

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur**

**Et**

**De la recherche Scientifique**  
**UNIVERSITE DE MOULOUD MAMMERY**  
**DE TIZI OUZOU**

**Faculté Génie électrique et informatique**  
**Département d'informatique**

**Les webs services pour une solution  
d'interopérabilité des applications E\_learning**

**Présenté par :**

**Mlle : KOUROGLI Akila**

**Encadré par :**

**Mr : KERBICHE Mohand**

**Année 2015/2016**

# *Remerciements*

*Je tiens à exprimer ma profonde gratitude à mon promoteur M<sup>r</sup>. Kerbiche pour son aide et son orientation pour la réalisation de ce projet.*

*Un grand merci à tous ceux qui ont contribué au bon déroulement de ce modeste travail.*

## Liste des figures

---

Figure 1.1 – Invocation d’un service Web par un client .....	3
Figure 1.2- Architecture d’invocation des services Web .....	6
Figure 1.3- Fonctionnement globale des services web .....	7
Figure 1.4 –Chemin emprunté par un message SOAP .....	11
Figure 1.5– structure d’un message SOAP .....	13
Figure 1.6- un exemple de message SOAP .....	13
Figure 1.7- Place d’un document WDSL dans l’invocation de service .....	18
Figure 1.8- Mécanisme d’accès aux services web .....	21
Figure 3.1 :Diagramme de contexte de l’application .....	50
Figure 3.2: Diagramme de cas d’utilisation « naviguer dans l’application ».....	51
Figure 3.3: Diagramme de cas d’utilisation « authentification » .....	52
Figure 3.4: Diagramme de cas d’utilisation « consulter module ».....	53
Figure 3.5 : Diagramme de cas d’utilisation « gérer module ».....	53
Figure 3.6 : Diagramme de cas d’utilisations «gestion filière».....	54
Figure 3.7 : Diagramme de cas d’utilisation « inscription apprenant » .....	55
Figure 3.8 : Diagramme de cas d’utilisation « inscription formateur » .....	55
Figure 3.9 : Diagramme de séquence « authentification ».....	63
Figure 3.10 : Diagramme de séquence « inscription apprenant » .....	64
Figure 3.11 : Diagramme de séquence « consulter cours».....	65
Figure 3.12 : Diagramme de séquence « consulter exercice » .....	66
Figure 3.14 : Diagramme de classe générale.....	69
Figure 4.1 : Interface de l’IDE Netbeans .....	77
Figure 4.2: Interface macromedia Dreamweaver .....	78
Figure 4.3 : Interface Apache Tomcat.....	79
Figure 4.4 : Interface PhpMyAdmin .....	80
Figure 4.5 : page d’accueil de l’application .....	83
Figure 4.6 : Espace administrateur .....	83
Figure 4.7 : formulaire d’inscription Apprenant .....	84
Figure 4.8 : formulaire d’inscription Formateur .....	84
Figure 4.9 : Espace Formateur .....	85
Figure 4.10 : Interface gérer les apprenants .....	86
Figure 4.11 : Interface des modules apprenant .....	86

# SOMMAIRE

---

## Chapitre 1: Services Web et l'interopérabilité

1. Introduction .....	1
1.1 Historique .....	1
1.2 Architecture de service orienté (SOA) .....	2
1.3 Concepts de l'architecture orientée services .....	2
1.4 Présentation des services Web .....	2
1.4.1 Définition d'un service Web .....	3
1.4.2 Types de services web .....	4
A. Services web SOAP .....	4
A.1 Architecture .....	5
A.2 Fonctionnement global des services web .....	6
A.3 Avantages des web services .....	8
A.4 Les technologies des Web Services .....	8
A.4.1 XML et les services Web .....	9
A.4.2 Le protocole SOAP ( <i>Simple Object Access Protocol</i> ) .....	10
A.4.3 SOAP via HTTP .....	14
A.4.4 SOAP et RPC .....	16
A.4.5 RPC et HTTP .....	17A.
4.6 Le langage WSDL .....	18
A.4.7 L'Annuaire UDDI .....	20
B. Services web REST .....	24B.
1. Définition de REST .....	24
B.2 Représentation .....	24
B.3 Architecture .....	25
B.4 Ressources .....	25
B.5 Représentation .....	26
1.5 La sécurité des web services .....	29
1.6 Interopérabilité .....	30
1.6.1 Définition .....	30
1.6.2 L'interopérabilité dans les technologies existantes .....	31

# SOMMAIRE

---

1.6.3 Le respect des normes .....	31
1.6.4 Initiatives WS-I .....	31
Conclusion.....	32
<b>Chapitre 2: E-Learning</b>	
2. Introduction .....	33
2.1 E-Learning.....	33
2.1 Apport des nouvelles technologies à l'enseignement.....	34
2.3 Les formes d'e -Learning .....	34
2.3.1 Les formes de communication .....	34
2.3.2 Les formes d'enseignement.....	36
Les avantages et les faiblesses du e-Learning .....	36
2.4.1 Les avantages du e-Learning .....	36
2.4.2 Les faiblesses d'e-Learning.....	38
2.5 Plates -formes d'enseignement à distance.....	39
2.6 Les principaux critères que doivent vérifier les plates-formes.....	40
2.7 Outils de communication .....	47
2.8 Acteurs de la plate-forme LMS .....	42
2.9 Quelques fonctionnalités de la plate-forme LMS .....	43
2.10 E-Learning en Algérie .....	45
Conclusion.....	47
<b>Chapitre 3 : Analyse et conception</b>	
3.Introduction .....	48
3.1Présentation de l'UML .....	48
3.2 Objectif du projet .....	48
3.3 Identification des acteurs et leurs besoins fonctionnels .....	49
3.4 Diagrammes représentatifs .....	50
3.4.1 Le diagramme de contexte de l'application .....	50
3.5 Description textuelle des cas d'utilisation.....	56
3.6Description graphiques des cas d'utilisation.....	62
3.7 la conception objet.....	68
Conclusion.....	70

# SOMMAIRE

---

## **Chapitre 4 : Réalisation**

4 Introduction .....	71
4.1 Le modèle relationnel .....	71
4.2 Description des outils de développement .....	75
4.2.1 Langage de programmation Java.....	75
4.2.2 Langages du web .....	80
4.3 Conclusion générale .....	87

Depuis l'apparition de l'informatique, l'être humain a toujours essayé d'exploiter cette science pour automatiser ses tâches quotidiennes de gestion, de communication, et d'enseignement...

Au cours de ces dernières années, les nouvelles technologies de l'information et de la communication ont connu un bouleversement marqué par l'apparition de l'Internet. Avec l'avènement de l'Internet, les distances se sont réduites et la communication est devenue aisée. Les outils de communication permettent aujourd'hui d'organiser des groupes, dont les membres peuvent être géographiquement dispersés, l'Internet offre plusieurs services : courrier électronique, forum de discussion, chat, commerce électronique, publication hypermédia, accès aux bases de données distantes, etc.

Le secteur public et le secteur privé sont confrontés à l'évolution rapide de cette nouvelle technologie qui modifie non seulement les moyens de communication et d'échange du monde économique, mais également de ceux des individus. Pour apprendre autrement et enseigner différemment, aujourd'hui une forte demande en temps réel de l'information ne cesse d'augmenter, avec un essor rapide de l'enseignement ou formation à distance ou (E-Learning), qui apporte d'avantage de mobilité et de possibilités d'apprentissage, sans se déplacer de chez soi ou de son lieu de travail.

Les travaux exposés dans ce mémoire consistent à créer des services web qui permettant l'interopérabilité des applications E-learning.

- ✓ Offrir un ensemble d'outils collaboratifs qui permettent une meilleure interaction entre apprenants et formateurs telle que : le chat, les forums de discussions et le service de messagerie
  - ✓ Gérer et administrer cette plate-forme et donc assurer en particulier les fonctions suivantes : La gestion des membres de la plate-forme .

Ce mémoire comporte quatre chapitres, et deux annexes :

- Le premier chapitre présente des généralités sur les WEB services et l'interopérabilité.
- Le deuxième chapitre présente le E-learning .
- Le troisième chapitre présente la conception de notre plate-forme.
- Enfin le quatrième chapitre décrit l'environnement de développement et l'implémentation de notre application.

## 1. Introduction :

Les Web Services sont le résultat de la collaboration exceptionnelle des joueurs majeurs des technologies de l'information qui se sont entendus sur un certain nombre de protocoles et d'approches qui favoriseront l'interopérabilité entre les plates-formes, les systèmes d'exploitation, les langages et les programmes. De ce fait, les *Web Services* sont donc plus un phénomène ou un concept, voire un contexte, qu'une technologie. Il est évident que les *Web Services* reposent sur diverses technologies, mais ils représentent surtout une volonté commune des manufacturiers, des organismes de standards et des utilisateurs de développer des outils permettant une réelle interopérabilité.

### 1.1 Historique

Les services web doivent leur origine à l'informatique distribuée et au développement du Web. Au cours des années 1980, les composants ont vu le jour marquant ainsi la fin de l'ère des applications monolithiques hébergeant l'ensemble des fonctionnalités nécessaires, et une nouvelle génération d'applications a vu le jour. Celle-ci est caractérisée par le fait que le code n'est plus compliqué en une seule unique entité, ce qui caractérisait les applications \*monolithiques, mais scindé en plusieurs composants compliqués indépendamment avec des interfaces et une sémantique bien définies. Ces composants sont construits pour être réutilisables et pour communiquer via une infrastructure fournie par le système d'exploitation, ouvrant ainsi la porte à leur exploitation de façon externe. Pour cela, il suffit que l'application appelante connaisse les appels de l'infrastructure par composants et l'interface des composants qu'elle veut appeler et utiliser.

Au fil des années, trois modèles de programmation distribués à base de composants se sont imposés. Il s'agit de DCOM/COM (Distributed Component Object Model/Component Object Model) de Microsoft, JAVA/RMI (Remote Method Invocation) de Sun Microsystems et CORBA (Common Object Request Broker Architecture) de l'OMG (Object Management Group).

Néanmoins, les modèles de composants distribués sont généralement confirmés dans un fonctionnement au sein de réseaux maîtrisés, comme par exemple un intranet d'entreprise. Ils sont complexes à mettre en œuvre, péniblement interopérables et peu compatibles avec les pare-feux. Ajoutons à cela qu'ils s'appuient sur des solutions propres à un éditeur.

L'insuffisance des technologies de programmation distribuée existantes, l'évolution d'internet et le succès du Web, ainsi que l'émergence du langage XML comme norme de structuration de données ont tous contribué à favoriser l'essor des services Web.

## **1.2 Architecture orientée services (SOA):**

SOA is not a concrete architecture it is something that leads to a concrete architecture. You might call it a style, paradigm, concept, perspective, philosophy, or representation. That is, SOA is not a concrete tool or framework you can purchase. It is an approach, a way of thinking, a value system that leads to certain concrete decisions when designing concrete software architecture.

Nicolaï M.Josuttis veut dire que SOA n'est pas une architecture, mais c'est un paradigme qui nous conduit à une architecture. C'est une approche, une voie de pensée qui nous conduit à des certaines décisions.

## **1.3 Concepts de l'architecture orientée service :**

L'architecture orientée service est axée autour de deux concepts fondamentaux qui sont :

### ➤ **Services :**

C'est-à-dire une fonction que l'on peut interroger à l'aide d'une requête et qui fournit une ou plusieurs réponses.

### ➤ **Interopérabilité :**

Il permet la propagation des fonctionnalités des services web via des systèmes hétérogènes.

## **1.4 Présentation des services Web :**

### **1.4.1 Définition d'un service Web :**

Un service Web est un composant logiciel qui interagit avec d'autres composants logiciels autonomes au moyen de protocoles universels, via le réseau Internet ou n'importe quel réseau local basé sur les protocoles standard du Web. Les applications clientes accèdent aux services Web à travers les protocoles Web (http, SMTP, ou autres) en se servant du protocole SOAP comme http et XML, permet aux applications clientes être implémentées dans n'importe quel langage de programmation, avec n'importe quel modèle de composants, et tourner sur n'importe quel système d'exploitation.

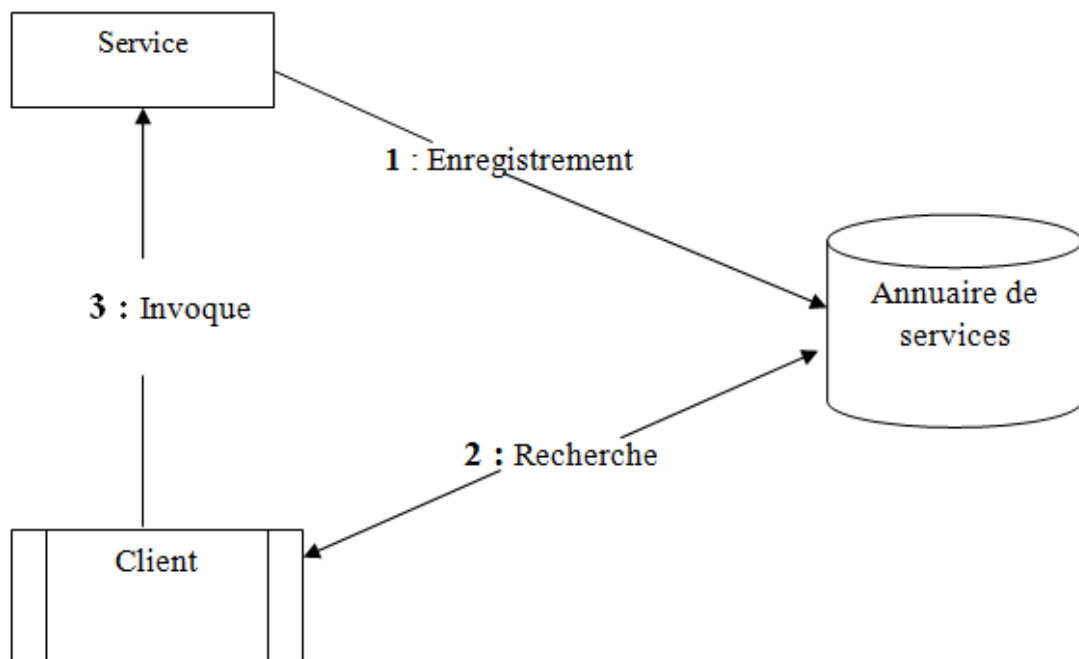
La technologie des services Web, initié par IBM et Microsoft puis normalisée par le W3C (World Wide Web Consortium), est maintenant acceptée par l'ensemble des acteurs de

l'industrie informatique sans exception. C'est surtout ce point qui fait des services Web une technologie révolutionnaire et qui rend aussi populaires.

Les services web sont mis à disposition à partir d'un serveur Web ou serveur d'applications à l'intention des utilisateurs ou d'autres applications interconnectées sur Internet. Ils présentent les deux caractéristiques suivantes :

- Un enregistrement facultatif auprès d'un service de recherche.
- Une interface publique avec laquelle le client invoque le service.

Chaque service Web peut s'enregistrer auprès d'un dépôt central de telle sorte que le client puisse le trouver lors d'une consultation de l'annuaire de services. Une fois le service localisé, le client en récupère la référence. Le client utilise alors le service en invoquant diverses méthodes que ce dernier implémente et publie, ceci au moyen d'une interface publique. Ainsi, chaque service doit publier son interface à l'intention de ses clients :



**Figure 1.1 – Invocation d'un service Web par un client.**

Le scénario d'invocation d'un service Web par un client sera explicité en détails plus loin dans ce chapitre.

Les services Web favorisent une approche modulaire des applications qui se caractérise par l'apparition de nouveaux modèles économiques. Plutôt que de vendre sa suite de logiciels en bloc, un éditeur commercialisera un composant fonctionnel qui sera invocable à travers le Web par une entreprise cliente. Ainsi, cette dernière assemblera les fonctions dont elle a besoin pour concevoir une application sur mesure. Elle sera à la même d'invoquer dynamiquement les systèmes informatiques des partenaires à travers le web pour fabriquer une offre personnalisée afin de répondre aux besoins de ses clients. Les services Web ont donc pour but d'utiliser Internet comme une infrastructure de communication pour le développement d'applications réparties à grande échelle.

#### **1.4.2 Types de services web :**

On distingue deux types de services web : **SOAP** et **REST**.

##### **A. Services web SOAP :**

SOAP (acronyme de Simple Object Access Protocol) est un protocole de RPC orienté objet bâti sur XML. Il permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également se faire par un autre protocole, comme SMTP.

Le protocole SOAP est composé de deux parties :

- Une enveloppe, contenant des informations sur le message lui-même afin de permettre son acheminement et son traitement.
- Un modèle de données, définissant le format du message, c'est-à-dire les informations à transmettre.

SOAP a été initialement défini par Microsoft et IBM, mais il est devenu une référence depuis une recommandation du W3C, utilisée notamment dans le cadre d'architectures de type **SOA**

(*Service Oriented Architecture* ) pour les services web WS.

### **A.1 Architecture :**

Pour comprendre le fonctionnement d'une architecture de services web SOAP, il faut commencer par revoir certains principes. Si l'on reprend la définition de Mark Colan, web service and XML Chief Advocate chez IBM, les web services SOAP sont des « applications modulaires basées sur internet qui exécutent des tâches précises et qui respectent un format spécifique ». Les services web donc ne sont que des applications modulaires qui peuvent être présentées, situées et invoquées dans un réseau et ce automatiquement. Ainsi, les applications peuvent faire appel à des fonctionnalités situées sur d'autres machines dans d'autres applications. Au final, on peut affirmer que le but initial d'un service web est de rendre possible l'utilisation d'un composant applicatif de façon distribuée.

Il s'agit maintenant d'identifier chaque acteur de ses web services et de comprendre comment ils interagissent les uns avec les autres. Les trois éléments les plus importants des services web sont Service Provider, Service Registry et Service Requester.

- **Service Provider :** C'est le fournisseur de services, dont il en est aussi le propriétaire. Il héberge un module logiciel, implémentant un ou plusieurs services Web, accessible par un client Web via le protocole SOAP.
- **Service Registry :** Annuaire contenant les descriptions des services Web publiés par les fournisseurs de services. Il est géré sur un serveur niveau application, entreprise ou mondial.
- **Service Requester :** Demandeur du service, qui est un responsable d'application cliente se liant à un service et invoquant ses fonctions par messages XML échangés via le protocole SOAP.

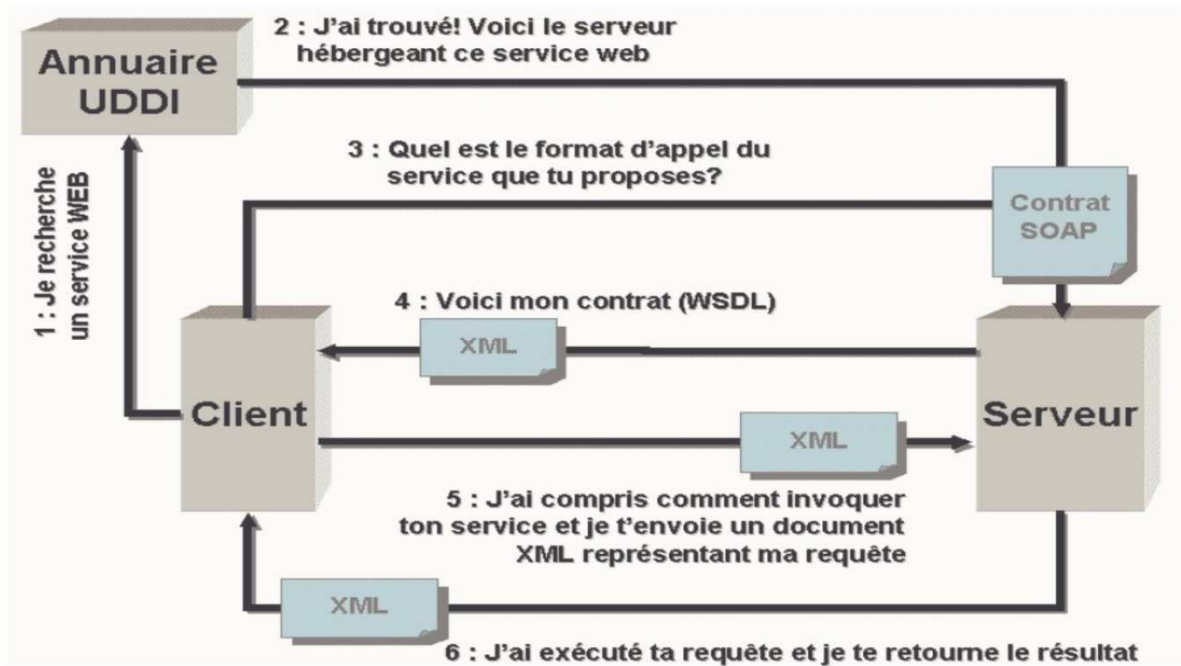


Figure 1.2- Architecture d'invocation des services Web

La figure 1.2 nous montre que les services Web peuvent être vus comme des composants logiciels, décrits au niveau de l'annuaire des services. Les demandeurs de services découvrent les caractéristiques fonctionnelles et techniques des services offerts dans l'annuaire. En suite, ils y entrent en liaison en invoquant les différents services depuis leur code applicatif pour réaliser leurs applications dont les services sont les briques de base. L'architecture permet aussi qu'un service Web puisse invoquer un autre service Web.

## A.2 Fonctionnement globale des services web

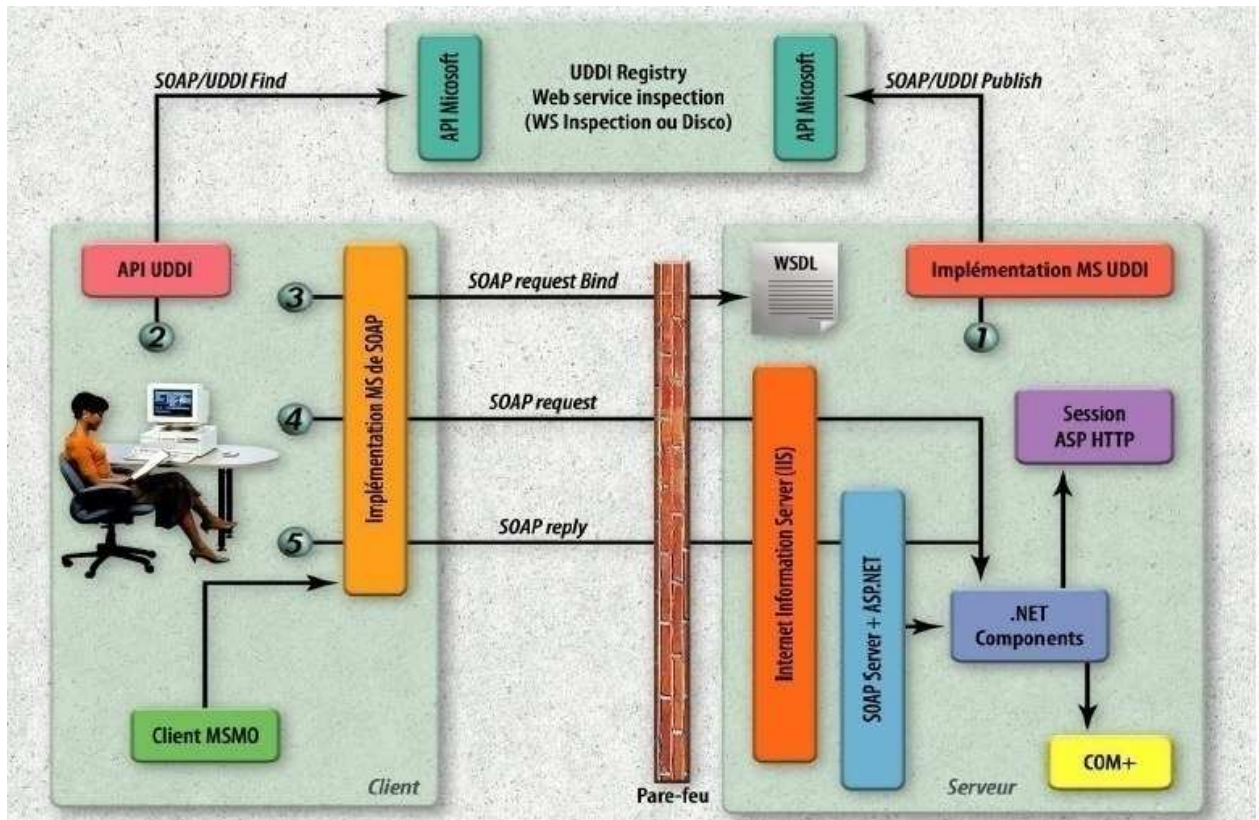


Figure 1.3- Fonctionnement des services web

Dans le scénario de fonctionnement des services Web, un fournisseur de services (Service Provider) héberge un module logiciel (plate forme d'exploitation) implémentant un ou plusieurs services Web accessibles via le réseau. Ce module implique des interfaces de connexion permettant : d'être à l'écoute d'une application cliente (généralement un navigateur Web), interrogation dans le cas de la récursivité (un service Web peut appeler un autre service Web) et la publication des fichiers WSDL (enregistrement de la description du service Web sur l'annuaire UDDI). L'annuaire des services est consultable sous plusieurs facettes (pages jaunes, pages vertes, pages blanches), cela dépend des critères transmis dans la requête de recherche (Find) émise du côté de l'application cliente. Ceux-ci peuvent englober les caractéristiques fonctionnelles, caractéristique techniques des services Web, taxinomie utilisée pour classifier l'entreprise qui l'a publié selon son activité, nom ou même la description de l'entreprise qui l'a publié. Suite à cette requête un fichier WSDL (réponse) sera renvoyé à l'application cliente (l'annuaire UDDI suit le modèle requête réponse), qui le reçoit via son interface de connexion WSDL, ce fichier contient des informations sur le service Web (interfaces des services, emplacement sur le Web) et l'entreprise qui l'a publié.

A ce niveau l'application cliente pourra se connecter physiquement à un emplacement bien précis sur le Web pour interroger le service Web qu'elle vient de découvrir via un protocole de transport (HTTP). Pour ce faire l'application cliente fait intervenir un processus de sérialisation/désérialisation (Encodeur SOAP) pour générer les messages SOAP, encapsulé dans une requête http. Le service Web réagi à son tour, en envoyant une réponse SOAP via le protocole http.

### **A.3 Avantages des web services**

De nos jours, la technologie des web services est populaire et couramment utilisée car elle offre des avantages intéressants pour les utilisateurs des systèmes distribués:

- Les web services réduisent le temps de mise en marche des services offerts par les diverses entreprises.
- Les web services permettent à des programmes écrits en des langages différents et sur des plates-formes différentes de communiquer entre eux par le biais de certaines normes. En d'autres termes, les web services permettent une meilleure interopérabilité entre les logiciels.
- Les web services utilisent des normes et protocoles ouverts.
- Grâce au protocole HTTP, les web services peuvent fonctionner malgré les pare-feu sans pour autant nécessiter des changements sur les critères de filtrage.
- Les protocoles et les formats de données sont offerts, le plus possible, en format texte pour que la compréhension du fonctionnement des échanges soit plus intuitive.
- Grâce aux web services, les coûts sont réduits par l'automatisation interne et externe des processus commerciaux.

### **A.4 Les technologies des Web Services :**

XML, SOAP (*Simple Object Access Protocol*), WSDL (*Web Services Description Language*) et UDDI (*Universal Description, Discovery and Integration*) sont les technologies dominantes des *Web Services*. Une pléthore d'autres technologies viendront, au fil du temps, garnir l'architecture des *Web Services*. Sans entrer dans tout les détails techniques, nous vous

exposons ici les grandes lignes de chacune de ces technologies et de celles, qui à court et moyen terme, risquent de se positionner à la suite.

