

UNIVERSITE DE DOUALA

---



---

## Mémoire

Présenté par **TOUKAM CHIEDJOU Eric Robin**

En vue d'obtenir le diplôme de  
MASTER II en IASIG

---

Informatique Appliquée aux Systèmes d'Informations Géographiques

---

# SYSTEME D'INFORMATION TOURISTIQUE DU CAMEROUN

---

**Sous la coordination de :**

L'encadreur Académique

Dr. TAGOUDJEU Joseph  
*Chargé de cours*

L'encadreur Professionnel

M. NYAMSI Christian  
*Directeur de Fire Software*

***Année Académique 2009 / 2010***

## **REMERCIEMENTS**

---

Je tiens à exprimer mes remerciements au Pr Jean Paul RUDANT pour l'intérêt accordé à cette formation.

Je remercie également Dr MVOGO Joseph qui nous a encouragés tout au long de notre formation.

Je souhaite manifester ma reconnaissance au Dr. TAGOUDJEU d'avoir accepté d'être mon encadreur, avec un suivi constant et un intérêt démontré tout au long de mon travail.

Je tiens à remercier Monsieur MOUTOME Marcelle Guillaume pour son soutien inébranlable et incontestable tout au long de notre formation.

Mes remerciements s'adressent également à tous les enseignants du Master 2 IASIG.

Je souhaite remercier l'ensemble des personnes qui m'ont soutenu pendant cette période. Je pense notamment à ma famille et à mes amis.

## SOMMAIRE

REMERCIEMENTS .....	2
SOMMAIRE .....	3
TABLE DES ILLUSTRATIONS .....	6
LISTE DES TABLEAUX .....	7
INTRODUCTION .....	8
I. CONTEXTE.....	8
II. PROBLEMATIQUE.....	8
III. OBJECTIFS .....	8
IV. RESULTATS OBTENUS.....	9
V. ENVIRONNEMENT DU STAGE.....	9
V.1. Historique de l'entreprise.....	9
V.2. Environnement et adresse complète de FIRE SOFTWARE.....	10
V.3. Les infrastructures.....	10
V.4. Personnels et secteurs d'activités.....	10
V.5. Partenariat.....	10
V.6. Organigramme.....	11
VI. ORGANISATION DU MEMOIRE.....	11
CHAPITRE I : ETAT DE L'ART .....	13
I. PREAMBULE.....	13
II. CONCEPT D'UN SIG .....	13
II.1. Définition et rôles .....	14
II.2. Fonctionnalités d'un SIG.....	16
II.2.1. Abstraction .....	16
II.2.2. Acquisition.....	16
II.2.3. Archivage.....	16
II.2.4. Analyse .....	17
II.2.5. Affichage .....	17
III. LE WEBMAPPING.....	17
III.1. Définitions.....	17

III.2. Principe du Webmapping.....	18
IV. LES SERVEURS CARTOGRAPHIQUES.....	19
IV.1. Définition .....	19
IV.2. Architecture des serveurs cartographiques.....	19
CHAPITRE II : CHOIX DES OUTILS A UTILISER .....	22
INTRODUCTION.....	22
I. ETUDE COMPARATIVE DES SERVEURS CARTOGRAPHIQUES .....	22
I.1. GéoServer .....	22
I.2. MapServer .....	23
II. ETUDE COMPARATIVE DES SERVEURS WEB.....	24
III. ETUDE COMPARATIVE DES SGBD SPATIAUX.....	25
IV. LES LANGAGES DE SCRIPT .....	28
V. AUTRES LOGICIELS UTILISES .....	29
VI. SYNTHESE DES DIFFERENTS OUTILS PRESENTES.....	29
CHAPITRE III : ANALYSE ET CONCEPTION.....	30
I. METHODES D'ANALYSE ET DE CONCEPTION .....	30
I.1. PREAMBULE.....	30
I.2. UML.....	31
I.2.1. Diagrammes structurels ou diagrammes statiques .....	32
I.2.2. Diagrammes comportementaux ou diagrammes dynamiques.....	33
I.2.3. Diagrammes d'interactions.....	33
II. CYCLE DE VIE DU LOGICIEL .....	34
III. EXPRESSION DES BESOINS ET FONCTIONNALITES .....	35
III.1.Exigences des besoins.....	35
III.2. Exigences fonctionnelles .....	36
III.3. Exigences non fonctionnelles.....	36
IV. ANALYSE ET CONCEPTION .....	37
IV.1. Les utilisateurs.....	37
IV.2. Les cas d'utilisation.....	37
IV.3. Diagrammes de séquence.....	41
IV.4. Diagrammes d'activités.....	43
IV.5. Diagramme de classes .....	45
IV.6. Diagrammes d'états-transition .....	46

CHAPITRE IV : IMPLEMENTATION ET TEST .....	48
I. ARCHITECTURE PHYSIQUE DU SYSTEME.....	48
I.1. Modèle de déploiement du système.....	48
I.2. Caractéristiques de la station de travail .....	48
II. REALISATION DU SIG.....	49
II.1. Calage des informations recueillies .....	49
II.2. Création de la base de données spatiale .....	49
II.3. Introduction des données dans la base de données.....	50
III. REALISATION DE LA CARTE.....	52
IV. RESULTATS.....	53
IV.1. Page d'accueil du SIG.....	53
IV.2. La page Administration .....	54
IV.3. Consultation et enregistrement de données .....	54
IV.4. La page Cartographie .....	55
CONCLUSION ET PERSPECTIVES .....	57
GLOSSAIRE.....	58
BIBLIOGRAPHIE.....	59

## TABLE DES ILLUSTRATIONS

---

Figure 1: Représentation du concept des S.I.G.....	15
Figure 2: Archivage des données.....	17
Figure 3: Architecture pour une solution « serveur cartographique » (attributaire et géographique gérés séparément) .....	19
Figure 4 : Architecture pour une solution « serveur cartographique » (SGBD avec cartouche spatiale) .....	20
Figure 5 : Diagramme de cas d'utilisation.....	39
Figure 6 : Diagramme de séquences d'enregistrement d'informations dans la base de données .....	42
Figure 7: Diagramme de séquence navigation sur la carte .....	43
Figure 8 : Diagramme d'activités authentification .....	44
Figure 9 : Diagramme d'activités création utilisateur.....	44
Figure 10 : Diagramme de classes.....	45
Figure 11 : Diagramme d'états-transitions utilisateur .....	46
Figure 12 : Diagramme d'états-transition carte .....	46
Figure 13 : Diagramme de déploiement.....	48
Figure 14 : Ajout d'une base de données.....	49
Figure 15 : Exemple de syntaxe de ogrinfo .....	52
Figure 16 : Connexion à la base de données d'une couche.....	52
Figure 17 : Page d'accueil du SIG.....	53
Figure 18 : Données attributaires d'une table.....	54
Figure 19 : Formulaire d'enregistrement des données d'une table.....	55
Figure 20 : Cartographie.....	56

## **LISTE DES TABLEAUX**

---

Tableau 1: Tableau comparatif des serveurs cartographiques .....	24
Tableau 2 : tableau comparatif des SGBD spatiaux.....	28
Tableau 3 : Synthèse des fonctionnalités .....	29
Tableau 4 : Caractéristiques de la station de travail.....	48
Tableau 5 : Options 1 de la commande shp2pgsql .....	51
Tableau 6 : Options 2 de la commande shp2pgsql .....	51

## **INTRODUCTION**

---

### **I. CONTEXTE**

Le développement de l'industrie touristique à travers la mise en œuvre des mesures et des actions spécifiques est l'une des préoccupations majeures du gouvernement camerounais<sup>1</sup>, notamment :

- La promotion du tourisme ;
- L'aménagement des sites touristiques et écotouristiques prioritaires ;
- L'amélioration du système de collecte des données statistiques.

### **II. PROBLEMATIQUE**

Au vue du contexte supra-décrit, la promotion du tourisme camerounais est confrontée à un certain nombre de difficultés, notamment :

- l'absence d'une base de données fiable relative au tourisme;
- l'insuffisance des moyens alloués à la promotion touristique ;
- les difficultés à obtenir des informations fiables à caractère touristique ;
- l'insuffisance quantitative et qualitative des supports touristiques.
- l'anarchie dans laquelle opèrent les acteurs du secteur du tourisme.

Toutefois, pour promouvoir le tourisme camerounais, il faut disposer d'un certain nombre d'informations fiables sur l'offre touristique du Cameroun (les sites touristiques et éco-touristiques, leur localisation, leur accessibilité, leurs ressources naturelles, les structures d'accueil et d'hébergement, les agences de voyages, les établissements de restauration, les établissements de loisirs, etc...).

### **III. OBJECTIFS**

Au vue des problèmes élaborés plus haut, la conception d'un système d'informations géographiques est nécessaire afin de permettre une vulgarisation du

---

<sup>1</sup> Document de Stratégie de Croissance et de l'Emploi (DSCE)



tourisme accentuée sur la géolocalisation ainsi qu'une meilleure gestion des informations liées au tourisme.

Ainsi, pour la mise sur pied de cette application, les objectifs sont les suivantes :

- Mettre en place la base de données géographique des données touristiques.
- Réaliser un noyau SIG dédié au tourisme devant évoluer progressivement.
- Diffuser les informations de ce SIG en ligne via une application web.
- Produire des requêtes, des opérateurs, des agrégations et des fonctions permettant la manipulation de ces données.

#### **IV. RESULTATS OBTENUS**

L'élaboration de ce projet n'étant pas du tout facile. Nous avons pu :

- mettre sur pied une base de données géographique des données géographiques ;
- créer un formulaire d'enregistrement qui permettra de mettre à jour constamment notre carte ;
- réaliser une carte touristique qui permettra à tout utilisateur de répertorier les informations sur chaque site sélectionné ;
- diffuser des informations via une application web de ce SIG.

#### **V. ENVIRONNEMENT DU STAGE**

##### **V.1. Historique de l'entreprise**

Crée par un collectif de deux Ingénieurs Informaticiens, FIRE SOFTWARE, bureau d'études spécialisé dans les prestations des services informatiques, prône la révolution informatique du 21<sup>ème</sup> siècle. C'est ainsi qu'armée d'une expérience indéniable de ses ingénieurs et consultants, elle met à la disposition de ses clients et partenaires des prestations informatiques très variées beaucoup plus spécialisées dans les domaines de la formation professionnelle, maintenance informatique, du développement des logiciels, du développement des Systèmes d'Informations Géographiques (SIG), conception/déploiement des solutions informatiques aux PME et PMI et surtout de la formation nécessaire pour le développement d'un pays.

## **V.2. Environnement et adresse complète de FIRE SOFTWARE**

Le siège social de l'entreprise est situé à Yaoundé sis au quartier Nsimeyong, plus précisément entre la Chapelle Nsimeyong et le carrefour Tamtam ; plus précisément à 100 m de la chapelle Nsimeyong dont son site web est : [www.firesoftwareonline.com](http://www.firesoftwareonline.com)







## **V.3. Les infrastructures**

L'entreprise FIRE SOFTWARE possède un cadre et des équipements qui sont à la pointe de la technologie informatique. Elle a en son sein :

- ✓ Une salle de formation climatisée équipée d'un vidéo projecteur et d'une dizaine d'ordinateurs hauts de gamme.
- ✓ Une salle de numérisation climatisée et équipée d'ordinateurs.
- ✓ Une salle de développement.




## **V.4. Personnels et secteurs d'activités**

L'entreprise FIRE SOFTWARE compte en son sein environs 08 personnels qualifiés et elle met à la disposition de ses Clients et Partenaires des prestations diversifiées dans le domaine de l'Informatique. Ainsi, ses prestations touchent pour l'essentiel les pôles suivants :

-  Développement des SIG.
-  Développement d'application pour PME/PMI.
-  Développement des sites web.
-  Maintenance informatique.
-  Mise en place des réseaux d'entreprise.
-  L'audit des systèmes d'information.

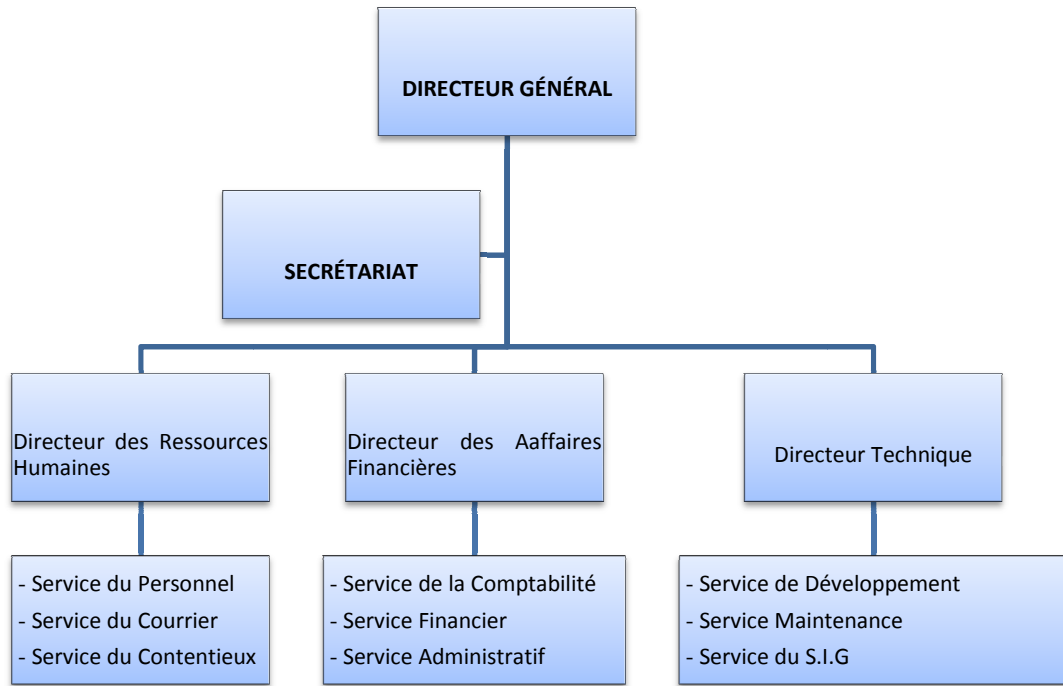
## **V.5. Partenariat**

Au regard de la qualité et l'efficacité du personnel ainsi que le travail effectué au sein de la société, une signature de partenariats a été établie entre les ONG, les entreprises publiques et l'entreprise FIRE SOFTWARE. Parmi ces partenaires nous pouvons citer :

-  Bénéficial Life, Assureur spécialisé dans les assurances vies.
-  Intek.
-  ISTAG, Institut Supérieur de Technologies Appliquées et de Gestion.

## V.6. Organigramme

Voici les différents termes utilisés dans l'organigramme de l'entreprise.



## VI. ORGANISATION DU MEMOIRE

Ce mémoire est composé de cinq parties présentant :

- Un état de l'art sur le domaine de la cartographie ;
- Le choix des outils utilisés dans le cadre du projet ;
- L'analyse et la conception de l'application à implémenter ;
- L'implémentation et le test du dit projet ;
- La conclusion et les perspectives.

