

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

**Ministère de l'Enseignement Supérieur et de la Recherche**

**Scientifique**

**Université Mouloud Mammeri de Tizi-Ouzou**



**Département Informatique**

**Projet de Fin d'études  
En vue de l'obtention du diplôme  
de Master en Informatique**

**Option : Ingénierie des Systèmes d'Information**

**Thème**

**Web Service de type SOAP pour l'interopérabilité des applications**

**E- learning**

**Elaboré par :**

**Ait Abdelmalek Sara**

**Touatioui Thinhinane**

**Proposé et Encadré par :**

**Mr Kerbiche**

**2017/2018**

## *Remerciements*

*Tout d'abord, nous remercions le bon Dieu de nous avoir donné le courage ainsi que la volonté pour la réalisation de ce projet.*

*Nous tenons à remercier notre promoteur pour ses critiques constructives, ses explications et suggestions pertinentes.*

*Nous exprimons à présent notre gratitude aux membres du jury pour avoir accepté d'évaluer et de juger notre modeste et humble travail.*

## *DEDICACES*

*Je dédie ce modeste travail à mes très chers parents qui m'ont tant encouragé tout en long de son élaboration, à mes chers frères, à ma chère amie Jasmine, à mes camarades de ma promotion et enfin à ma collègue exceptionnelle et très chère amie « Thinhinane » ainsi que sa famille.*



*Ait Abdelmalek Sara*

## *DEDICACES*

*Je dédie ce modeste travail à mes très chers parents, à mes chers frères ainsi qu'à ma sœur, à mes camarades de ma promotion et enfin à ma collègue exceptionnelle et très chère amie « Sara » ainsi que sa famille.*



*Touatioui Thinfinane*

## Résumé

En éducation, jusqu'au début des années 2000, l'enseignement se faisait dans une classe d'élèves avec un enseignant qui dirigeait le processus. Or, De nos jours, une nouvelle forme d'apprentissage a fait son apparition ; cette dernière est : « l'apprentissage en ligne ».

Le sujet abordé dans ce mémoire est l'interopérabilité dans les systèmes « e-learning », ce dernier est une industrie en pleine croissance : Il suffit d'un bon outil d'apprentissage en ligne pour que l'éducation soit facilitée pratiquement partout. En effet, la technologie a tellement progressé que le fossé géographique est comblé par l'utilisation d'outils qui vous donnent l'impression d'être dans une salle de classe traditionnelle.

En fait, l'objectif de ce présent travail n'est pas le développement et la mise en place d'un dispositif e-learning complet, mais plutôt d'implémenter, or d'intégrer une solution à certaines fonctionnalités qui doivent garantir un aspect technique, à savoir : **L'interopérabilité** ; auquel les systèmes e-learning sont confrontés et dont la satisfaction reste une nécessité. Cette solution consiste en l'intégration des **Services Web**. Ces derniers comportent deux types, à savoir : *SOAP* et *REST*, or notre travail est focalisé sur les Services Web de type **SOAP**.

**Mots clés** : éducation, e-learning, apprentissage en ligne, interopérabilité, Services Web, Services Web SOAP.

## ***Abstract***

In education, until the early 2000s , teaching was done in a class of students with a teacher leading the process. Nowadays, a new form of learning has appeared ; the latter is : "online learning".

The subject discussed in this memory is interoperability in e-learning systems, the latter is a growing industry : A good e-learning tool is enough to make education easier everywhere. In fact, technology has advanced so much that the geographic gap is filled by the use of tools that make you feel like you're in a traditional classroom.

In fact, the objective of this present work is not the development and the implementation of a complete e-learning device, but rather to implement, or to integrate a solution to certain functionalities that must guarantee an aspect technical, namely : interoperability ; that e-learning systems are facing and whose satisfaction remains a necessity. This solution is the integration of Web Services. There are two types of these, SOAP and REST, but our work is focused on SOAP Web Services.

**Keywords** : education, e-learning, e-learning, interoperability, web services, SOAP web services.

# SOMMAIRE

<b>Introduction générale</b> .....	1
------------------------------------	---

## **Chapitre I : Services Web et Interopérabilité**

<b>I-1- Introduction</b> .....	3
<b>I-2- Style d'architecture dans les systèmes informatiques</b> .....	3
<b>I-3- Architecture Orientée Services (SOA)</b> .....	4
I-3-1- Définition d'un service.....	4
I-3-2- Orchestration des services.....	4
I-3-3- Propriétés des services .....	5
I-3-4- Définition du style d'Architecture Orientée Service.....	5
I-3-5- Microservices :.....	6
I-3-6- SOA Vs Microservices : .....	6
I-3-7- Avantages de l'Architecture Orientée Services .....	6
<b>I-4- Les Services Web</b> .....	7
I-4-1- Définition d'un service web .....	7
I-4-2- Pourquoi les services web ?.....	8
I-4-4- Les standards utilisés par les services Web .....	10
I-4-5- Fonctionnement des services web.....	11
I-4-6- Types de services web .....	11
<b>I-5- Les services web de type SOAP</b> .....	12
I-5-1- Architecture .....	12
I-5-2- Protocole SOAP.....	12
I-5-3-Principes SOAP.....	13

<i>I-5-4- Structure d'un message SOAP .....</i>	<i>13</i>
<i>I-5-5- En- tête, Corps (requête/réponse) d'un message SOAP .....</i>	<i>14</i>
<i>I-5-5-1- L'enveloppe [envelop] .....</i>	<i>14</i>
<i>I-5-5-2- L'en-tête [header].....</i>	<i>14</i>
<i>I-5-5-3- Le corps [body] .....</i>	<i>15</i>
<i>I-5-6- L'encodage des messages SOAP.....</i>	<i>16</i>
<b><i>I-6- XML .....</i></b>	<b><i>18</i></b>
<i>I-6-1- XML- RPC .....</i>	<i>18</i>
<i>I-6-2- Utilisation de SOAP dans http .....</i>	<i>19</i>
<i>I-6-3- Utilisation de SOAP pour les appels de procédure à distance (RPC) .....</i>	<i>20</i>
<i>I-6-4- SOAP de style RPC .....</i>	<i>20</i>
<i>I-6-5- SOAP de style document .....</i>	<i>21</i>
<b><i>I-7- SOAP Vs RMI, DCOM et CORBA .....</i></b>	<b><i>22</i></b>
<b><i>I-8- Les avantages de SOAP .....</i></b>	<b><i>22</i></b>
<b><i>I-9- Les inconvénients de SOAP .....</i></b>	<b><i>23</i></b>
<b><i>I-10- Les problèmes liés à SOAP .....</i></b>	<b><i>23</i></b>
<b><i>I-11- WS-Security .....</i></b>	<b><i>24</i></b>
<b><i>I-12- WSDL.....</i></b>	<b><i>25</i></b>
<b><i>I-13- UDDI.....</i></b>	<b><i>26</i></b>
<b><i>I-14- Les services web de type REST .....</i></b>	<b><i>26</i></b>
<i>I-14-1- Définition .....</i>	<i>26</i>
<i>I-14-2- Utilisation de REST.....</i>	<i>28</i>
<i>I-14-3- Caractéristiques de REST .....</i>	<i>28</i>
<i>I-14-4- WADL.....</i>	<i>28</i>
<i>I-14-5- Les avantages de REST .....</i>	<i>28</i>
<i>I-14-6- Les inconvénients de REST.....</i>	<i>29</i>



<i>I-14-7- Le débat SOAP VS REST .....</i>	<i>30</i>
<b><i>I-15- Les avantages des services Web .....</i></b>	<b><i>31</i></b>
<b><i>I-16- Les inconvénients des services Web .....</i></b>	<b><i>31</i></b>
<b><i>I-17- Sécurité des services Web .....</i></b>	<b><i>31</i></b>
<b><i>I-18- Interopérabilité .....</i></b>	<b><i>33</i></b>
<i>I-18-1- Définition .....</i>	<i>33</i>
<i>I-18-2- Interopérabilité et services web .....</i>	<i>33</i>
<b><i>I-19- Conclusion.....</i></b>	<b><i>33</i></b>

## ***Chapitre II : Description d'un système e-learning***

<b><i>II-1- Introduction .....</i></b>	<b><i>34</i></b>
<b><i>II-2- e-Learning.....</i></b>	<b><i>34</i></b>
<i>II-2-1- Définition .....</i>	<i>34</i>
<i>II-2-2- L'histoire du e-learning.....</i>	<i>35</i>
<i>II-2-3- Les aspects de la formation en ligne (e-learning) .....</i>	<i>35</i>
<i>II-2-3-1- Formation exclusivement en ligne (sans tutorat).....</i>	<i>35</i>
<i>II-2-3-2- Formation exclusivement en ligne (avec tutorat) .....</i>	<i>36</i>
<i>II-2-4- Les outils de communication et de collaboration du e-learning .....</i>	<i>38</i>
<i>II-2-4-1- les outils synchrones.....</i>	<i>38</i>
<i>II-2-4-1- les outils asynchrones .....</i>	<i>39</i>
<i>II-2-5- Avantages et Inconvénients du e-learning de point de vue utilisateurs .....</i>	<i>40</i>
<i>II-2-5-1- Du côté de l'Apprenant .....</i>	<i>40</i>
<i>II-2-5-2- Du côté du Formateur .....</i>	<i>41</i>
<i>II-2-5-3- Du côté de l'entreprise .....</i>	<i>41</i>
<i>II-2-6- Avantages et Inconvénients du e-learning de manière générale.....</i>	<i>42</i>

<i>II-2-6-1-2- Inconvénients .....</i>	<i>42</i>
<i>II-2-7- Les standards du e-learning .....</i>	<i>43</i>
<i>II-2-8- Plate-forme e-learning .....</i>	<i>44</i>
<i>II-2-8- 2- Principaux critères que doivent vérifier les plates-formes e-learning.....</i>	<i>45</i>
<i>II-2-9- LMS ou système de gestion des cours et de l'apprentissage .....</i>	<i>46</i>
<i>II-2-9-1- Présentation .....</i>	<i>46</i>
<i>II-2-9-2- Composants.....</i>	<i>46</i>
<i>    II-2-9-2-1- Outils pédagogiques.....</i>	<i>46</i>
<i>    II-2-9-2-2- Outils de communication.....</i>	<i>46</i>
<i>    II-2-9-2-3- Principales plates-formes LMS .....</i>	<i>48</i>
<i>    II-2-9-2-4- Besoins des utilisateurs de la plate-forme LMS.....</i>	<i>49</i>
<i>II-2-9-3- Objectifs de la plate-forme LMS.....</i>	<i>50</i>
<i>II-2-9-4- Quelques fonctionnalités de la plate-forme LMS.....</i>	<i>50</i>
<i>II-2-10- Situation du e-learning en Algérie .....</i>	<i>51</i>
<i>II-2-10-1- Le développement de l'e-learning en Algérie .....</i>	<i>51</i>
<i>II-2-10-2- Exemple de plate-forme e-learning en Algérie.....</i>	<i>52</i>
<i>II-2-11- Situation du e-learning dans le monde.....</i>	<i>53</i>
<i>II-2-12- Une architecture de services Web pour l'apprentissage en ligne.....</i>	<i>55</i>
<b><i>II-3- Conclusion.....</i></b>	<b><i>57</i></b>

### **Chapitre III : Conception et Modélisation**

<b><i>III-1- Introduction.....</i></b>	<b><i>58</i></b>
<b><i>III-2- UML.....</i></b>	<b><i>58</i></b>
<i>    III-2-1- Définition .....</i>	<i>58</i>
<i>    III-2-2- Types de diagrammes UML .....</i>	<i>58</i>
<b><i>III-3- Problématique du système étudié.....</i></b>	<b><i>59</i></b>
<b><i>III-4- Objectif du projet .....</i></b>	<b><i>59</i></b>

<b>III-5- Partie Analyse .....</b>	<b>60</b>
III-5-1- Identification des acteurs .....	60
III-5-2- Identification des besoins .....	60
III-5-2-1- l'Administrateur .....	60
III-5-2-2- l'Apprenant .....	60
III-5-2-3- Le Formateur .....	61
III-5-3- Identification des cas d'utilisation .....	61
III-5-3-1- Administrateur.....	62
III-5-3-1-1- Diagramme des cas d'utilisation.....	62
III-5-3-1-2- Description textuelle du cas d'utilisation « S'authentifier ».....	63
III-5-3-2- Apprenant.....	64
III-5-3-2-1- Diagramme des cas d'utilisation.....	64
III-5-3-2-2- Description textuelle du cas d'utilisation « Modifier compte ».....	65
III-5-3-3- Formateur.....	66
III-5-3-3-1- Diagramme des cas d'utilisation.....	66
III-5-3-3-2- Description textuelle du cas d'utilisation « Gestion cours ».....	67
III-5-4- Identification des services web .....	68
<b>III-6- Partie Conception.....</b>	<b>68</b>
III-6-1- Diagrammes de séquences.....	69
III-6-1-1- Diagramme de séquence du cas d'utilisation « S'authentifier ».....	69
III-6-1-2- Modèle d'utilisation du service web SOAP correspondant .....	70
III-6-1-3- Diagramme de séquence du cas d'utilisation « modifier Compte » .....	71
III-6-1-4- Modèle d'utilisation du service web SOAP correspondant .....	72
III-6-1-5- Diagramme de séquence du cas d'utilisation « Ajouter Cours » .....	73
III-6-1-6- Modèle d'utilisation du service web SOAP correspondant .....	74
III-6-1-7- Diagramme de séquence du cas d'utilisation « Se déconnecter » .....	75
III-6-1-8- Modèle d'utilisation du service web correspondant .....	75

<i>III-6-2- Diagramme de classe général .....</i>	<i>76</i>
<b>III-7- Conclusion.....</b>	<b>77</b>

## **Chapitre IV : Réalisation**

<b>IV-1- Introduction.....</b>	<b>78</b>
<b>IV-2- Environnement et outils de développement et d'implémentation .....</b>	<b>78</b>
<i>IV-2-1- Java Enterprise Edition (JEE) .....</i>	<i>78</i>
<i>IV-2-2- Enterprise JavaBeans (EJB).....</i>	<i>79</i>
<i>IV-2-3- GlassFish.....</i>	<i>80</i>
<i>IV-2-4- Java Server Faces (JSF) .....</i>	<i>80</i>
<i>IV-2-5- PrimeFaces .....</i>	<i>80</i>
<i>IV-2-6- JPQL.....</i>	<i>81</i>
<i>IV-2-6-1- JPA .....</i>	<i>81</i>
<i>IV-2-6-2- Présentation de JPQL.....</i>	<i>81</i>
<b>IV-3- Modèle physique de données.....</b>	<b>81</b>
<i>IV-3-1- Table Formateur.....</i>	<i>81</i>
<i>IV-3-2- Table Apprenant.....</i>	<i>82</i>
<i>IV-3-3- Table Compte.....</i>	<i>82</i>
<i>IV-3-4- Table Cours.....</i>	<i>83</i>
<i>IV-3-4- Table Cours-Apprenant .....</i>	<i>83</i>
<i>IV-3-4- Table Chapitre .....</i>	<i>83</i>
<b>IV-4- Présentation des Interfaces résultantes suivi des Requêtes / Réponses SOAP .....</b>	<b>84</b>
<i>IV-4- 1- Une fenêtre pour l'Authentification .....</i>	<i>84</i>
<i>IV-4-2- Test Requête et Réponse SOAP.....</i>	<i>85</i>

<i>IV-4- 3- Fenêtre d'accueil.....</i>	<i>86</i>
<i>IV-4- 4- Page de Profil.....</i>	<i>87</i>
<i>IV-4- 5- Test Requête et Réponse SOAP.....</i>	<i>87</i>
<i>IV-4- 6- Page du Compte utilisateur.....</i>	<i>88</i>
<i>IV-4- 7- Test Requête et Réponse SOAP.....</i>	<i>89</i>
<i>IV-4-8- Test Requête et Réponse SOAP.....</i>	<i>90</i>
<i>IV-4-9- Page des Cours.....</i>	<i>91</i>
<i>IV-4-10- Test Requête et Réponse SOAP.....</i>	<i>92</i>
<i>IV-4-11- Test Requête et Réponse SOAP.....</i>	<i>93</i>
<i>IV-4- 12- Test Requête et Réponse SOAP.....</i>	<i>94</i>
<i>IV-4- 13- Test Requête et Réponse SOAP.....</i>	<i>95</i>
<i>IV-4- 14- Page des Chapitres.....</i>	<i>96</i>
<i>IV-4- 15- Page des cours et chapitres (coté Apprenant).....</i>	<i>97</i>
<i>IV-4-16- Test Requête et Réponse SOAP.....</i>	<i>98</i>
<b><i>IV-5- Conclusion.....</i></b>	<b><i>99</i></b>

## ***Conclusion***

<b><i>Conclusion générale.....</i></b>	<b><i>100</i></b>
--	-------------------

## ***Références***

<b><i>Références.....</i></b>	<b><i>101</i></b>
-------------------------------	-------------------

## Liste des figures

Figure I.1 : Orchestration de services .....	4
Figure I.2 : Fonctionnement des services web.....	11
Figure I.3 : Architecture des services web de type SOAP.....	12
Figure I.4 : Format d'un message SOAP. ....	14
Figure I.5 : Communication entre un client et un serveur en RPC. ....	19
Figure II.1 : Architecture de haut niveau de LearnServe. ....	56
Figure III.1 : Diagramme des cas d'utilisation associés à l'Administrateur. ....	62
Figure III.2 : Diagramme des cas d'utilisation associés à l'Apprenant. ....	64
Figure III.3 : Diagramme des cas d'utilisation associés au Formateur. ....	66
Figure III.4 : Diagramme de séquence du cas d'utilisation "S'authentifier".....	69
Figure III.5 : utilisation du service web de type SOAP "Authentification". ....	70
Figure III.6 : Diagramme de séquence du cas d'utilisation « Modifier un compte ». ....	71
Figure III.7: utilisation du service web de type SOAP "CompteWS". ....	72
Figure III.8 : diagramme de séquence du cas d'utilisation « Ajouter un cours ».....	73
Figure III.9 : utilisation du service web de type SOAP « CoursWS ». ....	74
Figure III.10 : Diagramme de séquence du cas d'utilisation « Se déconnecter ». ....	75
Figure III.11 : utilisation du service web de type SOAP " AuthentificationWS ". ....	75
Figure III.12 : Diagramme de classe correspondant.....	76
Figure IV.1 : Authentification apprenant/formateur. ....	84
Figure IV.2 : Requête SOAP associée à l'authentification (Login). ....	85
Figure IV.3: Réponse SOAP associée à l'authentification (Login). ....	85
Figure IV.4: Page d'accueil. ....	86
Figure IV.5 : Page de Profil. ....	87
Figure IV.6 : Requête SOAP associée à Profil.....	87
Figure IV.7 : Réponse SOAP associée à Profil. ....	88
Figure IV.8 : Page du Compte utilisateur.....	88

<i>Figure IV.9: Requête SOAP associée à l'obtention du compte en question.....</i>	<i>89</i>
<i>Figure IV.10 : Réponse SOAP associée à l'obtention du compte correspondant. ....</i>	<i>89</i>
<i>Figure IV.11: Requête SOAP associée à la modification du mot de passe du compte .....</i>	<i>90</i>
<i>Figure IV.12: Réponse SOAP associée à la modification du mot de passe du compte .....</i>	<i>90</i>
<i>Figure IV.13: Page des Cours du formateur.....</i>	<i>91</i>
<i>Figure IV.14 : Ajouter un cours donné. ....</i>	<i>91</i>
<i>Figure IV.15 : Requête SOAP associée à l'obtention des cours du formateur concerné. ....</i>	<i>92</i>
<i>Figure IV.16 : Réponse SOAP associée à l'obtention des cours du formateur concerné. ....</i>	<i>92</i>
<i>Figure IV.17 : Requête SOAP associée à l'ajout d'un cours donné. ....</i>	<i>93</i>
<i>Figure IV.18 : Réponse SOAP associée à l'ajout d'un cours donné. ....</i>	<i>93</i>
<i>Figure IV.19 : Requête SOAP associée à la modification d'un cours donné. ....</i>	<i>94</i>
<i>Figure IV.20 : Réponse SOAP associée à la modification d'un cours donné. ....</i>	<i>94</i>
<i>Figure IV.21 : Requête SOAP associée à la suppression d'un cours donné. ....</i>	<i>95</i>
<i>Figure IV.22 : Réponse SOAP associée à la suppression d'un cours donné. ....</i>	<i>95</i>
<i>Figure IV.23 : Page des chapitres associés au cours pris en considération.....</i>	<i>96</i>
<i>Figure IV.24 : Ajout d'un chapitre à un cours donné.....</i>	<i>96</i>
<i>Figure IV.25 : Page des cours d'un apprenant.....</i>	<i>97</i>
<i>Figure IV.26 : Page des chapitres associés à un cours d'un apprenant.....</i>	<i>97</i>
<i>Figure IV.27 : Requête SOAP associée à l'obtention des cours de l'apprenant concerné.....</i>	<i>98</i>
<i>Figure IV.28 : Réponse SOAP associée à l'obtention des cours de l'apprenant concerné.....</i>	<i>98</i>
<i>Figure IV.29 : Téléchargement d'un chapitre donné par l'apprenant concerné.....</i>	<i>99</i>

## Liste des tableaux

<i>Tableau I.1 : Comparatif entre SOA et Microservices. ....</i>	<i>6</i>
<i>Tableau I.2 : SOAP Vs RMI, DCOM et CORBA. ....</i>	<i>22</i>
<i>Tableau II.1 : Avantages et Inconvénients coté Apprenant.....</i>	<i>40</i>
<i>Tableau II.2 : Avantages et Inconvénients coté Formateur.....</i>	<i>41</i>
<i>Tableau II.3 : Avantages et Inconvénients coté Entreprise. ....</i>	<i>41</i>
<i>Tableau III.1 : description textuelle du cas d'utilisation « Authentification ».....</i>	<i>63</i>
<i>Tableau III.2 : description textuelle du cas d'utilisation « Modifier compte ».....</i>	<i>65</i>
<i>Tableau III.3 : Description textuelle du cas d'utilisation « Gestion des cours ». ....</i>	<i>67</i>
<i>Tableau III.4 : Identification des Services Web.....</i>	<i>68</i>



## Introduction générale

L'émergence des technologies de l'information et de la communication (TIC) et leur application dans plusieurs domaines avec des succès variables, implique la définition d'un grand nombre de systèmes logiciels. De tels systèmes sont implémentés dans des langages de programmation différents, en utilisant des types de ressources distincts.

L'apprentissage et l'enseignement est l'un de ces domaines d'application ; où l'apprentissage est différent et fonctionnel, à savoir : Plates-formes, référentiels, outils, types de contenu...etc.

Par ailleurs, ces systèmes devraient s'inter-opérer et ce, afin de fournir des services d'apprentissage meilleurs et plus utiles aux apprenants ainsi qu'aux formateurs.

Pour ce faire, les spécifications **d'Interopérabilité** doivent être mis en place ; en effet , ces dernières incarnent une véritable nécessité. Pour réaliser cette interopérabilité, les systèmes d'apprentissage, plus communément connu sous le terme « **e-Learning** » devraient pouvoir inter-opérer entre eux indépendamment des langages de programmations et des plates-formes. Sur ce, les **Services Web** rentrent en jeu afin d'assurer et de satisfaire cet aspect technique.

Ceci dit, notre travail consiste en résumé à intégrer la technologie des **Services Web** au sein de certaines fonctionnalités fournies par un système e-learning de manière générale et ce, en exploitant au mieux l'architecture Java EE, en utilisant des outils ainsi que des Frameworks favorisant l'implémentation de ces services web.

Par ailleurs, la finalité du travail demandé est de mener à bien l'analyse et la conception d'une solution afin de garantir l'aspect technique « **Interopérabilité** » d'un système e-learning classique et ce, en s'appuyant sur la technologie des **services web** ; en particulier de **type SOAP**, sujet omniprésent de ce projet de fin d'études.

Dès lors, ce présent mémoire est réparti comme suit :

Chapitre I : Services web et Interopérabilité.

Chapitre II : Description d'un système e-learning.

Chapitre III : Analyse et Conception

Chapitre IV : Réalisation.

Au final, nous avons pu aboutir aux résultats attendus, autrement dit aux objectifs fixés au départ du lancement de notre projet, à savoir : l'implémentation des Services Web de type Soap au sein d'une Architecture Client/serveur basée sur le modèle MVC (Model View Controller) correspondants aux Services e-learning cités ci-après : Authentification, Profile, Account et Courses.

## I-1- Introduction

« À l'ère de l'informatique dominée par le World Wide Web, la technologie **Services Web** est la clé pour permettre aux ordinateurs de communiquer machine à machine, programme à programme et pour cause, les services web facilitent la connexion de tous les types d'applications informatiques l'un à l'autre. Ils fournissent également une base **d'interopérabilité** grandement nécessaire dans un monde où les services informatiques et l'information numérique existent sous différentes formes et saveurs », explique *Marshall Breeding* dans son rapport « Introduction » dans le dernier numéro de *Library Technology Reports*.

Autrement, les bienfaits des Services Web ont nourrit les motivations des chercheurs et développeurs, choses qui ont poussé ces derniers à la conception et à la mise en place de ces dispositifs. Ceci dit, ce présent chapitre prendra en considération quelques notions au préalable et ce avant de passer à la présentation des *Services Web* et des concepts qui leurs sont associés.

## I-2- Style d'architecture dans les systèmes informatiques [1]

Un **style d'architecture** en informatique définit une famille de systèmes en termes de modèles de structure, d'un vocabulaire de composants, de connecteurs et de règles ou contraintes sur leurs relations. Ceci dit, chaque style d'architecture a ses propres caractéristiques et son évolution est généralement liée à celle de la technologie.

Les styles d'architecture sont donc des "modèles de conception" [design patterns] pour la structure et les interconnexions entre les systèmes logiciels. On distingue d'ailleurs trois grands styles d'architecture pour les systèmes distribués, à savoir :

- L'Architecture Orientées Objets,
- L'Architecture Orientées Ressources,
- L'Architecture Orientées Services.

Dans ce présent chapitre, nous allons nous focaliser sur le dernier style cité, c'est-à-dire : L'Architecture Orientée Services (SOA pour Services Oriented Architecture).

### I-3- Architecture Orientée Services (SOA)

Avant de définir ce qu'est une architecture orientée services, nous allons nous intéresser au préalable à certains concepts dont la connaissance est indispensable.

#### I-3-1- Définition d'un service [1]

Un **service** est une fonction logicielle autonome et sans état qui accepte des requêtes et qui renvoie des réponses à travers une interface standard bien définie sur un réseau. Un service est donc une unité de traitement qui fournit un résultat à un client [maître d'ouvrage]. En effet, le client ainsi que le fournisseur [maître d'œuvre] sont habituellement des agents logiciels qui agissent par délégation de leurs propriétaires.

De plus, les services ne doivent pas dépendre de l'état d'autres fonctions ou d'autres traitements externes.

Enfin, dans une architecture orientée services, tous les messages ou bien toutes les requêtes pour un service spécifique sont envoyées à l'adresse unique de celui-ci.

#### I-3-2- Orchestration des services

Les services peuvent être composés dans le but de réaliser un processus donné. L'orchestration leur permet justement de **communiquer sans avoir à se connaître** afin de préserver leur couplage lâche, c'est-à-dire leur indépendance. Pour ce faire, **un moteur d'orchestration se charge d'appeler les services** selon l'enchaînement désiré.

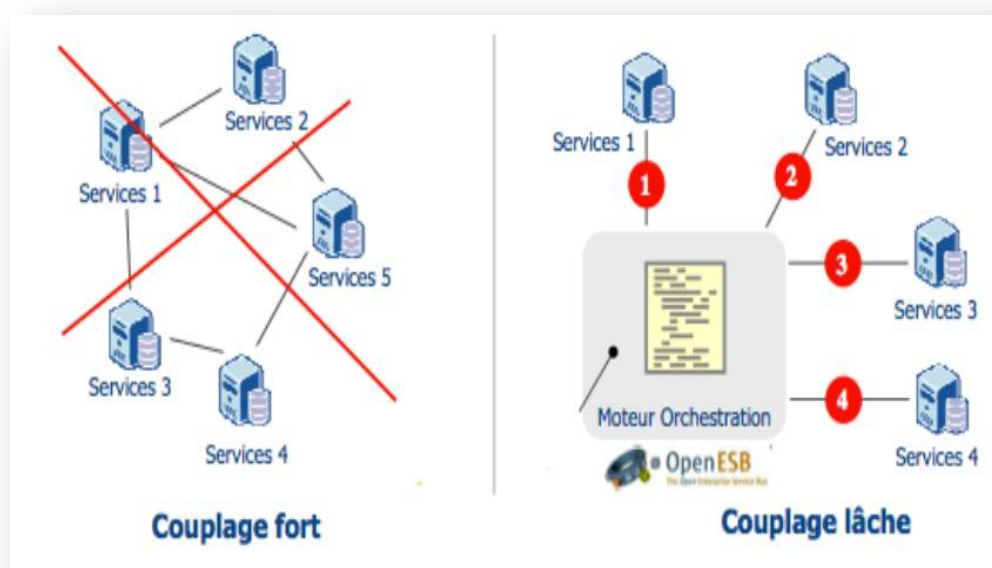


Figure I.1 : Orchestration de services

### I-3-3- Propriétés des services [2]

- ✓ Réutilisables et possèdent des contrats standardisés,
- ✓ Communiquent par *messages* à travers des *interfaces adressables*,
- ✓ Abstraits et <sup>1</sup>prédictibles,
- ✓ Modulaires et <sup>2</sup>granulaires,
- ✓ Autonomes et sans état (*stateless*),
- ✓ Moyens pour assurer une haute *Interopérabilité*,
- ✓ <sup>3</sup>Faiblement Couplés,
- ✓ Découvrables (dynamiquement),
- ✓ Composables.

