

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
*Ministère de l'enseignement supérieur et de la recherche scientifique*  
*Université Mouloud Mammeri de Tizi-Ouzou*  
*Faculté de Génie Électrique & d' Informatique*



*Département Informatique*

# *Mémoire*

*En vue d'obtention du diplôme de Master en Informatique*

*Option : Conduite De Projet Informatique*

## **Thème:**

*L'Adaptabilité entre les systèmes dans  
un réseau P2P*

*Réalisé par :*

*M<sup>elle</sup> : REKKAB Amel*

*Dirigé par :*

*M<sup>r</sup> .KERBICH.M*

**2014/2015**



# Remerciements

*Au terme de ce travail,  
Je tiens en premier lieu à remercier le Bon Dieu pour le courage et la patience  
qu'il m'a donné afin de mener ce projet à terme.*

*Je tiens vivement à remercier mon promoteur M<sup>r</sup> KERBICH.M D'avoir  
accepté de m'encadrer, ainsi que pour son aide précieux son travail  
encourageant, et ses conseils judicieux,*

*Je voudrais également exprimer mes sincères remerciements aux membres du jury  
qui m'ont fait l'honneur d'accepter de juger mon travail.*

*Enfin, mes remerciements s'adressent à tous ceux qui, de loin et de près ont aidé  
à l'élaboration de ce mémoire.*

# Dédicaces

*Je dédie ce modeste travail :*

*À ma chère adorable grand-mère.*

*À ma chers maman Salîha et mon chère papa Mourad que j'adore, pour leur aide et leur soutien tout au long de mes études, et qui ont fait de moi ce que je suis aujourd'hui et j'espère qu'un jour je serai capable de leur donner au moins le minimum car quoiqu'on face on arrivera jamais à leurs rendre tout.*

*À mes très chers frères et sœurs que j'aime :*

*À mon oncle Madjid*

*Et à toute ma famille*

*Que Dieu le tout puissant les protège.*

*À toi ma chère Lynda et à toute ta famille.*

*À tous mes amis (Mazighi, Takfa, Lydia, Katia, Meriam, Nassim, Samia, Chahira, Farida, Nawel, Hannne, Nadia, Nassima)*

*À ceux qui sont la source de mon inspiration et*

*De mon courage, à qui je dois de l'amour et de la*

*Reconnaissance.*

*Sans oublier mes camarades de la promotion 2015.*

*REKKAB Amel*

## Liste des figures

<b>Figure.I.1.</b> Classification des réseaux.....	2
<b>Figure.I.2.</b> Présentation du model OSI.....	12
<b>Figure.I.3.</b> Les principaux protocoles rencontrés sur un réseauTCP/IP .....	13
<b>Figure.II.1.</b> SOA :à la convergence de solution .....	14
<b>Figure.II.2.</b> Schéma montrant le fonctionnement des concepts de SOA .....	5
<b>Figure.II.3.</b> Fonctionnement d'un WSOA entre différents terminaux et les WebServices .....	16
<b>Figure.II.4.</b> Structure de composant de service .....	17
<b>Figure.II.5.</b> ESB et composants de service.....	18
<b>Figure.II.6.</b> Fonctionnement des services Web .....	22
<b>Figure.II.7.</b> Structure d'un message SOAP.....	27
<b>Figure.II.8.</b> Architecture de la conception ascendante .....	33
<b>Figure.II.9.</b> Architecture de la conception descendante.....	33
<b>Figure.II.10.</b> Architecture Client / Serveur .....	34
<b>Figure.II.11.</b> Architecture Peer To Peer .....	34
<b>Figure.III.1.</b> Les composants d'un système d'EIAO.....	40
<b>Figure.III.2.</b> Fonctionnement d'une plate forme e-learning.....	47
<b>Figure.III.3.</b> Page principale de moodle .....	49
<b>Figure.IV.1.</b> la démarche de modélisation de l'application .....	53
<b>Figure.IV.2.</b> Les acteurs participant à notre système.....	55
<b>Figure.IV.3.</b> Diagramme de contexte .....	56
<b>Figure.IV.4.</b> Diagramme de cas d'utilisation « acteur : étudiant».....	59
<b>Figure.IV.5.</b> Diagramme de cas d'utilisation « acteur : enseignant » .....	60
<b>Figure.IV.6.</b> Diagramme de cas d'utilisation « acteur : administrateur » .....	61
<b>Figure.IV.7.</b> Diagramme de séquence pour le cas d'utilisation <<s'inscrire à la plate forme>> .....	64
<b>Figure.IV.8.</b> Diagramme de séquence de cas d'utilisation <<s'authentifier>> .....	65
<b>Figure.IV.9.</b> Diagramme de séquence de cas d'utilisation « envoyer un message à un étudiant ou enseignant ou administration ».....	67
<b>Figure.IV.10.</b> Plan de représentation des schémas conceptuels ....	68
<b>Figure.IV.11.</b> diagramme du schéma logique du site département. ....	69
<b>Figure.IV.12.</b> Diagramme du schéma logique du la plate forme.....	70
<b>Figure.V.1.</b> Architecture de déploiement de la plate forme .....	72
<b>Figure.V.2.</b> Caractéristiques matérielles et logicielles .....	73
<b>Figure.V.3.</b> Page d'accueil Wamp .....	78

<b>Figure. V.4.</b> Administration de MySQL à partir de PhpMyAdmin .....	79
<b>Figure. V.5.</b> Accès à la base de données « notresite »à partir de PhpMyAdmin.....	80
<b>Figure. V.6.</b> Interface Dreamweaver .....	81
<b>Figure. V.7.</b> La page principale de la plate forme .....	82
<b>Figure. V.8.</b> Page d'accueil.....	82
<b>Figure. V.9.</b> Interface de la page du département.....	83
<b>Figure. V.10.</b> Interface de la page du département informatique . .....	83
<b>Figure. V.11.</b> Interface de la page d'administrateur de départemen .....	84
<b>Figure. V.12.</b> espace messagerie.....	84
<b>Figure. V.13.</b> liste des messages .....	85
<b>Figure. V.14.</b> affichage du message.....	85
<b>Figure. V.15.</b> envoyer un message .....	86
<b>Figure. V.16.</b> supprimer un message.....	86

## Sommaire

### *Introduction générale*

# *Chapitre I : les protocoles applicatifs sur Internet*

Introduction :	1
I.1. Les réseaux	1
I.1.1. Définition d'un réseau	1
I.1.2. Réseaux informatiques	1
Définition	1
Objectifs des réseaux informatiques	1
I.1.3. La classification des réseaux	2
• Classification selon l'étendue géographique	2
• Classification selon le mode de communication	3
I.2. Internet	3
I.2.1. Définition	3
I.2.2. Les services d'Internet	3
I.2.3. Fonctionnement	4
I.2.4. L'intranet et l'extranet	5
A. L'intranet	5
Définition	5
Les avantages d'un intranet	5
Les propriétés d'un intranet	5
B. L'extranet	5
Définition	5
I.3. Les protocoles	6
I.3.1. Définition	6
I.3.2. Quelques protocoles	6
A. Les protocoles de communication	6
A.1. Le protocole HTTP	6
A.2. Le protocole TCP	7
A.3. Protocole IP	7
A.5. Le protocole ICMP	7

A.6. Protocole ARP .....	7
A.7. Le protocole RARP .....	7
A.8. Protocole TELNET .....	8
A.9. Le protocole SMTP .....	8
B. DNS .....	8
C. LDAP .....	8
C.1. Définition .....	8
C.2. Les opération de LDAP .....	9
C.3. Pourquoi utiliser LDAP ? .....	9
D. CORBA .....	10
D.1. Définition du CORBA .....	10
D.2. Les avantages du CORBA .....	10
E. Le modele TCP/IP .....	10
E.1. Representation du modele OSI .....	11
E.2. Representation du modele TCP/IP .....	12
Conclusion .....	13

## *Chapitre II : L'adaptabilité et BDD reparties*

Introduction .....	14
II.1. Architecture orientée service .....	14
Introduction .....	14
II.1.1. Définition .....	14
II.1.2. Les concepts de SOA .....	15
II.1.2.1. Définition du concepts service .....	16
II.1.3. Les éléments de base de l'architecture orientée service .....	17
II.1.3.1. Composant de service .....	17
II.1.3.2. Bus d'entreprise .....	17
II.1.3.3. Contrat de service .....	18
II.1.3.4. Données d'échanges et données persistantes .....	18
II.1.4. Déploiement des SOA .....	19
II.1.5. Les apports des Architectures Orientées Services .....	19
II.1.5.1. Les Incovenients .....	19
II.1.5.2. Les Avantages .....	19
II.2. Services Web .....	20

Introduction .....	20
II.2.1.Définition .....	20
II.2.2.L'intérêt d'un service web .....	20
II.2.3.Les caractéristiques d'un service web .....	21
II.2.4.Architecture d'un service web .....	21
II.2.5.Fonctionnement des services web .....	22
II.2.6.Description en couches des services web .....	23
Couche transport.....	23
Couche communication .....	24
Couche description de service.....	24
Publication de service.....	24
II.2.7.Avantages et Inconvénients .....	24
Avantages.....	24
Inconvénients .....	24
II.3.XML/SOAP .....	24
II.3.1.XML.....	24
A. Introduction .....	24
B. Définition.....	25
C. Un document XML.....	25
D. Base de données XML .....	25
E. Les avantages de XML.....	26
II.3.2.SOAP.....	26
Introduction .....	26
A. Définition.....	26
B. Structure d'un message SOAP .....	27
C. L'enveloppe SOAP .....	28
II.4.Adaptabilité.....	29
II.4.1.Définition .....	30
II.4.2.Appports.....	30
II.4.3.Types de logiciels exploitant cette possibilité.....	30
II.4.4.Tableau récapitulatif .....	31
II.4.5.Interopérabilité .....	31
II.5.Les bases de données réparties .....	32
II.5.1.Définition .....	32
II.5.2.Buts de la répartition des bases de données .....	32

II.5.3.Conception d'une base de données répartie..	32
II.5.4.Architecture des bases de données réparties	33
II.5.5.Sécurité des bases de données réparties	34
II.5.5.1.Exemple de politique de sécurité	35
II.5.5.2.La protection d'une base de données	35
Conclusion	36

## *Chapitre III : E-Learning*

Introduction	37
III.1.Définition de la formation à distance	37
III.1.1.Les avantages et les inconvénients de la formation à distance	37
III.1.1.1.Les avantages de la formation à distance	37
III.1.1.2.Les inconvénients de la formation à distance	38
III.1.2.Le télé-enseignement	38
III.2.E-learning	38
III.2.1.Définition	38
III.2.2.Historique et évolution d'e-learning	39
III.2.2.1.EAO	39
III.2.2.2.EIAO	40
III.2.2.3.De l'EIAO à l'EIAD	41
III.2.2.4.Enseignements à distance ou formations à distance(EAD/FAD)	42
III.2.2.5.Formations ouverte et à distance	42
III.2.3.Typologie d'e-learning	42
III.2.4.Les différentes formules du E-learning	43
III.2.5.Les principales caractéristiques de E-learning	44
III.2.6.Les formes de communication et de collaboration de E-learning	44
III.2.7.Les outils de communication	45
III.3.Plate forme d'E-learning	46
III.3.1.Les acteurs d'une plate forme d'e-learning	47
III.3.1.1.Enseignant	47
III.3.1.2.Apprenant	47
III.3.1.3.Auteur	48
III.3.1.4.Administrateur	48
III.3.2.Exemples de plates formes d'e-learning	48

III.3.2.1.WebCT.....	48
III.3.2.2.Moodle.....	49
III.4.Situation de l'e-learning dans le monde.....	50
III.5.Avantages et faiblesses de l'e-learning.....	51
III.5.1.Avantages de l'e-learning.....	51
III.5.2.Faiblesses de l'e-learning.....	51
Conclusion.....	52

## *Chapitre IV : Elaboration d'un exemple*

Introduction.....	53
La démarche de modélisation de l'application.....	53
IV.1.Analyse.....	53
IV.1.1.Spécification des besoins.....	54
IV.1.2.Identification des acteurs.....	54
IV.1.2.1.Identification des acteurs de chaque département.....	55
IV.1.3.Diagramme de contexte.....	56
IV.1.4.Identification et representation des cas d'utilisation.....	56
IV.1.4.1.Les diagrammes de cas d'utilisation.....	58
IV.1.4.1.1.Définition.....	58
IV.2.Conception.....	62
IV.2.1.Elaboration avec les diagrammes de séquence.....	62
IV.2.2.Présentation de queleques diagrammes de séquences corespondant aux cas d'utilisation déjà décrits.....	63
IV.2.2.1.Cas d'utilisation « s'inscrire à la plate forme ».....	63
IV.2.2.2. Cas d'utilisation « s'authentifier ».....	65
IV.2.2.3. Cas d'utilisation « envoyer un message à etudiant/enseignant/administrateur ».....	66
IV.2.3.Diagramme de classe.....	68
IV.2.3.1.Diagramme de schema logique du site département.....	68
IV.2.3.2. Diagramme de schema logique de base de la plate forme.....	70
IV.2.3.3.Exemple de modèle logique de site département.....	71
Coclusion.....	71

## *Chapitre II : Réalisation et mise en œuvre*

Introduction .....	72
V.1.Architecture de l'application.....	72
V.2.Environnement logiciel et materiel.....	72
V.3.L'environnement de developpement.....	73
V.3.1.Langages de programmation .....	73
V.3.1.1.PHP.....	74
V.3.1.2.Structure d'une page PHP.....	74
V.3.1.3.Fonctionnement du PHP.....	75
V.3.2.Serveur Web Apache.....	76
V.3.3.Serveur de base de données MySQL.....	76
V.3.3.1.MySQL .....	76
V.3.3.2.Fonctionnalités du MySQL.....	77
V.4.Les outils de developpement.....	78
V.4.1.Wamp .....	78
V.4.2.PhpMyAdmin.....	79
V.4.2.1.Les fonctions de PhpMyAdmin.....	79
V.4.2.2.Utilisation de PhpMyAdmin .....	79
V.4.3.Macromedia dreamweaver8.....	80
V.5.Presentation de quelques interfaces de notre plate-forme.....	81
V.5.1.Page principale.....	81
V.5.2.Page d'accueil .....	82
V.5.3.la page de département .....	83
V.5.4.la page de département informatique .....	83
V.5.5.la page de l'administrateur de département .....	84
V.5.6.Espace messagerie .....	84
■ Espace messagerie .....	84
■ Liste des messages envoyés par des apprenants.....	85
■ Espace d'affichage du message .....	85
■ Espace pour envoyer un message.....	86
■ Suppression d'un message.....	86
Conclusion .....	87
Conclusion générale	
Bibliographie.	
Annexe.	

# Introduction générale

Dans la société actuelle, nous avons de plus en plus affaire à l'informatique. L'administration, l'enseignement, les entreprises, les hôpitaux, les citoyens, tous les acteurs de la société utilisent aujourd'hui massivement les systèmes informatiques pour gérer leurs activités, leurs données, leurs employés, leurs clients, et même leur argent. Comme de nouveaux besoins sont apparus avec toute organisation automatisée souhaite stocker et échanger ses informations qui sont géographiquement éloignées, ce qui rend la tâche de la collecte et de traitement et la mise à jour d'une grande quantité d'informations dispersées très difficile, de ce fait, l'amélioration de l'adaptabilité des systèmes d'informations est devenue une priorité pour les gérants des entreprises.

La solution qui s'impose est de distribuer les données et les organiser dans des bases de données sur différents sites de stockage. L'ensemble de ces sites constitue un système de bases de données réparties offrant la possibilité aux utilisateurs de manipuler les différentes bases via un réseau de manière transparente et réutilisable, comme dans une base de données globale. Les bases de données réparties offrent beaucoup d'avantages : une grande extensibilité, car ils peuvent croître progressivement selon le besoin et l'adaptabilité qui permet d'accéder à des nouveaux sites sans effectuer des mises à jours.

## Plan du mémoire

**Le premier chapitre** de ce mémoire s'intéresse à l'aspect technique des services, dans lequel, nous avons présenté des généralités sur les réseaux et Internet, en mettant l'accent sur les protocoles applicatifs sur Internet.

**Le deuxième chapitre** présente l'adaptabilité et les bases de données réparties entre les systèmes, l'architecture orienté service, l'évolution des services web ainsi que les technologies XML et SOAP. Une attention particulière est prêtée à l'adaptabilité et bdd réparties.

**Le troisième chapitre** traite le domaine de l'e-Learning, une attention particulière est prêtée à l'évolution de l'e-Learning partant de l'EAO (l'enseignement assisté par ordinateur) vers l'EIAO (l'enseignement intelligemment assisté par ordinateur) puis vers les EIAD (les environnements interactifs d'apprentissage à distance).

# Introduction générale

**Le quatrième chapitre** aborde en détail la démarche méthodologique suivie pour concevoir un exemple d'une plate forme e-Learning répondant aux exigences énumérées ci-dessus. **Le cinquième chapitre** décrit la mise en œuvre des solutions proposées dans le quatrième chapitre puis procède à la validation de nos propositions.

Pour conclure nous dressons un bilan sur notre travail, sur les résultats obtenus et nous ouvrons des perspectives de recherche.

**En fin de document** figure une annexe qui décrit l'UML et son extension pour le web.

**Introduction :**

L'avènement des réseaux a non seulement permis de relier différents ordinateurs mais aussi de contribuer à l'essor pris par l'échange continu d'information d'un bout à l'autre de la planète grâce au phénomène Internet.

Le présent chapitre comporte en premier lieu un recueil de notions de base sur les réseaux. L'Internet est aussi passé en revue. On retrouve ainsi les différents protocoles applicatifs sur internet.

**I.1. les réseaux : [12]**

Les réseaux ont vu leurs naissances d'un besoin d'échange des informations de manière simple et rapide entre des machines. Lorsque l'on travaillait sur une même machine, toutes les informations nécessaires au travail étaient centralisées sur cette machine. Pour des raisons de coûts ou de performances, on est venu à multiplier le nombre de machines. Les informations devaient alors être dupliquées sur les différentes machines de même site. On est donc arrivé à relier d'abord ces machines entre elles. Ce fut l'apparition des réseaux locaux. Ces réseaux étaient souvent des réseaux propriétaires. Plus tard on a éprouvé le besoin d'échange d'informations entre des sites distants. Les réseaux moyenne et longue distance commencèrent à voir le jour. Aujourd'hui, les réseaux se retrouvent à l'échelle planétaire. Le besoin d'échange de l'information est pleine évolution.

**I.1.1. Définition d'un réseau : [13]**

On peut associer un réseau à un ensemble de nœuds interconnectés par un ensemble d'arcs. La structuration d'un réseau peut être représentée sous forme d'un graphe. La topologie d'un réseau c'est-à-dire la localisation des nœuds et l'agencement des liens entre les nœuds peut être très varié.

**I.1.2. Réseaux informatique :****• Définition : [12]**

Un réseau informatique (en anglais: computer network) est l'ensemble des moyens matériels et logiciels mis en œuvre pour assurer les communications entre Ordinateurs, stations de travail et terminaux informatiques.

**• Objectifs des réseaux informatique : [12]**

Les réseaux permettent :

Le partage de fichiers.

Le transfert de fichiers.

Le partage d'application : compilateur, système de gestion de base de données(SGBD).

Le partage de périphériques.

L'interaction avec les utilisateurs connectés : messagerie électronique, vidéo conférence...

Le transfert de données en général (réseaux informatique).

Le transfert de la parole, de la vidéo et des données (réseaux à intégration de services ou multimédia).

### I.1.3. La classification des réseaux : [12]

On peut globalement classer les réseaux selon plusieurs critères tel que:

- **Classification selon l'étendue géographique :**

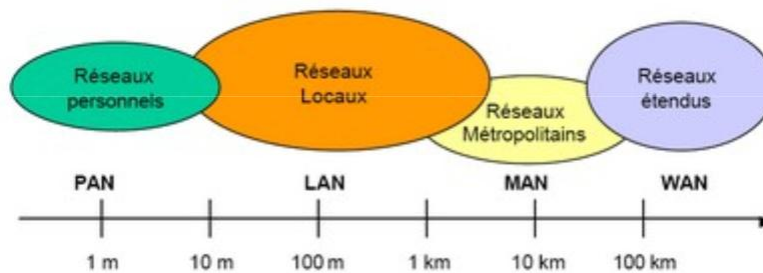


Figure.I.1.classification des réseaux

**A. Réseaux personnel(PAN):** PAN qui signifie « Personnel Area Network », est un réseau qui s'étend sur quelques mètres (connexion 'appareilsd périphérique tels que le clavier, l'imprimante, téléphone,...).

Exemple : Bluetooth, infrarouge.

**B. Réseaux locaux(LAN) :**LAN qui signifie « Local Area Network », est un réseau utilisé généralement par les entreprises pour relie ces différentes unités se trouvant dans le même bâtiment ou dans de différents bâtiments qui ont éloignés d'une centaine de mètres.

Exemple: Ethernet, Token ring, Wi-Fi.

**C. Réseaux métropolitain(MAN):** MAN qui signifie «Métropolitain Area Network», est un réseau dont les nœuds s'étendent sur quelques dizaines de kilomètres. Les fibres optiques sont généralement tiliséesu pour la réalisation

Exemple : Intranet, FDDI, ATM.

**D. Réseaux étendus (WAN):** WAN qui signifie « Wide Area Network », est un réseau couvrant une très grande zone géographique, allant jusqu'à l'échelle d'un pays, voire de la planète entière.

Ces réseaux sont soit terrestre (de grands réseauxde fibre optique), soit hertzien (comme les réseaux satellite).

Exemple : Internet, Satellite.

- **Classification selon le mode de communication :**

- [10] A. Mode diffusion :**

- Les réseaux en bus ou linéaire** il: consiste à relier tous les ordinateurs entre eux par l'intermédiaire de câble coaxial. Le mot « bus » désigne la ligne physique qui relie les machines du réseau.

- Les réseaux en anneau :** Dans un réseau possédant une topologie en anneau, les ordinateurs sont situés sur une boucle et communiquent chacun à leur tour.

- B. Mode point à point :**

- Les réseaux en étoile :** Cette topologie est basée sur un nœud d'interconnexion de tous les ordinateurs. Cette technique suppose qu'en plus des cartes réseaux, il faille un distributeur central : le hub.

- Les réseaux maillés :** est une évolution de la topologie en étoile, elle correspond à plusieurs liaisons point à point. Une unité réseau peut avoir (1,N) connexions point à point vers plusieurs unités.

## **I.2. Internet :**

Il est presque impossible de cerner une telle question, car c'est une technologie en constante évolution. Pour cela, on dira que Internet est un gigantesque réseau mondial d'ordinateurs qui englobe plusieurs autres réseaux de moindre taille. Ces derniers permettent eux-mêmes d'interconnecter les ordinateurs individuels.

### **I.2.1. Définition : [12]**

Le mot Internet vient du terme anglais « Inter-Network ». C'est un réseau international d'ordinateurs, ou plus précisément le réseau des réseaux d'ordinateurs qui communiquent entre eux grâce à un protocole d'échange de données standard (TCP/IP). Aujourd'hui Internet relie tous ceux qui ont un ordinateur. En bref, des dizaines de millions d'individus, de gouvernements ou d'organismes.

### **I.2.2. Les services d'Internet : [10]**

Internet a démarré avec des applications client/serveur simples, comme la messagerie électronique et transfert des fichiers, pour offrir aujourd'hui des services extrêmes complexes dans lequel on peut naviguer pour rechercher une formation précise.

- **E-mail (la messagerie électronique) :** permet aux usagers d'Internet d'échanger des messages écrits de manière efficace et rapide via des boîtes électroniques.
- **IRC (Internet Relay Chat) :** c'est une forme de communication interactive entre un individu et un autre par envoi de textes écrits.

- **New** : à l'inverse de mail ou la discussion set réalisée de l'émetteur vers le récepteur, les forums de discussion sont des moyens de discussion entre plusieurs personnes.
- **TELNET (Accès à des services à distance)** : grâce à FTP on peut transférer tout type de fichier (image, texte, logiciel, ...etc) d'une machine à une autre. Comme il permet à un usager d'accéder à un disque dur monté sur un ordinateur distant.
- **USENET (Le forums électronique)** : permet d'accéder à des groupes de discussions, sur lesquels les utilisateurs peuvent se trouver pour échanger des idées sur différents thèmes.
- **CHAT** : un « chat », appelé aussi tchat, est un espace permettant une « discussion textuelle » (par écrit) en temps réel entre plusieurs internautes.
- **WAIS (Wide Area Information Servers)** : C'est un service de recherche de données de différents formats (textes, images, vidéos, et sons) à l'aide de mot clé.
- **GOPHER** : permettent de retrouver de documents, des programmes et d'autres ressources sur internet en mode texte. La recherche se fait par choix dans le menu de ce système, ainsi la recherche permet au bout d'atteindre une ressource spécifique.
- **WWW (World Wide Web)** : c'est la partie qui marque le plus de succès et la renommée d'internet grâce à son aspect graphique et interactif.

### I.2.3. Fonctionnement : [10]

L'utilisateur peut se connecter à Internet en utilisant la méthode filaire ou bien sans fil à l'aide des dernières technologies comme le WIFI ( Wireless Fidelity),...

Dans le premier cas, l'accès se fait par le réseau téléphonique. Il faut un modem entre ce dernier et l'ordinateur qui permet de mettre en forme les signaux informatiques qui vont circuler sur les fils.

Dans le deuxième cas, l'accès se fait par login et un mot de passe.

Pour communiquer entre utilisateurs, les ordinateurs connectés à l'internet doivent parler la même langue en utilisant le protocole de communication TCP/IP (Transfert control Protocol /Internet Protocol). Son rôle est de définir la structure des paquets, l'adressage et la façon dont les paquets sont « routés » sur le réseau.

L'acheminement de l'information selon le protocole TCP/IP peut se résumer ainsi :

- L'information voyage sous forme d'octets (eux-mêmes formés de 0 et de 1) et par paquets à travers des fils (en cuivre ou en fibre optique,...)
- Chaque machine (serveur ou client) connectées sur Internet est identifiée par une adresse appelée IP, formée de chiffres, telle que 651.85.235.45. Cette adresse est formée également d'un nom en toutes lettres plus facile à retenir qu'une suite de chiffre, tel que : **http://www.google.com**.

Internet est répandu à travers le monde grâce aux répéteurs, ponts, routeurs, qui garantissent le transport des paquets d'octets de la source jusqu'au destinataire.

### I.2.4. L'intranet et l'extranet : [10]

#### A. L'intranet :

- **Définition : [10]**

Est un réseau informatique local c'est-à-dire utilise uniquement à l'intérieure d'une entreprise ou organisation quelconque.

Il consiste à utiliser les standards clients/serveurs de l'internet, tout comme l'utilisation des navigateurs internet (Mozilla, Opéra, Internet explorer,...etc) et des serveurs web (protocole http) pour réaliser un système d'information interne à une organisation ou une entreprise.

- **Les avantages d'un intranet :**

Technologie indépendante de tout constructeur et de toute la plateforme.

Coûts modérés pour l'acquisition de matériel et de logiciels pour les serveurs web.

Navigateur comme interface utilisateur unique.

Efforts réduits pour la mise à jour des logiciels.

- **Les propriétés d'un intranet :**

L'interconnexions des systèmes d'information d'une entreprise.

L'interconnexions sécurisé des réseaux augmentent, les pare-feu (firewalls) répondent à cette demande, en permettant à une entreprise d'isoler son réseau et ces ordinateurs de l'internet à une qualité limitée de données et de services.

Le serveur web de l'entreprise accessible aux employés et partiellement à des tiers externes.

#### B. L'extranet :

- **Définition : [10]**

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

L'accès à l'extranet doit être sécurisé dans la mesure où cela offre un accès au système d'information à des personnes situées en dehors de l'entreprise. Il peut s'agir soit d'une authentification simple (authentification par nom d'utilisateur et mot de passe) ou d'une authentification forte (authentification à l'aide d'un certificat). Il est conseillé d'utiliser **HTTPS** pour toutes les pages web consultées depuis l'extérieur afin de sécuriser le transport des requêtes et des réponses **HTTP** et d'éviter notamment la circulation du mot de passe en clair sur le réseau.

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

**I.3. les protocoles :****I.3.1. Définition : [13]**

Un protocole est une méthode standard qui permet la communication entre des processus (S'exécutant éventuellement sur différentes machines), c'est-à-dire un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau. Les protocoles sont classés par rapport à leurs fonctionnalités : le transport de fichiers (FTP), le contrôle de données (ICMP), etc...

**I.3.2. Quelques protocoles : [11]****A. Les protocoles de communications : [11]**

Sur Internet, les protocoles utilisés font partie d'une suite de protocoles, c'est-à-dire un ensemble de protocoles reliés entre eux. Cette suite de protocoles s'appelle TCP/IP. Le sigle TCP/IP signifie «Transmission Control Protocol/Internet Protocol».

TCP/IP représente d'une certaine façon l'ensemble des règles de communication sur Internet et se base sur la notion d'adressage IP, c'est-à-dire le fait de fournir une adresse IP à chaque machine du réseau afin de pouvoir acheminer des paquets de données. Etant donné que la suite de protocoles TCP/IP a été créée à l'origine dans un but militaire, elle est conçue pour répondre à un certain nombre de critères parmi lesquels :

- Le fractionnement des messages en paquets ;
- L'utilisation d'un système d'adresses ;
- L'acheminement des données sur le réseau (routage);
- Le contrôle des erreurs de transmission de données.

Elle contient, entre autre les protocoles suivants : HTTP, FTP, SMTP...

**A.1. Protocole HTTP : [11]**

Le protocole HTTP (HyperTexte transfert Protocol) est le protocole défini pour le web au dessus de TCP/IP. Il désigne un protocole au niveau application.

Il est destiné à transférer des données et à répondre au client en temps utile. C'est pour cela qu'une connexion est fermée à chaque fois qu'une requête a été satisfaite.

Le but du protocole http est de permettre un transfert de fichiers (essentiellement au format HTML) localisé grâce à une chaîne de caractère appelée URL (Uniform Resource Locator) entre un navigateur (le client) et un serveur web.

Les messages échangés du client vers le serveur permettent de réaliser les fonctions GET, POST, HEAD, PUT, DELETE, TRACE et OPTIONS. Les plus utilisés sont : GET, POST, HEAD :

**La méthode GET :** est la plus utilisée, elle permet aux clients de demander un document HyperText, en spécifiant sa localisation physique à l'aide d'une URL.

**La méthode POST :** primordiale pour le traitement des formulaires.

**La méthode HEAD :** permet d'offrir l'information sur le web, mais sans avoir besoin du document lui-même.

**La méthode DELETE :** permet d'effacer une URL.

**La méthode TRACE :** permet de voir ce que devient la requête du client à travers un serveur mandataire (proxy).

**La méthode OPTION :** permet de demander au serveur les options disponibles pour une URL donnée.

### **A.2. Le protocole TCP : [11]**

Le protocole TCP (Transmission Control Protocol) est l'un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle. Lorsque les données sont fournies au protocole IP, celui-ci les encapsule dans des datagrammes IP.

TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui se communiquent de contrôler l'état de la transmission.

### **A.3. Le protocole IP : [11]**

Le protocole IP (Internet Protocole) transfère des données en mode datagramme, c'est-à-dire que les paquets sont traités indépendamment les uns des autres. Ceci permet d'avoir des équipements relativement simples et rapides à l'intérieur du réseau.

### **A.4. Le protocole FTP : [11]**

Le FTP (File Transfer Protocole) est le protocole de transfert de fichier qui permet de télécharger une copie de données choisies par l'internaute, d'un ordinateur à un autre selon le modèle client/serveur. Les données téléchargées peuvent prendre la forme de logiciel, de fichier de toute nature (texte, graphique, images, sons ... etc.).

### **A.5. Le protocole ICMP : [11]**

Le protocole ICMP (Internet Control Message Protocole) organise un échange d'information permettant aux routeurs d'envoyer des messages d'erreurs à d'autres ordinateurs ou routeurs.

Etant donné le peu de contrôles que le protocole IP réalise, il permet non pas de corriger ces erreurs mais de faire part de ces erreurs aux protocoles des couches voisines. Ainsi, le protocole ICMP est utilisé par tous les routeurs pour signaler une erreur (appelé Delivery problem).

### **A.6. Le protocole ARP : [11]**

Le protocole ARP (Address Resolution Protocol) a un rôle très important parmi les protocoles de la couche Internet du modèle TCP/IP. Il permet de connaître l'adresse physique d'une carte réseau correspondante à une adresse IP, c'est pour cela qu'il s'appelle protocole de résolution d'adresse. Ainsi, pour faire correspondre les adresses IP aux adresses logiques, le protocole ARP interroge les machines du réseau pour connaître leurs adresses physiques, puis crée une table de correspondance entre les adresses logiques et les adresses physiques dans une mémoire cache.

### **A.7. Le protocole RARP :**

Le protocole RARP (Reverse Address Resolution Protocol) est beaucoup moins utilisé. De façon inverse, une machine qui se connecte sur le réseau peut connaître sa propre adresse physique et cherche son adresse IP. Au moment de son initialisation, cette machine doit contacter son serveur afin de déterminer son adresse IP.

Dans ce cas le protocole RARP permet à la machine d'utiliser son adresse logique sur Internet.

**A.8. Le protocole TELNET : [11]**

TELNET est un protocole standard permettant l'interfaçage de terminaux et d'applications à travers Internet. Ce protocole fournit les règles de base pour permettre de relier un client (Système composé d'un affichage et d'un clavier) à un ordinateur distant (coté serveur).

**A.9. Le protocole SMTP (Simple Mail Transfert Protocol) : [11]**

Le courrier électronique au sein d'Internet est géré par le protocole SMTP bâtis sur TCP. Il permet d'échanger des messages entre un expéditeur et un ou plusieurs destinataires pourvus que leurs adresses soient connues.

**B. DNS : [12]**

Le système **DNS** (Domain Name System) permet d'associer des noms symboliques à des adresses numériques. Comme les adresses IP, les noms symboliques sont **structurés et hiérarchiques**.

- Une partie désigne le **nom de la machine** (hostname).
- L'autre partie désigne le **nom de domaine** (domain name) auquel la machine appartient.

Dans chaque domaine, un serveur de nom ou serveur DNS est chargé de répondre aux requêtes des clients (les clients internes comme les clients externes au domaine). Le système DNS s'appuie sur le protocole de transport UDP (port 53).

Le rôle de DNS est donc de faire une équivalence entre un nom de machine et son adresse IP.

Pour ce faire, on utilise :

- Un système de nommage des ordinateurs qui est normalisé et hiérarchisé de manière à ce que chaque ordinateur de l'Internet porte un nom unique.
- Des serveurs de noms DNS qui comportent dans des bases de données le nom des ordinateurs et l'adresse IP correspondante.