Plus précisément, suite à cette introduction, le premier chapitre présente la cartographie interactive et les systèmes d'information géographique. Nous décrivons comment sont récupérées, stockées, traitées et représentées les données dans un logiciel de cartographie.

Le deuxième chapitre présente quelques outils utilisés qui interviennent dans le développement du projet.

La méthode d'analyse et de conception adoptée est présentée à travers le troisième chapitre.

L'implémentation et le test intervient au quatrième chapitre. Ce chapitre parle de la réalisation du SIG et des résultats estompés.

La conclusion porte sur le bilan du projet réalisé et les perspectives sur les modules à améliorer pour l'ergonomie totale de l'application.

## **CHAPITRE I : ETAT DE L'ART**

---

*L'état de l'art de ce mémoire porte en général sur la cartographie à travers lequel nous parlerons des Systèmes d'Information Géographique et des serveurs cartographiques.*

### **I. PREAMBULE**

Depuis 1966, on se réfère à la définition ratifiée par la commission de terminologie de l'ACI (Association Cartographique Internationale) qui définit la cartographie comme l'« ensemble des études et des opérations scientifiques, artistiques et techniques intervenant à partir des résultats d'observations directes ou de l'exploitation d'une documentation, en vue de l'élaboration de cartes et autres modes d'expression, ainsi que dans leur utilisation ».

Donc on pourra définir la cartographie comme la science de l'étude et de la réalisation des cartes.

Une carte est un média qui présente efficacement les données géographiques mais de manière statique et fixe. En fait, elle met en valeur un espace géographique et la localisation des éléments (villes, routes, hôtels) dans cet espace.

On peut aussi définir une carte comme une représentation conventionnelle, généralement plane, en positions relatives, de phénomènes concrets ou abstraits localisables dans l'espace.

En fait, le principe majeur de la cartographie est la représentation de données sur un support réduit représentant un espace généralement réel. L'objectif de la carte, c'est une représentation concise et efficace, la simplification de phénomènes complexes (politiques, économiques, sociaux, etc.) à l'œuvre sur l'espace représenté afin de permettre au public une compréhension rapide et pertinente.

### **II. CONCEPT D'UN SIG**

Traditionnellement, les données géographiques sont présentées sur des cartes utilisant des symboles, des lignes et des couleurs. La plupart des cartes ont une légende dans laquelle les éléments géométriques sont listés et expliqués (exemple : une ligne

noire épaisse pour des routes principales, une ligne noire mince pour les autres routes, etc.).

En fait, c'est à la fin des années 60 que le phénomène de SIG est apparu. Il résulte des besoins éprouvés par les géographes dans leur utilisation de l'outil informatique. A l'époque, il s'agissait d'exploiter les techniques récentes de photogrammétrie, de cartographie. L'objectif était la gestion et l'exploitation des données provenant de diverses sources :

- photogrammétrie numérique
- numérisation
- images scanner
- images radar

Durant les années 80, l'interactivité des interfaces graphiques liées au développement des outils cartographiques a été améliorée.

A la fin du XXème siècle, l'intégration des données est possible grâce à la télédétection (spatiale et aérienne). Ce qui a accrue la diffusion des systèmes d'information géographique.

De nos jours, avec l'essor des technologies liées à Internet, tout internaute peut créer et visualiser des cartes directement sur le Web.

Bref, avec l'informatique, la cartographie a apporté un nouveau mode de représentation. Ce qui a permis la mise sur pied de nouvelles applications et fait qu'en sorte, la géolocalisation est devenue un service incontournable aujourd'hui. Ainsi nous pouvons dire que les outils de cartographie reposent généralement tous sur un Système d'Information Géographique.

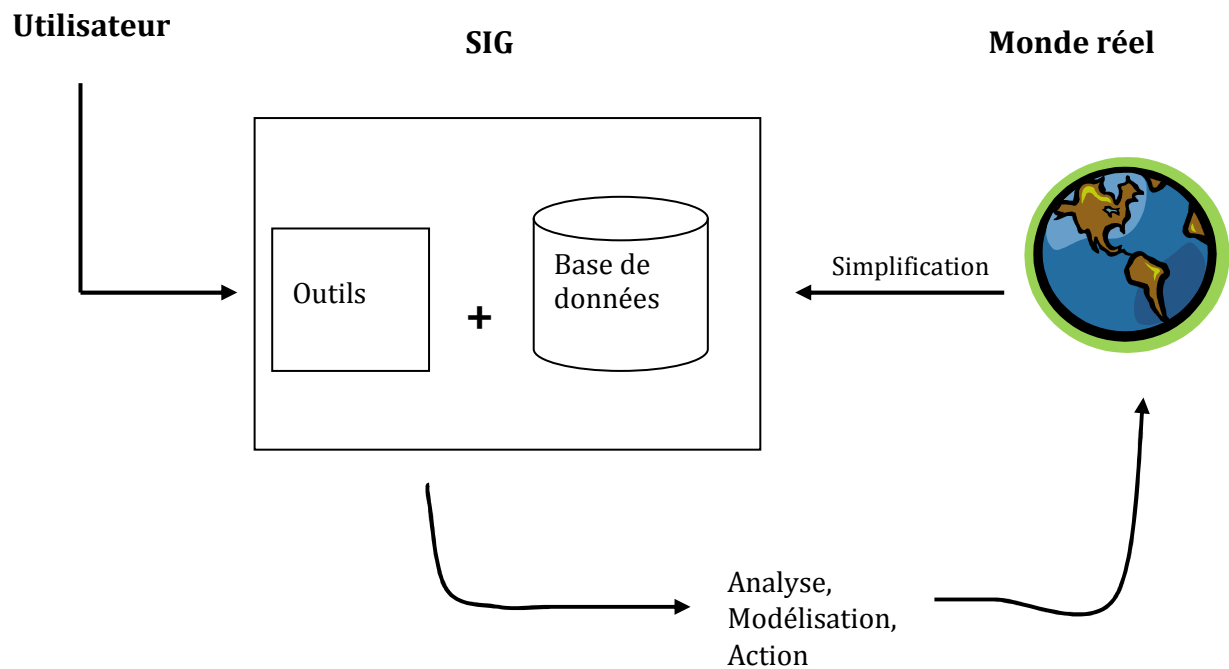
## **II.1. Définition et rôles**

Par rapport aux besoins des utilisateurs, on peut définir Système d'Information Géographique (SIG) comme un système de gestion de base de données pour la saisie, le stockage, l'extraction, l'interrogation, l'analyse et l'affichage des données localisées.

Une autre définition orientée plus vers les besoins des multiples décideurs, décrit un SIG comme étant un ensemble de données repérées dans l'espace, structurées, de façon à pouvoir en extraire des synthèses utiles à la décision. La finalité de cette seconde définition correspond bien à celle sous-jacente au terme Système d'Information.

Donc, un SIG est défini comme un système d'information capable d'organiser et de présenter des données alphanumériques spatialement référencées, ainsi que de produire des plans et des cartes.

Ainsi, un SIG combine la puissance d'une base de données (les requêtes, les analyses statistiques) et la représentation de l'information sur des cartes. C'est le moyen, à travers une standardisation de l'information géographique, non seulement de gérer un problème précis, mais aussi d'envisager les interdépendances.



**Figure 1: Représentation du concept des S.I.G.**

Les SIG gèrent les relations entre des objets spatialisés. Ces relations entre entités géographiques fournissent souvent des informations vitales. Par exemple les informations sur les réseaux d'eau potable et les bouches d'incendie seront capitales pour les pompiers avant et pendant les incendies.

Un SIG peut traiter des données géo-référencées et fournir des réponses aux questions relatives :

- à l'emplacement d'un objet donné,
- au développement spatial de certains phénomènes,
- aux changements survenus depuis une période donnée,
- aux impacts d'un évènement spécifique dans une région, etc.

## **II.2. Fonctionnalités d'un SIG**

Un SIG est basé principalement sur cinq fonctionnalités :

### **II.2.1. Abstraction**

Les SIG sont utilisés pour réaliser des descriptions du territoire permettant d'obtenir l'information nécessaire pour répondre à une problématique. Ils contiennent cette information sous plusieurs formes dont certaines sont des représentations d'éléments ou de phénomènes existants.

Bref, l'abstraction vise à représenter le monde réel qui est ainsi modélisé en fonction des besoins, ce qui permet de définir précisément le contenu du système c'est-à-dire organiser les données de manière à établir des relations entre les objets à travers des composants géométriques.

### **II.2.2. Acquisition**

L'acquisition des données s'effectue à travers la collecte des données géographiques. Cette collecte s'effectue par des moyens différents :

- Images satellites
- Images à base de la photographie aérienne
- Numérisation, etc...

Généralement, l'utilisation des images issues de la photographie aérienne nécessitent d'abord un traitement. Ce traitement permet la correction des aberrations optiques. Ensuite, égalisées radiométriquement, les images obtenues sont appelées orthophotographies ou orthoimages : c'est le processus d'équilibre chromatique.

En fait, c'est la distorsion causant la déformation de l'image ou à l'inclinaison de l'appareil photographique lors de la prise de vue qui est cause des aberrations optiques.

### **II.2.3. Archivage**

Le SIG sert à stocker les données et à les mettre à la disposition des utilisateurs du système

Bref, l'archivage des données revient à gérer la gestion des données tout en transférant vers un espace d'archivage les données récoltées.

On dénombre plusieurs objectifs issus de l'archivage :

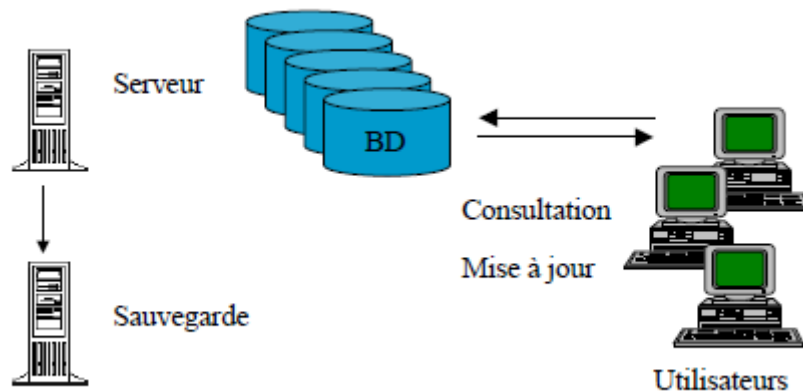
- Centraliser les données de manière normée ;



- Relier ces données les unes aux autres ;
- Permettre leur diffusion.

Ici, les avantages de l'archivage sont :

- d'assurer la cohérence et l'intégrité des données,
- de partager les bases,
- de simplifier leur mise à jour.



**FIGURE 2: Archivage des données**

#### **II.2.4. Analyse**

Ici, les données géographiques sont manipulées et interrogées afin de répondre aux requêtes des utilisateurs.

#### **II.2.5. Affichage**

Les règles de la cartographie doivent être respectées à travers une représentation graphique, ce qui permettra à l'utilisateur de mieux appréhender des phénomènes spatiaux.

Ainsi, après traitement des données, les SIG sont utilisés pour restituer les données sous différentes formes telles que les cartes, les graphiques, les tables statistiques ou tout autre fichier informatique exportable vers d'autres applications.

### **III. LE WEBMAPPING**

#### **III.1. Définitions**

Le Webmapping est le procédé permettant de concevoir, implémenter, générer et distribuer des cartes via le web.

Il correspond également à la manipulation des données géographiques via le web.

Le phénomène de distribution de cartes via le réseau Internet prend une réelle ampleur aujourd'hui parce qu'il répond aux besoins de diffusion de cartes en temps réel et à celui de contrôle et mise à jour des données sur un ordinateur distant.

Les données stockées et mises en relation dans les Systèmes de Gestion de Base de Données Relationnels (SGBDR) correspondent aux informations attributaires décrivant l'espace donné, tandis que les objets géographiques tels que le point, la ligne et le polygone sont des données géométriques référencées dans un plan en x par la longitude et en y par la latitude voire en z pour l'altitude.

Dans une conception en ligne, la visualisation des cartes passe par des programmes installés sur des serveurs cartographiques qui communiquent par des protocoles prédéfinis. La géométrie est gérée grâce à la cartouche spatiale du SGBDR.

L'implémentation du Webmapping fait face aux restrictions techniques telles que la limitation de la bande passante et la faible résolution d'affichage.

### **III.2. Principe du Webmapping**

Le Webmapping est basé sur une architecture client-serveur. Il utilise la pile de protocoles TCP/IP, qui permet à des ordinateurs branchés en réseau d'échanger de l'information via un navigateur web ou de transférer des fichiers via le protocole FTP.

L'architecture est de type client/serveur c'est-à-dire qu'il existe une série d'ordinateurs dits clients connectés à un serveur dédié qui lui-même communique vers l'extérieur (World Wide Web) ou avec des serveurs particuliers d'un réseau dédié par l'intermédiaire de leur adresse IP.

La solution la plus répandue actuellement dans le domaine de la mise en ligne de données cartographiques consiste à créer à la volée une image correspondant à la demande de l'utilisateur. Pour cela, il est le plus souvent fait appel à un serveur cartographique.

Un utilisateur émet à partir d'une machine cliente des requêtes spécifiques. Ces requêtes sont acheminées via le support de communication vers le serveur cartographique. Le serveur cartographique interprète cette requête et renvoie la carte sous la forme d'une image matricielle (gif, jpg...) ou vectorielle (svg...).

Les données peuvent être gérées par des logiciels spécifiques, les SGBDR, tels PostgreSQL, MySQL, Oracle, entre autres, qui peuvent être installés directement sur le

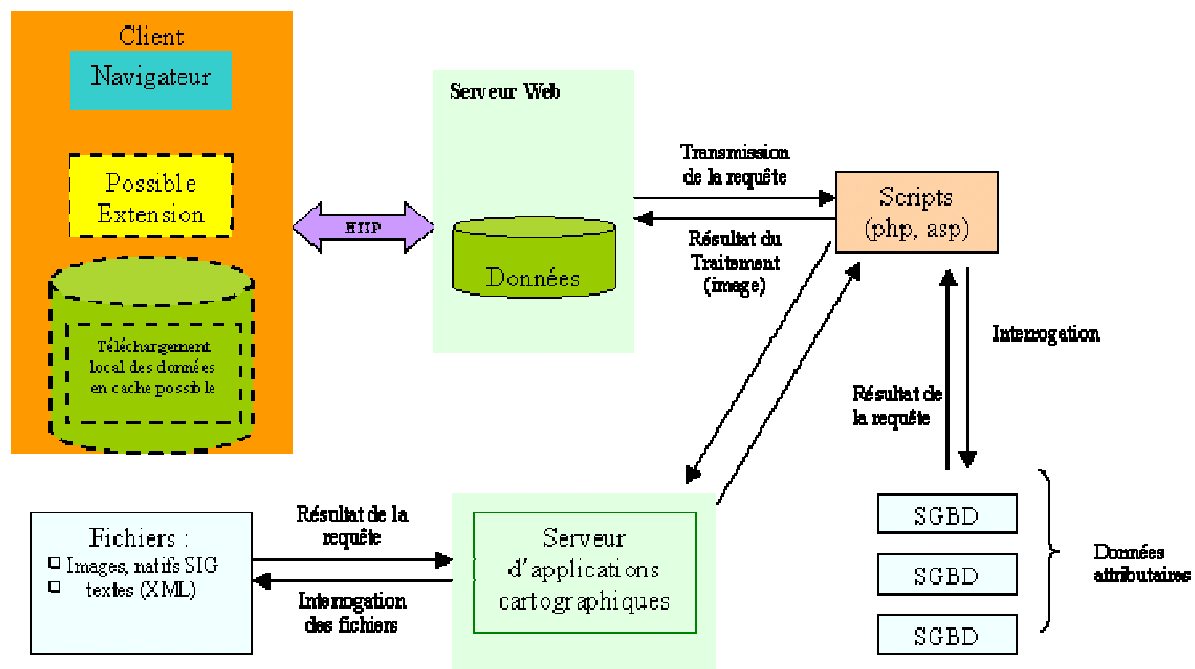
serveur contenant le serveur cartographique ou sur un autre serveur, distant. Qu'importe le lieu, l'objectif étant de pouvoir consulter et éditer des données à distance.

