

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire
Abd elhafid boussouf Mila

Institut des sciences et de la technologie

Département de Mathématiques et Informatique

**Mémoire préparé En vue de l'obtention du diplôme de
Licence
En:Filière Informatique**

THEME
CONCEPTION ET REALISATION D'UN SITE
WEB DYNAMIQUE DE VENTE EN LIGNE

Préparé par : - Belhamra zakarya

-Mosbah khoula

Encadré par : Boumassata Meriem

Grade : MAB

Année universitaire : 2014/2015



REMERCIEMENT

Avant tout développement sur cette expérience professionnelle, il apparait de commencer ce travail par des remerciements, à ceux qui nous ont beaucoup aidés

Nous tenons à exprimer nos vifs remerciements :

Tout d'abord Louange à dieu tout puissant de nous avoir aidé et éclairé le

Chemin pour achever notre travail et nos études.

Nos remerciements à nos très chers parents, frères, sœurs,

Collègues et amis respectives qui nous ont encouragés et soutenu

Durant tout notre parcours

*À notre encadreur mademoiselle **BOUMASSATA MERIEM** pour ses précieux conseils, sa disponibilité, la confiance qu'elle nous a toujours témoignée et la sollicitude dont elle nous a entouré, et ses conseils judicieux qu'il nous ont fait bénéficier.*

Nous tenons à remercier l'ensemble des enseignants Du primaire à l'université qui ont permis, par leurs conseils, pistes suggérés et commentaires, la réalisation de ce Travail de Fin d'Etude.

En fin, à tous ceux qui ont contribué de près ou de loin a l'achèvement de ce travail.

Dédicace

Je dédie ce modeste travail

*Ames chers parents, pour leur soutien, leur patience
et leur amour*

A ma cher grande mère

Atout ma grande famille

A mes fideles amis :

Salem, Sami, Hamza, Otman et, Yasser

A tous mes amis

Amon binôme Khoula et tout sa famille





Dédicace

Premièrement est avant tous nous remercions dieu le tout puissant de nous avoir donné le courage de réaliser ce modeste travail.

Un dédicace spécialement pour deux personnes qui sont mes tés très chers parent

Mon père dahmen l'homme qui m'a toujours aidée et m'a toujours soutenu durant toute mes réussites à compter de la première année d'école jusqu' à ce jour.

Il a aussi su m'éduquer civilement ainsi que religieusement et m'a donné tout son amour

Q 'un père peut donner à son fils ou sa fille. Je lui dis merci papa et sans

Oublier de lui dire je t'aime très fort papa de tout mon cœur

Saida mon adorable maman, sage femme, ma très chaire amie, elle ma bien aimé, malgré tout l'amour

Que je lui donne et toutes la reconnaissance que je lui dois mais je n'arrive jamais à lui rendre tout ce qu'elle m'a donnée toute sa vie, elle souffre pour me voir à l'aise. Mon adorable maman, tout ce que je peux te donner tout en étant très reconnaissante envers toi surtout pour ce que t'as donné pour moi c'est ces mots qui proviennent directement de mon cœur et de mon intérieur.

Je dédie aussi ce travail à mes chères frères et sœurs : abd al Malek, Zouhir, hadjer, Romaissa, sohaibe et raouf

A mes chers petits neveux: taha et Lokman

Pour tous les membres de la famille « mosbah », sans exception,

A Mon fiancé « HAMZA »

Je dédie aussi

Pour tous mes amis proches et lointains.

Pour chaque ami honnête, Fidel, adjoint, aimer, intéressés, de m'écouter lorsque j'en ai besoin, pour la compréhension, d'accepter mes erreurs et de me pardonner, pour leur aides et leur soutiens

Sur tout « Hallouma & Manel»

A tous ceux et celles qui me sont chers

Et bien sûr a mon binôme «ZAKI»

** KHOULA **



SOMMAIRE

INTRODUCTION GENERALE.....	01
----------------------------	----

Chapitre 01 : Internet et technologie web

1. INTRODUCTION.....	03
2. LES RESEAUX INFORMATIQUE.....	03
2.1 DEFINITION.....	03
2.2 TYPES DE RESEAUX	03
3. INTERNET.....	04
3.1 HISTORIQUE.....	04
3.2 DEFINITION.....	05
3.3 SERVICES D’INTERNET	06
3.4 LES PROTOCOLES D’INTERNET.....	06
4. TECHNOLOGIE WEB.....	07
4.1 HISTORIQUE	07
4.2 DEFINITION	07
4.3 SITE STATIQUE.....	07
4.4 URL.....	07
4.5 HYPERLIEN	08
4.6 NAVIGATEUR WEB.....	08
4.7 PAGE WEB	09
4.8 SITE WEB STATIQUE.....	09
4.9 SITE WEB DYNAMIQUE.....	09
4.10 LES LANGUAGES DU WEB.....	10
4.10.1 POUR SITE STATIQUE	11
4.10.2 POUR SITE DYNAMIQUE	11
5. ARCHITECTURE CLIENT/SERVEUR.....	11
5.1 DEFINITION.....	11
5.2 TYPES DE SERVEURS.....	12
5.3 AVANTAGES DE L’ARCHITECTURE CLIENT/SERVEUR.....	12
5.4 INCONVENIENTS DU MODELE CLIENT/SERVEUR.....	13
5.5 FONCTIONNEMENT D’UN SYSTEME CLIENT/SERVEUR.....	13
5.6 TYPES D’ARCHITECTURES CLIENT/SERVEUR.....	14
6. CONCLUSION	15

SOMMAIRE

Chapitre 02: Langage de Modélisation UML et Processus Unifié UP

1. INTRODUCTION.....	16
2. LANGAGE DE MODELISATION UML	16
2.1 HISTORIQUE	16
2.2 DEFINITION.....	16
2.3 CARACTERISTIQUES D’UML.....	17
2.4 AVANTAGES D’UML.....	17
2.5 INCONVENIENTS D’UML.....	18
2.6 LES DIFFERENTS TYPES DE DIAGRAMMES D’UML.....	18
2.6.1 LA VUE STATIQUE (STRUCTURELLE)	18
2.6.2 LA VUE DYNAMIQUE (COMPORTEMENTALE).....	20
3. LE PROCESSUS UNIFIE UP.....	23
3.1 DEFINITION.....	23
3.2 LES CARACTERISTIQUES DU PROCESSUS UNIFIE.....	23
3.3 LE CYCLE DE VIE D’UP.....	24
3.4 LES PHASES D’UP.....	24
4. METHODE DE DEVELOPPEMENT (METHODE BASEE SUR UP).....	25
4.1 EXPRESSION DES BESOINS	25
4.2 ANALYSE DES BESOINS.....	25
4.3 CONCEPTION.....	25
4.4 IMPLEMENTATION.....	26
4.5 TEST.....	26
5. CONCLUSION	26

Chapitre 3: Analyse et conception

1. INTRODUCTION.....	27
2. IDENTIFICATION DES BESOINS.....	27
2.1 PRESENTATION DU PROJET.....	27
2.2 LES GRANDS CHOIX TECHNIQUES.....	28
2.3 EXIGENCES FONCTIONNELLES.....	28
2.4 EXIGENCES OPERATIONNELLES.....	29
2.5 DESCRIPTION DU CONTEXTE.....	29

SOMMAIRE

2.6	DIAGRAMME DE CAS D'UTILISATION.....	32
2.7	DESCRIPTION DES CAS D'UTILISATION.....	34
2.8	LES DIAGRAMMES DE SEQUENCE SYSTEME.....	43
3.	ANALYSE DU DOMAINE.....	50
3.1	IDENTIFICATION DES CONCEPTS DU DOMAINE.....	50
3.2	MODELE DE DOMAINE.....	51
4.	CONCLUSION.....	58

Chapitre 04 : Implémentation

1.	INTRODUCTION.....	59
2.	ENVIRONNEMENT DE DEVELOPPEMENT.....	59
2.1	LES LANGAGES DE PROGRAMMATION.....	59
2.2	OUTILS UTILISES.....	63
3.	PRESENTATION DE QUELQUES INTERFACES DE NOTRE SITE.....	66
3.1	PAGE «ACCUEIL».....	66
3.2	PAGE «INSCRIPTION ».....	67
3.3	PAGE «CONSULTER ANNONCES DE VENTE».....	67
3.4	PAGE «AFFICHER DETAIL».....	68
3.5	PAGE « METTRE PROPOSITION ».....	68
3.6	PAGE «CONTACTER VENDEUR».....	69
3.7	PAGE «AJOUTER ANNONCE DE VENTE».....	69
3.8	PAGE «GERER DEMANDES».....	70
3.9	PAGE «LOUER VEHICULE».....	70
4.	CONCLUSION.....	71
	CONCLUSION GENERALE.....	72
	REFERENCES.....	73

LISTE DES FIGURES

CHAPITRES	FIGURES	PAGES
CHAPITRE 01	Figure 1 : Différents types de réseaux.	4
	Figure 2 : Navigateur web.	8
	Figure 3: Fonctionnement d'un site statique.	9
	Figure 4: Fonctionnement d'un site dynamique.	10
	Figure 5: Fonctionnement d'un système client/serveur.	13
	Figure 6: Architecture à deux niveaux.	14
	Figure 7: Architecture a 3 niveaux.	14
	Figure 8: Architecture multi- niveaux.	15
CHAPITRE 02	Figure 1 : Exemple d'un diagramme de classes.	19
	Figure 2 : Exemple d'un diagramme d'objets.	19
	Figure 3 : Exemple d'un cas utilisation.	20
	Figure 4 : Exemple d'un diagramme de séquence.	21
	Figure 5 : Exemple d'un diagramme d'états-transitions.	22
	Figure 6 : Exemple d'un diagramme d'activité.	23
CHAPITRE 03	Figure 1 : Diagramme de contexte.	32
	Figure 2 : Diagramme de cas d'utilisation du visiteur.	32
	Figure 3 : Diagramme de cas d'utilisation du client.	33
	Figure 4 : Diagramme de cas d'utilisation du vendeur.	33
	Figure 5 : Diagramme de cas d'utilisation du locateur.	33
	Figure 6 : Diagramme de cas d'utilisation.	33
	Figure 7 : Diagramme de séquence système du cas d'utilisation (S'inscrire).	43
	Figure 8 : Diagramme de séquence système du cas d'utilisation (S'authentifier).	43
	Figure 9 : Diagramme de séquence système du cas d'utilisation (Contacter client).	44
	Figure 10 : Diagramme de séquence système du cas d'utilisation (Gérer compte).	45
	Figure 11 : Diagramme de séquence système du cas d'utilisation (Consulter annonces de vente).	45
	Figure 12 : Diagramme de séquence système du cas d'utilisation (Consulter annonces de location).	46
	Figure 13 : Diagramme de séquence système du cas d'utilisation (Contacter vendeur).	46
	Figure 14 : Diagramme de séquence système du cas d'utilisation (Gérer annonces de vente).	47
	Figure 15 : Diagramme de séquence système du cas d'utilisation (Gérer annonces de location).	48
	Figure 16 : Diagramme de séquence système du cas d'utilisation (Gérer demandes).	48
	Figure 17 : Diagramme de séquence système du cas d'utilisation (Mettre proposition).	50
	Figure 18 : Diagramme de séquence système du cas d'utilisation (Louer véhicule).	51
	Figure 19 : Diagramme du modèle de domaine.	51
	Figure 20 : Diagramme du modèle de domaine (S'inscrire).	52
	Figure 21: Diagramme du modèle de domaine (S'authentifier).	52

LISTE DES FIGURES

	Figure 22 : Diagramme du modèle de domaine (Consulter annonces de vente).	53
	Figure 23 : Diagramme du modèle de domaine (Consulter annonces de location).	53
	Figure 24 : Diagramme du modèle de domaine (Contacter vendeur).	54
	Figure 25 : Diagramme du modèle de domaine (Louer véhicule).	54
	Figure 26 : Diagramme du modèle de domaine (Mettre proposition).	55
	Figure 27 : Diagramme du modèle de domaine (Gérer compte).	55
	Figure 28 : Diagramme du modèle de domaine (Contacter client).	56
	Figure 29 : Diagramme du modèle de domaine (Gérer annonces de vente).	56
	Figure 30 : Diagramme du modèle de domaine (Gérer annonces de location).	57
	Figure 31 : Diagramme du modèle de domaine (gérer demandes).	57
CHAPITRE 04	Figure 1: Une partie du code HTML de style de formulaire de notre site.	60
	Figure 2 : Une partie du code CSS de style de formulaire de notre site.	60
	Figure 3 : Une partie du code PHP de style de formulaire de notre site.	61
	Figure 4 : Une partie du code MYSQL de style de formulaire de notre site.	62
	Figure 5: Utilisation de la base de données avec XAMPP.	63
	Figure 6 : Structure de la base de données dans PHPMyAdmin	64
	Figure 7 : Un exemple d'utilisation de RAPID PHP Editor.	65
	Figure 8 : L'interface des navigateurs web.	66
	Figure 9: Page d'accueil.	66
	Figure 10 : Page d'inscription.	67
	Figure 11: Page de consultation d'annonces de vente.	67
	Figure 12 : Page du détail d'une annonce de vente.	68
	Figure 13 : Page de contacte de vendeur.	69
	Figure 14 : Page d'ajout d'une annonce de vente.	69
	Figure15: Page de gestion des demandes.	70
	Figure16 : Page de location d'un véhicule.	70

LISTE DES TABLEAUX

CHAPITRE	TABLEAUX	PAGE
CHAPITRE 03	Tableau 1 : Messages échangés entre visiteur et système.	30
	Tableau 2 : Messages échangés entre client et système.	30
	Tableau 3 : Messages échangés entre vendeur et système.	30
	Tableau 4 : Messages échangés entre locateur et système.	31
	Tableau 5 : Fiche description textuelle du cas (S'inscrire).	34
	Tableau 6 : Fiche description textuelle du cas (Consulter annonces de vente).	35
	Tableau 7 : Fiche description textuelle du cas (Consulter annonces de location).	36
	Tableau 8 : Fiche description textuelle du cas (Louer véhicule).	37
	Tableau 9 : Fiche description textuelle du cas (Gérer compte).	38
	Tableau 10 : Fiche description textuelle du cas (S'authentifier).	38
	Tableau 11 : Fiche description textuelle du cas (Gérer annonces de vente).	39
	Tableau 12 : Fiche description textuelle du cas (Gérer annonces de location)	40
	Tableau 13 : Fiche description textuelle du cas (Gérer demandes).	41
	Tableau 14 : Fiche description textuelle du (Mettre proposition)	42
	Tableau 15 : Fiche description textuelle du (Contacter vendeur).	42
	Tableau 16 : Fiche description textuelle du (Contacter client).	42

INTRODUCTION GENERALE

L'entrée d'Internet dans le monde de commerce a bouleversé les façons de vendre, de fabriquer les produits et de gérer les clients et les fournisseurs. Aujourd'hui, plus que jamais, il est primordial, pour les commerçants, de posséder un site Internet. Que ce soit simplement pour que les gens puissent entrer en communication avec eux ou pour agrandir la clientèle.

Donc la clientèle peut consulter le site Internet et trouver l'information par rapport au produit ou au service qui l'intéresse, et ce, de partout à travers le monde et en tout temps. Par le fait même, l'internaute peut naviguer sur la page web pour répondre à ses questions sans déranger l'un des employés qui peut continuer de vaquer à ses occupations pendant ce temps.

Avec l'augmentation de l'utilisation d'Internet en Algérie, la meilleure solution pour les commerçants algériens est la mise en place des sites web. Grâce à ces sites en ligne, ils pourront toucher un grand nombre de clients qui ne sont pas dans leurs sphères géographiques classiques, et qui ne se limitent plus à une ville ou une région, mais ils s'étendent au territoire national.

Notre projet consiste à concevoir et réaliser un site web pour la location et la vente de véhicules en ligne. L'objectif fondamental de notre site est de faciliter la communication entre les sociétés où individus, vendeurs où locataires de véhicules, et les clients qui peuvent consulter les informations sur les véhicules, négocier les prix, louer, où acheter des véhicules de chez eux.

➤ **Contenu du mémoire**

Le mémoire est constitué d'une introduction générale, quatre chapitres, et une conclusion générale.

Le premier chapitre : Consiste à une présentation générale d'internet et des technologies web.

Le deuxième chapitre : Composé essentiellement de deux parties : langage de modélisation UML et processus unifié UP utilisés pour la conception de notre site web.

Le troisième chapitre : Présente la conception du site qui est basée sur le langage UML et dirigée par le processus UP. Il est composé des deux parties suivantes :

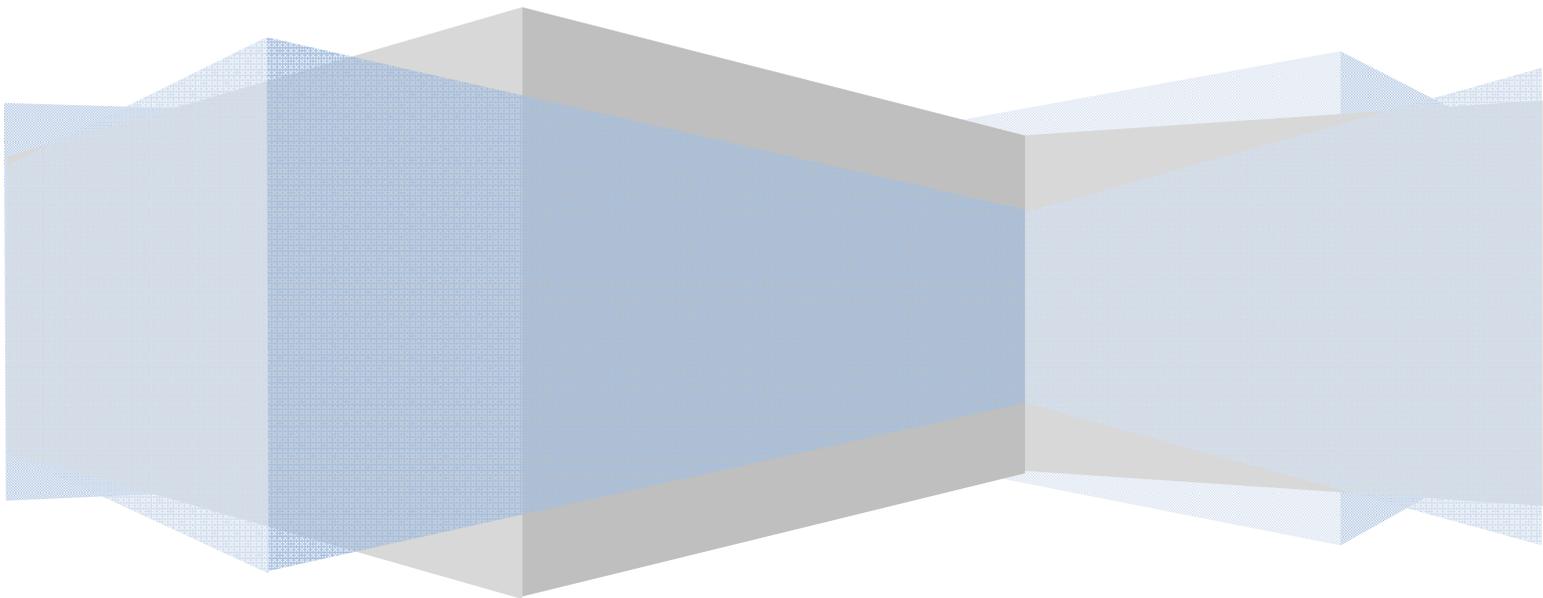
INTRODUCTION GENERALE

- ✓ Identification des besoins : Présente les acteurs et les cas d'utilisations. On utilise le diagramme de contexte, le diagramme de cas d'utilisation, les fiches descriptives et puis les diagrammes de séquence système.
- ✓ Analyse du domaine : Présente le modèle de domaine et les diagrammes de classes participantes.

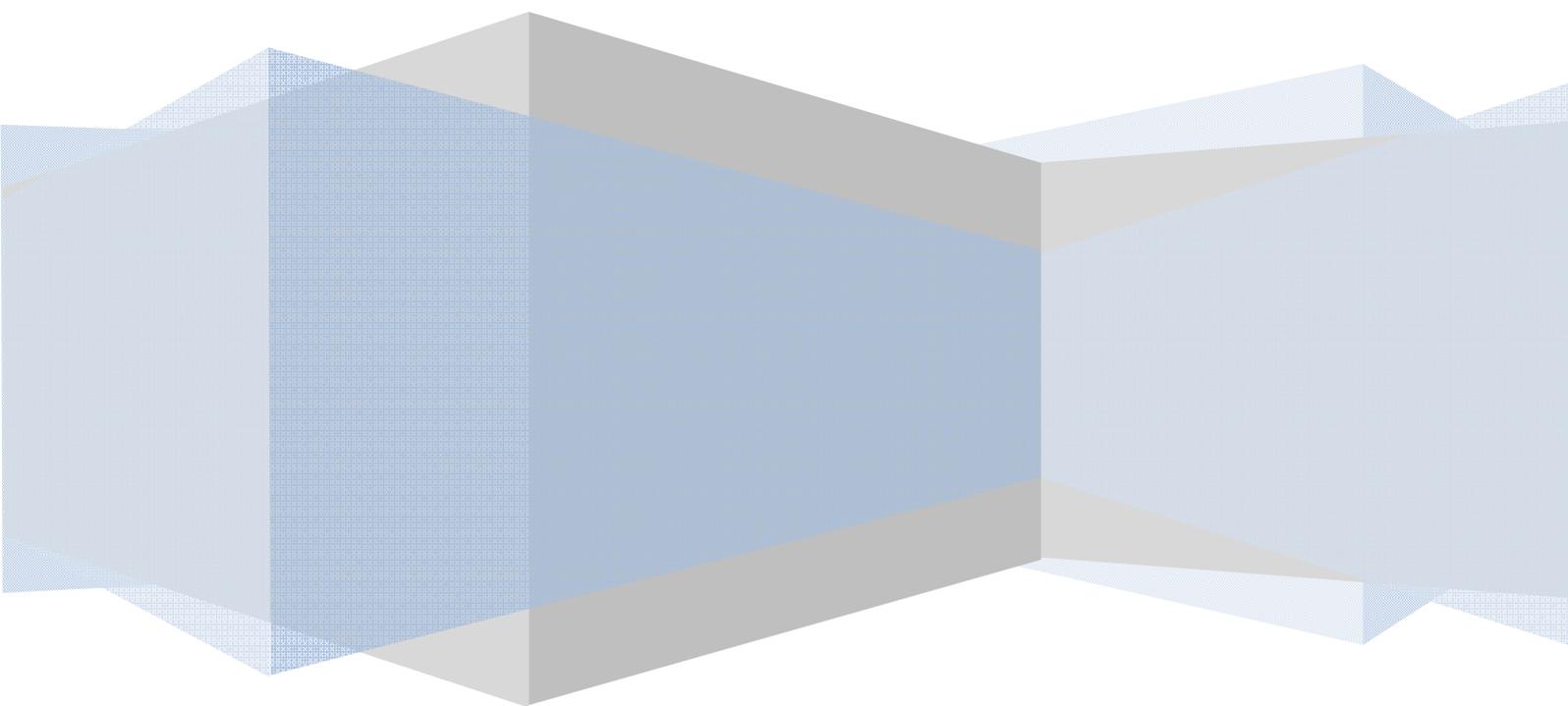
Le quatrième chapitre : Présente les outils de développement utilisés pour la réalisation du site : le langage HTML, le langage PHP, le système de gestion de base de données MySQL, et l'outil Rapid PHP. Il contient, aussi, une démonstration du fonctionnement de notre site en montrant les interfaces obtenues par l'exécution de quelques cas d'utilisation.

PARTIE 1

Etat de l'art



CHAPITRE 01 : **Internet et technologie** **web**



1. INTRODUCTION

Dans la vie quotidienne, internet est devenu aussi indispensable que la télévision, le téléphone, le réfrigérateur, etc. Lorsqu'on consulte un site web, c'est dans un but précis : trouver les réponses à nos questions, partager, communiquer, s'amuser, etc.

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

Dans ce chapitre, nous allons présenter les réseaux d'informatiques et leurs types. Ensuite, le réseau internet, ses services et ses protocoles. Après, nous allons présenter la technologie web et les sites web qui sont à la base de notre travail. Et en fin, nous présenterons l'architecture client/serveur.

2. LES RESEAUX INFORMATIQUES

2.1. Définition

Un réseau informatique est un ensemble d'équipements informatiques (ordinateur et périphériques) reliés entre eux grâce à des supports de communication (câble : réseau câblé, ou ondes radios : réseau sans fil) permettant la communication (transfert des informations électroniques) et le partage de ressources (matérielles et logicielles).

Les réseaux informatiques permettent :

- Le partage d'applications et des fichiers (échange du texte, vidéo, image, programmes, etc.).
- Le partage du matériel (imprimantes, lecteurs de CD-ROM, disques durs, etc.).
- La communication entre personnes (courrier électronique, discussion en direct, etc.).
- La communication entre processus (entre des machines industrielles).
- La garantie de l'unicité de l'information (bases de données).

2.2. Types de réseaux

On distingue différents types de réseaux selon leur distance :

- **PAN (Personal Area Network)**

Tous petits réseaux, de quelques mètres d'étendue, permettant d'interconnecter des machines personnelles : PC portable, téléphone mobile, agenda électronique, etc.

Les technologies d'interconnexion utilisées dans ce type de réseaux sont: USB, Bluetooth, Infrarouge.

- **LAN (Local Area Network)**

Il s'agit d'un ensemble d'ordinateurs appartenant à une même organisation, (Taille jusqu'à quelques kilomètres), reliés entre eux par un réseau. Par exemple : FDDI (Fiber Distributed Data Interface), réseau Ethernet.

- **MAN (Metropolitan Area Network)**

C'est un réseau qui interconnecte plusieurs LAN géographiquement proches (au maximum quelques dizaines de Km). Un MAN est formée de commutateurs ou de routeurs interconnectés par des liens hauts débit (en général en fibre optique).

- **WAN (Wide Area Network)**

Interconnecter plusieurs LAN et MAN, couvre un pays ou un continent. Les réseaux WAN sont opérés par de grands fournisseurs de réseaux [11].

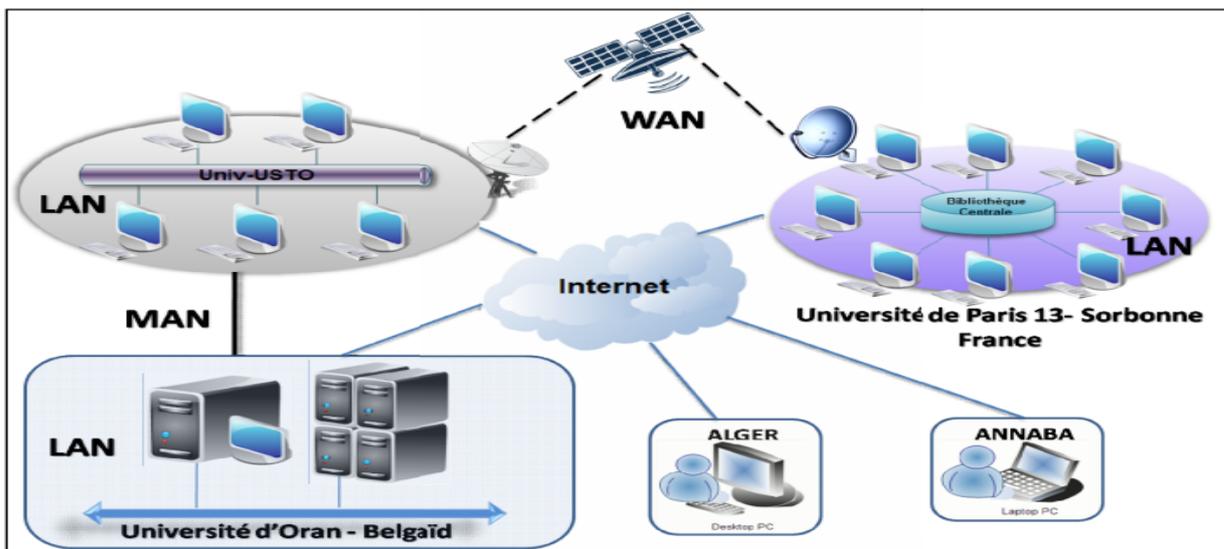


Figure 1 : Différents types de réseaux [10].

3. INTERNET

3.1. Historique

Internet est un ensemble de réseaux informatiques privés et publics interconnectés grâce à un protocole de communication commun. Son principe a été conçu par les milieux américains de la défense et de la recherche, à la fin des années 1960, puis il s'est progressivement généralisé au domaine civil et commercial. Limité à un millier environ en 1985, le nombre d'utilisateurs d'internet est passé à plus de 2 millions en 1994 et à près de 1,250 milliard en 2007.