Fondamentalement, SOAP, WSDL et UDDI sont des technologies issues de l'intérêt parmi des membres de la communauté Internet à développer un mécanisme pour échanger des documents XML sur le Web entre systèmes d'information.

#### **A.4.1 XML et les services Web :**

Le but de l'informatique distribuée était de permettre aux systèmes d'informations de faire apparaître leurs rouages (bases de données, services distribués, applications, etc.) comme des composantes fonctionnant avec une API commune. Cependant, les systèmes distribués à base de composants COBRA, JAVA/RMI et COM/DCOM ne répondaient que partiellement à ce but, du fait que leur implémentation utilisait des interfaces spécifiques, ce qui conduisait à des solutions propriétaires qui réduisaient la production coopérative des documents. L'arrivée des services Web XML qui proposaient un modèle objet distribués proche de celui de CORBA et DCOM sur le plan technique, a fait des systèmes d'informations de simples assemblages de composants de services XML indépendants capables de produire et de consommer des documents XML et qui communiquent entre eux à travers un canal de communication. Les services Web XML reposent sur une architecture objets distribués XML/HTTP, et ne nécessite pas de disposer d'une API commune, ils étendent les modèles JAVA, COM, CORBA à Internet et assurent l'extension de l'interopérabilité des composants existants. Le déploiement des services Web (SOAP, WSDL, UDDI) est assuré par le langage XML, qui est utilisé comme un format d'importation et d'exportation d'informations et dans les appels RPC pour l'invocation des procédures à distance. L'apport du langage XML et son impact sur l'informatique distribuée en particulier les services Web, sont résumés dans les points suivants :

- XML qui est un langage simple, définit un standard de description de données. Les messages XML sont auto descriptifs ils permettent le traitement des données sans en connaître le format. De tels messages sont indépendants des systèmes d'exploitation, des plates-formes, des langages de programmation et des formats d'affichage. Ceci facilite l'échange de données entre les différents partenaires.
- XML et les technologies qu'y sont liées (SOAP, WSDL, UDDI) forment un Framework des services Web. Une association qui révolutionné la manière dont les

applications communiquent entre elles. Ce langage a fait des services Web des applications à accès universel à partir de n'importe quel ordinateur ou appareil, indépendamment de sa technologie propriétaire de base.

- Les services Web XML ont fait d'Internet un réseau orientés services, sur lequel à peu près tous les services ou applications dont nous pourrions avoir besoin s'y trouvent.
- XML suivi par tous les éditeurs (Microsoft, Sun, Oracle, etc.) ce qui rend disponible les outils nécessaires pour le développement des services Web, tel est le cas pour les plates-formes J2EE de Sun Microsystems et .Net de Microsoft.

#### **A.4.2 Le protocole SOAP (*Simple Object Access Protocol*) :**

SOAP est un protocole de la famille XML servant à l'échange d'informations dans un environnement distribué et décentralisé. Il est considéré comme la technologie la plus importante des *Web Services*. Le standard SOAP a été proposé au W3C par Microsoft, IBM, Lotus, DevelopMentor et Userland.

Le standard SOAP définit trois éléments composant un message : l'enveloppe, l'entête du message et le corps du message. L'enveloppe définit le cadre pour décrire ce qui est dans le message et comment le traiter. Les règles d'encodages sont placées dans l'en-tête et servent à exprimer et définir le mécanisme de représentation des données. Le corps du message permet de transmettre les requêtes et les réponses entre les systèmes.

Plus de 80 implantations de la spécification SOAP v1.1 ont été développées jusqu'à présent. Cela démontre déjà la simplicité, la popularité et la justesse de l'approche qu'il offre pour transporter des données sur le Web.

#### **❖ Définition du protocole SOAP :**

SOAP est standard du consortium W3C, définissant un protocole qui assure des appels de procédures à distance (RPC) s'appuyant principalement sur le protocole HTTP et sur XML, mais aussi sur d'autres protocoles de communication (SMTP, POP, etc.).

L'objectif du protocole SOAP est de permettre l'invocation de services Web qui offrent des services sur des ports de connexion. Pour cela, il offre à l'application cliente les outils nécessaires pour invoquer des services distants en lui donnant l'illusion qu'ils sont locaux. Comme nous l'avons signalé SOAP est un protocole indépendant de toute plate-forme et de tout langage de programmation : il réalise le codage des appels en XML, qui est un codage

universel. Sur ce plan, il est beaucoup plus général par rapport à ce qu'offraient CORBA (IIOP/CDR), JAVA (RMI/JRMP), DCOM (DCE/ENDR) et RDA (SQL/XDR), tous spécifiques comme l'indiquent les sigles associés.

En fait, SOAP peut être vu comme un simple codage universel des appels et retours de procédure de type RPC. Il apporte cependant, un peu plus de fonctions.

#### ❖ Caractéristiques du protocole SOAP :

- SOAP permet une normalisation des échanges de données. Les données sont encodées en XML et échangées par des appels de procédures à distance en utilisant HTTP/SMTP/POP comme protocole de communication.
- Simple, extensible et permet le diagnostic des erreurs.
- Message unidirectionnel (Requête -> Réponse).
- Fonctionne de manière synchrone et asynchrone.
- Indépendant de la plate-forme et du langage.
- N'est pas perturbé par un pare-feu.

#### ❖ Modèle SOAP d'échange de messages :

La spécification SOAP définit un modèle d'échange de messages. Elle repose sur trois concepts de base : les messages sont des documents XML ; ils voyagent d'un émetteur vers un récepteur ; les récepteurs intermédiaires peuvent former une chaîne.

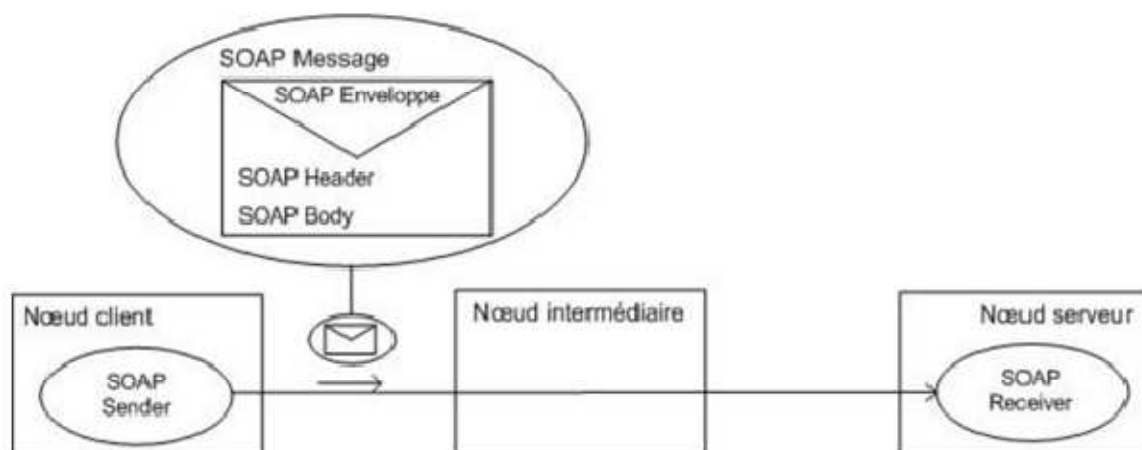


Figure 1.4 –Chemin emprunté par un message SOAP

### ❖ Document XML comme message

Les messages SOAP sont en langage XML, ce qui confère à SOAP plusieurs avantages par rapport aux autres protocoles de messagerie. Les messages XML peuvent être rédigés et lus par un développeur au moyen d'un éditeur de texte ce qui simplifie considérablement le processus de débogage par rapport à un autre protocole binaire complexe.

### ❖ Emetteurs et récepteurs

L'échange d'un message SOAP implique deux parties : un émetteur et un récepteur. Le message transite de l'émetteur vers le récepteur. Cette opération qui représente la plus petite unité de travail, s'avère insuffisante dans de nombreux cas. Il est plus souvent nécessaire d'échanger des messages par paires de requête/réponse. Cette méthode est celle utilisée par le protocole SOAP avec le transport HTTP et/ou la convention RPC. Les échanges de messages en sens unique sont également possibles. On peut parvenir à composer des échanges plus complexes en se basant sur un échange de message à sens unique entre un émetteur et un récepteur, ce qui nous donne la possibilité de construire des chaînes de messages.

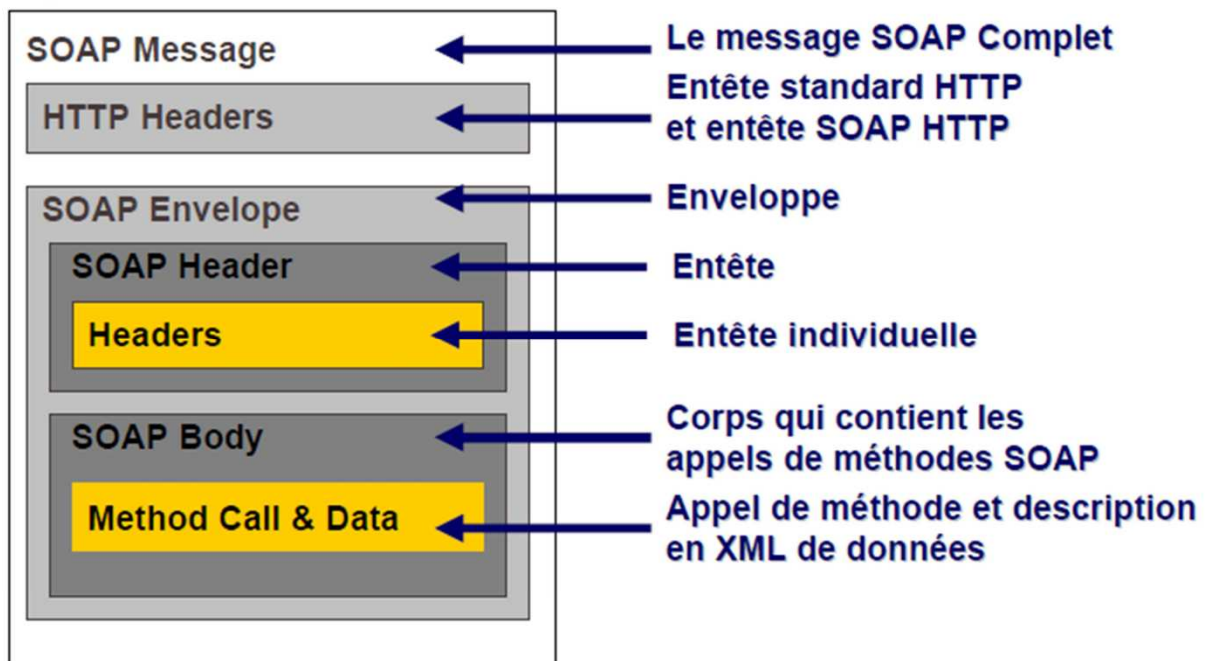
### ❖ Chaînes de messages

Les messages SOAP ne suivent pas nécessairement un modèle client-serveur classique. Ils pourraient être échangés de cette manière ou traités par une chaîne d'entités logiques. Ce concept d'entité logique qui effectue le traitement d'un message SOAP est appelé **point final**. Les points finaux sont des récepteurs de messages SOAP. Chaque point final est responsable de l'examen d'un message et de l'extraction de la partie qui lui a été adressée pour la traiter.

Les points finaux peuvent fonctionner à la fois comme émetteur et comme récepteur. Cette faculté permet la mise en place d'un système de traitement peut éventuellement être effectué à chaque étape. Les points finaux qui fonctionnent à la fois comme émetteur et comme récepteurs et transmettent à un autre point les messages qu'ils ont reçus sont appelés **intermédiaires**. Les intermédiaires et les points finaux sont identifiés par des URI (Universal Resource Identifier).

## ❖ Structure des messages SOAP

La structure de message SOAP est représentée ci-dessous



**Figure 1.5– structure d'un message SOAP.**

Un message SOAP comprend une charge utile c'est-à-dire les informations spécifiques à l'application.

Voici un exemple de message SOAP :

```
<soap:Envelope xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
  Soap:encodingStyle=http://schemas.xmlsoap.org/soap/encoding/>
  <soap:Header>
    <h:from xmlns:h=http://www.wrox.com/header>SoapGuy@wrox.com</h:from>
  </soap:Header>
  <soap:Body>
    <w:GetSecretIdentity xmlns:w=http://www.wrox.com/heroes/>
    <w:codename>XSLT-Man</w:codename>
  </w:GetSecretIdentity>
```

```
</soap:Body>  
</soap:Envelope>
```

**Figure 1.6- un exemple de message SOAP**

Les messages SOAP dépendent fortement des espaces de noms XML. Tous les éléments présentés ici ont un préfixe d'espace de noms. En effet pour qu'un message SOAP puisse transporter n'importe quelle charge utile XML supplémentaire il faut déterminer la portée de chacun de ses éléments de manière à éviter tout conflits sur les noms de ces éléments.

L'ensemble des éléments du message qui sont associés à l'espace de noms soap constitue les éléments standards d'un message SOAP, de même que les attributs. Les autres éléments peuvent être reliés aux extensions ou à la charge utile du message. Notre exemple contient trois éléments SOAP standard : **Envelope**, **Header** et **Body**. Il existe un quatrième élément standard qui n'apparaît pas dans notre exemple, l'élément **Fault**.

## ❖ Le transport

Le transport est la méthode qui permet d'acheminer un message SOAP d'un émetteur à un récepteur. La spécification SOAP a été développée de manière à permettre de séparer entre la définition du message et son transport.

### A.4.3 SOAP via HTTP :

HTTP constitue un excellent transport pour SOAP en raison de sa popularité. La combinaison de HTTP, protocole de transport pour le Web, et de SOAP, meilleur candidat au titre format de messagerie standard, permet d'obtenir un outil très puissant.

Il existe deux règles à respecter pour l'utilisation de HTTP comme transport de messages SOAP :

- Le mécanisme d'envoi d'un message SOAP via HTTP est la méthode **HTTP POST** standard. Un **HTTP POST** envoie un bloc de données à un URI donné sur le serveur Web. Dans le cas de messages SOAP, ce bloc de données est le message SOAP lui-même.
- Le message étant en XML, l'en-tête **Content-Type** du **HTTP POST** doit être **text/xml**. La réponse au message s'il y'en a une, est envoyée dans la réponse HTTP.

Voici un exemple de message SOAP transporté via HTTP :

- 1- POST/endpoint.asp HTTP/1.1
- 2- Content-Type text/XML
- 3- Content-Lenght : ###
- 4- SOAPAction : "urn :wroxheroes"
- 5- <soap :Envelope xmlns :soap="http :// schemas.xmlsoap.org/soap/envelope/"  
    Soap:encodingStyle="http://shemas.xmlsoap.org/soap/encoding/">  
    <soap:Header>  
    <h:from xmlns:h="http://www.wrox.com/header">  
        SoapGuy@wrox.com  
    </h:from>  
    </soap:Header>  
    <soap:Body>  
    <w:GetSecretIdentity xmlns:w="http:// www.wrox.com/heroes/">  
        <CodeName>XSLT-Man</CodeName>  
    </w:GetSecretIdentity>  
    </soap:Body>  
    </soap:envelope>

La première ligne contient la méthode **HTTP POST**, l'URI indiquant la position du point final ainsi que la version du protocole HTTP utilisée dans cette requête.

Les trois lignes suivantes sont des en-têtes HTTP. **Content-Type** indique le type MIME du contenu du POST : tous les messages SOAP doivent utiliser **text/xml**. **Content-Lenght** définit la taille en octets du contenu. Le dernier en-tête **SOAPAction** est spécifique à SOAP.

### ❖ En-tête SOAPAction:

L'en-tête SOAPAction, indique aux serveurs qu'un HTTP POST donné contient un message SOAP et la valeur de cet en-tête contient un URI informant de l'objet de ce message, c'est-à-dire la destination du message. Cela permet aux pare-feux et autres serveurs d'exécuter un traitement conditionnel, basé sur la présence de l'en-tête SOAPAction.

La version 1.1 imposait la présence de l'en-tête SOAPAction, même si cet en-tête peut être vide. Si tel est le cas, l'objectif du message est déduit de l'URI de la cible du POST. Dans les versions récentes du protocole HTTP, cet en-tête est retiré.

### ❖ Codes de statut

HTTP renvoie des informations relatives au statut sous forme de code de statut. Ces codes sont des nombres entiers, divisés en classes de 100.

- Tout code compris entre 200 et 299 indique une opération réussie
- Le code 500 représente une erreur interne serveur

Dans le cas d'une opération réussie, la première ligne de la réponse renvoyée sera :

**HTTP1.1/ 200 ok**

### A.4.4 SOAP et RPC :

La convention RPC peut être décrite comme une technique de sérialisation des appels de procédures à distance et des réponses comme messages SOAP.

Comme son partenaire HTTP, RPC SOAP utilise un modèle requête-réponse pour l'échange de messages. L'exécution d'un appel de procédure à distance avec le protocole SOAP nécessite uniquement la construction d'un message SOAP. Le message de requête SOAP envoyé au point final représente l'appel, et le message de réponse SOAP représente le résultat de cet appel.

### ❖ L'appel

L'exécution d'un appel de procédures à distance avec le protocole SOAP nécessite seulement la construction d'un message SOAP. La charge utile de ce message contient l'appel de méthode sérialisé, sous forme d'une structure dont le nom de l'élément correspond au nom de la méthode, et les éléments enfants de cette structure sont les paramètres entrants de cette méthode.

Considérons une méthode baptisée GetNumeroMois qui retourne le numéro du mois qu'elle reçoit à l'argument. L'espace de noms de cette méthode est `http://tempuri.org/`, et sa signature est :

```
Int GetNumeroMois(string mois) ;
```

L'appel de méthode sérialisé demandant le numéro du mois au moyen de la chaîne « Mars » ressemblerait à ce qui suit :

```
<q :GetNumeroMois xmlns :q="http://tempuri.org/">
```

```
<q :mois xsi :type="xsd :string">Mars</q :mois>
```

```
</q :GetNumeroMois>
```

### ❖ Réponse

Les résultats de l'appel sont renvoyés dans le message de réponse SOAP. La charge utile du message de réponse contient également la valeur résultant de la méthode.

Le résultat sérialisé de l'appel de la méthode précédente ressemblerait à ce qui suit :

```
<q :GetNumeroMoisResponse xmlns :q="http://tempuri.org/">
```

```
<q :moisNumero xsi :type="xsd :int">3</q :mois>
```

```
</q :GetNumeroMoisResponse>
```

Notons qu'aucune directive n'existe concernant le nom de la structure de réponse de la méthode, mais l'usage veut que l'on utilise le nom de l'appel de la méthode préfixé par **Réponse**.

### A.4.5 RPC et HTTP :

Pour exécuter un appel de procédures à distance, il faut un moyen de déplacement du message jusqu'à la position à distance. C'est dans cette situation, lorsqu'on combine RPC et HTTP pour appeler les services Web, que le protocole SOAP montre sa puissance.

Supposons que nous devions appeler une procédure à distance sur un serveur afin de retourner le code postal d'une commune. Notre hypothétique service Web serait à l'adresse <http://www.retournecodepostal.com/call.asp>. La méthode pourrait être invoquée à l'URL [www.retournecodepostal.com/call.asp](http://www.retournecodepostal.com/call.asp). Elle serait associée à l'URI de l'espace de noms <http://www.retournecodepostal.com/methode>, et l'en-tête SOAPAction pour cette méthode serait `urn :retournecodepostal`.

La signature de cette méthode : string RetourneCodePostal (string willaya, string commune) .

#### A.4.5 Le langage WSDL (Web Service Description language) :

##### ❖ Présentation de WSDL :

WSDL (Web Service Description language) est un format XML permettant de décrire un service Web, et ce, en définissant les opérations exposées par ce service et le format qu'un client doit employer pour le solliciter. Il met en place un contrat entre un client et le service, pour rendre la communication efficace entre les deux parties. Le client doit solliciter le service en lui envoyant une requête SOAP correctement formatée.

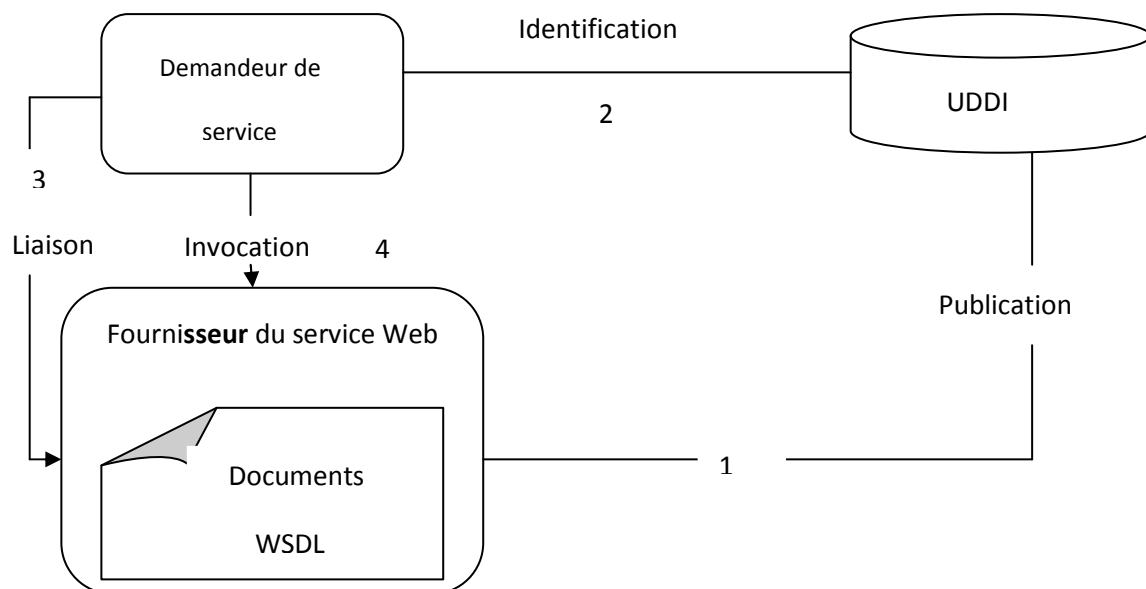


Figure 1.7- Place d'un document WSDL dans l'invocation de service

WSDL est né d'un effort commun entre IBM, Microsoft et Ariba. Il repose sur Http, MIME et SOAP comme mécanisme d'invocation d'objets distants, et sur la spécification des schémas XML comme système de type, tout en permettant l'utilisation d'autres langages de définition de type.

## ❖ Pourquoi WSDL ?

Pour correspondre à leur image de composants disponibles via Internet et utilisables sur n'importe quelle plate-forme et à partir de n'importe quel langage, les services Web doivent être auto descriptifs. C'est là que le langage WSDL entre en action. En effet, il permet de décrire ce qu'un service fait et comment les clients peuvent l'exploiter. Ainsi, tout service décrivant ses interfaces avec WSDL peut être invoqué, par voie de programmation, par n'importe quel client sans que ce dernier ne soit contraint de connaître les détails d'implémentation du service, ni la plate-forme ou le système d'exploitation sur le quel il tourne.

Le modèle d'échange supporté par WSDL est inspiré de la navigation maritime : des messages contenus dans des enveloppes sont échangés de port en port. Pour décrire les services, WSDL introduit, en plus de la notion de service et de définitions de types de données avec les schémas, les notions port, liaison et messages.

## ❖ Syntaxe WSDL

WSDL fournit une grammaire de description des services sous forme d'un ensemble de points finaux qui échangent des messages. Un document WSDL se compose de définitions présentant un service comme un ensemble de points finaux de réseau, ou ports. Chaque port est associé à une liaison spécifique. C'est cette liaison qui détermine comment un ensemble abstrait d'opérations et de messages est lié à un protocole particulier. Une liaison établit une correspondance entre un protocole donné et un type de port. Un type de port se compose d'un ensemble d'opérations, qui représentent un ensemble abstrait de « choses » que peut faire le service. Chaque opération est constituée d'un ensemble de messages abstraits, qui représentent les données communiquées au cours de l'opération. Chaque message contient une ou plusieurs données définies par des types.

La spécification d'un document WSDL offre un jeu de six définitions. La racine du document est <définitions>.

Les six éléments imbriqués sont :

- <types>
- <message>
- <portType>
- <binding>

- <port>
- <service>

#### **A.4.6 L'Annuaire UDDI :**

##### **❖ Définition**

L'annuaire UDDI permet aux utilisateurs de faire la publication des applications (des web services). En effet, UDDI peut contenir des informations sur les fournisseurs et les services qu'ils publient.

L'inscription d'un fournisseur de services à l'annuaire UDDI lui permet de se présenter et présenter ses services, l'adoption de cet annuaire par les fournisseurs permet l'accélération des échanges surtout les échanges commerciaux de type B2B. L'enregistrement des web services dans un annuaire UDDI, s'effectue auprès d'un opérateur en accédant au site web de ce dernier à partir d'un navigateur ou d'un outil intégré dans l'environnement de développement. Des recherches précises peuvent s'effectuer dans l'annuaire par catégorie de fournisseurs en utilisant des standards de taxinomie et d'identification de fournisseurs.

##### **❖ Pages de l'annuaire**

Les données retrouvées dans l'UDDI sont divisées en trois catégories :

**Pages blanches** : le référentiel comporte des informations sur les fournisseurs de services telles que le nom et les coordonnées du fournisseur.

**Pages jaunes** : le référentiel comporte des critères de catégorisation de services, les critères de catégorisation s'appuient sur des standards de classification de fournisseurs, les services sont décrits par des documents au format WSDL. Un fournisseur peut disposer de plusieurs entrées dans l'annuaire pour l'ensemble des différents services et produits qu'il publie.

**Pages vertes** : le référentiel comporte des informations techniques (WSDL) détaillée sur les services fournis telles que les informations sur les processus métier, les descriptions de services et les informations de liaison sur les services.

Les annuaires offrent des mécanismes standards de classifications. Ils peuvent être publics ou privés. Les annuaires publics sont hébergés par des sociétés comme IBM ou Microsoft.

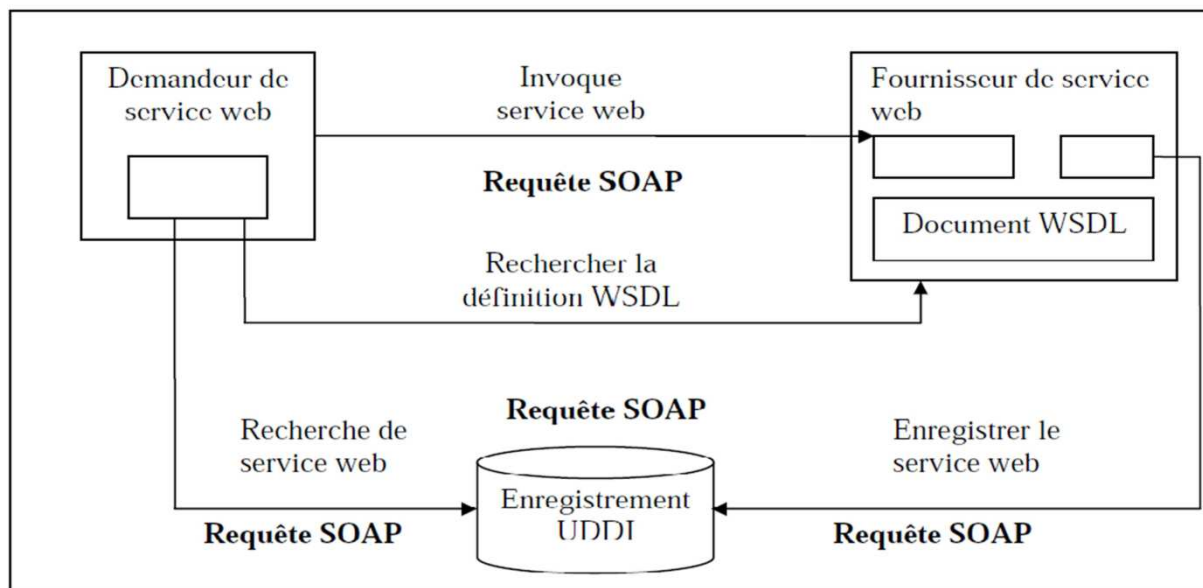
Les annuaires privés peuvent être hébergés par une société quelconque sur un réseau privé

ou sur internet.