**C. LDAP : [10]****C.1. Définition :**

LDAP est un protocole standard permettant de gérer des annuaires, c'est-à-dire d'accéder à des bases d'informations sur les utilisateurs d'un réseau par l'intermédiaire de protocoles TCP/IP.

Le protocole LDAP définit la méthode d'accès aux données sur le serveur au niveau du client, et non la manière de laquelle les informations sont stockées.

Ainsi LDAP fournit à l'utilisateur des méthodes lui permettant de :

- se connecter.
- se déconnecter.
- rechercher des informations.
- comparer des informations.

- insérer des entrées.
- modifier des entrées.
- supprimer des entrées.

D'autre part le protocole LDAP propose des mécanismes de chiffrement (SSL, ...) et d'authentification (SASL) permettant de sécuriser l'accès aux informations stockées dans la base.

### **C.2. Les opérations de LDAP :**

<b>Opérations</b>	<b>Description</b>
Abandon	Abandonne l'opération précédemment envoyées au serveur
Add	Ajoute une entrée au répertoire
Bind	Initie une nouvelle session sur le serveur LDAP
Compare	Compare les entrées d'un répertoire selon des critères
Delete	Supprime une entrée d'un répertoire
Extended	Effectue des opérations étendues
Rename	Modifie le nom d'une entrée
Search	Recherche des entrées d'un répertoire
Unbind	Termine une session sur le serveur LDAP

### **C.3. Pourquoi utiliser LDAP ?**

Le principal avantage du protocole LDAP réside dans la possibilité de réunir les informations concernant toute une organisation dans un lieu central. Par exemple, plutôt que de gérer des listes d'utilisateurs pour chaque groupe au sein d'une organisation, LDAP peut être utilisé comme un répertoire central accessible sur tout le réseau. De plus, puisque LDAP prend en charge les fonctions Secure Sockets Layer (SSL) et Transport Layer Security (TLS), des données confidentielles peuvent être protégées contre toute intrusion.

LDAP prend aussi en charge diverses bases de données parallèles pour y enregistrer des répertoires. Ainsi, les administrateurs disposent de la flexibilité nécessaire pour déployer la base de données la plus adaptée au type d'informations que le serveur doit disséminer. De plus, comme LDAP comporte une interface de programmation d'application (ou API de l'anglais Application Programming Interfaces) bien définie, le nombre d'applications compatibles avec LDAP est vaste et croissant aussi bien en quantité qu'en qualité.

**D. CORBA :****D.1. Définition du CORBA : [12]**

CORBA est un standard d'objets distribués. CORBA permet à une application de demander à un objet distribué d'effectuer une opération et d'en obtenir le résultat. L'application communique avec l'objet qui effectue l'opération. C'est une fonctionnalité clients/serveur basique où un client envoie une requête à un serveur et le serveur répond au client. Les données peuvent passer du client au serveur et sont associées à une opération particulière sur un objet particulier. Les données sont ensuite retournées au client sous la forme d'une réponse.

**D.2. Avantages du CORBA : [12]**

- CORBA supporte plusieurs langages. CORBA supporte aussi un mélange de ces langages dans une même application distribuée.
- CORBA supporte à la fois la distribution et l'orientation objets.
- CORBA est un standard industriel. Cela crée une concurrence entre les différents vendeurs et assure ainsi que des implémentations de qualité existent.
- Le standard CORBA permet aussi au développeur de pouvoir compter sur un degré élevé de compatibilité entre les différentes implémentations (sans jamais être totalement portable).
- CORBA permet un degré élevé d'interopérabilité. Cela permet d'assurer que des objets distribués construits sur différents produits CORBA pourront communiquer.
- Plus de 600 compagnies soutiennent CORBA.

**E. Le modèle TCP/IP (modèle en couches) : [11]**

Afin de pouvoir appliquer le modèle TCP/IP à n'importe quelle machine, c'est-à-dire indépendamment du système d'exploitation, le système de protocoles TCP/IP a été décomposé en plusieurs modules effectuant chacun une tâche précise. De plus, ces modules effectuent ces tâches les uns après les autres dans un ordre précis, c'est la raison pour laquelle on parle du modèle en couches.

Le terme de couche est utilisé pour évoquer le fait que les données qui transitent sur le réseau traversent plusieurs niveaux de protocoles. Ainsi, les données (paquets d'informations) qui circulent sur le réseau sont traitées successivement par chaque couche qui rajoute un élément d'information (appelé en-tête) puis sont transmises à la couche suivante.

Le but d'un système en couches est de séparer le problème en différentes parties (les couches) selon leur niveau d'abstraction.

Chaque couche du modèle communique avec une couche adjacente (celle du dessus ou celle du dessous). Chaque couche utilise ainsi les services des couches inférieures et en fournit à celle de niveau supérieur.

Le modèle TCP/IP est très proche du modèle OSI (Open Systems Interconnection) qui a été mis au point par l'organisation internationale des standards ISO (International Standard Organization) afin de normaliser les communications entre ordinateurs.

### **E.1. Présentation du modèle OSI : [11]**

OSI signifie Open Systems Interconnection, ce qui se traduit par Interconnexion de Systèmes Ouverts (ISO). Ce modèle a été mis en place par l'ISO afin de mettre en place un standard de communication entre les ordinateurs d'un réseau, c'est-à-dire les règles qui gèrent les communications entre les ordinateurs. En effet, aux origines des réseaux, chaque constructeur avait un système propre (on parle de système propriétaire). Ainsi de nombreux réseaux incompatibles coexistaient. C'est la raison pour laquelle l'établissement d'une norme a été nécessaire.

Le rôle du modèle OSI consiste à standardiser la communication entre les machines afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles .

Ce modèle comporte sept couches :

#### **La couche physique :**

La couche physique rassemble les moyens électriques, mécaniques, optiques ou hertziens par lesquels les informations sont transmises. Les unités de données sont des bits 0 ou 1.

#### **La couche liaison donnée:**

La couche liaison gère la fiabilité du transfert de bits d'un nœud à l'autre du réseau, comprenant entre autres les dispositifs de détection et de correction d'erreurs. L'unité de données à ce niveau est appelée trame.

#### **La couche réseau:**

Permet de détecter, de corriger les erreurs non réglées par la couche liaison et de gérer les adresses et le routage des données.

#### **La couche transport :**

Elle est chargée du transport des données et de la gestion des erreurs.

#### **La couche session :**

Permet l'ouverture et la fermeture des sessions sur les machines du réseau.

#### **La couche présentation :**

Définit le format des données.

#### **La couche application :**

Elle fournit des services utilisables par les applications installées sur le réseau.

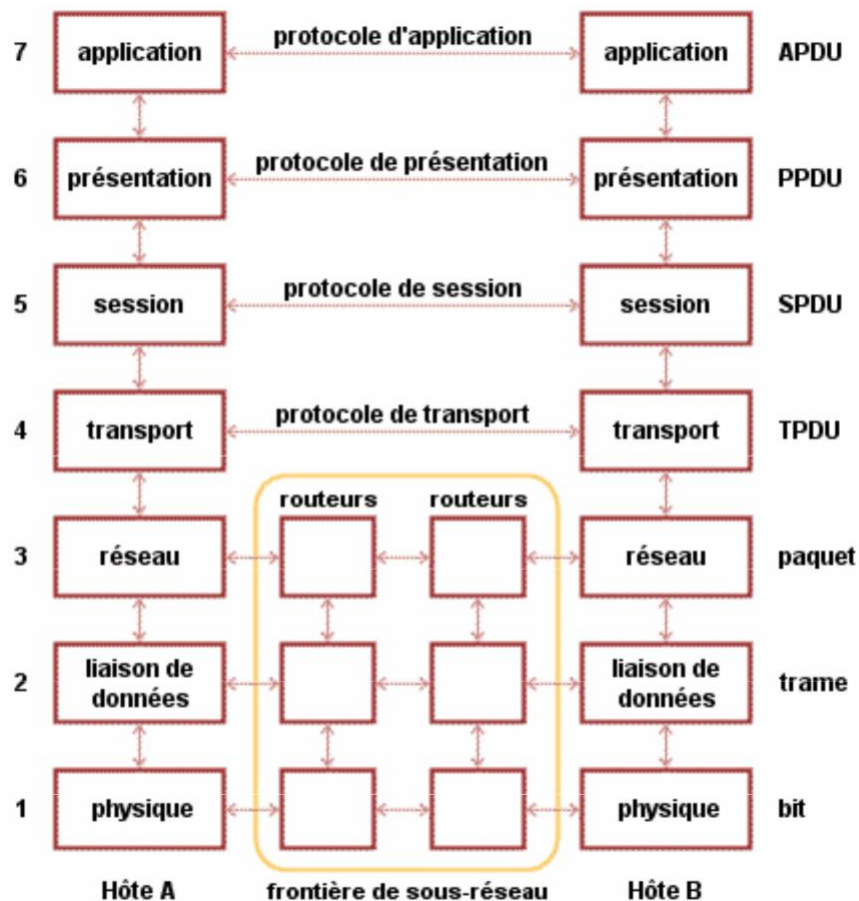


Figure I.2. Présentation du model OSI

## E.2. Présentation du model TCP/IP : [11]

Le modèle TCP/IP, inspiré du modèle OSI, reprend l'approche modulaire (utilisation de modules ou couches) mais contient uniquement quatre :

**Couche accès réseau :** spécifie la forme sous laquelle les données doivent être acheminées quel que soit le type de réseau utilisé.

**Couche transport :** elle assure l'acheminement des données, ainsi que les mécanismes permettant de connaître l'état de la transmission.

**Couche internet :** gère la circulation des paquets à travers le réseau en assurant leurs routages.

**Couche application :** elle englobe les applications standards du réseau (TELNET, SMTP, FTP,....).

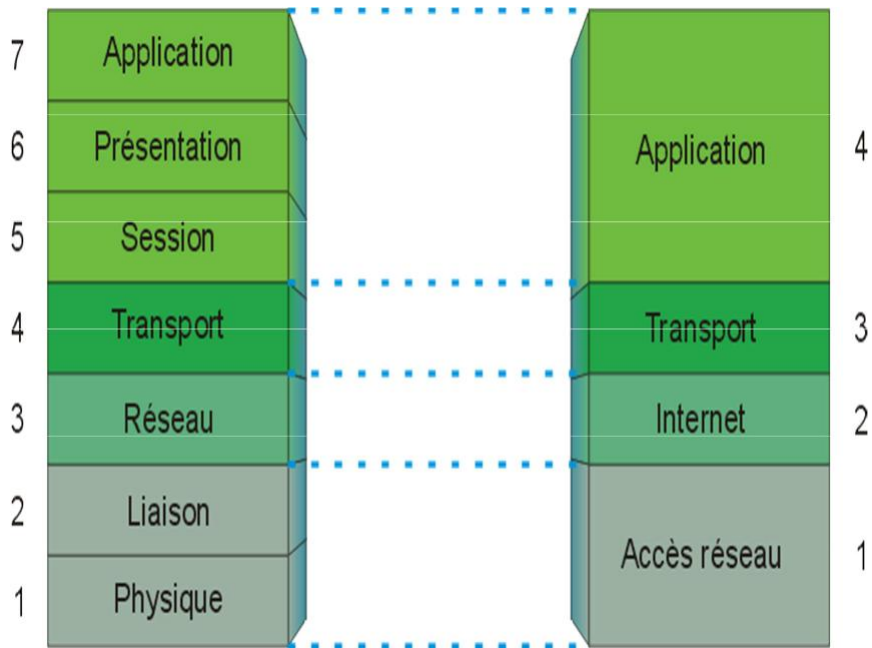


Figure.I.3.Les principaux protocoles rencontrés sur un réseau TCP/IP

**Conclusion :**

Nous avons vu dans ce chapitre certains concepts des nouvelles technologies de l'information et de la communication à savoir : les réseaux informatiques, les différents services offerts par Internet, ainsi les différents protocoles applicatifs sur Internet. Dans le chapitre suivant nous évoquerons l'adaptabilité et les bases de données réparties.

### Introduction :

Les Web Service ont un type d'architecture reposant sur les standards de l'internet. De ce fait Il s'agit d'une architecture orientée service permettant à des applications de communiquer entre elle directement sans se préoccuper des technologies sur lesquelles elles sont implémentées. Les Web Service se présentent comme étant la solution pour répondre à l'interopérabilité des systèmes.

### II.1. Architecture orientée service:

#### II.1.1.Définition: [2]

L'architecture orientée services (AOS) est le terme utilisé pour désigner un *modèle* d'architecture pour l'exécution d'applications logicielles réparties. Elle s'appuyant sur un ensemble de services simples. (SOA) s'est imposée aujourd'hui comme un thème majeur pour les systèmes d'information d'entreprise. Plus qu'une nouvelle technologie ou méthode, c'est la convergence de plusieurs approches existantes, et l'émergence d'un style d'architecture et de gouvernance de SI.

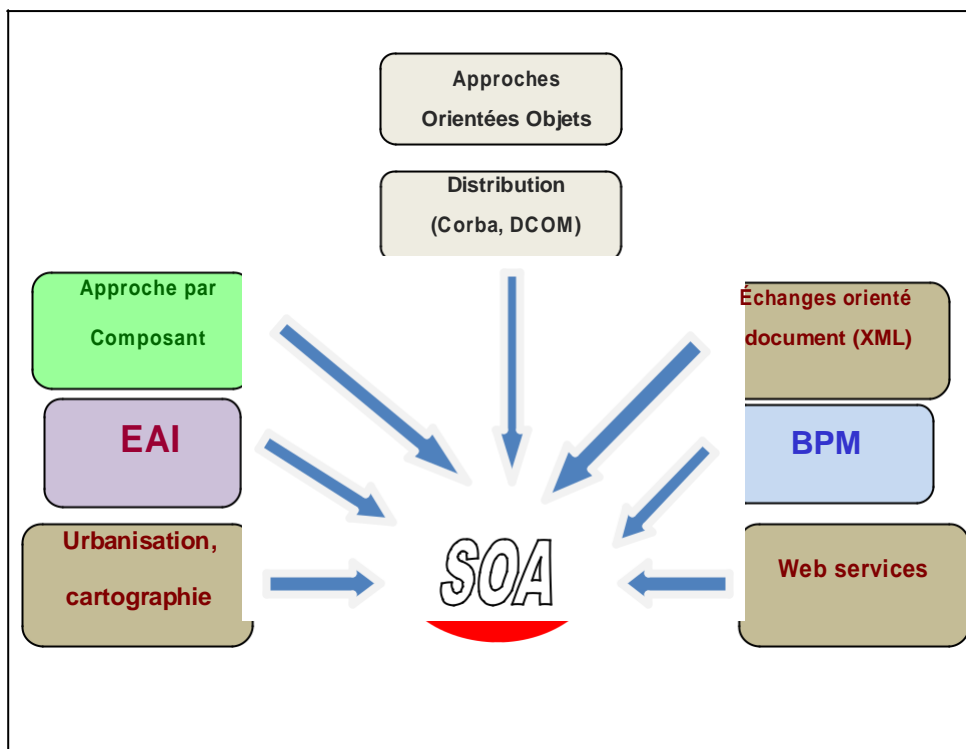


Figure.II.1.SOA : à la convergence de solutions

Dans cette optique, l'architecture occupe une place centrale. Instrument privilégié pour la construction et la maintenance du système, pivot sur lequel s'articulent le métier et sa traduction logicielle, elle constitue la référence pour l'organisation des projets, la construction technique et le plan de progression. Toutefois, l'architecture n'est pas un modèle abstrait, ni une cartographie fonctionnelle cible lointaine. Il s'agit plus pragmatiquement de la description des constituants du système et de leurs relations.

### II.1.2. Les concepts de SOA : [3]

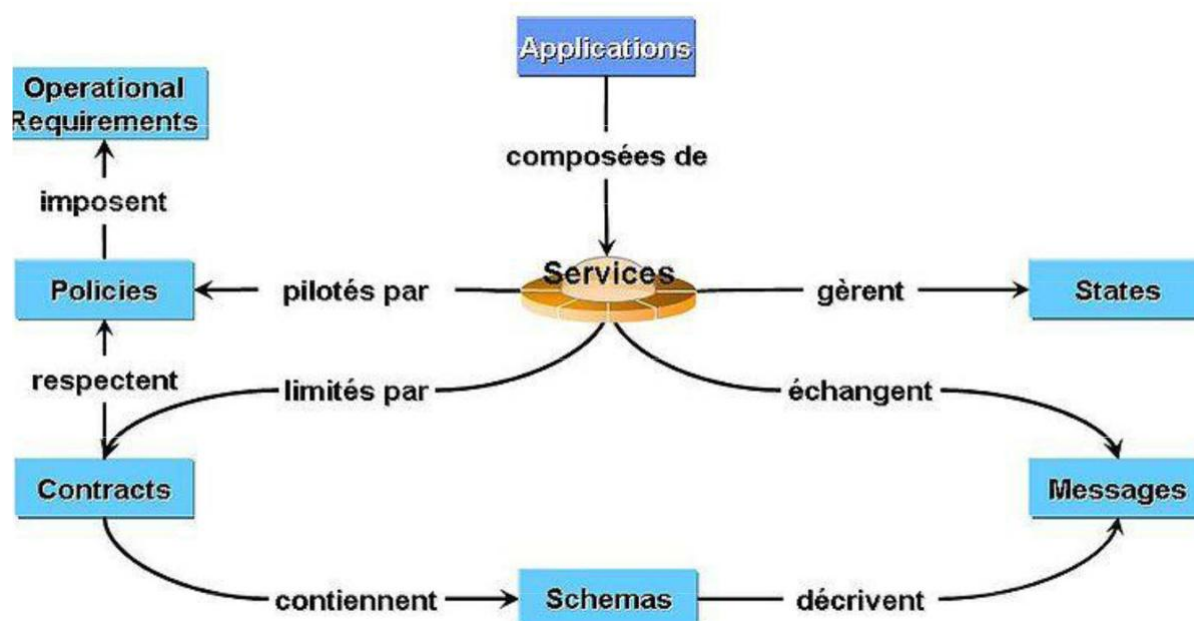


Figure.II.2.schéma montrant le fonctionnement des oncepts de SOA.

Une architecture orientée service se conforme à divers principes de gestion des services influençant directement le comportement intrinsèque d'une solution logicielle et le style de sa conception :

- L'encapsulation des services.
- Le faible couplage des services avec la maintenance d'une relation réduisant les dépendances.
- Le contrat de service adhérent à un accord de communication, collectivement défini avec un ou plusieurs documents de description.
- L'abstraction des services dissimulant la logique du service à l'extérieur.
- La réutilisation des services partageant la logique entre plusieurs services avec l'intention de promouvoir la réutilisation.
- La composition des services.
- L'autonomie des services.

- L'optimisation des services.
- La découverte des services depuis leur description extérieure.

L'architecture orientée service est une solution logicielle distribuée. Elle propose un mécanisme d'échange de messages sécurisé entre les systèmes d'informations.

### II.1.2.1. Définition du concept « service » :[1]

Les services au centre de ce schéma sont les unités atomiques d'une architecture orientée services. Ces services communiquent entre eux par le biais de messages. Ces communications peuvent être synchrones ou asynchrones. Le langage de programmation du service n'a aucune importance, ainsi que la plateforme d'exécution, matérielle et logicielle. Ces services sont limités par un ensemble de contrats, c'est-à-dire qu'un service doit respecter un ensemble de règles de fonctionnement.

Dans les WSOA (*WebService Oriented Architecture*), les *WebServices* sont des services disponibles via Internet par le biais d'URL. Pour les WSOA, les données sont stockées sur le serveur. On peut se connecter aux *WebServices* à l'aide de terminaux mobiles. La connexion à distance est réalisée par protocole http, on a donc une indépendance de tout langage de programmation et plateformes. La communication entre les différents acteurs du système est réalisée par la technologie XML. Le schéma suivant montre le fonctionnement d'un WSOA entre différents terminaux et les *WebServices*.

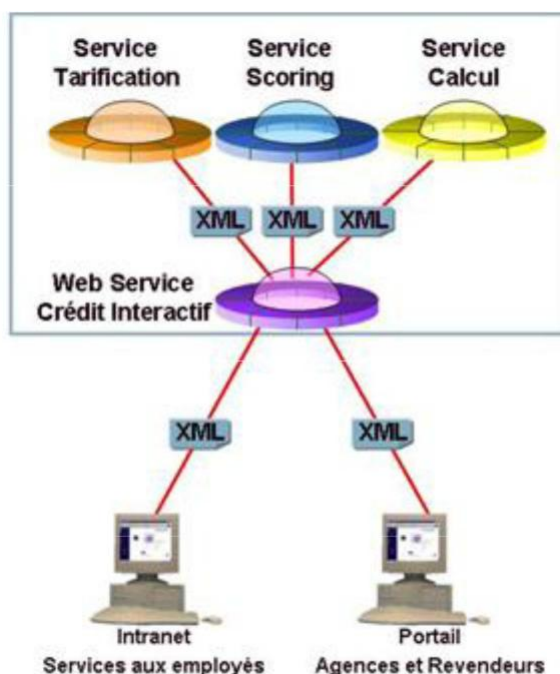


Figure.II.3.fonctionnement d'un WSOA entre différents terminaux et les WebServices.

II.1.3. Les éléments de base de l'architecture orientée service :[2]

II.1.3.1 Composant de service : [2]

Le *composant de service* est la brique de base de l'architecture. Il se décompose en deux parties : *la vue externe* (ou spécification de service), qui expose la facette service proprement dite, et la *vue interne*, qui décrit le contenu du composant. La vue externe est constituée par un ensemble d'opérations de service regroupées en interfaces (au sens UML).

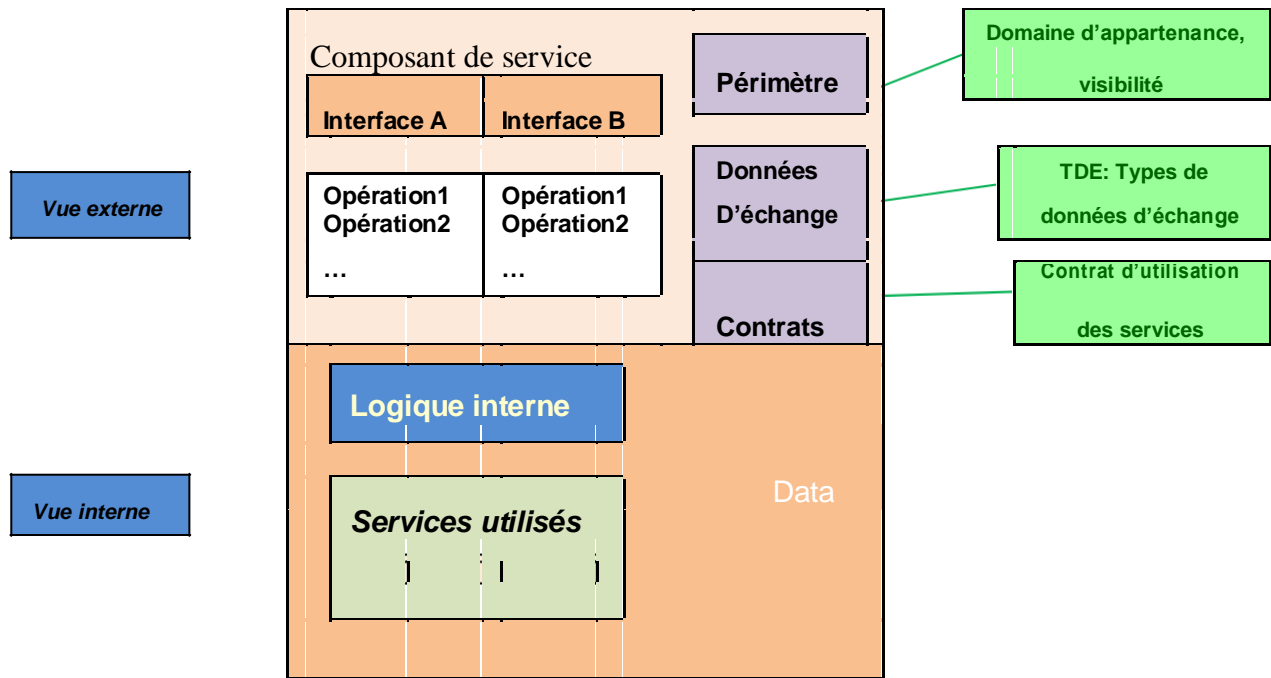


Figure.II.4. Structure du composant de service

La *vue interne* contient des informations relatives à la logique interne comme le détail des traitements ou les bases de données manipulées. On y trouve également les références vers les services utilisés par le composant. Cette vue est masquée aux consommateurs du composant de service. Elle est employée notamment par les architectes SI, qui travaillent sur la vision globale du système.

II.1.3.2 Bus d'entreprise : [2]

Le système est constitué d'un ensemble de *participants* communiquant qui jouent le rôle de *consommateur* et de *fournisseur* de service. Les consommateurs de service peuvent être divers : des applications, logiciels ou d'autres composants de service.

Les composants de service sont les fournisseurs de service du système, dans lesquels on distingue deux familles: les composants qui prennent en charge directement l'implémentation des services et ceux qui délèguent cette implémentation à un tiers (mainframe, ERP, application existante). Il faut néanmoins préciser que dans ce cas de figure, le composant n'est pas toujours assimilé à un simple passe-plat.

Des traitements d'adaptation sont fréquemment nécessaires (format des données, encapsulation de service) afin de mieux intégrer les besoins des consommateurs de service.

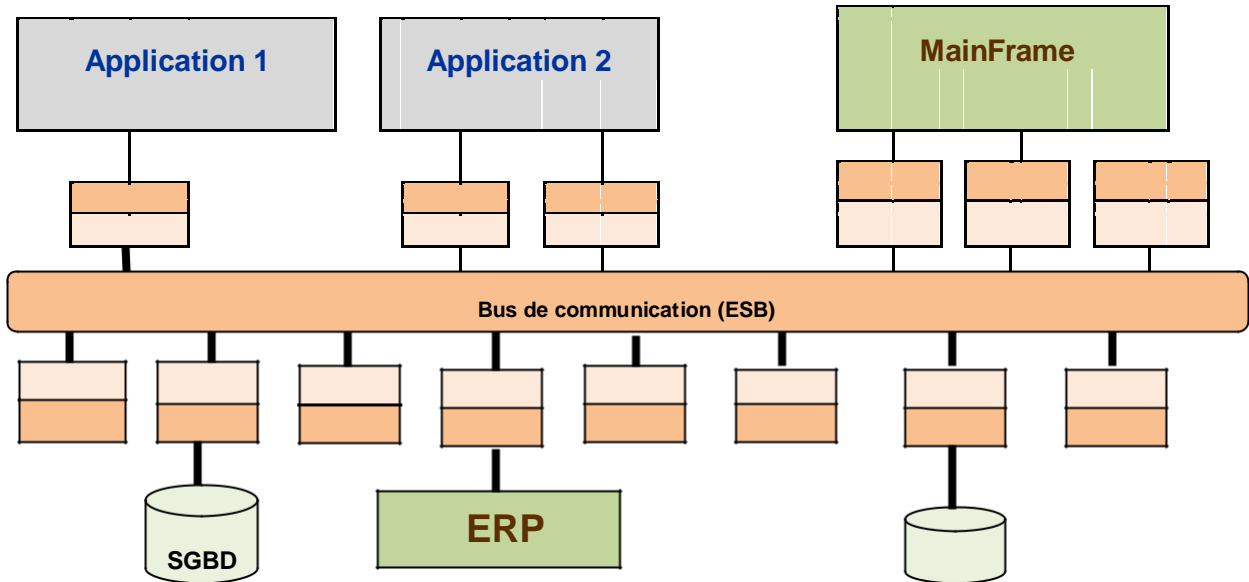


Figure.II.5. ESB et Composants de service

Le bus d'entreprise (ESB) agit comme la colonne vertébrale reliant ces participants d'une manière banalisée à travers les interfaces de services. Cela permet notamment les modifications d'implémentation sans remanier la structure de fonctionnement du système.

**II.1.3.3 Contrat de service : [2]**

Le contrat de service joue un rôle majeur : il détaille les conditions d'utilisation du service sous forme de pré et post conditions, protocoles, et contraintes

Type de contraintes	Exemples
Disponibilité	Taux d'indisponibilité par an, Plages horaires Contraintes de maintenance.
Performance	Temps de réponse moyen, minimum Nombre de transactions par seconde
Fiabilité	Taux d'erreur
Sécurité	Politique de droits d'accès, Non répudiation
Administration	Journalisation, Tableaux de bord, statistiques

Tableau.II. 1.Contraintes non fonctionnelles

**II.1.3.4 Données d'échanges et données persistantes : [2]**

La distinction entre les données d'échange et les données persistantes est inhérente aux architectures SOA, qui isolent les bases de données à l'aide de services d'accès. **Les données d'échange** sont les informations véhiculées entre les participants travers l'invocation des opérations de service.

**Les données persistantes** sont les informations contenues et gérées dans les bases de données. Ces informations sont structurées de façon habituelle (par exemple SGBD).

### II.1.4. Déploiement des SOA : [1]

Le déploiement des SOA peut être effectué sur des serveurs d'intégrations, EAI (*Enterprise Application Integration*). Ces serveurs organisent la circulation de l'information entre les applications et les rendent interopérables en développant des connecteurs, middleware, permettant d'interfacer ces applications utilisant des protocoles de communications différents généralement propriétaire. L'EAI permet donc la connexion aux applications, les conversions des informations dans un langage commun et le transport des informations d'une application vers une autre.

L'EAI permet un accès universel et un partage de toutes les données et composants d'un système d'information, qu'ils soient normalisés, propriétaires ou incompatibles. Il nécessite peu ou pas de modifications des applications ou structures de données qu'il intègre. Une architecture EAI n'est pas figée. Elle constitue un socle évolutif, réutilisable et dynamique suivant l'évolution des besoins métier spécifiques comme les évolutions structurelles de l'entreprise.

Cependant, les plateformes J2EE et .Net répondent aussi aux besoins des systèmes informatiques distribués et services Web.

### II.1.5. Les apports des Architectures Orientées Services : [1]

#### II.1.5.1. Inconvénients:

La mise en place d'un SOA a un coût élevé à la fois financier et humain.

Il faut former une équipe d'experts en conceptions ainsi que plusieurs équipes pour développer et administrer les différents services .

Si le fonctionnement de l'entreprise n'est pas organisé autour des services alors il est difficile d'utiliser un SOA et donc le coût de fonctionnement sera élevé.

En effet, les SOA ont un intérêt limité si l'entreprise ne base pas ses processus sur l'exploitation des services, il faut donc concevoir des *workflows* adaptés.

#### II.1.5.2. Avantages

les SOA disposent de tous les avantages d'une architecture client-serveur (la modularité, on peut facilement remplacer un service par un autre).

une plus grande tolérance aux pannes ainsi qu'une maintenance plus facile.

Il est très facile dans les SOA d'ajouter des nouveaux services sans venir perturber les existants ainsi que de les faire évaluer.

Ces services sont réutilisables, il est donc possible de bâtir de nouvelles applications faisant appel à des services tiers. Par exemple, un ERP d'entreprise peut être créé en faisant appel aux services SAP. Les services sont utilisés selon les besoins des entreprises et la mise à jour de ces services est réalisée de manière totalement transparente pour les consommateurs.

Les SOA ont une architecture n-tiers, c'est-à-dire la séparation de la présentation des données du traitement et de la base de données. Pour une architecture client/serveur, la présentation des données ainsi que le traitement sont réalisés par le client et le serveur utilisé pour la base de données. Par contre, en n-tiers, chaque serveur peut avoir une fonction unique. Par exemple, si on prend une architecture 3-tiers, on a les terminaux pour les requêtes http qui gèrent la partie présentation de l'interface homme-machine. On a un serveur d'application pour gérer les traitements et un dernier serveur de données qui gère les accès aux données.

## **II.2. Service web : [4]**

### **Introduction :**

La technologie des services Web est un moyen rapide de distribution de l'information entre clients, fournisseurs, partenaires commerciaux et leurs différentes plates-formes. Les services Web sont basés sur le modèle SOA.

#### **II.2.1. Définition :[4]**

Une technologie permettant à des applications de dialoguer à distance via Internet indépendamment des plates-formes et des langages sur lesquels elles reposent.

En d'autres termes, un service Web est tout simplement un programme accessible au moyen d'Internet, qui utilise un système de messagerie standard XML, et n'est lié à aucun système d'exploitation ou langage de programmation .

#### **II.2.2. L'intérêt d'un Service Web[4]:**

Les services Web fournissent un lien entre applications. Ainsi, des applications utilisant des technologies différentes peuvent envoyer et recevoir des données au travers de protocoles compréhensibles par tout le monde.

Les services Web sont **normalisés** car ils utilisent les standards XML et HTTP pour transférer des données et ils sont compatibles avec de nombreux autres environnements de développement. Ils sont donc indépendants des plates-formes.

C'est dans ce contexte qu'un intérêt très particulier a été attribué à la conception des services Web puisqu'ils permettent aux entreprises d'offrir des applications accessibles à distance par d'autres entreprises. Cela s'explique par le fait que les services Web n'imposent pas de modèles de programmation spécifiques. En d'autres termes, les services Web ne sont pas concernés par la façon dont les messages sont produits ou consommés par des programmes. Cela permet aux vendeurs d'outils de développement d'offrir différentes méthodes et interfaces de programmation au-dessus de n'importe quel langage de programmation, sans être contraints par des standards comme c'est le cas de la plate-forme *CORBA* qui définit des ponts spécifiques entre le langage de définition IDL et différents langages de programmation.

Les services Web représentent donc la façon la plus efficace de partager des méthodes et des fonctionnalités. De plus, ils réduisent le temps de réalisation en permettant de tirer directement parti de services existants.

### **II.2.3. Les caractéristiques d'un service Web :[4]**

La technologie des services Web repose essentiellement sur une représentation standard des données (interfaces, messageries) au moyen du langage XML. Cette technologie est devenue la base de l'informatique distribuée sur Internet et offre beaucoup d'opportunités au développeur Web.

Un service Web possède les caractéristiques suivantes :

- il est accessible via le réseau ;
- il dispose d'une interface publique (ensemble d'opérations) décrite en XML ;
- ses descriptions (fonctionnalités, comment l'invoquer et où le trouver ?) sont stockées dans un annuaire ;
- il communique en utilisant des messages XML, ces messages sont transportés par des protocoles Internet (généralement HTTP, SMTP, FTP.). ;

### **II.2.4. Architecture d'un service web: [4]**

Les services Web reprennent la plupart des idées et des principes du Web (HTTP, XML), et les appliquent à des interactions entre machines. Comme pour le **World Wide Web**, les services Web communiquent via un ensemble de technologies fondamentales qui partagent une architecture commune.

