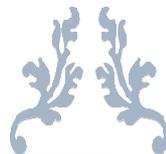


*REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE*

*UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU  
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE*

*DEPARTEMENT INFORMATIQUE*

*Mémoire  
De fin d'études*



En vue d'obtention de diplôme de Master 2 en informatique

*Thème : Conception et réalisation d'un web service pour une  
solution d'interopérabilité des applications e-learning*

Cas : E-Learning



REALISE PAR :

**BOUNDAOUI MERIEM**

**CHERIFI CELIA**

DERIGE PAR :

**M<sup>R</sup> KERBICHE**

**2014 /2015**

*Chapitre I : Services web et interopérabilités*

Figure 1.1 : Architecture des références des web.....	7
Figure 1.2 : Format d'un message SOAP.....	12
Figure 1.3 : Structure d'un message SOAP.....	12
Figure 1.4 : Exemple d'affectation de variable.....	15
Figure 1.5 : Exemple d'énumération.....	16
Figure 1.6 : Exemple d'un message SOAP.....	17
Figure 1.7 : Exemple d'une réponse serveur.....	18
Figure 1.8 : Structure générale d'un fichier WSDL.....	19
Figure 1.9 : Exemple du fichier WSDL du Web Service de Google.....	19
Figure 1.10 : Exemple d'un port du web Service de Google .....	20
Figure 1.11 : Exemple d'un message du web Service de Google.....	21
Figure 1.12 : Exemple d'un Port Type et Opérations du web Service de Google.....	22
Figure 1.13 : Exemple d'un binding du web Service de Google.....	23
Figure 1.14: Mécanisme d'accès aux services fournis par UDDI Register .....	24
Figure 1.15: Schéma représentant l'architecture REST .....	26
Figure 1.16 : La Requête GET .....	28
Figure 1.17: La Requête POST.....	28
Figure 1.18: La Requête PUT.....	28
Figure 1.18 : La Requête DELETE .....	29

---

*Chapitre 3: Analyse et conception*

---

Figure 3.1 : Diagramme de contexte.....	51
Figure3.2 : Diagramme des cas d'utilisation.....	52
Figure 3.3 : Diagramme des cas d'utilisation de l'acteur administrateur.....	53
Figure 3.4 : Diagramme de cas d'utilisation de l'acteur formateur .....	54
Figure 3.5 : Diagramme de cas d'utilisation de l'acteur apprenant .....	55
Figure 3.6 : Diagramme de séquence de cas d'utilisation <<inscription formateur>> .....	68
Figure 3.7 : Diagramme de séquence de cas d'utilisation <<authentification apprenant>> ..	69
Figure3.8 : Diagramme de séquence de cas d'utilisation <<accepter un formateur>>.....	70
Figure 3.9 : Diagramme de séquence de cas d'utilisation <<ajouter cours>> .....	71
Figure 3.10 : Diagramme de séquence de cas d'utilisation <<modifier PV des notes>>.....	72
Figure 3.11 : Diagramme de séquence de cas d'utilisation <<télécharger cours>>...	73
Figure 3.12: Diagramme de séquence de cas d'utilisation <<faire test>> .....	74
Figure3.13: Diagramme de classes globales.....	76
Figure 3.14 : Architecture générale du système.....	78
Figure 3.15 : Diagramme de déploiement de l'application.....	81

---

*Chapitre 4: Réalisation*

---

Figure 4.1 : Home page .....	94
Figure 4.2 : Page ajout .....	95
Figure4.3 : Page Profil apprenant .....	96
Figure4.4 : Page Cours.....	96
Figure 4.5 : Page home admin .....	97
Figure 4.6 : Gestion des apprenants .....	98

---

*Chapitre 1 : Services web et interopérabilités*

---

Tableau 1.1 : les types simples.....15

---

*Chapitre 2 : E-learning*

---

Tableau 2.1 : Coût comparé .....39

---

*Chapitre 4 : Réalisation*

---

Tableau 4.1 : Les règles de passage de l'objet au relationnel .....84

Tableau 4.2 : Table « compte » .....85

Tableau 4.3 : Table «administrateur» .....85

Tableau 4.4: Table «formateur» ..... 86

Tableau 4.5: Table «apprenant» .....86

Tableau 4.6: Table «Module» .....86

Tableau 4.7: Table «Cours» ..... 87

Tableau 4.8: Table «exercice» .....87

Tableau 4.9: Table «test» .....87

Tableau 4.10: Table «PV\_notes» .....88

Introduction générale .....	1
-----------------------------	---

---

## *Chapitre I : Services web et interopérabilité*

---

1.1. Introduction.....	3
1.2. Historique.....	4
1.3. Définitions.....	5
1.4. Motivations.....	5
1.5. Architecture.....	6
1.6. Technologie des web services .....	7
1.6.1. Le langage XML ( <i>eXtensible Markup Language</i> ) .....	8
1.6.1.1. Définition .....	8
1.6.1.2. XML et les services Web.....	8
1.6.2. Le protocole SOAP ( <i>Simple Object Access Protocol</i> ) .....	9
1.6.2.1. Définition .....	9
1.6.2.2. Principes de SOAP.....	9
1.6.2.3. Modèle d'échange de message SOAP.....	10
1.6.2.4. Relation avec le XML.....	10
1.6.2.5. Format d'un message SOAP.....	10
1.6.2.7. Modèles de données.....	15
1.6.2.8. Protocoles utilisés par SOAP.....	16
1.6.2.8.1. Le transport .....	16
1.6.3. Le langage WSDL ( <i>Web Service Description Language</i> ) .....	18
1.6.3.1. Présentation de WSDL .....	18
1.6.3.2. Description fonctionnelle et structurelle.....	18

a) Services.....	20
b) Port .....	20
c) Message .....	20
d) Port Type et Opérations .....	21
e) Bindings.....	22
1.6.4. Le protocole UDDI (Universal Description Discovery and Integration).....	23
1.6.4.1. Définition.....	23
1.6.4.2. Données du registre UDDI .....	24
1.6.5. Services Web REST .....	25
1.6.5.1. Présentation .....	25
1.6.5.2. Architecture .....	25
1.7. Objectifs des services web.....	29
1.7.1. Le couplage faible.....	29
1.7.2. La réutilisation.....	29
1.7.3. La découverte et la composition automatique.....	30
1.7.4. L'interopérabilité.....	30
1.8. L'interopérabilité.....	30
1.9. Les enjeux de l'interopérabilité .....	32
1. 1.10.	
Conclusion.....	32

---

## *Chapitre 2: E-learning*

---

2.1. Introduction .....	34
2.2. A qui s'adresse le e-l Définition.....	34
2.3. Définition .....	34
2.4. Principe.....	35
2.5. Le mode de diffusion de E-learning .....	35

2.6.	Les différentes formules du e-Learning.....	37
2.7.	Les avantages et les inconvénients du E-Learning.....	38
2.7.1.	Les avantages.....	38
2.7.2.	les inconvénients.....	40
2.8.	Le risque du E-Learning.....	40
2.9.	Les conditions de succès du E-Learning.....	41
2.10.	enjeux du e-Learning.....	42
2.11.	Différents aspects du e-Learning.....	43
2.12.	Les standards et les normes du e-Learning.....	43
2.12.1.	Intérêt de la normalisation pour la formation en ligne.....	43
2.12.2.	Les normes en e-Learning .....	44
2.12.3.	Pourquoi utiliser les standards ? .....	44
2.12.4.	Quelques standards en e-Learning .....	45
2.13.	Conclusion .....	47

---

### *Chapitre 3: Analyse et conception*

---

3.1.	Introduction.....	31
3.2.	Analyse.....	31
3.1.1.	Définition du besoin.....	31
3.2.2.	Identification des acteurs et des besoins .....	31
3.3.	Conception.....	32
3.3.1.	Diagramme de contexte .....	32
3.3.2.	Structuration des cas d'utilisation.....	33
3.3.3.	Description textuelle des cas d'utilisation .....	37
3.3.4.	Description graphique des cas d'utilisation.....	49
3.4.	Architecture du système .....	60

3.5. Description des différents services web .....	61
3.5.1. Le service web : ApprenantWS .....	61
3.5.2. Le service web : FormateurWS .....	61
3.5.3. Le service web : AdminWS .....	62
3.5.4. Le service web : NotesWS.....	62
3.5.5. Le service web : CoursWS.....	62
3.5.6. Le service web : ExercicesWS.....	62
3.5.7. Le service web : TestsWS .....	62
3.6. Le diagramme de déploiement .....	63
3.7. Conclusion .....	64

---

## *Chapitre 4: Réalisation*

---

4.1. Introduction.....	84
4.2. Les règles de passage de l'objet au relationnel.....	85
4.3. L'environnement technique de développement.....	88
4.3.1. Outils de développement.....	88
4.3.1.1. Le serveur web apache.....	88
4.3.1.2. Le module Tomcat.....	88
4.3.1.3. Le serveur de bases de données .....	89
4.3.1.4. L'IDE Netbeans .....	90
4.3.1.5. SoapUI.....	90
4.3.1.6. HibernateFramework .....	91
4.3.2. Les langages utilisés.....	91
4.3.2.1. J2EE.....	91
4.3.2.2. SQL.....	91

---

4.3.2.3. XHTML (Extensible HyperText Markup Language) .....	92
4.3.2.4. HTML.....	92
4.3.2.5. HTML5 (HyperText Markup Language 5) .....	92
4.3.2.6. Les feuilles de style (CSS) .....	93
4.3.2.7. JavaScript .....	93
4.4. Présentation des interfaces.....	94
4.5. Conclusion.....	99

Il existe aujourd'hui un certain nombre de plates-formes pour créer des applications. Chacune d'entre elles utilise généralement ses propres protocoles pour l'intégration des applications. En conséquence, les applications fonctionnant sur des plates-formes différentes n'ont qu'une faible capacité de partager leurs données. La prise de conscience de ces limites a entraîné un grand effort de standardisation des formats de données et d'échange de données. En effet, les regards se tournent de plus en plus vers un nouveau paradigme informatique : une intégration transparente des services Web qui dépasse les barrières logicielles et matérielles traditionnelles.

Au cœur de cette vision se trouve le concept d'interopérabilité, c'est-à-dire la capacité pour des systèmes disparates de communiquer et de partager des données de façon transparente. C'est l'objectif des services Web. Un service Web est une logique d'application programmable accessible à l'aide des protocoles Internet standard, que l'on peut aussi décrire comme l'implémentation de standards Web pour une communication transparente entre les machines et entre les applications.

Un certain nombre de technologies des services Web, telles que le protocole SOAP (*Simple Object Access Protocol*), le langage WSDL (*Web Service Description Language*) et le protocole HTTP (*HyperText Transfer Protocol*), sont actuellement utilisées pour transférer les messages entre les machines. La complexité de ces messages est très variable, pouvant aller de l'appel de méthode à la soumission d'un bon de commande. Une fonction courante - de niveau plus élevé - d'un service Web consiste à implémenter la communication de type RPC (*Remote Procedure Call*, appel de procédure à distance permettant à un programme sur un ordinateur d'exécuter un programme sur un autre ordinateur). Et les applications e-learning En particulier souffrent de ce grand problème.

Pour résoudre ce problème d'interopérabilité nous avons deux choix :

- ✓ Les services web de type REST
- ✓ Les services web de type SOAP

Vue les performances et la sécurité avancés dans les services web de type SOAP nous allons utiliser ce dernier type.

Pour aboutir à ce travail nous avons opté pour le plan suivant :

- ✓ Le premier chapitre intitulé « Services web et interopérabilité » explique en détails les web services, leur émergence, apport, architecture et les technologies qui les composent en plus du principe de l'interopérabilité.
- ✓ Le second chapitre intitulé « e-Learning », quant à lui, il est consacré à la présentation de la nouvelle méthode de formation à distance.
- ✓ Dans le troisième chapitre, nous abordons la démarche suivie dans la conception et l'analyse de notre application.
- ✓ Enfin, le quatrième chapitre décrit l'environnement de développement et l'implémentation de notre application.

L'objectif de ce chapitre est de présenter les principaux concepts des services web en commençant par une définition générale des services web, en précisant leurs architectures et objectifs.

Dans ce chapitre, une attention particulière est réservée au principe de l'interopérabilité en exposant sa définition et ses propriétés ainsi que les enjeux de l'interopérabilité et enfin quelques techniques utilisés pour sa réalisation.

## **1.1. Introduction**

L'amélioration des moyens de communication a depuis toujours été l'une des préoccupations majeures de l'humanité et avec l'accroissement de la complexité des systèmes, il devient également de plus en plus difficile de permettre la communication entre machines, c'est principalement pour répondre à cette problématique que sont nées les premières technologies des services web.

## **1.2. Historique [réf 1]**

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 des premières 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 compilé en une seule et unique entité, ce qui caractérisait les applications monolithiques, mais scindé en plusieurs composants compilés 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 réutiliser.

Au fil des années, trois modèles de programmation distribuée 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 confinés 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-feu. 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.

### 1.3. Définition [réf 1]

Les services Web représentent actuellement l'ensemble des standards les plus connus pour la réalisation des applications Internet, tels que le XML et le protocole HTTP.

Les Web services ont été proposés initialement par IBM et Microsoft, puis en partie standardisés par le consortium du World Wide Web : le W3C.

**Selon W3C :** « Un service Web est un système conçu pour permettre l'interopérabilité des applications à travers un réseau. Il est caractérisé par un format de description interprétable/compréhensible automatiquement par la machine « WSDL ». D'autres systèmes peuvent interagir avec les services Web selon la manière prescrite dans sa description et en utilisant des messages SOAP, généralement transmis via le protocole HTTP et sérialisés en XML et en d'autres standards du Web ».

### 1.4. Motivations

Un Web service est un mécanisme qui tend à donner plus d'interactions pour permettre à deux entités hétérogènes (entreprises, clients, applications, etc. ...) de dialoguer au travers du réseau Internet. Les logiciels écrits dans divers langages de programmation (C#, Visual Basic, Java, etc.), sur diverses plateformes (Linux, Windows, etc. ...) et avec diverses architectures peuvent employer des services Web pour échanger des données à travers des réseaux informatique. Chaque Web service doit pouvoir être découvert et invoqué dynamiquement par les applications.

Les aspects purement pratiques n'ont eu rien de fondamentalement novateurs. Au contraire, l'architecture des services Web s'est imposée (tout comme le langage XML) grâce à sa simplicité, à sa lisibilité et à ses fondations normalisées. L'objectif principal des services Web est de faciliter le plus possible l'accès aux applications entre entités et ainsi de simplifier les

échanges de données. Cette interopérabilité est due à l'utilisation de normes ouvertes. L'OSI et le W3C sont les comités de coordination responsables de l'architecture et de standardisation des services Web.

### 1.5. Architectures [réf 2]

L'architecture des web services est une instance de l'architecture de SOA ou AOS (Architecture Orientée service).

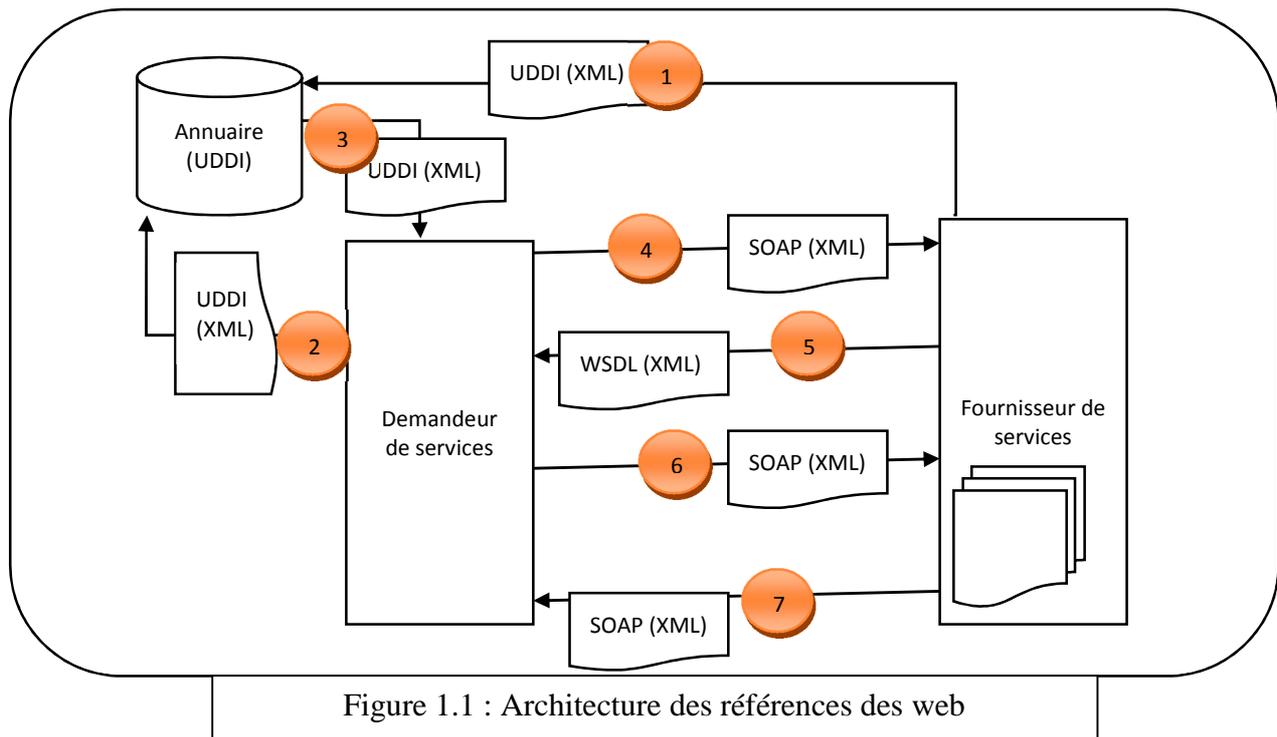
SOA est un modèle qui accepte de gérer l'hétérogénéité des applications en permettant à des applications ou services de communiquer et de travailler ensemble, quelques soit leurs plateformes ou localisations.

Ces principaux composants sont les messages échangés, les agents (entités concrètes qui implémentent un service (abstrait) et communiquent par l'échange des messages) qui agissent comme des demandeurs ou des fournisseurs de services et les mécanismes de transport qui permettent le flux de messages.

L'architecture des services web est une architecture qui définit les éléments globaux qui assurent l'interopérabilité des services web. Son but est de décrire un ensemble minimum de caractéristiques qui doivent être communes à tous les services web.

L'architecture des références des services web s'articule autour des trois rôles suivants :

- **Le fournisseur de service** : correspond au propriétaire du service. D'un point de vue technique, il est constitué par la plateforme d'accueil du service.
- **Le client** : correspond au demandeur du service. D'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer un service. L'application cliente peut être elle-même un service web.
- **L'annuaire des services** : correspond à un registre de descriptions de services offrant des facilités de publication de services à l'intention des clients



La figure 1.1 présente l'utilisation des technologies des web services par les acteurs fournisseur et demandeur :

1. Le fournisseur de service publie ses web services sur l'annuaire en utilisant UDDI
2. Le demandeur recherche un web services avec les caractéristiques voulus
3. L'annuaire trouve un service recherché et envoie les informations sur les fournisseurs du service et sur le service
4. le demandeur demande le contrat du service du fournisseur
5. Le fournisseur envoie le contrat du service (WSDL)
6. Le demandeur de service appelle le web services selon son contrat (appel en SOAP)
7. Le web services retourne le résultat de l'appel(SOAP)

## 1.6. Technologie des web services [réf 7]

L'originalité de l'infrastructure des services Web consiste à les mettre en place en se basant exclusivement sur les protocoles d'Internet tels que HTTP et les formats standards d'échange tels que XML. L'infrastructure des services Web s'est concrétisée autour de trois spécifications considérées comme des standards, à savoir SOAP, UDDI et WSDL.

## 1.6.1. Le langage XML (*eXtensibleMarkupLanguage*)

### 1.6.1.1. Définition

XML est un langage de balisage (*markup*) et un métalangage, contrairement à HTML qui possède un ensemble de balises de présentation prédéfinies, qui ne sont relatives qu'à la présentation du contenu c'est-à-dire rien ne permet à un logiciel de connaître le sens (la *sémantique*) du texte, XML va permettre d'inventer de nouvelles balises d'isolement d'informations ou d'agrégats élémentaires que peut contenir une page Web.

### 1.6.1.2. XML et les services Web

L'arrivée des services Web XML qui proposaient un modèle objet distribué a fait des systèmes d'information 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 est un langage simple, définit un standard de description de données. Les messages XML sont autodéscriptifs, 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 a 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 orienté 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 plateformes J2EE de Sun Microsystems et .NET de Microsoft.

## 1.6.2. Le protocole SOAP (*Simple Object Access Protocol*)

### 1.6.2.1. Définition

C'est un protocole de dialogue par appels de procédures à distance entre objets logiciels. Sa syntaxe d'utilisation est fondée sur XML et ses commandes sont envoyées sur Internet par l'intermédiaire du protocole HTTP mais aussi SMTP et POP sous forme de texte structuré.

Il permet aux systèmes objets distribués de solliciter et d'obtenir des services rendus par d'autres objets, il est moins lourd à mettre en œuvre que d'autres protocoles et c'est pour cela qu'il est de plus en plus adopté. Le protocole SOAP est une note du Consortium W3C dont Microsoft fait partie, mais qui n'est pas spécifique à Microsoft et Windows. IBM a également participé à l'élaboration de ce protocole.

### 1.6.2.2. Principes de SOAP

SOAP codifie simplement une pratique existante, à savoir l'utilisation conjointe de XML et HTTP. SOAP est un protocole minimal pour appeler des méthodes sur des serveurs, services, composants, objets.

Il a pour avantages de ne pas imposer une API ou un runtime, ni l'utilisation d'un ORB (CORBA, DCOM, ...) ou d'un serveur web particulier (Apache, IIS...) et en plus de ne pas imposer un modèle de programmation.

Une des volontés du W3C vis à vis de SOAP est de "ne pas réinventer une nouvelle technologie". SOAP a été construit pour pouvoir être aisément porté sur toutes les plateformes. La technologie SOAP peut être vu selon deux aspects :

- ❖ SOAP peut être vu comme un protocole d'échange de "message" :

- La requête contient un seul message (appel sérialisé d'une méthode sur un objet). --

La réponse contient un seul message (retour sérialisé d'un appel de méthode sur un objet).

- ❖ SOAP peut être vu comme un format d'échange de documents :

- La requête contient un document XML.
- Le serveur retourne une version transformée.

### **1.6.2.3. Modèle d'échange de message SOAP**

En dehors du protocole auquel SOAP est attaché, les messages sont routés sur un chemin appelé un 'message path', qui permet de transmettre un message à un destinataire en passant par un ou plusieurs noeuds intermédiaires.

Une application SOAP recevant un message SOAP doit traiter ce message en faisant les actions suivantes :

- Identifier toutes les parties du message SOAP ;
- Vérifier que toutes les parties mandataires sont supportées par l'application pour ce message et les traiter en conséquence. Si ce n'est pas le cas le message est supprimé ;
- Si l'application SOAP n'est pas la destination finale du message alors le message est forwardé.

### **1.6.2.4. Relation avec le XML**

Tous les messages SOAP sont encodés en XML. Une application SOAP doit inclure le namespace SOAP adéquat pour tous les éléments et les attributs définis par SOAP. Elle doit pouvoir supprimer les messages qui ont un namespace incorrect.

### **1.6.2.5. Format d'un message SOAP**

SOAP se sert le plus souvent de deux protocoles : HTTP et XML.

Un message SOAP est écrit en XML. HTTP est utilisé comme protocole de transport.

Les messages SOAP vont donc être encapsulés dans HTTP, ce qui permet une utilisation et une compatibilité très importante avec les réseaux et équipements existants. HTTP est le protocole de transport le plus utilisé mais il n'est pas impossible de trouver des implémentations de SOAP sur d'autres protocoles (avec SMTP par exemple).

Un message SOAP est un document XML qui possède une enveloppe SOAP et éventuellement une déclaration XML. L'enveloppe SOAP est composée d'un corps SOAP et éventuellement d'un en-tête.

Les règles de syntaxe sont les suivantes :

- Un message SOAP doit être codé en XML.
- Un message SOAP doit avoir une enveloppe SOAP.
- Un message SOAP peut avoir un entête SOAP (header).
- Un message SOAP doit avoir un corps SOAP (body).
- Un message SOAP doit utiliser l'espace de désignation de l'enveloppe SOAP.
- Un message SOAP doit utiliser l'espace de désignation d'encodage SOAP.
- Un message SOAP ne doit pas contenir une référence à une DTD.
- Un message SOAP ne doit pas contenir des instructions de type XML Processing.