## IV. LES SERVEURS CARTOGRAPHIQUES

### IV.1. Définition

Un *serveur cartographique* est un serveur doté des fonctionnalités d'analyser, d'interpréter des requêtes sur des données géographiques émises par des machines clientes et de générer de cartes correspondant aux dites requêtes.

### IV.2. Architecture des serveurs cartographiques



**Figure 3: Architecture pour une solution « serveur cartographique » (attributaire et géographique gérés séparément)**

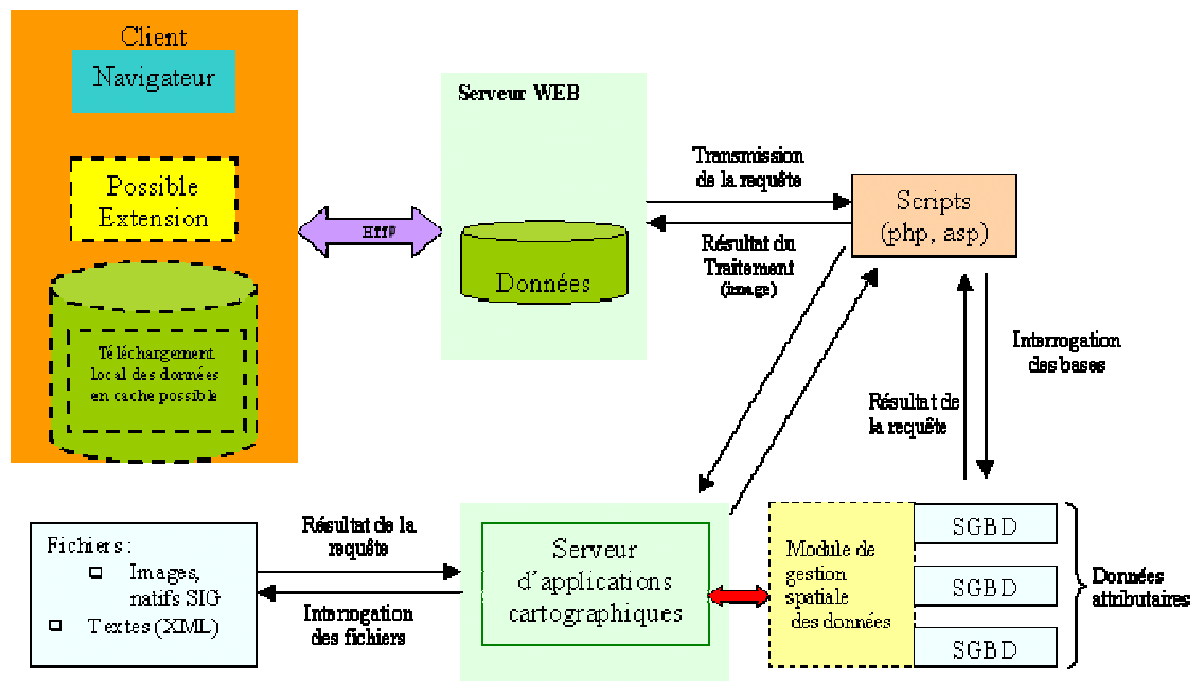
Source : [7]

Du côté serveur, on retrouve un serveur HTTP classique (logiciels tels Apache ou IIS (Internet Information Services) qui tourne en tâche de fond et permet aux serveurs de cartes d'accéder à l'Intranet et à l'Internet), un serveur d'applications cartographiques et des scripts écrits en langage de programmation serveur (ASP, JSP, PHP, servlets, ActiveX) permettant la transmission des requêtes aux éléments architecturaux concernés. Plusieurs bases de données attributaires peuvent être pontées et le serveur cartographique interroge des fichiers images, vectoriels

(structurés suivant la norme XML par exemple) ou natifs de SIG classique. Cet ensemble répond à toutes les requêtes de tous les clients. L'ordinateur serveur peut chercher cette information soit dans ses propres ressources, soit sur des serveurs de données distants. De plus, par définition même du serveur d'application, les sessions utilisateurs sont gérées : le contexte peut ainsi être connu et conservé.

Du côté client, une visionneuse suffit pour afficher la carte sur le browser - navigateur en français- un explorateur standard pour applications (Internet Explorer, Netscape).

Une partie des données comme les fonds cartographiques peut être chargée en local afin d'éviter des allers-retours trop gourmands entre client et serveur. Ici on parle de servlet (la visionneuse se trouve côté serveur).



**Figure 4 : Architecture pour une solution « serveur cartographique » (SGBD avec cartouche spatiale)**

Source : [7]

Dans ce type d'application, le serveur cartographique joue finalement un rôle de moteur de recherche entre les différentes bases de données permettant une restitution graphique.

### **III.3. Fonctionnalités des serveurs cartographiques**

Les serveurs cartographiques permettent au minimum les fonctionnalités suivantes :

- Affichage d'un jeu de données géo-référencées (avec gestion de différents systèmes de projection) à différents formats ;
- Sélection des données à afficher et/ou masquer ;
- Navigation au sein du jeu de données : zoom, déplacement, vue globale ;
- Interrogation des attributs correspondant à une sélection ;
- Localisation par requête attributaire (se positionner sur une commune) ;
- Fiche de renseignements par requête attributaire ;
- Mise à jour à distance de la base de données;
- Gestion de l'administration des données pour les modifications à distance ;
- Ecriture de remarques textuelles géo référencées conservées par le serveur et donc partagées ;
- Création de tampon, autres outils d'analyses spatiales ;
- Téléchargement des données visualisées;
- Extraction de données;
- Métadonnées disponibles;

## CHAPITRE II : CHOIX DES OUTILS A UTILISER

---

*De nombreux outils permettent aujourd'hui de visualiser des cartes mais également, pour certains, d'en créer. Dans ce chapitre, nous présenterons les différents outils disponibles en nous intéressant aussi bien à leurs fonctionnalités qu'à une étude comparative de ces outils et nous opterons sur le choix des outils pour mener à bien l'implémentation.*

### INTRODUCTION

Depuis quelques années déjà, l'utilisation et le déploiement des SIG ont beaucoup évolué, et cela en partie grâce à l'essor d'internet, par son augmentation des débits d'une part et par l'évolution des technologies employées d'autre part.

Sont alors apparus sur le marché des outils SIG adaptés Internet et permettant la diffusion de l'information géographique, l'analyse en ligne des données et les mises à jour. Cet essor des SIG doit beaucoup à nombre de logiciels libres qui ont su respecter les normes édictée par l'Open GIS Consortium<sup>2</sup> concernant l'interopérabilité des serveurs cartographiques par l'utilisation de services web.

### I. ETUDE COMPARATIVE DES SERVEURS CARTOGRAPHIQUES

#### I.1. GéoServer

GéoServer est un moteur cartographique Open Source développé en environnement JAVA, et basé sur la librairie GeoTools. GéoServer implémente de nombreuses spécifications Web Service de l'Open GIS Consortium (OGC). GéoServer permet rapidement de mettre en place un Web Service proposant les données vectorielles et/ou rasters.

GéoServer gère de nombreux formats de sortie pour les données (PNG, SVG, KML, JPEG, PDF...). De plus il est aisé pour un utilisateur averti de rajouter son propre module de sortie personnalisé pour des données vectorielles. C'est un outil de représentation cartographique lourd en ce sens où il nécessite l'installation d'un JDK 1.4 ou supérieur. Il

---

<sup>2</sup> OGC : Organisation internationale en charge de l'élaboration des standards et normes pour une information géographique « ouverte ».

dispose d'une vitesse d'affichage correcte mais d'une finesse d'image supérieure. Son interface est soignée et conviviale.

## I.2. MapServer

MapServer<sup>3</sup> est un moteur cartographique Open Source permettant d'afficher les données spatiales sur le web (cartes, images, et données vectorielles) et de construire des applications internet à référence spatiale. En plus de pouvoir parcourir des données géographiques, MapServer permet de créer des « cartes interactives » c'est-à-dire des cartes pouvant guider l'utilisateur vers un contenu spécifique.

Pour afficher des informations sur le navigateur Internet, MapServer utilise un fichier de configuration appelé Mapfile. Le Mapfile est le moteur de MapServer. Il se crée, se modifie avec un éditeur de texte.

Au cœur de MapServer se trouve une application CGI (Common Gateway Interface) pour la présentation sur le World Wide Web (WWW) de contenus dynamiques SIG ou résultant du traitement d'images. Le CGI est un outil puissant de génération des cartes mais il n'est pas possible de paramétrer l'interface de sortie de l'application.

MapServer intègre également un certain nombre d'applications autonomes pour la construction hors ligne de cartes, d'échelles et de légendes. Le module MapScript permet de l'interfacer à des pages html grâce aux langages PHP, Python, Perl, Java, C#.

Serveurs Cartographiques	GéoServer	Mapserver
<b>Avantages</b>	<ul style="list-style-type: none"><li>-GéoServer, solution libre</li><li>- Structure homogène : GeoAPI, GeoTools, respect des normes OGC</li><li>- Finesse de l'interface</li><li>-Haute résolution d'affichage</li><li>- Licence GPL</li></ul>	<ul style="list-style-type: none"><li>- MapServer, solution libre</li><li>- Les performances et la stabilité de MapServer</li><li>- L'adaptabilité et la flexibilité</li><li>- Compatibilité avec plusieurs langages de script</li><li>- Gère de nombreux formats vecteurs (.shp) et</li></ul>

<sup>3</sup> <http://mapserver.gis.umn.edu/index.html>

		raster (TIFF et GeoTIFF)  - Evolution rapide  -Connexion à des bases de données spatiales (PostGIS, Oracle)  -Grand nombre d'applications Open Source compatibles.
<b>Inconvénients</b>	- Lenteur d'affichage  - Compatible uniquement JDK 1.4 ou supérieur  - Documentation peu fournie	- Forte nécessité en développements  - Qualité du rendu graphique  - Lourdeur de l'installation

**Tableau 1: Tableau comparatif des serveurs cartographiques**

A partir de ce tableau comparatif, nous avons opté pour le choix vers MapServer qui permet une évolution rapide du noyau SIG et qui peut s'interfacer avec plusieurs langages de script.

## II. ETUDE COMPARATIVE DES SERVEURS WEB

Un serveur HTTP ou démon HTTP ou HTTPd (HTTP daemon) est un logiciel servant des requêtes respectant le protocole de communication client-serveur HTTP (HyperText Transfer Protocol), qui a été développé pour le World Wide Web.

Un ordinateur sur lequel fonctionne un serveur HTTP est appelé serveur Web. Le terme « serveur Web » peut aussi désigner le serveur HTTP (le logiciel) lui-même. Les deux termes sont utilisés pour le logiciel car le protocole HTTP a été développé pour le Web et les pages Web sont en pratique toujours servies avec ce protocole. D'autres ressources du Web comme les fichiers à télécharger ou les flux audio ou vidéo sont en revanche fréquemment servies avec d'autres protocoles.

Les serveurs HTTP les plus utilisés sont :

- Apache HTTP Server de Apache Software Foundation, successeur du NCSA<sup>4</sup> httpd

<sup>4</sup> Naational Center for Supercomputing Applications popularise le web.



- Internet Information Server de Microsoft (IIS)
- Sun ONE de Sun Microsystems (anciennement iPlanet de Netscape Communications Corporation)
- Le serveur Web Zeus de Zeus Technology

Le plus populaire est Apache HTTP Server qui sert environ 69% des sites Web en octobre 2005 selon Netcraft.

Serveur web	Apache	Microsoft IIS	Zeus
<b>Avantages</b>	<ul style="list-style-type: none"><li>- Gratuité</li><li>- Peu gourmand en ressources matérielles</li><li>- Excellente stabilité</li><li>- Multiplateformes</li></ul>	<ul style="list-style-type: none"><li>- Grande simplicité d'installation et d'utilisation</li><li>- Compatible uniquement Windows Server</li></ul>	<ul style="list-style-type: none"><li>- Peu gourmand en ressources matérielles</li><li>- Excellente stabilité</li><li>- Capacité à tenir de très gros pics de fréquentation</li><li>- Interface native avec certaines bases de données Multiplateformes</li></ul>
<b>Inconvénients</b>	<ul style="list-style-type: none"><li>- Installation et administration plus laborieuses (pas d'interface graphique)</li><li>- Pas de support technique</li></ul>	<ul style="list-style-type: none"><li>- Coût du support technique</li><li>- Très gourmand en ressources matérielles</li></ul>	<ul style="list-style-type: none"><li>- Coût de la licence (1900 €)</li><li>- Coût du support technique</li></ul>

**Tableau 3 : Tableau comparatif des serveurs Web**

A la fin de cette étude, nous allons porter notre choix sur le serveur Web Apache car il est libre, gratuit et multiplateformes et aussi parce que ses performances sont très élevées.

### III. ETUDE COMPARATIVE DES SGBD SPATIAUX

Cette section nous permettra de nous intéresser à une étude comparative de trois systèmes de gestion de base de données qui gèrent les cartouches spatiales. Il s'agit

de MySQL, Oracle et PostgreSQL. Cette étude a été réalisée par LINAGORA et CampToCamp pour le compte du CNES (Centre National d'Etudes Spatiales).

Les comparaisons ont été effectuées sur les versions MySQL 5.1 BETA, PostgreSQL 8.1.4 (et sa cartouche spatiale PostGIS 1.1.6), Oracle (et ses cartouches spatiales Oracle-10gR2\_spatial.ods, Oracle-10gR2\_locator.ods) suivant les points : Modèle Objet, Indexes Spatiaux, Système de Référence Spatiale, Prédicats, Opérateurs, Autres fonctions et Métadonnées.