### I-3-4- Définition du style d'Architecture Orientée Service [1]

Une architecture orientée services (notée SOA pour Services Oriented Architecture) est une architecture logicielle distribuée s'appuyant sur un ensemble de services simples. Elle permet de décomposer une fonctionnalité en un ensemble de fonctions basiques appelées services, fournies par des composants et de décrire finement le schéma d'interaction entre ces services.

Le style SOA se définit par :

- Un couplage faible [*loosely coupled*] entre les composants d'un système pour ne pas dépendre de l'état d'autres services et pour faciliter la réutilisation,
- Des services sans état [*stateless*] pour faciliter l'extensibilité [*scalability*] et l'éventuelle orchestration,
- Des services fortement interopérables, ce qui implique l'utilisation de vocabulaires de données très bien définis.

Cependant, plusieurs partenaires peuvent communiquer et échanger des données dans le contexte du SOA, indépendamment des plateformes et des langages de programmation.

---

<sup>1</sup>Prédictible : son comportement et sa réponse lors de la réception d'une requête ne varient pas.

<sup>2</sup>granularité : le service est un gros grain qui regroupe un ensemble d'interfaces cohérentes se rapportant à un même module fonctionnel.

<sup>3</sup>Faiblement Couplés : dépendance faible entre le consommateur et le service.

### I-3-5- Microservices [3]

Les microservices désignent à la fois une architecture et une approche de développement logiciel qui consiste à décomposer les applications en éléments plus simples et indépendants les uns des autres. Contrairement à une approche <sup>4</sup>*monolithique* classique, selon laquelle tous les composants forment une entité indissociable, les microservices fonctionnent en synergie pour accomplir les mêmes tâches, tout en étant séparés. Chacun de ces composants ou processus est un **microservice**. Granulaire et léger, ce type de développement logiciel permet d'utiliser un processus similaire dans plusieurs applications.

### I-3-6- SOA Vs Microservices

SOA	Microservices
les <b>services</b> ne peuvent pas fonctionner et être déployés indépendamment des autres services.	les <b>services</b> peuvent fonctionner et être déployés indépendamment des autres services.
les services partagent le stockage de données.	chaque service peut avoir un stockage de données indépendant, cependant quelques-uns peuvent aussi partager le même stockage de données.
une architecture SOA peut être un monolithe ou il peut être composé de plusieurs microservices.	Microservice doit être nettement plus faible que ce que la SOA a tendance à être.

**Tableau I.1 : Comparatif entre SOA et Microservices.**

### I-3-7- Avantages de l'Architecture Orientée Services [2]

- ✓ **Métier :**
  - Améliorer l'agilité et la flexibilité du métier (évolutivité),
  - Réduire en temps le cycle de développement des produits,
  - Faciliter la gestion des <sup>5</sup>*processus métiers*,
  - Améliorer le retour sur investissement ;
- ✓ **Technique :**
  - Réduire la complexité de la solution,
  - Construire les services une seule fois et les utiliser fréquemment,
  - Faciliter la maintenance.

---

<sup>4</sup>*Monolithique* : formé d'un seul bloc, ses composants sont interconnectés et interdépendants.

<sup>5</sup> *Processus métier* : un ensemble d'activités incluant une interaction entre des participants sous la forme d'échange d'informations visant à atteindre un objectif particulier.

Le style SOA peut s'appliquer à toutes les technologies pour tout type de réalisation. Cependant, la révolution SOA est tirée par les standards de l'Internet ; Il existe par ailleurs deux groupes de travail du W3C (*World Wide Web Consortium*) qui couvrent le sujet SOA : un sur l'*Architecture du World Wide Web* et un autre sur les *Web Services*[1] dont nous verrons les concepts dans ce qui suit.

#### **I-4- Les Services Web**

Lorsque l'architecture SOA s'appuie sur des web services, on parle alors de WSOA, c'est à dire : *Web Services Oriented Architecture*.

##### **I-4-1- Définition d'un service web**

Il s'agit d'une technologie permettant à des applications de dialoguer à distance via Internet, et ceci indépendamment des plates-formes et des langages sur lesquelles elles reposent. Pour ce faire, les services Web s'appuient sur un ensemble de protocoles Internet répandus, à savoir : XML, Http et ce afin de communiquer. Cette communication est basée sur le principe des demandes et réponses, effectuées avec des messages XML.

Les services web sont décrits par des documents *WSDL (Web Service Description Language)*, qui précisent les méthodes pouvant être invoquées, leurs signatures et les points d'accès du service (URL, port ...etc). Les services Web sont accessibles via *SOAP*, la requête et les réponses sont des messages XML transportés sur HTTP.

Il existe probablement autant de définitions des Web Services que d'entreprises qui les créent, cependant presque toutes ces définitions ont ce qui suit en commun :

- les Web Services proposent aux utilisateurs du Web des fonctionnalités pratiques grâce à un protocole Web standard (dans la plupart des cas, le protocole utilisé est SOAP)
- les Web Services offrent un moyen de décrire leurs interfaces suffisamment en détail pour permettre à un utilisateur de créer une application cliente capable de converser avec eux ; cette description est généralement fournie dans un document XML nommé WSDL (Web Services Description Language)

- les Web Services sont inscrits afin que les utilisateurs potentiels puissent les trouver facilement. Ceci est possible grâce à *UDDI (Universal Discovery Description and Integration)*.

### **I-4-2- Pourquoi les services web ?**

Les services web ont été mis en place afin de répondre à un certain nombre de besoins, à savoir :

- ✓ Améliorer les protocoles actuels (<sup>6</sup>*RPC*, <sup>7</sup>*DCOM*, <sup>8</sup>*RMI*) par une approche entièrement ouverte et interopérable.
- ✓ Faire interagir des composants hétérogènes, distants et indépendants avec un protocole standard (SOAP).
- ✓ Simplifier la communication entre ces composants.
- ✓ Ne pas créer de nouvelles technologies et se baser sur celles qui existent déjà (XML, HTTP).
- ✓ Dédiés aux applications B2B (Business to Business), EAI (Enterprise Application Integration), P2P (Peer to Peer).

En effet, Les services web sont très utilisés par les entreprises, ce qui leur permet d'exposer un certain nombre de services et d'échanger des informations entre elles.

---

<sup>6</sup>*RPC* (Remote Procedure Call) : est un protocole réseau permettant de faire des appels de procédures sur un ordinateur distant à l'aide d'un serveur d'applications. Ce protocole est utilisé dans le modèle client-serveur pour assurer la communication entre le client, le serveur et d'éventuels intermédiaires.

<sup>7</sup>*DCOM* (Distributed Component Object Model) : est une technique propriétaire de Microsoft qui permet la communication entre des composants logiciels distribués au sein d'un réseau informatique.

<sup>8</sup>*RMI* (Remote method invocation) : est une interface de programmation pour le langage Java qui permet d'appeler des méthodes distantes.



Les services web proposent un mécanisme facilitant :

- la communication entre applications hétérogènes : un service web développé dans une technologie peut être consommé par une application développée dans une autre technologie. Ceci est possible car les services web reposent sur des standards ouverts,
- l'exposition de fonctionnalités métiers aux applications internes mais aussi à des applications externes : dans ce dernier cas l'utilisation du protocole HTTP permet facilement de passer les pare-feux,
- la mise en œuvre d'une architecture SOA puisque les services web peuvent être une implémentation possible d'une telle architecture,
- L'utilisation de services web peut avoir plusieurs intérêts :
- l'exposition de fonctionnalités au travers du réseau : les traitements des opérations des services web peuvent être invoqués par une requête HTTP, ce qui peut permettre à plusieurs applications de consommer ces services web,
- la communication entre des applications et des systèmes hétérogènes : l'utilisation de standards ouverts permet la production et la consommation des services web par différentes technologies sur différents systèmes d'exploitation,
- la mise en œuvre des protocoles standards de l'industrie au niveau des couches transport, messaging, description et recherche permet de choisir entre plusieurs implémentations proposées et ainsi de ne pas dépendre d'un seul fournisseur,
- les échanges se font en utilisant l'infrastructure existante puisque les services web sont généralement invoqués en utilisant le protocole HTTP. Ceci permet de facilement passer un firewall pour permettre une invocation depuis l'extérieur,
- les services web permettent un couplage faible entre les fonctionnalités exposées et les applications qui les utilisent à tel point que les consommateurs et les producteurs peuvent être écrits pour des plates-formes ou des langages différents (Java, .Net, PHP)

- les services permettent de définir de nouvelles opportunités de business voir même de nouveaux modèles économiques en permettant de proposer des fonctionnalités à des partenaires par exemple. [4]

#### **I-4-4- Les standards utilisés par les services Web [4]**

L'intérêt des services web grandissant, des standards ont été développés pour assurer les besoins nécessaires à leur mise en œuvre. L'architecture des services web est composée de quatre grandes couches utilisant plusieurs technologies :

- Découverte : cette couche représente un annuaire dans lequel il est possible de publier des services et de les rechercher (UDDI est le standard)
- Description : cette couche normalise la description de l'interface publique d'un service web en utilisant WSDL (Web Service Description Language).
- Communication : cette couche permet d'encoder les messages échangés (SOAP est le standard)
- Transport : cette couche assure le transport des messages : généralement HTTP est mis en œuvre mais d'autres protocoles peuvent être utilisés (SMTP, FTP, ...)
- La description d'un service web permet à son consommateur de connaître l'interface du service.
- La communication permet de formaliser le format des messages échangés.

En plus de SOAP, WSDL et UDDI, il existe de nombreuses autres spécifications plus ou moins standard pour permettre la mise en œuvre de fonctionnalités manquantes dans ces standards comme la sécurité, la gestion des transactions, l'orchestration des services. Ces spécifications sont en cours de développement ou d'évolution ce qui les rend généralement immatures. De plus, fréquemment, il existe plusieurs spécifications ayant trait à un même sujet qui sont donc concurrentes. La mise en œuvre de ces spécifications n'est pas requise pour des services web basiques mais elle peut être nécessaire pour des besoins plus spécifiques.

#### I-4-5- Fonctionnement des services web [5]

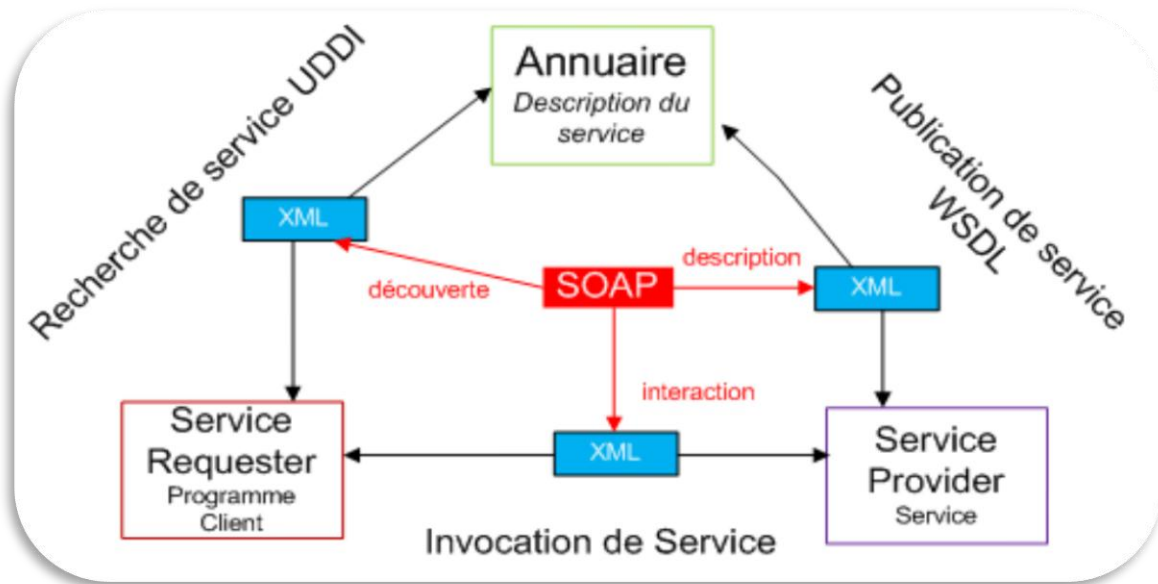


Figure I.2 : Fonctionnement des services web.

- Service Requester (client) : C'est n'importe quel consommateur du service Web. Le demandeur utilise un service Web existant en ouvrant une connexion réseau et en envoyant une demande en XML (REST, XML-RPC, SOAP).
- Service Provider (fournisseur de services) : le fournisseur de service met en application le service Web et le rend disponible sur Internet.
- Le registre de service : c'est un annuaire de services, celui-ci fournit un endroit central où les fournisseurs peuvent publier de nouveaux services et d'en trouver par rapport aux clients.

#### I-4-6- Types de services web [2]

On distingue deux types de services web, à savoir :

- **Les services web de type SOAP (Simple Object Access Protocol) :** utilisent les standards **UDDI** (annuaire), **WSDL** (contrat) et **SOAP** (protocole, messages / communication).
- **Les services web de type REST (Representational State Transfer) :** ils utilisent directement **HTTP** au lieu d'une enveloppe SOAP ainsi qu'une **URI** pour nommer et identifier une ressource et les méthodes HTTP (**POST**, **GET**, **PUT** et **DELETE**) pour effectuer les opérations de base **CRUD** (Create, Read, Update and Delete).

## I-5- Les services web de type SOAP

### I-5-1- Architecture

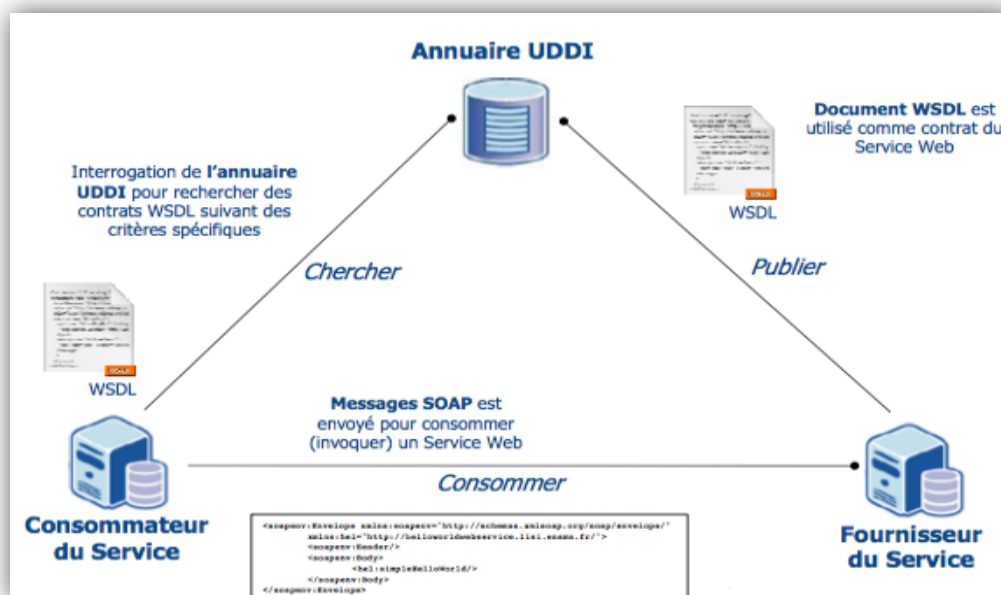


Figure I.3 : Architecture des services web de type SOAP

### I-5-2- Protocole SOAP

SOAP est un protocole ou bien un mécanisme qui permet d'échanger des messages XML entre applications. Il définit un ensemble de règles pour structurer des messages qui peuvent être utilisés dans de simples transmissions unidirectionnelles, mais il est particulièrement utile pour exécuter des dialogues requête-réponse RPC (Remote Procedure Call). Il n'est pas lié à un protocole de transport particulier mais HTTP est populaire. Il n'est pas non plus lié à un système d'exploitation ni à un langage de programmation, donc, théoriquement, les clients et serveurs de ces dialogues peuvent tourner sur n'importe quelle plate-forme et être écrits dans n'importe quel langage du moment qu'ils puissent formuler et comprendre des messages SOAP. Il s'agit donc d'un important composant de base pour développer des applications distribuées qui exploitent des fonctionnalités publiées comme services par des intranets ou internet.

De plus, SOAP a été développé à l'origine pour traverser via HTTP les pare-feux qui bloquaient les appels RPC « Remote Procedure Call » de DCOM ou <sup>9</sup>I/OP.

<sup>9</sup>I/OP : pour Internet Inter-ORB Protocol, c'est le protocole de communication utilisé par CORBA.

### **I-5-3-Principes SOAP [6]**

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.

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 et les technologies.

### **I-5-4- Structure d'un message SOAP [7] [1]**

SOAP définit un format pour l'envoi des messages. Ces derniers sont structurés en un document XML et comporte :

- Une enveloppe
- Un en-tête [Header]
- un corps [Body].

L'enveloppe définit la structure du message ; l'en-tête contient diverses informations (autorisations et transactions par exemple) et le corps transporte les informations sur l'appel et la réponse ainsi que sur les erreurs et les attachements éventuels.

Les messages SOAP peuvent être échangés entre des nœuds SOAP par une grande variété de protocoles. Cependant dans la majorité des cas, c'est le protocole HTTP qui est utilisé. Un nœud SOAP est donc identifié par son URI comme une ressource en REST. Les premières versions de SOAP imposaient l'usage de POST. La mise en œuvre de SOAP était donc assez lourde. SOAP ne permettait pas de bénéficier du mécanisme de cache fourni automatiquement par GET. Cette lacune a été comblée dans la version 1.2 de SOAP où la notion de SOAP Web Method a été introduite. Celle-ci permet d'utiliser les verbes HTTP autres que POST comme GET, PUT et DELETE.

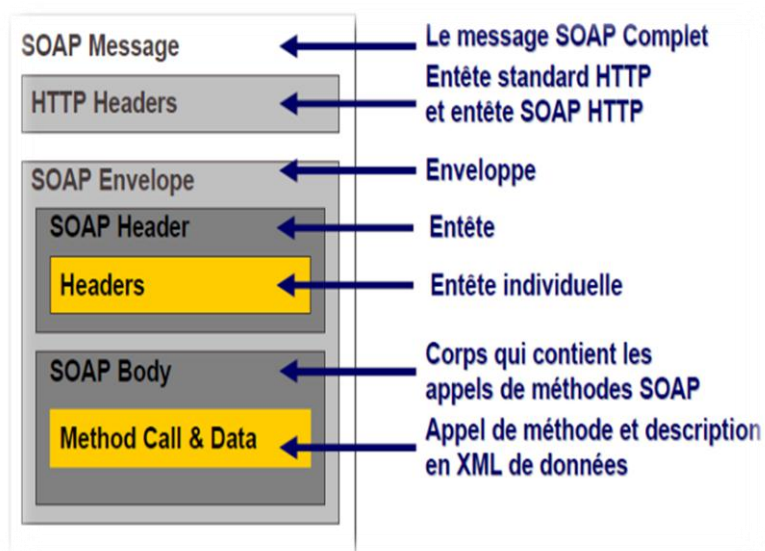


Figure I.4 : Format d'un message SOAP.

### I-5-5- En- tête, Corps (requête/réponse) d'un message SOAP [8]

#### I-5-5-1- L'enveloppe [envelop]

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
...
  Message information goes here
...
</soap:Envelope>
```

#### I-5-5-2- L'en-tête [header]

L'élément optionnel « SOAP Header » contient des informations spécifiques à l'application comme l'authentification, le paiement... etc concernant le message SOAP.

Si l'élément Header est présent, il doit être le premier élément enfant de l'élément Enveloppe.

**Remarque :** tous les éléments enfants immédiats de l'élément *Header* doivent être qualifiés pour l'espace de noms.

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
  <soap:Header>
    <m:Trans xmlns:m="https://www.w3schools.com/transaction/"
soap:mustUnderstand="1">234
    </m:Trans>
  </soap:Header>
  .....
</soap:Envelope>
```

L'exemple ci-dessus contient un en-tête avec un élément " Trans", un attribut "mustUnderstand" avec une valeur de 1 et une valeur de 234. SOAP définit trois attributs dans l'espace de noms par défaut. Ces attributs sont: mustUnderstand, acteur et encodingStyle. Les attributs définis dans l'en-tête SOAP définissent comment un destinataire doit traiter le message SOAP.

### I-5-5-3- Le corps [body]

L'élément SOAP Body requis contient le message SOAP réel destiné à l'extrémité ultime du message. Les éléments enfants immédiats de l'élément SOAP Body peuvent être qualifiés pour l'espace de noms.

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
```

```

<soap:Body>
  <m:GetPrice xmlns:m="https://www.w3schools.com/prices">
    <m:Item>Apples</m:Item>
  </m:GetPrice>
</soap:Body>

</soap:Envelope>

```

L'exemple ci-dessus demande le prix des pommes. Il faut noter que les éléments « m:GetPrice » et « Item » sont des éléments spécifiques à l'application. Ils ne font pas partie de l'espace de noms SOAP.

Une réponse SOAP pourrait ressembler à ceci :