3.2 Définition

INTERNET (INTERnational NETwork) est un réseau mondial associant des ressources de télécommunication et des ordinateurs serveurs et clients, destiné à l'échange de messages électroniques, d'informations multimédias et de fichiers.

Il fonctionne en utilisant un protocole standardisé TCP/IP (Transfert Control Protocol/Internet Protocol) qui permet l'acheminement de proche en proche de messages découpés en paquets indépendants.

3.3 Services d'internet

- **Echange de messages (e-mail)**

E-mail (courrier électronique) permet d'échanger (quasi instantanément) du courrier et des documents avec toute personne possédant une adresse électronique.

- **Les forums de discussion (News)**

News (Groupe News ou forum de discussion). Il s'agit d'une messagerie électronique ouverte, dans laquelle chacun peut écrire et tout le monde peut consulter.

- **Echange de fichiers (FTP)**

Transfert de fichiers (File Transfert Protocol). Ce service comme son nom l'indique, permet de transférer directement des fichiers d'une machine à une autre.

- **Gopher**

Ce service est semblable au service FTP, il permet toutefois seulement de retrouver des informations, mais n'autorise pas le téléchargement.

- **La connexion à distance (Telnet)**

Telnet est un protocole permettant d'émuler un terminal à distance, cela signifie qu'il permet d'exécuter des commandes saisies au clavier sur une machine distante. Ainsi, étant donné que la machine distante envoie les données à afficher, l'utilisateur a l'impression de travailler directement sur la machine distante.

- **Le Web (Word Wide Web)**

Le service le plus connu, le plus récent et maintenant le plus utilisée. Il s'agit d'un jeu de pages interconnectées qui constituent différents sites Web. Les pages Web utilisent des liens spéciaux qui prennent la forme de texte en surbrillance ou de graphismes. Il suffit de cliquer sur un lien pour être transporté vers une nouvelle page qui peut se trouver à l'autre bout de la planète. Sans efforts, le Web nous fait passer de pages en pages et de sites en sites quel que soit leur emplacement.

- **Le commerce électronique**

Se définit essentiellement comme un processus d'achat et de vente de produits et de services sur internet, il englobe la gestion des transactions d'achat et de transfert de fond sur des réseaux d'ordinateurs.

3.4 Les protocoles d'internet

- **HTTP**

Le protocole HTTP (HyperText Transfer Protocol) est le protocole le plus utilisé sur Internet depuis 1990. Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format HTML) localisés grâce à une chaîne de caractère appelée URL (Uniform Resource Locator) entre un navigateur (le client) et un serveur. L'URL est l'adressage standard de n'importe quel document, sur n'importe quel ordinateur en local ou sur Internet.

- **FTP**

Le protocole FTP (File Transfer Protocol) s'inscrit dans un modèle client/serveur, c'est-à-dire qu'une machine envoie des ordres (le client) et que l'autre attend des requêtes pour effectuer des actions (le serveur).

- **SMTP**

Le protocole SMTP (Simple Mail Transfer Protocol) est un protocole de communication utilisé pour transférer le courrier électronique vers les serveurs de messagerie électronique.

- **POP**

Le protocole POP (Post Office Protocol) est un protocole standard utilisé par les clients de messageries pour lire des informations provenant du serveur. Le protocole POP a été défini en 1984 parallèlement au protocole TCP/IP.

- **IMAP**

Le protocole IMAP (Internet Message Access Protocol) est un protocole qui permet au client de modifier le statut d'un message pour marquer des messages lus comme étant des messages non lus et inversement. De même, il permet d'enregistrer, de copier ou de supprimer des messages directement sur le serveur sans qu'ils aient été transférés vers le client.

- **TFTP**

Le protocole TFTP (Trivial File Transfer Protocol) est une méthode simple utilisée pour transférer des fichiers. Il s'appuie sur le protocole UDP (User Datagram Protocol) au niveau transport. Il se limite à l'écriture et à la lecture de fichiers entre deux hôtes réseau. Il ne possède pas de fonction d'authentification et il est impossible de lister les fichiers à distance [4].

4 TECHNOLOGIE WEB

4.1 Historique

Le WWW (World Wide Web) a été créé en 1990 par l'ingénieur du CERN Tim Berner Lee. Le 30 avril 1993, le CERN annonce que le « World Wide Web » sera libre d'utilisation pour tout le monde.

La toute première adresse internet du premier serveur web était « nxoc01.cern.ch », fonctionnant sur un ordinateur du CERN de marque NeXT. Une version ultérieure de cette toute première page html est consultable sur le site du W3C.

Avant l'introduction de HTML et HTTP, d'autres protocoles tel que FTP et Gopher étaient utilisés pour obtenir des fichiers depuis un serveur. Ces protocoles offraient une organisation hiérarchique de répertoires dans lesquels l'utilisateur naviguait et choisissait les fichiers à télécharger. Les documents étaient principalement des fichiers texte sans mise et des fichiers encodés par un traitement de texte.

4.2 Définition

Le World Wide Web(ou WWW, ou Web) est un très grand système d'information réparti sur un ensemble de sites connectés par le réseau Internet. Ce système est essentiellement constitué de documents hypertextes, ce terme pouvant être pris au sens large : textes, images, sons, vidéos, etc.

Chaque site propose un ensemble plus ou moins important de documents qui se transmettent sur le réseau par l'intermédiaire d'un programme serveur. Ce programme serveur dialogue avec un programme client qui peut être situé n'importe où sur le réseau. Le programme client prend le plus souvent la forme d'un navigateur.

4.3 Site web

Un site web est un ensemble de pages web visualisables dans un navigateur, ces pages web sont reliées entre elles par des liens permettant le passage de l'une à l'autre. Il y a toujours une page particulière, appelée page d'accueil, qui représente le portrait du site et à partir de laquelle on peut accéder aux autres pages du site en suivant les liens à l'intérieurs du site.

4.4 URL

Un **URL** « Uniform Resource Locator » fournit un schéma de nommage généralisé pour spécifier les ressources de l'Internet au moyen d'une chaîne de caractère **ASCII** (American Standard Code for Information Inter change) imprimables. Les caractères sont imprimables pour pouvoir envoyer des **URL** dans des courriers électroniques, les imprimer

sur des cartes de visites ou les afficher sur des tableaux. Un **URL** typique comporte quatre parties :

- **Le nom du protocole**

Indique au navigateur le protocole qu'il doit utiliser pour accéder à une ressource sur le serveur.

- **L'adresse du serveur**

Il s'agit d'un nom de domaine de l'ordinateur hébergeant la ressource demandée. Notons qu'il est possible d'utiliser l'adresse IP du serveur, ce qui rend par contre l'URL moins lisible.

- **Le numéro du port**

Identifie un programme qui s'exécute sur le serveur. On spécifie expressément un numéro de port, en le séparant du nom du serveur par deux points superposés.

- **Le chemin d'accès à une ressource cible**

Début par une barre de fraction, et se trouve après l'hôte et le numéro de port. La signification de ce champ dépend de la ressource à laquelle on accède.

4.5 Hyperlien

Un hyperlien, lien hypertexte ou simplement lien, est une référence dans un système hypertexte permettant de passer automatiquement d'un document consulté à un document lié. Un hyperlien a une source (ou origine) et une destination (ou cible). L'activation de l'élément source d'un hyperlien permet de passer automatiquement à sa destination.

La technique à la base des hyperliens du Web est également utilisée pour inclure des images ou des sous-pages dans les pages Web.

4.6 Navigateur web

Un navigateur Internet est un logiciel informatique qui permet à un utilisateur d'accéder à Internet. Au cœur du logiciel, il y a un moteur de rendu visuel qui interprète le contenu HTML dans une page Internet. Les navigateurs les plus connus sont : Internet Explorer, Safari, Firefox.



Figure 2 : Navigateur web [8].

4.7 Page web

Une page web est un document destiné à être consulté avec un navigateur web. Une page web est toujours constituée d'une ressource centrale (généralement un document HTML) et d'éventuelles ressources liées, automatiquement accessibles (par exemple, des images) [8].

4.8 Sites Web statiques

Ce sont des sites réalisés uniquement à l'aide des langages HTML et CSS. Ils fonctionnent très bien mais leur contenu ne peut pas être mis à jour automatiquement : il faut que le propriétaire du site (le webmaster) modifie le code source pour y ajouter des nouveautés.

Les sites statiques sont donc bien adaptés pour réaliser des sites « vitrine », pour présenter par exemple son entreprise, mais sans aller plus loin. Ce type de site se fait de plus en plus rare aujourd'hui.

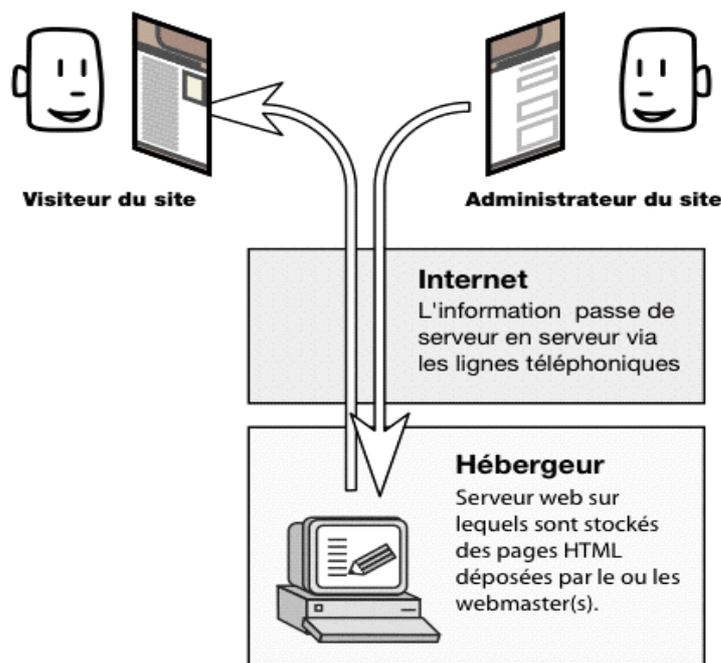


Figure 3: fonctionnement d'un site statique [10].

4.9 Sites web dynamiques

Plus complexes, ils utilisent d'autres langages en plus de HTML et CSS, tels que PHP et MySQL. Le contenu de ces sites web est dit « dynamique » parce qu'il peut changer sans l'intervention du web master. La plupart des sites web visités aujourd'hui, sont des sites dynamiques. Le seul pré requis pour apprendre à créer ce type de sites est de déjà savoir réaliser des sites statiques en HTML et CSS.

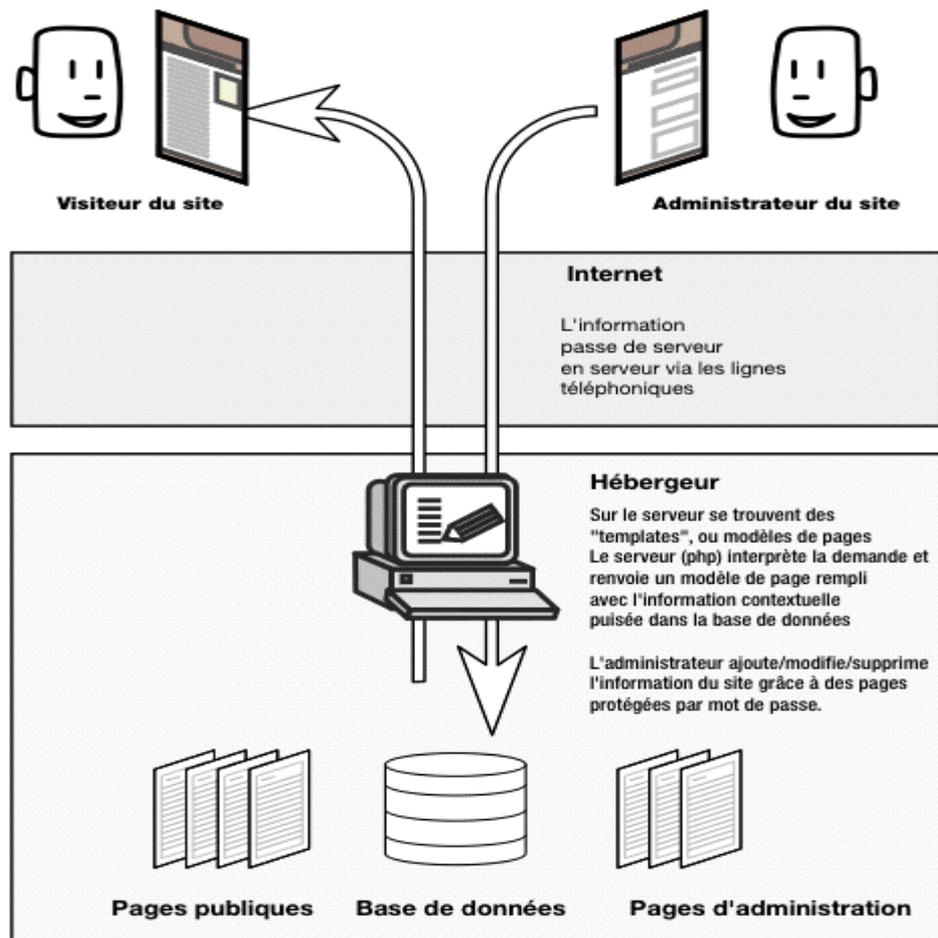


Figure 4: fonctionnement d'un site dynamique [10].

4.10 Les langages du web

4.10.1 Pour un site statique

- **HTML (Hyper Text Mark-up Language)**

HTML (Hyper Text Markup Language) est un langage permettant de décrire les différents composants d'un document : Définir des titres, construire des tableaux, mettre en formes des textes. Le HTML est un langage simple rédigé sous forme de texte et qui demande un simple éditeur de texte.

Le lien Hypertexte est un des principaux atouts du langage HTML [12].

- **CSS (Cascading Style Sheets)**

C'est le langage de mise en forme des sites web. Alors que le HTML permet d'écrire le contenu des pages web et de les structurer, le langage CSS s'occupe de la mise en forme et de la mise en page. C'est en CSS que l'on choisit notamment la couleur, la taille des menus et bien d'autres choses.

Ces langages sont la base de tous les sites web. Lorsque le serveur envoie la page web au client, il envoie en fait du code en langage HTML et CSS.

4.10.2 Pour un site dynamique

HTML et CSS sont indispensables. Cependant, ils ne suffisent pas pour réaliser des sites dynamiques. Il faut les compléter avec les langages PHP et MySQL.

- **PHP (HyperText PreProcessor)**

C'est un langage que seuls les serveurs comprennent et qui permet de rendre un site dynamique. C'est PHP qui « génère » la page web. Il peut fonctionner seul mais il prend tout son sens que s'il est combiné à un outil tel que MySQL.

- **SQL (Structure Query Language)**

C'est un langage normalisé d'intégration de base de données. Puis qu'il est normalisé, il est indépendant du type des bases de données, les mêmes commandes peuvent donc être exploitées quelle que soit la base utilisée (ex. MYSQL). Les commandes SQL peuvent ainsi gérer tout type d'action sur le serveur de base de données [13].

5. ARCHITECTURE CLIENT/SERVEUR

5.1 Définition

L'architecture client/serveur est un ensemble de machines clients liées à une machine serveur qui leur fournit des services.

L'architecture client/serveur propose un environnement ouvert et souple où la règle est l'assemblage multi vendeurs.

- **Client**

La machine client comporte une interface utilisateur généralement graphique, le client envoie des requêtes au serveur à chaque fois que l'utilisateur veut ramener des informations disponibles au niveau du serveur.

- **Serveur**

La machine serveur est généralement un ordinateur très puissant en termes de capacités de traitement et de stockage, le serveur fait référence à tout processus qui analyse et traite la requête et envoie la réponse.

- **Requête**

C'est un message envoyé par le client au serveur décrivant l'opération à exécuter.

- **Réponse**

C'est le message envoyé par le serveur à un client suite à l'exécution d'une requête.

5.2 Types de serveurs

- **Serveur de fichier**

Permet de partager des données à travers un réseau. Les serveurs de fichiers sont utiles pour partager des fichiers sur un réseau ils sont indisponibles pour créer des banques des documents, d'image, de dessins industriels ou autre grand collection d'objets.

- **Serveur d'application**

Permet d'utiliser un programme sur un serveur à partir de tous les postes clients simultanément, principalement des applications qui utilisent des bases de données (gestion de fabrication, commerciale, comptabilité, stock, ...). Ces applications doivent être programmées pour gérer les partages.

- **Serveur d'impression**

Partage des imprimantes. Certaines imprimantes réseaux peuvent être directement connectées sur le réseau sans passer par un PC, des boîtiers spécifiques peuvent également être utilisés, quand ce n'est pas directement intégré dans l'imprimante.

- **Serveur de base de données**

Logiciel qui permet la création de bases de données mais aussi la récupération de données contenues dans les bases. Le client émet des requêtes SQL, sous forme de messages en direction du serveur. Le code traitant la requête ainsi que les données résident sur la même machine, le serveur utilise sa propre capacité de traitement pour rechercher les données demandées. Le résultat de chaque requête SQL, est envoyé sur le réseau.

5.3 Avantages de l'architecture client/serveur

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

- **Des ressources centralisées**

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

- **Une meilleure sécurité**

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

- **Une administration au niveau serveur**

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

- **Un réseau évolutif**

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

5.4 Inconvénients du modèle client/serveur

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

- **Un coût élevé**

Dû à la technicité du serveur.

- **Un maillon faible**

Le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui. Heureusement, le serveur a une grande tolérance aux pannes (notamment grâce au système RAID (**R**edundant **A**rray **o**f **I**nexpensive **D**isks)).

5.5 Fonctionnement d'un système client/serveur

Un système client/serveur fonctionne selon le schéma suivant :

Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur, le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port [8].

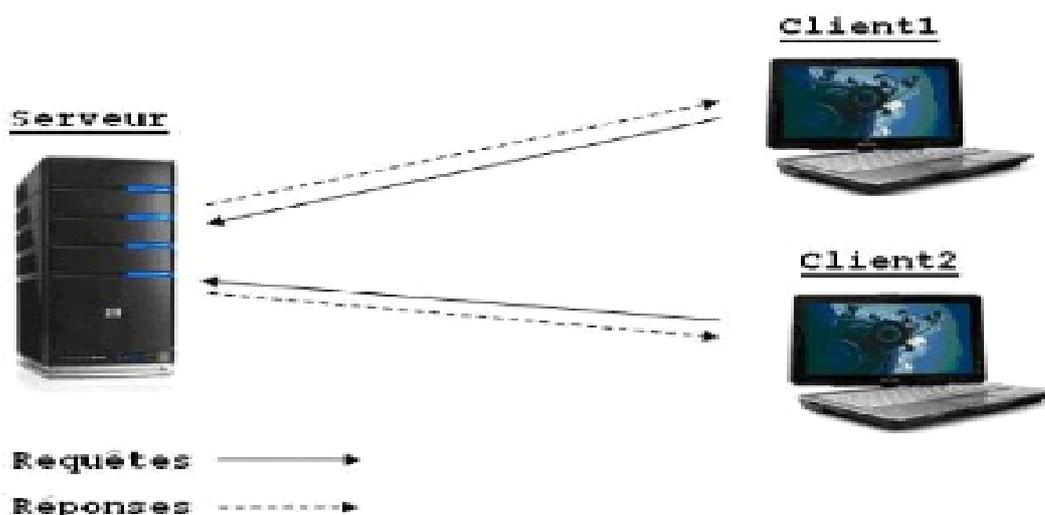


Figure 5: Fonctionnement d'un système client/serveur [10]

5.6 Types d'architectures client/serveur

• Architecture à deux niveaux

L'architecture à deux niveaux caractérise le système client/serveur dans lequel le client demande une ressource au serveur qui la lui fournit directement (le serveur ne fait appel à aucune application pour fournir ce service).

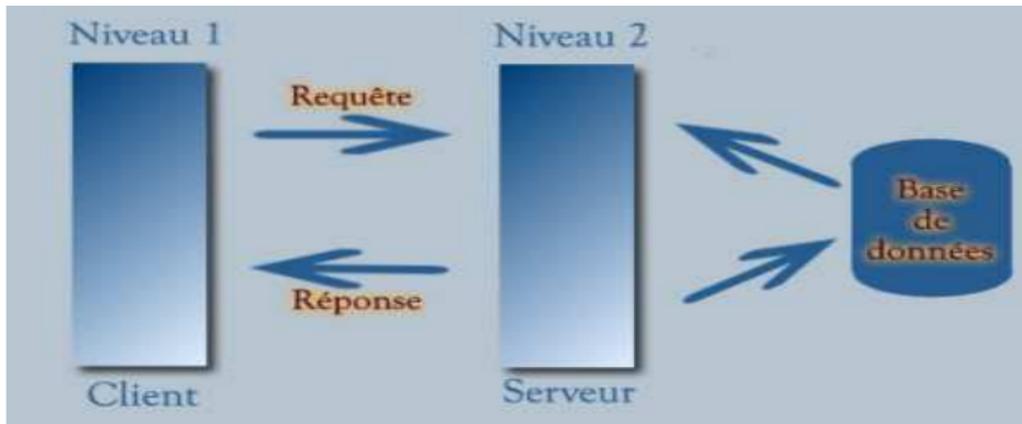


Figure 6: Architecture à deux niveaux [18].

• Architecture à 3 niveaux

Dans l'architecture à 3 niveaux, il existe un niveau intermédiaire :

- Un client, c'est-à-dire l'ordinateur demandeur de ressources (niveau 1), équipée d'une interface utilisateur (un navigateur web par exemple) chargée de la présentation.
- Le serveur d'application chargé de fournir la ressource mais faisant appel à un autre serveur (niveau 2).
- Le serveur secondaire (serveur de base de données), fournissant au serveur d'application les données dont il a besoin (niveau 3) pour retourner directement la réponse vers le demandeur.

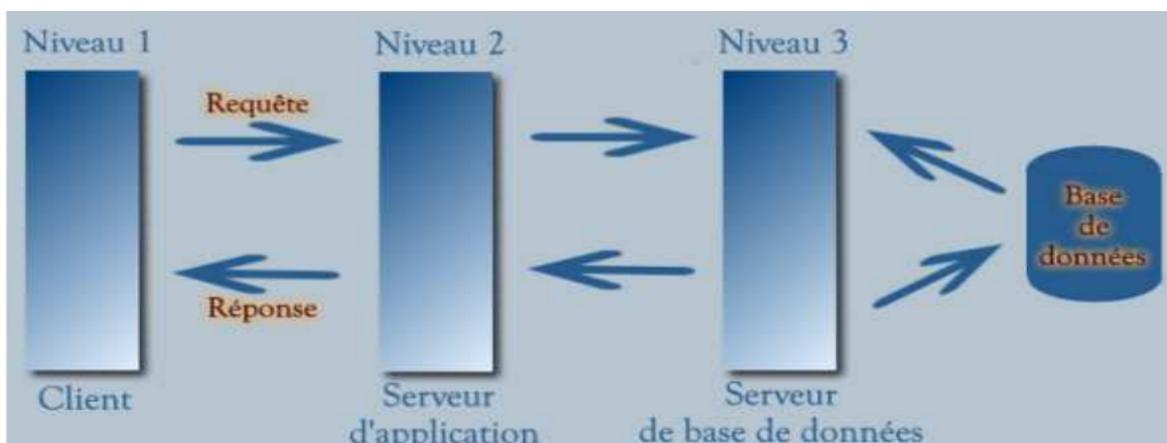


Figure 7: Architecture à 3 niveaux [10].

- **Architecture multi- niveaux**

Chaque serveur effectue un service spécialisé ainsi, un serveur peut 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 [17].

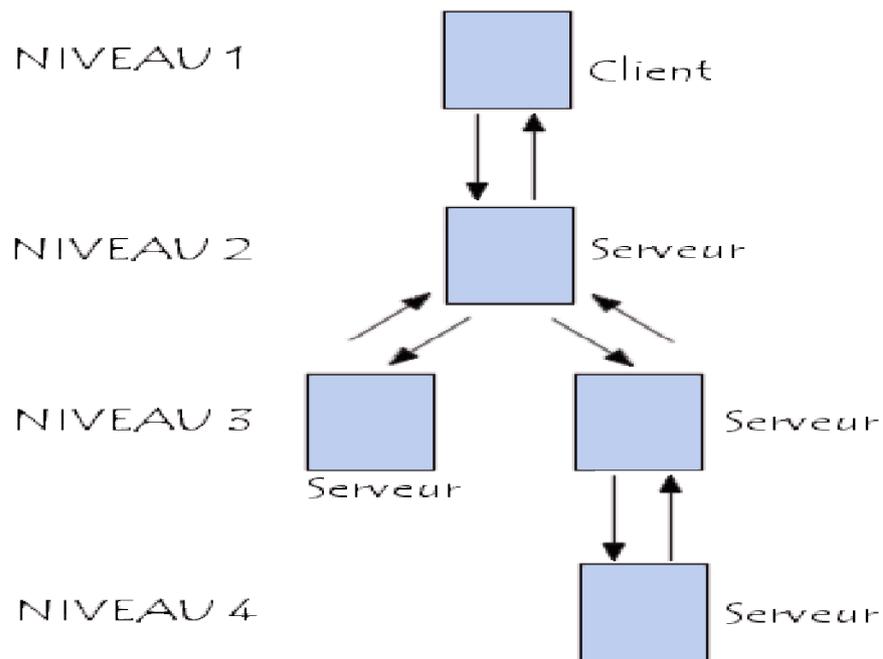


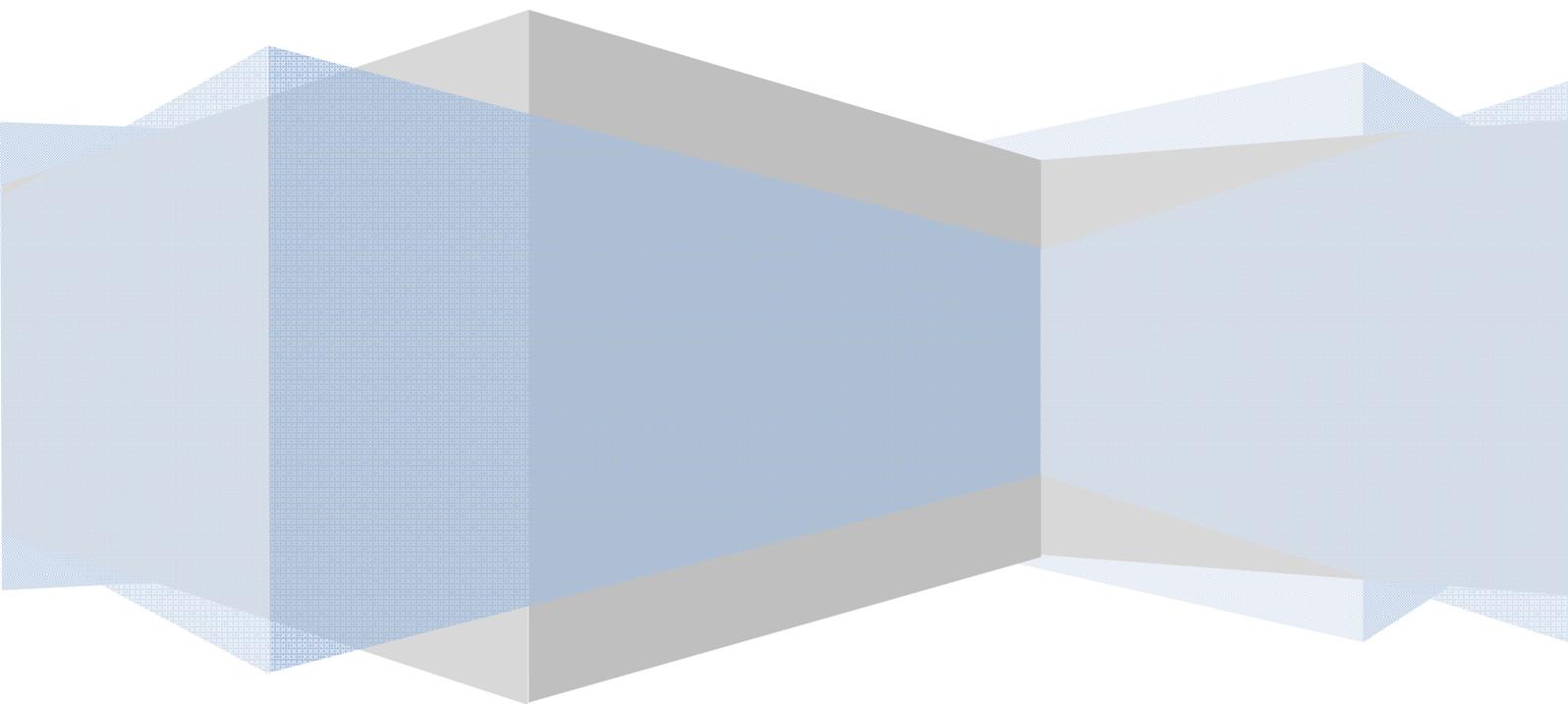
Figure 8: Architecture multi- niveaux [10].