Un message SOAP va être constitué d'une enveloppe et d'un corps. C'est à l'intérieur du corps (body) que l'on trouve le contenu :

```

//enveloppe du message
<soap:Envelope
//espace de nommage pour l'enveloppe SOAP
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
//on définit le type d'encodage du message
SOAP:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
//espace de nommage pour les types de variables
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
//espace de nommage pour l'appel de méthodes
xmlns:tns="http://soapinterop.org/">
//corps
<soap:Body>
//espace de nommage pour la fonction getStateName
<m:getStateNamexmlns:m="http://soapware.org/">
//on spécifie un argument à passer à getStatName
  <statenumxsi:type="xsd:int">41</statenum>
</m:getStateName>

//gestion d'erreurs
<soap:fault>
</soap:fault>
</soap:Body>

```

Figure 1.2 : Format d'un message SOAP

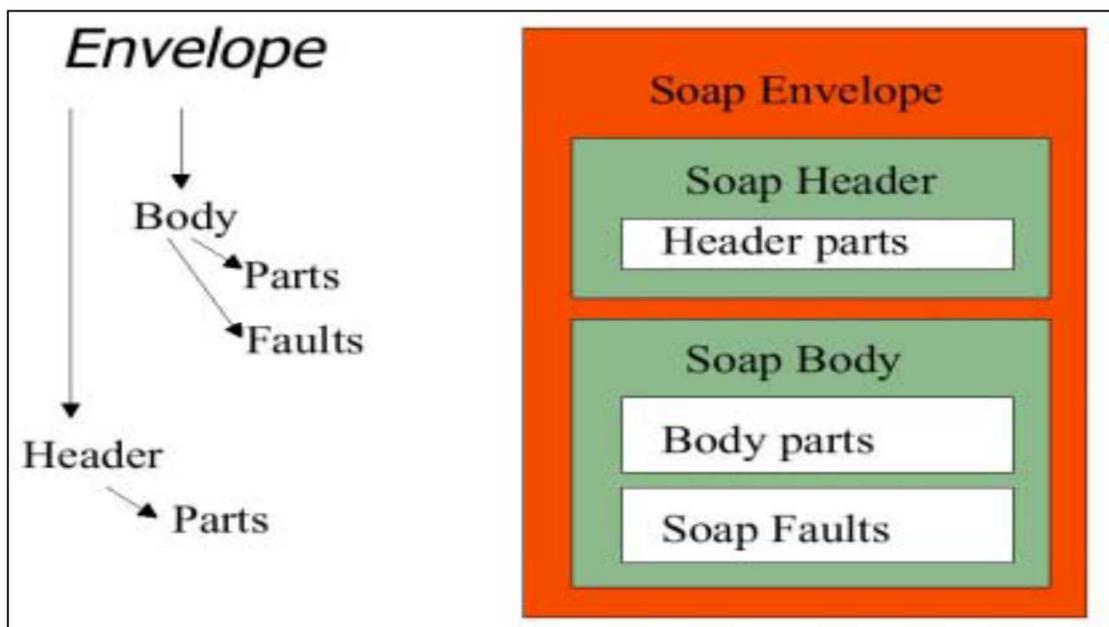


Figure 1.3 : Structure d'un message SOAP

**❖ L'enveloppe SOAP**

Un message SOAP contient :

- L'enveloppe : est l'élément de base du document XML, qui représente le message ;
- L'entête (header) : est un mécanisme générique qui permet d'ajouter des fonctionnalités à un message SOAP de manière décentralisé sans agrément entre les parties qui communiquent. C'est ici qu'il est indiqué si le message est mandataire ou optionnel ;
- Le corps (body) : Offre un mécanisme simple d'échange des informations mandataires destinées au receveur du message SOAP. Cette partie contient les paramètres fonctionnels tels que le nom de l'opération à invoquer, les paramètres d'entrée et de sortie des rapports d'erreurs.

Les règles de grammaires sont les suivantes :

**☑ Enveloppe :**

- i. Le nom de l'élément est 'envelope' ;
- ii. L'élément doit être présent dans un message SOAP ;
- iii. L'élément peut contenir une déclaration de namespace et des attributs additionnels.

**☑ En-tête (header) :**

- i. Le nom de l'élément est 'Header' ;
- ii. L'élément peut être présent dans un message SOAP. S'il est présent, il doit être le premier enfant d'un élément SOAP enveloppe ;
- iii. L'élément peut contenir une collection d'en-têtes qui doivent être ses enfants immédiats.

**☑ Corps (body) :**

- i. Le nom de l'élément est 'Body' ;
- ii. L'élément doit être présent dans un message SOAP et doit être un enfant immédiat d'une enveloppe SOAP. Il doit suivre le header s'il est présent. Sinon il doit être le fils immédiat de l'enveloppe ;
- iii. L'élément peut contenir une collection de body qui sont tous les fils immédiats d'un body.

### ❖ L'en-tête SOAP

L'élément Header est encodé en tant que premier enfant de l'enveloppe SOAP. Tous les enfants immédiats du header sont appelés des entrées du header.

Les règles d'encodage pour les entrées du header sont les suivantes :

- Une entrée d'header est identifiée par son nom d'élément, qui consiste en le namespace URI et le nom local ;
- L'attribut SOAP : 'encodingStyle' peut être utilisé pour indiquer le style de codage de l'entrée
- L'attribut SOAP : 'mustUnderstand' peut être utilisé pour indiquer comment traiter l'entrée.

### ❖ Le corps du message SOAP (Body)

L'élément body est encodé en tant que fils direct de l'élément SOAP enveloppe. Si un header est présent alors le body doit être le fils immédiat de celui-ci.

Tous les enfants immédiats de l'élément body sont appelés des entrées du corps et chaque entrée est encodée comme un élément indépendant dans l'élément body.

Les règles d'encodage des entrées du body sont les suivantes :

- Une entrée body est identifiée par son nom d'élément, qui correspond au namespace URI et au nom local ;
- L'attribut SOAP 'encodingStyle' peut être utilisé pour indiquer le style d'encodage pour l'entrée.

SOAP définit une entrée body, qui est l'entrée 'Fault' utilisée pour le rapport d'erreurs.

### Message 'Fault'

L'élément Fault de SOAP est utilisé pour acheminer des erreurs ou des informations sur le statut à travers un message SOAP. S'il est présent, l'élément Fault de SOAP doit impérativement apparaître comme une entrée du 'body' et ne doit en aucun cas apparaître plus d'une fois à l'intérieur de l'élément 'body'.

### 1.6.2.7. Modèles de données

Les éléments ainsi véhiculés peuvent contenir deux types de donnée :

➤ les types simples :

Type	Exemple
<b>Int</b>	58502
<b>Float</b>	314159265358979E+1
<b>negativeInteger</b>	-32768
<b>String</b>	Louis 'Satchmo ' Armstrong

Tableau 1.1 : les types simples

La déclaration d'un type simple s'effectue en deux temps :

1. Il est nécessaire de fixer le type ainsi que le nom de la variable que l'on souhaite manipulée.
2. Comme pour tout langage, nous pouvons ensuite affecter la variable à une valeur.

**Exemple:**

```
<element name="age" type="int"/>
<element name="height" type="float"/>
<element name="displacement" type="negativeInteger"/>
<age>45</age>
<height>5.9</height>
<displacement>-450</displacement>
```

Figure 1.4 : Exemple d'affectation de variable

➤ Les types composés :

**Exemple :** Les énumérations

Contrairement aux éléments simples, il faut trois étapes pour utiliser une énumération :

1. Il faut déclarer un nouvel élément qui possède le type que l'on s'apprête à créer
2. Ensuite, on définit les valeurs de l'énumération
3. On utilise l'énumération en affectant la valeur souhaitée.

**Exemple :**

```
<element name="EyeColor" type="tns:EyeColor"/>
<simpleType name="EyeColor" base="xsd:string">
  <enumeration value="Green"/>
  <enumeration value="Blue"/>
  <enumeration value="Brown"/>
</simpleType>
<Person>
  <Name>Henry Ford</Name>
  <Age>32</Age>
  <EyeColor>Brown</EyeColor>
</Person>
```

Figure 1.5 : Exemple d'énumération

**1.6.2.8. Protocoles utilisés par SOAP**

SOAP se sert le plus souvent de deux protocoles : HTTP et XML.

**1.6.2.8.1. Le transport****SOAP via HTTP**

HTTP constitue un excellent transport pour SOAP. 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.

L'une des caractéristiques de SOAP est de faire abstraction de la couche de transport sous-jacente. Bien que la pratique la plus répandue soit d'utiliser SOAP au-dessus de HTTP, il existe aussi des implantations de SOAP au-dessus d'autres protocoles. La manière d'implanter SOAP au-dessus d'un protocole de transport donné est appelée une liaison SOAP (« SOAP binding » en anglais).

Une liaison SOAP définit, en particulier, l'encodage des messages (nécessaire en particulier lorsque le protocole sous-jacent utilise un format binaire), la méthode pour l'échange de messages, l'encodage des noms d'opérations (appelés « SOAP Actions »), et la façon dont différents messages (y compris les messages d'erreur) appartenant à la même interaction sont corrélés. Par exemple, la liaison SOAP au-dessus de HTTP définit que les messages sont encodés dans un « type de média » appelé « application/soap+xml » (c'est-à-dire en XML avec

quelques extensions), que le nom de l'opération correspondant à une requête est donné dans un en-tête HTTP appelé « SOAPAction », et que les messages dans une interaction sont échangés au travers des méthodes POST et GET fournies par HTTP. D'autres règles définissent la manière dont les messages appartenant à un même échange (y compris les messages d'erreur) sont corrélés en exploitant les caractéristiques des méthodes POST et GET.

**Exemple:** Message SOAP encapsulé dans une requête HTTP

Voici une requête SOAP typique (avec les en-têtes HTTP) pour un appel à une méthode RPC nommée EchoString, qui prend une chaîne comme paramètre

```
POST /test/simple.asmx HTTP/1.1
Host: 131.107.72.13
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://soapinterop.org/echoString"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://soapinterop.org/"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <tns:echoString>
      <inputString>string</inputString>
    </tns:echoString>
  </soap:Body>
</soap:Envelope>
```

Figure 1.6 : Exemple d'un message SOAP

Le serveur répond alors:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://soapinterop.org/"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <tns:echoStringResponse>
      <Return>string</Return>
    </tns:echoStringResponse>
  </soap:Body>
</soap:Envelope>
```

Figure 1.7 : Exemple d'une réponse serveur

### 1.6.3. Le langage WSDL (*Web Service Description Language*)

#### 1.6.3.1. Présentation de WSDL

Afin de décrire les services disponibles sur un serveur, un langage de description de service a été mis en place : le WSDL (*Web Service Description Language*). Ce langage est basé sur XML. Les fichiers WSDL contiennent donc la description de l'accès à des Web services ainsi que des messages qui seront échangés avec les services en question. (Arguments, types de donnée...), et définit les opérations exposées par ce service et le format qu'un client doit employer pour le solliciter. Une fois muni de cette description, appelée aussi parfois « contrat », le client va pouvoir dialoguer avec les services de manière adéquate.

#### 1.6.3.2. Description fonctionnelle et structurelle

Le schéma suivant nous montre les différents éléments décrits dans un fichier WSDL.

```

<definitions>
<type/>
Description des types de données structurés
<message/>
Description des messages
<portType/>
Description de l'ensemble des opérations
<binding/>
Description des protocoles de communication
<service/>
Description des ports d'accès
</definitions>

```

Figure 1.8 : Structure générale d'un fichier WSDL

Un fichier WSDL commence par `<definition>` et finit par `</definition>`. C'est à l'intérieur de cette espace que l'on va déclarer tous les éléments constituant la description.

Nous allons prendre l'exemple du Web Service de Google pour montrer comment se construit un fichier WSDL.

Début de la description des services avec déclaration des différents schémas utilisés

Nom unique identifiant cet ensemble de service au niveau du serveur

```

<definitions name="urn:GoogleSearch"
  targetNamespace="urn:GoogleSearch"
  xmlns:typens="urn:GoogleSearch"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
...
</definition>

```

Figure 1.9 : Exemple du fichier WSDL du Web Service de Google

**a) Services**

Un service est la mise à disposition d'une ou plusieurs méthodes. On peut imaginer par exemple une classe avec plusieurs méthodes invocables à distance. La classe sera le service, et chaque méthode sera une opération sur ce service.

La définition d'un service se fait par l'utilisation de `<service></service>`.

**b) Port**

Pour chaque service on va définir des ports par lesquels ce service sera disponible. En effet, il est possible de rendre disponible un service sur plusieurs supports différents : HTTP GET, SOAP, SMTP...

**c) Message**

Les messages sont les éléments échangés entre le client et le serveur lors d'une opération sur le port d'un service. Ces messages sont constitués de plusieurs parties (part) qui représentent chacune une donnée avec un type associé.



#### d) Port Type et Opérations

Une méthode peut être représentée par une opération qui prend un message en entrée et renvoie un message en sortie. Ainsi chaque opération représentée par `<operation></operation>` indique les flux de messages en entrée et en sortie correspondants en définissant les éléments `<input/>` et `<output/>`. Enfin, la collection de toutes les opérations d'un service est rassemblée dans un port Type.

Définition d'une opération  
(méthode)

```
<portType name="GoogleSearchPort">

  <operation name="doGetCachedPage">
    <input message="typens:doGetCachedPage"/>
    <output message="typens:doGetCachedPageResponse"/>
  </operation>

  <operation name="doSpellingSuggestion">
    <input message="typens:doSpellingSuggestion"/>
    <output message="typens:doSpellingSuggestionResponse"/>
  </operation>

  <operation name="doGoogleSearch">
    <input message="typens:doGoogleSearch"/>
    <output message="typens:doGoogleSearchResponse"/>
  </operation>

</portType>
```

Figure 1.12 : Exemple d'un Port Type et Opérations du web Service

### e) Bindings

Enfin, pour compléter la description, il faut relier certains éléments entre eux. C'est le rôle des *bindings* représentés par les balises `<binding></binding>`. On va spécifier notamment tout ce qui concerne l'encodage des données dans les messages en indiquant les règles que l'on utilise.

Définition de la couche transport utilisée

Pour chaque opération, on définit les règles d'encodage des données