Ils ont été conçus pour être réalisés sur de nombreux systèmes développés et déployés de façon indépendante. Les technologies utilisées par les services Web sont HTTP, WSDL, REST, XML-RPC, SOAP et UDDI.

#### ***REST***

REST (*Representational State Transfer*) est une manière de construire une application pour les systèmes distribués comme le World Wide Web.

#### ***XML-RPC***

XML-RPC est un protocole simple utilisant XML pour effectuer des messages RPC. Il est indépendant de la plate-forme, ce qui lui permet de communiquer avec diverses applications.

**SOAP**

SOAP (*Simple object Access Protocol*) est un protocole standard de communication, décrit en XML et standardisé par le W3C. Il se présente comme une enveloppe pouvant contenir des données ou des pièces jointes. Il circule sur le protocole HTTP et permet d'effectuer des appels de méthodes à distance.

**WSDL**

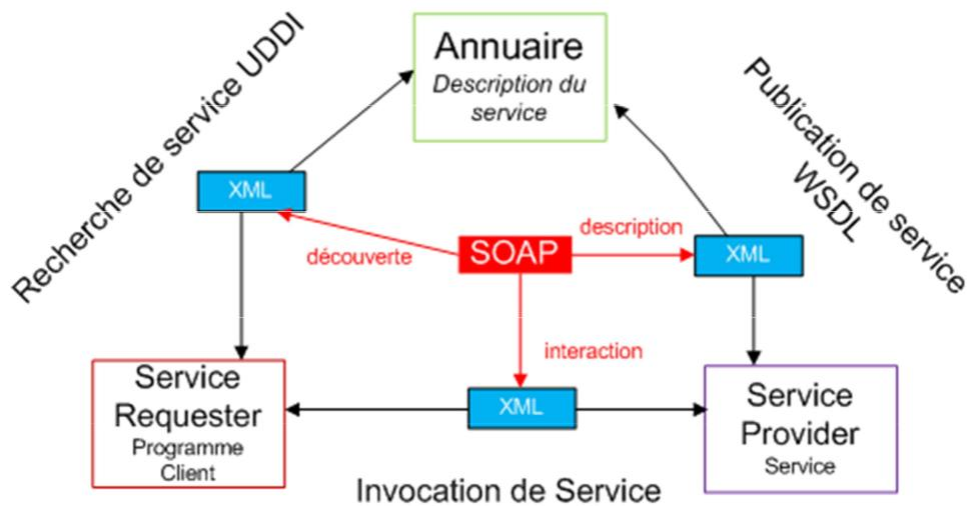
WSDL (*Web Services Description Language*) est un langage de description standard. Il est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées.

**UDDI**

UDDI (*Universal Description, Discovery and Integration*) est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services Web aux clients.

**II.2.5. Fonctionnement des services Web : [4]**

Le fonctionnement des services Web s'articule autour de trois acteurs principaux illustrés par le schéma suivant :



**Figure.II.6. Fonctionnement des services Web**

Décortiquons ce schéma.

### *Service provider service*

Le fournisseur de service met en application le service Web et le rend disponible sur Internet.

### *Service requester programme client*

C'est n'importe quel utilisateur du service Web. Le demandeur utilise un service Web existant en ouvrant une connexion réseau et en envoyant une demande en XML (REST, XML-RPC, SOAP).

### *Annuaire service registry*

Le registre de service est un annuaire de services. Le registre fournit un endroit central où les programmeurs peuvent publier de nouveaux services ou en trouver. Les interactions entre ces trois acteurs suivent plusieurs étapes :

- **La publication du service** : le fournisseur diffuse les descriptions de ses services Web dans l'annuaire.
- **La recherche du service** : le client cherche un service particulier, il s'adresse à un annuaire qui va lui fournir les descriptions et les URL des services demandés afin de lui permettre de les invoquer.
- **L'invocation du service** : une fois que le client récupère l'URL et la description du service, il les utilise pour l'invoquer auprès du fournisseur de services.

## II.2.6. Description en couche des services Web : [4]

Les services Web emploient un ensemble de technologies qui ont été conçues afin de respecter une structure en couches sans être dépendante de façon excessive de la pile des protocoles. Cette structure est formée de quatre couches majeures :

**Découverte de services** UDDI

**Description de services** WSDL

**Communication** SOAP

**Transport** HTTP

Couches technologiques des services Web.

### *A. Couche transport*

Cette couche est responsable du transport des messages XML échangés entre les applications. Inclus HTTP, SMTP, FTP, et de nouveaux protocoles tels que BEEP.

***B. Couche communication***

Cette couche est responsable du formatage des données échangées de sorte que les messages peuvent être compris à chaque extrémité.

***C. Couche description de service***

Cette couche est responsable de la description de l'interface publique du service Web. Le langage utilisé pour décrire un service Web est WSDL qui est la notation standard basée sur XML pour construire la description de l'interface d'un service.

***D. Couche publication de service***

Cette couche est chargée de centraliser les services dans un registre commun, et de simplifier les fonctionnalités de recherche et de publication des services Web.

**II.2.7. Avantages et inconvénients : [4]****Avantages**

- Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- Les services Web utilisent des standards et protocoles ouverts.
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges.
- Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les services Web existants.

**Inconvénients :**

- Les normes de services Web dans certains domaines sont actuellement récentes.
- Les services Web souffrent de performances faibles comparées à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM.

**II.3. XML/SOAP:****II.3.1. XML: [5]*****A. Introduction :***

XML (eXtensible Markup Language) est un format universel qui permet de structurer et d'organiser des documents et des données sur le Web. La version 1.0 de cette recommandation a été publiée par le W3C en février 1998 et ce format est devenu depuis incontournable.

Ce succès est bien sûr lié au développement d'Internet mais il tient aussi en grande partie aux objectifs initiaux que le groupe de travail s'était fixé et qui tiennent en quelques mots : simple (à construire, à lire, à traiter), précis (pas d'ambiguïté, règles syntaxiques strictes), universel (conforme à Unicode, indépendant de la plateforme logicielle), extensible.

### ***B. Définition :***[5]

XML est un sous-ensemble, une version simplifiée de SGML, et tout comme lui, c'est un langage à balises générique *markup( language)* : il établit les règles syntaxiques servant à marquer un document et à en dégager la structure mais il ne définit aucun jeu de balises (contrairement à HTML, par exemple). La définition de ces balises et de leur sémantique appartient au concepteur qui construit le document.

### **C. Un document XML :** [5]

Le corps d'un document XML est constitué d'un ou plusieurs éléments délimités par des balises ouvrantes et fermantes. Ces éléments sont organisés entre eux dans une structure arborescente. Dans l'exemple suivant :

```
<debut>
  <element1> Premier element </element1>
  <element2> Second element </element2>
</debut>
```

- debut est la racine du document ;
- element1 et element2 sont les fils de début ;
- debut est le père d'element1 et element2 ;
- element1 et element2 sont frères. Les éléments possèdent un contenu délimité par les balises.

Ce contenu peut être :

- simple s'il s'agit de texte uniquement ;
- mixte si l'élément possède à la fois un contenu simple et d'autres éléments ;
- vide : dans ce cas, la balise ouvrante est aussi fermante. Exemple : <saut\_de\_page/>.

### **D. Base de données XML :**[5]

Il s'agit d'une base de données dédiée au modèle XML. Cela signifie que les données sont stockées dans des documents XML, que les fonctions d'extraction et de mise à jour sont celles définies pour les données XML, que les méthodes d'accès sont compatibles avec un protocole tel que SOAP et que les standards définis par le W3C ou l'ISO sont tout ou partie mis en œuvre. Il existe :

- des bases « natives XML » développées sur le seul modèle XML ;

- des bases relationnelles dans lesquelles le XML est projeté (on dit « mappé »), ces bases étant restrictives car les modèles XML et relationnels ne sont pas égaux ;
- des bases XML qui utilisent un moteur objet ;
- des bases XML qui s'appuient sur des bases hiérarchiques.

### E. Les avantages de XML :

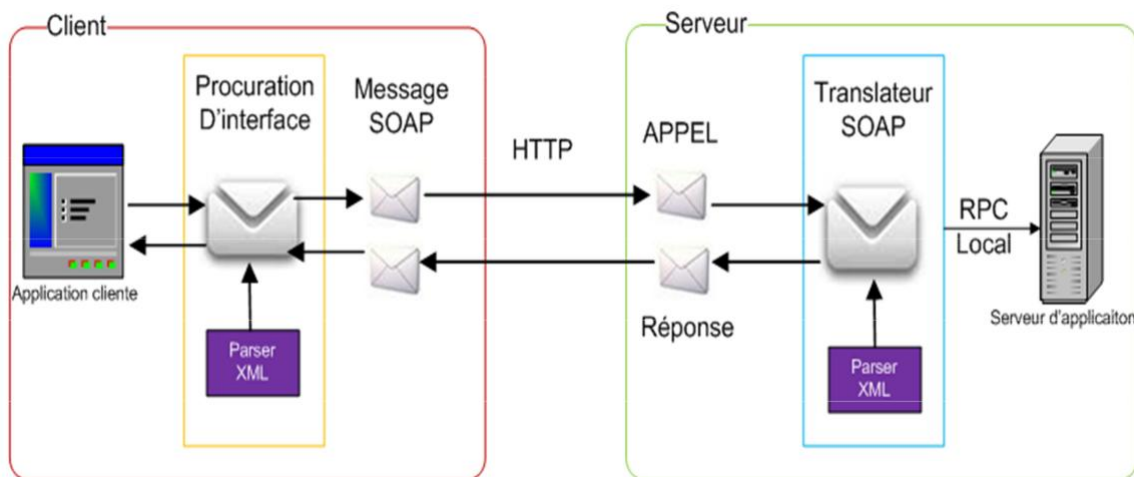
Voici les principaux avantages de XML :

- La lisibilité : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML.
- Une structure arborescente : permettant de modéliser la majorité des problèmes informatiques.
- Universalité et portabilité : les différents jeux de caractères sont pris en compte
- Déployable : il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme http.
- Intégrabilité : un document XML est utilisable partout application pourvue d'un parser (c'est-à-dire un logiciel permettant d'analyser un code XML).
- Extensibilité : un document XML doit pouvoir être utilisable dans tous les domaines d'applications.

### II.3.2. SOAP : [4]

#### Introduction :

SOAP est un protocole d'invocation de méthodes sur des services distants. **Basé sur XML**, SOAP a pour principal objectif **d'assurer la communication entre machines**. Le protocole permet d'appeler une méthode RPC et d'envoyer des messages aux machines distantes via HTTP. Ce protocole est très bien adapté à l'utilisation des services Web, car il permet de fournir au client une grande quantité d'informations récupérées sur un réseau de serveurs tiers :



#### A. Définition :[4]

Beaucoup de définitions normalisées de SOAP ont été proposées. Une particulièrement intéressante définit SOAP comme étant une spécification pour une omniprésence, basée sur XML et sur des infrastructures distribuées.

- **Spécification** car SOAP est un document qui définit le modèle de communication. L'idée de base est que si les deux parties ont créés programmes de mêmes spécifications, ils seront en mesure d'interagir de façon transparente.
- **Omniprésent** car SOAP est défini à un niveau suffisamment élevé d'abstractions que tout système d'exploitation et combinaison de langages de programmation peuvent être utilisés pour créer des programmes compatibles SOAP.
- **Basé sur XML**, SOAP est construit sur XML, ce qui signifie que les documents SOAP sont des documents XML construits en fonction d'un cahier de charges plus strict.
- **Infrastructure distribuée**, SOAP ne précise pas quelles données peuvent être déplacées ou bien quels appels de fonctions peuvent avoir lieu sur elle. Les applications construites sur la spécification SOAP peuvent déplacer les données d'un ordinateur A à un ordinateur B et par la suite à une autre application écrite sur la même spécification.

**B. Structure d'un message SOAP : [4]**

Un message SOAP est composé de deux parties obligatoires : l'enveloppe SOAP et le corps SOAP ; et une partie optionnelle : l'en-tête SOAP.

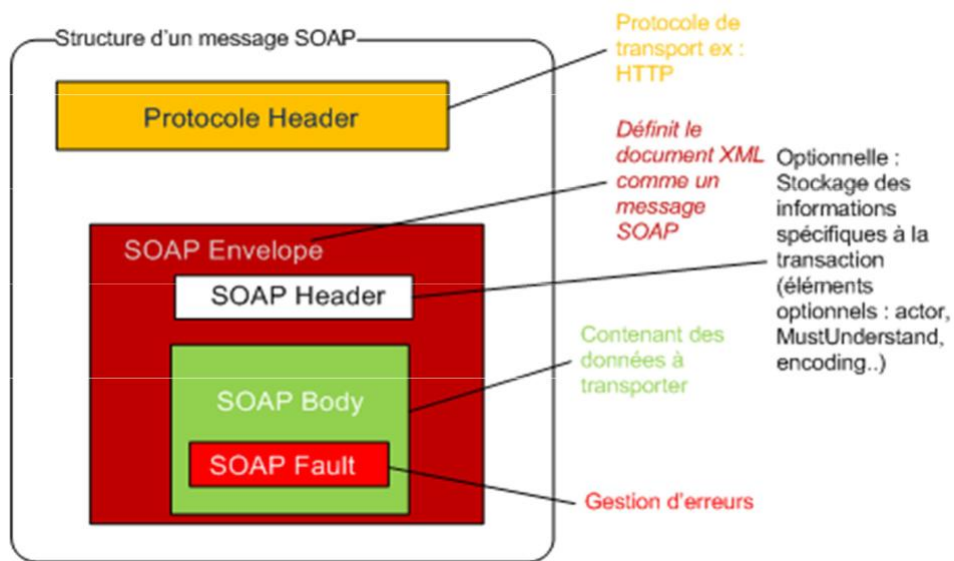


Figure.II.7. Structure d'un message SOAP

- **SOAP envelope** (enveloppe) est l'élément de base du message SOAP. L'enveloppe contient la spécification des espaces de désignatio (namespace) et du codage de données.
- **SOAP header** (entête) est une partie facultative qui permet d'ajouter des fonctionnalités à un message SOAP de manière décentralisée sans agrément entre les parties qui communiquent.

- **SOAP body** (corps) est un container pour les informations mandataires à l'intention du récepteur du message, il contient les méthodes et les paramètres qui seront exécutés par le destinataire final.
- **SOAP fault** (erreur) est un élément facultatif défini dans le corps SOAP et qui est utilisé pour reporter les erreurs.

### C. L'enveloppe SOAP : [4]

L'enveloppe SOAP sert de conteneur aux autres éléments du message SOAP, elle est définie au début par la balise `<soap:Envelope>` et se termine par la balise `</soap:Envelope>`. Les messages SOAP ne peuvent pas être envoyés en lots, autrement dit l'enveloppe contient un seul message constitué d'un entête facultatif (SOAPheader) et d'un **corps obligatoire** (SOAP body).

#### Le corps SOAP

Le corps SOAP est un élément obligatoire dans le message SOAP. Il contient l'information destinée au receveur. Le corps *body()* doit fournir le nom de la méthode invoquée par une requête ainsi que les paramètres associés celle-ci.

Le contenu du corps SOAP est utilisé pour spécifier un appel de méthode à un ordinateur distant avec les valeurs de paramètre. Par exemple, la demande du solde d'un compte bancaire.

#### L'en-tête SOAP

L'en-tête SOAP est un élément facultatif dans un message SOAP. Toutefois, si un en-tête est présent, il doit être le premier élément qui paraît dans l'enveloppe SOAP.

L'en-tête d'un message SOAP commence avec la balise `<soap:Header>` et se termine avec la balise `</soap:Header>`. Trois attributs associés à l'en-tête SOAP peuvent être utilisés :

- **soap:mustUnderstand** : cet attribut prend la valeur 1 ou 0. La valeur 1 signale que le récepteur doit reconnaître l'information présente dans l'en-tête et que son traitement est obligatoire. La valeur 0 indique que l'en-tête peut être ignoré par le récepteur.
- **soap:role** : sert à indiquer le destinataire SOAP auquel un bloc d'en-tête SOAP particulier est destiné.
- **soap:relay** : est utilisé pour indiquer si un bloc d'en-tête SOAP ciblé sur un récepteur SOAP doit être réacheminé (relayé) s'il n'est traité.

#### Message d'erreur SOAP

Afin de récupérer le plus grand nombre d'erreurs, l'approche SOAP se base essentiellement sur le bon usage de la balise `<soap:fault>` qui est contenue dans le corps SOAP.

Cette balise est utilisée pour communiquer un problème qui a eu lieu dans la tentative de réalisation de la demande adressée au service Web. L'élément d'erreur est facultatif et figure uniquement dans les messages de réponse. La balise <soap:fault> peut contenir quatre autres balises facultatives :

- **faultcode** : Il contient un code indiquant la nature du problème.
- **faultstring** : C'est la traduction en langage naturel du code d'erreur.
- **faultactor** : indique le service qui a généré l'erreur.
- **detail** : cet élément doit contenir autant d'informations ueq possible sur l'état du serveur à l'instant de l'apparition de l'erreur. Il contient souvent des valeurs de variables au moment de l'échec.

Quatre types de codes d'erreur sont définis par la spécification :

- **soap:Server** : indique qu'une erreur s'est produite sur le serveur, mais pas avec le message lui-même.
- **soap:Client** : signifie que le message reçu contient une erreur .
- **soap:VersionMismatch** : cette erreur se produit lorsque les versions des protocoles SOAP utilisés par le client et le serveur sont différentes.
- **soap:MustUnderstand** : cette erreur est générée lorsqu'un élément dans l'en-tête ne peut pas traiter alors qu'il est marqué comme obligatoire.

#### II.4. Adaptabilité :[6]

"L' **adaptabilité**" est la capacité du système de s'adapter aux personnalisations expressément demandées par l'utilisateur. En termes d'action, l' **adaptabilité** renvoie à un processus dit *statique*, basé essentiellement sur des connaissances disponibles ou acquises par le système avant que les interactions utilisateur-système ne soient engagées. Ainsi, les adaptations sont réalisées lors de l'initialisation du système qui se présente dans une version adaptée à l'utilisateur. De plus, les connaissances utilisées par le système semblent inchangées au cours d'utilisation.

Quant à l' **adaptativité**, elle a une vision plus *dynamique* du processus d'adaptation. En effet, les connaissances sont acquises ou modifiées par le système au cours des interactions. En effet, le système procède à des adaptations pendant que l'utilisateur interagit avec lui. Par ailleurs, l'utilisateur prend le contrôle de ses actions (initiation, proposition, sélection, exécution) sur le système dans l'adaptativité alors qu'il n'exerce aucun contrôle sur le système dans l'adaptabilité.

L'adaptation, qu'elle soit adaptabilité ou adaptativité, nécessite la définition d'un ou plusieurs modèles d'utilisateur. Le concepteur doit avoir une certaine représentation de l'utilisateur

**II.4.1. Définition : [6]**

Exprime la capacité du logiciel à communiquer et à utiliser les ressources d'autres logiciels comme, par exemple, les documents créés par une certaine application sans qu'il subisse des mises à jours.

**II.4.2. Apports : [6]**

Entre l'usage des concepteurs et celle des utilisateurs, les résultats montrent souvent des écarts considérables. C'est une des principales raisons qui pousse à considérer d'autant plus fortement l'utilité du concept d'adaptation. Comme le soulignent **Bastien et Scapin (1998)** lorsqu'ils évoquent que : « l'adaptabilité d'un système concerne sa capacité à réagir selon le contexte, et selon les besoins et préférences des utilisateurs ». Cela signifie que l'utilisateur, au gré de sa pratique et de son niveau, serait sensé bénéficier des fonctions et des commandes différentes : dans le but d'éviter des effets négatifs sur l'utilisateur, en fonction de son contexte et de ses besoins. En d'autres termes, c'est au système de s'adapter à l'utilisateur, selon le niveau d'expertise, en lui offrant des moyens disponibles non seulement pour personnaliser les fonctions, mais aussi pour répondre aux exigences souhaitées par une multitude d'actions. Dans le domaine de la formation, plus un environnement d'apprentissage offre des caractéristiques d'adaptation, plus l'apprentissage ne peut être individualisé.

Autrement dit, un logiciel adaptable est en mesure de répondre à un besoin spécifique de la part de l'utilisateur en cours d'utilisation. Pour illustrer notre argumentation, nous prendrons l'exemple d'une des possibilités existantes sur un logiciel de messagerie instantanée qui est MSN. Ce logiciel: "Plus rapide que le courrier électronique, c'est un outil idéal pour les conversations et la solution parfaite lorsque vous ne pouvez pas être présent en personne". Nous ne nous attarderons pas sur le logiciel en lui-même, mais seulement sur le concept d'adaptabilité. Aussi, nous l'aurons compris, l'idée de la messagerie instantanée est bien de pouvoir correspondre rapidement avec d'autres usagers. Mais si, en cours de communication l'utilisateur A désire parler à l'utilisateur B, en incluant d'autres utilisateurs tels que C et D dans la même conversation, le besoin alors ne sera plus le même. Et c'est justement à partir de ce point de vue que nous pouvons reconnaître que la fonction "inviter quelqu'un à rejoindre la conversation" que A va activer (en cliquant sur l'icône y relatif) devient une source d'adaptabilité pour nos usagers.

**II.4.3. Types de logiciels exploitant cette possibilité : [6]**

En guise d'exemples, nous proposons les logiciels suivants:

**Adaptabilité**

Certains systèmes d'exploitation et en général tous les logiciels dits "open source" sont construits sur le principe de l'adaptabilité. Exemple Linux permet à l'utilisateur de modifier le code source afin de l'adapter à ses besoins mais ceci demande des connaissances spécifiques.

**Adaptativité**

Logiciels de reconnaissance vocale: après une période d'apprentissage, durant laquelle le logiciel se familiarise avec la voix de l'utilisateur, le programme est capable de reconnaître les mots prononcés pour les transcrire par écrit. Exemple : Dragon Naturally Speaking.

**II.4.4. Tableau récapitulatif : [6]**

<b>Adaptabilité</b>	<b>Adaptativité</b>
L'utilisateur émet une demande explicite, le système s'adapte pour y répondre.	Le système s'adapte tout seul grâce à l'observation de l'utilisateur.
En amont de l'interaction utilisateur / système.	Pendant l'interaction utilisateur / système.
Statique.	Dynamique.
Dépend surtout du but de l'utilisateur.	Dépend surtout du but de l'action.

**II.4.5. Interopérabilité : [7]**

L'Interopérabilité est le fait que plusieurs systèmes, qu'ils soient identiques ou radicalement différents, puissent communiquer sans ambiguïté. Compte tenu du fait que ces systèmes sont produits par des constructeurs divers, avec des méthodes variées, et qu'ils répondent à des besoins spécifiques, l'idée la plus simple consiste à définir une base explicite, une norme, que chaque élément va « implanter » dans son propre fonctionnement. Cette norme joue un double rôle : elle est d'abord un indicateur de la façon dont le dialogue entre les différents éléments doit s'opérer – et cristallise donc les besoins de ce dialogue ; elle est ensuite une passerelle de communication, qui va pouvoir éventuellement s'adapter aux besoins changeants des éléments. La norme est alors proche d'une interface.

La notion d'interface est essentielle pour aborder l'interopérabilité. On ne peut parler de l'interopérabilité que lorsque les interfaces sont complètement définies, connues et librement utilisables. Les interfaces sont beaucoup moins complexes que les systèmes qui les utilisent. De plus, elles sont stables sur de longues durées car indépendantes des évolutions de ces systèmes.

**Interopérabilité en informatique :**

L'interopérabilité est une notion absolument cruciale pour les réseaux de télécommunication mondiaux comme le téléphone et l'Internet. Par essence, des matériels divers et variés sont mis en œuvre dans ces réseaux hétérogènes aux côtés d'une panoplie encore plus vaste de matériels informatiques et de logiciels.

Mais elle est cruciale aussi et surtout pour l'ensemble de l'économie, car aujourd'hui, dans presque tous les domaines d'activité, dans l'industrie et dans les services, et même en agriculture, des systèmes informatiques gèrent des données, pilotent des systèmes de contrôle,

des systèmes de gestion, et sont interconnectés d'une entreprise à l'autre par des réseaux informatiques (internet, extranet, messageries électroniques, ...).

### II.5. Les bases de données réparties :

#### II.5.1. Définition :

Une base de données répartie *BDR* est une collection de base de données localisées sur différents sites, généralement distants, mises en relations les unes avec les autres à travers un réseau d'ordinateurs, perçues pour l'utilisateur comme une base de données unique. Elle permet de rassembler des données plus ou moins hétérogènes, disséminées dans un réseau sous forme d'une base de données globale, homogène et intégrée.

#### **Définition d'un système de gestion de base de données répartie:**

Système gérant une collection de BD logiquement reliées, réparties sur différents sites en fournissant un moyen d'accès rendant la distribution transparente.

Un SGBD réparti doit rendre la répartition des bases de données transparentes aux utilisateurs. La base de données étant répartie, il faut également répartir certaines fonctionnalités du SGBD.

#### II.5.2. Buts de la répartition des bases de données:

Les bases de données réparties ont une architecture plus adaptée à l'organisation des entreprises décentralisées.

**Plus de fiabilité:** les bases de données réparties ont souvent des données répliquées. La panne d'un site n'est pas très importante pour l'utilisateur, qui s'adressera d'autre site.

**Meilleures performances :** réduire le trafic sur le réseau est une possibilité d'accroître les performances. Répartir une base de données sur plusieurs sites permet de répartir la charge sur les processeurs et sur les entrées/ sorties.

**Faciliter l'accroissement:** l'accroissement se fait par l'ajout de machines sur le réseau.

#### II.5.3. Conception d'une base de données répartie :

On distingue deux méthodes de conception :

##### **Conception ascendante :**

Cette approche se base sur le fait que la répartition est déjà faite, mais il faut réussir à intégrer les différentes BDs existantes en une seule BD globale.

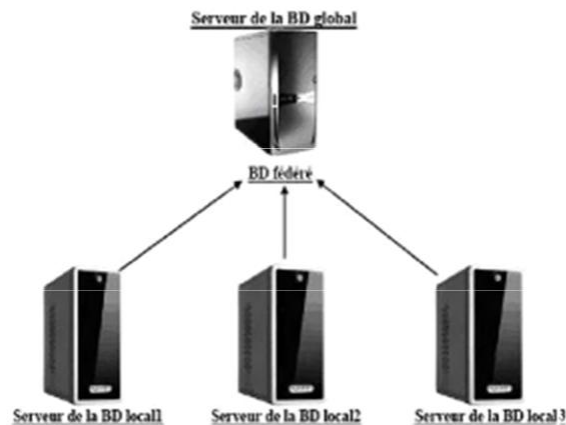


Figure II.8. Architecture de la conception ascendante. [17]

**Conception descendante :**

On commence par définir un schéma conceptuel global de la base de données répartie, puis on le distribue sur les différents sites en des schémas conceptuels locaux.

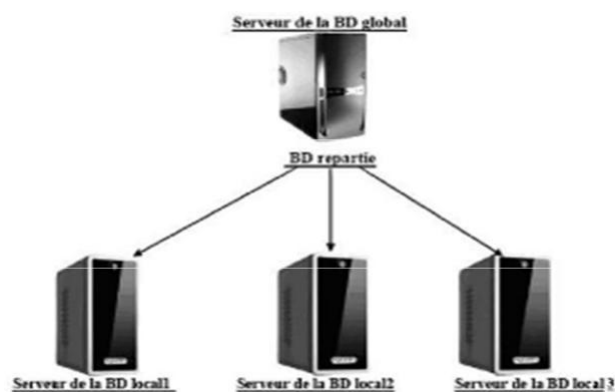


Figure II.9. Architecture de la conception descendante. [17]

**II.5.4. Architecteur des bases de données réparties:**

Du point de vue organisationnel on distingue deux architectures :

**Architecture client/serveur :** , c'est le SGBDR qui se charge de trouver la solution à la requête en question, et de retourner un résultat aux clients.

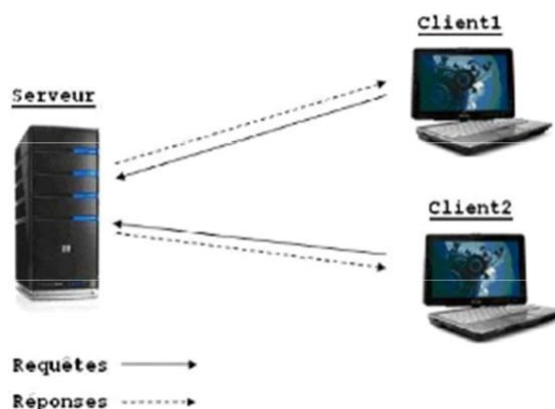


Figure II.10. Architecture Client / Serveur [17].

### Architecture Peer To Peer:

C'est un type de communication pour lequel toutes les machines ont une importance équivalente. Il n'y a pas de machine qui a une importance hiérarchique par rapport aux autres. Dite aussi, l'architecture totalement répartie.

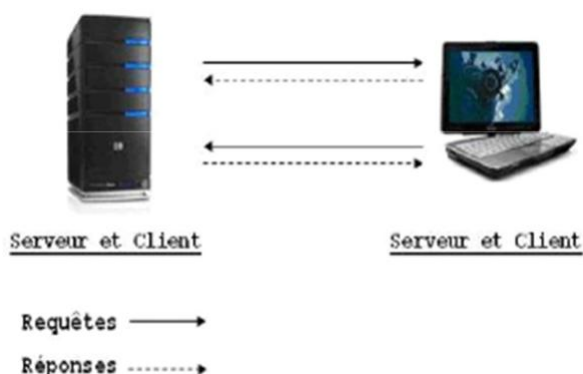


Figure II.11. Architecture Peer To Peer [17].

Chacune de ces architectures possède des avantages et des inconvénients. Le Client / Serveur avec sa structure plus hiérarchique est très sensible aux problèmes de panne des serveurs, bloquant ainsi les clients. En revanche, la prise de décision des serveurs est rapide.

Pour l'architecture Peer-To-Peer, comme les machines sont strictement équivalentes, la panne d'une machine peut rarement rendre le système un peu lent. Mais cette architecture engendre énormément de communication pour toute décision.

### II.5.5. Sécurité des bases de données réparties :

La sécurité se base sur trois principes :

- **Confidentialité** : L'information protégée ne doit pas être accessible aux utilisateurs ou un programme non autorisés (militaires ou commerciaux).
- **Intégrité** : Les données ne peuvent pas être modifiées que par les utilisateurs habilités à le faire (Des pannes de système).
- **Disponibilité** : Il s'agit de détecter ou d'empêcher des dénis de service (lorsqu'un utilisateur ne parvient pas à accéder dans un délai raisonnable, à une information pour laquelle il a une autorisation d'accès).

### II.5.5.1. Exemple de politique de sécurité :

**Authentification** : est la procédure mise en œuvre notamment pour vérifier l'identité d'une entité et s'assurer que l'identité fournie correspond à l'identité de cette entité préalablement enregistrée. (identifiant et mot de passe)

**Cryptographie** : désigne l'ensemble des méthodes ou techniques *chiffrement*, (*signature numérique et certificat*) permettant de garantir intégrité, authenticité, confidentialité des informations sensibles, et on appelle la personne qui la pratique un *cryptographe*.

**Création des messages d'erreurs** : Si vous n'êtes pas prudent, un utilisateur malveillant peut déduire des informations importantes sur votre application à partir des messages d'erreur qui s'affichent. Respectez les règles ci-dessous.

- N'écrivez pas de messages d'erreurs reproduisant des informations pouvant être utilisées par des utilisateurs malveillants, telles qu'un nom d'utilisateur.
- Configurez l'application de façon à ne pas afficher d'erreurs détaillées.
- Utilisez l'élément de configuration pour contrôler les personnes qui peuvent consulter les exceptions à partir du serveur.
- Créez une gestion des erreurs personnalisée pour les situations sujettes aux erreurs, telles que l'accès aux bases de données.

**II.5.5.2. La protection d'une base de données** : Pour la protection d'une base de données il faut:

Connaître son besoin (les utilisateurs, la manière dont la connexion s'effectue directement ou par le biais d'une interface).

Une sécurité (doit être pour l'ensemble des éléments, surtout dans le cas d'un applicatif accédant à la base).

Supervision (Un suivi des indicateurs de la base de données doit être assuré afin de détecter les anomalies) .

Sensibiliser les DBA(L'administrateur doit être sensibilisé aux problématiques de sécurité)

Gestion des comptes(Les comptes par défaut doivent être verrouillés et les mots de passe remplacés pour respecter les normes de sécurité)

Méthodes d'accès (L'entrée sur la base de données doit être autorisée selon des méthodes précises.

Chiffrer les flux de données (Les informations envoyées en réponse à une requête ne doivent pas circuler en clair sur le réseau).

### **Conclusion:**

A travers les différents points développés dans le présent chapitre, nous avons pu constater l'intérêt particulier porté à l'adaptabilité et les bases de données réparties entre les systèmes et aux différents problèmes auxquels ce type de solution a pu remédier.

Nous avons pu également, détailler et expliquer certains concepts des nouvelles technologies de l'information et de la communication à savoir : l'architecture orientée service et l'intérêt des services web, ainsi les caractéristiques de XML/SOAP. Une présentation des domaines d'études est primordiale pour le prochain chapitre sur les généralités de e-Learning.

## **Introduction**

Depuis quelques années, les nouvelles technologies de l'information et de la communication (NTIC) s'intègrent de plus en plus dans la vie privée et la vie professionnelle. L'application des NTIC au domaine de la formation a conduit à la création d'une nouvelle réalité appelée e-learning. De manière générale, l'e-learning désigne l'apprentissage par le biais d'outils et de ressources numériques et électroniques accessibles à distance. Depuis l'ère de l'enseignement assisté par ordinateur où le véritable outil était alors le logiciel utilisé, qui porte avec lui une stratégie pédagogique et une conception de l'apprentissage, l'évolution de la technologie a introduit la possibilité de travailler à distance, à plusieurs, où on veut, quand on veut. Cela a contribué au développement d'une seconde approche dans laquelle l'ordinateur est perçu comme un moyen de permettre un apprentissage par la découverte et l'exploration en donnant le contrôle à l'apprenant.

Dans ce chapitre nous allons présenter l'e-learning d'une manière générale. Premièrement, nous nous intéresserons à l'historique de l'e-learning et son évolution au cours de ces dernières années, puis, nous mettrons la lumière sur ses aspects à travers un panorama des principales plates formes développées, et nous mettrons finalement en évidence les points forts et les faiblesses de l'e-learning.

