

504/1  
M/004.40

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études Master

12/791

Filière : Informatique

Option : Ingénierie des Medias

Thème :

---

**Apprentissage collaboratif basé service**

---

Encadré Par :

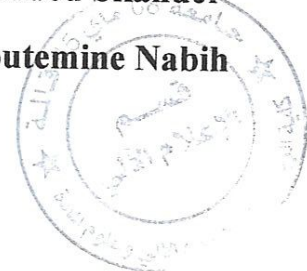
Mr : Hadjeris Mourad

Présenté par :

Hadded Skander

Boutemine Nabih

Juin 2012



## *Remerciements*

*En préambule à ce mémoire, On tient à remercier, tout d'abord **Dieu** qui nous a permis de goûter le plaisir de la vie estudiantine.*

*On souhaite adresser nos remerciements les plus sincères aux personnes qui nous ont apportés leur aide et qui ont contribué à l'élaboration de ce modeste travail ainsi qu'à la réussite de cette formidable formation.*

*On tient à remercier sincèrement Monsieur **HADJERIS**, qui, en tant qu'encadreur de mémoire, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi que pour l'inspiration, l'aide et le temps qu'il a bien voulu nous consacrer et sans qui ce mémoire n'aurait jamais vu le jour.*

*On n'oublie pas **nos parents** pour leur contribution, leur soutien et leur patience.*

**SKANDER,**

*Je ne remercierai jamais assez ma famille: ma mère, mon père, ma sœur, mes deux frères, ma tante, mon oncle, ma fiancée, pour leur encouragements et leur soutien.*

## *NABIH,*

*Merci à mes **parents** pour leur patience, encouragements et confiance, merci à mon frère qui m'a toujours aidé et supporté dans mes décisions et surtout à **Guerfi Abdelhamid**.*

*Enfin, on exprime notre sincère gratitude et nos respects à tous les enseignants du département d'Informatique qui ont contribué à notre formation durant toutes ces dernières années.*

*Un grand merci à l'université **08 MAI 1945** qui nous à ouvert ses portes au moment ou d'autres étaient fermés, ce précieux diplôme lui ai dédié*

*Enfin, on adresse nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire.*

*Merci à tous et à toutes.*

## *Dédicaces*

### *Nabih*

*A mes amis qui ont passés avec moi tous le cycle universitaire, à l'ami de toujours et mon binôme **Skander**, à mon deuxième frère **Abdou**, à **Leilou, Riad, djo et Didina**,*

*A tous mes amis, des signes d'amitié pour*

*Ahdelaziz, Hamza, Brahim, Mustafa, Manel, Mounia, Rym, , Aref, Hmed, Mus et*

*Orchidée*

## *Skander*

*Je dédie ce mémoire à ma chère mère, et tout le reste de ma famille ainsi qu'à tous mes amis qui m'ont accompagné tout au long de mon cycle universitaire, et avec qui j'ai passé des moments inoubliables, à mon bras droit NABIH, Abdou, Riad, Leylou, Didina, Djo et tous les autres ...*

*Une spéciale dédicace à ma fiancée*

*<<Miryam.>>*



# Table des matières

Introduction générale.....	1
<b>Chapitre I : L'apprentissage collaboratif et les services web</b>	
1.1 Introduction.....	4
1.2 Définition de l'apprentissage collaboratif.....	4
1.3 Importance de l'apprentissage en ligne.....	4
1.4 Concept de base de l'e- learning.....	5
1.5 Les services web.....	7
1.5.1 Définition.....	7
1.5.2 Intérêt d'un service web.....	7
1.5.3 Les caractéristiques d'un service web.....	8
1.5.4 Architecture générale d'un service web.....	8
1.5.5 Fonctionnements des services web.....	11
1.5.6 Description en couche des web services.....	12
1.5.7 Le Protocol de communication SOAP.....	13
1.5.7.1 Structure d'un message SOAP.....	15
1.5.8 Le langage de description WSDI.....	16
1.5.9 L'annuaire des services UDDI.....	22
1.5.9.1 Consultation de l'annuaire.....	22
1.5.9.2 Structures de données UDDI.....	23
1.5.9.3 L'interface UDDI.....	24
1.6 Conclusion.....	25

## Chapitre II : Etude comparative entre plateforme

2.1	Introduction .....	27
2.2	Définition des plateformes.....	27
2.3	Principe de la plateforme.....	27
2.4	Les plateformes de formation on line.....	28
2.5	Le tracking.....	29
2.6	Outils de communication .....	29
2.7	Sélection de plateforme pour la comparaison.....	30
2.7.1	Moodle version 1.8.2.....	31
2.7.1.1	Les points faibles de la plateforme.....	34
2.7.2	Sakai version 2.4.0.....	34
2.7.2.1	Les pointes faibles de la plateforme.....	36
2.8	Comparaison.....	37
2.8.1	Communauté, dynamisme, international.....	37
2.8.2	Documentation.....	39
2.8.3	Gestion du nombre d'utilisateurs.....	40
2.8.4	Outils collaboratifs.....	41
2.8.5	Adaptabilité et modularité de la plateforme.....	41
2.8.6	Installation et gestion.....	42
2.8.7	Adaptation possible de la charte graphique.....	44
2.8.8	Système d'exploitation, navigateurs, plug-ins clients.....	44
2.8.9	Ergonomie et utilisabilité.....	45
2.8.10	Grille récapitulative des points clés par plateforme.....	46
2.9	Résultats.....	47
2.10	Résumé.....	47

### **Chapitre III : Conception et modélisation**

3.1	Introduction .....	49
3.2	Architecture générale .....	49
3.2.1	Description de l'architecture.....	50
3.3	Diagramme des cas d'utilisations.....	50
3.4	Diagramme de séquence.....	53
3.5	Diagramme d'activité.....	56
3.6	Conclusion.....	58

### **Chapitre IV : implémentation et environnement de développement**

4.1	Introduction .....	60
4.2	Langage utilisé .....	60
4.3	Editeur utilisé.....	62
4.4	Implémentation.....	65
4.5	Conclusion.....	71
	Conclusion général.....	72
	Bibliographie.....	73

## Table des figures

Figure 1. 1- Principe de base plus l'ensemble des éléments de la formation à distance.....	5
Figure 1. 2-Architecture d'un web service.....	9
Figure 1. 3-Fonctionnement des services web.....	11
Figure 1. 4-Couches technologiques des services web.....	12
Figure 1. 5-Représentation du protocole SOAP.....	14
Figure 1. 6-Structure d'un message SOAP.....	15
Figure 1. 7-Structure d'un document WSDL.....	16
Figure 1. 8-Schéma général de l'annuaire UDDI.....	23
Figure 2. 1-Page d'accueil d'un cours Moodle/version utilisateur.....	33
Figure 2. 2-Page d'accueil d'un apprenant sur sakai.....	35
Figure 3. 1-Architecture général.....	49
Figure 3. 2-Cas d'utilisations.....	51
Figure 3. 3-Diagramme de séquence de l'inscription et l'identification/Apprenant.....	53
Figure 3. 4-Diagramme de séquence de l'étape connexion/Enseignant.....	54
Figure 3.5-Diagramme de séquence tuteur.....	55
Figure 3. 6-Diagramme d'activité apprenant.....	56
Figure 3. 7-Diagramme d'activité professeur.....	57
Figure 4. 1-Logo du langage utilisé.....	61
Figure 4. 2-L'interface de l'éditeur choisi.....	64
Figure 4. 3-.page d'accueil E-EARN.....	65
Figure 4. 4-.page d'inscription apprenant.....	66
Figure 4. 5-.page de connexion.....	67
Figure 4. 6-interface compte apprenant.....	68
Figure 4. 7-interface compte professeur.....	68
Figure 4. 8-interface compte tuteur.....	69
Figure 4. 9-page d'erreur.....	69
Figure 4.10- clic bouton test.....	70
Figure 4.11 –page de passage de test.....	71