```
<binding name="GoogleSearchBinding" type="typens:GoogleSearchPort">
<soap:binding style="rpc"
  transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="doGoogleSearch">
<soap:operation soapAction="urn:GoogleSearchAction" />
<input>
  <soap:body use="encoded"
    namespace="urn:GoogleSearch"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</input>
<output>
  <soap:body use="encoded"
    namespace="urn:GoogleSearch"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</output>
</operation>
</binding>
```

Figure 1.13 : Exemple d'un binding du web Service de Google

Dans la pratique, les développeurs utilisent des outils de développement générant le document WSDL plutôt de rédiger à la main une description de service, ceci parce que dans la plupart des cas, un service Web publie plusieurs méthodes et plus le nombre de méthodes est important, plus la description WSDL devient importante, donc la rédiger manuellement sera une opération très complexe.

## 1.6.4. Le protocole UDDI (Universal Description Discovery and Integration)

### 1.6.4.1. Définition

UDDI a été conçu en 2000 à l'initiative d'un ensemble d'industriels (Ariba, IBM, Microsoft), en vue de devenir le registre standard de la technologie des services Web. Pour convenir à la technologie des services Web, les services référencés dans UDDI sont accessibles par l'intermédiaire du protocole de communication SOAP, et la publication des informations concernant les fournisseurs et les services doit être spécifiée en XML afin que la recherche et l'utilisation soient faites de manière dynamique et automatique. UDDI constitue un méta-service possédant des fonctions de publication et de recherche en fournissant comment interagir avec un service et permet une immatriculation mondiale. Il peut être implanté au niveau mondial, sur Internet, ou au niveau d'une entreprise, sur intranet.

L'annuaire UDDI propose un cadre technique indépendant des plates-formes et ouvert pour permettre aux entreprises d'enregistrer et de découvrir leurs services Web.

UDDI repose sur le protocole SOAP, et de ce fait, les requêtes et réponses sur l'annuaire sont des messages SOAP.

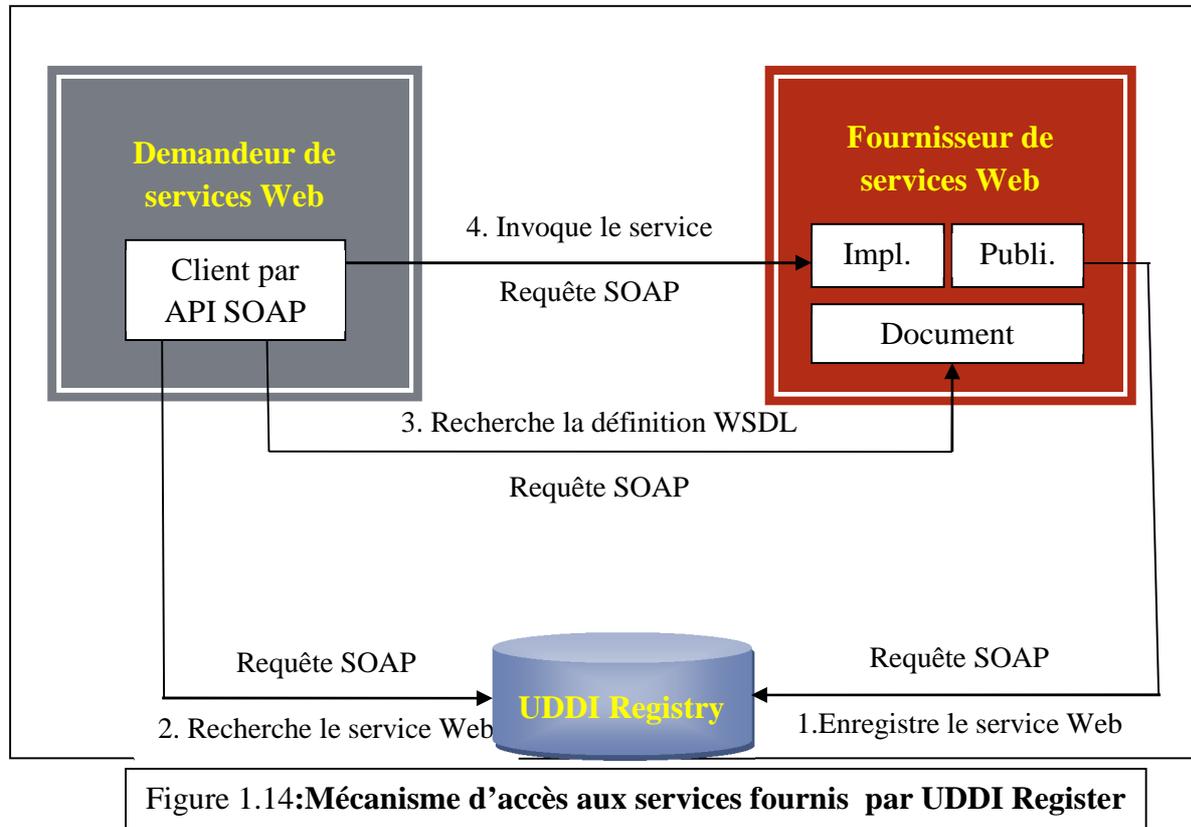


Figure 1.14: Mécanisme d'accès aux services fournis par UDDI Register

#### 1.6.4.2. Données du registre UDDI

La recherche et la sélection dans UDDI reposent sur la publication préalablement décrite du service et de son fournisseur. Le futur client peut connaître par l'intermédiaire de l'UDDI : les fournisseurs de service, les services proposés par un fournisseur donné, les moyens d'invoquer un service. Pour apporter aux clients la réponse à ces questions, UDDI organise l'ensemble des informations qu'il contient en trois catégories, spécifiées en XML. Chacune d'elles peut être utilisée pour faire une recherche via UDDI.

- **Les pages blanches** : ce composant permet de connaître des informations à propos du fournisseur proposant le service. Cette description contient toutes les informations jugées pertinentes pour identifier le fournisseur (organisation) telles que : son nom, son adresse physique, numéros de téléphone, fax, sites Web, etc. Le futur client du service retrouve dans les pages blanches les informations que le fournisseur a renseignées dans l'élément *Business Entity* lors de la publication.

- **Les pages jaunes** : les pages jaunes d'UDDI contiennent, au format WSDL, la description (non technique) des services renseignée par les fournisseurs eux mêmes. Les services proposés sont répertoriés en catégories.
- **Les pages vertes** : elles comportent les informations techniques liées aux services Web et basée sur leur description WSDL (Processus marketing, descriptions des services rendus, programmation, Plateforme, Implémentation..).

## 1.6.5. Services Web REST

### 1.6.5.1. Présentation

REST est l'Acronyme de **RE**presentational**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 à SOAP ou HTTP mais un style d'architecture inspiré de l'architecture du web basé sur le protocole http.

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 utilisé par les sites communautaires (réseaux sociaux) leur permettant de proposer à leurs clients une API facile à utiliser. Des sites comme Facebook, Twitter proposent ainsi de telles API évitant à leurs clients de devoir passer par la case SOAP.

Les Web Services RESTful sont basés 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 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.

### 1.6.5.2. 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 unitairement 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.

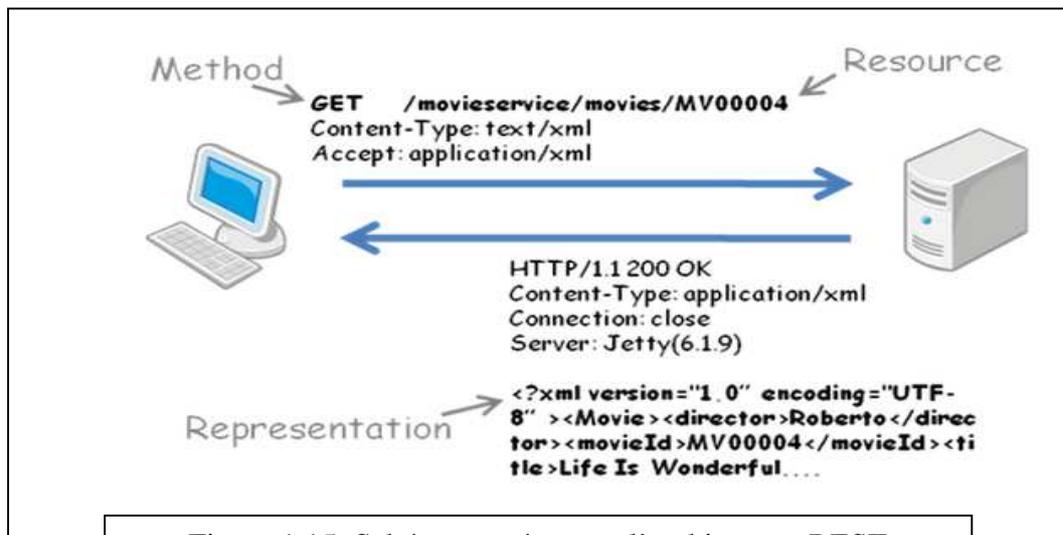


Figure 1.15: Schéma représentant l'architecture REST

- **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 média (xml,json,html) peut avoir un impact direct sur la perception de performance d'un système réparti d'hypermédia que peut avoir l'utilisateur. Un format de données qui place l'information de la plus importante à la moins importante, 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.

### ❖ 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 requêtes.

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 à partir du serveur, la récupération d'une page HTML, ou la soumission d'un formulaire. En fait, HTTP permet d'effectuer huit types 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

**Méthode GET:** La méthode GET renvoie une représentation de la ressource telle qu'elle est sur le système.



Figure 1.16 : La Requête GET

**Méthode POST** : La méthode POST crée une nouvelle ressource sur le système

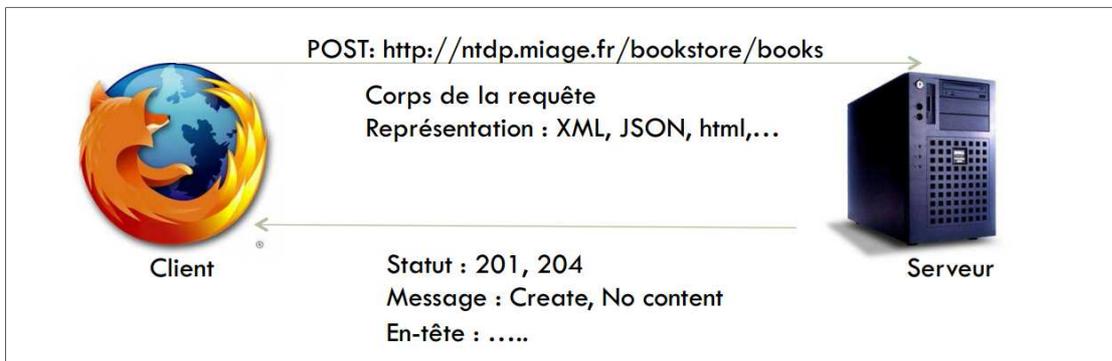


Figure 1.17: La Requête POST

**Méthode PUT** : Mise à jour de la ressource sur le système



Figure 1.18: La Requête PUT

**Méthode DELETE:** Supprime la ressource identifiée par l'URI sur le serveur.



Figure 1.18 : La Requête DELETE

## ❖ JAX\_RS

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

### 1.7. Objectifs des services web

Quatre objectifs fondamentaux sont visés par L'approche de service Web:

#### 1.7.1. Le couplage faible

Le couplage est une métrique indiquant le niveau d'interaction entre deux ou plusieurs composants logiciels. Nous parlons de couplage fort si les composants échangent beaucoup d'information et de couplage faible dans le cas contraire. Vu que la communication avec les services Web est réalisée via des messages décrits par le standard XML caractérisé par sa généricité et son haut niveau d'abstraction, les services Web permettent la coopération d'applications tout en garantissant un faible taux de couplage. Par conséquent, il est possible de modifier un service sans briser sa compatibilité avec les autres services composant l'application.

#### 1.7.2. La réutilisation

L'avantage de la réutilisation est qu'elle permet de réduire les coûts de développement en réutilisant des composants déjà existants. Dans le cas de l'approche service Web, l'objectif de la séparation des opérations en services autonomes est en effet pour promouvoir leur réutilisation. Ainsi, lorsqu'un client définit ses exigences, il est généralement possible de réutiliser des services déjà existants pour satisfaire une partie des exigences. Ceci facilite la maintenance de l'application et permet un gain de temps considérable.

#### 1.7.3. La découverte et la composition automatique

La découverte et la composition sont des étapes importantes qui permettent la réutilisation des services. En effet, il faudra être en mesure de trouver et de composer un service afin de pouvoir en faire usage. En exploitant les technologies offertes par Internet et en utilisant un

ensemble de standards pour la publication, la recherche et la composition, l'approche services Web tend à diminuer autant que possible l'intervention humaine en vue de permettre une découverte et une composition automatiques des services les plus complexes. En effet, pour réaliser son application, un développeur peut simplement interroger un moteur de recherche de services afin de trouver le service adéquat et à l'aide de langages de coordination appropriés il peut l'intégrer avec le reste des services de son application.

#### 1.7.4. L'interopérabilité

Elle permet à des applications écrites dans des Langages de programmation différents et s'exécutant sur des plateformes différentes de communiquer entre elles. En manipulant différents standards que ce soit XML ou les protocoles d'Internet, les services Web garantissent un haut niveau d'interopérabilité des applications et ceci indépendamment des plateformes sur lesquelles elles sont déployées et des langages de programmation dans lesquels elles sont écrites. Ainsi, en s'appuyant sur un format d'échange de messages standard et sur l'ubiquité de l'infrastructure d'Internet, l'interopérabilité est donc une caractéristique intrinsèque aux services Web.

### 1.8. Interopérabilité

#### 1.8.1. Définition [ref 9]

- **Définition linguistique de l'interopérabilité**

De point de vue linguistique, le terme interopérabilité est composé de deux parties :

- le préfixe "**Inter**" : qui ajoute au radical auquel il est relié le sens de la coopération entre plusieurs acteurs ;
- le mot "**opérabilité**" : qui est le nom tiré de l'adjectif "opérable" qui, en qualifiant un nom signifie que ce dernier peut opérer.

Le résultat de l'association de ces deux parties signifie que plusieurs acteurs ont le pouvoir d'opérer chacun sur l'autre et vice versa.

- **Définition informatique de l'interopérabilité**

L'interopérabilité des systèmes d'information en général et les systèmes d'information géographique en particulier peut être vue comme la capacité de ces systèmes à échanger des services et des données.

Elle peut être aussi définie comme la capacité que possède un produit ou un système, avec des interfaces intégralement connues, à fonctionner avec d'autres produits ou systèmes sans restrictions d'accès ou de mise en œuvre.

Un exemple parfait de systèmes interopérables est le téléphone. Toutes les interfaces sont des normes générées par l'UIT. Il est possible ainsi de téléphoner quelque soit la marque de téléphone du correspondant ou les matériels utilisés par les différents opérateurs.

Il est important de faire la distinction entre l'interopérabilité et la compatibilité. La compatibilité est une relation binaire et concerne un ensemble fini de systèmes.

- A et B sont dits compatibles si leurs constructions respectives leur permettent de communiquer et travailler ensemble ;
- A et B sont interopérables si, en représentant une ou plusieurs normes externes, ils arrivent à être compatibles.

L'interopérabilité est basée sur des conventions adoptées entre les systèmes communicants concernant les protocoles d'échange de messages, la désignation des procédures à activer ou les codes d'erreurs retournés.

L'interopérabilité possède, à la fois, une dimension statique (la compatibilité des types de données) et une dimension dynamique (la compatibilité des procédures).

Il existe plusieurs organismes publiant des normes pour l'interopérabilité des systèmes d'informations, nous citons :

- DMTF: Distributed Management Task Force
- IETF : Internet Engineering Task Force
- ISOC : Internet Society
- OASIS : Organization for the Advancement of Structured Information Standards
- W3C : World Wide Web Consortium

L'interopérabilité repose principalement sur l'inter connectivité des systèmes et des composants, qui est elle-même contrôlée par des normes et standards informatiques de communication devant être respectées.

Les standards de communication consistent en des spécifications techniques qui se traduisent par des définitions et des règles d'ingénierie, de façon à assurer que les produits, les traitements et les services remplissent bien leurs rôles.

Il est donc clair que les messages échangés entre les systèmes d'information géographique représentent le point d'intérêt de l'interopérabilité. Ces messages possèdent une syntaxe (une forme) et une sémantique (un sens) d'une manière que plusieurs formes peuvent correspondre à un seul sens. À cet effet, il existe deux types d'interopérabilité, une interopérabilité syntaxique et une interopérabilité sémantique.

- l'interopérabilité syntaxique définit la nature, le type et le format des messages échangés, et conduit à la notion de système ouvert permettant d'assurer l'hétérogénéité. cependant, cette interopérabilité est insuffisante car la cohérence de la forme des messages ne permet pas, par elle-même, d'avoir une cohérence des significations de ces messages.
- l'objectif de l'interopérabilité sémantique est d'assurer que les échanges effectués entre les systèmes interconnectés conservent leur sens, c'est-à-dire que les systèmes communicants ont une compréhension commune de la signification des données et des services échangés.

Il est possible de résumer les objectifs motivant l'interopérabilité comme suit :

- la réduction des coûts d'acquisition des données ;
- l'amélioration des processus d'aides à la décision ;
- l'ajout de nouvelles fonctionnalités aux systèmes existants ;
- la migration vers de nouveaux systèmes tout en préservant ceux déjà existants.

### **1.9. Les enjeux de l'interopérabilité**

L'interopérabilité est une notion absolument cruciale pour le réseau mondial et internet ainsi que les systèmes d'information et les services web. Par essence, des matériels divers et variés sont mis en œuvre dans ces réseaux hétérogènes aux côtés d'une panoplie encore plus vaste de matériels informatiques et de logiciels mis à la disposition des concepteurs de ces réseaux.

Elle est aussi cruciale pour l'ensemble de l'économie, car dans la quasi-totalité des domaines d'activités dans l'industrie, les services, et même l'agriculture, on emploie des systèmes informatiques, qui communiquent de nos jours d'une entreprise à l'autre par le biais de réseaux informatiques (internet, extranet, messagerie électronique).

Aussi, l'interopérabilité est une nécessité de plus en plus évidente dans le domaine du commerce électronique. En effet, le demandeur de services et le fournisseur de services sont obligés d'interopérer en utilisant des structures et des types de données communs.

### **1.10. Conclusion**

Dans ce chapitre nous avons apporté un éclairage sur les différentes solutions et technologies des services web. Nous présenterons dans le chapitre suivant une méthode d'apprentissage exploitant ces technologies qui est le e-Learning

## **2.1. Introduction**

L'enseignement à distance a connu une évolution spectaculaire, aujourd'hui tout le monde en parle. L'utilisation des TIC (Technologies de l'Information et de la Communication) a considérablement développé les organisations, les métiers, les techniques, ainsi que les moyens et les pratiques des apprenants.

Il est à noter que tout cela fait que l'EAD (Enseignement À Distance) a désormais pris le nom d'e-Learning.

## **2.2. A qui s'adresse le e-learning?**

Le e-Learning est souvent employé dans les grandes entreprises en association avec le Knowledge Management pour former les salariés plus rapidement qu'à l'ordinaire. Ainsi que dans les universités où elles proposent des formations à distance ou encore à tout particulier qui voudrait entamer un apprentissage en ligne.

## **2.3. Définition ref 12]**

Abréviation de "electronic Learning" ; que l'on peut traduire par apprentissage par des moyens électroniques. Selon la définition de la Commission Européenne le e-Learning est : « L'utilisation des nouvelles technologies multimédia et de l'internet pour améliorer la qualité de l'apprentissage en facilitant l'accès à des ressources et des services, ainsi que les échanges et la collaboration à distance ». Le e-Learning définit tout dispositif de formation qui utilise un réseau local, étendu ou internet pour diffuser, interagir ou communiquer, ce qui inclut l'enseignement à distance, en environnement distribué, l'accès à des sources par téléchargement ou en consultation sur le net.

Il peut faire intervenir du synchrone ou de l'asynchrone, des systèmes tutorés, des systèmes à base d'autoformation.

L'e-Learning résulte donc de l'association de contenus interactifs et multimédia (son, texte, animation), de supports de distribution (PC, internet, intranet, extranet), d'un ensemble d'outils logiciels qui permettent la gestion d'une formation en ligne et d'outils de création de formations interactives. L'accès aux ressources est ainsi considérablement élargi de même que les possibilités de collaboration et d'interactivité.