### **III.1. Définition de la formation à distance : [9]**

La formation à distance consiste principalement en « la séparation de l'étudiant et du professeur dans l'espace et dans le temps ». Dans sa forme la plus simple, la formation à distance vise principalement à fournir une formation pour les personnes isolées - géographiquement. La communication entre les étudiants et les instructeurs se fait de deux façons, soit synchrone (en temps réel), soit asynchrone (différée dans le temps), notamment à l'aide des TIC.

Les établissements sont divisés en deux types, soit unimodal et bimodal. Le type unimodal signifie que l'établissement offre uniquement de la formation à distance. Le type bimodal désigne un établissement qui offre à la fois de la formation à distance et de l'enseignement en mode présentiel.

#### **III.1.1. Les avantages et inconvénients de la formation à distance : [9]**

##### **III.1.1.1. Les avantages de la formation à distance :**

L'enseignement à distance est apparu au 19<sup>ème</sup> siècle. Il est né de la volonté de rendre accessible la formation pour tous. La formation à distance vise principalement :

- Les personnes éloignées des établissements de formation ;
- Les personnes voulant concilier activité salariale et formation ;
- Les hommes et femmes au foyer ;
- Les personnes handicapées ;
- Les expatriés ;
- Les personnes préférant l'e-Learning par choix.

La formation à distance s'est imposée aujourd'hui comme une méthode de formation à part entière car elle possède de nombreux avantages:

- Elle s'adapte aux contraintes de temps et à l'organisation de chacun ;
- Elle permet d'alterner formation pratique et théorique (formation en alternance, stages en milieu d'application) ;
- Elle permet de travailler de façon autonome et indépendante ;
- Elle développe l'esprit de curiosité et de recherche ;
- Le coût de la formation est amoindri (locaux, déplacement des formateurs) ;
- Elle permet d'éviter l'inconvénient du transport et l'éloignement géographique.

### **III.1.1.2. Les inconvénients de la formation à distance :**

Malgré de nombreux avantages non négligeables, la formation à distance dispose de nombreux inconvénients :

- L'appréhension de l'outil informatique et les éventuelles difficultés d'utilisation ;
- La mauvaise gestion de l'autonomie due à une absence d'encadrement ;
- L'obligation de motivation et de forte implication ;
- La plupart du temps pas de contact direct avec le formateur.

### **III.1.2. Le télé-enseignement [9]:**

Est une forme particulière d'enseignement à distance ; c'est une application adaptée notamment à l'enseignement spécialisé, soit dans les universités, permettant de bénéficier des meilleurs cours existants, soit comme une option à bas coût pour la formation continue des professionnels, en utilisant des outils de télécommunication. Les avantages de la télé-enseignement :

Le matériel pédagogique est disponible et mis à jour régulièrement.

On peut refaire les mêmes leçons autant de fois que nécessaire.

Les cours sont disponibles n'importe où dans le monde.

Les contrôles de connaissances peuvent être automatisés.

Les apprenants sont chez eux et suivent les cours d'une université à l'étranger. En offrant aux apprenants du matériel on les habitue à une plus grande autonomie. Le télé-enseignement dès l'école primaire accroît les chances de succès.

## **III.2. E-Learning:**

### **III.2.1. Définition: [8]**

Le terme anglais « e-learning » correspond en français à l'apprentissage électronique. L'e-learning est un processus d'apprentissage à distance s'appuyant sur des ressources multimédias, qui permet à une ou plusieurs personnes de se former à partir de son ordinateur. Ce mode d'apprentissage tire parti de l'usage des technologies de l'information et de la communication (TIC). En effet, ces TIC offrent une pléiade de supports multimédias (texte, graphisme, son, vidéo, etc.) qui permettent de révolutionner l'approche pédagogique, d'employer des méthodes plus ludiques où l'interactivité joue un grand rôle, de diversifier les outils employés, de s'adapter davantage au processus d'apprentissage de l'apprenant, qui devient le pilote de sa formation.

Le terme e-learning peut désigner particulièrement un dispositif de formation dont les principaux objectifs peuvent être définis comme l'autonomie d'apprentissage, la formation à distance, l'individualisation des parcours de formation et le développement des relations pédagogiques en ligne.

### **III.2.2. Historique et évolution de l'e-learning :[7]**

La formation à distance n'est pas un phénomène récent puisqu'elle existe depuis plusieurs années. Au début par correspondance pour les adultes n'ayant pas pu achever leur enseignement secondaire ou supérieur.

Au fil des années, cette méthode a connu une évolution marquée, depuis le papier acheminé par poste ou par fax, ensuite la diffusion hertzienne via la radio et les émissions spécialisées des chaînes de télévision arrivant ainsi à l'enseignement assisté par ordinateur (EAO).

#### **III.2.2.1.EAO : [8]**

L'EAO se définit comme étant l'utilisation de l'outil informatique dans un objectif pédagogique, d'enseignement ou de formation.

Un didacticiel est un logiciel d'EAO, spécialisé dans une ou plusieurs matières, mettant un apprenant en interaction avec la matière enseignée dans le but de lui faire acquérir des connaissances. Il existe plusieurs sortes de didacticiels :

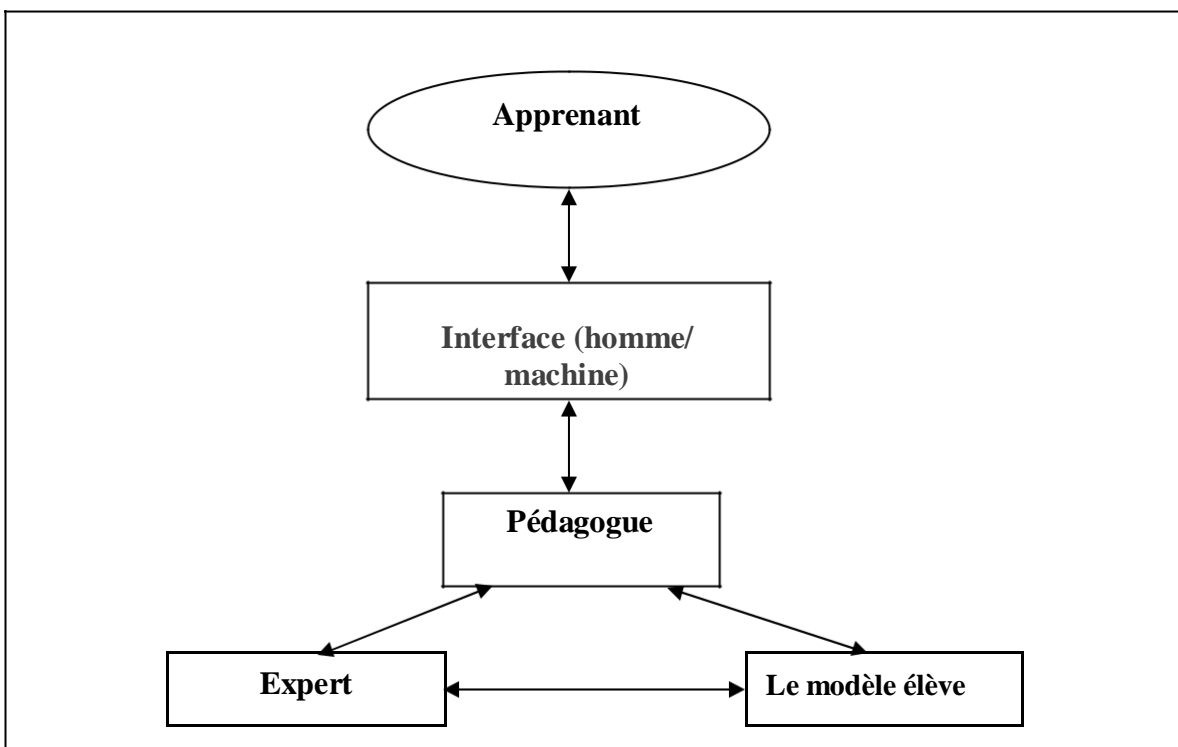
- **Le didacticiel de test** : est un logiciel à but purement évaluatif ou diagnostic. L'apprenant doit répondre à un questionnaire séquentiel, ouvert ou à choix multiple, dans un domaine particulier.
- **Le didacticiel informatif** : est un logiciel sous forme d'une suite de pages-écrans, qui sert à proposer et à présenter des connaissances dans un domaine précis sans chercher à vérifier leurs acquisitions par l'apprenant.
- **Le didacticiel d'entraînement ou drill** : combine le didacticiel de test et le didacticiel informatif.
- **Le didacticiel de simulation** : est un logiciel basé sur le principe pédagogique qui consiste à familiariser l'apprenant avec un modèle en lui donnant l'occasion de voir les différentes formes du modèle en faisant varier ses paramètres.
- **Le tutoriel** : utilise toutes les formes précédentes, c'est la forme la plus ambitieuse d'EAO, tant dans son but que dans les moyens mis en œuvre. Le tutoriel prend en compte :
  - Les divers sous objectifs du but pédagogique, qui ne doivent pas être traités d'une manière séquentielle.
  - Pour chaque objectif, diverses stratégies utilisées: module informatif, test, simulation.
  - La structure de suivi de l'apprenant, qui doit permettre un diagnostic final ou partiel.
  - Eventuellement une structure d'orientation, destinée à aiguiller l'apprenant selon son niveau initial, déclaré ou mesuré.

- **Le pédagogue:** est un didacticiel possédant en plus les caractéristiques suivantes :
  - Une adéquation pédagogique qui prime tout raffinement concernant les canaux de communication.
  - Les questions « ouvertes » le sont réellement (il n’y a pas de QCM caché) et le traitement des réponses en langue naturelle se fait intelligemment.
  - Les modules d’aides sont appuyés par des filtres logiciels pédagogiques accédant à une encyclopédie thématique.

**III.2.2.2.EIAO : [8]**

Afin d’améliorer les possibilités d’adaptation, les années 80 sont marquées par l’utilisation des techniques d’intelligence artificielle dans les systèmes d’apprentissage. Les systèmes tuteurs intelligents (STI) marquent le passage de l’EAO à l’EIAO alors décliné en Enseignement Intelligemment Assisté par Ordinateur.

Pour s’adapter à l’élève en situation d’apprentissage, le modèle général des STI repose sur une triple expertise : celle du domaine à enseigner (expert du domaine), celle de l’enseignement (module pédagogique) et celle des compétences et connaissances de l’élève (modèle de l’élève), comme le montre la figure suivante :



**Figure III.1. Les composants d’un système d’EIAO**

- **L’interface apprenant :**  
 Cette interface permet à l’apprenant de communiquer avec le pédagogue. Il est important qu’elle soit ergonomique, conviviale et facile à utiliser afin que l’apprenant se sente à l’aise dans son apprentissage.

- **Le pédagogue :**

Le pédagogue est le noyau du système. Il a la charge de coordonner les interactions avec l'apprenant pendant la session d'enseignement. Il choisit le scénario à lancer ; en tenant compte du niveau de connaissance de l'apprenant, de l'objectif d'enseignement fixé et d'un certain nombre de variables pédagogiques.

- **L'expert :**

L'expert a pour tâches :

- La génération de problèmes.
- La résolution de problèmes.
- Le test des réponses de l'élève (diagnostic des erreurs).
- L'explication.

- **Le modèle élève**

Les systèmes d'EIAO se caractérisent par l'individualisation de l'enseignement, d'où l'utilisation d'un modèle propre à chaque apprenant. Le modèle élève représente les connaissances et les aptitudes de l'apprenant qui s'enrichit au fur et à mesure de l'apprentissage. Il contient les connaissances liées au domaine (ce que l'apprenant sait, ne sait pas, ce qu'il sait faire et ne sait pas faire), les connaissances indépendantes du domaine enseigné (les mécanismes d'apprentissage préférés par l'apprenant) ainsi que les connaissances servant à l'identification de l'apprenant.

### **III.2.2.3. De l'EIAO à l'EIAD : [8]**

Puis apparaît l'EIAD, qui rajoute la dimension « à distance » à l'EIAO. Cette dimension supplémentaire va faire apparaître le besoin de prise en compte de l'individu à distance et du renforcement du lien social, qui va faire intervenir les outils de communication synchrones et asynchrones entre individus, que ce soit pour les relations enseignant-élève que pour les relations entre élèves. La formation repose désormais à la fois sur une dimension communicationnelle et sur une dimension éducative. La communication et l'interaction entre des machines et des humains distribués géographiquement, donnent naissance aux Environnements Informatiques pour l'Apprentissage Humain (EIAH). Un EIAH est un environnement qui intègre des agents humains (élève ou enseignant) et artificiels (informatiques) et leur offre des conditions d'interaction, localement ou à travers les réseaux informatiques, ainsi que des conditions d'accès à des ressources formatives (humaines et/ou médiatisées) locales ou distribuées.

Les EIAD peuvent être définis comme étant des environnements cherchant à créer des conditions de construction de connaissances chez l'apprenant à partir d'interactions avec un système informatique et avec d'autres acteurs (apprenants, enseignants).

**III.2.2.4. Enseignement à distance ou formation à distance (EAD/FAD) : [8]**

La formation à distance (FAD) est un moyen qui permet de se former sans se déplacer sur le lieu de formation et sans nécessiter la présence physique d'un formateur. La transmission des connaissances et les activités d'apprentissage se situent en dehors de la relation directe en face à face, dite « en présentiel » entre l'enseignant et l'apprenant. La formation à distance couvre l'ensemble des dispositifs techniques et des modèles d'organisation qui ont pour but de fournir un enseignement ou un apprentissage à des individus qui sont distants de l'organisme de formation.

**III.2.2.5. Formation ouverte et à distance : [8]**

Le terme « formation ouverte et à distance » (FOAD) est la traduction et de l'adaptation par le gouvernement français du terme anglais « Open and Distance Learning ». Contrairement à la FAD, la formation ouverte ou Open Learning, permet aux apprenants des entrées et des sorties permanentes. Elle correspond à un mode d'organisation pédagogique diversifié qui s'appuie sur des apprentissages à distance, en auto-formation. En formation ouverte, l'apprenant peut alterner des séquences individuelles et collectives. L'auto-formation est un processus par lequel l'individu détermine son itinéraire d'apprentissage (rythme, contenu, temps de travail) de façon autonome et en étant éventuellement en relation avec un tuteur (enseignant ou formateur) ou un groupe structuré.

Notons que le terme FOAD est le terme couramment utilisé pour désigner une situation éducative dans laquelle la transmission des connaissances se situe en dehors de la relation directe entre l'enseignant et l'apprenant. En effet, le terme « FOAD » permet d'englober les deux processus éducatifs, l'enseignement à distance (point de vue enseignant) et l'apprentissage à distance (point de vue apprenant).

**III.2.3. Typologie d'e-Learning : [9]**

La catégorisation du e-Learning peut être conçue selon des critères différents :

La définition large de l'e-Learning avancée par les instances scientifiques de l'Union européenne distingue 4 types de projets e-Learning, du plus simple au plus complexe :

**Niveau 1 : un outil d'aide aux formateurs :**

Utiliser le réseau pour préparer et/ou suivre un cours présentiel

création de communautés virtuelles, échanges... : début du tutorat et importance de l'animation mais pas encore de contenu

soutien à la formation classique

**Niveau 2 : mise à disposition des savoirs via le réseau :**

un portail donnant un accès à des documents pédagogiques

complément de formation en libre service (vient comme un « plus » par rapport au présentiel et s'organise autour).

**Niveau 3 : un « guidance » à distance des apprenant s :**

Avec définition de parcours pédagogiques (en général mix à distance/en classe), tutorat et suivi important, changement du processus de formation, de l'organisation, des habitudes, implication de nombreux acteur.

**Niveau 4 : formation intégrée aux dispositifs de gestion des compétences et des connaissances :**

Projet RH dont la formation est un des aspects : le e-Learning vient en support à la gestion, la capitalisation, la mise à disposition d u savoir de l'entreprise et vient en support également au développement des compétences (identification des compétences par activité, par fonction puis réalisation de parcours correspondants aux compétences identifiées et à développer).

En se référant à ces modèles, il en ressort que le e-Learning peut être conçu comme une pratique émergente de management adoptée par les entreprises pour la formation du capital humain. Cette conception prend une autre coloration dans le contexte de l'enseignement : le e-Learning renvoie aussi bien aux formations d'entreprise qu'à la formation professionnelle continue comme aux initiatives en matière d'enseignement supérieur.

Dans notre travail, on va approcher le e-Learning dans le domaine de l'enseignement et plus particulièrement, l'enseignement supérieur.

**III.2.4. Les différentes formules du E-Learning : [9]**

Lorsque les cours sont donnés grâce à un ordinateur , relié à un Intranet (les cours sont alors limités au personnel de l'entreprise) ou à un Extranet (l'accès est alors élargi à des personnes extérieures).

Trois situations peuvent alors se présenter : le cours magistral synchrone, l'assistance et l'autoformation connectée.

**Un cours magistral synchrone**

Un formateur est relié en mode audio-visuel à plusieurs stagiaires (en moyenne quatre à huit) situés dans des lieux géographiques qui peuvent être très différents. L'avantage du E-Learning est alors double.

Pour les stagiaires, il offre la possibilité de se former depuis leur lieu de travail, voir même depuis leur domicile. C'est confortable et cela permet aussi de gagner du temps. Pour

l'employeur, l'avantage financier est considérable : pas de coûts liés au déplacement des stagiaires, pas de surcoûts liés au temps perdu.

#### **L'assistance**

Dans cette phase, vous allez travailler individuellement, mais vous disposez d'une assistance audio-visuelle en direct.

Le formateur reste à votre disposition et peut, à distance, prendre le contrôle de votre poste de travail. Il peut ainsi partager les logiciels ou les documents. L'assistance peut également se manifester de manière asynchrone, au moyen de messageries ou de forums informatiques.

#### **L'auto-formation connectée**

Les personnes en formation peuvent à partir des ressources pédagogiques ou des parcours de formation disponibles sur le serveur internet ou intranet, suivre un cours magistral enregistré, en audio et en vidéo, sous forme par exemple, de séances de rattrapage.

Ce qui explique, aujourd'hui, le phénomène d'engouement pour le e-Learning est la souplesse de cet outil pour l'utilisateur et son faible coût d'utilisation, une fois les investissements de départ réalisés. La personne en formation peut beaucoup plus facilement concilier formation et temps de travail et est pratiquement affranchie des contraintes de lieu et de temps. En outre, elle peut plus facilement apprendre à son rythme et en utilisant la ou les méthodes de son choix.

Mais le problème est l'important coût d'investissement en amont pour mettre en place de tels supports. Difficile de les rentabiliser à court terme.

### **III.2.5. Les principales caractéristiques de e-Learning : [9]**

- Internet est le moyen d'accès à la matière. Vous avez accès à la matière d'études 7 Jours sur 7 et 24h/24.
- L'étudiant est l'élément central. Il peut définir lui-même son rythme, son Environnement et sa méthode de travail.
- L'environnement d'études peut être adapté aux besoins de l'entreprise.
- Le tout est soutenu par un réseau d'expériences qui peuvent être partagées.

### **III.2.6. Les formes de communication et de collaboration d'e-Learning : [9]**

Les outils de communication et de collaboration jouent un rôle essentiel en e-Learning puisqu'ils rendent possible le tutorat, facilitent le travail des tuteurs et permettent la construction d'une véritable communauté d'apprentissage. Ils se subdivisent en deux catégories :

**les formes synchrones.** Ils permettent de communiquer en temps réel entre des personnes distantes géographiquement : messagerie instantanée (chat, clavardage, etc.), téléphonie par le net, audioconférence et vidéoconférence, etc. En effet, pour "chatter", il faut que les "chatteurs" se trouvent en même temps face à leur ordinateur respectif ;

**les formes asynchrones.** Ils permettent des échanges décalés dans le temps et l'espace : courrier électronique (e-mail), forum (de discussions), portfolio, wiki, blog et vlog, etc. Quand une personne envoie un mail, le récepteur peut le lire quelques minutes, heures ou jours plus tard (visionnez les démos du Forum et du Wiki).

### III.2.7. Les outils de communication : [9]

**Le mail :** courriel ou e-mail courrier électronique, échange de messages, via la messagerie électronique. **Le forum :** asynchrone, animé par un formateur ou animateur de site web (il a le nom de modérateur pour les forums du web), et fonctionne par messages écrits.

**La foire aux questions :** les questions de chaque apprenant et leurs réponses données par un formateur, voir un autre apprenant peuvent être consultées.

**lettre de diffusion ou news :** La permet de chaque apprenant et leurs réponses données par groupe d'abonnés ou d'inscrits.

**Les wiki :** est un outil qui permet de mettre en place un site web collaboratif avec une hiérarchie dans ses pages et s'il en a besoin de modifier leur contenu ou tout simplement ajouter de nouvelles pages. Créer un web mis à jour régulièrement devient alors très facile.

Dans le cadre d'e-learning, il sera intéressant d'ouvrir un wiki privé, réservé aux membres d'un groupe. Ils pourront ainsi bâtir une base de connaissances structurée qui peut être mise à jour ou corrigée facilement.

Des systèmes d'authentications permettent de sécuriser les pages et de ne les rendre éditables que par certains groupes ou même de bloquer les pages pour éviter des débordements.

**Les blogs :** est un outil qui permet de publier des articles rédigés par une ou plusieurs personnes. Les sujets peuvent varier d'un journal intime anonyme à un recensement de nouvelles sur un sujet particulier.

Les articles sont généralement présentés du plus récent au plus ancien. Il est donc considéré que les visiteurs viennent régulièrement prendre connaissance des nouveaux articles publiés.

Un système où les visiteurs peuvent ajouter un commentaire est dans la majorité des cas intégré ce qui permet à chacun de réagir à un article donné.

Dans le cas d'e-learning, les blogs pourraient être utilisés par un groupe. Ce qui permettrait à chacun de proposer des articles être utilisés, de proposer des liens ou des nouvelles par le biais de cet outil. Les étudiants construisent donc leur savoir eux-mêmes à travers leurs recherches qu'ils pourront publier et consulter.

**L'audioconférence et la vidéoconférence** permettent à deux ou plusieurs personnes éloignées de communiquer entre elles en temps réel. Le son et l'image sont émis et réceptionnés. Cet outil permet dans une application d'e-learning d'offrir un apprentissage à distance. L'enseignant peut envoyer en même temps des schémas, images et autres ressources visuelles aux étudiants se situant sur différents sites. De

plus, l'étudiant a la possibilité de prendre la parole pour poser des questions ou y répondre. Il peut être aussi utilisé pour la diffusion de cours ou de conférences. Dans ce cas, si ces transmissions sont enregistrées, on pourrait avoir une utilisation asynchrone de cet outil.

**Le chat** : ou (échanges de messages en temps réel via un logiciel spécifique. La messagerie instantanée : permet des communications textuelles uniquement entre deux personnes. L'émetteur choisit, son interlocuteur à partir d'une liste de présence des connectés à la session synchrone. Le message s'affiche immédiatement dans la machine du destinataire.

**Tableaux blancs partagés**: ces outils permettent de travailler à plusieurs sur des données textuelles, et/ou graphique en prenant le concept de tableau sur lequel tout le monde peut écrire.

**Le streaming** : ou webcasting (diffusion ou web diffusion) est un système informatique qui permet de diffuser en temps réel, c'est-à-dire sans avoir besoin de télécharger un fichier sans le visualiser, un flux audio-vidéo depuis un serveur de streaming vers plusieurs clients.

L'utilisateur devra installer préalablement un plug-in ou un client de streaming. A l'aide de son navigateur il pourra ensuite, en cliquant sur le lien adéquat le visualiser instantanément (ou presque) sur son poste de travail.

Deux modes de diffusion sont possibles sur Internet : en direct ou à la demande. Dans le mode direct, l'utilisateur se connecte à un flux en cours qui est retransmis de la même manière à tous les clients. Mais pour la vidéo à la demande (VOD), l'utilisateur fait une requête qui est traitée uniquement pour lui.

### **III.3. Plate forme d'e-learning: [8]**

Une plate forme électronique pour la formation ouverte et à distance est une application qui assiste la conduite des enseignements à distance. Ce type de logiciel regroupe les outils nécessaires aux quatre principaux types d'utilisateurs (enseignant, apprenant, auteur, administrateur) d'un dispositif qui a pour finalité la consultation à distance de contenus pédagogiques, l'individualisation de l'apprentissage et le télé-tutorat. Une plate forme d'e-learning doit généralement remplir les fonctions suivantes :

- La gestion du contenu, qui recouvre la production et le stockage de ressources pédagogiques (création de cours, bibliothèque de formation).
- La gestion de la formation, qui recouvre la gestion administrative (inscription des apprenants, gestion des accès et des acteurs) et l'accès aux cours (connexion, identification et téléchargement).
- La gestion de l'interactivité, qui recouvre l'accompagnement asynchrone de l'apprenant (messagerie, forum, outils collaboratifs) et les classes virtuelles synchrones (vidéo-conférences, partage d'applications).
- La gestion des compétences, qui recouvre la mise en rapport des compétences individuelles avec les besoins de formation pour bâtir des programmes adaptés.

Certaines plates formes couvrent d'autres fonctions comme l'organisation des ressources. Le fonctionnement d'une plate-forme est illustré par la figure suivante :

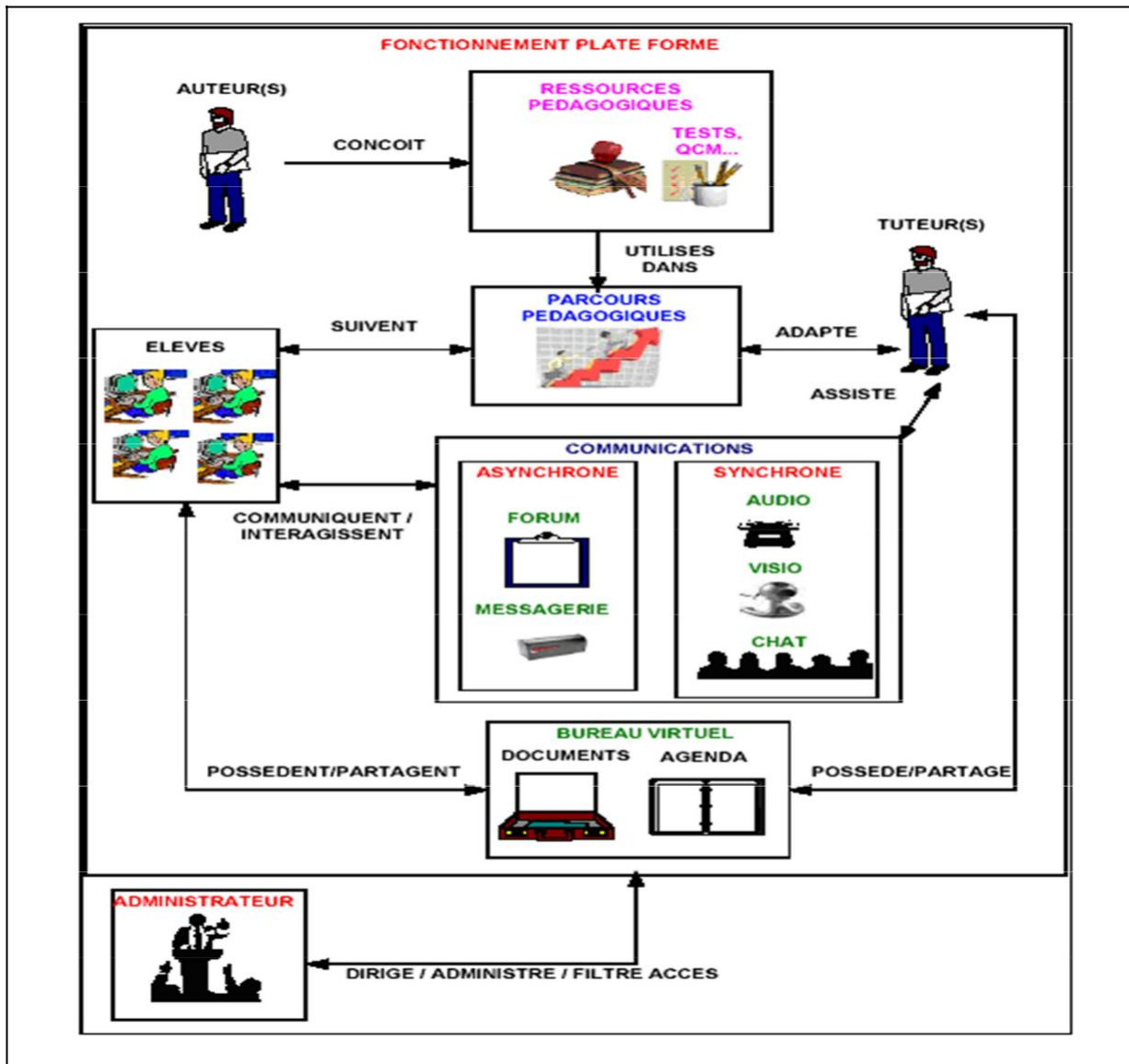


Figure III.2. Fonctionnement d'une plate forme d'e- Learning

### III.3.1. Acteur d'une plate forme e-learning:

#### III.3.1.1. Enseignant : [8]

L'enseignant crée des parcours pédagogiques types et individualisés de son enseignement, incorpore des ressources pédagogiques multimédias et effectue un suivi des activités des apprenants.

#### III.3.1.2. Apprenant : [8]

L'apprenant consulte en ligne ou télécharge les contenus pédagogiques qui lui sont recommandés, organise et a une vue de l'évolution de son travail, effectue des exercices,

s'auto évalue et transmet des devoirs à corriger. Enseignants et apprenants communiquent individuellement ou en groupe, créent des thèmes de discussion et collaborent à l'élaboration de documents en commun.

**III.3.1.3. Auteur : [8]**

L'auteur est en charge de créer du matériel pédagogique destiné aux enseignants. Mais ce rôle peut être également endossé par l'enseignant ponctuellement pour remédier à une situation donnée, par exemple dans l'optique de la préparation d'un examen, créé du contenu complémentaire pour enrichir le support existant.

**III.3.1.4. Administrateur : [8]**

L'administrateur a pour rôle, d'installer les applications de la plateforme, d'assurer la maintenance et la gestion des accès au système.

**III.3.2. Exemples de plateformes d'e-learning : [8]**

Présentons brièvement quelques plateformes d'e-learning sélectionnées parmi un nombre important de dispositifs présents sur le marché de la formation en ligne.

**III.3.2.1. WebCT : [8]**

WebCT (Web Course Tools), est une plateforme développée par l'université canadienne British Columbia. Il s'agit d'un environnement d'enseignement/apprentissage destiné à développer et diffuser des téléCours et à faciliter l'intégration d'un ensemble d'outils de communication, d'évaluation, de gestion, etc. WebCT est l'une des plateformes commerciales de téléformation les plus vendues dans le monde. Elle est traduite en plusieurs langues et utilisée par des milliers d'institutions. Dans WebCT, chaque acteur humain (administrateur, concepteur de cours, tuteur et apprenant) dispose d'un espace de travail sous forme d'une interface (accessible via un mot de passe) regroupant un ensemble d'outils propres à son profil. Les principales fonctionnalités de WebCT sont regroupées dans le tableau suivant :

Espace administrateur	Espace enseignant	Espace apprenant
-Gestion d'utilisateurs (profils, données, ...) -Suivi du bon fonctionnement technique de la plate-forme	-Création de cours -Suivi des cours -Création des groupes et des forums de discussion -Consultation des notes et statistiques à l'aide d'un carnet de notes.	-Découverte des contenus -Lancement des activités -Communication/collaboration et soutien à l'apprentissage (calendrier, courriel, forums, messagerie instantanée).

**Tableau III.1. Fonctionnalités de WebCT**

III.3.2.2. Moodle : [8]

Moodle (Object-Oriented Dynamic Learning Environment) est une plate forme « open source » mise à disposition des utilisateurs librement suivant la licence GPL (General Purpose License). Cela signifie que Moodle bénéficie d'une *copyright*, mais que l'utilisateur (administrateur, enseignant, ...) dispose d'un certain nombre de libertés. Il a le droit de copier, d'utiliser et de modifier Moodle pour autant qu'il s'engage à mettre à disposition des autres le code source, à ne pas modifier ni supprimer la licence originale et les copyrights et à appliquer la même licence à tous les travaux dérivés. D'un point de vue technique, Moodle est basée sur du code PHP et fait appel à une base de données MySQL. La figure III.3 représente la page principale de Moodle. Les fonctionnalités principales de la plate forme Moodle sont classées en fonction des acteurs et sont résumées dans le tableau III.2.

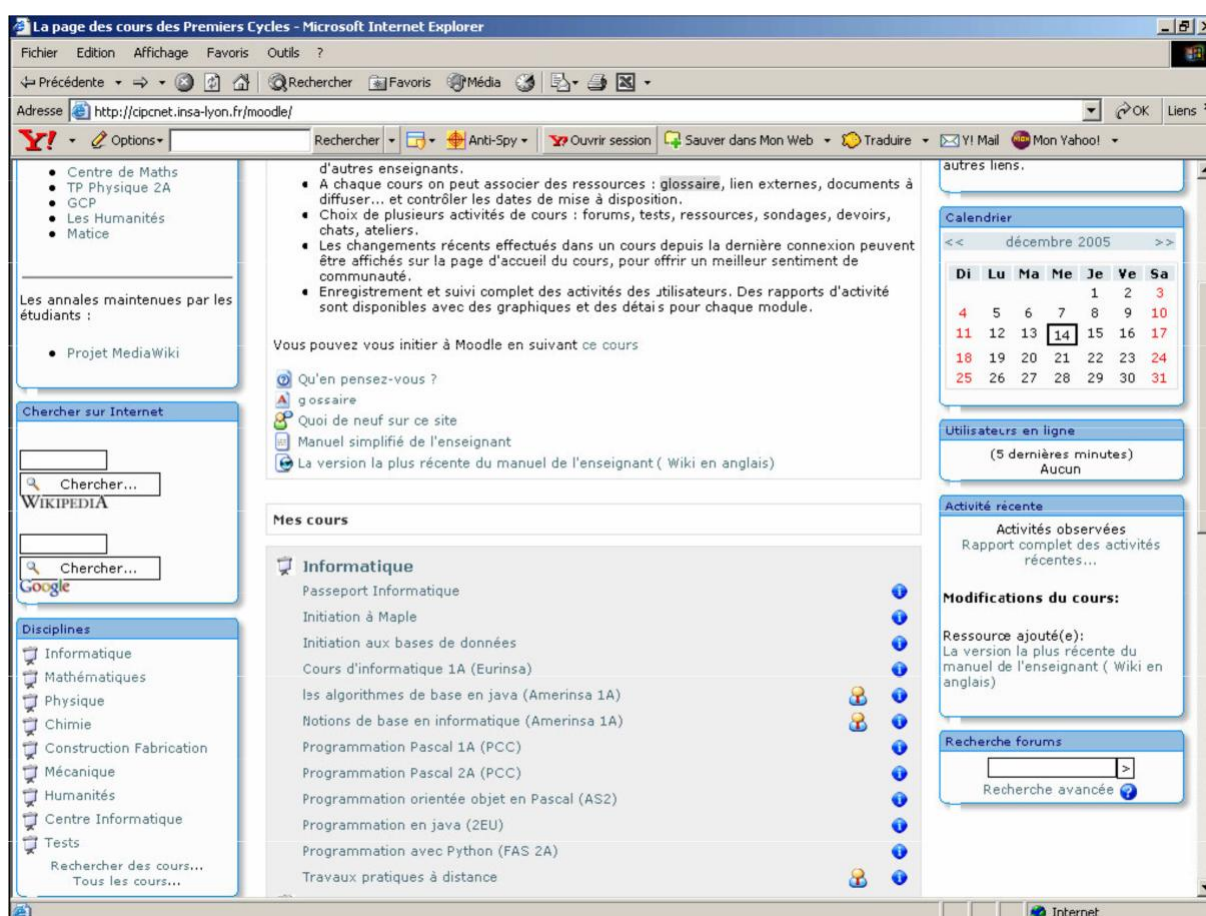


Figure III.3. Page principale de Moodle

Espace administrateur	Espace enseignant	Espace apprenant
-Inscription et suivi des apprenants -Gestion des dossiers des apprenants -Création de groupes d'utilisateurs avec restrictions paramétrables	-Travail en groupes -Tutorat (communication enseignant-apprenant synchrone et asynchrone) -Evaluation des apprenants	-Accès aux supports de cours -Espace privé -Communication asynchrone apprenant-apprenant (forums) -Communication synchrone apprenant-apprenant par messagerie instantanée

Tableau III.2. Fonctionnalités de Moodle

### III.4. Situation de l'e-Learning dans le monde : [8]

Les principaux pays actuellement exportateurs de services de formation par Internet sont : les États-Unis, la Nouvelle-Zélande, l'Australie et le Canada.