```

<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
  soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
  <soap:Body>
    <m:GetPriceResponse xmlns:m="https://www.w3schools.com/prices">
      <m:Price>1.90</m:Price>
    </m:GetPriceResponse>
  </soap:Body>
</soap:Envelope>

```

### I-5-6- L'encodage des messages SOAP [4]

On distingue deux formats de messages SOAP, à savoir :

- remote procedure call : (RPC) permet l'invocation d'opérations qui peuvent retourner un résultat.
- message oriented (Document) : données au format XML définies dans un schéma XML



Les règles d'encodage (Encoding rules) précisent les mécanismes de sérialisation des données dans un message. Il existe deux types :

- Encoded : les paramètres d'entrée de la requête et les données de la réponse sont encodées en XML dans le corps du message selon un format particulier à SOAP
- Literal : les données n'ont pas besoin d'être encodées de façon particulière : elles sont directement encodées en XML selon un schéma défini dans le WSDL

Le style et le type d'encodage permettent de définir comment les données seront sérialisées et désérialisées dans les requêtes et les réponses.

La combinaison du style et du type d'encodage peut prendre plusieurs valeurs :

- RPC/Encoded
- RPC/Literal
- Document Encoded : cette combinaison n'est pas implémentée
- Document/Literal
- Wrapped Document/Literal : extension du Document/Literal proposé par Microsoft

Le style RPC/Encoded a largement été utilisé au début des services web : actuellement ce style est en cours d'abandon par l'industrie au profit du style Document/Literal. C'est pour cette raison que le style RPC/Encoded n'est pas intégré dans le WS-I Basic Profile 1.1.

Le style et le type d'encodage sont précisés dans le WSDL. L'appel du service web doit obligatoirement se faire dans le style précisé dans le WSDL puisque celui-ci détermine le format des messages échangés.

Avant de définir les styles d'encodage du protocole SOAP, nous débuterons à priori avec la description de certains concepts dont la connaissance est indispensable, à savoir : le standard XML, le protocole RPC, l'utilisation de SOAP dans http et L'utilisation de SOAP pour RPC.

## I-6- XML [9]

Le langage XML (eXtended Markup Language) est un format général de documents orienté texte ; Il s'est imposé comme un standard incontournable de l'informatique. Il est aussi bien utilisé pour le stockage de documents que pour la transmission de données entre applications. Sa simplicité , sa flexibilité et ses possibilités d'extension ont permis de l'adapter à de multiples domaines allant des données géographiques au dessin vectoriel en passant par les échanges commerciaux.

Les avantages du XML sont multiples, à savoir :

- *Lisibilité* : il est facile pour un humain de lire un fichier XML car le code est structuré et facile à comprendre. En principe, il est même possible de dire qu'aucune connaissance spécifique n'est nécessaire pour comprendre les données comprises à l'intérieur d'un document XML.
- *Disponibilité* : ce langage est libre et un fichier XML peut être créé à partir d'un simple logiciel de traitement de texte (un simple bloc- note suffit).
- *Interopérabilité* : quel que soit le système d'exploitation ou les autres technologies, il n'y a pas de problème particulier pour lire ce langage.
- *Extensibilité* : De nouvelles balises peuvent être ajoutées à souhait.
- Tous les navigateurs internet récents intègrent un parseur XML, pour lire les documents de ce langage informatique.

### I-6-1- XML- RPC [10]

**XML-RPC** est un protocole RPC (Remote Procedure Call), une spécification simple et un ensemble de codes qui permettent à des processus s'exécutant dans des environnements différents de faire des appels de méthodes à travers un réseau.

XML-RPC permet d'appeler une fonction sur un serveur distant à partir de n'importe quel système (Windows, Mac OS X, GNU/Linux) et avec n'importe quel langage de programmation. Le serveur est lui-même sur n'importe quel système et est programmé dans n'importe quel langage.

Cela permet de fournir un Service web utilisable par tout le monde sans restriction de système ou de langage.

Les processus d'invocation à distance utilisent le protocole HTTP pour le transfert des données et la norme XML pour la structuration des données.

XML-RPC est conçu pour permettre à des structures de données complexes d'être transmises, exécutées et renvoyées très facilement. XML-RPC est une alternative aux Services Web WS-\*, dont SOAP.

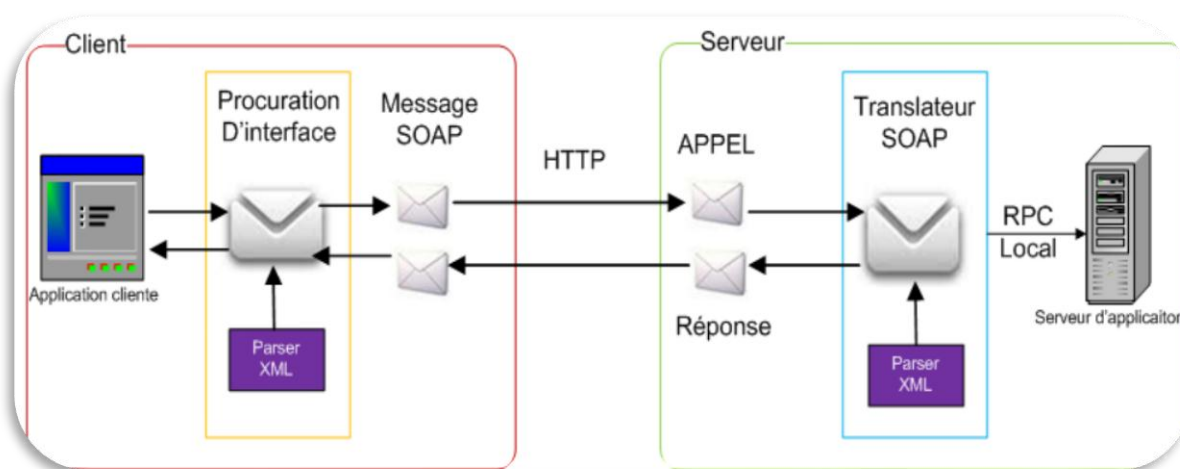


Figure I.5 : Communication entre un client et un serveur en RPC.

### I-6-2- Utilisation de SOAP dans http [11] :

Le fait de lier SOAP à HTTP nous permet d'utiliser la forme et la flexibilité décentralisée de SOAP tout en bénéficiant des fonctionnalités utiles du protocole HTTP. Par ailleurs, associer SOAP à HTTP ne signifie pas que SOAP va redéfinir la sémantique de HTTP mais plutôt que la sémantique de SOAP mappe naturellement la sémantique de HTTP.

SOAP suit naturellement le modèle de message requête/réponse HTTP en fournissant des paramètres de requête SOAP dans les requêtes HTTP et des paramètres de réponse SOAP dans les réponses HTTP.

Les applications HTTP doivent absolument utiliser le type de support "text/xml" conformément à RFC 2376, lorsqu'elles insèrent des corps d'entités SOAP dans des messages HTTP.

### **I-6-3- Utilisation de SOAP pour les appels de procédure à distance (RPC) [11] :**

L'un des objectifs de conception de SOAP est d'encapsuler et d'échanger les appels RPC en bénéficiant de l'extensibilité et de la flexibilité de XML. Cette section définit une représentation uniforme des appels et réponses de procédure à distance. L'utilisation de SOAP pour RPC est orthogonale à la liaison du protocole SOAP et l'utilisation de HTTP comme liaison de protocole permet de mapper les appels RPC sur les requêtes HTTP ainsi que les réponses RPC sur les réponses HTTP. Toutefois, l'utilisation de SOAP pour RPC ne se limite pas à la liaison de protocole HTTP.

Pour effectuer un appel de méthode, il faut fournir les informations suivantes :

- L'URI (Uniform Ressource Identifier) de l'objet de destination,
- Un nom de méthode,
- Une signature de méthode optionnelle,
- Les paramètres de la méthode,
- Les données d'en-tête optionnelles.

SOAP se sert de la liaison de protocole pour fournir un mécanisme de transfert des URI. Par exemple, pour HTTP l'URI de la requête indique la ressource concernée par l'invocation. Hormis la nécessité d'utiliser un URI valide, SOAP ne place aucune restriction quant à la forme de l'adresse.

Dans ce qui suit, nous allons passer à la description des styles d'encodage associés au protocole SOAP :

### **I-6-4- SOAP de style RPC :**

Les appels de procédure distante (RPC) effectués sur des protocoles Web ne sont essentiellement pas différents de ceux effectués sur d'autres protocoles, tels que IIOP, DCOM ou JRMP. Les appels sont généralement synchrones (en d'autres termes, le client attend que la méthode revienne avant de continuer). Zéro ou plusieurs paramètres de types différents sont passés dans l'appel pour fournir des informations à traiter, et zéro ou plusieurs valeurs de retour sont générées pour fournir les sorties de la méthode distante au

client. Les appels de méthode distante sont remis à une certaine forme de répartiteur sur le serveur distant qui détermine la méthode à appeler et organise le flux régulier des paramètres et des valeurs de retour.

Pour le fonctionnement de type RPC, les implémentations SOAP sont conformes à la description précédente. La différence avec SOAP (et d'autres mécanismes RPC basés sur le Web, tels que XML-RPC) réside dans le fait qu'il utilise des transports standard à usage général, tels que HTTP, avec une description d'appel de méthode textuelle en XML. Tous les paramètres et les valeurs de retour sont codés en XML dans le corps SOAP, tandis que les informations sur le service et la méthode à appeler sont fournies dans l'en-tête de transport et éventuellement dans l'en-tête SOAP. Lorsqu'il est envoyé via HTTP, l'en-tête et le corps SOAP sont enveloppés dans un autre document XML - l'enveloppe SOAP - et cette enveloppe constitue le corps d'une requête HTTP POST.

Un message SOAP HTTP sera envoyé à un routeur SOAP qui prendra la forme d'une servlet HTTP (pour une implémentation Java). Le routeur SOAP examinera les informations d'en-tête HTTP et SOAP et décidera comment il doit transférer le corps du message. Cela impliquera d'instancier ou d'appeler un composant particulier ou une classe qui recevra le message. Le routeur SOAP, ou ses classes auxiliaires, effectuera également la conversion des paramètres basés sur XML en objets et primitives Java pouvant être transmis dans le cadre de l'appel de service.

#### **I-6-5- SOAP de style document :**

Dans un appel SOAP de type document, la couche SOAP envoie un document XML complet au serveur sans attendre un résultat en retour. Le message peut contenir n'importe quel type de données XML, pour peu qu'elles soient adaptées au service distant. En mode document (document style encoding), le développeur à la main sur tout, du choix du protocole de transport (HTTP, SMTP...etc) à la sérialisation, en passant par le format des enveloppes SOAP.

## I-7- SOAP Vs RMI, DCOM et CORBA

<i><b>SOAP</b></i>	<i><b>RMI</b></i>	<i><b>DCOM</b></i>	<i><b>CORBA</b></i>
Multi langages	Mono langages	Multi langages	Multi langages
Multiplateformes	Multiplateformes	Mono plateformes	Multiplateformes
Capable de franchir les firewalls (Pare-feux)	Problème de performance et de sécurité.	Faible diffusion	Installation coûteuse

**Tableau I.2 : SOAP Vs RMI, DCOM et CORBA.**

## I-8- Les avantages de SOAP [12]

Au départ, les architectes d'entreprise ont privilégié les services Web de type SOAP. Les principaux avantages de ce protocole étant :

- L'indépendance vis à vis du langage de programmation,
- L'indépendance vis à vis de la plateforme où ils sont déployés,
- son extensibilité,
- l'utilisation du standard XML pour l'échange des messages,
- le contenu des messages échangés qui incluent le format d'échange cela permet d'avoir un contrat personnalisé entre le fournisseur et le consommateur du service,
- l'utilisation du protocole HTTP pour le transport qui permet de s'affranchir des problématique de firewalls: en réalité SOAP est indépendant de la couche de transport puisqu'il l'a redéfini (entête, format d'échange, type des données échangées, ...). Cependant, HTTP est le protocole de transport le plus largement utilisé par SOAP,
- SOAP définit sa propre sécurité appelée : WS Security.

### I-9- Les inconvénients de SOAP [13]

- Lent : SOAP utilise un format XML qui doit être analysé pour être lu. Il définit de nombreuses normes qui doivent être suivies lors du développement des applications SOAP. Il est donc lent et consomme plus de bande passante et de ressources.
- Lourdeur et complexité.
- Dépendant de WSDL : SOAP utilise WSDL et n'a aucun autre mécanisme pour décrire le service.

### I-10- Les problèmes liés à SOAP [12]

Les avantages précédant devaient permettre une interopérabilité entre des applications hétérogènes développées en Java, .NET ou PHP. Cependant, les architectes ont été confrontés à des problèmes parmi eux :

- SOAP étant extensible, plusieurs standards sont apparus, regroupés sous la forme WS-\* :

WS-Addressing, WS-Policy, WS-Security, WS-Federation, WS-Reliable Messaging, WS-Coordination, WS-Atomic Transaction, WS-RemotePortlets, etc. Leur nombre croissant a abouti à l'augmentation de la complexité perçue de SOAP. De plus, des différences d'implémentations ont aboutis à des problèmes d'interopérabilité entre applications hétérogènes.

- Pour que les applications puissent interagir entre elles, la description des Web services en WSDL et la définition d'un dictionnaire de données sont nécessaires dans l'entreprise. Cependant, dès qu'une erreur de conception de ce dictionnaire apparaît, l'implémentation des services Web devient problématique et coûteuse. Ajouter à cela, le coût de mise en œuvre des fichiers de description des services Web (le WSDL) et des fichiers de données (XSD). En résumé, le coût de mise en œuvre est souvent très élevé.
- La conception des services Web en elle-même : si un service est mal conçu et cela peut arriver très vite, le nombre de données qu'il renvoie peut-être très important. Ajouter à cela, la verbosité de SOAP, très rapidement, des problèmes de performance

sont rencontrés qui induisent souvent à une remise en cause de la conception elle-même des services.

Le coût et la complexité d'implémentation générés par les inconvénients de SOAP ont remis en cause, complètement, les avantages de son utilisation. Les réseaux sociaux, dans un premier temps, avec leur volume de données échangées puis les autres entreprises, ensuite, se sont positionnées sur un autre moyen de mettre en œuvre des services Web moins coûteux, plus simple à implémenter et surtout plus efficace. Cette méthode qui n'est pas nouvelle puisqu'elle est basée sur les fondamentaux du protocole HTTP est : l'architecture REST qu'on verra un peu plus tard.

### **I-11- WS-Security**

WS-Security est un standard pour sécuriser des services web SOAP uniquement. Ce dernier aborde les aspects classiques comme l'authentification, l'habilitation mais également des besoins plus rares comme le chiffrement, la signature. Cette norme s'applique uniquement au niveau des messages échangés et n'a aucun impact sur le protocole et le transport choisis. Il repose sur l'ajout d'éléments dans les headers/en-têtes SOAP. Ce standard appelé aussi WSS (Web Services Security) tente de normaliser les échanges sécurisés de messages et de garantir l'interopérabilité entre systèmes.

Des Profils sont disponibles pour couvrir l'ensemble des problématiques liées à la sécurité (authentification, chiffrement et signature). Le consortium OASIS vise la standardisation des spécifications WS-\* pour permettre une meilleure interopérabilité entre les technologies.

#### *WS-Security : une norme de plus en plus accessible :*

La norme WS-Security est vaste mais la complexité inhérente à la norme est de plus en plus masquée et devient même dans certains cas complètement externe aux développements applicatifs.

En Java par exemple, l'adoption de nouveaux frameworks et leur intégration avec Spring a favorisé la mise en place de WS-Security par simple déclaration au niveau de la configuration Spring. Un service SOAP peut donc intégrer un chiffrement ou une signature par simple ajout d'un bloc XML.



## I-12- WSDL [1]

WSDL [Web Services Description Language] est un langage de description de services Web. C'est un document XML qui permet de décrire chaque service SOAP sous la forme d'une documentation en ligne. Ce document décrit les messages échangés, habituellement sous la forme d'un schéma XML, ainsi que les détails des interfaces publiques de ce service.

Un document WSDL est donc un simple document XML avec quatre éléments principaux `<portType>`, `<message>`, `<type>` et `<binding>` :

- `<portType>` définit le service web , les opérations qui peuvent être effectuées et les messages. C'est l'équivalent d'un nom de fonction classique ;
- `<message>` définit les données de chaque opération. C'est l'équivalent des paramètres d'une fonction classique;
- `<type>` définit les types de données en utilisant la notation des schémas XML ;
- `<binding>` définit le format des messages et les détails du protocole de chaque port.

Cette documentation n'est pas réservée à SOAP ; elle est disponible en ligne. En effet, elle permet de simplifier l'utilisation des services car elle est exploitable directement par les outils de développement ou de programmation.

En résumé **WSDL** est un contrat entre un client et un serveur qui fait état :

- ✓ Des spécifications d'interfaces qui décrivent toutes les méthodes publiques,
- ✓ Des spécifications relatives aux types de donnée de messages mis en œuvre dans les requêtes -réponses,
- ✓ Des informations liées au protocole de transport utilisé,
- ✓ Des informations d'adresse permettant de localiser le service décrit.

En un mot, WSDL définit le contrat existant entre un client et un serveur sans dépendance particulière pour une plateforme ou un langage donné.

## I-13- UDDI [7]

Né d'une collaboration entre IBM et ARIBA, l'UDDI est l'équivalent des pages jaunes dans les services Web. Tout comme avec les pages jaunes classiques, on peut y rechercher une entreprise offrant les services dont on a besoin et consulter le service offert.

Bien entendu, on peut proposer un service Web sans l'inscrire, mais si on souhaite toucher un public plus important, l'UDDI permet aux clients éventuels de trouver des services.

Une entrée du répertoire UDDI est constituée d'un fichier XML qui décrit une entreprise et les services qu'elle offre. Chaque entrée du répertoire UDDI est constituée de trois parties :

- Les "**pages blanches**" décrivent l'entreprise qui offre le service : nom, adresse, contacts ...etc.
- Les "**pages jaunes**" comportent les catégories industrielles.
- Les "**pages vertes**" décrivent l'interface vers le service avec suffisamment de détail pour qu'il soit possible d'écrire une application permettant d'utiliser le service Web.

L'inscription pour les entreprises se fait par Internet à l'adresse suivante : <http://www.uddi.org/register.html>. A l'heure actuelle, une première version de l'UDDI est disponible sur <http://www.uddi.org>.

## I-14- Les services web de type REST

### I-14-1- Définition [14]

*REST (REpresentational State Transfer)* n'est pas un protocole, un standard ou encore un format. Il s'agit d'un style d'architecture pour les systèmes distribués. Le terme a été introduit par Roy Fielding (un des principaux auteurs de la spécification HTTP) dans une thèse en 2000.

En effet, lorsque vous surfer sur le Web vous faites du REST sans le savoir. Prenons un exemple simple : vous souhaitez accéder aux informations concernant votre film préféré via un site spécialisé. Vous ouvrez donc votre navigateur et saisissez dans la barre d'adresse l'URL (par exemple [HTTP://www.touslesfilms.com/monfilmpréfére](http://www.touslesfilms.com/monfilmpréfére)) de la page dédiée à ce film (on suppose que vous la connaissez). Le navigateur va envoyer une requête HTTP GET à

cette URI afin d'obtenir la ressource correspondante. Le serveur va analyser cette requête et renvoyer une réponse HTTP. Celle-ci sera composée d'un en-tête et d'un corps. Le corps contiendra du html (par exemple) qui s'affichera dans le navigateur. Comment ce dernier sait-il que la requête s'est bien déroulée (pas d'erreur dans l'adresse par exemple) et que les informations renvoyées sont au format html ? Grâce à l'en-tête de la réponse. Celui-ci est composé d'informations comme:

- Un code de statut HTTP indiquant le déroulement de la requête. Un code de statut OK permet au navigateur de savoir que tout s'est déroulé correctement. Vous connaissez sans doute le code Not Found (Erreur 404) indiquant que la ressource demandée est introuvable.
- Un type de contenu (Content-Type) indiquant le format des données du corps de la réponse HTTP. C'est ce qui permet au navigateur de savoir que le contenu est du html et de l'interpréter comme tel.

Toutes ces notions (et quelques autres) sont la base du style d'architecture REST.

Un des concepts important de REST est la notion de ressource. Chaque ressource est accessible par une URI (Uniform Resource Identifier). La ressource étant une notion abstraite, le client et le serveur communiquent en s'échangeant une représentation de la ressource. Le format de cette représentation peut être du XML, du JSON, une image, un fichier vidéo, etc.

Les ressources sont accessibles via un ensemble uniforme de commandes fourni par HTTP (essentiellement GET, POST, PUT et DELETE) qui permettent de spécifier l'opération à effectuer sur une ressource.

La communication entre le client et le serveur est "sans état". Ainsi chaque requête du client vers le serveur doit contenir toutes les informations nécessaires pour que cette demande soit comprise, et elle ne peut tirer profit d'aucun contexte stocké sur le serveur. L'état de la session est donc entièrement détenu par le client.

### **I-14-2- Utilisation de REST [15]**

- Utiliser dans le développement des applications orientés ressources (ROA) ou orientées données (DOA).
- Les applications respectant l'architecture REST sont dites : RESTful.

### **I-14-3- Caractéristiques de REST [15]**

- 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.
  - Minimisation des ressources systèmes ( pas de gestion de session , ni d'état ).
- Interface uniforme basée sur les méthodes http (GET, POST, PUT, DELETE).
- Les architectures RESTful sont construites à partir de ressources uniquement identifiées par des URI(s).

### **I-14-4- WADL [15]**

Web Application Definition Language est un langage de description des services REST au format XML. Il est une spécification de W3C initié par SUN ([www.w.org/Submission/wadl](http://www.w.org/Submission/wadl)).

- Il décrit les éléments à partir de leur type (Ressources, Verbes (méthodes), Paramètre, type de requête, Réponse).
- Il fournit les informations descriptives d'un service permettant de construire des applications clientes exploitant les services REST.

### **I-14-5- Les avantages de REST [12]**

Les architectures de type REST deviennent, de plus en plus, des standards dans les entreprises aujourd'hui. La plupart des consultants en informatique de gestion vous le diront. Cependant, au-delà de l'effet de mode, la mise en œuvre de services Web basée sur cette architecture a de quoi séduire. Les principales motivations pour le choix de cette architecture sont :

- l'indépendance vis à vis du langage de programmation,

- l'indépendance vis à vis de la plateforme sur laquelle ils sont déployés,
- la plus grande simplicité d'implémentation car la couche transport n'a pas besoin d'être redéfinie et il n'est plus nécessaire de créer un dictionnaire de données comme on le fait avec SOAP.
- la non intégration du format d'échange au sein des messages. Cela induit une verbosité beaucoup plus faible et donc une performance accrue et une plus grande souplesse dans l'implémentation,
- l'utilisation de multiples formats pour l'échange de données (XML, JSON, HTML),
- qu'elle est plus proche de la conception et de la philosophie initiale du Web (URI, GET, POST, PUT et DELETE).

#### **I-14-6- Les inconvénients de REST [12]**

Comme les services Web de type SOAP, ceux basés sur l'architecture REST ont le gros avantage d'être interopérables et indépendants de leur environnement de déploiement avec, en bonus, de meilleures performances. Cependant, ils ont des défauts conséquents à leur adhésion au protocole de transport http, à savoir :

- une limitation en matière de sécurité,
- il n'est pas fiable et ne garantit pas la bonne transmission des messages,
- la limitation en termes d'actions possibles (CRUD uniquement),
- les navigateurs actuels implémentent mal les opérations PUT et DELETE.

Dans la réalité, ces limitations ne sont pas vraiment un frein. La sécurité peut être introduite de plusieurs façons différentes (tokens d'authentification, HTTPS, des outils comme Spring Security en java). Les problèmes de standardisation des navigateurs sont détournés par surcharge de la méthode POST. Le vrai frein au choix de REST pour la mise en œuvre de services web est le besoin propre à chaque application.

### I-14-7- Le débat SOAP VS REST [16]

On sait que la façon la plus répandue d'utiliser SOAP consiste à s'appuyer sur le protocole HTTP en utilisant la liaison SOAP- HTTP. Dans cette liaison, il est possible d'associer plusieurs opérations à la même URL. Par exemple, un service de ventes peut être placé à l'URL [www.unservicedevende.fr/serviceweb](http://www.unservicedevende.fr/serviceweb), et toutes les opérations de ce service (demande de devis, placement de bon de commande, suivi de bon de commande...etc.) peuvent être fournies sur cette même URL. Une application donnée peut alors appeler chacune de ces opérations en utilisant la méthode http : POST, et en incluant le nom de l'opération concernée par la requête dans l'en-tête http « SOAPAction ».

Cette approche, ainsi que d'autres caractéristiques de SOAP, ont fait l'objet de nombreuses critiques. Les arguments principaux mis en avant contre SOAP reposent sur le fait que :

- a) Il rajoute peu de fonctionnalités au-dessus de ce qu'il est déjà possible de faire avec HTTP et XML (sans les extensions apportées par SOAP) ;
- b) En associant plusieurs opérations à une même URL : il rend difficile, voir impossible, l'utilisation de l'infrastructure de « caching » associée au protocole HTTP, qui constitue sans doute l'un des points forts de HTTP.

Une approche alternative pour l'implantation des services Web appelées REST a été définie [Fielding 2000]. Dans cette approche, chaque opération d'un service est associée à une URL distincte et l'accès à chaque URL peut être réalisé en utilisant l'une des quatre méthodes fournies par HTTP : POST, GET, PUT et DELETE. Le contenu des messages est alors encodé en XML, et la distinction entre en-tête et contenu de message est laissée à la charge des applications qui requièrent cette distinction. Le résultat est un ensemble de conventions plus simples que celles de SOAP, et la possibilité d'utiliser des bibliothèques existantes pour l'échange de messages sur HTTP, éliminant ainsi le besoin de plates-formes implantant SOAP, qui dans certains cas peuvent être considérées comme étant plus « lourdes » et plus difficiles à utiliser.

Plusieurs services Web populaires disponibles à l'heure actuelle utilisent l'approche REST. Ceci est le cas notamment de la maison de ventes au enchères en ligne « eBay », qui rend une partie de ses fonctionnalités, accessibles sous forme de services Web « style REST » (voir

<http://developer.ebay.com/rest>). Il en est de même du site de vente par Internet Amazon (voir <http://www.amazon.com/webservices>). Le débat SOAP vs. REST est encore ouvert et les avis divergent considérablement dans la communauté.

### **I-15- Les avantages des services Web [2] [17]**

- ✓ Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- ✓ Les services Web utilisent des standards et protocoles ouverts.
- ✓ Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges.
- ✓ Basés sur le protocole HTTP, les services Web peuvent fonctionner au travers de nombreux pare-feu sans nécessiter des changements sur les règles de filtrage.
- ✓ Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les services Web existant.

### **I-16- Les inconvénients des services Web [2] [17]**

- ✓ Les normes de services Web dans certains domaines sont actuellement récentes.
- ✓ Les services Web souffrent de performances faibles comparés à d'autres approches de l'informatique répartie telles que : RMI, CORBA, ou DCOM.
- ✓ Par l'utilisation du protocole HTTP, les services Web peuvent contourner les mesures de sécurité mises en place au travers des pare-feux.

### **I-17- Sécurité des services Web [18]**

Les architectures SOA se sont généralisées petit à petit au sein des entreprises pour construire des systèmes capables d'offrir des fonctionnalités partagées via des services.

Ces services peuvent être internes et ne concerner qu'une organisation ou être ouverts sur l'extérieur dans le cadre d'échanges B2B (Business to Business).

Dès lors que l'on propose de la valeur ajoutée ou transporte des données dites sensibles, ces services sont alors confrontés à des buts contradictoires, à savoir :

- Exposer de l'information et la rendre facilement accessible à un tiers (personne ou système),
- Sécuriser l'information pour la rendre uniquement consommable par des personnes ou des systèmes habilités à la voir et à l'utiliser.

*La sécurité* est donc au cœur des préoccupations des entreprises pour garantir la cohérence et la pérennité des systèmes. Pour ce faire, la sécurité doit être prise en compte dès la conception des services web et ceci à tous les niveaux :

- ✓ Conception, spécification.
- ✓ Architecture
- ✓ Développement
- ✓ Déploiement
- ✓ Exécution & exploitation.

Au-delà du tout sécuritaire, une approche pragmatique de la sécurité pilotée par les risques est préférable. Rappelons les fonctions de sécurité disponibles qui pourront s'appliquer à des services :

- *Authentification* : le mécanisme qui permet de vérifier l'identité d'une personne/d'un système.
- *Habilitation* : le droit d'accéder ou non à une fonctionnalité, à une donnée.
- *Intégrité* (ou signature) : la non modification d'une donnée échangée.
- *Imputabilité* : traçabilité des actions d'un individu sur un système.
- *Confidentialité* (ou chiffrement) : la non lisibilité de la donnée par un tiers ne partageant pas un secret. [18]

Des constructeurs d'appliances comme IBM vont également dans ce sens en proposant des boîtiers dédiés à la sécurité des web services. Ces derniers sont capables de modifier à la volée des messages SOAP pour leur appliquer des fonctions de sécurité.



## **I-18- Interopérabilité**

### **I-18-1- Définition [19]**

L'interopérabilité est la capacité que possède un produit ou un système dont les interfaces sont intégralement connues, à fonctionner avec d'autres produits ou systèmes existants ou futurs et ce sans restriction d'accès ou de mise en œuvre.

### **I-18-2- Interopérabilité et services web [20]**

L'objectif initial des services web est de fournir un ensemble de standards permettant d'exposer des services de manière interopérable.

La mode du tout Web Service a rapidement mis en exergue les manques du triptyque de départ (WSDL, SOAP, UDDI) qui n'est plus qu'un diptyque. Par exemple, les aspects transactionnels en sont absents, ce qui impose de gérer des mécanismes de compensation. C'est alors que se sont mis à fleurir les WS-\*. Même après consolidation et épuration, la confusion autour de ces standards est palpable et il semble illusoire d'aboutir à un modèle qui soit interopérable out-of-the-box.

C'est face à ce constat que s'est créé la Web Service Interoperability Organization (WS-I org.), consortium industriel dont l'objectif est d'établir et de diffuser un ensemble de best practices autour des standards Web Service, en vue de garantir l'interopérabilité des différentes implémentations et utilisations qui sont faites de cette pile de standards.

## **I-19- Conclusion**

Dans ce présent chapitre, nous avons abordé des concepts diverses, à savoir : l'Architecture Orientée Services (SOA), les microservices, les Services Web, le fonctionnement et caractéristiques de ces derniers ainsi que leurs avantages et inconvénients ; sans oublier les différents standards associés à cette technologie. Au final, nous avons décrit un standard pour la sécurisation des services web et illustrer la relation « Services Web - Interopérabilité ». En effet, ce chapitre nous a permis de bien comprendre les notions de base associées aux Services Web et le rôle de ces derniers afin d'assurer et de satisfaire l'interopérabilité d'un système d'information de manière générale et qui correspond d'ailleurs à l'aspect technique constituant la problématique de ce projet.

Ceci dit, le chapitre qui suit sera consacré au e-learning.