MySQL	PostgreSQL	Oracle
<b><u>Modèle Objet</u></b> <i>(Il s'agit d'un modèle permettant de gérer les objets géométriques)</i>		
<ul style="list-style-type: none"> <li>- Point</li> <li>- Linestring</li> <li>- Polygon</li> <li>- GeometryCollection</li> <li>- MultiPoint</li> <li>- MultiLineString</li> <li>- MultiPolygon</li> <li>- 2D (X-Y)</li> </ul>	<ul style="list-style-type: none"> <li>- Point</li> <li>- Linestring</li> <li>- Polygon</li> <li>- GeometryCollection</li> <li>- MultiPoint</li> <li>- MultiLineString</li> <li>- MultiPolygon</li> <li>- CircularString</li> <li>- CompoundCurve</li> <li>- CurvePolygon</li> <li>- MultiCurve</li> <li>- 2D (X-Y)</li> <li>- 3D (X-Y-Z)</li> <li>- 3D (X-Y-M)</li> <li>- 4D (X-Y-Z-M)</li> </ul>	<ul style="list-style-type: none"> <li>- Point</li> <li>- Linestring</li> <li>- Polygon</li> <li>- GeometryCollection</li> <li>- MultiPoint</li> <li>- MultiLineString</li> <li>- MultiPolygon</li> <li>- Rectangle</li> <li>- ArcLinestring</li> <li>- ArcPolygon</li> <li>- CompoundLinestring</li> <li>- CompoundPolygon</li> <li>- Circle</li> <li>- 2D (X-Y)</li> <li>- 3D (X-Y-Z)</li> <li>- 3D (X-Y-M)</li> <li>- 4D (X-Y-Z-M)</li> </ul>
<b><u>Système de Référence Spatial</u></b> <i>(permet la gestion des systèmes de coordonnées, le changement de systèmes et la prise en compte de coordonnées géocentriques)</i>		
<ul style="list-style-type: none"> <li>- Stockage SRID</li> <li>- Pas de gestion des SRS</li> </ul>	<ul style="list-style-type: none"> <li>- Moteur de projection</li> </ul>	<ul style="list-style-type: none"> <li>- Package SRS: sdo_cs</li> <li>- Schéma complet</li> </ul>

	PROJ4 - Définitions issues d'EPSG (2670+ systèmes) - Ajout de systèmes possible - Pas de transformation implicite (mêmes SRID pour les objets) - Faible support des coordonnées géocentriques	décrivant les SRS - Définitions issues d'EPSG (codes EPSG + codes Oracle) - Gestion géocentrique - Conversion d'unités - Transformations complexes et implicites (filtres uniquement) - Quelques restrictions
<p align="center"><b><u>Prédicats</u></b></p> <p align="center"><i>(Des exemples de prédicats répondants aux spécifications de l'OGC sont : <b>Crosses, Touches, Within, Disjoint</b>)</i></p>		
- Uniquement sur les box  - Déclenchent l'utilisation des index spatiaux	- Tous implémentés  - Respect du nommage norme  - Ne déclenchent pas l'utilisation des index  - Robustes	- Tous implémentés, ou prédicats équivalents - Non respect du nommage - Déclenchent l'utilisation des index
<p align="center"><b><u>Opérateurs</u></b></p> <p align="center"><i>(Opérateurs répondant à la norme OGC. Il s'agit de <b>Union</b> (mot réservé SQL...), <b>Intersection, Difference, Symmetric Difference</b>(SymDifference), <b>Buffer, Convexhull</b>)</i></p>		
- Pas de support	- Support de tous les opérateurs  - Respect du nommage OGC	- Support de tous les opérateurs  - Noms spécifiques - Uniquement Oracle Spatial
<p align="center"><b><u>Métadonnées</u></b></p> <p align="center"><i>(La norme OGC définit deux tables : <b>spatial_ref_sys</b>: qui stocke des Systèmes de Référence Spatiaux et <b>geometry_columns</b>: qui stocke des tables contenant des colonnes spatiales ainsi que les noms des colonnes géométriques)</i></p>		
- Pas de gestion des	- Gestion des métadonnées  - Respect des règles de	- Gestion des métadonnées  - Noms spécifiques: vue

métadonnées	nommage	USER_SDO_GEOM_ METADATA - Schéma complet pour la gestion des SRS
<b><u>Présentation des évaluations quantitatives des cartouches spatiales</u></b>		
<ul style="list-style-type: none"> <li>- Requête généralement les plus rapides des trois systèmes</li> <li>- Jointures entre tables très rapides</li> <li>- Temps prometteurs</li> </ul>	<ul style="list-style-type: none"> <li>- Jointures entre tables relativement lentes</li> <li>- Buffer, ConvexHull prenant beaucoup de temps</li> <li>- Opérateurs performants</li> </ul>	<ul style="list-style-type: none"> <li>- Buffer, ConvexHull très performants</li> <li>- Jointures entre tables très longues (plusieurs dizaines de minutes)</li> <li>- Union de gros polygones très longue (plusieurs heures)</li> <li>- Ordre des opérandes important</li> </ul>

**Tableau 2 : tableau comparatif des SGBD spatiaux**

L'analyse ci-haut menée démontre la supériorité en termes de fonctionnalités d'Oracle et ses cartouches spatiales Oracle-10gR2\_spatial.ods, Oracle-10gR2\_locator.ods. Mais le fait est que Oracle est propriétaire et payant. Nous choisirons donc comme SGBDR le couple PostgreSQL<sup>5</sup>/PostGIS<sup>6</sup> parce qu'il est libre et gratuit et presque aussi complet que Oracle.

#### IV. LES LANGAGES DE SCRIPT

En ce qui concerne les langages de script, Perl, PHP et Python ont quasiment les mêmes fonctionnalités. La différence entre ces langages de script repose sur le fait que PHP est plus adapté à la programmation Web et au développement de sites Web dynamiques. En fait, c'est langage de script le plus populaire, la documentation et des tutoriels sont plus facilement accessibles.

<sup>5</sup> <http://www.postgresql.org>

<sup>6</sup> <http://www.postgis.org>

## V. AUTRES LOGICIELS UTILISES

Pour la réalisation de notre travail, nous avons eu à utiliser comme logiciels :

- PHP PgAdmin

Cette application est une interface qui permet d'administrer le système de gestion de base de données via le serveur Web.

- Quantum GIS Tethys

C'est une solution SIG cliente qui a l'avantage d'être Open Source. Doté d'une interface conviviale et en français, QGIS permet de réaliser un nombre d'opérations intéressantes.

- Il peut indifféremment importer des couches vecteurs ou rasters.
- Il se connecte à PostGIS pour récupérer les géométries et valeurs attributaires des tables choisies.
- Il possède des outils d'édition des couches chargées.

Il est à noter que nous avons utilisé ms4w qui est un package contenant le serveur web Apache, le moteur cartographique MapServer et qui gère le langage de script PHP.

## VI. SYNTHESE DES DIFFERENTS OUTILS PRESENTES

Le tableau ci-dessous récapitule les différentes fonctionnalités de chaque outils que nous venons de présenter.

Distribution	Version	Fonction
Apache http Server (ms4w 2.3.1)	2.2.10	Serveur web
Mapserver (ms4w 2.3.1)	5.2.1	Moteur cartographique
PostgreSQL/PostGIS	8.4 (PostGIS 1.5)	SGBD
PHP (ms4w 2.3.1)	5.2.6	Interpréteur de script
PHP PgAdmin	4.1	Interface d'administration de la BD
Quantum GIS Tethys	1.4.0-1	Client permettant de visualiser et traiter des données géographiques

Tableau 3 : Synthèse des fonctionnalités

## **CHAPITRE III : ANALYSE ET CONCEPTION**

---

*Dans ce chapitre, nous présenterons les exigences des besoins et fonctionnalités. Nous y ferons l'analyse de ces besoins et nous en ressortirons une approche de modélisation de l'architecture logique du système.*

### **I. METHODES D'ANALYSE ET DE CONCEPTION**

Une méthode d'analyse et de conception a pour objectif de permettre la formalisation des étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client.

La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités, de performance, de robustesse, de maintenance, de sécurité, d'extensibilité, etc...

La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, afin d'en faciliter la réalisation.

Pour la modélisation de notre application, nous utiliserons la méthode UML (*Unified Modeling Language*).

#### **I.1. PREAMBULE**

Pour faire face à la complexité croissante des systèmes d'information, de nouvelles méthodes et outils ont été créées. La principale avancée des quinze dernières années réside dans la programmation orientée objet (P.O.O.).

Face à ce nouveau mode de programmation, les méthodes de modélisation classique (telle MERISE) ont rapidement montré certaines limites et ont dû s'adapter. De très nombreuses méthodes ont également vu le jour comme Booch, OMT ...

Dans ce contexte et devant le foisonnement de nouvelles méthodes de conception « orientée objet », l'Object Management Group (OMG) a eu comme objectif de définir une notation standard utilisable dans les développements informatiques basés sur l'objet. C'est ainsi qu'est apparu UML (*Unified Modified Language* « langage de modélisation

objet unifié »), qui est issu de la fusion des méthodes Booch, OMT (Object Modelling Technique) et OOSE (Object Oriented Software Engineering).

Issu du terrain et fruit d'un travail d'experts reconnus, UML est le résultat d'un large consensus. De très nombreux acteurs industriels de renom ont adopté UML et participent à son développement.

En l'espace d'une poignée d'années seulement, UML est devenu un standard incontournable.

## **I.2. UML**

UML est une méthode de modélisation orientée objet développée en réponse à l'appel à proposition lancé par l'OMG (*Object Management Group*) dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets.

En effet, c'est en fin 1997 qu'UML devient une norme OMG qui est organisme à but non lucratif qui fut créé en 1989 à l'initiative de grandes sociétés (HP, Sun, Unisys, American Airlines, Philips...), dont son rôle est de promouvoir des standards qui garantissent l'interopérabilité entre applications orientées objet, développées sur des réseaux hétérogènes.

UML comble une lacune importante des technologies objet. Il permet d'exprimer et d'élaborer des modèles objet. Il a été pensé pour servir de support à une analyse basée sur les concepts objet.

UML est un langage formel car défini par un métamodèle. Le métamodèle d'UML décrit de manière très précise tous les éléments de modélisation (les concepts véhiculés et manipulés par le langage) et la sémantique de ces éléments (leur définition et le sens de leur utilisation).

En fait, UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet.

Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions. L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions. Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en font un langage universel.

Toutefois, UML est un langage de modélisation. La plupart des méthodes proposent un langage de modélisation et un processus. Le langage de modélisation représente la

conception et le processus envisage la marche à suivre lors de la conception. UML est utilisée pour spécifier un logiciel et/ou pour concevoir un logiciel. Il est intuitif, plus simple et homogène.

Bref, la puissance et l'intérêt d'UML est qu'il normalise la sémantique des concepts qu'il véhicule.

UML comporte treize types de diagrammes qui permettent de visualiser et de manipuler les éléments de la modélisation. Ils sont dépendants hiérarchiquement et se complètent. Ils se répartissent en trois grands groupes :

- Diagrammes structurels ou diagrammes statiques
- Diagrammes comportementaux ou diagrammes dynamiques
- Diagrammes d'interactions

### **I.2.1. Diagrammes structurels ou diagrammes statiques**

Ces diagrammes permettent de visualiser, spécifier, construire l'aspect structurel du système d'informations. Ces diagrammes sont :

– *le diagramme de classes* qui permet de montrer la structure interne du système. Il représente la structure statique en termes de classes et de relations.

– *le diagramme d'objets* qui permet la représentation des instances et s'utilise de manière ponctuelle pour montrer l'effet d'une interaction.

– *le diagramme de composants* qui décrit le système modélisé sous forme de composants réutilisables et mettent en évidence leurs relations de dépendance. Bref, Il montre la mise en œuvre physique des modèles de la vue logique avec l'environnement de développement.

– *le diagramme de déploiement* qui représente le déploiement des composants sur les dispositifs matériels (ordinateurs, réseaux, périphériques, etc...), leur disposition physique (connexions) et la disposition des exécutables (représentés par des composants) sur ces éléments matériels. Bref, il montre la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.

– *le diagramme de paquetages* qui représente l'organisation logique du modèle et les relations entre packages.

– *le diagramme de structures composites* qui représente l'organisation interne d'un élément statique complexe.



### **I.2.2. Diagrammes comportementaux ou diagrammes dynamiques**

Ils modélisent les aspects dynamiques du système c'est-à-dire les différents éléments qui sont susceptibles de subir des modifications. Ce sont :

- *le diagramme de cas d'utilisation*

Ce diagramme représente des fonctions du système du point de vue de l'utilisateur. Il décrit les possibilités d'interaction entre le système et les acteurs c'est-à-dire toutes les fonctionnalités que doit fournir le système.

- *le diagramme d'états-transitions* qui représente le comportement d'une classe en termes d'état. Il montre la manière dont l'état du système (ou de sous-parties) est modifié en fonction des événements du système.

- *le diagramme d'activités* qui représente le comportement d'une opération en termes d'actions et le déclenchement d'événements en fonction des états du système. Ils permettent de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

### **I.2.3. Diagrammes d'interactions**

Ces diagrammes permettent de représenter les objets et leur interaction à travers le système.

- *le diagramme d'interaction ou collaboration* qui est une variante du diagramme d'activités où les nœuds sont des interactions. Il montre comment les instances de classes communiquent pour réaliser une certaine fonctionnalité.

- *le diagramme de séquence* qui représente temporellement des objets et leurs interactions et séquentiellement le déroulement des traitements et des interactions entre les éléments du système et/ou des acteurs.

- *le diagramme de communication* qui élabore la représentation temporelle des objets et leurs interactions ainsi que la représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.

- *le diagramme de temps* qui représente les interactions où l'aspect temporel est mis en valeur et fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utiles pour la maîtrise d'ouvrage sont les diagrammes d'activités, de cas d'utilisation, de classes, d'objets, de séquence et

d'états-transitions. Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'œuvre à qui ils permettent de formaliser les contraintes de la réalisation et la solution technique.

Etant donné que nous allons utiliser comme méthode d'analyse UML, il est important de donner quelques concepts importants de l'approche objet.

## **II. CYCLE DE VIE DU LOGICIEL**

Le développement d'un logiciel se fait suivant un cycle appelé le cycle de vie du logiciel. Le cycle de vie est décomposé en phases de développement :

- **Définition des objectifs**

Il consiste à définir la finalité du projet et son inscription dans une stratégie globale. C'est pendant cette phase qu'est défini un schéma directeur dans le cas de la création ou de la rénovation d'un système d'information complet d'une entreprise prenant en compte la stratégie de l'entreprise.

- **Analyse des besoins et faisabilité**

Dans cette phase, on exprime, recueille et formalise les besoins du demandeur (le client) et de l'ensemble des contraintes.

- **Spécifications**

Ici, on doit dégager clairement les fonctions, les interfaces (machines, utilisateurs, environnement), les contraintes (performance, mémoires, logiciels, machine). En fait, cette phase définit le quoi du logiciel après dialogue avec le client.

- **Conception générale et Conception détaillée**

Dans la phase de conception générale, on formalise les spécifications élaborées de l'architecture générale du logiciel.

Dans la phase de conception détaillée, on définit de façon précise chaque sous-ensemble du logiciel.

- **Codage et tests unitaires**

Chaque module est ensuite codé et testé indépendamment des autres.

A l'issue de cette phase les produits intermédiaires sont :

- les modules codés et testés
- la documentation de chaque module
- les résultats des tests unitaires.

- **Intégration**

Ici, on assure l'interfaçage des différents modules du logiciel. Chaque module testé est intégré avec les autres suivant le plan d'intégration et l'ensemble est testé conformément au plan de tests.