## Table des tableaux

Tableau 2. 1- Propriété de la plate forme /Moodle.....	38
Tableau 2. 2- Propriété de la plate forme /Sakai.....	39
Tableau 2. 3- <i>Documentation</i> /Moodle.....	39
Tableau 2. 4- <i>Documentation</i> /Sakai.....	40
Tableau 2. 5- Gestion du nombre d'utilisateur /Moodle.....	40
Tableau 2. 6- Gestion du nombre d'utilisateur /Sakai.....	40
Tableau 2. 7- Outils collaboratifs /Moodle.....	41
Tableau 2. 8- Outils collaboratifs /Sakai.....	41
Tableau 2. 9- Adaptabilité et modularité /Moodle.....	42
Tableau 2. 10- Adaptabilité et modularité /Sakai.....	42
Tableau 2. 11- Intégration de spécifications et standar <i>DS</i> /Moodle.....	43
Tableau 2. 12- Intégration de spécifications et standards /Sakai.....	43
Tableau 2. 13- Installation et gestion /Moodle.....	44
Tableau 2. 14- Installation et gestion /Sakai.....	44
Tableau 2. 15- Adaptation de la charte graphique /Moodle.....	47
Tableau 2. 16- Adaptation de la charte graphique /Sakai.....	47
Tableau 2. 17- Systèmes d'exploitation, navigateurs , plug-ins clients /Moodle.....	48
Tableau 2. 18- Systèmes d'exploitation, navigateurs , plug-ins clients /Sakai.....	48
Tableau 2. 19-Grille récapitulatif.....	47
Tableau 2. 20- Résultats.....	47

## Introduction générale:

L'utilisation des ordinateurs comme moyen d'enseignement a permis l'évolution des systèmes de formation et le renforcement de leur efficacité.

L'ÉAO (Enseignement Assisté par Ordinateur) a réalisé le passage de simples cours magistraux présentés par l'enseignant à des cours programmés performants et attrayants dans lesquels on a mis l'accent sur l'importance de l'interactivité, et depuis l'apparition d'internet la composition ordinateur+internet a donné naissance à la fin des années 1980 à l'apprentissage collaboratif en ligne expérimenté dès l'apparition de l'enseignement en ligne sous la dénomination « computer conferencing » par courrier électronique d'abord, par forums ensuite, connue aussi dans le milieu universitaire sous le nom de « e-formation » ou « e-learning », synonyme moderne de formation à distance et qui est devenue aujourd'hui une spécialité à part entière dans le domaine de l'informatique.

Les enseignants expérimentant l'apprentissage collaboratif en ligne se réjouissent de pouvoir superviser un plus grand nombre de groupes d'étudiants collaborant en ligne que lorsqu'ils collaborent en face à face (ou en mode « présentiel »). L'apprentissage collaboratif ou l'e-learning combine les avantages des enseignants présentiels (interaction) et à distance (flexibilité) et évite leurs inconvénients. Il constitue ainsi un nouveau type d'enseignement indépendant qui utilise les forums et chats, et facilite ainsi la création de liens sociaux, en réintroduisant les notions de classe et de groupe.

Notre travail porte l'accent essentiellement sur l'apprentissage collaboratif basé service, car un service web permet la communication et l'échange de données entre applications et systèmes hétérogènes et ce à travers une présentation standardisée des services offerts et d'un protocole standard de communication permettant de structurer les messages échangés entre les composants logiciels. L'objectif des services web est donc de faciliter l'accès aux applications et ainsi de simplifier l'échange de données. Leur particularité réside dans le fait qu'ils utilisent la technologie internet comme infrastructure pour la communication entre les composants logiciels. Les architectures orientées service constituent un paradigme de conception et de réalisation applicable aux différents niveaux d'interaction d'un système communicant.

Le web service utilise le protocole http pour les échanges de données. Il est établi sur les standards existants comme : SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language), UDDI (Universal Description, Discovery, and Intergration).

## **Chapitre 1**

### **L'apprentissage collaboratif et les services web**



## 1.1 Introduction :

*L'apprentissage collaboratif en ligne* fait réjouir plusieurs enseignants l'ayant expérimenté, car ce nouveau mode qui diffère de la méthode classique permet de superviser un plus grand nombre de groupes d'étudiants collaborant en ligne, et ce grâce aux *services* de messagerie, forum et autres. Nous allons voir dans ce premier chapitre qui utilise cette méthode et grâce à quels outils techniques elle a pu s'imposer si rapidement.

## 1.2. Définition de l'apprentissage collaboratif:

Dans un livre dédié à l'apprentissage collaboratif, 35 pages entières ont été consacrées à sa définition. Nous proposons la suivante: est qualifiée d'apprentissage collaboratif toute activité d'apprentissage réalisée par un groupe d'apprenants ayant un but commun, étant chacun source d'information, de motivation, d'interaction, d'entraide... et bénéficiant chacun des apports des autres, de la synergie du groupe et de l'aide d'un formateur facilitant les apprentissages individuels et collectifs. [MAR 04]

## 1.3 Importance de l'apprentissage en ligne :

Comme l'enseignement en ligne, l'apprentissage collaboratif en ligne fait bénéficier d'une grande flexibilité de temps et de lieu (stimulant l'autonomie et la réflexion) ainsi que d'une excellente interaction asynchrone. C'est pourquoi la nature collective du « computer conferencing » peut-être l'élément le plus fondamental et critique sous tendant l'élaboration de théories ainsi que la conception et la mise en œuvre des activités éducatives en ligne. L'apprentissage collaboratif constitue l'apport pédagogique le <sup>plus</sup> important de l'enseignement en ligne. Dès l'apparition du web, l'e-learning a suscité un grand engouement et il a pris un essor rapide sous forme de matériel didactique affiché sur le web, souvent sans interaction humaine, puis parfois en interaction par courriel entre chaque apprenant et son tuteur, enfin par des forums permettant l'interaction de chaque apprenant avec un tuteur et avec ses pairs. Mais les activités d'apprentissage collaboratif en ligne en petit groupe ont été négligées par beaucoup de réalisateurs de formations en ligne, car étant trop pressés pour multiplier les formations en ligne pour se consacrer à la conception et à l'animation

d'activités de petits groupes alors que celle-ci constituent l'innovation la plus bénéfique de l'enseignement en ligne. [CHR 03]

#### 1.4. Concept de base du e-learning :

Le e-learning consiste à utiliser les ressources de l'informatique et de l'internet pour acquérir, à distance, des connaissances. Ce type d'apprentissage, constituant une des facettes du e-learning, se différencie par l'utilisation de la technologie web. Ces applications et ces contenus sont dispensés via un ordinateur relié à intranet ou à internet. Les contenus constituent une étape plus avancée que l'utilisation des CD-Rom où les cours sont dispensés en salle.

