, et comme le client, un canal d'achat sécurisé digne de confiance
- **La banque** : elle occupe dans ce schéma un rôle primordial dans la relation commerciale du commerce électronique. La relation de la banque avec son client s'enrichit, en particulier en développant progressivement le concept de banque à domicile, mais aussi en développant de nouveaux canaux de distribution.

### II.5- Les différents types du e-commerce :

On distingue deux types de commerce électronique, la différence entre ces deux derniers revient à la catégorie dans laquelle se trouve le client. Il peut être un particulier comme il peut être une entreprise.

#### B2B

B2B (business-to-business ou entreprise-entreprise) ce sont des relations entre entreprises pour vendre, informer etc. des produits ou services [AHM 01]

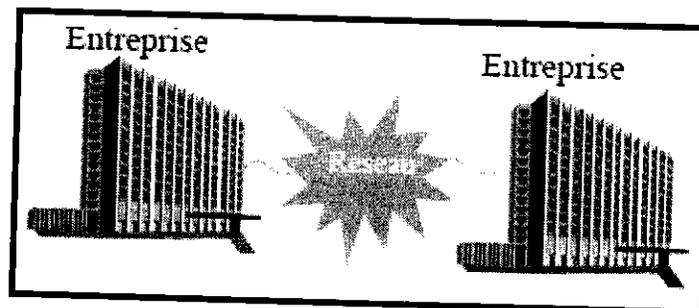
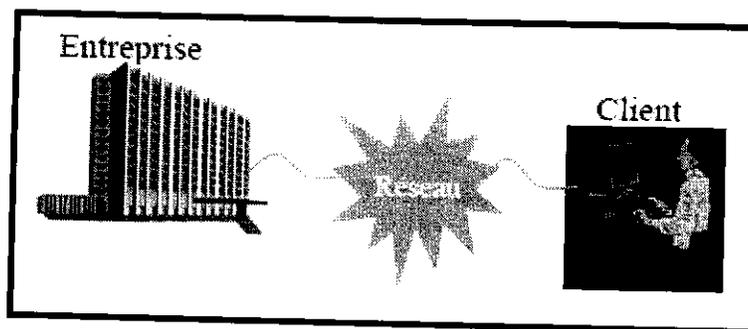


Figure 2.1: Marché Business to Business

#### B2C

B2C (business-to-consumer ou entreprise-client) c'est le type de site e-commerce qu'on va réaliser qui désigne l'échange de produits, services ou renseignements entre entreprises et clients, au sens strict concerne l'utilisation de supports électroniques pour tout ou partie des relations commerciales entre une entreprise et les particuliers: [AHM 01]

- Publicité.
- Présentation de catalogue.
- Service d'information interactif.
- Commande en ligne.
- Distribution.
- Service après-vente.



**Figure 2.2: Marché Business to Consumer**

Cependant, d'autres catégories se sont ajoutées : [AHM 01]

### **C2A**

C2A (consumer-to-administration ou consommateur-administration) offre aux consommateurs des services administratifs.

### **B2A**

B2A (Business-to-administration ou entreprise-administration) offre aux entreprises des services administratifs.

### **B2G**

B2G (business-to-government ou entreprise-gouvernement) englobe l'échange de renseignements, le fait de remplir des formules financières ou réglementaires, le paiement de taxes ou d'honoraires et toute autre transaction entre une entreprise et une entité gouvernementale (municipale, régionale, provinciale ou nationale).

### **C2C**

C2C (consumer-to-consumer ou client-client) : est un commerce entre consommateurs (ventes aux enchères).

### II.6- Les fonctionnalités d'un site e-commerce:

Quel que soit la nature ou le type de commerce qu'exerce une entreprise dès qu'elle opte pour une solution de e-commerce, cette dernière doit apporter un plus pour l'entreprise à savoir :

- la gestion et la présentation d'un **catalogue produits** structuré en un ensemble de rayons et contenant des ensembles dynamiques d'attributs liées aux différentes catégories d'articles ou services proposés.
- Une structure de **base de données** capable d'organiser et de stocker les différents types de produits et les clients.
- Un **panier d'achat** que les clients peuvent utiliser pour stocker les articles ou services sélectionnés disponible à tout instant.
- Des formulaires permettant aux clients de **passer des commandes** en introduisant leurs adresses, mot de passe et leur carte de crédit de manière sécurisée.
- Des outils de services à la clientèle permettant aux clients de visiter le site et de suivre leurs commandes.
- Un **outil de recherche** facilitant la recherche d'un produit suivant des critères variés.
- Une **communauté en ligne** qui permet de rapprocher les clients du site et leur permettant de bavarder, d'échanger des idées et de s'informer sur les nouveaux produits et services soit à travers un groupe de discussion (forum) ou par les lettres d'informations envoyées dans leurs courriers électroniques.

### III- Services de paiement électronique :

Une grande variété de moyens de paiement électronique caractérise les solutions de commerce électronique existantes. On distingue trois modes de paiements :

#### a\ Le crédit :

- Carte de crédit : C'est une carte à puce délivrée par une banque.
- Chèque Electronique : C'est un montant de crédit délivré par la banque en chèque électronique.

#### b\ Le débit :

Dans ce cas chaque client possède un compte de dépôt dans un établissement bancaire ou assimilé. Le débit se fera directement sur ce compte par le serveur et le client alimente régulièrement ce compte.

c) Le porte-monnaie électronique :

Dans ce cas chaque client possède un compte de dépôt dans un établissement bancaire: le porte monnaie est un compte parallèle à ce compte dépôt, qui convient le client avec la banque, d'un versement mensuel régulier du compte. Et à chaque fois qu'un marchand désirera se faire rétribuer, la banque débitera ce compte.

Le schéma de la **figure 2.3** illustre les fonctions essentielles des transactions de paiement par carte de crédit que l'on retrouve dans les moyens de paiement en général.

Dans les moyens de paiement électronique les plus avancés, ces fonctions se répartissent en trois modules : le module de paiement du consommateur, le module de paiement du commerçant, et la passerelle vers le système bancaire [MIC 99].

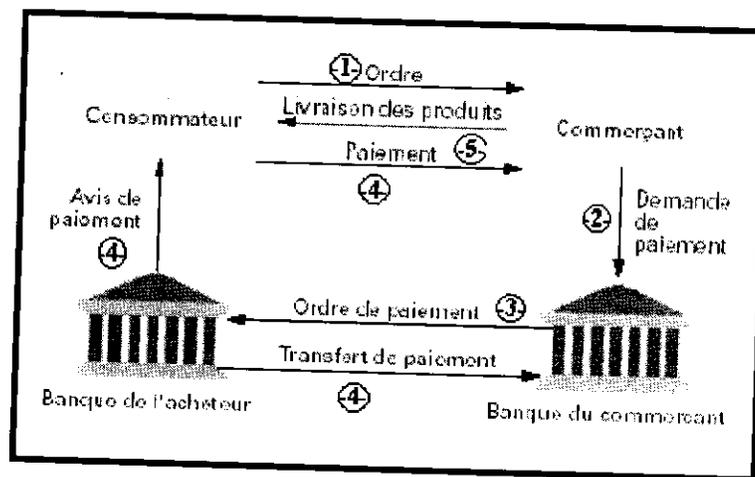


Figure 2.3 : Schéma de paiement électronique par carte de crédit.

- 1 : le consommateur effectue sa commande et remplit les formulaires du bon de commande.
- 2 : le commerçant demande de paiement par sa banque il lui fournit le numéro du carte de crédit et le nom de la banque du consommateur.
- 3 : la banque du commerçant effectue une demande de paiement au banque du consommateur.
- 4 : la banque du consommateur transfère le paiement vers la banque du commerçant et un avis de paiement vers le consommateur.
- 5 : le commerçant livre les produits.

IV- Les conditions nécessaires pour le commerce électronique :

Le e-commerce est possible que dans la confiance. Or la sécurité est un préalable nécessaire et non suffisant à celle-ci. Actuellement, bon nombre de réticences à l'égard d'Internet sont liées à l'absence, réelle ou supposée de sécurité, notamment autour du

paiement. Les experts de l'Organisation pour la coopération économique et le développement l'OCDE ont identifié les conditions minimales pour assurer celle-ci.

**Un besoin d'authentification et de certification :** acheteurs et vendeurs ont besoin d'être assurés de l'identité du partenaire, de la légitimité à vendre du vendeur et de la solvabilité de l'acheteur. C'est une information minimale au dénouement de transactions commerciales.

**Besoin de confidentialité :** la confidentialité est actuellement proposée par des solutions de cryptographie ou, ce qui résout également le problème d'authentification, par des lecteurs de carte à puce. Pour les consommateurs, la protection de la vie privée est également en cause.

**Besoin de sécurité du paiement :** plusieurs systèmes de paiement sécurisés ont été mis au point, utilisant la cryptographie ou un boîtier de lecture de carte bancaire à puce. Il reste toutefois du chemin à parcourir pour assurer leur interopérabilité et l'unification des garanties accordées. Car les systèmes techniques destinés à garantir la sécurité des paiements électroniques sont nombreux.

**Besoin de sécurité juridique :** la loi qui s'applique au contrat, les voies de recours dont les contractants disposent en cas de non exécution doivent être clairement connues. La propriété intellectuelle, la vie privée doivent pouvoir être protégées.

**La fiscalité :** la virtualité des échanges, lorsqu'il s'agit de biens numérisés transportés sur les réseaux (logiciels, information) est une source d'évasion fiscale.

**La protection des consommateurs :** puisque chaque consommateur laisse une "trace" de ses attitudes de recherche et de "flânerie" dans les galeries marchandes virtuelles, incite à porter une attention spécifique à la protection des données personnelles.

**La surveillance de la concurrence :** les problèmes de concurrence liés au développement du commerce électronique ne sont pas encore approfondis par les experts, le développement de nouvelles structures de marché étant encore embryonnaire. Mais il est certain que les réseaux de partenariat et les places de marché électroniques qui sont constitutifs du développement du commerce électronique vont poser en termes nouveaux l'analyse concurrentielle.

**V- Les risques dans le commerce électronique:**

- Le numéro de la carte peut-être lu et détourné à des fins frauduleuses, le montant de la transaction peut-être modifié, le porteur de la carte peut contester à tort un achat en vente à distance. Le premier obstacle est la confidentialité et l'intégrité des données. Il n'y a aucune garantie que les messages envoyés n'aient pas été interceptés.
- Le second obstacle est l'identification des interlocuteurs. Comme les échanges sont faits de manière électronique, comment acheteurs et commerçants peuvent-ils s'identifier à distance, quand chacun est représenté par des caractères numériques sur un écran ou une disquette.

Le chiffrement des transactions garantit la confidentialité des informations comme le fait le standard SSL (Secure Socket Layer). Mais de nombreuses compagnies se sont associées pour développer de nouveaux systèmes de sécurité [YLE 97] [DBM 00] adaptés aux cartes à puces, comme le protocole SET (Secure Electronic Transaction) d'origine américaine et le protocole français C-SET (Chip-Secured Electronic Transaction).

**VI- Communications sécurisées :**

Lors des transactions commerciales via Internet les utilisateurs doivent fournir des informations personnelles et confidentielles, la saisie du numéro de carte de crédit du client doit s'effectuer de manière sécurisée, c'est pourquoi les clients s'attendent à ce que nous prenions les mesures nécessaires pour que ces informations restent confidentielles.

Pour sécuriser le site e-commerce, nous devons réfléchir à la configuration de notre système d'exploitation, de notre logiciel serveur web, et notre base de données.

**VI.1- Cryptographie :**

La cryptographie est à la base des mécanismes de confidentialité, d'intégrité et d'authentification, elle consiste à crypter tous types d'informations de manière qu'elles soient incompréhensibles aux personnes non concernées. [JER 03]

L'un des astuces de cryptographie consiste à chiffrer les données lors de leur transmission, par exemple au lieu d'obtenir un numéro de carte de crédit sous la forme 1111 1144 2242, il apparaîtra sous la forme 9857 4568 2598. La technologie utilisée pour sécuriser un serveur web utilise une technique de chiffrement appelée cryptographie par clé publique.

**VI.2- Fonctionnement de la cryptographie par clé publique :**

Voici une explication : prenez deux nombre premiers énormes, et multipliez-les pour obtenir un nombre encore plus impressionnant. Prenez ensuite ce nombre gigantesque puis effectuer une opération mathématique sur le message pour obtenir un message codé. Pour décoder ce message, il suffit d'effectuer l'opération mathématique inverse.

Pour parvenir le client la clé de chiffrement du message sans que personne ne l'intercepte lors de sa transmission il suffit à utiliser deux nombres et à générer un nombre gigantesque à partir de ceux-ci, l'un des deux nombres est dit clé privée. Le deuxième est disponible pour tout le monde, il s'agit de la clé publique. Combinées les deux clés s'appellent paire de **clés privée/publique**.

**VI.3- Certificats :**

Un certificat est un ensemble de données utilisé par le navigateur web pour identifier le serveur qui gère la requête HTTP.

Lorsque nous connectons à un serveur web sécurisé, il nous envoie un certificat. Une clé publique peut être extrait de ce certificat, puis utilisée dans le processeur de chiffrement, ce qui nous permet de communiquer au serveur web des informations sensibles comme le numéro de carte de crédit du client.

Les connexions établies avec un serveur web sécurisé peuvent être identifiées grâce à la mention HTTPS qui apparaît dans la chaîne d'URL, contrairement à l'habituel http (le <S> représentant le terme sécurisé).

**VI.4- Les protocoles de sécurité :**

Il y a plusieurs protocoles de sécurité, nous citons: [IPS 04]

**a\ SSH :**

SSH (Secure Shell) est un protocole de sécurité qui vous permet d'établir une connexion sécurisée à un serveur sur lequel les protocoles SSH et SFTP (Secure File Transfer Protocol) sont installés.

SSH crypte toutes les communications entre client et serveur. Quand une connexion SSH est établie, SFTP est le protocole utilisé pour toutes les tâches réalisées par le biais de cette connexion sécurisée.

**b\ PGP :**

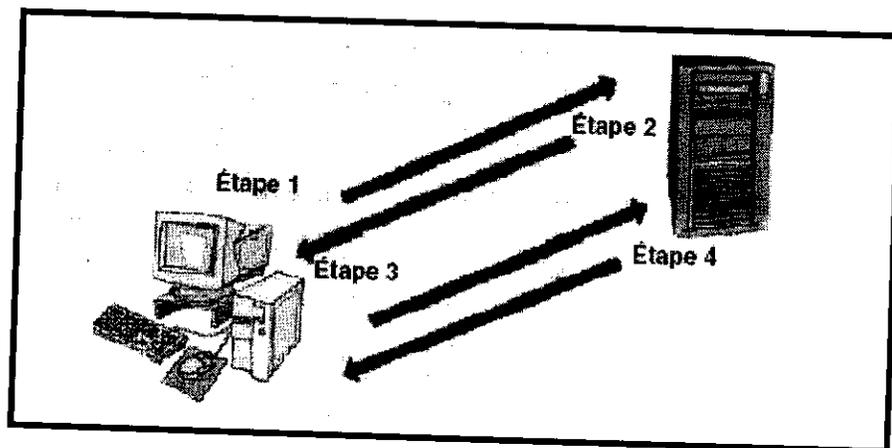
PGP (Pretty Good Privacy) est une méthode de cryptage de fichiers à base de clé. Grâce à cette méthode, seul le destinataire peut recevoir et décrypter les fichiers. PGP permet de sécuriser les communications par messagerie électronique, mais sa technologie est également applicable à FTP.

PGP utilise deux clés cryptographiques pour sécuriser les fichiers. Une clé publique sert à crypter le fichier et seule la clé privée correspondante peut le décrypter.

**c\ SSL : c'est le protocole utilisée dans notre travail**

SSL est un protocole de cryptage/décryptage des données transmises par connexion Internet directe. Quand un client établit une connexion SSL avec un serveur, toutes les données échangées avec ce serveur sont codées à l'aide d'un algorithme mathématique complexe qui rend difficile le décodage des données interceptées.

Voici une illustration étape par étape du fonctionnement de SSL.



**Figure 2.4 : Fonctionnement de SSL**

**Étape 1 :** Le client établit la connexion initiale avec le serveur et demande l'établissement d'une connexion SSL. Si SSL implicite est utilisé, la connexion initiale est cryptée. Si SSL explicite est utilisé, le contact initial n'est pas crypté.

**Étape 2 :** Si sa configuration est correcte, le serveur envoie au client son certificat et sa clé publique.

**Étape 3 :** Le client compare le certificat du serveur à une base de données d'autorités approuvées. Si le certificat figure dans cette liste, le client fait confiance au serveur et passe à l'étape 4. Si le certificat est absent de cette liste, l'utilisateur doit l'ajouter à la base de données d'autorités approuvées avant de passer à l'étape 4.

**Étape 4** : Le client utilise cette clé publique pour crypter une clé de session qu'il envoie au serveur. Si le serveur demande le certificat du client à l'étape 2, celui-ci doit alors l'envoyer.

**Étape 5** : Si le serveur est configuré pour la réception de certificats, il compare celui reçu à ceux figurant dans la liste de sa base de données d'autorités approuvées et accepte ou rejette la connexion.

Si la connexion est rejetée, un message d'erreur est envoyé au client. Si la connexion est acceptée, ou si le serveur n'est pas configuré pour la réception de certificats, il décode la clé de session du client avec sa propre clé privée et renvoie un message de réussite au client, en ouvrant par la même occasion un canal sécurisé pour les données.

### **VII. Conclusion :**

Dans ce deuxième chapitre nous avons vu de pré les différents aspects du commerce électronique en présentant ses différents types, ses fonctionnalités, ses avantages et ses risques, ainsi que la partie sécurité. Dans le but de bien préparer les différents modules de notre application.

# Chapitre III

## Chapitre III : Applications Web : Architecture et langages de programmation

### I. Introduction :

Les technologies mises en œuvre dans les projets de e-commerce s'appuient sur les standards Internet, c'est-à-dire sur l'utilisation d'une interface Web, accessible à partir d'un navigateur, permettant d'accéder à des services proposés par l'entreprise. Le Web est basé sur un mode de fonctionnement dit Client/Serveur ou l'utilisateur interagit avec un service par l'intermédiaire de requêtes.

Grâce à ce type de mécanisme, les personnels, les clients ou les fournisseurs d'une entreprise peuvent avoir accès à une sélection de services fournis par l'entreprise par l'intermédiaire d'un serveur et dispose instantanément d'une information ou échanger des informations.

### II.1- ARCHITECTURE DES APPLICATIONS WEB :

#### II.1.1- Application Web :

Une application Web est une application mettant en œuvre des fonctionnalités métier à l'aide d'un navigateur Web, c'est-à-dire qu'une application Web est un site Web dont l'état applicatif est affecté par les actions de l'utilisateur.

L'architecture d'une application Web est basée sur le modèle client/serveur. Elle contient quatre composants principaux à savoir : le serveur Web, les serveurs d'application, une connexion réseau et des navigateurs clients. [FRA 06]

#### II.1.2- Architecture client-serveur :

Le terme serveur s'applique à tout programme qui offre un service que l'on peut atteindre à travers le réseau. Le serveur accepte des demandes issues du réseau, il les traite et il renvoie le résultat au demandeur.

Un programme qui s'exécute devient un client lorsqu'il émet une demande vers un serveur et il attend une réponse.

Une application est bâtie selon une architecture client-serveur lorsqu'elle est composée de deux programmes, coopérant l'un avec l'autre à la réalisation d'un même traitement. La première partie, appelée module client, est installée sur le poste de travail alors que la seconde, appelée module serveur, est implantée sur l'ordinateur (ou même des ordinateurs éventuellement situés dans des lieux géographiquement différents) chargé de rendre le service.

a\ **Fonctionnement :**

Une architecture client-serveur web fonctionne comme suit : [DAV 00]

- Un logiciel serveur met à disposition des informations.
- Un logiciel client (navigateur) se connecte à ce serveur via un réseau pour demander à recevoir ces informations en émettant une requête http. Le serveur répond au client en émettant une réponse http.
- Un réseau réalise la communication entre le client et le serveur.

Le schéma de la figure 3.1 illustre son fonctionnement

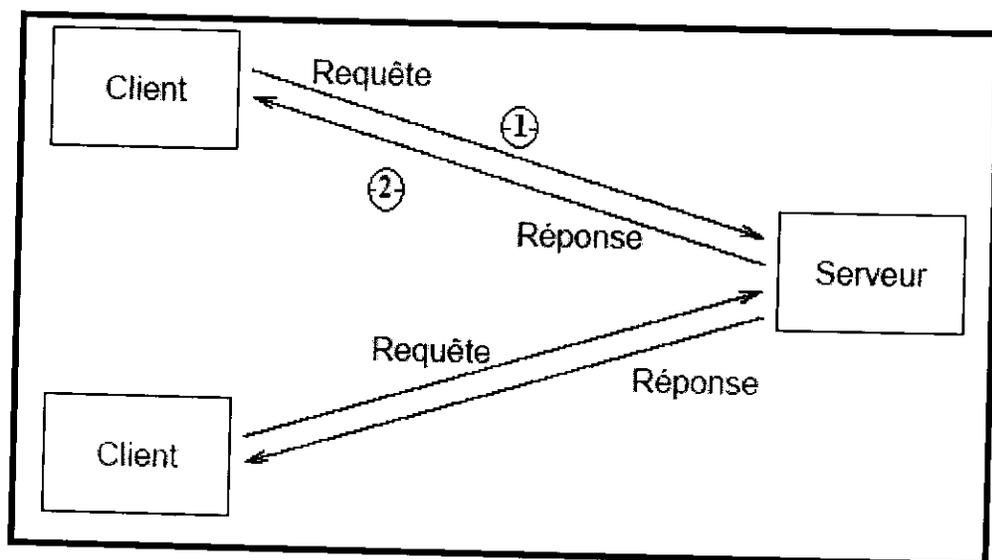


Figure 3.1 : fonctionnement d'un système client/ serveur

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

b\ **Caractéristiques :**

Un système client/serveur partage les caractéristiques spécifiques suivantes [ROB 99] :

- **Service** : le modèle client/serveur est essentiellement une relation entre les processus tournant sur des machines séparées. Le processus serveur est un fournisseur de services tandis que le processus client est un consommateur de services.
- **Partage des ressources** : un serveur peut traiter plusieurs requêtes client en même temps et contrôler leurs accès aux ressources.

- **Intégrité** : le code et les données du serveur sont gérés de façon centralisée, ce qui garantit un moindre coût de maintenance et une meilleure intégrité des données partagées.
- **Échange de messages** : clients et serveurs sont des systèmes qui interagissent au moyen de messages. Le message est un mécanisme d'émission des demandes de service et des réponses à celles-ci.
- **Transparence à la localisation** : les processus client et serveur peuvent résider sur la même machine ou, par l'intermédiaire d'un réseau, sur deux machines différentes interconnectées. Le logiciel client/serveur masque aux clients la localisation du serveur en redirigeant les demandes de service si nécessaire. Un programme peut être client, serveur ou les deux.

II.1.3- Les différents types d'architecture Client/Serveur : [JFP 06]

II.1.3.1- Présentation de l'architecture à 2 niveaux :

L'architecture à deux niveaux (aussi appelée architecture 2-tiers, tiers signifie étage en anglais) caractérise les systèmes clients/serveurs dans lesquels le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service (voir figure 3.2)

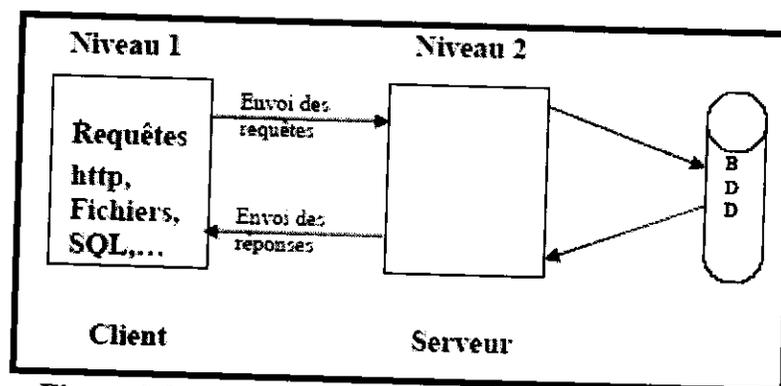


Figure 3.2 : Architecture Client/Serveur à 2 niveaux

II.1.3.2- Présentation de l'architecture à 3 niveaux :

Dans l'architecture à 3 niveaux (appelée aussi architecture 3-tiers), il existe un niveau intermédiaire (voir figure 3.3), c'est-à-dire que l'on a généralement une architecture partagée entre :

- **Le client** : c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur Web) chargée de la présentation.

- Le serveur d'application (appelé aussi middleware) : le serveur chargé de fournir la ressource mais faisant appel à un autre serveur qu'est le serveur de la base de données.
- Le serveur secondaire : (généralement un serveur de base de données), fournissant au serveur d'application les données dont il a besoin.

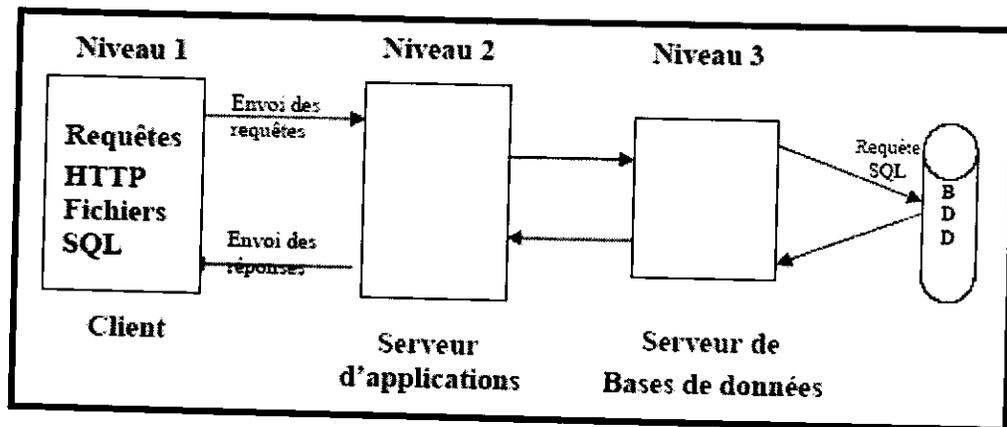


Figure 3.3 : Architecture Client/Serveur à 3 niveaux

- Comparaison des deux types d'architecture :

L'architecture à 3 niveaux permet de fournir par rapport à l'architecture à 2 niveaux :

- Une plus grande flexibilité/souplesse.
- Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau.
- De meilleures performances, étant donné le partage des tâches entre les différents serveurs.

II.1.3.3- Présentation de l'architecture multi niveaux :

Dans l'architecture à 3 niveaux, chaque serveur (niveaux 1 et 2) 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.

Le schéma (voir **figure 3.4**) suivant illustre la configuration multi niveaux :

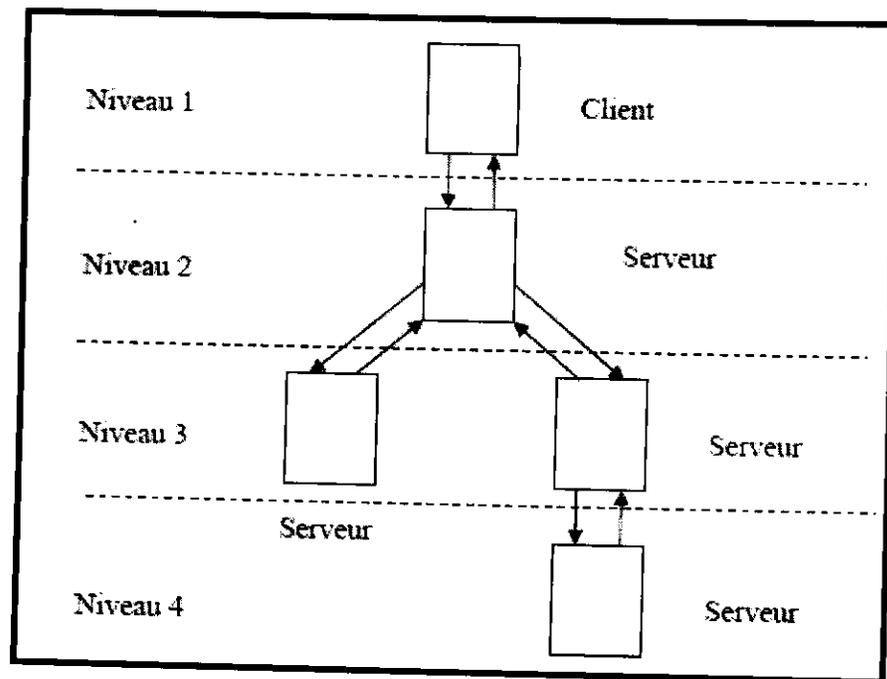


Figure 3.4 : Architecture Client/Serveur multi niveaux

#### II.1.4- Avantages et inconvénients de l'architecture client/serveur :

L'architecture client/serveur est particulièrement recommandée pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont [FMC 02] :

- **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 cette architecture, 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 modifications majeures.