- **Qualification**

Dans cette phase, on vérifie la conformité du logiciel aux spécifications initiales, procéder aux tests. Cette phase termine le développement.

A l'issue de cette phase le logiciel est prêt à la mise en exploitation.

- **Documentation et mise en production**

On produit les informations nécessaires pour l'utilisation du logiciel et pour des éventuelles évolutions futures. Ensuite, on livre et valide le produit.

- **Maintenance**

C'est pendant cette phase que tous les efforts de documentation faits pendant le développement seront particulièrement appréciés de même que la transparence de l'architecture et du code. On entreprend les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) du logiciel.

Ces phases sont échelonnées dans le temps. Une phase se termine par la remise d'un (ou plusieurs) document(s) validé(s) conjointement par l'utilisateur et le chef de projet. Une phase de développement se termine lorsque la revue de cette phase est faite. Une phase ne peut commencer que lorsque la précédente est terminée. La décomposition en phases de développement permet donc le suivi du projet par l'utilisateur.

### **III. EXPRESSION DES BESOINS ET FONCTIONNALITES**

#### **III.1.Exigences des besoins**

Pour la gestion des données touristiques, l'application devrait remplir les fonctions suivantes :

- disposer d'une base de données d'informations géographiques relatives à la carte touristique ;
- permettre la mise à jour de manière régulière des informations contenues dans la base de données ;
- connaître les statistiques exactes de fréquentation des sites éco-touristiques ;

- répertorier les infrastructures touristiques (hôtels, restaurants, voies d'accès, établissements de loisirs, sites touristiques) et les introduire dans la base de données.
- gérer les utilisateurs de la base de données (droits et paramétrage de l'application).
- offrir aux touristes la possibilité à travers Internet de consulter la carte et de l'imprimer au 1/200 000è.

### **III.2. Exigences fonctionnelles**

Après présentation des besoins, le système d'informations touristiques regroupera les fonctionnalités suivantes :

- *Enregistrement d'informations dans la base de données*

Un utilisateur pourra introduire de nouvelles données dans le système d'informations géographiques du tourisme via l'interface utilisateur de l'application.

- *Requêtes sur les données du système d'informations géographique*

Un utilisateur pourra interroger le système pour recueillir des informations sur des données du SIG.

- *Recherche de donnée*

Un internaute doit pouvoir effectuer une recherche de donnée sur le site, en fonction de mots clés ou une recherche rapide.

- *Analyse thématique*

L'utilisateur doit pouvoir afficher les informations en fonction des thèmes.

- *Zoom à une échelle de 1/200000*
- *Impression/exportation*

La carte générée doit pouvoir être imprimée et les informations recueillies doivent être exportées sous des formats usuels.

### **III.3. Exigences non fonctionnelles**

- la seule ressource nécessaire au client pour l'utilisation de l'application devra être un ordinateur muni d'un navigateur web ;
- le serveur cartographique doit être accessible via une page d'accueil d'interface simple et ergonomique ;

- la recherche d'informations sur le site doit être simple, rapide et les informations affichées doivent répondre de manière concise aux données recherchées ;
- des administrateurs pourront effectuer des mises à jour à distance sur la base de données.

#### IV. ANALYSE ET CONCEPTION

Le système est conçu pour permettre la gestion des données liées au tourisme. Parmi les diagrammes UML supra-cités, nous en énumérons quelques qui permettront d'avoir une vue sur les aspects du fonctionnement du système. Mais avant, nous élaborons les utilisateurs du système.

##### IV.1. Les utilisateurs

Comme utilisateur, nous aurons :

- **L'administrateur**

C'est le super utilisateur du système. Il gère le système, modifie les droits des utilisateurs, ajoute ou supprime des profils.

- **L'administrateur de la base de données**

Il est chargé d'administrer la base de données du SIG. En fait, ses fonctions seraient d'effectuer des mises à jour sur la base de données et d'ajouter de nouvelles couches.

- **Le responsable**

Il a des fonctions spécifiques telles que l'insertion, la modification et de suppression de données dans les tables de la base de données via l'interface d'administration.

- **Le visiteur**

Il peut consulter la carte et l'imprimer ou l'exporter sous un certain format.

##### IV.2. Les cas d'utilisation

Encore appelé *use cases*, on présentera les différents cas des utilisateurs cités ci-dessus :

- ✓ **L'administrateur du système**

Les cas d'utilisation de l'administration à effectuer sont les suivants (*voir figure 4*) :

- Authentification : ce cas d'utilisation correspond à la saisie du compte et du mot de passe pour l'accès à la zone Administration;

- Créer Profil : il peut créer un ensemble de droits d'un profil;

- Modifier Profil: il peut modifier des droits d'un profil ;

- Créer Utilisateur: l'administrateur peut créer un utilisateur;

- Modifier Utilisateur: de même, il peut modifier le profil d'un utilisateur.

#### ✓ ***L'administrateur de la base de données***

Ses cas d'utilisation sont les suivants (*voir figure 4*) :

- Authentification

- Ajouter table : il peut ajouter une nouvelle table dans la base de données ;

- Modifier table : il élaborera la modification des attributs d'une table dans la base de données.

#### ✓ ***Le Responsable***

Ces actions sont les suivantes (*voir figure 4*) :

- Authentification

- Ajouter Donnée : il peut ajouter une donnée sur une couche liée à une table de la base de donnée via l'interface cartographique ;

- Modifier Donnée : il peut modifier les attributs d'une ligne d'une table de la base de données ;

- Supprimer donnée : il peut supprimer une donnée dans la table de la base de données.

#### ✓ ***Le Visiteur***

Les cas d'utilisations du visiteur sont les suivants (*voir figure 4*) :

- Sélectionner couche : il peut sélectionner la couche qui sera affichée par la carte ;

- Afficher carte ;

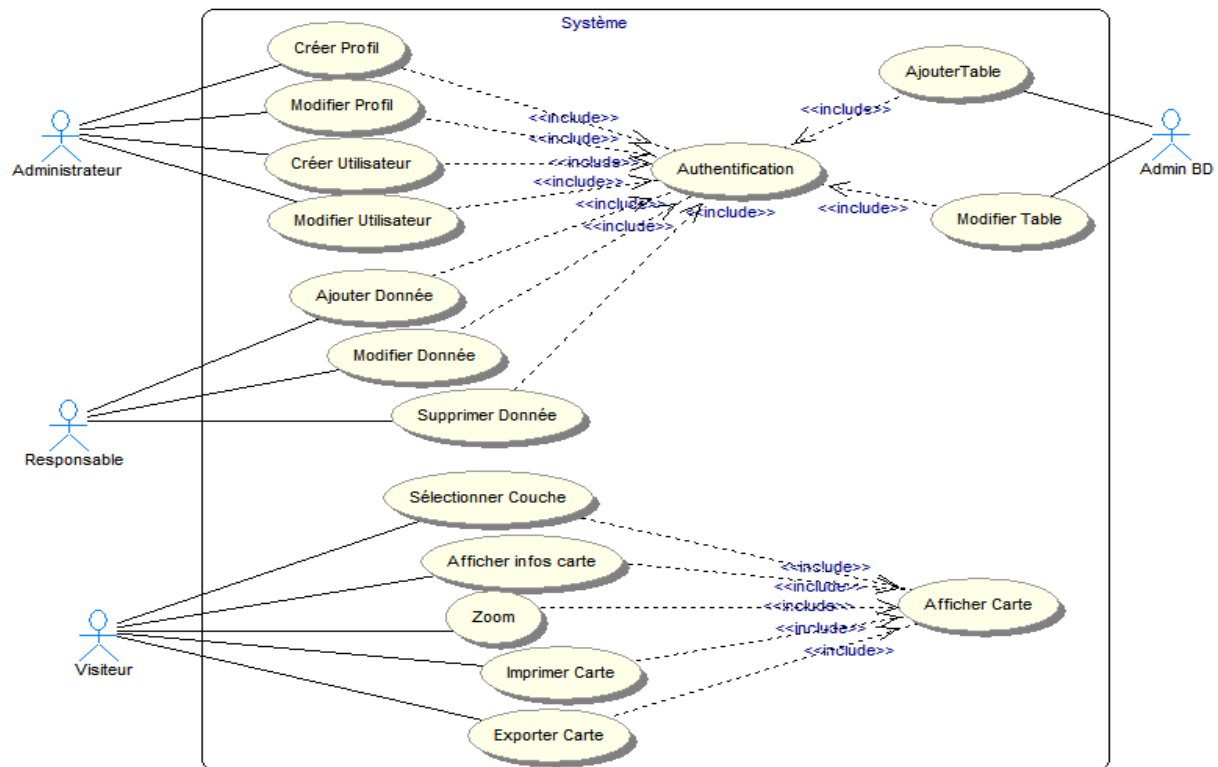
- Afficher des informations sur la carte ;

- Zoom : afficher la carte à une échelle plus grande ou plus petite ;

- Rechercher : recherche une donnée précise s'il a une connaissance de ce nom ;

- Imprimer carte ;

- Exporter carte.



**Figure 5 : Diagramme de cas d'utilisation**

Après avoir élaboré le diagramme de cas d'utilisation, nous allons à présent décrire les scénarios textuels de deux use cases notamment l'authentification et la création d'un utilisateur.

#### ⇒ **Cas d'authentification**

Acteur : Administrateur, Responsable

Objectif : Authentification de l'utilisateur pour accès à la zone d'administration.

Pré condition : Utilisateur actif

Post Condition : L'utilisateur est authentifié

Scénario nominal :

1. Le système affiche la page d'accueil
2. L'utilisateur sélectionne le menu Administration
3. Le système affiche le formulaire d'authentification
4. L'utilisateur entre les informations dans le formulaire
5. Le système vérifie la validité des informations
6. Le système enregistre la date et l'heure d'authentification, ainsi que l'identifiant de l'utilisateur authentifié

7. Le système affiche la page Administration correspondant au profil de l'utilisateur.

Alternatives :

5.a. Les informations entrées par l'utilisateur ne sont pas valides

- i) le système renvoie un message d'erreur
- ii) le système affiche la page d'accueil

Le cas d'utilisation se termine par un échec.

⇒ **Cas de création d'utilisateur**

Acteur : Administrateur

Pré condition : Administrateur Authentifié

Post Condition : Utilisateur crée

Scénario nominal :

1. Le système affiche la page Administration
2. L'utilisateur sélectionne l'onglet Ajouter Utilisateur
3. Le système affiche le formulaire d'enregistrement d'utilisateur
4. L'administrateur entre les informations concernant l'utilisateur dans le formulaire
5. Le système vérifie que les informations ne sont pas déjà présentes dans la base de données
6. Le système enregistre les informations dans la base de données
7. Le système envoie un message de validation réussie.

Alternatives :

4.a. Erreur sur le formulaire

- i) Un message d'erreur s'affiche
- ii) Le système réaffiche le formulaire

5.a. les informations entrées sont déjà présentes dans la base de données

- i) Un message d'erreur s'affiche

Dans ce cas, il y a échec.

⇒ **Cas d'ajout d'une donnée (voir figure 5)**

Acteur : Responsable

Pré condition : Responsable Authentifié



Post Condition : Donnée ajoutée

Scénario nominal :

1. Le système affiche la page Administration
2. L'utilisateur sélectionne la table à ajouter une donnée
3. Le système affiche les informations relatives de la table proprement dite
4. L'utilisateur sélectionne l'onglet Ajouter donnée
5. Le système affiche le formulaire d'enregistrement de donnée
6. L'utilisateur entre les informations concernant la table dans le formulaire
7. Le système vérifie que les informations ne sont pas déjà présentes dans la base de données
8. Le système enregistre les informations dans la base de données
9. Le système envoie un message d'ajout réussi.

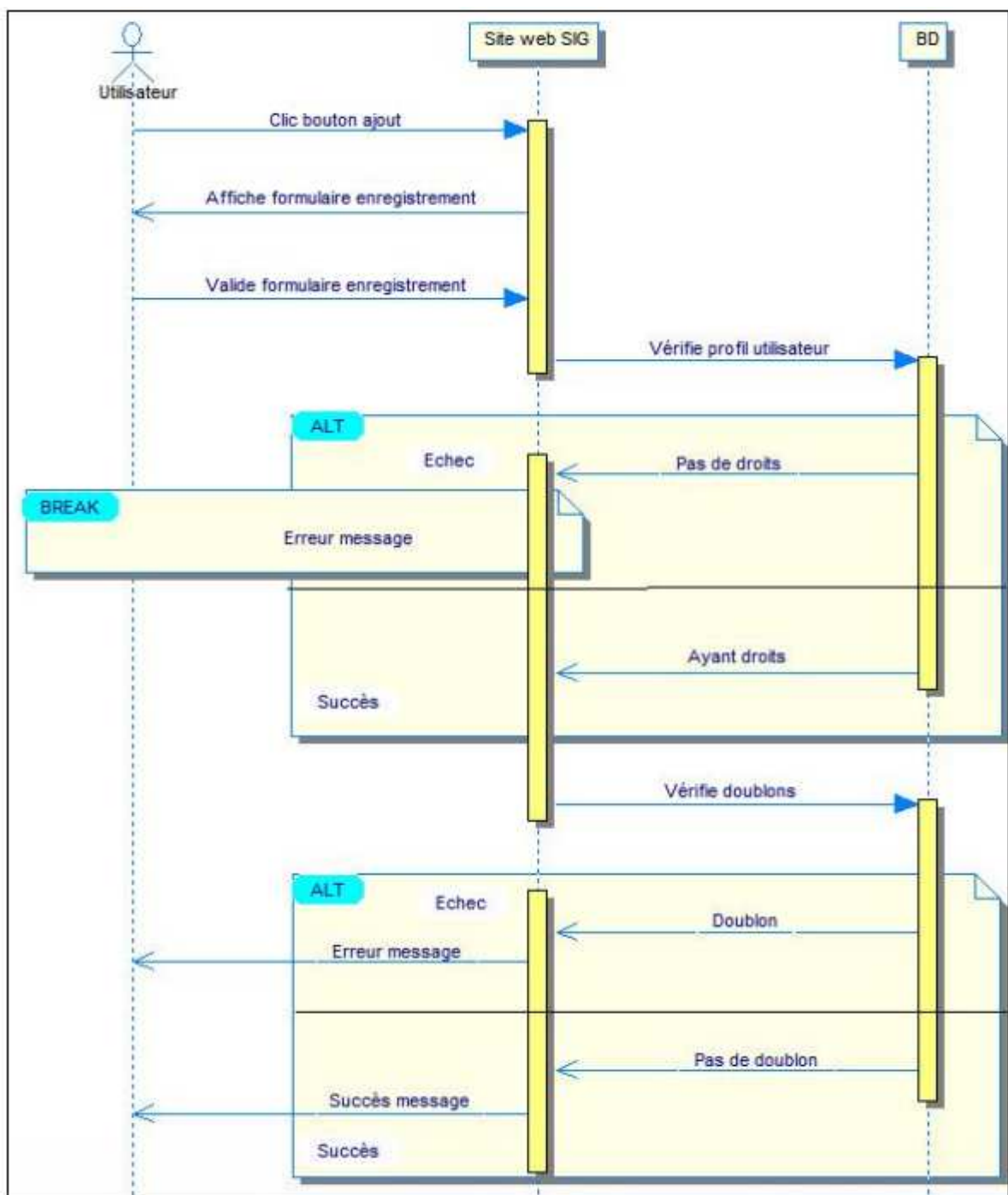
Alternatives :

- 6.a. Erreur sur le formulaire
  - i) Un message d'erreur s'affiche
  - ii) Le système réaffiche le formulaire
- 7.a. les informations entrées sont déjà présentes dans la base de données
  - i) Un message d'erreur s'affiche

Dans ce cas, il y a échec.