**NB :** Les annuaires publics sont moins développés que les annuaires privés en terme de sécurité.

### ❖ Mécanismes d'accès aux services fournis par l'UDDI

La communication (requêtes / réponses) avec un annuaire UDDI repose sur le protocole de transport SOAP



**Figure 1.8- Mécanisme d'accès aux services web**

Un enregistrement UDDI, a deux types de clients :

Les fournisseurs de services.

Les utilisateurs de ces services.

UDDI comporte des pages qui fournissent des informations sur les fournisseurs (nom, coordonnées,...etc.) et des pages qui comportent la description, au format WSDL, des web services. Un fournisseur peut disposer de plusieurs entrées dans l'annuaire pour l'ensemble de services qu'il propose, et des pages qui disposent d'informations techniques détaillées sur les produits proposés.

## ❖ L'interface UDDI

L'interface UDDI est définie sous forme de documents UDDI et implémentée sous la forme de web service SOAP. Elle est composée des modules suivants :

**Interrogation** : Cette interface permet de rechercher des informations dans un répertoire UDDI et de lire les différents enregistrements enregistrés suivant le modèle de données UDDI.

**Publication** : Cette interface permet de publier des informations dans un répertoire UDDI conformément à son modèle de données.

**Sécurité** : Cette interface est utilisée pour obtenir et révoquer les jetons d'authentification nécessaires pour accéder aux enregistrements protégés dans un annuaire

UDDI.

**Contrôle d'accès et propriété** : Cette interface permet de transférer la propriété d'informations (qui est à l'origine attribuée à l'utilisateur ayant publié ces informations) et de gérer les droits d'accès associés.

**Abonnement** : Cette interface permet à un client de s'abonner à un ensemble d'informations et d'être avertis lors des modifications de ces informations. Tous les répertoires UDDI doivent gérer un avertissement par polling (le client interroge le serveur pour savoir si des modifications ont eu lieu sur les données auxquelles il est abonné). Une fonctionnalité optionnelle est également prévue permettant au client de communiquer au serveur la définition d'un web service sur lequel il souhaite être prévenu en cas de modification.

**Réplication interne (Nœuds d'un même annuaire)** : A côté des interfaces utilisateurs que nous venons de voir, UDDI définit également l'interface permettant de synchroniser les nœuds d'un même annuaire UDDI.

**Réplication externe (interrogation, publication, abonnement)** : La réplication externe par duplication d'informations entre différents annuaires UDDI n'a pas donné lieu à la définition d'une interface spécifique mais se fait en utilisant les interfaces d'interrogation (pour la lecture dans un annuaire), publication (pour la publication dans un autre annuaire) et éventuellement abonnement (pour pouvoir propager les modifications ultérieures).

## ❖ L'usage de l'annuaire UDDI

L'annuaire UDDI permet, la publication et la découverte des web services:

**La publication de services web:** Un web service peut être publié en publiant sa description, après sa production bien sur. Cette description peut être générée manuellement ou automatiquement à l'aide des outils qui peuvent générer des parties WSDL et de créer des entrées dans UDDI à partir de méta données par exemple. Un UDDI peut être de plusieurs types et leurs utilisations dépendent du domaine des services à publier, à savoir :

**Nœud UDDI pour une application interne :** les nœuds UDDI se trouvent derrière le firewall, avec ce type de nœud on a plus de contrôle sur l'enregistrement, l'accessibilité, la disponibilité et les spécifications de publication lors de la publication de service.

**Nœud portail UDDI :** il se trouve à l'extérieur du firewall du fournisseur de service ou entre les firewalls. Sur ce type de nœud on publie des web services pour les partenaires externes tout en implémentant des mécanismes d'accès sélectifs selon le profil des utilisateurs.

**Nœud catalogue du partenaire avec UDDI :** les web services peuvent être publiés sur un nœud du catalogue du partenaire avec UDDI (derrière le firewall). Le partenaire est choisi avec une autorisation d'accès spécifique.

**Nœud de place de marché UDDI :** il s'agit de relations interentreprises et de contrôle du partage de l'information entre systèmes d'information.

**La découverte de services web:** C'est la recherche et la localisation d'un web service particulier dans un annuaire de services décrivant le nom du fournisseur, l'objectif de chaque service. Donc il s'agit de l'acquisition des descriptions de web services et leur utilisation.

## ❖ Opérateurs d'annuaires UDDI actuels

Opérateurs	Adresse du site
HP	<a href="http://uddi.hp.com/uddi/index.jsp">http://uddi.hp.com/uddi/index.jsp</a>
IBM	<a href="http://www.3.ibm.com/services/uddi/v2beta/protect/registry.html">http://www.3.ibm.com/services/uddi/v2beta/protect/registry.html</a>
Microsoft	<a href="http://uddi.microsoft.com/register.aspx">http://uddi.microsoft.com/register.aspx</a>
NTT	<a href="http://www.ntt.com">www.ntt.com</a>
SAP	<a href="http://www.sap.com">www.sap.com</a>

## B. Services web REST :

### B . 1 Définition de REST :

On a plusieurs définitions sur l'architecture REST, on cite quelque unes :

- Selon Roy Thomas Fielding: «REST is a hybrid style derived from several of the network-based architectural styles and combined with additional constraints that define a uniform connector interface» [12].

Roy Thomas Fielding a créé cette architecture « REST » selon sa définition le REST est un modèle hybride dérivé de plusieurs modèles basés sur les concepts réseau.

- Selon Martin Kalin: « REST is a style of software architecture for distributed hypermedia systems; that is, systems in which text, graphics, audio, and other media are stored across a network and interconnected through hyperlinks. »[7]

On conclut du point de vue de “ Martin Kalin“ que REST est un style d'architecture pour les systèmes hypermédia distribués, c'est un système auquel le texte, les graphiques, l'audio et autre média sont stockés sur un réseau et reliés entre eux par des hyperliens.

### B.2 Présentation :

REST est l'acronyme de **R**épresentation **S**tate **T**ransfert défini par Roy Fielding en 2000 dans la thèse de doctorat à l'université de Californie ; REST n'est pas un protocole ou un format, contrairement a SOAP, HTTP ou RCP, mais un style d'architecture inspire de l'architecture du web base sur le protocole http, ne fait qu'utiliser les principes fondamentaux du web.

Utilisant le protocole HTTP, REST permet l'envoi de messages sans enveloppe SOAP et dans un encodage libre (XML, JSON, binaire, simple texte). Il est actuellement très utilise par sites communautaires(réseaux sociaux) leur permettant de proposer a leurs clients une API évitant a leurs clients de devoir passer par la case SOAP.

Les services web RESTful sont base sur un type d'architectures fondées sur les concepts de ressources. Le principe est qu'un composant logiciel puisse lire ou modifier une ressource en utilisant une représentation (XML ou JSON par exemple) de cette ressource.

Les services basés sur REST utilisent le protocole http, défini par la norme RFC2616 pour permettre au serveur et au client de communiquer ensemble. Ce protocole fournit les opérations nécessaires à la manipulation des ressources : GET, POST, PUT, DELETE.

On parle de RESTful pour tout système suivant l'architecture REST.

## **B.3 Architecture :**

**REST** est une architecture orientée ressource : une ressource est l'information de base dans une architecture REST.

Les architectures RESTful sont construites à partir de ressources uniquement identifiées : chaque ressource est identifiée individuellement et accessible par son URI unique pour procéder à diverses opérations GET, POST, PUT et DELETE, qui sont des opérations supportées nativement par http.

La manipulation des ressources se fait à travers des représentations : les ressources ont des représentations définies.

Les services REST sont sans états (stateless) chaque requête envoyée au serveur doit contenir toutes les informations relatives à son état et est traitée indépendamment de toutes autres requêtes.

## **B.4 Ressources :**

L'abstraction principale de l'information dans REST est la ressource. Toute information pouvant être nommée peut être une ressource : un document ou une image, un service temporel (par exemple 'le temps d'aujourd'hui à Marseille'), une collection d'autres ressources, un objet non virtuel (par exemple une personne), ainsi de suite. En d'autres termes, tout concept pouvant être la cible d'une référence hypertexte d'un auteur doit entrer dans la définition d'une ressource. C'est une correspondance conceptuelle à un ensemble d'entités et ce n'est pas l'entité correspondant à cette association à un moment particulier dans le temps. [thèse de Roy T. Fielding - traduction du chapitre 5 : REST]

Cette définition abstraite d'une ressource fournit les concepts clés de l'architecture du web.

Elle permet de généraliser en enveloppant de nombreuses sources d'information sans les distinguer artificiellement par leur type ou leur mise en œuvre. Ensuite, elle permet de faire

une liaison entre une référence et une représentation, permettant ainsi une négociation de contenu se basant sur les caractéristiques de la demande. Enfin, elle permet à un auteur de mettre en exergue un concept plutôt qu'une représentation change n'existe plus (en partant du principe que l'auteur a utilisé le bon identifiant) .[thèse de Roy T. Fielding - traduction du chapitre 5 : REST]

REST utilise un identifiant de ressource URI pour identifier la ressource particulière impliquée dans une interaction entre les composants (Client et Serveur).

Une ressource est un objet identifiable sur le système (cours, étudiant, note) par une URI unique qui a un rôle important et doit suffire pour identifier une ressource sur le site service web, exemple ([http://localhost:8080/univers\\_elearning/webapi/cours/22](http://localhost:8080/univers_elearning/webapi/cours/22)).

Une ressource peut subir quatre opérations de bases CRUD correspondant aux quatre principaux types de requête http (GET, PUT, POST, DELETE).

### **B.5 Représentation :**

une représentation désigne les données échangées entre le client et le serveur pour une ressource, ce format peut être sous différents types :JSON, XML, XHTML text/plain.

Les composants REST effectuent des actions sur une ressource en utilisant une représentation pour capturer l'état courant de cette ressource et en transférant cette représentation entre les composants. Une représentation est une séquence d'octets. Des noms comme document, fichier, entité de message http, instance ou variante sont utilisés pour désigner une représentation de façon générale.

La conception d'un type de media (XML,JSON,HTML) peut avoir un impact direct sur la perception de performance d'un système réparti hypermédia que peut avoir l'utilisateur. un format de données qui place l'informatique de plus en plus importante au tout début, a pour résultat une meilleure perception de performance par l'utilisateur par rapport à un autre système qui ne le fait pas.

#### **❖ JSON :**

JSON est l'acronyme de 'JavaScript Object Notation' qui est un format d'échange de données basé sur JavaScript, il est indépendant du langage de programmation.

Le site officiel présente (en juin 2013) des solutions d'intégrations de JSON pour 55 langages de programmation. Il sert à faire communiquer des applications dans un environnement hétérogène.

JSON est constitué de trois types de données :

**JSONObject** : commence par un '{' et se termine par '}' et compose d'une liste non ordonnée de paire clefs/valeurs. Une clef est suivie de ':' et les paires clef/valeur sont :

**JSONArray** : liste ordonnée d'objets commençant par '[' et se termine par ']', les objets sont séparés l'un de l'autre par ',', un JSONArray est une collection de JSONObject.

**Value** : un objet JSON peut être soit un string entre « "" » ou un nombre (entier, décimal) ou un booléen (true, false) ou un objet ou nul.

Ces types de données sont suffisamment génériques et abstraits pour pouvoir être représentés dans n'importe quel langage de programmation, aussi pour représenter n'importe quelle donnée concrète.

Le principal avantage de JSON est qu'il est simple à mettre en œuvre par un développeur.

Il est notamment utilisé comme langage de transport de données par AJAX et les services web : le type MIME *application/json* est utilisé pour transmettre par le protocole http.

### ❖ **Stateless** :

Toutes les interactions REST sont sans état. C'est-à-dire que chaque requête contient toutes les informations nécessaires pour qu'un connecteur (client et serveur) puisse comprendre la demande et ce indépendamment de toutes les requêtes qui ont pu l'avoir précédées.

Mais cette contrainte diminue la performance du réseau en répétant les données envoyées dans une série de paquets.

Cette contrainte accomplit quatre fonctions :

- Enlève tout besoin pour les connecteurs de maintenir l'état de l'application entre les requêtes, réduisant ainsi la consommation de ressources physiques ;
- Permet à des interactions d'être traitées en parallèle sans exiger de compréhension sémantique par le mécanisme de traitement ;

- Permet a un intermédiaire de regarder et de comprendre une requête de façon isolée, ce qui peut être nécessaire quand les services sont modifiés dynamiquement ; etc.
- Force toutes les informations qui pourraient être factorisées dans la réutilisation d'une réponse en cache à être présenté dans chaque requête.

Un autre type de connecteur, le connecteur de cache, peut être situé sur l'interface d'un connecteur client ou serveur afin de conserver des réponses, pouvant être mises en cache, relatives aux interactions actuelles, de sorte qu'elles puissent être réutilisées pour des requêtes ultérieures. Un cache peut être employé par un client pour éviter la répétition de communication réseau, ou par un serveur pour éviter de répéter le processus de production d'une réponse. Dans les deux cas, il sert à réduire la latence des interactions.

#### ❖ Méthodes HTTP :

Le protocole http décrit les instructions détaillées de la façon dont un navigateur doit envoyer ses requêtes au serveur.

HTTP permet au navigateur de récupérer divers types d'informations a partir du serveur, la récupération d'une page HTML, ou la soumission d'un formulaire. En fait, http permet d'effectuer huit type de requêtes différentes sur le serveur.

REST s'appuie sur le protocole http pour réaliser les opérations CRUD sur les ressources :

- CREATE ➡ POST
- READ ➡ GET
- UPDATE ➡ PUT
- DELETE ➡ DELETE

#### ❖ JAX\_RS :

Acronyme de Java API pour RestfullWeb Services qui est une spécification qui décrit la mise en œuvre des services REST web cote serveur. Son architecture se repose sur l'utilisation des classes et des annotations pour développer les services web.

Plusieurs implémentations sont développées autour de cette spécification :

JERSEY : implémentation de référence fournie par Oracle(<http://jersey.java.net>)

CXF : fournie par Apache(<http://cfx.apache.org>)

RESTEasy : fournie par JBOSS

RESTLET : L'un des premiers Framework implémentant REST pour Java.

## 1.5 La sécurité des web services

Vu que le domaine d'application majeur des web services est les échanges interentreprises (*Business to Business* ou *B2B*). La gestion de la sécurité est un élément fondamental de la technologie des web services.

Notre but ici ne consiste pas à traiter en détail des méthodes utilisées dans le domaine de la sécurité informatique. Par contre, il est nécessaire de rappeler les concepts rencontrés :

- Confidentialité et intégrité garantissent qu'un échange est confidentiel et intact.
- L'authentification consiste à déterminer l'identité, au sens large (utilisateur, machine, application).
- La non-répudiation qui garanti qu'un acteur ne puisse renier la dernière opération qu'il a effectué.

En règle générale, ces fonctionnalités s'appuient sur des processus de cryptage/décryptage grâce à des clés (publiques et privées).

Au sein d'Internet, la mise en œuvre de ces concepts s'appui, au niveau du transport, sur le protocole *Secure Socket Layer* (SSL<sup>1</sup>). C'est ainsi que les web services sécurisés utilisent non pas SOAP/HTTP, mais SOAP/HTTPS. Il s'agit d'une utilisation de HTTP en combinaison avec le protocole SSL.

Une alternative prometteuse au chiffrement par SSL, est le standard proposé par W3C : *XML Encryption Standard*. Ce dernier défini les spécifications de cryptage et de décryptage d'un document XML, ou de seulement une partie.

Bien que les standards au formalisme XML soient en perpétuelle évolution, un consensus semble se faire autour de propositions suivantes : SAML (*Security Assertions Markup Language*), XKMS (*Public Key management*) et WS-Security. De plus, il ne faut pas oublier l'importance des "pare feu" (*firewall*). Éléments techniques indispensables à la prévention des intrusions dans les réseaux informatiques.

- **WS-Security**

La spécification WS-Security est supportée depuis 2002 par de grands acteurs du domaine de la sécurité informatique, notamment, Microsoft, IBM et VeriSign.

Cette proposition est la plus sérieuse car elle définit une infrastructure complète de la gestion de la sécurité, adaptée aux web services. En effet, WS-Security prévoit une extension du protocole SOAP afin de supporter tous les mécanismes de sécurité : authentification, confidentialité et intégrité. Mais surtout, cette spécification offre la possibilité de véhiculer des jetons de sécurité. En fait, le message SOAP d'un service Web serait crypté en utilisant la spécification XML-Encryption et intégrerait le jeton, vraisemblablement au format X509 ou Kerberos. Toutes ces informations prendraient place dans l'entête du message SOAP.

En résumé, SAML est un langage qui couvre les problématiques d'authentification et d'habilitations des transactions. XKMS vise à standardiser les opérations relatives au traitement des clés publiques à des fins de confidentialité, d'authentification et de répudiation. En fin, WS-Security est un langage proposé pour sécuriser les web services, il s'agit d'une extension de SOAP incluant cryptage, signature, etc.

## **1.6 Interopérabilité :**

### **1.6.1 Définition :**

Interopérabilité se réfère principalement à la libre circulation des informations et des données à travers de multiples services Web hébergés sur la plate-forme unique ou multiple dans un des environnements hétérogènes. Une des questions les plus pertinentes qui est souvent posée est la façon de définir un «Service Web». Il peut être défini en termes plus simples comme "service qui répond à d'autres services" qui héberge des applications multiples comprenant des données variées et des infos sur une plate-forme unique sur le Web en utilisant des standards ouverts comme XML (Extensible Markup Language), SOAP (Simple Object Access Protocol) et WSDL (Web Services Description Language) et UDDI (Universal Description Discovery and Integration). Comme par W3C [World Wide Web Consortium] le service Web peut être défini comme "Un service Web est un système logiciel conçu pour soutenir l'interaction interopérable machine-to-machine sur un réseau. Il dispose d'une interface décrite dans un format capable de la machine-processus (spécifiquement WSDL).

D'autres systèmes d'interagir avec le service Web dans une manière prescrite par sa description en utilisant des messages SOAP, généralement transmis via le protocole HTTP avec une sérialisation XML en conjonction avec d'autres normes Web- liés ". Une des choses les plus importantes à garder à l'esprit tout en comprenant le concept de services Web est que le service Web ne va jamais être utilisé directement par l'être humain (s) .En fait, toutes les informations disponibles à travers le service Web est principalement destiné pour le logiciel comme cela est le logiciel qui communique directement avec le service web et pas les êtres humains. À notre connaissance et la compréhension il est d'abord un tel papier qui non seulement aborde la question de l'interopérabilité, mais aussi le design de service Web dans ASP.NET qui répond à la requête reçue du client Java.

### **1.6.2 L'interopérabilité dans les technologies existantes :**

Les services Web devraient être interopérables dans une certaine mesure que les classes, les tableaux et structures conçus et développés en utilisant J2SE (Java 2 Standard Edition), ou J2ME (Java 2 Mobile Edition), ou J2EE (Java 2 Enterprise Edition), l'architecture doit être capable de communiquer avec le tableau, classes et structures conçues dans le C # (C sharp) d'application .NET. De la même manière leur faudrait flux continu de données entre les applets conçues dans J2SE à Midlets conçus J2ME aux servlets conçu J2ME. Les mêmes données ou des renseignements si nécessaire doivent être mis à la disposition de l'application conçue en VB.NET ou ASP.NET. J2EE offre deux l'architecture est pour les services Web conçus pour exploiter les concepts de l'informatique distribuée à savoir de. RMI (Remote Method Invocation) et CORBA (Common Object Request Broker Architecture) .RMI est utiliser principalement pour la réalisation du JAVA - la communication de JAVA qui est aux deux extrémités de l'application, les programmes sont écrits en Java, donc pas de problèmes de l'interopérabilité de surgir entre les applications.RMI utilise le concept de distance, Interfaces d'objets distants, Stubs et squelettes pour mener à bien la communication entre les programmes client et serveur écrit en Java. Cependant l'architecture CORBA qui est un standard défini par l'Object Management Group est principalement utilisé pour la réalisation de la communication entre l'application Java et de la non-application Java tels que ceux application conçue dans le COBOL, Pascal, FORTRAN, ALGOL, C ++ et C # .NET, etc. CORBA utilise des fichiers spéciaux appelé comme langage de définition d'interface (IDL) pour spécifier les interfaces objets présents aux applications de monde extérieur. CORBA précise ensuite une application de IDL pour un langage de mise en œuvre spécifique, comme C ++ ou Java. Correspondances standard existent pour C, C ++, Ada, Simula, Smalltalk,

Python, Ruby on Rails. Donc, l'effort est toujours d'aller dans la connectivité sans faille entre Java et l'application C# .NET.

### **1.6.3 Le respect des normes :**

L'effort doit toujours être mis sur la conception d'un service Web qui se conforme strictement aux normes et standards de WS-I (Web Services Organisation normes d'interopérabilité) et W3C (World Wide Web) Consortium.

Si nous voulons mesurer comment interopérable un service Web est qu'il doit être mesurée en termes de celui de l'adhésion ou de la déviation de l'ensemble existant de normes et standards. Si un service Web suit de près ces ensemble standard de lignes directrices et de normes, alors il en est une qui est hautement interopérable et si ne fonctionne pas, alors il est considéré comme faiblement interopérable.

### **1.6.4 Initiatives WS-I :**

Aujourd'hui, les services Web sont emballés dans la FRM (Web Services Framework de ressources) [document Microsoft] au comprend le XML, SOAP et WSDL et UDDI (Universal Description Discovery and Integration) et al. Afin d'atteindre un niveau élevé de l'interopérabilité entre les différentes plates-formes et des technologies une organisation sous le nom de Web Service Interopérabilité [WS-I] [14] a été fondée .WS-I est une organisation industrielle ouverte affrété pour établir les meilleures pratiques pour le Web l'interopérabilité des services, pour certains groupes de normes de services Web, à travers de multiples plates-formes, systèmes d'exploitation et langages de programmation. Le profil WS-I Basic établit services Web de base des spécifications (SOAP, WSDL, UDDI, XML Schéma, HTTPS) qui doit être utilisée de concert pour développer des services Web interopérables. À ce jour, WS-I a produit Basic Profile 1.0 et 1.1.

### **Conclusion :**

Dans ce chapitre nous nous sommes intéressés aux technologies sur lesquelles sont basés les web services. Nous avons analysé le moyen de transmission et de transaction qu'est SOAP. Ce dernier permet l'échange d'informations structurées grâce au format XML. D'autre part, nous avons vu un langage de description de services, le WSDL basé aussi sur XML, qui permet de procurer aux applications clientes de nombreux renseignements sur le contenu d'un service et sur les façons de l'invoquer. Puis, nous avons abordé le registre annuaire pour les web services, nommé UDDI et considéré comme un standard. Enfin, nous avons illustré quelques moyens fondamentaux qui participent à la sécurisation des web services. Dans le prochain chapitre nous allons parler du E-Learning.

## 2. Introduction

Le nombre de plateformes E-Learning, qui sont basées sur les services web, est de plus en plus croissant. Un web service peut être défini comme un programme autonome qui s'exécute sur le web. Un web service est décrit par une description WSDL (Web Services Description Language), qui est enregistrée dans des registres UDDI (Universal Description, Discovery and Intégration) afin de faciliter sa recherche (découvre Une plateforme E-Learning peut être vu comme un ensemble de services web qui coopèrent entre eux pour fournir certaines fonctionnalités aux acteurs de la plateforme. Ainsi, il est possible d'utiliser / réutiliser des services externes qui appartiennent à d'autres plateformes E-Learning. Mais avant d'utiliser / réutiliser ces services, il est nécessaire de les localiser. Cette localisation (découverte) est une opération importante qui doit être automatique et efficace.

### 2.1 E-Learning:

L'E-Learning est un processus d'apprentissage à distance, qui repose sur la mise en place d'un dispositif de contenus pédagogiques via un réseau de type Internet ou Intranet et permet ainsi à une ou plusieurs personnes de se former à partir d'un ordinateur.

Les supports multimédias utilisés peuvent combiner du texte, des graphismes, du son, de l'image, de synthèse, de l'animation et même de la vidéo. Ces supports permettent une nouvelle approche pédagogique, avec l'emploi de méthodes plus attrayantes où l'interactivité joue un grand rôle, et avec la possibilité de s'adapter davantage au processus d'apprentissage de l'apprenant.

L'utilisateur peut se former à son rythme, en fonction de ses besoins et de ses disponibilités, ce qui est particulièrement important à une époque où la formation se décline tout au long d'une vie.

## 2.2 Apport des nouvelles technologies à l'enseignement :

Les progrès de l'informatique et des réseaux tel que Internet, ou plus largement des nouvelles technologies de l'information et de la communication (NTIC) a permis de créer des systèmes d'enseignement ou le maître mot est « multimédia ».

N'importe qui peut désormais mettre un cours en ligne, mais construire une ressource pédagogique de qualité avec ces nouvelles technologies ne veut pas dire qu'il suffit de mettre en ligne un polycopié en y ajoutant quelques animations. Les Technologies de l'Information et de la Communication Educatif (TICE) doivent apporter une réelle plus value par rapport aux ressources informatiques déjà existantes.

## 2.3. Les formes d'E –Learning :

Nous commencerons par différencier les formes de communication dans l' E-Learning puis les types d'apprentissage sous-jacents.

### 2.3.1 Les formes de communication :

Il existe deux formes de communication, la forme synchrone et la forme asynchrone :

#### ➤ La forme synchrone :

Correspond à une situation dans laquelle les apprenants peuvent se communiquer directement via des chats, des visioconférences ou audioconférence. L'interactivité est aussi immédiate qu'en présent elle (un ou plusieurs formateurs et tous les apprenants sont regroupés au même instant pour la réalisation de la formation).

- **Les principaux avantages de la formation synchrone**

- ❖ La dynamique de groupe s'installe plus rapidement et les apprenants s'interagissent intensivement à l'écran avec les formateurs et entre eux ;

- ❖ On peut rapidement créer du contenu prêt à diffuser ;

- ❖ Utilisation de langage oral et visuel.

- **Les principaux inconvénients de la formation synchrone**

- ❖ Il faut posséder le matériel technique nécessaire (Webcam, casque, carte vidéo) et une bande passante suffisamment large ;

- ❖ Elle ne permet pas à chaque apprenant d'évoluer à son propre rythme et de choisir l'ordre des éléments du cours à sa guise ;

- ❖ Réduit le taux de contributions des acteurs faute de temps limité des séances d'enseignement ce qui ne permet pas aux apprenants d'approfondir certains aspects du cours ;

- ❖ Volatilité des sessions synchrones.

➤ **La forme asynchrone :**

Décrit une situation où il n'y a pas de contact simultané (temps réel) entre l'apprenant et le formateur elle correspond à un mode de communication alterné. La communication s'opère essentiellement par courrier électronique (e-mail) et forum de discussion.