L'architecture client/serveur a quelques inconvénients parmi lesquels :

- **Un coût élevé** : dû à la technicité du serveur (un serveur doit être puissant et rapide, afin de pouvoir gérer le plus rapidement possible les requêtes d'un grand nombre de processus)
- **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.

## II.2- LES LANGAGES DE PROGRAMMATION (SCRIPTS):

Le serveur Web reconnaît les fichiers à leurs extensions différentes de celles des pages *HTML* statiques, si le document appelé par le client comporte des ordres de programmation (scripts), alors le serveur Web lance l'interpréteur des scripts, qui traduit le document demandé et exécute le code source de la page. Les commandes figurant dans la page sont interprétées et le résultat prend la forme d'une page *HTML* publiée à la place du code source dans le même document.

Il faut préciser qu'il existe deux types de scripts un est nommé langage côté client et l'autre côté serveur.

### II.2.1- Langages coté serveur :

Les langages côté serveur sont indépendants du client (le navigateur): tout le travail d'interprétation du programme est réalisé par le serveur. Ils sont donc compatibles avec tous les navigateurs.

### II.2.2- Langages coté client

Les langages côté client sont indépendants du serveur. Même si un hébergeur n'autorise pas l'utilisation d'un langage serveur, il vous est toujours possible de faire appel à un langage côté client tel Java Script.

Ces langages sont qualifiés de côté client car ils sont exécutés sur la machine de l'internaute, pas sur le serveur web. Ils n'ont donc pas accès aux données de ce dernier.

Les principales différences entre les scripts côté client et les scripts côté serveur sont:

- Le navigateur exécute les scripts côté client après avoir chargé complètement la page.
- Le code des scripts côté client fait partie de la page et peut donc être lu par l'utilisateur (par exemple via la commande Affichage/Code Source)
- Le code côté client doit être pris en charge par le navigateur, et les différences entre navigateurs rendent parfois difficiles l'écriture des scripts.
- Le script côté serveur est exécuté par le serveur avant d'être transmis au navigateur.
- Le code d'un script côté serveur n'est pas transmis au navigateur, et il n'est donc pas lisible par l'utilisateur.
- Le script côté serveur est indépendant des possibilités du navigateur mais impose un équipement adéquat sur le serveur web.

Les scripts exécutés côté client et côté serveur peuvent être mélangés (on peut mettre du code *ASP* et *Javascript* dans la même page).

#### Exemples :

##### • Langage Côté client (Javascript) :

```
<Html>
<Head>
<Title>Exemple de Java script</Title>
</Head>
<Body>
<BR>
<P align="center"> Exemple d'un message d'alerte :</p>
<script language="JavaScript">
Alert ("Bienvenue sur le site de commerce électronique www.citroën-Meklati.dz");
</script>
</Body>
```

Ce script affiche un message d'alerte.

- **Langage Côté serveur (ASP) :**

```
<%@ LANGUAGE="VBSCRIPT" %>
<html>
<Head>
<Title>Exemple de script ASP</Title>
</head>
<Body>
<% Response. Write (FormatDateTime (Now (), 2) & " «) %>
</Body></html>
```

Ce script a pour effet l'affichage de la date système serveur.

### II.2.3- Les Scripts utilisés :

Les Scripts utilisables pour la réalisation des sites Web sont :

- **Java** : c'est un langage inventé par la société Sun, son principal avantage est d'être indépendant du système d'exploitation. On peut théoriquement, réaliser un programme *Java* fonctionnant sur un serveur sous Unix et le transférer sans modification sur un serveur *Windows NT*. L'une des particularités est de pouvoir fonctionner côté client ou côté serveur. De nombreux sites ont recours à des modules Java, des « applets » récupérées à droite et à gauche.
- **Java Script** : A ne pas confondre avec *Java*, *JavaScript* a été mis au point par Netscape, Si *JavaScript* peut fonctionner du côté serveur, il est généralement utilisé côté client. Les instructions sont directement insérées dans les pages *HTML* et traitées par le navigateur. C'est un langage très largement employé sur Internet.
- **VBScript** : développé par *Microsoft*, *VBScript* est un langage généralement utilisé côté serveur. Microsoft a ainsi voulu concurrencer *JavaScript* en tablant sur le succès de *Visual Basic* dont est dérivé *VBScript*. Lorsqu'on utilise *VBScript* côté serveur, la compatibilité est évidemment assurée avec tous les navigateurs, comme pour tout langage côté serveur.

- **PERL** : reste une référence sur l'Internet. Rapide, puissant, c'est un langage universel. Il est toutefois plus complexe à mettre en oeuvre que l'ASP et le PHP. De plus, avec ce langage, on est obligé de gérer la page entière par programmation.
- **PHP (personnel hypertexte page)**: est un langage incrusté au HTML et interprété ou compilé côté serveur. Il dérive du C et du Perl dont il reprend la syntaxe. Comme il supporte tous les standards Web et qu'il est gratuit, il s'est rapidement répandu sur le Web.
- **ASP (Active Server Pages)** : c'est avec ce langage que nous réalisons notre application, C'est un langage signé Microsoft, très proche de Visuel Basic, l'ASP est souvent associé aux serveurs Microsoft (IIS, Internet Information Server).

#### II.2.4- ASP : définitions et fonctionnalités :

##### II.2.4.1- Définition : [JEN 05]

ASP constitue l'abréviation de Active Server Page qui est une technologie développée par Microsoft. Il s'agit de scripts qui s'exécutent sur le serveur Web, suite au chargement d'une page HTML. Les scripts ASP remplissent une fonction similaire aux autres technologies de scripts côté serveur comme Perl, PHP et autres.

Les applications ASP sont des programmes conçus à partir de plusieurs sources :

- du Visual Basic Script (VBScript), ou du JScript (le Javascript de Microsoft)
- des composants et objets ASP.

##### II.2.4.2- Plateforme requise :

Le système d'exploitation de l'ASP est évidemment Microsoft Windows (Windows 95, 98, Millenium, XP, 2000, Serveur ...etc.)

Ensuite, un serveur Internet spécifique à chaque système d'exploitation doit être installé et configuré proprement.

- *PWS (Personal Web Server)* sur *Windows 95, 98 et Millennium*
- *IIS 4.0 (Internet Information Server)* sur *NT 4.0 Server*,
- *IIS 5.0* sur *NT 2000 Server et XP*.

#### II.2.4.3- Fonctionnement de l'ASP :

On insère directement du code (*VBScript* ou *JScript*) dans la page *HTML*, à l'endroit où une opération spécifique doit être effectuée.

Lorsqu'un navigateur demande cette page *ASP*, le serveur interprète et exécute le code qui a été inséré et renvoie le résultat au client sous forme de page *HTML* standard.

Il existe deux façons d'insérer du code *ASP* dans les pages *HTML*. On peut utiliser deux types de balises de délimitation :

Dans le fichier *HTML*, les scripts se distinguent des balises *HTML* en employant des délimiteurs différents: Ainsi, les commandes *ASP* utilisent les délimiteurs `<%` et `%>`.

Le deuxième exemple utilisant la balise `<script>` permet de préciser le langage de programmation à utiliser pour le script *ASP* de la page. Il est alors indispensable de refermer cette balise `</script>` après avoir terminé les ordres de script.

L'exemple suivant illustre ces deux méthodes, en utilisant des commandes *VBScript*.

```
<%
code dans le langage par défaut (certainement VBScript)
%>
<%@ language="VBScript" %>
<%
Code VBScript
%>
<SCRIPT language="JavaScript" runat="Server">
Code JavaScript
</SCRIPT>
```

#### II.2.4.4- Le modèle objet ASP : [AHM 01]

*ASP* utilise différents objets pour décrire les interactions client/serveur. Ces objets en *ASP* sont au nombre de six (voir **figure 3.5**) (pour plus de détails sur ces objets, voir l'Annexe B) :

- l'objet *Request*
- l'objet *Response*
- l'objet *Application*

- l'objet *Session*
- l'objet *Server*
- l'objet *ASPError*

Ces objets peuvent être constitués de trois éléments : des *propriétés*, des *méthodes* et des *collections*. Une collection est un tableau associatif. ASP va par exemple utiliser une collection spéciale de l'objet *Request* (*Request.Form*) pour rassembler le contenu d'un formulaire par champ. On accède à ces trois éléments en utilisant la notation objet en vigueur dans le langage qu'on utilise pour écrire le code ASP de la page (généralement *objet.element*).

Ces différents objets ont différents domaines d'intervention. Cela peut se représenter de la façon suivante :

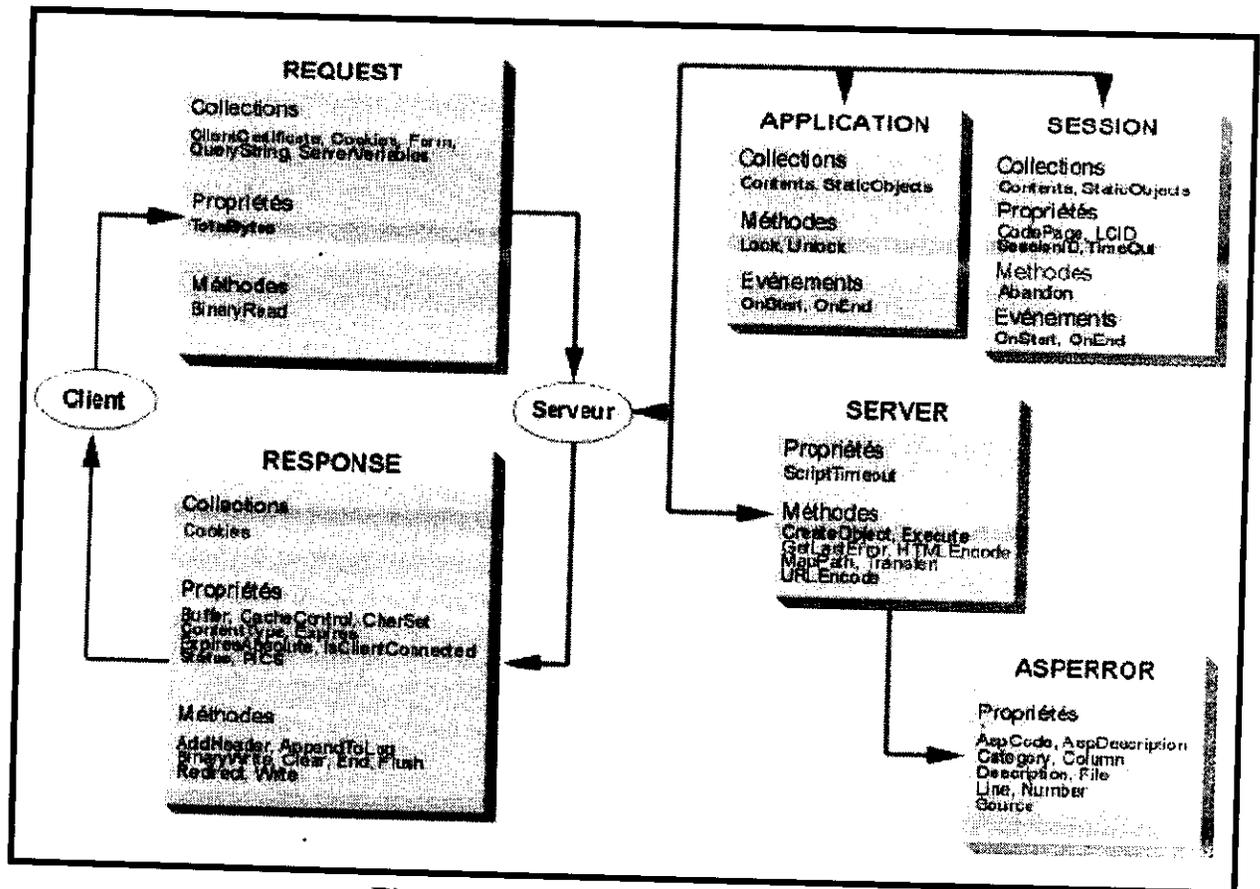


Figure 3.5 : Le modèle objet de l'ASP

### II.2.5- Variables, constantes et types de données :

#### II.2.5.1- Constantes :

Une constante peut avoir le même type qu'une variable, seule la déclaration est différente. Pour définir une constante, on fait appel à l'instruction *Const*.

**Exemple :**

```
Const site
```

```
Site = www.citroën-Meklati.dz
```

#### II.2.5.2- Utilisation des variables :

En *ASP*, les variables appartiennent toujours à un type de données fondamental : *Variant*.

Un *Variant* est un type de données spécial qui peut contenir différents types d'information, en fonction de son utilisation. Parce que le *Variant* est le seul type de données de *VBScript*, c'est aussi le type de données retourné par toutes les fonctions de *VBScript*.

On déclare des variables explicitement dans un script par l'intermédiaire des instructions *Dim*, *Public* et *Private* et l'affectation est réalisée à l'aide de l'opérateur (=).

Par exemple :

```
Dim prénom = "Billel"
```

### II.2.6- ASP et le service de messagerie (Mail) :

*ASP* étant un langage consacré au Web, et le service le plus utilisé sur Internet étant la messagerie électronique, donc il est naturel que *ASP* permette d'envoyer des e-mails.

En fait, ce service de messagerie peut être supporté par de nombreux protocoles : le plus couramment utilisé étant ***SMTP*** (*Simple Mail Transfert Protocol*).

En *ASP*, l'envoi d'e-mails s'effectue par le biais de l'objet *ASPmail*. Ce composant permet à partir de scripts présents dans des pages *ASP*, d'envoyer des e-mails de manière simple.

En premier lieu, il faut créer l'objet *ASPmail*, attribuer les propriétés nécessaires au fonctionnement de cet objet et envoyer le mail par la méthode d'envoi d'*ASPmail*.

La création de l'objet *ASPmail* se fait toujours par cette ligne typique de déclaration d'objet `<% Set nom_de_l'objet = Server.CreateObject ("SMTPsvg.Mailer") %>`.

### II.3- ASP ET L'INTERFAÇAGE AVEC LES BASES DE DONNEES :

Pour communiquer avec une base de données à travers une page *ASP*, il faut utiliser un composant *ADO* (*ActiveX Data Objects*). Il fournit un certain nombre d'objets susceptibles de se connecter à une base de données et de l'interroger.

#### II.3.1- ADO (ActiveX Data Objects):

Le langage *ASP* possède plusieurs composants *Active Server* implémentés par des fichiers *DLL* (*Dynamic link Library*), bibliothèque de liaison dynamique, localisés sur le serveur. En résumé, ce sont des *ActiveX* dont les méthodes et propriétés sont stockées dans une *DLL*. Les composants serveur effectuent diverses tâches utiles pour le fonctionnement dynamique des pages Web.

*ADO* est un modèle objet d'accès aux données. Il permet d'adresser des données qui proviennent de bases de données relationnelles (*SQL Server, Oracle...*), ou d'autres sources de données non relationnelles telles que des fichiers texte, des sources de données non Microsoft... C'est un moyen d'accéder de manière unique à tout type de données.

Cette *API* (*application programming interface*) passe obligatoirement par *OLE DB* (*Object Linking and Embedded DataBase*) qui constitue une alternative récente de *ODBC* (*Open DataBase Connectivity*).

#### II.3.2- L'API ODBC.:

*ODBC* est une *API* qui permet à des applications d'accéder aux données d'un large éventail de systèmes de gestion de bases de données relationnelles.

*ODBC* repose sur une architecture constituée de *DLL*. Au sommet, on trouve « *ODBC32.DLL* » qui définit l'*API* elle-même et appelle des *DLL* spécifiques à la base de données connue sous le nom de pilote (driver). A l'aide de la base de registres, elle détecte les pilotes disponibles et choisit le pilote approprié.

#### II.3.3- L'API OLE DB:

*OLE DB* est une technologie qui a pour but de résoudre certaines contraintes d'*ODBC*. Cette technologie autorise l'accès à tout type de données, permet de gérer l'aspect distribué des sources de données. Cette technologie a pour but à terme de remplacer la technologie *ODBC*.

Les fournisseurs *OLE DB* (voir **figure 3.6**) fournissent non seulement un mécanisme d'accès aux données relationnelles, mais également aux données non relationnelles.

Voici quelques fournisseurs des bases de données relationnelles :

- Le Provider *OLE DB* pour *ODBC* aussi nommé *KAGERA* ou *MSDASQL*.
- Le Provider *OLE DB* pour *SQL Server*, nommé *SQLOLEDB*
- Le Provider *OLE DB* pour *Oracle* nommé *MSDAORA*

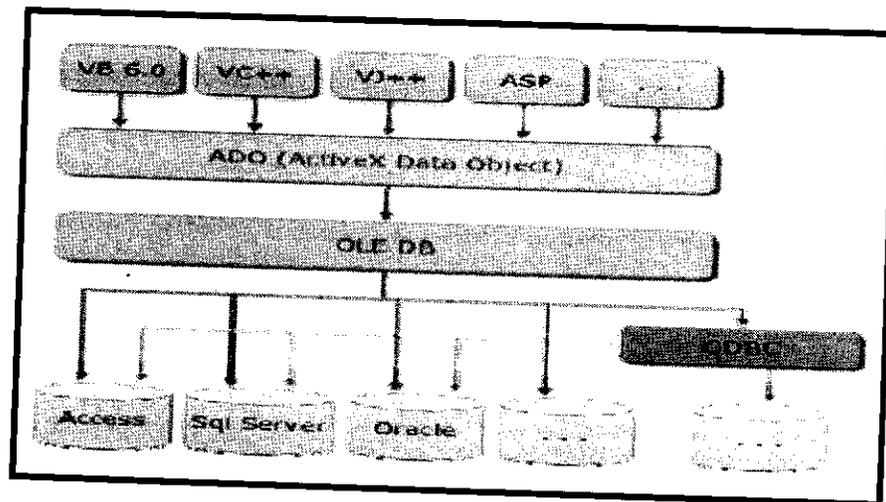


Figure 3.6 : Les fournisseurs d'accès *OLEDB*.

#### II.3.4- Le modèle objet ADO :

Le schéma de la figure 3.7 montre les objets et les relations existantes entre les objets de ce modèle.

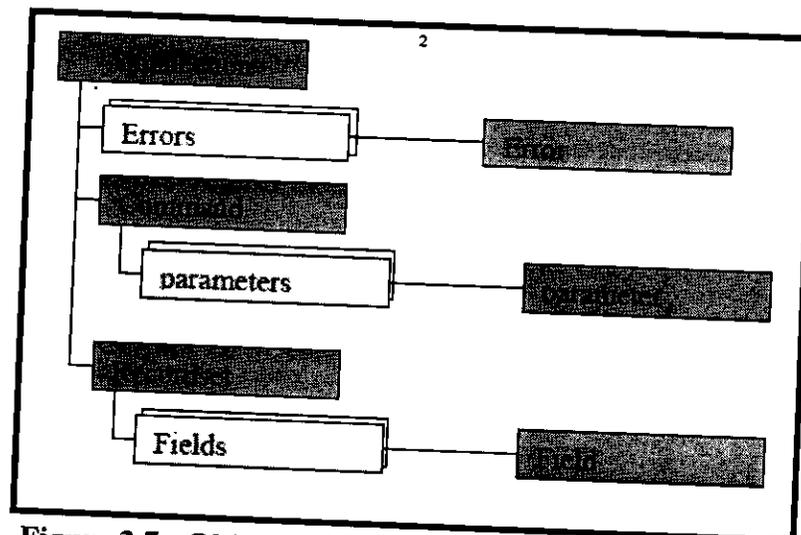


Figure 3.7 : Objets et Collections du modèle objet ADO.

Les objets *Connection*, *Recordset* et *Command* sont les objets les plus significatifs de ce modèle.

*ADO* est un modèle objet qui propose une grande flexibilité au programmeur. Il y a donc souvent différentes possibilités pour réaliser la même tâche. Par exemple, pour exécuter une requête, il est possible d'utiliser la méthode *Execute* de l'objet *Command* ou bien de l'objet *Connection*. Le seul réel lien entre *ADO* et la base est ce qu'on appelle la chaîne de connexion. Cette chaîne permet simplement de spécifier le pilote que *OLEDB* se devra d'instancier. Ainsi via *OLE DB*, on accède à la base de données souhaitée.

Le principe est toujours de créer l'objet *ADO*, de le traiter et de le détruire. Les informations sur la base doivent être transmises à l'objet *ADO* à l'aide d'une de ses deux méthodes :

#### II.3.4.1- Le Système DSN (DataSource Name):

Très répandu, il permet une connexion en utilisant un "nom virtuel" défini sur le serveur. Il s'agit en fait d'un "pont" vers le serveur de la base de données.

#### II.3.4.2- Le DSN-Less (Sans DSN) :

C'est la meilleure solution mais qui demande une bonne connaissance des chaînes de connexion puisque on n'a pas à déclarer la base de données dans le *DSN* du serveur. C'est à dire qu'on doit envoyer nous-même ce que le *DSN* envoie. Chaque base de données a son propre "Driver ODBC" ou "Provider OLEDB" donc sa propre chaîne de connexion.(Des exemples de chaînes de connexion sont cités dans l'annexe B).

L'exemple ci-dessous illustre l'utilisation des deux méthodes citées auparavant :

```
<%
Dim objConnexion, ChaineConn
' Créer l'objet ADO
Set objConnexion = Server.CreateObject ("ADODB.Connection")

' Ouvrir la base avec DSN
ChaineConn="DSN = le nom de la source de données;uid=usager;pwd=motpasse"

' ou ouvrir la base avec une chaîne de connexion OLEDB pour SQL Server
chaineConn = "provider=SQLOLEDB.1; Initial Catalog=NomBaseDeDonnées ;
DataSource=nom du serveur BDD; User ID=nom d'utilisateur ; Password=mot de
passe;"

objConnexion.open ChaineConn
' Traitement...
```

```
' Fermer la connexion  
objConnexion.close  
' Détruire l'objet  
set objConnexion=nothing  
%>
```

## II.4- ASP ET LA COMMUNICATION CLIENT-SERVEUR :

### II.4.1- Les formulaires :

Le point de départ de la programmation de pages Web interactives est constitué par les formulaires HTML, ces derniers constituent une solution intéressante pour transférer des données du client vers le serveur http.

La structure générale d'un script de traitement de formulaire doit:

- Implémenter le formulaire lui-même.
- Procéder à des contrôles de saisie.
- Effectuer des actions suivant que la saisie est validée ou non.

ASP permet de récupérer, au niveau du serveur Web, les données saisies sur un formulaire HTML. Quand vous soumettez un formulaire HTML, deux méthodes de postages des informations sont utilisables.

#### II.4.1.1- La méthode GET :

Pour la méthode GET, les données sont concaténées à la suite de l'URL demandée. Les données concaténées suivront le modèle suivant :

$$\text{variable1=valeur1\&variable2=valeur2\&.....variableN = valeurN.}$$

La collection *QueryString* de l'objet *Request* permet de pouvoir récupérer des données concaténées à la suite de l'URL demandée (soit postées par GET, soit un lien cliqué contenait des données supplémentaires).

#### II.4.1.2- La méthode POST :

Cette méthode garantit que les données ne seront pas stockées dans l'URL. Pour récupérer les données soumises via la méthode *POST*, on utilise la collection *Form* de l'objet *Request*.

L'exemple suivant montre ce que pourrait être le code *HTML* d'un formulaire de saisie d'un login et un mot de passe.

```
<CENTER> <H1>Veuillez vous identifier</H1> <BR><BR>
  <FORM Action="identification.asp" Method="GET">
  <FORM Action="identification.asp" Method="POST">
  Login : <INPUT Name="Login" Type="text"> <BR><BR>
  Password : <INPUT Name="Password" Type="password" <BR><BR>
            <INPUT Type="Submit" Value="Se connecter">
  </FORM>
</CENTER>
```

Le code suivant montre comment récupérer les données postées par *GET* et *POST* du formulaire au dessus.

```
<%@Language="VBScript" %>
<%
  DIM login, mot_passe
  ' Récupération des données postées par GET
  login = Request.QueryString ("Login")
  mot_passe = Request.QueryString ("Password")
  ' Récupération des données postées par POST
  login = Request.Form ("Login")
  mot_passe = Request.Form ("Password")
%>
```

#### II.4.2- Les variables d'environnements :

Une variable d'environnement est une valeur en lecture seule définie sur le serveur qui permet de récupérer certaines informations utiles, comme par exemple l'adresse *IP* du visiteur, la version du serveur, le type du navigateur utilisé. Pour utiliser une variable d'environnement, on utilise la méthode *ServerVariables* disponible pour l'objet *Request*

Syntaxe : *Request.ServerVariables ("VARIABLE\_HTTP")*

Diverses informations peuvent être récupérées par l'objet *Request*, associé à sa collection *ServerVariables* pointant des variables d'environnement *HTTP*.

Les différentes données ainsi extraites, permettent de mettre en place des mécanismes particuliers, comme l'adaptation d'une application *ASP* à la configuration de l'ordinateur client ou à d'autres paramètres.

**Exemple :** L'exécution d'un code en fonction du navigateur du visiteur.

```
<%  
    If InStr (Request.ServerVariables ("HTTP_USER_AGENT"), "MSIE6.0") > 0 Then  
        'Code pour internet Explorer 6  
    Else  
        'Code pour les autres navigateurs  
    End If  
%>
```

## II.5- ASP ET LA GESTION DES TRANSACTIONS :

Une transaction correspond à un ensemble d'instructions *SQL* devant s'exécuter en un tout unique: si l'on réalise un virement d'un compte bancaire à un autre, il faut débiter le premier, d'une certaine somme, pour en créditer le second. Si l'une de ces deux étapes échoue, il faut annuler la transaction bancaire.

Les transactions permettent de sortir de ce genre de problématique. Si une transaction est initiée, les modifications faites par les instructions *SQL* de mise à jour ne sont pas définitivement appliquées sur la base. Pour appliquer définitivement les modifications, il faudra valider la transaction : selon le *SGBDR* utilisé, il faudra utiliser les ordres *SQL Commit (Oracle)* ou *CommitTrans (SQL Server)*.

Si on cherche, au contraire, à annuler la transaction, il faudra utiliser les ordres *SQL Rollback (Oracle)* ou *RollbackTrans (SQL Server)*, toujours selon le *SGBDR* utilisé.

## II.6- CONCLUSION :

Dans ce troisième chapitre on a donné un aperçu général sur l'architecture des sites Web dynamiques, l'architecture client/serveur à deux et trois niveaux (respectivement 2 et 3-tiers) et enfin les langages de programmation utilisés pour développer les sites Web dynamiques. Ensuite dans le prochain chapitre on traite la démarche que nous avons suivie pour concevoir notre site e-commerce, la conception de la base de données utilisée pour stocker la masse d'informations de notre application ainsi que la conception de l'application (architecture, fonctionnalités,...etc).

# Chapitre IV

## **Chapitre IV : La Conception**

### **I- Introduction :**

Nous rappelons que l'objectif de notre travail est la réalisation d'un site e-commerce pour le compte du concessionnaire des voitures Meklati (Citroën), dans les trois premiers chapitres nous avons donné une introduction générale sur le réseau Internet à savoir le web, le commerce électronique et l'architecture des sites Web dynamiques.

Le présent chapitre sera consacré à l'analyse qui aura pour but la définition des besoins des utilisateurs, et la conception qui définira l'architecture générale de l'application. Nous avons choisis la méthode de conception « OMT » qui sera appliquée en adoptant le langage de modélisation objet UML. (Voir Annexe A)

### **II- l'analyse des besoins :**

Notre site e-commerce est un site B2C (business-to-consumer ou entreprise-client) qui désigne l'échange de produits, services ou renseignements entre entreprises et clients, et dans notre cas l'entreprise est le concessionnaire des voitures Meklati qui vend des voitures de la marque Citroën et des pièces de rechange, l'analyse des besoins se concentre sur les fonctionnalités requise par le concessionnaire, donc notre application web doit satisfaire au maximum les exigences du concessionnaire, et pour cela elle doit supporter les fonctionnalités suivantes :

- La création d'une base de données qui permet la gestion des clients, des véhicules et des pièces de rechange.....etc.
- Conception d'un système de catalogue.
- La création d'un panier virtuel qui permet aux clients d'enregistrer les véhicules ou les pièces de rechange désirées.
- Chaque produit (véhicules et pièces de rechange) est présenté en détail sur sa propre page.
- A tout moment, le client peut accéder aux formulaires du bon de commande, dans lequel il saisit ses coordonnées et les informations nécessaires au paiement, ou imprimer un devis pour commander par fax ou courrier.
- L'envoi des données est crypté pour garantir la sécurité.