Aux États-Unis, les réalisations sont nombreuses et diversifiées. L'e-Learning croît à une vitesse vertigineuse. Au niveau universitaire, Virtual University Campus est le plus important portail d'éducation sur Internet. Cette université produit plus de 500 cours de qualité à faible coût.

En Australie, le programme national australien d'éducation et de formations flexibles est un plan stratégique sur cinq ans auquel sont affectés 22,4 Millions € par an. À cela s'ajoutent les participations des différentes provinces qui ont leurs propres financements dans ce domaine. Sur l'ensemble des étudiants des universités australiennes, 14% (soient 95.300) reçoivent une éducation à distance.

En Nouvelle-Zélande, le gouvernement a créé un comité de pilotage pour l'e-Learning. Ce comité, composé de neuf experts, donne des avis au gouvernement quant aux actions à mettre en œuvre pour faciliter, développer et coordonner l'enseignement à distance. Les exportations de services d'éducation en Nouvelle-Zélande sont estimées entre 380 et 560 Millions € par an.

Au Canada, l'éducation, incluant l'e-Learning, est une compétence provinciale. La majorité des dix provinces et des trois territoires ont identifié l'e-Learning comme une priorité. Une étude récente montre que 57% des 134 collèges et universités canadiens offrent des cours en ligne (entre 1 et 340 cours par établissement), soit un total de l'ordre de 3 000 cours.

En Algérie, un projet e-Learning est lancé récemment par le ministère de l'enseignement supérieur et de la recherche scientifique.

En résumé, tous les pays ont réagi rapidement à l'évolution de la formation ouverte et à distance en y affectant des crédits importants et en se dotant des structures nécessaires pour accompagner et organiser son développement.

### **III.5. Avantages et faiblesses de l'e-Learning : [8]**

#### **III.5.1. Avantages de l'e-Learning**

- La formation est ouverte à toute personne, quels qu'en soient son âge, son niveau d'instruction, sa catégorie socioprofessionnelle, etc.
- Assouplissement des contraintes géographiques et logistiques induisant une baisse du coût de la formation pour les apprenants.
- L'apprentissage est plus rapide et plus durable. En effet, des études montrent que les courbes d'apprentissage peuvent être améliorées par l'e-Learning et que la mémorisation des informations est souvent meilleure. Les temps d'apprentissage sont en général plus faibles qu'en présentiel (environ 0%5 plus faibles). Bien utilisé, l'e-Learning permet donc d'améliorer la qualité de la formation.
- L'e-Learning permet l'accès à de nouvelles compétences qui sont plus que jamais indispensables aux exigences de la vie moderne. Chacun peut se familiariser avec les nouvelles technologies comme l'ordinateur, les systèmes multimédias et l'Internet.
- L'apprentissage est personnalisé car l'apprenant peut choisir le temps à passer sur chaque module d'une formation en fonction de ses acquis. Il peut adapter le rythme du cours à son niveau (moins de stress, moins de frustration).
- L'e-Learning coûte globalement moins cher que le présentiel du fait qu'il élimine les coûts liés à la rémunération des professeurs, la gestion des salles de cours et les déplacements.
- L'apprenant est le centre de concentration et non plus le formateur. Donc, l'apprenant est incité à être un émetteur et de participer d'une manière significative à la formation et ne plus se limiter à être un récepteur d'informations et de savoirs comme c'est le cas des étudiants à la traditionnelle.
- Créer un sentiment de liberté et de confiance de l'apprenant en lui-même. Les sentiments d'intimité et de honte des collègues et du formateur en cas de faute par exemple se dissipent en e-Learning. L'apprenant étudie tout seul, face à son ordinateur et n'est pas observé des autres (excepté le cas de visioconférence).

#### **III.5.2. Faiblesses de l'e-Learning**

- L'absence physique de l'enseignant avec tout son poids d'émotions, d'autorité et d'expressivités humaines.
- Les problèmes techniques afférents au fonctionnement des systèmes de formation (perturbation du réseau de communications, pannes des ordinateurs, terminaux ou serveurs, attaques des documents électroniques de cours par des virus ou des pirates, etc.)

- L'accès à l'outil informatique est nécessaire. En effet, l'utilisation de l'outil informatique limite la diffusion de l'e-learning auprès d'une partie des collaborateurs. Ceci est un obstacle par exemple dans l'industrie.
- L'e-learning limite les interactions entre les individus. Certains mécanismes de communication ne peuvent pas être reproduits (langage du corps par exemple), alors qu'ils jouent un rôle important dans la diffusion du savoir.

**Conclusion :**

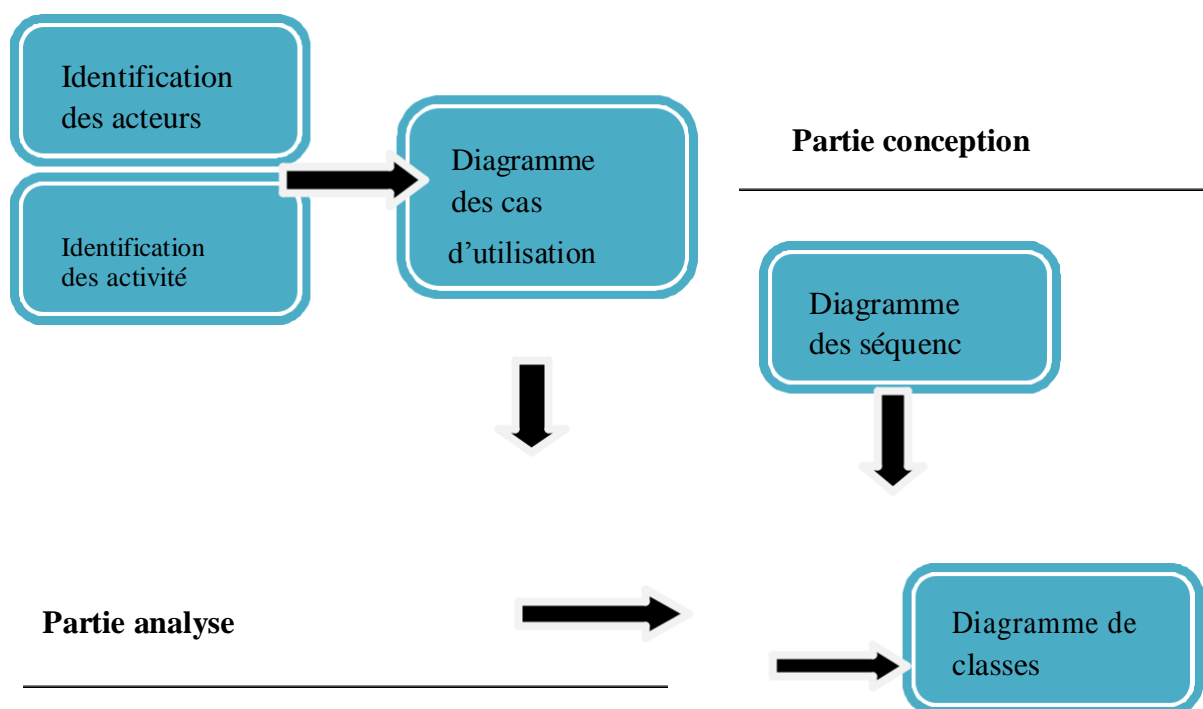
Au niveau de ce chapitre nous avons présenté quelques généralités relatives à l'EAO, EIAO et au domaine de e-Learning et ces plates-formes. Nous avons aussi énuméré quelques avantages et inconvénients d' e-Learning. Dans le chapitre suivant nous évoquerons l'élaboration d'un exemple.

**Introduction :**

Pour l'étude de l'exemple on utilise la méthode UML, la méthode qui se base sur deux parties l'analyse et la conception, et avant d'entamer la conception, il faut impérativement passer par la phase d'analyse qui permet d'identifier les différents acteurs qui intéressent avec le système ainsi que leurs besoins. Puis on passe à la phase de conception en s'appuyant sur les résultats de la phase d'analyse.

**La démarche de modélisation de l'application.**

Vu qu'UML ne propose pas de démarche de développement nous allons suivre la démarche suivante : analyse, conception et réalisation mais avant de commencer le processus d'analyse et conception nous allons présenter les objectifs et les fonctionnalités attendus de notre application.



**Figure IV.1.la démarche de modélisation de l'application**

**IV.1. Analyse :**

Cette phase comprend l'identification des besoins de système, les acteurs participants, leurs interactions avec le système ainsi que les cas d'utilisation.

**IV.1.1. Spécification des besoins :**

Notre projet consiste à concevoir une application adaptable et une base de données répartie et son implémentation d'une base de données centralisée vers une base de données distribuée sur plusieurs sites.

Notre cas d'étude s'appuie sur un exemple d'une plateforme Learning pour faculté génie électrique et d'informatique, ou nous avons choisie d'intégrer les données des quatre départements d'informatique, automatique, électronique et électrotechnique à un service commun de la faculté.

La problématique réside au niveau des données partagées entre les départements et le service commun et comment faire circuler ce flux d'information d'une manière transparente en vue des acteurs de la plateforme et comment assurer l'adaptabilité de cette application.

L'objectif de notre projet est de proposer une bonne solution à la problématique posée précédemment par une bonne conception et intégration d'une base de données répartie,

Donc pour la conception de la base de données de la faculté nous allons procéder d'une façon ascendante on suivant les étapes suivantes :

Etudier les besoins et les acteurs des sites locaux et établissement des schémas conceptuels locaux.

Etablissement du schéma d'intégration on utilisant l'approche ascendante.

Réalisation du schéma globale après intégration.

Traduction des schémas conceptuels obtenus en schémas relationnels logiques en appliquant les règles de passages UML –relationnel.

**IV.1.2. Identification des acteurs :**

Comme nous avons cité déjà notre système comporte plusieurs sites (sites locaux et un site global) donc nous allons identifier les acteurs de chaque site comme le monter le schéma suivant :

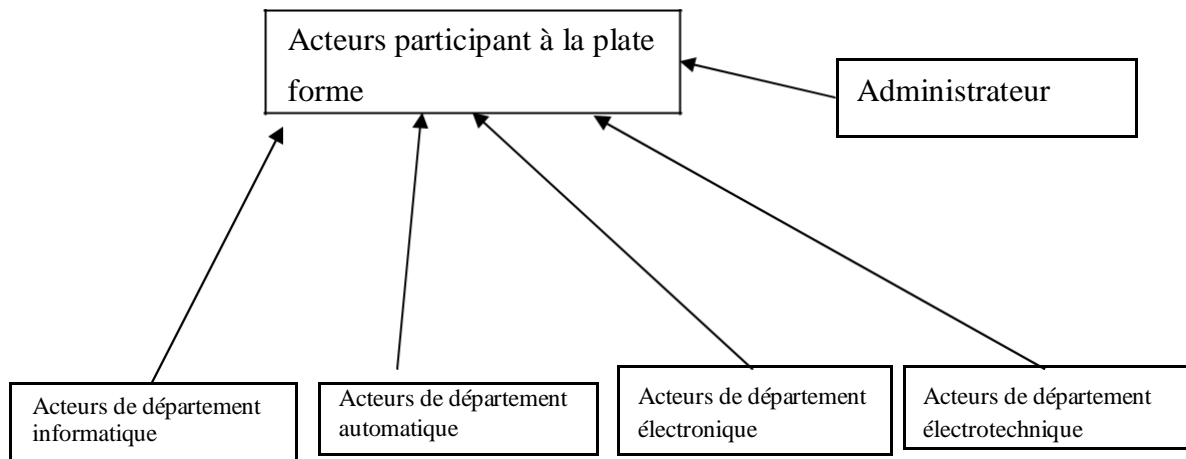


Figure IV.2. Les acteurs participant à notre système.

**IV.1.2.1. Identification des acteurs de chaque département (informatique, automatique, électronique, électrotechnique).**

- Les acteurs d'un département sont nombreux et ils sont tous concernés par la plate forme dans notre cas on se limite aux acteurs suivant :

Enseignant : est un acteur qui peut accéder à la plate forme pour avoir les informations et les applications spécifiques au département.

Etudiant : est un acteur concerné juste par les applications et les données spécifiques au département.

L'administrateur de département : c'est l'administrateur de site département.

**Administrateur :**

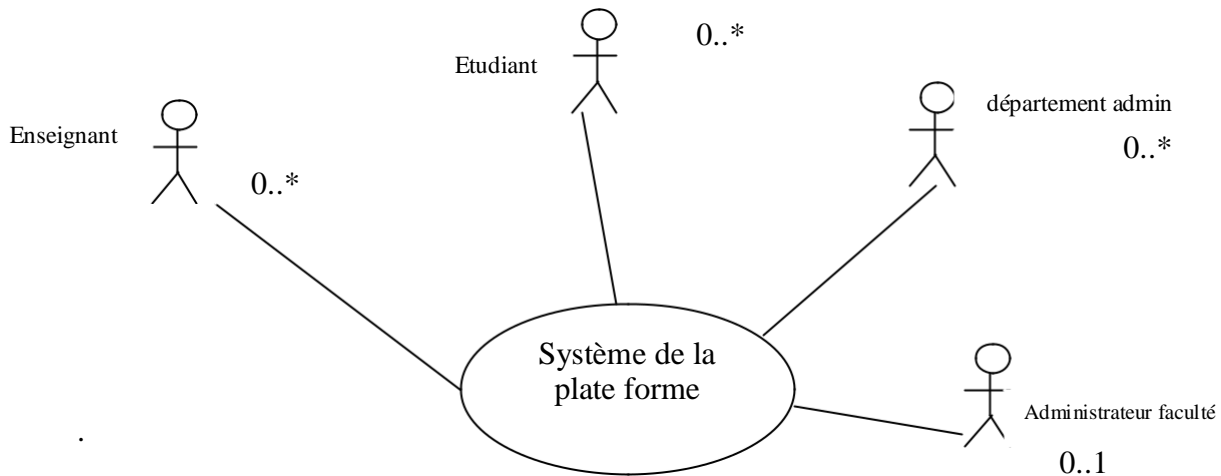
département admin.

Plate forme admin.

Pour l'administrateur du système réparti, l'administrateur de la plate forme s'occupe de la gestion de la BDD répartie.

**IV.1.3.Diagramme de contexte :**

Le diagramme suivant illustre le contexte de notre travail



**Figure IV.3.Diagramme de contexte**

**IV.1.4.Identification et représentation des cas d'utilisation :**

**Définition :**

Un cas d'utilisation désigne une fonctionnalité visible de l'extérieur du système dont on désire la fonctionnalité .un cas d'utilisation doit répondre à un besoin en définissant une partie du comportement attendu du système sans révéler sa structure interne.

Le tableau récapitulatif suivant présente les différentes taches réalisées par les acteurs de système sur les différents sites.

Site	Acteur	Cas d'utilisation	Base locale	Base répartie
		Espace pédagogique		
	Etudiant	T1 :S'inscrire à la plate forme. T2 :S'authentifier. T3 : Accéder a son espace . T4 : Modifier son profil. T5 : Consulter sa messagerie. T6 : Envoyer un message à un enseignant T7 : Envoyer un message à l'administration. T8 : Consulter son emploi de temps. T9 : Voir les filières et les spécialités. T10 : Voir les modules d'une spécialité. T11 : participer au forum. T12 : Consulter les cours, exercices et qcm.	OUI	
	Enseignant	T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11. Idem qu'étudiant. T12 : Envoyer et répondre aux messages d'Etudiants. T13 : Déposer des cours, exercices, qcm.	OUI	
	département Admin	T14 : Se connecter. T15 : Consulte ces informations. T16 : Gestion des enseignants (valider, supprimer). T17 : Gestion des étudiants (valider inscription, supprimer). T18 : Lire et répondre aux messages reçus. T19 : Gestion des formations.	OUI	<b>oui</b>
	Administrateur Faculté	T20 : Se connecter. T21 : Gestion de la plate forme. T22 : Répondre aux messages reçus.	oui	<b>oui</b>

**IV.1.4.1. Les diagrammes de cas d'utilisations :**

Après l'identification des différents acteurs ainsi que les cas d'utilisation qui sont mis en œuvre par ces acteurs, nous terminons la phase d'analyse par l'élaboration des différents diagrammes des cas d'utilisation associés à chaque acteur définie précédemment.

**II.4.1.1 Définition :**

**Cas d'utilisation:** (en anglais *use case*) permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Le format de représentation d'un cas d'utilisation est complètement libre mais UML propose un formalisme et des concepts issus de bonnes pratiques.

Le diagramme de cas d'utilisation permet de représenter visuellement la séquence d'actions réalisées par un système. Il est représenté par une boîte rectangulaire, produisant un résultat sur un acteur, appelé acteur principal, et ceci indépendamment de son fonctionnement interne.

**Relation entre cas d'utilisation :**

Il existe trois types de relation standards entre cas d'utilisation :

**Include:** le cas d'utilisation incorpore explicitement et de manière obligatoire un autre cas d'utilisation à l'endroit spécifié,

**Extend:** le cas d'utilisation incorpore implicitement de manière facultative un autre cas d'utilisation à l'endroit spécifié.

**Généralisation:** les cas d'utilisation descendants héritent des propriétés de leurs parents.

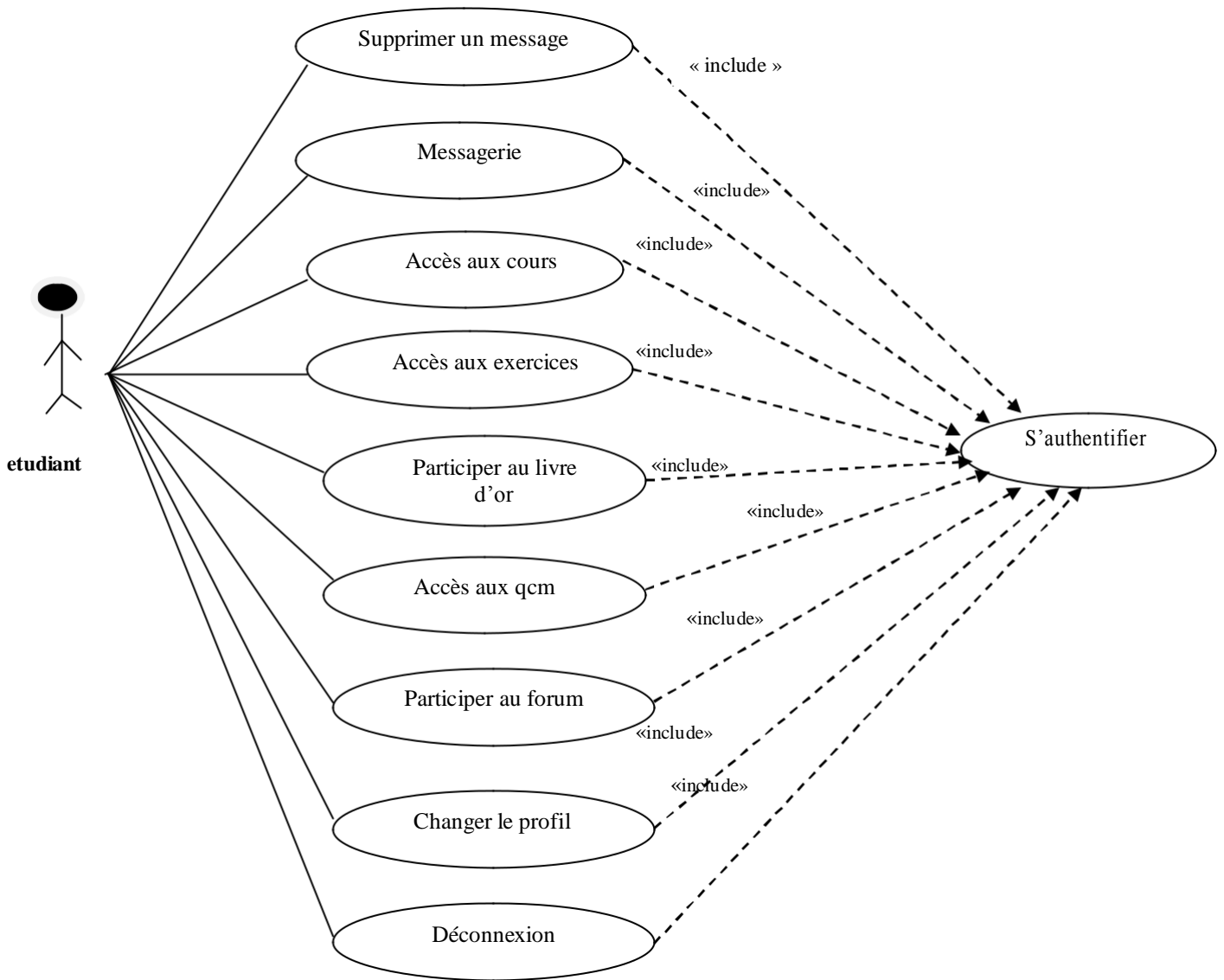


Figure .IV.4.Diagramme de cas d'utilisation « acteur : étudiant»

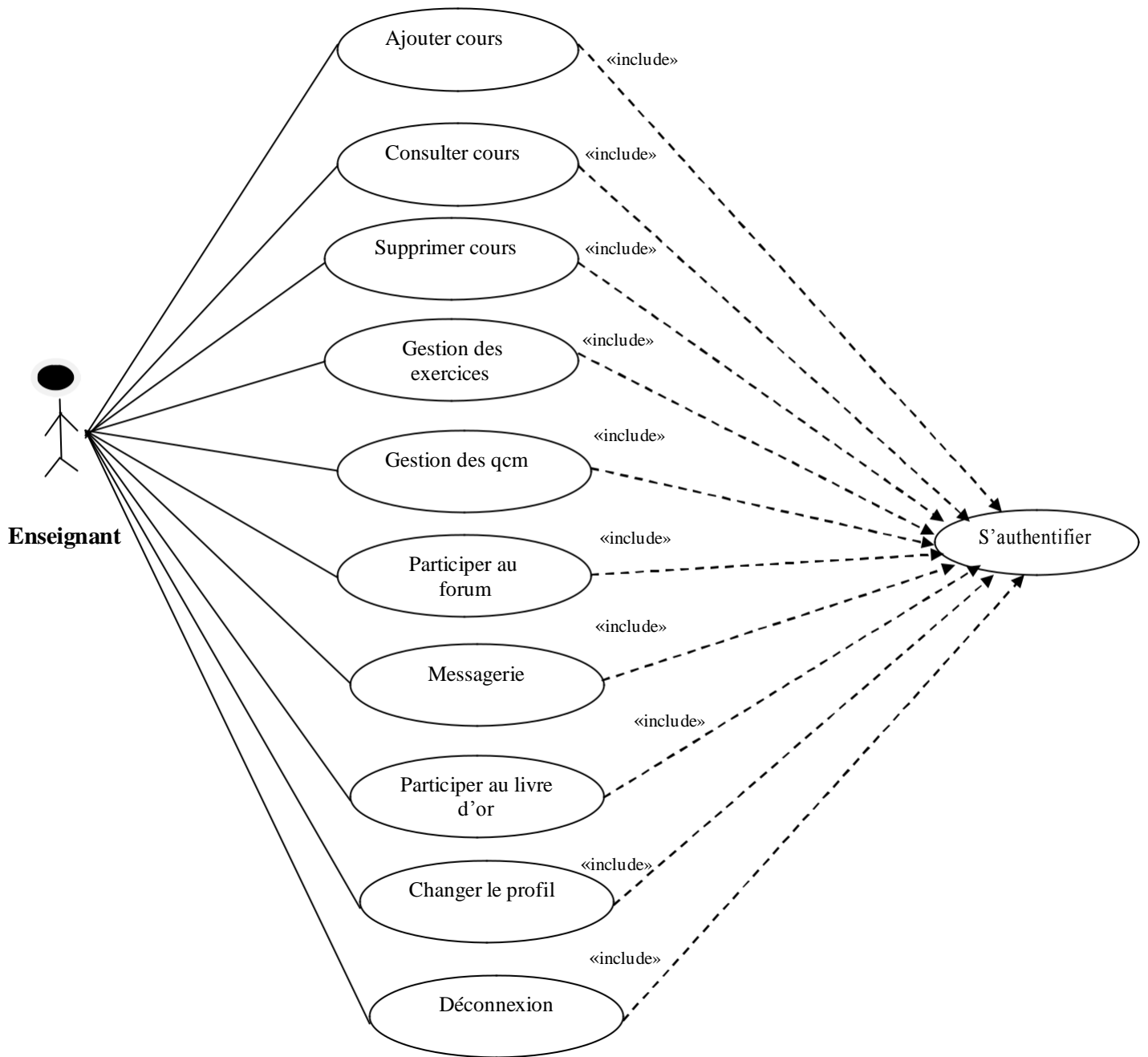


Figure.IV.5.Diagramme de cas d'utilisation « acteur : enseignant »

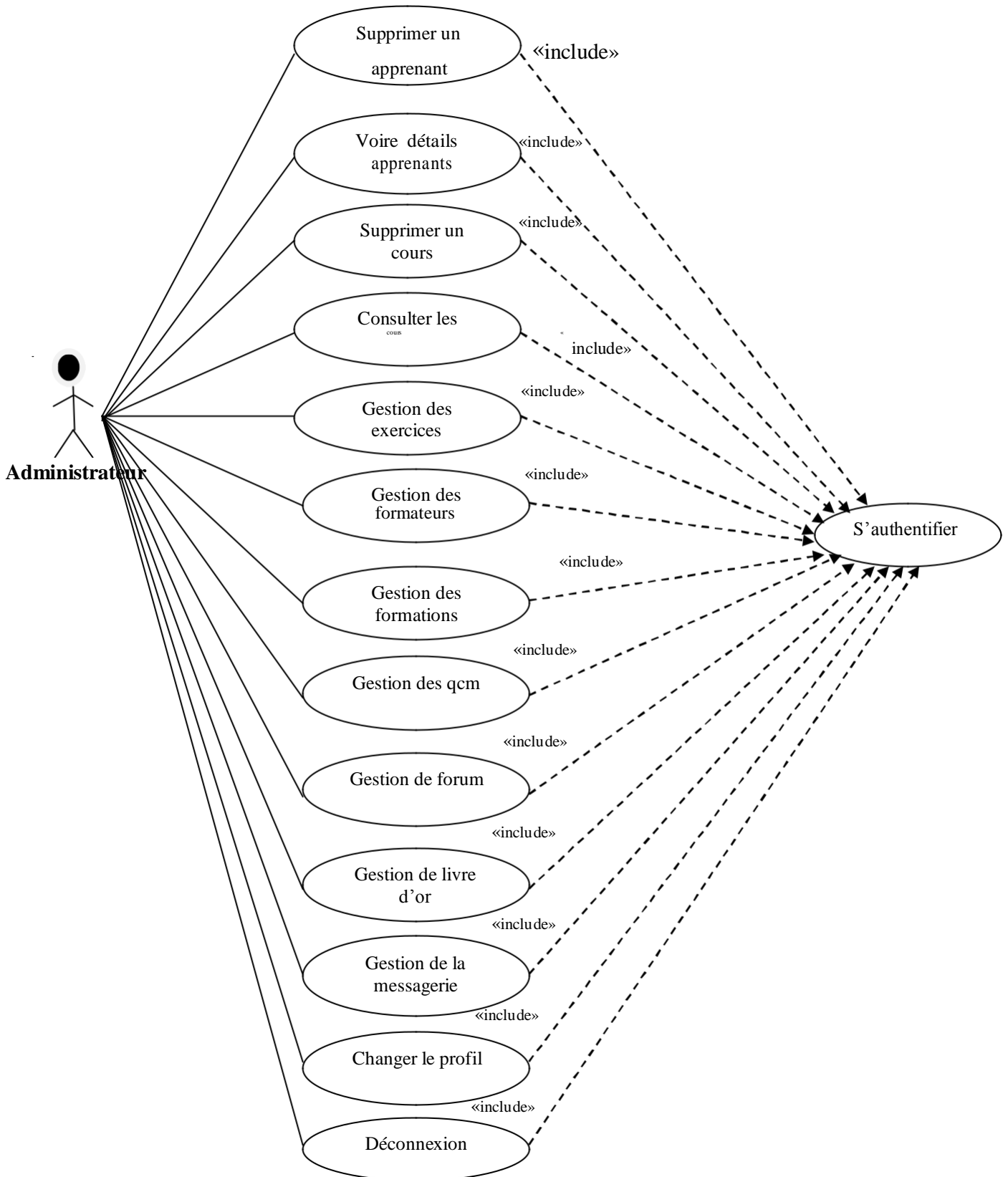


Figure.IV.6.Diagramme de cas d'utilisation « acteur : administrateur »

**IV.2.Conception :**

Après avoir analysé les différents acteurs participant, ainsi leur cas d'utilisation ; Nous allons passer dans ce qui concerne la phase de conception qui se base essentiellement sur les différents diagrammes de séquences et diagrammes de classe que nous allons présenter dans ce qui suit :

**Les diagrammes de séquence :** il présente la vue dynamique du système. L'objectif du diagramme de séquence est de présenter les interactions des objets en indiquant la chronologie des échanges. Cette représentation se réalise par cas d'utilisation.

**Les diagrammes de classes :** représentent la vue statique des objets qui leur donnent naissance à la BDD, intérêt majeur de modéliser les entités d'un système.

Autrement dit, ils expriment les relations existantes entre les pages client et serveur. Dans ce qui suit on représentera quelques diagrammes de séquences et de classes, correspondant aux cas d'utilisation déjà décrits.

**IV.2.1. Elaboration avec les diagrammes de séquence:**

Les diagrammes de séquence permettent de représenter les interactions entre objet selon un point de vue temporel.

- **Les objets interface :**

Un objet interface représente l'interface entre l'acteur et le système

Icône : 

- **Les objets entités :**

Les objets entités sont des objets décrits dans plus d'un cas d'utilisation.

Icône : 

- **Les objets de contrôle :**

Il représente les activités des processus du système, il dirige les activités des objets interfaces et entités.



Icône :

**IV.2.2.Présentation de quelques diagrammes de séquence correspondant aux cas d'utilisation déjà décrits :**

Nous avons choisie de présenter quelques exemples des diagrammes de Séquences correspondant aux cas d'utilisation présenté précédemment.

Comme nous l'avons cité précédemment notre démarcheprocède d'une manière ascendante donc nous allons schématisés quelques diagrammes de séquence de quelques taches déroulent au niveau des sites locaux.

Site locale	Acteurs	Diagrammes de séquences de cas d'utilisation
Site département « informatiques »	Etudiant	<ul style="list-style-type: none"> <li>- S'inscrire a la plate forme.</li> <li>- S'authentifier.</li> </ul>
	Enseignant	- envoyer un message à un étudiant ou administration
	admin_département	- Valider une inscription (enseignant, étudiant).

**IV.2.2.1.Cas d'utilisation « S'inscrire a la plate forme » :**

Il contient les objets suivants :

Ce cas d'utilisation contient les objets suivant :

**Les objets de type interface :**

- Page principale.
- Formulaire d'inscription.
- Page de confirmation.

**Les objets de type entité :**

- Inscription

**Les objets de type contrôle :**

- Enregistrer dans la base de données.