Le problème principal de cette forme de communication est le temps qui peut s'écouler entre une question d'un apprenant et la réponse du formateur, temps durant lequel l'apprenant évolue dans son apprentissage, si bien que la réponse reçue ne sera peut être plus en rapport avec sa nouvelle situation d'apprentissage.

- **Les principaux avantages de la formation asynchrone :**

- ❖ Les apprenants évoluent à leur propre rythme ;

- ❖ Ils peuvent adapter l'ordre dans lequel ils appréhendent les éléments du cours ;

- ❖ Ils peuvent revoir et approfondir certains aspects du cours à leur guise ;

❖ Un temps de réflexion est donné à chaque acteur (apprenant, formateur) pour donner leurs contributions.

- **Les principaux inconvénients de la formation asynchrone :**

❖ L'apprenant est seul devant son écran. En cas de difficulté, il ne peut s'appuyer sur aucune aide extérieure (qui renforce le sentiment d'isolement) ;

❖ Le temps qui peut s'écouler entre une question d'un apprenant et la réponse du formateur peu être long.

### **2.3.2 Les formes d'enseignement :**

L'e-Learning peut être classé en cinq méthodes d'enseignement principales :

- L'auto-formation en solitaire,
- L'auto-formation avec tutorat,
- L'auto- formation avec tutorat asynchrone,
- La classe virtuelle en temps réel,
- La visio formation personnalisée.

## **2.4 Les avantages et les faiblesses du e-Learning :**

### **2.4.1 Les avantage du e-Learning :**

- ❖ **Pour les apprenants**

L'e- Learning offre de nombreux avantages pour les apprenants Il permet notamment :

- D'acquérir un diplôme ou une qualification dans un domaine donné,

- De compléter un enseignement ou une formation (perfectionnement),
- De se doter de compétences nouvelles a des fins d'enseignement ou professionnelles (actualisation de connaissances).
- D'étudier dans des conditions particulières mieux adaptées,
- À chacun d'être plus responsable de son apprentissage ou de sa formation,
- Un gain de temps considérable,
- Une facilité d'accès (dématérialisation du lieu),
- Un suivi personnalisé (tuteur en ligne),
- D'aller plus loin par le biais de tuteurs adaptés à chaque classe d'apprenants ou à l'aide de liens Internet vers d'autres sites traitant du même sujet.
- La possibilité d'échange avec les autres apprenants à une grande échelle à l'aide de forum ou de chat.

#### ❖ Pour les formateurs

Le e-Learning permet pour le formateur:

- Un gain de temps, pas de déplacement,
- Un enrichissement dynamique du contenu,
- Une évaluation des pré-requis plus facile grâce à l'utilisation de jeux ou de QCM interactifs,
- Une évaluation plus précise grâce à des tests en continu tout le long de l'apprentissage.

#### ❖ Pour les entreprises

Dans le cadre de la formation à distance le e-Learning permet :

L'économie des coûts liés au déplacement (transport, hébergement, coût de l'indisponibilité),

- La possibilité d'avoir un nombre très important d'apprenants,
- Plus de flexibilité vis à vis des formations,
- La réduction des coûts globaux.

#### **2.4.2 Les faiblesses d'e-Learning :**

Les faiblesses d'e- Learning sont nombreuses. Elles sont parfois liées à la manière dont les dispositifs sont conçus et parfois aux acteurs eux- mêmes:

- La formation et la motivation des tuteurs.
- Les apprenants ne disposent pas des compétences techniques minimales.
- Les forts préjugés des novices.
- La nécessité d'un équipement coûteux.
- Les activités prennent plus de temps que prévu.
- Les effectifs se concentrent très souvent autour de compétences qualifiées.
- Les formations ne sont pas accessibles à tous.
- Le challenge des contenus pédagogiques à améliorer n'est pas encore gagné.
- L'effort d'apprentissage par l'e-Learning est plus important que par la formation classique.

En effet l'apprenant ne peut être passif : il est l'acteur de sa formation.

- Les contraintes techniques de la plate-forme utilisée, chaque connexion peut être un problème potentiel, et le choix d'une plate-forme adaptée à tout le monde n'est pas évident.

### **2.5 Plates -formes d'enseignement à distance:**

Une plate-forme d'enseignement à distance (EAD), appelée parfois LMS (Learning Management System), est un système informatique de gestion de formation et de gestion de contenu de formation. Les dernières générations intègrent ou s'interfacent avec des outils de communication tels que : chat, messagerie instantanée, visioconférence, etc.

Au sein de projets concrets de formation, elles favorisent l'apprentissage en ligne et le travail collaboratif.

Il s'agit d'une composante d'un dispositif e-Learning mais ce n'est pas la seule. Il est entre autre un des outils des Technologies de l'information et de la communication (TIC) et des Technologies de l'information et de la communication pour l'éducation (TICE).

Une plate-forme pédagogique est un logiciel qui assiste la conduite des formations présent elles et à distance. Elle est basée sur des techniques de travail collaboratif et regroupe les outils nécessaires aux trois principaux acteurs de la formation : apprenant, tuteur, administrateur.

Elle fournit à chaque acteur un dispositif qui a pour première finalité l'accès à distance au contenu pédagogique, l'auto apprentissage, l'autoévaluation et la télé tutorat via l'utilisation des moyens de travail et de communication à plusieurs : visioconférence, e-mail, forums, chats, etc.

Le but est donc de combler la perte de cohésion et de stimulation de la salle que peut sentir l'apprenant devant sa machine.

## 2.6 Les principaux critères que doivent vérifier les plates-formes :

1. Le coût total de la mise en place de la plate-forme (gratuite ou non), dans le cas d'une plate-forme payante, vérifier les prestations fournies.

2. Les besoins de maintenance de la plate-forme d'un point de vue strictement informatique, tout ce qui devra être géré par les administrateurs (voir également le coût en personnel de maintenance).

3. La sécurité de la plate-forme d'un point de vue purement informatique (filtrer les IPs qui auront accès aux cours) et au niveau des cours (login, mot de passe).

4. Le nombre d'utilisateurs maximum que pourra supporter la plate-forme.

5. La possibilité de mettre en place des QCM (Questions à Choix Multiples) sur la plateforme, s'il est possible d'insérer une image, si le système de questions est interactif (avec animations), s'il y a possibilité de faire un tirage aléatoire des questions.

6. Un forum de discussion dont l'accès sera étroitement lié à la sécurité interne de la plateforme.

7. La technologie utilisée, du point de vue du langage pour pouvoir éventuellement faire évoluer le système et sa capacité d'accueil (XML, MySQL, PHP...).

8. Un système de gestion de mails, pour qu'un formateur puisse éventuellement envoyer des mails à l'une de ses classes, également lui permettre de gérer des groupes.

9. La façon de poster un cours.

10. Les moyens requis pour consulter la plate-forme (browser, système d'exploitation...).

11. La possibilité de diffuser les cours par vidéo soit en temps réel, soit en différé.

## 2.7 Outils de communication :

Les plates-formes intègrent un certain nombre d'outils de communication, la liste des outils disponibles dépend de la richesse de la plate forme.

Ces outils sont souvent très usitées sur le Web et ne sont en rien spécifique au e-Learning, ils font simplement partie intégrante de la plate forme et ne nécessitent pas de configuration spécifique sur les postes des apprenants (par exemple, il n'est pas nécessaire de configurer un client messagerie pour profiter de ce service).

On peut citer :

**La messagerie** : C'est un espace de communication entre les membres du groupe apprenants et formateurs. Ce système permet l'envoi et la réception de mails avec ou sans fichier attaché.

Il peut être interne à la plateforme et ne nécessite pas d'avoir un e-mail personnel.

**Le forum** : pouvant être public ou réservé à un groupe, permet aux apprenants et aux formateurs, de poster des messages qui seront accessibles à l'ensemble des membres du groupe de formation, de répondre aux messages déjà postés et ainsi engager une discussion sur un sujet donné).

**Le Chat** : permet à l'ensemble des membres du groupe de discuter en temps réel.

**Les documents pédagogiques partagés** : Une zone commune peut être utilisée pour mettre à disposition du groupe différents documents sous forme numérique (Word, Excel, PDF, etc.)

**Les news** : La possibilité de créer des news permettra, par exemple, d'informer ses apprenants de nouveaux rendez-vous, ou de leur communiquer diverses informations.

**Les sondages** : permettra de poser des questions ouvertes ou de proposer des choix de réponses. Ultérieurement, les réponses obtenues sur un cours en particulier pourront être analysées.

**Le Bloc-notes** : C'est un espace privé à chaque utilisateur. Il peut y noter toutes les informations qu'il souhaite ce qu'il a retenu de sa formation, des actions à mener...

**Les Glossaires** : Des glossaires peuvent être créés, destinés à l'ensemble des utilisateurs ou un groupe en particulier. Les différents termes avec leur signification sont enregistrés, un moteur de recherche permet aux utilisateurs de faciliter la recherche d'un terme dans les glossaires.

## **2.8 Acteurs de la plate-forme LMS :**

Il y a trois principaux types d'utilisateurs accédant à la plate-forme : *l'administrateur, les apprenants et les enseignants (formateurs).*

### **1. Enseignant :**

Chaque enseignant est responsable d'un ou plusieurs modules d'enseignement. La plate-forme permet à l'enseignant de structurer les modules en différentes briques de contenus de cours. Il peut ajouter un contenu de cours en téléchargeant sur le serveur de la plate-forme n'importe quel type de fichier. Il peut mettre à jour et supprimer chaque contenu.

### **2. Apprenant :**

Un apprenant peut consulter ces différentes briques de cours et éventuellement les télécharger pour une lecture ultérieure hors connexion.

### **3. Administrateur :**

L'administrateur, de son côté, assure l'installation et la maintenance du système, gère les droits d'accès, crée des utilisateurs, les intitulés de module etc. Il y a quatre types de contenus de cours : exercice, ressource, test, concept.

## 2.9 Quelques fonctionnalités de la plate-forme LMS :

### A. Modules d'édition de contenus de cours:

Les modules sont divisés en topic (semaines de travail) différents selon les enseignants concernés dans chaque topic, il y a une liste de travaux à faire de 4 types (exercices, ressources à lire, concepts, tests).

Chaque module doit avoir un code l'identifiant. Par exemple, pour système d'exploitation, le code serait SE (il faut savoir comment déterminer automatiquement ce code de manière à ce qu'il soit significatif). Il est important de mettre ce code à côté du nom de module et des contenus de cours proposés. Ainsi, l'étudiant pourra facilement se repérer.

Par ailleurs, le forum est associé au module et un forum général où les étudiants peuvent communiquer librement sur tout un tas de sujets indépendants de leur formation.

### B. Gestion des inscriptions :

La plate-forme dispose d'un système de gestion des inscriptions. Ce système permet l'inscription aux formations utilisant la plate-forme. Il s'agit d'un formulaire à remplir en ligne, qui permet différents types d'inscriptions :

- Inscription en ligne automatique avec validation d'administrateur de type « staff »
- Inscription manuelle par un administrateur en rentrant directement un utilisateur.
- Inscription manuelle par un administrateur en rentrant un fichier contenant une liste d'utilisateurs (*en format Excel*).

### C. Dépôts de fichiers :

Le dépôt de fichiers sert aux étudiants pour rendre les tests et aux professeurs pour récupérer ces fichiers « rendus de test » et mettre à disposition d'éventuelles corrections.

Il y a un dépôt commun par module, les étudiants disposent d'un certain temps pour mettre les « rendus de test » dans le dépôt.

Chaque élément du dépôt contient les informations suivantes :

- Auteurs (nom, prénom) ;
- Type (rendu de test ou rendu d'exercices ou correction) ;
- Nom physique du fichier ;
- Module associé ;
- Rendu avant la date limite (oui/non).

Les professeurs du module associé au dépôt peuvent lire tous les fichiers déposés et ils peuvent déposer les fichiers de type « correction » dans le dépôt.

Les étudiants peuvent lire les fichiers qu'ils ont déposés et les fichiers de type correction.

Les administrateurs peuvent avoir accès aux dépôts des modules et supprimer ou modifier ces dépôts. La modification des dépôts consiste à supprimer ou ajouter des fichiers dans le dépôt.

### **C. Gestion des notes :**

La plate-forme dispose d'un système de gestion des notes de chaque apprenant. Il y a une partie de cette gestion des notes qui sera gérée par les enseignants eux-mêmes et une autre partie qui sera gérée par l'administrateur.

La gestion des notes comprend différentes fonctionnalités :

- Ajout des notes manuellement ;
- Ajout des notes par import de fichiers ;
- Consultation des notes par les étudiants ;
- Création automatique de bulletins de notes ;
- Création de tableaux récapitulatifs de notes.

## 2.10 E-Learning en Algérie :

### A- Facteurs et indicateurs technologiques :

Par rapport aux années précédentes, l'Algérie a fait un grand saut technologique en matière de développement et de structures dans les TIC . Nous pouvons lister plusieurs critères ayant contribué à ce progrès :

Alors qu'elle était carrément inexistante en l'an 2000, l'ADSL a dépassé les 700.000

Km en 2006 le réseau de fibre optique passe à 26.000 km contre 7.000 km seulement en 2000 (projet transsaharien Annaba-Lagos) les nombreux projets de satellites : Alsat1, Alsat2 etc. Le marché multi opérateurs en progression la densité (fixe et mobile) est passé de 5,28%

en 2000 à près de 65% à fin juillet 2006, Abonnés aux réseaux mobile et fixe a atteint, 8600 000 en 2006 contre 54 000 en 2000 Lignes spécialisées 34 MB, Autres ressources ADSL ...

Opération Ousratic IPC/Famille, avec la formule de microcrédit initié par le ministère de la Poste et des Technologies de l'information et de la communication plus de 16 milliards de DA ont été consacrés par le gouvernement au développement et l'intégration des technologies de l'information et de la communication dans les collèges Projet du Ministère de l'éducation nationale, fin 2009, de doter toutes les écoles de laboratoires qui disposeront tous d'un équipement informatique complet ce qui a contribué à l'augmentation du nombre d'internautes à plus de 3.000.000 d'internautes en 2006.

### B- Expérimentations professionnelles :

#### 1- Djaweb « <http://elearn.djaweb.dz> »:

Djaweb, filiale d'Algérie Télécom, a lancé le service « E-Learning » par carte prépayée, réalisé en partenariat avec Thomson et Microsoft. Premier du genre en Afrique, ce service propose, via Internet, un contenu de 4.000 cursus de formation dans le domaine des Technologies de l'information et de la communication et du développement des compétences professionnelles. Il s'agit, entre autres, de l'initiation à l'informatique aux certifications les plus connues des grands éditeurs informatiques (Microsoft, Oracle, Cisco, IBM, Novell Comptia, SAP...). La stratégie d'entreprise, la stratégie marketing, les finances dans

l'entreprise, le leadership, la gestion des performances et bien d'autres cursus conçus par les plus grandes.

universités anglo-saxonnes. Un programme de formation est proposé par Djaweb, aux clients et ce, pour l'obtention de la certification Microsoft. Le programme comprend 16 modules, pour une durée d'une année avec un volume horaire de 200 heures.

## **2- DZCampus.com, la première plate-forme E-Learning en Algérie :**

Actions technologiques " Actech ", spécialisée dans la communication multimédia, notamment dans la conception, la réalisation et l'animation de sites Internet et le développement d'applications web interactives, et " Comform Communications ", une entreprise algérienne spécialisée dans les études, la recherche, la formation, l'expertise et le conseil en communication tous domaines confondus, en partenariat avec la Bibliothèque Nationale ont lancés la première plate-forme e-learning ouverte en Algérie.

Elle s'adresse aux entreprises, institutions et organismes de formation d'Algérie. DZCampus.com propose plus de quarante modules de formation en ligne avec tutorat et quiz d'évaluation dans les domaines suivants: management, gestion de projet, langues, bureautique- informatique, graphisme, gestion-secrétariat, enseignement général.

Elle offre différents espaces dédiés à l'entreprise, au formateur, au stagiaire ainsi qu'un espace commun et un "Agora"; elle permet de capitaliser le savoir et le savoir-faire des utilisateurs et contribue ainsi à développer les compétences individuelles et à améliorer la performance de l'entreprise. DZCampus.com est dotée d'un outil de visioconférence et utilise l'outil plateforme Dokeos1.6.5

**Conclusion :**

Il existe sur le web un volume considérable de ressources pédagogiques, on a présenté quelques plates formes actuelles qui offrent une certaine ouverture sur le web, d'où la nécessité de développement des spécifications descriptives de ces ressources pour faciliter leurs localisation et réutilisation et assurer leurs interopérabilité. Dans le chapitre suivant nous allons présenter la conception de notre application E-learning.

### 3. Introduction :

Pour développer des applications informatiques, il est nécessaire d'utiliser une démarche méthodologique et rigoureuse. Alors le choix d'une méthode est très important.

Dans le cadre de notre projet nous avons opté pour le langage de modélisation par objet à savoir UML. Plusieurs raisons nous ont conduits à adopter UML dans notre modélisation. En effet : Il a été normalisé par l'OMG, qui est une organisation internationale se chargeant de la standardisation des technologies objets. C'est donc un langage standard compréhensible par tout le monde. Malgré son évolutivité, il est stable et facilite la compréhension du système grâce à ses représentations simples. Il permet d'utiliser les principes et les concepts objet pour enrichir la phase de la conception des systèmes.

#### 3.1 Présentation d'UML :

##### 3.1.1. Définition de l'UML :

UML (*Unified Modeling Language*, que l'on peut traduire par "*langage de modélisation unifié*") est un langage graphique de modélisation des données et des traitements.

De plus, il est une notation permettant de modéliser un problème de façon standard. Il est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE, principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson.

UML est à présent un standard défini par l'Object Management Group (OMG), et est devenu désormais la référence en termes de modélisation objet.

Pour plus de détails, se référer à l'annexe.

#### 3.2. Objectif du projet :

Notre travail consiste à réaliser une application E-Learning interopérable rendant ainsi l'accès ouvert à un grand nombre d'utilisateurs. Pour cela, nous devons trouver les fonctionnalités qui doivent être interopérables et qui seront ensuite réalisées à l'aide des services web.

### 3.3. Identification des acteurs et leurs besoins fonctionnels :

Dans notre plate forme on distingue quatre principaux acteurs, pour chaque acteur on a identifié les fonctionnalités lui permettant d'accomplir son rôle dans le système :

➤ Administrateur :

L'administrateur représente l'acteur chargé de réguler l'activité sur le système, il assure un ensemble d'actions tels que :

- Accès direct à la base de données de la plate-forme ;
- Il s'occupe entre autre de l'inscription finale des acteurs dans la plate-forme (validation des prés inscriptions) ;
- Gestion des comptes utilisateurs (apprenants et formateurs) ;
- Gestion des sessions de prés inscription des visiteurs ;

➤ Formateur :

Le formateur est la personne chargée de suivre les apprenants durant le processus d'apprentissage et cela en :

- Assurant le suivi des cours ;
- Rechercher et faire un cours ;
- Assurant le suivi des exercices ;

➤ Apprenants :

L'apprenant est toute personne qui suit un cours dans son propre espace réservé, seul ou au sein d'un groupe, avec ou sans tuteur dans le but d'atteindre les objectifs pédagogiques .Consulter et télécharger des cours, des exercices ;

- Télécharger un cours ;
- Recherché un cours ;
- Rechercher un exercice ;
- Télécharger un exercice ;

➤ Visiteur :

Le visiteur est toute personne qui visite le site sans être ni un apprenant ni un formateur.

L'ensemble des actions qu'un visiteur peut effectuer sont :

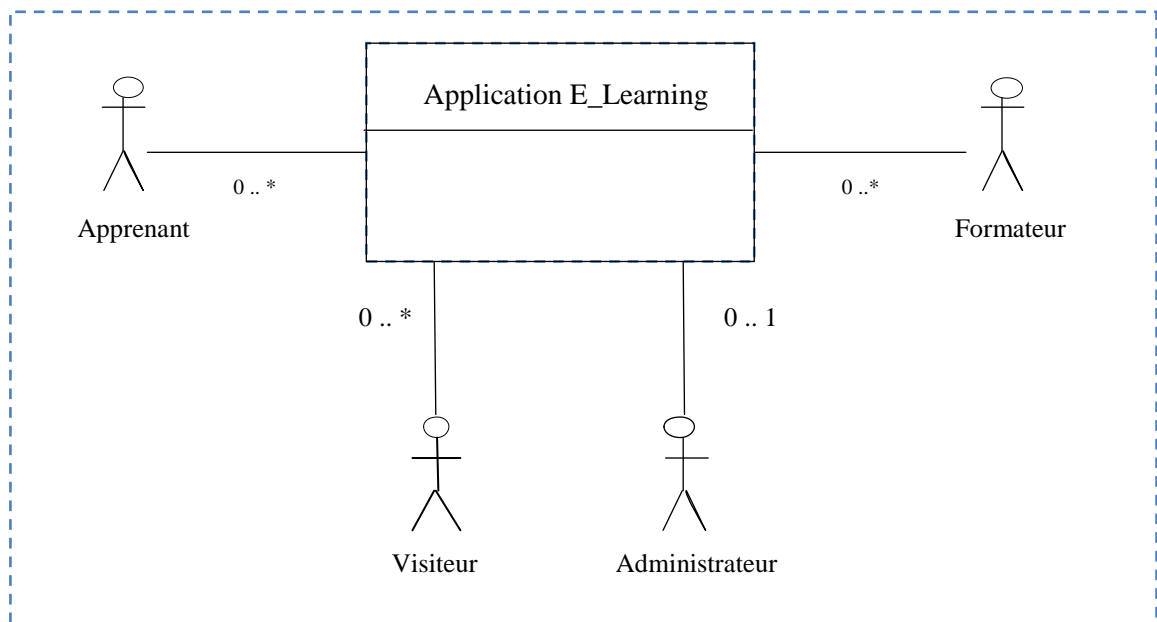
- Se préinscrire en tant qu'apprenant ;
- Se préinscrire en tant que formateur ;
- Contacter l'administrateur pour d'éventuelles informations ;

### 3.4. Diagrammes représentatifs :

#### 3.4.1. Le diagramme de contexte de l'application :

Le diagramme de contexte permet de spécifier le nombre d'instances d'acteurs connectés à ce système à un moment donné.

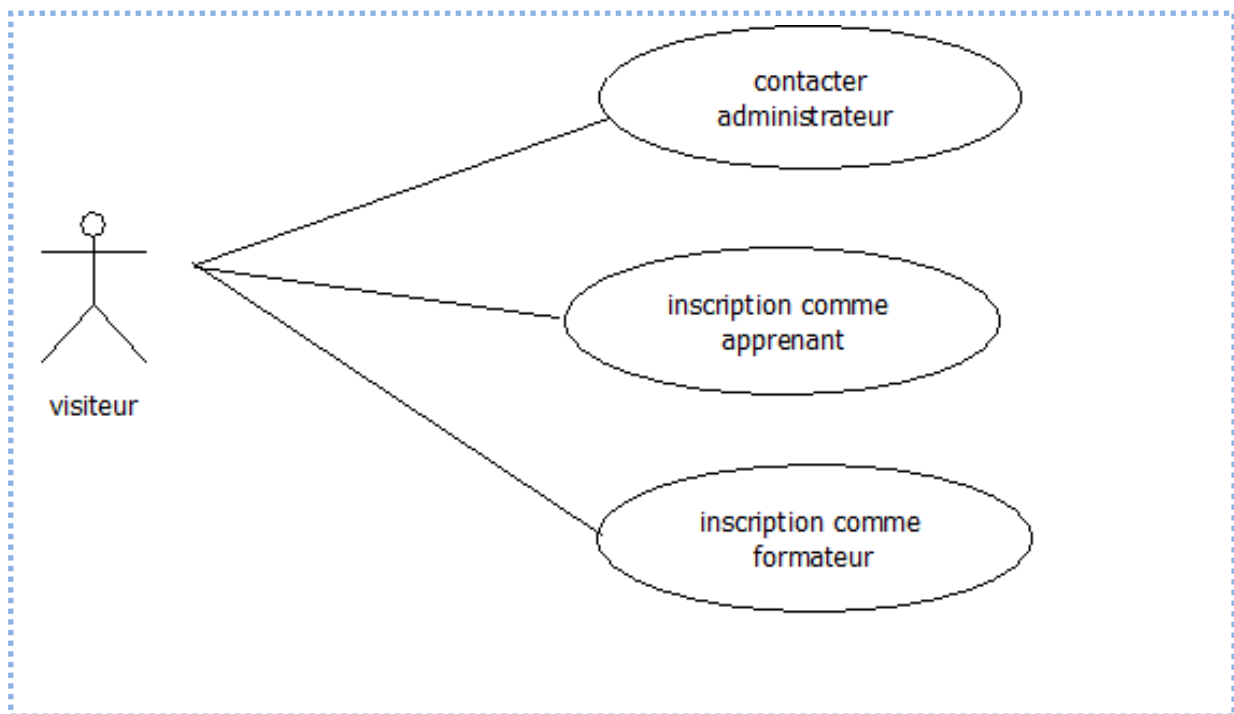
La définition des besoins fonctionnels des acteurs présentée ci-dessous nous permet de dégager le diagramme de contexte suivant :



**Figure 3.1** : Diagramme de contexte de l'application

### 3.4.2 Détermination des cas d'utilisation

Etant donné que les acteurs sont identifiés, nous sommes en mesure de modéliser les cas d'utilisation de ces acteurs. Un cas d'utilisation décrit ce que l'utilisateur veut fondamentalement faire avec le système. Il s'agit d'une représentation macroscopique des interactions entre acteurs et le système fondée sur une sémantique "faible" qui rend ce modèle aisément compréhensible par les utilisateurs.