- La création d'un module de recherche (recherche par mot clé) pour les produits.
- La réalisation d'un outil d'administration du site (gestion du catalogue des produits, Suivi des commandes ....).
- La réalisation d'un forum de chat pour les clients.
- Utilisation l'XML pour l'échange de données.

Donc l'objectif principal de notre site est de permettre aux internautes de passer des commander et les payer directement sur le web (30% pour une voiture et 100% pour une pièce de rechange) ou imprimer un devis pour commander par fax ou courrier. Et de leurs donner la possibilité de rechercher des voitures ou des pièces de rechange par mot-clé, de se constituer un panier virtuel.

### **II.1- Présentation du site e-commerce :**

Notre application Web comportera trois grands volets à savoir :

- La partie statique : c'est l'ensemble des pages Web statiques (A propos,...).
- La partie dynamique (le catalogue des produits, les prix, les formulaires...) accessible par tous les visiteurs du site.
- La partie administration de la base de données accessible uniquement par l'administrateur.

### **Scénario d'utilisation :**

Si un client veut acheter une véhicule ou une pièce de rechange, il va le chercher à l'aide d'un moteur de recherche (si le client connaît le nom du produit), ou flâner dans les rayons du site (sinon), après avoir trouvé le produit désiré, il le met dans le panier et il peut rechercher un nouveau produit.

Le client peut accéder à son panier et modifier la quantité ou supprimer une véhicule ou une pièce de rechange.

A tout moment le client peut passer sa commande, et remplit les formulaires de bon de commande.

Si le client dispose d'une carte de crédit bancaire il passe sa commande, sinon il peut imprimer sa commande pour commander par fax. Deux cas se présentent alors :

**A- Paiement par carte de crédit :**

Le client va payer 30% du véhicule et 100% de pièce de rechange, l'administrateur envoie la commande vers le fournisseur, si les produits sont disponibles, le fournisseur informe l'administrateur, le concessionnaire (l'administrateur) récupère les produits, puis informe le client par e-mail ou par téléphone.

**B- Paiement par virement bancaire :**

Le client règle le montant de la commande par virement bancaire, celui-ci retourne le bon de commande et l'avis de paiement par fax, l'administrateur vérifie le paiement, et reprend la procédure du premier cas.

Pour réaliser notre application, nous adopterons une architecture à 3-tiers (voir **figure 4.1**).

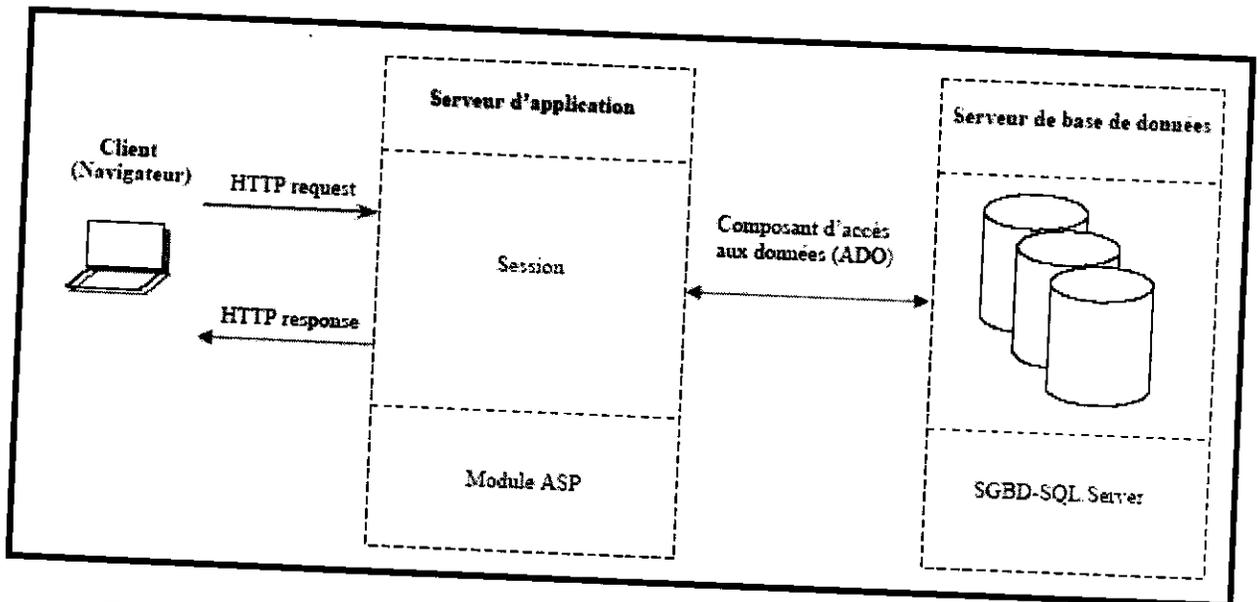


Figure 4.1 : l'architecture à 3-tiers du site Web de commerce électronique.

**III- Conception du site e-commerce :**

**III.1- Identification des acteurs :**

Les acteurs du site web sont les suivants :

- Le client : le personne qui visite le site pour rechercher des produits, ou passer éventuellement une commande, c'est l'acteur le plus important.

- L'administrateur : les responsables du contenu rédactionnel du site, qui s'occupent du suivi des commandes des clients, le bon fonctionnement et la maintenance du site web.
- Le service de paiement : qui assure une passerelle entre notre site et le réseau privé de la banque (vérifier c'est la carte de crédit est valable ou non, ainsi que lors du paiement il transfère le montant du compte du client vers le compte du concessionnaire).

III.2- Identification des cas d'utilisation et les diagrammes de séquence :

III.2.1- Représentation générale des cas d'utilisations :

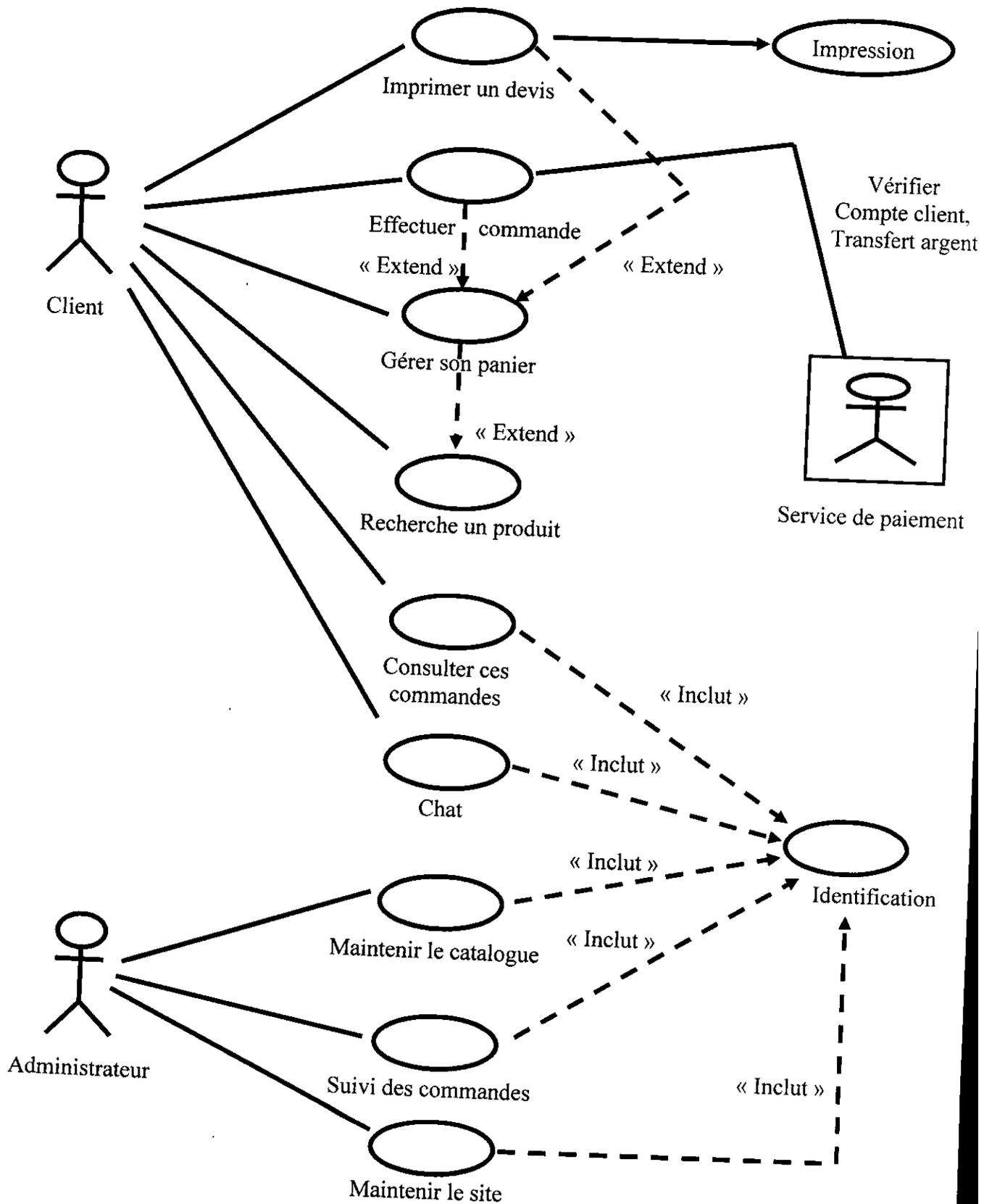


Figure 4.2 : Diagramme de cas d'utilisation

On peut donner les différents scénarios concernant les diagrammes des cas d'utilisations qui seront utilisés et leurs diagrammes de séquences.

**III.2.2- cas d'utilisation - recherche un produit - :**

Le client veut trouver le plus rapidement possible le véhicule ou la pièce de rechange désirées dans l'ensemble du catalogue qui sont organisées en deux rayons le premier rayon pour les voitures et le deuxième pour les pièces de rechange, mais il y a deux outils de recherche, il peut effectuer une recherche séquentielle (flâner dans les rayons de site) ou bien à l'aide d'un moteur de recherche (par un mot clé) qu'on va le prendre comme le cas nominale.

**Condition :**

Catalogue non vide

**Scénario :**

- 1- Le client entre un mot-clé ou le nom du produit et lance la recherche.
- 2- Le système affiche le véhicule ou la pièce de rechange trouvée dans une nouvelle page.
- 3- Le client sélectionne le produit.
- 4- Le système affiche une fiche détaillée du produit (l'image, le prix,.....etc.).
- 5- Le client clique sur le bouton acheter et le produit est met dans son panier virtuel.

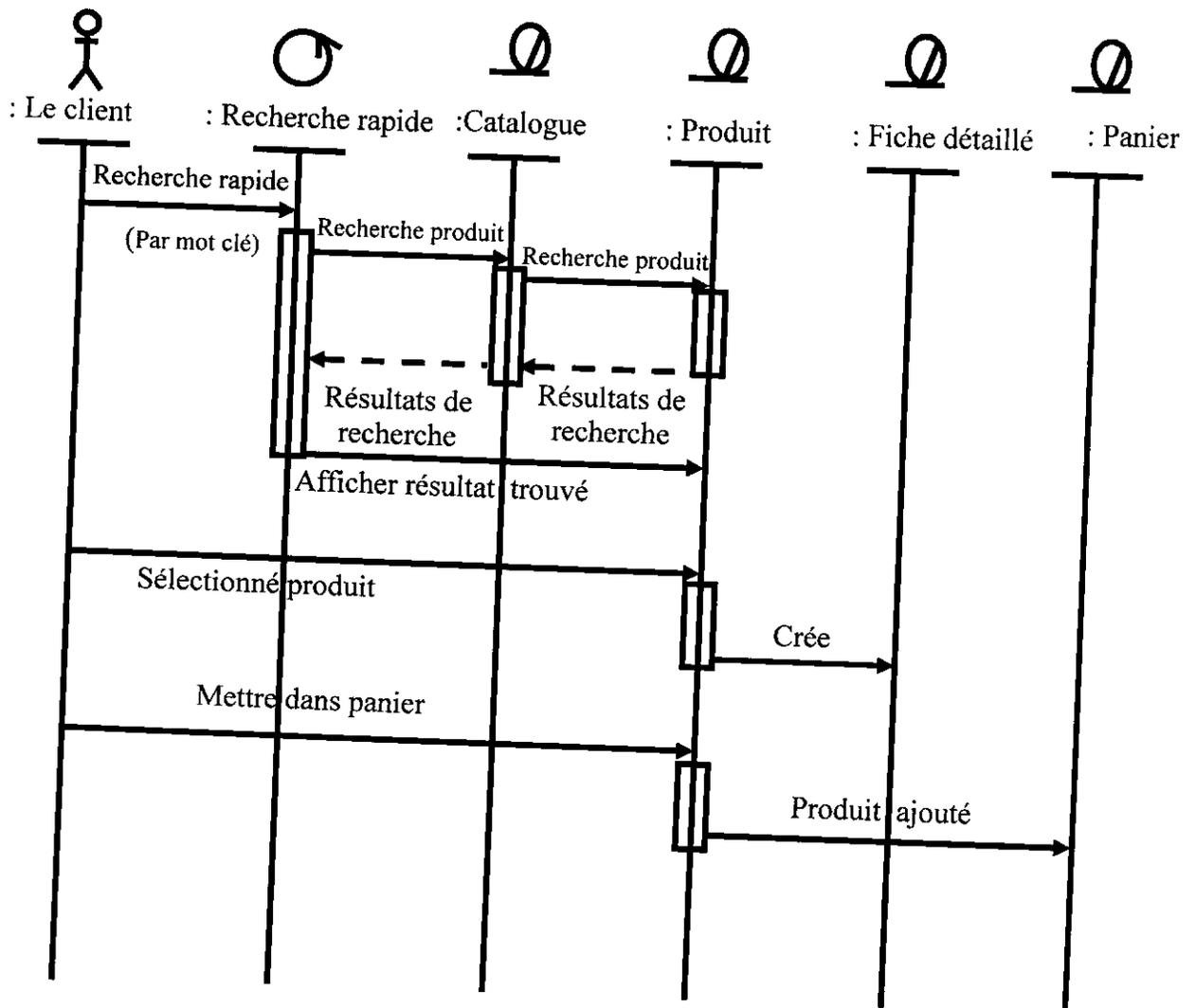


Figure 4.3 : Diagramme de séquence – recherche produit –

**Scénario d'exception <recherche séquentielle > :**

- 1- Le client sélectionne l'un des deux rayons disponibles.
- 2- Le système affiche l'ensemble du catalogue du rayon sélectionné.
- 3- Aller à l'étape 3.

**Scénario d'exception <produit non trouvé > :**

- 1) le système n'a pas trouvé le produit et il affiche un message « produit non trouvé ».
- 2) aller à l'étape 1.

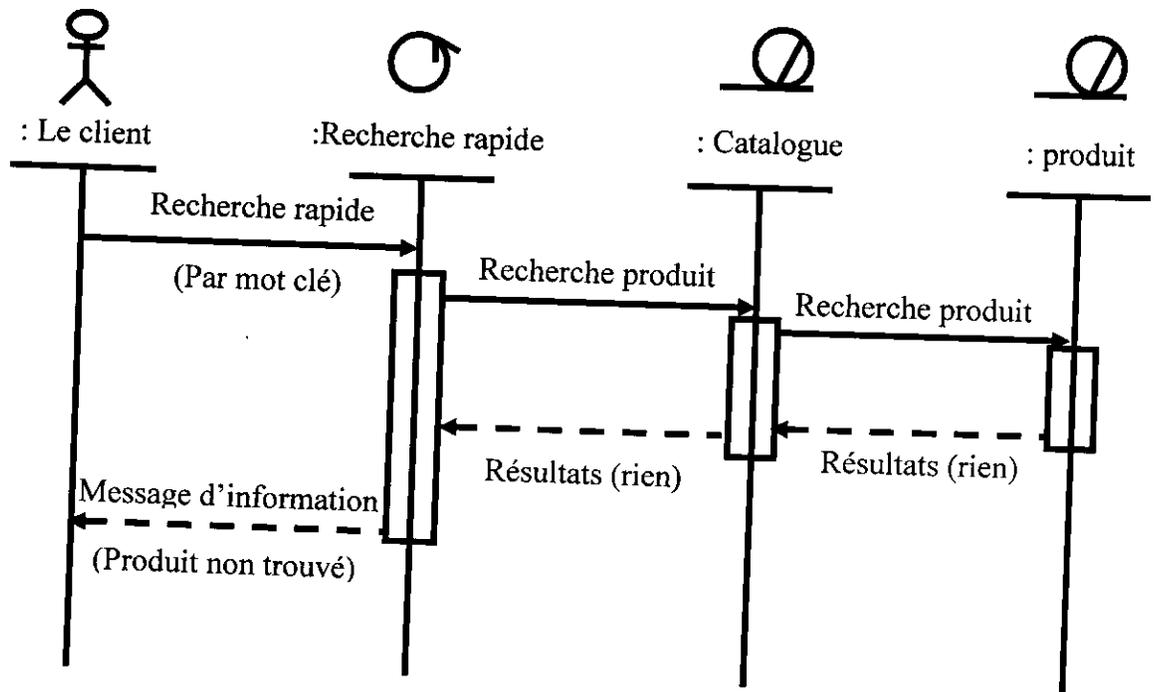


Figure 4.4 : Diagramme de séquence < Exception (produit non trouvé) >

### III.2.3- cas d'utilisation - gérer son panier - :

Si le client est intéressé par le produit recherché, il le met dans son panier virtuel et puis il peut ajouter un nouveau produit, modifier ou supprimer un produit déjà enregistré dans le panier.

**Condition :**

Panier non vide

**Scénario :**

- 1- Le client met le produit (véhicule ou pièce de rechange) recherché dans le panier.
- 2- Il demande l'accès à son panier.
- 3- Le système affiche l'état de son panier.
- 4- Le client peut modifier ou supprimer les lignes du panier.
- 5- Le client peut sortir du panier pour faire une nouvelle recherche, effectuer une commande ou imprimer un devis pour commander par fax.

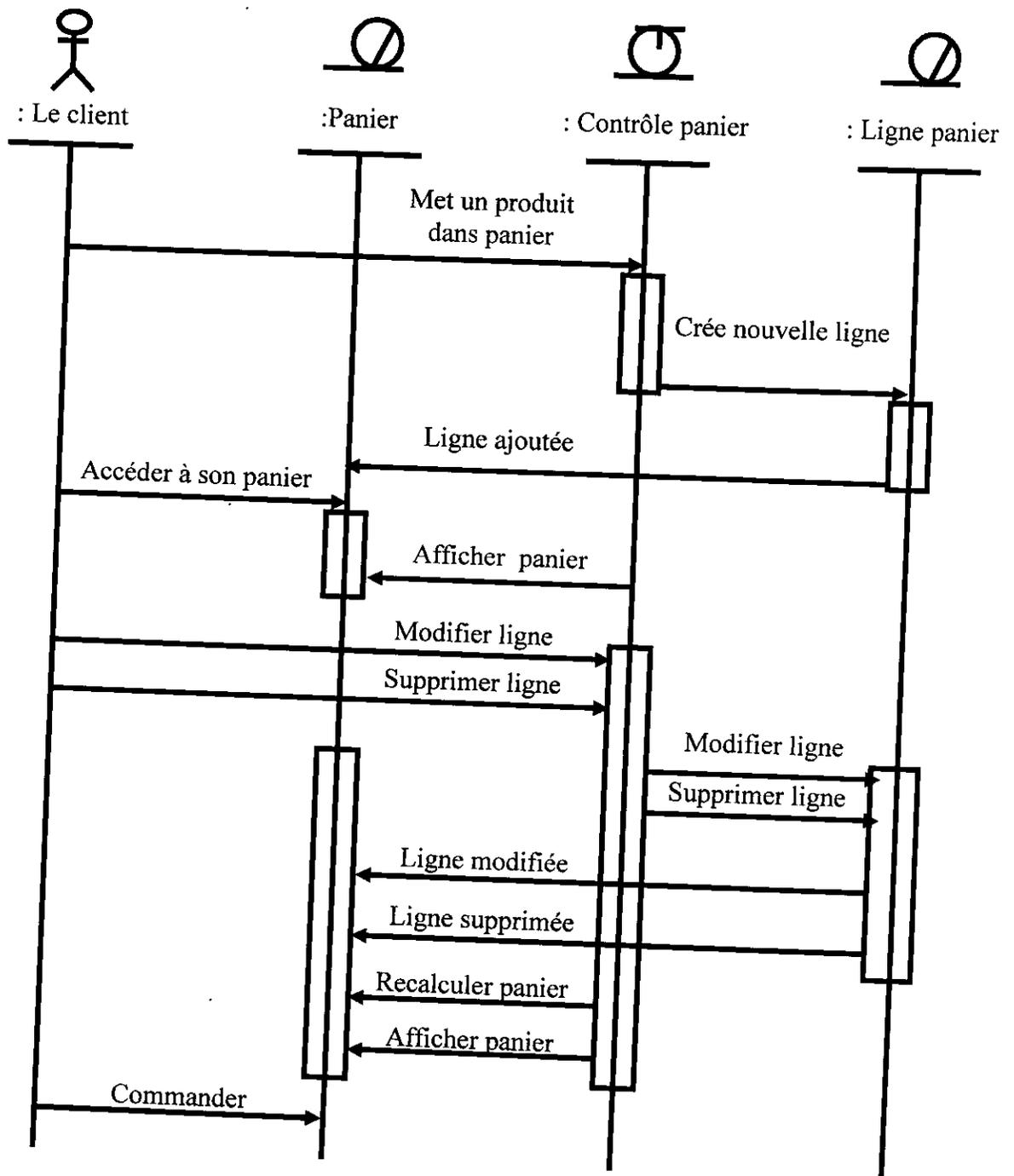


Figure 4.5 : Diagramme de séquence - gérer panier -

**Scénario d'exception <panier vide > :**

- 3- Le système affiche un message d'information « le panier est vide ».
- 4- Aller à recherche un produit.

**III.2.4- cas d'utilisation - effectuer une commande - : (par carte de crédit)**

Le client peut effectuer une commande à condition que le panier ne soit pas vide.

**Scénario**

- 1- Le client clique sur le bouton effectuer une commande.
- 2- Le système affiche des formulaires à remplir par Le client qui contiennent des informations concernent le paiement, l'inscription, la carte de crédit.....etc.
- 3- Le client saisit l'ensemble des informations.
- 4- Le système affiche un récapitulatif de la commande.
- 5- Le client valide sa commande.
- 6- Le système envoi les informations de la carte de crédit concernant le client pour effectue le transfert d'argent au service de paiement.
- 7- Le service de paiement confirme le transfert d'argent.
- 8- Le système confirme la prise de la commande.

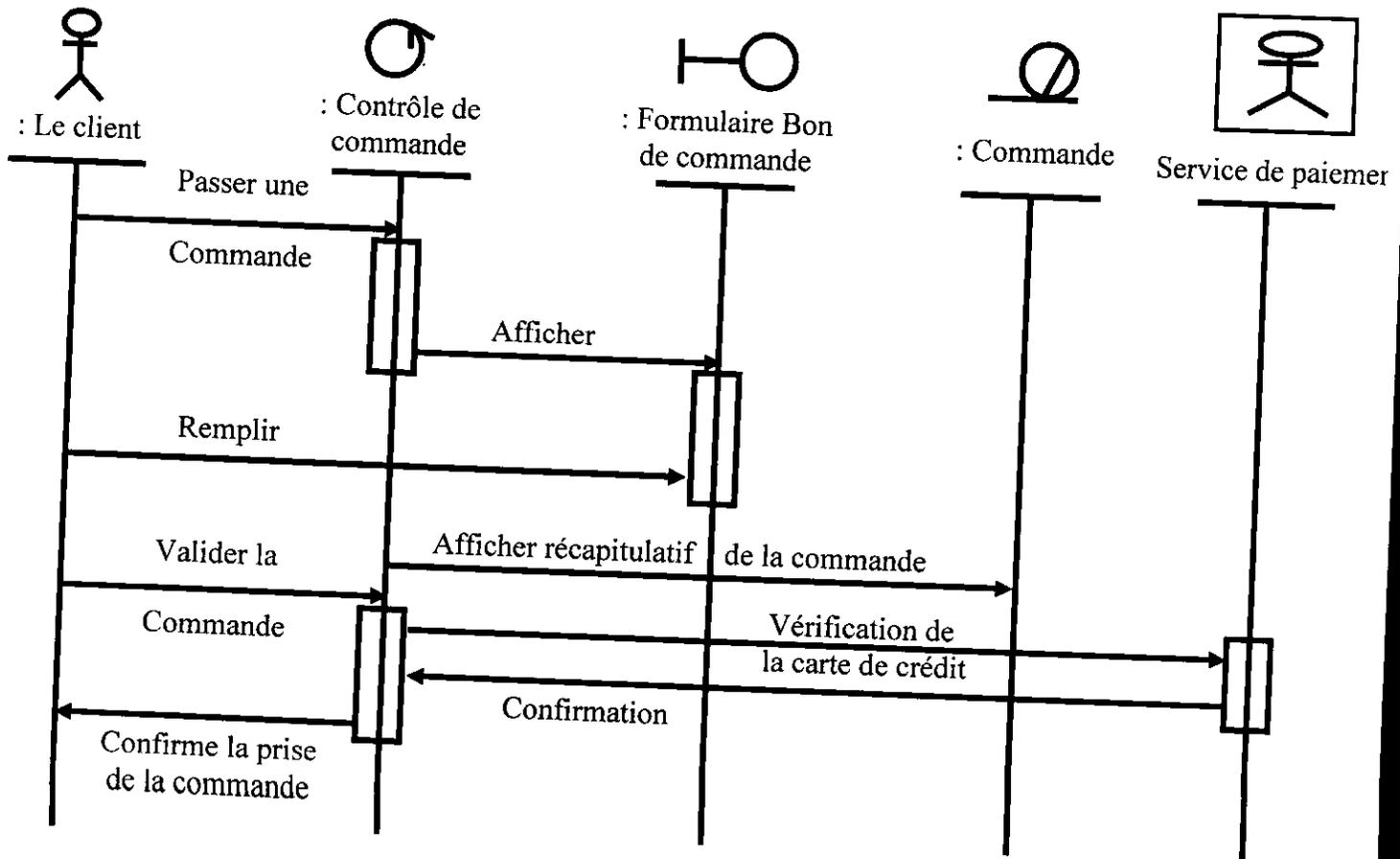


Figure 4.6 : Diagramme de séquence - effectuer une commande -

**Scénario d'exception <carte de crédit non valide > :**

- 7- Le service de paiement signal le système de la non validité de la carte de crédit.
- 8- Le système informe le client que sa carte de crédit n'est pas valide.

**III.2.5- cas d'utilisation - imprimer un devis (un bon de commande) - :**

Le client peut ne pas effectuer sa commande par e-mail (exemple : ne dispose pas une carte de crédit bancaire) mais le fait par fax ou courrier dans ce cas Le client demande l'impression du devis.

**Condition :**

Panier non vide.

**Scénario**

- 1- Le client clique sur le bouton effectuer une commande.
- 2- Le système affiche des formulaires à remplir par Le client qui contiennent des informations concernant l'inscription.....etc.
- 3- Le client saisit l'ensemble des informations.
- 4- Le système affiche un récapitulatif du devis.
- 5- Le client clique sur le bouton 'imprimer devis'.
- 6- Le système génère le bon de commande au format PDF.
- 7- Le client valide l'impression.
- 8- Le système lance l'impression.