## Chapitre II

### II-1- Introduction

La fonction principale des plates-formes **e-Learning** est de fournir à l'apprenant les bonnes activités avec les bons outils au bon moment en fonction de ses besoins. Cela nécessite l'application de mécanismes d'animation et de coordination des modules et des activités pédagogiques. Si un système e-Learning est une collection d'activités ou de processus, nous pouvons découper ses fonctionnalités en un certain nombre de fonctions autonomes qui peuvent alors être réalisées séparément sous la forme d'applications autonomes ou de services, en utilisant la technologie des **services Web**.

### II-2- e-Learning

#### II-2-1- Définition [21]

Le **e-learning** est une nouvelle forme d'apprentissage à distance qui repose sur la mise à disposition de contenus pédagogiques via internet ou un réseau intranet. Ces contenus sont organisés en modules qui peuvent être assemblés en parcours de formation personnalisé. Ces modules peuvent aussi être installés directement sur des ordinateurs grâce à des supports électroniques « off line » tels que CD-Rom ou DVD-Rom.

Le e-learning permet donc de se former seul face à un ordinateur sur son lieu de travail ou à domicile. Cette nouvelle approche de la formation autorise une grande souplesse d'utilisation pour l'apprenant qui peut se former à son rythme en fonction de ses besoins et de ses disponibilités.

La combinaison de différents contenus multimédias (textes, illustrations dynamiques , voix-off, vidéos...etc ) crée des environnements de travail riches , attractifs et efficaces ayant un impact fort sur la compréhension et la mémorisation par l'apprenant. Sans oublier, L'interactivité qui permet aussi de stimuler l'intérêt de l'apprenant notamment lors des exercices autocorrectifs.

Le **e-learning** recouvre toutes les méthodes de formation s'appuyant sur l'outil informatique. Cette définition large inclut plusieurs axes indépendants :

- Supports en ligne ou hors-ligne,

- Apprentissage individuel ou collectif,
- Formateur présent, à distance, voir absent.

Ceci dit, d'autres termes sont parfois utilisés autre que « e-learning », tels que :

- Formation en ligne,
- Apprentissage en ligne,
- Enseignement en ligne,
- Formation à distance,
- Enseignement à distance.

### **II-2-2- L'histoire du e-learning**

Discipline pédagogique relativement récente, le ***e-learning*** a tout de même une histoire à raconter. L'expression apparaît aux États-Unis au cours des années 1990 ; mais les premiers « cours en ligne » remontent aux années 1970 et 1980. Ils sont notamment proposés par l'Institut de technologie du New Jersey, l'Université de Guelph ou encore la British Open University.

Néanmoins, le e-learning se développe véritablement en même temps que les technologies liées à Internet que constituent les e-mails, les forums et les logiciels de partage de documents. En Europe, avant la démocratisation du Web "à la maison", le e-learning connaît une première phase de développement avec les supports CD-Rom, qui permettent d'installer des logiciels d'apprentissage sur les ordinateurs personnels.

Avec l'arrivée du Web 2.0, le e-learning change et s'enrichit : outils collaboratifs, plateforme LMS, amélioration continue des savoirs et des savoirs-faire, échanges entre apprenants ; en somme, le e-learning est passé à la vitesse supérieure !

### **II-2-3- Les aspects de la formation en ligne (e-learning) [22]**

On peut distinguer deux aspects de la formation en ligne, à savoir :

**II-2-3-1- Formation exclusivement en ligne (sans tutorat) :** Ce type de formation est proche du service fourni par un cédérom ou d'une vidéo de formation. Cette formation n'apporte

pas beaucoup de valeur ajoutée. On la trouve souvent sur les «portails de formation» aux Etats-Unis.

### **II-2-3-2- Formation exclusivement en ligne (avec tutorat) :**

C'est le modèle de référence des « portails de formation », plutôt destiné au grand public. L'acheteur paie en ligne sa formation, un tuteur lui est attribué et lui propose un programme de travail. Dans ce cas, la formation peut être :

#### **II-2-3-2-1- Asynchrone :**

##### **▪ Présentation :**

La formation asynchrone est une méthode d'apprentissage s'adaptant aux disponibilités de l'apprenant. Celui-ci a accès à un ou plusieurs instruments (exemples : vidéo, enregistrement audio, texte, logiciel d'apprentissage virtuel) qu'il utilisera à sa guise. Le suivi de la formation avec le formateur ou avec les membres d'un groupe d'apprentissage se fera par voie indirecte ( courriel, forum de discussion...etc ).

##### **▪ 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.

#### **II-2-3-2-2- Synchrone :**

##### **▪ Présentation :**

La formation synchrone, au contraire de la formation asynchrone, se caractérise par l'interaction directe et en temps réel entre les apprenants et les formateurs. C'est la méthode la plus traditionnelle, celle qui s'approche le plus de la classe magistrale.

Lorsque tous les apprenants d'un groupe sont simultanément en ligne avec leur formateur et échangent entre eux au moyen du clavardage, partagent des applications, visionnent les mêmes écrans ou encore reçoivent des images de visioconférence (Webcast), ils pratiquent l'apprentissage virtuel synchrone.

##### **▪ Principaux avantages de la formation virtuelle synchrone :**

- ✓ Les apprenants interagissent à l'écran avec les formateurs ; le langage oral et visuel sont utilisés.
- ✓ On peut rapidement créer du contenu prêt à diffuser.
- ✓ La dynamique de groupe s'installe plus rapidement.

### **II-2-3-2-3- Formation mixte :**

#### **▪ Présentation :**

C'est plutôt le modèle des universités, écoles et organismes de formation. En mettant en ligne le contenu de la formation, des tests, des évaluations et un tutorat, cela permet de réduire le temps de présentiel et d'individualiser la formation. Le présentiel permet de faire le point, de répondre à des interrogations, voir d'approfondir le sujet.

Ce modèle de formation combine donc les éléments de l'apprentissage en ligne et de l'apprentissage traditionnel en classe.

#### **▪ Quelques exemples de formules d'apprentissage mixte :**

Avant de bénéficier d'un enseignement traditionnel en classe, les apprenants reçoivent du matériel ayant recours aux technologies de l'information (Internet, cédéroms). Le temps de rencontre est diminué par cette préparation et la formation en classe se concentre sur les échanges approfondis.

- ❖ Entre deux séances de formation en classe, les participants poursuivent leur formation au moyen d'échanges virtuels structurés et d'outils de collaboration.
- ❖ Une classe se réunit en temps réel (mode synchrone : vidéoconférence, clavardage) et se poursuit en différé (forum de discussion, échange de courriels).
- ❖ Du support (coaching) est apporté aux apprenants, de façon virtuelle ou sur les lieux de travail, afin de faciliter l'intégration des connaissances acquises à travers une formation virtuelle.

#### **▪ 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.

## II-2-4- Les outils de communication et de collaboration du e-learning [23]

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

### II-2-4-1- les outils synchrones :

La formation en ligne permet l'usage de ressources synchrones. Ces dernières impliquent un environnement d'apprentissage dans lequel le formateur et les apprenants sont réunis simultanément. Dans ce contexte, les apprenants doivent participer aux activités selon un horaire précis et le formateur doit dispenser la matière et les rétroactions en temps réel.

A titre d'exemple :

- Classe virtuelle,
- Audio ou vidéoconférence,
- Chat,
- Messagerie instantanée.

### ✓ **Avantages :**

- ❖ **Permettre au formateur d'interagir en temps réel avec les apprenants :** et ainsi de juger instantanément leur niveau d'engagement et de compréhension de la matière. Le formateur peut s'adapter et fournir des explications supplémentaires pour clarifier des concepts. De la même manière, les apprenants peuvent recevoir une rétroaction instantanée de la part du formateur s'ils ont des questions ou des problèmes.
- ❖ **Offrir un espace de rencontre aux équipes de travail dispersées géographiquement :** qui, autrement, ne seraient pas en mesure d'interagir simultanément ou de se rencontrer.
- ❖ **Encourager l'esprit de groupe, la résolution de problèmes et la collaboration en temps réel** entre les apprenants, lesquels sont « connectés » simultanément les uns avec les autres.

- ❖ **Réduire les coûts liés à la formation en classe** : Selon the e-Learning Guild, les organisations qui misent sur la formation en classe consacrent, en moyenne, environ 70 % de leur budget de formation total au déplacement, à l'hébergement et à la nourriture.

#### **II-2-4-1- les outils asynchrones :**

À l'opposé, les ressources e-learning asynchrones impliquent un environnement d'apprentissage dans lequel le formateur et les apprenants n'ont pas besoin d'être connectés au même moment. Le contenu pédagogique peut être consulté au moment le plus opportun et l'apprenant peut compléter sa formation de façon individuelle. Des exemples :

- ❖ Cours d'autoformation en ligne,
- ❖ Groupe de discussion (Forums),
- ❖ Matériel pédagogique accessible en ligne (exercices, évaluations à rétroactions différées...etc),
- ❖ Tutoriels,
- ❖ Blogues.

#### **✓ Avantages :**

- ❖ **Offrir de la flexibilité à l'apprenant** : qui peut intégrer aisément la formation asynchrone dans son horaire de travail régulier et optimiser son temps.
- ❖ **Permettre à l'apprenant d'apprendre à son rythme** : il peut parcourir à nouveau une portion de la matière qui est mal comprise ou prendre plus de temps pour examiner les aspects d'une notion avant de répondre à une question ou de poursuivre son apprentissage. À l'inverse, il peut aussi sauter des modules portant sur des connaissances qu'il maîtrise déjà.
- ❖ **Transmettre un contenu lourd et fastidieux** : comme de la théorie axée sur des procédures ou des processus. Ce type de contenu est généralement transmis plus efficacement avec la formation en ligne asynchrone.
- ❖ **Prolonger la réflexion et la discussion** : en permettant à l'apprenant de contribuer lorsqu'il le souhaite en partageant aux autres apprenants du contenu supplémentaire

*ou en renchérissant sur des concepts sans interrompre le rythme ou allonger le temps alloué en classe. Cela peut favoriser une réflexion plus approfondie.*

- ❖ **Permettre à l'apprenant de réfléchir sur son propre parcours de formation et d'améliorer l'efficacité de ses apprentissages.**

## **II-2-5- Avantages et Inconvénients du e-learning de point de vue utilisateurs**

[24] [25]

Si la formation e-learning détient de nombreux atouts, le système possède aussi ses limites. Le tableau ci-dessous tend à énumérer les avantages et les inconvénients attribués au e-learning, du point de vue des trois acteurs de la formation e-learning : l'apprenant , le formateur et l'entreprise.

### **II-2-5-1- Du côté de l'Apprenant :**

Avantages	Inconvénients
Apprenant acteur de sa formation e-learning	Appréhension de l'outil informatique, réticence face aux nouvelles technologies
Interactivité et attractivité du contenu e-learning	Gestion de son autonomie car pas de cadre de travail imposé
Flexibilité et adaptabilité selon ses disponibilités (heure et lieu)	Gestion de sa motivation et de son implication dans sa formation e-learning, effort de concentration
Formation à son propre rythme indépendamment des autres apprenants	Pas de contacts directs avec le formateur (sauf en cas de formation blended learning)
Auto-évaluation en cours et en fin de cursus	
Suivi personnel de son avancement dans la formation e-learning et bilan des résultats obtenus grâce au tracking	

**Tableau II.3 : Avantages et Inconvénients coté Apprenant.**



**II-2-5-2- Du côté du Formateur :**

Avantages	Inconvénients
Pré-requis pour évaluer le niveau des apprenants	Pas de contacts directs avec l'apprenant (sauf en cas de formation blended learning)
Formation sur des outils interactifs et attractifs pour lui et les apprenants	Transformation du rôle de « sachant » en rôle de « chef d'orchestre »
Flexibilité et adaptabilité selon les disponibilités (heure, lieu)	Passage d'une communication majoritairement orale à écrite
Suivi des apprenants depuis la plateforme (tracking)	

**Tableau II.4 : Avantages et Inconvénients coté Formateur.****II-2-5-3- Du côté de l'entreprise :**

Avantages	Inconvénients
Formation « de masse » (nombre d'apprenants illimité)	Entreprises parfois mal informées et frileuses face aux nouvelles technologies
Economie sur les coûts indirects liés aux formations en présentiel (déplacement, logement etc.)	Pas de contrôle sur la motivation, l'implication et la gestion de leur formation par les apprenants
Flexibilité et adaptabilité selon les disponibilités des apprenants (heure et lieu)	Investissement en matériels informatiques et en logiciels si les outils ne sont pas disponibles en mode SaaS
Individualisation et ajustement des parcours de formation suivant les compétences et les objectifs pédagogiques prédéfinis	Contenus e-learning parfois difficiles à concevoir pour des formations dans des domaines très pointus
Contraintes logistiques réduites (pas de location de salle, de déplacement des salariés, de logement etc.) Conduite du changement à mettre en œuvre dans le service formation	Conduite du changement à mettre en œuvre dans le service de formation.
Reporting précis des formations et analyse des résultats automatisée grâce au tracking	
Supports de formation pérennes et actualisables	

**Tableau II.5 : Avantages et Inconvénients coté Entreprise.**

## II-2-6- Avantages et Inconvénients du e-learning de manière générale [23]

### II-2-6-1- Avantages :

- ✓ ***Pas de limite d'âge*** : la formation n'est ouverte à toute personne, quel que soit son âge, son niveau d'instruction ou sa catégorie socioprofessionnelle.
- ✓ L'accès aux informations, aux savoirs et aux savoirs- faire ***sans limites de distance***.
- ✓ Favorise la ***créativité et l'esprit de découverte***.
- ✓ ***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***.
- ✓ ***Autonomie de la formation*** : les conditions spatio-temporelles de la formation sont choisies par l'apprenant.
- ✓ ***L'apprenant est le centre de concentration*** et non plus le formateur : l'apprenant est incité à être un émetteur de feed-back et de participer d'une manière significative à la formation et ne plus se limiter à être un récepteur d'informations et de savoirs comme c'est le cas des étudiants à la traditionnelle.
- ✓ 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 l'observation et le captage.
- ✓ ***Créer un sentiment de liberté et de confiance chez l'apprenant*** : les sentiments d'intimité et de honte des collègues et du formateur en cas de faute par exemple se dissipent en cas du e-learning. L'apprenant étudie tout seul, face à son ordinateur et il n'est pas observé par les autres ( excepté le cas de visioconférence ).
- ✓ ***Performance*** : de nombreuses études mettent en avant de meilleurs résultats de la formation en e-learning que lors d'une formation présentielle traditionnelle.

### II-2-6-1-2- Inconvénients :

- ✓ ***L'absence physique de l'enseignant*** avec tout son poids d'émotions, d'autorité et d'expressivités humaines.
- ✓ ***Les problèmes techniques*** afférents au fonctionnement des systèmes de formation (perturbation du réseau de communication ; pannes des ordinateurs, terminaux ou

serveurs ; attaques des documents électroniques de cours par des virus ou des pirates informatiques ...etc).

- ✓ **Autodiscipline** : les apprenants doivent faire preuve de rigueur et de discipline, particulièrement s'ils sont isolés dans une formation à distance.
- ✓ **Isolement** : les contacts avec les autres apprenants de la formation et avec le formateur sont réduits, voir inexistant.
- ✓ **Maîtrise des outils** : le e-learning nécessite une maîtrise suffisante des outils informatiques et d'Internet pour pouvoir suivre la formation.

## II-2-7- Les standards du e-learning [26]

Le e-learning a donné lieu à plusieurs standards, à savoir : AICC, IMS, SCORM, LOM et QTI.

### ❖ Présentation :

Face au développement du e-learning, des normes ont été créées pour encadrer ses pratiques. Ainsi, les contenus utilisés dans le cadre de l'e-learning répondent généralement à un ou plusieurs des standards tel que : AICC, IMS, SCORM, LOM et QTI. Ces derniers ont plusieurs objectifs dont les principaux sont :

- Uniformiser l'indexation des divers contenus,
- Permettre le partage des contenus pédagogiques entre différents environnements,
- Assurer l'interopérabilité des contenus entre eux et avec les plateformes d'e-learning.

A des fins d'illustration, l'un de ces standards est présenté ci-après :

### c) SCORM :

SCORM est un modèle visant à standardiser l'indexation et le partage des contenus pédagogiques utilisés en e-learning. Il regroupe lui-même plusieurs normes (AICC, LOM, IMS...etc).

Il considère les domaines suivants :

- La gestion des contenus : métadonnées, empaquetage,
- L'environnement d'exécution et la communication avec le LMS,

- Le modèle de navigation : interprétation des règles de navigation déclenchées par l'action de l'apprenant.

## **II-2-8- Plate-forme e-learning [27]**

### **II-2-8- 1- Définition :**

Une plate-forme e-learning ou bien pédagogique est un logiciel qui assiste la conduite des formations présentielles et à distance. Elle est basée sur des techniques de travail collaboratif et regroupe les outils nécessaires aux trois principaux acteurs de la formation : apprenant, tuteur et administrateur. Elle fournit à chaque acteur un dispositif qui a pour première finalité l'accès à distance au contenu pédagogique, l'auto apprentissage, l'autoévaluation et le télé tutorat via l'utilisation des moyens de travail et de communication à plusieurs, tel que : la visioconférence, e-mail , forums, chats, annotations, etc. Le but étant de combler la perte de cohésion et de stimulation de la salle que peut sentir l'apprenant devant sa machine.

- ✓ L'usage de ce système est relativement standard, le tuteur crée des parcours de formation, incorpore des ressources pédagogiques multimédias et de suivi des activités des apprenants.
- ✓ L'apprenant, peut consulter en ligne ou télécharger les contenus pédagogiques qui lui sont recommandés, effectuer des exercices, s'auto évaluer et transmettre des travaux à son tuteur pour les corriger. La communication entre apprenant et tuteur peut être individuelle ou en groupe. De plus, Il est possible de créer des thèmes de discussion et collaborer à des travaux communs en utilisant des moyens de travail et de communication à plusieurs.
- ✓ L'administrateur, de son côté, assure l'installation et la maintenance du système, gère les droits d'accès, crée des liens vers d'autres systèmes et ressources externes.

Ainsi, une plate-forme peut comporter des fonctionnalités relatives à la gestion des compétences, à la gestion des ressources pédagogiques, à la gestion de la qualité de la formation.

## **II-2-8- 2- Principaux critères que doivent vérifier les plates-formes e-learning :**

- ❖ Le coût total de la mise en place de la plate-forme (gratuite ou non). Dans le cas d'une plate-forme payante, vérifier les prestations fournies.
- ❖ Les besoins de maintenance de la plate-forme d'un point de vue strictement informatique, tout ce qui devra être géré par les administrateurs (voir également le coût en personnel de maintenance).
- ❖ La sécurité de la plate-forme d'un point de vue purement informatique (filtrer les IPs qui auront accès aux cours) et au niveau des cours (login, mot de passe).
- ❖ Le nombre d'utilisateurs maximum que pourra supporter la plate-forme.
- ❖ La possibilité de mettre en place des QCM (Questions à Choix Multiples) sur la plate-forme, s'il est possible d'insérer une image et si le système de questions est interactif (avec animations)
- ❖ Un forum de discussion (dont l'accès sera étroitement lié à la sécurité interne de la plateforme).
- ❖ La technologie utilisée, de point de vue du langage pour pouvoir éventuellement faire évoluer le système et sa capacité d'accueil (XML, MySQL, Php...).
- ❖ Un système de gestion de mails, pour qu'un professeur puisse éventuellement envoyer des mails à l'une de ses classes, également lui permettre de gérer des groupes.
- ❖ La façon de poster un cours (à partir de documents PowerPoint, enregistrer au format HTML et différents formats).
- ❖ Les moyens requis pour consulter la plate-forme (browser, OS).
- ❖ La possibilité de diffuser les cours par vidéos soit en temps réel, soit en différé.

## II-2-9- LMS ou système de gestion des cours et de l'apprentissage [27]

Le e-learning s'appuie sur des contenus pédagogiques mis à disposition des apprenants. Le *LMS (Learning Management System)* ou *SGC (Système de Gestion des Cours)* est donc au cœur du e-learning. Il s'agit souvent de systèmes modulaires pour lesquels on peut choisir différents composants qui offrent diverses fonctionnalités.

### II-2-9-1- Présentation

Le LMS est l'interface informatique entre le formateur et les apprenants. Ses fonctionnalités dépassent de loin le simple dépôt de documents à télécharger. La plupart des logiciels de LMS permettent d'activer ou de désactiver des modules en fonction des besoins identifiés par le formateur et les apprenants.

### II-2-9-2- Composants

#### II-2-9-2-1- Outils pédagogiques :

Le cœur de métier du LMS reste la gestion des contenus pédagogiques. Les composants suivants sont donc généralement un point commun des plateformes d'e-learning :

- ✓ *Affichage des contenus pédagogiques,*
- ✓ *Tracking* : le principe est de suivre le parcours des apprenants (résultats des évaluations, temps passé sur les modules...etc ). Cela permet au formateur d'évaluer la qualité des contenus pédagogiques et l'assiduité des apprenants,
- ✓ *Quiz* : ils permettent l'évaluation et l'auto-évaluation des apprenants,
- ✓ *Outils de conception* du contenu pédagogique.

#### II-2-9-2-2- Outils de communication :

En addition des outils pédagogiques, le LMS propose des moyens de communication diverses entre l'apprenant et le formateur ou entre co-apprenants, tel que :

- ❖ **Messagerie** : c'est un système de communication asynchrone ( c'est-à-dire en décalé), qui permet aux membres d'un groupe ( apprenants / enseignants ) de

communiquer et l'envoi ainsi que la réception de mails avec ou sans fichier attaché .De plus , Il peut être interne à la plate-forme et ne nécessite pas d'avoir un e-mail personnel.

- ❖ **forums** : système de communication asynchrone où tous les participants au forum peuvent suivre toutes les discussions,
- ❖ **chat ou messagerie instantanée** : système de communication synchrone ( c'est-à-dire en simultané) permettant à plusieurs personnes de discuter en temps réel,
- ❖ **audio ou vidéoconférence** : chaque participant équipé d'un micro et d'une caméra peut prendre part à la vidéoconférence. C'est le moyen de communication le plus proche d'une interaction en présentiel,
- ❖ **Tableau blanc interactif** : il peut s'agir d'un tableau interactif réel, utilisé par le formateur lors de sessions vidéo et utilisable à distance. Il peut aussi s'agir d'une zone partagée dans le navigateur sur laquelle les utilisateurs peuvent partager des schémas ou des textes.
- ❖ **Documents partagés** : un dépôt de documents partagés permet de distribuer des documents structurés à une partie ou à tous les apprenants,
- ❖ **Wiki** : site web librement éditable,
- ❖ **Sondages** : ils permettent de recueillir l'avis des apprenants sur les contenus, les dates, les fonctionnalités de la plateforme...etc.
- ❖ **Calendrier et agenda** : ils servent à définir des dates butoir pour certains événements (évaluations, sessions présentiels...etc).
- ❖ **Les FAQs** : cet espace permet aux tuteurs et aux administrateurs formuler la réponse à des questions récurrentes afin de satisfaire un bon nombre de personnes visitant la

plate-forme et leurs questionnements sur l'utilisation ou les modalités liées à la plate-forme.

- ❖ **Les news** : la possibilité de créer des news permettra aux enseignants d'informer ses apprenants de nouveaux rendez-vous, ou de leur communiquer diverses informations.

### **II-2-9-2-3- Principales plates-formes LMS :**

Il existe de nombreuses solutions open-source pour mettre en œuvre une plateforme e-learning. Les plus connues sont : Claroline, Ganesha, Moodle, Docebo , Dokeos et Sakai.

#### ❖ **Moodle**

Moodle a été conçu comme une plateforme d'e-learning extrêmement modulaire. Il est compatible avec les normes SCORM, AICC, QTI et IMS. La communauté d'utilisateurs de Moodle est l'une des plus importantes dans le domaine de l'e-learning open-source. De nombreuses sociétés proposent de l'assistance et du conseil autour de Moodle.

#### ❖ **Dokeos**

Dokeos est un projet dérivé du projet Claroline. Plusieurs éditions sont disponibles , à savoir :

- Free : la version libre et gratuite à support limité,
- Education : inclut la version Free et dispose de fonctionnalités orientées éducation ainsi qu'un support commercial,
- Pro : inclut la version Free et dispose de fonctionnalités spécifiques au monde professionnel ainsi qu'un support commercial,
- Medical : inclut la version Free et dispose de fonctionnalités spécifiques au monde médical ainsi qu'un support commercial.

#### ❖ **Sakai**

Sakai a été développé pour un contexte universitaire. Il dispose des fonctionnalités d'un LMS classique. Celui-ci propose en plus, des outils pour la collaboration des apprenants dans le cadre des groupes de travail ou des groupes de recherche.



#### **II-2-9-2-4- Besoins des utilisateurs de la plate-forme LMS :**

Une plate-forme est un système d'information qui doit satisfaire certains besoins et critères , tels que :

- ✓ Reposer sur les technologies de l'Internet,
- ✓ Satisfaire les normes (LOM, AICC, IMS, SCORM...etc ) ,
- ✓ Permettre de gérer plusieurs types d'activités pédagogiques : cours, exercices, communication...etc.

Ce type de système regroupe des outils nécessaires aux principaux utilisateurs, à savoir : l'enseignant , l'apprenant et l'administrateur. Ces dispositifs assurent en finalité la consultation à distance des contenus pédagogiques , le suivi de l'apprentissage et les interactions entre les trois type d'utilisateurs.

##### **❖ *Besoins de l'enseignant :***

- Création d'un contenu pédagogique multimédia,
- Création des parcours pédagogiques types ou individualisés,
- Suivi des activités des étudiants,
- Mettre à jour les contenus pédagogiques.

##### **❖ *Besoins de l'apprenant :***

- Consulter en ligne ou télécharger les contenus pédagogiques qui lui sont recommandés,
- Effectuer les exercices qui lui sont soumis,
- Répondre à des questionnaires proposés,
- Obtenir une évaluation de son parcours individualisé.

Sachant que les enseignants et les apprenants ont des besoins communs de communication mutuelle ou en groupe par des thèmes de discussion et de collaboration à des documents communs.

❖ ***L'administrateur d'un système e-Learning :***

- Installer et assurer la maintenance de la plate-forme,
- Créer les utilisateurs de la plate-forme,
- Gère les droits d'accès des utilisateurs,
- Crée des liens avec les systèmes d'information externes (scolarité , catalogues , ressources pédagogiques...etc ).

### **II-2-9-3- Objectifs de la plate-forme LMS**

Une plateforme LMS a comme objectifs de permettre :

- ✓ La mise à disposition des ressources pédagogiques ( textes, exercices, tests, évaluations... etc) ,
- ✓ La communication entre les formateurs et les apprenants au moyen d'un forum, chat ou dépôts de fichiers) ,
- ✓ Faire la gestion administrative des formations (inscriptions, bulletins de notes...etc) ,
- ✓ La plateforme LMS permet de mettre à disposition tout type de contenus de cours. En effet, Il y a une grande liberté dans les formats des fichiers de cours. Cette liberté est appréciée des enseignants qui veulent chacun avoir leur propre type de support ( Slides PowerPoint , documents Word , html , pdf...etc).

### **II-2-9-4- Quelques fonctionnalités de la plate-forme LMS**

- Modules d'édition des contenus de cours,
- Gestion des inscriptions,
- Dépôts de fichiers,
- Gestion des notes,
- Module « actualités ».

## **II-2-10- Situation du e-learning en Algérie**

### **II-2-10-1- Le développement de l'e-learning en Algérie [28]**

La formation par Internet a connu un premier succès en Algérie en 2007 grâce au Centre National de l'Enseignement Professionnel à Distance (CNEPD) et son projet de certification ICDL, «International Computer Driving Licence». Visant à mesurer les compétences en informatique sur un standard international, cette certification a rencontré un vif succès et ne comptait pas moins d'une dizaine de millions de candidats.

Face à un tel engouement, l'Algérie a lancé, dès 2008, une plateforme interactive connue sous le nom de Tarbiatic. Celle-ci permettait de relier tous les acteurs de la formation et de l'enseignement, en favorisant la mobilité des élèves et des enseignants. Elle a engendré l'inscription de près de 60 établissements scolaires et de 20 000 élèves en seulement une année d'existence.

Si cette plateforme visait les candidats aux épreuves du bac, elle a réussi à convaincre d'autres acteurs de l'enseignement à adopter la solution de l'e-learning et à miser sur le jumelage des connaissances. Ainsi, plusieurs universités algériennes telles que Sétif, Batna, Constantine ou Biskra ont travaillé main dans la main pour encourager la formation à distance. Plusieurs autres portails e-learning ont également vu le jour et ont investi dans l'apprentissage à distance. Parmi ces initiatives, nous citerons entre autres les portails des technologies de l'information et de la communication ou l'Université Virtuelle Algérienne.

### II-2-10-2- Exemple de plate-forme e-learning en Algérie [29]

En 2017, une plateforme E-learning 100% algérienne a été lancée. Elle est destinée aux entreprises pour répondre à des besoins de formations de masse pour les grandes entreprises et organismes publics.

« **Beeform** propose des formations qualifiantes sur mesure, que ce soit pour des formations standards comme les outils bureautiques, les ressources humaines, le management, ainsi que des formations spécifiques telles que le transfert de compétences sur de nouvelles procédures internes (métiers) , mise à niveau des collaborateurs lors de l'installation d'un nouveau système d'information et toute autre opération nécessitant des formations rapides et de masse », a souligné Omar Ali Yahia, le fondateur de la plateforme E-learning « **Beeform** » lors d'une cérémonie de lancement de la plateforme à Alger.

« Le point positif dans cette initiative c'est qu'elle est basée en Algérie. Jusque-là, l'Algérie importe le savoir et on utilise des plateformes installées aux Etats-Unis, en France et en Allemagne, et enfin, on a vu naître une plateforme DZ avec un contenu DZ », a déclaré Younes Grar, expert en TIC, présent à la cérémonie.

De son côté, Hacen Chaib, directeur général de Algeria Learning Center (ALC) et un des signataires de la première convention de partenariat avec **Beeform** a déclaré que son école compte apporter du contenu de formation en langues étrangères, notamment l'Anglais qui est très demandé dans les entreprises et « cette plateforme sera un socle de formations sur le territoire national et même au-delà, ce qui fera de l'Algérie un pays exportateur de savoir ».

Pour en citer les premiers pas de première plateforme algérienne de l'e-Learning, Omar Ali Yahia a souligné que **Beeform** « accompagne le projet de la smart city de Tipasa avec tout le

transfert de compétences que cela implique. Ceci s'est matérialisé avec la signature d'une convention de coopération le jour de lancement ».