Le e-learning est <sup>(UN)</sup> un domaine révolutionnaire, un changement dans le champ de l'apprentissage. Les instructions qu'il offre en ligne peuvent être fournies n'importe quand et n'importe où par une variété de solutions d'apprentissages électroniques.

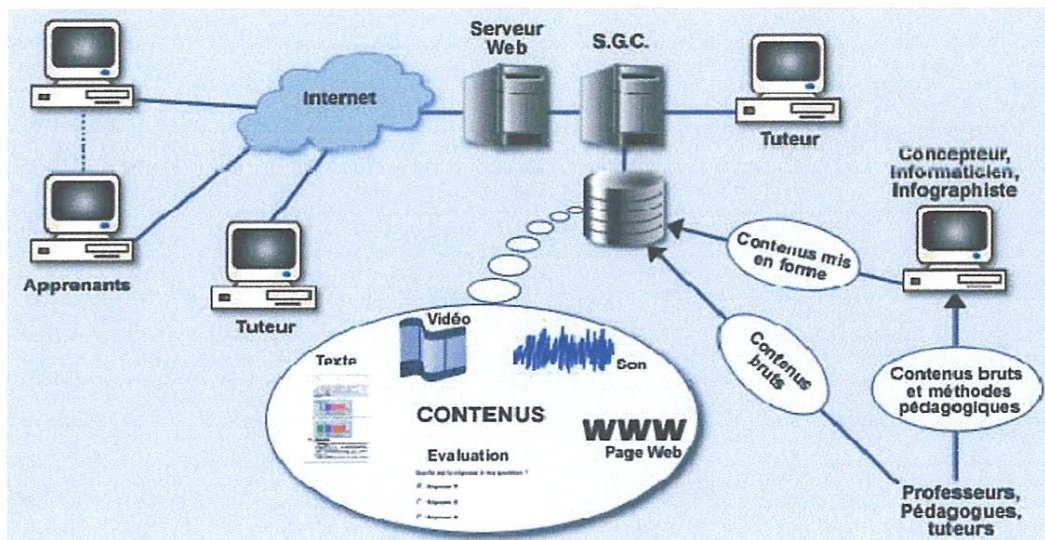


Figure 1-1. Principe de base plus l'ensemble des éléments de la formation à distance [ANN 07]

D'après le schéma ci-dessus, nous constatons qu'il y a certains acteurs tels que : l'apprenant, tuteur et le concepteur. Qui sont ces acteurs ? Quels sont leur rôle ? Nous allons commencer par définir chacun d'eux.

- a) **L'apprenant** : les apprenants sont les individus (hommes, femmes ou enfants) désirant suivre une formation. Cette formation est suivie depuis leurs postes de travail respectifs ; le contenu de la formation est organisé selon un parcours pédagogique.
  
- b) **Le tuteur** : son rôle est fondamental. Il ne s'agit plus de transmettre des connaissances, mais d'accompagner l'apprenant dans ses acquisitions, de lui faire acquérir un maximum d'autonomie. Il est là pour combattre la sensation d'isolement que pourrait ressentir l'apprenant dans une classe virtuelle et pour maintenir sa motivation par des encouragements, des questionnements, des suggestions d'approfondissement. Il est de plus en plus fréquent que les formations soient associées à la mise en place d'une communauté d'apprentissage que le tuteur sera chargé d'animer.
  
- c) **Le formateur** : Il définit le parcours en fonction des compétences possédées par l'apprenant avant la formation, en fonction des objectifs pédagogiques visés par la formation et en fonction des résultats obtenus lors des évaluations intermédiaires : le parcours est donc individualisé.
  
- d) **Le concepteur** : Il conçoit le cours et travaille à partir du scénario pédagogique fourni par l'enseignant. Il a en charge la médiatisation des contenus, c'est-à-dire la mise en forme des ressources pédagogiques et leur intégration sur la plateforme. [BAR 08]

Tous ces acteurs peuvent par la suite communiquer entre eux grâce aux services offerts par la plateforme comme : le chat, les messageries, les forums... etc. Cela s'effectue par le biais de programmes qui fonctionnent avec des protocoles. Ces programmes sont : les services web, que nous allons détailler dans le prochain sous-titre.

détaillé

Section



## 1.5. Les services web :

### 1.5.1 Définition :

Un service web est un programme décrit en XML, identifié par un URI (Uniform Resource Identifier) proposant diverses fonctionnalités et utilisé grâce à des protocoles qui sont eux aussi décrit en XML et basés sur l'échange de messages, ces derniers seront transmis par HTTP, FTP, SMTP...etc.

La notion de **service** désigne essentiellement des sites Web qui ne se contentent pas de fournir une information statique mais permettent d'effectuer certaines actions ou modifications. Donc le terme web service désigne une notion mettant l'accent sur les technologies du web. [1]

### 1.5.2 Intérêt d'un web service :

Les services Web fournissent un lien entre applications. Ainsi, des applications utilisant des technologies différentes peuvent envoyer et recevoir des données au travers de protocoles compréhensibles par tout le monde. Les services Web sont **normalisés** car ils utilisent les standards XML et HTTP pour transférer des données et ils sont compatibles avec de nombreux autres environnements de développement. Ils sont donc indépendants des plates-formes. C'est dans ce contexte qu'un intérêt très particulier a été attribué à la conception des services Web puisqu'ils permettent aux entreprises d'offrir des applications accessibles à distance par d'autres entreprises. Cela s'explique par le fait que les services Web n'imposent pas de modèles de programmation spécifiques. En d'autres termes, les services Web ne sont pas concernés par la façon dont les messages sont produits ou consommés par des programmes. Cela permet aux vendeurs d'outils de développement d'offrir différentes méthodes et interfaces de programmation au-dessus de n'importe quel langage de programmation, sans être contraints par des standards comme c'est le cas de la plate-forme *CORBA* qui définit des ponts spécifiques entre le langage de définition IDL et différents langages de programmation. Ainsi, les fournisseurs d'outils de développement peuvent facilement différencier leurs produits avec ceux de leurs concurrents en offrant différents niveaux de sophistication. Les services Web représentent donc la façon la plus efficace de partager des méthodes et des



fonctionnalités. De plus, ils réduisent le temps de réalisation en permettant de tirer directement parti de services existants. [1]

### 1.5.3. Les caractéristiques d'un web service :

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

Un service web possède les caractéristiques suivantes :

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

L'intégration d'application en implémentant des services Web produit des systèmes faiblement couplés, le demandeur du service ne connaît pas forcément le fournisseur. Ce dernier peut disparaître sans perturber l'application cliente qui trouvera un autre fournisseur en cherchant dans l'annuaire. [2]

### 1.5.4. Architecture générale d'un service web :

Les services Web reprennent la plupart des idées et des principes du Web (HTTP, XML), et les appliquent à des interactions entre machines. Comme pour le World Wide Web, les services Web communiquent via un ensemble de technologies fondamentales qui partagent une architecture commune. Ils ont été conçus pour être réalisés sur de nombreux systèmes développés et déployés de façon indépendante. Les technologies utilisées par les services Web sont HTTP, WSDL, REST, XML-RPC, SOAP et UDDI. [1]