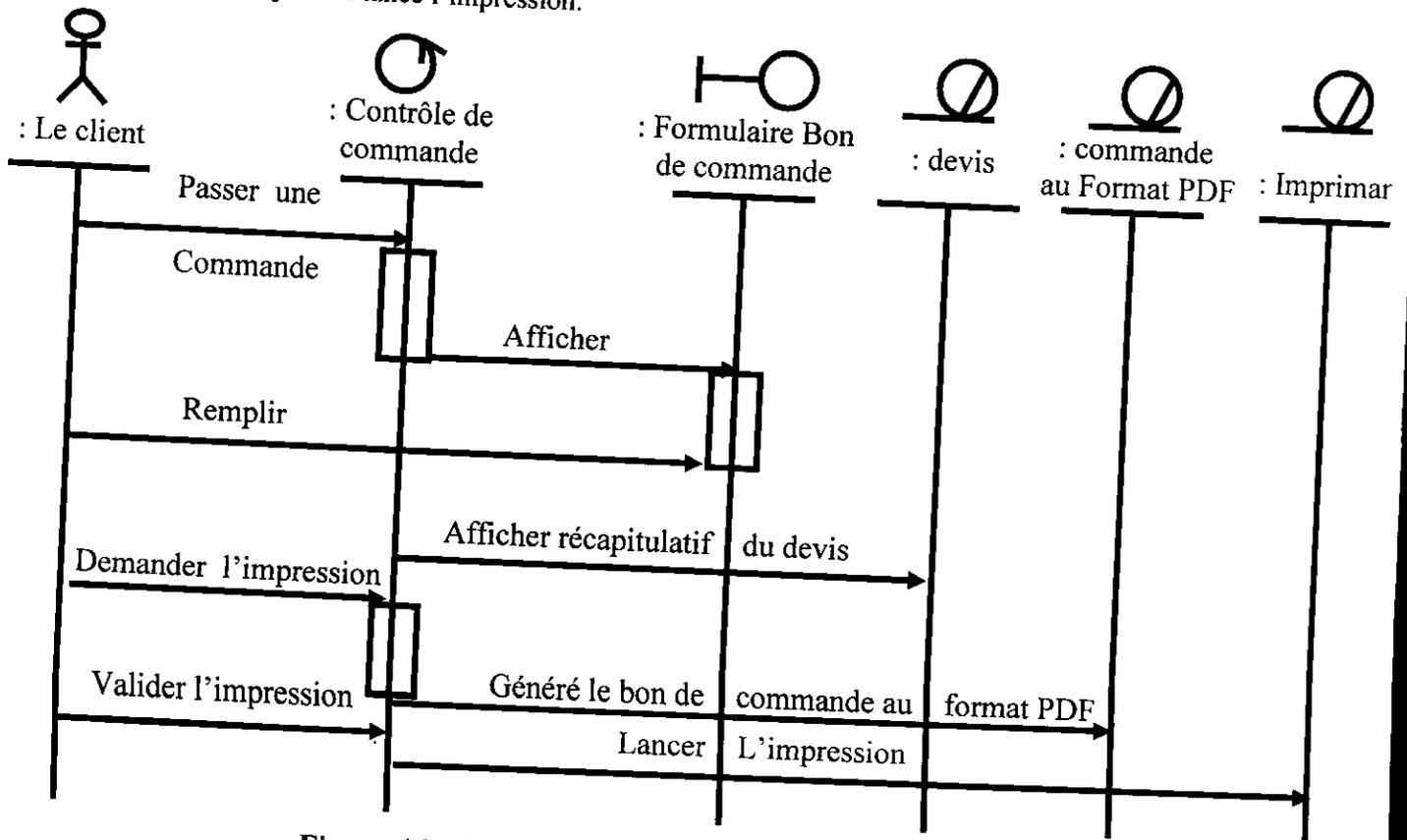


Figure 4.7 : Diagramme de séquence - imprimer un devis -

**III.2.6- cas d'utilisation - consulter ces commandes - :**

Dans ce cas Le client est déjà un client inscrit et il veut voir l'état de sa commande.

**Scénario**

- 1- le système demande au client d'entrée son e-mail et son mot de passe.
- 2- Le client entre l'e-mail et le mot de passe.
- 3- Le système affiche un récapitulatif de sa commande et son état.

**Scénario d'exception <le mot de passe n'est pas valide > :**

- 3- Le système affiche un message d'erreur « ce mot de passe n'est pas valide ».
- 9- Aller à l'étape 1.

**III.2.7- cas d'utilisation - entrée dans le forum - :**

Un client peut utiliser le forum de discussion pour la communication avec d'autre client ou pour laisser un commentaire.

**Scénario**

- 1- le système demande au client d'entrée son e-mail et son mot de passe.
- 2- Le client entre l'e-mail et le mot de passe.
- 3- Le système affiche le contenu du forum.
- 4- Le client demande d'écrire un commentaire.
- 5- Le système affiche une zone d'écriture.
- 6- Le client note son message et le valide.

**III.2.8- cas d'utilisation - maintenir le catalogue - :**

L'administrateur est le responsable du catalogue du produit (véhicules et pièces de rechange) présenté dans le site donc il à la possibilité d'ajouter, modifier ou supprimer les produits ou les rayons.

**Scénario**

- 1- le système demande à l'administrateur d'entrer son login et son mot de passe.
- 2- L'administrateur entre le login et le mot de passe.
- 3- Le système affiche son environnement du travail.

- 4- L'administrateur fait les opérations désirées (ajouter, modifier ou supprimer les produits ou les rayons).
- 5- L'administrateur valide ses opérations.
- 6- Le système confirme la validation.

**Scénario d'exception <le mot de passe n'est pas valide > :**

- 3- Le système affiche un message d'erreur « ce mot de passe n'est pas valide ».
- 4- Aller à l'étape 1.

**III.2.9- cas d'utilisation - suivi les commandes - :**

L'administrateur est le responsable de suivi les commandes puis les traiter.

**Scénario**

- 1- le système demande à l'administrateur d'entrée son login et son mot de passe.
- 2- L'administrateur entre le login et le mot de passe.
- 3- Le système affiche son environnement du travail.
- 4- L'administrateur clique sur le bouton « lister des commandes »
- 5- Le système affiche la liste des commandes classées par date.
- 6- L'administrateur choisit une commande à traiter.
- 7- L'administrateur valide la commande traitée.
- 8- Le système confirme la validation.

**Scénario d'exception <le mot de passe n'est pas valide > :**

- 3- Le système affiche un message d'erreur « ce mot de passe n'est pas valide ».
- 4- Aller à l'étape 1.

**III.3- DIAGRAMME DES CLASSES :**

Après avoir donné les différents scénarios concernant les diagrammes des cas d'utilisations qui seront utilisés et leurs diagrammes de séquences, nous allons maintenant introduire l'étape typiquement orientée de l'analyse qui décompose le domaine d'intérêt en classes conceptuelles représentant les entités significatives de ce domaine, si l'on emploie la notation UML on peut représenter les éléments suivants :

- Les classes conceptuelles ou les objets du domaine.
- Les associations entre les classes conceptuelles.

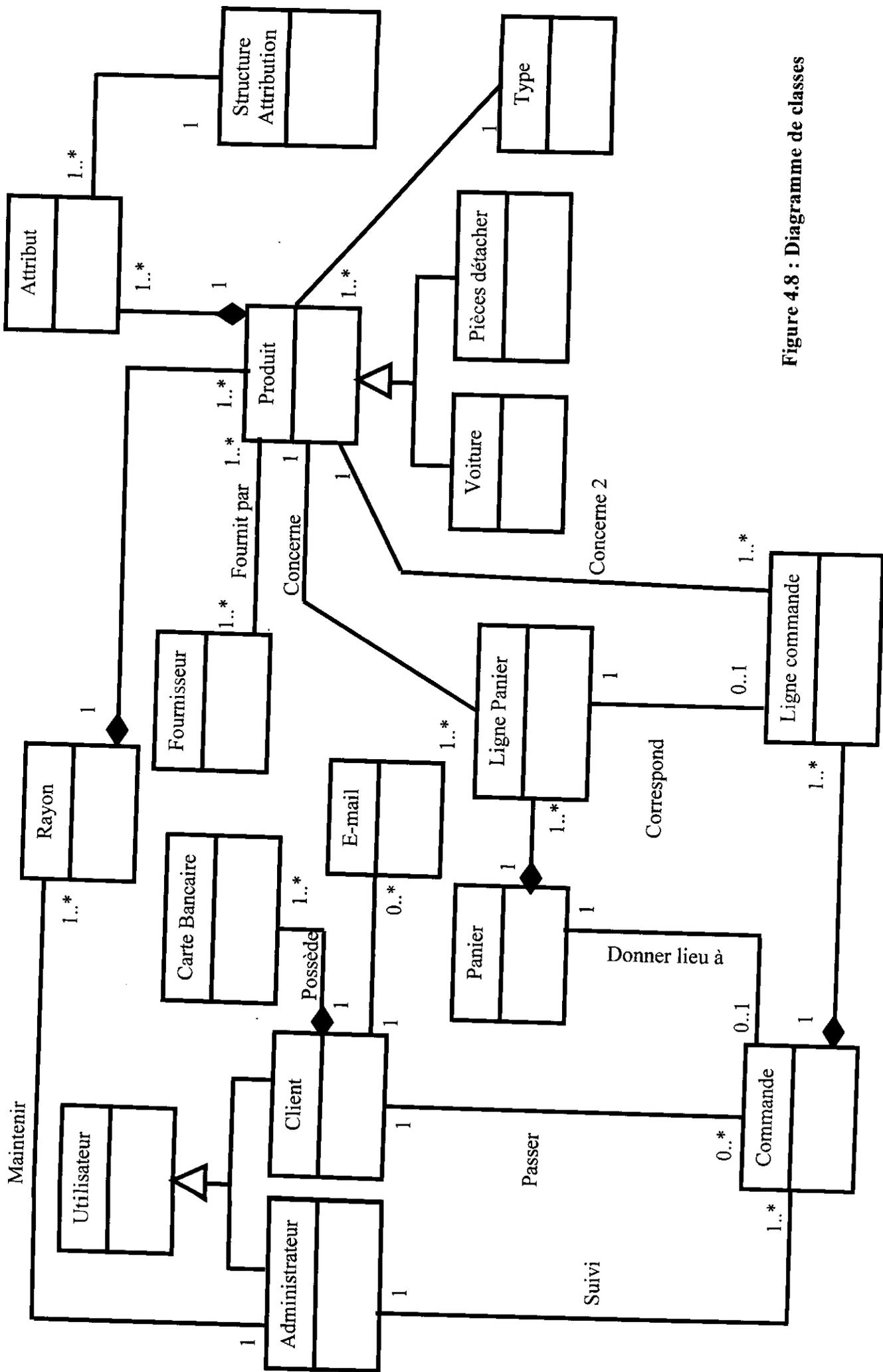


Figure 4.8 : Diagramme de classes

### **III.4- DIAGRAMMES D'ACTIVITES :**

Le diagramme d'activités est un des diagrammes d'UML, qui modélise la navigation d'un site web. Nous allons nous servir d'un certain nombre d'éléments standard; En effet un diagramme d'activités contient un nombre restreint d'éléments, à savoir :

- Des activités.
- Des transitions entre activités, pouvant porter des conditions.
- Des branchements conditionnels.
- Un début et une ou plusieurs terminaisons possibles.

On va donner les diagrammes d'activités des quatre cas d'utilisations suivants :

- Recherche produit.
- Gérer panier.
- Effectuer commande.
- Maintenir le catalogue.

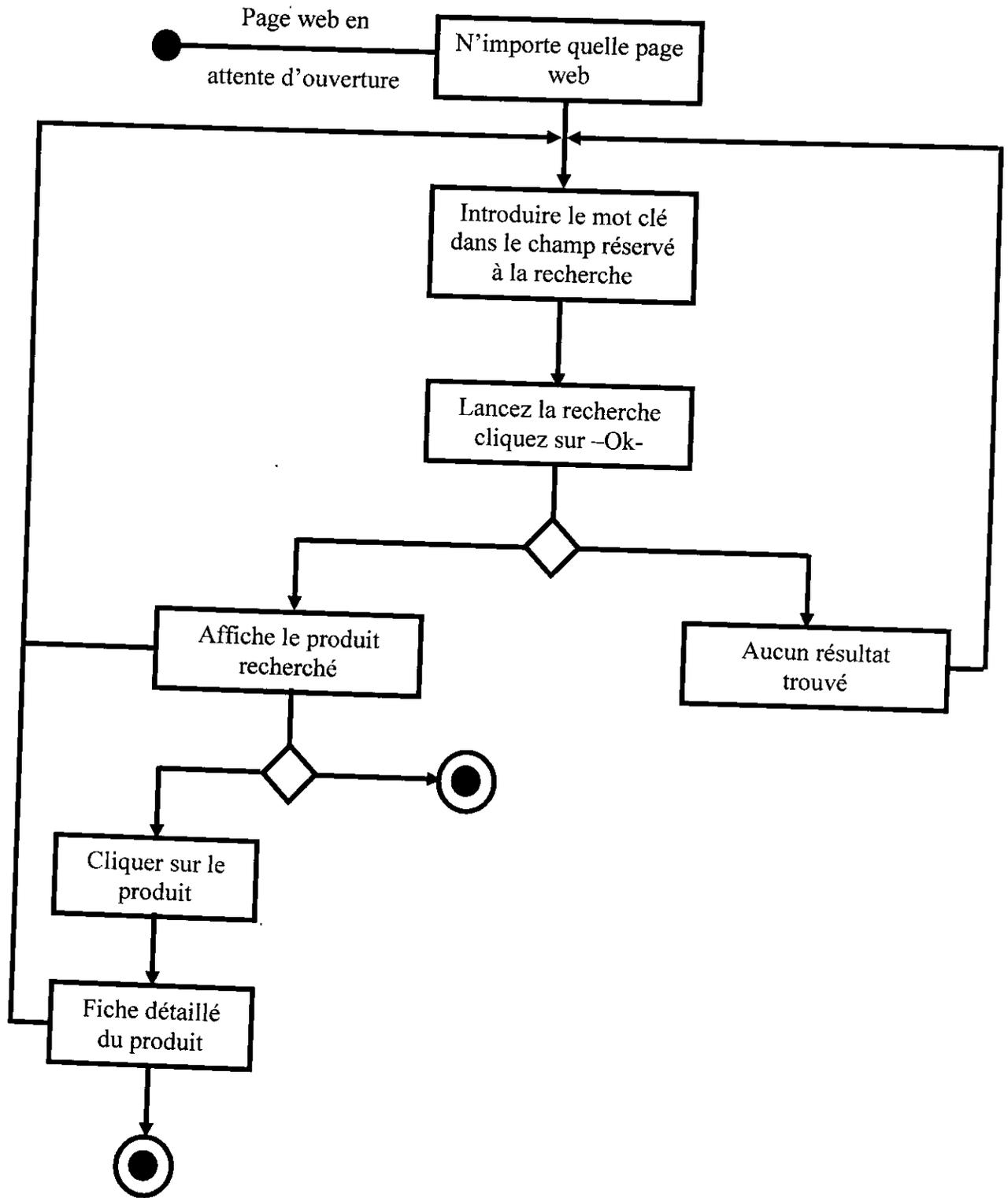


Figure 4.9 : Diagramme d'activité - Recherche produit -

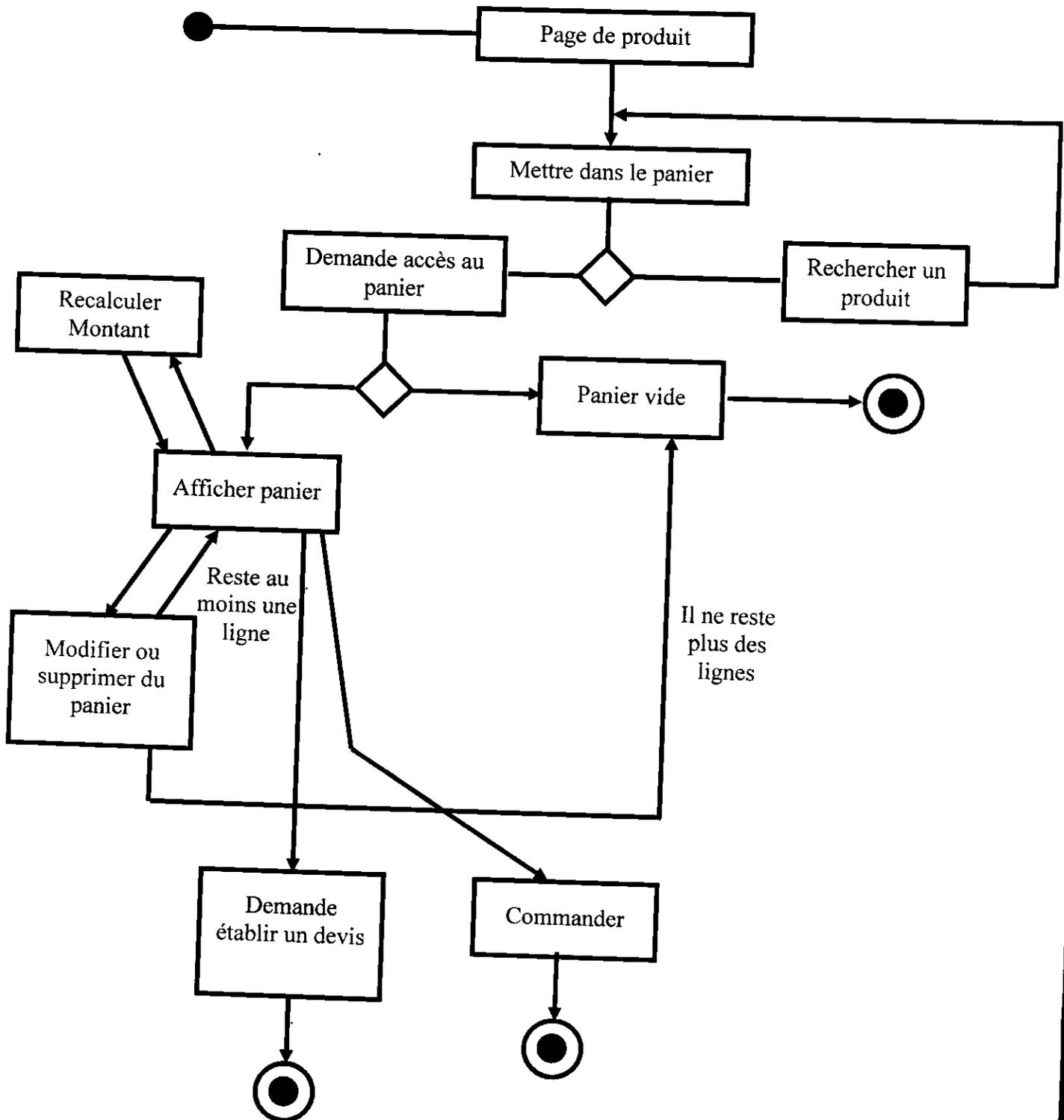


Figure 4.10 : Diagramme d'activité - Gérer panier -

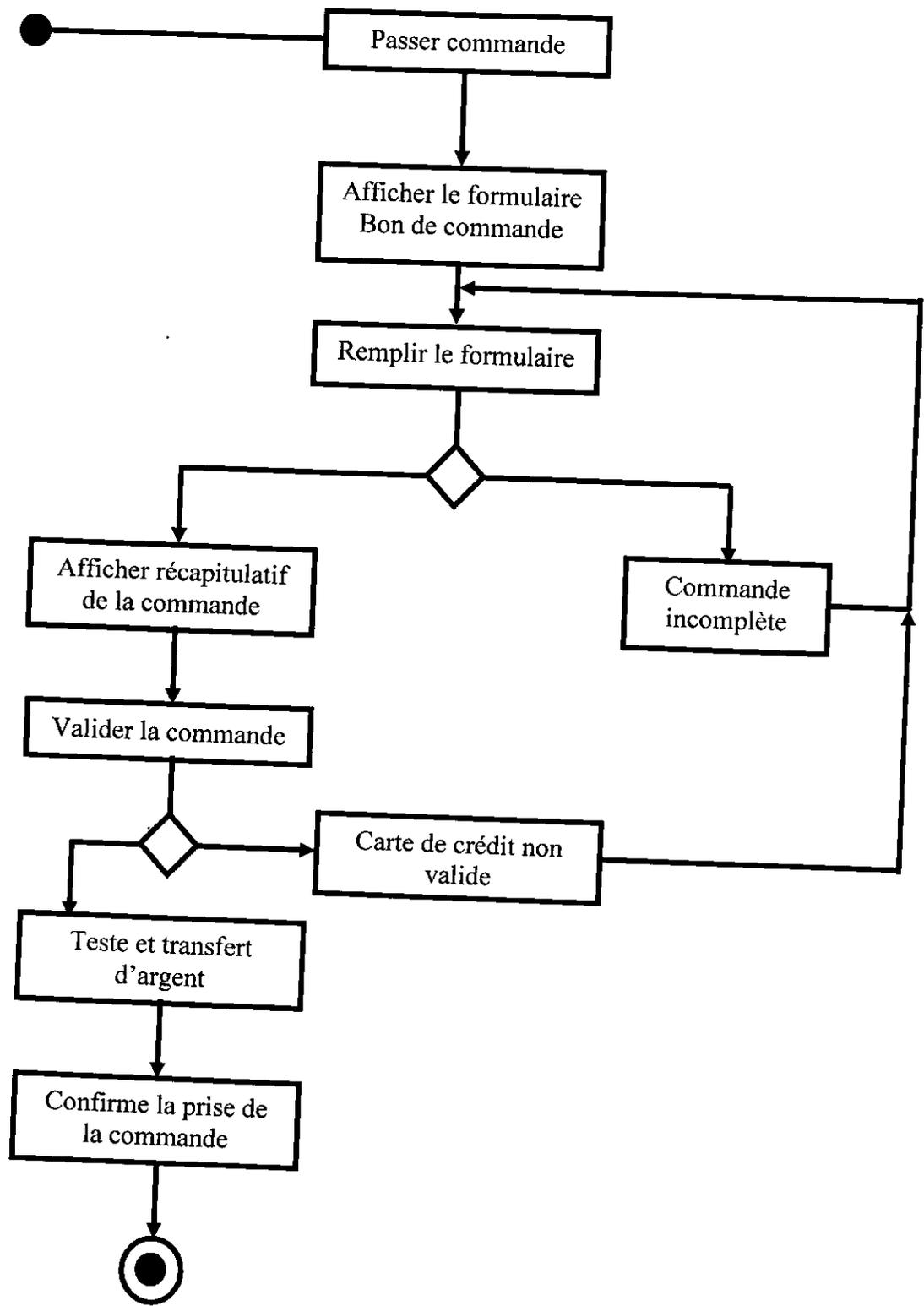


Figure 4.11 : Diagramme d'activité - Effectuer commande -

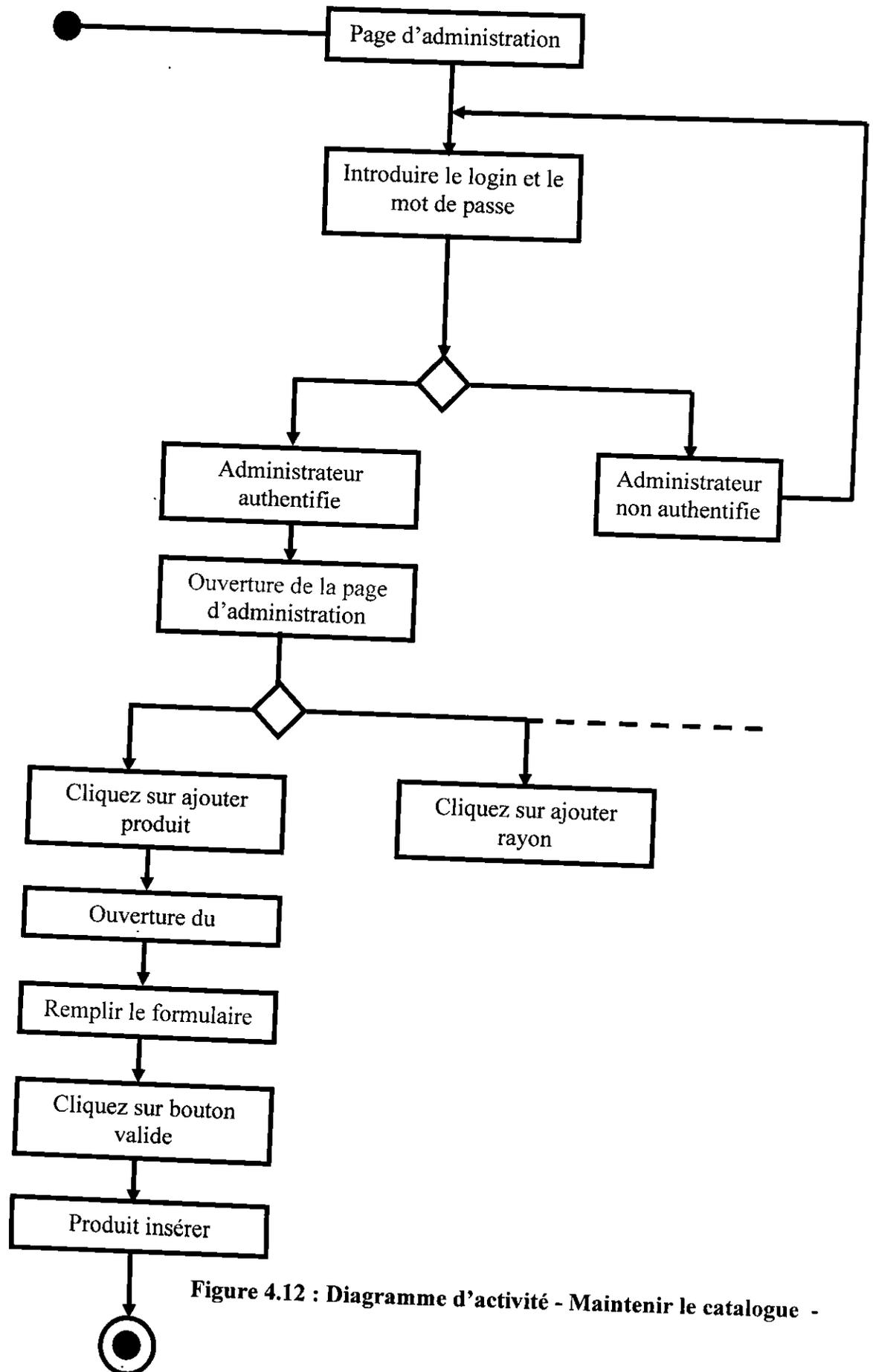


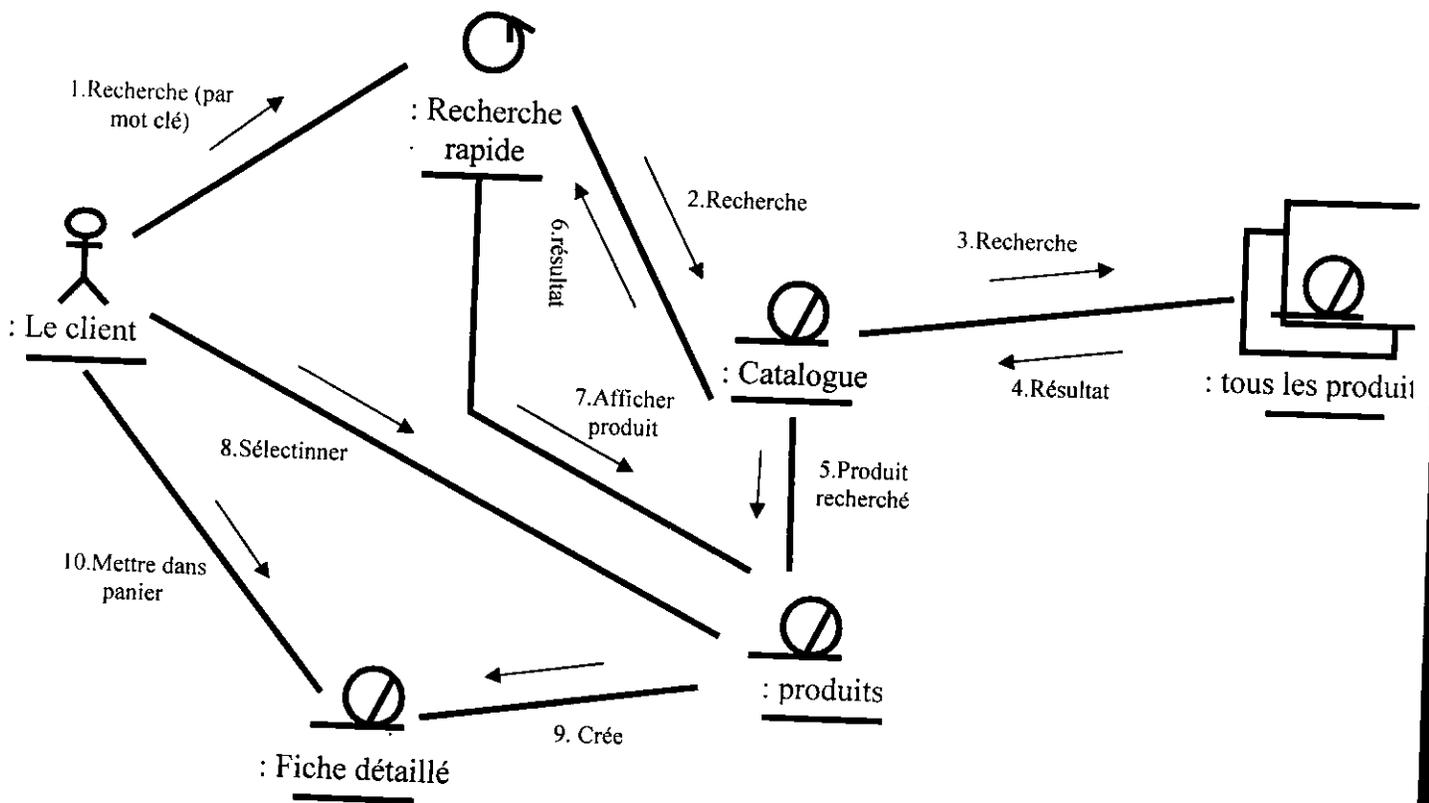
Figure 4.12 : Diagramme d'activité - Maintenir le catalogue -

**III.5- DIAGRAMMES DE COLLABORATION :**

Les diagrammes de collaboration illustrent les interactions entre objets sous forme de graphe ou de réseaux. Les objets peuvent être placés en tout point du diagramme. Ils sont connectés par des liens qui indiquent qu'une forme de navigation entre ces objets est possible. Tout message entre objets est représenté par une expression et une petite flèche indiquant son sens de circulation. Chaque lien permet la circulation de plusieurs messages et chaque message est assorti d'un numéro d'ordre.