6 CONCLUSION

Dans ce chapitre nous avons fait une revue sur le réseau internet et les services accessibles via lui. Aussi, nous avons parlé de la technologie web et les sites web dynamiques, ce qui nous a aidé à comprendre le principe de leur fonctionnement.

Dans le chapitre suivant, nous allons voir la présentation des concepts de base de langage de modélisation UML et du processus unifié UP qui permettent d'exprimer de manière uniforme l'analyse et la conception de notre site web.

Chapitre 02 :
Langage de modélisation
UML et processus unifié
UP



1. INTRODUCTION

Au cours de ces dernières années le langage de modélisation UML (Unified Modeling Language) est devenu un standard de fait pour décrire tout système logiciel. La norme UML a été décrite en même temps qu'une méthode d'analyse et de conception des systèmes logiciels ; le Processus Unifié (UP).

Dans le cadre de ce chapitre, nous allons définir quelques généralités portant sur la méthode et outils mettant en évidence la réalisation de notre projet. Nous allons commencer par présenter le langage de modélisation unifié UML (Unified Modeling Language), définir la démarche générique du processus de développement logiciel qui l'accompagne et enfin nous allons présenter les principaux concepts de base de la méthode de développement basée sur UP que nous allons suivre pour réaliser notre projet.

2. UML

2.1 Historique

Les méthodes utilisées dans les années 1980 pour organiser la programmation impérative (notamment Merise) étaient fondées sur la modélisation séparée des données et des traitements.

Lorsque la programmation par objets prend de l'importance au début des années 1990, la nécessité d'une méthode qui lui soit adaptée devient évidente. Plus de cinquante méthodes apparaissent entre 1990 et 1995 (Booch, Classe-Relation, Fusion, HOOD, OMT, OOA, OOD, OOM, OOSE, etc.) mais aucune ne parvient à s'imposer. En 1994, le consensus se fait autour de trois méthodes :

- OMT de James Rumbaugh (General Electric) : fournit une représentation graphique des aspects statique, dynamique et fonctionnel d'un système.
- OOD de GradyBooch : définie pour le (Department of Defense), introduit le concept de paquetage (package)
- OOSE d'Ivar Jacobson (Ericsson) : fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou use cases).

L'unification a progressé par étapes. En 1995, Booch et Rumbaugh (et quelques autres) se sont mis d'accord pour construire une méthode unifiée, Unified Method 0.8 ; en 1996, Jacobson les a rejoints pour produire UML 0.9 (notons le remplacement du mot méthode par le mot langage, plus modeste). Les acteurs les plus importants dans le monde du logiciel s'associent alors à l'effort (IBM, Microsoft, Oracle, DEC, HP, Rational, Unisys etc.) et UML 1.0 est soumis à l'OMG.

L'OMG adopte en novembre 1997 UML 1.1 comme langage de modélisation des systèmes d'information à objets. La version d'UML en cours en 2008 est UML 2.1.1 et les travaux d'amélioration se poursuivent [1].

2.2 Définition

UML, c'est l'acronyme anglais pour « Unified Modeling Language ». UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc. Réaliser ces diagrammes revient donc à modéliser les besoins du logiciel à développer [2].

2.3 Caractéristiques du langage de modélisation UML

Le méta modèle permet de donner des bases solides et rigoureuses à ce langage graphique, dont les représentations graphiques ne sont là que pour véhiculer des concepts de réalisation. Le méta-modèle d'UML est en fait un langage formel possédant les caractéristiques suivantes:

- Un langage sans ambiguïtés.
- Un langage universel pouvant servir de support pour tout langage orienté objet.
- Un moyen de définir la structure d'un programme.
- Une représentation visuelle permettant la communication entre les acteurs d'un même projet.
- Une notation graphique simple, compréhensible même par des non informaticiens.
- Générique et expressif.
- Syntaxe et sémantique définies.
- Flexible (configurable, extensible). [2]

2.4 Avantages d'UML

- Un certain consensus autour de l'utilisation d'UML : standard de fait dans l'industrie.
- Notation avec une syntaxe très riche, tout en restant intuitive.
- Intégration dans des ateliers de génie logiciel avec production de squelettes de codes et autres transformations automatiques des modèles.
- Langage de contraintes OCL (Object Constraint Language) pour spécifications précises à utiliser en complément.

- Il facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel [3].

2.5 Inconvénients d'UML

- Notation majoritairement graphique pouvant se révéler insuffisante ou trop chargée d'un point de vue expressivité.
- Sémantique floue ou mal définie pour certains types de diagrammes.
- Lien parfois difficile entre les vues et diagrammes d'une même application [3].

2.6 Les différents types de diagrammes UML

Il existe 2 types de vues du système qui comportent chacune leurs propres diagrammes :

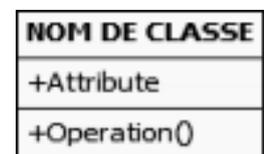
2.6.1 La vue statique (structurelle)

- **Diagramme de classes**

Est considéré comme le plus important dans un développement orienté objet, exprime la structure statique du système en termes de classes et de relations entre ces classes [9].

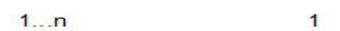
Il comporte les concepts suivants :

- **Classe** : Une classe décrit un groupe d'objets ayant les mêmes propriétés (attributs), un même comportement (opérations), et une sémantique commune (domaine de définition).



- **Relation** : il ya 4 relations principales :

- ❖ **Association** : une relation entre deux classes qui indique qu'il peut avoir des liens entre les instances des classes associés.



- ❖ **Agrégation** : une forme particulière d'association où un tout est relié à ses parties.



- ❖ **Composition** : un cas particulier de l'agrégation implique qu'un élément ne peut appartenir qu'à un seul agrégat composite.



- ❖ **Dépendance** : Quand une classe utilise une autre classe, par exemple comme membre ou comme paramètre d'une de ces fonctions, elle "dépend" ainsi de cette classe.



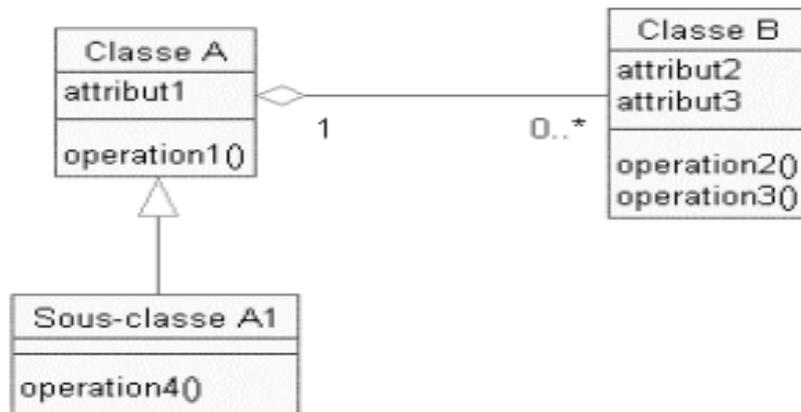


Figure 1 : Exemple d'un diagramme de classes.

- **Diagramme d'objets**

Ce diagramme permet la représentation d'instances des classes et des liens entre instances.

Il comporte 2 concepts :

- **Association** : une association décrit un ensemble de liens qui sont des instances des relations.
- **Objet** : décrit l'instance d'une classe.

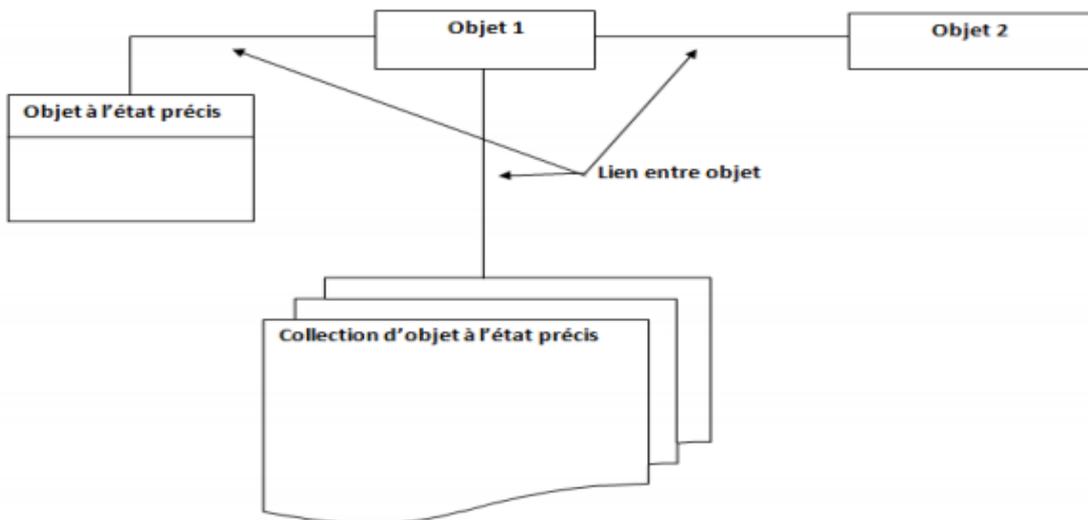


Figure 2 : Exemple d'un diagramme d'objets.

- **Diagramme de composants**

Ce diagramme représente les différents constituants du logiciel au niveau de l'implémentation d'un système. Un composant représente une entité logicielle d'un système (fichiers de code source, programmes, documents, fichiers de ressource).

- **Diagramme de déploiement**

Il sert à représenter les différents matériels et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent avec eux.

2.6.2 La vue dynamique (comportementale)

- **Diagramme de cas d'utilisation**

Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système. Il constitue un des diagrammes les plus structurants dans l'analyse d'un système. Il comporte 3 concepts :

- **Cas utilisation** : Un cas d'utilisation représente une unité cohérente d'une fonctionnalité fournie par un système spécifiée par une séquence d'actions que le système peut exécuter en interagissant avec les acteurs du système.
- **Acteur** : Un acteur est un ensemble cohérent de rôles joués par des entités externes qui interagissant avec le système.
- **Relation (includ, extend, généralisation/spécialisation)** : Elle exprime l'interaction existante entre un acteur et un cas d'utilisation.

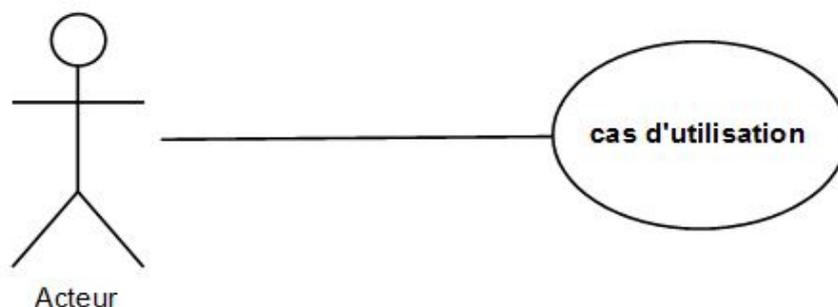


Figure 3 : Exemple d'un cas utilisation.

- **Diagramme de collaboration**

Ce type de diagramme est un croisement entre un diagramme d'objets et un diagramme de séquence. A la différence du diagramme de séquence qui modélise l'interaction dans un format de type ligne-colonne, le diagramme de collaboration emploie une disposition libre des objets tels qu'on les trouve dans un diagramme d'objets. Ceci facilite la vision de toutes les interactions impliquant un objet particulier.

- **Diagramme de séquence**

Il représente le déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs. Le diagramme de séquence peut servir à illustrer un cas d'utilisation. Il comporte les concepts suivants :

- Les concepts de message, d'objet et d'acteur sont déjà définis.
- **Ligne de vie** : représentation de l'existence d'un élément participant dans un diagramme de séquence, il peut être un acteur ou un objet de système.
- **Activation** : Les activations, sont modélisées par des boîtes rectangulaires sur la ligne de vie. Elles indiquent quand l'objet effectue une action.

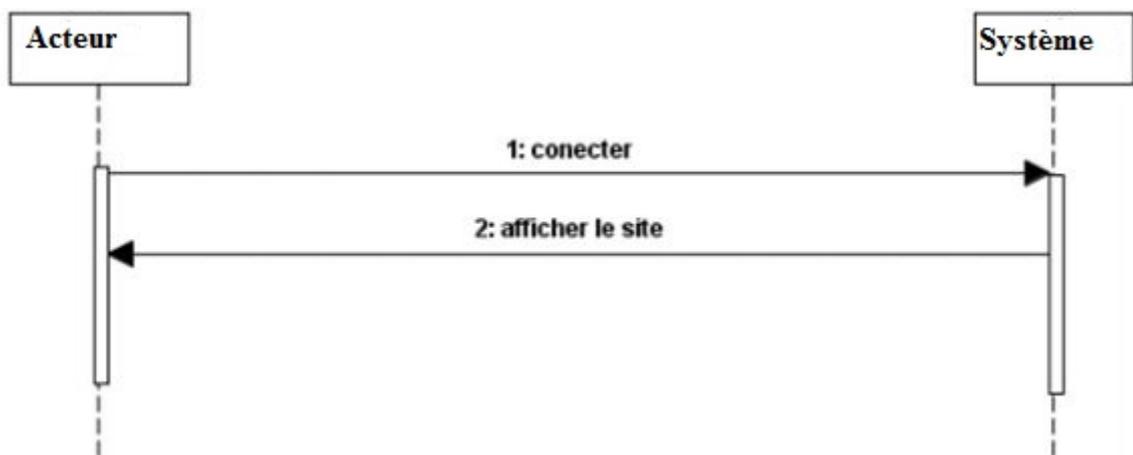


Figure 4 : Exemple d'un diagramme de séquence.

- **Diagramme d'états-transitions**

Il montre la manière dont l'état du système (ou de sous-parties) est modifié en fonction des événements du système. Il comporte les concepts suivants :

➤ **Etat** : La notation de l'état décrit le mode de l'entité, il existe deux types :

❖ **Etat initial** : un état de contrôle à partir duquel le flot débute lorsque l'activité enveloppante est invoquée.



❖ **Etat final** : un pseudo état qui indique que le diagramme d'états-transitions, ou l'état enveloppant, est terminé.



➤ **Transition** : Une transition définit la réponse d'un objet à l'occurrence d'un événement.

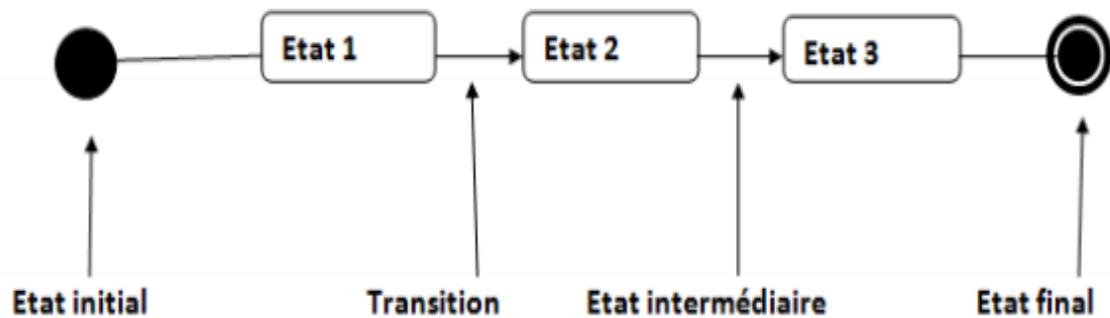


Figure 5 : Exemple d'un diagramme d'états-transitions.

• Diagramme d'activité

Ce diagramme permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. Il comporte 3 concepts :

- **Etat d'activité** : l'état d'activité marque une action faite par un objet. Il est représenté par un rectangle arrondi.
- **Transition**.
- **Couloir** : Dans un diagramme d'activité, on peut placer les activités dans des couloirs qui représentent des systèmes.
- **Etat initial** : l'état initial marque le point d'entrée à la première activité. Il est représenté par un cercle plein. Il ne peut y avoir qu'un seul état initial sur un diagramme.
- **Etat final** : l'état final marque la fin du déroulement des opérations modélisées. Il peut y avoir des états finaux multiples sur un diagramme. Ils sont représentés par un cercle plein entouré d'un autre cercle [6].

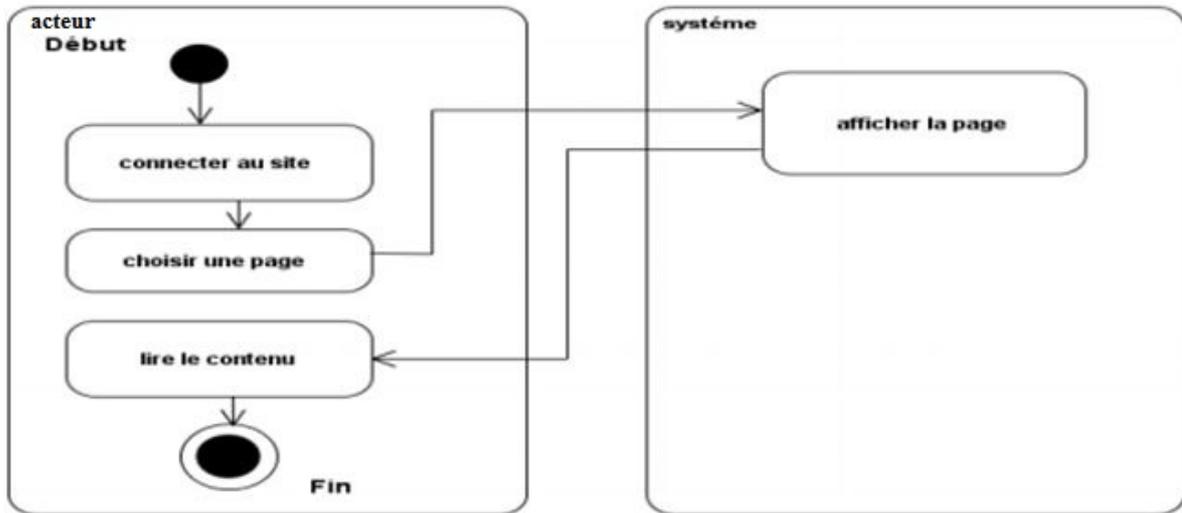


Figure 6 : Exemple d'un diagramme d'activité [7].

3. LE PROCESSUS UNIFIÉ UP

3.1 Définition

Le Processus Unifié (UP) est un processus de développement logiciel construit sur UML (Unified Modeling Language). Il définit une séquence d'étapes, en partie ordonnée, qui concoure à l'obtention d'un système logiciel ou à l'évolution d'un système existant, pour produire des logiciels de qualité qui répondent aux besoins des utilisateurs dans des temps et des coûts prévisibles. Le processus UP est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.

3.2 Les caractéristiques du processus unifié

- **UP est itératif et incrémental**

L'itération est une répétition d'une séquence d'instructions ou d'une partie de programme, un nombre de fois fixé à l'avance ou tant qu'une condition définie n'est pas remplie, dans le but de reprendre un traitement sur des données différentes [3].

- **UP est centré sur l'architecture**

Tout système complexe doit être décomposé en parties modulaires afin d'en faciliter la maintenance et l'évolution. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML, et pas seulement documentée en texte.

- **UP est guidé par les cas d'utilisation d'UML**

Le but principal d'un système informatique est de satisfaire les besoins de client. Le processus de développement sera donc accès sur l'utilisateur. Les cas d'utilisation permettent d'illustrer ces besoins. Ils détectent puis décrivent les besoin fonctionnels et leur ensemble constitue le modèle de cas d'utilisation qui dicte les fonctionnalités complètes du système.

- **UP est piloté par les risques**

Les risques majeurs du projet doivent être identifiés au plus tôt mais surtout levés le plus rapidement. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.

3.3 Le cycle de vie d'UP

Le processus unifié répète un certain nombre de fois une série de cycles. Tout cycle se conclut par la livraison d'une version du produit aux clients et s'articule en 4 phases :

- Création.
- Elaboration.
- Construction.
- Transition.

Chacune d'entre elles se subdivise à son tour en itérations.

3.4 Les phases d'UP

- **Analyse des besoins**

L'analyse des besoins donne une vue du projet sous forme de produit fini. Cette phase porte essentiellement sur les besoins principaux (du point de vue de l'utilisateur), l'architecture générale du système, les risques majeurs, les délais et les coûts.

- **Elaboration**

L'élaboration reprend les éléments de la phase d'analyse des besoins et les précise pour arriver à une spécification détaillée de la solution à mettre en œuvre. L'élaboration permet de préciser la plupart des cas d'utilisation, de concevoir l'architecture du système et surtout de déterminer l'architecture de référence.

- **Construction**

La construction est le moment où l'on construit le produit. L'architecture de référence se métamorphose en produit complet. Le produit contient tous les cas d'utilisation que les chefs de projet, en accord avec les utilisateurs ont décidé de mettre au point pour cette version.

- **Transition**

Le produit est en version bêta. Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts. Cette phase suppose des activités comme la formation des utilisateurs clients, la mise en œuvre d'un service d'assistance et la correction des anomalies constatées. Tout simplement la phase de transition permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux [5].

4. METHODE DE DEVELOPPEMENT (METHODE BASEE SUR UP)

Le processus UP (Unified Process) constitue un cadre général très complet de processus de développement. Pour réaliser notre projet, nous avons choisi de suivre une méthode simple et générique basée sur le processus UP, et qui comporte les étapes suivantes :

4.1 Expression des besoins

L'expression des besoins comme son nom l'indique, permet de définir les différents besoins :

- Inventorier les **besoins principaux** et fournir une liste de leurs fonctions.
- Recenser les **besoins fonctionnels** (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation.
- Appréhender les **besoins non fonctionnels** (techniques) et livrer une liste des exigences.

Le modèle de cas d'utilisation présente le système du point de vue de l'utilisateur et représente sous forme de cas d'utilisation et d'acteur, les besoins du client.

4.2 Analyse

Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution. Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système. Le modèle d'analyse s'écrit dans le langage des développeurs et peut être considéré comme une première ébauche du modèle de conception. L'analyse des besoins permet d'élaborer le diagramme de classes et le diagramme d'activité.

4.3 Conception

La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation. Elle

détermine les principales interfaces et les transcrit à l'aide d'une notation commune. Elle constitue un point de départ à l'implémentation :

- Elle décompose le travail d'implémentation en sous-systèmes.
- Elle crée une abstraction transparente de l'implémentation.

La conception permet d'élaborer les diagrammes d'interaction qui permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse.

4.4 Implémentation

Permet d'implémenter le système sous forme de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutables et d'autres éléments du même type.

Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous formes de codes sources.

4.5 Test

Les tests permettent de vérifier des résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun [3].

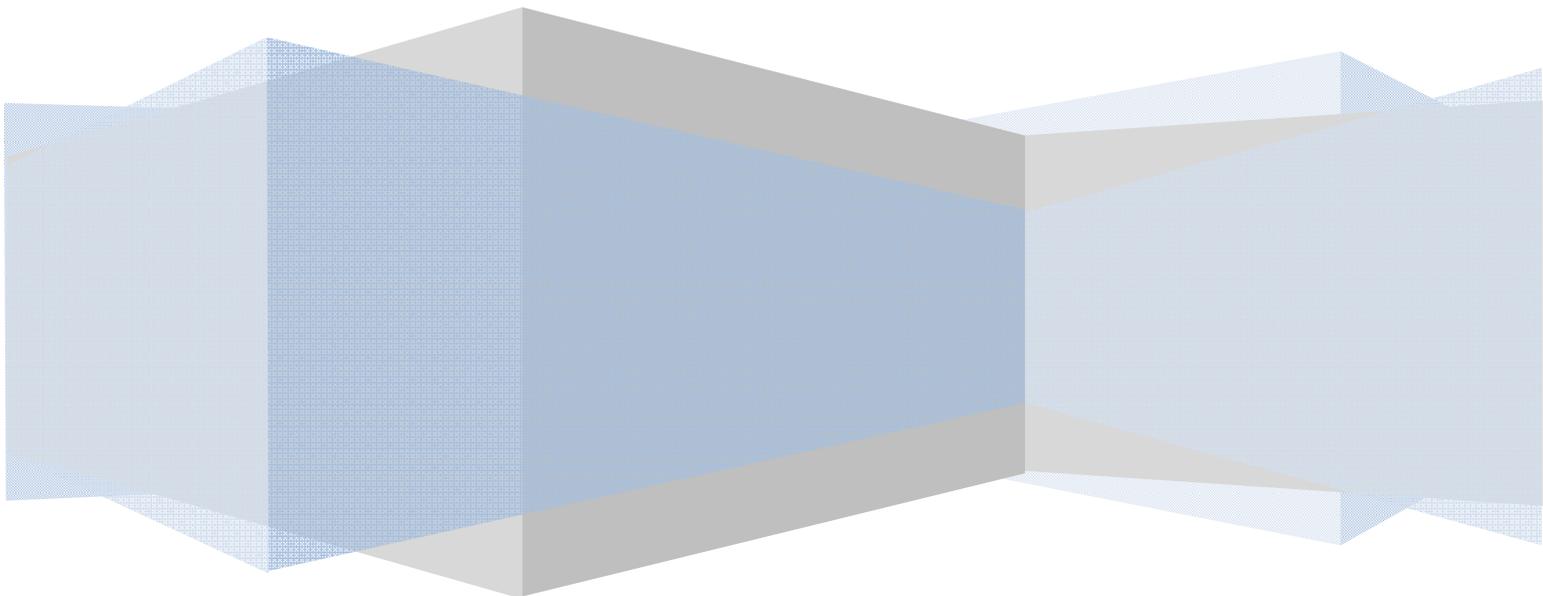
5. CONCLUSION

Nous avons présenté, dans ce chapitre, le langage de modélisation UML et ses diagrammes, le processus unifié UP, ainsi qu'une méthode de développement simple et générique basée sur UP.

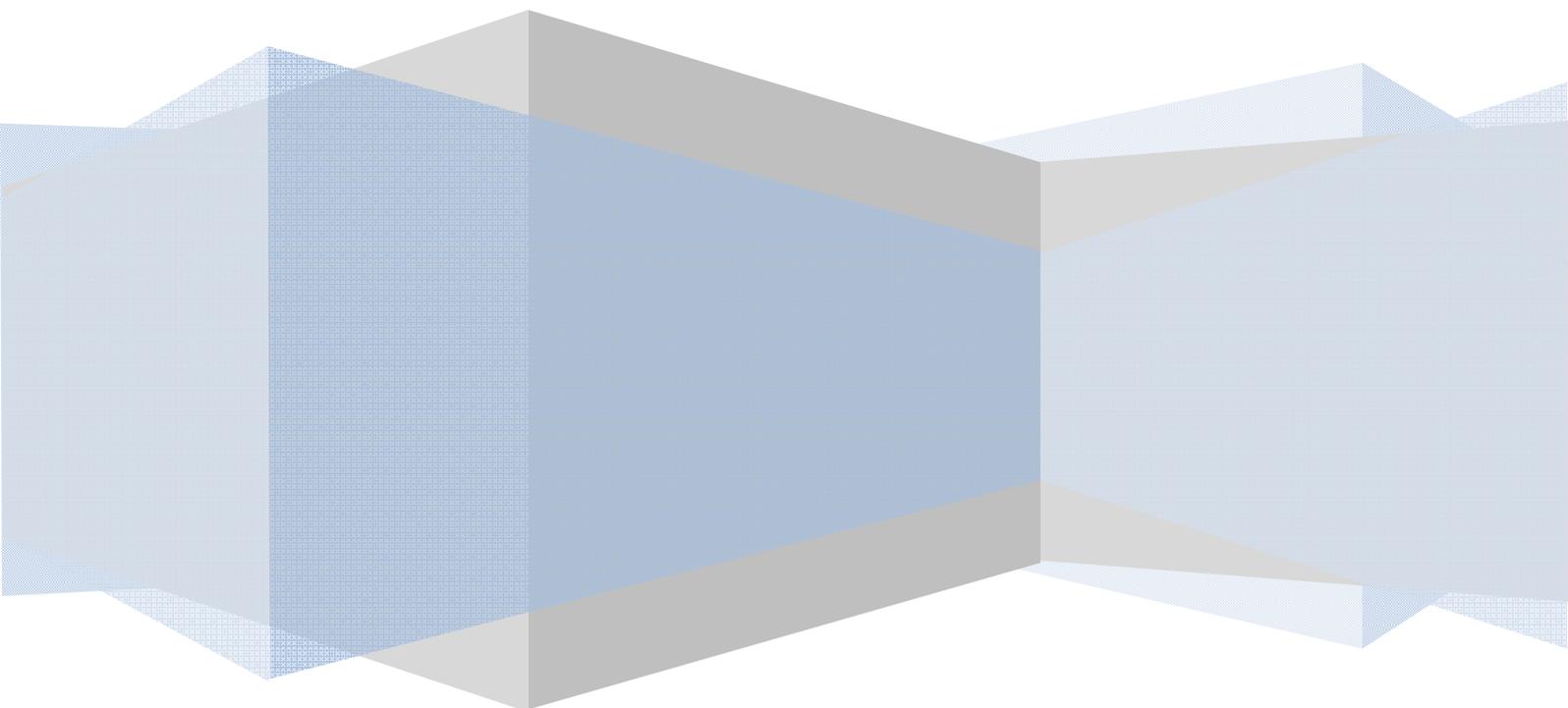
Dans le chapitre suivant, nous allons entamer les étapes d'expression des besoins, d'analyse et de conception définies dans la méthode précédente, pour la conception de notre site web.