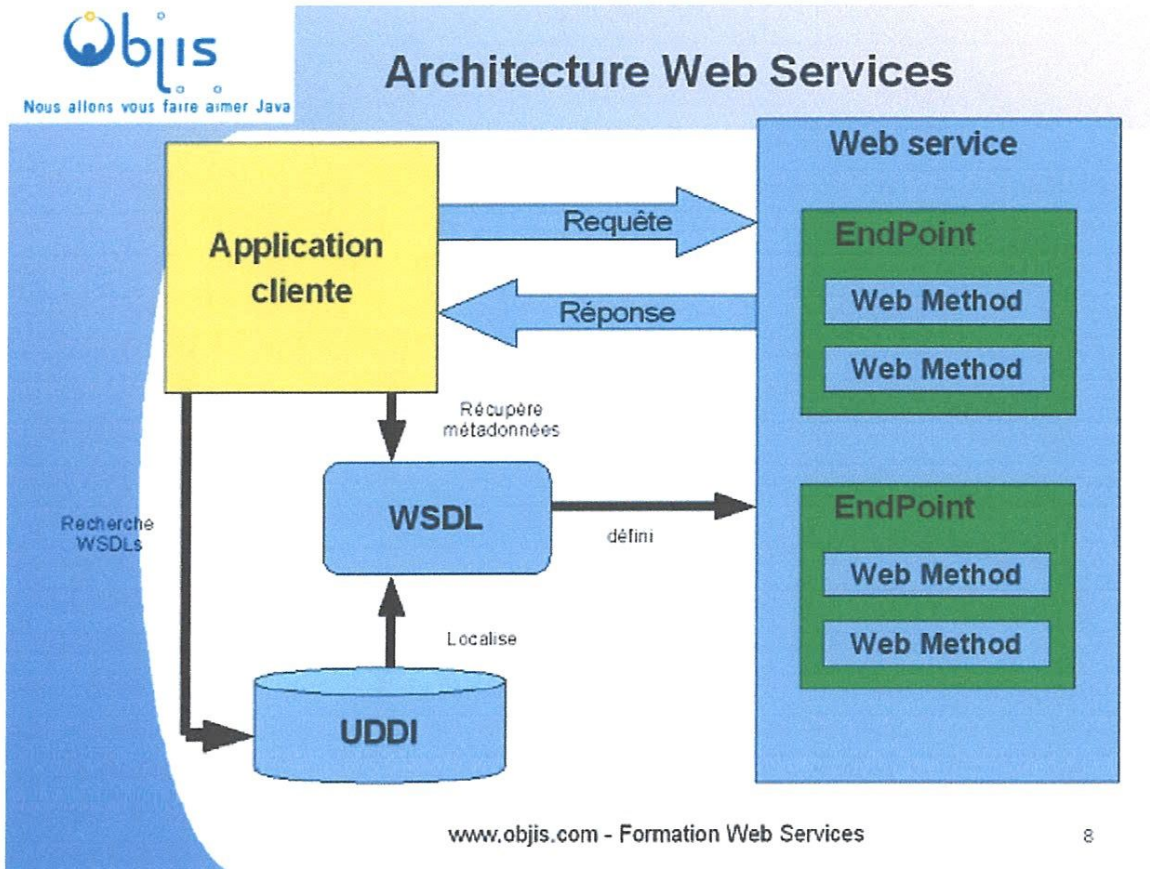


Figure 1.2: Architecture d'un web service [3]

a) REST:

REST (Representational State Transfer) est une architecture de services Web. Élaborée en l'an 2000 par Roy Fielding, l'un des créateurs du protocole HTTP, du serveur Apache HTTPd et d'autres travaux fondamentaux, REST est une manière de construire une application pour les systèmes distribués comme le World Wide Web.

b) XML-RPC:

XML-RPC est un protocole simple utilisant XML pour effectuer des messages RPC . Les requêtes sont écrites en XML et envoyées via HTTP POST. Les requêtes sont

intégrées dans le corps de la réponse HTTP. XML-RPC est indépendant de la plateforme, ce qui lui permet de communiquer avec diverses applications. Par exemple, un client Java peut parler de XML-RPC à un PerlServer !

c) SOAP:

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

d) WSDL:

WSDL (Web Services Description Language) est un langage de description standard. C'est l'interface présentée aux utilisateurs. Il indique comment utiliser le service Web et comment interagir avec lui. WSDL est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées.

e) UDDI :

UDDI (Universal Description, Discovery and Integration) est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services Web aux clients. Les informations qu'il contient peuvent être séparées en trois types :

- les pages blanches qui incluent l'adresse, le contact et les identifiants relatifs au service Web
- les pages jaunes qui identifient les secteurs d'affaires relatifs au service Web
- les pages vertes qui donnent les informations techniques

Nous allons étudier plus en détail ces trois dernières technologies.(WSDL,UDDI,SOAP) [4]



### 1.5.5. Fonctionnement des services Web :

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

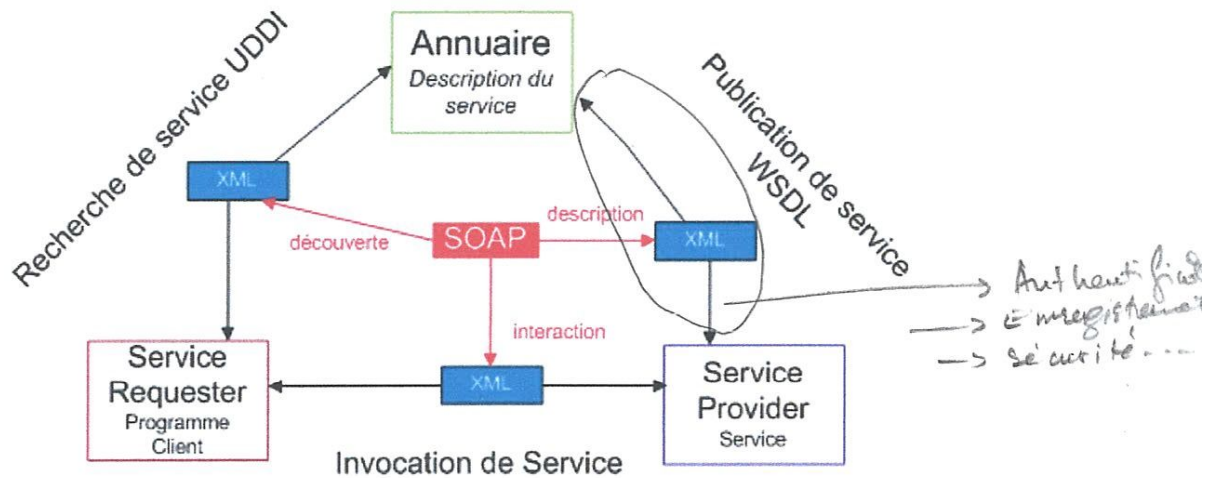


Figure 1.3 : Fonctionnement des services web [1]

- ✓ **Service provider service** : Le fournisseur de service met en application le service Web et le rend disponible sur Internet.
- ✓ **Service requester programme client** : C'est n'importe quel 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)
- ✓ **Annuaire service registry** : Le registre de service est un annuaire de services. Le registre fournit un endroit central où les programmeurs peuvent publier de nouveaux services ou en trouver. Les interactions entre ces trois acteurs suivent plusieurs étapes :



### 1.5.6. Description en couche des web services :

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

<b>Découverte de services</b>	UDDI
<b>Description de services</b>	WSDL
<b>Communication</b>	SOAP
<b>Transport</b>	HTTP

Figure 1.4 : Couches technologiques des services Web [1]

- Le transport de messages XML-RPC ou SOAP est assuré par le standard HTTP.
- SOAP ou XML-RPC prévoit la couche de communication basée sur XML pour accéder à des services Web.
- La description d'un service Web se fait en utilisant le langage WSDL. WSDL expose l'interface du service.
- La publication et la découverte des services Web sont assurées par le biais du référentiel UDDI. Un référentiel UDDI est un catalogue de services Web.