On va donner les diagrammes de collaboration des trois cas d'utilisations suivants :

- Recherche produit.
- Gérer panier.
- Effectuer commande.



**Figure 4.13 : Diagramme de collaboration – recherche produit -**

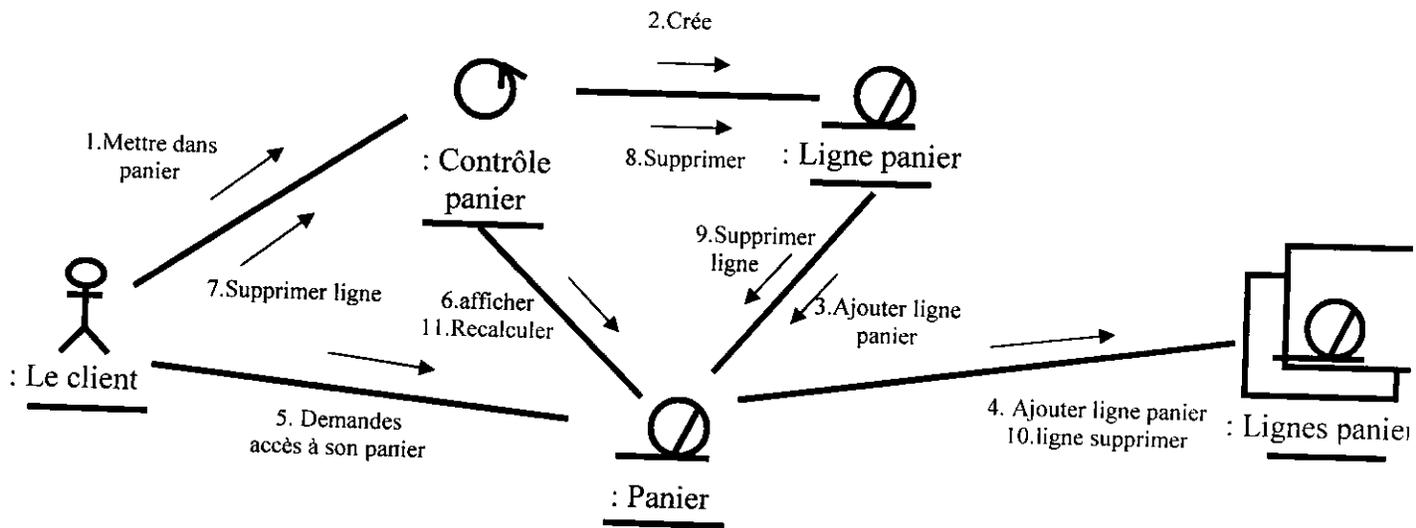


Figure 4.14 : Diagramme de collaboration – Gérer panier -

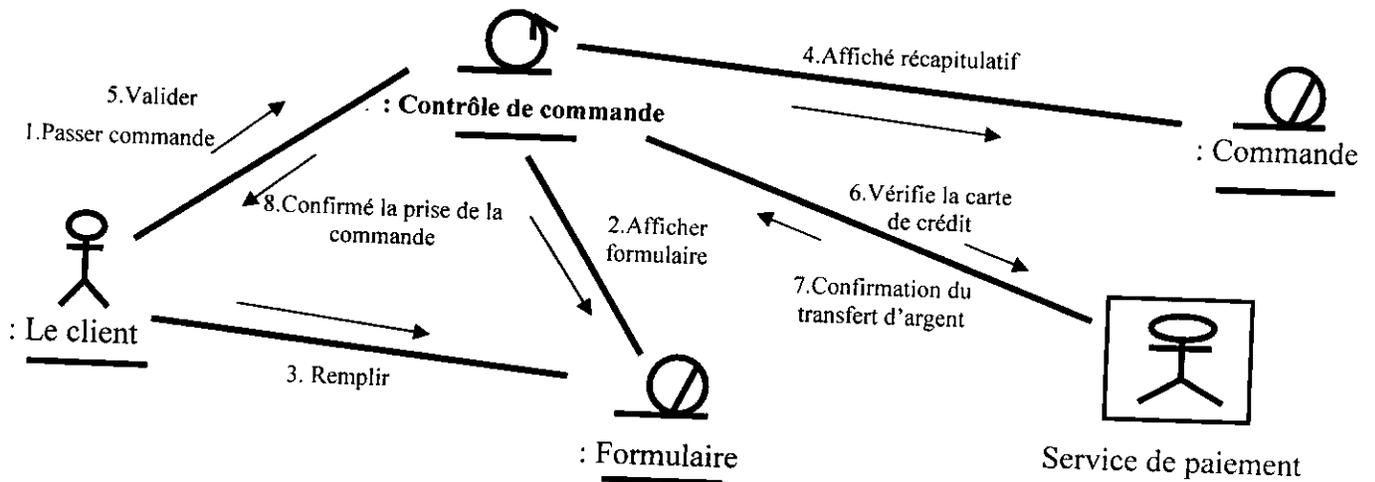


Figure 4.15 : Diagramme de collaboration – effectuer commande -

### III.6- Diagramme de déploiement :

Pour concrétiser les fonctionnalités précédentes, nous avons besoin d'une architecture matérielle en client serveur. Un serveur ouvert sur Internet et des postes pour les utilisateurs ayant la possibilité d'accéder à L'Internet, et un serveur de données.

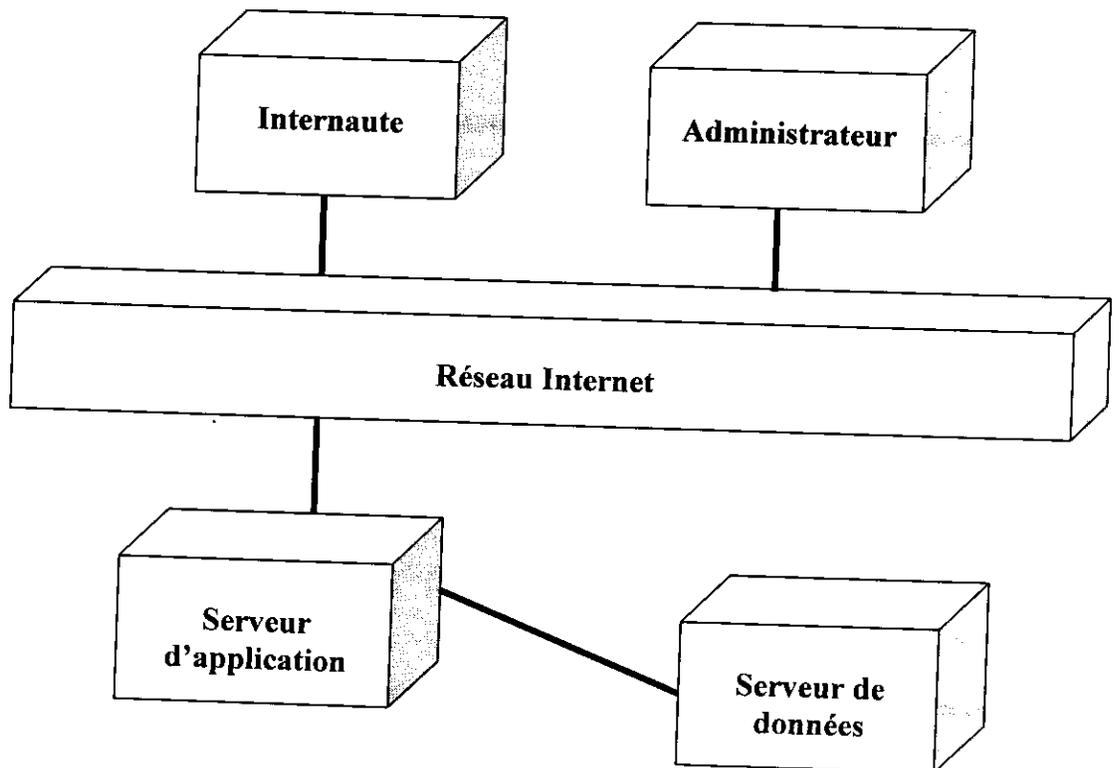


Figure 4.16 : Diagramme de déploiement

### III.7- Conception de la base de données :

Il s'agit de rendre cohérent une masse d'informations et de la mettre à disposition de l'administrateur du site d'un côté et des clients d'un autre côté, Notre système inclut une base de données relationnelle qui va contenir les données propres à l'application. Pour passer du model objet au modèle relationnel il faut appliquer les règles de passage.

**III.7.1- Règles de passage du model objet au modèle relationnel :****1)Représentation des classes d'objets en tables :**

- chaque classe est représentée par un ou plusieurs tables.

**2)Représentation des associations d'objets en tables :**

- Une association un à plusieurs est représentée par une table distincte ou peut être enfouie comme clé étrangère dans la table pour l'une ou l'autre des classes. Pour les associations un à plusieurs ou un à un, il n'y a pas de cycle, on dispose de l'option supplémentaire qui consiste à ranger l'association et les objets liés dans une seule table.

Ayant conscience que cela peut introduire une redondance et violer des formes normales.

- Les noms de rôles sont incorporés en tant que partie du nom de l'attribut de la clé étrangère.
- Les associations n-aires ( $n > 2$ ) se représentent par des tables distinctes. Ce qui aide parfois à promouvoir une association n-aires en une classe.
- Une association qualifiée se représente en une table distincte avec au moins trois attributs, la clé primaire de chaque classe liée est le qualificatif.
- Les agrégations suivent les mêmes règles que les associations.

**3)Représentation de la généralisation de l'héritage simple en tables :**

- On représente chaque superclasse et chaque sous-classe par une table.
- S'il n'y a de table de superclasse, les attributs sont dupliqués dans chaque table de sous-classe.
- S'il n'y a de table de sous-classe, on apporte tous les attributs de sous-classe dans la superclasse.

**III.7.2- Traduction du modèle Objet En base de Données relationnelle :**

En appliquant les règles de passage au modèle objet, on obtient la représentation logique de notre base de données. Les tables de la base de données avec leurs attributs : (voir le script de la base de données Annexe C)

- Carte** (Idcarte, \*Idclient, Type, Numero, Nomsurcarte, Moisexpiration, Anneeexpiration)
- Clients** (Idclient, prenom\_client, nom\_client, email\_client, Motdepasse, adresse, ville, codepostal, téléphone)
- Commandes** (Idcommande, \*Idclient, \*Idcarte, Creee, Achevee, Etat)
- Fournisseurs** (Idfournisseur, nom\_fournisseur, url\_fournisseur, Premcontactcom, Derncontactcom, Emailcontactcom)
- Lignes commande** (Idligne, \*Idproduit, Quantite, Prixunite, Total)
- Paniers** (Idpanier, Delaisexpiration)
- Produits** (Idproduit, \*Idfournisseur, \*Idrayon, \*Idtype, nom\_produit, Details, Description, Prix, Urlimage)
- Rayons** (Idrayon, Idprayeron, Nom\_rayon)
- Lignes panier**(Idarticlepanier, \*idpanier, \*idproduit, quantite)
- Structure attribution** (Idstructure, \*Idtype, nom\_structure, Typedonnee) : qui contient les attributs des véhicules et pièces de rechange (couleurs, puissance, .....etc).
- Attributs** (Idattribut, \*Idstructure, \*Idproduit, Valeurlong, Valeurchaine, Valeurdouble, Valeurdate, Valeurbooleen) : qui contient les valeurs des attributs (rouge, vert, 1200...etc).
- Types** (Idtype, nom\_type)
- E-mail** (Idemail, \*idclient, sujet, message, dateprevue)
- Administrateur** (Idadministrateur, Login, Motdepasse)

**Note** : Les attributs soulignés représentent les clés primaires, et ceux précédé d'une (\*) des clés étrangère.

**IV- Conclusion :**

Ce chapitre traite de la démarche que nous avons suivi pour concevoir notre application Web, définition des besoins des utilisateurs, la conception qui définira l'architecture générale de l'application, la conception de la base de données utilisée pour stocker la masse d'informations de notre application ainsi que la conception de l'application web (scénario d'utilisation, ...etc).

# Chapitre V

## Chapitre V : Implémentation du Site Web

### I- INTRODUCTION :

Ce présent chapitre est consacré à la présentation des illustrations pratiques de notre application web. La navigation sur le site réalisé est divisée en deux parties, la partie accessible par tous les visiteurs et la partie administration réservée uniquement à l'administrateur du site.

Pour chaque partie, nous allons présenter juste quelques interfaces de navigation que nous considérons les plus importantes.

### II- CONFIGURATION LOGICIELLE :

Nous avons adopté une architecture à 3 niveaux, le principe d'une architecture à 3 niveaux est relativement simple : il consiste à séparer la réalisation des trois parties suivantes : stockage des données, logique applicative, présentation. Cette séparation signifie qu'il est possible de déployer chaque partie sur un serveur indépendant, toutefois cela n'est pas obligatoire. La mise en place de ce type d'architecture permet une plus grande évolutivité du système.

Il est ainsi possible de commencer par déployer les deux serveurs (base de données et WEB) sur la même machine, puis de déplacer le serveur applicatif web sur une autre machine lorsque la charge devient excessive.

Pour la réalisation de notre système on a choisit :

- **Système d'exploitation** : Windows XP version professionnelle.
- **Serveur Web**: Internet Information Server 5.0.
- **Serveur de base de données** : Serveur SQL SERVER 2000, avec une interface Graphique, Analyseur de requêtes pour la création et l'interrogation de la base.
- **Langages de programmation** :
  1. Langage HTML sous l'éditeur Microsoft InterDev pour la partie Statique du Web.
  2. Langage de script ASP 3.0 pour la partie dynamique.
- **Le Navigateur WEB** : Internet Explorer.
- **Le protocole de sécurité** : Secure Socket Layer (SSL).

**III - DEVELOPPEMENT A TROIS NIVEAUX :**

Le modèle à trois niveaux représente une architecture de création d'applications distribuées, et c'est celui que nous suivrons pour créer le site e-commerce de Citroën-Meklati. Dans toute application, qu'elle soit simple ou complexe, des services applicatifs interviennent. Chaque service joue un rôle particulier dans l'application et inclut des éléments caractéristiques tels que des :

- **Service de bases de données**, permettant à une application d'interroger et de manipuler des bases de données.
- **Services d'interface utilisateur**, permettant à une application de gérer l'interface utilisateurs.
- **Service d'impression**, permettent à une application d'imprimer des documents.
- **Services réseaux**, permettant aux serveurs et aux clients de communiquer.
- **Service de fichiers**, permettant à une application de manipuler des fichiers sur les disques.
- **Services d'intégration**, permettent à une application de communiquer avec d'autres services et applications.

La figure 5.1 suivante illustre la répartition de ces rôles, ainsi que des services énumérés ci-dessus, entre les différents niveaux.

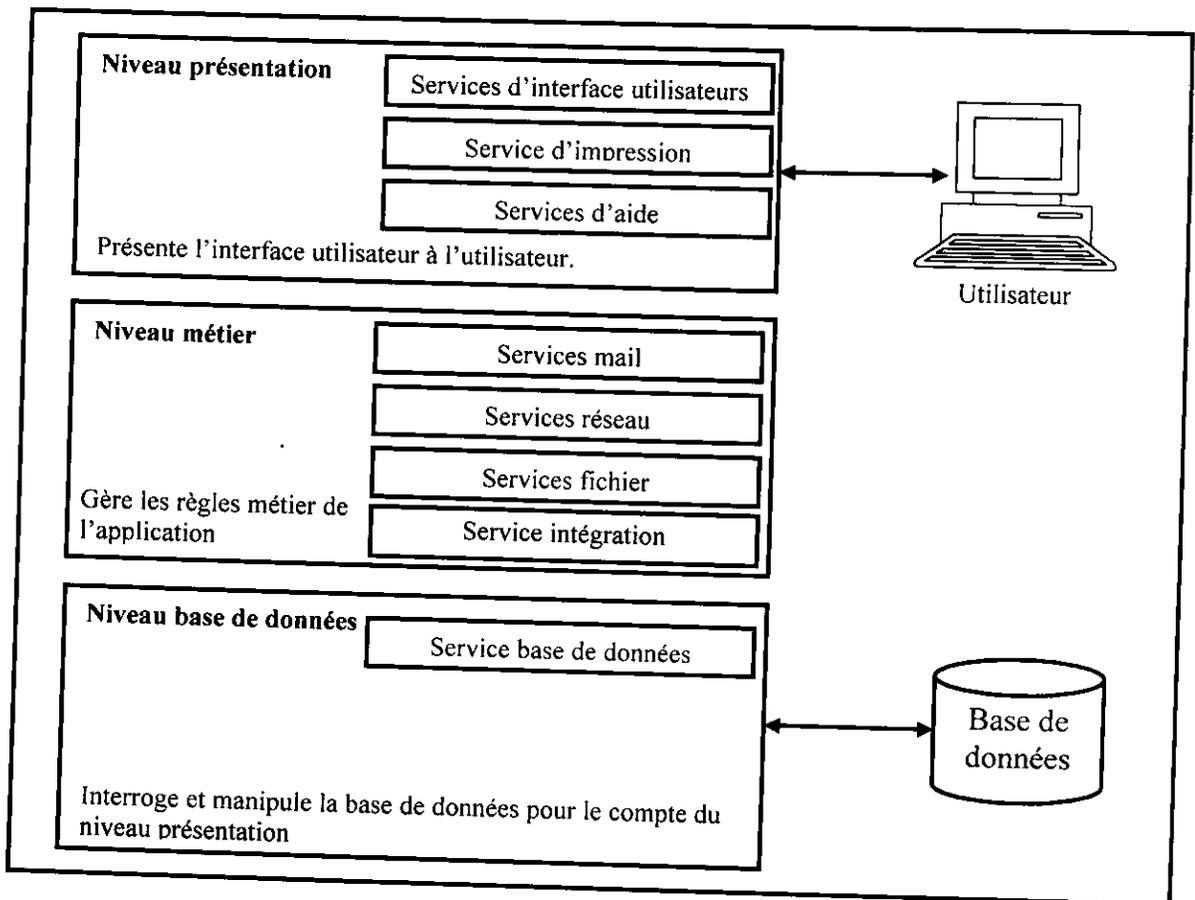


Figure 5.1 : les différents niveau d'une application distribuée orienté web

Détaillons maintenant chacun de ces rôles en notant leurs implications pour la mise en place de Citroën-Meklati

### III-1 Niveau présentation :

Le niveau présentation est chargé de fournir les services requis par l'application pour permettre à l'utilisateur de la voir et de la manipuler. En d'autres termes, elle présente une interface à l'utilisateur.

Dans l'application distribuée orienté web de Citroen-Mekalit, le niveau de présentation se composera d'un ensemble de pages ASP utilisant un serveur web acceptant ASP (Microsoft Internet Information Server) afin de pourvoir fournir, au navigateur de l'utilisateur la présentation de l'application au format HTML. Dans le cadre de ce mémoire, c'est Visual InterDev 6.0 qui a été utilisé pour la création de ces pages.

### **III-2 Niveau Métier :**

Le niveau métier gère les règles métier pour l'application. Puisque l'ensemble des opérations métier suit des règles et procédures définies par les besoins métiers, alors notre applicatif doit suivre les mêmes règles.

La programmation orienté objet (POO) s'appuie sur le principe que les entités et les relations existant dans le monde réel peuvent être représentés dans du code sous la forme d'objets.

Nous allons exécuter les objets métier nécessaires à la création de Citroen-Meklati sous la forme d'objet ActiveX côté serveur (appelés Citroencommerce) écrits en Visual Basic (technologie développées par Microsoft).

### **III-3 Niveau base de données :**

Le niveau base de données est chargé d'interroger et de manipuler la base de données à la demande du niveau métier.

Dans le cadre de notre mémoire, nous avons utiliser une base de données SQL Server afin d'assurer le stockage des données.

### **III-4 L'objet CitroenCommerce :**

Dans les faits, nous allons créer le niveau métier de l'application sous la forme d'un composant ActiveX côté serveur.

Voici un résumé des tâches générales de commerce électronique que le moteur pourra accomplir :

- Gestion du catalogue des produits : la liste des produits en vente sur le site.
- Gestion du panier d'achats : garder une trace de qui veut acheter quoi.
- Gestion des clients : gérer toute la saisie des commandes.
- Exécution des commandes : transformer le contenu du panier en commandes.

Notre but est d'élaborer cet objet métier que nous pourrons utiliser dans le cas du site Citroën-Meklati.

La figure 5.2 illustre la structure hiérarchique de notre modèle objet CitroenCommerce

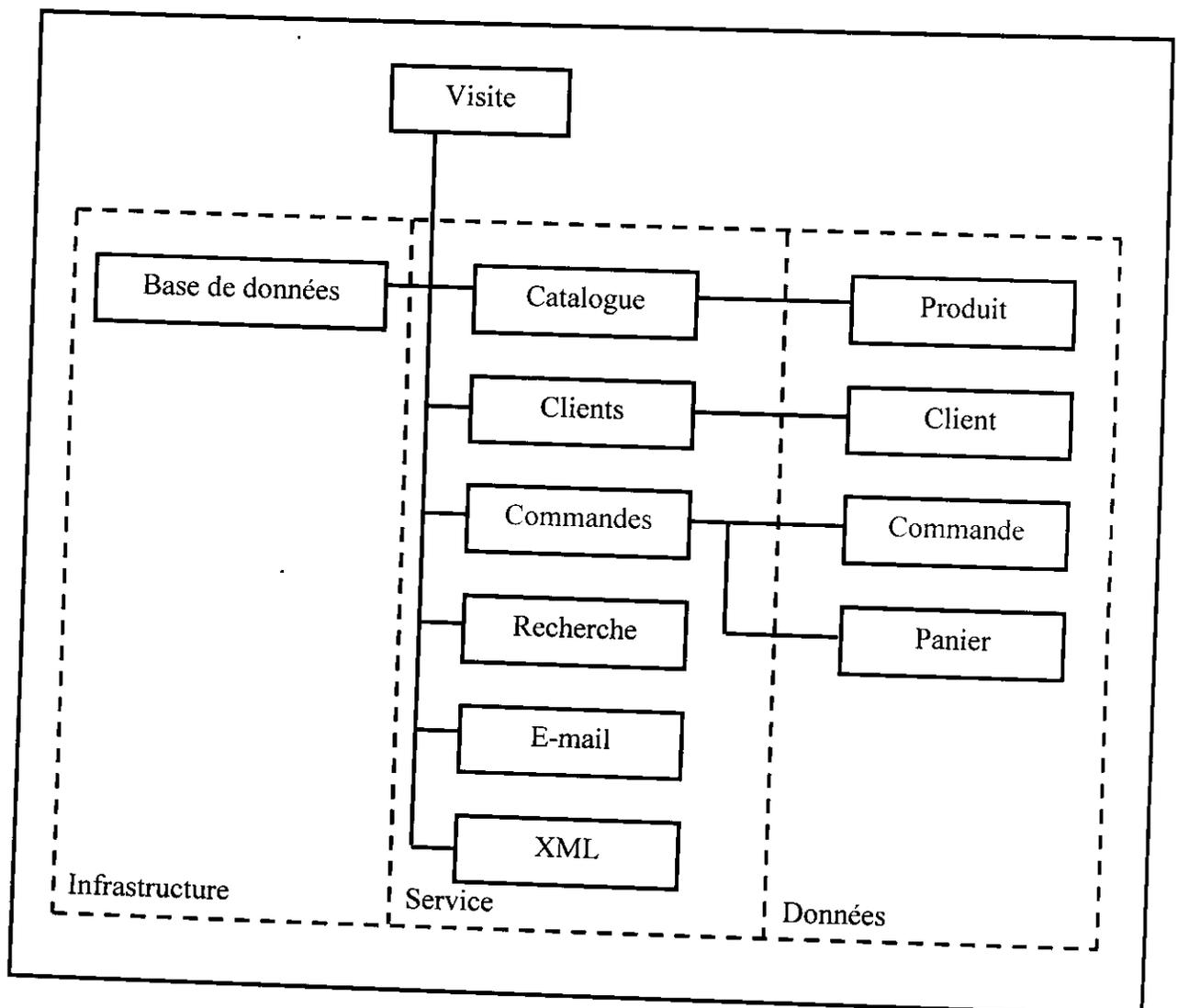


Figure 5.2 : Structure hiérarchique du Modèle Objet CitroenCommerce

### III-5 Conception du modèle :

En développant le niveau métier de notre application, nous allons construire une série d'objet que nous pouvons répartir en trois groupes distincts :

#### ➤ Objet d'infrastructure

- *Base de données* : cet objet simplifie nos communications avec la base de données (qui sont assurées par l'objet ADO *connexion*). Cet objet n'est pas accessible directement depuis le code ASP.

#### ➤ Objet de service

- *Catalogue* : cet objet assure l'accès au catalogue des produits. Il permet d'insérer des rayons et des produits, d'interroger les fournisseurs, les rayons et les produits.

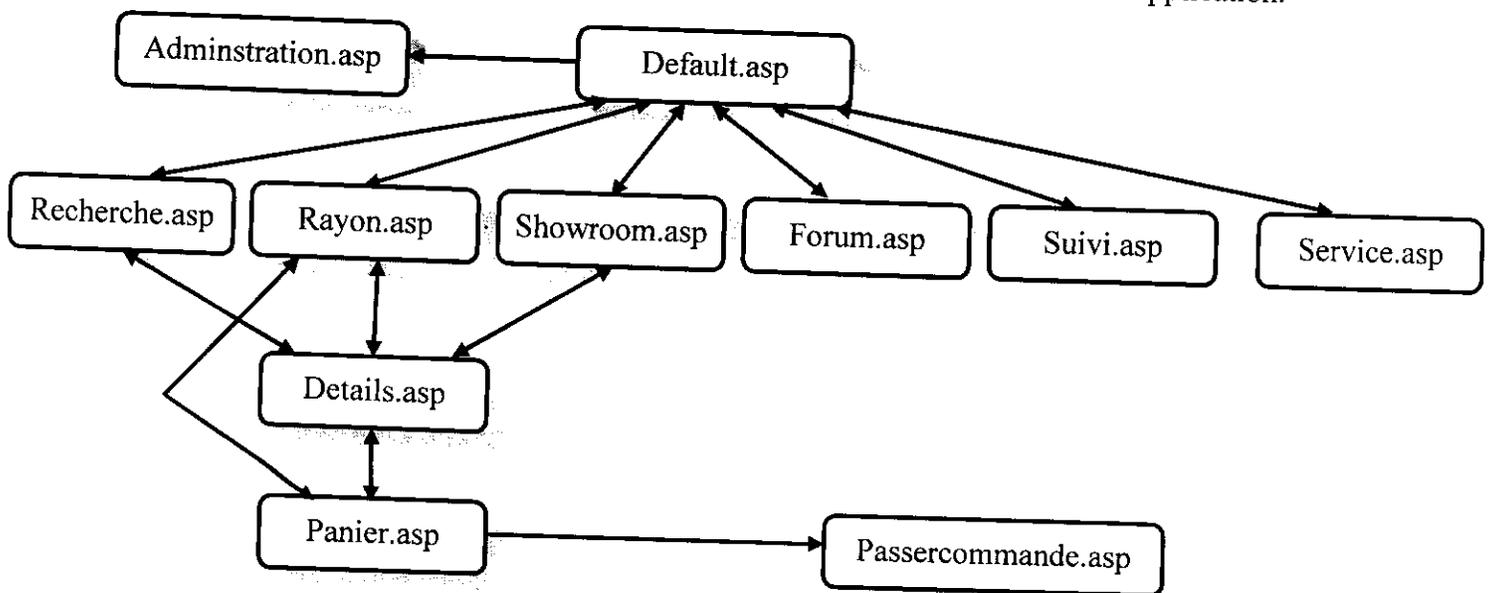
- *Clients* : cet objet gère les clients. Permet de créer de nouveaux clients, gérer les informations relatives à leurs cartes de crédit.
- *Commandes* : cet objet gère les commandes. Son rôle est de transformer un panier d'achat en commande.
- *Rechercher* : cet objet permet d'effectuer des recherches dans la table produits.
- *E-mail* : grâce à cet objet, il est possible d'envoyer des e-mails aux clients à une date et heure données.
- *XML* : cet objet permet de publier des données de notre base données en XML et d'importer des données XML dans notre système.