**Figure 3.2 :** Diagramme de cas d'utilisation navigué dans l'application.

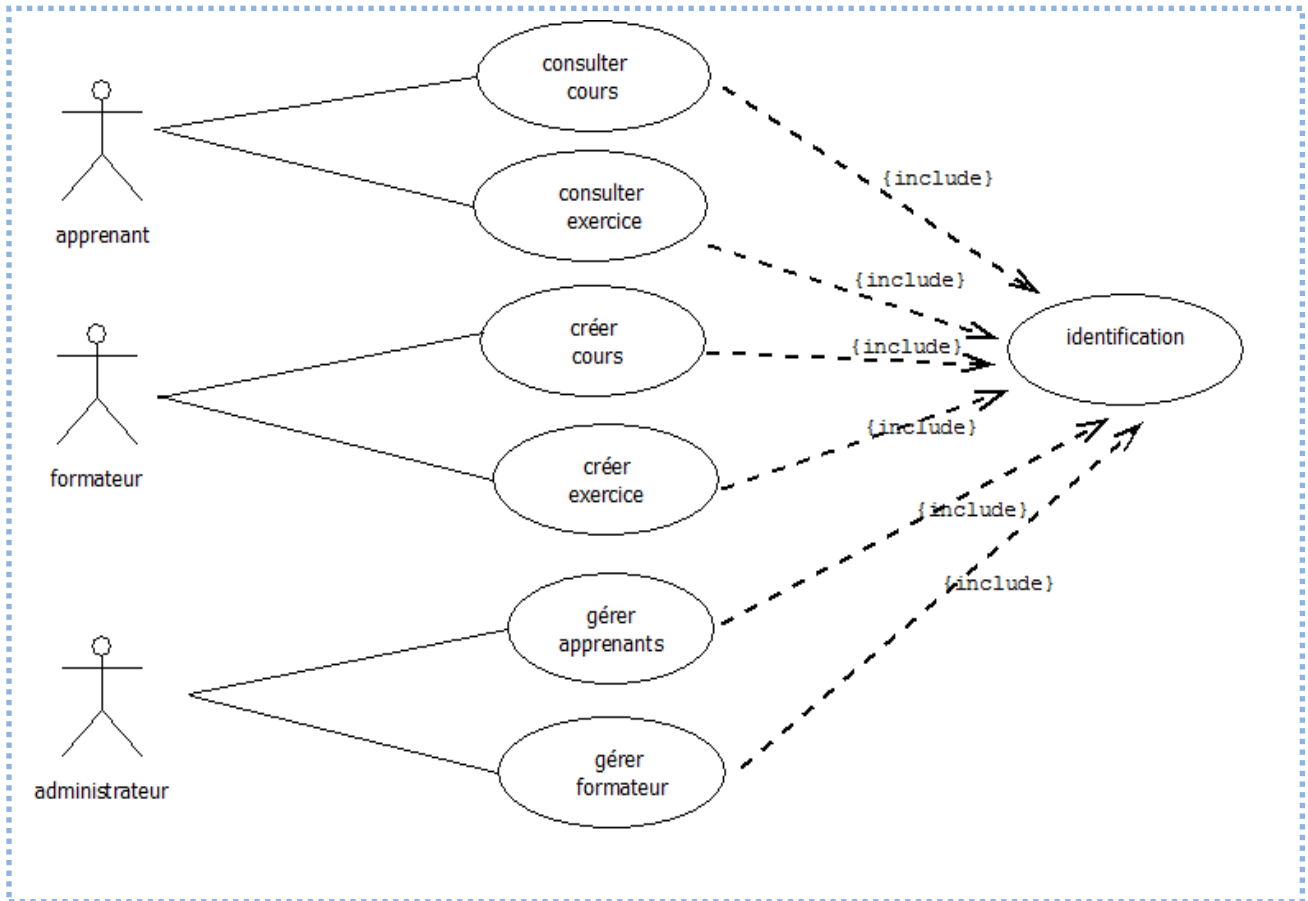
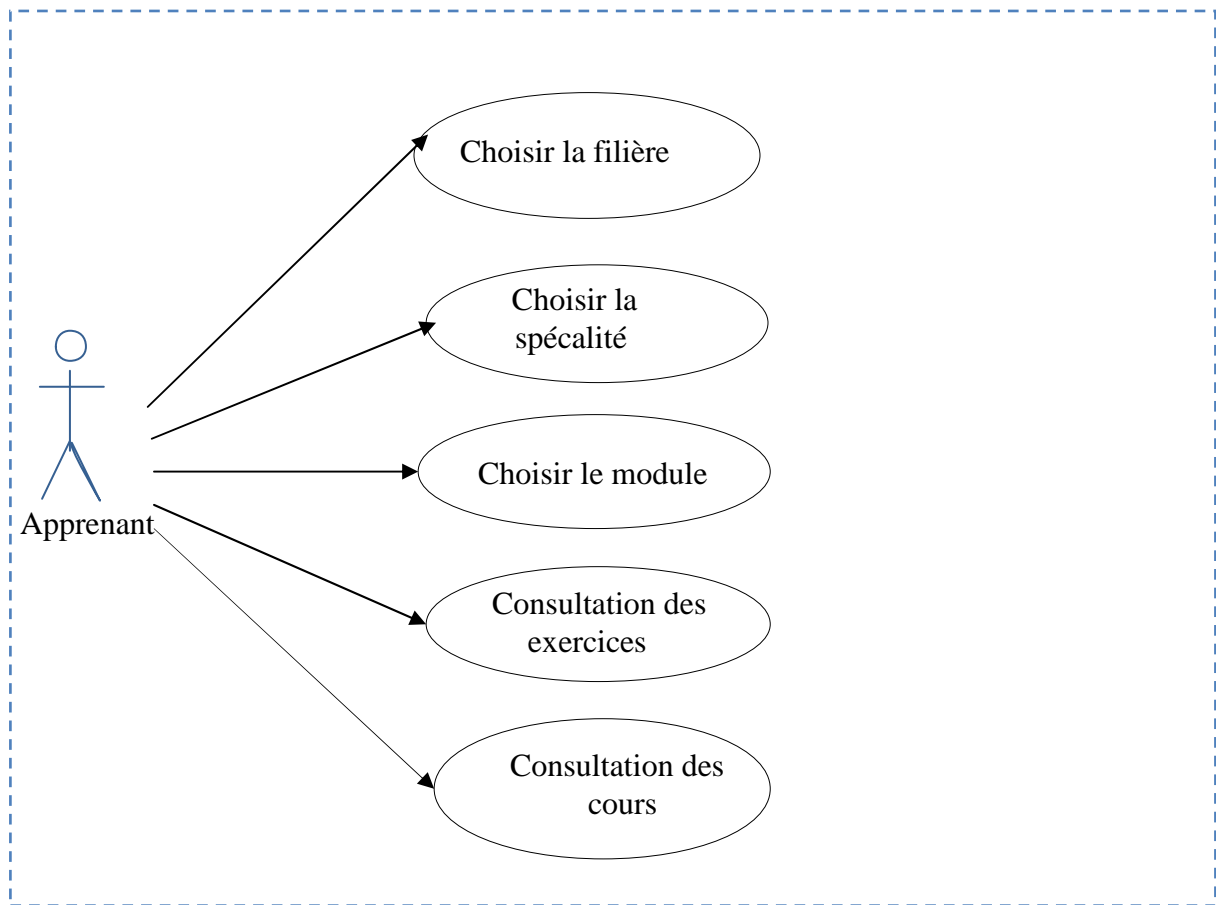
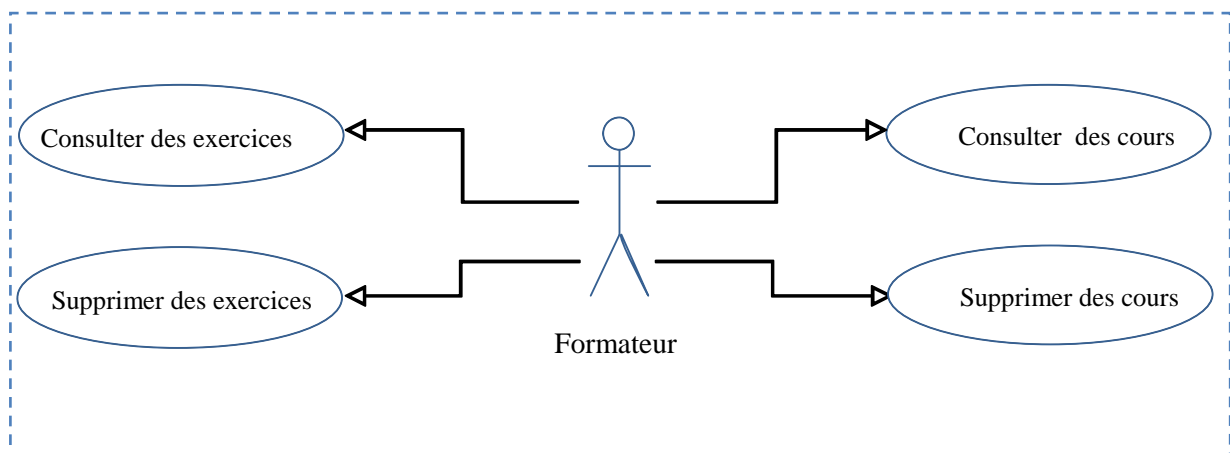


Figure 3.3 : Diagramme de cas d'utilisation « Authentification ».



**FIGURE 3.4 : Diagramme de cas d'utilisation « consulter module »**



**FIGURE 3.5 : Diagramme de cas d'utilisation « Gérer module »**

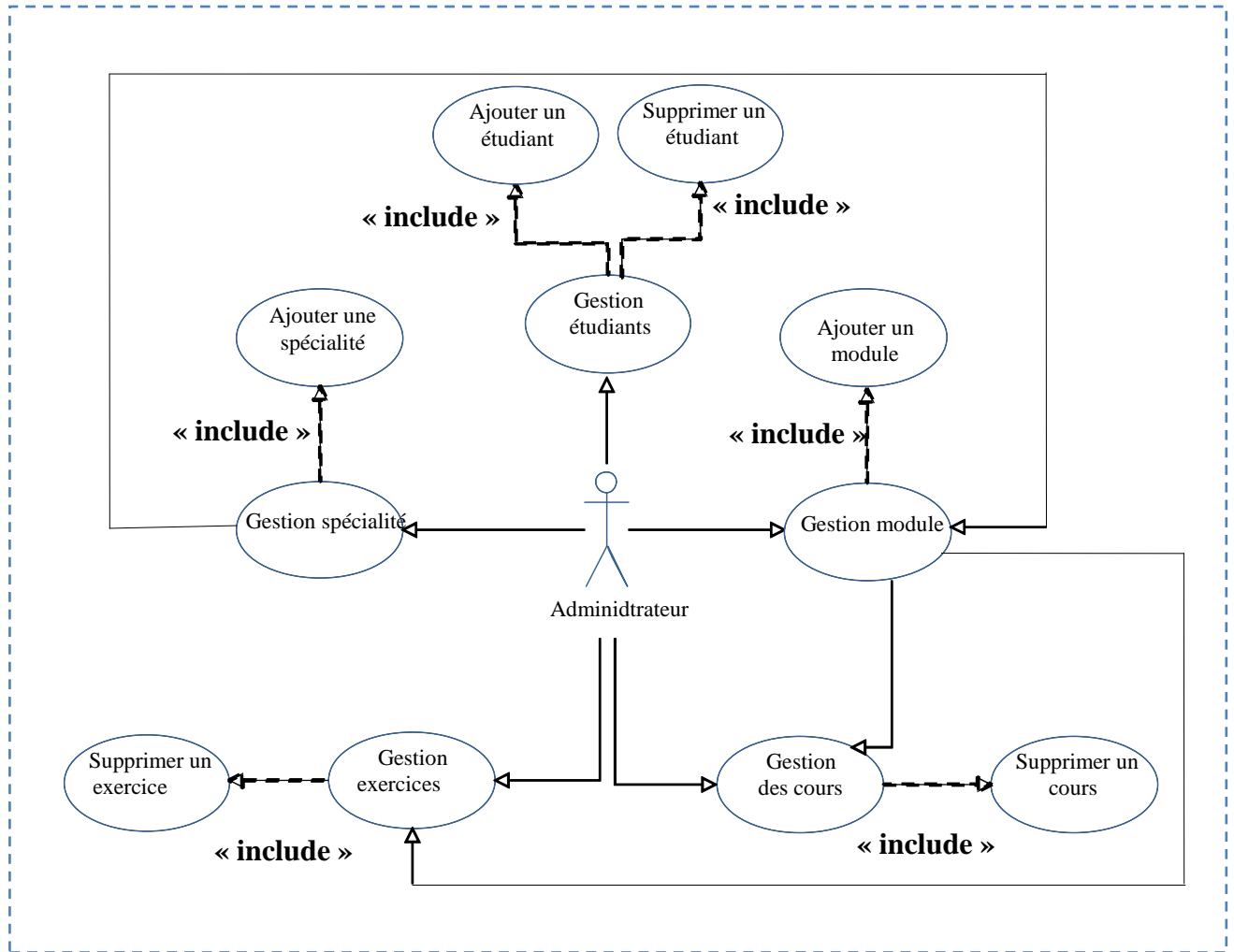


FIGURE 3.6 : Diagramme de cas d'utilisation « Gestion filière»

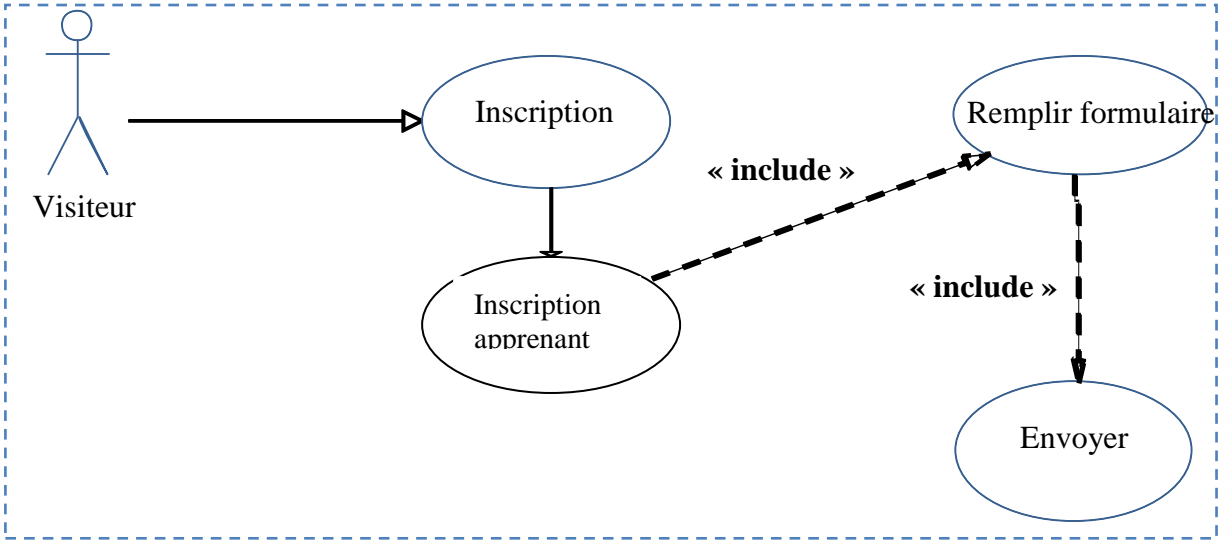


FIGURE 3.7: Diagramme de cas d'utilisation « Inscription apprenant »

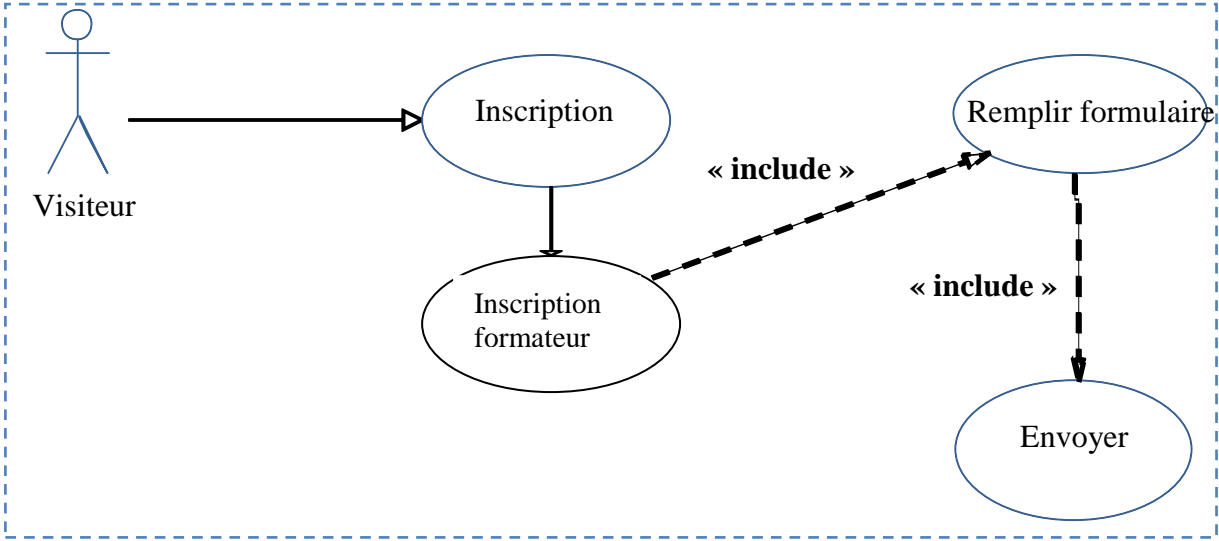


FIGURE 3.8: Diagramme de cas d'utilisation « Inscription formateur »

### 3.5 Description textuelle des cas d'utilisation

Après avoir déterminé les cas d'utilisation de l'application, on les décrit de manière textuelle avec leurs scénarios argumentés à la fin avec quelques cas de séquences qu'on appellera diagramme de séquence du système.

#### 3.5.1 Cas d'utilisation « Consulter module »

➤ **Identification du cas d'utilisation :**

**Cas d'utilisation :** Consulter cours ;

**Acteur :** Apprenant ;

**Résumé :** L'utilisateur accède à son espace personnel dans l'application, et choisit de consulter un cours d'un module donné de la spécialité de sa filière.

➤ **Description des scénarios :**

**Scénario nominal**

1. Après authentification,
2. Le système affiche la page de l'espace personnel et l'utilisateur clique sur « envoyer »,
3. Le système affiche la filière où l'utilisateur est inscrit, ensuite il clique dessus,
4. Le système affiche la liste des spécialités de la filière choisie auparavant,
5. L'utilisateur choisit une spécialité en cliquant dessus,
6. Le système lui affiche la page des modules de sa spécialité, et l'utilisateur en sélectionne un,
7. Le système lui affiche le choix « cours », et le membre décide de consulter les cours de ce module en cliquant sur « consulter » ou bien « télécharger »,
8. Le système lui affiche l'ensemble de tous les cours du module choisi, et le membre visualise le contenu de l'un de ces cours en cliquant sur son lien pour le télécharger.

### 3.5.2 Cas d'utilisation : « Consulter exercices »

➤ **Identification du cas d'utilisation :**

**Acteur :** Apprenant ;

**Résumé :** L'utilisateur accède à son espace personnel dans l'application, et choisit de consulter un exercice d'un module donné de la spécialité de sa filière.

➤ **Description des scénarios :**

**Scénario nominal**

9. Après authentification,
10. Le système affiche la page de l'espace personnel et l'utilisateur clique sur « envoyer »,
11. Le système affiche la filière où l'utilisateur est inscrit, ensuite il clique dessus,
12. Le système affiche la liste des spécialités de la filière choisie auparavant,
13. L'utilisateur choisit une spécialité en cliquant dessus,
14. Le système lui affiche la page des modules de sa spécialité, et l'utilisateur en sélectionne un,
15. Le système lui affiche le choix « exercices », et le membre décide de consulter les exercices en cliquant sur « consulter »,
16. Le système lui affiche les exercices du module choisi.

### 3.5.3 Cas d'utilisation « Gestion de filière »

➤ **Identification du cas d'utilisation :**

**Cas d'utilisation :** Créer une nouvelle spécialité;

**Acteur :** Administrateur

**Résumé :** Cette fonctionnalité permet à l'administrateur de mettre à jour une filière en lui ajoutant une nouvelle spécialité.

➤ **Description des scénarios :**

**Scénario nominal :**

1. Après authentification,
2. L'administrateur atteint son interface, et clique sur suivant pour continuer,
3. Le système affiche une interface contenant les choix : « Apprenant », « Formateur », « Cours », « Exercices » et « Filières »,
4. L'administrateur décide de gérer les spécialités et clique sur « Filières »,
5. Le système lui retourne une interface contenant l'ensemble de toutes les filières.
6. L'administrateur choisit une filière et clique dessus,
7. Le système affiche la liste des spécialités de la filière choisie auparavant ainsi que le choix « nouvelle spécialité »,
8. L'administrateur crée une nouvelle spécialité en cliquant sur « nouvelle spécialité »,
9. Le système lui affiche la page de création d'une spécialité,
10. L'administrateur remplit les champs nécessaires et clique sur « Valider »,
11. Le système retourne un message de confirmation de création d'une spécialité.

### 3.5.4 Cas d'utilisation : « Créer un nouveau module »

➤ **Identification du cas d'utilisation :**

**Acteur :** Administrateur

**Résumé :** Cette fonctionnalité permet à l'administrateur de mettre à jour une spécialité en lui ajoutant un nouveau module.

➤ **Description des scénarios :**

**Scénario nominal :**

1. Après authentification,
2. L'administrateur atteint son interface, et clique sur suivant pour continuer,
3. Le système affiche une interface contenant trois choix « Apprenant », « Formateur », « Cours », « Exercices » et « Filières »,
4. L'administrateur décide de gérer les spécialités et clique sur « Filières »,

5. Le système lui retourne une interface contenant l'ensemble de toutes les filières.
6. L'administrateur choisit une filière et clique dessus,
7. Le système affiche la liste des spécialités de la filière choisie auparavant ainsi que le choix « nouvelle spécialité »,
8. L'administrateur clique sur une spécialité,
9. Le système lui affiche l'interface des modules de la spécialité ainsi que le choix « nouveau module »,
10. L'administrateur crée un nouveau module en cliquant sur « nouveau module »,
11. Le système lui affiche la page de création d'un module,
12. L'administrateur remplit les champs nécessaires et clique sur « Valider »,
13. Le système retourne un message de confirmation de création d'un module.

### 3.5.5 Cas d'utilisation : « Supprimer un cours »

#### ➤ Identification du cas d'utilisation :

**Acteur** : Administrateur

**Résumé** : Cette fonctionnalité permet à l'administrateur de supprimer un cours déjà publié par un enseignant.

#### ➤ Description des scénarios :

##### Scénario nominal :

1. Après authentification,
2. L'administrateur atteint son interface, et clique sur suivant pour continuer,
3. Le système affiche une interface contenant trois choix « Apprenant », « Formateur », « Cours », « Exercices » et « Filières »,
4. L'administrateur décide de gérer les cours et clique sur « Cours »,
5. Le système lui retourne une interface contenant l'ensemble de tout les cours publiés.
6. L'administrateur choisit un cours et clique sur supprimer,
7. Le système retourne un message de confirmation de suppression d'un cours.

### 3.5.6 Cas d'utilisation « Créer un nouveau cours»

➤ **Identification du cas d'utilisation :**

**Acteur :** Formateur

**Résumé :** Cette fonctionnalité permet à l'enseignant de mettre à jour un module en lui ajoutant un nouveau cours.

➤ **Description des scénarios :**

**Scénario nominal :**

1. Après authentification,
2. Le formateur atteint son interface, et clique sur suivant pour continuer,
3. Le système affiche la page de l'espace personnel avec les choix : « exercices », « cours », « publier exercices » et « publier cours»
4. Le formateur décide de publier un nouveau cours en cliquant sur « publier cours»,.
5. Le système lui affiche la page de création d'un cours,
6. Le formateur remplit les champs nécessaires et clique sur « Valider »,
7. Le système retourne un message de confirmation de création d'un cours.

### 3.5.7 Cas d'utilisation : « Créer un nouvel exercice »

➤ **Identification du cas d'utilisation :**

**Acteur :** Formateur

**Résumé :** Cette fonctionnalité permet à l'enseignant de mettre à jour un module en lui ajoutant un nouvel exercice.

➤ **Description des scénarios :**

**Scénario nominal :**

1. Après authentification,
2. Le formateur accède à sa propre interface, et clique sur suivant pour continuer,
3. Le système affiche la page de l'espace personnel avec les choix : « exercices », « cours », « publier exercices » et « publier cours »
4. Le formateur décide de mettre en ligne un nouveau exercice en cliquant sur « publier exercices »,
5. Le système lui affiche la page de création d'un exercice ,
6. Le formateur remplit les champs nécessaires et clique sur « Valider »,
7. Le système retourne un message de confirmation de création d'un exercice.

### 3.5.8 Cas d'utilisation « Authentification »

➤ **Identification du cas d'utilisation :**

**Acteur :** Formateur/Apprenant/Administrateur.

**Résumé :** Cette fonctionnalité permet à chaque utilisateur de s'authentifier par un login et mot de passe pour accéder à son espace.

➤ **Description des scénarios :**

**Scénario nominal :**

1. L'utilisateur atteint le site ;
2. Le système lui affiche la page d'accueil;
3. L'utilisateur choisit son profil ;
4. Le système affiche la page d'authentification ;
5. L'utilisateur saisit son login et son mot e passe, puis il clique sur « envoyer » ;
6. Le système vérifie les informations saisies avec celles de la base de données;
7. Apres validation, l'utilisateur atteint son espace.

### 3.5.9 Cas d'utilisation « Inscription »

➤ **Identification du cas d'utilisation :**

**Cas d'utilisation :** Inscription apprenant.

**Acteur :** visiteur.

**Résumé :** Cette fonctionnalité permet au visiteur de s'inscrire en tant qu'apprenant.

➤ **Description des scénarios :**

**Scénarios nominal :**

1. L'utilisateur atteint le site ;
2. Le système lui affiche la page d'accueil;
3. L'utilisateur choisit son profil en cliquant sur le bouton « visiteur » ;
4. Le système lui affiche la page s'inscrire ;
5. L'utilisateur clique sur « apprenant ».
6. Le système lui affiche le formulaire d'inscription ;
7. Après remplissage du formulaire, l'utilisateur clique sur « envoyer » ;
8. Le système vérifie la validité des informations du formulaire, et retourne un message de confirmation d'inscription.

### 3.6 Description graphique des cas d'utilisation :

Pour documenter les cas d'utilisation, la description textuelle est indispensable, car elle seule permet de communiquer facilement avec les utilisateurs et de s'entendre sur la terminologie métier employée.

En revanche, avec le texte seul il est difficile de montrer comment les enchaînements se succèdent. Il est alors recommandé de compléter la description textuelle par un ou plusieurs diagrammes dynamiques UML. Pour notre cas nous utilisons les diagrammes de séquences pour décrire les scénarios nominaux des cas d'utilisation. Dans la suite de ce paragraphe nous présentons seulement à titre d'exemple quelques diagrammes.