Lors de la cérémonie de lancement, plusieurs conventions de coopération ont été signées, notamment avec l'Ecole des Hautes Etudes Commerciales (EHEC) de Koléa, le Centre de recherche sur l'information scientifique et technique (CERIST), Algeria Learning Center (ALC), ISSAL, COSOFT et ACM Consulting.

### **II-2-11- Situation du e-learning dans le monde [23]**

Les principaux pays actuellement exportateurs de services de formation par Internet sont : les États-Unis, la Nouvelle-Zélande, l'Australie et le Canada. Aux États-Unis, les réalisations sont nombreuses et diversifiées. Une caractéristique originale est la forte implication des géants de l'édition qui s'investissent massivement dans l'enseignement à distance.

Au niveau universitaire, Virtual University Campus est le plus important portail d'éducation sur Internet. Cette université a produit plus de 500 cours de qualité à faible coût. Par ailleurs, on trouve des dizaines d'universités virtuelles offrant un ensemble de cours en ligne extrêmement riche. Le gouvernement a également créé « The Distance Education and Training Council (DETC) ».

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

En Nouvelle-Zélande, le gouvernement a créé un comité de pilotage pour le e-learning. Ce dernier, composé de neuf experts, donne des avis au gouvernement quant aux actions à mettre en œuvre pour faciliter, développer et coordonner l'enseignement à distance.

Les exportations de services d'éducation en Nouvelle-Zélande sont estimées entre 380 et 560 Millions € par an.

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

Au Japon, le Parlement a voté une loi fondamentale sur la formation dans une société en réseau grâce aux technologies avancées des télécommunications : « I.T. Basic Law ». Cette loi est entrée en application le 6 janvier 2001.

En 2000, l'Institut National de l'Education Multimédia (NIME) a adressé vingt et une recommandations aux universités pour promouvoir de façon satisfaisante et crédible la formation à distance par l'usage des TIC.

En Grande-Bretagne, le gouvernement britannique investit lourdement en concentrant ses moyens sur deux initiatives : University for Industry et UK eUniversities Worldwide. La première est l'outil national pour la formation des salariés britanniques, la seconde a pour vocation d'être un vecteur de formation initiale et continue de dimension internationale. Par ailleurs, les initiatives des universités continuent à s'exprimer dans le cadre de Scottish Knowledge ou de l'Open University.

À l'University for Industry (UFI), les cours sont accessibles sur l'ensemble du territoire dans plus de mille centres de proximité et UK eUniversities Worldwide se présente comme la première initiative nationale d'enseignement à distance dans l'enseignement supérieur. Un comité de direction, « e-learning holding company limited » a été créé.

La Communauté européenne met en œuvre un plan d'action e-learning : « penser l'éducation de demain ». Depuis l'adoption de l'« initiative e-learning », en mai 2000, trois groupes de travail ont été mis en place : un avec les États membres, un au sein de la Commission européenne et un troisième avec les industriels.

L'adoption par le Conseil des ministres de l'éducation européens, le 12 février 2001, du rapport sur « les objectifs concrets futurs des systèmes d'éducation et de formation » fournit le cadre général du plan d'action.

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

## **II-2-12- Une architecture de services Web pour l'apprentissage en ligne [30]**

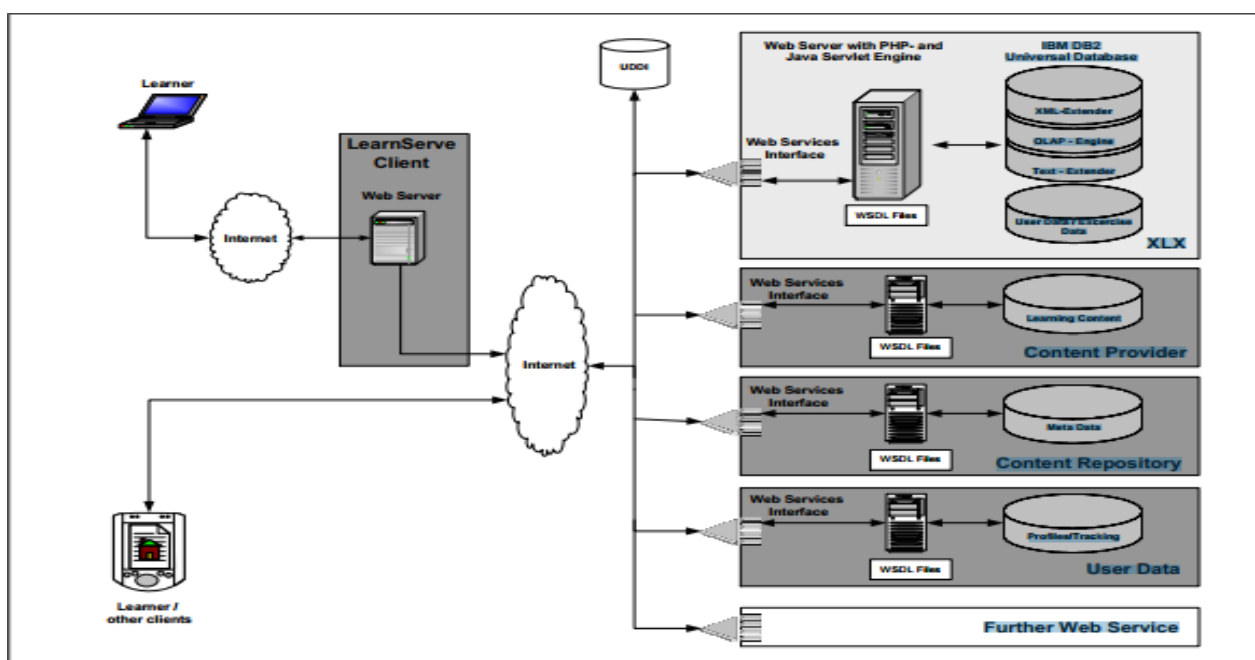
Un service Web est essentiellement un composant logiciel autonome doté d'une URI unique (Uniform Resource Identifier est une adresse unique).

La prémisses de base est que les services Web ont un fournisseur et des utilisateurs ou des abonnés. Ils peuvent être combinés pour en créer de nouveaux avec une fonctionnalité plus complexe. Clairement, Les services Web doivent être interopérables. De plus, ils doivent être indépendants des systèmes d'exploitation et des langages de programmation et ils devraient être capable d'interagir les uns avec les autres.

Pour atteindre ces objectifs, les services Web sont généralement basés sur des normes ; Actuellement, les plus courantes sont les spécifications XML SOAP (Simple Object Access Protocol), UDDI (Description Universelle, Découverte et Intégration) et WSDL (Web Services Description Language).

Les avantages d'une architecture de services Web sont bien reconnus dans le domaine interentreprises (B2B), où les entreprises l'utilisent déjà pour l'intégration d'applications d'entreprise. Même en termes de l'interopérabilité des systèmes B2C (Business-to-Consumer), les services Web obtiennent actuellement importance.

Comme le montre la *figure 1*, LearnServe est divisé en deux parties : un logiciel client et des services Web fournis par plusieurs fournisseurs. Le client LearnServe est le point d'accès pour les utilisateurs qui peuvent utiliser l'apprentissage prestations de service. Ces services sont implémentés sur des serveurs distribués et incluent notamment la création de services de contenu, d'exercices, de suivi et de découverte ainsi que des services de communication tels que les forums de discussions.



**Figure II.1 : Architecture de haut niveau de LearnServe.**



## **II-3- Conclusion**

A travers ce chapitre, nous avons procédé à la présentation et à la description de l'apprentissage en ligne et à distance, autrement dit, le e-learning. En effet, nous avons illustré les différents aspects du e-learning, les outils de communications qui lui sont associés, ses avantages et inconvénients ainsi que des exemples de plates-formes existantes et populaires sur le marché. Pour finir, nous nous sommes intéressés à la situation du e-learning en Algérie et dans le monde.

Ceci dit, ce chapitre nous a permis d'avoir un aperçu sur le fonctionnement d'un système e-learning et d'en tirer les fonctionnalités de ce dernier.

Dans le chapitre qui suit, nous allons entamer la partie Analyse et Conception.

### III-1- Introduction

Nous avons vu dans le chapitre précédent les concepts principaux liés à un système e-learning ainsi qu'une brève description de la situation de ce dernier en Algérie et dans le reste du monde.

Dans ce présent chapitre, on débutera avec la présentation du langage de modélisation « UML » ; Par la suite, on enchaînera par l'identification des besoins fonctionnels et la présentation des diagrammes UML associés au système e-learning dans lequel nos services web de type SOAP seront susceptibles d'intégrer.

### III-2- UML

#### III-2-1- Définition [31]

UML (Unified Modeling Language), ou bien langage de modélisation unifié a été pensé pour être un langage de modélisation visuelle commun et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure et leur comportement.

Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure ainsi que le comportement du système étudié et des objets qui s'y trouvent.

#### III-2-2- Types de diagrammes UML [32] [33]

L'UML utilise des éléments et les associe de différentes manières pour former des diagrammes qui représentent les aspects statiques ou structurels d'un système, ainsi que des diagrammes comportementaux qui capturent les aspects dynamiques de celui-ci. A ce titre, L' UML propose 14 diagrammes depuis sa version 2.3 (UML 2.3) ; on peut d'ailleurs les distinguer selon trois catégories, à savoir :

- ✓ **Les diagrammes de structure** : incluent le diagramme de classes, le diagramme d'objets, le diagramme de composants, le diagramme de structure composite, le diagramme de packages, le diagramme de déploiement et le diagramme de profils.

- ✓ **Les diagrammes de comportement** : incluent le diagramme des cas d'utilisation (utilisé par certaines méthodologies lors de la collecte des exigences), le diagramme d'activité et le diagramme de machine d'état.
- ✓ **Les Diagrammes d'Interaction** : incluent le diagramme de séquence, le diagramme de communication, le diagramme de temps et le diagramme de Vue d'Interaction.

Nous allons à présent définir les diagrammes que nous allons utiliser dans ce présent chapitre :

- **Le diagramme des cas d'utilisation** : celui-ci représente une fonctionnalité particulière d'un système donné, créé pour illustrer la relation entre les fonctionnalités et leurs contrôleurs internes ou externes (acteurs).
- **Le diagramme de séquence** : montre comment les objets interagissent entre eux et leur ordre d'apparition. Ils représentent des interactions pour un scénario particulier.
- **Le diagramme de classe** : c'est le diagramme UML le plus utilisé. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques. Ceci dit, Une classe décrit la structure, le comportement et le type d'un ensemble d'objets (instances d'une classe).

### III-3- Problématique du système étudié

- Les utilisateurs d'une plate-forme e-learning, à savoir les apprenants ainsi que les formateurs doivent pouvoir se connecter et de naviguer sur celle-ci et ce de n'importe quel endroit, autrement dit une accessibilité à distance ; de ce fait, le problème se pose au niveau de l'aspect technique « **Interopérabilité** » qui doit être justement garantie au sein des systèmes e-learning pour pouvoir palier le problème en question.
- Inexistence des **Services Web** au sein des systèmes e-learning pour pouvoir garantir l'interopérabilité de ces derniers.

### III-4- Objectif du projet

- ✓ Mise en place des **Services Web de type SOAP**.
- ✓ Garantir **l'interopérabilité** du système e-learning grâce à la mise en place des services web.

## III-5- Partie Analyse

### III-5-1- Identification des acteurs

Les acteurs concernés par le système sont comme suit :

- **L'administrateur** : c'est le responsable de la plate-forme e-learning dont dispose l'entité pris en considération.
- **L'Apprenant** : c'est l'acteur ayant intégré la plateforme e-learning à des fins d'apprentissage en suivant une formation à distance ou bien de mettre à jour ses connaissances dans tel ou tel domaine.
- **Le formateur** : c'est la personne responsable de l'aspect pédagogique de la plateforme e-learning.

### III-5-2- Identification des besoins

Dans cette partie, nous allons décrire ce que les acteurs cités précédemment pourront faire via une plate-forme e-learning.

#### III-5-2-1- l'Administrateur :

- S'authentifier en tant qu'administrateur sur la plate-forme.
- Gérer les comptes des apprenants et des formateurs.
- Mettre à jour son compte.
- Gérer la messagerie interne.
- Programmer des séances pédagogiques.
- Se déconnecter.

#### III-5-2-2- l'Apprenant :

- S'authentifier en tant qu'apprenant sur la plate-forme.
- Consulter son profil.
- Modifier son compte.
- Consulter son compte.
- Consulter ses cours.
- Télécharger des supports de cours.
- Gérer la messagerie interne.
- Se déconnecter.

**III-5-2-3- Le Formateur :**

- S'authentifier en tant que formateur sur la plate-forme.
- Consulter son profil.
- Modifier son compte.
- Consulter son compte.
- Gestion des cours.
- Gérer la messagerie interne.
- Se déconnecter.

**III-5-3- Identification des cas d'utilisation**

Un cas d'utilisation constitue un moyen de recueillir et de décrire les besoins des acteurs du système ; il permet également de décrire l'interaction entre ces derniers (utilisateurs du cas) et le système lui - même.

Ceci dit, la description de l'interaction est réalisée suivant le point de vue de l'utilisateur. De plus, La représentation d'un cas d'utilisation met en jeu trois concepts, à savoir : l'acteur, le cas d'utilisation et l'interaction entre l'acteur et le cas d'utilisation en question.

A présent, nous allons définir les cas d'utilisation appropriés aux besoins cités précédemment en termes d'acteurs. Autrement dit, nous allons décrire les cas d'utilisation associés à chaque acteur du système e-learning pris en considération.

### III-5-3-1- Administrateur :

#### III-5-3-1-1- Diagramme des cas d'utilisation :

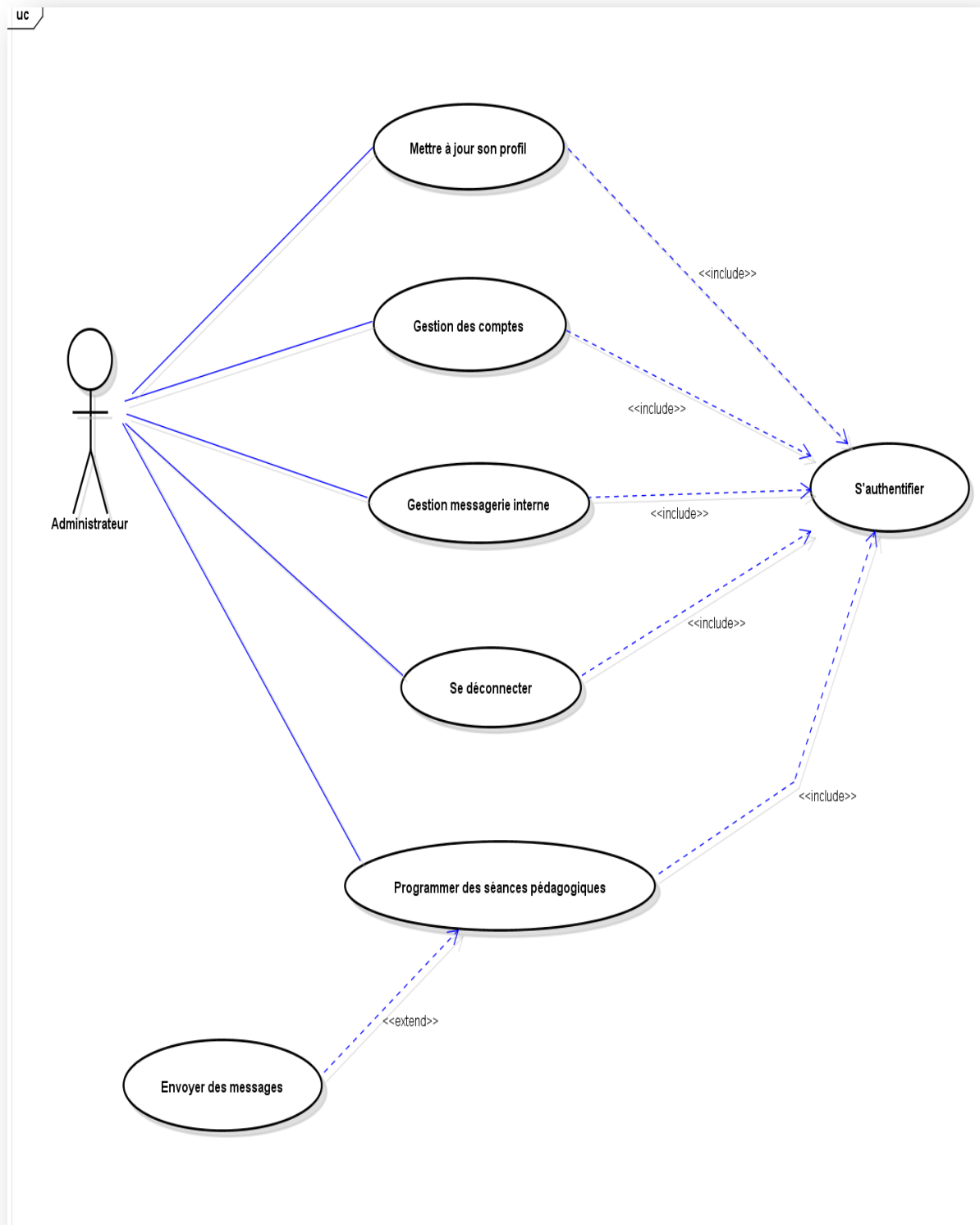


Figure III.1 : Diagramme des cas d'utilisation associés à l'Administrateur.

### III-5-3-1-2- Description textuelle du cas d'utilisation « S'authentifier » :

<b>Nom</b> : S'authentifier sur la plate-forme e-learning.
<b>But</b> : s'identifier afin d'accéder à un compte utilisateur donné.
<b>Résumé</b> : chacun des acteurs (utilisateurs de la plateforme) va devoir saisir son login suivi de son mot de passe afin d'accéder à leurs comptes respectifs.
<b>Acteurs concernés</b> : administrateur, formateur, apprenant.
<b>Pré conditions</b> : chaque acteur accède à la page d'accueil de la plate-forme e-learning considérée pour pouvoir s'authentifier.
<b>Scénario nominal</b> : 1-L'acteur envoie une demande de connexion. 2- Le système affiche une page d'authentification. 3- L'acteur saisit son login et son mot de passe. 4- L'acteur valide les informations saisies.
<b>Exceptions</b> : Exception 1 : le système signale que l'un des champs obligatoires est vide. Exception 2 : le système signale que l'un des champs remplis est incompatible avec le type de données adéquat.
<b>Post conditions</b> : l'acteur en question est authentifié, il accédera donc au compte qui lui est attribué.
<b>Besoins spéciaux</b> : garantir l'interopérabilité grâce aux services web ; en particulier ceux de type SOAP et pour cause : la confidentialité des informations saisies est requise dans ce cas.

**Tableau III.6 : description textuelle du cas d'utilisation « Authentification ».**

### III-5-3-2- Apprenant :

#### III-5-3-2-1- Diagramme des cas d'utilisation :

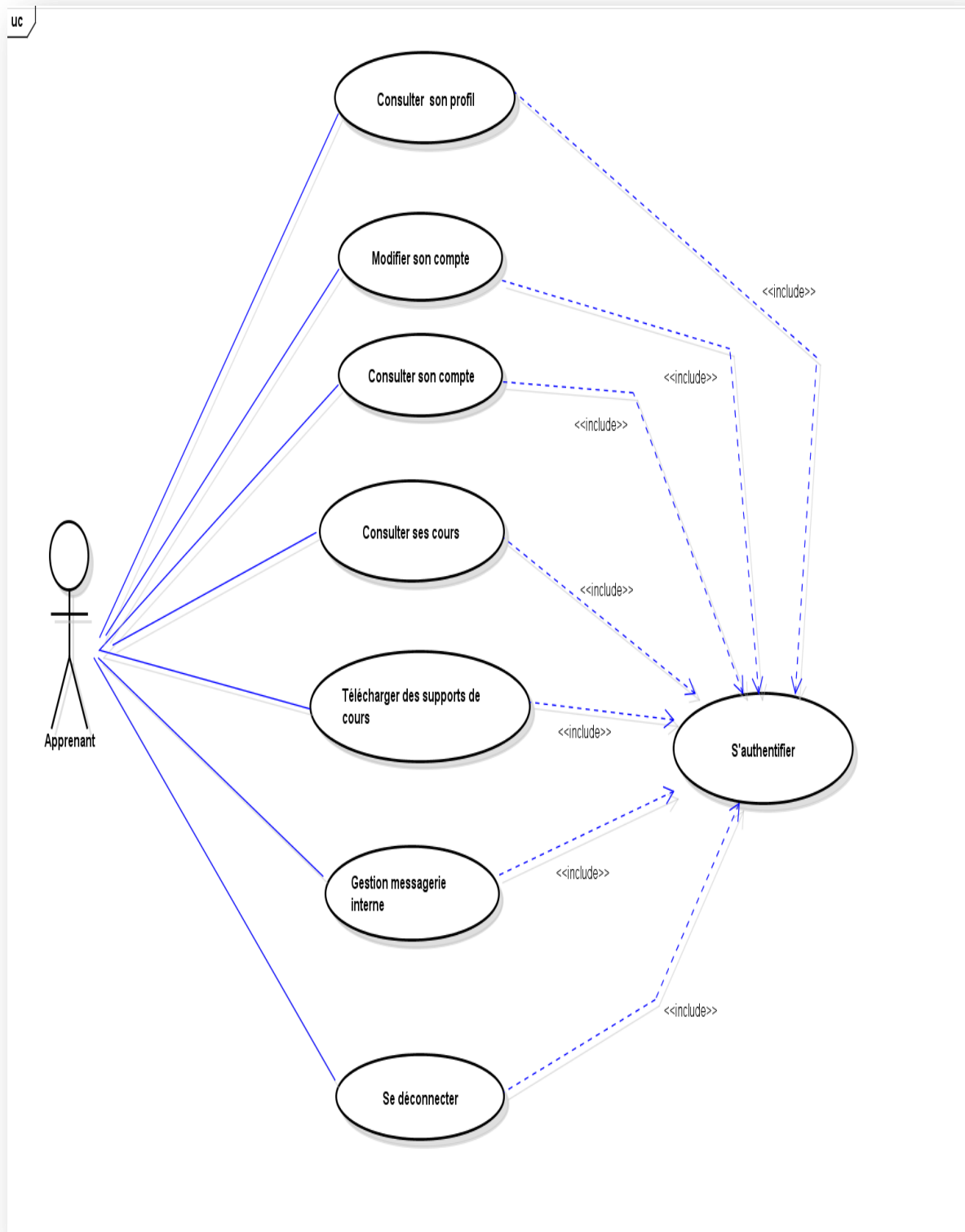


Figure III.2 : Diagramme des cas d'utilisation associés à l'Apprenant.



**III-5-3-2-2- Description textuelle du cas d'utilisation « Modifier compte » :**

<b>Nom</b> : Modifier le compte.
<b>But</b> : permettre aux acteurs de mettre à jour leur compte approprié au sein de la plate-forme e-learning.
<b>Acteur concerné</b> : Administrateur, Apprenant, Formateur.
<b>Pré - conditions</b> : l'acteur concerné doit être authentifié sur la plate-forme e-learning.
<p><b>Scénario nominal</b> :</p> <ol style="list-style-type: none"> <li>1- L'acteur accède à la plate-forme e-learning.</li> <li>2- Il s'authentifie en saisissant son e-mail, mot de passe ainsi que le type (Administrateur/ Apprenant ou bien Formateur)</li> <li>3- Il accède par la suite à son espace, autrement dit au compte qui lui est associé.</li> <li>4- Il clique sur « modifier compte » et il apporte les modifications voulues.</li> <li>5-Il valide enfin les modifications apportées en vue d'être enregistrées.</li> </ol>
<p><b>Exceptions</b> :</p> <p><b>Exception 1</b> : le système signale que l'un des champs obligatoires est vide.</p> <p><b>Exception 2</b> : le système signale que l'un des champs remplis est incompatible avec le type de données correspondant.</p> <p><b>Exception 3</b> : le système signale que le mot de passe saisi au niveau de la confirmation du nouveau mot de passe n'est pas équivalent à celui du champ associé au nouveau mot de passe.</p>
<b>Post - conditions</b> : le compte de l'acteur concerné est bel et bien modifié.
<p><b>Besoins spéciaux</b> : garantir l'interopérabilité grâce aux services web ; en particulier ceux de type SOAP et pour cause : la confidentialité des informations et les transactions possibles (Ajouter, Modifier, Consulter et Supprimer).</p>

**Tableau III.2 : description textuelle du cas d'utilisation « Modifier compte ».**

### III-5-3-3- Formateur :

#### III-5-3-3-1- Diagramme des cas d'utilisation :

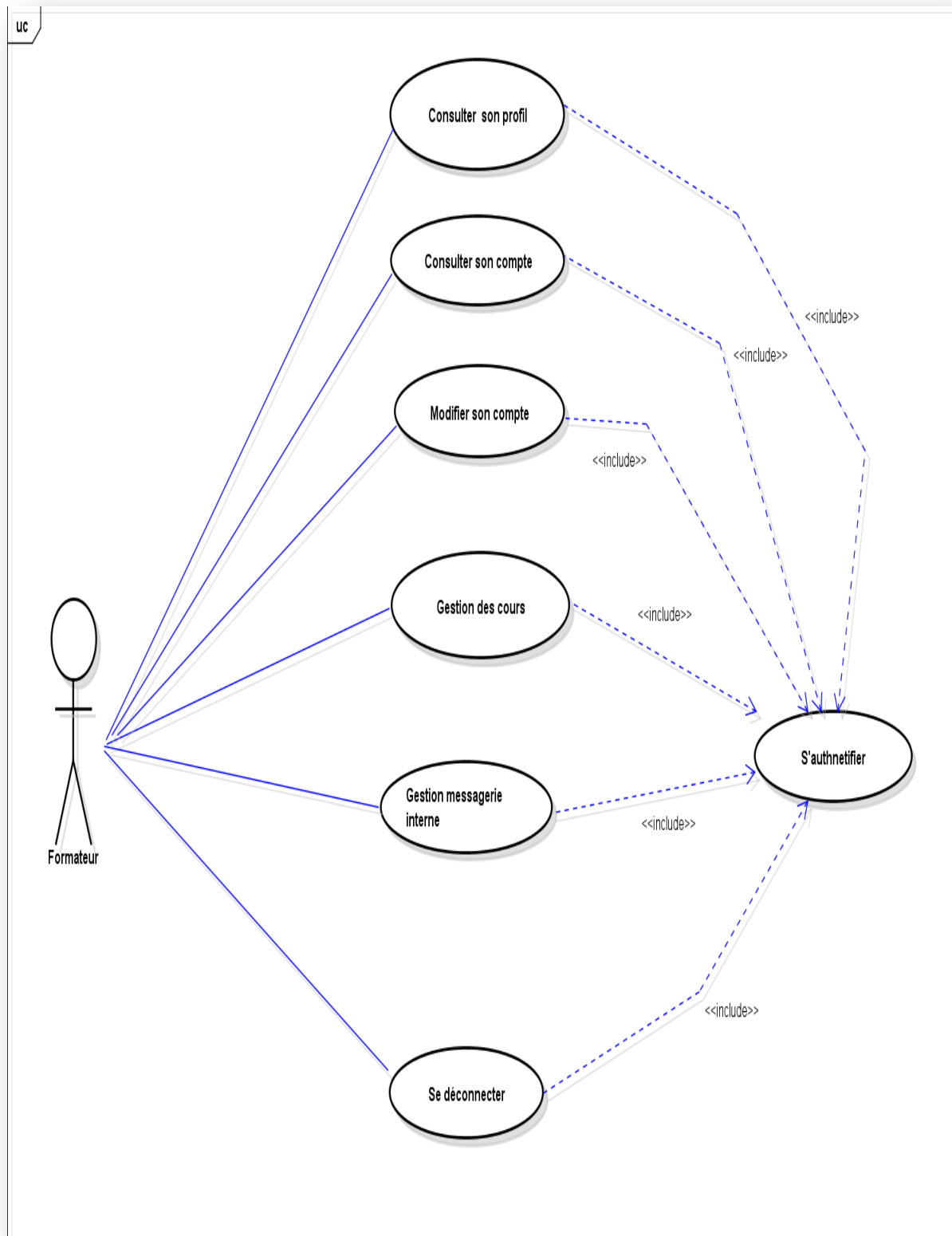


Figure III.3 : Diagramme des cas d'utilisation associés au Formateur.

### III-5-3-3-2- Description textuelle du cas d'utilisation « Gestion cours » :

<b>Nom</b> : gestion des cours.
<b>But</b> : permettre au formateur de gérer ses cours, c'est-à-dire d'ajouter, de modifier, de consulter et de supprimer un cours donné au sein de la plate-forme e-learning.
<b>Acteur concerné</b> : le formateur.
<b>Pré-conditions</b> : le formateur en question doit être authentifié sur la plate-forme e-learning.
<p><b>Scénario nominal</b> :</p> <p>1- Le formateur s'authentifie sur la plate-forme e-learning et il accède au compte qui lui est associé.</p> <p>2- Il clique sur « section cours ».</p> <p>3- Il choisit d'ajouter, de modifier, de consulter ou bien de supprimer un cours donné selon ce qui est voulu par le formateur.</p>
<p><b>Exceptions</b> :</p> <ul style="list-style-type: none"> <li>- Le système signale des erreurs liées à l'incompatibilité des informations saisies avec le type de données requis.</li> </ul>
<b>Post- conditions</b> : traitement voulu effectué (ajout, modification, consultation ou bien suppression).
<p><b>Besoins spéciaux</b> : garantir l'interopérabilité grâce aux services web ; en particulier ceux de type SOAP et pour cause : les transactions possibles (Ajouter, Modifier, Consulter et Supprimer un cours donné).</p>

**Tableau III.7 : Description textuelle du cas d'utilisation « Gestion des cours ».**

### III-5-4- Identification des services web

Après l'analyse des diagrammes des cas d'utilisation de notre système ainsi que **les points d'interopérabilité** éventuels, nous allons présenter ci-après les **Services Web** correspondants sachant que notre travail se focalisera sur ceux de **type SOAP**.

Points Interopérables du Système	
Services Web SOAP	Services Web REST
<i>Service Authentification</i>	<i>Service Gestion Messagerie</i>
<i>Service Compte</i>	
<i>Service Gestion des cours</i>	
<i>Service Gestion des TDs,TPs</i>	

**Tableau III.8 : Identification des Services Web.**

### III-6- Partie Conception

Dans cette partie du chapitre, nous allons présenter des diagrammes de séquence qui correspondent à l'analyse du système e-learning étudié ainsi que la conception de la solution apportée au problème d'interopérabilité que suscitent certaines fonctionnalités de ce dernier. Ceci dit, chaque diagramme de séquence sera suivi par un schéma modélisant le processus lié au service web de type soap qui sera intégré en vue de rendre les fonctionnalités en question **interopérables**.

### III-6-1- Diagrammes de séquences

#### III-6-1-1- Diagramme de séquence du cas d'utilisation « S'authentifier » :

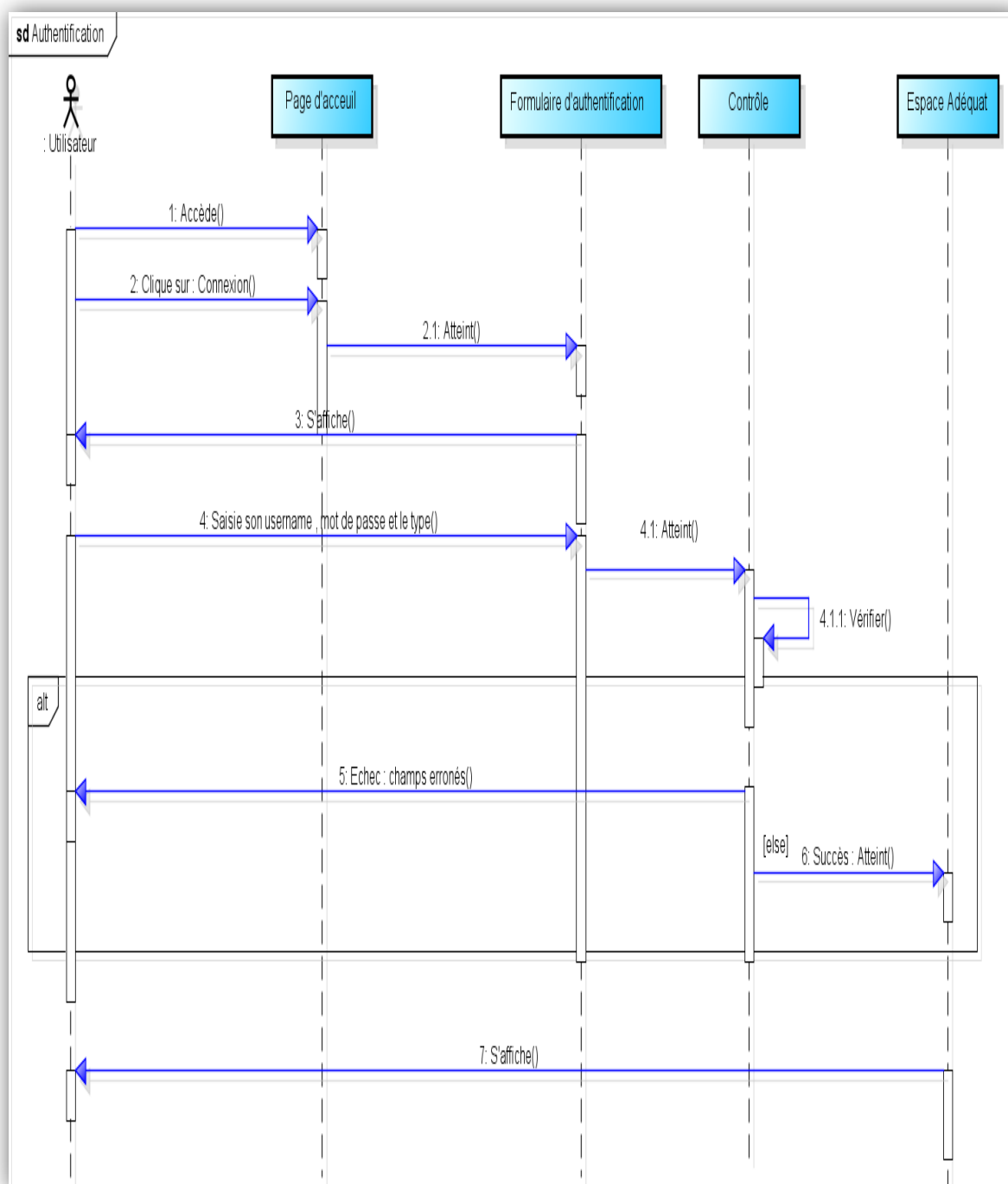


Figure III.4 : Diagramme de séquence du cas d'utilisation "S'authentifier".

### III-6-1-2- Modèle d'utilisation du service web SOAP correspondant :

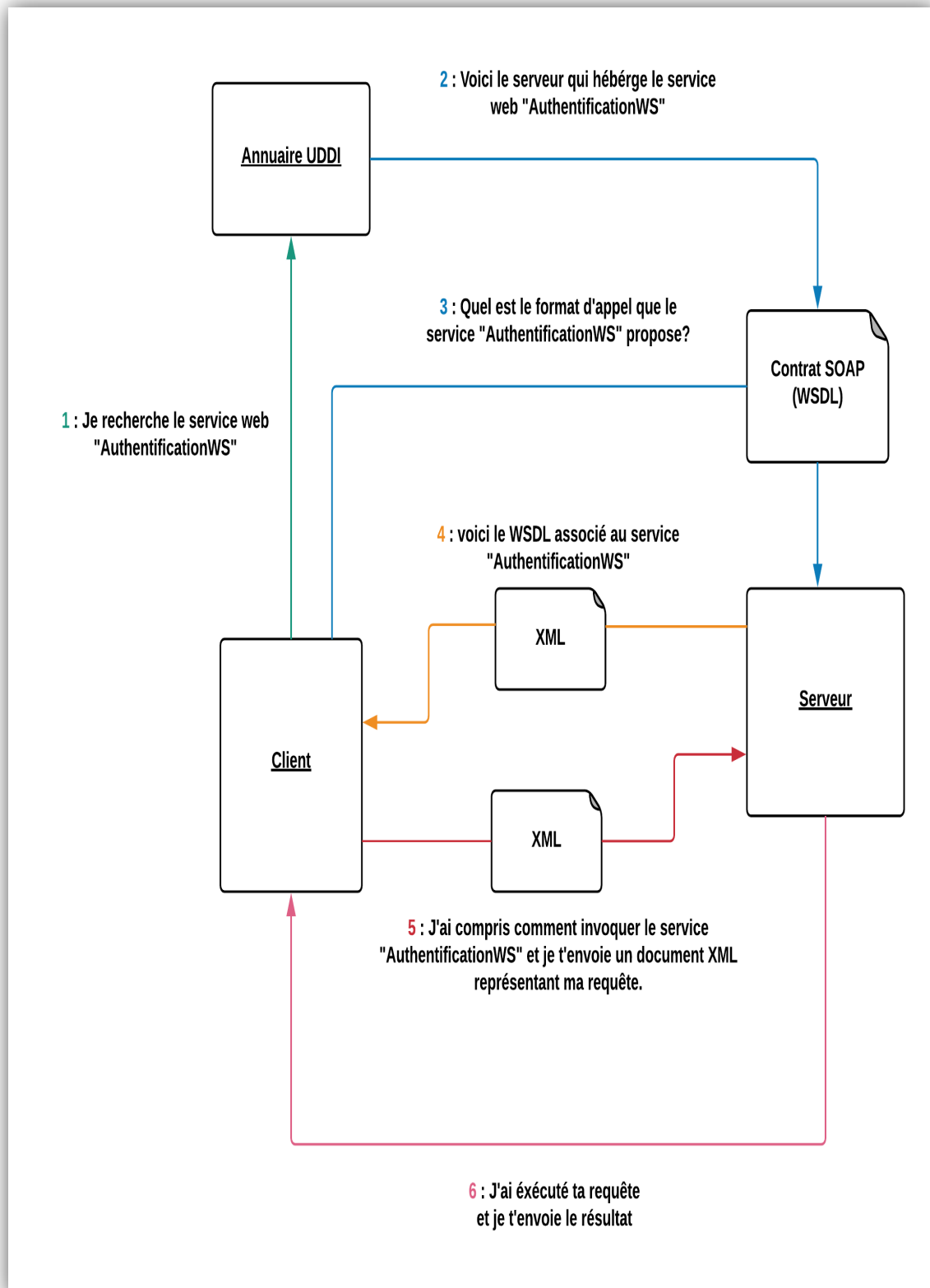


Figure III.5 : utilisation du service web de type SOAP "Authentication".

### III-6-1-3- Diagramme de séquence du cas d'utilisation « modifier Compte » :

Le diagramme ci-après décrit la séquence correspondante à une modification d'un mot de passe au niveau d'un compte donné. C'est pareil dans le cas où on désire modifier le nom d'utilisateur (username) ou bien le type (c'est-à-dire le rôle joué par l'acteur concerné).

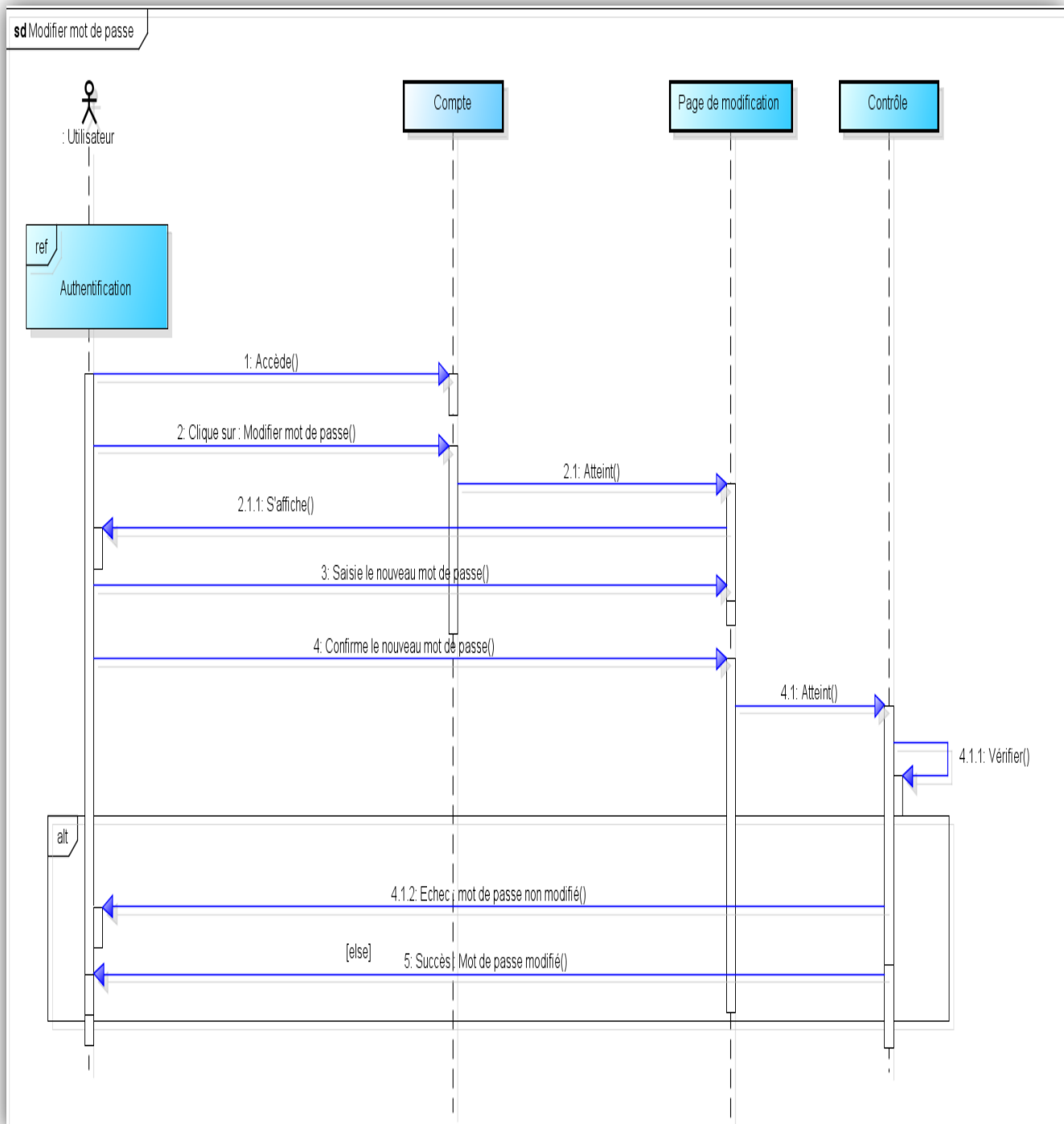


Figure III.6 : Diagramme de séquence du cas d'utilisation « Modifier un compte ».

### III-6-1-4- Modèle d'utilisation du service web SOAP correspondant :

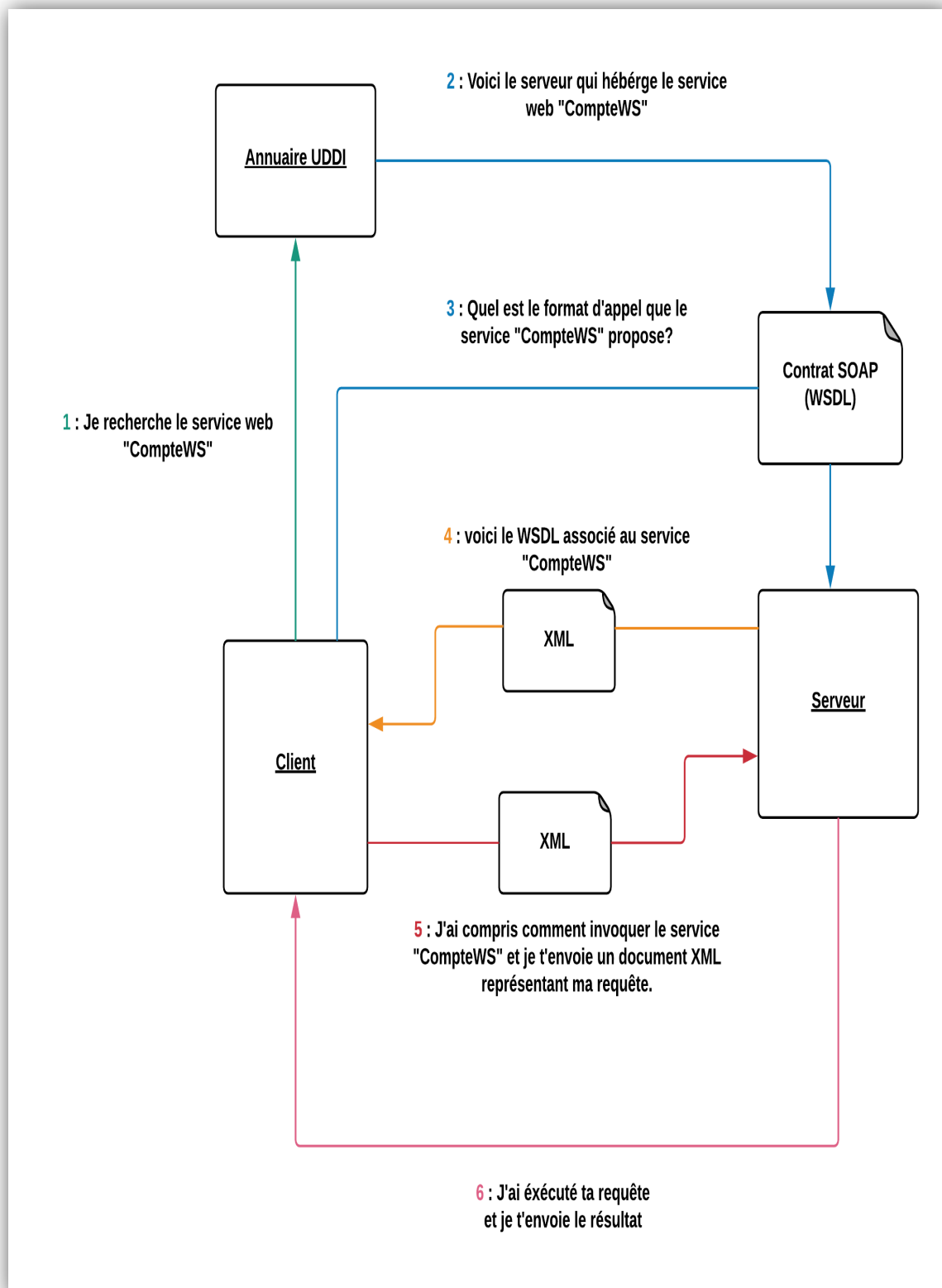


Figure III.7 : utilisation du service web de type SOAP "CompteWS".



### III-6-1-5- Diagramme de séquence du cas d'utilisation « Ajouter Cours » :

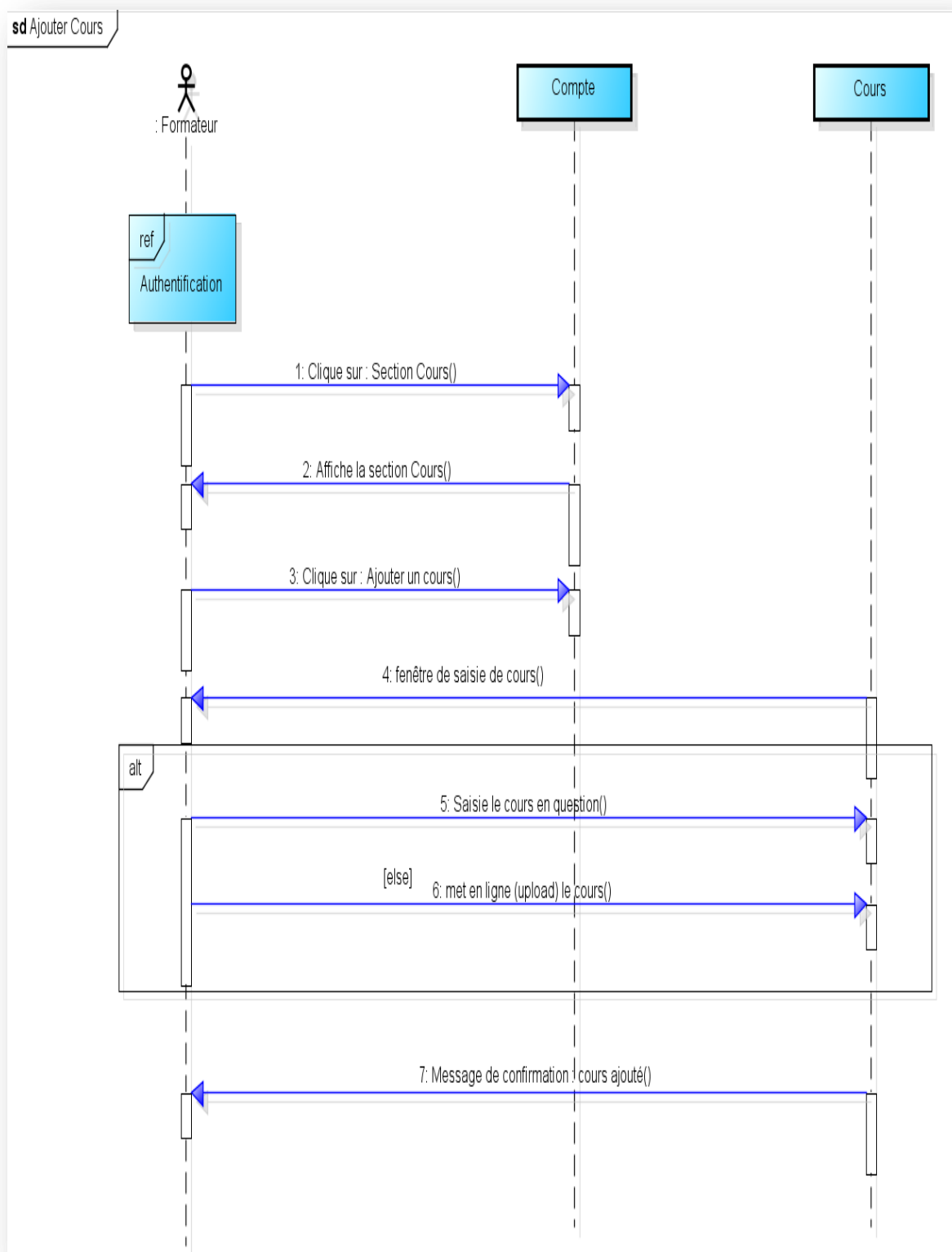


Figure III.8 : diagramme de séquence du cas d'utilisation « Ajouter un cours ».

### III-6-1-6- Modèle d'utilisation du service web SOAP correspondant :

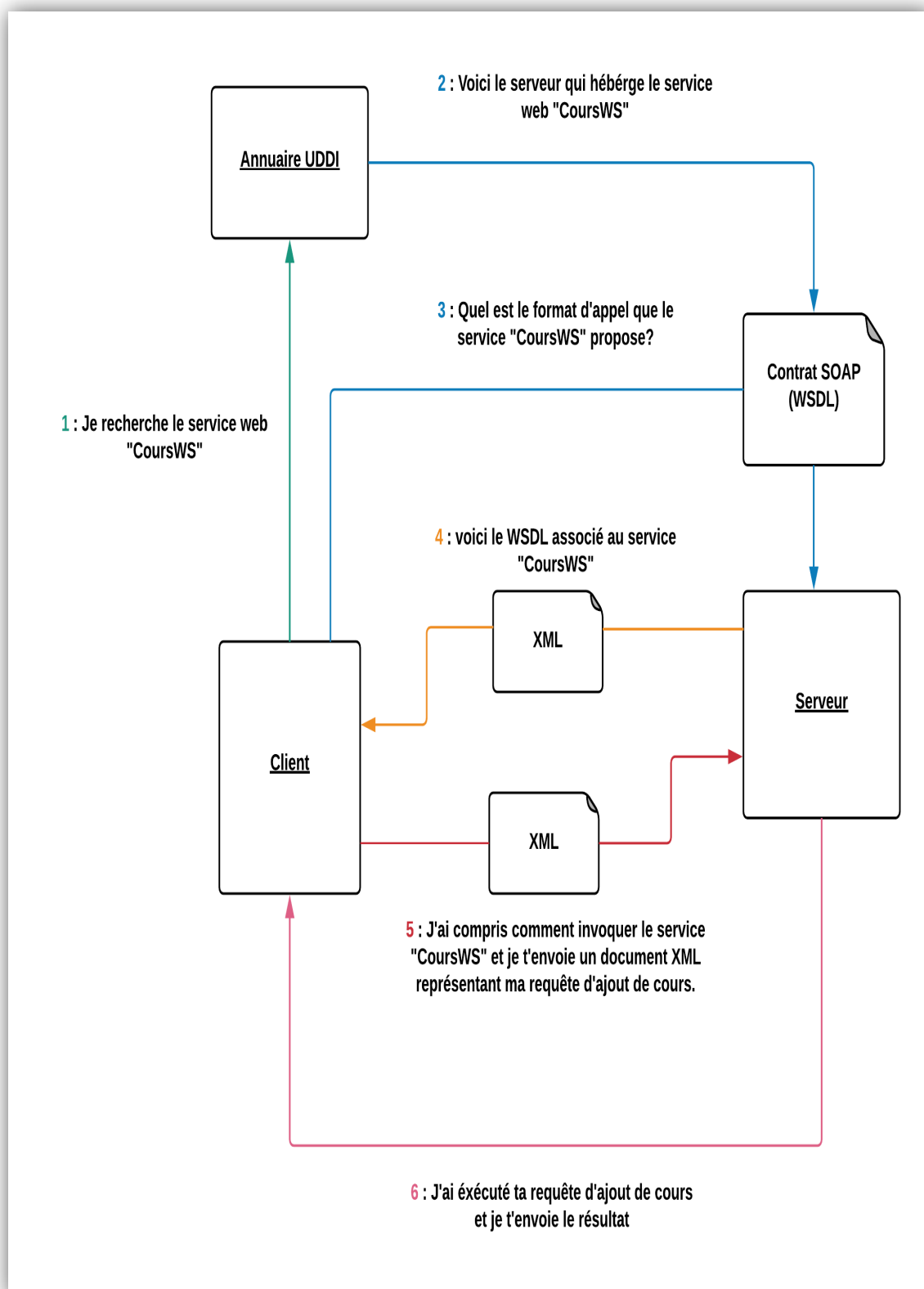


Figure III.9 : utilisation du service web de type SOAP « CoursWS ».

### III-6-1-7- Diagramme de séquence du cas d'utilisation « Se déconnecter » :

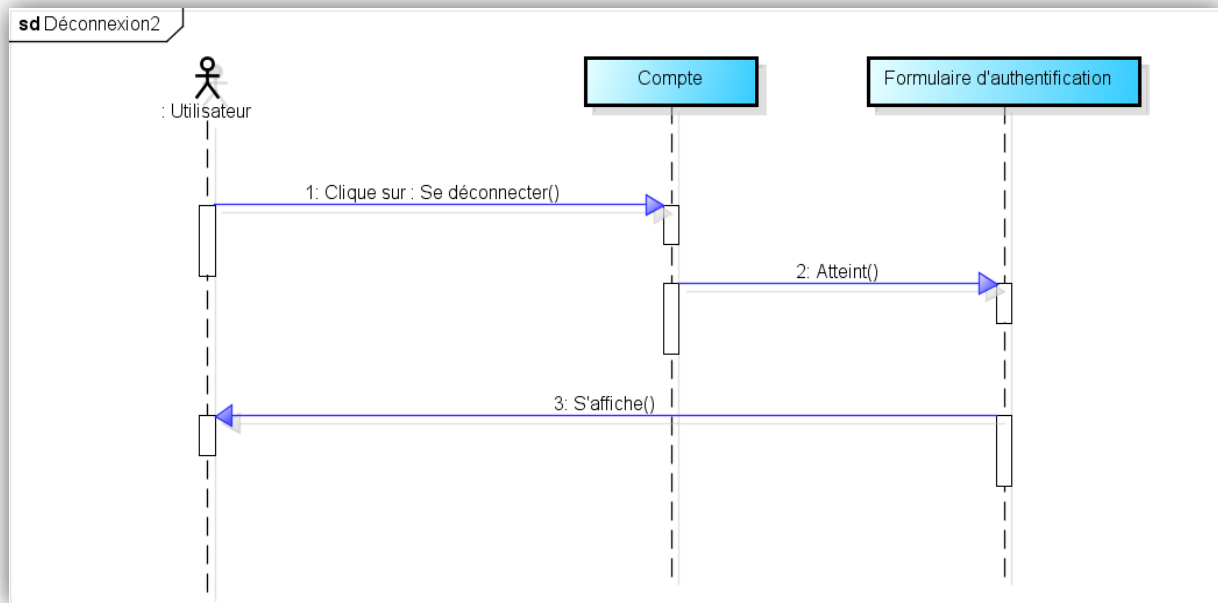


Figure III.10 : Diagramme de séquence du cas d'utilisation « Se déconnecter ».

### III-6-1-8- Modèle d'utilisation du service web correspondant :

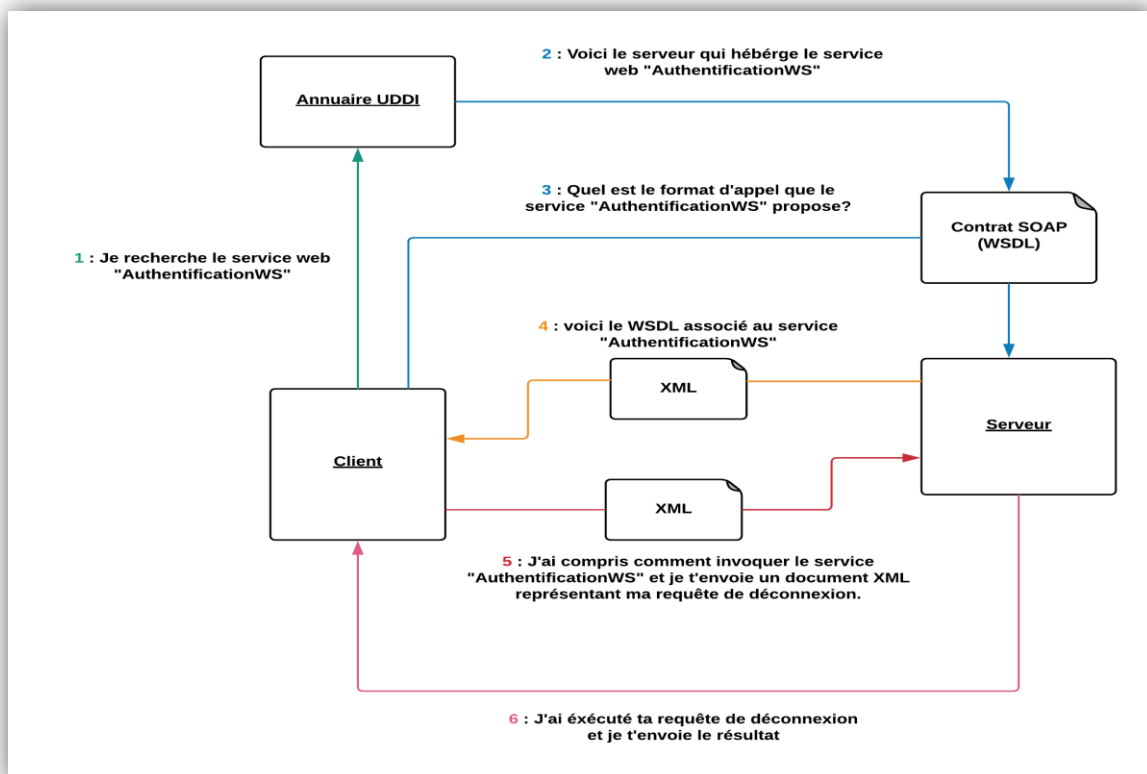


Figure III.11 : utilisation du service web de type SOAP " AuthentificationWS ".

### III-6-2- Diagramme de classe général

Ci-dessous le diagramme de classes général associé au système e-learning en prenant en compte certaines fonctionnalités de ce dernier ; sachant que notre travail va se baser en particulier sur les classes mises en évidence.

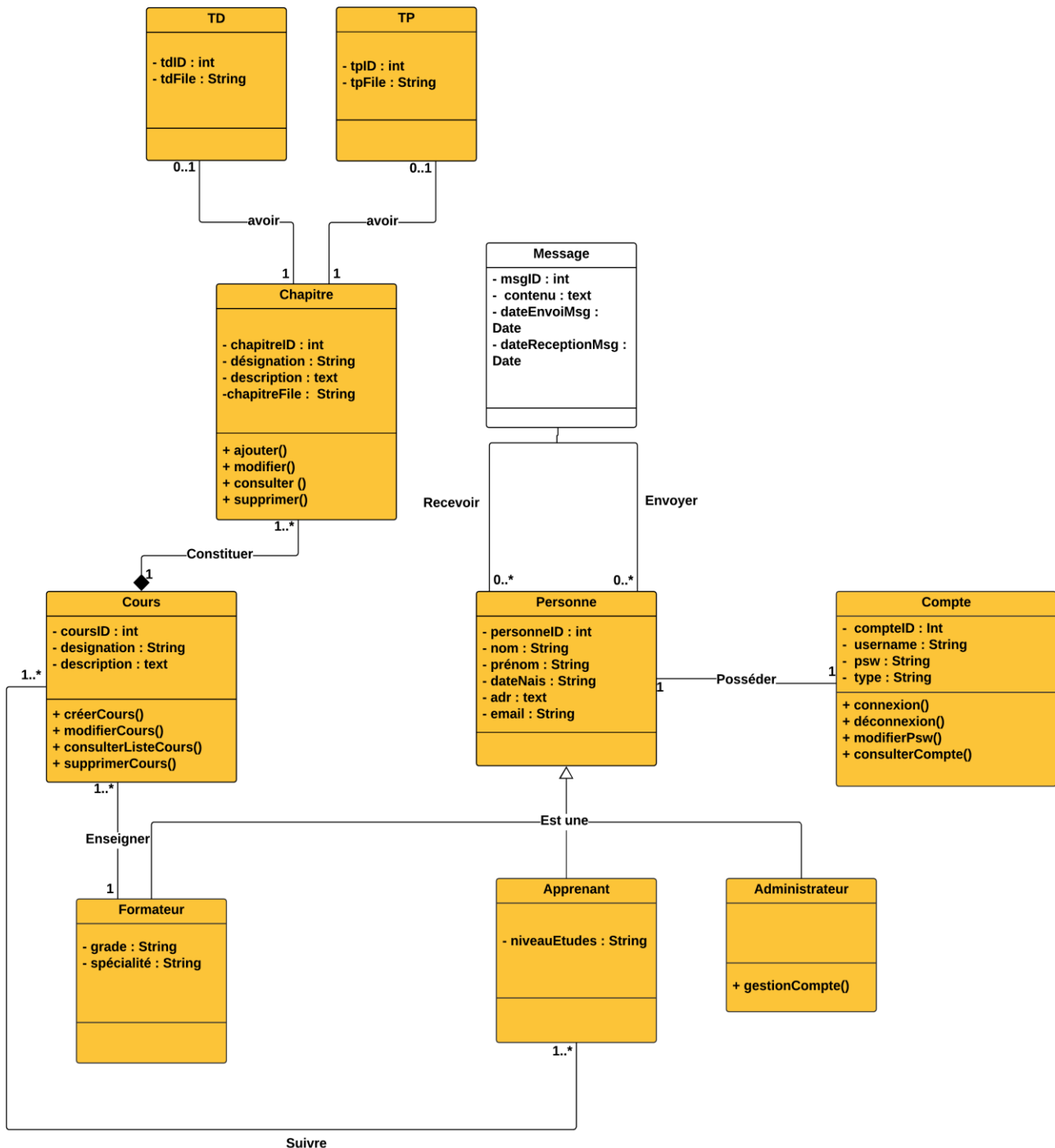


Figure III.12 : Diagramme de classe correspondant.

### III-7- Conclusion

Dans ce troisième chapitre, nous avons mis en évidence l'objectif de notre travail. Pour ce faire, nous avons proposé une mini plateforme avec quelques fonctionnalités e-learning qui intégreront nos **Services Web de type SOAP**. Ceci dit, celle-ci est fondée sur une analyse et une conception modélisées à l'aide des diagrammes associés au Langage de Modélisation Unifié « UML ». A ce titre, nous avons :

- ✓ Spécifié les besoins fonctionnels de notre système en terme d'acteurs ;
- ✓ Conçu les diagrammes des cas d'utilisation ainsi que les diagrammes de séquence.
- ✓ Elaboré un diagramme de classe général en mettant en évidence les classes associées à nos services web de type SOAP.

Au final, le chapitre qui suit sera consacré à la partie **réalisation** ; en y présentera les différents outils de développement utilisés afin de mettre en œuvre le travail demandé.

## IV-1- Introduction

Après avoir achevé l'étape d'analyse et conception, on va entamer dans ce qui suit la partie réalisation et implémentation qui consiste en la traduction de la solution citée précédemment en code source.

En effet, ce présent chapitre couvre la création et la mise en œuvre des **Services Web** de type **SOAP** ainsi que la présentation des outils qui ont permis à ces derniers de voir le jour. Nous présenterons, par la suite, la **vue** résultante correspondante au modèle **MVC (Model Controller View)** et qui constitue d'ailleurs la consommation de nos **Services Web** fraîchement déployés.

## IV-2- Environnement et outils de développement et d'implémentation

Pour la réalisation de ce projet, nous avons opté pour les outils suivants :

- ✓ **Java Enterprise Edition (JEE)** comme spécification / plateforme de développement ;
- ✓ **Enterprise JavaBeans (EJB)** comme architecture de composants logiciels côté serveur et ce, du côté **Back End** ;
- ✓ **Netbeans8.1RC** comme IDE ;
- ✓ **GlassFish** comme serveur d'application ;
- ✓ **Java** comme langage de programmation ;
- ✓ **Java Server Faces (JSF)** comme Framework et **PrimeFaces** comme API et ce, du côté **Front End**;
- ✓ **JPQL** comme langage de requêtes ;
- ✓ **MYSQL serveur** comme serveur de bases de données ;
- ✓ **SOAP UI** comme testeur des requêtes/réponses SOAP.

### IV-2-1- Java Enterprise Edition (JEE) [32]

Java Platform, Enterprise Edition, Java EE ou Jakarta EE (anciennement Java 2 Platform, Enterprise Edition, ou J2EE) est une spécification pour la plate-forme Java d'Oracle, destinée aux applications d'entreprise.

La plate-forme étend Java Platform Standard Edition (Java SE) en fournissant une API de mapping objet-relationnel, des architectures distribuées et des services web. Elle se fonde principalement sur des composants modulaires exécutés sur un serveur d'applications.

Pour ce faire, Java EE définit les éléments suivants :

- une plate-forme (Java EE Platform) pour héberger et exécuter les applications, incluant outre Java SE des bibliothèques logicielles additionnelles du Java Development Kit (JDK) ;
- une suite de tests (Java EE Compatibility Test Suite) pour vérifier la compatibilité ;
- une réalisation de référence (Java EE Reference Implementation), dénommée GlassFish ;
- un catalogue de bonnes pratiques (Java EE BluePrints) ;
- un code script.

#### **IV-2-2- Enterprise JavaBeans (EJB) [32]**

Enterprise JavaBeans (EJB) est une architecture de composants logiciels côté serveur pour la plateforme de développement Java EE.

Cette architecture propose un cadre pour créer des composants distribués (c'est-à-dire déployés sur des serveurs distants) écrit en langage de programmation Java, hébergés au sein d'un serveur applicatif permettant de représenter des données (EJB dit *entité*), de proposer des services avec ou sans conservation d'état entre les appels (EJB dit *session*), ou encore d'accomplir des tâches de manière asynchrone (EJB dit *message*). Tous les EJB peuvent évoluer dans un contexte transactionnel.

De la version 1.0 à la version 2.1, un EJB était accompagné d'un ou plusieurs fichiers de déploiement écrits en XML qui permettait au serveur applicatif de déployer correctement l'objet au sein d'un conteneur. C'était notamment dans ces fichiers de déploiement que le développeur avait la possibilité de préciser le cadre transactionnel dans lequel l'objet allait s'exécuter. Depuis la version 3.0, le modèle EJB utilise le principe d'annotation java (meta-données) pour spécifier toute la configuration et les propriétés transactionnelles de l'objet. Le fichier de code source de l'EJB se suffit à lui-même.

C'est le serveur applicatif qui est chargé de la création, la destruction et l'activation de ses composants en fonction des besoins. Le client via un appel RMI (ou une de ses dérivées) va rechercher un EJB par son nom logique <sup>1</sup>*JNDI* et appeler une ou plusieurs méthodes de cet objet.

#### **IV-2-3- GlassFish [34]**

GlassFish est un projet de serveur d'applications Java créé par Sun Microsystems qui permet à de nombreux développeurs de générer des technologies d'entreprise pratiques et évolutives, ainsi que des services supplémentaires pouvant être installés en fonction de leurs préférences. Il s'agit d'un logiciel libre sous licence GNU General Public License (GPL) et Common Development and Distribution License (CDDL). GlassFish a été acquis par Oracle en 2010.

#### **IV-2-4- Java Server Faces (JSF) [35]**

JSF, alias "Java Server Faces" est un Framework de type MVC, destiné aux applications web respectant l'architecture J2EE.

Le premier objectif de JSF est de procurer un environnement de développement permettant de construire une interface de type web, sans devoir toucher au code HTML et JavaScript. Ceci est réalisé par la mise en place d'un mapping entre l'HTML et les objets concernés. JSF est donc basé sur la notion de composants. Le principal atout de JSF est qu'il tient compte des différentes expériences acquises non seulement avec des technologies de type standard comme les Servlets, les JSP,... mais aussi avec les technologies de type Framework comme Struts. Dès lors, JSF propose un Framework qui puisse être mis en œuvre par des outils facilitant le développement d'applications web.

#### **IV-2-5- PrimeFaces**

Primefaces est une librairie open source de composants graphiques pour les applications JSF (Java Server Faces) créée par la compagnie Turque PrimeTek. Le développement de Primefaces remonte en 2008 et la première version sera publiée Janvier 2009. Il s'agit d'un ensemble de plus de 100 composants graphiques. A part leur beau look and feel, les composants intègrent pleinement les fonctionnalités AJAX et HTML5, sont responsives et compatibles avec la plupart des navigateurs modernes.



## IV-2-6- JPQL [32]

Avant de définir le langage JPQL, il faudrait au préalable parler de Java Persistence API (JPA).

### IV-2-6-1- JPA :

Java Persistence API (abrégée en JPA), est une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java. La persistance dans ce contexte recouvre trois zones :

- l'API elle-même, définie dans le paquetage javax.persistence ;
- le langage Java Persistence Query (JPQL) ;
- l'objet/les métadonnées relationnelles.

### IV-2-6-2- Présentation de JPQL [32] :

Java Persistence Query Language (JPQL) est un langage de requête orienté objet indépendant de la plateforme, défini dans la spécification Java Persistence API (JPA). JPQL sert à exécuter des requêtes sur des entités persistées en base de données mais en travaillant sur les entités Java correspondantes aux tables plutôt que sur les tables elles-mêmes.

## IV-3- Modèle physique de données

### IV-3-1- Table Formateur :

<i>Champ</i>	<i>Type</i>	<i>Contraintes</i>	<i>Désignation</i>
<b>formID</b>	INT	Primary Key, auto_increment	Identifiant du formateur
<b>nomF</b>	VARCHAR(20)		Nom du formateur
<b>prenomF</b>	VARCHAR(15)		Prénom du formateur
<b>grade</b>	VARCHAR(20)		Grade du formateur
<b>specialite</b>	VARCHAR(20)		Spécialité du formateur
<b>numcF</b>	INT	Foreign Key	L'identifiant du compte formateur

## IV-3-2- Table Apprenant :

<i>Champ</i>	<i>Type</i>	<i>Contraintes</i>	<i>Désignation</i>
<b>applID</b>	INT	Primary Key, auto_increment	Identifiant de l'apprenant
<b>nomA</b>	VARCHAR(20)		Nom de l'apprenant
<b>prenomA</b>	VARCHAR(15)		Prénom de l'apprenant
<b>adr</b>	VARCHAR(60)		Adresse de l'apprenant
<b>niveauEtudes</b>	VARCHAR(20)		Niveau d'Etudes de l'apprenant
<b>compteID</b>	INT	Foreign Key	L'identifiant du compte Apprenant

## IV-3-3- Table Compte :

<i>Champ</i>	<i>Type</i>	<i>Contraintes</i>	<i>Désignation</i>
<b>ID</b>	INT	Primary Key, auto_increment	Identifiant du compte
<b>usernameU</b>	VARCHAR(20)		Nom utilisateur
<b>password</b>	VARCHAR (20)		Mot de passe
<b>typeU</b>	VARCHAR (15)		Type utilisateur

**IV-3-4- Table Cours :**

<i>Champ</i>	<i>Type</i>	<i>Contraintes</i>	<i>Désignation</i>
<b>coursID</b>	INT	Primary Key, auto_increment	Identifiant du cours
<b>designation</b>	VARCHAR(20)		
<b>description</b>	TEXT		
<b>nF</b>	INT	Foreign key	Numéro formateur (ID du formateur)

**IV-3-4- Table Cours-Apprenant :**

<i>Champ</i>	<i>Type</i>	<i>Contraintes</i>	<i>Désignation</i>
<b>coursID</b>	INT	Primary Key	Identifiant du cours
<b>appID</b>	INT	Primary Key	Identifiant de l'apprenant

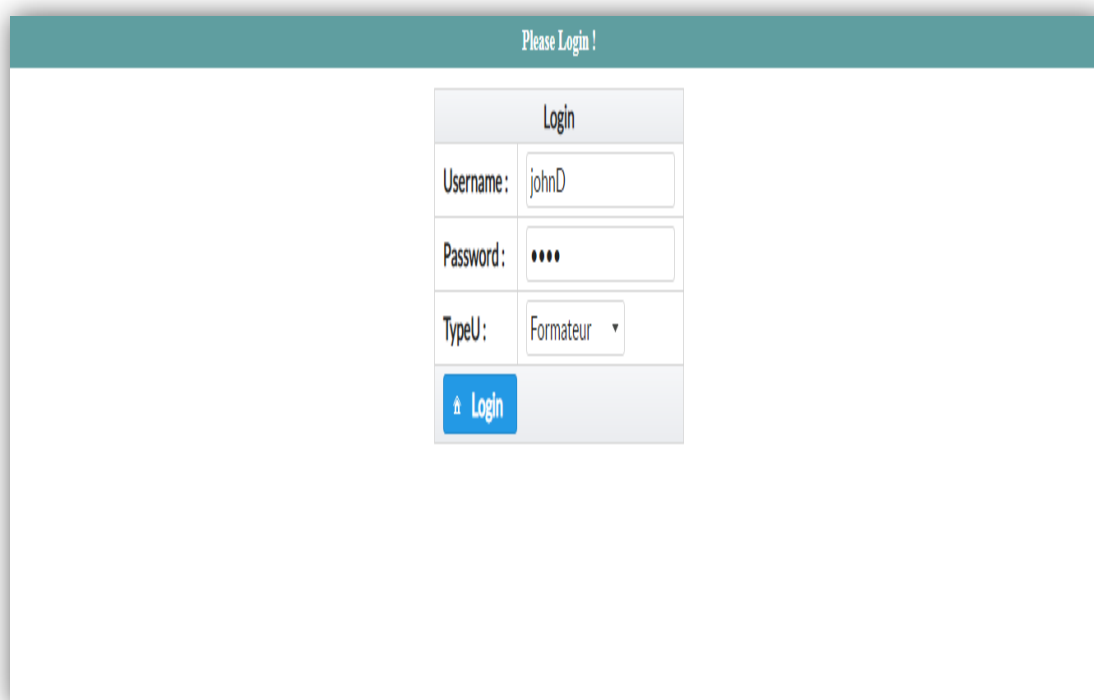
**IV-3-4- Table Chapitre :**

<i>Champ</i>	<i>Type</i>	<i>Contraintes</i>	<i>Désignation</i>
<b>chapitreID</b>	INT	Primary Key, auto_increment	Identifiant du cours
<b>designationC</b>	VARCHAR(20)		
<b>descriptionC</b>	TEXT		
<b>numC</b>	INT	Foreign key	Numéro cours
<b>ChapterFile</b>	mediumblob		
<b>TDFile</b>	mediumblob		
<b>TPFile</b>	mediumblob		

## IV-4- Présentation des Interfaces résultantes suivi des Requêtes / Réponses SOAP correspondantes

Notre solution WS SOAP est composée des interfaces ci-après :

**IV-4- 1- Une fenêtre pour l'Authentication** : l'acteur concerné (apprenant/formateur) saisie son nom d'utilisateur, mot de passe ainsi que son type (rôle joué par celui-ci) ; en cas d'erreur un message lui sera affiché.



The image shows a web-based login interface. At the top, there is a teal header bar with the text "Please Login !". Below this, the main content area is white and contains a login form. The form has a title "Login" in a light blue box. It includes three input fields: "Username:" with the value "johnD", "Password:" with four black dots, and "TypeU:" with a dropdown menu showing "Formateur". At the bottom of the form is a blue button with a white house icon and the text "Login".

Figure IV.6 : Authentification apprenant/formateur.

#### IV-4-2- Test Requête et Réponse SOAP :

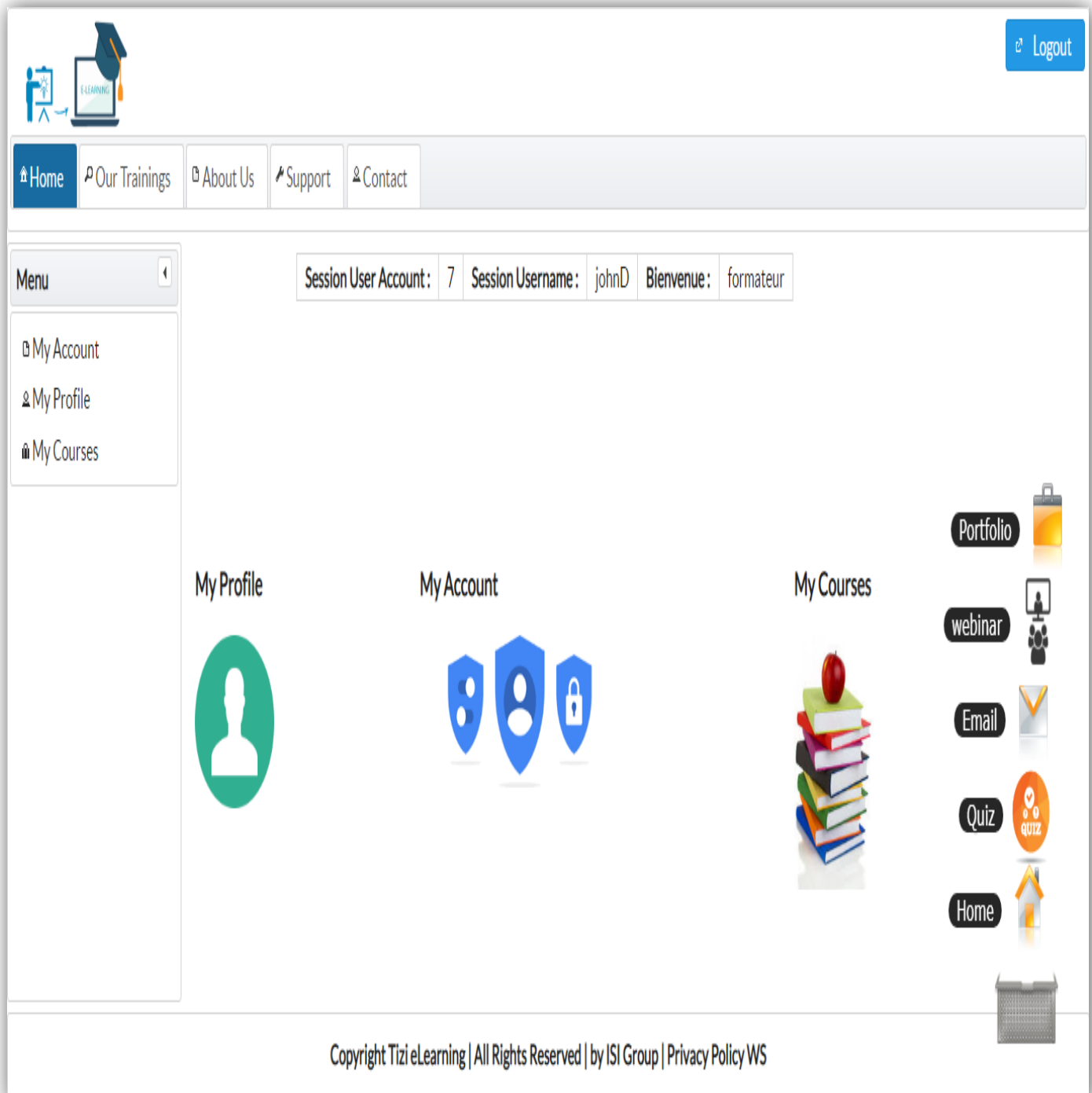
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:log="http://loginEJB/">
  <soapenv:Header/>
  <soapenv:Body>
    <log:login>
      <!--Optional:-->
      <arg0>johnD</arg0>
      <!--Optional:-->
      <arg1>judo</arg1>
      <!--Optional:-->
      <arg2>formateur</arg2>
    </log:login>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure IV.7 : Requête SOAP associée à l'authentification (Login).

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:loginResponse xmlns:ns2="http://loginEJB/">
      <return>
        <id>7</id>
        <password>judo</password>
        <typeU>formateur</typeU>
        <usernameU>johnD</usernameU>
      </return>
    </ns2:loginResponse>
  </S:Body>
</S:Envelope>
```