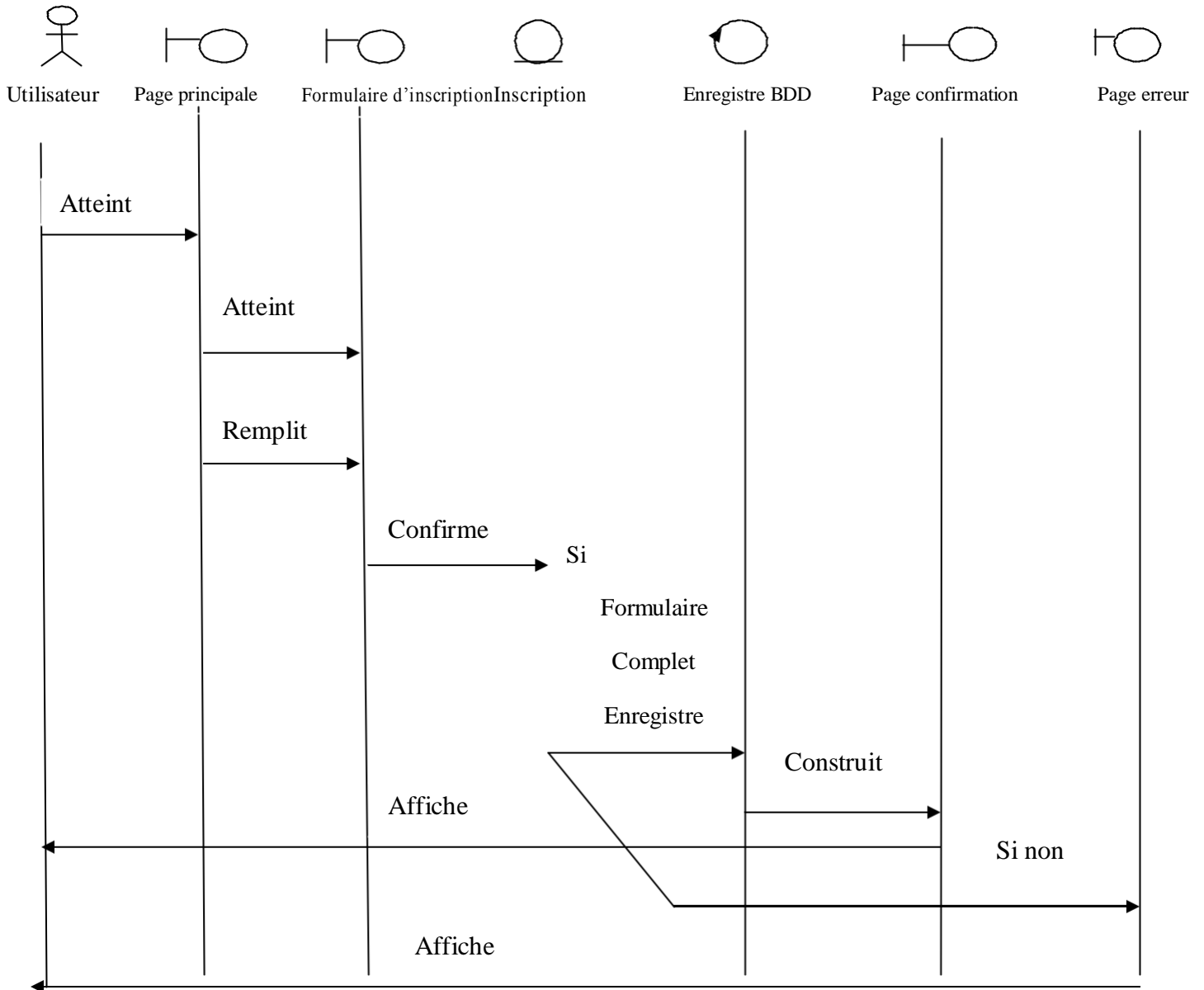


Figure.IV.7.Diagramme de séquence pour le cas d'utilisation <<s'inscrire à la plate forme>>

1. L'utilisateur atteint la page principale (page d'accueil) de la plate forme en saisit l'URL.
2. L'utilisateur clique sur le lien inscription et le système lui affiche le formulaire d'inscription.
3. L'utilisateur remplit le formulaire d'inscription.
4. L'utilisateur confirme son inscription avec le bouton valider.
5. Le système vérifie si le formulaire est complet, il enregistre l'inscription dans la base de données.
6. Le système construit la page de confirmation.

7. Le système affiche la page de confirmation
8. Si non il redirige vers la page d'erreur.
9. Le système affiche un message d'erreur (page d'erreur).

**IV.2.2.2.Cas d'utilisation « s'authentifier » :**

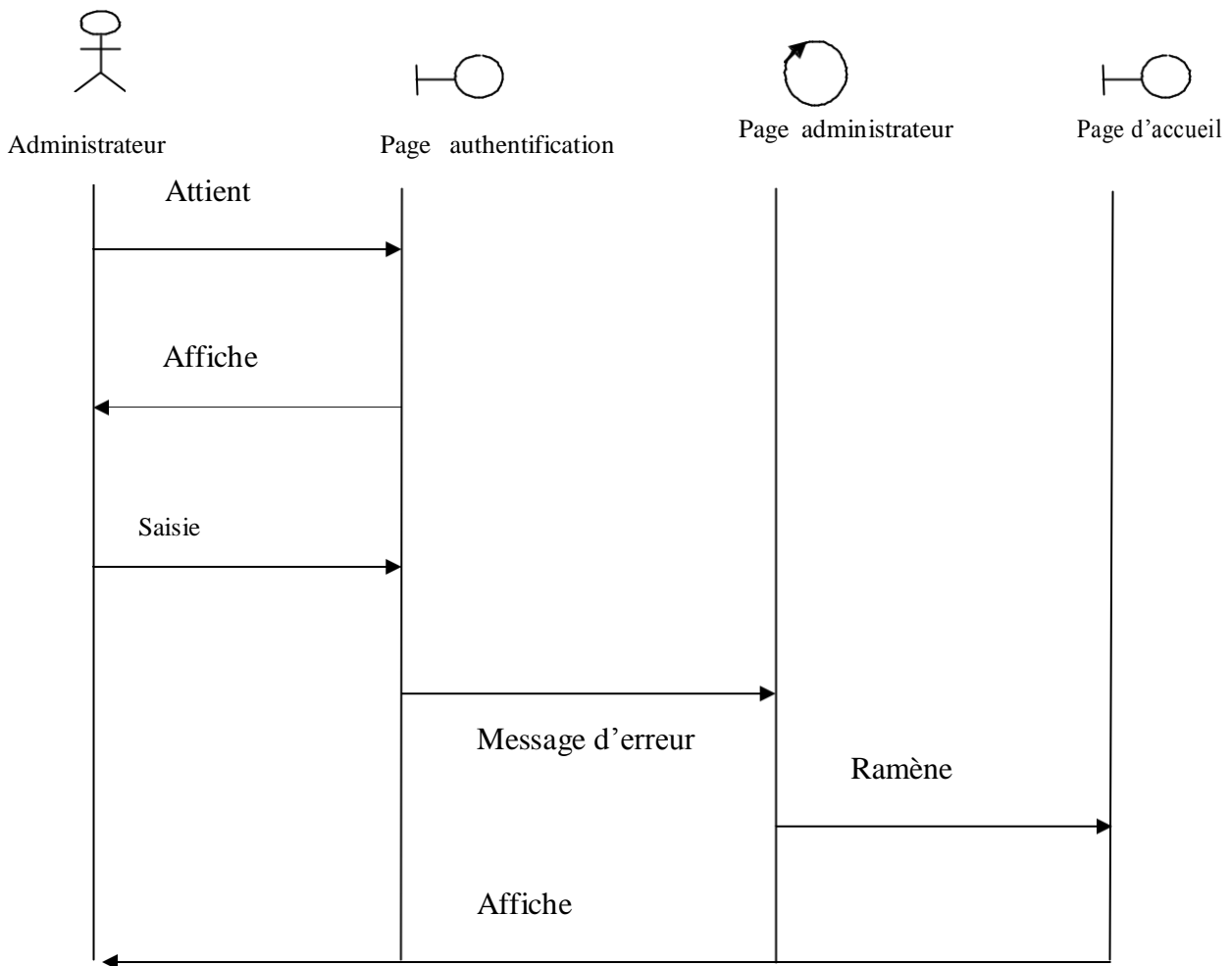
Il contient les objets suivants :

**Objet interface :**

- Page principale
- Page d'authentification
- Espace personnel

**Objet contrôle :**

- La recherche dans la base de données.



**Figure.IV.8: Diagramme de séquence de cas d'utilisation :<<s'authentifier>>**

1. L'administrateur atteint la page d'authentification .
2. Le système l'affiche la page d'authentification
3. L'administrateur saisit ses coordonnées.
4. Le système vérifie les informations saisies.
5. Le système construit l'interface d'administrateur si les informations sont correctes si non il affiche un message d'erreur.

**IV.2.2.3.Cas d'utilisation « Envoyer un message à un étudiant ou enseignant ou administration » :**

*Il contient les objets suivants :*

***Objet interface :***

- Page messagerie
- Page nouveau message
- Page de confirmation

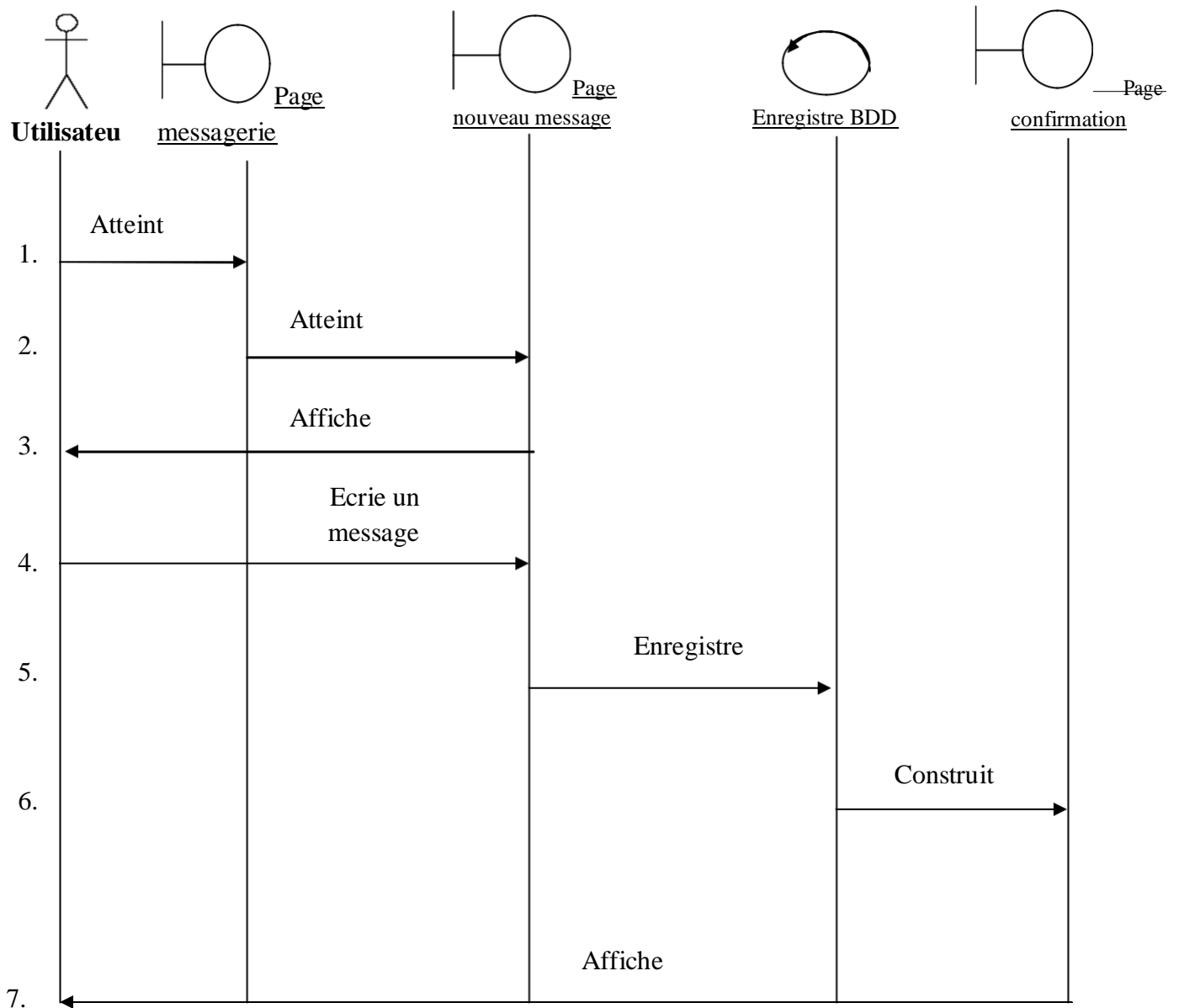
***Objet contrôle :***

- Enregistrer dans la base de données

***Objet entité:***

- Message

***Remarque :*** dans ce diagramme l'utilisateur c'est étudiant et enseignant.



**Figure IV.9 : Diagramme de séquence de cas d'utilisation « envoyer un message à un étudiant ou enseignant ou administration »**

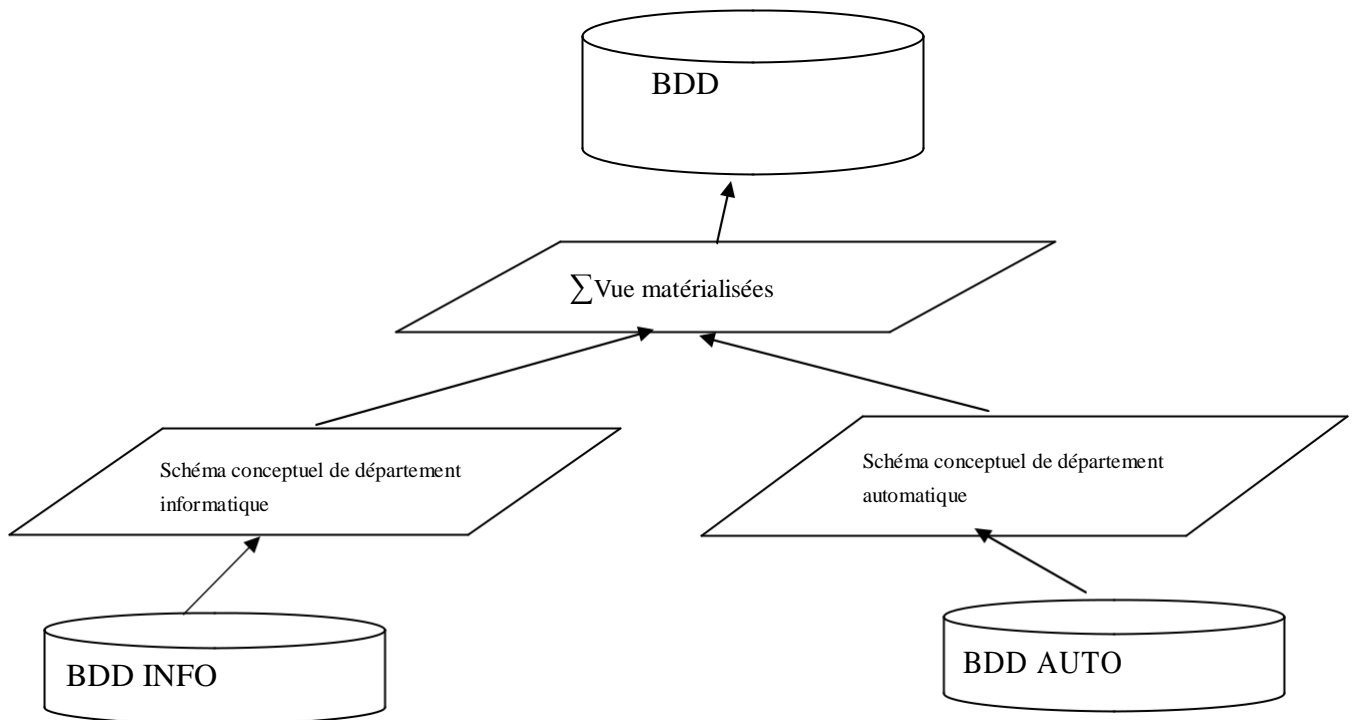
1. L'utilisateur atteint la page de messagerie après l'étape d'authentification
2. L'utilisateur clique sur le lien nouveau message et le système lui affiche la page de nouveau message
3. L'utilisateur saisit le message et l'adresse de de stinataire et clique sur le bouton envoyer
4. Le système enregistre le message dans la base de données
5. Le système construit la page de confirmation
6. Le système affiche la page de confirmation de l'envoi de message.

**IV.2.3.Diagramme de classe :**

**Définition:** un diagramme de classe est la vue statique des objets qui donnent naissance à la base de données, leur intérêt majeur est de modéliser les entités d'un système.

Autrement dit, ils expriment les relations existantes entre les pages client et serveur.

Nous allons présenter les schémas conceptuels locaux des bases locales, et le schéma conceptuel global de la base de données comme le montre la figure suivante.



**Figure IV.10:** *Plan de représentation des schémas conceptuels.*

**IV.2.3.1.diagramme du schéma logique du site département**

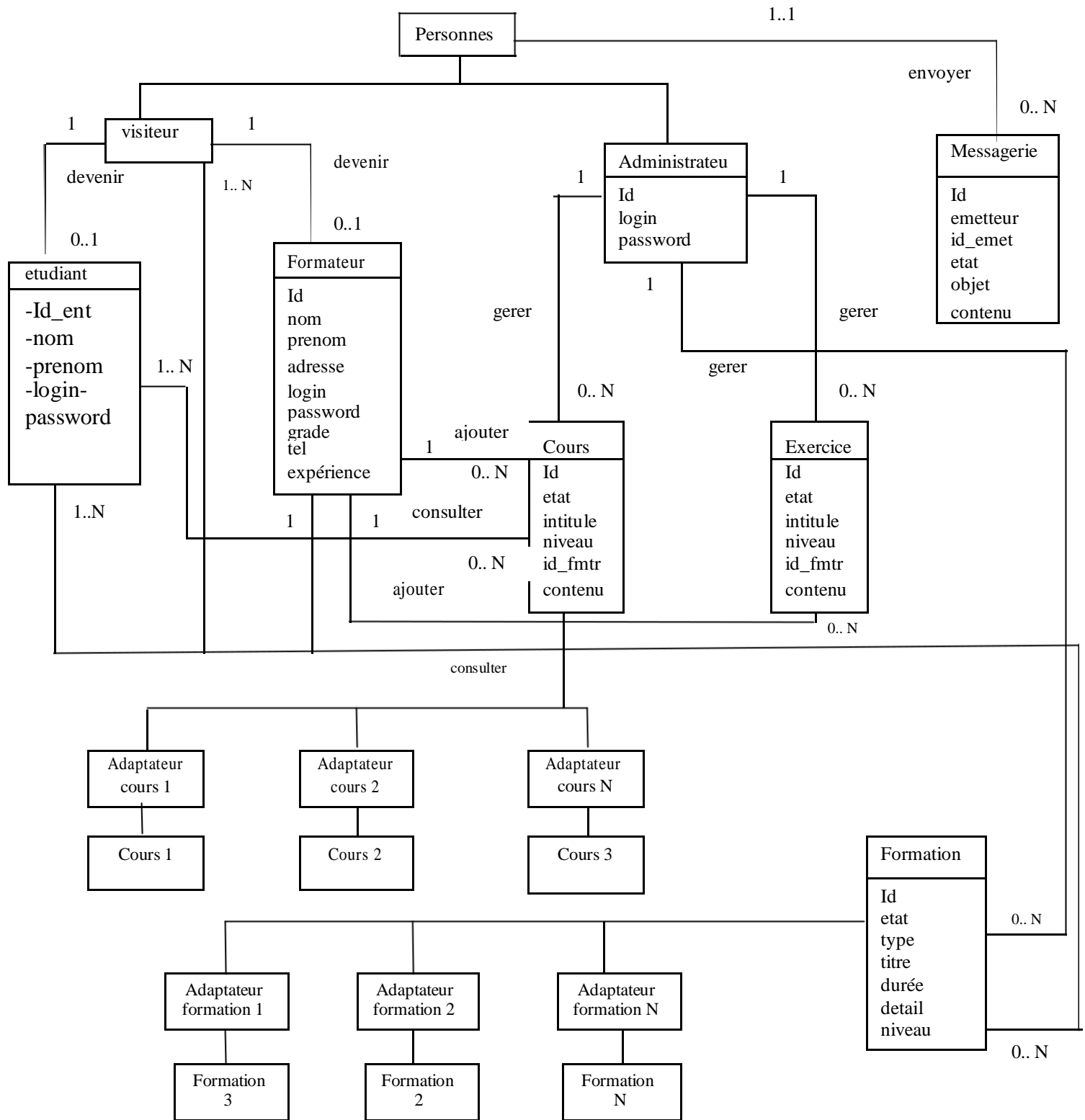


Figure IV.11. diagramme du schéma logique du site département.

IV.2.3.2. Diagramme du schéma logique de la base faculté :

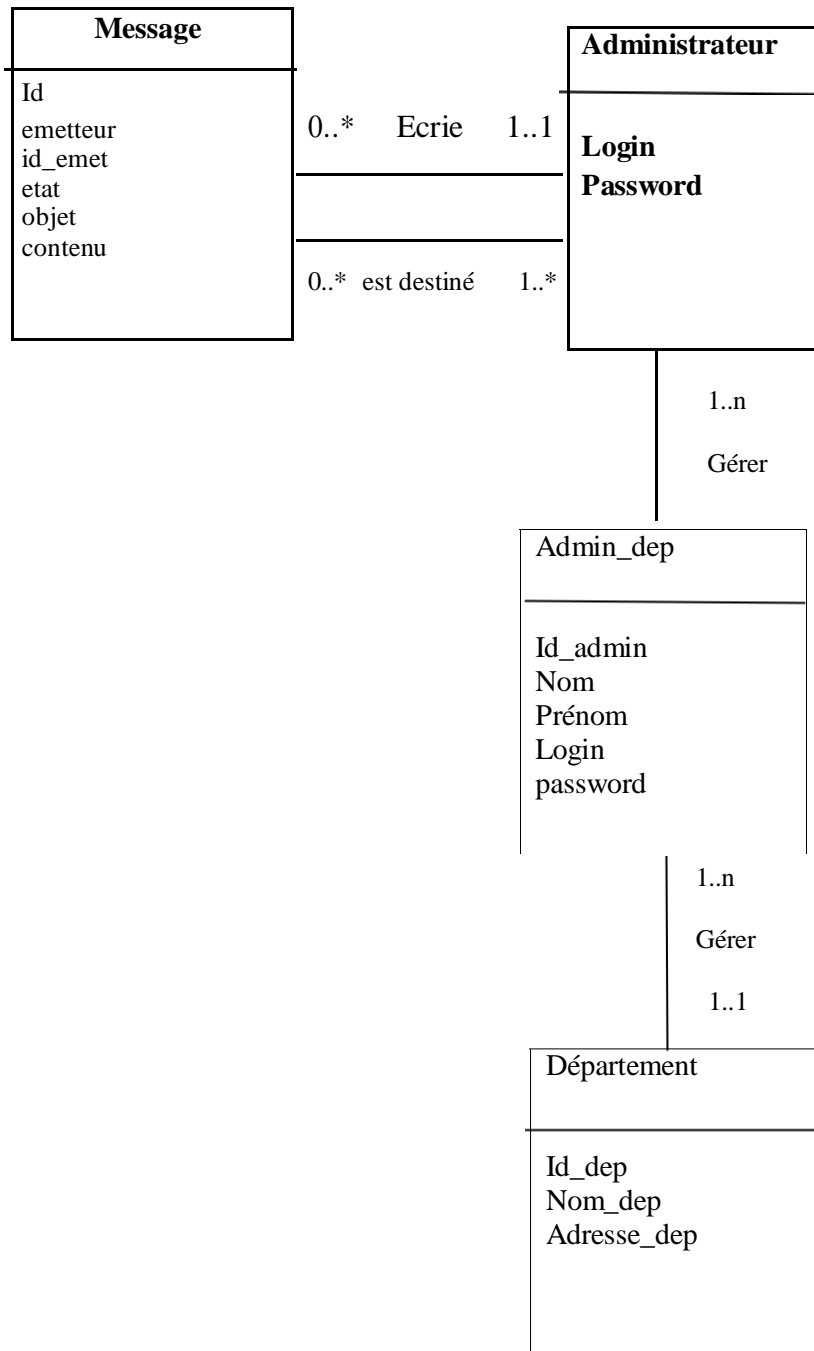


Figure IV.12. Diagramme du schéma logique de la plate forme

**IV.2.3.3.Exemple de modèles logiques de site de département :**

**Etudiant** (id\_etud, nom, prénom, date\_naiss, adresse, mail, Tel, login, pwd, id\_group)

**Administrateur** (id, login, password).

**Cours** (id, état, intitulé, nom\_fmtr, prénom\_fmtr, formation, contenu).

**Exercice** (id, état, niveau, nom\_fmtr, prénom\_fmtr, formation, contenu).

**Formateur** (id, nom, prénom, adresse, login, password, pays, date\_de\_naissance, sexe, grade, num\_tel, expérience).

**Livre\_dor** (id, date, heure, pseudo, message, statut)

**Messagerie\_admin** (id, emetteur, nom\_emet, prénom\_emet, état, bjet, contenu).

**Messagerie\_app** (id\_app, emetteur, nom\_emet, prénom\_emet, état, bjet, contenu, id).

**Messagerie\_fmtr** (id\_fmtr, emetteur, nom\_emet, prénom\_emet, état, bjet, contenu, id).

**Forum\_rep** (id, id\_sujet, nom\_emet, prénom\_emet, rep).

**Forum\_suj** (id, suj, nom\_emet, prenom\_emet).

**Conclusion :**

Dans ce chapitre j'ai présenté le processus de conception de mon application en deux niveaux, le niveau applicatif et le niveau de données. En premier lieu, j'ai commencé l'analyse et la conception par le niveau applicatif qui concerne les fonctionnalités et les traitements de l'application, ensuite j'ai passé au niveau de données qui m'a permis d'avoir le modèle logique de la base de données.

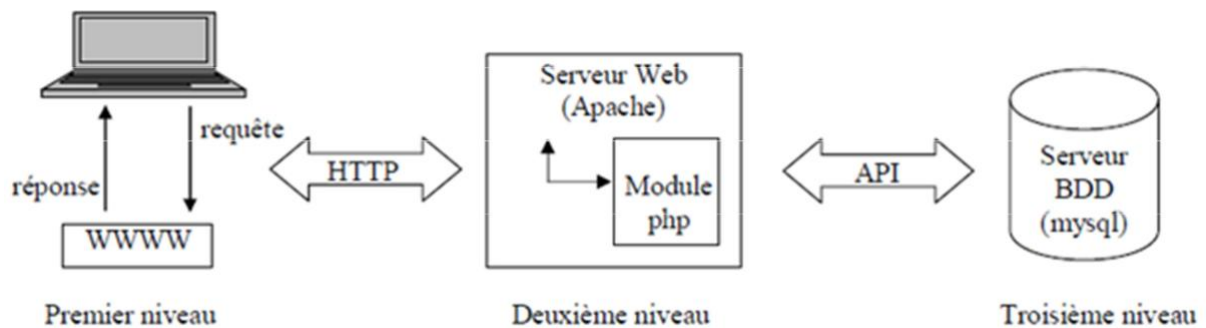
Le chapitre suivant sera consacré à la partie réalisation de notre plateforme ainsi que la différentes fonctionnalités dont elle dispose.

**Introduction :**

Après avoir présenté dans le chapitre précédent les différentes étapes d’analyse et de conception, nous allons présenter dans ce dernier chapitre l’environnement de développement, les outils qui ont servi à la réalisation de notre application, et nous terminerons par la présentation de ses fonctionnalités à travers ses différentes interfaces. Plusieurs solutions existent pour le développement d’une application web dynamique, nous avons opté pour la solution PHP/MySQL pour réaliser notre application.

**V.1.Architecture de l’application :**

La plate forme comporte trois niveaux, analogue à l ’architecture trois tiers du modèle client serveur. L’architecture de notre application peut être représentée graphiquement comme suit :



**Figure.V.1.L’architecture de déploiement de La plate forme**

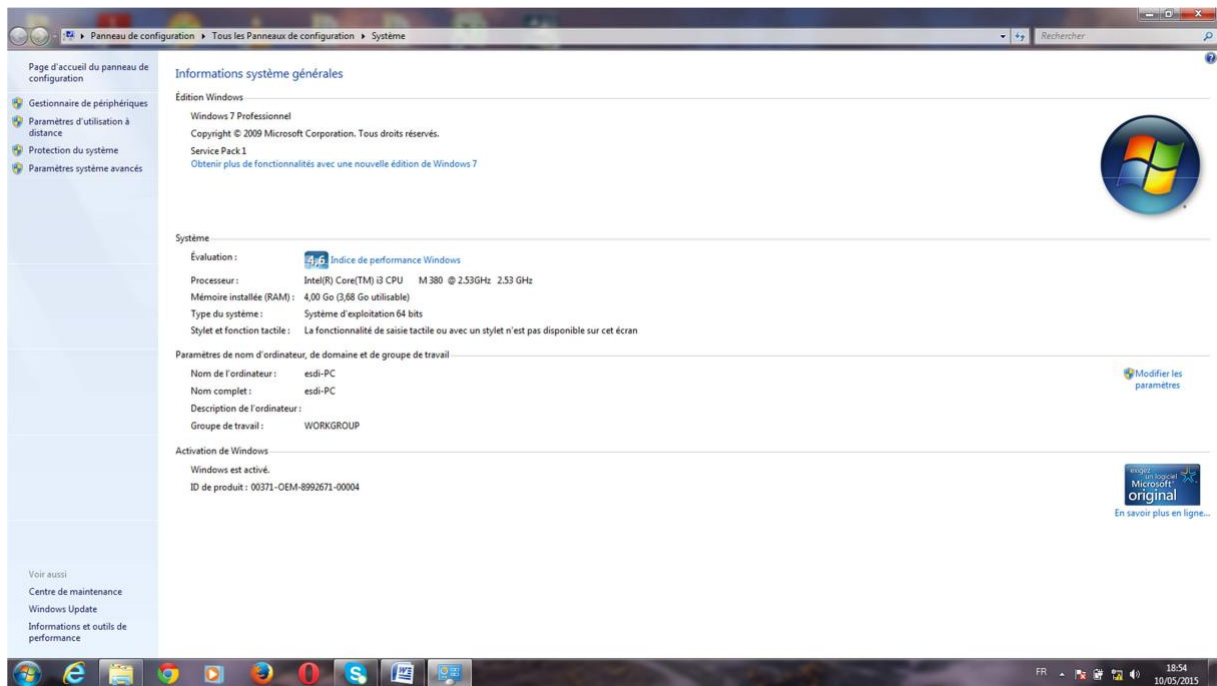
**Premier niveau :** il comprend le navigateur qui interprète et affiche les interfaces utilisateur relatives aux différents services de l’application ;

**Deuxième niveau :** le deuxième niveau est constitué d’un serveur Web Apache doté de module PHP (l’interpréteur PHP est installé comme module Apache) ;

**Troisième niveau :** le troisième niveau est composé d’un serveur de base de données (MYSQL) performant souple et disposant d’un jeu de commande SQL large.

**V.2.Environnement logiciel et matériel :**

Pour la réalisation de notre travail, nous avons utilisé un micro-ordinateur portable, ces caractéristiques sont :



**Figure.V.2.caractéristiques matérielles et logicielles**

Pour la réalisation de notre application, nous avons fait appel à plusieurs outils (logiciel).

- **Système d'exploitation :** Windows 7.
- **Serveur Web :** Serveur apache offert par WAMP Serveur.
- **Serveur de base de données :**
  - Un serveur MySQL.
  - Une interface graphique PHPMysqlAdmin pour manipuler les bases de données .
- **Langages utilisés :**
  - Pour la partie statique de la plate forme : le langage HTML.
  - Pour la partie dynamique de la plate forme : le langage PHP.
  - Langage de requête MySQL pour interroger la base dedonnées .

### ***V.3.L'environnement de développement :***

#### **V.3.1. Langages de programmation : [19]**

Pour satisfaire les besoins de l'application à réaliser, nous avons utilisé les langages suivants :

- ✚ **Le langage HTML :** est un langage de balisage pour la création des pages statiques de l'application qui constitueront l'interface utilisateur.
- ✚ **Le langage de requête SQL :** pour l'élaboration des requête d'interrogation deal base de données.

- ☛ Le langage de script côté client java *script* : pour les contrôles des informations au niveau client.
- ☛ *Le langage de script PHP* : pour la partie dynamique du site. Il constitue le langage pilier de l'application, c'est-à-dire, que c'est sur lui que reposent le dynamisme de l'application, c'est la raison pour laquelle nous allons mieux le détailler dans ce qui suit :

### V.3.1.1. PHP [18] :

PHP (Hypertext Preprocessor , Pré processeur Hypertexte PHP ) est un langage de script côté serveur, c'est à dire que les scripts sont exécutés avant que la page ne soit envoyée au navigateur, Ses principaux avantages sont :

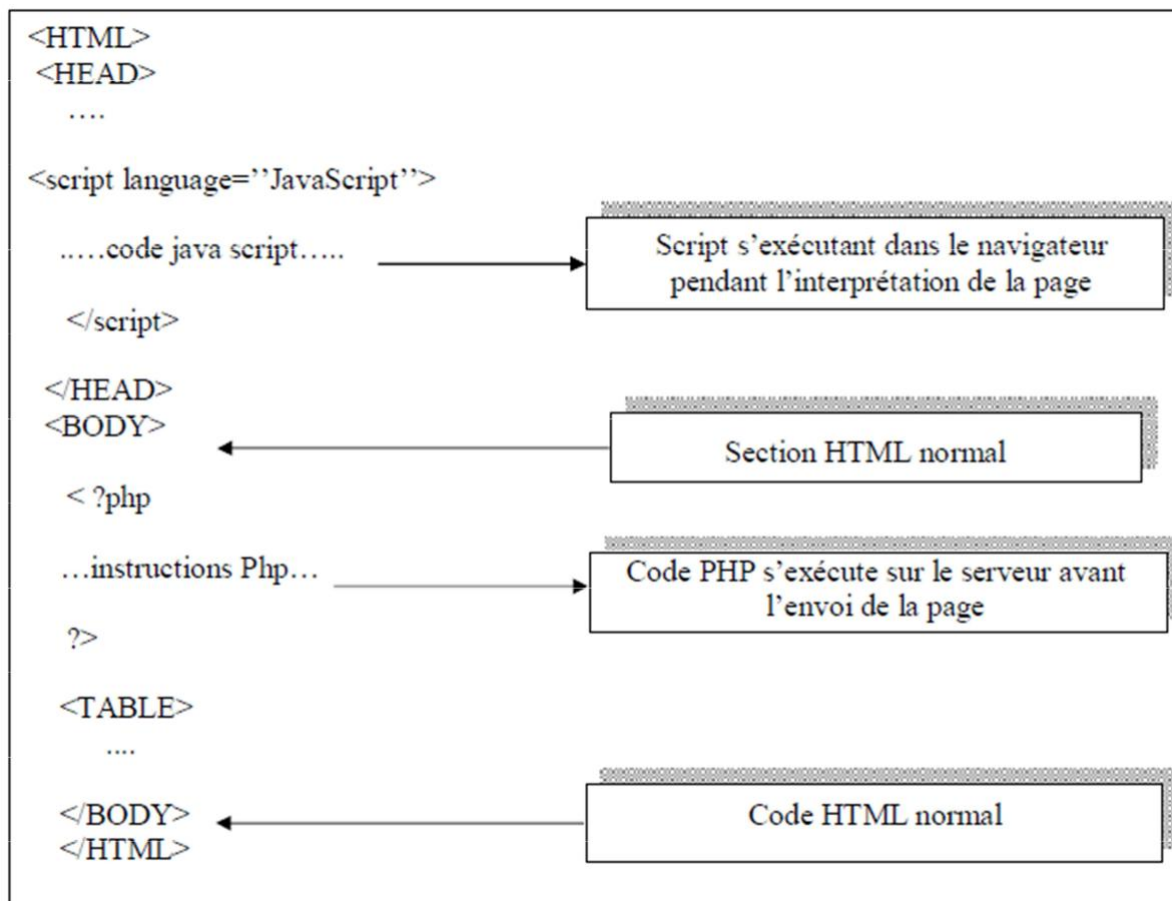
- ☛ PHP est facile à utiliser, il permet de créer avec un minimum d'efforts des pages Web dynamiques destinées aux applications Internet multimédia et de E-commerce.
- ☛ PHP est multi plate forme, il existe pour les différentes versions de Windows, Unix et Linux, ainsi que pour de nombreux serveurs Web dont APACHE et IIS.
- ☛ PHP est libre, donc « OPEN SOURCE », on peut ajouter de nouvelles fonctionnalités sans attendre une nouvelle version.
- ☛ PHP utilise le moteur de scripts Zend, qui améliore la rapidité de traitement
- ☛ PHP a été conçu pour fonctionner sur le Web, la connexion et l'interrogation d'une base de données sont extrêmement simple (peuvent être accomplies en deux ou trois lignes de code).
- ☛ La plus grande qualité et le plus important avantage du langage PHP est le support d'un grand nombre de bases de données (plus de 20). PHP possède des fonctions ODBC, permettant de s'interfacer de manière conviviale à toute base de données ODBC.
- ☛ Avec plus de 1200 fonctions utilisables dans des applications très variées, PHP couvre pratiquement tous les domaines en rapport avec les applications Web.