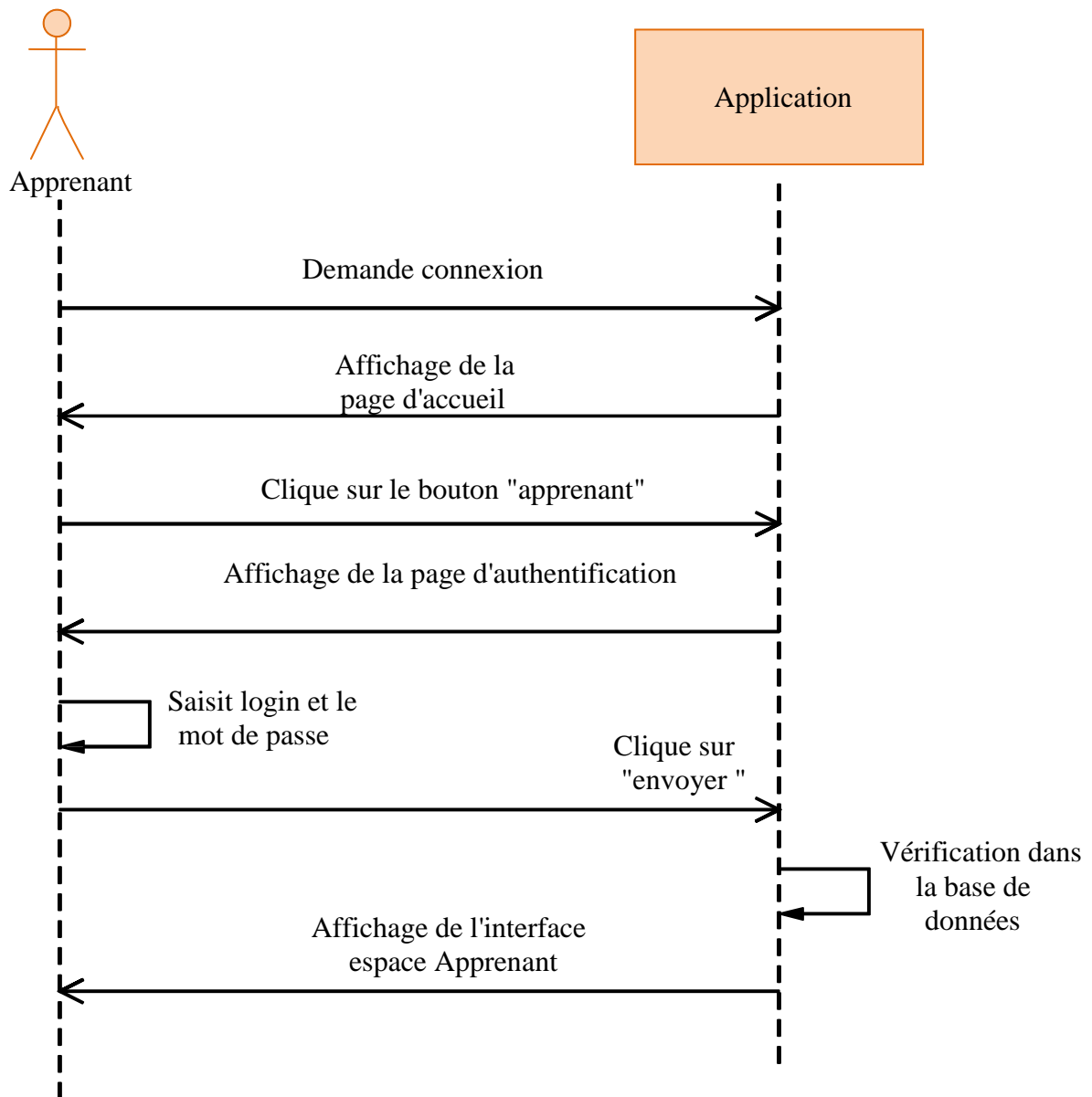


FIGURE 3.9 : Diagramme de séquence « Authentification »

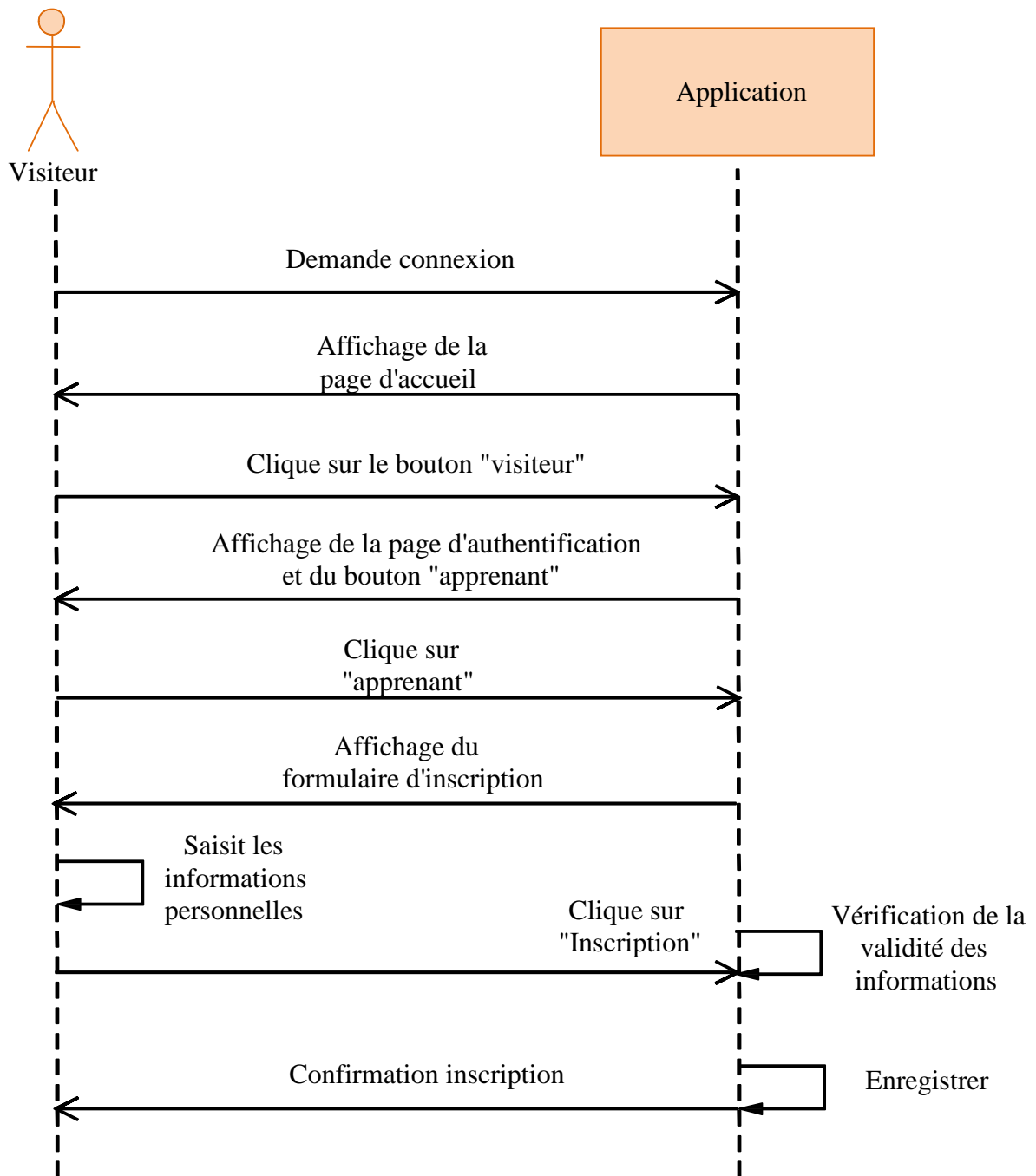


FIGURE 3.10 : Diagramme de séquence « Inscription apprenant »

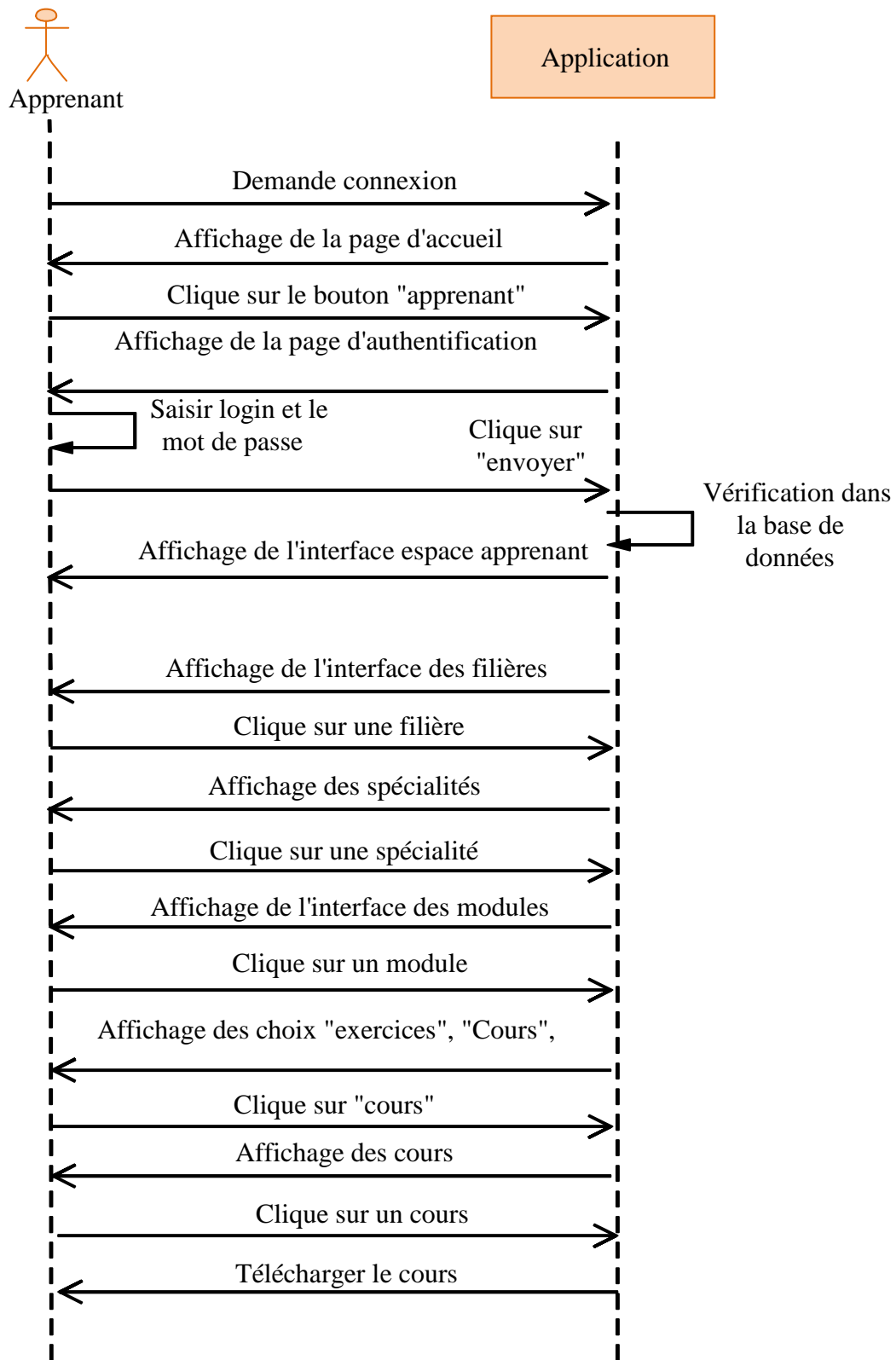


FIGURE 3.11 : Diagramme de séquence « Consulter cours »

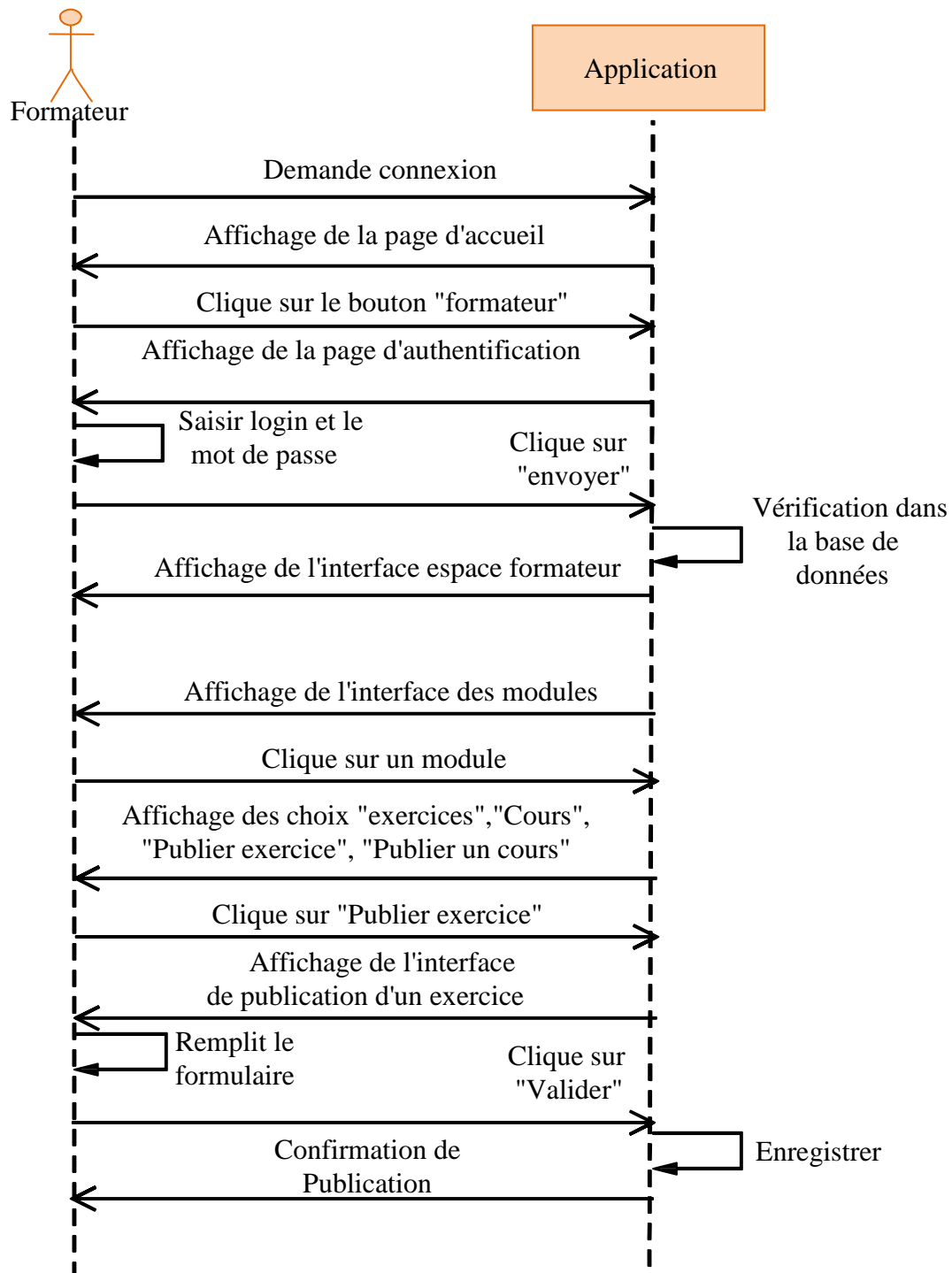


FIGURE 3.12 : Diagramme de séquence « consulter exercices »

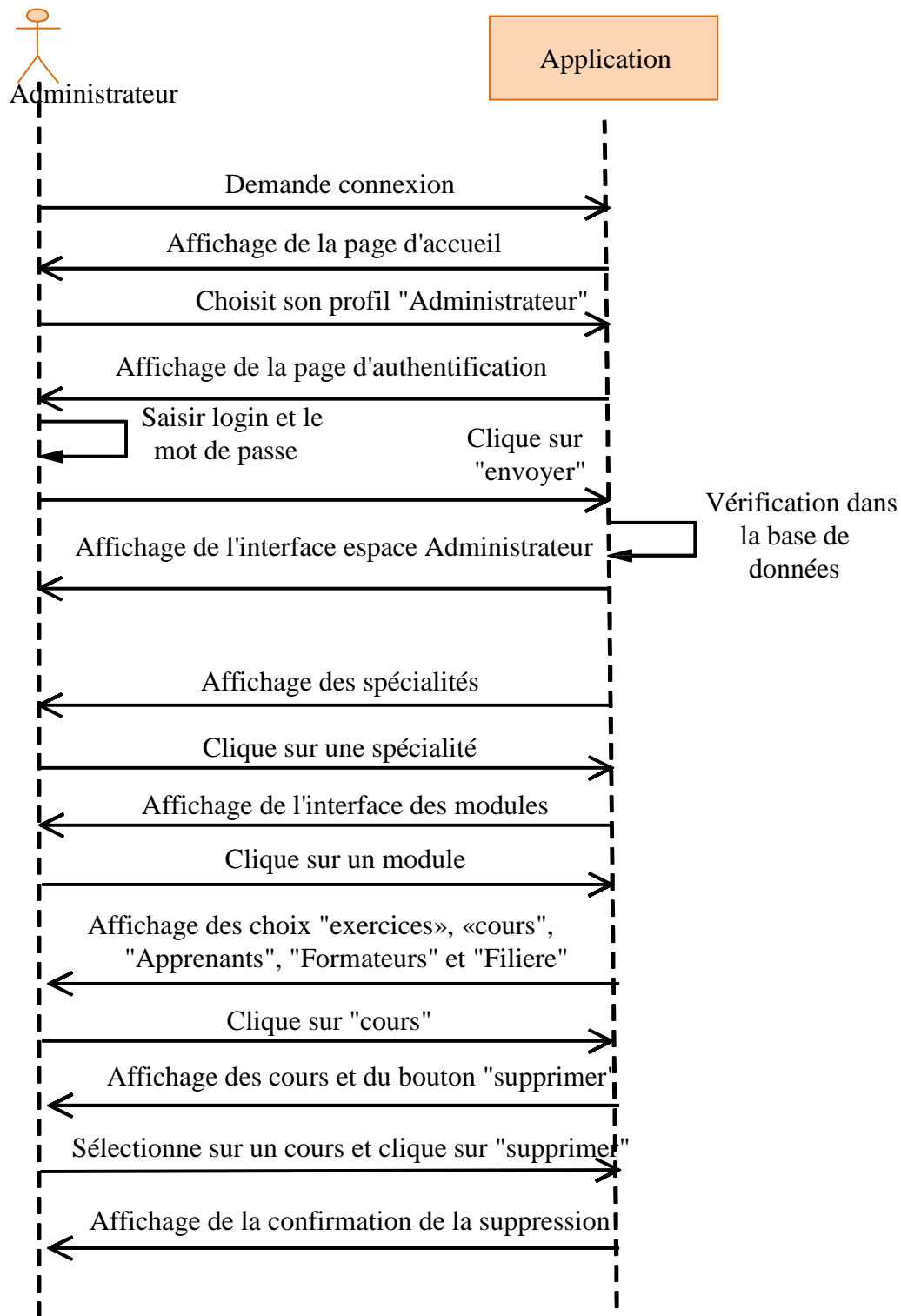


FIGURE 3.13 : Diagramme de séquence « Supprimer cours »

### **3.7 La conception objet :**

La conception est souvent considérée à tort, comme un simple enrichissement des résultats obtenus dans l'analyse. Elle se base sur le savoir-faire qui peut être interne au projet ou acquis de l'extérieur sous forme d'outils, de composants réutilisables ou plus largement de cadres de développement. Elle vise principalement à préciser le modèle d'analyse de telle sorte qu'il puisse être implémenté avec les composants de l'architecture, c'est la phase la plus complexe du projet.

Cette partie de ce chapitre est consacrée à décrire la manière de concevoir la structure de la base de données.

#### **A. Les bases de données :**

Une base de données est une collection de données sur un domaine d'application particulier, où les propriétés des données ainsi que les relations sémantiques entre ces données sont spécifiées en utilisant les concepts proposés par le modèle de données.

Le langage standard de requête structurée des bases de données relationnelles est le SQL.

SQL signifie Structured Query Language c'est-à-dire langage d'interrogation structuré. Il permet d'interroger une base de données, d'en modifier des informations. C'est un langage universel d'interrogation des bases de données, qui permet aussi à différents systèmes d'échanger des données entre eux.

#### **B. Le niveau conceptuel de la base de données :**

Après avoir élaborés les diagrammes de séquence des différents cas d'utilisations déjà décrits, nous allons élaborer le diagramme de classes général qui représente une vue conceptuel de la base de données.

##### **❖ *Le diagramme de classe :***

Le diagramme de classe a toujours été le diagramme le plus important dans les méthodes orientées objets. Il représente la structure statique d'un système. Il contient principalement des classes, ainsi que leurs associations, mais on peut aussi y trouver des objets.

L'intérêt majeur du diagramme de classes est de modéliser les entités du système.

❖ Le schéma conceptuel :

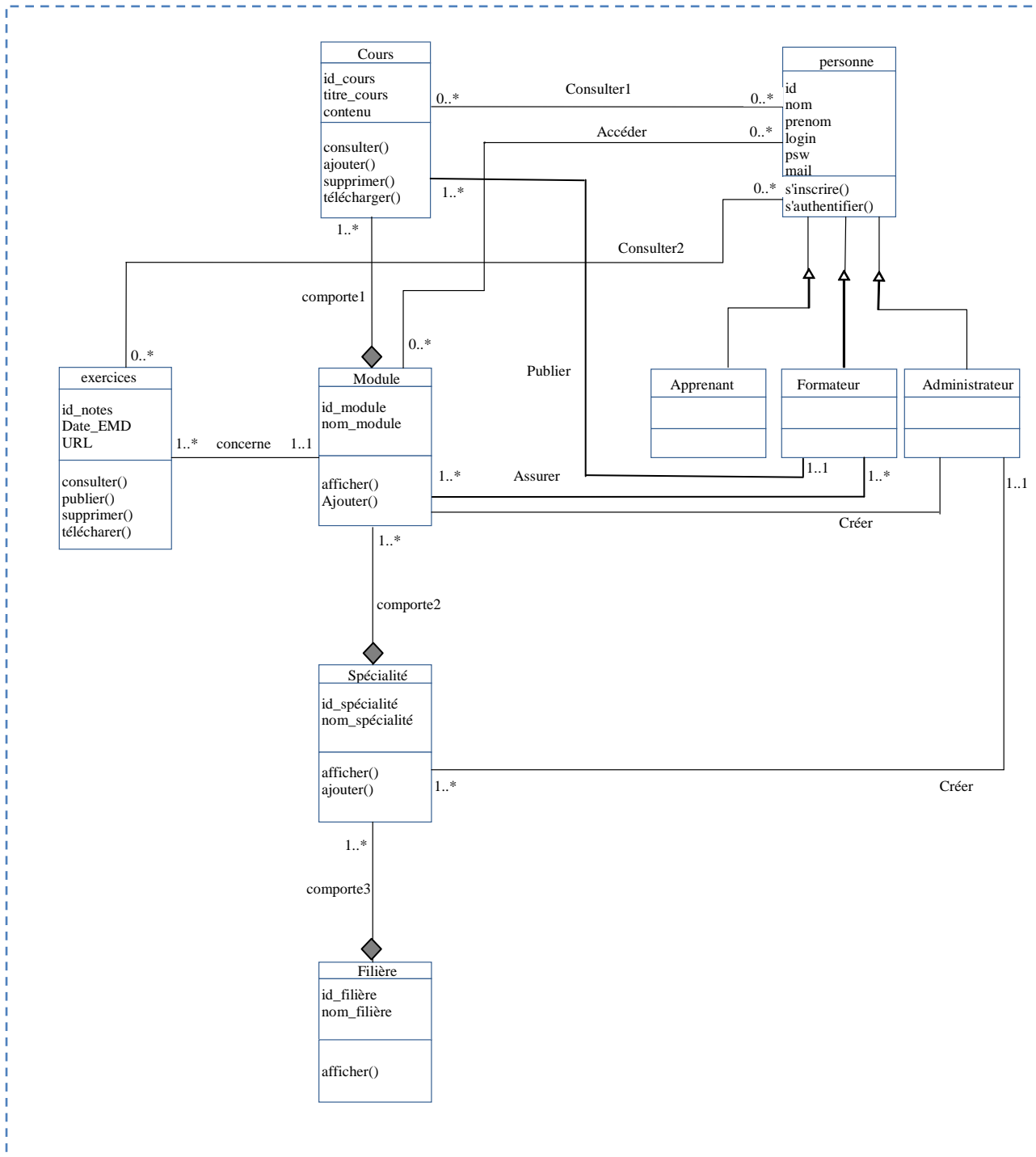


FIGURE 3.14: Diagramme de classe général

### **Conclusion :**

Ce chapitre est consacré à l'analyse et à la conception de l'application avec le langage UML pour le web.

Nous avons donc commencé par la définition des besoins qui ont été traduits par la suite en formalismes graphiques à l'aide des diagrammes offerts par ce langage.

La partie conception a mis en évidence les différents composants de l'application et cela dans le but de construire notre base de données.

Donc à ce stade nous sommes assez armés pour mettre sur pied notre application, ce qui va être l'objet du chapitre suivant tout en exposant l'environnement de développement.

## 4. Introduction

Après avoir fait une étude détaillée des différents modules constituant notre application, nous allons nous pencher sur la mise en œuvre de notre système ; en choisissant les différents outils de développement correspondants à des critères bien définis pour atteindre des objectifs bien précis.

Dans le chapitre précédent, nous avons détaillé l'étape de conception de notre système. Les résultats de cette étape seront employés dans la partie réalisation que nous détaillerons dans ce présent chapitre.

Nous commençons par la présentation de l'environnement de développement puis nous présentons quelques interfaces de notre application.

Notre solution se base sur les services web (nous avons présenté ce concept en détails dans le chapitre I). Dans le souci de répondre aux différents besoins fonctionnels de notre système.

### 4.1 Le modèle relationnel :

Le modèle relationnel présente l'univers des données décrits dans le modèle conceptuel (diagramme de classe), en tenant compte le type de base de données choisie ; en d'autre terme le modèle relationnel traduit le modèle conceptuel en un formalisme compressible par la machine.

Les relations récences du schéma sont :

**Administrateur** ( id\_admin, nom\_admin, prenom\_admin, login\_admin, pass\_admin).

**Formateur** ( id\_for, nom\_for, prenom\_for, login\_for, pass\_for, adress, sexe, email).

**Apprenant**( id\_app, nom\_app, prenom\_app, login\_app, pass\_app, adress, diplôme, email, date\_naissance,sexe, id\_fil\*).

**Factulte** (id\_fac, nom\_fac).

**Filiere** ( id\_fil ,nom\_fil, id\_fac\*).

**Specialite** (id\_spec, nom\_spec, id\_fil).

**Module** ( id\_mod, nom\_mod,id\_spec\*).

**Module\_formateur** (id\_mod\_for, charge\_mod, id\_mod\*)

**Cours** (id\_cours, titre, url, description, id\_mod\*).

**Exercice** (id\_exo, lib, contenu\_exo, id\_cours\*).

Remarque :

- Les attributs soulignés sont des clés primaires.
- Les attributs avec étoile sont des clés étrangères.

#### A. Structure physique des tables de la base de données

La base de données que nous allons à présent construire contiendra les différentes tables qui comportent toutes les données nécessaires au bon fonctionnement de notre application.

**Table 1** : « Apprenant »

Nom du champ	Description	Type de données	Clé (s)
id_app	identifiant de l'apprenant	Int (5)	Primaire
nom_app	nom de l'apprenant	Varchar (20)	
prenom_app	prenom de l'apprenant	Varchar (20)	
login_app	Nom utilisateur apprenant	Varchar (25)	
pass_app	Mot de passe apprenant	Varchar (25)	
adress	Adress apprenant	Varchar (50)	
email	Email apprenant	Varchar (25)	
sexe	Sexe de l'apprenant	Varchar (10)	
Diplome	Diplôme de l'apprenant	Varchar (25)	
Date_naissance	Date de naissance de l'apprenant	date	
id_fil	Identifiant de la filiere	Int (5)	Etrangère

**Table 2** : «Formateur»

Nom du champ	Description	Type de données	Clé (s)
id_for	identifiant du formateur	Int (5)	Primaire
nom_for	nom du formateur	Varchar (20)	
prenom_for	prenom du formateur	Varchar (20)	
login_for	Nom utilisateur formateur	Varchar (25)	
pass_for	Mot de passe formateur	Varchar (25)	
adress	Adress du formateur	Varchar (50)	
sexe	Sexe du formateur	Varchar (10)	
email	Email du formateur	Varchar (50)	

**Table 3** : «Administrateur»

Nom du champ	Description	Type de données	Clé (s)
id_admin	identifiant de l' admin	Int (5)	Primaire
nom_admin	nom de l' admin	Varchar (20)	
prenom_admin	prenom de l' admin	Varchar (20)	
login_admin	Nom utilisateur admin	Varchar (25)	
pass_admin	Mot de passe admin	Varchar (25)	

**Table 4** : «Faculté»

Nom du champ	Description	Type de données	Clé (s)
id_fac	identifiant de la faculte	Int (5)	Primaire
nom_fac	nom de la faculte	Varchar (20)	

**Table 5** : «Filière»

Nom du champ	Description	Type de données	Clé (s)
id_fil	identifiant de la filière	Int (5)	Primaire
nom_fil	nom de la filière	Varchar (20)	
id_fac	Identifiant de la faculte		Etrangère

**Table 6**: «Spécialité»

Nom du champ	Description	Type de données	Clé (s)
id_spec	identifiant de la spécialité	Int (5)	Primaire
id_fil	identifiant de la filière	Int (5)	Etrangère
nom_spec	nom de la spécialité	Varchar (20)	

**Table 7** : «Module»

Nom du champ	Description	Type de données	Clé (s)
id_mod	identifiant du module	Int (5)	Primaire
nom_mod	nom de du module	Varchar (20)	
id_spec	Identifiant de la specialite	Int (5)	Etrangère

**Table 8**: «Modue\_Formateur»

Nom du champ	Description	Type de données	Clé (s)
id_mod_for	identifiant du module du formateur	Int (5)	Primaire
charge_mod	chargé du module	Varchar (20)	
id_mod*	Identifiant du module	Int (5)	Etrangère

**Table 9:** «Cours»

Nom du champ	Description	Type de données	Clé (s)
<u>id_cours</u>	identifiant du cours	Int (5)	Primaire
description	Description du cours	Varchar (50)	
titre	titre du cours	Varchar (50)	
url	L'emplacement du cours	Varchar (50)	
id_mod*	identifiant du module	Int (5)	Etrangère

**Table 10:** Exercice

Nom du champ	Designation du champ	Type de donnée	Clé(s)
<u>id_exo</u>	Identifiant de l'exercice	Int (10)	Primaire
Lib	Nom de l'exercice	VARCHAR(30)	
contenu_exo	Contenu de l'exercice	VARCHAR(100)	
id_cours	Identifiant du cours	VARCHAR(10)	Etrangère

## 4.2 Description des outils de développement :

### 4.2.1. Langage de programmation Java :

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton de Sun Microsystems. Mais c'est également un environnement d'exécution.