#### a) Couche transport :

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

#### b) Couche communication :

Cette couche est responsable du formatage des données échangées de sorte que les messages peuvent être compris à chaque extrémité. Actuellement, deux styles architecturaux totalement

différents sont utilisés pour ces échanges de données. Nous avons d'un côté l'architecture orientée opérations distribuées (protocoles RPC) basée sur XML et qui comprend XML-RPC et SOAP et de l'autre côté une architecture orientée ressources Web, REST (Representational State Transfer) qui se base uniquement sur le bon usage des principes du Web (en particulier, le protocole HTTP).

#### **c) Couche description de services :**

Cette couche est responsable de la description de l'interface publique du service Web. Le langage utilisé pour décrire un service Web est WSDL qui est la notation standard basée sur XML pour construire la description de l'interface d'un service. Cette spécification définit une grammaire XML pour décrire les services Web comme des ensembles de points finaux de communication (ports) à travers lesquels on effectue l'échange de messages.

#### **d) Couche publication de services :**

Cette couche est chargée de centraliser les services dans un registre commun, et de simplifier les fonctionnalités de recherche et de publication des services Web. Actuellement, la découverte des services est assurée par un annuaire UDDI (Universal Description, Discovery, and Integration).

#### **1.5.7. Le Protocol de communication SOAP :**

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

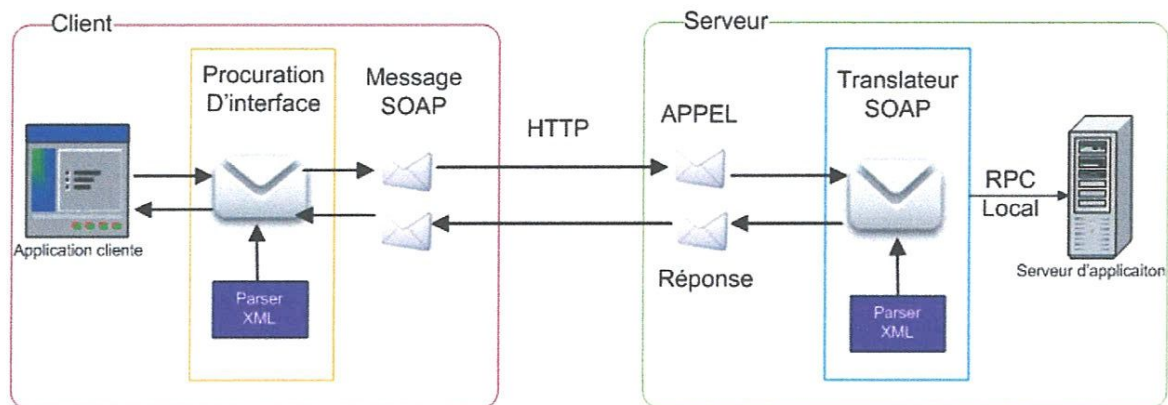


Figure 1.5 : Représentation du protocole SOAP

SOAP est bien plus populaire et utilisé que XML-RPC. C'est une recommandation du W3C. D'après cette recommandation, SOAP est destiné à être un protocole léger dont le but est d'échanger des informations structurées dans un environnement décentralisé et distribué. 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 plates-formes et les technologies existantes.

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

- **Spécification** car SOAP est un document qui définit le modèle de communication. L'idée de base est que si les deux parties ont créé des programmes de mêmes spécifications, elles seront en mesure d'interagir de façon transparente.
- **Omniprésente** car SOAP est défini à un niveau suffisamment élevé d'abstractions que tout système d'exploitation et combinaison de langages de programmation peuvent être utilisés pour créer des programmes compatibles SOAP.
- **Basé sur XML**, SOAP est construit sur XML, ce qui signifie que les documents SOAP sont des documents XML construits en fonction d'un cahier de charges plus strict.

- **Infrastructure distribuée**, SOAP ne précise pas quelles données peuvent être déplacées ou bien quels appels de fonctions peuvent avoir lieu sur elle. Les applications construites sur la spécification SOAP peuvent déplacer les données d'un ordinateur A à un ordinateur B et par la suite à une autre application écrite sur la même spécification.

### 1.5.7.1. Structure d'un message SOAP :

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

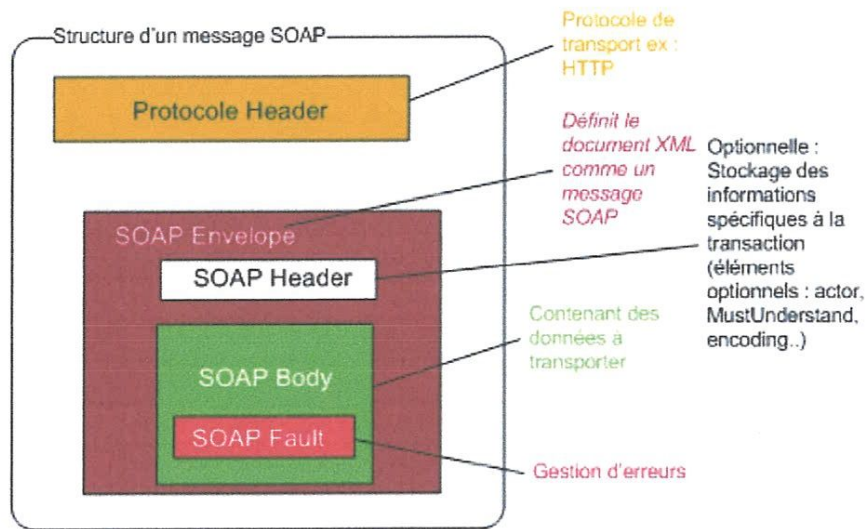


Figure 1.6 : Structure d'un message SOAP [1]

- **SOAP enveloppe** (enveloppe) est l'élément de base du message SOAP. L'enveloppe contient la spécification des espaces de désignation (namespace) et du codage de données.
- **SOAP header** (entête) est une partie facultative qui permet d'ajouter des fonctionnalités à un message SOAP de manière décentralisée sans agrément entre les parties qui communiquent. C'est ici qu'il est indiqué si le message est mandataire ou optionnel. L'entête est utile surtout, quand le message doit être traité par plusieurs intermédiaires.
- **SOAP body** (corps) est un *container* pour les informations mandataires à l'intention du récepteur du message. Il contient les méthodes et les paramètres qui seront exécutés par le destinataire final.



- **SOAP fault** (erreur) est un élément facultatif défini dans le corps SOAP et qui est utilisé pour reporter les erreurs.