PARTIE 2

Etude de cas



Chapitre 03 : **Analyse et conception**



1. INTRODUCTION

Dans ce chapitre nous présentons les étapes d'expression des besoins, d'analyse et de conception définies dans la méthode de développement inspirée des étapes du processus UP que nous avons choisi pour la conduite de notre projet. Les étapes d'analyse et de conception sont basées sur les diagrammes UML.

Nous présentons dans ce chapitre les diagrammes de cas d'utilisation, les diagrammes de séquence et les diagrammes de classes.

2. IDENTIFICATION DES BESOINS

2.1. PRESENTATION DU PROJET

Nous allons créer un site web qui offre des services à des internautes algériens et qui consiste à un site intermédiaire entre les vendeurs de véhicules occasion, les locataires de véhicules et les clients (qui achètent ou louent des véhicules). Ce site permet aux clients de choisir parmi les véhicules exposés par le vendeur ou le locateur.

Les services offerts par notre site sont :

- **La consultation du site**

Le visiteur du site peut consulter toutes les informations sur les véhicules exposés pour la vente ou la location.

Le visiteur peut devenir un acheteur, un vendeur ou un locateur, en créant un compte personnel sur le site. L'inscription dans le site se fait par l'enregistrement des informations personnelles (nom, prénom, adresse mail...). Ses informations varient selon le type de compte choisi.

Dans ce cas, le site permet à l'utilisateur d'accéder à d'autres services. Toute action réalisée selon le type de compte nécessite une authentification.

- **La gestion du compte**

L'utilisateur inscrit dans le site peut modifier les informations de son compte, le supprimer, l'activer ou le désactiver.

- **L'achat de véhicule**

Le site permet au client d'acheter un véhicule exposé dans le site. Le client peut consulter les informations du véhicule et mettre sa proposition de prix ou bien contacter le vendeur pour plus d'information.

Le vendeur consulte les propositions mises sur son véhicule, il choisit le client qui a mis la meilleure proposition et le contacte par email.

- **La location de véhicule**

Le site permet au client de louer un véhicule exposé dans le site. Le client peut consulter les informations du véhicule et le réserver dans une période de temps à condition qu'il ne soit pas réservé dans cette période.

- **La gestion des véhicules de vente**

Le site permet au vendeur de gérer ses annonces de vente. Le vendeur peut ajouter une nouvelle annonce en entrant les informations du nouveau véhicule (modèle, marque, année, kilométrage, etc.). Il peut, aussi, modifier ou supprimer une annonce.

- **La gestion des véhicules de location**

Le site permet au locateur de gérer ses annonces de location. Le locateur peut ajouter une nouvelle annonce en entrant les informations du nouveau véhicule (modèle, marque, année, kilométrage, etc.). Il peut, aussi, modifier ou supprimer une annonce.

2.2 LES GRANDS CHOIX TECHNIQUES

Les outils de développement utilisés pour réaliser ce projet sont :

- **UML** : comme langage de Modélisation.
- **UP** : comme processus de développement logiciel.
- **Client/serveur** : comme architecture du logiciel.
- **HTML/CSS/PHP** : comme langages de programmation.
- **Rapide PHP 2015**: comme éditeur de texte.
- **XAMPP** : Serveur.
- **MySQL** : comme SGBDR (Serveur de Gestion de Base de Données Relationnelle).

2.3 EXIGENCES FONCTIONNELLES

Le visiteur peut :

- Accéder au site sans authentification.
- S'inscrire dans le site comme client, vendeur, ou locateur.
- Consulter les annonces de vente de véhicules.
- Consulter les annonces de location de véhicules.

Le client peut :

- Accéder à son compte après authentification.
- Modifier, activer, désactiver, ou supprimer son compte.
- Consulter les annonces de vente de véhicules.

- Consulter les annonces de location de véhicules.
- Louer un véhicule.
- Mettre une proposition de prix pour acheter un véhicule.
- Contacter un vendeur.

Le vendeur peut :

- Accéder à son compte après authentification.
- Modifier, activer, désactiver, ou supprimer son compte.
- Gérer ses annonces de vente.
- Contacter un client.

Le locateur peut :

- Accéder à son compte après authentification.
- Modifier, activer, désactiver, ou supprimer son compte.
- Gérer ses annonces de location.

2.4 EXIGENCES OPERATIONNELLES

- **La sécurité** : lors de sa connexion le vendeur, le locateur, ou le client, doit être authentifié par une adresse email et un mot de passe.
- **La qualité (interface)** : les éléments de l'interface doivent être clairs et bien organisés avec un ordre attirant qui simplifie la navigation entre les pages.

2.5 DESCRIPTION DU CONTEXTE

➤ Identification des acteurs

- **Visiteur** : c'est la personne qui visite le site pour consulter les informations du site ou créer un compte pour devenir un client, un vendeur, ou bien un locateur.
- **Client** : c'est une personne déjà inscrite dans le site, et qui peut accéder au site pour : consulter les annonces de vente, consulter les annonces de location, mettre une proposition, ou bien louer un véhicule.
- **Vendeur** : c'est une personne déjà inscrite dans le site, et qui peut accéder au site pour : gérer des annonces de vente et gérer les demandes d'achat.
- **Locateur** : c'est une personne déjà inscrite dans le site, et qui peut accéder au site pour : gérer des annonces de location.

➤ Identification des messages

Les messages échangés entre le visiteur et le système :

Visiteur	Système
1. Demande d'inscription	4. Confirmation d'inscription.
2. Demande de consultation des annonces de vente.	5. Affichage de la liste des annonces de vente.
3. Demande de consultation des annonces de location.	6. Affichage de la liste des annonces de location.

Tableau 1 : Messages échangés entre visiteur et système.

Les messages échangés entre le client et le système

Client	Système
7. Demande d'authentification.	12. Affichage des différentes fonctionnalités du site pour le client.
8. Demande de gestion du compte.	13. Confirmation de modification, d'activation, de désactivation, ou de suppression du compte.
9. Demande de mettre proposition.	14. Confirmation de mise de proposition.
10. Demande de contacter vendeur.	15. Accès à la boîte e-mail.
11. Demande de location de véhicule.	16. Confirmation de location de véhicule.

Tableau 2 : Messages échangés entre client et système.

Les messages échangés entre le vendeur et le système :

Vendeur	Système
17. Demande d'authentification.	22. Affichage des différentes fonctionnalités du site pour le vendeur.
18. Demande de gestion du compte.	23. Confirmation de modification, d'activation, de désactivation, ou de suppression du compte.
19. Demande de gestion des annonces de vente.	24. Confirmation d'ajout, de modification, ou de suppression des annonces de vente.
20. Demande de gestion des demandes.	25. Affichage de la liste des demandes.
21. Demande de contacter client.	26. Accès à la boîte e-mail.

Tableau 3 : Messages échangés entre vendeur et système.

Les messages échangés entre le locateur et le système :

Locateur	Système
27. Demande d'authentification.	30. Affichage des différentes fonctionnalités du site pour le locateur.
28. Demande de gestion du compte.	31. Confirmation de modification, de désactivation, ou de suppression du compte.
29. Demande de gestion des annonces de location.	32. Confirmation d'ajout, de modification, ou de suppression des annonces de location.

Tableau 4 : Messages échangés entre locateur et système.

- **Diagramme de contexte**

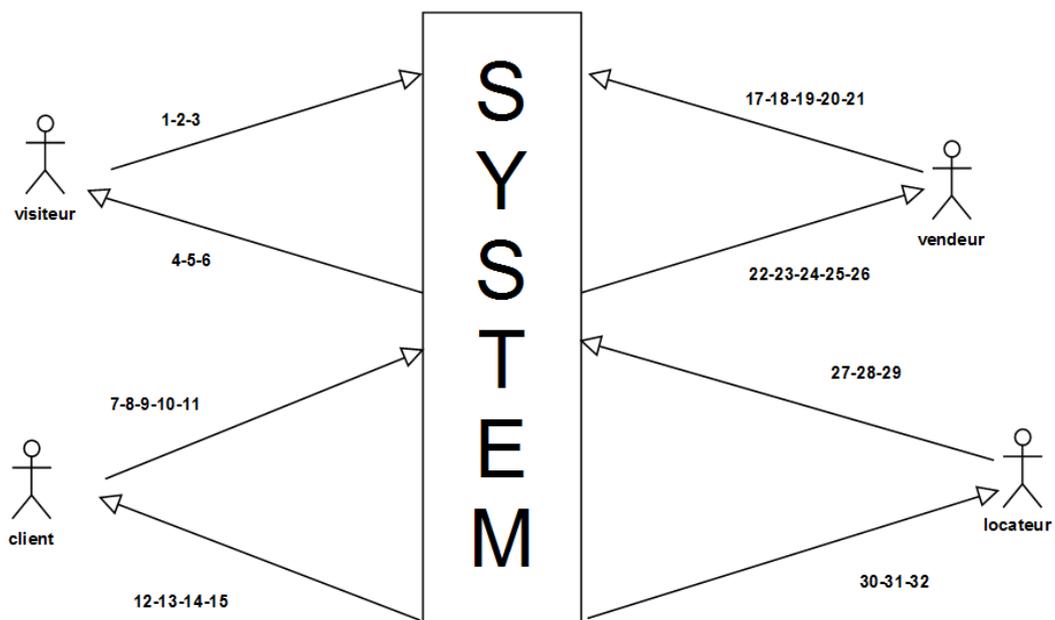


Figure 1 : Diagramme de contexte.

2.6 DIAGRAMME DE CAS D'UTILISATION

- **Visiteur**

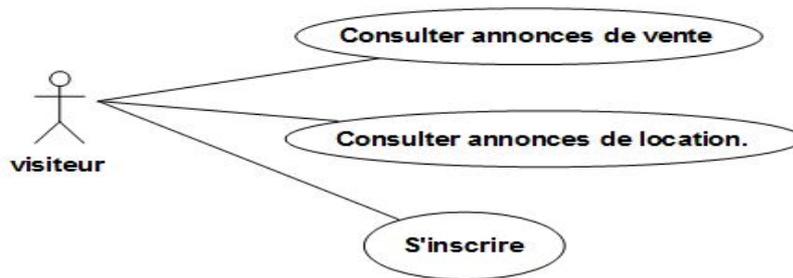


Figure 2 : Diagramme de cas d'utilisation du visiteur.

- **Client**

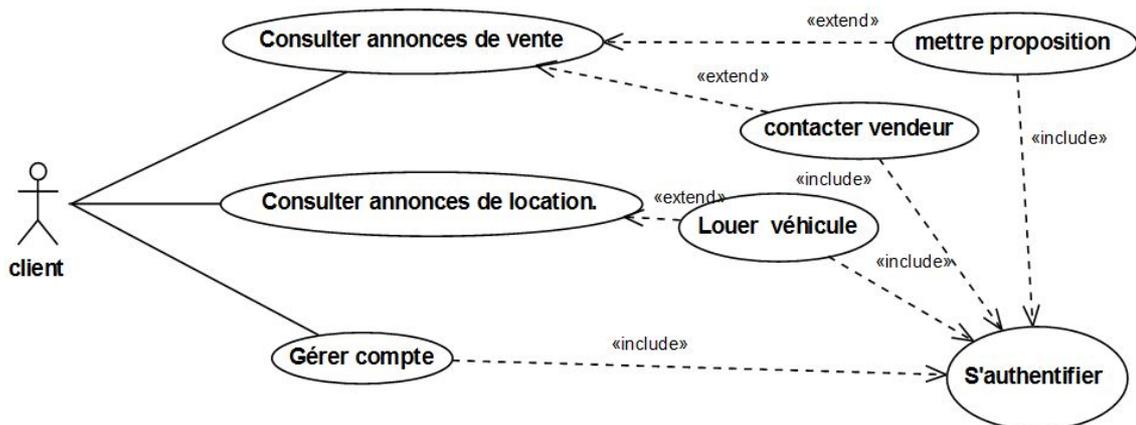


Figure 3 : Diagramme de cas d'utilisation du client.

- **Vendeur**

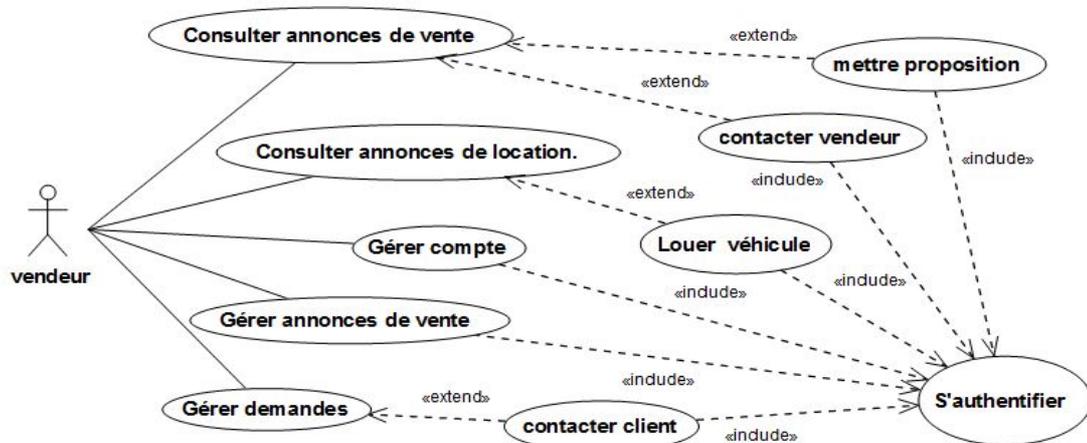


Figure 4 : Diagramme de cas d'utilisation du vendeur.

• **Locateur**

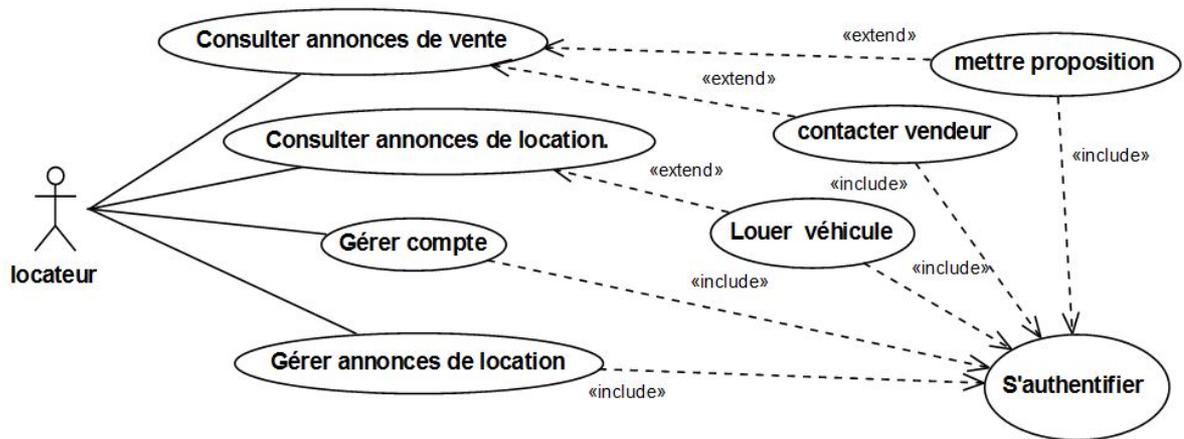


Figure 5 : Diagramme de cas d'utilisation du locateur.

• **Diagramme de cas d'utilisation**

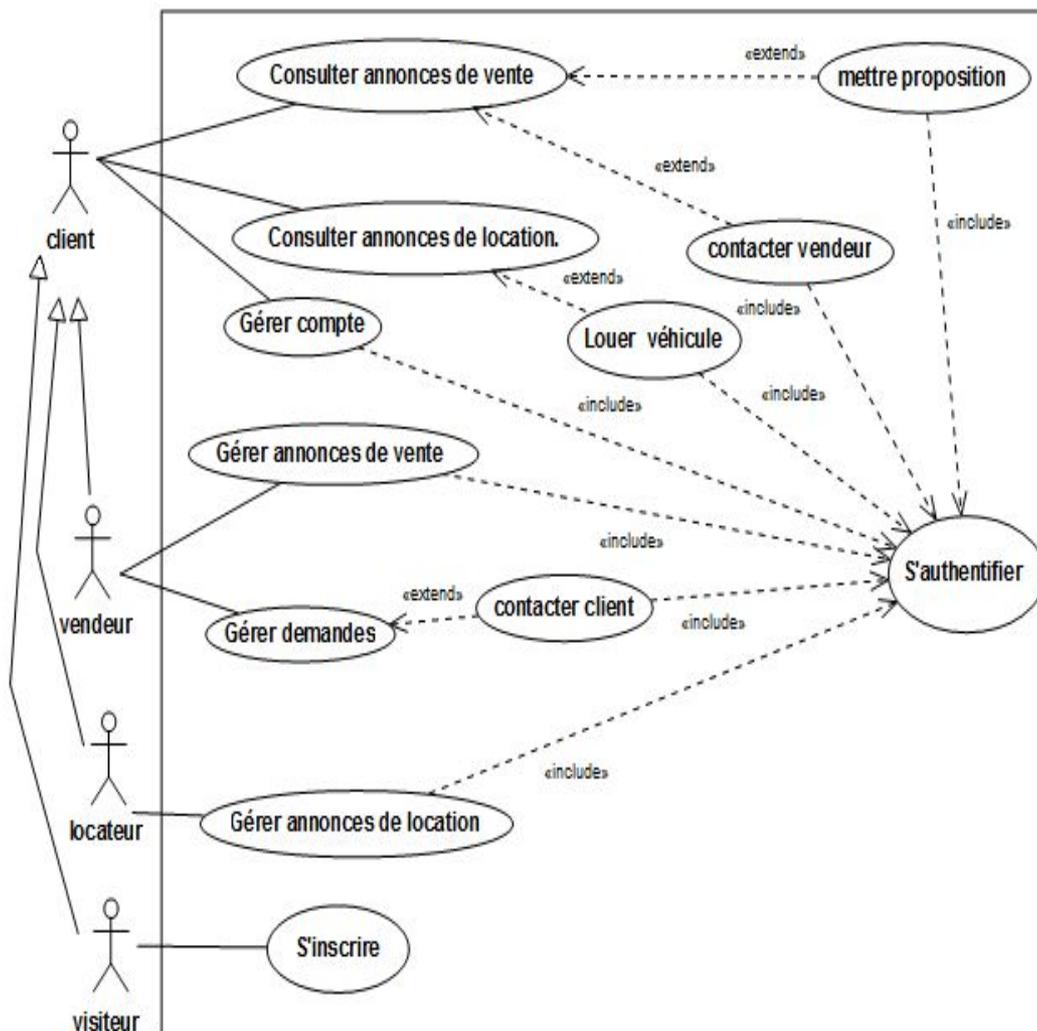


Figure 6 : Diagramme de cas d'utilisation.

2.7 DESCRIPTION DES CAS D'UTILISATION

- **S'inscrire**

Cas d'utilisation	S'inscrire
Acteur	Visiteur
Objectif	Permet au visiteur de s'inscrire dans le site.
Pré-condition	/
Post-condition	Le visiteur est enregistré dans la base de données.
Scénario nominal	<ol style="list-style-type: none"> 1. Le visiteur demande de s'inscrire dans le site. 2. Le système affiche le formulaire d'inscription. 3. Le visiteur entre les informations d'inscription et les envoie. 4. Le système vérifie les informations entrées. 5. Le système enregistre le visiteur dans la base de données et affiche un message de confirmation.
Scénario alternatif	<p>(Alternatif 1)</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur (Cet utilisateur existe déjà). 2. Retourner à l'étape numéro 2 du scénario nominal. <p>(Alternatif 2)</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur (Les informations sont mal remplies). 2. Retourner à l'étape numéro 2 du scénario nominal.

Tableau 5 : Fiche description textuelle du cas (S'inscrire).

- **Consulter annonces de vente**

Cas d'utilisation	Consulter annonces de vente
Acteur	Utilisateurs (visiteur, client, vendeur, locateur)
Objectif	Permet à l'utilisateur de consulter les annonces de vente publiées dans le site.
Pré-condition	/
Post-condition	Afficher les informations.

Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur de mande de consulter les annonces de vente. 2. Le système affiche un choix (Toutes les véhicules, Recherche par critère). 3. L'utilisateur choisit Toutes les véhicules. 4. Le système affiche la liste des annonces enregistrées dans la base de données. 5. L'utilisateur sélectionne une annonce. 6. Le système affiche les détails de l'annonce. 7. Le système affiche les opérations (mettre proposition, contacter vendeur).
Scénario alternatif	<p>(Alternatif 1)</p> <ol style="list-style-type: none"> 1. L'utilisateur choisit Recherche par critère. 2. Le système affiche les champs de recherche (catégorie, type, etc.). 3. L'utilisateur entre les informations selon les critères (catégorie, type, etc.) et les envoie. 4. Le système affiche la liste des annonces correspondantes. 5. Retourner à l'étape numéro 5 du scénario nominal. <p>(Alternatif 2)</p> <ol style="list-style-type: none"> 1. L'utilisateur choisit l'opération Contacter vendeur. 2. Le système appelle le cas d'utilisation « Contacter vendeur ». <p>(Alternatif 3)</p> <ol style="list-style-type: none"> 1. L'utilisateur choisit l'opération Mettre proposition. 2. Le système appelle le cas d'utilisation « Mettre proposition ».

Tableau 6 : Fiche description textuelle du cas (Consulter annonces de vente).

- **Consulter annonces de location**

Cas d'utilisation	Consulter annonces de location
Acteur	Utilisateurs (visiteur, client, vendeur, locateur)
Objectif	Permet à l'utilisateur de consulter les annonces de location publiées dans le site.
Pré-condition	/
Post-condition	Afficher les informations.

Scenario nominal	<ol style="list-style-type: none"> 1. L'utilisateur demande de consulter les annonces de location. 2. Le système affiche un choix (Toutes les véhicules, Recherche par critère). 3. L'utilisateur choisit Toutes les véhicules. 4. Le système affiche la liste des annonces enregistrées dans la base de données. 5. L'utilisateur sélectionne une annonce. 6. Le système affiche les détails de l'annonce. 7. Le système affiche l'opération (Louer véhicule).
Scénario alternatif	<p>(Alternatif 1)</p> <ol style="list-style-type: none"> 1. L'utilisateur choisit Recherche par critère. 2. Le système affiche les champs de recherche (catégorie, type, etc.). 3. L'utilisateur entre les informations selon les critères (catégorie, type, etc.) et les envoie. 4. Le système affiche la liste des annonces correspondantes. 5. Retourner à l'étape numéro 5 du scénario nominal. <p>(Alternatif 2)</p> <ol style="list-style-type: none"> 1. L'utilisateur choisit l'opération Louer véhicule. 2. Le système appelle le cas d'utilisation « Louer véhicule».

Tableau 7 : Fiche description textuelle du cas (Consulter annonces de location).

- **Louer véhicule**

Cas d'utilisation	Louer véhicule
Acteur	Utilisateurs (client, vendeur, locateur)
Objectif	Permet à l'utilisateur de louer un véhicule.
Pré-condition	L'utilisateur doit être authentifié.
Post-condition	Mise à jour de la base de données.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'opération Louer véhicule. 2. Le système demande à l'utilisateur de saisir les dates (début et fin) de location. 3. L'utilisateur saisie les informations et les envoie.

	<p>4. Le système vérifié l'état du véhicule.</p> <p>5. Le système met à jour la base de données et affiche un message (Confirmation de location).</p>
Scénario alternatif	<p>(Alternatif 1)</p> <p>1. Le système affiche un message d'erreur (Ce véhicule n'est pas disponible en ces dates).</p> <p>2. Retourner à l'étape numéro 2 du scénario nominal.</p>

Tableau 8 : Fiche description textuelle du cas (Louer véhicule).

• **Gérer compte**

Cas d'utilisation	Gérer compte
Acteur	Utilisateurs (client, vendeur, locateur)
Objectif	Permet à l'utilisateur de gérer son compte (modification, suppression, activation, désactivation).
Pré-condition	L'utilisateur doit être authentifié.
Post-condition	Mise à jour de la base de données.
Scénario nominal	<p>1. L'utilisateur demande de gérer son compte.</p> <p>2. Le système affiche un choix (Modifier compte, Supprimer compte, Activer compte, Désactiver compte).</p> <p>3. L'utilisateur choisit Modifier compte.</p> <p>4. Le système affiche le formulaire contenant les informations du compte.</p> <p>5. L'utilisateur modifie les informations et les envoie.</p> <p>6. Le système met à jour la base de données et affiche un message de confirmation.</p>
Scénario alternatif	<p>(Alternatif 1)</p> <p>1. L'utilisateur choisit Supprimer compte.</p> <p>2. Le système met à jour la base de données et affiche un message de confirmation.</p> <p>(Alternatif 2)</p> <p>1. L'utilisateur choisit Désactiver compte.</p> <p>2. Le système met à jour la base de données et affiche un message de confirmation.</p>

	(Alternatif 3) 1. L'utilisateur choisit activer compte. 2. Le système met à jour la base de données et affiche un message de confirmation.
--	--------------------------------------------------------------------------------------------------------------------------------------------------

Tableau 9 : Fiche description textuelle du cas (Gérer compte).

- **S'authentifier**

Cas d'utilisation	S'authentifier
Acteur	Utilisateurs (client, vendeur, locateur)
Objectif	Permet à l'utilisateur d'accéder à sa session.
Pré-condition	L'utilisateur doit être déjà inscrit.
Post-condition	L'utilisateur est authentifié par le système.
Scénario nominal	1. L'utilisateur demande au système de s'authentifier. 2. Le système demande à l'utilisateur de saisir l'identifiant et le mot de passe. 3. L'utilisateur entre l'identifiant et le mot de passe et les envoie. 4. Le système vérifie l'existence de l'utilisateur dans la base de données. 5. Le système affiche la session de l'utilisateur.
Scénario alternatif	1. Le système affiche une erreur (L'utilisateur n'existe pas). 2. Retourner à l'étape numéro 2 du scénario nominal.

Tableau 10 : Fiche description textuelle du cas (S'authentifier).

- **Gérer annonces de vente**

Cas d'utilisation	Gérer annonces de vente
Acteur	Vendeur
Objectif	Permet au vendeur de gérer ses annonces de vente (ajouter, modifier, supprimer).
Pré-condition	Le vendeur doit être authentifié.
Post-condition	Mise à jour de la base de données.
Scénario nominal	1. Le vendeur demande de gérer ses annonces. 2. Le système affiche un choix (Ajouter annonce, Modifier annonce,

	<p>Supprimer annonce).</p> <ol style="list-style-type: none"> 3. Le vendeur choisit Ajouter annonce. 4. Le système affiche un formulaire pour entrer les informations de l'annonce. 5. Le vendeur entre les informations de l'annonce et les envoie. 6. Le système vérifie les informations entrées. 7. Le système enregistre l'annonce dans la base de données et affiche un message de confirmation.
Scénario alternatif	<p>(Alternatif 1)</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur (Les informations sont mal remplies). 2. Retourner à l'étape numéro 4 du scénario nominal. <p>(Alternatif 2)</p> <ol style="list-style-type: none"> 1. Le vendeur choisit Modifier annonce. 2. Le système affiche la liste des annonces. 3. Le vendeur sélectionne une annonce. 4. Le système affiche le formulaire contenant les informations de l'annonce. 5. Le vendeur modifie les informations et les envoie. 6. Le système met à jour la base de données et affiche un message de confirmation. <p>(Alternatif 3)</p> <ol style="list-style-type: none"> 1. Le vendeur choisit Supprimer annonce. 2. Le système affiche la liste des annonces. 3. Le vendeur sélectionne une annonce et confirme la suppression. 4. Le système met à jour la base de données et affiche un message de confirmation.

Tableau 11 : Fiche description textuelle du cas (Gérer annonces de vente).

• **Gérer annonces de location**

Cas d'utilisation	Gérer annonces de location
Acteur	Locateur
Objectif	Permet au locateur de gérer ses annonces de location (ajouter, modifier, supprimer).