Java peut être séparé en deux parties. D'une part, le programme écrit en langage Java et d'autre part, une machine virtuelle (JVM) qui va se charger de l'exécution du programme Java. C'est cette plateforme qui garantit la portabilité de Java. Il suffit qu'un système ait une machine virtuelle Java pour que tout programme écrit en ce langage puisse fonctionner.

**J2EE :** Le langage de programmation utilisé est le J2EE de SUN, un langage orienté objet qui combine toutes les possibilités de manipulation des données proposées par SQL, et ce grâce à l'API JDBC. Il offre de nombreux avantages :

- C'est un langage de programmation évolué qui facilite tout les accès à la base grâce à la puissance de SQL.
- Performant, facile à la programmation et permet la portabilité.
- Supporte le modèle Client-Serveur.
- Supporte la programmation Orienté Objet.
- Permet la modularité et l'ouverture sur d'autres langages L3G.

### ***IDE (NetBeans) :***

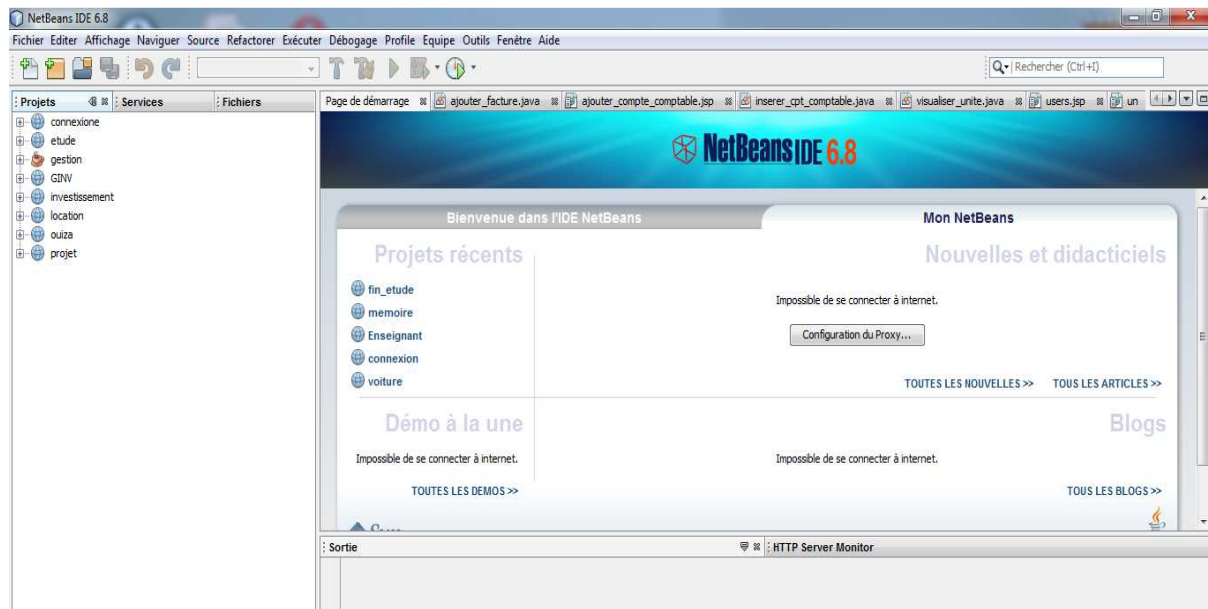
C'est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus Java, NetBeans permet également de supporter différents autres langages, comme Python, C,C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et des pages web). NetBeans est disponible sous Windows, Linux ...

Il supporte les principaux langages suivants :

- ❖ Java (Java SE, Java ME JavaFX Java EE), Java doc,
- ❖ JSP, JSF
- ❖ Ruby et Ruby on Rails: Ruby 1.9, Ruby on Rails 3, JRuby 1.4,
- ❖ Groovy et Grails,
- ❖ PHP (dont les environnements Zend et Symfony)
- ❖ JavaScriptC, C++, Fortran. Netbeans ne requiert pas l'utilisation d'un compilateur particulier. A noter le support des bibliothèques Qt. Les plates-formes supportées sont Microsoft Windows, Linux, Mac OS, Solaris 10 et OpenSolaris,
- ❖ Python,
- ❖ HTML, XHTML, RHTML,
- ❖ XML,
- ❖ DTD,
- ❖ CSS

NetBeans comprend un explorateur de bases de données qui supporte toutes les bases relationnelles pour lesquelles un connecteur JDBC existe (selon les versions des gestionnaires de bases de données): JavaDB (Derby) MySQL, PostgreSQL, Oracle, Microsoft SQL, PointBase, jTDS, IBM Redistributable DB2, ...

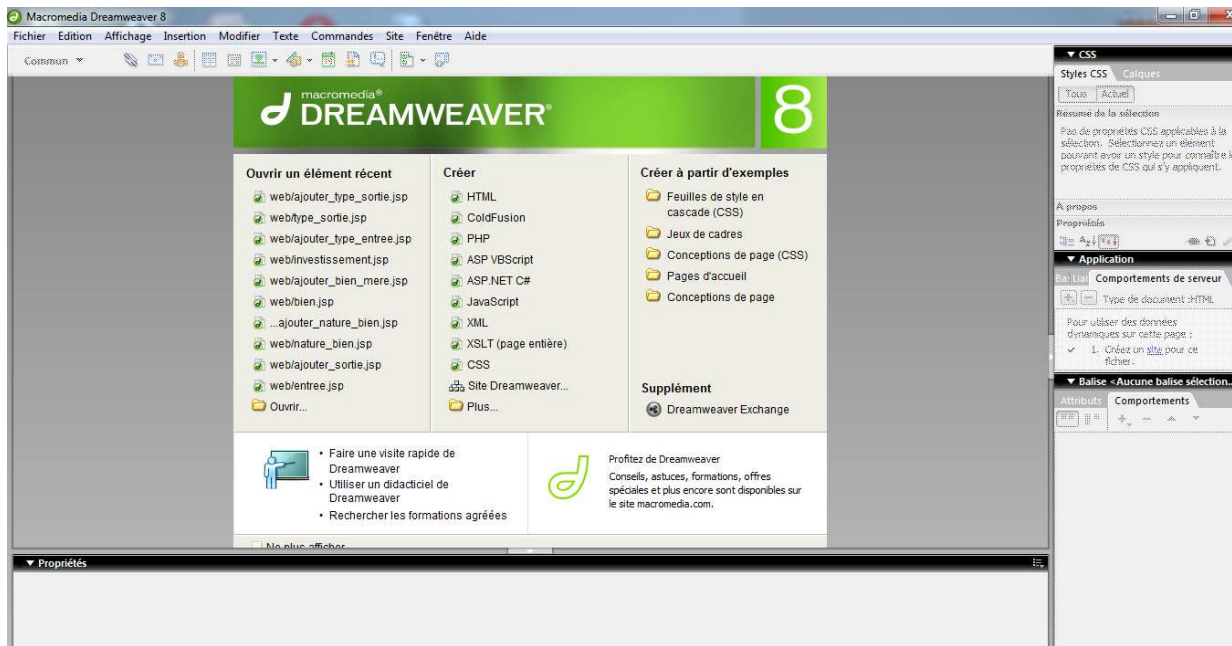
L'explorateur comprend un éditeur de requêtes, un gestionnaire intégré de bases de données MySQL.



**Figure 4.1** : Interface de l'IDE NetBeans

### ***Macromedia Dreamweaver :***

Macromedia Dreamweaver est aujourd'hui considéré comme le logiciel de création des interfaces graphiques le plus performant du marché. Est un outil convivial et très puissant destinée à la conception, au codage et au développement de pages quelque soit l'environnement du travail utilisé.



**Figure 4.2 :** Interface de Macromedia Dreamweaver de pages graphiques.

### *Le serveur web Apache :*

Le serveur Apache est un serveur web qui s'occupe du traitement des requêtes des clients, le transfert des pages HTML au browser et l'exécution des programmes sur la machine serveur. Nous avons utilisé le serveur web Apache, du fait qu'il est le serveur le plus utilisé.

Notre choix pour apache est motivé par les nombreux avantages qu'il présente :

Extrême portabilité (fonctionne sous les différentes plates-formes UNIX et Windows).

- Modestie des exigences matérielles requises pour un niveau élevé de performance.
- Gratuit.
- Extensible, modulaire et configurable.
- Etc....

Dans notre cas apache est intégré à IDE NetBeans 6.8 dans sa version 6.0.20, qui s'appelle Apache Tomcat 6.0.20

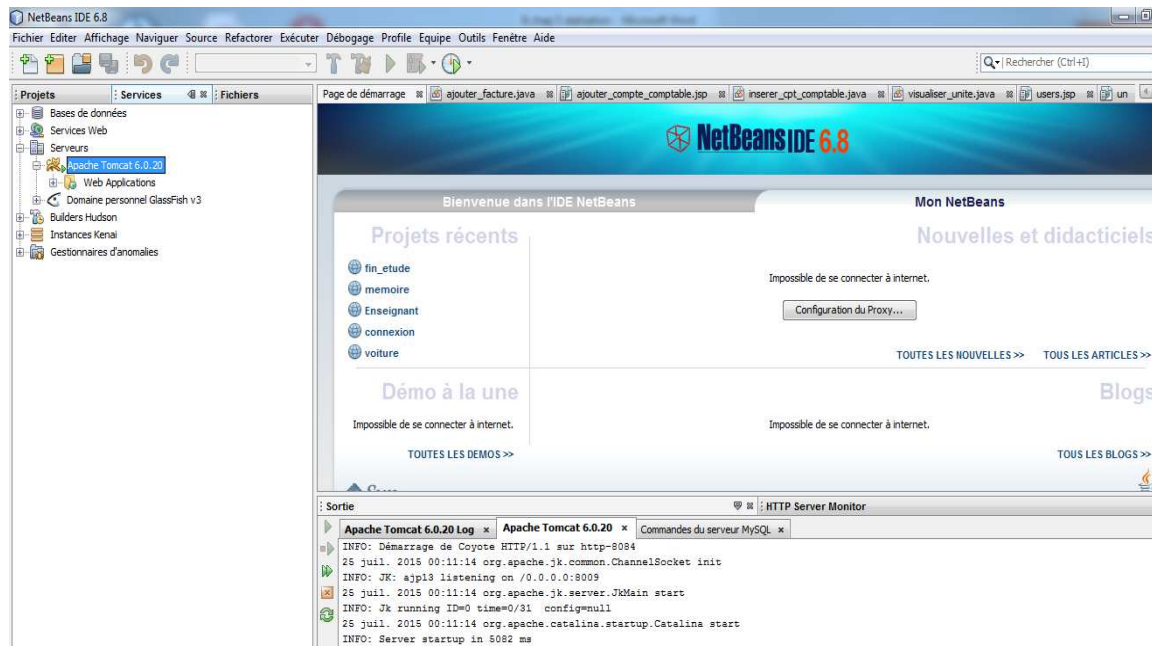


Figure 4.3 : Apache Tomcat 8.0.27 démarré sur NetBeans 8.1

### ***Le SGBD (MYSQL):***

MySQL est un gestionnaire de base de données libre, basé sur un langage de requêtes SQL, il est très utilisé dans les projets libres et dans le milieu industriel.

MySQL est un SGBD relationnel développé dans un souci de performances élevées.

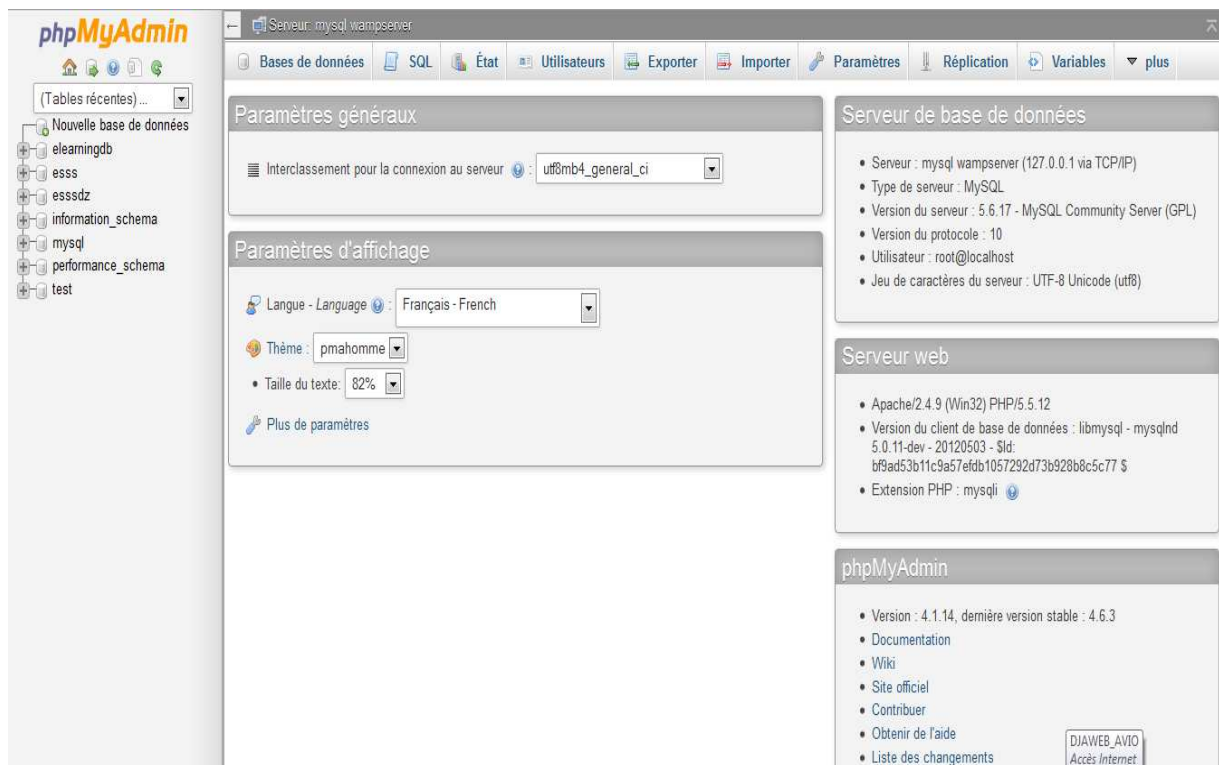
On est opté pour Phpmyadmin qui permet de manipuler MySQL de manière très simple.

### ***PhpMyAdmin :***

L'outil PhpMyAdmin est développé en PHP, il offre une interface graphique pour l'administration de base de données MySQL via un navigateur web.

Les fonctions principales de PhpMyAdmin sont :

- ❖ Création de nouvelles bases de données,
- ❖ Création/Suppression/Modification de tables,
- ❖ L'édition, l'ajout et la suppression des champs,
- ❖ L'exécution des commandes et des requêtes SQL



**Figure 4.4** : L'interface PhpMyAdmin

## 4.2.2. Les Langages du web :

### A. Le coté client :

- **Un langage de description HTML :**

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

HTML, dans sa première mouture, a été inventé par l'équipe de Tim Berners Lee en même temps qu'HTTP.

---

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

- **CSS :**

Le CSS (*Cascading Style Sheets* : feuilles de style en cascade) sert à décrire le style du contenu d'un fichier HTML et XML.

L'HTML aussi dispose d'outils pour la mise en pages. Mais la tendance actuelle est d'utiliser l'HTML pour décrire le type et la structure du contenu (texte, images..), et le CSS comme outil pour la mise en page et la définition du style. Le but est de distinguer la structure du document, du style de son contenu, ceci afin d'en faciliter la manipulation.

- **JavaScript:**

JavaScript est un langage de scripts. Il est programmé sous forme de texte et intégré tel quel au code HTML de la page. Le script est le code source du langage.

Java Script est interprété, c'est-à-dire que le script est lu, analysé et exécuté directement depuis le code source sous sa forme textuelle par le navigateur sans aucune transformation. Il n'y a pas de phase de compilation comme avec d'autres langages de programmation qui compilent le code source en un code exécutable. Java Script est exécuté côté client. Les informaticiens utilisent l'expression « Langage client » pour le désigner.

## **B. Le coté serveur :**

Les dernières technologies Web coté serveur sont ASP (Active Server Page), les Servlets Java, JSP (Java Server Page) et PHP (Personnel Home Page).

- **Les Servlets :**

Une servlet est un programme java qui utilise des modules supplémentaires figurant dans l'API java. Son exécution se fait dans un moteur de Servlet ou conteneur de Servlet permettant d'établir le lien entre la Servlet et le serveur Web. Les servelets permettent donc de gérer des requêtes HTTP et de fournir au client une réponse HTTP dynamique (donc de créer des pages Web dynamiques).

- **Les JSP :**

Les JSP (*Java Server Page*) est l'une des composantes principales de la programmation Java, qui permet de combiner le langage de marquage (HTML ou XML) avec des fragments de code Java, pour produire une page dynamique. Chaque page est automatiquement compilée en une servlet (par le moteur de JSP) à sa première demande et puis exécutée.

En d'autres termes, JSP désigne un fichier texte simple, composé de code HTML ou XML et d'éléments JSP (code Java abrégé). Lorsqu'un client demande une page JSP de serveur qui n'a pas été exécutée, elle est d'abord transmise au moteur de JSP. Ce moteur compile la page en une servlet, l'exécute et envoie au client. Ensuite, le moteur de servlet du serveur Web exécute la page compilée.

- **SoapUI :**

SoapUI est une application open source permettant le test de web service dans une architecture orientée services (SOA). Ses fonctionnalités incluent l'inspection des web services, l'invocation, le développement, la simulation, les tests fonctionnels, les tests de charge et de conformité.

SoapUI a été publié pour la première fois en septembre 2005 sous licence public générale limitée GNU. Il est entièrement basé sur la plateforme Java et utilise Swing pour l'interface utilisateur. Ce qui signifie que SoapUI est multiplateforme. SoapUI supporte aujourd'hui Eclipse et Netbeans

- **HibernateFramework :**

Hibernate est un Framework open source gérant la persistance des objets en base de données relationnelle.

Hibernate est adaptable en terme d'architecture, il peut donc être utilisé aussi bien dans un développement client lourd, que dans un environnement web légère type apache tomcat ou dans un environnement java EE complet : WebSphere, JBoss Application Server et Oracle WebLogic Server.

Hibernate apporte une solution aux problèmes d'adaptation entre le paradigme objet et les SGBD en remplaçant les accès de la base de données par des appels à des méthodes objet de haut niveau.

### 4.2.3. Présentation des interfaces :

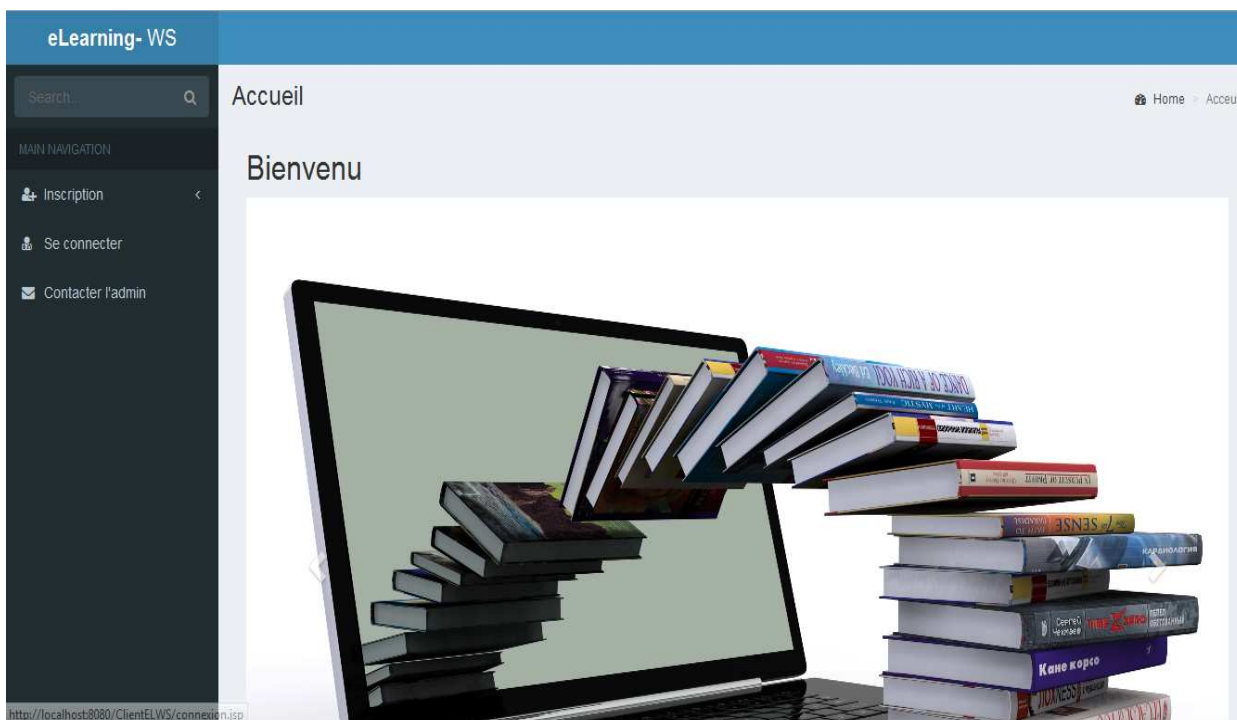


Figure 4.5 : page d'accueil

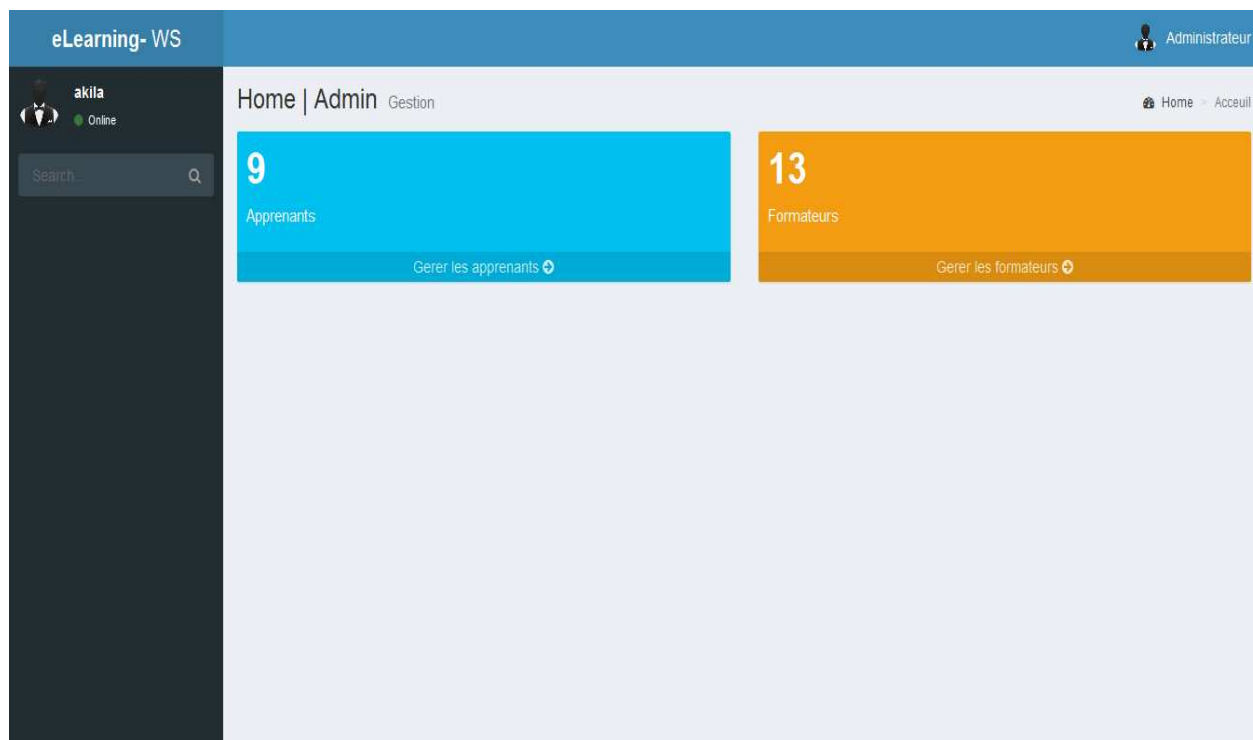


Figure 4.6 : espace administrateur

The screenshot shows the 'Inscription' page for a new learner in the Learning-WS system. The page title is 'Inscription' with the subtitle 'Inscrire un nouvel apprenant'. The breadcrumb trail is 'Home > Inscription'. The main heading is 'Formulaire d'inscription d'un nouvel apprenant'. The form contains the following fields:

- nom**: A text input field with the placeholder text 'Votre nom .....
- Prénom**: A text input field with the placeholder text 'Votre prénom .....
- Email**: A text input field with the placeholder text 'Votre email .....
- Sexe**: A dropdown menu with 'Male' selected.
- Adresse**: A large text area with a placeholder text '.....'.
- Diplome**: A text input field (partially visible at the bottom).

**Figure 4.7: Inscription apprenant**

The screenshot shows the 'Inscription' page for a new teacher in the eLearning-WS system. The page title is 'Inscription' with the subtitle 'Inscrire un nouveau formateur'. The breadcrumb trail is 'Home > Inscription'. The main heading is 'Formulaire d'inscription d'un nouveau formateur'. The form contains the following fields:

- nom**: A text input field with the placeholder text 'Votre nom .....
- Prénom**: A text input field with the placeholder text 'Votre prénom .....
- Email**: A text input field with the placeholder text 'Votre email .....
- Sexe**: A dropdown menu with 'Male' selected.
- Adresse**: A large text area with a placeholder text '.....'.
- Login**: A text input field with the placeholder text 'Votre nom .....