### 1.5.8. Le langage de description WSDL :

Un document WSDL se compose d'un ensemble d'éléments décrivant les types de données utilisés par le service, les messages que le service peut recevoir, ainsi que les liaisons SOAP associées à chaque message. Le schéma suivant illustre la structure du langage WSDL qui est un document XML, en décrivant les relations entre les sections constituant un document WSDL

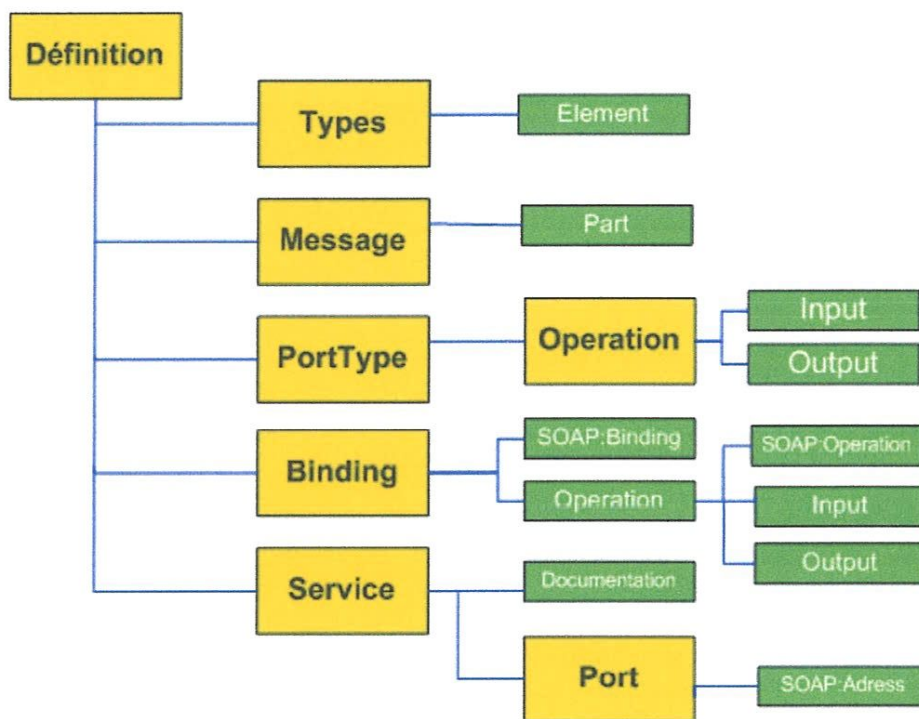


Figure 1.7 : Structure d'un document WSDL [1]

Le fichier WSDL contient la structure suivante :

- **Types** : fournit la définition de types de données utilisés pour décrire les messages échangés.

- **Messages** : représente une définition abstraite (noms et types) des données en cours de transmission.
- **PortTypes** : décrit un ensemble d'opérations. Chaque opération a zéro ou un message en entrée, zéro ou plusieurs messages de sortie ou d'erreurs.
- **Binding** : spécifie une liaison entre un <portType> et un protocole concret (SOAP, HTTP...).
- **Service** : indique les adresses de port de chaque liaison.
- **Port** : représente un point d'accès de services défini par une adresse réseau et une liaison.
- **Opération** : c'est la description d'une action exposée dans le port.

Le document WSDL peut être divisé en deux parties. Une partie pour les **définitions abstraites**, tandis que la deuxième contient les **descriptions concrètes**. La description concrète est composée des éléments qui sont orientés vers le client pour le service physique. Les trois éléments concrets XML présents dans un WSDL sont :

- <wsdl:service> ;
- <wsdl:port> ;
- <wsdl:binding>.

La description abstraite est composée des éléments qui sont orientés vers la description des capacités du service Web. Ses éléments abstraits définissent les messages SOAP de façon totalement indépendante de la plate-forme et de la langue. Cela facilite la définition d'un ensemble de services pouvant être implémentés par différents sites Web. Les quatre éléments abstraits XML qui peuvent être définis dans un WSDL sont :

- <wsdl:types> ;
- <wsdl:message> ;
- <wsdl:operation> ;
- <wsdl:portType>



### ✚ L'élément types :

L'élément <types> décrit tous les types de données utilisés entre le client et le serveur. Ces types sont l'équivalent en structures C++ ou Java à des classes qui ne contiennent que des données et pas de méthodes. WSDL n'est pas liée exclusivement à un système de typage, mais il utilise le XML schéma de la spécification W3C.

### ✚ L'élément message :

L'élément <message> comprend la section Messages. Si nous envisageons les opérations comme des fonctions, alors un élément <message> définit les paramètres pour cette fonction. L'exemple suivant représente les messages correspondant à l'ajout d'un nouveau client à un service Web.

Chaque élément enfant <part> de l'élément <message> correspond à un paramètre et possède un attribut de nom et de type, tout comme un paramètre de fonction a un nom et un type. Les paramètres d'entrée sont définis dans un élément <message> unique et séparé des paramètres de sortie, qui se trouvent dans leur propre élément <message>. Le message addCustomer va ajouter un nouveau client (customer) au service Web par l'envoi d'une instance de l'élément client que nous avons défini dans l'élément <type>. Le nom d'un élément <message> de sortie se termine par Response. Dans cet exemple, le message de réponse est confirmationResponse, il renvoie au client un nombre entier lui indiquant le succès de l'opération.

### ✚ L'élément opération :

L'élément <operation> est analogue à un appel de méthode en Java ou d'une sous-routine dans Visual Basic. La différence est que seulement trois messages sont autorisés dans une opération :

- **Input Message** : définit les données que le service Web s'attend à recevoir.
- **Output Message** : définit les données que le service Web prévoit d'envoyer en réponse.
- **Fault Message** : définit les messages d'erreurs qui peuvent être retournés par le service Web.

Plusieurs types d'opération peuvent être déclarés dans un document WSDL :

- **Request/Response** : le client envoie la demande, et le service répond.
- **Solicit/Response** : un service Web envoie un message au client, et le client répond.
- **One-way** : un client envoie un message au service Web, mais ne s'attend à aucune réponse.
- **Notification** : un service Web envoie un message au client, mais n'attend pas de réponse.

#### ✚ L'élément portType :

Un port est simplement une suite d'opérations. De nombreux langages de programmation appellent cela une bibliothèque, un module ou une classe, mais dans le monde de l'échange de messages, les points de connexion sont des ports, et la définition abstraite d'un port est appelée <portType>.

L'élément <portType> contient l'ensemble des opérations que peut effectuer un service Web. Cependant, il ne fournit pas d'informations sur la façon de se connecter directement à ce service. Il prévoit un point d'arrêt où un client peut obtenir des informations sur tous les traitements offerts par un service Web. La syntaxe d'un portType est définie comme suit :

Code : XML - [Sélectionner](#)

```
1 <wsdl:portType name="newCustomerPortType">
2 <wsdl:operation name="createNewCustomer">
3 <wsdl:input message="addCustomer"/>
4 <wsdl:output message="confirmationResponse"/>
5 </wsdl:operation>
6 </wsdl:portType>
```

L'élément <PortType> défini dans l'exemple est identifié par un nom unique newCustomerPortType. Il contient l'opération createNewsCustomer de type Request/Rcponse qui va ajouter un nouveau client au service Web en utilisant les messages d'entrée (input) et de sortie (output) définis précédemment dans la section message.