Pré-condition	Le vendeur doit être authentifié.
Post-condition	Mise à jour de la base de données.
Scénario nominal	<ol style="list-style-type: none"> 1. Le locateur demande de gérer ses annonces. 2. Le système affiche un choix (Ajouter annonce, Modifier annonce, Supprimer annonce). 3. Le locateur choisit Ajouter annonce. 4. Le système affiche un formulaire pour entrer les informations de l'annonce. 5. Le locateur entre les informations de l'annonce et les envoie. 6. Le système vérifie les informations entrées. 7. Le système enregistre l'annonce dans la base de données et affiche un message de confirmation.
Scénario alternatif	<p>(Alternatif 1)</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur (Les informations sont mal remplies). 2. Retourner à l'étape numéro 4 du scénario nominal. <p>(Alternatif 2)</p> <ol style="list-style-type: none"> 1. Le locateur choisit Modifier annonce. 2. Le système affiche la liste des annonces. 3. Le locateur sélectionne une annonce. 4. Le système affiche le formulaire contenant les informations de l'annonce. 5. Le locateur modifie les informations et les envoie. 6. Le système met à jour la base de données et affiche un message de confirmation. <p>(Alternatif 3)</p> <ol style="list-style-type: none"> 7. Le locateur choisit Supprimer annonce. 8. Le système affiche la liste des annonces. 9. Le locateur sélectionne une annonce et confirme la suppression. 10. Le système met à jour la base de données et affiche un message de confirmation.

Tableau 12 : Fiche description textuelle du cas (Gérer les annonces de location)

- **Gérer demandes**

Cas d'utilisation	Gérer demandes
Acteur	Vendeur
Objectif	Permet au vendeur de consulter les demandes d'achat sur ses véhicules et de choisir un client.
Pré-condition	Le vendeur doit être authentifié.
Post condition	Le client sera informé.
Scénario nominal	<ol style="list-style-type: none"> 1. Le vendeur demande de consulter les demandes d'achat sur ses véhicules. 2. Le système affiche la liste des annonces. 3. Le vendeur choisit une annonce. 4. Le système affiche la liste des demandes des clients. 5. Le vendeur choisit un client. 6. Le système affiche les opérations (Confirmer, Contacter client). 7. Le vendeur choisit l'opération (Confirmer). 8. Le système envoie un message à la boîte e-mail du client. 9. Le système met à jour la base de données et affiche un message de confirmation.
Scénario alternatif	<ol style="list-style-type: none"> 1. Le vendeur choisit l'opération Contacter Client. 2. Le système appelle le cas d'utilisation « Contacter client ».

Tableau 13 : Fiche description textuelle du cas (Gérer demandes).

- **Mettre proposition**

Cas d'utilisation	Mettre proposition
Acteur	Utilisateurs (client, vendeur, locateur).
Objectif	Permet à l'utilisateur de mettre une proposition pour acheter un véhicule.
Pré-condition	l'utilisateur doit être authentifié
Post-condition	Mise à jour de la base de données.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'opération Mettre proposition. 2. Le système demande à l'utilisateur de saisir le prix.

	<ol style="list-style-type: none"> 3. L'utilisateur entre le prix et l'envoi. 4. Le système met à jour la base de données et affiche un message (Veuillez attendre une réponse vendeur dans 24 h).
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tableau 14 : Fiche description textuelle du (Mettre proposition).

• **Contacteur vendeur**

Cas d'utilisation	Contacteur vendeur
Acteur	Utilisateurs (client, vendeur, locateur).
Objectif	Permet à l'utilisateur de contacter le vendeur d'un véhicule.
Pré-condition	L'utilisateur doit être authentifié
Post-condition	/
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur demande de Contacter le vendeur. 2. Le système affiche l'adresse e-mail du vendeur. 3. L'utilisateur sélectionne l'adresse pour contacter le vendeur. 4. Le système donne accès à la messagerie électronique.

Tableau 15 : Fiche description textuelle du (Contacteur vendeur).

• **Contacteur client**

Cas d'utilisation	Contacteur client
Acteur	Vendeur.
Objectif	Permet au vendeur de contacter un client.
Pré-condition	Le vendeur doit être authentifié.
Post-condition	/
Scénario nominal	<ol style="list-style-type: none"> 1. Le vendeur demande de Contacter le client. 2. Le système affiche l'adresse e-mail du client. 3. Le vendeur sélectionne l'adresse pour contacter le client. 4. Le système donne accès à la messagerie électronique.

Tableau 16 : Fiche description textuelle du (Contacteur client).

2.8 LES DIAGRAMMES DE SEQUENCE SYSTEME

➤ S'inscrire

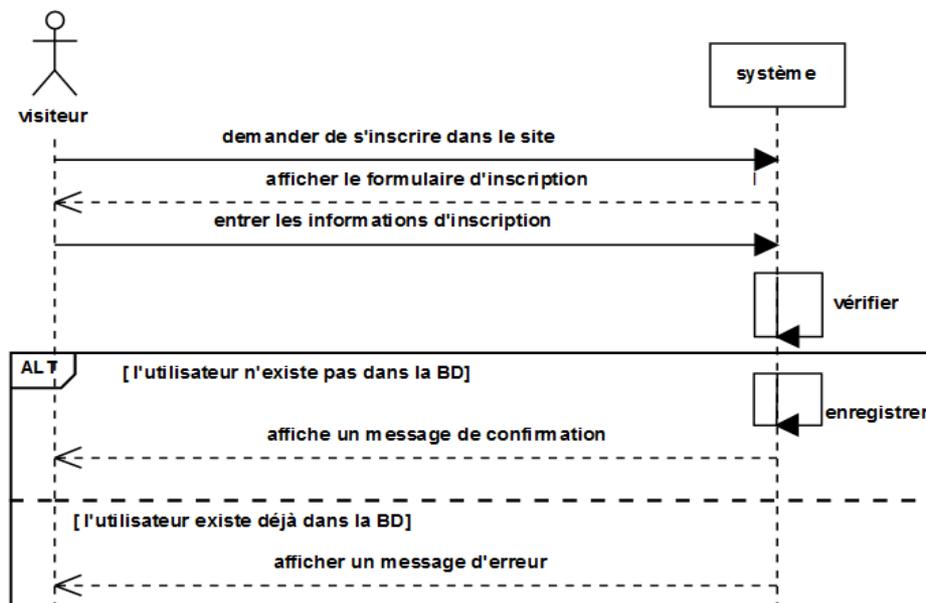


Figure 7 : Diagramme de séquence système du cas d'utilisation (S'inscrire).

➤ S'authentifier

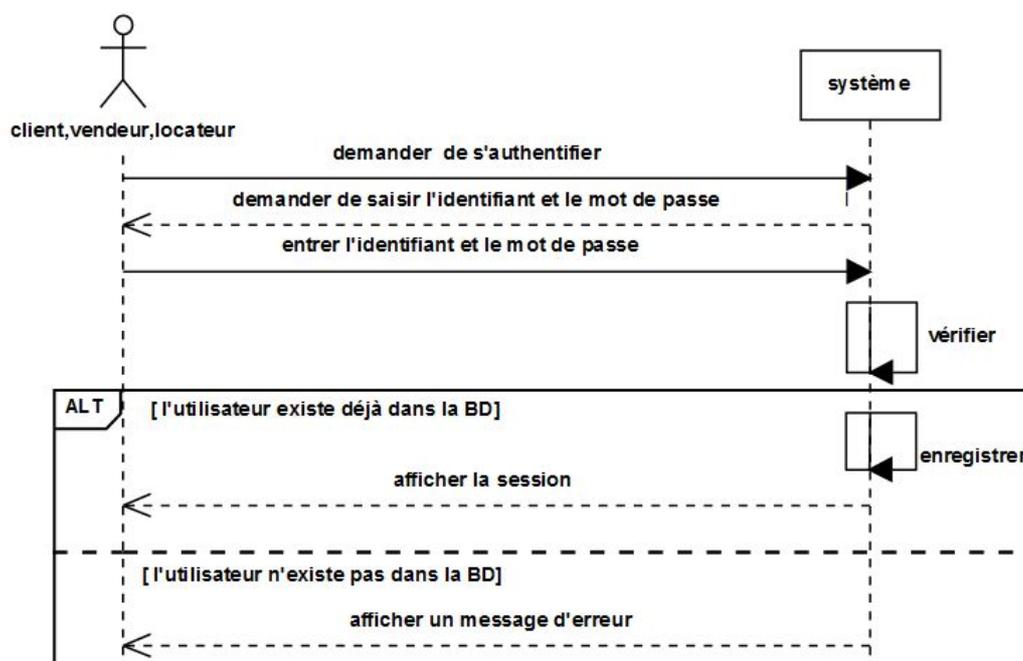


Figure 8 : Diagramme de séquence système du cas d'utilisation (S'authentifier).

➤ **Contacteur client**

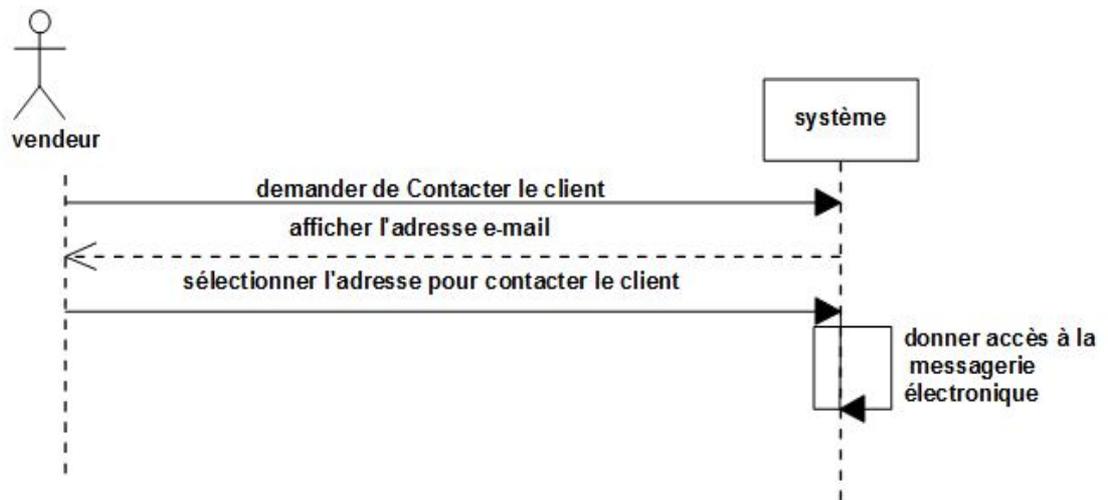


Figure 9 : Diagramme de séquence système du cas d'utilisation (Contacter client).

➤ **Gérer compte**

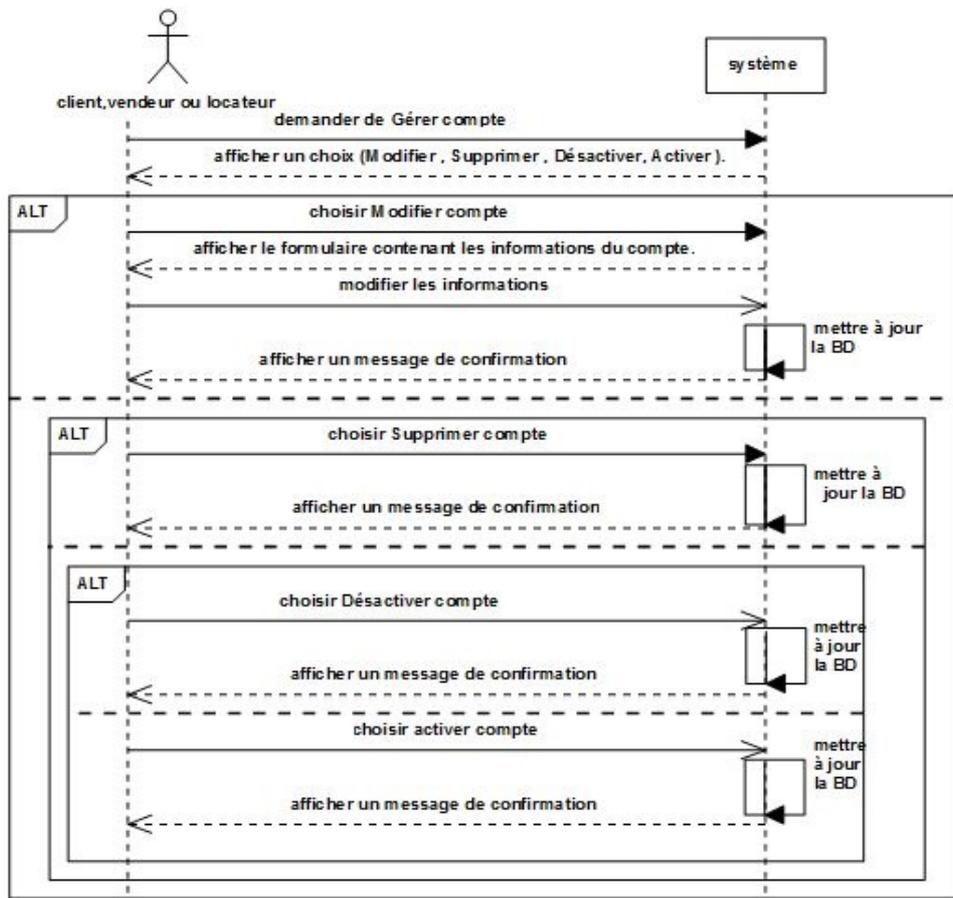


Figure 10 : Diagramme de séquence système du cas d'utilisation (Gérer compte).

➤ Consulter annonces de vente

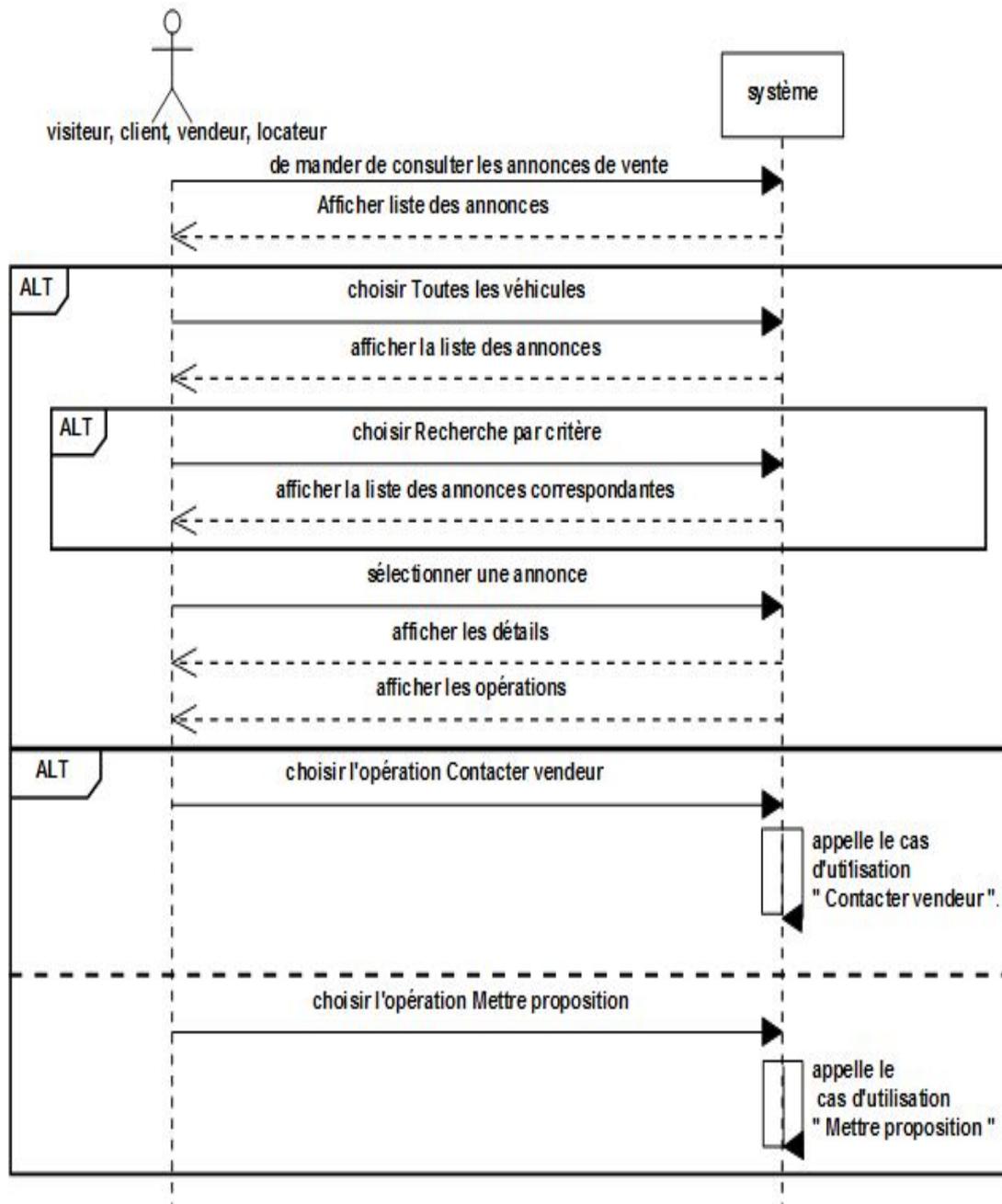


Figure 11 : Diagramme de séquence système du cas d'utilisation (Consulter annonces de vente).

➤ Consulter annonces de location

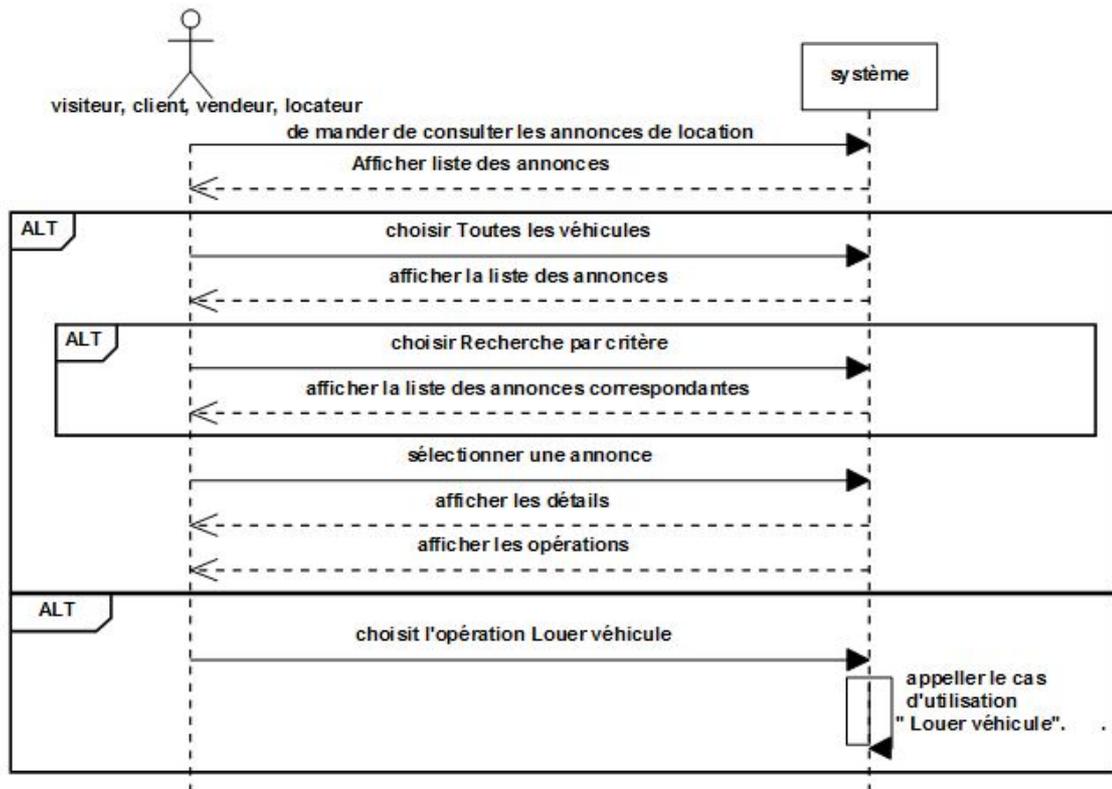


Figure 12 : Diagramme de séquence système du cas d'utilisation (Consulter annonces de location).

➤ Contacter vendeur



Figure 13 : Diagramme de séquence système du cas d'utilisation (Contacter vendeur).

➤ Gérer annonces de vente

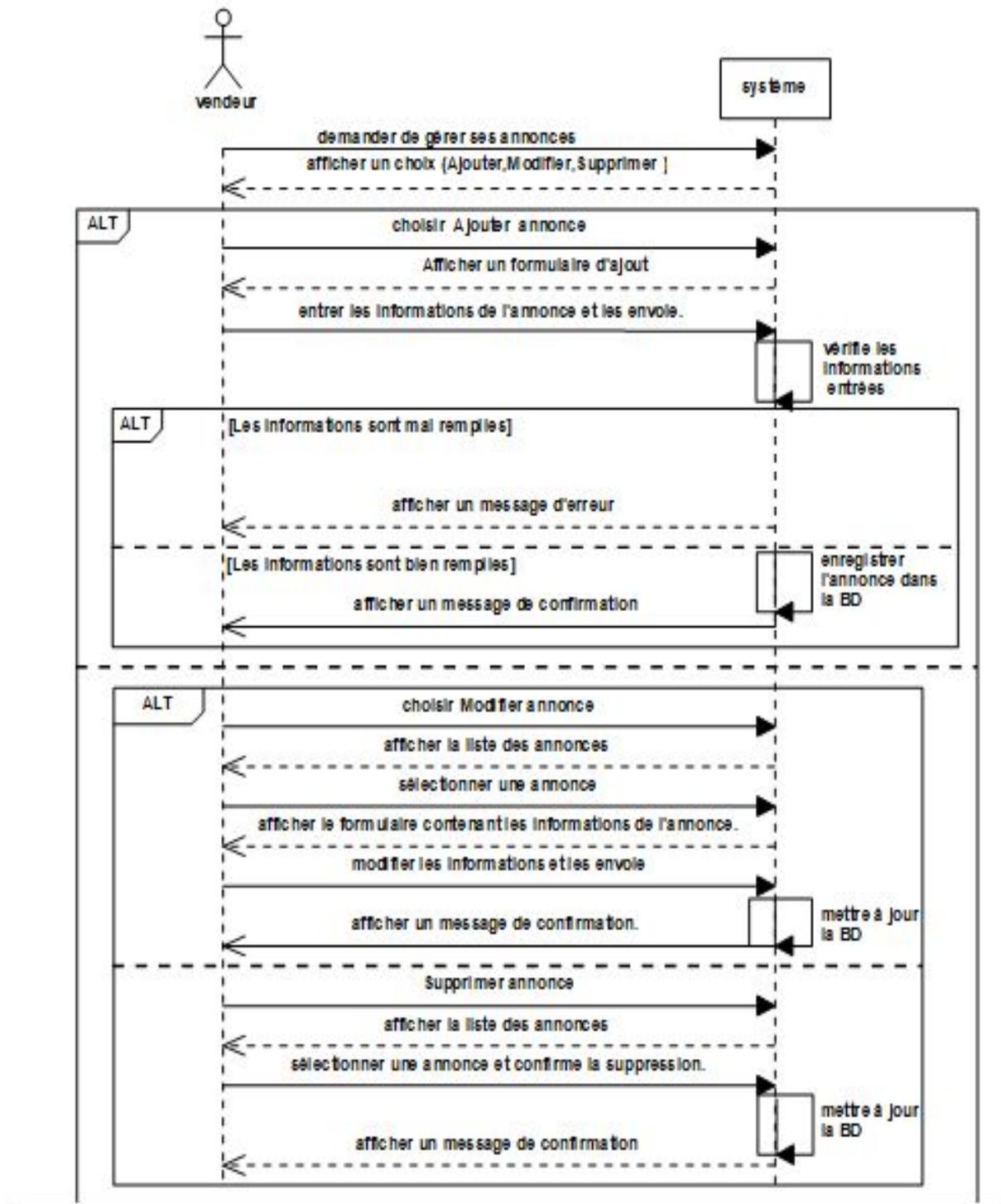


Figure 14 : Diagramme de séquence système du cas d'utilisation (Gérer annonces de vente).

➤ Gérer annonces de location

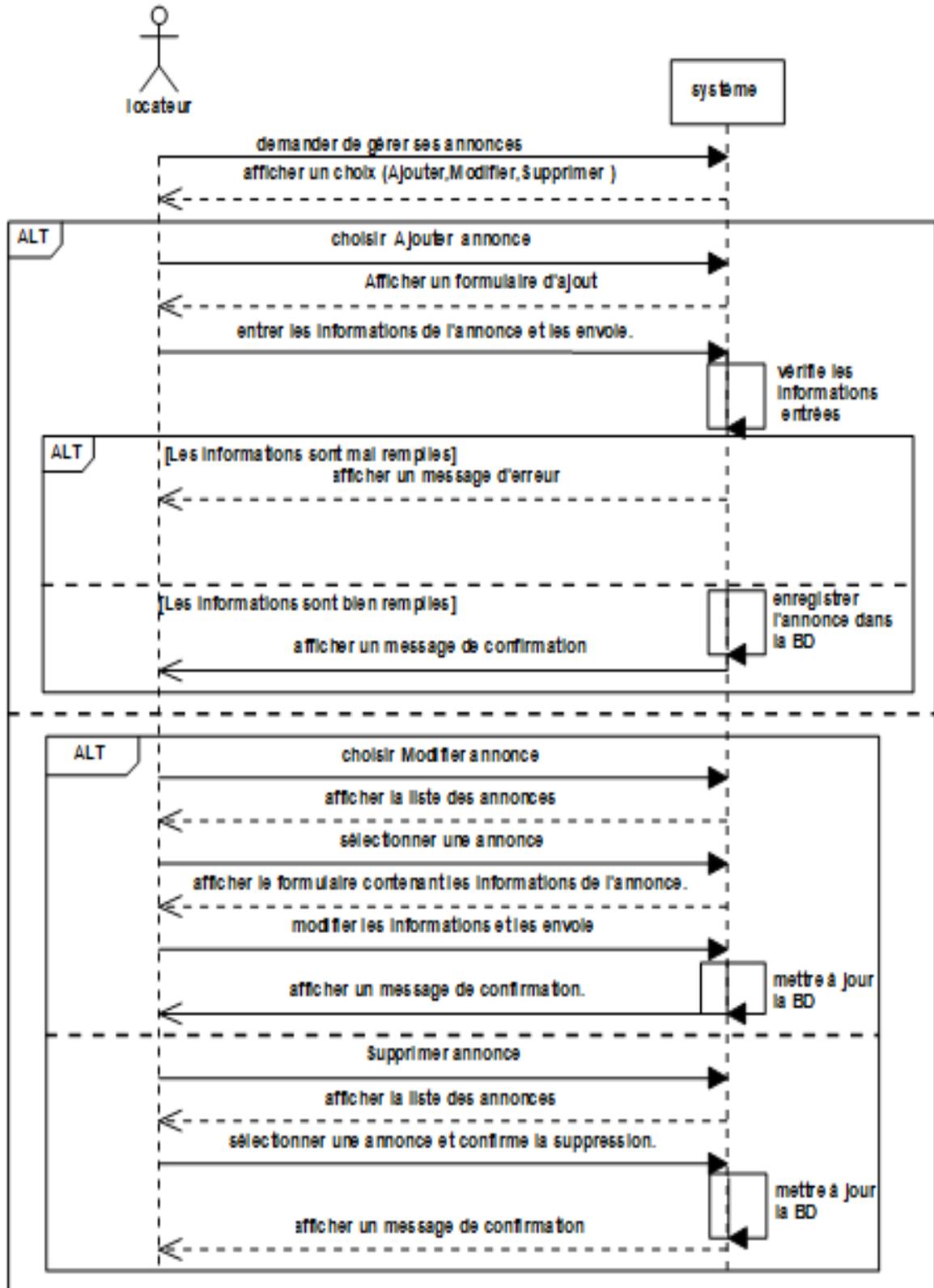


Figure 15 : Diagramme de séquence système du cas d'utilisation

(Gérer annonces de location).