### V.3.1.2. Structure d'une page PHP :

Le code PHP fait partie intégrante de la page HTML, il est inclus entre :

- < ? PHP .....instruction..... ? >
- < ? php .....instruction..... ? >
- < ? .....instruction..... ? >
- < % .....instruction..... % >
- < script langage= « PHP » >  
.....instruction.....  
< / script >

Ce code permet au navigateur de passer en mode PHP.



Il est à noter la différence avec :

- ✚ Les autres langages de scripts coté serveur tels que le Perl ou les servelets Java, qui doivent écrire un programme avec de nombreuses lignes de commande afin d'afficher une page HTML, alors que on écrit une page HTML avec du code PHP inclus à l'intérieur.
- ✚ Les autres langages de script coté client comme leJava script sont exécutées par le navigateur, alors que le code Php est exécuté sur el serveur. Si on a un script Php sur un serveur, le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat.

### V.3.1.3. Fonctionnement du PHP [18] :

Le serveur Web reconnaît à l'extension des fichiers , différente de celle des pages HTML simples, si le document appelé par le client comporte du code PHP.

- ✚ Le serveur Web lance l'interpréteur PHP ;
- ✚ L'interpréteur PHP traduit le document demandé et exécute le code source de la page ;

- Les commandes figurant dans la page sont interprétées et le résultat prend la forme d'une page HTML publiée à la place du code source dans le même document ; La
- page modifiée est envoyée au client pour y être affichée par le navigateur.

De cette façon la page Web est créée dynamiquement, c'est à dire au moment même où le client y accède. Cela permet donc de modifier la page avant qu'elle ne soit envoyée au client en fonction de dialogue avec l'utilisateur.

### **V.3.2. Serveur Web Apache :[9]**

Le logiciel apache est un serveur http en Open source, c'est le fruit de travail d'un groupe volontaire « **the Apache Groupe** », la première version est apparue en 1995, par la suite des centaines d'utilisateurs ont contribué à son amélioration, c'est le serveur Web le plus utilisé au monde actuellement, car il englobe plusieurs avantages dont :

- C'est un serveur gratuit (peut être téléchargé à partir du site du groupe Apache à l'adresse '<http://www.apache.org>').
- Un niveau de performance élevé pour des besoins matériels modestes.
- Extensible, modulaire et configurable.
- Robuste.
- Très portable contrairement à IIS (Internet Information Services) de Microsoft qui tourne seulement sous Windows, le serveur Web apache dispose d'une version pour chaque plate forme (Linux, Windows,...).

### **V.3.3. Serveur de base de données MySQL [16] :**

#### **V.3.3.1. MySQL :**

- Est un véritable serveur de bases de données SQL multiutilisateur et multitraitement. Cela permet d'établir des connexions rapides et d'utiliser la même mémoire cache pour plusieurs requêtes.
- MySQL est basé sur une bibliothèque de gestion de donnée éprouvée depuis de nombreuses années et faisant appel à des index d'arbres binaires. Grâce à cela, le cœur du système peut afficher une performance remarquable, tout particulièrement dans les accès indexés.

**Les principaux objectifs de MYSQL sont :**

1. la rapidité.
2. la robustesse.
3. la facilité d'utilisation.

### V.3.3.2.Fonctionnalités du MySQL :[16]

La liste suivante décrit quelques fonctionnalités importantes de MySQL :

- ☛ **Multitraitement** : MySQL est multitraitement en utilisant les threads du noyau. Il peut utiliser plusieurs CPU.
- ☛ **Langues** : Le serveur peut fournir au client les messages d'erreurs en plusieurs langues.
- ☛ **APIs** : Les applications de bases de données MySQL peuvent être écrites en C, C++, JAVA, PHP, PYTHON et TCL.
- ☛ **Multi plateformes** : Prise en charge de plus de 20 plates-formes de système d'exploitation Win95/98, NT UNIX et LINUX.
- ☛ **Tables** : MySQL stocke chaque table sous forme de fichier distinct dans le répertoire de la base de données. La taille maximale d'une table comprise entre 4Go est la taille maximale de fichier acceptée par le système d'exploitation. Le mélange des tables de différentes bases de données est supporté dans une même requête.
- ☛ **Système de droits flexibles et sécurisé** : Système de droits flexible et sécurisé de mots de passe, et qui autorise une vérification faite sur l'hôte : le serveur détermine l'identité du client grâce à l'hôte depuis lequel il se connecte et le nom d'utilisateur qu'il spécifie. Puis, l'associer avec les droits d'utilisation des commandes « **select** », « **insert** », « **update** » et « **delete** » sur cette base (le serveur vérifie chaque requête émise pour voir si elle est autorisée). Les mots de passe sont sécurisés depuis que la gestion des mots de passe est cryptée entre le client et le serveur.
- ☛ **Rapide, Robuste et Convivial** : MySQL est de trois à quatre fois plus rapide que la plupart des autres bases de données commerciales : les fonctions SQL sont implémentées à travers des classes de bibliothèques extrêmement optimisées. En général, il n'y a plus d'allocation mémoire après une requête d'initialisation.

Les développeurs de MySQL (TCX) utilisent MySQL depuis 1996 dans un environnement de plus de 40 bases de données contenant 10000 tables, dont plus de 500 contiennent plus de 7 millions d'enregistrements. C'est environ 100 Giga Octets de données critiques.

Sa gestion est aisée puisqu'elle ne nécessite pas des compétences poussées en administration de bases de données.

- ☛ **Economique** : MySQL est un SGBD, « OPEN SOURCE ».

V.4.Les outils de développement :

V.4.1.Wamp :[9]

WampServer est une plate-forme de développement Web sous Windows. Il vous permet de développer des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin et SQLite Manager pour gérer plus facilement vos bases de données.

**lancer wampserver:**

On ne peut pas proprement parler du lancement d'wampserver, il s'agit en fait de la mise en route du serveur apache et de serveur MYSQL. A l'installation, un raccourci vers wampserverest crée dans le répertoire « Démarrer/programmes/wampserver ». Une fois wampserver lancé, une icône se place dans la barre des tâches à coté de l'horloge. Un clic droit permet d'accéder à différents menus :

**Fichier log** : renvoie aux erreurs générées par apache et MYSQL.

**Configuration** : donne accès aux différentes configurations du wampserver ;

**Administration** : ouvre la page d'administration des alias et du répertoire des bases de données « phpMyAdmin » ;

**Démarrer/arrêter** démarrer/arrêter: Apache MySQL ;

**Redémarrer** :redémarrer apache et MySQL ;

**Quitter** : ferme wampserver ;

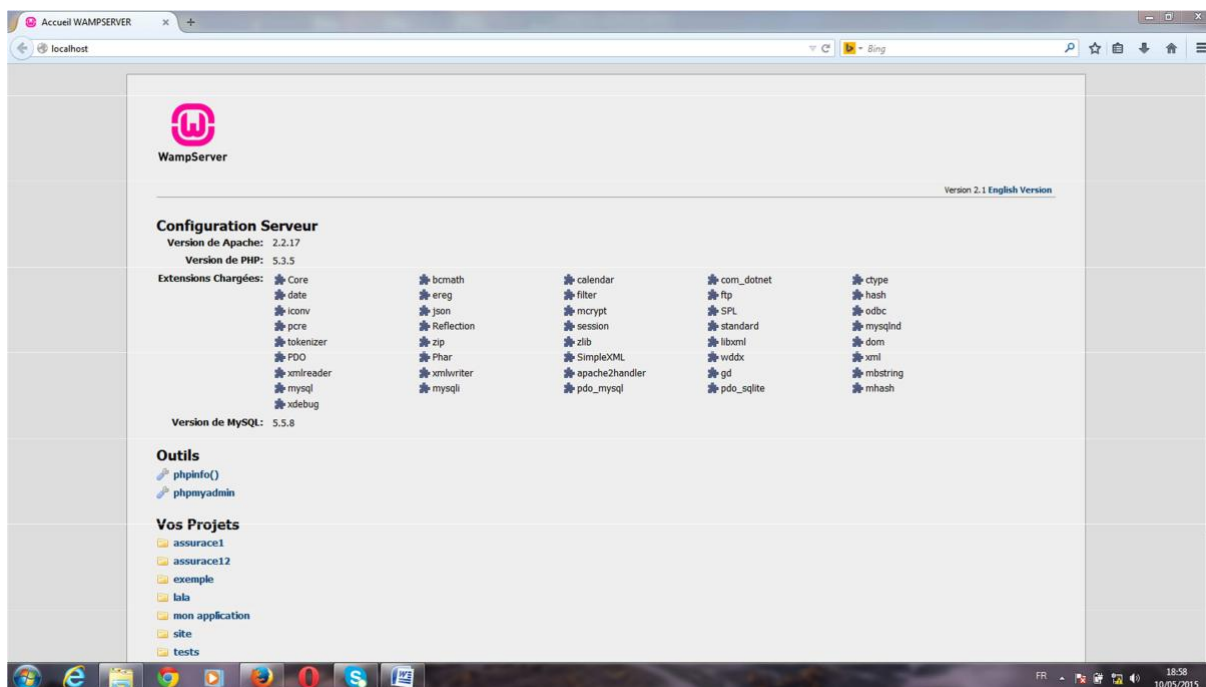


Figure V.3. Page d'accueil Wamp

Pour que les pages PHP soient interprétées, il est impératif de placer les fichiers dans le répertoire www ou dans un alias crée. Pour visualiser les pages, il suffit alors d'ouvrir le « Web local » ou d'accéder aux alias via la page d'administration.

## V.4.2. PhpMyAdmin :

phpMyAdmin est un utilitaire rendant plus conviviale l'administration de base de données MySQL. Il consiste en un ensemble de scripts PHP permettant d'administrer des bases de données MySQL en passant par un navigateur web.

### V.4.2.1. les fonctions de PhpMyAdmin :

- ✚ Création et suppression de Base de données.
- ✚ Création, modification, copie et suppression de tables.
- ✚ Edition, ajout et suppression des champs.
- ✚ Exécution des commandes SQL.
- ✚ Création d'index.
- ✚ Chargement de fichier dans les tables.

### V.4.2.2. Utilisation de PhpMyAdmin :

Pour accéder à PhpMyAdmin, il faut d'abord vérifier que wamp est lancé et que le serveur fonctionne, après on peut accéder à partir de « l'administration ». Pour ouvrir l'administration il suffit de faire une clique droite sur l'icône et sélectionner « Administration ». Une page web apparaît, au milieu de celle-ci il ya un bouton PhpMyAdmin avec une simple clique là-dessus, la page d'accueil de PhpMyAdmin s'affiche dans la fenêtre du navigateur, accompagné d'un champ de sélection de base de données présente sur l'hôte MySQL par défaut.

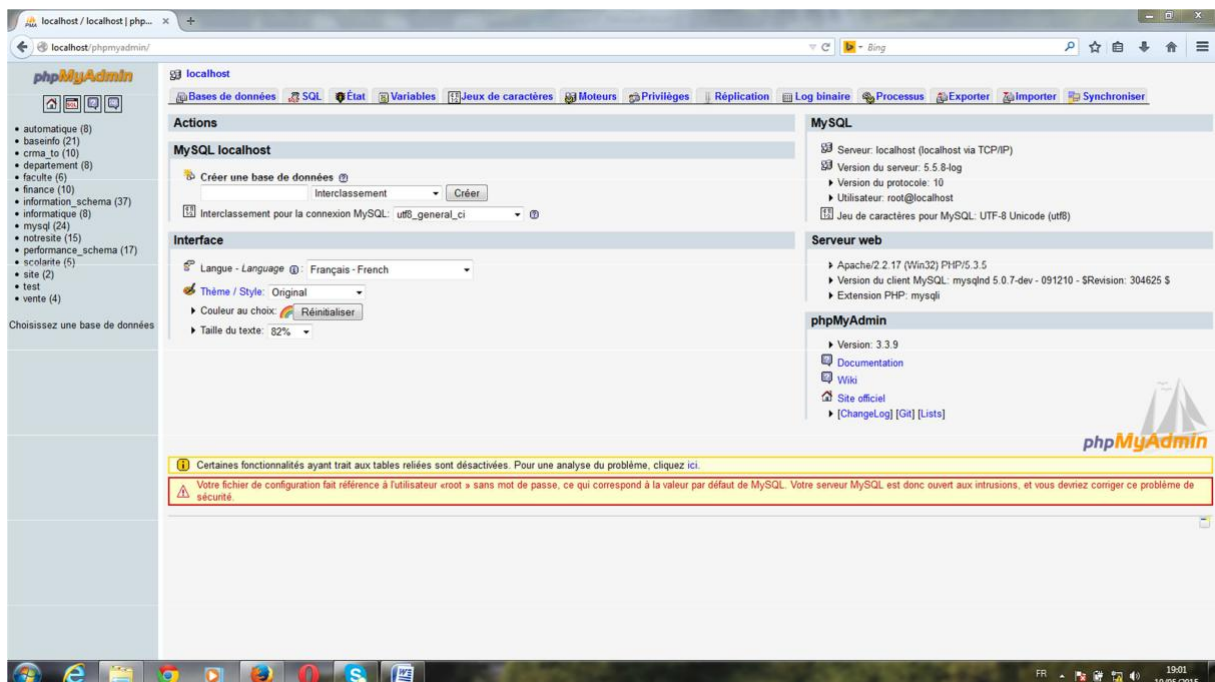


Figure.V.4.Administration de MySQL à partir de PhpM yAdmin

Pour afficher le contenu de la base personnelle par exemple, il faut cliquer sur la ligne correspondante à la base de données personnelle dans la partie gauche de la fenêtre du navigateur. Toutes les tables de cette base de données seront affichées :



Figure.V.5.Accès à la base de données « notresite »à partir de PhpMyAdmin

### V.4.3. Macromedia Dreamweaver 8 :

Macromedia Dreamweaver8 est un éditeur HTML professionnel destiné à la conception, au codage et au développement de sites, de pages et d'applications Web. Quel que soit l'environnement de travail utilisé (codage-manuel HTML ou environnement d'édition visuel), Dreamweaver propose des outils qui vous aideront à créer des applications Web. Les fonctions d'édition visuelles de Dreamweaver vous permettent de créer rapidement des pages sans rédiger une seule ligne de code. Vous pouvez afficher tous les éléments ou actifs de votre site et les faire glisser directement d'un panneau convivial dans un document. Rationalisez les tâches de développement en créant et en modifiant des images dans Macromedia Fireworks ou toute autre application graphique, puis en les important directement dans Dreamweaver, ou en ajoutant des objets Flash Macromedia.

Dreamweaver propose également un environnement de codage complet comprenant des outils de modification du code (comme la coloration du code et la création de balises) ainsi que des documents de référence sur les feuilles de style encascade (CSS - Cascading StyleSheets), JavaScript et ColdFusion Markup Language (CFML). La technologie RoundtripHTML de Macromedia permet d'importer des documents HTML codés manuellement sans en modifier le code pour que vous puissiez ensuite reformater ce dernier avec le style de formatage de

votre choix. Dreamweaver permet également de créer des applications Web reposant sur des bases de données dynamiques au moyen de technologies serveur comme CFML, ASP.NET, ASP, JSP et PHP. Dreamweaver est entièrement personnalisable. Vous pouvez créer vos propres objets et commandes, modifier les raccourcis clavier ou encore rédiger un code JavaScript pour intégrer de nouveaux comportements, inspecteurs de propriétés et rapports de site aux fonctionnalités de Dreamweaver .

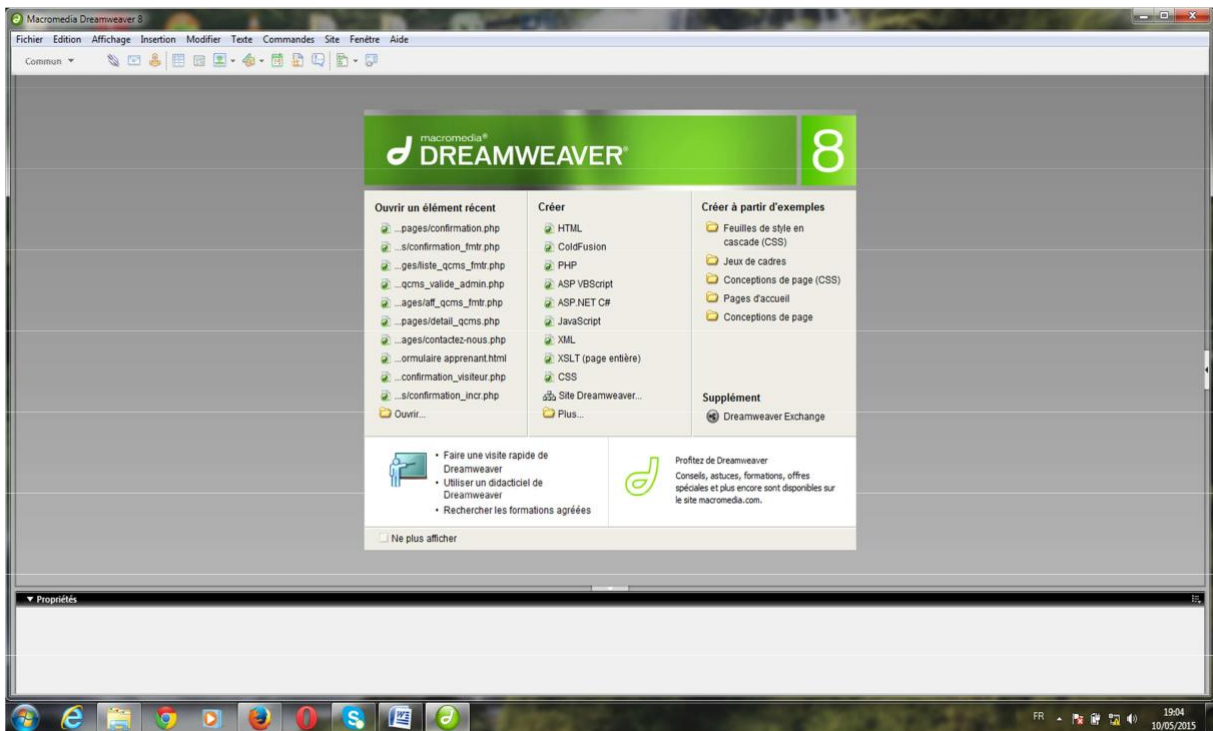


Figure.V.6.Interface Dreamweaver

## V.5. Présentation de quelques interfaces de notre plate-forme :

### V.5.1. Page principale :

Dès que le visiteur saisit l' url de la plate forme, la page ci-dessous s' affiche.

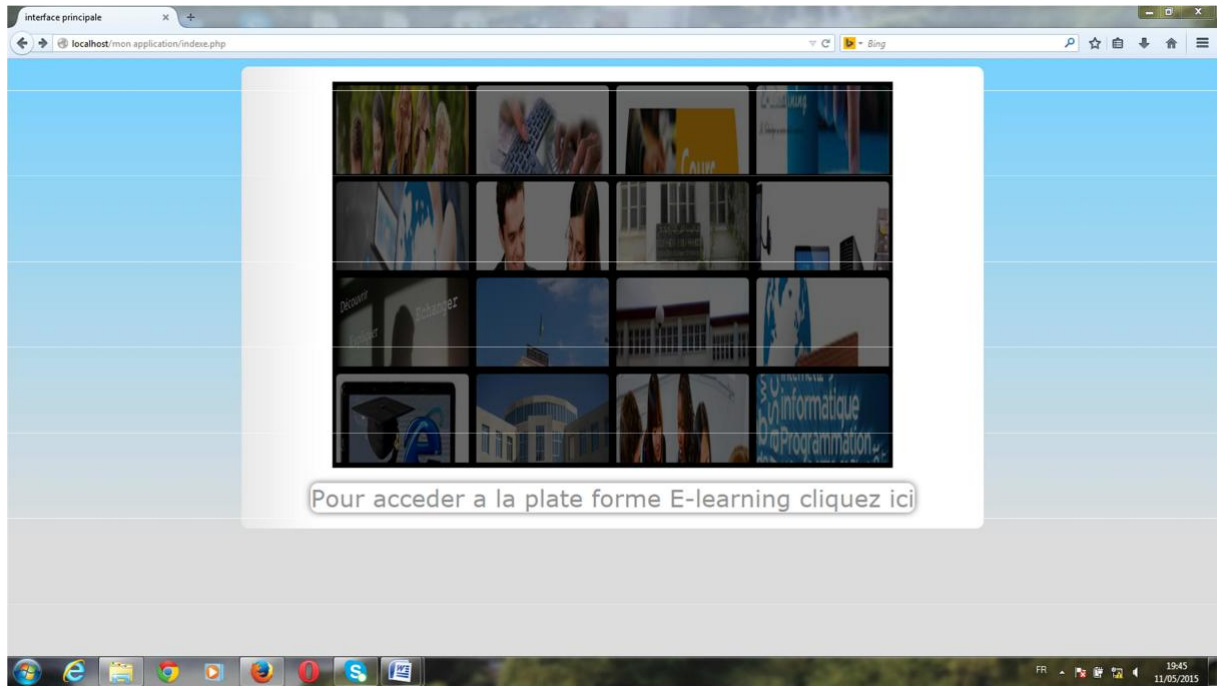


Figure.V.7.La page principale de la plate forme.

### V.5.2.La page d'accueil :

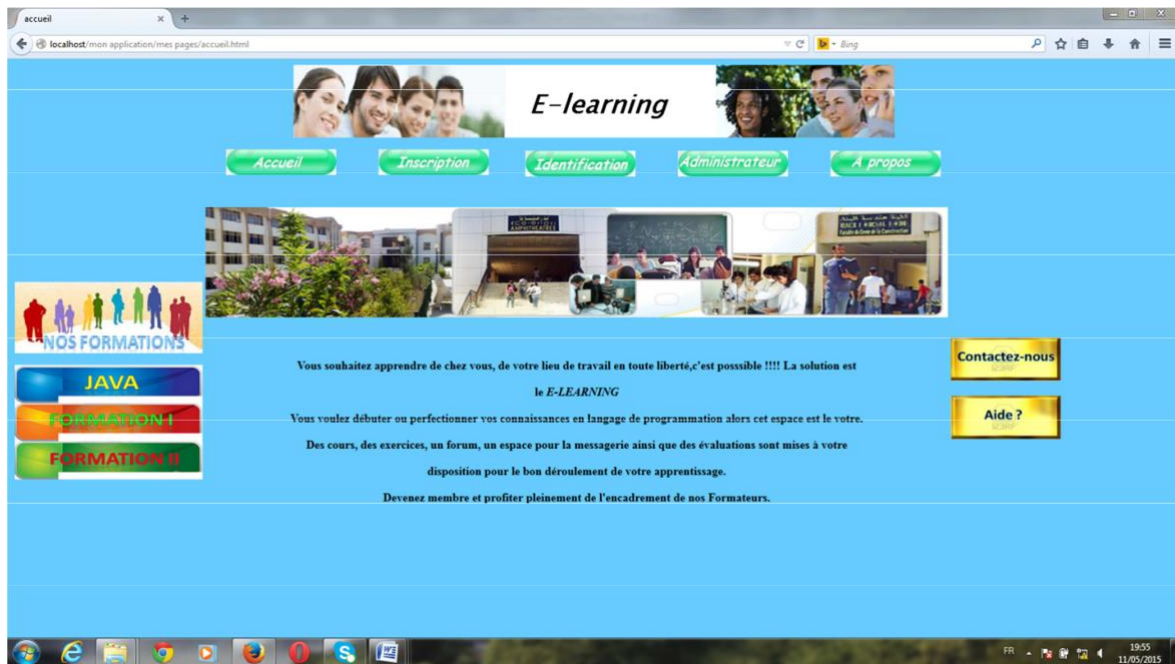


Figure.V.8.Interface de la page d'accueil.

V.5.3.la page de département :

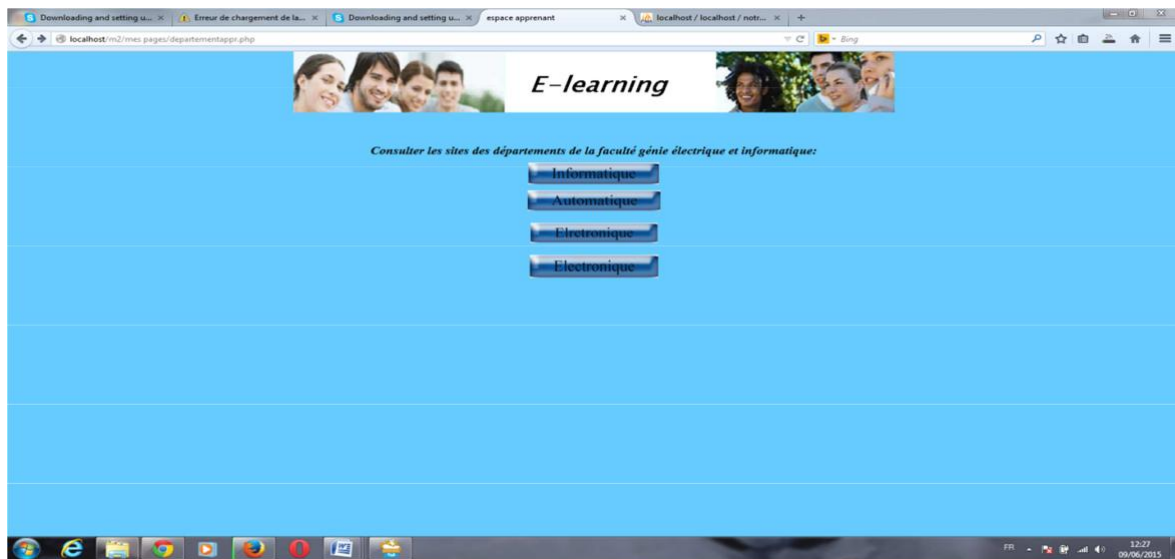


Figure.V.9.Interface de la page du département.

V.5.4.la page de département informatique :



Figure.V.10.Interface de la page du département informatique.

V.5.5.la page de l'administrateur de département :

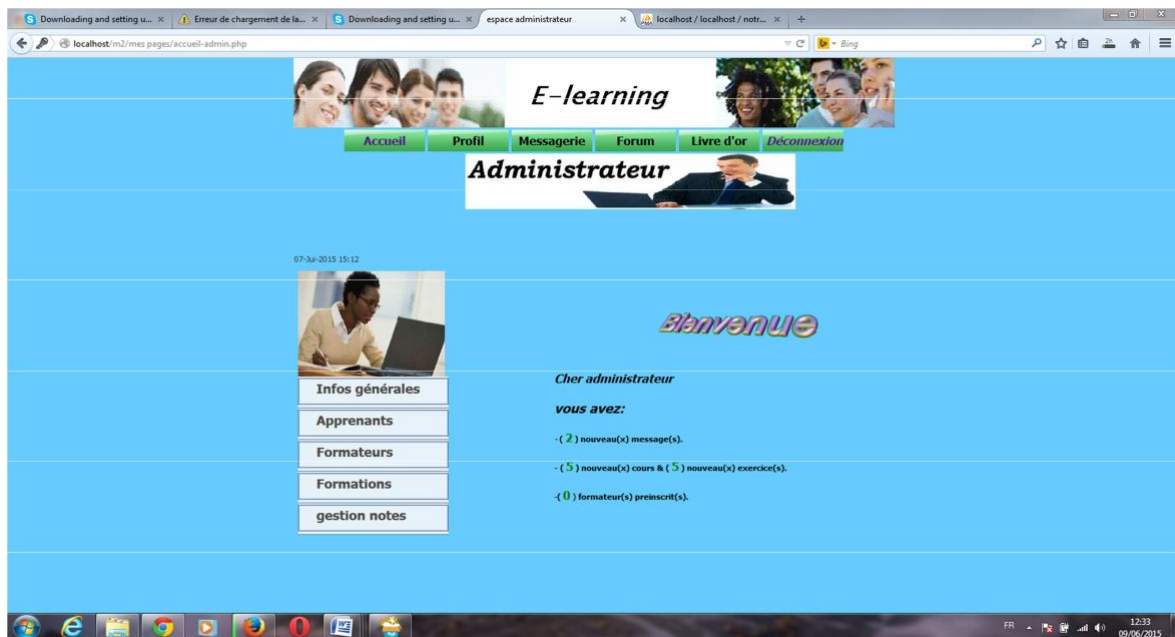


Figure.V.11.Interface de la page d'administrateur de département.

V.5.6.Espace messagerie :

Permet l'intercommunication entre les trois acteurs principaux à savoir : l'administrateur, le formateur et l'apprenant.