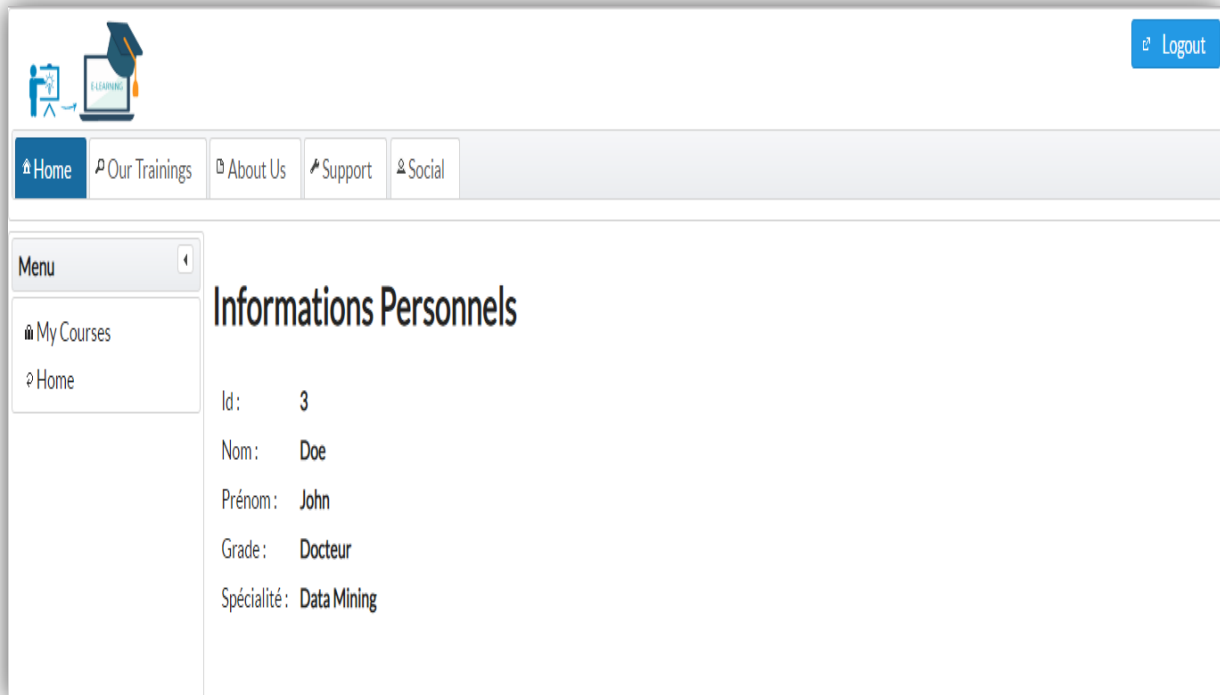
Figure IV.3 : Réponse SOAP associée à l'authentification (Login).