➤ Gérer demandes

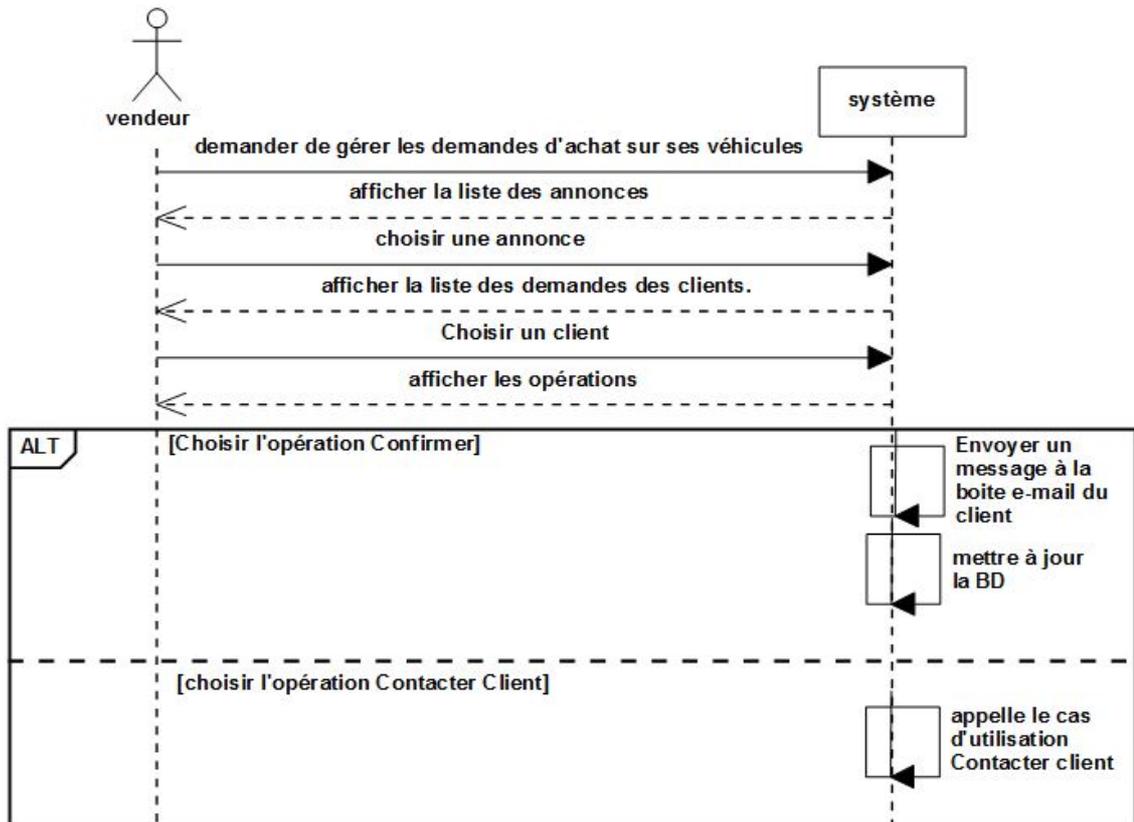


Figure 16 : Diagramme de séquence système du cas d'utilisation (Gérer demandes).

➤ Mettre proposition

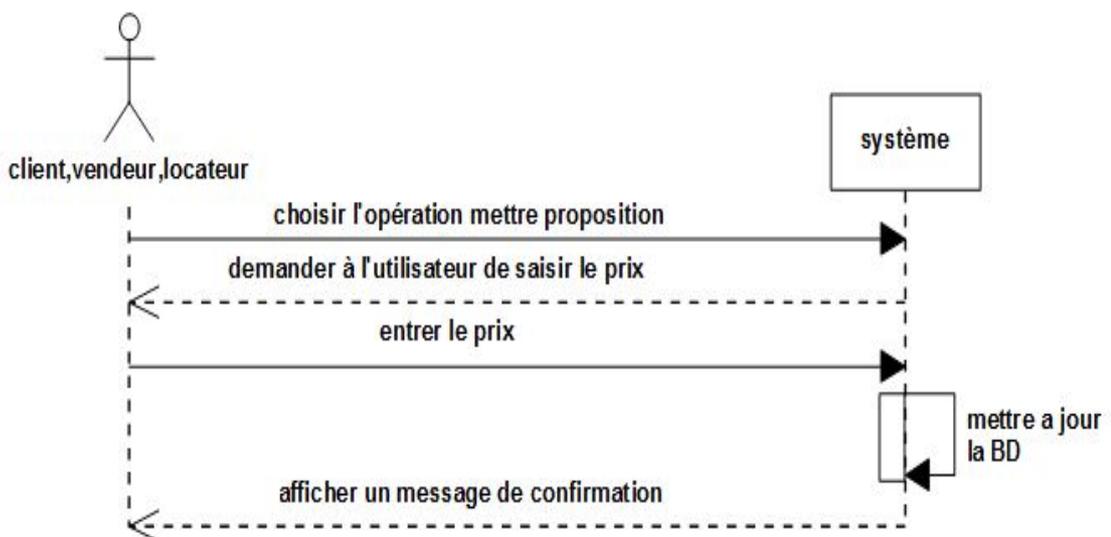


Figure 17 : Diagramme de séquence système du cas d'utilisation (Mettre proposition).

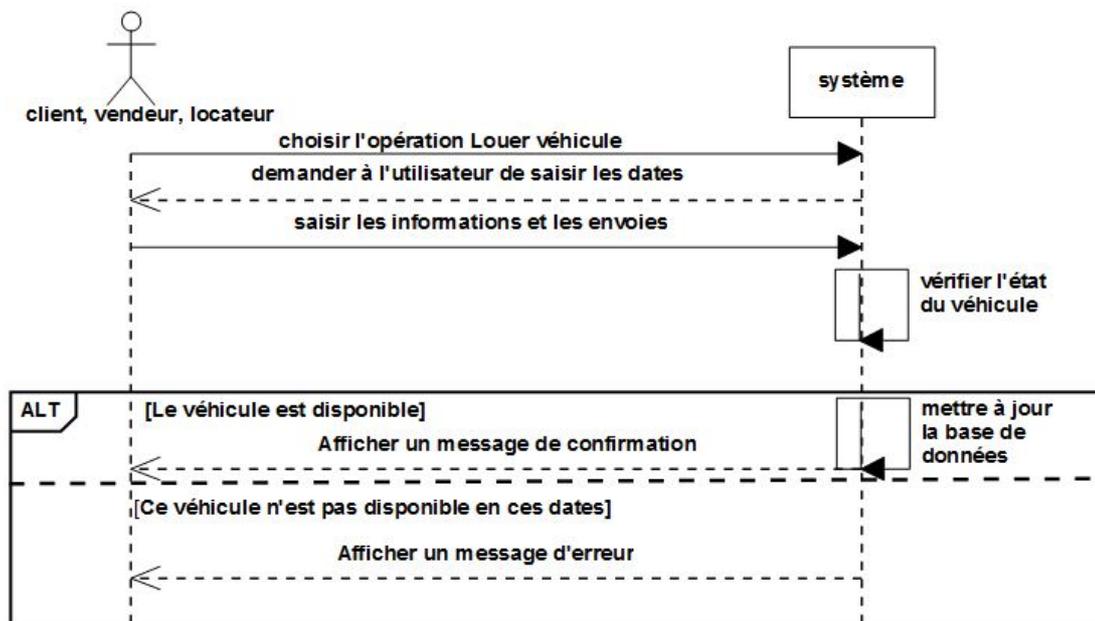
➤ **Louer véhicule**

Figure 18 : Diagramme de séquence système du cas d'utilisation (Louer véhicule).

3. ANALYSE DU DOMAINE

3.1 IDENTIFICATION DES CONCEPTS DU DOMAINE

Nous allons prendre les cas d'utilisations un par un et nous poser pour chacun la question suivante : quel sont les concepts métier qui participent à ce cas d'utilisation ?

➤ **S'inscrire**

Visiteur.

➤ **S'authentifier**

Client, vendeur, locateur.

➤ **Consulter annonces de vente**

Visiteur, client, vendeur, locateur.

➤ **Consulter annonces de location**

Visiteur, client, vendeur, locateur.

➤ **Gérer annonces de vente**

Vendeur.

➤ **Gérer annonces de location**

Locateur.

➤ **Contacteur vendeur**

- Client, vendeur, locateur.
- **Contacteur client**
Vendeur.
- **Gérer demandes**
Vendeur.
- **Mettre proposition**
Client, vendeur, locateur.
- **Gérer compte**
Client, vendeur, locateur.
- **Louer véhicule**
Client, vendeur, locateur.

3.2. MODELE DE DOMAINE

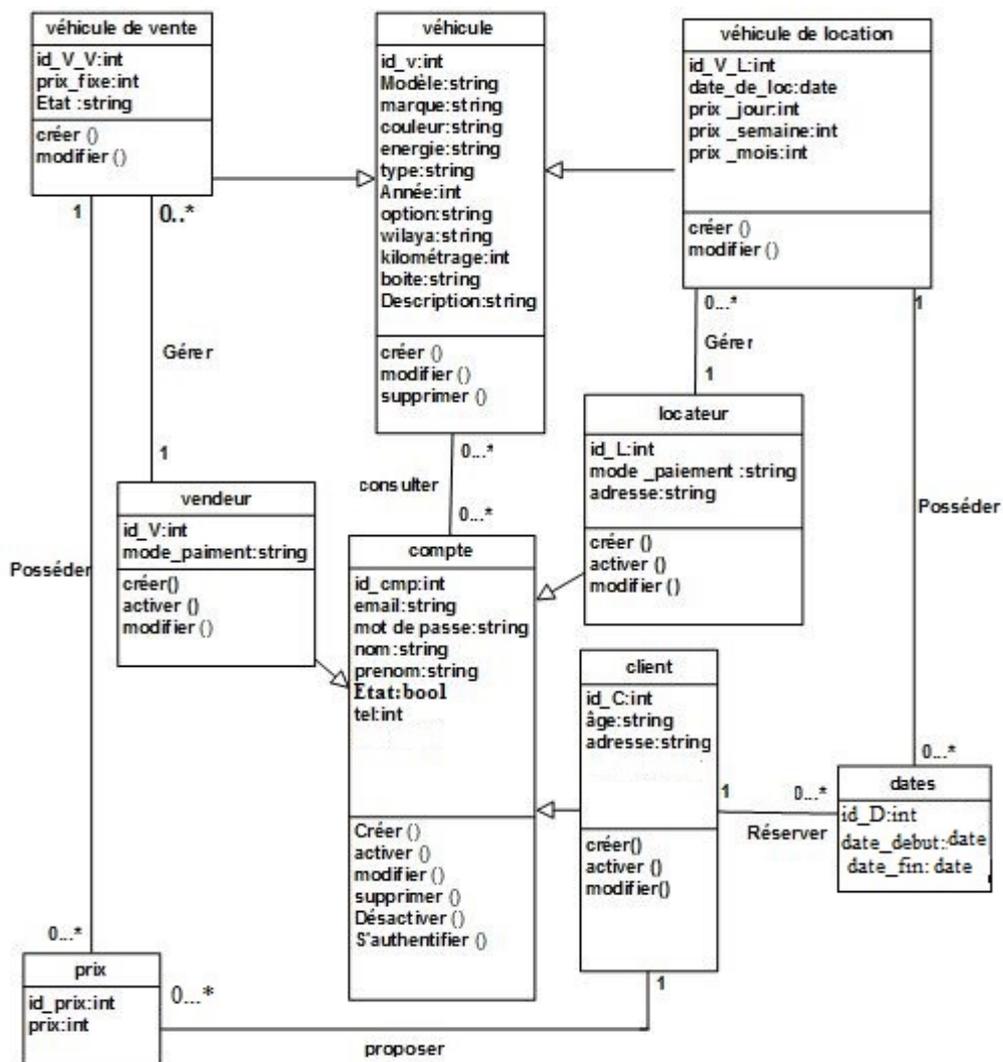


Figure 19 : Diagramme du modèle de domaine.

➤ S'inscrire

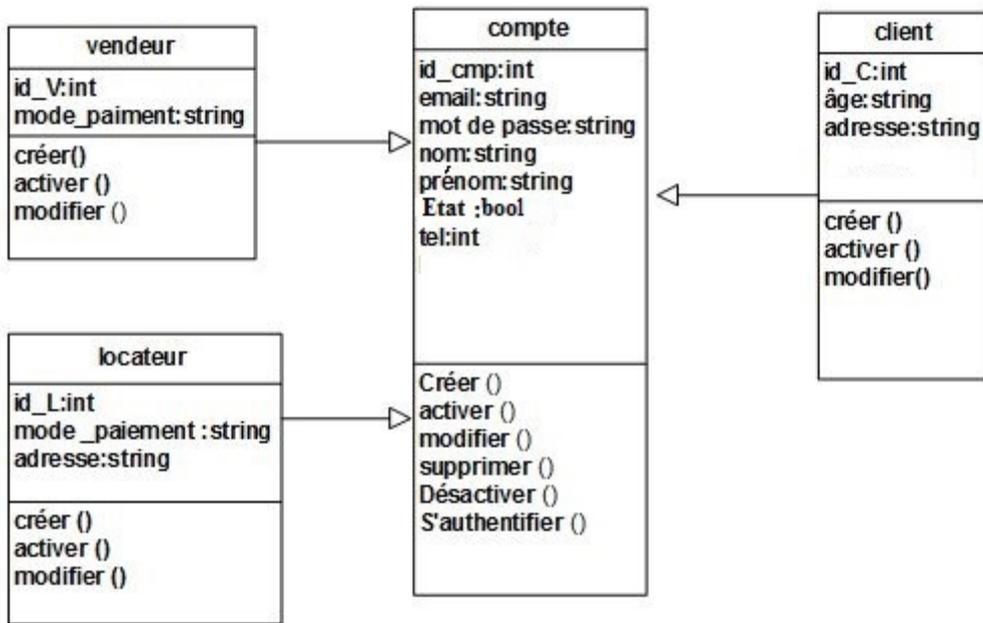


Figure 20 : Diagramme du modèle de domaine (S'inscrire).

➤ S'authentifier

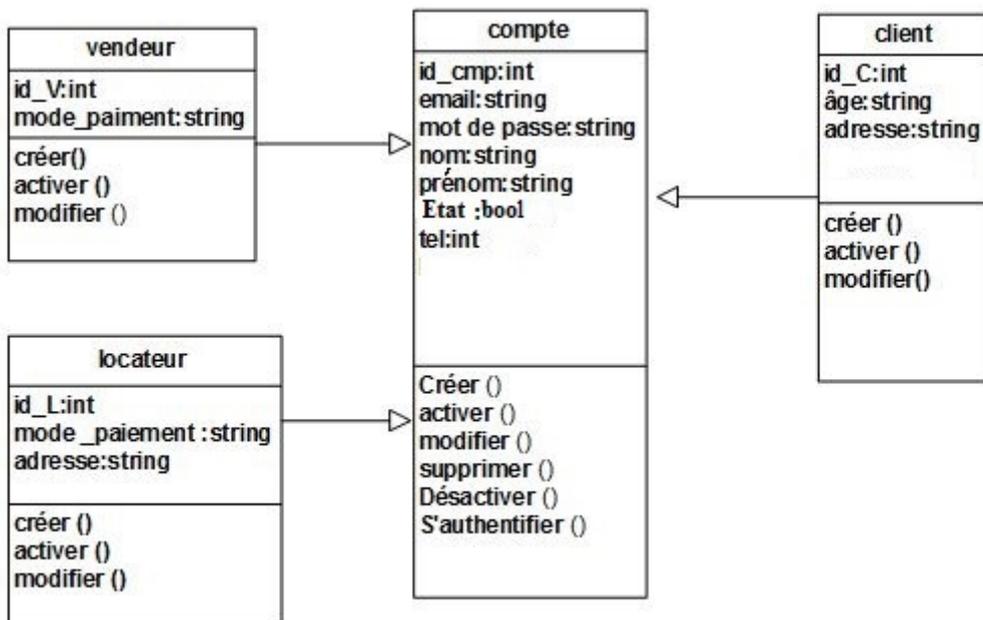


Figure 21: Diagramme du modèle de domaine (S'authentifier).

➤ Consulter annonces de vente

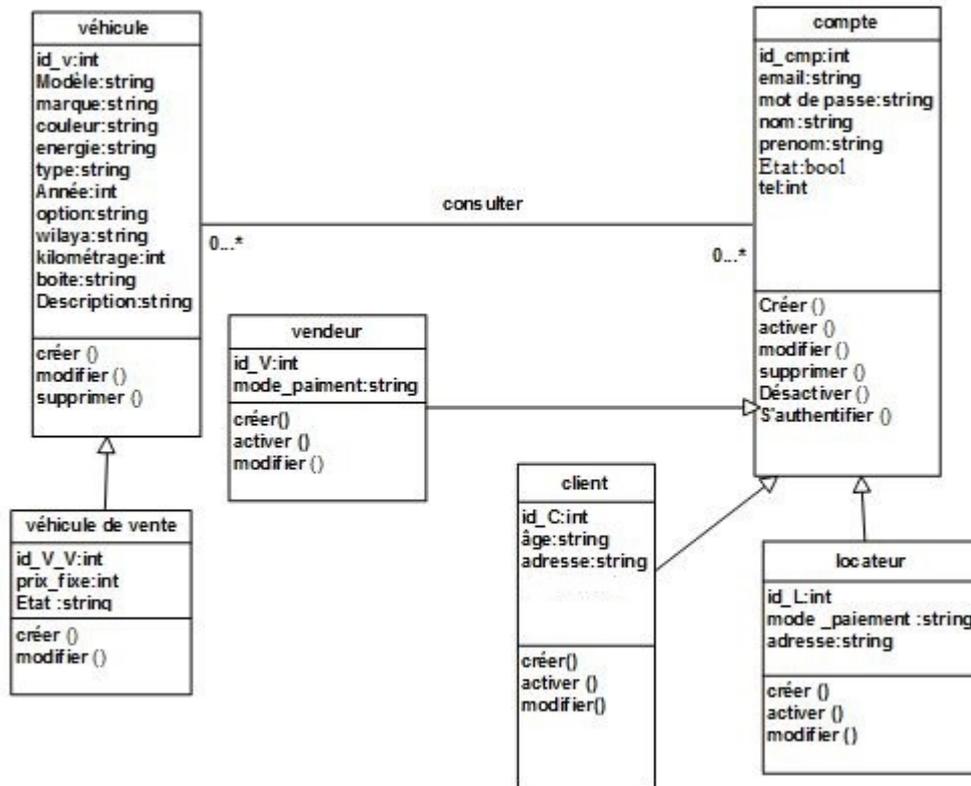


Figure 22 : Diagramme du modèle de domaine (Consulter annonces de vente).

➤ Consulter annonces de location

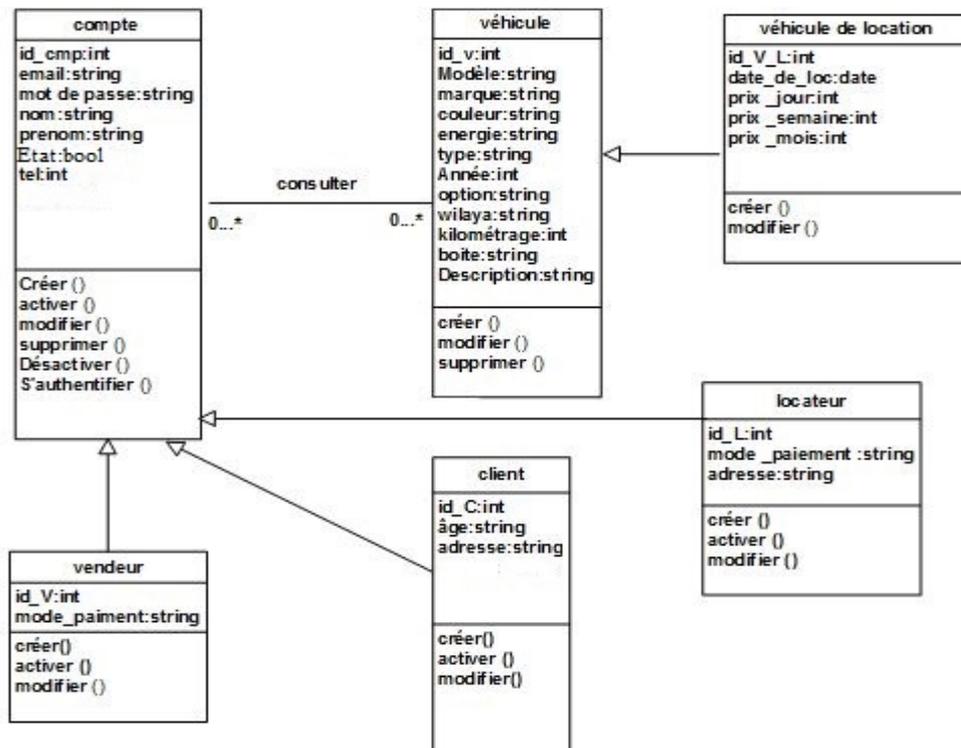


Figure 23 : Diagramme du modèle de domaine (Consulter annonces de location).

➤ Contacter vendeur

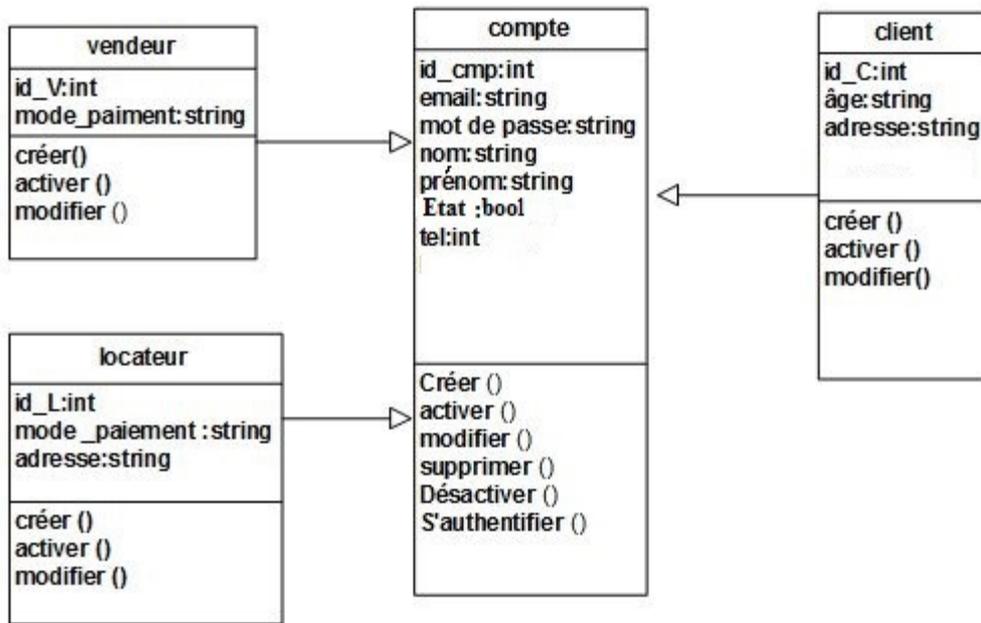


Figure 24 : Diagramme du modèle de domaine (Contacter vendeur).

➤ Louer véhicule

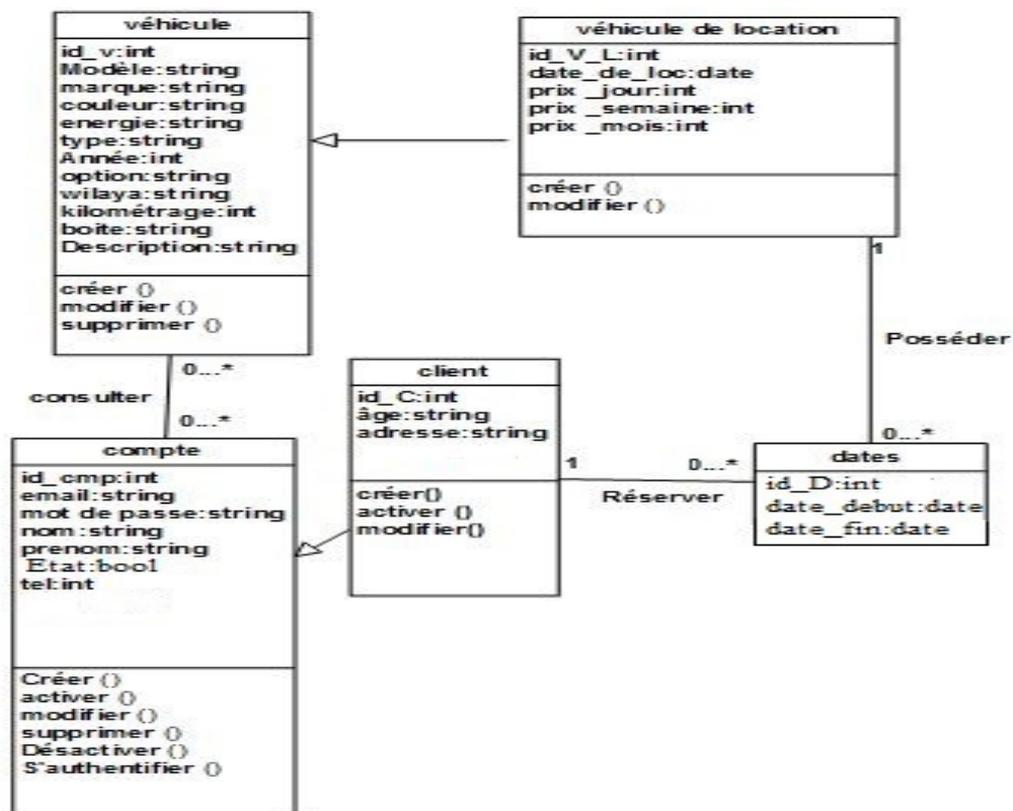


Figure 25 : Diagramme du modèle de domaine (Louer véhicule).

➤ Mettre proposition

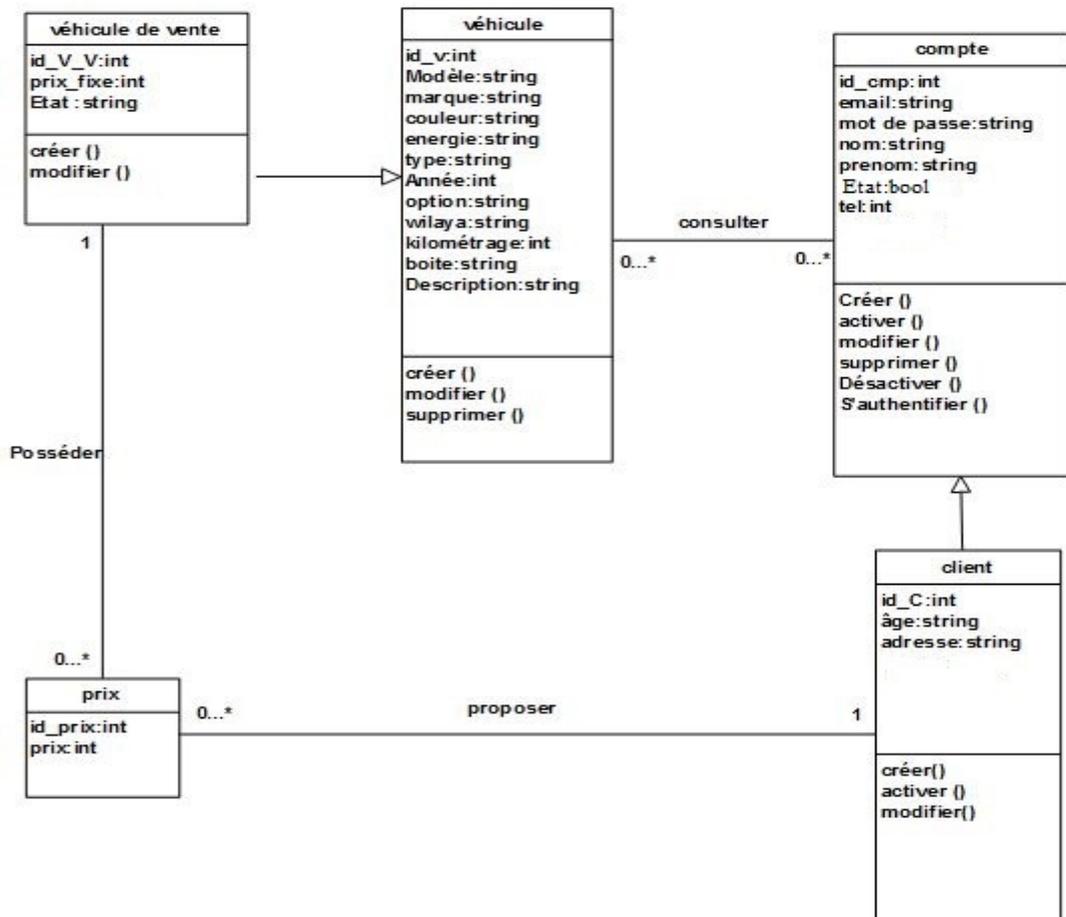


Figure 26 : Diagramme du modèle de domaine (Mettre proposition).

➤ Gérer compte

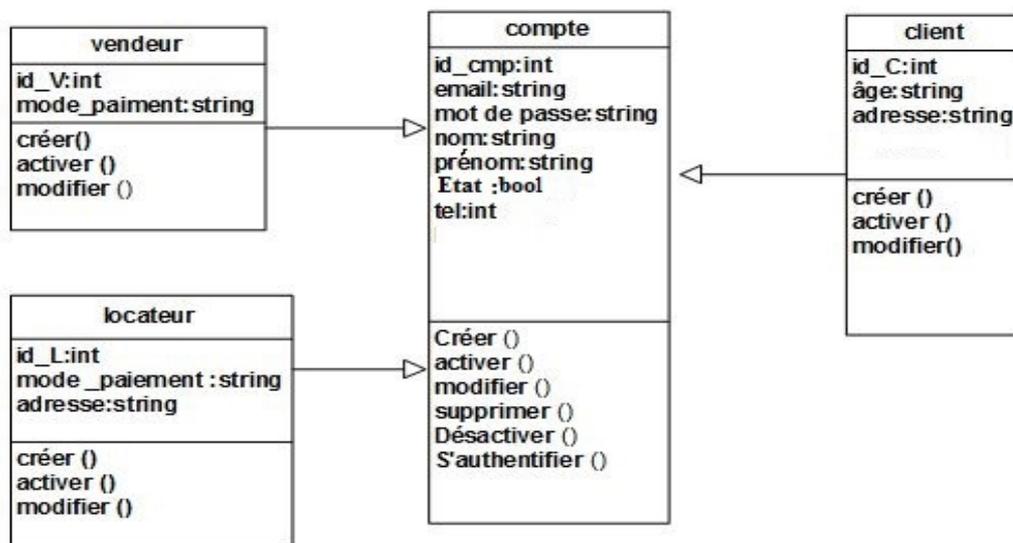


Figure 27 : Diagramme du modèle de domaine (Gérer compte).