➤ **Objet de données**

- *Produit* : cet objet représente un produit unique stocké dans la base de données. Il peut renvoyer des informations le concernant.
- *Client* : cet objet représente unique stocké dans la base de données. Il peut renvoyer des informations concernant les cartes de crédit et les commandes qui ont été passées.
- *Commande* : cet objet représente une seule commande stockée dans la base de données.
- *Panier* : cet objet représente un panier unique stocké dans la base de données.

**IV- ENCHAINEMENT DES PAGES WEB DU SITE :**

Dans ce qui suit, nous allons présenter l'enchaînement des pages web de notre site e-commerce, **La figure 5.3** montre la séquence des pages web de notre application.



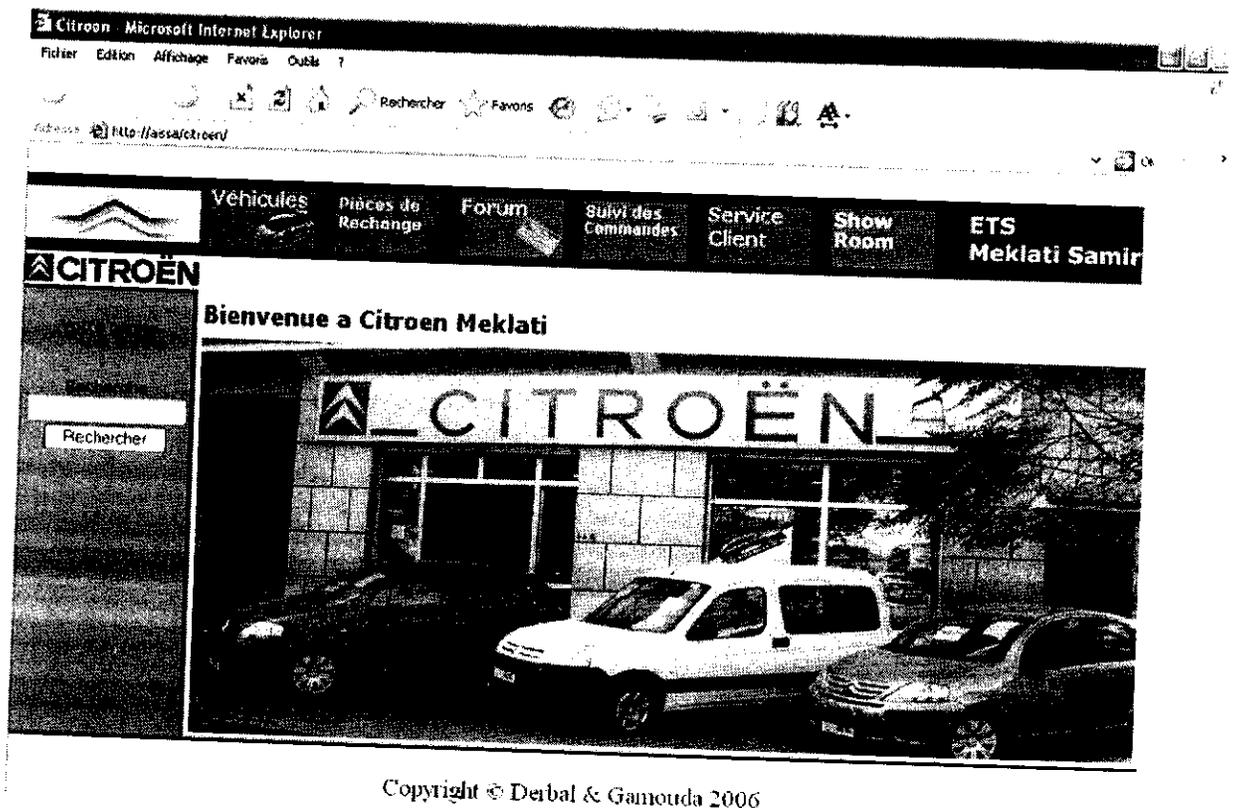
**Figure 5.3 : Enchaînement des pages Web**

## V- LES INTERFACES DU SYSTEME :

### V.1- COTE CLIENT (VISITEUR):

#### V.1.1- Page d'accueil :

La première page qu'un visiteur voit s'afficher lorsqu'il accède sur le site est la page d'accueil, montrée à la **figure 5.4** Elle contient des liens permettant de naviguer librement sur le site et un formulaire de recherche simple, d'où on peut effectuer une recherche simple d'un véhicule ou d'une pièces de rechange par nom.



**Figure 5.4: La page d'accueil**

**V.1.2- formulaire de recherche :** il est présent dans la majorité des pages du site. La **figure 5.5** présente une partie de la page d'accueil qui contient le formulaire. Il offre aux visiteurs du site la possibilité de taper un mot clé, la validation de ce formulaire est réalisée en cliquant sur le bouton « Rechercher ».

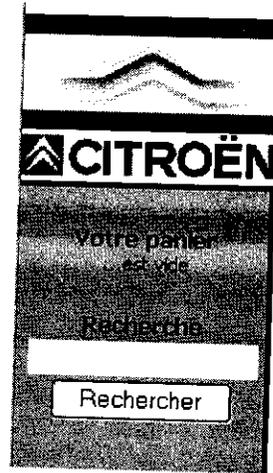


Figure 5.5 : Le formulaire de recherche

**VI- CONSULTATION DES RAYONS :**

Afin de permettre aux visiteurs du site de visualiser l'ensemble des véhicules et pièces de rechange CITRÖEN proposés par le concessionnaire, nous avons divisé nos produits en deux rayons principaux :

- 1- le rayon de niveau supérieur *Véhicules* possède trois sous rayons.
- 2- Le rayon *pièce de rechange*.

Nous avons créé la table Rayons avec les champs suivant :

*IDRayon* : permet l'identification de chaque ligne de la table.

*NomRayon* : permet de stocker le nom du Rayon

*IDPRayon* : sera utilisé pour structurer les rayons en arborescence.

L'ensemble des rayons (véhicules et pièces de rechange) est composé selon la figure 5.6 :

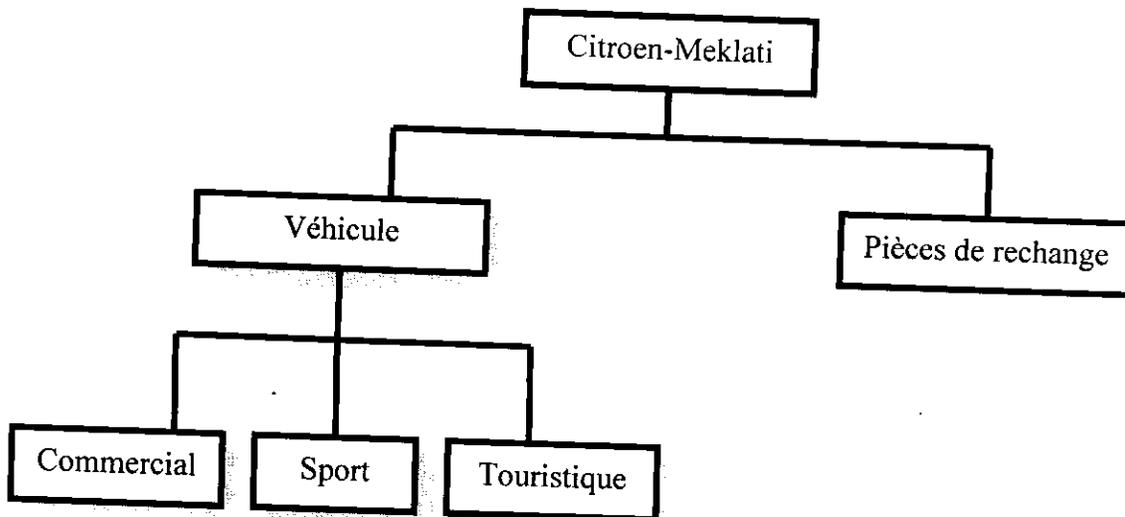


Figure 5.6 : Schéma de structure des rayons

Nous allons expliquer comment le champ IDPRayon nous permet de représenter l'arborescence :

IDRayon	Nom	IDPRayon
4	Commercial	2
5	Sport	2

Ces rayons ont tous deux leur IDPRayon initialisé à 2, ce qui signifie que leur parent est le rayon possédant l'IDRayon 2, dans ce cas Véhicule.

IDRayon	Nom	IDPRayon
2	Véhicule	Null

Enfin, considérons le rayon Véhicule. La valeur de son IDPRayon est de Null, ce qui signifie qu'il se trouve au sommet de l'arborescence des rayons. Notons que le rayon Citroen-meklati n'a pas d'IDRayon. En effet, nous ne créerons pas de rayon spécifique pour ce thème.

Sur la base de ces données, nous pouvons trouver les rayons de même niveau que *Véhicule* au moyen d'une instruction SELECT qui recherche les rayons partageant le même parent que Véhicule :

```
SELECT * FROM Rayons Where IDRayon = 2
```

Nous pouvons de même trouver tous les rayons enfants de Véhicule en recherchant les rayons dont le champ IDPRayon est positionné sur l'IDRayon du rayon Véhicule

```
SELECT * FROM Rayons Where IDPRayon = 2
```

La figure 5.7 présente la page web qui affiche l'ensemble des rayons des véhicules qui est trié selon trois catégories (touristique, commerciale et sport).

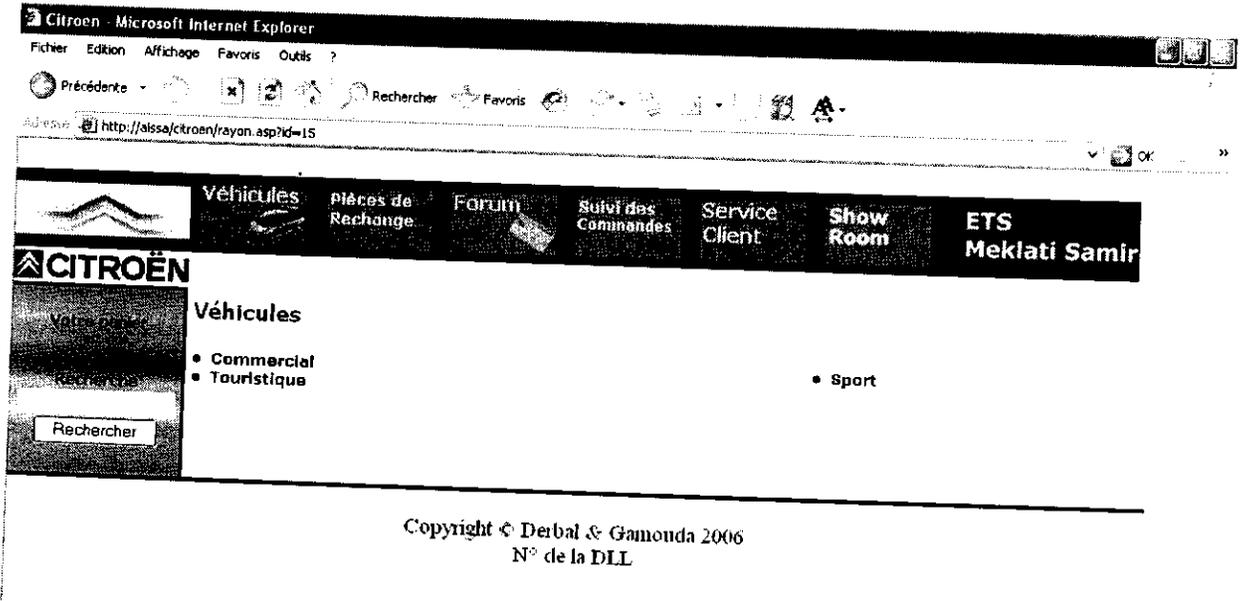


Figure 5.7 : les rayons des véhicules selon catégories

La figure 5-8 : présente la page web qui affiche l'ensemble des véhicules touristique.

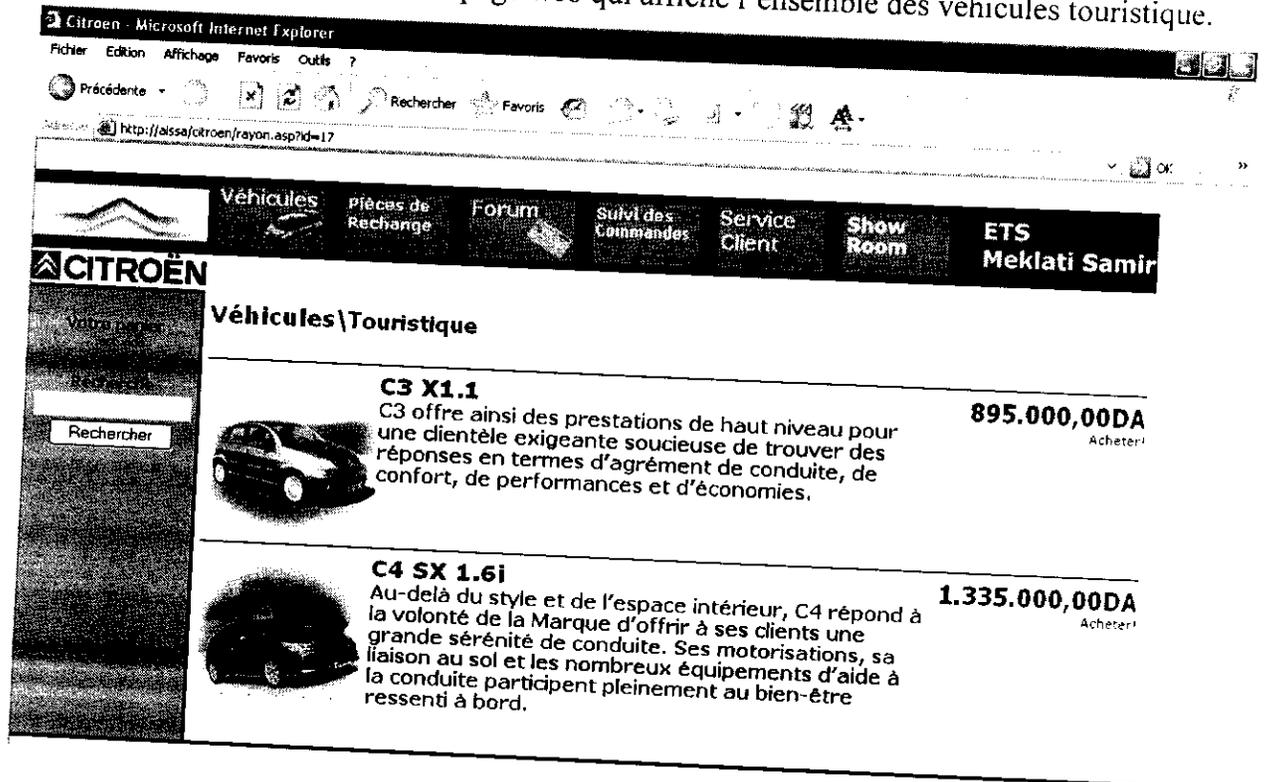


Figure 5.8 : Rayon des véhicules touristique

VII- Le panier d'achat :

Cette page affiche le contenu du panier, où le client peut effectuer ses modifications, ou passer sa commande.

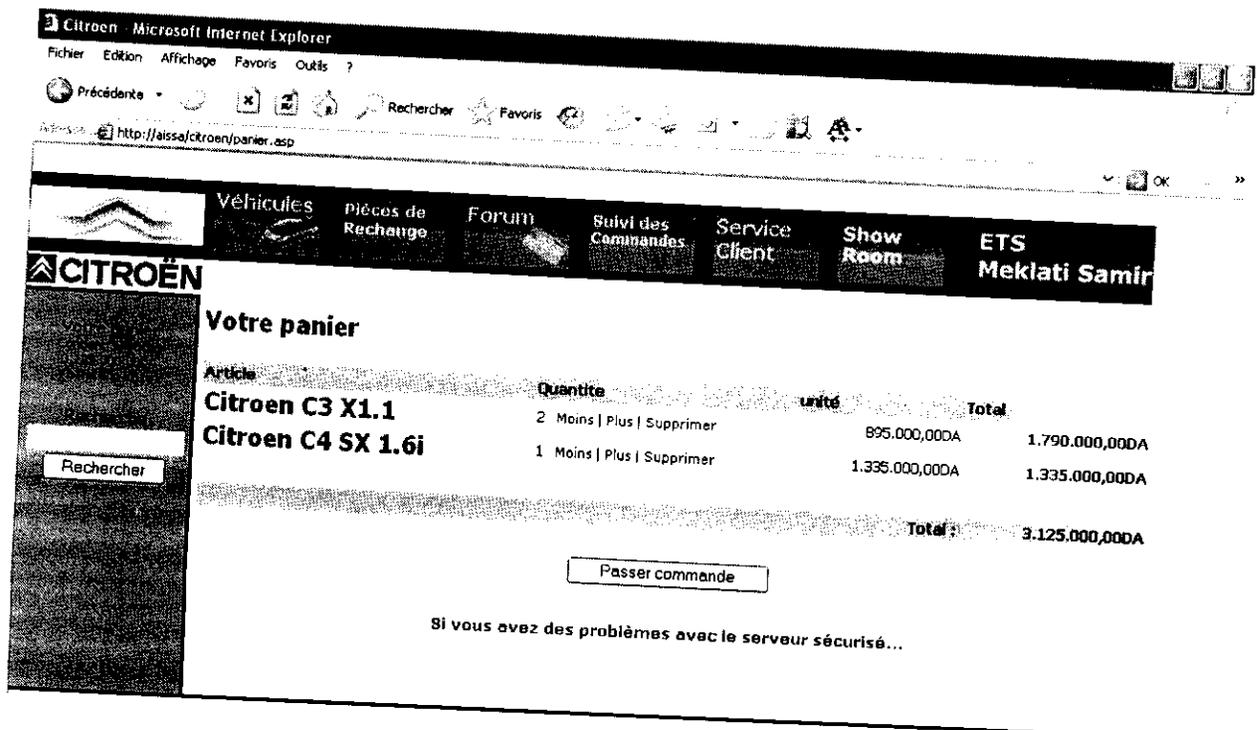


Figure 5.9 : Page Web qui Affiche le contenu du panier

VIII. Passer commande :

La figure 5.10 présente l'un des formulaires qui va être rempli par le client pour passer sa commande et la Figure 5.11 présente la commande finale.

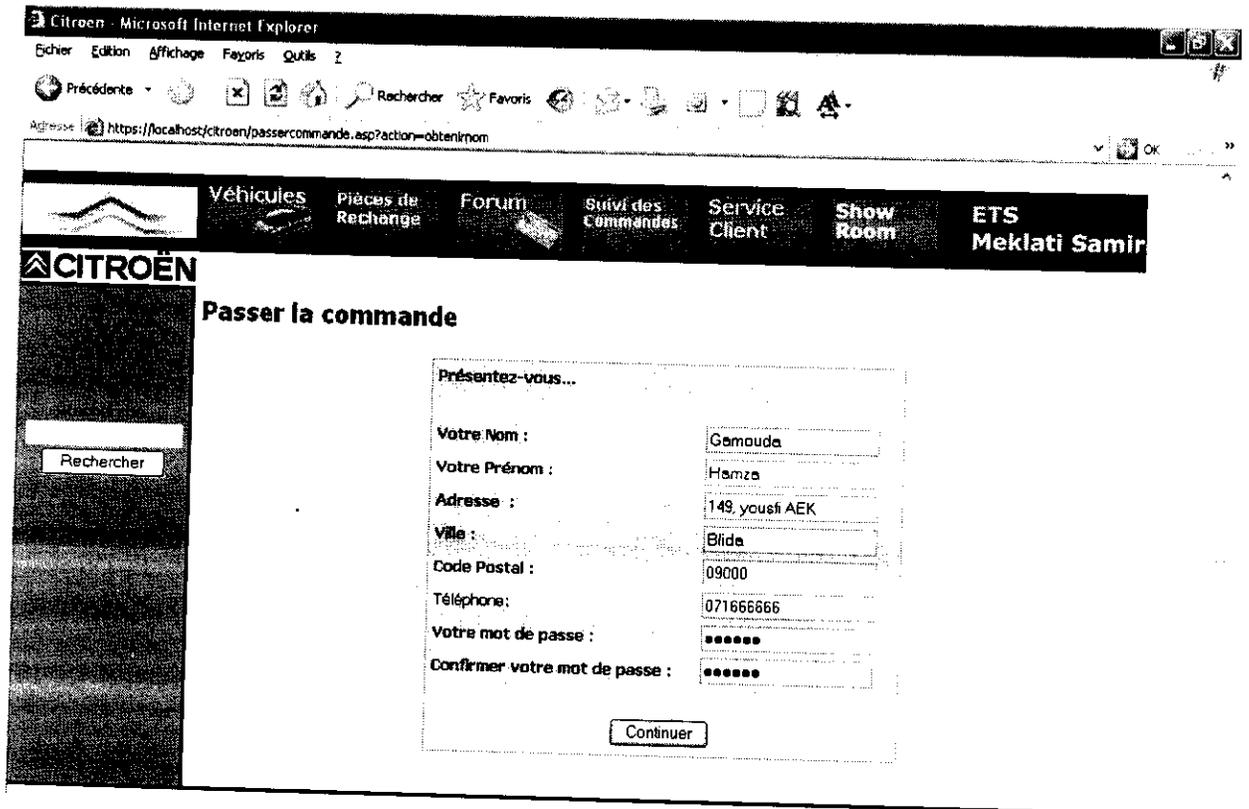


Figure 5.10 : Formulaire pour passer une commande

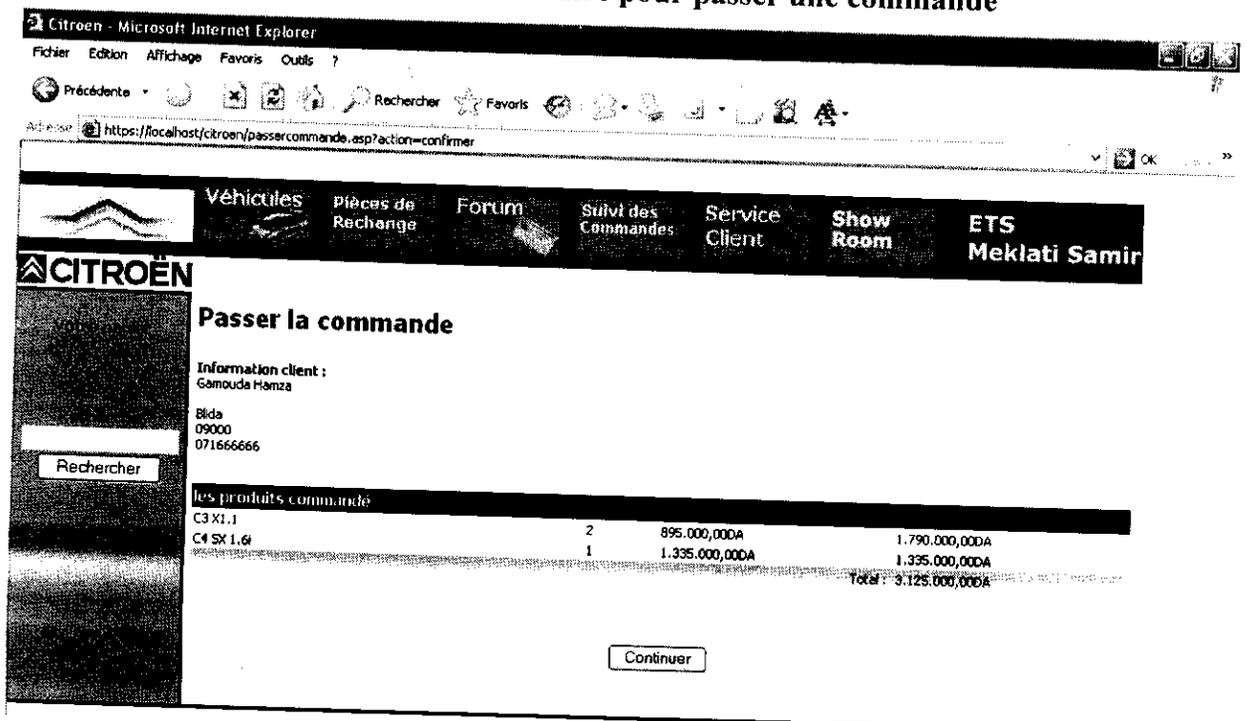


Figure 5.11 : La commande finale



La figure 5.14 présente un formulaire pour la saisie des données concernant la carte de crédit du client.

The screenshot shows a web browser window with the following details:

- Browser: Microsoft Internet Explorer
- Address bar: https://localhost/citroen/passercommande.asp?action=obtenircarte
- Page Title: Passer la commande
- Form Title: Veillez entrer les informations de votre carte de crédit...
- Form Fields:
  - Type de carte : (Selectionner) v
  - Nom inscrit sur la carte :
  - Numéro de la carte :
  - Dates d'expiration :
- Buttons: Rechercher (on the left sidebar), Continuer (at the bottom right of the form)

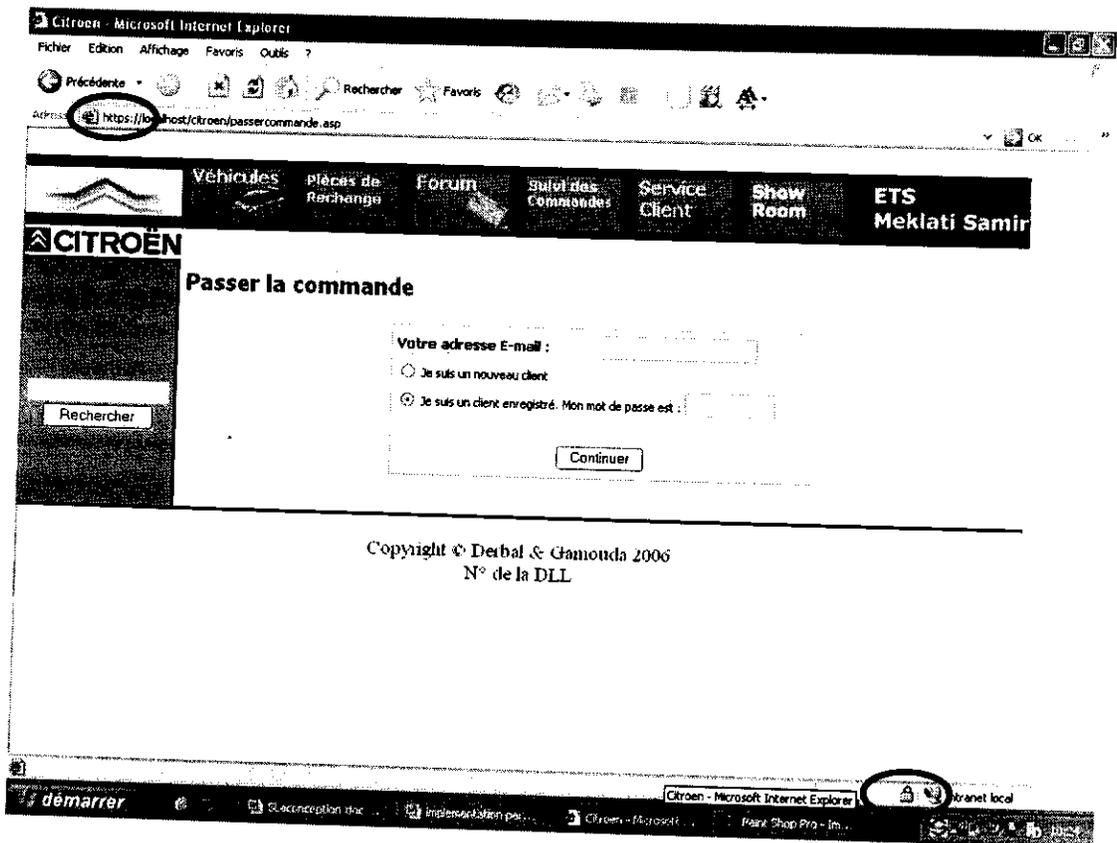
Figure 5.14 : formulaire de carte de crédit

Après la saisie des données, notre système doit communiquer avec un service de paiement électronique, celui-ci communique avec la banque pour valider les informations de la carte de crédit, vérifier le compte du client, transférer l'argent du compte du client vers le compte du concessionnaire, cette partie de notre système n'est pas implémenté parce qu'il n'existe pas encore un service de paiement électronique implanté en Algérie à l'instar des services de paiement installé en France ou aux Etats-Unis (exemple : DataCash, www.datacash.com)

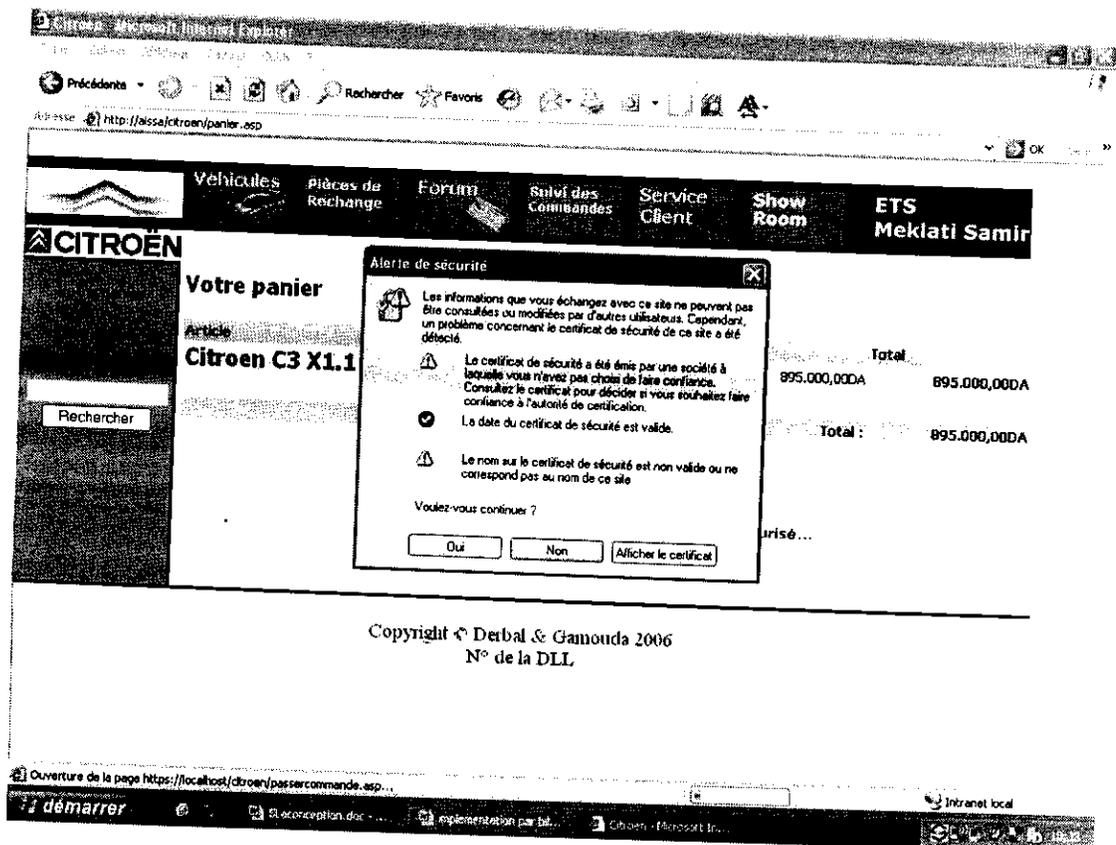
## IX- SECURITE DE PAIEMENT : (LE PROTOCOLE SSL)

Comme nous l'avons mentionné précédemment, lors des transactions commerciales via Internet les utilisateurs doivent fournir des informations personnelles et confidentielles. Pour passer une commande, le client doit saisir les références de sa carte de crédit dans un formulaire, Ces références sont transmises de façon sécurisée.