**IV-4-3- Fenêtre d'accueil :** elle représente la page Welcome dont laquelle l'acteur concerné peut accéder à des informations techniques ou pédagogiques le concernant et ce, à travers son profil, les informations sur son compte et enfin ses cours.



**Figure IV.4 : Page d'accueil.**

**IV-4- 4- Page de Profil :** celle-ci décrit les informations personnelles associées à l'acteur concerné.



**Figure IV.5 : Page de Profil.**

#### **IV-4- 5- Test Requête et Réponse SOAP**

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:log="http://loginEJB/">
  <soapenv:Header/>
  <soapenv:Body>
    <log:getprofilF>
      <arg0>7</arg0>
    </log:getprofilF>
  </soapenv:Body>
</soapenv:Envelope>
```

**Figure IV.6 : Requête SOAP associée à Profil.**

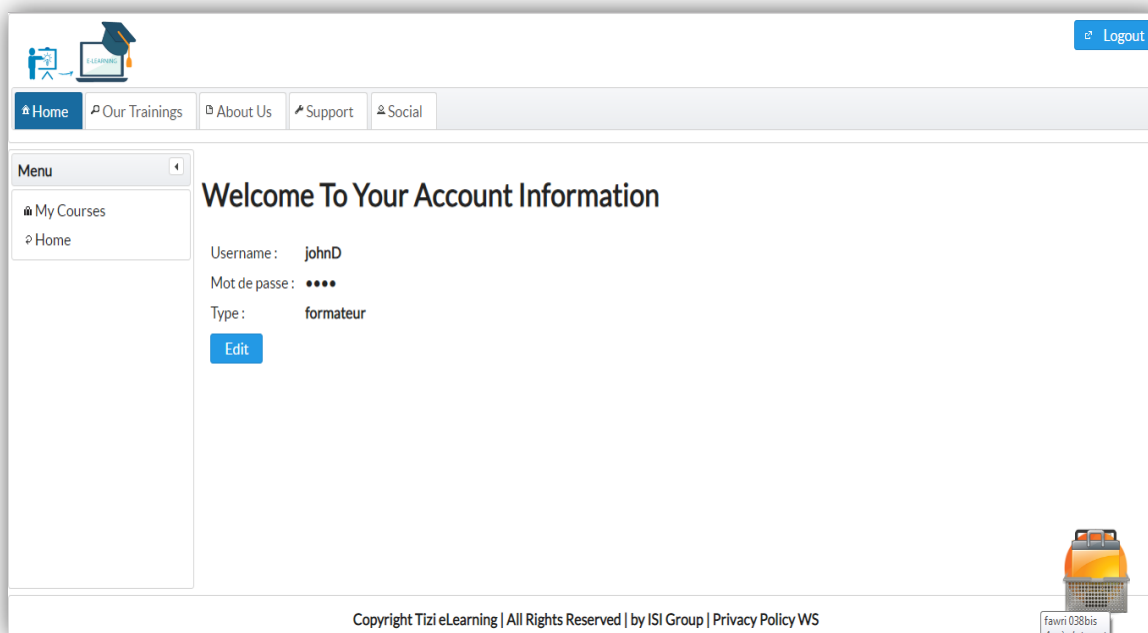
```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getprofilFResponse xmlns:ns2="http://loginEJB/">
      <return>
        <formID>3</formID>
        <grade>Docteur</grade>
        <nomF>Doe</nomF>
        <numcF>
          <id>7</id>
          <password>judo</password>
          <typeU>formateur</typeU>
          <usernameU>johnD</usernameU>
        </numcF>
        <prenomF>John</prenomF>
        <specialite>Data Mining</specialite>
      </return>
    </ns2:getprofilFResponse>
  </S:Body>
</S:Envelope>

```

**Figure IV.7 : Réponse SOAP associée à Profil.**

**IV-4-6- Page du Compte utilisateur :** celle-ci décrit les informations associées au compte de l'acteur concerné. Ce dernier a la possibilité d'éditer, or modifier le mot de passe (Password) associé à son compte actuel.



**Figure IV.8 : Page du Compte utilisateur.**



#### V-4-7- Test Requête et Réponse SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:log="http://loginEJB/">
  <soapenv:Header/>
  <soapenv:Body>
    <log:getAccount>
      <arg0>7</arg0>
    </log:getAccount>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure IV.9 : Requête SOAP associée à l'obtention du compte en question.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getAccountResponse xmlns:ns2="http://loginEJB/">
      <return>
        <id>7</id>
        <password>judo</password>
        <typeU>formateur</typeU>
        <usernameU>johnD</usernameU>
      </return>
    </ns2:getAccountResponse>
  </S:Body>
</S:Envelope>
```

Figure IV.10 : Réponse SOAP associée à l'obtention du compte correspondant.