## 2.4. Principe [ref 12]

Le principe étant de pouvoir accéder à des cours depuis un poste distant, les lieux nécessaires au suivi d'un cursus de formation (établissements, classes, bibliothèques, université) n'existent plus physiquement, ils sont remplacés par le S.G.C (Système de Gestion des Cours ou Systèmes de Gestion des Contenus). Le SGC est le cœur du système de formation à distance, c'est lui qui fait le lien entre les apprenants, les cursus, les tuteurs, les ressources et les contenus présents dans le système.

L'apprenant, via une plate-forme, se verra attribuer un certain nombre de modules de cours, d'exercices, d'évaluations qu'il devra effectuer en tenant compte d'une planification établie. Le tuteur (ou formateur) se charge de gérer les apprenants qu'il doit suivre. Ainsi il pourra leur affecter des ressources à consulter, des cours à étudier, des évaluations afin de se rendre compte de la bonne assimilation des contenus proposés. La communication entre tous ces acteurs se fait via Internet.

Il y a quelques années encore, il était très difficile de concilier des contenus de qualité avec un confort d'utilisation maximum, en effet, les contraintes liées au bas débit faisaient que les seuls contenus réellement exploitables étaient constitués de pages au format HTML ou de document texte.

Le haut débit, accessible aujourd'hui à une très large population a permis l'exploitation de tous les formats du contenu même les plus lourds (audio et vidéo).

## 2.5. Les modes de diffusion du E-Learning

Suivre une formation en e-Learning ne signifie pas être seul face à son ordinateur sans personne pour échanger sur les concepts abordés au cours de la formation ou pour apporter un support technique ou pédagogique.

En effet, il existe de nombreuses possibilités de communication en e-Learning, différentes de celles que l'on utilise en formation présentielle (chat, forum...).

De façon simplifiée, on distingue 3 modes de diffusion du e-Learning :

- **Asynchrone** : c'est une méthode de formation en temps différé qui permet à l'apprenant d'accéder à la formation à sa guise et autant de fois qu'il le désire (contenus sous forme de textes, animations multimédias...). Le dialogue entre les formateurs et les apprenants peut se faire via e-mail, messagerie instantanée, forums de discussion, SMS et partage de documents.

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.
- **Synchrone** : c'est une méthode en temps réel basée sur la parole et l'écoute. L'apprenant est en contact avec son formateur et les autres membres de sa classe virtuelle par partage d'application ou visioconférence. Les étudiants connectés posent leurs questions et reçoivent la réponse en temps réel, quelque soit la distance à condition bien sûr que la bande passante soit à la hauteur. Le dialogue entre les formateurs et les apprenants peut se faire via : chat, webcam, micro, partage d'applications, fonctionnalités de prise en main à distance, tableau blanc, Téléphone...

Les principaux avantages de la formation synchrone :

- Les apprenants interagissent intensivement à l'écran avec les formateurs. Le langage oral et visuel est utilisé.
- Le modèle de la classe est familier.
- On peut rapidement créer du contenu prêt à diffuser.
- La dynamique de groupe s'installe plus rapidement.

• **Mixte** : c'est une méthode qui combine les éléments asynchrones et synchrones. Il peut aussi être utilisé pour homogénéiser les savoirs avant une présentation, une conférence, une intervention ou une formation.

Les principaux avantages de la formation mixte :

- Des études en milieu universitaire ont démontré que les acquis de connaissances étaient meilleurs avec la formation mixte.
- Les classes virtuelles en temps réel permettent aux apprenants d'assister aux cours à partir de n'importe où.
- Lorsque le format permet d'enregistrer le déroulement et de sauvegarder les éléments échangés pendant le cours, les apprenants peuvent revoir à leur guise les aspects qu'ils souhaitent réviser.
- En combinant l'utilisation des logiciels de formation standardisés et prêts à l'emploi à un échange en classe, on peut adapter la formation générique à la situation particulière de son entreprise. Il s'agit d'une méthode peu coûteuse de personnalisation d'un programme de formation générique.

## 2.6. Les différentes formules du e-Learning

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

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

### • Un cours magistral synchrone

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

Pour les stagiaires, il offre la possibilité de se former depuis leur lieu de travail, voir même depuis leur domicile. C'est confortable et cela permet aussi de gagner du temps. Pour l'employeur, l'avantage financier est considérable : pas de coûts liés au déplacement des stagiaires, pas de surcoûts liés au temps perdu.

### • L'assistance

Dans cette phase le travail est individuel avec assistance audio-visuelle en direct.

Le formateur reste à disposition et peut, à distance, prendre le contrôle d'un poste de travail. Il peut ainsi partager les logiciels ou les documents et, le cas échéant, converser avec les personnes en formation.

L'assistance peut également se manifester de manière asynchrone, au moyen de messageries ou de forums informatiques.

### • L'auto-formation connectée

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

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

## 2.7. Les avantages et les inconvénients du E-Learning [réf 13]

### 2.7.1. Les avantages

E-Learning présente un intérêt considérable et offre des opportunités uniques pour les personnes qui pourraient avoir un accès limité à l'éducation et à la formation. Il intègre des approches novatrices et créatives à l'instruction et fournit un accès sans précédent aux ressources et aux informations.

Certains des avantages sont résumés ci-après :

- La formation est ouverte à toute personne, quel que soit son âge, son niveau d'instruction, sa catégorie socioprofessionnelle, etc.
- L'accès aux informations, aux savoirs et aux savoirs faire sans limites de distance.
- Favorisation de la créativité et l'esprit de découverte.
- Le e-Learning permet l'accès à de nouvelles compétences qui sont plus que jamais indispensables aux exigences de la vie moderne. Chacun peut se familiariser avec les nouvelles technologies comme l'ordinateur, les systèmes multimédias et l'Internet.
- Formation sur place : pas de déplacements ce qui favorise un gain de temps, une économie en argent et des conditions optimales de formation (à la maison par exemple) sans oublier que cet avantage est très bénéfique pour les personnes handicapées.
- Un formateur peut s'adresser à un grand nombre d'apprenants tout en assurant une relation individualisée avec chacun d'eux.
- Une réduction des coûts :
  - Les frais de transport ou d'hébergement des stagiaires qui grèvent considérablement les budgets sont supprimés.
  - Les frais liés à la prestation du formateur sont moindres ou lissés.
  - Le temps d'apprentissage personnel est réduit.

Le tableau ci-dessous présente le coût comparé d'une formation en présentiel classique et d'une e-formation pour 15 personnes sur 5 jours (source CISCO System 1999) :

Postes	Formation	e-formation (en \$)
Déplacement (15 personnes)	15 000	0
Hôtel et repas (15 x 5j)	15 000	0
Taxi et autres frais	600	0
Temps de voyage (15 pers x 5h AR x 50\$/h)	7 500	0
Temps de formation (15 pers x 40h x 50\$/h)	30 000	30 000
Location de salle (5j)	500	0
Formateur à plein temps (5j à 1500 \$ par jour)	7 500	0
Tuteur à temps partiel (1h/j x 5j x 150 \$/h)	0	750
Manuel de stage (15 x 20 \$)	300	0
Réseau bande passante (15 pers. x 40h x 40\$ par sem)	0	3600
Location PC /amortissement	0	600
Total	76 400	34 950
Total par apprenant	5 093	2 330

Tableau 2.1 : Coût comparé

- Choisir les thèmes voulus, construire les cours à apprendre, définir les orientations de la formation et adaptation aux objectifs assignés et fixés aux exigences de l'apprenant (selon ses désirs, son niveau actuel, sa qualité professionnelle, l'exigence du travail, etc.).
- Le suivi de l'étudiant est personnalisé et non plus généralisé sur tout un groupe. Ceci assure une meilleure assimilation des connaissances par tous les apprenants en plus de l'adaptation de l'évolution de la formation au rythme et aux capacités de compréhension de chaque étudiant.
- Le e-Learning est basé sur des solutions multimédias interactives qui attirent l'attention de l'apprenant, stimulent ses capacités de compréhension et d'interprétation et l'incitent à se concentrer et à assimiler rapidement par le système de l'observation et du captage.
- Créer un sentiment de liberté et de confiance de l'apprenant en lui-même. L'apprenant étudie tout seul, face à son ordinateur et n'est pas observé des autres (excepté le cas de visioconférence).
  - Le grand nombre d'étudiants qui ne cesse d'augmenter représente un problème au niveau des infrastructures d'enseignement (nombre restreint d'instituts, temps de formation limité, nombre de formateurs réduit, nombre de spécialités croissant, etc.).

- La qualité et la quantité de connaissances, d'informations et de savoirs faire progressent d'une manière très rapide. Les moyens actuels de formation traditionnelle ne permettent pas d'accéder à toutes ces informations surtout qu'elles sont réparties partout dans le monde, chose qui est plus accessible à travers la formation à distance.
- Les informations recueillies à travers la formation à distance sont constamment mises à jour du fait que les nouveautés sont rapidement détectées.

### 2.7.2. Les inconvénients

- **Les problèmes techniques afférents au fonctionnement des systèmes de formation:** perturbation du réseau de communications, pannes des ordinateurs, terminaux ou serveurs, attaques des documents électroniques de cours par des virus ou des pirates, etc.
- **La mise en place de l'infrastructure technique et la création des contenus sont coûteuses :** un investissement est nécessaire au lancement d'une politique du e-Learning. Cet investissement (coût fixe et risque important) remplace le paiement à la prestation de formation (coût variable et risque faible). Par ailleurs, le marché n'est pas encore consolidé et l'offre en matière de cours spécialisés reste limitée suivant les domaines.
- **L'accès à l'outil informatique est nécessaire :** l'utilisation de l'outil informatique limite la diffusion du e-Learning auprès d'une partie des collaborateurs.
- **La diffusion de cours e-Learning nécessite des équipements multimédia :** l'équipement de l'entreprise doit permettre la diffusion des contenus des cours. Cela concerne notamment les postes utilisateurs (PC récents, logiciels installés compatibles, éventuellement réseau avec suffisamment de bande passante). Cet inconvénient tend à perdre de son importance au fil des années, étant donné l'évolution des technologies.
- **L'E-Learning limite les interactions entre les individus :** Certains mécanismes de communication ne peuvent pas être reproduits (langage du corps par exemple), alors qu'ils jouent un rôle important dans la diffusion du savoir.

## 2.8. Le risque du E-Learning

Le taux d'abandon des formations e-Learning est plus important que dans le monde traditionnel. Les individus doivent être responsabilisés et motivés car l'e-Learning demande plus d'autonomie et d'initiative que les formations classiques.

Le tutorat permet de réduire ce risque, mais il subsiste en général. Le médium de diffusion, à savoir l'outil informatique, peut également encore aujourd'hui constituer un frein à l'adoption du e-Learning.

## 2.9. Les conditions de succès du E-Learning

### • L'E-Learning ne peut pas se substituer complètement à l'humain

L'intervention d'un tuteur pendant la formation est indispensable pour répondre (par téléphone, e-mail, forum ou chat) aux questions des apprenants. Un rôle d'accompagnement des apprenants est également indispensable pour s'assurer que les individus s'astreignent à suivre la formation, en réservant du temps en dehors de leurs tâches habituelles.

### • L'infrastructure de l'entreprise doit être capable d'accueillir l'e-Learning

L'infrastructure technique de l'entreprise (serveurs, réseau, plate-forme LMS) ainsi que les postes de diffusion du e-Learning doivent être adaptés aux formations diffusées.

### • Les thèmes des cours doivent être sélectionnés judicieusement

En fonction du profil de la population à former, de sa taille et des connaissances à enseigner, l'entreprise doit comparer les avantages et inconvénients du e-Learning et du présentiel avant de choisir le mode de diffusion.

#### • Dans la mise en place :

##### - **Spécifique au média Internet**

Il ne faut pas se contenter de mettre un cours traditionnel ou un livre en ligne. Les individus n'ont pas nécessairement la patience de lire des pages ou de suivre une vidéo sur un écran. Ils doivent également interagir avec le support. Cette interactivité et la logique de séquencement du parcours de l'apprenant sont mises en place par des ingénieurs pédagogiques.

##### - **Réutilisable**

Il est souvent possible de réutiliser du matériel développé dans le cadre d'une formation pour d'autres cours (questionnaires, chapitres,...). Ceci est essentiel pour assurer un bon retour sur l'investissement.

##### - **Opérationnel**

Les modules de cours doivent être développés et améliorés afin d'être intuitifs et utilisables par tous le public.

**- Adaptable**

Les modules de formation doivent pouvoir être modifiés ou corrigés facilement afin de s'adapter aux changements (de procédures, de métiers, etc.).

**• Pendant le fonctionnement****- Accessible et souple**

Les cours e-Learning doivent être accessibles à tout moment. Les apprenants doivent pouvoir choisir les chapitres qui les intéressent et y consacrer le temps qu'ils souhaitent.

**- Feedback**

Les apprenants doivent pouvoir connaître leur niveau par rapport au cours (avant, pendant et après).

**- Personnalisable**

Le système d'apprentissage doit pouvoir s'adapter aux besoins de la plupart des utilisateurs: recherche d'un complément d'information, acquisition d'une expertise, exemples pratiques,... L'apprenant doit pouvoir également choisir l'ordre dans lequel il va suivre les chapitres d'une formation.

**- Mesurable**

Il est indispensable de pouvoir mesurer la qualité de l'apprentissage par les individus. Cela permet de leur proposer une personnalisation plus poussée des formations. Cette caractéristique permet à la fois une amélioration de la qualité des modules de cours proposés et une amélioration de la qualité des formations réalisées (en termes d'apprentissage).

**- Communautaire**

Il est important de créer une communauté autour du e-Learning afin de conserver la motivation des individus et de développer les compétences en continu. Le tuteur et la classe sont des éléments clés.

## 2.10. Les enjeux du e-Learning

Les enjeux du e-Learning sont notamment de :

- Rendre plus efficaces, plus solides, plus adaptés les processus d'apprentissage et l'accès à la connaissance.
- Bénéficier des avantages des technologies éducatives (interactivité, simulation...).
- Bénéficier des avantages de la formation à distance (plus grande autonomie, élimination de contraintes...).

## 2.11. Différents aspects du e-Learning

### • L'usage des cédéroms éducatifs

On peut lire, écouter, regarder une œuvre dans l'ordre voulu par l'auteur. Naviguer, c'est choisir son parcours, sélectionner ses centres d'intérêts.

Le cédérom est également réputé par son interactivité. L'apprenant peut questionner l'application qui le répond, et même lui pose à son tour des questions pour suivre avec exactitude son degré d'évolution.

### • La visioconférence

Ou "vidéoconférence", c'est se parler en se voyant sur écran. Pour être plus précis, disons que la visioconférence est une télécommunication audiovisuelle interactive en temps réel permettant à des personnes ou groupes de personnes de dialoguer oralement à distance tout en échangeant entre les sites participants les images animées, des interlocuteurs ainsi que des textes, documents graphiques, audiovisuels ou informatiques, ou même une navigation en direct sur le Web.