 Espace messagerie :

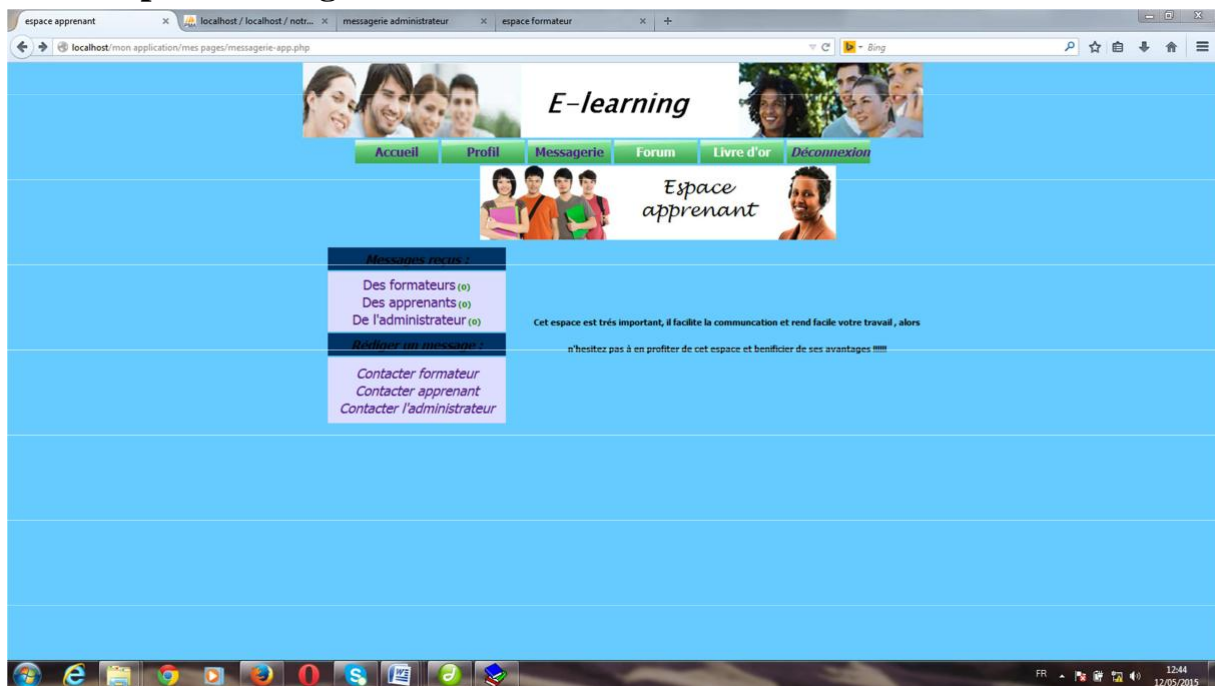


Figure.V.12.espace messagerie

 Liste des messages envoyés par des apprenants :

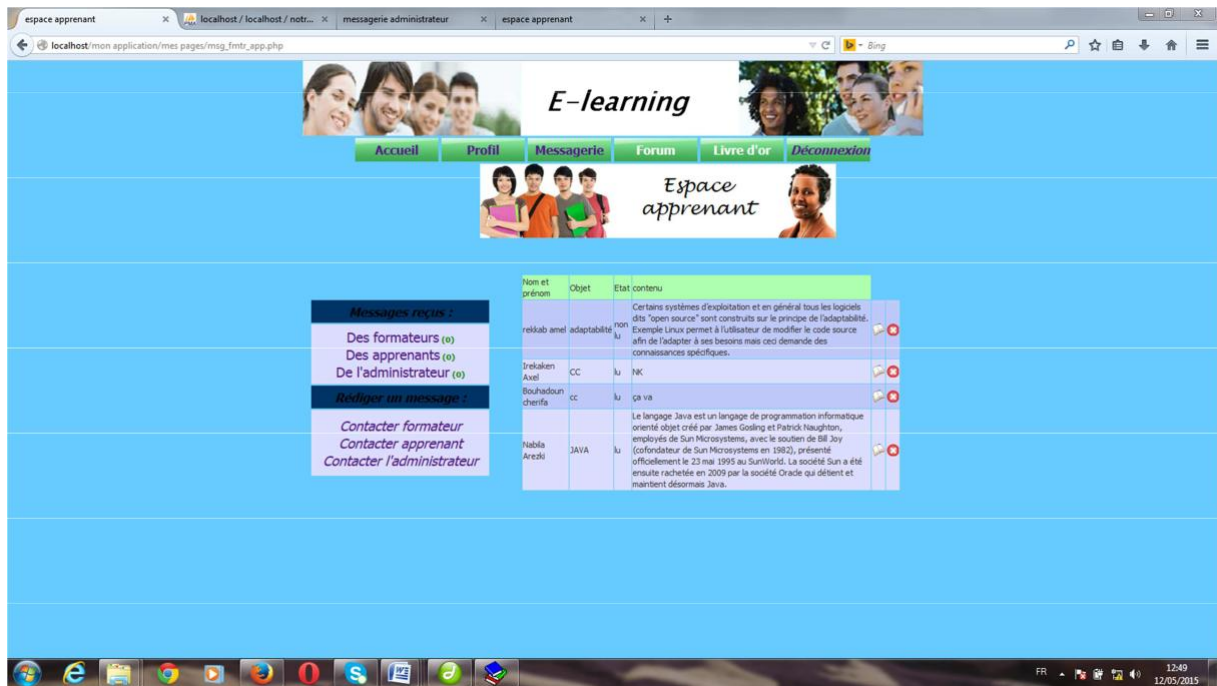


Figure.V.13. liste des messages

 Espace d'affichage du message :

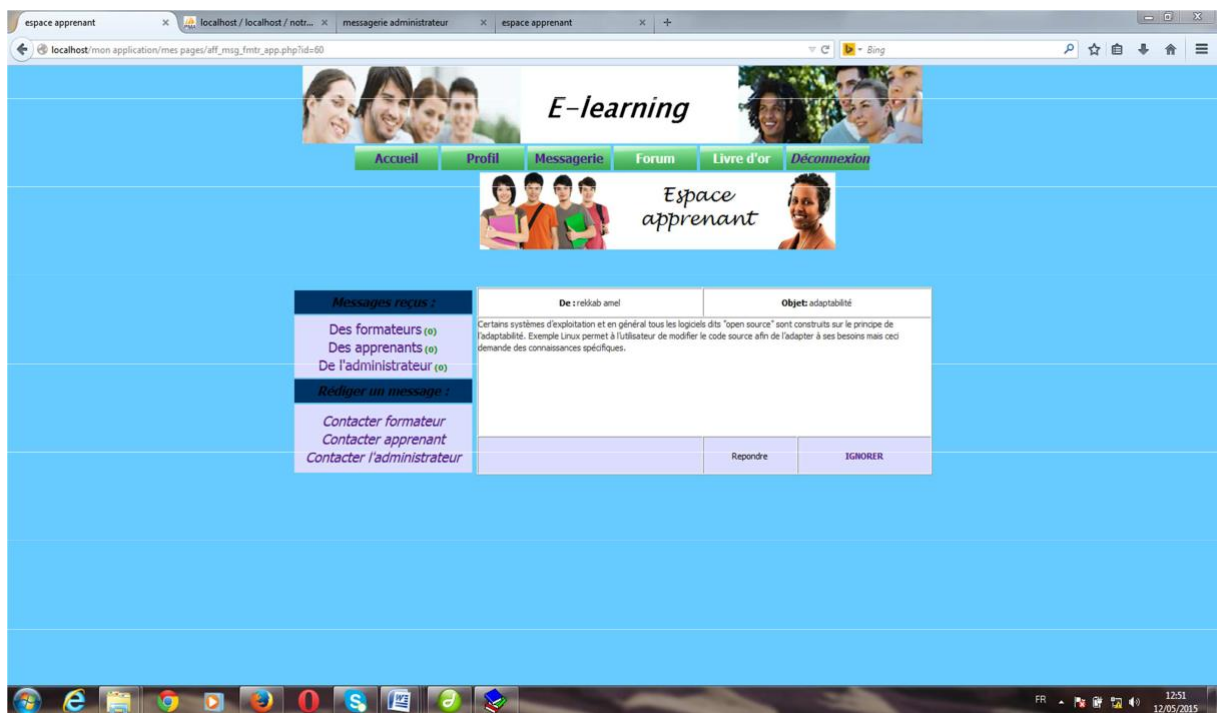


Figure.V.14.affichage du message

## ✚ Espace pour envoyer un message (après avoir sélectionné un destinataire):

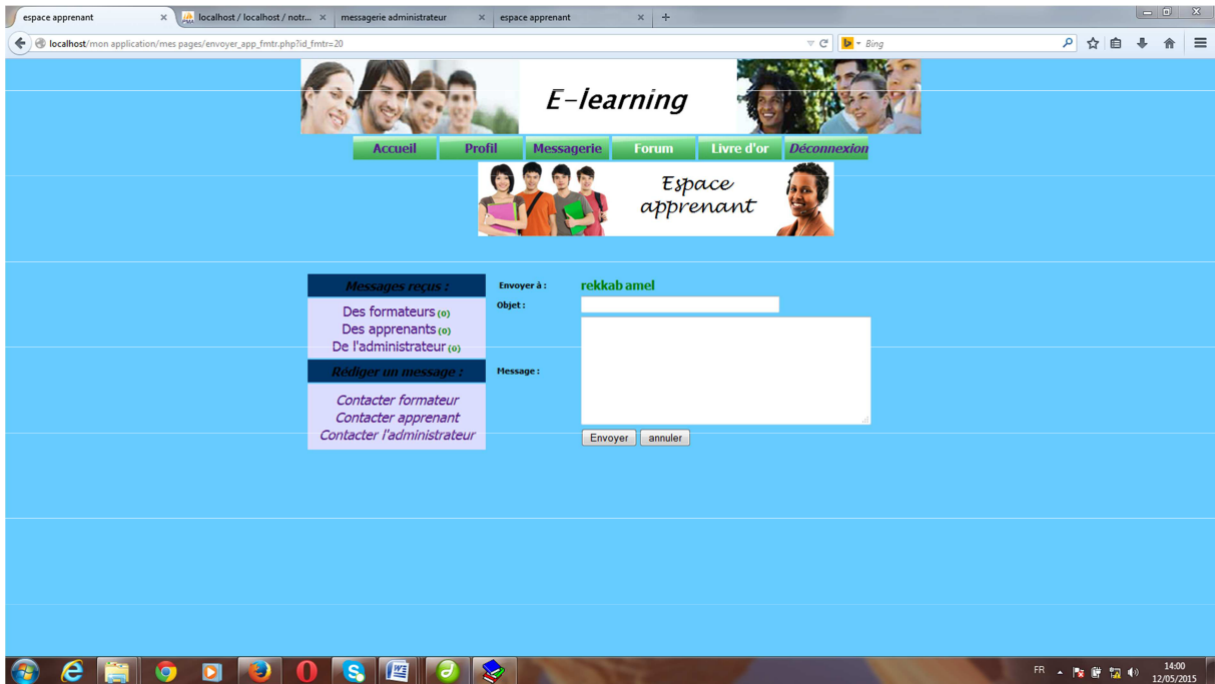


Figure.V.15. envoyer un message

## ✚ Suppression d'un message :

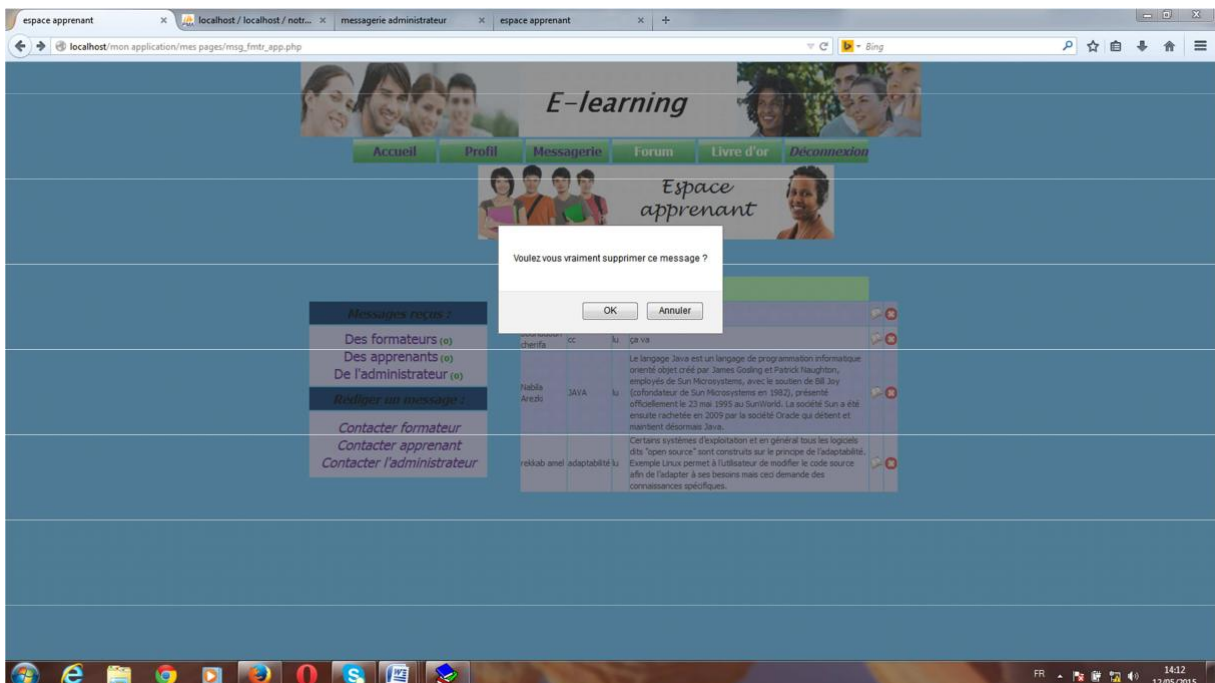


Figure.V.16. supprimer un message

**Conclusion :**

Dans ce dernier chapitre nous avons présenté en premier lieu les outils logiciels qui nous ont permis de réaliser notre travail, à savoir l'environnement de développement, les langages de programmation et les logiciels de création des pages web. Puis quelques interfaces essentielles de la plate forme.

# Conclusion générale

Dans ce mémoire nous avons présenté une démarche pour mettre en œuvre la conception d'une plate forme e Learning et l'implémentation d'une base de données distribuée pour cette plate forme basant sur le mécanisme d'intégré facilement des autres systèmes.

Le travail réalisé nous a ramené à étudier plusieurs domaines dans le cadre de l'adaptabilité, sécurité informatique et notammenten base de données réparties. Mon mémoire est articulé autours de deux parties ;la première est théorique, elle contient trois chapitres, le premier traite les protocoles applicatifs sur internet, le deuxième parle de l'adaptabilité et bases de données réparties en général,n le troisième traite un domaine spécifique dans le cadre de e Learning.

La seconde partie de mon mémoire est la partie pratique, dans laquelle nous avons implémenté une plate forme e Learning réparties, une conception a été nécessaire pour mieux contourner toutes les fonctionnalités de chaque casd'utilisation

Au cours de l'élaboration de mon mémoire, j'ai acquis plusieurs connaissances qui s'avèrent bénéfiques dans le cadre de ma formationet qui sont un complément essentiel pour la consolidation de plusieurs données théoriques acquises tout au long de ma formation académique.

Ce travail peut être encore amélioré en envisageantles perspectives suivantes :

Compléter le processus de fédération des bases de onnéesd par l'extension de cette dernière vers d'autres systèmes.

Mettre en place des mécanismes de l'adaptabilité etde sécurité.

## BIBLIOGRAPHIE

---

- [1] : rapport-gl-service-oriented-architecture.pdf(les architectures orientées services SOA)
- [2]: livre blanc-SOA Architecture logique
- [3] : Architecture orientée service wikipédia
- [4]: les services web wikipédia
- [5]: eyrolles-services web avec J2EE et .Net
- [6] : Adaptation — EduTech Wiki.htm (page réaliser dans le cadre du cours conception des « Environnement Informatisé d'Apprentissage » de formation Maltt, au TECFA).
- [7] : mémoire de fin d'études Master II « conceptio et réalisation d'une plate forme e-Learning adaptable supportant des télé cours synchrone »Réalisé par ZENINE NAIMA.
- [8] : MEMOIRE DE MAGISTER, sujet : « **Environnement de travail collaboratif dédié aux travaux pratiques à distance** », **SPECIALITE : INFORMATIQUE, OPTION : SYSTEMES INFORMATIQUES, Présenté par « RAMDANE MOHAMED »**
- [9] : mémoire de fin d'études : «**conception et réalisation d'un site web pour E-learning** » cas « l'école 2IntPartner » dirigé par Mr DIB AHMED, promotion 2013/2014.
- [10] : <http://www.wikipédia.org/wiki/>
- [11] : mémoire de fin d'étude « Développement d'uncrawler pour des documents PDF et HTML avec application de l'algorithme DFS (Depth First Search) réalisé par BELMILOUD SAMIRA et BELMILOUD LOUIZA
- [12] : <http://www.commentcamarche.net>
- [13] : Maldeleine Doussy « Informatique et communication »Edition Bréal ; France 2005.
- [14] : « concevoir des application web avec UML » Jim conallen, eyrolles, 2000.
- [15] : :Grady Booch, James Rambough et Ivar Jacobso « Le guide de l'utilisateur UML », Edition Eyrolles, 2001.
- [16] : Gilles Hunault, Un petit tuteur MYSQL, Angers, janvier 2000.
- [17] : Manuel d'aide fourni avec le logiciel macromedia DREAMWEAVER 8.
- [18] : Jean Engels : « PHP cours et exercices », EYROLLES 2005.
- [19] : Daniel Koch, Oliver Kurten, Florion Hormon : « Html, XML, XHTML, Micro application » grand livre.

**Introduction :**

La complexité croissante des systèmes d'information étaient derrière l'apparition des différents modes de modélisation, certains de ces modes ont montres des limites, d'où la nécessité d'adapter ces modes à ces systèmes.

Dans l'objectif d'accomplir ces insuffisances, plusieurs méthodes sont apparues. Le groupe OMG a développé une notation standard utilisable dans les développements informatique basés sur l'objet, ainsi c'était l'apparition de l'UML (**Unified Modelling Language**) « Langage de Modélisation objet Unifié ».

**I.les processus de développement d'application web:****Besoins et cas d'utilisation :**

Cette phase détermine l'expression des besoins, en décrivant sans ambiguïté le système logiciel à développer (besoins non fonctionnels) et l'élaboration des cas d'utilisation qui sont formels et qui permettent de consigner et d'exprimer les interactions et les dialogues des utilisateurs (nommés acteurs) avec le système besoins( fonctionnels).

**Analyse :**

C'est à travers des activités d'analyse que l'on peut définir à partir de l'examen des besoins un modèle conceptuel du système à développer. C'est à dire un modèle capable d'être concrétisé sous forme logiciel. Les artefacts de cette étape incluent les classes et les collaborations détaillées, les diagrammes de séquences, les diagrammes d'états et les diagrammes d'activités.

**La conception :**

Cette phase traite les artefacts produits durant l'étape d'analyse et elle est consacrée à la réalisation des cas d'utilisation et la définition de la vue logique du système. Les activités de la conception se focalisent sur les diagrammes de classes et d'interaction, en outre l'élaboration des classes et des collaborations, les activités de conception incluent les deux tâches suivantes :

Le découpage des objets en couches, telle que couche client et couche serveur. La définition et le découpage des interfaces utilisateurs ou des pages Web.

En fin, le modèle de conception peut être directement traduit en code source

**L'implémentation :**

A ce stade de processus, les cas d'utilisations sont terminés. Le problème a été analysé en profondeur et nous avons défini une conception appropriée à l'architecture. Nous pouvons alors, entreprendre l'étape d'implémentation qui inclut les activités suivantes :

La traduction de la conception en code source et en composants

; La réalisation des tests unitaires ;

La rétro ingénierie.

## II. Présentation d'UML :

UML est une norme OMG (*Object Management Group*), un langage qui permet de représenter des modèles. Il a été conçu pour servir de support à une analyse basée sur les concepts objets.

C'est un langage formel servant de support de communication. Il facilite la compréhension des solutions objets grâce à l'utilisation des notations graphiques qui permettent de visualiser ces solutions objets.

## III. Modélisation avec UML :

Un modèle est une abstraction de la réalité, une vue subjective mais pertinente de la réalité. Un modèle reflète des aspects importants de la réalité, il en donne une vue juste et pertinente.

UML propose plusieurs modèles qui sont des descriptions du système étudié, et parmi ces modèles :

- Le modèle de classe
- Le modèle des cas d'utilisation
- Le modèle d'interaction
- Le modèle des états
- Le modèle de réalisation
- Le modèle de déploiement

Ces modèles sont élaborés au moyen de diagrammes qui spécifient un aspect précis du système.

On peut organiser une présentation d'UML autour d'un découpage en vues, ou bien en différents diagrammes, selon qu'on sépare plutôt les aspects fonctionnels des aspects architecturaux, ou les aspects statiques des aspects dynamiques.

Nous adopterons plutôt dans la suite un découpage en diagrammes, mais nous commençons par présenter les différentes vues, qui sont les suivantes :

- La vue fonctionnelle, interactive, qui est représentée à l'aide de diagrammes de cas et de diagrammes des séquences. Elle cherche à appréhender les interactions entre les différents acteurs/utilisateurs et le système, sous forme d'objectif à atteindre d'un côté et sous forme chronologique de scénarios.

- La vue structurelle, ou statique, réunit les diagrammes de classes et les diagrammes de packages. Les premiers favorisent la structuration des données et tentent d'identifier les objets/composants constituant le programme, leurs attributs, opérations et méthodes, ainsi que les liens ou associations qui les unissent. Les seconds s'attachent à regrouper les classes fortement liées entre elles en des composants les plus autonomes possibles.

- La vue dynamique, qui est exprimée par les diagrammes d'états. Cette vue est plus algorithmique et orientée « traitement », elle vise à décrire l'évolution (la dynamique) des objets complexes du programme tout au long de leur cycle de vie. De leur naissance à leur mort. Le diagramme d'activité est une sorte d'organigramme correspondant à une version simplifiée du diagramme d'états. Il permet de modéliser des activités qui se déroulent en parallèle les unes aux autres, quand ce parallélisme peut poser problème. En général, les diagrammes d'états à eux seuls ne permettent pas de faire apparaître les problèmes spécifiques posés par la synchronisation des processus en concurrence, pour assurer la cohérence du comportement et l'absence d'inter blocage.

#### **IV. Les diagrammes UML :**

##### **IV.1. Le diagramme des cas d'utilisation (vue fonctionnelle) :**

Les cas d'utilisation (use case) modélisent une interaction entre le système informatique à développer et un utilisateur ou acteur interagissant avec le système. Plus précisément, un cas d'utilisation décrit une séquence d'actions réalisées par le système qui produit un résultat observable pour un acteur.

##### **IV.2. Le diagramme des classes (vue structurelle) :**

Un diagramme des classes décrit le type des objets ou données du système ainsi que les différentes formes des relations statiques qui les relient entre eux. On distingue classiquement deux types principaux de relations entre objets :

- Les associations, utilisées dans les modèles entité/association.
- Les sous-types, particulièrement en vogue en conception orientée objets, puisqu'ils s'expriment très bien à l'aide de l'héritage en programmation.

##### **IV.3. Les diagrammes de séquences (vue fonctionnelle) :**

Les diagrammes de séquences mettent en valeur les échanges de messages (déclenchant des événements) entre acteurs et objets (ou entre objets et objets) de manière chronologique, l'évolution du temps se lisant de haut en bas.

Chaque colonne correspond à un objet (décrit dans le diagramme des classes), ou éventuellement à un acteur, introduit dans le diagramme des cas. La *ligne de vie* de l'objet représente la durée de son interaction avec les autres objets du diagramme.

Un diagramme de séquences est un moyen semi-formel de capturer le comportement de tous les objets et acteurs impliqués dans un cas d'utilisation. On peut faire apparaître de nombreuses informations de contrôle le long de la ligne de vie d'un objet.

Un diagramme des séquences permet de vérifier que tous les acteurs, les classes, les associations et les opérations ont bien été identifiés dans les diagrammes de cas et de classes.

#### **VI.4. Les diagrammes d'états (vue dynamique) :**

Les diagrammes d'états décrivent tous les états possibles d'un objet. Ils indiquent en quoi ses changements d'états sont induits par des événements.

Si les diagrammes de séquences regroupent tous les objets impliqués dans un unique cas d'utilisation, les diagrammes d'états indiquent tous les changements d'états d'un seul objet à travers l'ensemble des cas d'utilisations dans lequel il est impliqué. C'est donc une vue synthétique du fonctionnement dynamique d'un objet.

Les diagrammes d'états identifient pour une classe donnée le comportement d'un objet tout au long de son cycle de vie (de la naissance ou état initial, symbolisée par le disque plein noir, à la mort ou état final, disque noir couronné de blanc).

#### **IV.5. Le diagramme d'activité (vue dynamique) :**

Le diagramme d'activité est un cas particulier du diagramme d'états, dans lequel à chaque état correspond une activité constituant un élément d'une tâche globale à réaliser.

Le but de ce diagramme est de mettre en évidence les contraintes de « séquentialité » et de parallélisme qui pèsent sur la tâche globale.

### **V. Extension d'UML pour le web :**

L'extension d'UML pour le web définit un ensemble de stéréotype, d'étiquettes et de contraintes, qui rendent possible la modélisation web. Ces stéréotypes et ces contraintes sont appliqués sur certains des composants propres aux applications Web permettant ainsi de représenter au sein du même modèle et sur les mêmes diagrammes que ceux qui décrivent le reste du système.

#### **V.1. Stéréotype :**

C'est une extension du vocabulaire d'UML permettant l'association d'une nouvelle signification à un élément du modèle. Il est représenté par une chaîne de caractère entre guillemet (« »).

##### **1. Classe :**

- **Page serveur « server page »**

**Description:** elle représente une page web possédant des scripts exécutés par le serveur. Ils interagissent avec des ressources serveur telles que les bases de données.

**Icône :**



**Contraintes :** Les pages serveur ne peuvent avoir de relation qu'avec des objets sur le serveur

**Étiquette :** Moteur de script qui peut être un langage ou le moteur qui doit être utilisé pour exécuter ou interpréter cette page.

#### • Page client « client page »

**Description:** Une instance d'une page client est une page web formatée en HTML. Les pages client peuvent contenir des scripts interprétés par les navigateurs lorsque celles-ci sont restituées par ces derniers. Les fonctions des pages clients correspondent aux fonctions des scripts des pages web. Les pages clients peuvent avoir des associations avec d'autres pages clients ou serveurs.

**Icône :**



**Contraintes:** Aucune.

**Étiquette :**

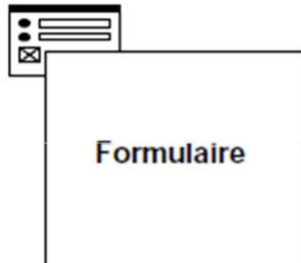
- Titre (Title) : Titre de la page tel qu'il est affiché par le navigateur.
- Base (Base) : URL de base pour déréférencer les URL relatives.
- Corps (Body) : Ensemble des attributs de la balise <body>, qui définit des caractéristiques par défaut du texte et de l'arrière-plan.

#### • Formulaire « form »

**Description:** Une classe stéréotype « form » est un ensemble de champs de saisie faisant partie d'une page client. A une classe formulaire correspond une balise HTML « form », les attributs de cette classe correspondent aux éléments de saisie d'un formulaire HTML (zone de saisie, zone de texte, bouton d'option).

Un formulaire n'a pas d'options : toute opération qui interagit avec le formulaire appartient à la page qui la contient.

**Icône :**



**Contraintes :** Aucune.

**Étiquette :** les méthodes GET ou POST : utilisée pour soumettre les données à l'URL de l'attribut action de la balise HTML <form>.

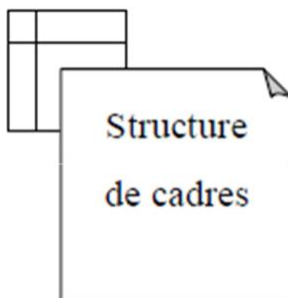
### • Structure de cadres « Frameset »

**Description:** Une structure de cadres est un conteneur de plusieurs pages web.

La zone d'affichage rectangulaire est divisée en cadres rectangulaires, à chaque cadre peut être associé un nom unique de cible « Target ».

Une classe stéréotypée « framset » est directement associée à une structure de cadres de pages web par la balise HTML <framset>.

**Icône :**



**Contraintes:** Aucune.

**Étiquette :**

○ Rangées (rows) : Valeur de l'attribut rows de la balise HTML <framset>.

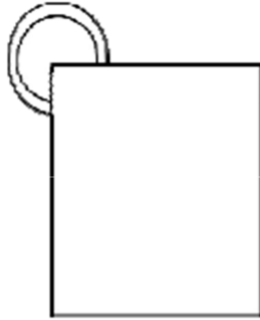
C'est une chaîne de pourcentages séparés par des virgules, définissant les hauteurs relatives des cadres.

○ Colonnes (cols) : Valeur de l'attribut cols de la balise HTML <frameset>. C'est une chaîne de pourcentages séparés par des virgules définissant les largeurs des cadres.

**• Cible « Target »**

*Description:* Une cible est une zone nommée dans la fenêtre du navigateur dans laquelle des pages web peuvent être affichées. Le nom de la classe stéréotypée est ce lui de la cible. Habituellement, une cible est le cadre d'une structure défini dans une fenêtre ; ce pendant, une cible peut être une toute nouvelle instance d'un navigateur : une fenêtre. Une association « targeted link » spécifie la cible ou une page web doit être affichée.

*Icône :*



*Contraintes:* Pour chaque client du système le nom de la cible doit être unique. Par conséquent sur un même client, il ne peut exister qu'une seule instance d'une même cible.

*Étiquettes:* Aucune.

**• Objet JavaScript « JavaScript Object »**

*Description:* Sur un navigateur compatible java script. Il est possible de simuler des objets personnalisés à l'aide de fonctions JavaScript. Les objets java script ne peuvent exister que dans le contexte de page client.

*Icône:* Aucune.

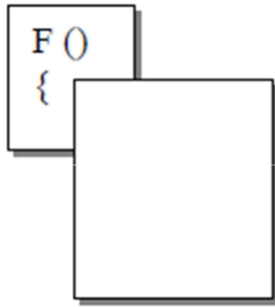
*Contraintes:* Aucune.

*Étiquettes:* Aucune.

**• Objet script client « Client Script Object »**

*Description:* Un objet script client est un ensemble qui regroupe des scripts particuliers dans un fichier, lequel est inclus dans une requête distincte du navigateur client. Ces objets regroupent des lots de fonctions couramment utilisés à travers une application ou d'une entreprise.

*Icône:*



*Contraintes:* Aucune.

*Étiquettes:* Aucune.

## 2. Association :

### • Lien « link »

*Description:* Un lien est un pointeur d'une page client vers une autre page. Dans un diagramme de classe, un lien est une association entre une page client et une autre page client ou une page serveur. A un lien correspond une balise HTML ancre.

*Icône :* Aucune.

*Contraintes:* Aucune.

*Étiquettes:* La liste des noms de paramètres qui doivent être passés avec la demande de la page liée.

### • Lien cible « targeted link »

*Description:* Similaire à une association lien. Un lien cible est un lien dont la page associée est affichée dans une cible. A un lien cible correspond une balise HTML, dont l'attribut target prend la valeur de la cible.

*L'icône:* Aucune.

*Contraintes:* Aucune.

*Étiquettes:* Liste des paramètres et le nom de la cible (target name), nom de la cible ou la page vers laquelle pointe le lien, doivent être passés avec la demande de la page liée.

### • Contenu de cadre « fram content »

*Description :* Une association « contenu de cadre » est une association d'agrégation qui traduit l'apparence d'une page ou d'une cible à un cadre.

Une association contenue de cadre peut aussi pointer vers une structure de cadre aboutissant dans ce cas à des cadres imbriqués.

*Icône :* Aucune.

*Contraintes :* Aucune.

**Étiquette :**

Rangée : entier qui indique la rangée du cadre dans la structure de cadres auquel appartient la page, ou la cible associée.

Colonne : entier qui indique la colonne du cadre dans la structure des cadres auquel appartient la page, ou la cible associée.

**• Soumet « submit »**

**Description:** Une association de soumission « submit » se trouve toujours entre un formulaire et une page serveur. Les formulaires soumettent les valeurs de leurs champs au serveur pour qu'il les traite. Le serveur web traite la page serveur, qui accepte et utilise les informations du formulaire et les utilise.

**Icône:** Aucune.

**Contraintes :** Aucune.

**Étiquettes:** Listes des noms de paramètres qui doivent être passés avec la demande de la page liée.

**• Construit « build »**

**Description:** La relation « build » est une relation particulière qui fait le pont entre les pages clients et les pages serveurs. L'association « build » indique quelle page serveur est responsable de la création d'une page client. C'est une relation orientée, puisque la page client n'a pas connaissance de la page qui est à l'origine de son existence.

Une page serveur peut construire plusieurs pages client, en revanche, une page cliente ne peut être construite que par une seule page serveur.

**Icône:** Aucune.

**Contraintes:** Aucune.

**Étiquettes:** Aucune.

**• Rediriger « redirect »**

**Description:** Une relation « redirect », est une association unidirectionnelle avec une autre page web, pouvant être dirigée à partir d'une page client ou serveur vers une page client ou serveur.

**Icône :** Aucune.

**Contraintes:** Aucune.

**Étiquette:** le délai (delay), délai que doit observer une page client avant de se rediriger vers la page destination. Cette valeur correspond à l'attribut content de la balise <META>.

**3. Attributs :****• Éléments de saisie « input element »**

**Description:** Un élément de saisie correspond à la balise <input> d'un formulaire HTML. Les étiquettes associées à cet attribut stéréotype correspondant aux attributs de la balise <input>.

Les attributs obligatoires de la balise HTML <input> sont renseignés de la manière suivante : l'attribut « name » prend la valeur du nom de l'élément de saisie et l'attribut « value » prend celle de sa valeur initiale.

**Icône :** Aucune.

**Contraintes :** Aucune.

**Étiquette :**

Type (Type) : Le type de l'élément de saisie : texte, numérique, mot de passe, case à cocher, bouton d'option, bouton « submit » ou bouton « reset ».

Taille (Size) : Définit la largeur visible allouée à l'écran en caractères.

Longueur max (maxlength) : C'est le nombre maximal de caractères que peut saisir l'utilisateur .

#### • Sélection d'éléments « select element »

**Description :** Contrôle de saisie employé dans le formulaire, il permet à l'utilisateur de sélectionner une ou plusieurs valeurs dans une liste. La plupart des navigateurs restituent ce contrôle par une liste d'option ou une liste déroulante.

**Icône :** Aucune.

**Contraintes :** Aucune.

**Étiquette :**

Taille(Size) : définit le nombre d'éléments qui doivent être affichés simultanément.

Multiple (Multiple) : valeur booléenne qui indique que plusieurs éléments peuvent être sélectionnés conjointement.

#### • Zone de texte « text area element »

**Description :** C'est un contrôle de saisie, employé dans les formulaires, qui permet l'écriture de plusieurs lignes de texte.

**Icône :** Aucune.

**Étiquette :**

Ligne (Rows) : Nombre de lignes de texte visibles.

Colonne (cols) : Largeur visible du texte en largeur de caractères moyens.

### 4. Composant :

#### • Page web « web page »

**Description :** Un composant page est une page web. Il peut contenir des scripts client ou serveur. Le plus souvent, le composant page est un fichier texte accessible au serveur web, mais il peut être un module compilé, chargé et exécuté par le serveur web. Dans les deux cas, le serveur web produit, à partir du composant page, un document au format HTML, qui est renvoyé en réponse à la requête du navigateur

*Icône :*



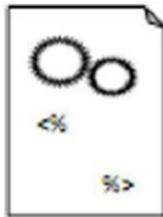
*Contraintes:* Aucune.

*Étiquettes :* Chemin (path), chemin requis pour spécifier la page web sur le serveur web. Cette valeur doit être relative au répertoire racine du site de l'application web.

### • Page ASP « ASP page »

*Description:* Une page ASP (Active Serveur Pages) est une page web qui implémente du code ASP coté serveur. Ce stéréotype n'est pertinent que dans un environnement d'application basé sur les ASP.

*Icône :*



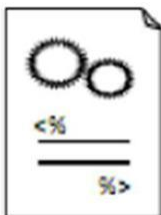
*Contraintes:* Aucune.

*Étiquettes:* Identique à celles de la page web.

### • Page JSP « JSP page »

*Description:* Une page JSP (Java serveur Page) est une page Web qui implémente du code JSP coté serveur. Ce stéréotype n'est pertinent que dans un environnement d'application basé sur les JSP.

*Icône :*



*Contraintes:* Aucune.

*Étiquettes:* Identique à celles de la page web.

- **Servlet « Servlet »**

*Description* : Un servlet est un composant Java. Il n'est pertinent que dans un environnement d'application compatible avec les servlets de SUN.

*L'icône* :



*Contraintes*: Aucune.

*Étiquettes*: Identique à celle de la page web.

## V.2. Règles de cohérence sémantique :

- **Réalisation de composant :**

En principe, les composants pages web peuvent réaliser les classes stéréotypées « serveur page », « client page », « form », « JavaScript Object », « client script Object », « frameset » et « target ». Quand l'environnement de développement associé (ASP ou JSP) est en place, les pages web peuvent réaliser, au lieu des pages « serveur page », selon le cas, des classes stéréotypées « ASP page » ou « JSP page ».

- **Généralisation**

Tous les éléments de modélisation impliqués dans eunmême généralisation doivent être du même stéréotype.

- **Association**

Une page client peut avoir au plus une relation « build » avec une page serveur, mais une page serveur peut avoir plusieurs relations « build » avec différentes pages clients. En plus de la combinaison standard d'UML, les combinaisons de stéréotypes présentées au tableau ci-dessous sont permises :

A \ DE	« client page »	« Serveur page »	« Frameset »	« target »	« form »
« client page »	« Link » « redirect » « Targeted Link »	« Link » « redirect » « target link »	« Link » « redirect » « target Link »	«dépendance »	« agrégation »
« serveur page »	« build » « redirect »	« redirect »	« build » « redirect »		
« frameset »	« frame content »		« frame content »	«Frame conte »	
« target »					
« form »	« aggregated by »	« submit »			

*Tableau : Association entre page.*

## VI. Avantages et inconvénients d'UML :

### VI.1. Les points forts d'UML :

UML est un langage formel normalisé et un support de communication performant. Son caractère polyvalent et sa souplesse en font un langage universel, il offre :

- Un gain de précision
- Un gage de stabilité
- Le cadrage de l'analyse.
- Une facilité de compréhension des représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

### VI.2. Les points faibles d'UML :

Même si UML n'est pas à l'origine des concepts objets, il en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle. Ceci dit, UML présente tous de même quelques défauts parmi les elquon citera les suivants :

- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.
- Le processus (non couvert par UML) est une autre clé de la réussite d'un projet.

## Conclusion :

Dans cette annexe, nous avons présenté les concepts de base du langage de modélisation UML qu'on a utilisé pour la conception de notre application, UML reste incontournable pour la conception des applications orientées objets.