### L'élément binding :

L'élément <binding> permet d'obtenir les informations nécessaires pour connecter physiquement un service Web. Il décrit les spécifications concrètes de la manière dont le service sera implémenté : protocole de communication et format des données pour les opérations et messages définis par un portType particulier. Le langage WSDL possède des extensions internes pour définir des services SOAP. De fait, les informations spécifiques à SOAP se retrouvent dans cet élément. L'élément <binding> a deux objectifs. Tout d'abord, il sert de lien entre les éléments abstraits et les éléments concrets dans le WSDL. Ensuite, il fournit un conteneur pour des informations telles que le protocole et l'adresse du service Web.

### L'élément port :

Un port définit un point d'accès individuel, en spécifiant une adresse unique pour une liaison binding. La syntaxe d'un <port> est la suivante :

Code : XML - [Sélectionner](#)

```
1 <wsdl:port binding="newCustomerBinding" name="newCustomerPort">
2   <soap:address location="http://www.stevepotts.com:1776/soap/servlet/rpcrouter">
3 </wsdl:port>
```

L'élément port contient deux attributs : l'attribut name et l'attribut binding. L'attribut name donne un nom unique parmi tous les ports définis dans le document WSDL. Dans notre exemple, le nom du port est newCustomerPort. L'attribut binding fait référence à l'élément binding newCustomerBinding défini dans la section binding du document WSDL. Le port contient un élément <soap:address> qui spécifie à l'aide de l'attribut location une URL représentant l'adresse du port. Dans notre exemple, l'adresse du port est : «[.]www.stevepotts.com:1776/soap/servlet/rpcrouter ».

Un port ne doit pas définir plus d'une adresse. Un port ne doit pas définir d'autres informations de liaisons autres que celles de l'adresse.

### 📌 L'élément service :

L'élément <service> définit les ports soutenus par le service Web. Il contient aussi un élément <documentation> qui fournit la documentation lisible par l'homme. Pour chaque liaison supportée, un élément port est désigné. L'élément <service> n'est donc qu'une simple collection de ports.

### L'élément définition :

L'élément racine dans un document WSDL est <wsdl:definition>. Il contient un attribut targetNamespace qui définit un certain nombre d'espaces de noms namespace auquel tous les noms déclarés dans un élément du document WSDL appartiennent, ce qui permet d'éviter les conflits de nommage.

La syntaxe de l'élément <definition> :

### Code : XML - Sélectionner

```
1 <wsdl:definition name="customerExemple" targetNamespace="http://www.stevepotts.com/customer.wsdl"
2     xmlns:soap="http://www.schemas.xmlsoap.org/wsdl/soap/"
3     xmlns:wsdl="http://www.schemas.xmlsoap.org/wsdl/"
4     xmlns="http://www.stevepotts.com/customer.xsd">
```

Dans cet exemple, l'attribut targetNamespace a pour valeur l'URL : «[.]www.stevepotts.com/customer.wsdl». Cela signifie que tous les noms déclarés dans ce document WSDL appartiennent à cet espace de noms. Le reste du document WSDL apparaît sous cet élément et la fin de ce document est indiquée par </wsdl:definition>.



### 1.5.9. L'annuaire des services UDDI :

L'annuaire des services UDDI est un standard pour la publication et la découverte des informations sur les services Web. La spécification UDDI est une initiative lancée par *ARIBA*, Microsoft et *IBM*. Cette spécification n'est pas gérée par le W3C mais par le groupe OASIS. La spécification UDDI vise à créer une plate-forme indépendante, un espace de travail (framework) ouvert pour la description, la découverte et l'intégration des services des entreprises.

#### 1.5.9.1. Consultation de l'annuaire :

L'annuaire UDDI se concentre sur le processus de découverte de l'architecture orientée services (SOA), et utilise des technologies standards telles que XML, SOAP et WSDL qui permettent de simplifier la collaboration entre partenaires dans le cadre des échanges commerciaux. L'accès au référentiel s'effectue de différentes manières.

- Les pages blanches comprennent la liste des entreprises ainsi que des informations associées à ces dernières (coordonnées, description de l'entreprise, identifiants...).
- Les pages jaunes recensent les services Web de chacune des entreprises sous le standard WSDL.
- Les pages vertes fournissent des informations techniques précises sur les services fournis.

Les entreprises publient les descriptions de leurs services Web en UDDI, sous la forme de fichiers WSDL. Ainsi, les clients peuvent plus facilement rechercher les services Web dont ils ont besoin en interrogeant le registre UDDI. Lorsqu'un client trouve une description de service Web qui lui convient, il télécharge son fichier WSDL depuis le registre UDDI. Ensuite, à partir des informations inscrites dans le fichier WSDL, notamment la référence vers le service Web, le client peut invoquer le service Web et lui demande d'exécuter certaines de ses fonctionnalités.

Le scénario classique d'utilisation d'UDDI est illustré ci-dessous. L'entreprise B a publié le service Web S, et l'entreprise A est client de ce service :

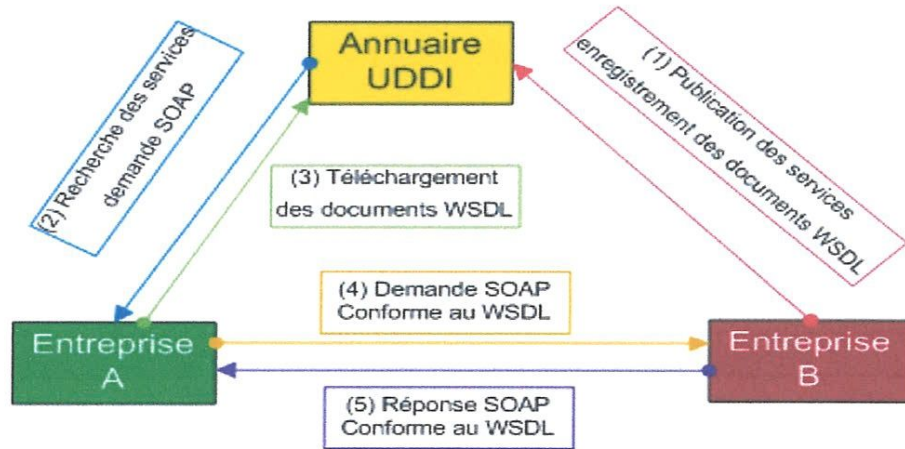


Schéma Général de l'annuaire UDDI

Figure 1.8 : schéma général de l'annuaire UDDI [1] *fonctionnement des services*

### 1.5.9.2. Structures de données UDDI :

Un registre UDDI se compose de quatre types de structures de données, le **businessEntity**, le **businessService**, le **bindingTemplate** et la **tModel**. Cette répartition par type fournit des partitions simples pour faciliter la localisation rapide et la compréhension des différentes informations qui constituent un enregistrement.

#### a) **BusinessEntity (entité d'affaires) :**

Les « businessEntities » sont en quelque sorte les pages blanches d'un annuaire UDDI. Elles décrivent les organisations ayant publié des services dans le répertoire. On y trouve notamment le nom de l'organisation, ses adresses (physiques et Web), des éléments de classification, une liste de contacts ainsi que d'autres informations.