## 2.12. Les standards et les normes du e-Learning [réf 14]

### 2.12.1. Intérêt de la normalisation pour la formation en ligne

La normalisation pour la formation en ligne répond à cinq objectifs :

- **L'accessibilité** facilite la recherche, l'identification, l'accès aux contenus et composants de formation en ligne.
- **La réutilisabilité** permet d'utiliser les mêmes contenus et composants à différentes fins, dans différentes applications, dans différents produits, dans différents contextes et via différents modes d'accès.
- **La durabilité** permet d'éviter un nouveau développement ou une réingénierie des formats de contenus et des composants dans le cas de changements du support logiciel et technique.
- **L'adaptabilité** est rendue possible par la modularisation des contenus et des composants pour mieux répondre aux besoins des utilisateurs.
- **L'interopérabilité** permet de faire fonctionner ensemble des composants logiciels grâce à des interfaces communes.

### 2.12.2. Les normes en e-Learning [ref 13]

La liste des normes existantes à ce jour dans le monde du e-Learning :

- ISO/CEI 2382-36 Vocabulary for e-learning
- ISO/CEI 19779-1 Collaborative workplace: data model
- ISO/CEI 10780-1 Text-based collaborative learning communication (publié en 2006)
- ISO/CEI 24703-13 Participant identifier (publiée en 2004), définition des types de données qui peuvent être associées aux participants à l'étude, à l'éducation et à la formation
- ISO/CEI Conceptual model for competences
- ISO/CEI 19788 (2 parties), Metadata for learning resources:
  - Compatible avec IEEE LOM 1484.12.1 et Dublin Core ISO 15836 ;
  - utilise les catégories de façon uniquement nominative ;
  - permet l'utilisation du web sémantique ;
  - intégrera la description schématique de profils décrivant les éléments recommandés, optionnels, recommandés-optionnel et non-utilisés.
- ISO/CEI 19796 (4 parties dont 1 publiée en 2005) analyse, et met en application la gestion de qualité et la garantie d'une approche de qualité. Elle servira à comparer différentes approches existantes et à harmoniser ces derniers vers un modèle commun de qualité.
- ISO/CEI 19786 Participant accommodation information, Information de logement liée aux participants dans l'apprentissage, l'éducation, et la formation
- ISO/CEI 24751 Individualized adaptability and accessibility, La portée de cette norme à parties multiples est de fournir un cadre commun pour faciliter les besoins et les préférences d'accessibilités de ressources et d'interfaces utilisateur d'étude appropriées.
- ISO/CEI 24725-2 Profile for rights expression language (publié en 2006) décrit les dispositifs qui sont uniques au champ de l'étude, de l'éducation et de la formation.

### 2.12.3. Pourquoi utiliser les standards ?

On utilise les standards d'E-Learning pour assurer l'interopérabilité entre plates-formes, protéger des investissements dans le développement continu, favoriser l'échange de contenu entre systèmes e-learning, définir une structure de base de contenu pédagogique, introduire les méta-descriptions et les métadonnées et assurer la réutilisation des documents pédagogiques.

#### 2.12.4. Quelques standards en e-Learning

- **SCORM** (Sharable Content Object Reference Model)
- **AICC** (Aviation industry computer based training committee)
- **Learning Object Metadata (LOM)**

##### **SCORM** (Sharable Content Object Reference Model)

Dans le principe originel, SCORM permet aux concepteurs pédagogiques de distribuer leurs contenus sur une large variété de plates-formes e-Learning sans efforts de réécriture. Le modèle SCORM est une suite de normes techniques qui permet de construire des plates-formes e\_Learning et des contenus de formation de manière à fonctionner ensemble. Quelles que soient les versions de SCORM, toutes standardisent deux points distincts : la description du contenu (content packaging) et les échanges de données entre le cours et le LMS (exchanging data at runtime).

1. Le système de description de contenu détermine un ensemble de propriétés du cours : Définition du titre du cours et de ses leçons, liste des objectifs et des pré-requis, URL de lancement des leçons, définitions des notes de passage... Au cœur de la description SCORM un fichier appelé « imsmanifest » contient toutes les informations requises par les LMS pour référencer le cours. Le fichier « manifest » est un fichier XML décrivant la structure du cours tant du point de vue de l'apprenant que du point de vue de l'organisation physique des fichiers.
2. L'environnement d'exécution spécifie comment le contenu communique avec le LMS lorsque le module est en cours de lecture. Il y a 2 étapes importantes dans le processus de communication. Premièrement, le contenu doit trouver le LMS puis, une fois le LMS identifié, le contenu peut communiquer avec une série de commandes « Get » et « Set » complétées par un vocabulaire adéquat. Une commande de type « Get » permet de récupérer une donnée du LMS et une commande de type « Set » permet de mettre à jour une donnée sur le LMS. Par exemple avec ce jeu de commande, il est possible de récupérer le nom de l'apprenant (Get) ou d'enregistrer son score obtenu (Set) sur une leçon.

- **AICC (Aviation industry computer based training committee)**

L'AICC publie des recommandations et un ensemble de spécifications permettant de gérer le chargement d'un contenu dans une plate-forme de formation à distance (LMS), de standardiser la communication entre le contenu et le LMS, ainsi que d'adapter la pédagogie du contenu en fonction de l'apprenant.

Ces spécifications ont été écrites initialement pour un environnement exécutable (Windows) ce qui explique que les fichiers de descriptions des contenus (CIF pour Course Import Files) soient basés sur un format de fichier INI et CSV. Il existe 7 fichiers dont 4 sont obligatoires (.CRS : Course, .DES : Descriptor, .AU : Assignable Unit et .CST : Course Structure Table) et 3 facultatifs en fonction du niveau pédagogique (.PRE : Prerequis, .CMP : Completion Requirements, .ORT : Objectives Relationships).

L'AICC définit deux modes de communication entre le LMS et le contenu :

- API pour Application Programming Interface
- HACP pour HTTP AICC Communication Protocol

Dans les deux cas, l'activité doit être lancée avec au minimum deux paramètres :

- AICC\_URL : URL de la page implémentant la communication AICC côté LMS
- AICC\_SID : Système ID placé par le LMS permettant d'associer les informations à une activité.

- **Learning Object Metadata (LOM)**

Le Learning Object Metadata (LOM) est un schéma de description de ressources d'enseignement et d'apprentissage. Le LOM peut être utilisé pour décrire des ressources tant numériques que non numériques.

Dans le cadre de la formation en ligne, SCORM 2004 spécifie que les métadonnées lorsqu'elles sont incluses dans un paquet manifest doivent être conformes au standard IEEE LOM pour le modèle de données LOM et le schéma XML.

Le LOM a été, dès ses débuts, proposés pour suivre un processus de normalisation au niveau ISO. Le projet de normalisation s'est en définitive développé en optant pour des choix de modélisation et des options techniques différentes de ceux proposés par le LOM. Ce processus a abouti en 2011 à la publication de la norme ISO/IEC 19788 - Metadata for Learning Resources et conduit certains pays à établir des correspondances entre le LOM et cette nouvelle norme.

### **2.13. Conclusion**

Dans ce chapitre, nous avons présenté le nouveau mode d'apprentissage qui est l'e-Learning dont Il apparaît clairement que ce dernier a un impact considérable sur le système éducatif présent.

A ce stade, l'étape analyse et conception de notre système e-Learning peut être lancée. Elle sera l'objectif du chapitre qui suit.

### 3.1. Introduction

Dans le but d'une meilleure organisation et une bonne maîtrise de travail, tout processus de développement d'application ou systèmes informatiques doit suivre une méthode ou démarche bien définie.

Dans ce chapitre, nous allons entamer le processus par une analyse qui mettra en évidence les différents acteurs intervenant dans le système cible ainsi que leurs besoins. La phase conception, s'appuie sur les résultats de la phase analyse donnera la modélisation de la solution des objectifs à atteindre. Pour se faire, notre démarche va s'appuyer sur le langage UML, conçu pour la visualisation, la spécification et la construction des systèmes logiciels.

### 3.2. Analyse

Cette partie a pour objectif la spécification de manière claire de notre application e-Learning. Pour ce faire, il est nécessaire de déterminer globalement ce qui se trouve dans le champ de l'application. De ce fait, on s'intéresse dans cette phase à la définition des besoins ainsi qu'au domaine couvert par l'application.

#### 3.1.1. Définition du besoin

Notre travail consiste à faire fonctionner notre application e-Learning de façon à assurer l'interopérabilité rendant ainsi l'accès ouvert à un grand nombre d'utilisateurs et cela malgré la différence entre leur architecture, structure et leur langage de programmation et leur fonctionnement sur différents systèmes d'exploitation.

#### 3.2.2. Identification des acteurs et des besoins

Dans l'organisation humaine du projet, on distingue principalement les acteurs suivants : l'apprenant, le formateur et l'administrateur de l'application.

**Apprenant** : toute personne inscrite et suivant une formation sur l'application. Après une inscription, le système lui réserve un espace propre à lui. L'apprenant peut ensuite après chaque authentification :

- ✓ Consulter et télécharger les cours relatifs aux modules de son niveau d'inscription.
- ✓ Consulter et télécharger les exercices relatifs aux modules de son niveau d'inscription.
- ✓ Faire les tests de fin niveau relatifs aux modules de son niveau d'inscription.

- ✓ Consulter et télécharger le PV de notes.

**Formateur** : toute personne chargée d'enseigner ses formateurs durant leur cursus. Le formateur peut ensuite après chaque authentification :

- ✓ Publier un cours
- ✓ Publier un exercice ;
- ✓ Modifier un cours.
- ✓ Supprimer un cours,
- ✓ Supprimer un exercice.

**Administrateur** : une personne ayant pour mission la gestion et la maintenance de la plateforme e-Learning à distance. L'administrateur peut ensuite après chaque authentification :

- ✓ Accepter un formateur.
- ✓ Refuser un formateur.
- ✓ Supprimer un formateur.
- ✓ Supprimer un apprenant.
- ✓ Ajouter un module.
- ✓ Supprimer un module.
- ✓ Publier un test ;
- ✓ Supprimer un test ;
- ✓ Publier PV de notes ;
- ✓ Modifier PV de notes.

### 3.3. Conception

#### 3.3.1. Diagramme de contexte :

Le diagramme de contexte est un modèle conceptuel de flux qui permet d'avoir une vision globale des interactions entre le système et les liens avec l'environnement extérieur. Il permet aussi de bien délimiter le champ de l'étude.

Pour notre cas le contexte est donné par la figure suivante :

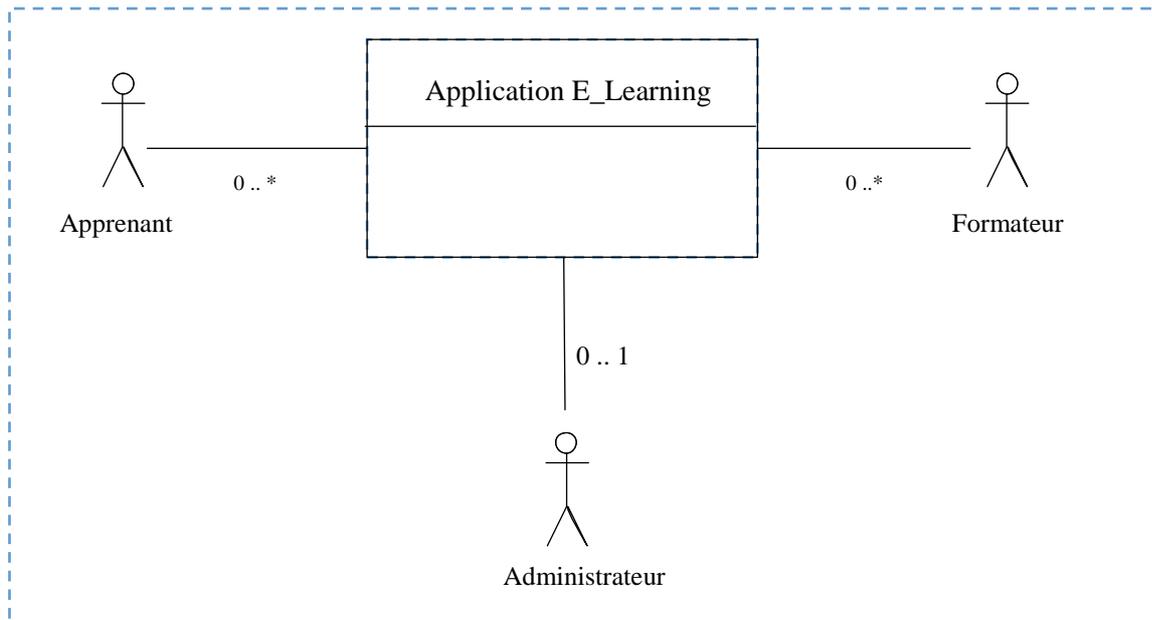


Figure 3.1 : Diagramme de contexte

### 3.3.2. Structuration des cas d'utilisation

Nous les regroupons par domaine fonctionnel en faisant figurer l'acteur principal devant les fonctions auxquelles il a accès :

- L'apprenant peut accéder aux fonctions suivantes : s'inscrire, s'authentifier, consulter cours, consulter exercices, faire le test.
- Le formateur peut accéder aux fonctions suivantes : s'inscrire, s'authentifier, gérer cours, gérer exercices.
- L'administrateur peut accéder aux fonctions suivantes : s'authentifier, gérer formateurs, gérer apprenants, gérer modules, gérer PV de notes, gérer test.
- Le visiteur peut accéder à la fonction suivante : S'inscrire.

La figure suivante modélise les cas d'utilisation de ce système :

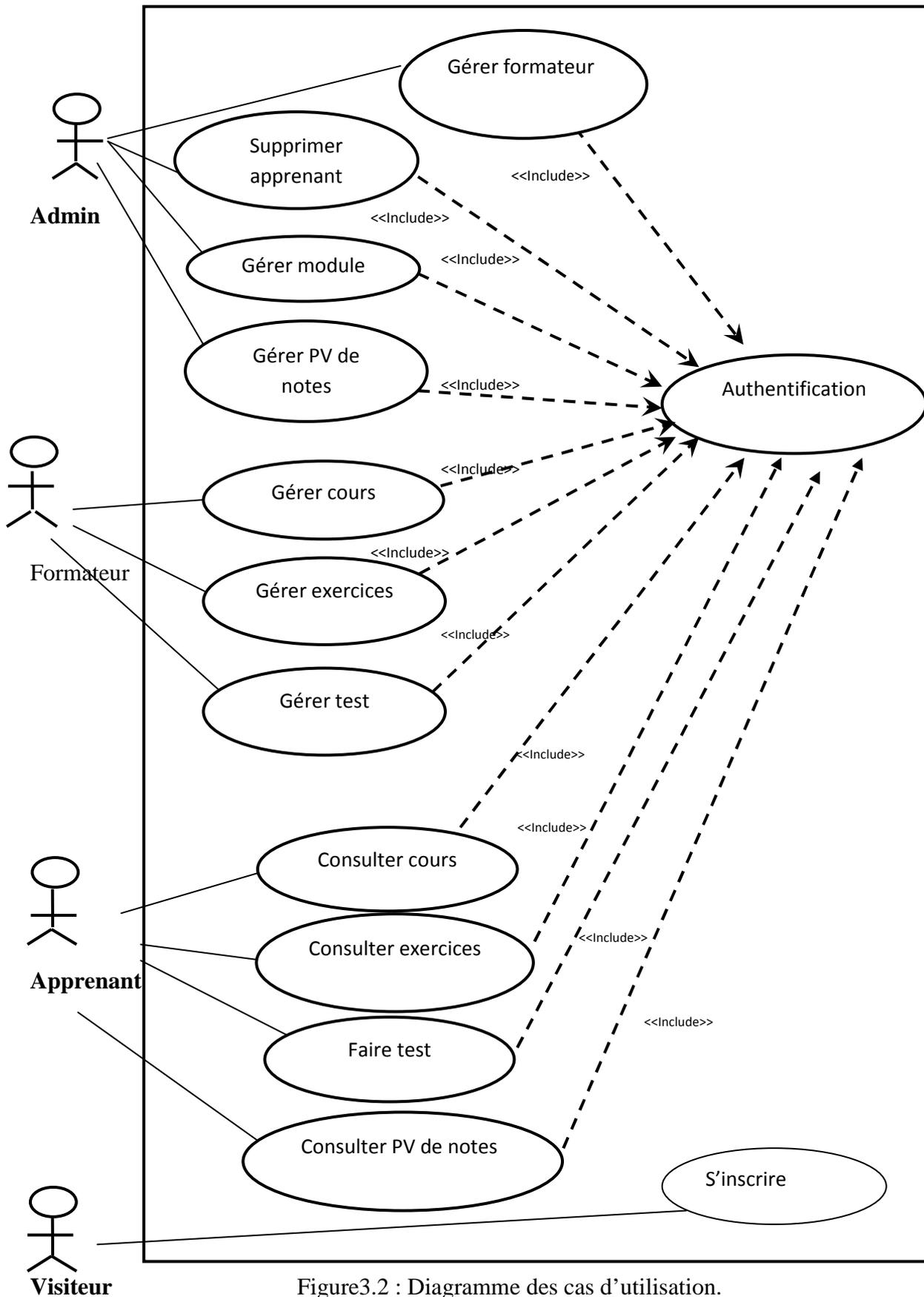


Figure3.2 : Diagramme des cas d'utilisation.

Voici les différents cas d'utilisation que nous avons modélisés :

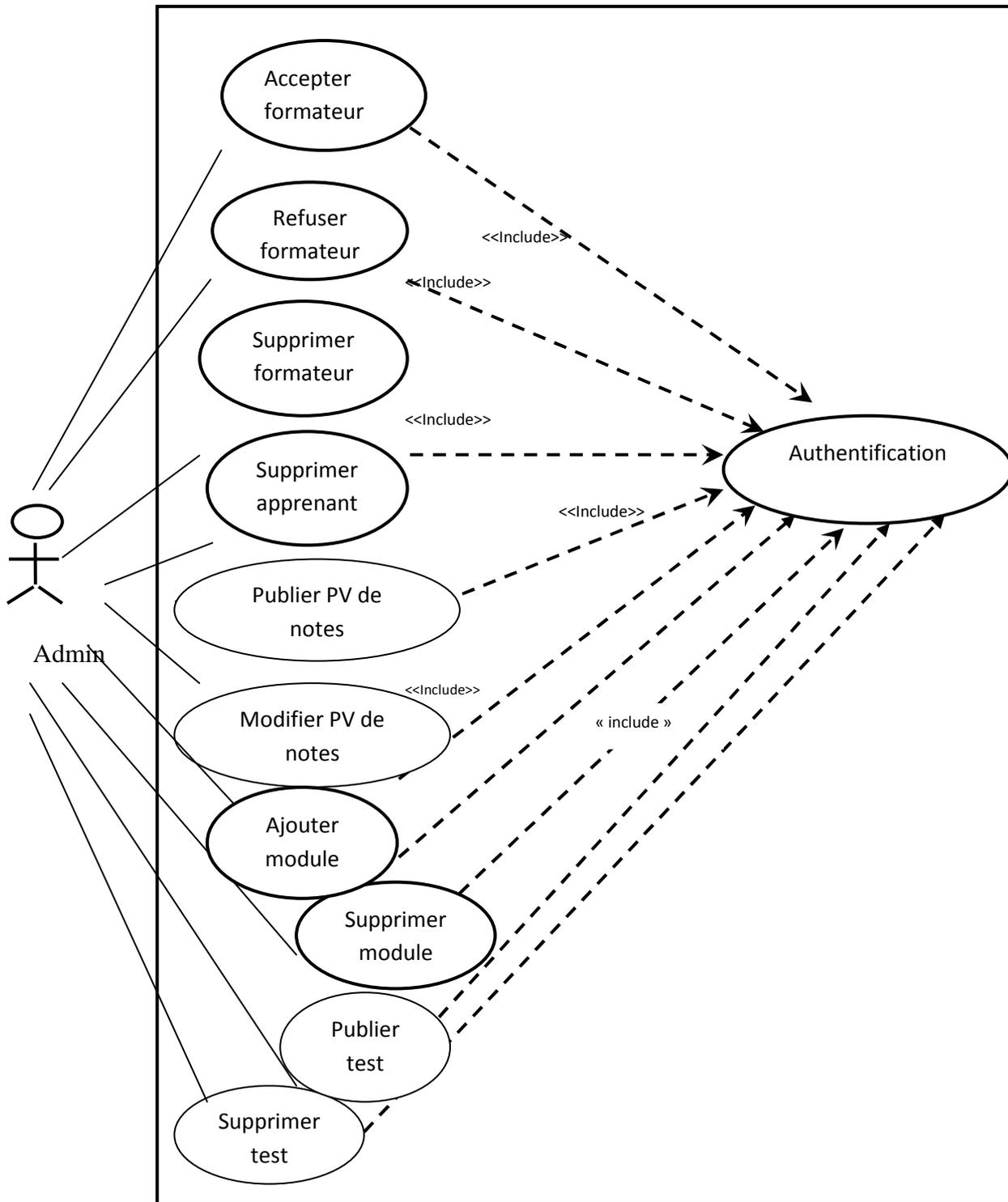


Figure 3.3 : Diagramme des cas d'utilisation de l'acteur administrateur

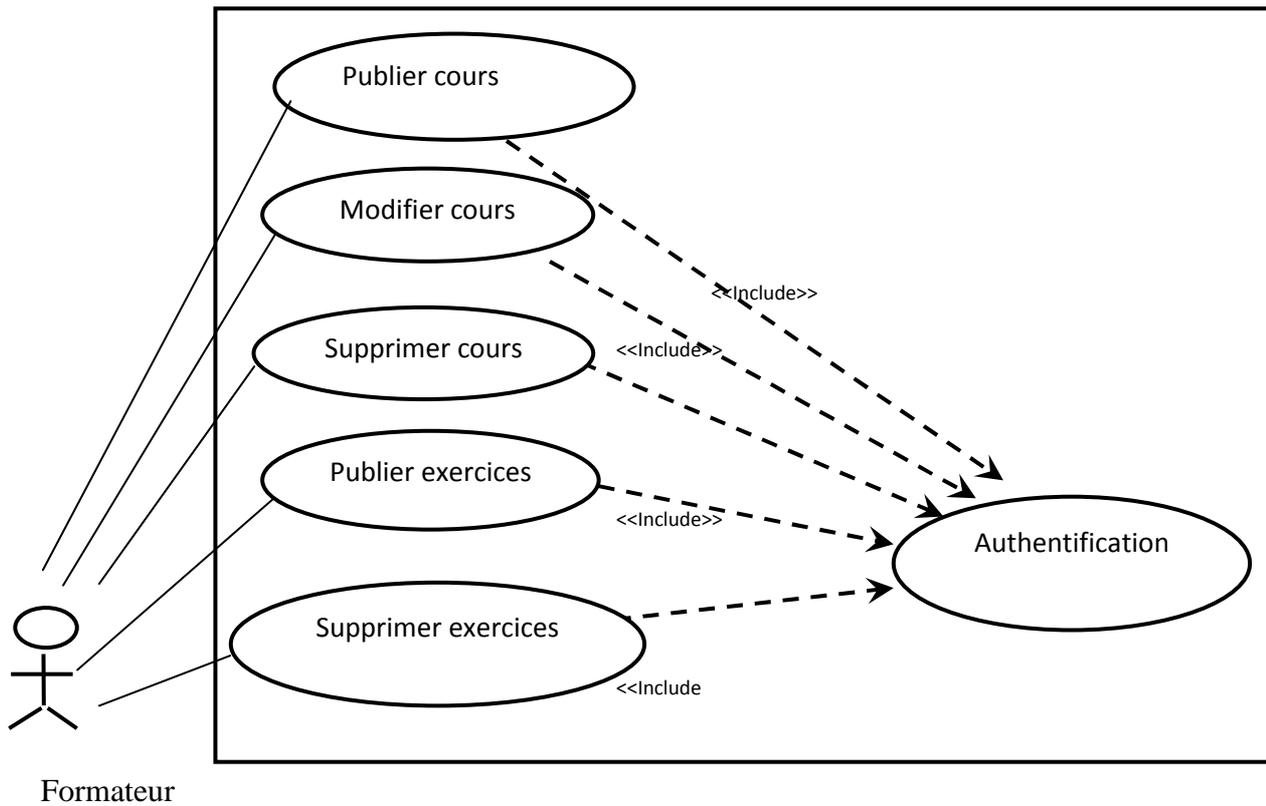


Figure 3.4 : Diagramme de cas d'utilisation de l'acteur formateur

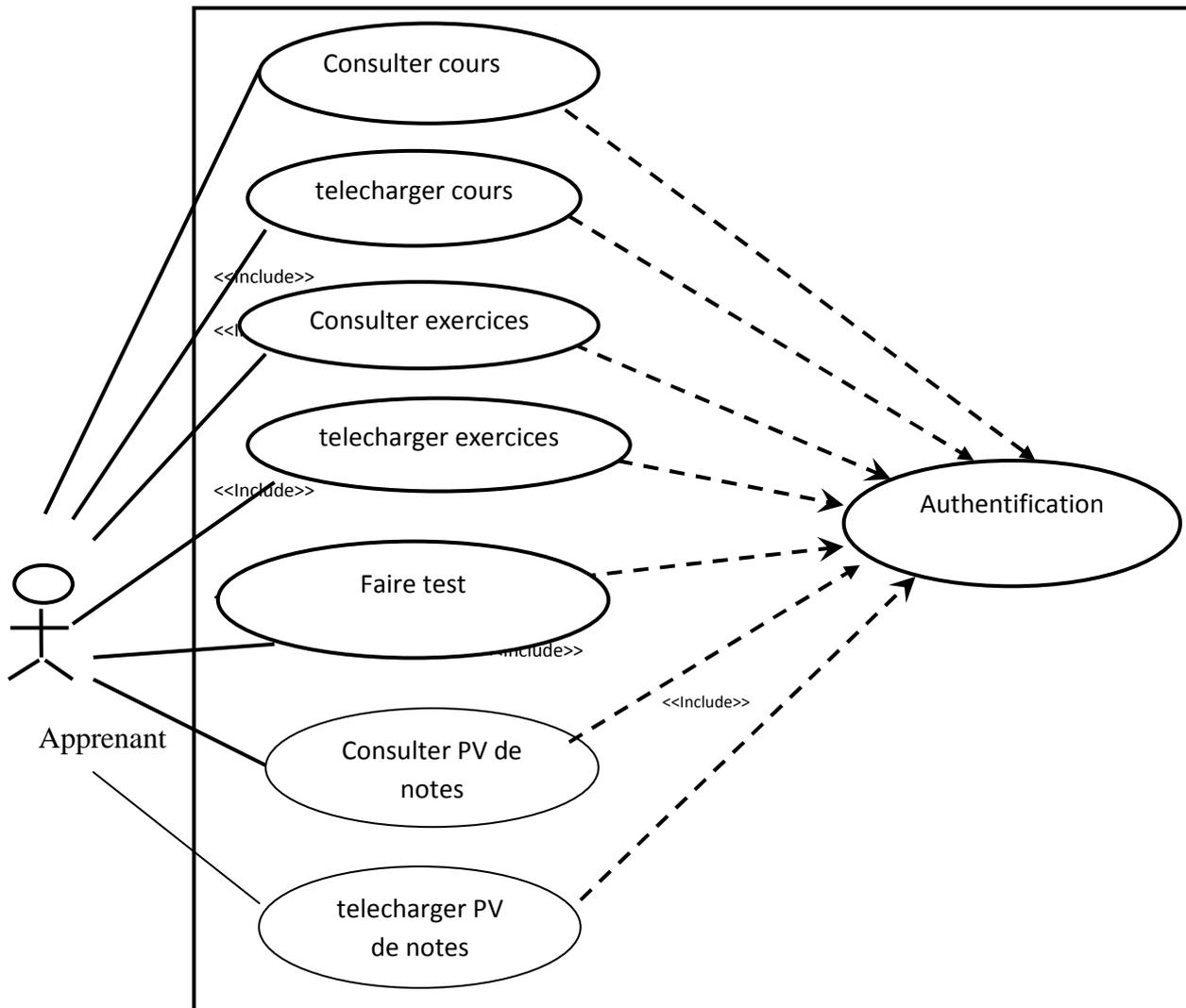


Figure 3.5 : Diagramme de cas d'utilisation de l'acteur apprenant

### 3.3.3. 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.

#### ❖ Cas d'utilisation « Gestion des formateurs »

**Cas d'utilisation :** accepter ou refuser un formateur ;

**Acteur :** Administrateur

**Résumé :** Cette fonctionnalité permet à l'administrateur d'ajouter de nouveaux formateurs.

**Description des scénarios :**

1. Après authentification,
2. L'administrateur atteint son interface, et clique sur suivant pour continuer,
3. Le système affiche une interface contenant le choix : « Gestion Formateurs », « Gestion Apprenant », « Gestion Modules », « Gestion Test », « Gestion PV de notes».
4. L'administrateur décide de gérer les formateurs et clique sur « Gestion Formateurs »,
5. Le système lui retourne une interface contenant les choix : «liste des formateurs», « Messages »,
6. L'administrateur décide de gérer les messages et clique sur « messages» des visiteurs inscrits en tant que formateurs.
7. L'administrateur choisit d'accepter un dossier d'un visiteur en cliquant sur «accepter Formateur»,
8. Le système envoie un e-mail de confirmation au visiteur et l'ajoute à la liste des formateurs»,
9. Le système retourne un message de confirmation d'ajout du formateur,

**Scénario alternatif :**

7. L'administrateur choisit de refuser un dossier d'un visiteur inscrits en tant que formateur en cliquant sur «refuser Formateur»,
8. Le système envoie un e-mail de refus au visiteur et le supprime de la liste d'attente,
9. Le système retourne un message de confirmation de refus du formateur,

**Cas d'utilisation :** Supprimer un formateur ;

**Acteur :** Administrateur

**Résumé :** Cette fonctionnalité permet à l'administrateur de mettre à jour la liste des formateurs.

**Description des scénarios :**

1. Après authentification,
2. L'administrateur atteint son interface, et clique sur suivant pour continuer,
3. Le système affiche une interface contenant le choix : « Gestion Formateurs », « Gestion Apprenant », « Gestion Modules ».
4. L'administrateur décide de gérer les formateurs et clique sur « Gestion Formateurs »,

5. Le système lui retourne une interface contenant les choix : «liste des formateurs », « Messages »,
6. L'administrateur décide de gérer la liste des formateurs et clique sur «liste des formateurs »,
7. Le système lui retourne une interface contenant l'ensemble des formateurs.
8. L'administrateur choisit un formateur et le supprime en cliquant sur «supprimer Formateur»,
9. Le système retourne un message de confirmation de suppression et supprime le formateur de la liste,

#### ❖ Cas d'utilisation « Gestion des apprenant»

**Cas d'utilisation :** Supprimer un formateur ;

**Acteur :** Administrateur

**Résumé :** Cette fonctionnalité permet à l'administrateur de mettre à jour la liste des apprenants.

**Description des scénarios :**

1. Après authentification,
2. L'administrateur atteint son interface, et clique sur suivant pour continuer,
3. Le système affiche une interface contenant le choix : « Gestion Formateurs », « Gestion Apprenant », « Gestion Modules », « Gestion Test », « Gestion PV de notes».
4. L'administrateur décide de gérer les apprenants et clique sur « Gestion Apprenant »,
5. Le système lui retourne une interface contenant la liste des apprenants.
6. L'administrateur choisit un apprenant et le supprime en cliquant sur «supprimer Apprenant»,
7. Le système supprime l'apprenant de la liste et retourne un message de confirmation de la suppression.

**❖ Cas d'utilisation « Gérer module »**

**Cas d'utilisation :** Créer un nouveau module ;

**Acteur :** administrateur

**Résumé :** Cette fonctionnalité permet à l'administrateur d'ajouter de nouveaux modules.

**Description des scénarios :**

1. Après authentification, L'administrateur atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « Gestion Formateurs », « Gestion Apprenant », « Gestion Modules », « Gestion Test », « Gestion PV de notes».
3. L'administrateur décide de gérer les modules et clique sur « Gestion Modules »,
4. Le système lui retourne une interface contenant les choix : «liste des Modules », « Ajouter un Module »,