Les connexions établies avec un serveur web sécurisé peuvent être identifiées grâce à la mention HTTPS qui apparaît dans la chaîne d'URL, contrairement à l'habituel http (le <S> représentant le terme sécurisé), dans notre site e-commerce les connexions HTTPS sont établies à l'aide du protocole **Secure Socket Layer (SSL)**, qui détecte si les données en cours de transmission ont été altérées tout au long de leur parcours via internet.



Lorsque nous connectons à un serveur web sécurisé, il nous envoie un certificat. Une clé publique peut être extraite de ce certificat, puis utilisée dans le processeur de chiffrement, ce qui nous permet de communiquer au serveur web des informations sensibles comme le numéro de carte de crédit de notre client.



Il y a plusieurs organismes de certification qui sont capable de fournir en tout fiabilité une certificat pour sécuriser notre site, nous avons choisir l'organisme VeriSing (www.VeriSing.com) pour obtenir une certificat avec une validité limité.

Une certificat est émis par un organisme de certification pour :

- garantir au visiteur du site que les pages contenant des informations sécurisées par le certificat appartient réellement au site auquel elles prétendent appartenir ;
- fournis une clé de chiffrement utilisé conjointement par le serveur et le navigateur pour créer un canal hautement sécurisé par lequel passent les versions chiffrées du trafic HTTP.

#### **VI- UTILISATION L'XML POUR L'ECHANGE DE DONNEES :**

Intégrer des systèmes informatiques à d'autres systèmes reste une tâche difficile soulevant de nombreux problèmes et se révélant relativement coûteuse, XML a trouvé une large utilisation dans le domaine du commerce électronique B2B et B2C.

On a utilisé l'XML pour l'échange de données entre le concessionnaire et le fournisseur :

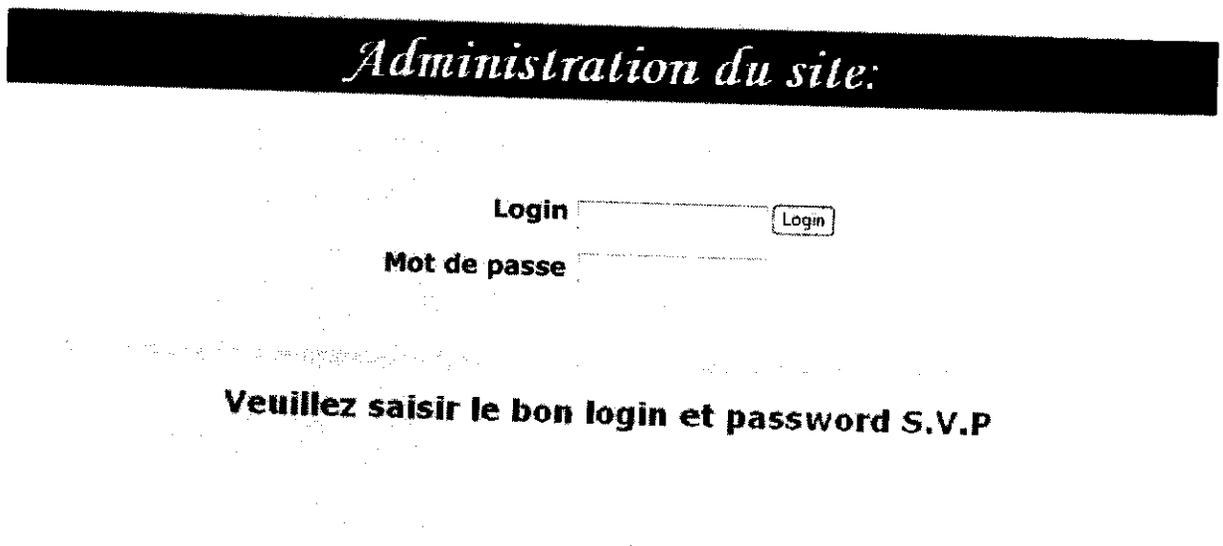
- Le concessionnaire envoie la commande du client vers le fournisseur.
- Le fournisseur informe l'administrateur que les véhicules ou les pièces de rechange commandés sont disponible.
- Le fournisseur envoie la mise à jour des listes des prix concernant les véhicules ou les pièces de rechange.

Comme nous avons motionné précédemment notre composant ActiveX CitroenCommerce comporte l'objet XML qui permet de publier des données de notre base données en documents XML, ces documents sont stocker dans un répertoire sur le serveur contenant la liste des documents XML, cette solution est simple à mettre en œuvre en utilisant le FTP pour envoyer les données dans ce répertoire, en permettant à notre fournisseur de copier les fichiers du répertoire.

**X- COTE ADMINISTRATEUR :****X.1- Accès à la page d'administration du site :**

Cette partie est dédiée uniquement à l'administrateur du site, qui a le droit d'insérer, de supprimer, de mettre à jour l'ensemble du catalogue du site, ainsi que d'autres actions possibles à savoir le changement des paramètres de connexion administrateur (login, mot de passe).

Elle sera accessible par le biais du nom de l'administrateur (Login) et de son mot de passe. La page de la **figure 5.15** permet la saisie et l'accès au menu d'administration.



**Administration du site:**

Login

Mot de passe

**Veuillez saisir le bon login et password S.V.P**

**Figure 5.15: La page de connexion à la session d'administration**

**X.2- Interface de l'ajout d'un nouveau véhicule :**

Parmi les tâches les plus importantes de l'administrateur, c'est la maintenir du catalogue (véhicules et pièces de rechange), par exemple pour ajouter un véhicule l'administrateur doit activer le lien hypertexte « ajouter un produit », une fois c'est fait, un formulaire de saisie s'affiche (voir **Figure 5.16**) permettant ainsi à l'administrateur d'introduire toutes les informations relatives à cet véhicule

La validation de ce formulaire s'effectue par le bouton « Ajouter un produit ».

Adresse [http://eissa/citroen\\_f/admin/default.asp?action=ajouterproduit](http://eissa/citroen_f/admin/default.asp?action=ajouterproduit)

### Citroen Administration

Ajouter un Produit

Nom Produit:

Détails :

Rayon : (Selection) ▼

Type : (Selection) ▼

Prix :

Fournisseur : (Selection) ▼

URL Image :

Description :

Figure 5.16 : La page d'ajout d'une véhicule

#### Conclusion :

Dans ce dernier chapitre nous avons présentées la configuration adoptée pour développer notre propre application Web (langage de script, serveur web et SGBDR), les différents niveau de notre application distribuée orienté web, Nous avons aussi évoquée quelques illustrations de navigation au sein de notre site Web dynamique.

### **Conclusion générale & Perspectives :**

De nos jours, les organisations veulent profiter des avantages cumulés du Web afin d'offrir leurs services en ligne. Dans ce cadre, de nombreux sites Web à caractère commercial ont vu le jour afin de présenter les produits et services au monde extérieur.

Le travail présenté dans ce mémoire illustre l'exploitation de la technologie des bases de données au sein des sites Web dynamiques. Afin d'illustrer les concepts du e-commerce, nous les avons appliqués aux concessionnaires de voiture Meklati (CITRÖEN).

L'objectif de ce projet est de réaliser un site de commerce électronique B2C, avec un moteur de recherche simple et un forum de discussion, nous avons réalisé une base de données afin de stocker la masse d'informations relatives à la gestion du catalogue du produit. Ensuite, nous avons intégré cette même base de données au sein de notre site Web dynamique accessible aux utilisateurs du réseau Internet, nous avons choisi une solution d'échange d'informations et de données simple à mettre en œuvre, cette solution repose sur l'XML, comme format d'échange de données.

Dans ce mémoire, nous avons donné un aperçu général sur Internet, le Web, l'XML et le commerce électronique. Ensuite, nous avons présenté les différentes architectures de développement des applications Web ainsi que les langages de programmation (script) qui sont utilisés. Par la suite, nous avons présenté la démarche que nous avons suivie afin de réaliser notre Application, l'architecture à trois tiers et la configuration logicielle adoptées pour notre site. Enfin, nous avons présenté l'implémentation de notre application de commerce électronique.

Un des points forts de notre travail c'est qu'on a mis en œuvre un squelette de site e-commerce qui peut être appliqué sur d'autres domaines de commerce électronique (exemple : une librairie en ligne).

Néanmoins d'autres actions en perspective peuvent être accomplies afin de compléter notre travail à savoir :

- Améliorer l'aspect visuel du site (design).
- Concevoir et éventuellement implémenter la partie service de paiement, une fois que les techniques bancaires se seront améliorées en Algérie.

## Bibliographie & Webographie

- [AHM 01] : Ahmed Agouzal, 'Le commerce électronique et le e-marketing', Université Aix-marseille III, 2001.
- [ALA 00] : Alain Michard "XML langage et application". Eyrolles décembre 2000.
- [DAV 00] : David DOUSSOT & Pierre-Yves CLOUX & Aurélien GERON, '*Les architectures client-serveur Internet et Intranet*', Dunod, Paris, 2000.
- [DBM 00] : D. Darseyne, B Moulin, '*Le Commerce Electronique*', Projet short, 2000.
- [DEN 05] : DENIO DUARTE "une méthode pour l'évolution de schémas XML préservant la validité des documents". université francois-rabelais-tours juillet 2005.
- [DOU 01] : Douglas COMER, '*TCP/IP Architecture, Protocoles et Applications*', Dunod, Paris, 2001.
- [FMC 02] : François-Marie Colonna, 'L'architecture Client – Serveur', Octobre 2002.
- [FRA 06] : François Pillou, 'tout sur les réseaux et Internet', DUNOD, 2006.
- [FRA 97] : François Laissus, 'Internet Protocol', Novembre 1997.
- [FRE 99] : Fred Gault, Richard St-Pierre, 'Une vérification de la réalité pour définir le commerce électronique', Projet de définitions du commerce électronique, Statistique Canada, 1999.
- [FYV 04] : F.Y. Villemin, 'Le commerce électronique', 2004.
- [GAR 99] : George GARDARIN, '*Internet/intranet et bases de données*', Eyrolles, Paris, 1999.
- [GUY 98] : Guy Vastersavendts, 'Utiliser Internet et ses services', facultés universitaire notre-dame de la paix, Octobre 1998.
- [HUN 01] : D.Hunter, C.Cagle, D.Gibbons, N.Ozu, J.Pinnock, P.Spence "initiation à XML avec trois études de cas détaillées". Eyrolles Janvier 2001.
- [IPS 04] : Ipswitch Inc, 'Guide de la sécurité', Etats-Unis, juin 2004.
- [JEN 05] : Jean François Pillou 'tout sur le webmastering', dunod 2005.

- [JER 03] : Jérôme Daniel, 'Service Web : concepts techniques et outils', Vuibert, Paris, 2003.
- [JFP 06] : Jean François Pillou, 'tout sur les système d'information', DUNOD, 2006.
- [KNA 99] : Kneb.F Jean-christophe Bermadac "construire une application XML avec deux études d cas réels". Eyrolles Juin 1999.
- [MAL 02] : Malika Aboubekr, Suzanne Rivard, 'Commerce Électronique et Conflits de Canaux de Distribution : Un État de la Question', Rapport de Projet, Montréal, Avril 2002.
- [MIC 99] : Michel PONTACQ, *Commerce électronique*, Bull Enterprise Information Systems, 1999.
- [OSC 98] : Oscar Figueiredo, 'Introduction à la Création de Documents pour le World Wide Web HTML 4.0', école polytechnique fédérale de lausanne, avril 1998.
- [PAS IRD] : Pascal RENAUD 'Historique de l'Internet du Nord au Sud', l'Institut de recherche pour le développement, IRD, Genève.
- [RAH 03] : Rahim.S, Afrit.S, " Apport XML pour systèmes d'aides a l'investgation pour les firewalls", 2003.
- [ROB 99] : Robert ORFALI, Dan HARKEY et Jeri EDWARDS, '*Client/serveur*', VUIBERT, Paris, 1999.
- [W3C 06] : <http://www.w3c.org/Xml>, cours et articles, 2006
- [W3O 06]: <http://www.w3.org/Consortium>, 2006.
- [YAC 01] : Yacouba Goita "les applications du XML à la production 'objetpédagogiques interactifs" .université Joseph Fourier Septembre 2001
- [YLE 97] : Y. Lejeune, E. Pivot, D Guillet, *Télépaiement Sécurisé sur Internet*, EPITA, 1997

## Annexe A : LANGAGE UML

Aujourd'hui, le standard industriel de modélisation objet est UML. Il est sous l'entière responsabilité de l'OMG.

### 1. DEFINITION :

UML (Unified Modelling Language) traduit en français langage de modélisation unifié se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML unifié à la fois les notations et les concepts orientés objet, il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage.

UML est le résultat de la combinaison des efforts de trois méthodologies les plus réputés : Grady Booch auteur de la méthode objet Booch, Ivar Jacobson auteur de la méthode objet OOSE et James Rumbaugh auteur de la méthode objet OMT.

#### 1.1 Quelques généralités :

UML est conçu pour modéliser divers types de systèmes, de taille quelconque et pour tous les domaines d'application (gestion, scientifique, temps réel, système embarqué).

UML permet de diviser le système d'information (d'une organisation) en le système métier et le système informatique. Le système métier modélise les aspects statiques et dynamiques de l'activité selon une vision externe et une vision interne (en ignorant l'implémentation technique) tandis que le système informatique recouvre la partie automatisée du système métier concrétisant les choix effectués parmi les différentes technologies d'actualité. Les concepts manipulés sont les mêmes, à chacun de ces deux niveaux d'abstraction.

UML se compose d'une part des éléments de modélisation qui représentent toutes les propriétés du langage et d'autre part des diagrammes (de cas d'utilisation, de classes, d'objets, d'états-transitions, d'activités, de séquence, de collaboration, de composants et de déploiement) qui en constituent l'expression visuelle et graphique.

UML n'impose pas de processus de développement logiciel particulier, même si celui sous-jacent est un processus itératif (précisant à chaque itération les degrés d'abstraction), incrémental (i.e. en divisant le développement en étapes aboutissant chacune à la construction de tout ou partie du système), centré sur l'architecture (au niveau de la modélisation comme de la production), conduit par les cas d'utilisation (modélisant l'application à partir des modes d'utilisation attendus par les utilisateurs), piloté par les risques (afin d'écartier les causes majeures d'échec).

UML prend en compte de manière complètement intégrée l'ingénierie des besoins (cas d'utilisation).

UML est automatisable pour générer du code à partir des modèles vers les langages et les environnements de programmation.

UML est générique, extensible (en plus de couvrir les possibilités des différentes technologies objet existantes) et configurable.

UML se veut intuitif, simple et cohérent.

## **1.2 Historique :**

UML est né en octobre 1994 chez Rational Software Corporation à l'initiative de G. Booch et de J. Rumbaugh.

UML 1.1 a été standardisé par l'OMG (Object Management Group) le 17 novembre 1997 suite à la demande émanant de la collaboration de plusieurs entreprises (Hewlett-Packard, IBM, i-Logix, ICON Computing, IntelliCorp, MCI Systemhouse, Microsoft, ObjecTime, Oracle, Platinum Technology, Ptech, Rational Software Corporation, Reich Technologies, Softeam, Sterling Software, Taskon et Unisys).

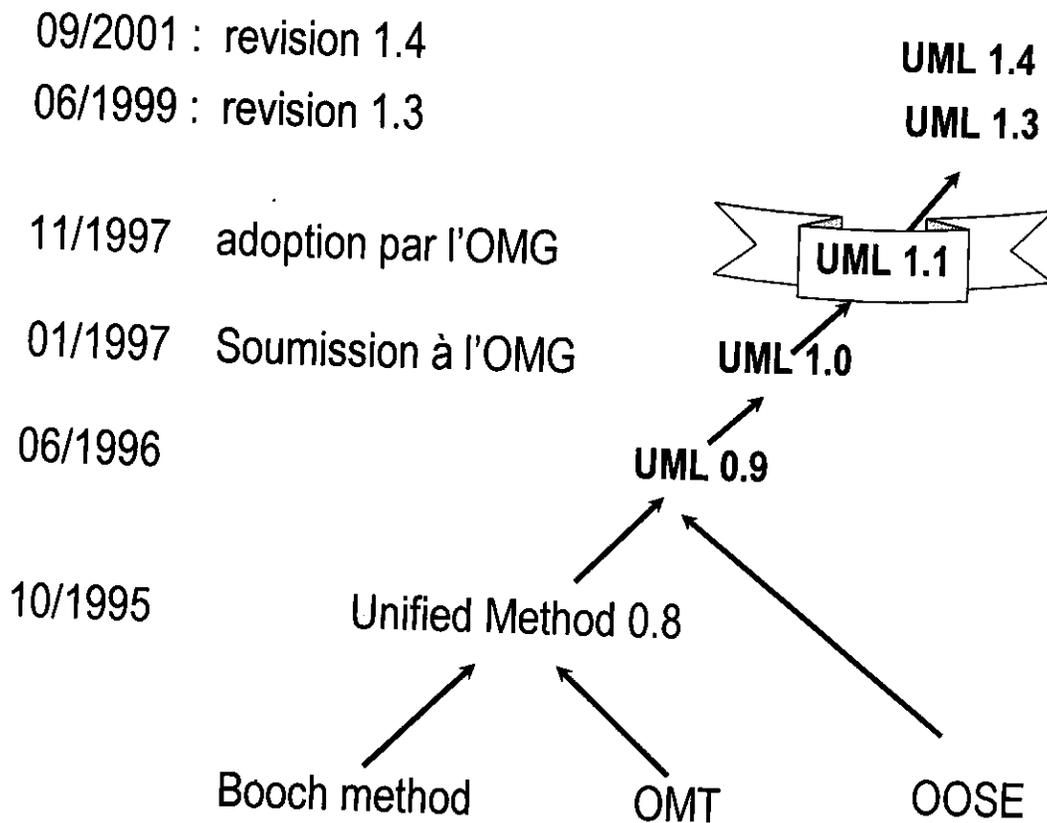


Figure A-1 : Historique d'UML

## 2. LES CONCEPTS DE BASE :

### 2.1 Modèle et Métamodèle :

Pour formaliser UML, tous les concepts ont été eux-mêmes modélisés avec UML. Cette définition récursive, appelée méta modélisation, présente le double avantage de permettre de classer les concepts par niveau d'abstraction, de complexité et de domaine d'application, et aussi de faire la preuve de la puissance d'expression de la notion, capable entre autres de se présenter elle-même.

Le métamodèle d'UML permet de définir :

- Les éléments de modélisation (les concepts manipulés par le langage).
- La sémantique de ces éléments (leur définition et le sens de leur utilisation).

**2.2 Modèle :**

Un modèle est une abstraction de la réalité, l'abstraction est un des piliers de l'approche objet. Il s'agit d'un processus qui consiste à identifier les caractéristiques intéressantes d'une entité en vue d'une utilisation précise.

**2.3 Objet :**

L'objet est une entité atomique formée de l'union d'un état et d'un comportement.

En UML, un objet se représente sous forme d'un rectangle souligné à l'intérieur qui contient soit le nom de l'objet, soit le nom et la classe de l'objet (séparés par un double point), soit seulement le nom de la classe de l'objet précédé par un double point (l'objet est alors dit anonyme).

**2.4 Les classes :**

Une classe est un type abstrait caractérisé par de propriétés (attributs et méthodes) communes à un ensemble d'objets et permet de créer des objets ayant ses propriétés.

En UML, en général, chaque classe est représentée sous forme d'un rectangle divisé en trois compartiments :

- Le nom de la classe.
- Les attributs.
- Les opérations.

Ne pas représenter les attributs et les méthodes d'une classe n'implique pas quelle n'en contient pas.

**2.5 Les packages**

Les packages permettent de regrouper des classes, des associations et même d'autres packages.

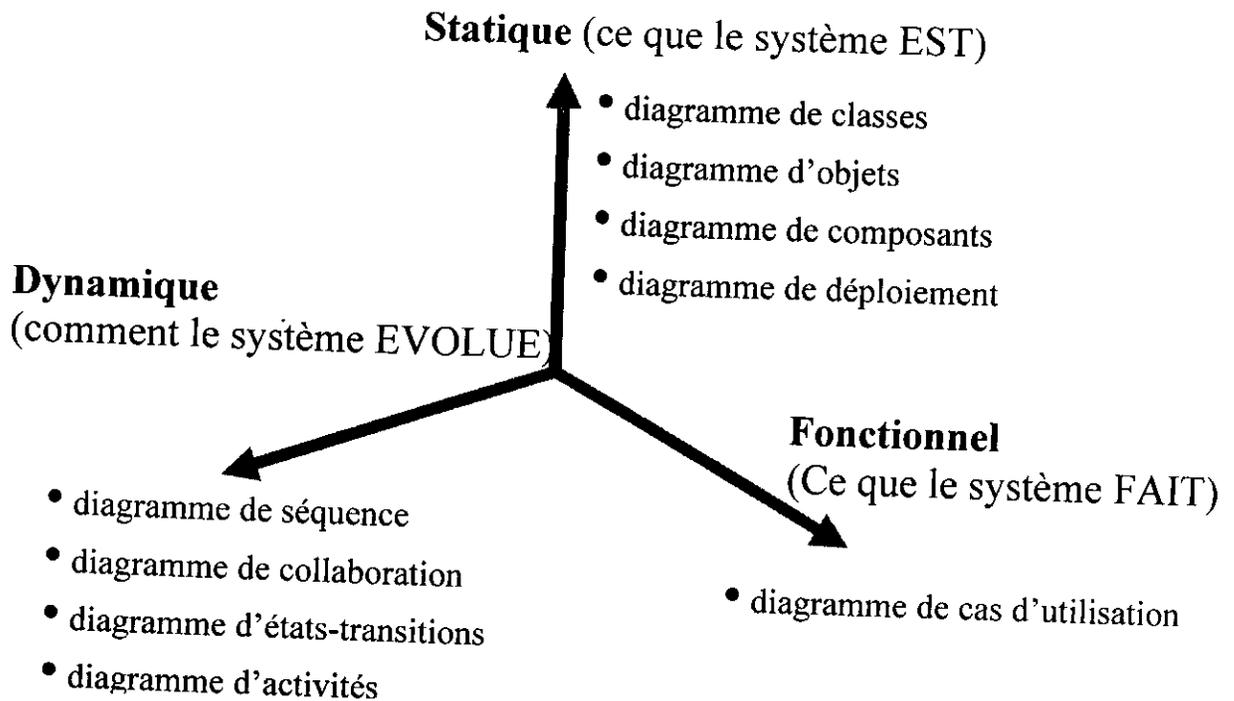
Un package regroupe le plus souvent un ensemble d'entité qui correspond à une fonctionnalité bien définie.

### 3. La modélisation UML :

UML s'articule autour de plusieurs types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particulière d'un système logiciel.

UML modélise le système suivant trois modes de représentation : le premier décrit les services fonctionnels rendus par le système, le second concerne la structure statique du système et la troisième concerne la fonctionnement dynamique du système.

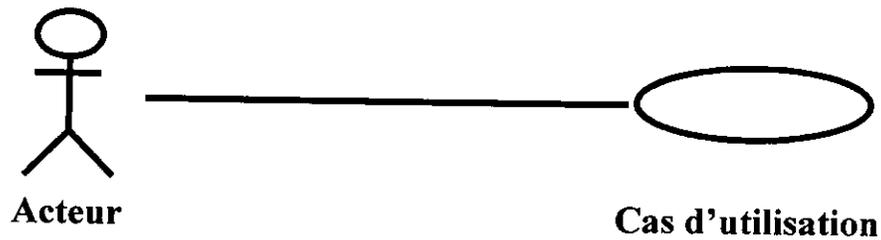
Les trois représentations sont nécessaires et complémentaires pour schématiser la façon dont est composé le système et comment ses composantes fonctionnent entre elles



**Figure A-2 : les 3 axes de modélisation**

Le mode de représentation fonctionnel s'appuie exclusivement sur le diagramme de cas d'utilisation

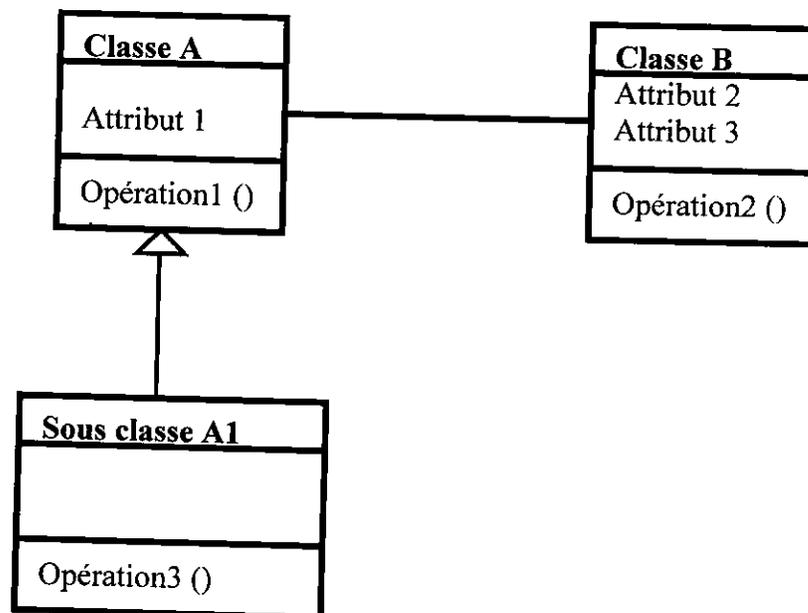
- **Le diagramme de cas d'utilisation** : Il est destiné à représenter les besoin des utilisateurs par rapport au système.