**Figure 4.8: Inscription formateur**

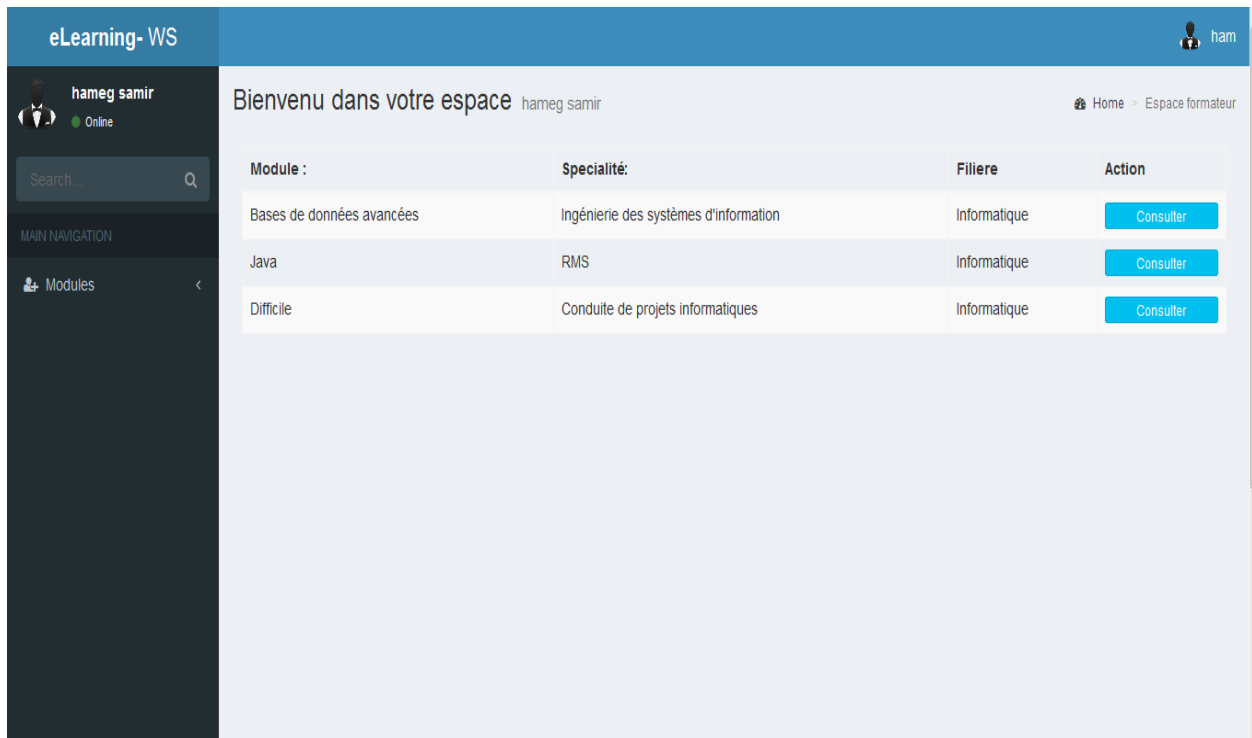


Figure 4.9 : Espace formateur

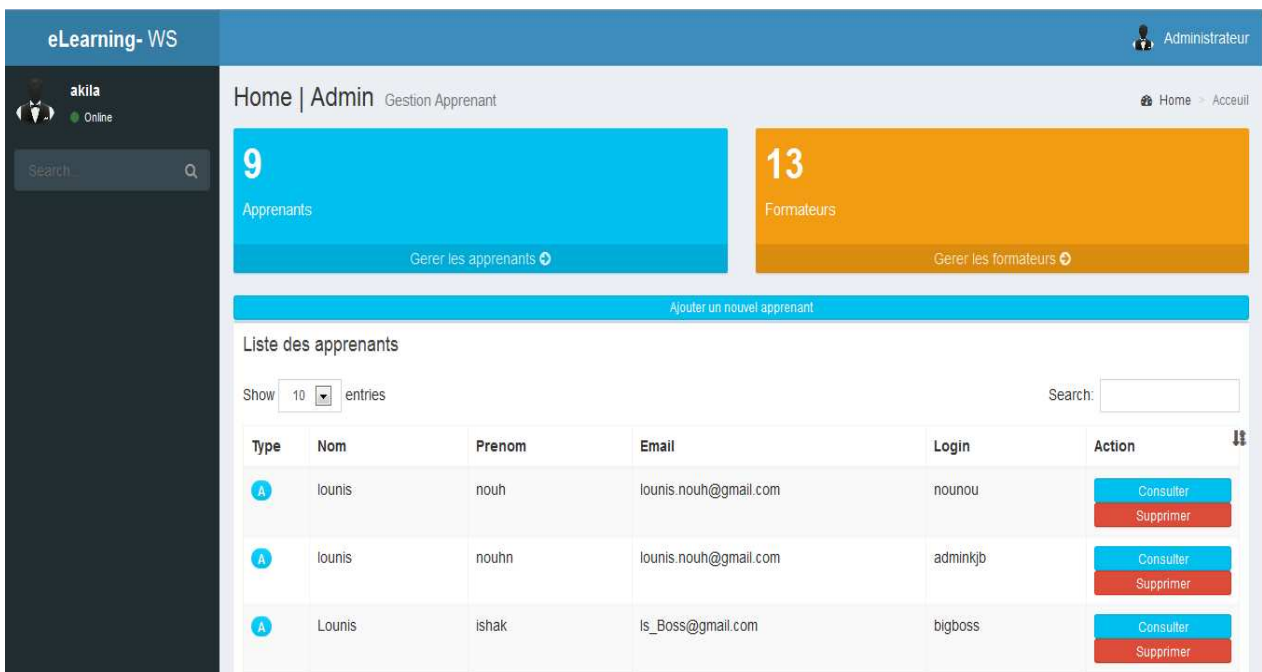
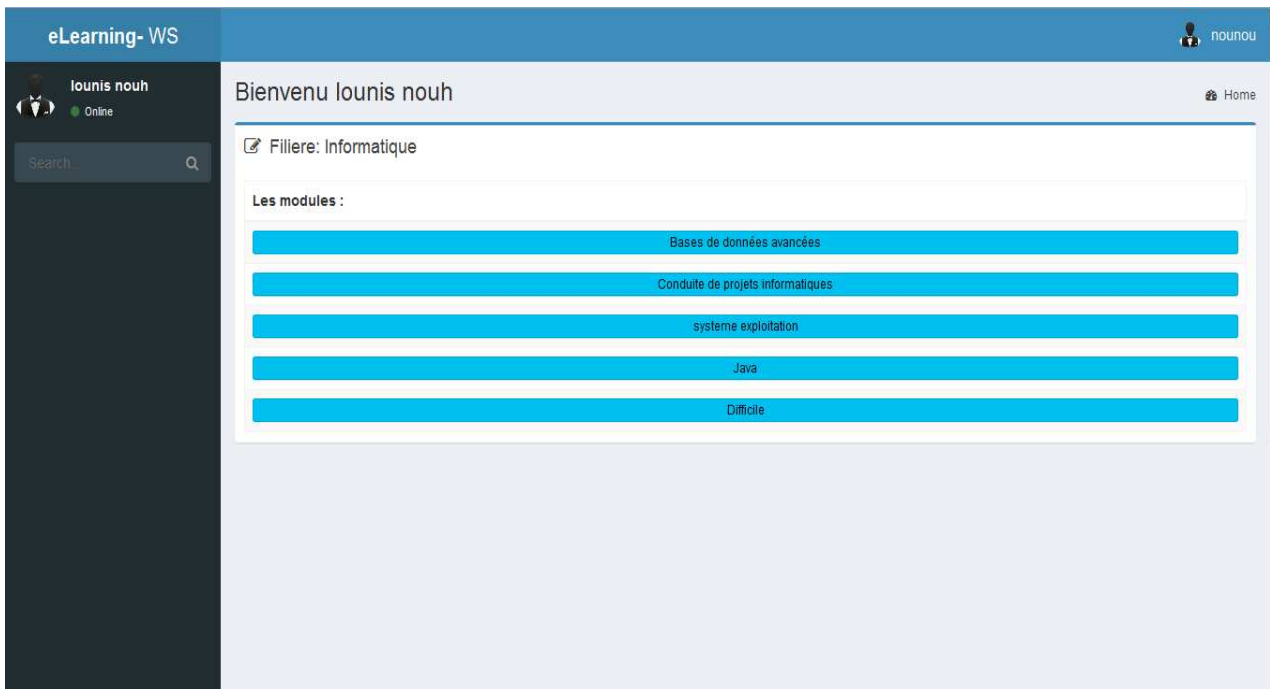


Figure 4.10: gérer des apprenants



**Figure 4.11 : modules apprenant**



**Figure 4.12 : Consulter des cours et leurs exercices.**

### **Conclusion :**

Dans ce chapitre nous avons présenté l'environnement technique dans lequel nous avons développé notre application ainsi que les outils que nous avons utilisés. Nous avons aussi décrit quelques interfaces de notre application.

L'objectif de notre travail était de concevoir et de réaliser une plate forme d'enseignement à distance et implémenté des services web dans cette plate forme, Cette plate-forme devrait implémenter des outils permettant de couvrir les besoins des différents acteurs afin de pouvoir conduire un processus d'enseignement de qualité.

Ce travail peut ce résumer en deux parties :

- Dans la première partie (Chapitre I, Chapitre II et Chapitre III), nous avons fait une étude théorique sur les WEB services et l'interopérabilité et l'état de l'art de l'enseignement à distance. Nous avons défini et suivi, ensuite, le langage de modélisation UML et son extension pour le web afin de bien concevoir notre application (Chapitre III).
- Dans la deuxième partie (Chapitre IV), nous avons concrétisé notre modèle conceptuel en réalisant une application web qui répond aux objectifs fixés, à savoir : l'implémentation des services web dans la plate forme d'enseignement à distance et la création des contenus pédagogiques.

### **Les bénéfices de notre travail :**

Le travail que nous avons accompli nous a permis :

- D'avoir les notions de bases d'une nouvelle discipline en pleine expansion si bien applicable à l'enseignement en ligne : l'E-Learning et le web service.
- De nous initier aux différentes étapes à suivre pour la conduite d'un vrai projet informatique en incluant la définition des besoins, l'étude de faisabilité en réalisant des maquettes et la transformation des modèles de la conception en codes sources exécutables.
- D'acquérir de nouvelles connaissances sur les langages HTML, JAVAScript, le PHP, MYSQL, JAVA le langage de modélisation UML, et aussi d'utiliser et de maîtriser plusieurs logiciels tels que Macromedia Dreamweaver, Macromedia Flash et le système de gestion de bases de données MySQL PHP MyAdmin et du logiciel Wampserver et Netbeans.

# Bibliographie

---

- 1) Mr. Mokrani Idir , web service pour le e-commerce, diplôme d'ingénieur d'état en informatique option système parallèles et distribués, UMMTO tizi-ouzou 2009
- 2) Mr. Loumani Jugurtha & Mr. Tatoulte Hassen, conception et réalisation d'une plate forme de télé-td (travaux dirigés à distance), diplôme de licence LMD en informatique 2011.
- 3) Mr. Mohamed el Amine Bouabid, conception à l'exploitation des travaux pratiques en ligne : application d'une approche générique à l'enseignement de l'informatique, thèse doctorat de l'université de Toulouse (université Toulouse III Paul Sabatier) soutenue et présenté le 07 décembre 2012.
- 4) Mr. Dali Yahia Mohamed, sélection de services web à base de QoWS, thèse master en informatique spécialité (Modèles Intelligents et Décision (MID)) université Abou Bekr Belkaid Telemcen Algerie 2011
- 5) M<sup>elle</sup> .Debiane Thilelli & Mr. Aboute Amyas, conception et réalisation d'une application web cas: gestion du personnel ENIEM, thèse licence en informatique UMMTO Tizi-Ouzou 2011
- 6) Mr. Sylvain Rampacek, sémantique, interactions et langages de description des services web complexes, thèse doctorat de l'université de Reins Champagne\_Ardenne, présenté et soutenue le 10 novembre 2006.
- 7) Mr. Bemba Coulibaly, un langage et un environnement de conception et de développement de services web complexes, université Paris Dauphine laboratoire LAMSADE, présenté et soutenue le 5 juin 2009.
- 8) M<sup>elle</sup> . Cyrielle Lablanche & Mr. Florens Seine & Mr. Sébastien Gastaud, les web services, thèse de licence en informatique, université de Nice\_Sophia Antipolis 2004/2005
- 9) Doc : implémentation d'une chaîne d'édition générique pour télé Tps (Arnaud Lelevé, Hcene Ben mohamed et Patrick Prévôt) laboratoire I.C.T.T, I.N.S.A de Lyon.
- 10) Doc : conception et implémentation de scénarios pédagogiques riches avec des portails communautaires.  
<http://tecfa.unige.ch/proj/seed/catalog/docs/queret03/>
- 11) Doc : guide du développeur d'applications web JBuildreX, Borland Excellence Endures.

# Bibliographie

---

- 12) Doc : Didier Donsez, panorama sur les web services, université Joseph Fourier (Grenoble1).
- 13) Arnaud Iéleuve, Cecile Meyer, Patrick Prevot, Télé-TP premiers pas vers une modélisation, Laboratoire ICTT, INSA de Lyon.
- 14) Yves Laloum, les web services CNAM Paris.

Livre : Services web avec J2EE et .NET conception et réalisation,  
ÉDITIONS EYROLLES 61, bd Saint-Germain 75240 Paris Cedex  
05www.editions-eyrolles.com

## Bibliographie

- 1: Stéphane Borzmeyer, Les Web services : connecter des applications, 2003
  - 2: Fielding, Roy Thomas, Architectural Styles and the Design of Network-based Software Architectures, 2000
  - 3: Fielding and al, Hypertext Transfer Protocol -- HTTP/1.1, 1999
  - 5: J. Gregorio, B. de hOra, The Atom Publishing Protocol, 2007
  - 4: D. Crockford, The application/json Media Type for JavaScript Object Notation (JSON), 2006
  - 6: Leonard Richardson, Sam Ruby, RESTful Web Services, 2007
- 15)
- 16)

## Les sites utilisés :

- 1) <http://fr.wikipedia.org>.
- 2) <http://sunmicrosystem.com>.
- 3) [\[NET-10/11\] : www.netalya.com/fr/reseaux1.asp](http://www.netalya.com/fr/reseaux1.asp)  
[laurent-audibert.developpez.com/Cours-UML/](http://laurent-audibert.developpez.com/Cours-UML/)
- 4) [www.siteduzero.com](http://www.siteduzero.com)
- 5) [www.commentcamarche.net](http://www.commentcamarche.net)
- 6) [www.developpez.com](http://www.developpez.com)

**I) Langage de modélisation UML****1. Introduction**

La démarche adoptée nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse.

Pour bien représenter l'outil qu'on va concevoir, un recours à un langage bien structuré et complet s'avère indispensable. UML (Unified Modeling language), défini par l'OMG, est à présent un standard de modélisation.

**2. Définition**

UML (en anglais *Unified Modeling Language* ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toute sorte de système ne se limitant pas au domaine informatique.

La modélisation UML utilise des diagrammes qui permettent de préciser d'une manière claire la structure et le comportement d'un système.

UML comporte treize diagrammes, diagramme de classes, diagramme de composants, diagramme de structurer composite, diagramme de déploiement, diagramme d'objet, diagramme des paquetages, diagramme d'activité, diagramme d'état, diagramme de cas d'utilisation, diagramme de séquences, diagramme global d'interaction et diagramme de temps.

Nous n'allons pas serte étudier tous les diagrammes existant, mais nous allons juste spécifier les diagrammes qui vont nous permettre de voir comment interagit l'application que nous allons développer dans son contexte de fonctionnement. Pour se faire la connaissance de l'entreprise (ou bien d'un service de l'entreprise) pour laquelle l'application est prévue est nécessaire.

### Les éléments de modélisation

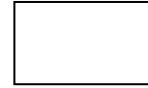
- Les objets :

Une entité d'un monde réel ou virtuel



- Les classes :

La description d'un ensemble d'objets



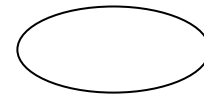
- Les états :

Une étape de la vie d'un objet



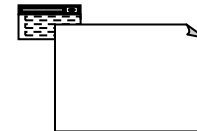
- Les cas d'utilisation :

Une manière dont un acteur utilise le système



- Les notes

Un commentaire, une explication ou une annotation.



### Les relations

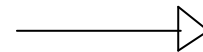
- L'association :

Une connexion sémantique entre instances.



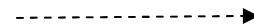
- La généralisation :

Une relation de classification.



- La dépendance :

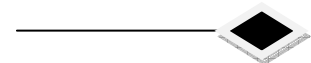
L'utilisation d'un élément par un autre.



- Composition :

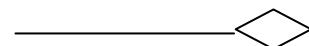
C'est une relation qui signifie que

les données de la classe cible (composante) sont des données de la classe source (composite).



- Agrégation :

Est une composition faible.



- Etat initial

L'état initial marque le point d'entrée de la première activité.

Il est représenté par un cercle plein. Il ne peut y avoir qu'un seul état initial sur un diagramme.



- Etat final :

L'état final marque la fin du déroulement de l'opération finale multiple sur un diagramme. Ils sont représentés par un cercle plein entouré d'un autre cercle.



- Barre de Synchronisation :

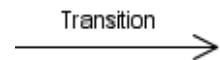
Souvent, certaines activités peuvent être faites en parallèle. Pour dédoubler le traitement "Fork", ou le reprendre quand des activités multiples ont été accomplies ("join"), des barres de synchronisation sont utilisées. Celles-ci sont modélisées par des rectangles pleins, avec des transitions multiples entrantes ou sortantes



- La transition :

Quand un état d'activité est accompli, le traitement passe à un autre état d'activité. Les transitions sont utilisées pour marquer ce passage.

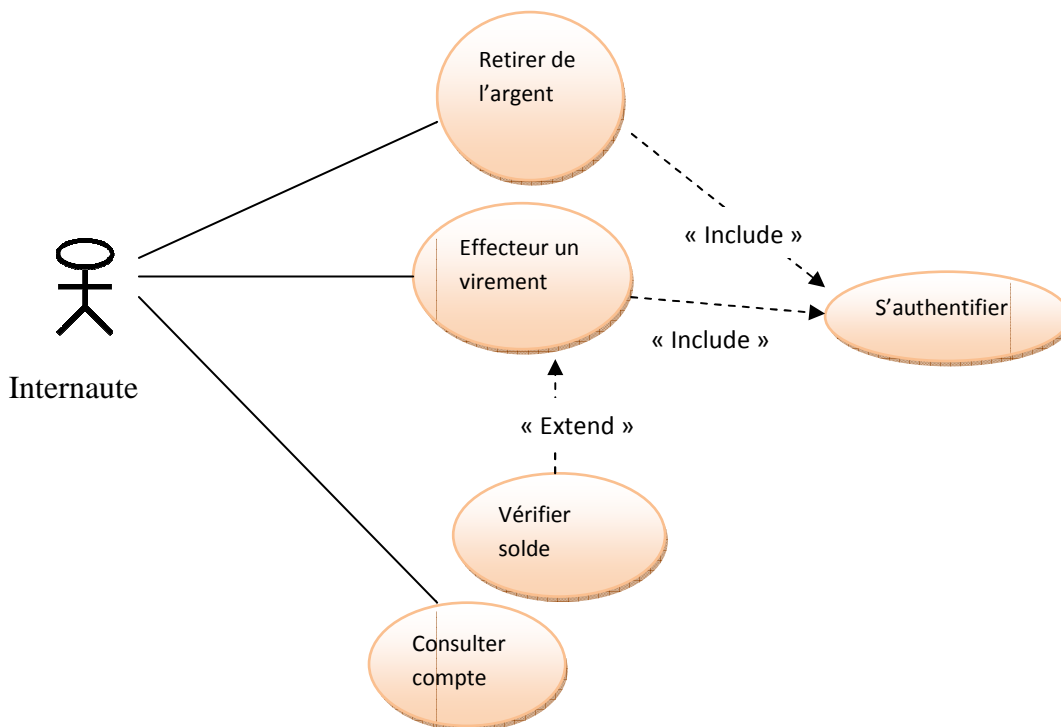
Les transitions sont modélisées par des flèches.



### 3. Les diagrammes utilisés

- **Diagramme de cas d'utilisation (use case) :** il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.

## Exemple :



## Commentaires :

▪ Acteur: entité externe qui agit sur le système (opérateur, autre système...). Un acteur peut consulter ou modifier l'état du système. En réponse à l'action de cet acteur, le système fournit un service qui correspond à son besoin.

Exemple: Internaute est un Acteur

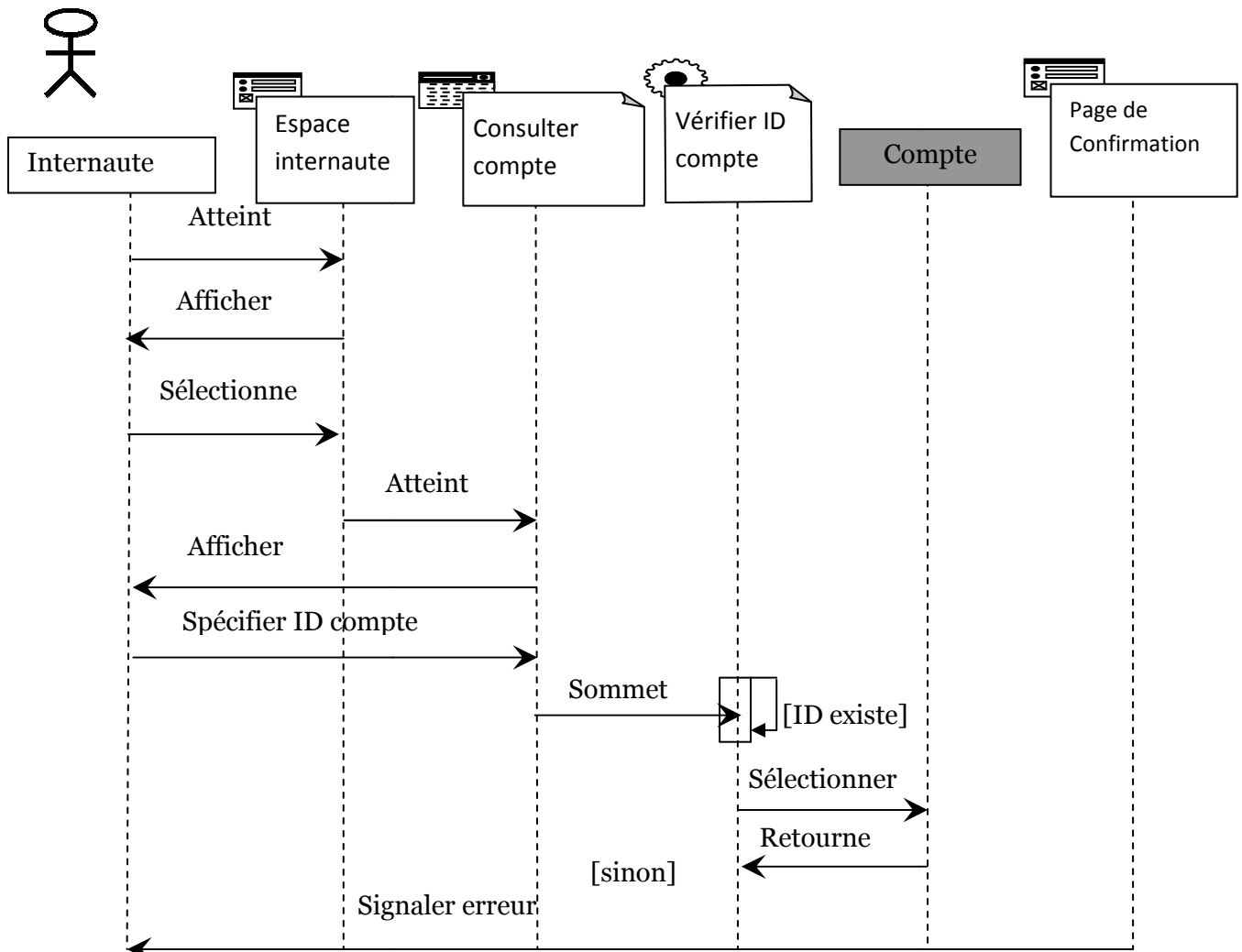
▪ La relation extend : est une relation entre 2 instances de cas d'utilisation telle que A extend B signifie que le comportement de B peut être complété par le comportement de A. La relation extend indique une possibilité, un complément possible.

▪ La relation Include : est une relation entre 2 instances de cas d'utilisation telle que la réalisation de l'un nécessite la réalisation de l'autre.

- **Diagramme de séquence :** Il représente la succession chronologique des opérations réalisées par un acteur : saisir une donnée, consulter une donnée et lancer un traitement. Le diagramme de séquence indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre. Dans un diagramme de séquence, les objets sont associés à une ligne de vie. La dimension verticale de celle-ci représente l'écoulement du temps (du haut vers

le bas). Notons que la disposition des objets sur l'axe horizontal n'est pas importante dans ce type de diagrammes. Les messages sont représentés par des flèches et leur ordre est donné par leurs positions sur la ligne de vie. Le concept de message unifie toutes les formes de communication entre objets.

### Exemple : consulter un compte.

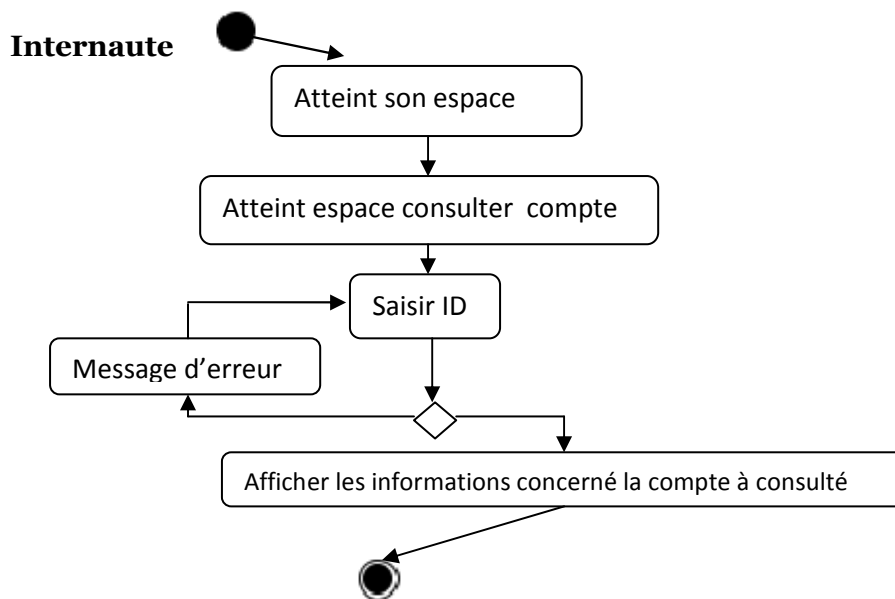


1. L'internaute atteint son espace.
2. Le système affiche son contenu.
3. L'internaute atteint le formulaire consulter compte.
4. Le système affiche son contenu.
5. Le manager sélectionne consulter compte.
6. Le système affiche son contenu.

7. Le manager spécifie la compte à consulter (en donnant son identifiant).
8. Le système spécifie si l'identifiant existe dans le cas positif le système sélectionne la compte dans la base de données et la retourne au internaute sinon le système renvoie un message d'erreur.

- **Diagramme d'activité:** permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

**Exemple :**



- **Diagramme de classe:** Le diagramme de classe représente l'architecture conceptuelle du système. Il décrit les classes que le système utilise ainsi que leurs liens, que ceux-ci représentent un emboîtement conceptuel (héritage) ou une relation organique (agrégation).

**Exemple :**