### **IV.3. Diagrammes de séquence**

A cette étape, nous allons en présenter deux de notre système, à savoir les diagrammes de séquence d'enregistrement des informations dans la base de données et de navigation sur la carte.



**Figure 6 : Diagramme de séquences d'enregistrement d'informations dans la base de données**

Dans la figure ci-dessous, il est question de montrer les différentes séquences de navigation que peut effectuer un utilisateur sur la carte.

Ainsi, il peut sélectionner une couche qui s'affichera après requête. Il peut en outre effectuer un zoom, déplacer la carte et surtout avoir les informations par rapport aux différentes couches sélectionnées.

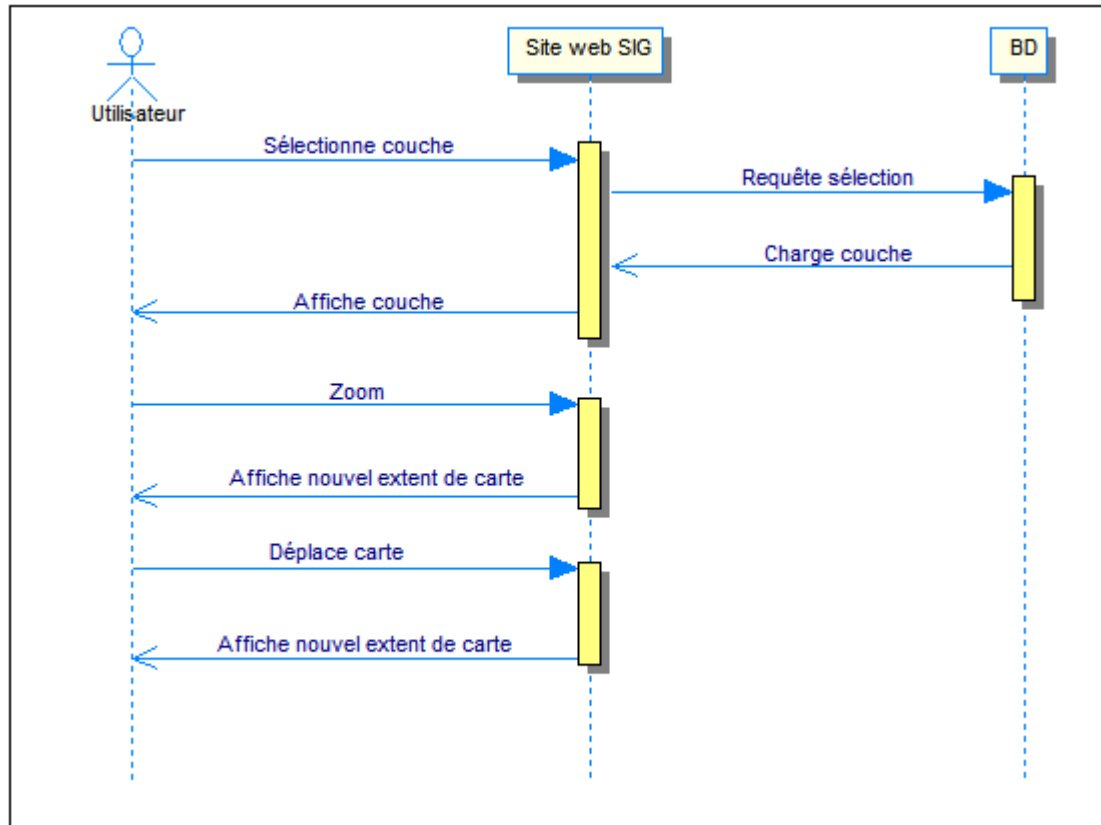


Figure 7: Diagramme de séquence navigation sur la carte

#### IV.4. Diagrammes d'activités

Après avoir donné les différents cas d'utilisation, nous allons à présent décrire ces use cases grâce à des diagrammes d'activités. La description portera sur les use cases Authentification et Créer utilisateur.

Pour le cas d'authentification, l'utilisateur entre les informations y référant par rapport à son profil. Si ces informations sont valides, le système vérifie le profil et lui permet d'accéder à son compte. Mais si le compte n'est pas valide, le système renvoie le formulaire d'authentification.

Pour le cas de création d'utilisateur, l'administrateur doit remplir un formulaire. Si le formulaire rempli est complet, le système vérifie s'il n'y a pas doublon et crée l'utilisateur en lui attribuant un profil. Mais s'il y a doublon, le système renvoie le formulaire de création d'utilisateur qui permettra à l'utilisateur de le remplir à nouveau.

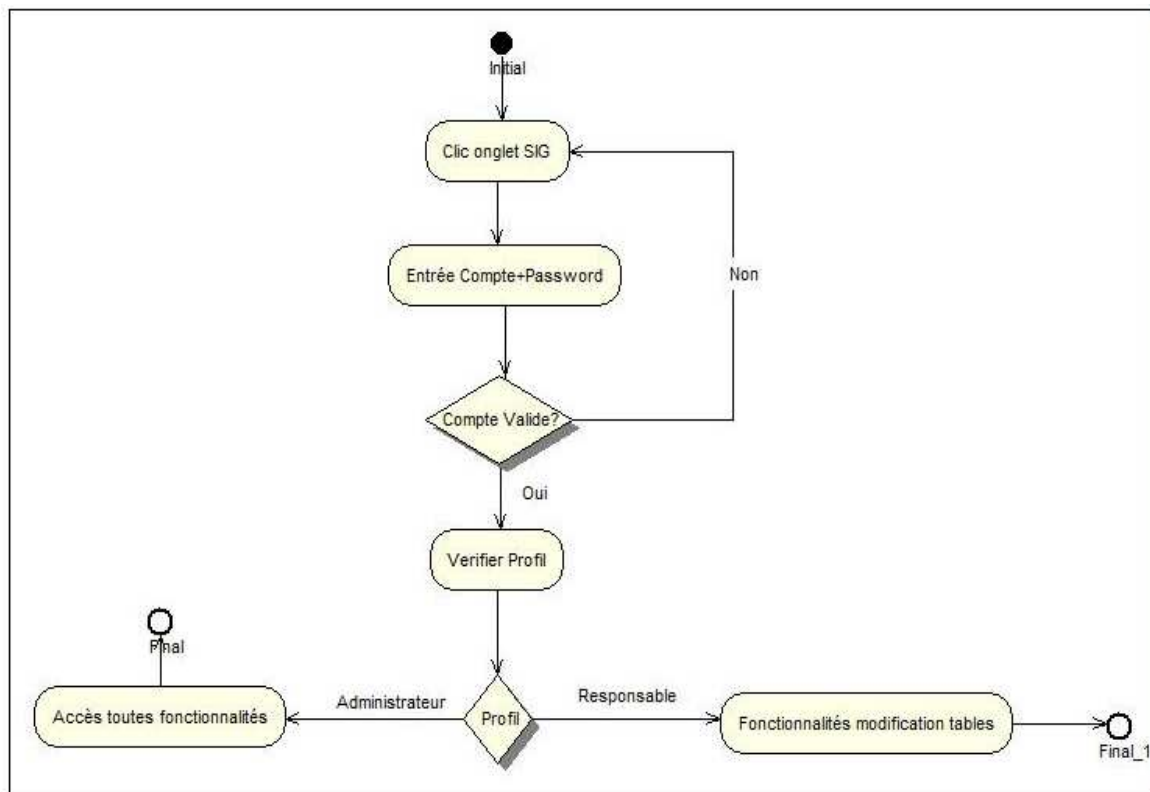


Figure 8 : Diagramme d'activités authentification

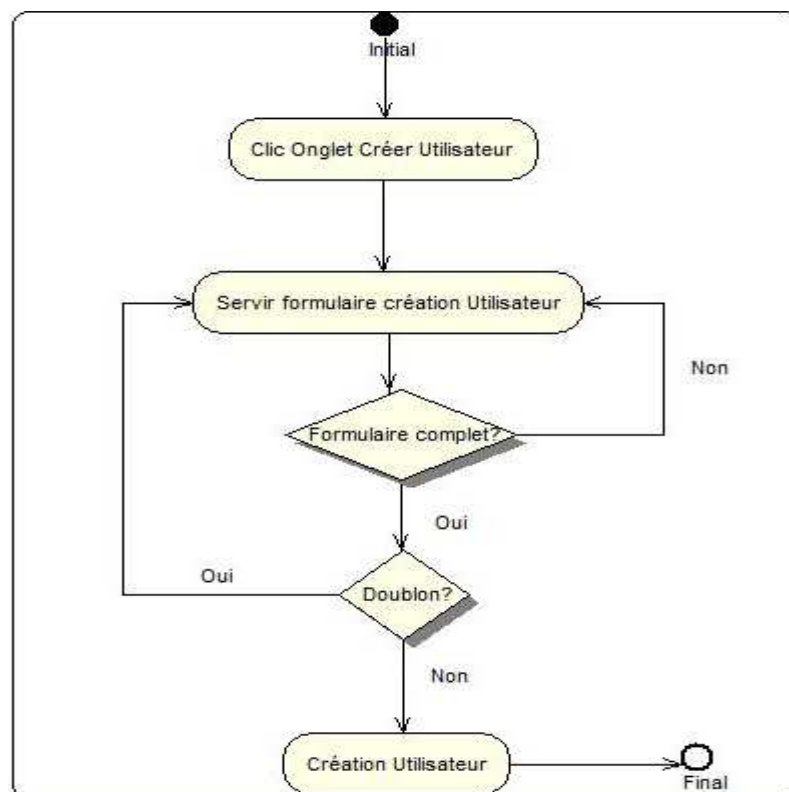


Figure 9 : Diagramme d'activités création utilisateur

## IV.5. Diagramme de classes

A cette étape, nous savons que ce type de diagramme élabore la structure interne du système. En fait, il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation.

Dans ce diagramme, nous identifierons par exemple les entités administratives telles que Région, Département, Localité.

A cela, on ajoutera les entités Site touristique, Hôtel, Guide, Aéroport, Restaurant, etc... qui seront les tables contenant les données à rechercher.

Enfin on identifiera les entités telles que Profil qui énuméra un ensemble de droits, Droit qui donnera les droits de modification de la table ou de ses données dans une base de données et ainsi que Utilisateur.

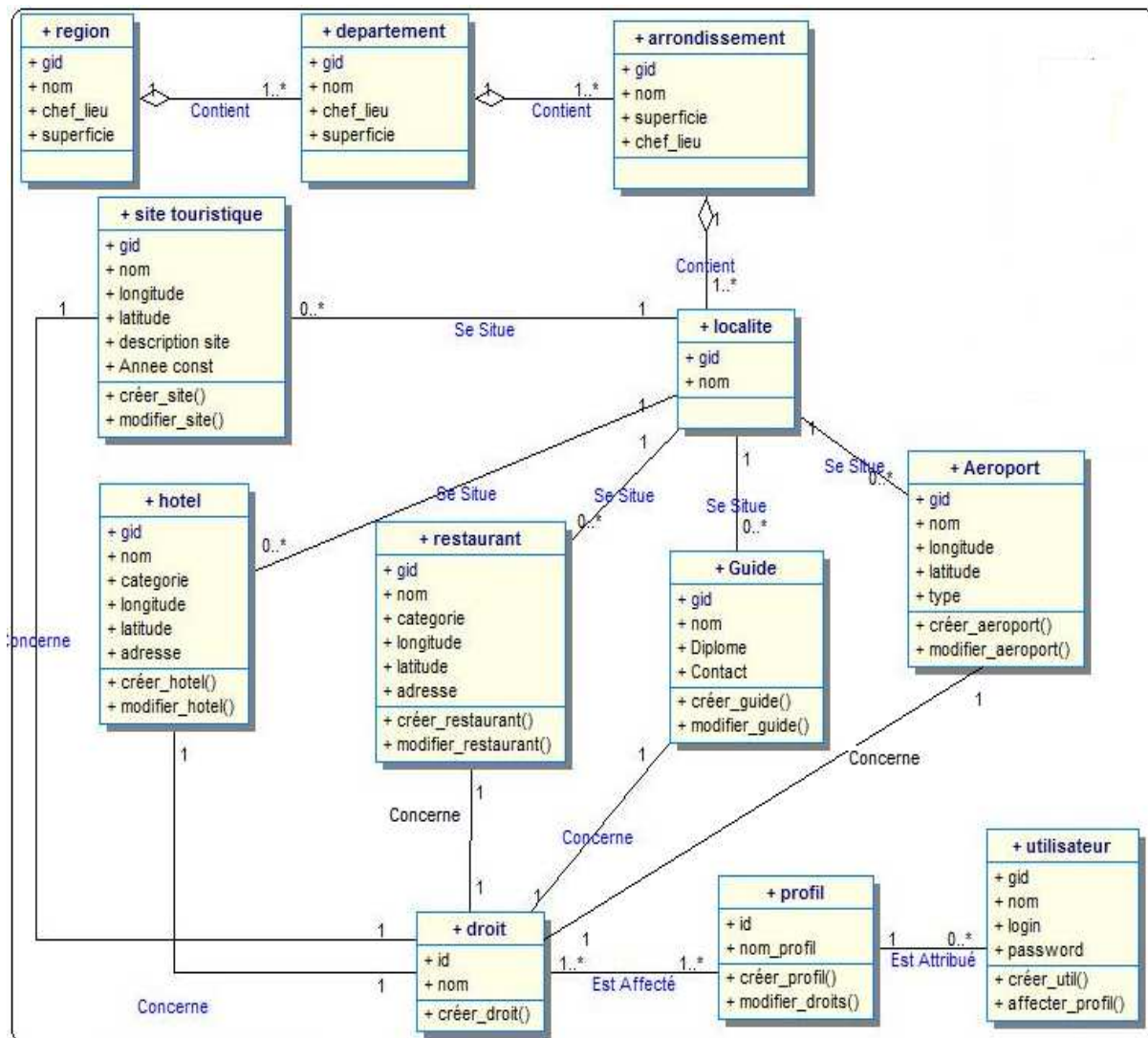


Figure 10 : Diagramme de classes

#### IV.6. Diagrammes d'états-transition

Le diagramme montre la manière dont l'état du système (ou de sous-parties) est modifié en fonction des évènements du système.

Dans cette partie, nous allons présenter les instances de classe Utilisateur et Carte à travers les diagrammes d'états-transition.