➤ **Contacteur client**

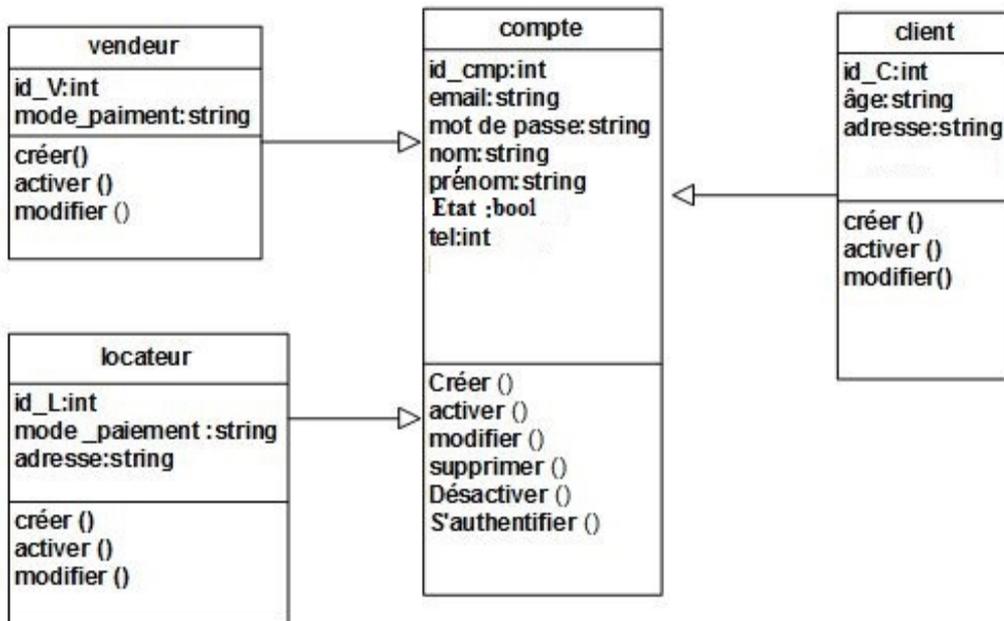


Figure 28 : Diagramme du modèle de domaine (Contacteur client).

➤ **Gérer annonces de vente**

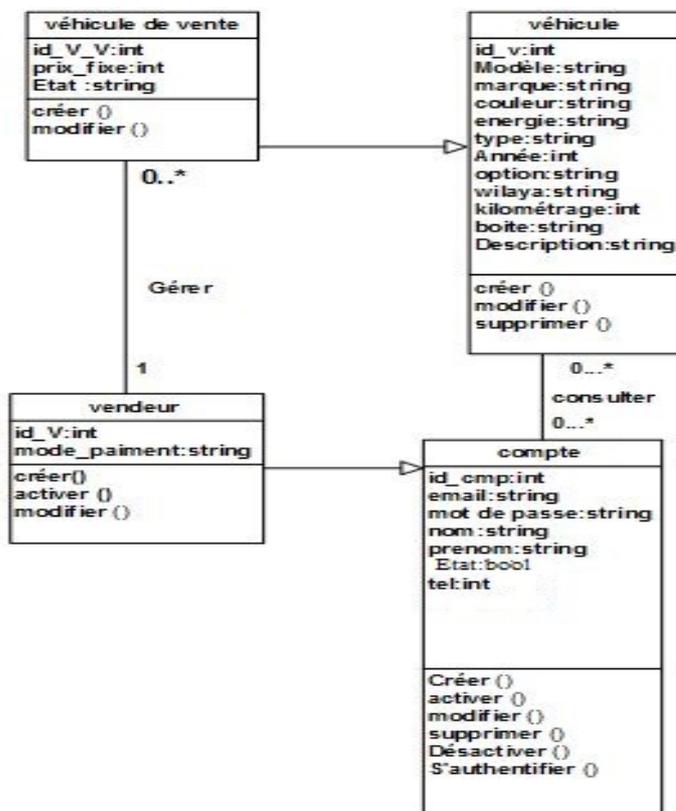


Figure 29 : Diagramme du modèle de domaine (Gérer annonces de vente).

➤ Gérer annonces de location

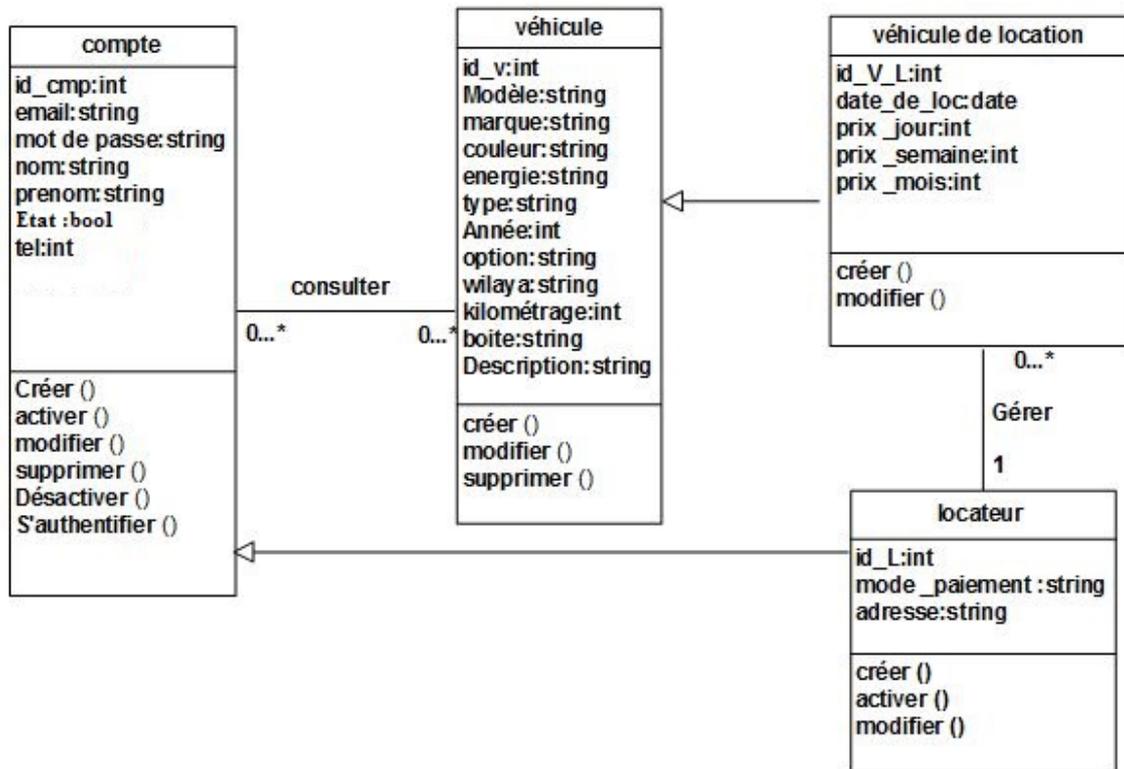


Figure 30 : Diagramme du modèle de domaine (Gérer annonces de location).

➤ Gérer demandes

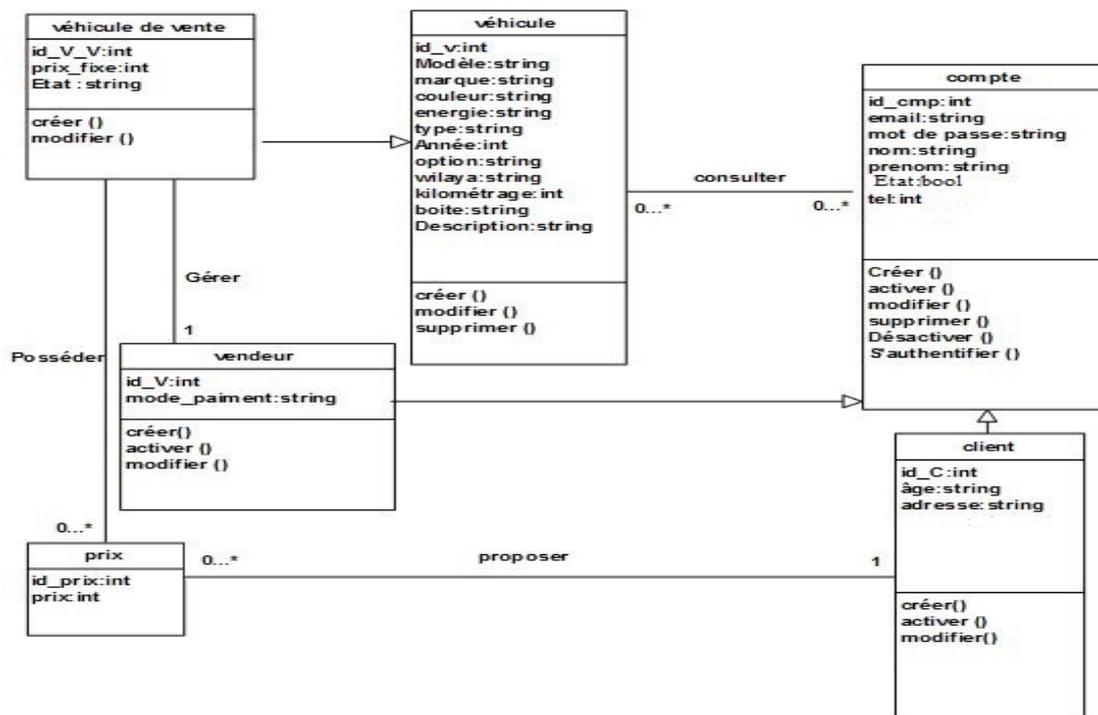


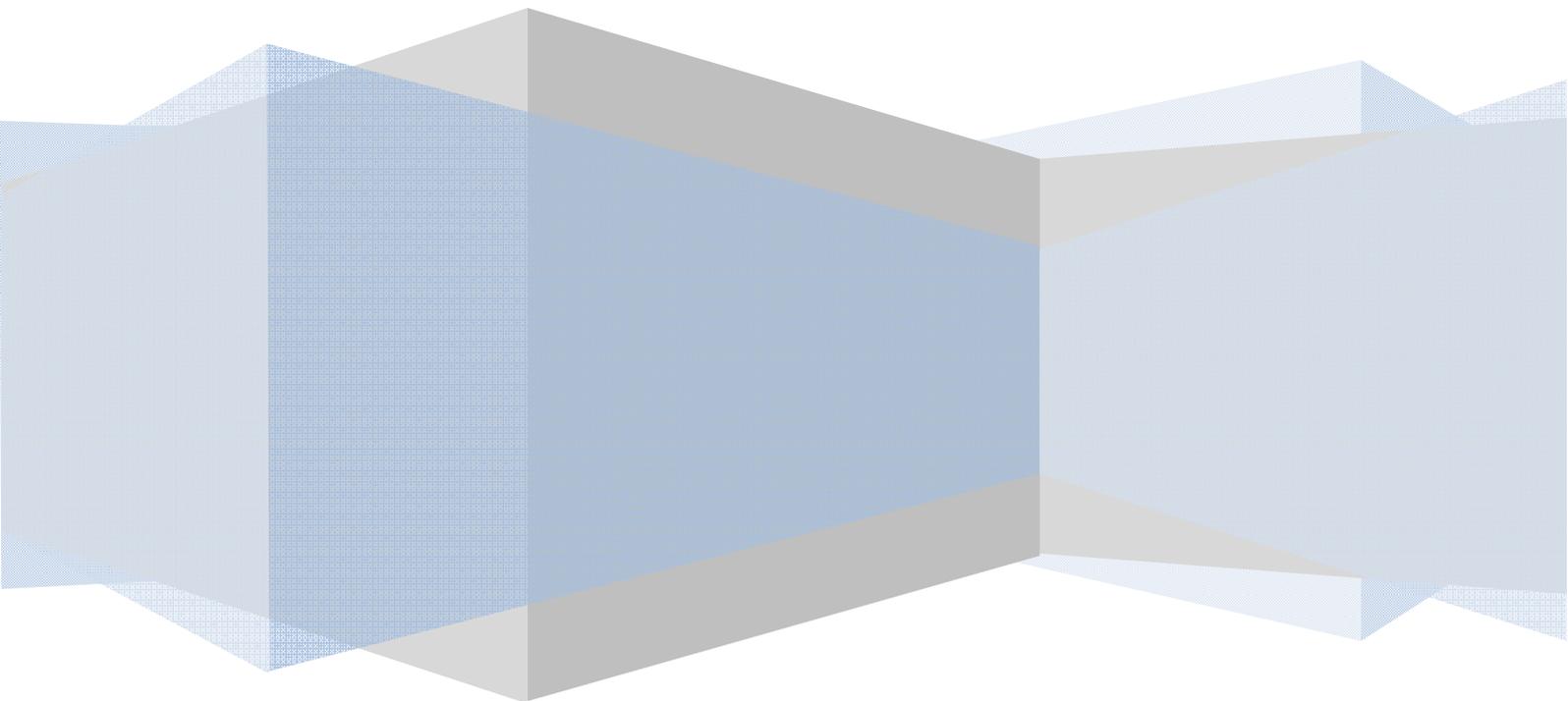
Figure 31 : Diagramme du modèle de domaine (gérer demande)

4. CONCLUSION

Dans ce chapitre, nous avons détaillé l'analyse et la conception de notre site. Pour cela, nous avons élaboré le diagramme de contexte, le diagramme de cas d'utilisation, les diagrammes de séquences, le diagramme de classes et les diagrammes de classes participantes aux cas d'utilisations.

Ce chapitre prépare la phase d'implémentation qui est l'objet du chapitre suivant.

Chapitre 04 : Implémentation



1. INTRODUCTION

Dans ce chapitre, nous faisons une présentation générale des différentes techniques de programmation web. Nous présentons le langage principal de production de pages hypertextes HTML et le langage de mise en forme des sites web CSS. Ainsi que le langage de script utilisé pour développer des sites web interactifs PHP. Nous présentons, aussi, le serveur de bases de données MySQL et le serveur XAMPP.

Ensuite, nous exposons quelques pages de notre site web pour donner une image globale de son fonctionnement.

2. ENVIRONNEMENT DE DEVELOPPEMENT

2.1 LES LANGAGES DE PROGRAMMATION

➤ HTML

HTML (Hypertext Markup Language) est un langage de formatage de document hypertexte. Il sert à définir la structure et le contenu des pages web, donc c'est la base de toute page web.

HTML est un langage permettant de décrire les différents composants d'un document : titres, formes de paragraphes, tableaux, couleurs, insertion d'images, intégration des aides à la navigation, etc. Parmi ses caractéristiques, HTML est un langage facile à utiliser ainsi que son mode de construction est simple (rédigé sous forme de texte à l'aide d'un éditeur de texte tel que : Note pad sous Windows).

Les liens hypertextes sont l'un de ses principaux atouts, un lien permet d'accéder rapidement à un autre emplacement dans un document, ou bien à une autre page sur internet.

Mais son principal inconvénient est le manque de commandes de mise en forme sophistiquées [23].

```

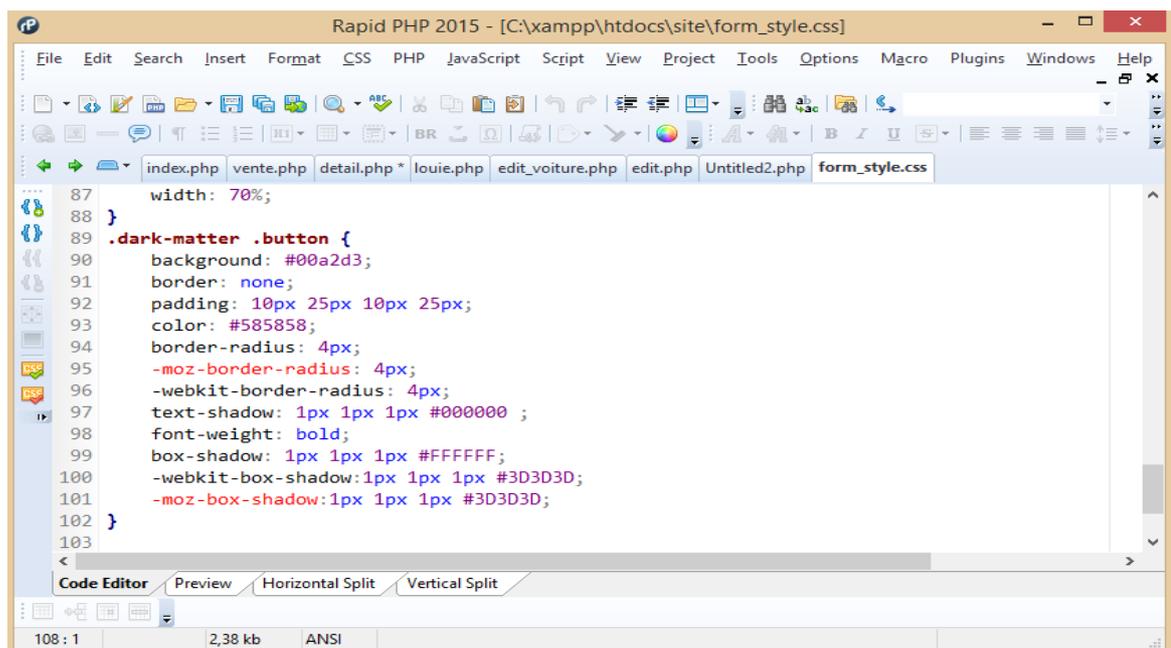
41 <html>
42 <body>
43 <center>
44 <table bgcolor="CCCCCC" border="0"width="70%">
45 <tr>
46 <td width="62%">
47 <object classid="clsid:166B1BCA-3F9C-11CF-8075-444553540000" codebase="http://download.macromedia.com/pub/shockwave/cabs/dir
48 <param name="src" value="images/banner.swf" />
49 <embed src="swf/banner.swf" pluginspage="http://www.adobe.com/shockwave/download/" width="100%" height="150"></embed>
50 </object>
51 </td>
52 </tr>
53 </table>
54 <br>
55 <table bgcolor="123654" width="80%" border="1" align="center">
56 <tr>
57 <td width="8%"align="center"><a href="index.php"><b><font color="cccccc">&rsquo;&rsquo;Home </b> </a></td>
58 <td width="15%"align="center"><a href="vente.php"><b><font color="cccccc">&rsquo;&rsquo;Consulter annonces de vente </b> </a><
59 <td width="16%"align="center"><a href="louie.php"><b><font color="cccccc">&rsquo;&rsquo;Consulter annonces de location </b></
60 <td width="10%"align="center"><a href="register.php"> <b><font color="cccccc">&rsquo;&rsquo;s'inscrire </b></a></td>
61 <td width="10%"align="center"><a href="login.php"><b><font color="cccccc">&rsquo;&rsquo;s'authentifier</b></a></td>
62 <td width="10%"align="center"><a href="contactus.php"method="get"><b><font color="cccccc">&rsquo;&rsquo;Contact Us</b></a></td>
63 </tr>
64 </table><br>
65 <font size="4"color="blue ">merci pour votre confiance...</font>
66 </body>

```

Figure 1: Une partie du code HTML de style de formulaire de notre site.

➤ CSS

CSS (Cascading Style Sheets) est le langage de mise en forme des sites web. Alors que le HTML permet d'écrire le contenu des pages web et de les structurer, le langage CSS s'occupe de la mise en forme et de la mise en page. C'est en CSS que l'on choisit notamment la couleur, la taille des menus et bien d'autres choses encore [14].



```

87 width: 70%;
88 }
89 .dark-matter .button {
90 background: #00a2d3;
91 border: none;
92 padding: 10px 25px 10px 25px;
93 color: #585858;
94 border-radius: 4px;
95 -moz-border-radius: 4px;
96 -webkit-border-radius: 4px;
97 text-shadow: 1px 1px 1px #000000 ;
98 font-weight: bold;
99 box-shadow: 1px 1px 1px #FFFFFF;
100 -webkit-box-shadow: 1px 1px 1px #3D3D3D;
101 -moz-box-shadow: 1px 1px 1px #3D3D3D;
102 }
103

```

Figure 2 : Une partie du code CSS de style de formulaire de notre site.

➤ PHP

PHP (HyperText preprocessor) est un langage de programmation bien connu pour l'extension des pages web avec des fonctionnalités dynamiques. Tandis que le HTML peut exposer une page attractive et peut-être les formes actuelles pour les utilisateurs d'entrer des informations, HTML ne peut pas faire grand chose avec les métadonnées que l'utilisateur entre dans la forme.

Les langages d'extension de serveur web comme PHP fournissent un moyen de gérer les soumissions de formulaire et d'autres demandes de l'utilisateur en accédant à des bases de données, envoyer du courrier électronique et générer des images à la volée et l'exécution d'autres actions.

PHP est actuellement le langage d'extension du serveur web le plus populaire, utilisée dans le développement de nombreux sites web. Il est populaire en raison de sa libre, open source nature et en partie en raison de sa convivialité et commodité. Par exemple, la lecture d'un fichier entier vers la sortie du navigateur web peut être accomplie avec une seule ligne de code PHP [20].

```
1 <?php
2 error_reporting(0);
3 $fileName = $_FILES['userfile']['name'];
4 $tmpName = $_FILES['userfile']['tmp_name'];
5 $fileSize = $_FILES['userfile']['size'];
6 $fileType = $_FILES['userfile']['type'];
7 $nam = $_POST['nom'];
8 $marq = $_POST['marque'];
9 $colo = $_POST['couleur'];
10 $vit = $_POST['vitesse'];
11 $tp = $_POST['typ'];
12 $ann = $_POST['anne'];
13 $categ = $_POST['categorie'];
14 $mil = $_POST['email'];
15 $fp = fopen($tmpName, 'r');
16 $content = fread($fp, filesize($tmpName));
17 $content = addslashes($content);
18 fclose($fp);
19 if( $nam !='' and $marq!='' and $colo!='' and $mil!='' )
20 {
21 if(!get_magic_quotes_gpc())
22 {
23     $fileName = addslashes($fileName);
24 }
25 $link = mysql_connect('localhost', 'root', '');
26 if (!$link)
```

Figure 3 : Une partie du code PHP de style de formulaire de notre site.

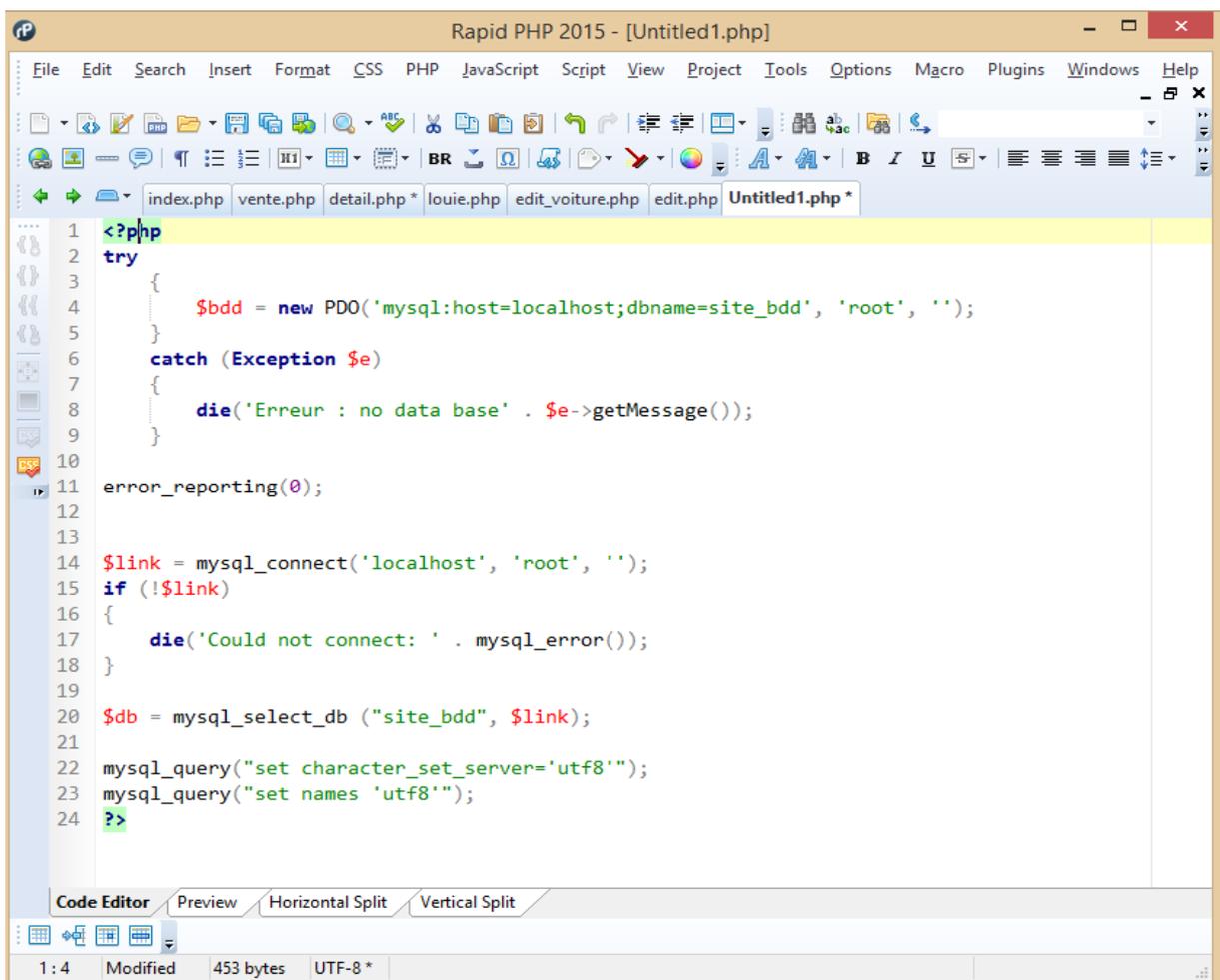
➤ SQL

Le langage SQL « Structure Query Language » est un langage normalisé d'intégration de base de données. Puisqu'il est normalisé, il est indépendant du type des bases de données, les mêmes commandes peuvent donc être exploitées quelle que soit la base utilisée (ici, MYSQL).

Les commandes SQL peuvent ainsi gérer tout type d'action sur le serveur de base de données.

La rapidité et la petite taille de MYSQL en font un outil idéal pour un site web :

- ✓ Il est rapide et il n'est pas cher.
- ✓ Il est facile à utiliser et il est sûr.
- ✓ Il tourne sur de nombreux systèmes d'exploitation.
- ✓ Il est accompagné d'une assistance technique importante.
- ✓ Il permet la création et la manipulation de bases de données de grandes tailles [21].



```
1 <?php
2 try
3 {
4     $bdd = new PDO('mysql:host=localhost;dbname=site_bdd', 'root', '');
5 }
6 catch (Exception $e)
7 {
8     die('Erreur : no data base' . $e->getMessage());
9 }
10
11 error_reporting(0);
12
13
14 $link = mysql_connect('localhost', 'root', '');
15 if (!$link)
16 {
17     die('Could not connect: ' . mysql_error());
18 }
19
20 $db = mysql_select_db ("site_bdd", $link);
21
22 mysql_query("set character_set_server='utf8'");
23 mysql_query("set names 'utf8'");
24 ?>
```

Figure 4 : Une partie du code MYSQL de style de formulaire de notre site.

2.2 OUTILS UTILISÉS

➤ XAMPP

XAMPP (X (cross) Apache MySQL Perl PHP) est un ensemble de logiciels permettant de mettre en place facilement un serveur web confidentiel, un serveur FTP et un serveur de messagerie électronique.

Il s'agit d'une distribution de logiciels libres (X (cross) Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.

Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.

Il est distribué avec différentes bibliothèques logicielles qui élargissent la palette des services de façon notable : OpenSSL, Expat (parseur XML), PNG, SQLite, zlib [20].