5. L'administrateur clique sur « Ajouter Module».
6. Le système affiche une interface contenant le formulaire d'ajout,
7. L'administrateur remplit le formulaire et clique sur « Ajouter »,
8. Le système retourne un message de confirmation d'ajout du module.

**Cas d'utilisation :** Supprimer un module ;

**Acteur :** administrateur

**Résumé :** Cette fonctionnalité permet à l'administrateur de supprimer un module existant.

**Description des scénarios :**

1. Après authentification, L'administrateur atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « Gestion Formateurs », « Gestion Apprenant », « Gestion Modules », « Gestion Test », « Gestion PV de notes».
3. L'administrateur décide de gérer les modules et clique sur « Gestion Modules »,
4. Le système lui retourne une interface contenant les choix : «liste des Modules », « Ajouter un Module »,
5. L'administrateur clique sur « liste des Modules»,

6. Le système affiche une interface contenant la liste des modules,
7. L'administrateur sélectionne un module et clique sur « Supprimer »,
8. Le système supprime le module de la liste et retourne un message de confirmation.

❖ **Cas d'utilisation « Gérer test »**

**Cas d'utilisation :** ajouter un test ;

**Acteur :** Administrateur

**Résumé :** Cette fonctionnalité permet à l'administrateur d'ajouter un nouveaux test.

**Description des scénarios :**

1. Après authentification, l'administrateur atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « Gestion Formateurs », « Gestion Apprenant », « Gestion Modules », « Gestion Test », « Gestion PV de notes».
3. l'administrateur décide de gérer les tests et clique sur « Gestion tests »,
4. Le système lui retourne une interface contenant les choix : «ajouter test », «liste des tests »,
5. l'administrateur clique sur « Ajouter test».
6. Le système affiche une interface contenant le formulaire d'ajout,
7. l'administrateur remplit le formulaire et clique sur « Ajouter »,
8. Le système retourne un message de confirmation d'ajout du test.

❖ **Cas d'utilisation « Gérer PV de notes »**

**Cas d'utilisation :** publier un PV de notes ;

**Acteur :** administrateur

**Résumé :** Cette fonctionnalité permet à l'administrateur de publier le PV de notes.

**Description des scénarios :**

1. Après authentification, l'administrateur atteint son interface, et clique sur suivant pour continuer,

2. Le système affiche une interface contenant le choix : « Gestion Formateurs », « Gestion Apprenant », « Gestion Modules », « Gestion Test », « Gestion PV de notes».
3. l'administrateur décide de gérer les PV de notes et clique sur « Gestion PV de notes »,
4. Le système lui retourne une interface contenant les choix : «Publier PV de notes », «liste des PV de notes »,
5. l'administrateur clique sur «Publier PV de notes ».
6. Le système affiche une interface contenant le formulaire d'ajout,
7. l'administrateur remplit le formulaire et clique sur « Publier »,
8. Le système retourne un message de confirmation de la publication.

**Cas d'utilisation** : modifier un PV de notes ;

**Acteur** : administrateur

**Résumé** : Cette fonctionnalité permet l'administrateur de modifier le PV de notes.

**Description des scénarios** :

1. Après authentification, l'administrateur atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « Gestion Formateurs », « Gestion Apprenant », « Gestion Modules », « Gestion Test », « Gestion PV de notes».
3. l'administrateur décide de gérer les PV de notes et clique sur « Gestion PV de notes »,
4. Le système lui retourne une interface contenant les choix : «Publier PV de notes », «liste des PV de notes »,
5. l'administrateur clique sur «liste des PV de notes ».
6. l'administrateur sélectionne un PV de notes et clique sur « Modifier».
7. l'administrateur modifie le PV de notes et clique sur « Confirmer »,
8. Le système retourne un message de confirmation de la modification.

❖ **Cas d'utilisation « Gérer cours »**

**Cas d'utilisation** : Créer un nouveau cours ;

**Acteur** : Formateur

**Description des scénarios :**

1. Après authentification, le formateur atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « Gestion cours », « Gestion exercices ».
3. Le formateur décide de gérer les cours et clique sur « Gestion Cours »,
4. Le système lui retourne une interface contenant les choix : «liste des cours », « Ajouter cours »,
5. Le formateur clique sur « Ajouter cours».
6. Le système affiche une interface contenant le formulaire d'ajout,
7. Le formateur remplit le formulaire et clique sur « Ajouter »,
8. Le système retourne un message de confirmation d'ajout du cours.

**Cas d'utilisation :** Supprimer un cours ;

**Acteur :** formateur

**Résumé :** Cette fonctionnalité permet à l'administrateur de supprimer un cours qu'il a déjà publié.

**Description des scénarios :**

1. Après authentification, Le formateur atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « Gestion cours », « Gestion exercices ».
3. Le formateur décide de gérer les cours et clique sur « Gestion cours »,
4. Le système lui retourne une interface contenant les choix : «liste des cours », « Ajouter cours»,
5. Le formateur clique sur « liste des cours»,
6. Le système affiche une interface contenant la liste des cours,
7. Le formateur sélectionne un cours et clique sur « Supprimer »,
8. Le système supprime le cours de la liste et retourne un message de confirmation de la suppression.

**Cas d'utilisation :** Modifier un cours ;

**Acteur :** formateur

**Résumé :** Cette fonctionnalité permet à l'administrateur de modifier un cours qu'il a déjà publié.

**Description des scénarios :**

1. Après authentification, Le formateur atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « Gestion cours », « Gestion exercices ».
3. Le formateur décide de gérer les cours et clique sur « Gestion cours »,
4. Le système lui retourne une interface contenant les choix : «liste des cours », « Ajouter cours»,
5. Le formateur clique sur « liste des cours»,
6. Le système affiche une interface contenant la liste des cours,
7. Le formateur sélectionne un cours et clique sur « Modifier»,
8. Le système affiche une interface contenant le formulaire de modification,
9. Le formateur remplit le formulaire et clique sur « Modifier »,
10. Le système retourne un message de confirmation de la modification du cours.

❖ **Cas d'utilisation « Gérer exercices »**

**Cas d'utilisation :** ajouter un exercice ;

**Acteur :** Formateur

**Résumé :** Cette fonctionnalité permet au formateur d'ajouter de nouveaux exercices.

**Description des scénarios :**

1. Après authentification, le formateur atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « Gestion cours », « Gestion exercices ».
3. Le formateur décide de gérer les exercices et clique sur « Gestion Exercices »,

4. Le système lui retourne une interface contenant les choix : «ajouter exercice », «liste des exercice »,
5. Le formateur clique sur « Ajouter exercice».
6. Le système affiche une interface contenant le formulaire d'ajout,
7. Le formateur remplit le formulaire et clique sur « Ajouter »,
8. Le système retourne un message de confirmation d'ajout de l'exercice.

**❖ Cas d'utilisation « Consulter cours »**

**Cas d'utilisation** : télécharger un cours ;

**Acteur** : Apprenant

**Résumé** : Cette fonctionnalité permet à l'apprenant de télécharger un cours publié par son formateur.

**Description des scénarios :**

1. Après authentification, l'apprenant atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « consulter cours », « consulter exercices », « faire tests », « consulter PV de notes ».
3. L'apprenant décide de consulter les cours et clique sur « consulter cours »,
4. Le système lui retourne une interface contenant la liste des modules,
5. L'apprenant choisi un module et clique dessus,
6. Le système affiche une interface contenant la liste des cours du module qu'il a choisi,
7. L'apprenant sélectionne un cours et clique sur télécharger.

**❖ Cas d'utilisation « Consulter exercices »**

**Cas d'utilisation** : télécharger un exercice ;

**Acteur** : Apprenant

**Résumé** : Cette fonctionnalité permet à l'apprenant de télécharger un exercice publié par son formateur.

**Description des scénarios :**

1. Après authentification, l'apprenant atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « consulter cours », « consulter exercices », « faire tests », « consulter PV de notes ».
3. L'apprenant décide de consulter les exercices et clique sur « consulter exercices »,
4. Le système lui retourne une interface contenant la liste des modules,
5. L'apprenant choisi un module et clique dessus,
6. Le système lui retourne une interface contenant la liste des cours du module,
7. L'apprenant choisi un cours et clique dessus,
8. Le système affiche une interface contenant la liste des exercices du cours qu'il a choisi,
9. L'apprenant sélectionne un exercice et clique sur télécharger.

**❖ Cas d'utilisation « faire test »**

**Cas d'utilisation :** faire test ;

**Acteur :** Apprenant

**Résumé :** Cette fonctionnalité permet à l'apprenant de faire le test d'un module.

**Description des scénarios :**

1. Après authentification, l'apprenant atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « consulter cours », « consulter exercices », « faire tests », « consulter PV de notes ».
3. L'apprenant décide de consulter les tests et clique sur « faire test »,
4. Le système lui retourne une interface contenant le test,
5. L'apprenant répond aux questions du test et clique sur « envoyer ».
6. Le système envoi les réponses au compte du formateur et affiche un message de confirmation d'envoi

**❖ Cas d'utilisation « Consulter PV de notes »**

**Cas d'utilisation :** télécharger PV des notes ;

**Acteur :** Apprenant

**Résumé :** Cette fonctionnalité permet à l'apprenant de télécharger le PV des notes publié par son formateur.

**Description des scénarios :**

1. Après authentification, l'apprenant atteint son interface, et clique sur suivant pour continuer,
2. Le système affiche une interface contenant le choix : « consulter cours », « consulter exercices », « faire tests », « consulter PV de notes ».
3. L'apprenant décide de consulter les exercices et clique sur « consulter PV de notes »,
4. Le système lui retourne une interface contenant les PV de notes,
5. L'apprenant sélectionne un PV de notes et clique sur télécharger.

**❖ Cas d'utilisation « Authentification »**

**Cas d'utilisation :** Authentification.

**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 :**

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 « s'authentifier»,
6. Le système vérifie les informations saisies avec celles de la base de données,
7. Apres validation, l'utilisateur atteint son espace.

**Scénario alternatif :**

Mot de passe erroné

L'enchaînement nominal démarre au point 6 :

7. le système affiche un message d'erreur indiquant que le login ou le mot de passe est erroné.

L'enchaînement reprend au point 5.

**❖ Cas d'utilisation « Inscription »**

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

**Acteur :** visiteur.

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

**Description des scénarios :**

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 « Apprenant » ,
4. Le système lui affiche une page contenant un formulaire d'authentification ainsi qu'un bouton « s'inscrire » ,
5. L'utilisateur clique sur « s'inscrire » ,
6. Le système lui affiche le formulaire d'inscription,
7. Après remplissage du formulaire, l'utilisateur clique sur « Inscription » ;
8. Le système vérifie la validité des informations du formulaire, et retourne un message de confirmation d'inscription.

**Scénario alternatif :**

Les données du formulaire sont erronées.

L'enchaînement nominal démarre au point 8 ;

9. le système affiche un message.

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

**Acteur :** visiteur.

**Résumé :** Cette fonctionnalité permet au visiteur de s'inscrire en tant que formateur.

**Description des scénarios :**

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 « Apprenant » ,
4. Le système lui affiche une page contenant un formulaire d'authentification ainsi qu'un bouton « s'inscrire » ,
5. L'utilisateur clique sur « s'inscrire »,
6. Le système lui affiche le formulaire d'inscription ,
7. Après remplissage du formulaire, l'utilisateur clique sur « Inscription » ;
8. Le système vérifie la validité des informations du formulaire, et retourne un message de confirmation d'inscription.

**Scénario alternatif :**

Les données du formulaire sont erronées.

L'enchaînement nominal démarre au point 8 ;

9. le système affiche un message.

### 3.3.4. Description graphique des cas d'utilisation

Après avoir décrit les scénarios des cas d'utilisation, nous allons élaborer les diagrammes de séquences. L'objectif de ce type de diagramme offert par UML est de représenter les interactions entre objets en précisant la chronologie des échanges de messages. Les scénarios sont des instances des cas d'utilisation et sont traduits en diagrammes de séquences.

Vu le nombre important de cas d'utilisation et de scénarios, nous n'allons décrire que quelques exemples de cas d'utilisation :

- Inscription formateur ;
- Authentification apprenant ;
- accepter un formateur
- ajouter un cours ;
- modifier PV de notes ;
- télécharger un cours ;
- faire test.

Diagramme de séquence de cas d'utilisation <<inscription formateur>>

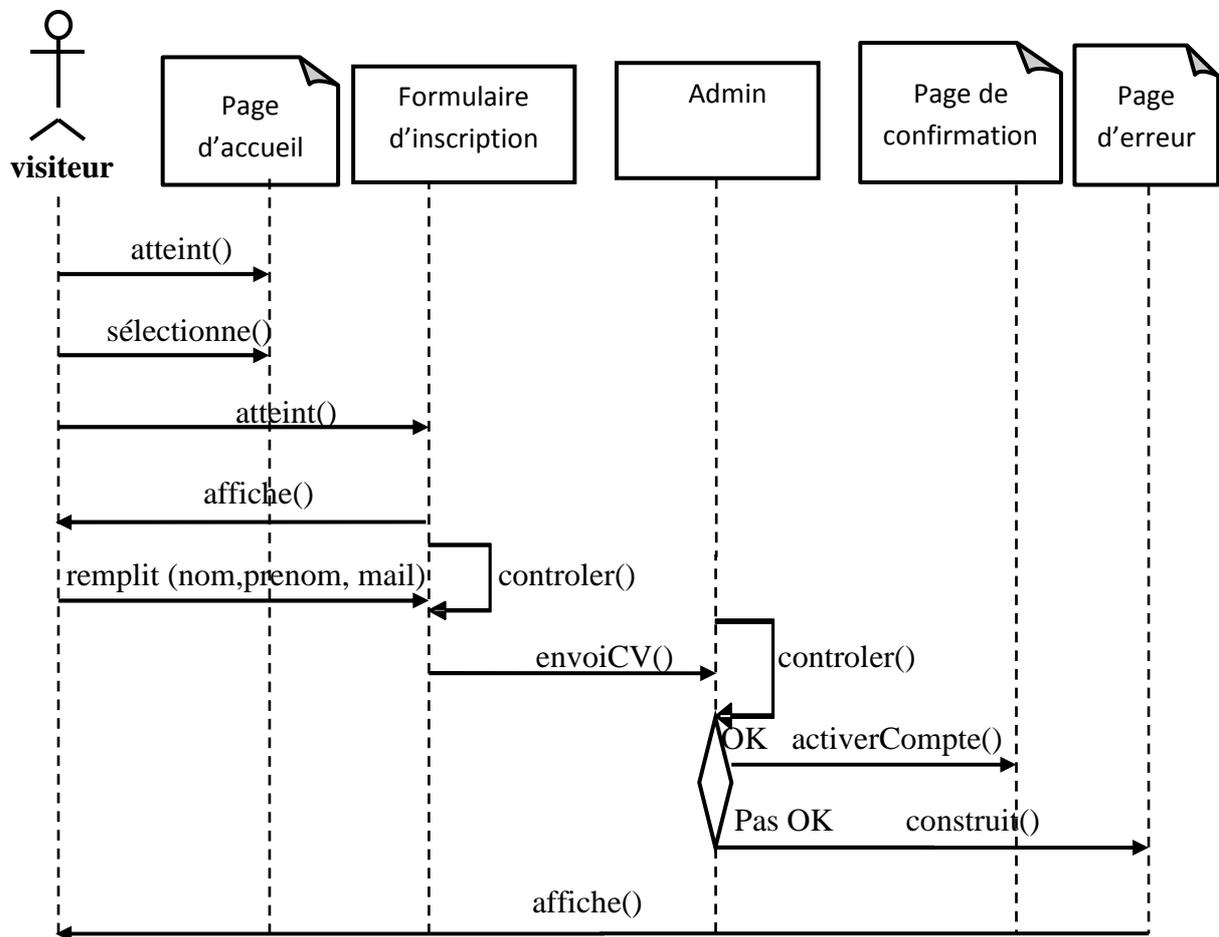


Figure 3.6 : Diagramme de séquence de cas d'utilisation <<inscription formateur>>

## Diagramme de séquence de cas d'utilisation &lt;&lt;authentification apprenant&gt;&gt;

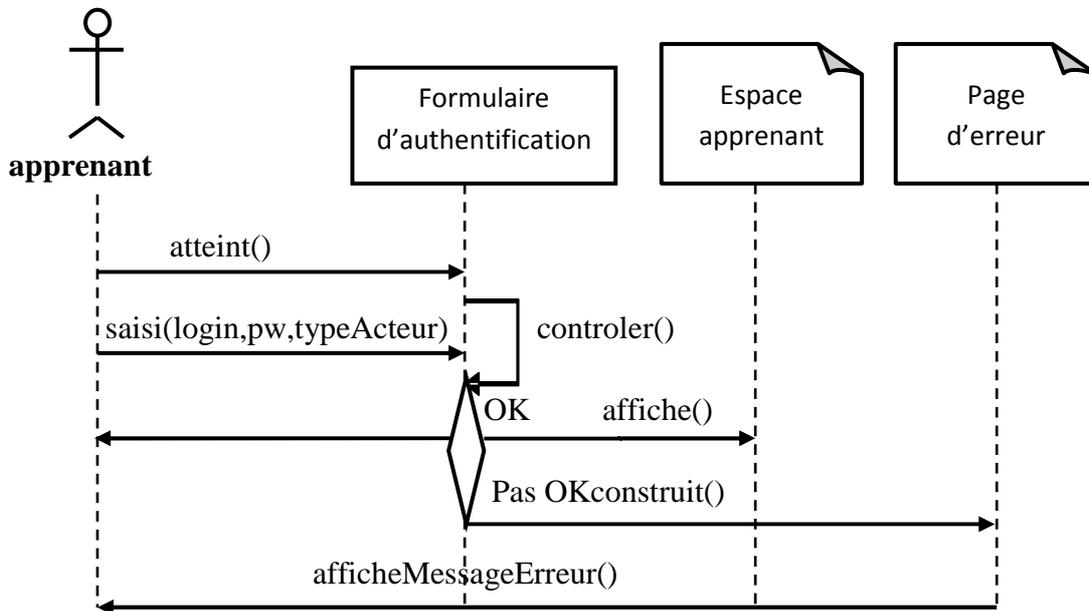


Figure 3.7 : Diagramme de séquence de cas d'utilisation &lt;&lt;authentification apprenant&gt;&gt;

Diagramme de séquence de cas d'utilisation <<accepter un formateur>>

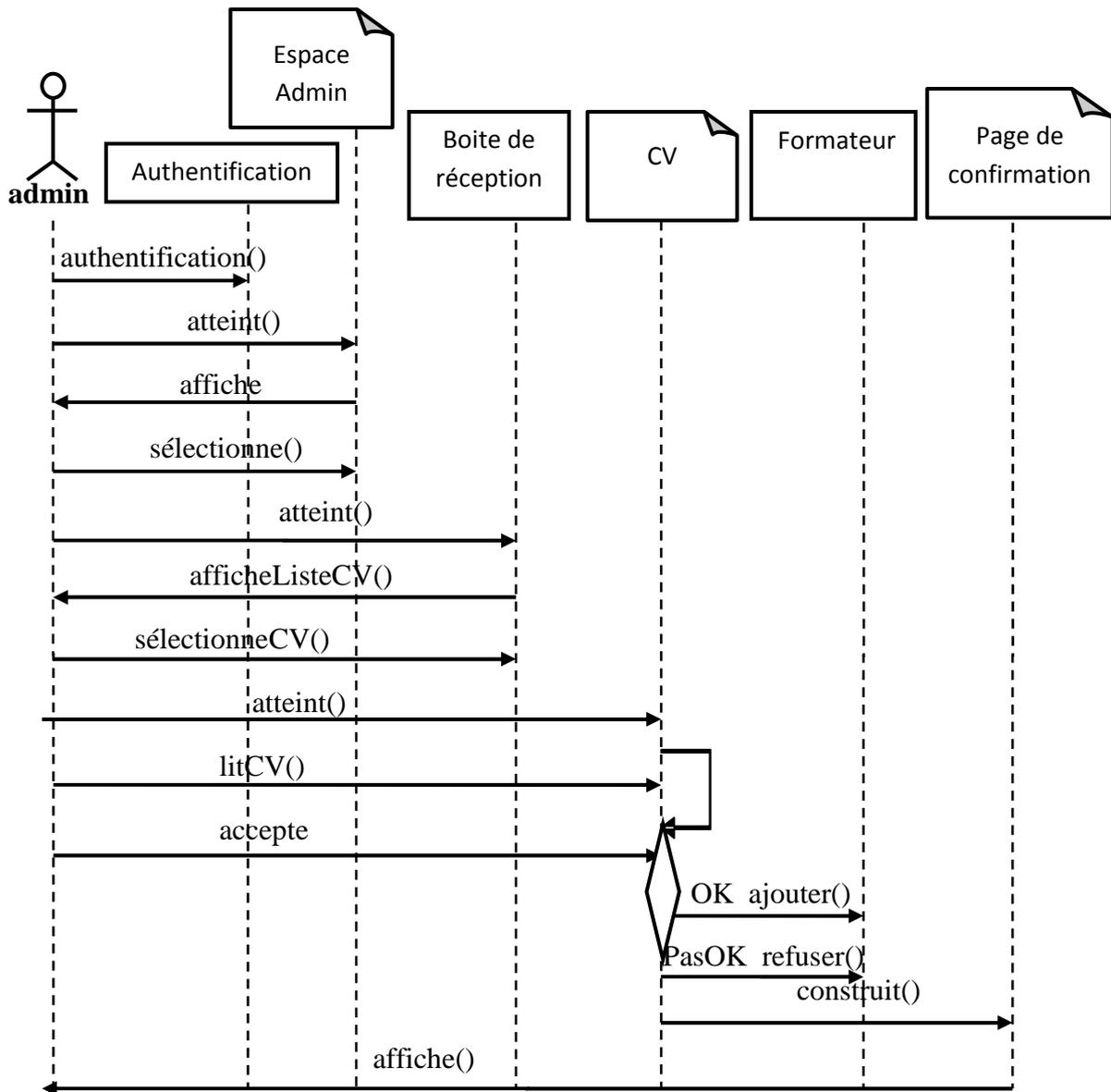


Figure3.8 : Diagramme de séquence de cas d'utilisation <<accepter un formateur>>

Diagramme de séquence de cas d'utilisation <<ajouter cours>>

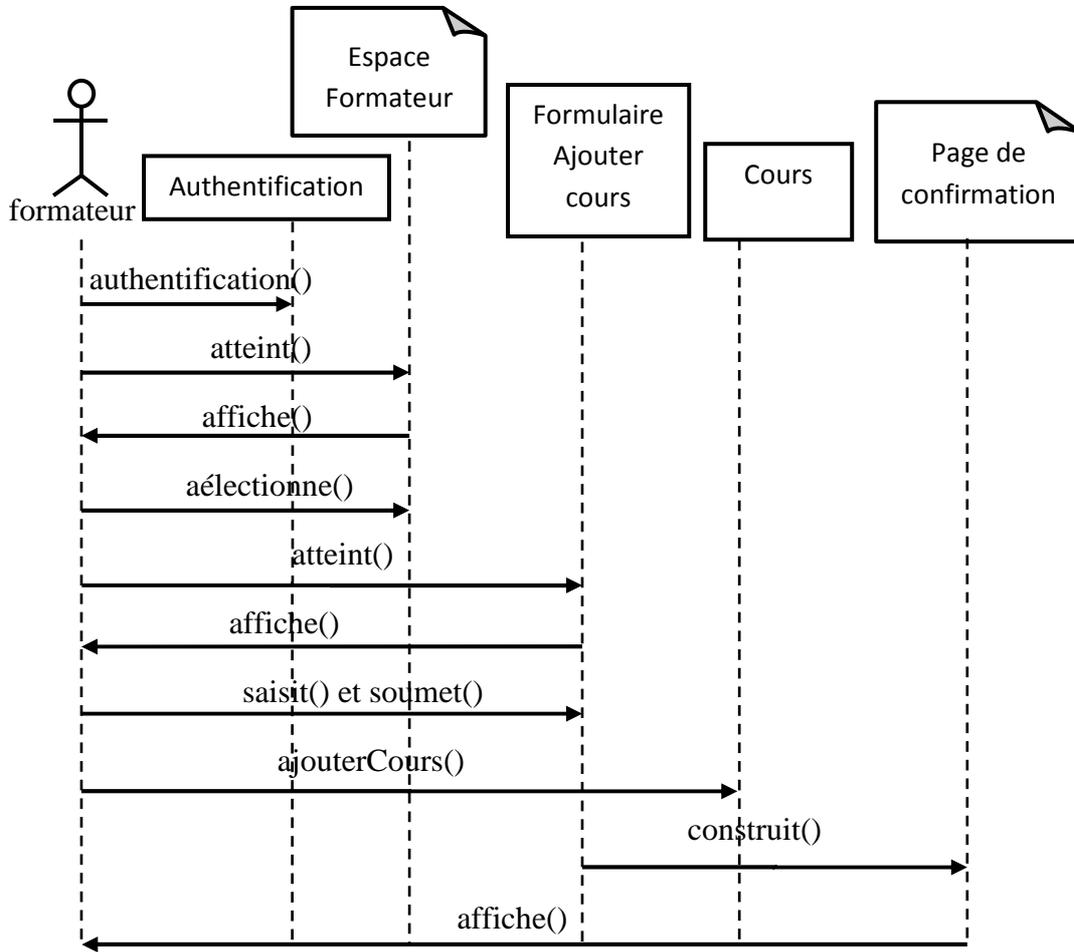


Figure 3.9 : Diagramme de séquence de cas d'utilisation <<ajouter cours>>

Diagramme de séquence de cas d'utilisation <<modifier PV des notes>>

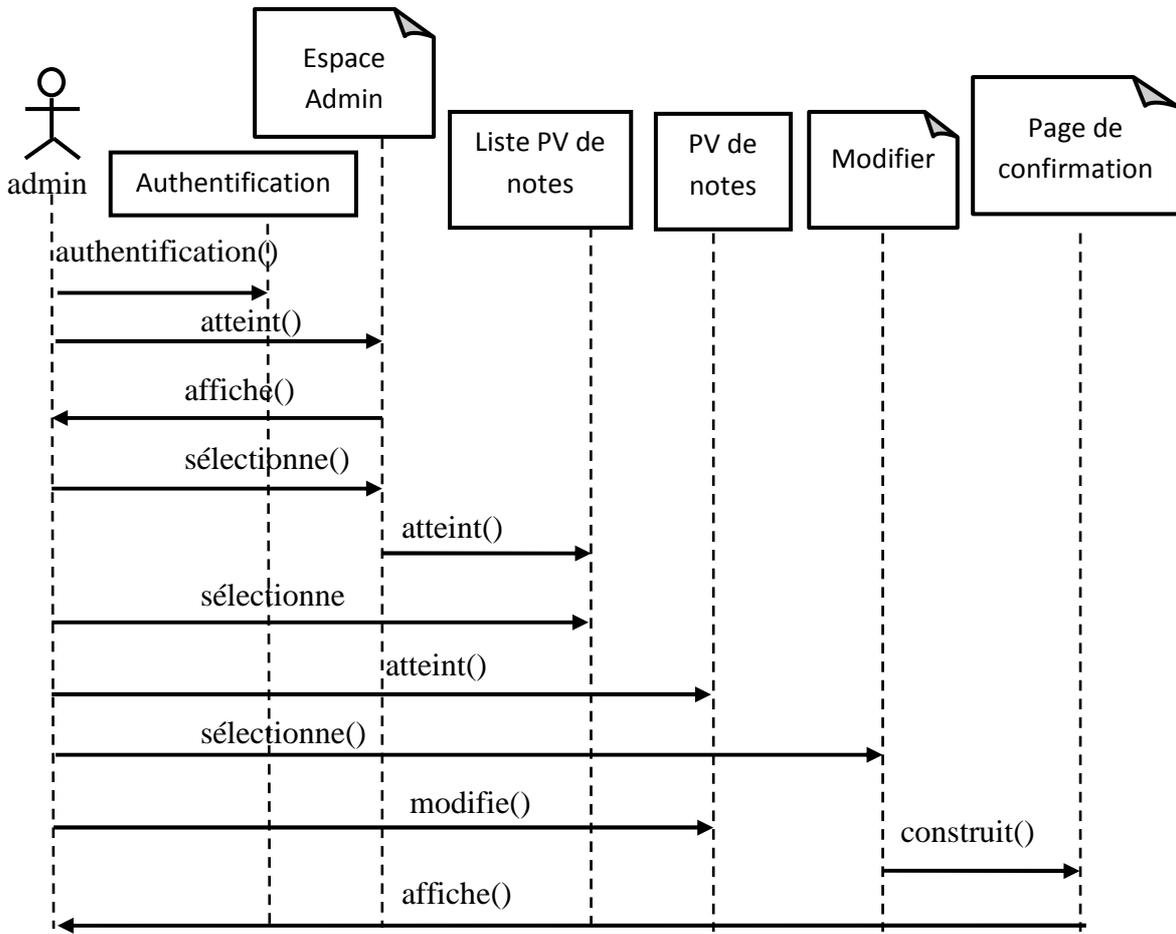


Figure 3.10 : Diagramme de séquence de cas d'utilisation <<modifier PV des notes>>

Diagramme de séquence de cas d'utilisation <<télécharger cours>>

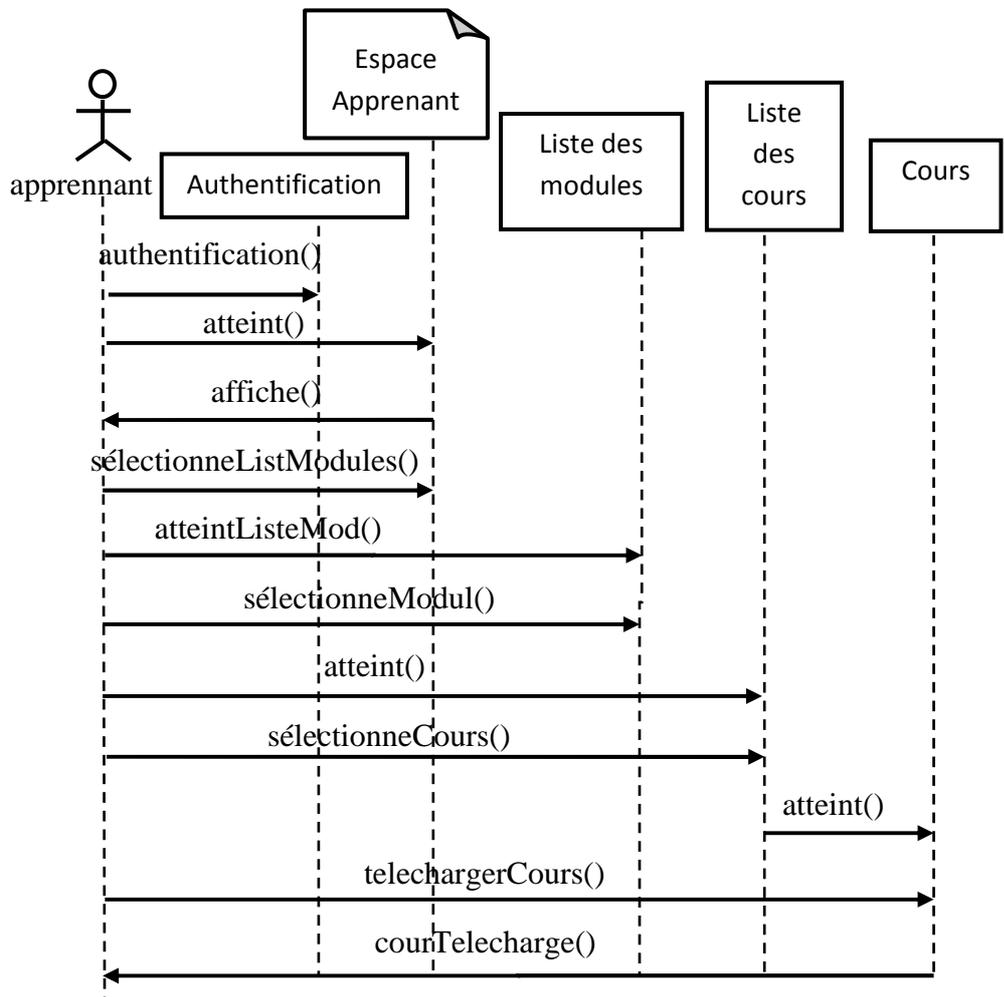


Figure 3.11 : Diagramme de séquence de cas d'utilisation <<télécharger cours>>

## Diagramme de séquence de cas d'utilisation &lt;&lt;faire test&gt;&gt;

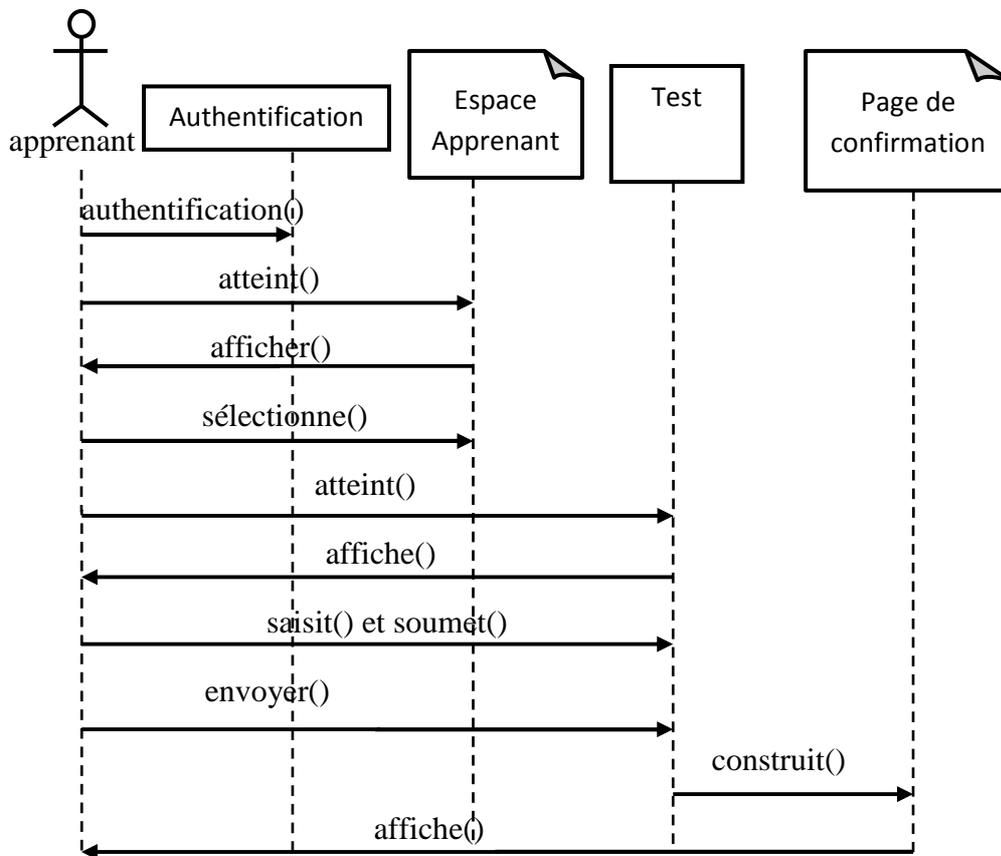


Figure 3.12 : Diagramme de séquence de cas d'utilisation &lt;&lt;faire test&gt;&gt;

**A. 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.

**➤ Les règles de gestion**

- Un acteur peut être un administrateur, un formateur ou un apprenant ;
- Un acteur pédagogique peut être un formateur ou un apprenant ;
- Un acteur est propriétaire d'un seul et un seul compte ;
- Un formateur est chargé d'un ou plusieurs modules ;
- Un cours concerne un ou plusieurs modules ;
- Un exercice concerne un ou plusieurs cours ;
- Un test concerne un et un seul module ;
- Un PV de notes concerne un ou plusieurs modules.

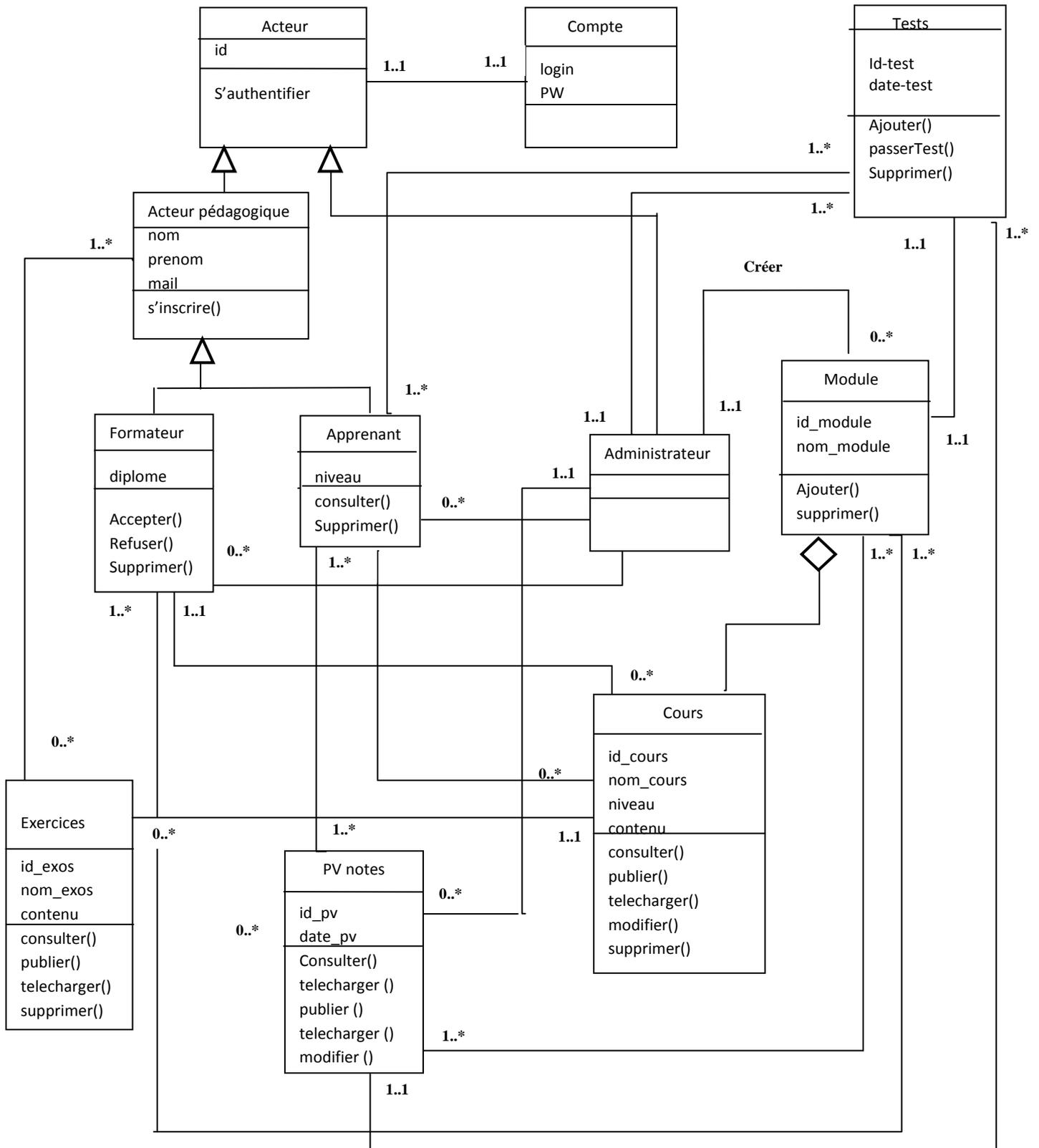


Figure 3.13 : Diagramme de classes globales.

### 3.4. Architecture du système

Comme résultat de l'analyse et la conception basée sur UML, le système que nous motterons en place sera constitué de deux modules complémentaires :

- **Module serveur** : est modélisé par l'ensemble des services web mis en œuvre pour la manipulation, gestion et récupération des ressources sollicitées par les clients, depuis une base de données.
- **Module client** : C'est l'application web facilitant la consultation du compus numérique et qui gère la manière dont sont affichées les informations sur l'interface.

Notre application est sensée être interopérable c'est-à-dire s'adapter aux diverses contraintes logicielles et matérielles liés aux terminaux, permettant ainsi d'atteindre un nombre important d'utilisateurs

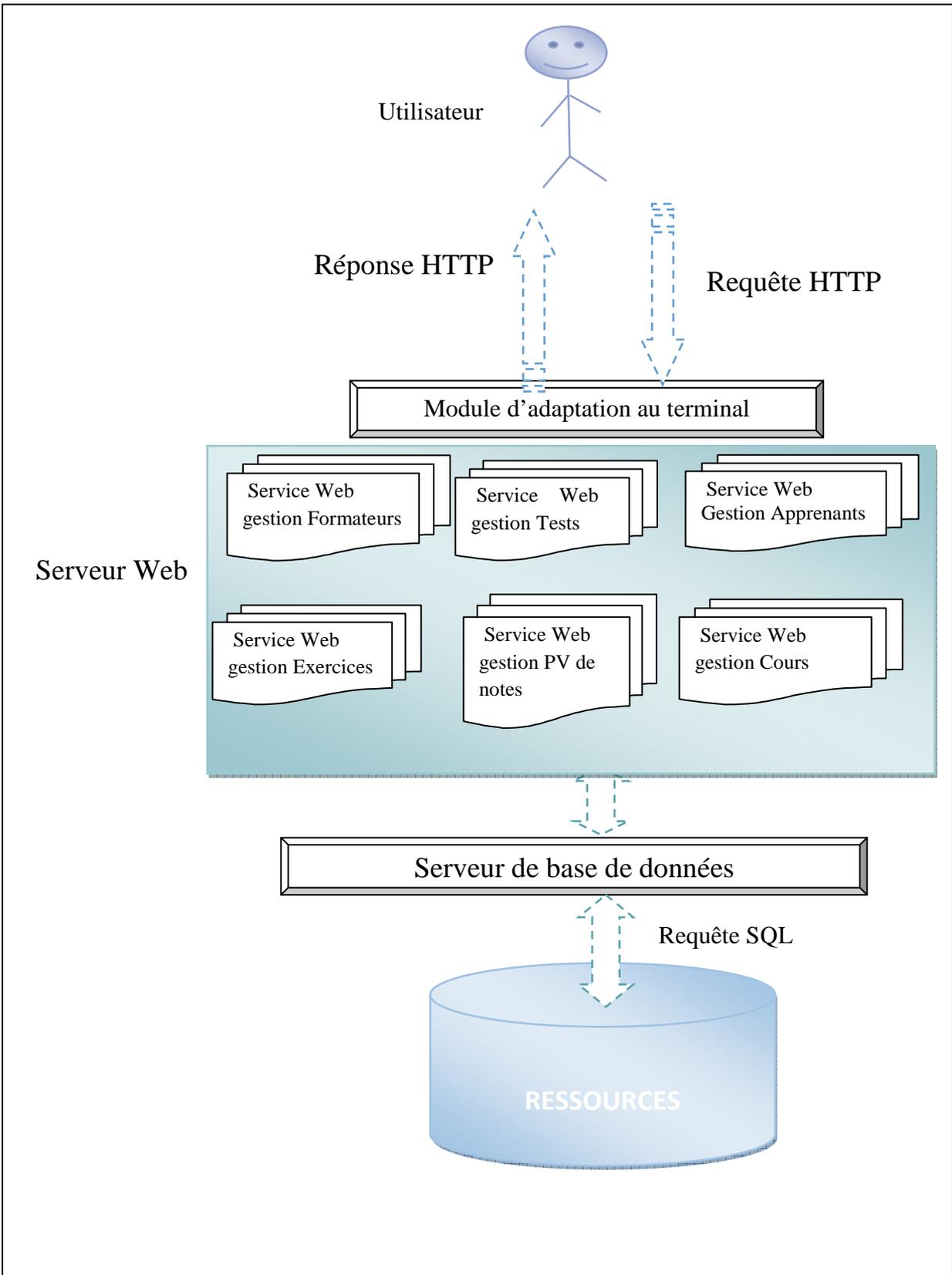


Figure 3.14 : Architecture générale du système