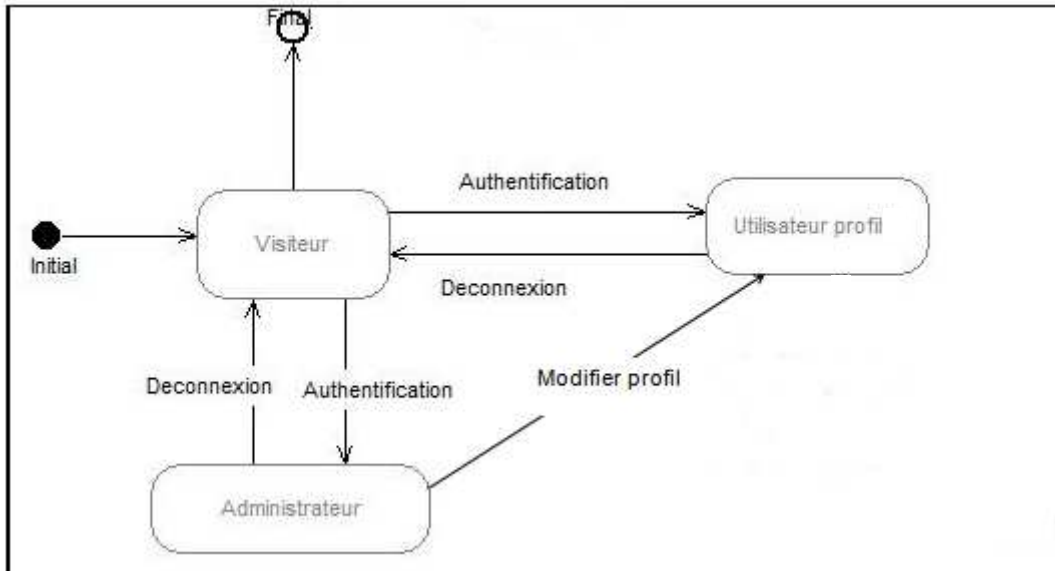


Figure 11 : Diagramme d'états-transitions utilisateur

Si un visiteur a des droits d'accès en tant administrateur, il peut modifier le profil d'un utilisateur. Un administrateur et un utilisateur tout en s'authentifiant peuvent se déconnecter sans ambiguïté.

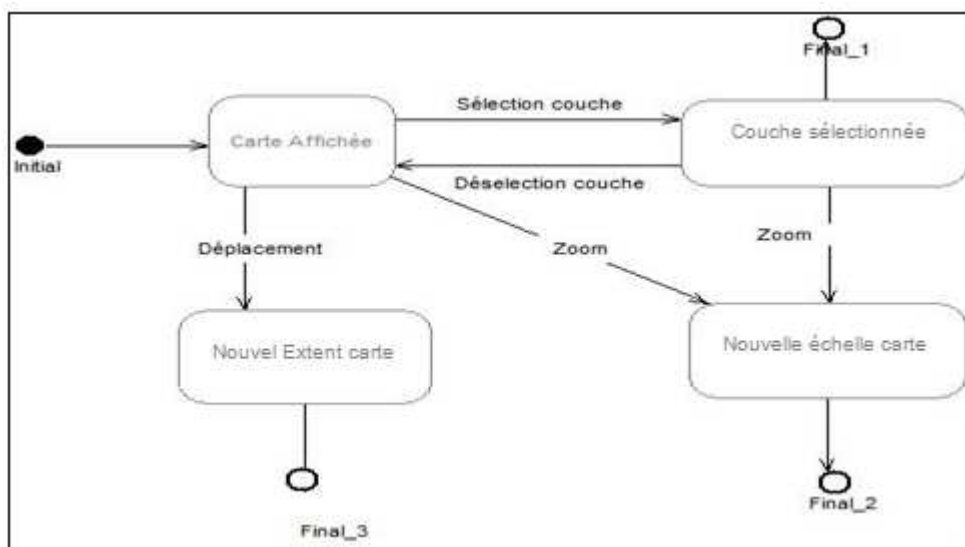


Figure 12 : Diagramme d'états-transition carte

Avec une carte, un utilisateur peut sélectionner une couche, déplacer une carte dans la zone d'affichage et également effectuer un zoom pour modifier l'échelle de la carte.

## CHAPITRE IV : IMPLEMENTATION ET TEST

Les chapitres précédents nous ayant permis de schématiser conceptuellement le fonctionnement de notre système et de déterminer les outils à utiliser pour mettre sur pied notre application, la dernière étape de notre travail consiste à présenter l'architecture physique du système, des outils matériels à utiliser ainsi que des résultats obtenus.

### I. ARCHITECTURE PHYSIQUE DU SYSTEME

#### I.1. Modèle de déploiement du système

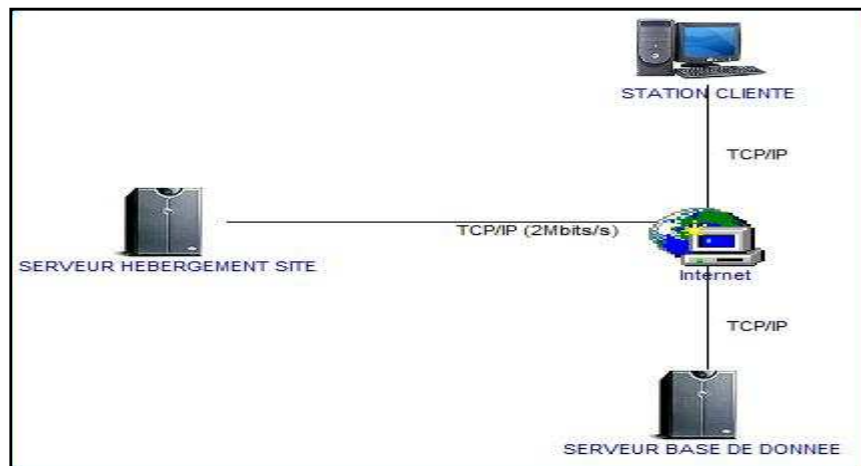


Figure 13 : Diagramme de déploiement

#### I.2. Caractéristiques de la station de travail

Le poste de travail utilisé est doté des caractéristiques suivantes :

<b>Processeur</b>	Intel Pentium 2.4Ghz
<b>Mémoire Vive</b>	1 GB
<b>Carte Graphique</b>	128 MB
<b>Disque dur</b>	80 GB

Tableau 4 : Caractéristiques de la station de travail



## II. REALISATION DU SIG

### II.1. Calage des informations recueillies

Les données recueillies sont consignées dans un fichier, puis elles ont été introduites dans le logiciel de cartographie MapInfo afin d'y être numérisées. Grâce à celui-ci, elles ont pu être « calées », c'est à dire positionnées dans leur emplacement universel, et ce dans un système de projection donné. Il est important de retenir le système de projection utilisé, car c'est celui-ci qui détermine le SRID qui sera précisé lors de l'importation des fichiers shape dans la base de données.

Le système de projection utilisé lors du calage des informations est le système UTM/WSG84 zone 32N.

### II.2. Création de la base de données spatiale

Lors de la configuration de notre système de gestion de base de données PostgreSQL, nous devons tout d'abord préciser l'adresse du serveur hôte et également le numéro de port utilisé. Le serveur est localhost (127.0.0.1) et le port est 5432.

Pour la création de la base de données spatiale, nous devons premièrement lancer la page d'accueil de PostgreSQL (PgAdminIII). Dans le volet Navigateur d'objets, on double clique sur l'onglet PostgreSQL 8.4 (localhost : 5432). Après avoir renseigné le champ mot de passe utilisateur, on effectue un clic droit sur l'icône Base de données et on sélectionne *Ajouter une base de données*.

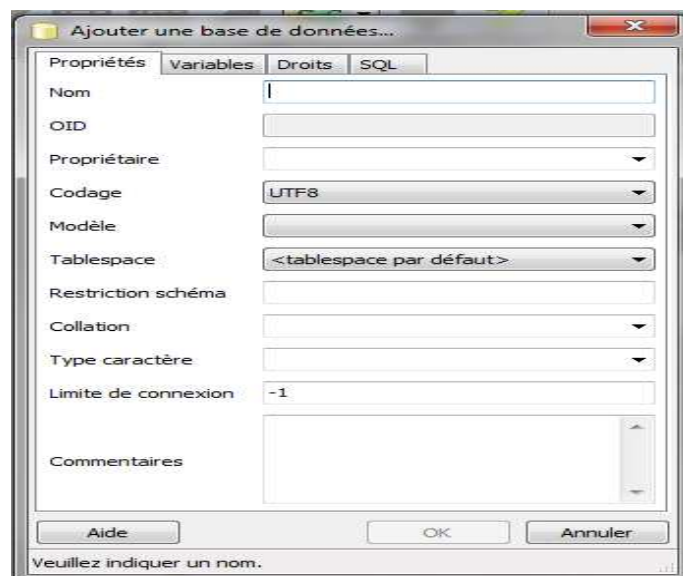


Figure 14 : Ajout d'une base de données

Après avoir entré le nom de la base de données, (dans notre cas tourisme), et sélectionné le propriétaire postgres, on choisit comme modèle template\_postgis.

### II.3. Introduction des données dans la base de données

Pour introduire les données acquises et numérisées dans la base de données, nous avons utilisé l'utilitaire shp2pgsql de PostgreSQL qui permet d'importer des Shapefiles dans une base de données PostgreSQL. Pour utiliser shp2pgsql, il faut déplacer les Shapefiles dans le répertoire bin de PostgreSQL. La syntaxe d'importation de Shapefiles est la suivante :

*shp2pgsql [options1] [nom-fich-shape] [nom-table-BD] | psql [nom-BD] [options2]*

Les options possibles sont les suivantes :

#### **Options1**

-a	Ajoute les données du fichier vecteur dans la table de la base de données. On utilise cette option pour charger plusieurs fichiers qui contiennent les mêmes attributs et les mêmes types de données.
-c	Crée une nouvelle table et la remplit avec le contenu du fichier vecteur (Shape). C'est le mode par défaut.
-d	Supprime la table de la base de données avec les données du fichier vecteur (Shape).
-D	Cette option peut être combiné avec -a, -c et -d. En fait, elle utilise le format d'export de PostgreSQL pour extraire les données. On utilise ceci pour l'insertion d'une grande quantité de données.
-i	Convertit tous les entiers en entiers 32 bits standards.
-I	Crée un index GiST sur la colonne géométrique.
-k	Respecte la syntaxe (cas : majuscules/minuscules) des identifiants (colonne, schéma et attributs).
-p	Produit uniquement le code de création des tables, sans ajouter les données. Cela peut être utilisé lorsque vous avez besoin de séparer complètement les étapes de création de la table et de son remplissage.
-s	Crée et remplit les tables géométriques avec le SRID spécifié.
-W	Spécifie l'encodage des données en entrée (du fichier .dbf). Lorsque cette option

	est utilisée, tous les attributs du .dbf sont convertis de l'encodage spécifié en UTF-8. La sortie SQL résultante contiendra une commande SET CLIENT_ENCODING to UTF8, donc le serveur principal sera capable de reconvertir de l'UTF-8 dans l'encodage choisi dans configuration de la base de données.
--	--

**Tableau 5 : Options 1 de la commande shp2pgsql**

**Options 2**

-h	Précise le nom de l'hôte du serveur de base de données
-p	Précise le port utilisé par la base de données
-U	Précise le nom de l'utilisateur de la base de données

**Tableau 6 : Options 2 de la commande shp2pgsql**

**Exemple de syntaxe d'importation de shapefile :**

« *Shp2pgsql -W latin1 -s 32632 province\_region.shp region/psql -d tourisme -h localhost -p 5432 -U postgres* ».

Avant l'importation des données dans la base de données, il est nécessaire d'obtenir un certain nombre d'informations sur le Shapefile à importer, entre autres :

- le système de projection ;
- l'encodage utilisé ;
- le type géométrique de la table ;
- l'Extent de la carte (coordonnées géographiques du coin inférieur gauche de la carte et de son coin supérieur droit) qui sera utilisé lors de la création du Mapfile.

Ces informations sont acquises grâce à l'utilitaire ogrinfo de la librairie gdal-ogr de Apache. La syntaxe d'utilisation de cet utilitaire est la suivante :

***"ogrinfo -al -so [nom\_fichier\_shape]"***

```

C:\ms4w---\Apache\cgi-bin>ogrinfo -al -so Query1_region.shp
INFO: Open of 'Query1_region.shp'
      using driver 'ESRI Shapefile' successful.

Layer name: Query1_region
Geometry: Polygon
Feature Count: 1
Extent: <12.495894, 1.641500> - <16.187500, 6.135040>
Layer SRS WKT:
GEOGCS["Lat Long for MAPINFO type 0 Datum",
  DATUM["MAPINFO",
    SPHEROID["World Geodetic System of 1984",6378137,298.257223563],
    PRIMEM["Greenwich",0],
    UNIT["Degree",0.017453292519943295]],
  PROJ4[""],
  PROJNAME: String <1.0>
  PROVINCE: String <30.0>
  CHEF_LIEU: String <30.0>
  SUPERFICIE: Real <31.15>

C:\ms4w---\Apache\cgi-bin>_
  
```

Figure 15 : Exemple de syntaxe de ogrinfo

Après importation, un champ est créé automatiquement, le champ *the\_geom* qui stockera l'index géométrique des objets. C'est cet index qui permettra au serveur cartographique de charger les géométries de la base de données vers la carte.

### III. REALISATION DE LA CARTE

Les tables à charger de la base de données par MapServer sont précisées dans le Mapfile. On y précise également l'Extent de la carte à réaliser.

Pour connecter MapServer à la base de données dans le Mapfile, on précise dans l'objet Layer le type de connexion, la donnée à charger. Voici un exemple de connexion à la base de données d'une couche :

```

LAYER

  NAME chute
  CONNECTION "user=postgres password=postgres dbname=tourisme host=127.0.0.1"
  CONNECTIONTYPE POSTGIS
  DATA "the_geom from (select * from chute) foo USING UNIQUE gid USING SRID=32632"
  STATUS Off
  TYPE POINT
  LABELCACHE on
  LABELITEM "nom_site"
  
```

Figure 16 : Connexion à la base de données d'une couche.