**Figure A-3 : Diagramme de cas d'utilisation**

Le mode de représentation statique ou structurel s'appuie sur les diagrammes ci contre :

- **Le diagramme de classes** : est le point centrale dans un développement orienté objet. En analyse, il a pour objet de décrire la structure des entités manipulées par les utilisateurs, Il contient des classes, des interfaces, des collaborations, et des relations : de dépendance, de généralisation et d'association.



**Figure A-4 : Diagramme de classes**

- **Le diagramme d'objet** : La représentation des instances des classes constitue l'objectif de ce diagramme.

- **Le diagramme de composants** : représente les concepts de configuration logicielle. Il s'agit de montrer comment s'agencent des composants tels que les fichiers source et les packages de code.
- **Le diagramme de déploiement** : correspond à la fois à la structure du réseau informatique qui prend en charge le système logiciel, et à la façon dont les composants y sont installés.

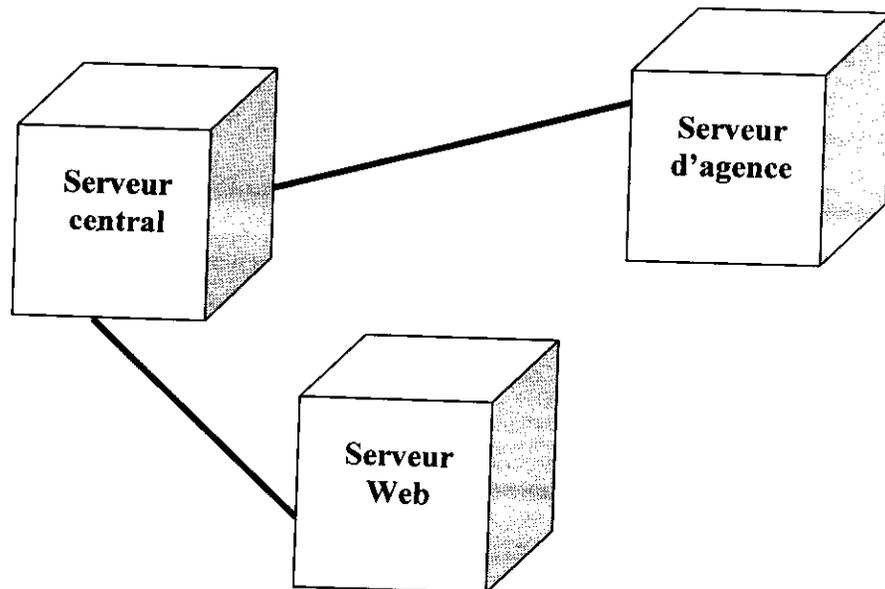


Figure A-5 : Diagramme de déploiement

Le mode de représentation dynamique ou comportemental s'appuie sur les diagrammes ci-après :

- **Le diagramme d'activités** : Ils donnent une version des enchaînements des activités propres à une opération ou à un cas d'utilisation.

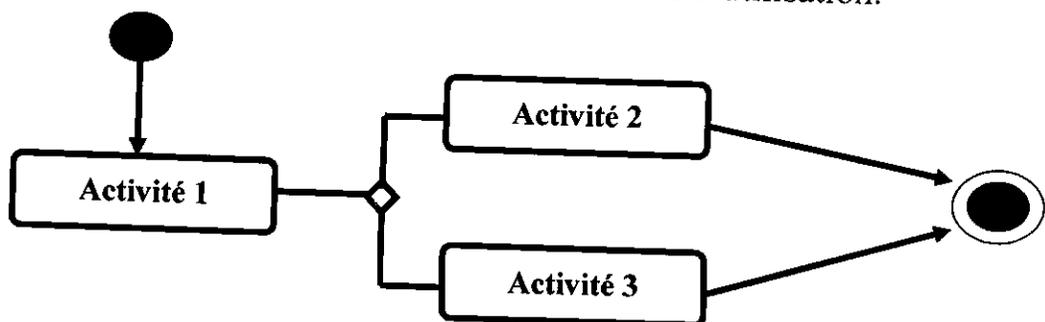


Figure A-6 : Diagramme d'activités

- **Le diagramme d'états-transitions** : représente le cycle de vie commun aux objets d'une même classe

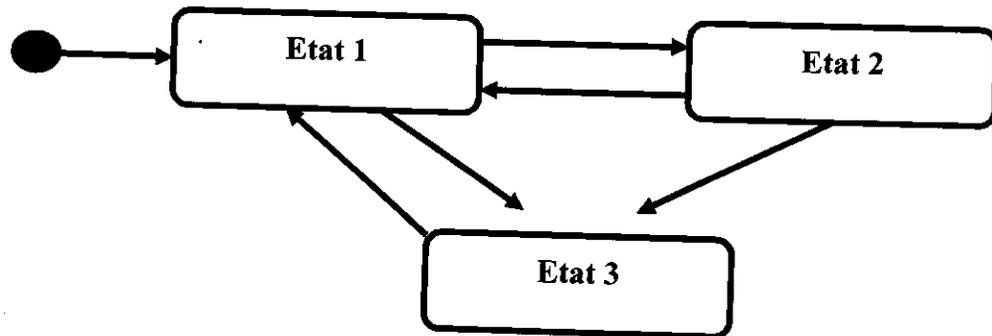


Figure A-7 : Diagramme d'états-transitions

- **Le diagramme de séquence** : Il permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.

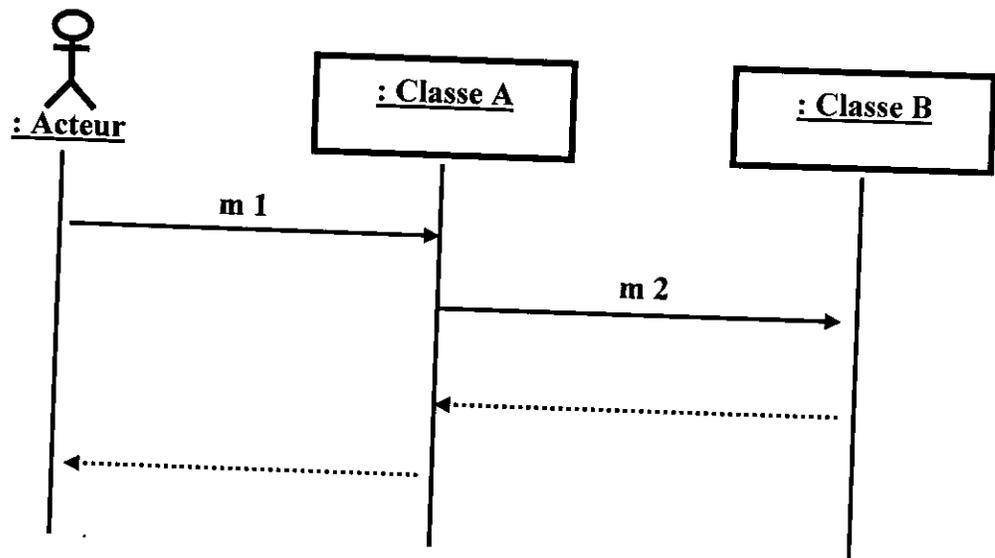


Figure A-8 : Diagramme de séquence

- **Le diagramme de collaboration** : C'est une représentation des scénarios des cas d'utilisation qui mettent plus l'accent sur les objets et les messages échangés.

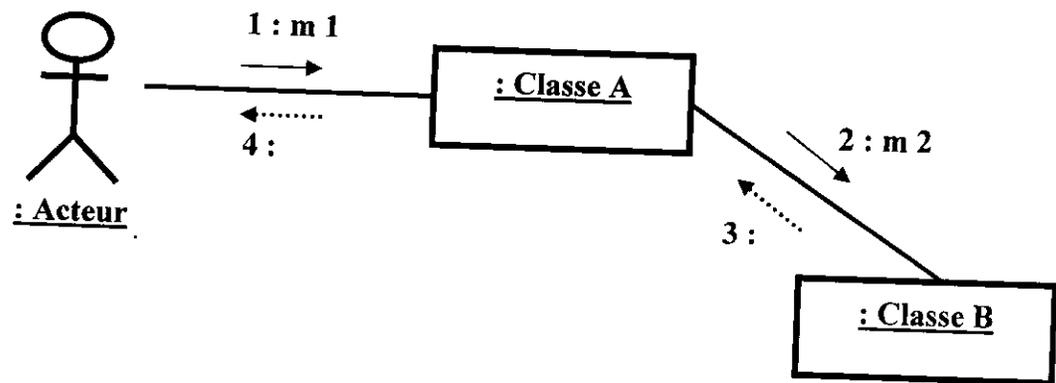


Figure A-9 : Diagramme de collaboration

#### 4. les points forts d'UML :

- UML est un langage formel et normalisé, il permet un gain de précision et un gage de stabilité, ce qui encourage l'utilisation d'outils.
- UML est un support de communication performant, il cadre l'analyse et il facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

#### 5. les points faibles d'UML :

- La mise en pratique d'UML nécessite un apprentissage et passe par un période d'adaptation.
- UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifié les différentes approches et en donne une définition plus formelle.
- Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue.

## Annexe B : Les objets ASP et les chaînes de connexion ODBC, OLE DB

### 1. Description des objets ASP:

#### 1.1 L'objet Request :

L'objet Request représente la requête que fait le navigateur auprès du serveur pour récupérer la page ASP. Il rassemble dans différentes collections le message HTTP que le client a envoyé au serveur. Par exemple, cet objet contiendra dans une collection spéciale le contenu du formulaire que l'on aura envoyé.

#### 1.2 L'objet Response :

L'objet Response sert à représenter la réponse HTTP du serveur à la requête du client. Ce sont d'une part les entêtes de la réponse (les statuts HTTP par exemple), et le corps de la réponse (la partie de la réponse contenant les données intéressantes) d'autre part. C'est cet objet qui permet d'écrire dynamiquement du code HTML à partir du code ASP avec notamment la méthode **Write**.

#### 1.3 L'objet Application :

L'objet Application représente l'application Web au sens large. C'est un gros container qui contient des données dynamiques valables pour tous les visiteurs du site à tout instant (le nombre de visiteurs en cours par exemple). Pour schématiser, l'objet Application contient les variables globales de l'application.

L'objet Application peut fonctionner comme une collection : on peut ajouter dynamiquement des éléments de l'une des deux façons suivantes :

```
Application ("prénom") = "Bille" ' valeur scalaire
```

```
Set Application("connexion") = Server.CreateObject("ADODB.Connection") ' objet
```

Deux événements **OnEnd** et **OnStart** permettent d'exécuter des actions (comme par exemple initialiser des variables) au moment du démarrage ou de l'arrêt de l'application.

#### 1.4 L'objet Session :

Dans le langage ASP, une session représente une période de temps dans laquelle un visiteur navigue dans un ensemble de pages ASP. L'objet Session permet de définir des variables, des constantes, des procédures à la manière de l'objet Application. Mais son utilisation se situe au niveau de l'utilisateur, c'est-à-dire de la personne qui arrive sur votre site. Grâce à l'objet Session, le serveur web se souvient de vous.

L'objet Session stocke des informations pour une session particulière du visiteur comme par exemple, la liste des articles qu'il est en train de commander sur un site marchand. Ces informations sont valables pour une durée déterminée (qui peut être infinie). Par défaut, les sessions expirent au bout de 20 minutes. Son ouverture est assurée automatiquement par le serveur.

La syntaxe de l'affectation d'une valeur à une variable dans l'objet Session est la suivante :

```
Session ("Nom-Variable")=Variable.
```

```
Session (pays) =ALGERIE
```

```
Set Session ("connexion") = Server.CreateObject ("ADODB.Connection") ' objet
```

**Note** : pour changer la durée d'expiration par défaut d'une session on utilise la propriété :**TimeOut** de l'objet Session.

Syntaxe : Session.TimeOut = "la durée" (en minutes).

**1.5 L'objet Server :**

L'objet Server représente le serveur Web et le moteur ASP. Cet objet permet de gérer quelques paramètres du serveur Web et instancier les objets de l'utilisateur.

Exemple : *DIM Enregistrement*

```
Set Enregistrement = Server.CreateObject ("ADODB.Recordset")
```

**1.6 L'objet ASPError :**

L'objet ASPError permet de recueillir des informations sur les erreurs survenues lors de l'exécution du code ASP de la page.

On repère un objet AspError en utilisant la méthode **Server.GetLastError**

**2. Description des objets et collections ADO :**

Objet ou Collection	Utilisation
Objet Connection	Permet d'établir les connexions entre le client et la source de données (la base de données).
Objet Command .	Permet de réaliser des commandes, telles que des requêtes SQL ou des mises à jour d'une base.
Objet Recordset	Permet de voir et de manipuler les résultats d'une requête.
Collection Parameters	Est utilisée lorsque la requête de l'objet Command nécessite des paramètres.
Collection Errors	La collection Errors et l'objet Error sont accédés au travers de l'objet Connection, lorsqu'une erreur est générée. Il permet donc d'obtenir des informations précises sur la cause de l'erreur.
Collection Fields .	La collection Fields et l'objet Field sont accédés au travers de l'objet Recordset, une fois que celui-ci contient des données.

### 3. Chaîne de connexion ODBC :

•**SQLServer:**

DRIVER = {SQLServer}; server = NomServeur; uid = nonDusager;  
pwd = motDePasse; Database = baseDeDonnees;

•**ACCESS:**

DRIVER = {Microsoft Access Driver(\*.mdb); DBQ=D:\inetpub\db\base.mdb;  
Uid = nonDusager; pwd = motDePasse;

•**Oracle:**

DRIVER = {Microsoft ODBC forOracle}; Server=OracleServer.world;  
Uid = nonDusager; pwd = motDePasse; database = baseDeDonnees;

•**MySQL:**

DRIVER = {mySQL}; server = CheminMonServeur; Option = 16834;  
Database = baseDeDonnées;

### 4. Chaînes de connexion OLE DB :

**SQL Server:**

Provider = SQLOLEDB.1; DataSource = Nom du Serveur; Initial  
Catalog = NomBaseDeDonnées;  
User ID = nonDusager; Password = motDePasse;

**ACCESS:**

Provider = MICROSOFT.JET.OLEDB.4.0; DataSource = chemin de la base ;  
User ID=nom d'utilisateur ; Password = mot de passe;

**Oracle:**

Provider = OraOLEDB.Oracle; DataSource = chemin d'accès à la base;  
User ID=nom D'utilisateur; Password = mot de passe;

## Annexe C : Scripte SQL

```

*****
*****
                                création de la table RAYONS
                                *****
CREATE TABLE [dbo].[RAYONS] (
    [IDRAYON] [int] IDENTITY (1, 1) NOT NULL ,
    [IDPRAYON] [int] NULL ,
    [NOM_rayon] [varchar] (64) COLLATE French_CI_AS NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[RAYONS] WITH NOCHECK ADD
    PRIMARY KEY CLUSTERED
    (
        [IDRAYON]
    ) ON [PRIMARY]
GO

```

```

*****
*****
                                création de la table LIGNESPANIER
                                *****
CREATE TABLE [dbo].[lignespanier] (
    [idarticlepanier] [int] IDENTITY (1, 1) NOT NULL ,
    [idpanier] [int] NOT NULL ,
    [idproduit] [int] NOT NULL ,
    [quantite] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[articlespanier] WITH NOCHECK ADD
    PRIMARY KEY CLUSTERED
    (
        [idarticlepanier]
    ) ON [PRIMARY]
GO

```

```

*****
*****
                                création de la table ATTRIBUTS
                                *****
CREATE TABLE [dbo].[attributs] (
    [idattribut] [int] IDENTITY (1, 1) NOT NULL ,
    [idstructure] [int] NOT NULL ,
    [idproduit] [int] NOT NULL ,
    [valeurlong] [int] NULL ,
    [valeurchaine] [varchar] (1024) COLLATE French_CI_AS NULL ,
    [valeurdouble] [float] NULL ,
    [valeurdate] [datetime] NULL ,

```

```

    [valeurbooleen] [tinyint] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[attributs] WITH NOCHECK ADD
    PRIMARY KEY CLUSTERED
    (
        [idattribut]
    ) ON [PRIMARY]
GO

```

```

*****
*****
                création de la table CARTES
*****

```

```

CREATE TABLE [dbo].[cartes] (
    [idcarte] [int] IDENTITY (1, 1) NOT NULL ,
    [idclient] [int] NOT NULL ,
    [type] [varchar] (16) COLLATE French_CI_AS NOT NULL ,
    [numero] [varchar] (20) COLLATE French_CI_AS NOT NULL ,
    [nomsurcarte] [varchar] (64) COLLATE French_CI_AS NOT NULL ,
    [moisexpiration] [int] NOT NULL ,
    [anneeexpiration] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[cartes] WITH NOCHECK ADD
    PRIMARY KEY CLUSTERED
    (
        [idcarte]
    ) ON [PRIMARY]
GO

```

```

*****
*****
                création de la table CLIENTS
*****

```

```

CREATE TABLE [dbo].[clients] (
    [idclient] [int] IDENTITY (1, 1) NOT NULL ,
    [prenom_client] [varchar] (32) COLLATE French_CI_AS NULL ,
    [nom_client] [varchar] (32) COLLATE French_CI_AS NULL ,
    [email_client] [varchar] (64) COLLATE French_CI_AS NULL ,
    [motdepasse] [varchar] (32) COLLATE French_CI_AS NULL ,
    [adresse] [varchar] (64) COLLATE French_CI_AS NULL ,
    [ville] [varchar] (32) COLLATE French_CI_AS NULL ,
    [codepostal] [int] COLLATE French_CI_AS NULL ,
    [telephone] [varchar] (12) COLLATE French_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[clients] WITH NOCHECK ADD
    PRIMARY KEY CLUSTERED
    (

```

```

        [idclient]
    ) ON [PRIMARY]
GO
ALTER TABLE [dbo].[clients] WITH NOCHECK ADD
    CONSTRAINT [DF__clients__creee__0DAF0CB0] DEFAULT (getdate()) FOR [creee]
GO

```

```

*****
***** création de la table COMMANDES *****

```

```

CREATE TABLE [dbo].[commandes] (
    [idcommande] [int] IDENTITY (1, 1) NOT NULL ,
    [idclient] [int] NOT NULL ,
    [idcarte] [int] NULL ,
    [creee] [datetime] NOT NULL ,
    [achevee] [datetime] NULL ,
    [etat] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[commandes] WITH NOCHECK ADD
    PRIMARY KEY CLUSTERED
    (
        [idcommande]
    ) ON [PRIMARY]
GO
ALTER TABLE [dbo].[commandes] WITH NOCHECK ADD
    CONSTRAINT [DF__commandes__creee__164452B1] DEFAULT (getdate()) FOR
[creee],
    CONSTRAINT [DF__commandes__etat__173876EA] DEFAULT (0) FOR [etat]
GO

```

```

*****
***** création de la table E-mail *****

```

```

CREATE TABLE [dbo].[fireandforget] (
    [idfire] [int] IDENTITY (1, 1) NOT NULL ,
    [idclient] [int] NOT NULL ,
    [sujet] [varchar] (64) COLLATE French_CI_AS NOT NULL ,
    [Message] [varchar] (4096) COLLATE French_CI_AS NOT NULL ,
    [dateprevue] [datetime] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[fireandforget] WITH NOCHECK ADD
    CONSTRAINT [PK_fireandforget] PRIMARY KEY CLUSTERED
    (
        [idfire]
    ) ON [PRIMARY]
GO

```

```

*****
*****
***** création de la table FOURNISSEURS *****
CREATE TABLE [dbo].[fournisseurs] (
    [idfournisseur] [int] IDENTITY (1, 1) NOT NULL ,
    [nom_fournisseur] [varchar] (64) COLLATE French_CI_AS NULL ,
    [url_fournisseur] [varchar] (64) COLLATE French_CI_AS NULL ,
    [premcontactcom] [varchar] (32) COLLATE French_CI_AS NULL ,
    [derncontactcom] [varchar] (32) COLLATE French_CI_AS NULL ,
    [emailcontactcom] [varchar] (64) COLLATE French_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[fournisseurs] WITH NOCHECK ADD
    PRIMARY KEY CLUSTERED
    (
        [idfournisseur]
    ) ON [PRIMARY]
GO

```

```

*****
*****
***** création de la table LIGNESCOMMANDE *****
CREATE TABLE [dbo].[lignescommande] (
    [idligne] [int] IDENTITY (1, 1) NOT NULL ,
    [idproduit] [int] NULL ,
    [quantite] [int] NULL ,
    [prixunite] [float] NULL ,
    [total] [float] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[lignescommande] WITH NOCHECK ADD
    PRIMARY KEY CLUSTERED
    (
        [idligne]
    ) ON [PRIMARY]
GO

```

```

*****
*****
***** création de la table PANIERS *****
CREATE TABLE [dbo].[paniers] (
    [idpanier] [int] IDENTITY (1, 1) NOT NULL ,
    [delaisexpiration] [datetime] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[paniers] WITH NOCHECK ADD
    PRIMARY KEY CLUSTERED
    (

```

```

        [idpanier]
    ) ON [PRIMARY]
GO
ALTER TABLE [dbo].[paniers] WITH NOCHECK ADD
    CONSTRAINT [DF__paniers__delaise__07F6335A] DEFAULT (getdate() + 1) FOR
[delaisexpiration]
GO

```

```

*****
*****
*****      création de la table PRODUITS      *****
*****

```

```

CREATE TABLE [dbo].[produits] (
    [idproduit] [int] IDENTITY (1, 1) NOT NULL ,
    [nom_produit] [varchar] (64) COLLATE French_CI_AS NOT NULL ,
    [details] [varchar] (256) COLLATE French_CI_AS NOT NULL ,
    [description] [text] COLLATE French_CI_AS NULL ,
    [idrayon] [int] NOT NULL ,
    [idtype] [int] NOT NULL ,
    [prix] [float] NOT NULL ,
    [idfournisseur] [int] NOT NULL ,
    [urlimage] [varchar] (128) COLLATE French_CI_AS NULL ,
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [dbo].[produits] WITH NOCHECK ADD
    CONSTRAINT [PK__produits__78B3EFCA] PRIMARY KEY CLUSTERED
    (
        [idproduit]
    ) ON [PRIMARY]
GO

```

```

*****
*****
*****      création de la table STRUCTURE ATTRIBUTION      *****
*****

```

```

CREATE TABLE [dbo].[structureattribution] (
    [idstructure] [int] IDENTITY (1, 1) NOT NULL ,
    [idtype] [int] NOT NULL ,
    [nom_structure] [varchar] (64) COLLATE French_CI_AS NOT NULL ,
    [typedonnee] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[structureattribution] WITH NOCHECK ADD
    PRIMARY KEY CLUSTERED
    (
        [idstructure]
    ) ON [PRIMARY]

```

GO

\*\*\*\*\*  
 \*\*\*\*\*  
**création de la table TYPES** \*\*\*\*\*

```
CREATE TABLE [dbo].[types] (
  [idtype] [int] IDENTITY (1, 1) NOT NULL ,
  [nom_type] [varchar] (32) COLLATE French_CI_AS NULL
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[types] WITH NOCHECK ADD
  PRIMARY KEY CLUSTERED
```

```
(
  [idtype]
) ON [PRIMARY]
GO
```

\*\*\*\*\*  
 \*\*\*\*\*  
**création de la table ADMINISTRATEURS** \*\*\*\*\*

```
CREATE TABLE [dbo].[administrateurs] (
  [idadministrateur] [int] IDENTITY (1, 1) NOT NULL ,
  [login] [varchar] (32) COLLATE French_CI_AS NULL ,
  [motdepasse] [varchar] (32) COLLATE French_CI_AS NULL
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[administrateurs] WITH NOCHECK ADD
  PRIMARY KEY CLUSTERED
```

```
(
  [idadministrateur]
) ON [PRIMARY]
GO
```

\*\*\*\*\*  
 \*\*\*\*\*  
**création de la vue VPRODUITS** \*\*\*\*\*

```
CREATE VIEW VPRODUITS AS
SELECT FOURNISSEURS.DERNCONTACTCOM,
FOURNISSEURS.EMAILCONTACTCOM,
FOURNISSEURS.NOM_FOURNISSEUR, FOURNISSEURS.PREMCONTACTCOM,
FOURNISSEURS.URL_FOURNISSEUR, PRODUITS.DESCRPTION,
PRODUITS.DETAILS, PRODUITS.IDFOURNISSEUR, PRODUITS.IDPRODUIT,
PRODUITS.IDRAYON, PRODUITS.IDTYPE,
PRODUITS.NOM_PRODUT, PRODUITS.PRIX, PRODUITS.URLIMAGE,
RAYONS.IDPRAYON, RAYONS.NOM_RAYON FROM Produits
fournisseurs ON Produits.IDfournisseur = fournisseurs.iDfournisseur INNER JOIN
```

rayons ON  
Produits.IDDrayon = rayons.IDDrayon

```

*****
*****
                création de la vue VATTRIBUTS
CREATE VIEW VATTRIBUTS AS
SELECT ATTRIBUTS.IDPRODUIT, ATTRIBUTS.VALEURdate,
ATTRIBUTS.VALEURBOOLEEN, ATTRIBUTS.VALEURCHAINE,
ATTRIBUTS.VALEURDOUBLE, ATTRIBUTS.VALEURLONG,
STRUCTUREATTRIBUTION.IDSTRUCTURE,
STRUCTUREATTRIBUTION.IDTYPE,
STRUCTUREATTRIBUTION.NOM_STRUCTURE,
STRUCTUREATTRIBUTION.TYPEDONNEE
FROM STRUCTUREATTRIBUTION LEFT OUTER JOIN
ATTRIBUTS ON
STRUCTUREATTRIBUTION.IDStructure = Attributs.IDStructure

```

```

*****
*****
                création de la vue VE-mail
CREATE VIEW dbo.ve-mail
AS
SELECT  dbo.clients.prenom_client, dbo.clients.nom_client, dbo.clients.email_client, dbo.
e-mail.idfire, dbo. e-mail.idclient, dbo. e-mail.sujet,
        dbo. e-mail.Message, dbo. e-mail.dateprevue
FROM    dbo. e-mail INNER JOIN
        dbo.clients ON dbo. e-mail.idclient = dbo.clients.idclient

```

```

*****
*****
                création de la vue VARTICLESPANIER
CREATE VIEW varticlespanier AS
SELECT  articlespanier.idarticlepanier, articlespanier.idpanier, articlespanier.idproduit,
articlespanier.quantite,
        VPRODUITS.NOM_FOURNISSEUR, VPRODUITS.DERNCONTACTCOM,
        VPRODUITS.EMAILCONTACTCOM, VPRODUITS.PREMCONTACTCOM,
        VPRODUITS.IDFOURNISSEUR, VPRODUITS.URLIMAGE,
        VPRODUITS.NOM_PRODUI,
        VPRODUITS.DETAILS, VPRODUITS.PRIX, articlespanier.quantite *
        VPRODUITS.PRIX AS ligneprix,
        articlespanier.quantite * VPRODUITS.COUT AS ligne cout
FROM    VPRODUITS INNER JOIN
        articlespanier ON VPRODUITS.IDPRODUIT = articlespanier.idproduit

```

## Résumé

Le présent travail a pour objectif la conception et la réalisation d'un site Web de commerce électronique répondant aux besoins du concessionnaire de voiture Citroen-Meklati en matière de vente en ligne des voitures et pièces de rechange. Pour atteindre cet objectif, nous avons couplé deux technologies : le Web et les Bases de Données.

Pour ce faire, nous avons opté pour une architecture 3-tiers « navigateur/serveur/SGBD », un SGBDR « SQL Server » et des langages de scripts : ASP pour le traitement et la formulation des requêtes et des résultats, VBScript pour le contrôle des pages.

Le site offre plusieurs fonctionnalités : consultation des rayons, suivi de la commande, un outil de recherche des produits par mot clé ... etc.

**Mot clés :** Commerce électronique, E-Business, application Web, architecture 3-tiers, pages Web dynamiques, ASP, SQL Server, VBSCRIPT.

## Abstract

This aim of the work presented in this thesis has a principal goal for the design and the realization of a Web site of electronic business, for the needs of the dealer of cars Citroen-Meklati as regards to sales on line of cars and spare part. To achieve this goal, we join two technologies: the Web and Data bases.

With this intention, we choose an architecture 3-third "navigator/server/SGBD", a SGBDR "SQL Server" and languages of scripts: ASP for the treatment, formulation of the requests and the results, VBScript for the pages control.

The site offers several services: consultation of stores, follow the order, and a tool of research of the products by key word ... etc.

**Key words:** E-Business, Web application, 3-tiers architecture, dynamic Web pages, ASP, SQL Server, VBSCRIPT.