### 3.5. Description des différents services web

Pour répondre aux différents besoins fonctionnels de notre système qui est de permettre l'interopérabilité et la communication avec des applications existantes, en supportant des interfaces de différentes architectures et d'élargir l'audience des utilisateurs, nous avons opté pour le découpage suivant :

- ❖ Le service web ApprenantWS
- ❖ Le service web FormateurWS
- ❖ Le service web PvNotesWS
- ❖ Le service web CoursWS
- ❖ Le service web ExercicesWS
- ❖ Le service web TestsWS

Dans ce qui suit nous présentons une description détaillée des différents services web implémentés.

#### 3.5.1. Le service web : ApprenantWS

Ce service web implémente cinq web méthodes, ses fonctions principales sont :

- ✓ Inscrire les apprenants
- ✓ Vérifier les informations introduites par un apprenant dans le but de s'authentifier
- ✓ Supprimer un apprenant à travers son id
- ✓ Fournir des informations sur un apprenant donné d'après son id
- ✓ Et permettre de récupérer les informations personnelles de l'ensemble des apprenants inscrits (Afficher la liste des apprenants)

#### 3.5.2. Le service web : FormateurWS

Ce service web implémente cinq web méthodes. Principalement il gère :

- ✓ L'inscription des formateurs
- ✓ L'authentification des formateurs

- ✓ La suppression des formateurs à travers leurs identifiant
- ✓ Fournir des informations sur un formateur donné d'après son id
- ✓ Et permettre de récupérer les informations personnelles de l'ensemble des formateurs inscrits (Afficher la liste)

### **3.5.3. Le service web : AdminWS**

Ce service implémente une seule fonctionnalité, le principal rôle de ce service est de permettre à l'administrateur de s'authentifier.

### **3.5.4. Le service web : NotesWS**

- ✓ Affiche les notes d'un apprenant
- ✓ Spécifier un niveau d'un apprenant à partir des résultats du test passé par ce dernier (Si note est supérieur à 50% l'apprenant passe au prochain niveau)

### **3.5.5. Le service web : CoursWS**

Les principales fonctions de ce service web sont :

- ✓ Ajouter un nouveau cours
- ✓ Supprimer un cours à travers son identifiant

### **3.5.6. Le service web : ExercicesWS**

Les principales fonctions de ce service web sont :

- ✓ Ajouter un nouvel exercice
- ✓ Supprimer un exercice à travers son identifiant

### **3.5.7. Le service web : TestsWS**

- ✓ Ajouter un nouveau test
- ✓ Supprimer un test à travers son identifiant

### 3.6. Le diagramme de déploiement

Un diagramme de déploiement permet de donner la structure d'une plate-forme technique, mais aussi de spécifier la localisation des nœuds constitués par des unités distribuées, de préciser où se trouvent les processus et de montrer comment les objets se créent et se déplacent dans une architecture distribuée.

Notre diagramme de déploiement permet de représenter l'environnement de développement de notre application dont l'implémentation se base sur une architecture client/serveur trois tiers :

- ✓ Le premier niveau de cette architecture qui est le est le niveau présentation est constitué d'un navigateur Web tel que Mozilla firefox.
- ✓ Le deuxième niveau est le niveau applicatif (logique applicative) qui est pris en charge par le serveur Apache tomcat.
- ✓ Le troisième niveau qui fournit au niveau intermédiaire les données dont il a besoin, est pris en charge dans notre cas par le SGBD MySQL.

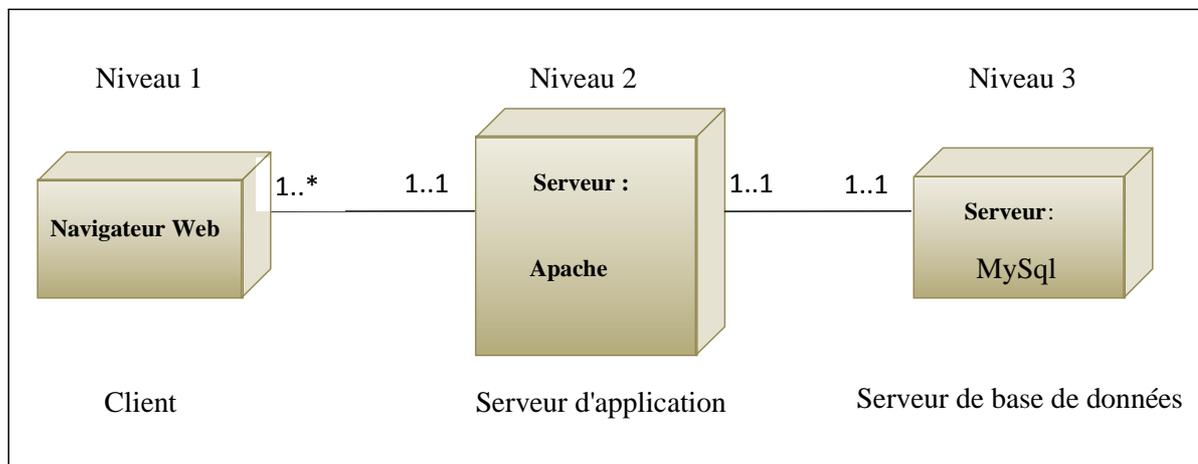


Figure 3.15 : Diagramme de déploiement de l'application

### 3.7. 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 web, ce qui va être l'objet du chapitre suivant tout en exposant l'environnement de développement.

## 4.1. Introduction

Après avoir présenté dans le chapitre précédent la conception et le fonctionnement global de notre système, nous arrivons dans ce chapitre à la mise en œuvre de notre application. Ce chapitre sera divisé en deux parties : la première partie sera consacrée à la description de l'environnement et des outils de développement de notre application. La deuxième partie contiendra les détails techniques d'implémentation des différentes fonctionnalités offertes par notre système.

## 4.2. Les règles de passage de l'objet au relationnel [réf 17]

Dans cette phase le problème de la gestion est posé. Ce problème concerne la manière de gérer les données du système. La solution la plus adaptée pour les systèmes d'information est l'utilisation d'une base de données.

Les SGBDs orienté objet semble plus promoteurs. Le recours aux SGBDs relationnels est le plus concret car ces derniers offrent des solutions aux problèmes d'intégrité et de référence.

UML	Relationnel
Classe	Table
Attribut d'une classe	Les champs de la table
Association	Association
Objet d'une classe	Un enregistrement d'une table
Un lie	Une jointure entre deux tables

Tableau 4.1 : Les règles de passage de l'objet au relationnel

### Schéma relationnel :

Compte (login,PW) ;

Administrateur (id\_admin, login\*) ;

Formateur (id\_form,nom\_form, prenom\_form, mail\_form, diplôme, login\*) ;

Apprenant (id\_app, nom\_app, prenom\_app, mail\_app, niveau, login\*) ;

Module (id\_mod, nom\_mod, id\_admin\*);

Cours (id\_cours, intitule, contenu\_cours, niveau, id\_form\*);

Exercice (id\_exos, nom\_exos, contenu\_exos ,id\_cours\*);

Test (id\_test, date\_test, id\_mod\*, id\_admin\*);

PV\_notes (id\_PV, date\_PV, id\_admin\*);

## 4.2. Le modèle physique de données

Le modèle physique de données est la traduction du modèle logique de données dans un langage de description de données spécifique au système de gestion de base de données utilisé.

Le modèle physique des données est l'implémentation des données (tables) issues du modèle logique des données en machine afin d'aboutir à la description des fichiers de base de données.

**Table 1** : « compte »

Nom du champ	Description	Type de données	Clé (s)
<b>login</b>	Nom utilisateur de l'acteur	Varchar (20)	Primaire
<b>PW</b>	Mot de passe de l'acteur	Varchar (8)	

Tableau 4.2 : Table « compte »

**Table 2** : «administrateur»

Nom du champ	Description	Type de données	Clé (s)
<b>id_admin</b>	identifiant de l' admin	Int (2)	Primaire
<b>login</b>	Nom utilisateur de l'admin	Varchar (20)	Etrangere

Tableau 4.3 : Table «administrateur»

Table 3 : «formateur»

Nom du champ	Description	Type de données	Clé (s)
<b>id_form</b>	identifiant du formateur	Int (10)	Primaire
<b>nom_form</b>	nom du formateur	Varchar (30)	
<b>prenom_form</b>	prenom du formateur	Varchar (30)	
<b>mail_form</b>	Mail du formateur	Varchar (60)	
<b>diplome</b>	Diplôme du formateur	Varchar (30)	
<b>login</b>	Nom utilisateur de l'admin	Varchar (20)	Etrangere

Tableau 4.4: Table «formateur»

Table 4: «apprenant»

Nom du champ	Description	Type de données	Clé (s)
<b>id_app</b>	identifiant de l'apprenant	Int (10)	Primaire
<b>nom_app</b>	nom de l'apprenant	Varchar (30)	
<b>prenom_app</b>	prenom de l'apprenant	Varchar (30)	
<b>mail_app</b>	Mail apprenant	Varchar (60)	
<b>niveau</b>	Niveau de l'apprenant	ENUM (A, B, C, D)	
<b>login</b>	Nom utilisateur de l'admin	Varchar (20)	Etrangere

Tableau 4.5: Table «apprenant»

Table 5: «Module»

Nom du champ	Description	Type de données	Clé (s)
<b>id_mod</b>	identifiant du module	Int (10)	Primaire
<b>nom_mod</b>	nom du module	Varchar (30)	
<b>Id_admin</b>	Identifiant de l'admin	Int (2)	Etrangere

Tableau 4.6: Table «Module»

Table 6: «Cours»

Nom du champ	Description	Type de données	Clé (s)
<b>id_cours</b>	identifiant du cours	Int (10)	Primaire
<b>intitule</b>	Intitulé du cours	Int (30)	Etrangère
<b>Contenu_cours</b>	contenu du cours	Varchar (100)	
<b>niveau</b>	niveau du cours	ENUM (A, B, C, D)	
<b>Id_form</b>	Identifiant du formateur	Int (10)	Etrangere

Tableau 4.7: Table «Cours»

Table 7: «exercice»

Nom du champ	Description	Type de données	Clé (s)
<b>id_exos</b>	identifiant du PV de notes	Int (10)	Primaire
<b>nom_exos</b>	identifiant de l'apprenant	Int (30)	
<b>Contenu_exos</b>	Contenu de l'exercice	Varchar (100)	
<b>Id_cours</b>	Identifiant du cours	Int (10)	Etrangere

Tableau 4.8: Table «exercice»

Table 8 : «test»

Nom du champ	Description	Type de données	Clé (s)
<b>id_test</b>	identifiant du test	Int (10)	Primaire
<b>date_test</b>	Date du test	Date	
<b>id_mod</b>	identifiant du module	Int (10)	Etrangère
<b>id_admin</b>	Identifiant de l'admin	Int (2)	Etrangere

Tableau 4.9: Table «test»

Table 9: «PV\_notes»

Nom du champ	Description	Type de données	Clé (s)
id_pv	identifiant du PV de notes	Int (3)	Primaire
date_pv	Date du PV	DATE	
id_admin	Identifiant de l'admin	Int (2)	Etrangere

Tableau 4.10: Table «PV\_notes»

### 4.3. L'environnement technique de développement

#### 4.3.1. Outils de développement

Notre environnement de développement est constitué d'un ensemble d'outils à savoir :

##### 4.3.1.1. Le serveur web apache

Tout développement de site Web requiert un serveur Web qui s'occupe du traitement des requêtes des clients et l'exécution des programmes sur la machine serveur.

Le serveur Web Apache est développé par un groupe de personne autonome. L'objectif était de développer un serveur http (Web) puissant et utilisable gratuitement.

Le serveur web apache est très bien placé parmi d'autres serveurs, puisqu'il :

- représente plus de deux tiers des serveurs actuellement installés.
- Il présente un niveau de performances élevé pour des exigences matérielles modestes
- Il est gratuit et robuste.

##### 4.3.1.2. Le module Tomcat

Le conteneur de servlets choisi est le moteur Tomcat 8.0 développé par la fondation Apache. Le dialogue entre le moteur de servlet et le serveur Web s'effectue à l'aide d'un module logiciel appelé connecteur. Le module Tomcat du serveur Apache a été développé à partir des sources de Sun Microsystems. Il représente une implémentation de référence pour les servlets. Tomcat peut fonctionner seul, mais cela n'est pas une solution efficace.

Inexploitation il est préférable d'associer Tomcat avec un serveur http plus puissant, qui se chargera du contenu statique.

### 4.3.1.3. Le serveur de bases de données

#### ➤ Le serveur MySQL

MySQL est un véritable serveur de base de données SQL Multiutilisateurs et multitraitements. Cela permet d'établir des connexions rapides et d'utiliser la même mémoire cache pour plusieurs requêtes.

MySQL est basé sur une bibliothèque de gestion de données éprouvée depuis de nombreuses années et faisant appel à des index d'arbres binaires. Grâce à cela, le cœur du système peut afficher une performance remarquable, tout particulièrement dans les accès indexes.

Les principaux objectifs de MySQL sont la rapidité, la robustesse et la facilité d'utilisation.

#### Fonctionnalité de MySQL

- **Multitraitement** : MySQL est multitraitement en utilisant les threads du noyau. il peut utiliser plusieurs CPU.
- **Langues** : Le serveur peut fournir au client les messages d'erreurs en plusieurs langues.
- **APIs** : Les applications de base de données MySQL peuvent être écrites en C, C++, Eiffel, java, perl, PHP, python et TCL.
- **Multi plates-formes** : Prise en charge de plus de 20 plates formes de système d'exploitation (Win95/98 et NT, UNIX, LINUX).
- **Tables** : MySQL stocke chaque table sous forme de fichier distinct dans le répertoire de la base de données. La taille maximale d'une table est comprise entre 4 GO et la taille maximale du fichier accepté par le système d'exploitation.

#### ➤ L'Interface graphique PhpMyAdmin

C'est une interface conviviale gratuite réalisée en langage PHP, permettant d'administrer des bases de données MySQL via un navigateur web. PhpMyAdmin permet de :

- ✓ Créer /supprimer de nouvelles bases.
- ✓ Créer /modifier/supprimer /copier des tables.
- ✓ Afficher /ajouter /modifier /supprimer des tuples dans les tables.

- ✓ Effectuer des sauvegardes de la structure et /ou des données.
- ✓ Effectuer n'importe quelle requête.
- ✓ Gérer les privilèges des utilisateurs.

#### 4.3.1.4. L'IDE Netbeans

Netbeans est un environnement de développement intégré (Integrated Development Environment) qui a été créé à l'initiative de Sun Microsystems. Il présente toutes les caractéristiques indispensables à un environnement de qualité, que ce soit pour développer en Java, Ruby, C/C++ ou même PHP.

NetBeans est sous licence OpenSource, il permet de développer et déployer rapidement et gratuitement des applications graphiques Swing, des Applets, des JSP/Servlets, des architectures J2EE, dans un environnement fortement personnalisable.

L'IDE NetBeans repose sur un noyau robuste, la plateforme NetBeans, que vous pouvez également utiliser pour développer vos propres applications Java, et un système de plugins performant, qui permet d'avoir un IDE modulable.

NetBeans détient un support de développement d'applications Web avec des améliorations pour l'édition des JSP et JSF, la gestion serveur et le support des dernières versions de Tomcat.

Enfin cet IDE possède un débogueur de grande qualité ainsi qu'une interface graphique améliorée.

#### 4.3.1.5. 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

#### 4.3.1.6. 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.

### 4.3.2. Les langages utilisés

#### 4.3.2.1. 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.

#### 4.3.2.2. SQL

C'est un langage de manipulation de base de données mis au point dans les années 70 par IBM. Il permet notamment :

- La manipulation des tables (création, suppression, modification de la structure des tables).
- La manipulation des bases de données (sélection, modification et suppression d'enregistrements).
- La gestion des droits d'accès aux modifications.

### 4.3.2.3. XHTML (Extensible HyperText Markup Language)

XHTML est un langage de balisage qui sert à écrire des pages web. Conçu à l'origine comme le successeur de HTML, XHTML se fonde sur la syntaxe définie par XML, plus récente, mais plus simple que celle définie par SGML sur laquelle repose HTML. Il s'agissait en effet à l'époque de tirer parti des bénéfices techniques attendus de la simplification offerte par XML.

### 4.3.2.4. HTML

L'HyperText Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade). HTML est initialement dérivé du *Standard Generalized Markup Language* (SGML).

### 4.3.2.5. HTML5 (HyperText Markup Language 5)

HTML5 Est la dernière révision majeure d'HTML. Cette version a été finalisée le 28 octobre 2014. HTML5 spécifie deux syntaxes d'un modèle abstrait défini en termes de DOM : HTML5 et XHTML5. Le langage comprend également une couche application avec de nombreuses API, ainsi qu'un algorithme afin de pouvoir traiter les documents à la syntaxe non conforme. Le travail a été repris par le W3C en mars 2007 après avoir été lancé par le WHATWG. Les deux organisations travaillent en parallèle sur le même document afin de maintenir une version unique de la technologie. Le W3C vise la clôture des ajouts de fonctionnalités le 22 mai 2011 et une finalisation de la spécification en 2014, et encourage les développeurs Web à utiliser HTML 5 dès maintenant.

#### 4.3.2.6. Les feuilles de style (CSS)

Elles ont été mises au point afin de compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation. En effet, le HTML offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte, toutefois chaque élément possède son propre style, indépendamment des éléments qu'ils entourent. Grâce aux feuilles de style, lorsque la charte graphique d'un site composé de plusieurs centaines de pages web doit être changée, il suffit de modifier la définition des feuilles de style en un seul endroit pour changer l'apparence du site tout entier !

Elles sont appelées « *feuilles de style en cascade* » car il est possible d'en définir plusieurs et que les styles peuvent être hérités en cascade.

#### 4.3.2.7. JavaScript

JavaScript est un langage de scripts qui est incorporé aux balises Html, permet d'améliorer la présentation et l'interactivité des pages web. JavaScript est donc une extension du code Html des pages web. Les scripts qui s'ajoutent aux balises Html, peuvent en quelques sortes être comparés aux macros d'un traitement de texte. Ces scripts vont être gérés et exécutés par le navigateur lui-même sans devoir faire appel aux ressources du serveur.

Son objectif principal est d'introduire de l'interactivité avec les pages Html et effectuer des traitements simples sur le poste de travail de l'utilisateur tel que les contrôles de saisis pour valider les champs d'un formulaire, ouvrir ou fermer des nouvelles fenêtre ou encore gérer des éléments graphiques.

Les balises annonçant un code JavaScript sont les suivantes :