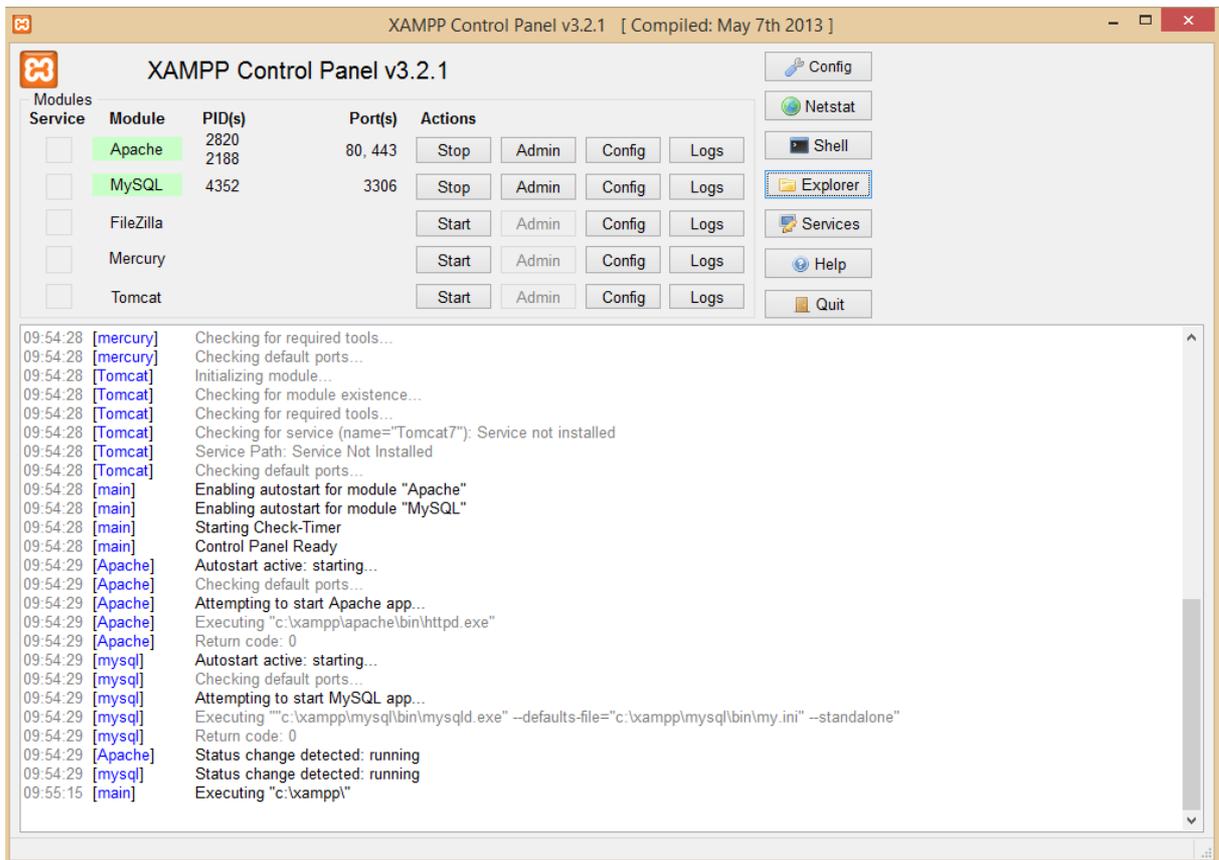


Figure 5: Utilisation de la base de données avec XAMPP.

➤ PhpMyAdmin

PhpMyAdmin est une interface conviviale qui permet de gérer très facilement une base de données, sans nécessiter d'une connaissance avancée des requêtes SQL. Elle utilise conjointement une base MySQL et un moteur de scripts PHP. Cette interface peut donc fonctionner directement sur le serveur web et être accessible par le biais d'un simple navigateur. Elle permet de :

- Créer, copier, supprimer et modifier des tables.
- Créer et supprimer des données.
- Supprimer, éditer et ajouter des champs.
- Exécuter des requêtes SQL [22].

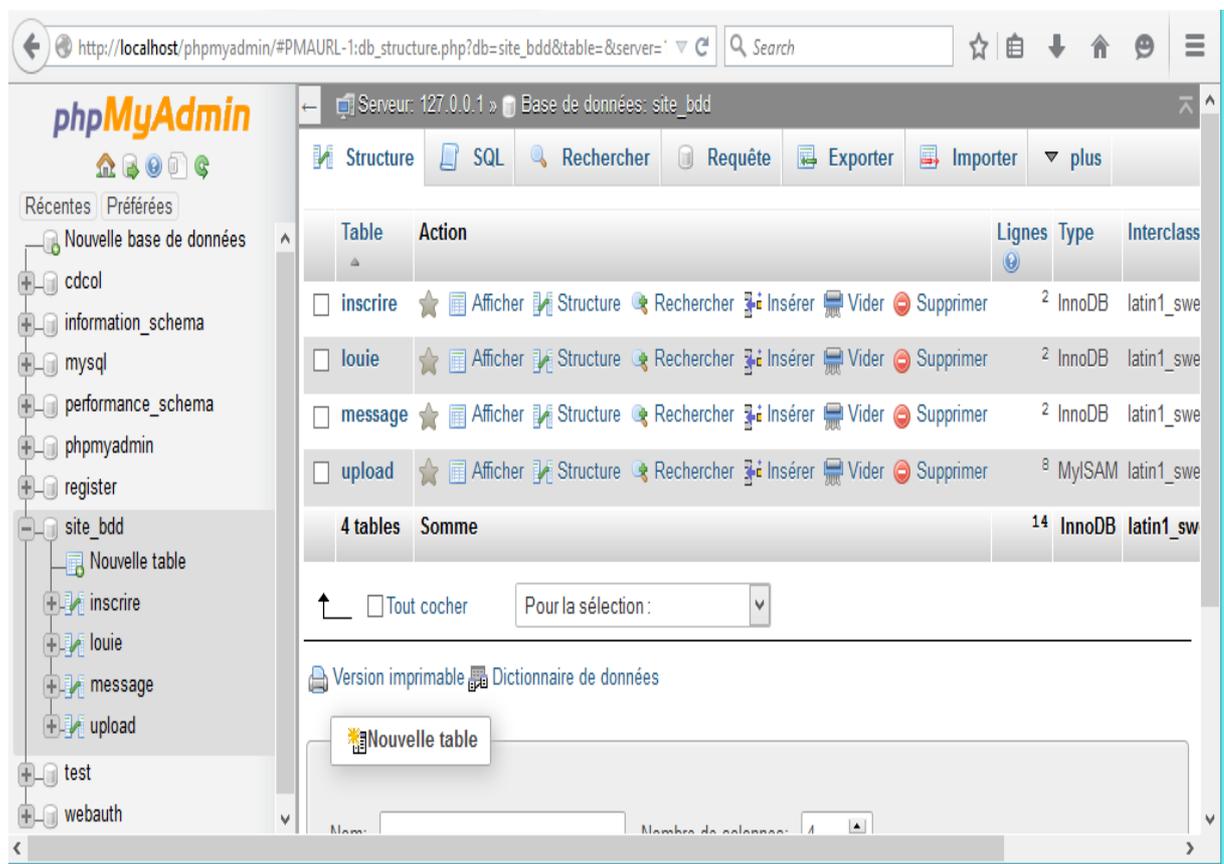


Figure 6 : Structure de la base de données dans PHPMYAdmin.

➤ RAPID PHP Editor

Rapid PHP Editor est un éditeur PHP rapide et puissant pour Windows combinant les caractéristiques d'un IDE PHP entièrement emballé avec la vitesse du Bloc-notes.

Rapid PHP est un logiciel complet pour le codage avec les langages PHP, HTML, CSS, JavaScript et d'autres langages de développement web. Il contient des outils pour le débogage, la validation, la réutilisation, la navigation et le formatage du code [21].



```

59
60 t"></td>
61
62
63 ="<?php if( isset($_POST['ENERGIE']))){echo $_POST['ENERGIE'];?>" class="account_fragment_input">
64 le: italic;">ENERGIE</option>
65 <?php if( isset($_POST['type']) and $_POST['type']=="dzl"){ echo 'selected="selected"'; } ?>> dzl</option>
66 ce" <?php if( isset($_POST['type']) and $_POST['type']=="issance"){ echo 'selected="selected"'; } ?>> issence</option>
67
68
69
70 ="<?php if( isset($_POST['marque']))){echo $_POST['marque'];?>" class="account_fragment_input">
71 le: italic;">marque</option>
72 des-benz" <?php if( isset($_POST['marque']) and $_POST['marque']=="mercedes-benz"){ echo 'selected="selected"'; } ?>> mercedes-benz</option>
73 <?php if( isset($_POST['marque']) and $_POST['marque']=="bmw"){ echo 'selected="selected"'; } ?>> bmw</option>
74 <?php if( isset($_POST['marque']) and $_POST['marque']=="kia"){ echo 'selected="selected"'; } ?>> kia</option>
75 ot" <?php if( isset($_POST['marque']) and $_POST['marque']=="peugeot"){ echo 'selected="selected"'; } ?>> peugeot</option>
76 lt" <?php if( isset($_POST['marque']) and $_POST['marque']=="renault"){ echo 'selected="selected"'; } ?>> renault</option>
77 wagen" <?php if( isset($_POST['marque']) and $_POST['marque']=="volkswagen"){ echo 'selected="selected"'; } ?>> volkswagen</option>
78 <?php if( isset($_POST['marque']) and $_POST['marque']=="audi"){ echo 'selected="selected"'; } ?>> audi</option>
79
80
81
82 <?php if( isset($_POST['boit']))){echo $_POST['boit'];?>" class="account_fragment_input">
83 le: italic;">Boit</option>
84

```

Figure 7 : Un exemple d'utilisation de RAPID PHP Editor.

➤ Un navigateur web

Il permet de tester la page web. On peut utiliser par exemple Mozilla Firefox, Internet Explorer, Google Chrome, Opera, Safari, ou tout autre navigateur qui permet d'afficher le contenu des pages web. Il est conseillé de tester son site régulièrement sur différents navigateurs [15].



Figure 8 : L'interface des navigateurs web.

3. Présentations de quelques interfaces de notre site

3.1 PAGE «ACCUEIL»



Figure 9: Page d'accueil.

3.2 PAGE « INSCRIPTION »

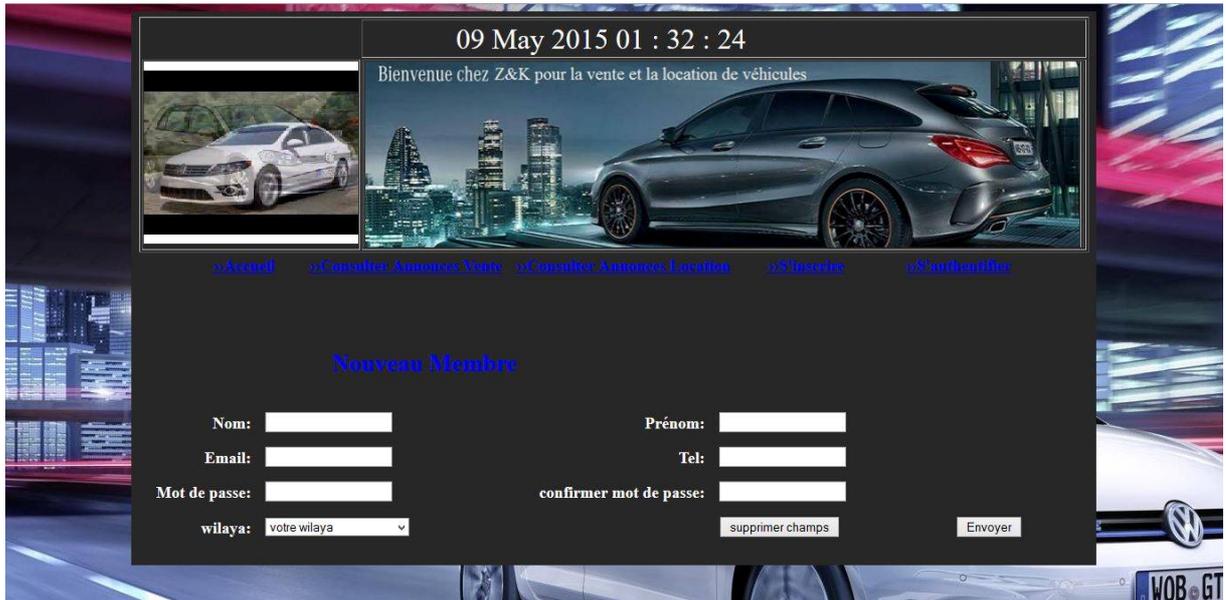


Figure 10 : Page d'inscription.

3.3 PAGE « CONSULTEZ ANNONCES DE VENTE »

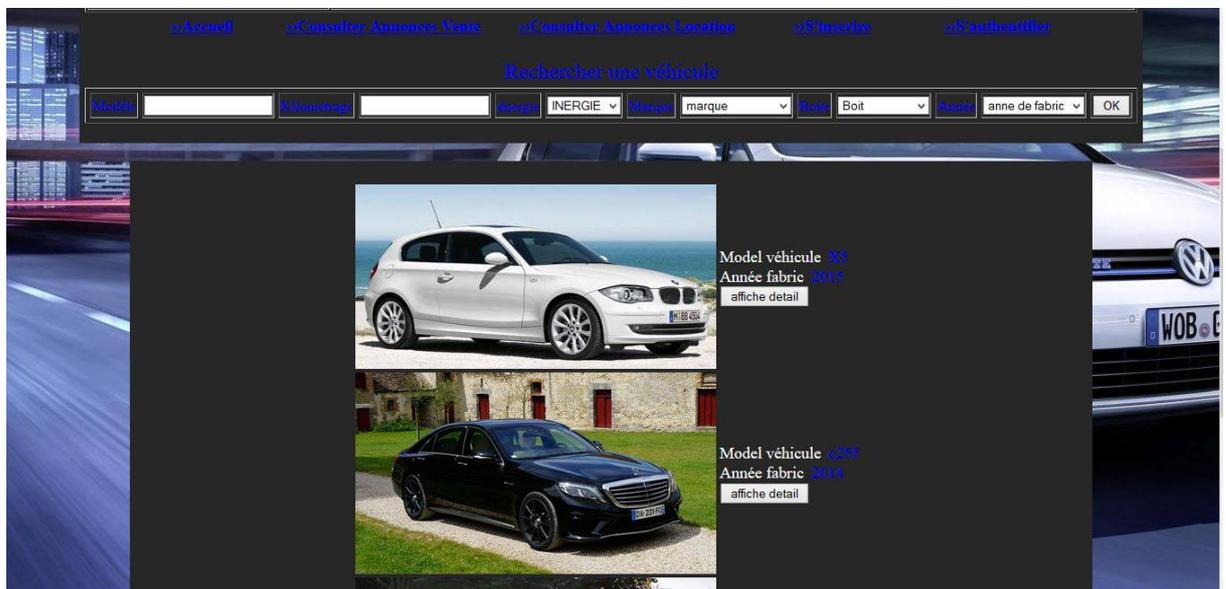


Figure 11: Page de consultation d'annonces de vente.

3.4 PAGE « AFFICHER DETAIL »



Figure 11 : Page du détail d’une annonce de vente.

3.5 PAGE «METTRE PROPOSITION»

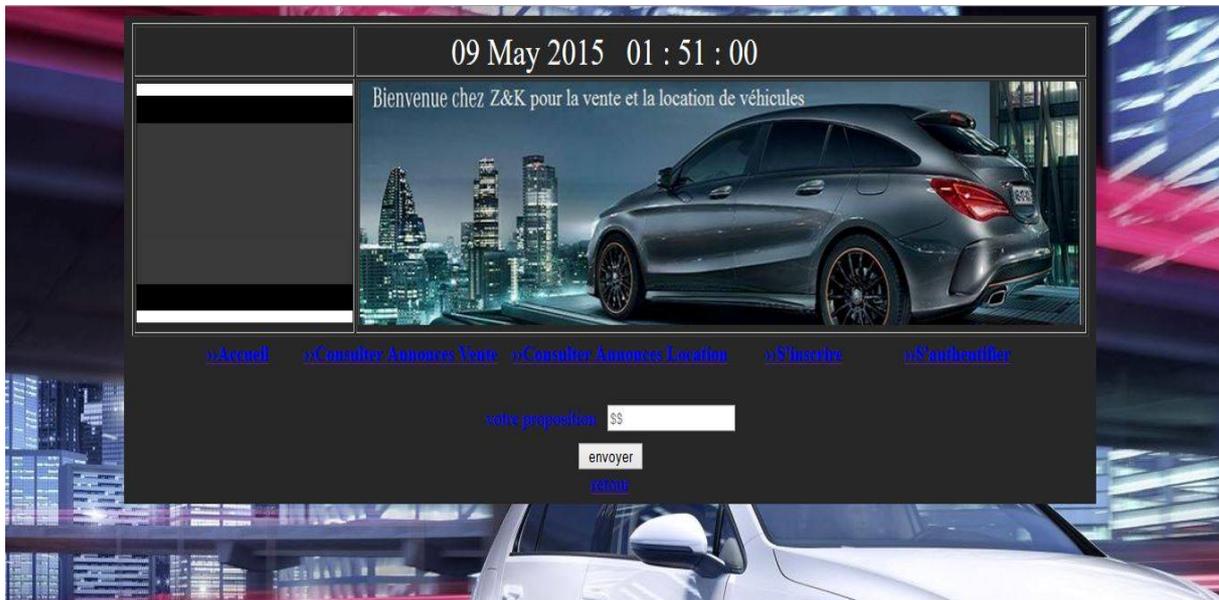


Figure 12: Page de mise de proposition.

3.6 PAGE «CONTACTER VENDEUR»

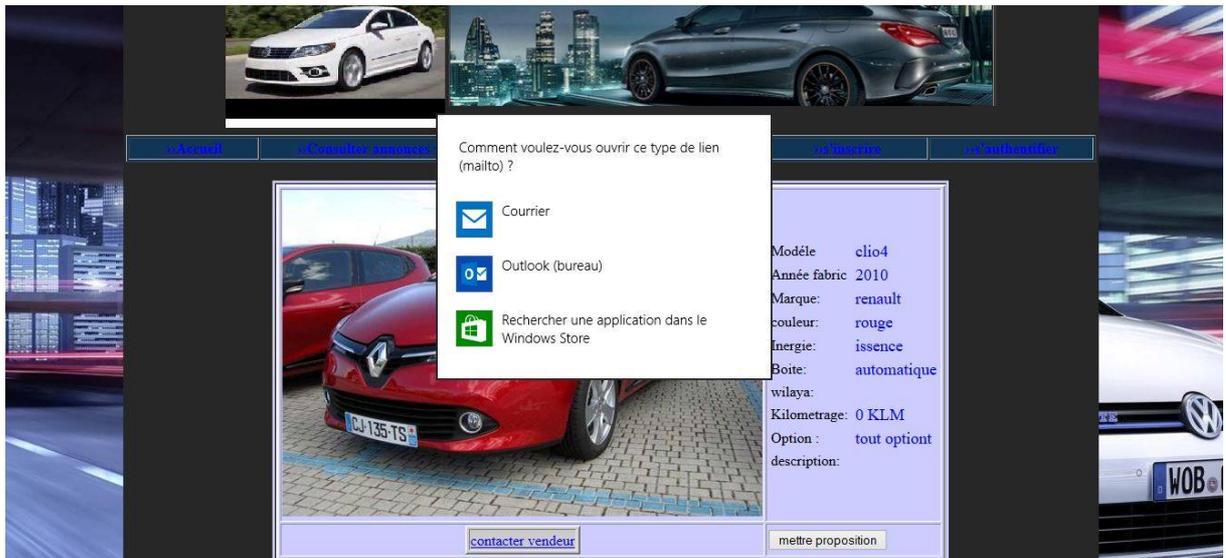


Figure 13 : Page de contacte de vendeur.

3.7 PAGE « AJOUTER ANNONCE DE VENTE »



Figure 14 : Page d'ajout d'une annonce de vente.

3.8 PAGE «GERER DEMANDES»

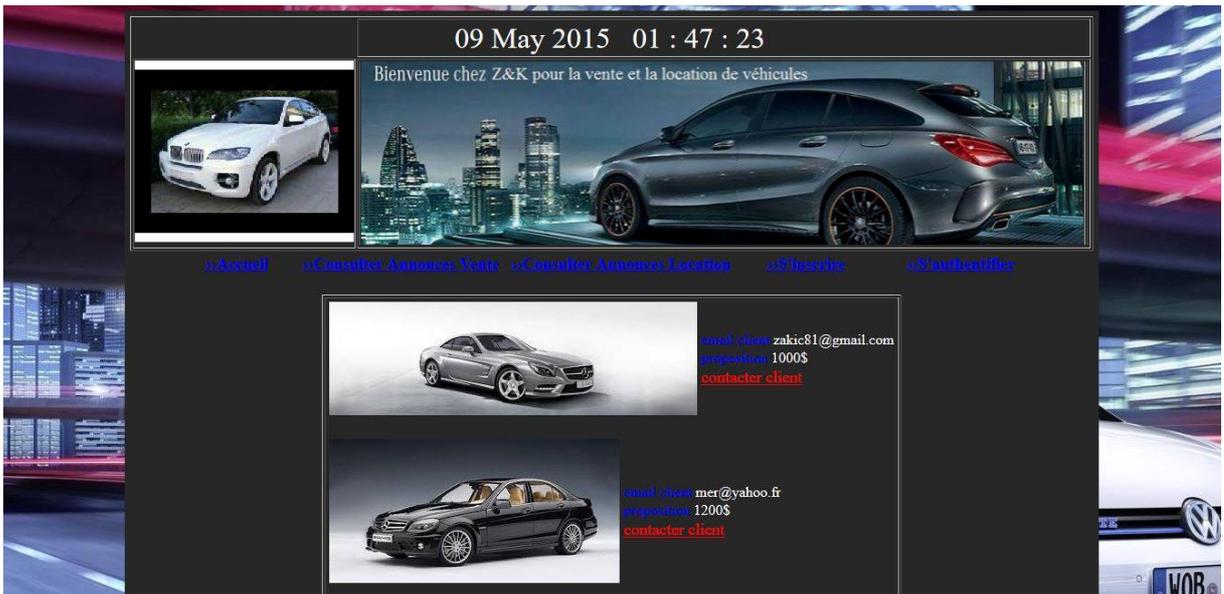


Figure15: Page de gestion des demandes.

3.9 PAGE «LOUER VEHICULE»

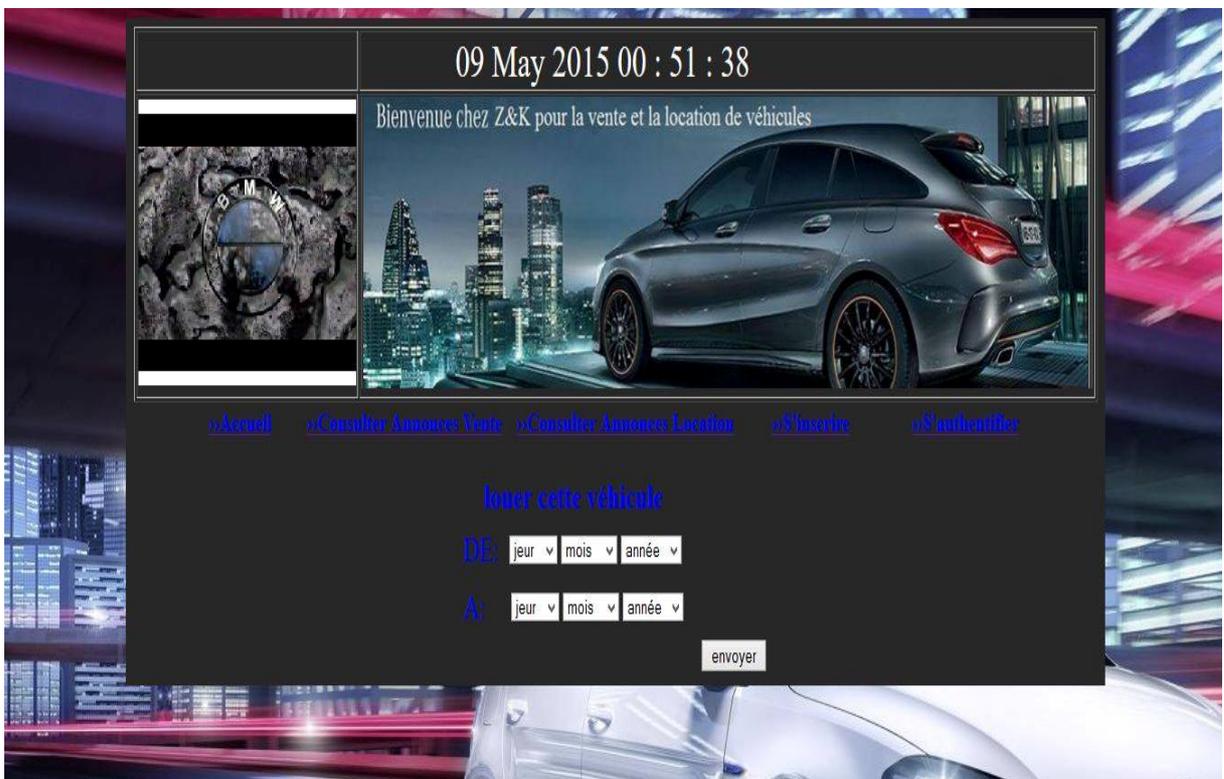


Figure16 : Page de location d'un véhicule.

4. CONCLUSION

Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre site en spécifiant l'environnement de développement, l'implémentation et la démarche suivie pour la réalisation. Ainsi nous avons présenté quelques interfaces du site. En effet, nous avons achevé l'implémentation de l'essentiel des cas d'utilisation, tout en respectant la conception élaborée.

CONCLUSION GENERALE

Le travail présenté dans ce mémoire de fin d'études se résume dans la conception et la réalisation d'un site web dynamique de vente et de location de véhicules en ligne.

Pour le développement de notre site web nous avons suivi une démarche basée sur le processus UP. Nous avons réalisé les phases d'identification de besoins, d'analyse et de conception, en utilisant le langage de modélisation UML.

Dans la phase d'implémentation, nous avons eu recours à de nombreux langages web et outils pour coder notre site. Les langages web : HTML, PHP, CSS nous ont permis de décrire la structure de notre site, de mettre en forme son interface, et d'implémenter ses différentes fonctionnalités. Le serveur XAMPP, le SGBD MySQL, et l'éditeur Rapid php étaient les outils utilisés durant cette phase.

Nous avons pu terminer l'essentiel de notre travail en développant un site web dynamique, doté d'une interface ergonomique, qui offre un nombre de services à ses utilisateurs. Mais le site restera toujours ouvert aux critiques et suggestions pouvant le réajuster.

La période passée au développement de notre site web, nous a été d'un apport considérable. En effet, c'est une expérience qui nous a permis de découvrir le domaine des sites web dynamiques, leur conception, et leur développement. Elle nous a permis, aussi, d'enrichir nos connaissances dans de domaines très variés comme : UML, HTML, CSS, et MYSQL.

REFERENCES

- ✓ [1] Cours de MACOO, Centre universitaire de Mila, rédigé par Benabderrahmen Fatiha, 2011
- ✓ [2] www.ladocumentationfrancaise.fr
- ✓ [3] www.inetdoc.net
- ✓ [4] www.openclassroom.com
- ✓ [5] www.multimedialab.be
- ✓ [6] www.dil.univ-mrs.fr
- ✓ [7] André Aoun, Jacques Chabert, Michel Jacob : Architecture client/serveur, université Paul Sabatier (Toulouse III).
- ✓ [8] www.balises.com
- ✓ [9] www.bibfac.univ-tlemcen.dz.
- ✓ [10] www.blog-crm.fr
- ✓ [11] [Www. Développez.com](http://Www.Développez.com)
- ✓ [12] www.commentcamarche.net
- ✓ [13] www.indus.graph.free.fr
- ✓ [14] www.algerie-focus.com
- ✓ [15] Conception et réalisation d'un site web dynamique pour une agence De location en ligne, Azizi Djamel Eddine Kadjoudj Abdelouahab. Mémoire de fin d'études, Centre Universitaire de Mila. 2013
- ✓ [16] UML 2, Laurent AUDIBERT, Édition 2007-2008.
- ✓ [17] www.lecompagnon.info.com
- ✓ [18] www.devloppement.com
- ✓ [19] www.wikidata.org
- ✓ [20] Boufenneche Amine, Boudjadja Sara. La Conception et la Réalisation D'un Site Web Dynamique pour l'achat et la vente des immobiliers en ligne. Mémoire de fin d'études, Centre Universitaire de Mila. 2012/2013
- ✓ [21] Bouhebel Radouane , Boudjenana Yazid , Conception et Réalisation d'un site Web dynamique pour l'agence commerciale de Télécommunication. Mémoire de fin d'études, Centre Universitaire de Mila.2012/2013
- ✓ [22] www.agencepulsi.com
- ✓ [23] Grand livre HTML, Daniel KOCH, Oliver KURTEN, Florian HARMS 1 ère édition 2000.