#### IV-4-8- Test Requête et Réponse SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:log="http://loginEJB/">
  <soapenv:Header/>
  <soapenv:Body>
    <log:UpdateAccount>
      <arg0>7</arg0>
      <!--Optional:-->
      <arg1>judoBis</arg1>
    </log:UpdateAccount>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure IV.11 : Requête SOAP associée à la modification du mot de passe du compte utilisateur pris en considération.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:UpdateAccountResponse xmlns:ns2="http://loginEJB/">
  </S:Body>
</S:Envelope>
```

Figure IV.12 : Réponse SOAP associée à la modification du mot de passe du compte utilisateur pris en considération.

**IV-4-9- Page des Cours :** celle-ci décrit des informations pédagogiques, à savoir : les cours associés au formateur concerné. De ce fait, ce dernier a la possibilité de consulter, d'ajouter, d'éditer et de supprimer un cours donné ; autrement dit, la gestion des cours dépend du formateur.

**List of Courses**

CourseID	Designation	Description	Action
14	Data Mining I	Introduction au Data Mining : Le Data Mining est en fait un terme générique englobant toute une famille d'outils facilitant l'exploration et l'analyse des données contenues au sein d'une base de données. Les techniques mises en action lors de l'utilisation de cet instrument d'analyse et de prospection sont particulièrement efficaces pour extraire des informations significatives et pertinentes depuis de grandes quantités de données.	<a href="#">View</a>   <a href="#">Update</a>   <a href="#">Delete</a>
15	Data Mining II	Apprentissage Supervisé : Les arbres de décision. Un arbre de décision est une représentation graphique de la procédure de classification (modèle de prédiction) où : a- Un nœud spécifie un test que doit subir un certain attribut ; b- Chaque branche (arc) correspond à une valeur possible de l'attribut ; c- Une feuille indique une classe.	<a href="#">View</a>   <a href="#">Update</a>   <a href="#">Delete</a>

Copyright Tizi eLearning | All Rights Reserved | by ISI Group | Privacy Policy WS

**Figure IV.13 : Page des Cours du formateur.**

**Create / Add a Course**

Designation :

Description :

[ADD](#)

Copyright Tizi eLearning | All Rights Reserved | by ISI Group | Privacy Policy WS

**Figure IV.14 : Ajouter un cours donné.**

#### IV-4-10- Test Requête et Réponse SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:log="http://loginEJB/">
  <soapenv:Header/>
  <soapenv:Body>
    <log:getCoursesF>
      <arg0>3</arg0>
    </log:getCoursesF>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure IV.15 : Requête SOAP associée à l'obtention des cours du formateur concerné.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getCoursesFResponse xmlns:ns2="http://loginEJB/">
      <return>
        <coursID>14</coursID>
        <description>Généralités et Principes du Data Mining.</description>
        <designation>Data Mining I</designation>

      </return>
      <return>
        <coursID>15</coursID>
        <description>Apprentissage Supervisé : Les arbres de décision.</description>
        <designation>Data Mining II</designation>

      </return>
    </ns2:getCoursesFResponse>
  </S:Body>
</S:Envelope>
```

Figure IV.16 : Réponse SOAP associée à l'obtention des cours du formateur concerné.

#### IV-4-11- Test Requête et Réponse SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:log="http://loginEJB/">
  <soapenv:Header/>
  <soapenv:Body>
    <log:ajouterCours>
      <!--Optional:-->
      <arg0>
        <!--Optional:-->
        <coursID?</coursID>
        <!--Optional:-->
        <description>Causes des éruptions volcaniques</description>
        <!--Optional:-->
        <designation>Volcans</designation>
        <!--Optional:-->
        <nf>
          <!--Optional:-->
          <formID>1</formID>
          <!--Optional:-->
          <grade>docteur</grade>
          <!--Optional:-->
```

Figure IV.17 : Requête SOAP associée à l'ajout d'un cours donné.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:ajouterCoursResponse xmlns:ns2="http://loginEJB/">
  </S:Body>
</S:Envelope>
```

Figure IV.18 : Réponse SOAP associée à l'ajout d'un cours donné.

#### IV-4-12- Test Requête et Réponse SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:log="http://loginEJB/">
  <soapenv:Header/>
  <soapenv:Body>
    <log:UpdateCourse>
      <arg0>1</arg0>
      <!--Optional:-->
      <designation>Géologie : Bases II </designation>
      <!--Optional:-->
      <arg2>Les bases de la géologie</arg2>
    </log:UpdateCourse>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure IV.19: Requête SOAP associée à la modification d'un cours donné.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:UpdateCourseResponse xmlns:ns2="http://loginEJB/">
  </S:Body>
</S:Envelope>
```

Figure IV.20 : Réponse SOAP associée à la modification d'un cours donné.

#### IV-4-13- Test Requête et Réponse SOAP

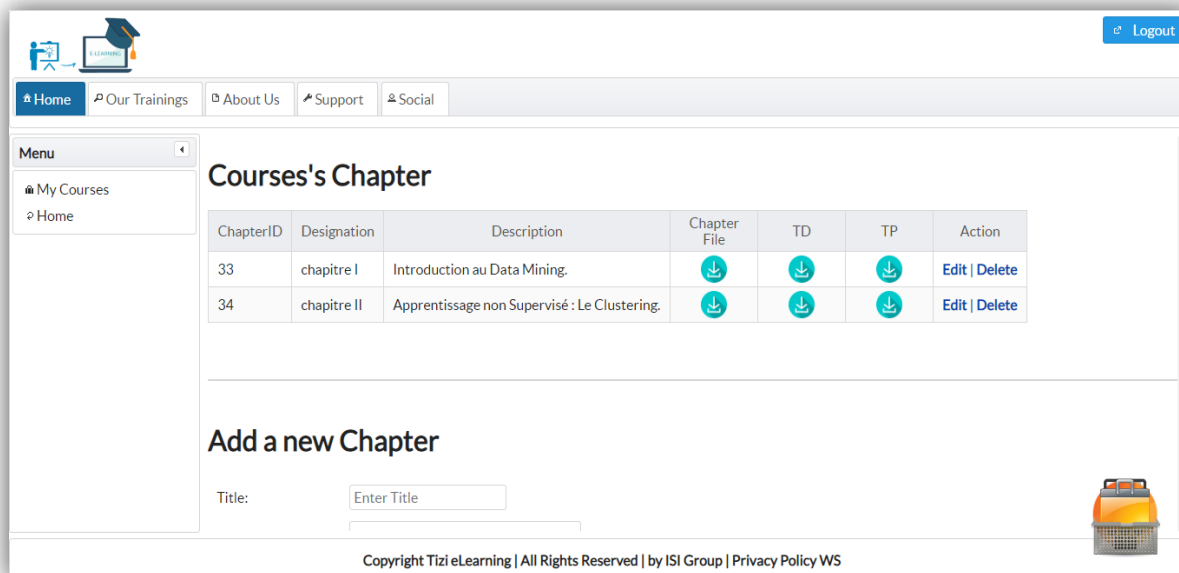
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:log="http://loginEJB/">
  <soapenv:Header/>
  <soapenv:Body>
    <log:removeCourse>
      <!--Optional:-->
      <arg0>
        <!--Optional:-->
        <courseID>14</courseID>
        <!--Optional:-->
        <description>Causes des éruptions volcaniques</description>
        <!--Optional:-->
        <designation>Volcans</designation>
        <!--Optional:-->
        <nf>
          <!--Optional:-->
          <formID>1</formID>
          <!--Optional:-->
          <grade>docteur</grade>
          <!--Optional:-->
          <nomExat</nomExat>
        </nf>
      </arg0>
    </log:removeCourse>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure IV.21 : Requête SOAP associée à la suppression d'un cours donné.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:removeCourseResponse xmlns:ns2="http://loginEJB/">
    </S:Body>
</S:Envelope>
```

Figure IV.22 : Réponse SOAP associée à la suppression d'un cours donné.

**IV-4- 14- Page des Chapitres** : représente les chapitres associés à un cours particulier du formateur concerné. De ce fait, ce dernier a la possibilité d'éditer et de supprimer un chapitre donné.



**Courses's Chapter**

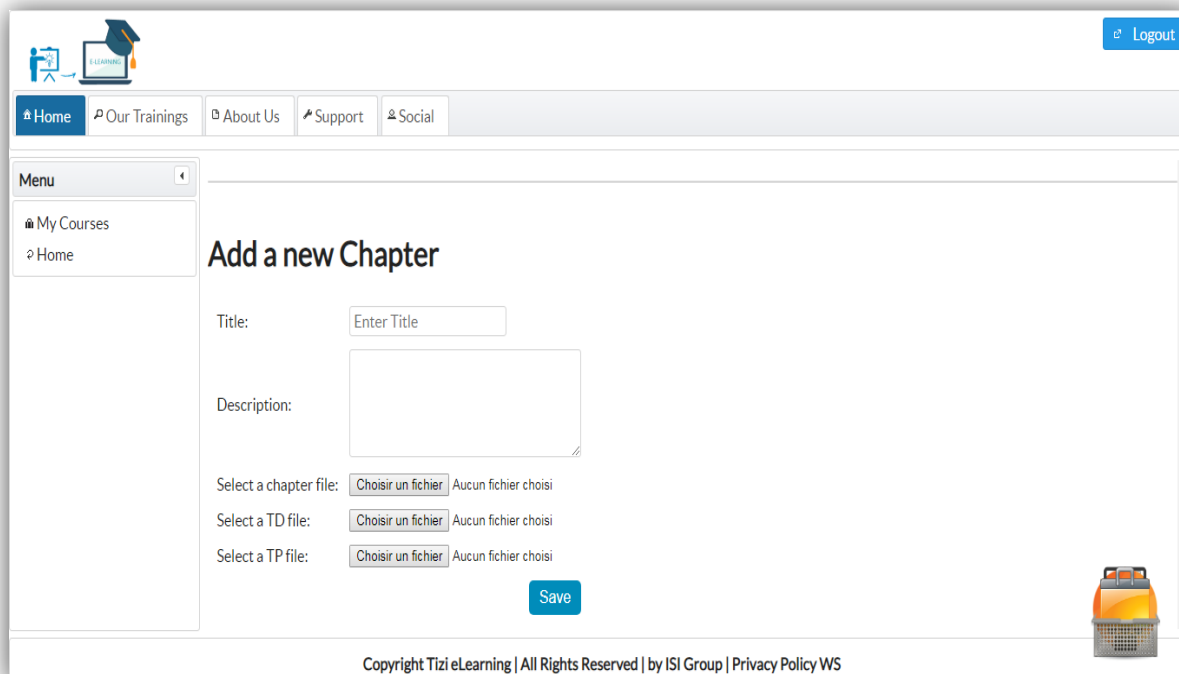
ChapterID	Designation	Description	Chapter File	TD	TP	Action
33	chapitre I	Introduction au Data Mining.				<a href="#">Edit</a>   <a href="#">Delete</a>
34	chapitre II	Apprentissage non Supervisé : Le Clustering.				<a href="#">Edit</a>   <a href="#">Delete</a>

**Add a new Chapter**

Title:

Copyright Tizi eLearning | All Rights Reserved | by ISI Group | Privacy Policy WS

**Figure IV.23 : Page des chapitres associés au cours pris en considération.**



**Add a new Chapter**

Title:

Description:

Select a chapter file:  Aucun fichier choisi

Select a TD file:  Aucun fichier choisi

Select a TP file:  Aucun fichier choisi

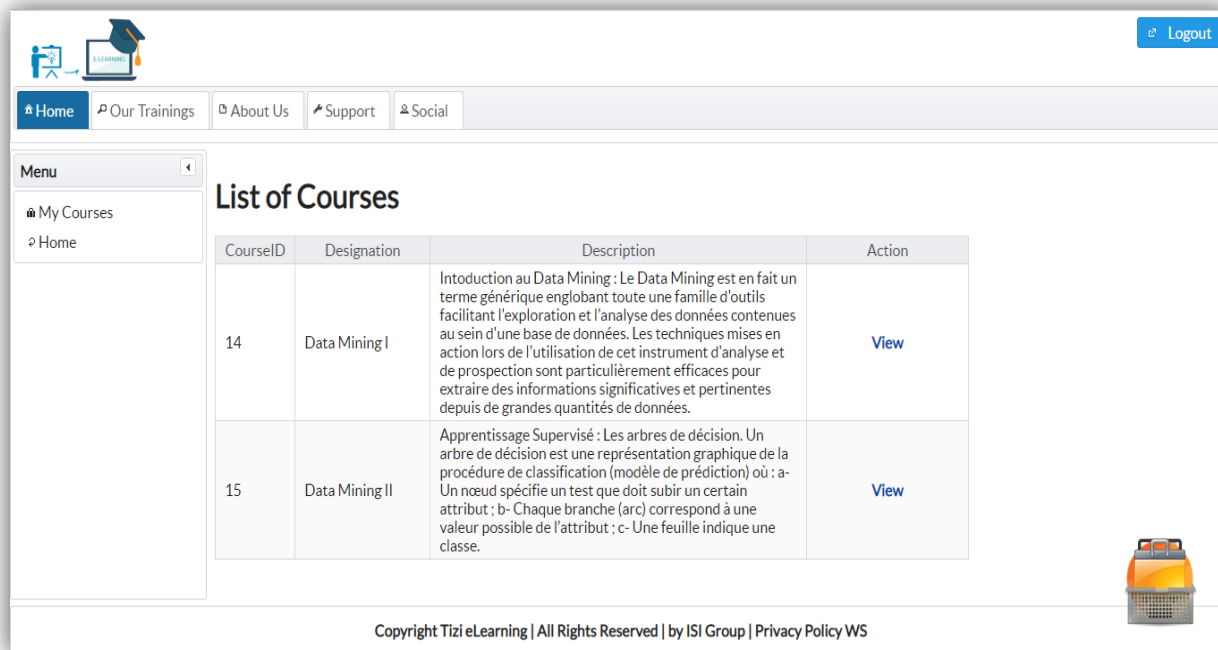
Copyright Tizi eLearning | All Rights Reserved | by ISI Group | Privacy Policy WS

**Figure IV.24 : Ajout d'un chapitre à un cours donné.**



Dans ce qui suit (**coté Apprenant**), la page des cours d'un apprenant pris en considération suivi de celle associée aux chapitres correspondants ; Sachant que l'apprenant a la possibilité de Consulter ses cours (TDs et TP éventuellement).

#### IV-4- 15- Page des cours et chapitres (coté Apprenant) :

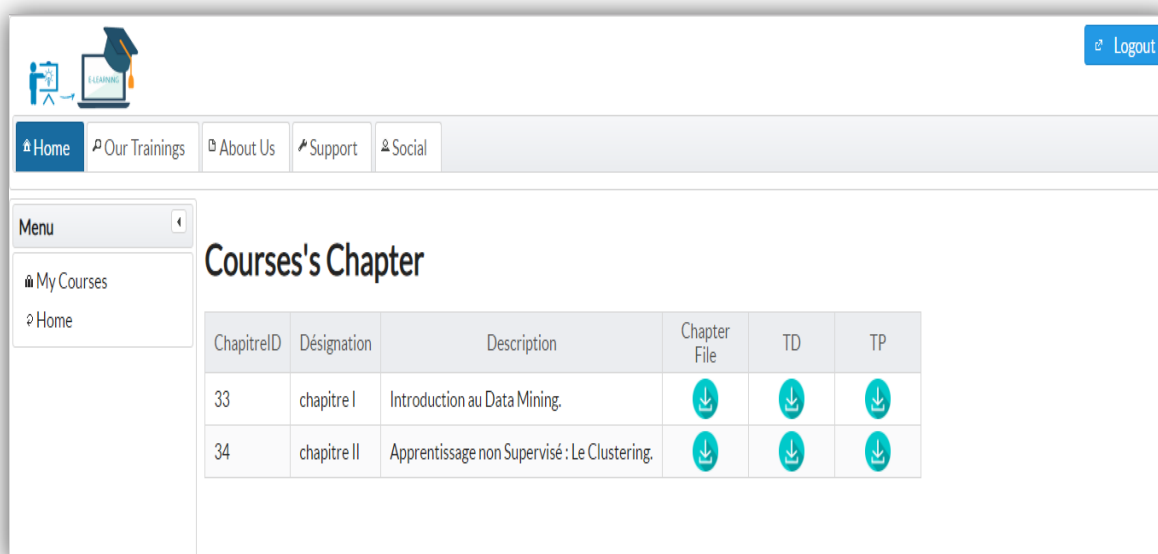


**List of Courses**

CourseID	Designation	Description	Action
14	Data Mining I	Introduction au Data Mining : Le Data Mining est en fait un terme générique englobant toute une famille d'outils facilitant l'exploration et l'analyse des données contenues au sein d'une base de données. Les techniques mises en action lors de l'utilisation de cet instrument d'analyse et de prospection sont particulièrement efficaces pour extraire des informations significatives et pertinentes depuis de grandes quantités de données.	<a href="#">View</a>
15	Data Mining II	Apprentissage Supervisé : Les arbres de décision. Un arbre de décision est une représentation graphique de la procédure de classification (modèle de prédiction) où : a- Un nœud spécifie un test que doit subir un certain attribut ; b- Chaque branche (arc) correspond à une valeur possible de l'attribut ; c- Une feuille indique une classe.	<a href="#">View</a>

Copyright Tizi eLearning | All Rights Reserved | by ISI Group | Privacy Policy WS

Figure IV.25 : Page des cours d'un apprenant.



**Courses's Chapter**

ChapterID	Désignation	Description	Chapter File	TD	TP
33	chapitre I	Introduction au Data Mining.	<a href="#">↓</a>	<a href="#">↓</a>	<a href="#">↓</a>
34	chapitre II	Apprentissage non Supervisé : Le Clustering.	<a href="#">↓</a>	<a href="#">↓</a>	<a href="#">↓</a>

Figure IV.26 : Page des chapitres associés à un cours d'un apprenant.

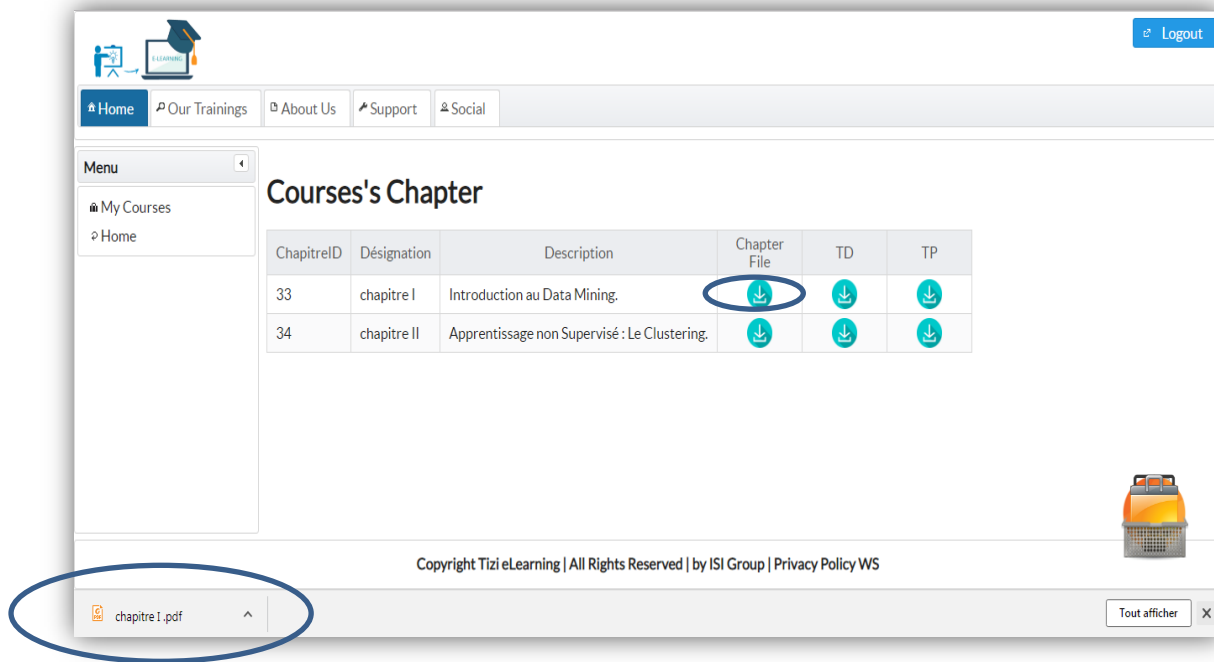
#### IV-4-16- Test Requête et Réponse SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:log="http://loginEJB/">
  <soapenv:Header/>
  <soapenv:Body>
    <log:getCourses>
      <arg0>4</arg0>
    </log:getCourses>
  </soapenv:Body>
</soapenv:Envelope>
```

**Figure IV.27 : Requête SOAP associée à l'obtention des cours de l'apprenant concerné.**

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getCoursesResponse xmlns:ns2="http://loginEJB/">
      <return>
        <coursID>14</coursID>
        <description>Généralités et Principe du Data Mining</description>
        <designation>Data Mining I</designation>
      </return>
      <return>
        <coursID>15</coursID>
        <description>Apprentissage Supervisé : Les arbres de décision.</description>
        <designation>Data Mining II</designation>
      </return>
    </ns2:getCoursesResponse>
  </S:Body>
</S:Envelope>
```

**Figure IV.28 : Réponse SOAP associée à l'obtention des cours de l'apprenant concerné.**



**FigureIV.29 : Téléchargement d'un chapitre donné par un Apprenant.**

#### **IV-5- Conclusion**

A travers ce chapitre, nous avons procédé à la présentation et à la description des outils de développement qui ont servis à la réalisation du travail demandé. Nous avons également montré la partie Vue de notre application web, autrement dit, les interfaces résultantes ; sans oublier les couples requêtes / réponses associés aux *Services Web* de *type SOAP*.

## ***Conclusion Générale***

Cette section représente une clôture de ce présent mémoire qui nous a permis de découvrir le monde de la formation sous un autre angle, c'est-à-dire le « e-learning » mais plus essentiellement, nous a fait connaître et apprécier une technologie hors pair, à savoir « les *Services Web* » accompagnée de leurs galaxie, qui était d'ailleurs le sujet phare de notre projet ; sans oublier l'un des aspects techniques le plus convoité de nos jours, ce qu'on appelle « *l'interopérabilité* ».

Cette interopérabilité justement est beaucoup plus nécessaire aujourd'hui et pour cause, les utilisateurs ont besoin de services spécifiques. Dans le paysage e-Learning, la situation est similaire à ceci. En effet, Il existe plusieurs LMS, différents systèmes de portefeuille, référentiels, contenus...etc. Ces derniers sont développés dans des langages de programmation distincts, utilisent différents contextes et la plupart du temps ne sont pas en mesure de communiquer avec d'autres systèmes.

Pour ce faire, une solution a été pensée et dès lors vu le jour ; à savoir : les fameux *Services Web*.

Sur ce, notre projet de fin d'études a été pour nous un beau challenge. En effet, celui-ci nous a offert une opportunité pour élargir nos connaissances et capacités dans le domaine de la programmation orientée objet ainsi que l'exploitation de l'architecture JavaEE ; néanmoins, comme tout autre travail, des améliorations ou bien une extension peuvent être apportées ; à titre d'exemple : la mise en ligne (upload) de fichiers multiples ou bien diverses.

Au final, nous espérons que notre présent mémoire servira de documentation pour les futurs étudiants et leur sera d'un très grand intérêt.

## ***Références***

### ***Bibliographie***

- [1] Jean-Paul Figer, Architectures orientées Services (SOA), pluralitas non est ponenda sine necessitate, 2006.
- [2] Fatima-Zahra BELOUADHA, SOA (SERVICE ORIENTED ARCHITECTURE), Options IQL et TI, Filière Génie Informatique.
- [6] Pierrick Vanneau, Mathieu Samson, Julien Donel, *JINI – SOAP : Présentation*, 2005.
- [15] Amosse Edouard, Doctorant, comprendre l'Architecture des web services Rest.
- [16] M. Dumas, M.-C. Fauvet, Intergiciel et Construction d'Applications Réparties, 2008.
- [17] Ludovic Denoyer, Architecture Orientée Services, 2015.
- [19] N.Denos, K.Silini, Travailler dans un environnement numérique évolutif, D1.3 :Tenir compte des enjeux de l'interopérabilité.
- [25] FORMATION E-LEARNING, L'ESSENTIEL À RETENIR, LIVRE BLANC , Talentsoft Learning.
- [30] Peter Westerkamp, E-Learning as a Web Service, University of Munster "", Institut für Wirtschaftsinformatik "Leonardo-Campus 3, D-48149 Munster, Germany "pewe@wi.uni-muenster.de

## ***Webographie***

- [3] <https://dzone.com/articles/microservices-vs-soa-2>
- [4] <https://www.jmdoudoux.fr/java/dej/chap-service-web.htm#service-web-2>
- [5] <https://openclassrooms.com/fr/>
- [7] <http://www-igm.univ-mlv.fr/~dr/XPOSE2004/woollams/soap.html>
- [8] [https://www.w3schools.com/xml/xml\\_soap.asp](https://www.w3schools.com/xml/xml_soap.asp)
- [9] <http://glossaire.infowebmaster.fr/xml/>
- [10] [www.wikipedia.org](http://www.wikipedia.org)
- [11] <https://msdn.microsoft.com/fr-fr/library/bb469912.aspx#RFC2376>
- [12] <http://amine.benkirane.over-blog.com/2012/05/architecture-logicielle-soap-vs.-rest-web-services>
- [13] <http://androidhubb.blogspot.com/2016/08/advantages-and-disadvantages-of-soap.html>
- [14] <https://badger.developpez.com/tutoriels/dotnet/web-service-rest-avec-wcf-3-5/#LI>
- [18] <https://blog.octo.com/securite-des-services-web-1ere-partie/>
- [20] <http://blog.xebia.fr/2009/02/18/web-service-interoperability-ws-i/>
- [21] [http://benhur.telug.quebec.ca/SPIP/inf9013/IMG/pdf/M3\\_colDef.pdf](http://benhur.telug.quebec.ca/SPIP/inf9013/IMG/pdf/M3_colDef.pdf)
- [22] [https://www.procomptable.com/papier\\_recherche/mmbf.htm#\\_Toc35069933](https://www.procomptable.com/papier_recherche/mmbf.htm#_Toc35069933)

- [23] <http://ellicom.com/blogue/formation-en-ligne/e-learning-101-synchrone-et-asynchrone-quoi-comment-et-pourquoi/#.WqrdqsPwblV>
- [24] [http://www.e-doceo.net/telechargement/livre\\_blanc\\_e-learning.pdf](http://www.e-doceo.net/telechargement/livre_blanc_e-learning.pdf)
- [26] <https://e-learning.prestataires.com/conseils/quest-ce-que-le-learning>
- [27] <https://www.researchgate.net/publication/237609489>
- [28] <http://www.altissia.com/blog/fr/developpement-learning-algerie/>
- [29] <http://maghrebemergent.info/high-tech/82040-beeform-lance-la-premiere-plateforme-e-learning-100-algerienne.html>
- [31] <https://www.lucidchart.com/pages/what-is-UML-unified-modeling-language>
- [32] [fr.wikipedia.org](http://fr.wikipedia.org)
- [33] <http://www.uml.org/what-is-uml.htm>
- [34] <https://www.techopedia.com/definition/27238/glassfish>
- [35] [http://gardeux-vincent.eu/Documents/ProjetJEE/HPV\\_JSF\\_Castor/jsf.html](http://gardeux-vincent.eu/Documents/ProjetJEE/HPV_JSF_Castor/jsf.html)
- [36] <https://www.supinfo.com/articles/single/6402-primefaces-java-introduction>