## IV. RESULTATS

### IV.1. Page d'accueil du SIG

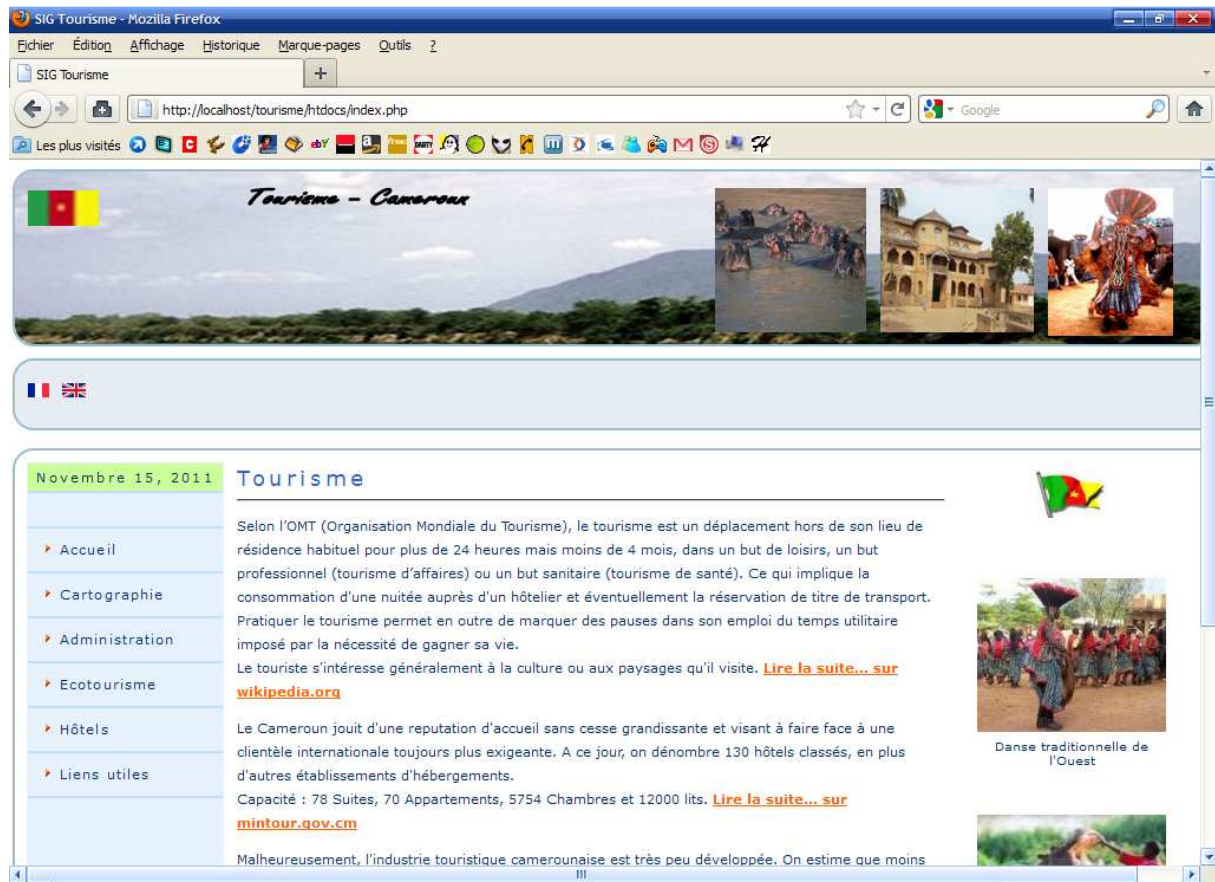


Figure 17 : Page d'accueil du SIG

Depuis la page d'accueil, l'utilisateur peut accéder aux fonctionnalités du système. Il peut avoir accès aux données attributaires du SIG, à la carte où il peut s'authentifier pour passer en mode privilégié.

En outre, il peut avoir des informations sur des différents sites naturels et de constructions, des loisirs et les agences de voyage, la structure d'accueil et d'hébergements des établissements tels que les hôtels, les restaurants, etc...

De plus, le visiteur peut accéder à la carte. Il clique sur le lien *Cartographie*, ce qui lui permet d'effectuer le zoom ainsi qu'une recherche sur une couche bien précise sur la carte, d'afficher une région de la carte et une information sur la carte.

Par ailleurs, si l'utilisateur a des privilèges, il peut y accéder en cliquant sur le lien *Administration*.

## IV.2. La page Administration

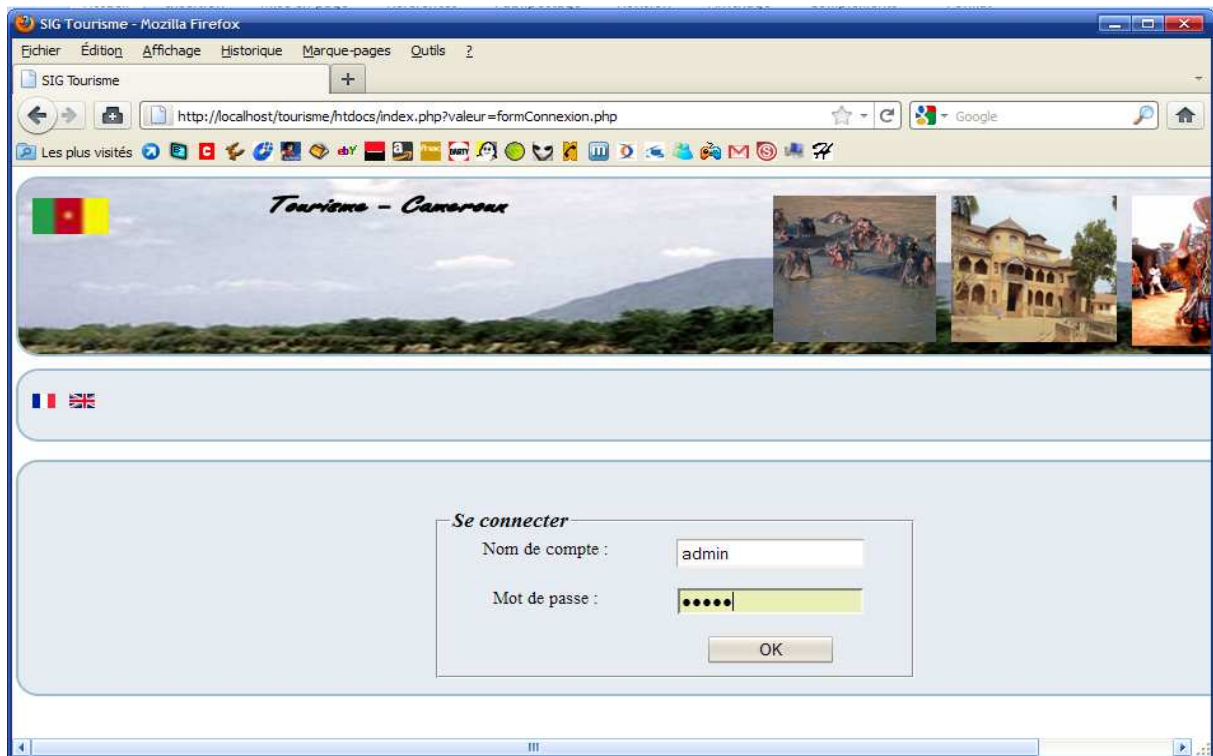


Figure 17 : Authentification d'un utilisateur

A la page Administration, l'utilisateur doit s'authentifier pour passer en mode privilégié.

## IV.3. Consultation et enregistrement de données

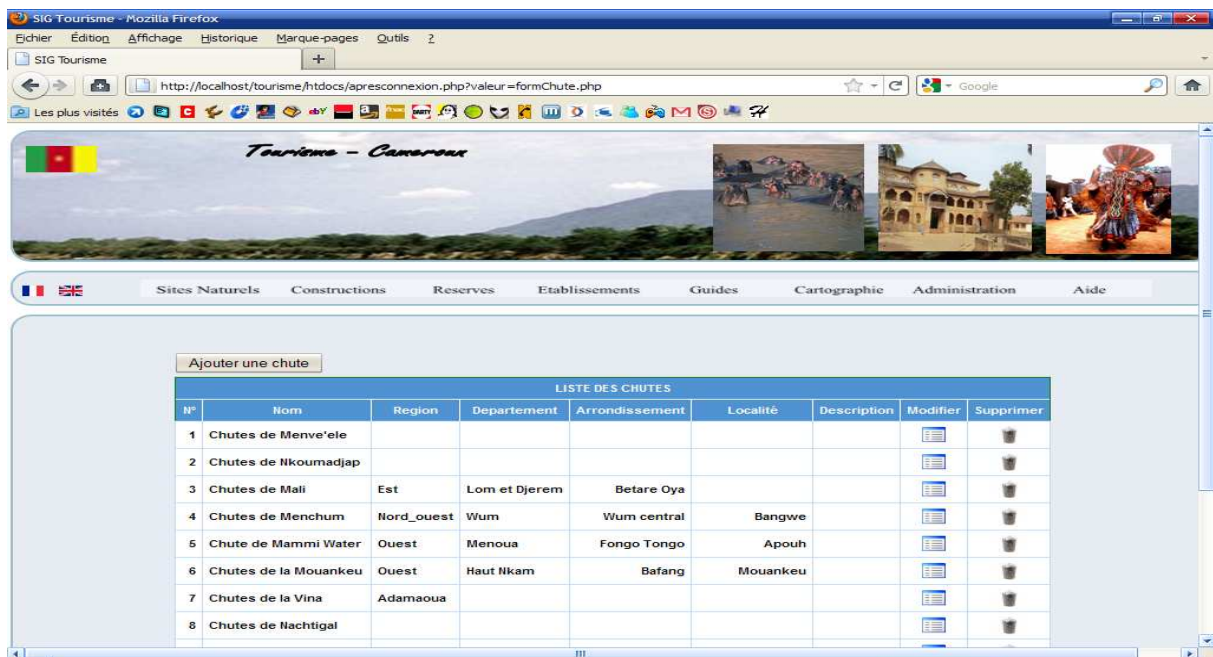


Figure 18 : Données attributaires d'une table



Les informations attributaires sont chargées dans la base, et les coordonnées géographiques sont traitées pour générer l'index géographique *the\_geom*, qui servira pour l'affichage des données sur la carte. Pour éditer une couche, il faut au préalable avoir les droits relatifs à cette couche.

Toutefois, pour enregistrer une nouvelle couche, il faut cliquer sur le bouton *Ajouter une couche* pour afficher le formulaire d'enregistrement ci-dessous.

The screenshot shows a web browser window titled 'SIG Tourisme - Mozilla Firefox'. The address bar displays 'http://localhost/tourisme/htdocs/apresconnexion.php?valeur=formChute.php'. The page header includes the text 'Tourisme - Cameroun' and a navigation menu with links: 'Sites Naturels', 'Constructions', 'Reserves', 'Etablissements', 'Guides', 'Cartographie', 'Administration', and 'Aide'. The main content area is titled 'Enregistrement d'une chute' and contains a form with three sections: 'Infos', 'Localisation', and 'Caractéristiques'. The 'Infos' section has a 'Nom \*' field. The 'Localisation' section has fields for 'Région', 'Département', 'Arrondissement', 'Localité', 'Longitude \*', and 'Latitude \*'. The 'Caractéristiques' section has a 'Description du site' field. A red warning message 'les champs suivis de \* sont obligatoires !' is displayed above the 'Localisation' section. At the bottom right of the form are 'Enregistrer' and 'Effacer' buttons.

Figure 19 : Formulaire d'enregistrement des données d'une table

L'utilisateur entre les informations concernant la table et valide en cliquant sur le bouton *Enregistrer*.

Un utilisateur peut par la suite avoir le droit de modification ou de suppression d'une couche donnée.

#### IV.4. La page Cartographie

Dans cette page, nous avons la liste des tables du SIG affichées sous forme de légende ; nous avons également une carte de référence, la barre d'échelle ainsi que les





## **CONCLUSION ET PERSPECTIVES**

---

Parvenus au terme de notre projet, nous avons pu concevoir un noyau SIG permettant de gérer les données relatives au tourisme au Cameroun et nous avons également pu réaliser une plateforme de cartographie dynamique en ligne qui servira de vitrine pour le tourisme camerounais.

La solution technique adoptée est basée sur un serveur cartographique, logiciel permettant d'analyser et de répondre aux requêtes géographiques des clients.

La méthode d'analyse et de conception utilisée est la méthode UML qui permet de spécifier et de concevoir un logiciel.

Les outils logiciels utilisés sont Open source c'est-à-dire libres et avec un code réutilisable. Ainsi, après les études menées pour le choix des logiciels, nous avons opté pour le serveur cartographique MapServer, comme SGBD PostgreSQL avec sa cartouche spatiale PostGIS ; le serveur Web utilisé est Apache et le langage de script est PHP. Il est à noter que tous ces outils sont téléchargeables sur Internet.

La plateforme utilisée comme système d'exploitation a été un environnement Microsoft Windows. Néanmoins l'application peut aussi s'installer sur un environnement LINUX.

Bref, Le monde de la cartographie est en constante évolution. Ainsi, tous les outils cités ci-dessus (serveurs, bases de données, clients) voient leurs fonctionnalités s'étoffer et se perfectionner tous les jours.

Par ailleurs, les contraintes temporelles imparties n'ont pas permis la gestion des fonctionnalités d'impression/export, ainsi que l'audit des fréquentations des sites touristiques.

En perspectives, pour l'évolution de l'application, nous pourrions implémenter les fonctionnalités d'impression/export, les fonctionnalités d'audit des fréquentations des sites touristiques ainsi qu'une extraction routière à superposer sur la carte ; ceci dans le but de pouvoir répertorier les différents conduits aux sites touristiques.

## **GLOSSAIRE**

---

### **Carte**

En géographie, une carte est une représentation d'un espace géographique. Elle met en valeur l'étendue de cet espace, sa localisation relative par rapport aux espaces voisins, ainsi que la localisation des éléments qu'il contient.

### **Cartographie**

Science désignant l'étude et la réalisation de cartes.

### **Géolocalisation**

La géolocalisation est la localisation d'un objet sur une carte à l'aide de positions géographiques.

### **SIG - Système d'Information Géographique**

Outil informatique permettant le stockage, la gestion, l'analyse et la visualisation de données géographiques.

## BIBLIOGRAPHIE

---

1. *Document de Stratégie pour la Croissance et l'Emploi*, 2009, PDF disponible sur <http://www.afdb.org/fileadmin/uploads/afdb/Documents/Project-and-Operations/Cameroon%20DSCE2009.pdf>
2. Dr. Marcellin Julius NKENLIFACK, Support de cours, *Ingénierie des systèmes logiciels*, version 2009.
3. Nathalie LOPEZ, Jorge MIGUEIS et Emmanuel PICHON, *Intégrer UML dans vos projets*, Eyrolles, 1999.
4. Raphaël THOMAS, *Evolutions d'outils dédiés à l'analyse territoriale et à l'analyse spatiale dans le cadre du projet HyperCarte*, mémoire soutenu en juin 2008.
5. Julien GAULMIN, *Programmer en PHP*, Version 1.44.
6. Philippe DROUOT, Toofik MEDJAMIA et Cédric PROTIERE, *Création et gestion de sites Web professionnels*, Africa Computing, 2003.
7. Site internet de l'architecture pour une solution « serveur cartographique », consulté en décembre 2010, disponible sur <http://mappemonde.mgm.fr/num8/internet/int05401.html>
8. <http://www.cartographie.ird.fr/publi/documents/sig1.pdf>
9. William Steinmetz et Brian Ward, *PHP Clés en main : 76 scripts efficaces pour enrichir vos sites web*, Pearson, version 5 et 6.
10. Olivier SIGAUD, *Introduction à la modélisation orientée objets avec UML*, Edition 2005-2006.
11. Laurent Piechocki, *UML, le langage de modélisation objet unifié*, Edition 2007.
12. [http://fr.wikipedia.org/wiki/Tourisme\\_au\\_Cameroun](http://fr.wikipedia.org/wiki/Tourisme_au_Cameroun)
13. [http://en.wikipedia.org/wiki/Tourisme\\_in\\_Cameroun](http://en.wikipedia.org/wiki/Tourisme_in_Cameroun)
14. <http://www.pole-geomatique-auvergne.org/>

15. <http://www.forumsig.org>
16. <http://www.portalsig.org>
17. <http://mapserver.gis.umn.edu/docs/reference/mapfile>
18. <http://uml.free.fr>
19. <http://www.developpez.com>