```
<script langage= "JavaScript">Mettez ici votre code </script >.
```

#### 4.4. Présentation des interfaces

Nous vous présentons, dans ce qui suit, quelques interfaces de notre application :

- **La page d'accueil :** C'est le point d'entrée à la plateforme. Elle permet aux différents utilisateurs d'accéder à leur profil, Et la page d'authentification des apprenants

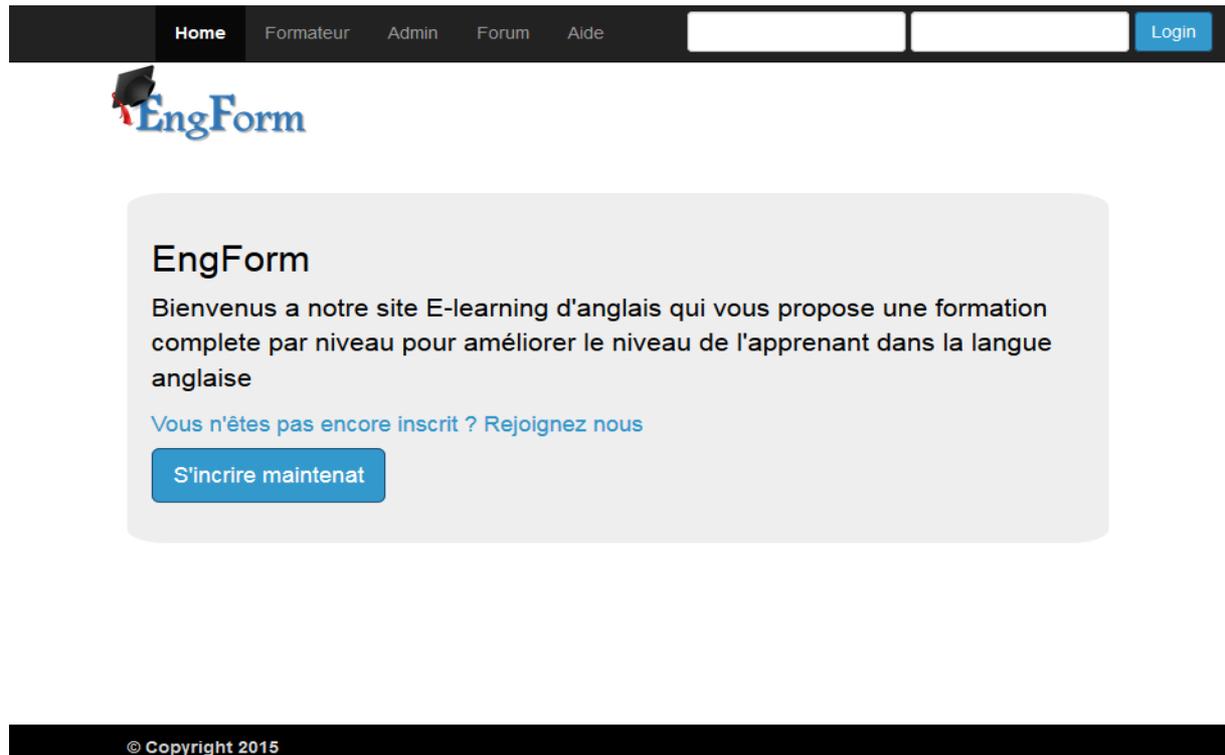
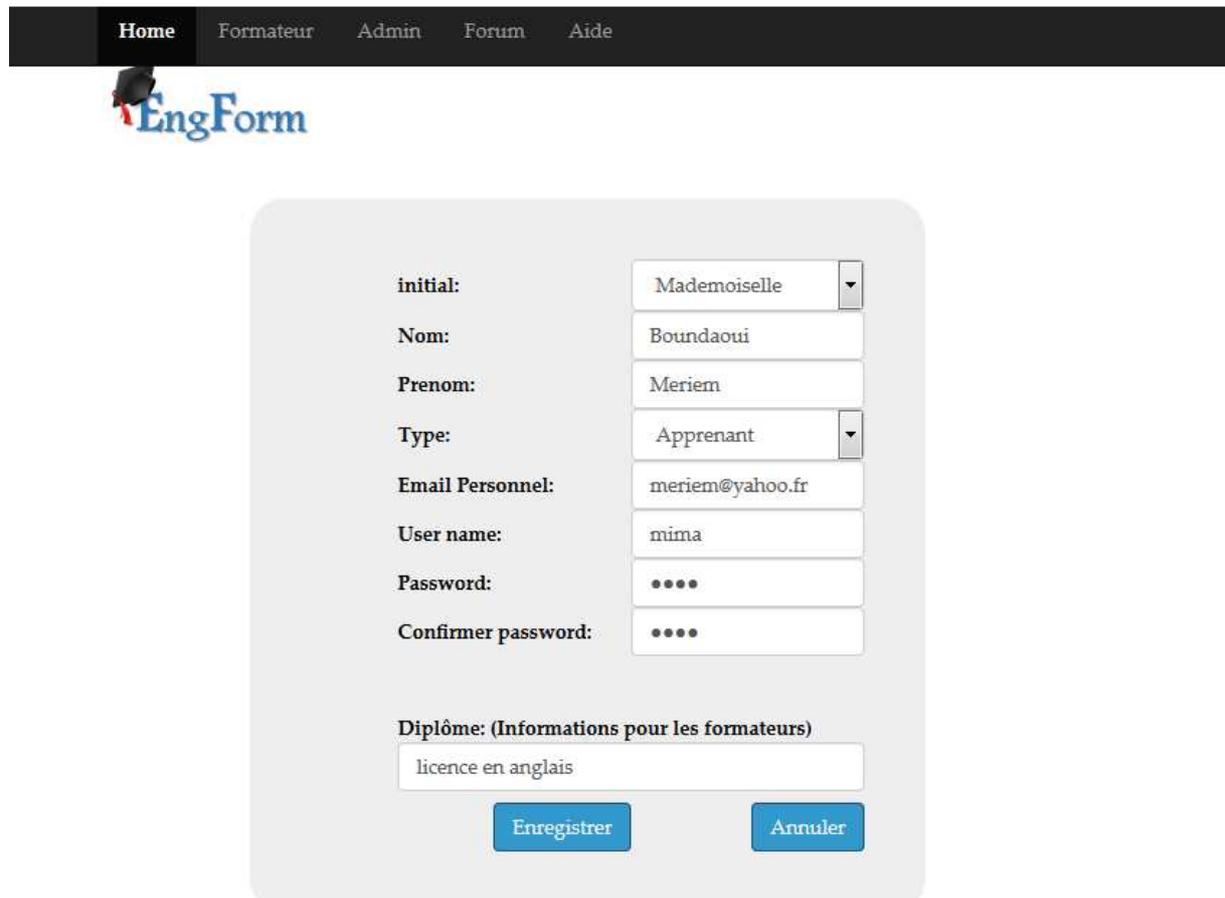


Figure 4.1 : Home page

➤ **La page inscription d'apprenant ou formateur :** cette page nous permet d'enregistrer des apprenants ou des formateurs



The screenshot shows the registration page for EngForm. At the top, there is a navigation bar with links for Home, Formateur, Admin, Forum, and Aide. Below the navigation bar is the EngForm logo. The main content area contains a registration form with the following fields:

- initial: Mademoiselle (dropdown menu)
- Nom: Boundaoui
- Prenom: Meriem
- Type: Apprenant (dropdown menu)
- Email Personnel: meriem@yahoo.fr
- User name: mima
- Password: ••••
- Confirmer password: ••••
- Diplôme: (Informations pour les formateurs) licence en anglais

At the bottom of the form, there are two buttons: Enregistrer and Annuler.

Figure 4.2 : Page ajout

➤ **Profil apprenant :** c'est la page où l'apprenant peut accéder aux différents cours et exercices et tests.



Figure4.3 : Page Profil apprenant

➤ **Page cours:** cette page permet à l'apprenant de voir tous les cours qui sont destinés à son niveau et les télécharger.

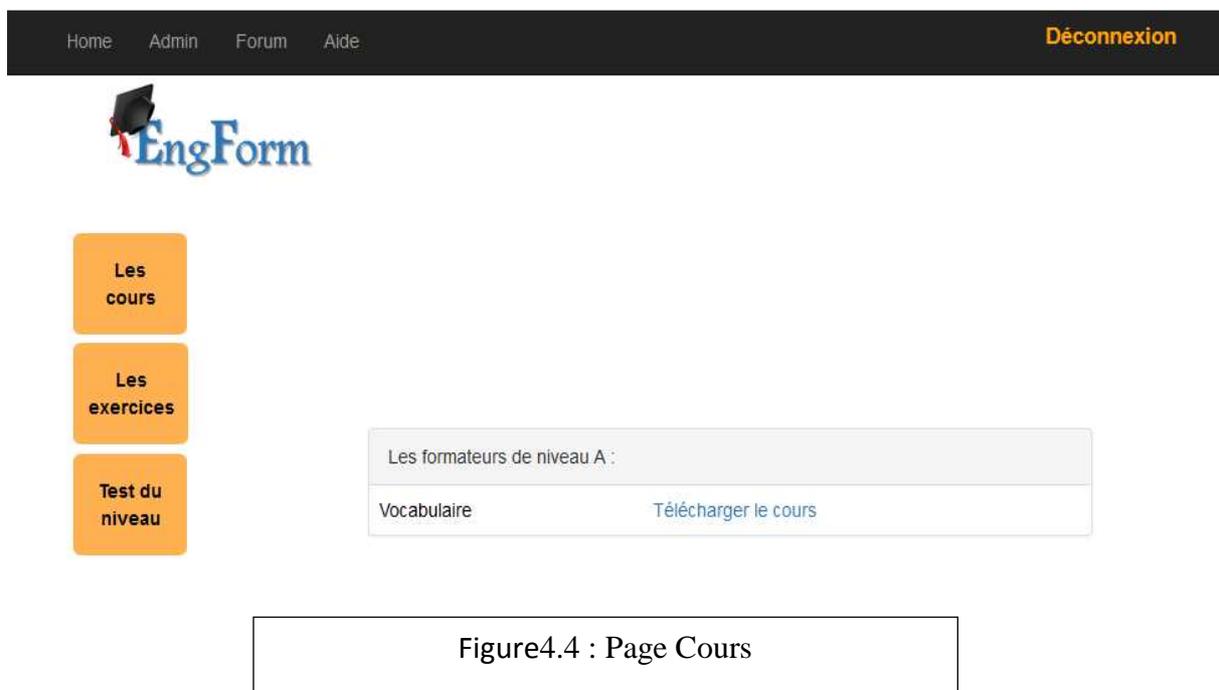


Figure4.4 : Page Cours

➤ **Page Home admin:** c'est de cette page que l'administrateur gère son site

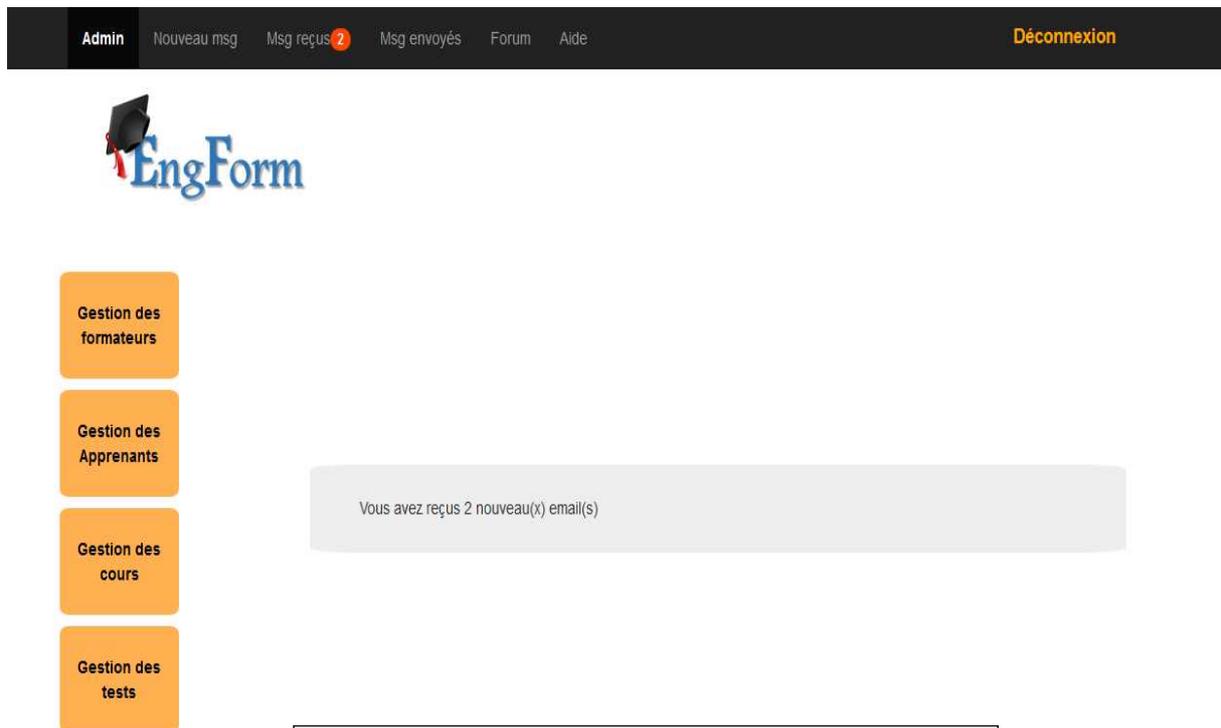
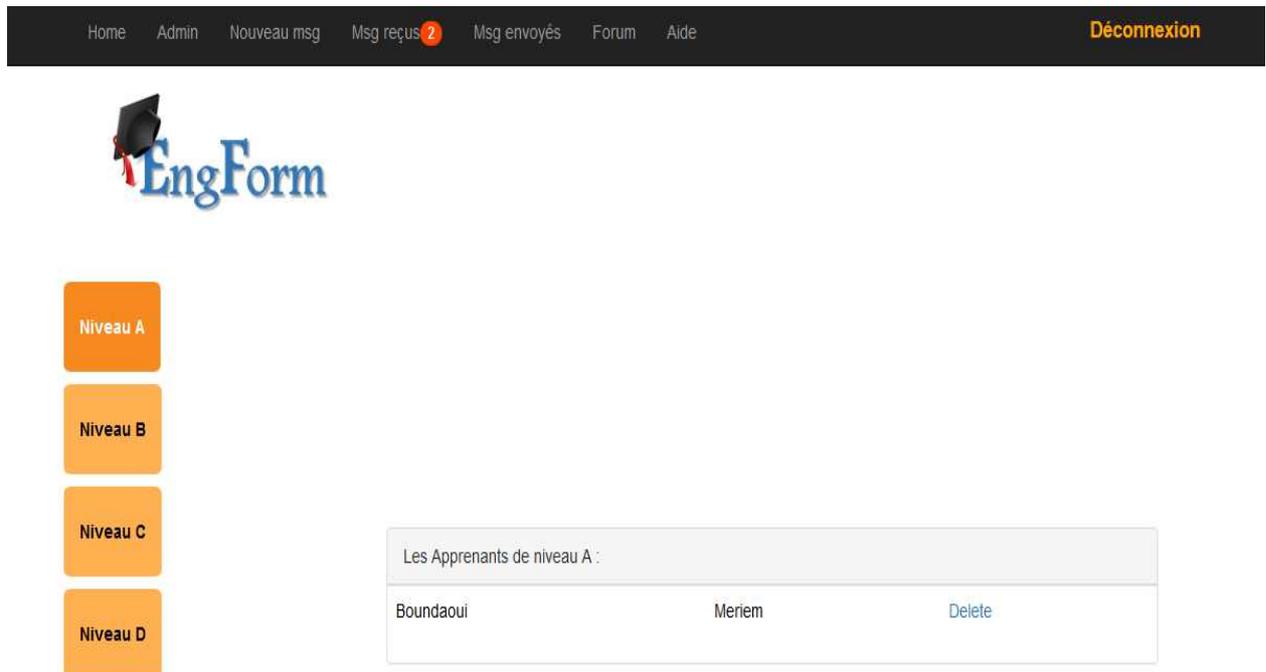


Figure 4.5 : Page home admin

➤ **Page gestion des apprenants** : cette page permet à l'administrateur de visualiser les informations des apprenants et de les supprimer.



The screenshot shows the EngForm application interface. At the top, there is a navigation bar with links for Home, Admin, Nouveau msg, Msg reçus (2), Msg envoyés, Forum, Aide, and Déconnexion. Below the navigation bar is the EngForm logo, which includes a graduation cap icon. On the left side, there are four orange buttons labeled Niveau A, Niveau B, Niveau C, and Niveau D. The main content area displays a table titled "Les Apprenants de niveau A :". The table has one row with the name "Boudaoui Meriem" and a "Delete" link.

Les Apprenants de niveau A :		
Boudaoui	Meriem	<a href="#">Delete</a>

Figure 4.6 : Gestion des apprenants

### 4.5. 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.

Nous avons présenté à travers ce travail une étude relative à la conception et la réalisation d'une application de formation à distance accessible depuis différents supports.

L'intégration de la technologie web dans le domaine de la formation à distance engendre d'importants avantages tels que : la possibilité de travailler dans des points géographiquement dispersés, ce qui permet à l'utilisateur d'accéder à l'information où il en a besoin et de gagner du temps sans se soucier du type du terminal d'accès. Ceci permet au domaine de l'e-Learning de devenir de plus en plus large qu'auparavant.

Ce travail nous a permis :

- ✓ De nous familiariser avec les technologies et approches de développement web multiplateforme.
- ✓ D'approfondir nos connaissances théoriques et pratiques en rapport avec les services web, les réseaux et les bases de données ...
- ✓ D'acquérir de nouvelles connaissances sur les langages HTML, Java EE, le langage de modélisation UML, et aussi d'utiliser plusieurs logiciels tels que, Netbeans 8 et le système de gestion de bases de données MySQL.
- ✓ D'avoir les éléments de base d'une nouvelle discipline qui ne cesse pas de se développer aussi bien dans le secteur public que dans le secteur privé : L'e-learning ou enseignement à distance.

L'application que nous avons réalisée correspond aux objectifs que nous nous étions fixés au départ. Ce projet étant notre première expérience, nous a permis de mettre en pratique les connaissances que nous avons acquies tout au long de notre cursus universitaire.

Nous espérons que notre travail pourra servir de référence aux prochaines promotions, ce projet reste ouvert à toute amélioration et extension notamment celle concernant :

- Enrichissement de la plateforme pédagogique avec les outils de collaboration (forum, chat ..).
- Ajouter une autre approche de développement multiplateforme (mobile).

## 1. Introduction

La modélisation est la conception d'un modèle, et ce dernier n'est en fait qu'une simplification de la réalité qui permet de mieux comprendre le système à développer.

## 2. Présentation d'UML [réf 15]

UML (Unified Modeling Language), traduisez "**langage de modélisation unifié**" n'est pas une méthode ni une description normative des étapes de la modélisation : les auteurs d'UML (GradyBooch, James Rumbaugh, et Ivar Jacobson), en effet estiment qu'il n'était pas opportun de définir une telle méthode en raison de la diversité des cas particuliers. Ils ont préférés être plus de modèles, et définir un langage graphique permettant de représenter et de communiquer les divers aspects d'un système d'information (aux graphiques sont bien sur associés des textes qui expliquent leur contenu).

## 3. Qu'est-ce que UML ?

UML est le facteur de standardisation, car il est impossible de prétendre imposer une méthode, une manière de faire, à toute l'industrie .En revanche on peut prétendre définir un langage de modélisation qui, s'il est suffisamment général, sera adopté comme moyen de communication.

UML est un langage qui permet de représenter des modèles, mais il ne définit pas le processus d'élaboration d'un modèle. Cependant, dans le cadre de la modélisation d'une application informatique, les auteurs d'UML préconisent d'utiliser une démarche :

- ❖ Itérative et incrémentale,
- ❖ Guidée par les besoins des utilisateurs du système,
- ❖ Centrée sur l'architecture logicielle.

## 4. À quoi sert UML ?

UML permet de définir et de visualiser un modèle, à l'aide de diagrammes. Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle ; c'est une perspective du modèle, pas "le modèle" lui-même. Chaque type de diagramme UML possède une structure (les types des éléments de Modélisation qui le composent sont prédéfinis).Un type de diagramme UML véhicule une sémantique précise (un type de diagramme Offre toujours la même vue d'un système). Combinés, les différents types de

diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système. Par extension et abus de langage, un diagramme UML est aussi un modèle (un diagramme modélise un aspect du modèle global).

## 5. Historique [réf 16]

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

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

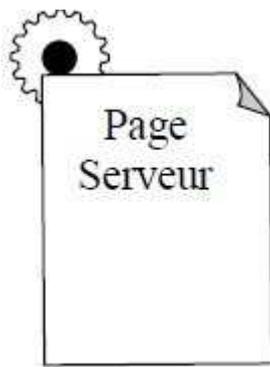
La version actuelle (depuis juin 1999) est UML 1.3 (la version 1.4 sera bientôt prête, afin de préparer la prochaine version 2.0).

## 6. Stéréotypes

### 6.1. Classe

#### 6.1.1. Page serveur « Server page »

##### Icône



##### Description

Représente une page Web possédant des scripts qui interagissent avec des ressources serveur telles que les bases de donnée, ces scripts sont exécuté par le serveur.

## Contraintes

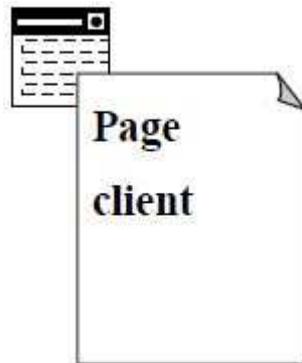
Les pages serveur ne peuvent avoir de relation qu'avec des objets sur le serveur.

## Étiquette

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

### 6.1.2. Page client « client page »

#### Icône



#### Description

Une instance d'une page client est une page Web formatée en HTML. Les pages clients peuvent contenir des scripts interprétés par les navigateurs lorsque celles-ci sont restituées par ces derniers. Les fonctions des pages clients correspondent aux fonctions des scripts de la page web.

#### Contrainte

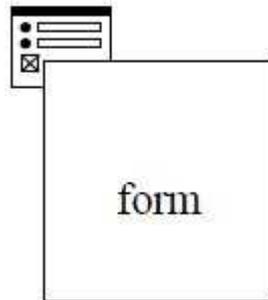
Aucune.

#### Étiquette :

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

### 6.1.3. Formulaire

#### Icône



#### Description :

Une classe stéréotypée « form » est un ensemble de champs de saisie faisant partie d'une page client. A une classe formulaire correspond une balise HTML `<form>`, les attributs de cette classe correspondent aux éléments de saisie d'un formulaire HTML (zone de saisie, zone de texte, boutons d'option.).

Un formulaire n'a pas d'opérations, puisqu'il peut les encapsuler. Toute opération qui interagit avec le formulaire appartient à la page qui la contient.

#### Contraintes :

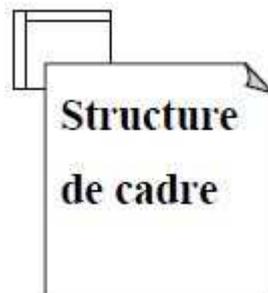
Aucune.

#### Étiquette :

GET ou POST – Méthodes utilisées pour soumettre les données à l'URL de l'attribut action de la balise HTML `<form>`.

### 6.1.4. Structure de cadres « frameset »

#### Icône



## Description

Une structure de cadres est un conteneur de plusieurs pages Web. La zone d'affichage rectangulaire est divisée en cadres rectangulaires inscrits. A chaque cadre peut être associé un nom unique de cible « Target ».

Le contenu d'un cadre peut être une page Web ou une structure de cadre.

Une classe stéréotypée « frameset » est directement associée à une structure de cadre de page Web par la balise HTML <frameset >.

Une structure de cadre est une page client qui peut posséder des opérations et des attributs.

## Contraintes

Aucune.

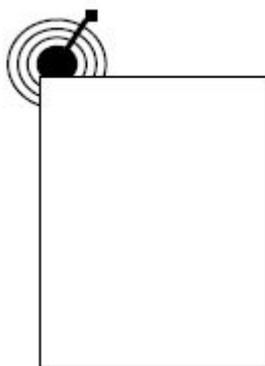
## Étiquette

Rangées (rows) : valeur de l'attribut rows de la balise HTML <frameset>. C'est une chaîne de pourcentages séparés par des virgules, définissant les hauteurs relatives des cadres.

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

### 6.1.5. Cible « target »

#### Icône



## Description

Une cible est une zone nommée dans la fenêtre du navigateur dans laquelle des pages Web peuvent être affichées. Le nom de la classe stéréotypée est celui de la cible. Habituellement, une cible est le cadre d'une structure de cadre définie dans une fenêtre ; cependant, une cible peut-être une toute nouvelle instance de navigateur : une fenêtre. Une association « targeted link » spécifie la cible où une page Web doit être affichée.

**Contraintes**

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

**Étiquette**

Aucune.

**6.1.6. Objet Java Script « Java Script object »****Icône :**

Aucune.

**Description :**

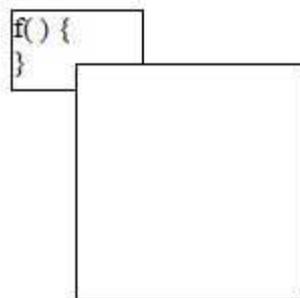
Sur un navigateur compatible Java Script, il est possible de simuler des objets personnalisés à l'aide de fonctions Java Script. Les objets Java Script ne peuvent exister que dans le contexte de pages client.

**Contraintes :**

Aucune.

**Étiquette :**

Aucune.

**6.1.7. Objet ScriptClient « ClientScript object »****Icône****Description :**

Un Objet Script Client est un ensemble qui regroupe des scripts client particuliers dans un fichier. Lequel est inclus dans une requête distincte du navigateur client. Ces objets regroupent souvent des lots de fonctions couramment utilisées au travers d'une application ou d'une entreprise.

**Contraintes**

Aucune.

**Étiquette**

Aucune.

**6.2. Association****6.2.1. Lien « link »****Icône**

Aucune.

**Description**

Un lien est un pointeur d'une page client vers une autre page. Dans un diagramme de classes, un lien est une association entre une page client et une autre page client ou une page serveur.

A un lien correspond une balise ancre HTML.

**Contraintes**

Aucune.

**Étiquette**

Paramètres (paramètres) : liste de noms de paramètres qui doivent être passés avec la demande de la page liée.

**6.2.2. Lien cible « targeted link »****Description**

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

**Icône**

Aucune.

**Contraintes**

Aucune.

**Étiquette**

Paramètres (Paramètres) : liste de noms de paramètres qui doivent être passés avec la demande de la page liée.

Nom de la cible (targetname) : nom de la cible ou la page vers laquelle pointe le lien qui doit être affichée.

### 6.2.3. Contenu de cadre « frame content »

#### Icône

Aucune.

#### Description

Une association contenue de cadre est une association d'agrégation qui traduit l'appartenance d'une page ou d'une cible à un cadre.

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

#### Contraintes

Aucune.

#### Étiquette

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

**Colonne (Col) :** entier qui indique la colonne du cadre dans la structure de cadre auquel appartient la page, ou la cible associée.

### 6.2.4. Soumet « submit »

#### Icône :

Aucune.

#### Description

Submit est une association qui se trouve toujours entre un formulaire et une page serveur. Les formulaires soumettent les valeurs de leurs champs au serveur, par l'intermédiaire de pages serveur, pour qu'il les traite. Le serveur Web traite la page serveur, qui accepte et utilise les informations du formulaire.

#### Contraintes

Aucune.

#### Étiquette

**Paramètres (parameters) :** une liste de noms de paramètres qui doivent être passés avec la demande de la page liée.

### 6.2.5. Construit « build »

#### Icône

Aucune.

**Description**

La relation « build » est une relation particulière qui fait le pont entre les pages client et les pages serveur. L'association « build » identifie quelle page serveur est responsable de la création d'une page client. C'est une relation orientée, puisque la page client n'a pas connaissance de la page qui est à l'origine de son existence. Une page serveur peut construire plusieurs pages client, en revanche, une page client ne peut être construite que par une seule page serveur.

**Contraintes**

Aucune.

**Étiquette**

Aucune.

**6.2.6. Redirige « redirect »****Icône**

Aucune

**Description**

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

**Contraintes**

Aucune.

**Étiquette**

Délai (delay) : délai que doit observer une page client avant de rediriger vers la pagedestination. Cette valeur correspond à l'attribut Content de la balise <META>.

**6.2.7. IIOP (Internet Inter-ORB Protocol)****Description :**

IIOP est un type spécial de relation entre objets sur le client et objets sur le serveur. IIOP est un mécanisme de communication client/serveur différent de HTTP. On pourra trouver ce type de relation entre des JavaBeans sur le client et des EJB (Entreprise JavaBeans) sur le serveur.

**Icône**

Aucune.

**Contraintes**

Aucune.

### Étiquette

Aucune.

## 6.2.8. RMI (RemoteMéthode Invocation)

### Icône

Aucune.

### Description

RMI est un mécanisme qui permet à des applets Java, ou à des JavaBeans, d'envoyer des messages à des JavaBeans situés sur d'autres machines.

On pourra trouver ce type de relation entre des JavaBeans, ou des applets, sur le client et des EJB (Entreprise JavaBeans) sur le serveur.

### Contraintes

Aucune.

### Étiquette

Aucune.

## 6.3. Attribut

### 6.3.1. Élément de saisie « input element »

#### Icône

Aucune.

#### Description

Un élément de saisie correspond à la balise <input> d'un formulaire HTML... Les étiquettes associées à cet attribut stéréotypé, correspondent aux attributs de la balise <input>.

Les attributs obligatoires de la balise HTML <input> sont renseignés de la manière suivante :

L'attribut name prend la valeur du nom de l'élément de saisie et l'attribut value prend celle de sa valeur initiale.

#### Contraintes

Aucune.

#### Étiquette

**Type (Type) :** Le type de l'élément de saisie : texte, numérique, mot de passe, case à cocher, bouton d'option, bouton SUBMIT ou bouton RESET.

**Taille (size) :** définit la largeur visible allouée à l'écran en caractères.

- ❖ **Longueur Max (Max length)** : nombre maximal de caractères que peut saisir l'utilisateur.

### 6.3.2. Sélection d'éléments « select element »

#### Icône

Aucune.

#### Description

Contrôle de saisie employé dans les formulaires, il permet à l'utilisateur de sélectionner une ou plusieurs valeurs dans une liste. La plupart des navigateurs restituent ce contrôle par une liste d'options ou une liste déroulante.

#### Contraintes :

Aucune.

#### Étiquettes

- ❖ **Taille (Size)** : définit le nombre d'éléments qui doivent être affichée simultanément.
- ❖ **Multiple (Multiple)** : valeur booléenne qui indique que plusieurs éléments peuvent être sélectionnés conjointement.

### 6.3.3. Zone de texte « texte area element »

#### Icône

Aucune.

#### Description

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

#### Contraintes

Aucune.

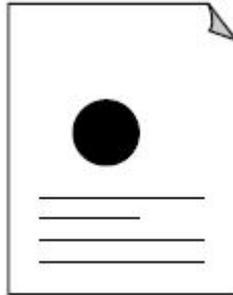
#### Étiquette

- ❖ **Ligne (rows)** : nombre de lignes de texte visibles.
- ❖ **Colonnes (cols)** : largeur visible du texte en largeurs de caractères moyennes.

## 6.4. Composant

### 6.4.1. Page web « web page »

#### Icône



#### Description

Un composant page est une page web. Il peut être requis d'après son nom par un navigateur.

Un composant page peut contenir des scripts client ou serveur.

Le plus souvent, le composant page est un fichier texte accessible au serveur Web, mais il peut également être un module compilé, chargé et exécuté par le serveur web. Dans les deux cas, le serveur Web produit, à partir du composant page, un document au format HTML, qui est renvoyé en réponse à la requête du navigateur.

#### Contraintes

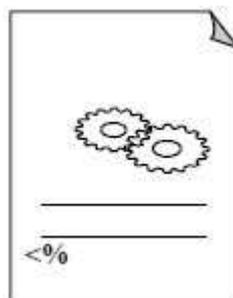
Aucune

#### Étiquette

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

### 6.4.2-Page ASP « ASP page »

#### Icône



**Description**

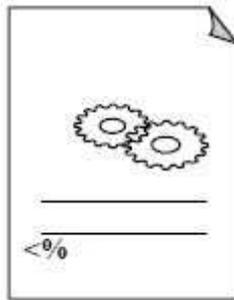
Une page ASP est une page Web qui implémente du code ASP coté serveur. Ce stéréotype n'est pertinent que dans un environnement d'application basé sur les ASP (Active Server Page) de Microsoft.

**Contraintes**

Aucune.

**Étiquette**

Identique à celles de la page web.

**II.4.3. Page JSP « JSP page »****Icône****Description**

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

**Contraintes**

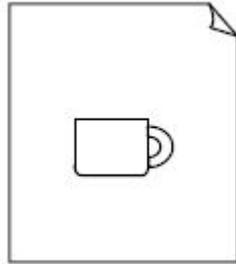
Aucune.

**Étiquette**

Identique à celles de la page web.

#### 6.4.4. Servlet « Servlet »

##### Icône



##### Description

Une servlet est un composant Java. Ce stéréotype n'est pertinent que dans un environnement d'application compatible avec les servlets de Sun.

##### Contraintes

Aucune.

##### Étiquette

Identique à celles de la page web.

#### 6.4.5. Bibliothèque de scripts « script library »

##### Icône



##### Description

Une bibliothèque de scripts est un composant qui propose un ensemble de sous-routines. Ou de fonctions, pouvant être inclus à d'autres composants pages web.

##### Contraintes

Aucune.

##### Étiquette

Identique à celles de la page web.

## 7. Règles de cohérence sémantique

### ❖ Réalisation de composant

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

### ❖ Généralisation

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

### ❖ Association

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

A \ De	« client page »	« server page »	« frameset »	« target »	« form »
« client page »	« link » « redirect » « targeted link »	« link » « redirect » « targeted link »	« link » « redirect » « targeted link »	« dependenc »	« aggregation »
« server page »	« build » « redirect »	« redirect »	« build » « redirect »		
« frameset »	« frame content »		« frame content »	« frame conte »	
« target »					
« Form »	« aggregated by »	« submit »			

Figure 5.1: Combinaisons valides d'associations de stéréotypes

### Référence Webographique

[réf 1] : [https://fr.wikipedia.org/wiki/Service\\_web](https://fr.wikipedia.org/wiki/Service_web)

[réf 2]: <http://www.jmdoudoux.fr/java/dej/chap-service-web.htm>

[réf 3]: <https://netbeans.org/kb/docs/websvc/rest.html>

[réf4]:<http://mbaron.developpez.com/tutoriels/soa/developpement-services-web-jaxws-netbeans/>

[réf 5] : <http://www.abricocotier.fr/4224-faire-un-service-web-avec-netbeans-et-glassfish-le-tutoriel>

[réf 6]: <http://www.jmdoudoux.fr/java/dej/chap-service-web.htm>

[réf 7]: <https://openclassrooms.com/courses/les-services-web>

[réf 8] : <http://www.objjis.com/formation-java/comprendre-web-services-architecture-wsdl-uddi-soap-soa.html>

[réf 9] :[https://fr.wikipedia.org/wiki/Interop%C3%A9rabilit%C3%A9\\_en\\_informatique](https://fr.wikipedia.org/wiki/Interop%C3%A9rabilit%C3%A9_en_informatique)

[réf 10] :<https://msdn.microsoft.com/fr-fr/library/bb469919.aspx>

[réf 11] : <http://www.paraclub-arcachon.com/conforme.html>

[réf 12] : <http://e-learning.prestataires.com/>

[réf 13] :[https://fr.wikipedia.org/wiki/Formation\\_en\\_ligne](https://fr.wikipedia.org/wiki/Formation_en_ligne)

[réf 14 ] : <http://eduscol.education.fr/numerique/dossier/archives/metadata/normes-et-standards>

[réf 15] : <https://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml>

[réf 16] : [https://fr.wikipedia.org/wiki/UML\\_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

[réf 17] : <http://www.commentcamarche.net/contents/1013-le-modele-relationnel>