#### b) **BusinessService (service d'affaires) :**

Les « businessServices » sont en quelque sorte les pages jaunes d'un annuaire UDDI. Elles décrivent de manière non technique les services proposés par les différentes organisations. On y trouve essentiellement le nom et la description textuelle des services ainsi qu'une référence à l'organisation proposant le service et un ou plusieurs « bindingTemplate ».



#### c) **BindingTemplate (modèle de rattachement) :**

UDDI permet de décrire des services Web utilisant HTTP, mais également des services invoqués par d'autres moyens (SMTP, FTP...). Les « bindingTemplates » donnent les coordonnées des services. Ce sont les pages vertes de l'annuaire UDDI. Ils contiennent notamment une description, la définition du **point d'accès** (une URL) et les éventuels « tModels » associés.

#### d) **tModel (index) :**

Les « tModels » sont les descriptions techniques des services. UDDI n'impose aucun format pour ces descriptions qui peuvent être publiées sous n'importe quelle forme et notamment sous forme de documents textuels (XHTML, par exemple). C'est à ce niveau que WSDL intervient comme le vocabulaire de choix pour publier des descriptions techniques de services.

### 1.5.9.3. L'interface UDDI :

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

- **Interrogation inquiry** : cette interface permet de rechercher des informations dans un répertoire UDDI et de lire les différents enregistrements suivant le modèle de données UDDI.
- **Publication** : cette interface permet de publier des informations dans un répertoire UDDI conformément à son modèle de données.
- **Sécurité** : cette interface est utilisée pour obtenir et révoquer les jetons d'authentification nécessaires pour accéder aux enregistrements protégés dans un annuaire UDDI.
- **Contrôle d'accès et propriété custody and ownership transfer** : cette interface permet de transférer la propriété d'informations (qui est à l'origine attribuée à l'utilisateur ayant publié ces informations) et de gérer les droits d'accès associés.
- **Abonnement Subscription** : cette interface permet à un client de s'abonner à un ensemble d'informations et d'être averti lors des modifications de ces informations. [5]

## 1.6 Conclusion :

L'apprentissage collaboratif est une méthode qui postule que l'échange interpersonnel est la base de la construction des connaissances. Il constitue l'apport pédagogique le plus important de l'enseignement en ligne.

En ce qui concerne les services Web, ce sont des protocoles standards utilisés pour établir un domaine d'intégration des applications dans des plateformes d'apprentissage en ligne.

L'un des facteurs ayant contribué au succès des services Web est sans doute l'utilisation des standards Internet tels que XML et HTTP. En conséquence, tout système capable d'analyser du texte et de communiquer via un protocole de transport Internet standard peut communiquer avec un service Web.

La combinaison entre l'apprentissage en ligne et les services web, conduit vers des plateformes différentes. La question qui se pose est de savoir quelles sont les fonctionnalités que doit posséder une plate-forme, et quelles sont les normes qu'elle doit respecter. C'est ce que nous verrons lors du chapitre suivant dans lequel nous ferons une étude comparative entre les plateformes.

## **Chapitre2**

### **Etude comparative entre plates-formes**

## 2.1 Introduction :

Avec la progression des technologies, les métiers ont évolué, donc les besoins pour la formation initiale ou continue ont augmenté. Vues cette émergence technologique et la génération internet qui s'impose de jour en jour, nous devons penser à un apprentissage rapide et efficace. Cette apprentissage existe aujourd'hui et est représenté en plateforme, des plateformes qui diffèrent d'un point de vue contextuel, fonctionnel, structurel et/ou conceptuel. Nous nous intéresserons dans ce chapitre à ces différences entre plateformes et allons en faire une étude comparative.

## 2.2 Définition des plates formes:

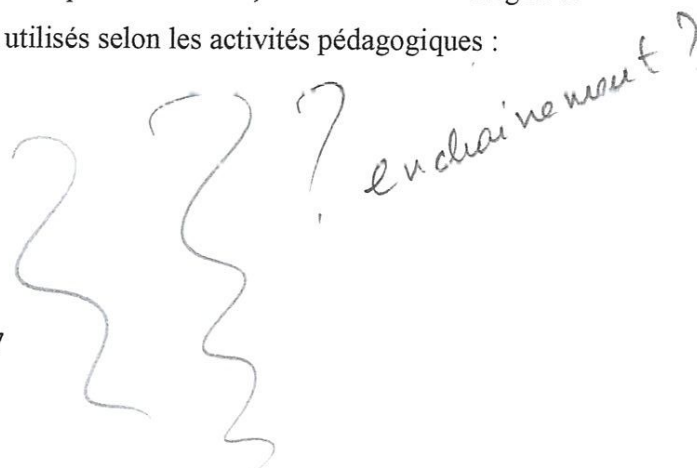
Une plate forme est un logiciel installé sur un serveur informatique. Pour l'utilisateur, le logiciel se présente sous la forme d'un site web auquel il peut se connecter et accéder notamment aux différentes formations prévues par le formateur.

Coté formateur, la plateforme de formation permet de nombreuses actions, par exemple, effectuer simplement le suivi détaillé de la formation.

Les plateformes offrent des outils de gestion et d'administration : gestion des cours, gestion des apprenants, suivi des apprenants (tracking), élaboration de parcours pédagogiques, auto-inscription des apprenants. Des modules complémentaires peuvent enrichir les fonctionnalités disponibles : rendu de travaux, visibilité des cours, gestion des notes, moteur de recherche, glossaire intégré, outils collaboratifs Wiki, espace personnel, Portfolio, outils de sondage, de statistiques, fil RSS... [HEL 07]

## 2.3 Principe de la plate-forme :

La plate-forme met à disposition des apprenants un ensemble de cours et de ressources accessibles à travers un réseau local d'entreprise ou à partir d'internet, en utilisant un navigateur classique. Différents outils sont mis à disposition utilisés selon les activités pédagogiques :





**a) Mode synchrone : (communication simultanée)**

Les apprenants et tuteurs communiquent par clavardage (clavier + bavardage), partage d'applications ou en utilisant la vidéo conférence.

**b) Mode asynchrone :**

Utilisé pour des échanges différés, grâce à des outils tels que : la messagerie, blogs ou forums. Ces derniers favorisent donc les échanges plus personnels

**2.4 Les plates-formes de formation on-line :**

Une plateforme pédagogique est un logiciel qui assiste la conduite des formations présentielle et à distance. Elle est basée sur des techniques de travail collaboratif et regroupe les outils nécessaires aux trois principaux acteurs de la formation : apprenant, tuteur, administrateur. Elle fournit à chaque acteur un dispositif qui a pour première finalité l'accès à distance au contenu pédagogique, l'auto apprentissage, l'autoévaluation et le télé tutorat via l'utilisation des moyens de travail et de communication à plusieurs : visioconférence, e-mail, forums, chats, annotations, etc. Le but est donc de combler la perte de cohésion et de stimulation de la salle que peut sentir l'apprenant devant sa machine. [HEL 07]

L'usage de ces systèmes est relativement standard : le tuteur crée des parcours de formation type, 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. 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, etc.

Avec l'évolution des techniques, des infrastructures de réseau et des normes, le nombre de plateformes et environnements de formation ont augmenté de manière significative.

Dans cette partie, nous mentionnons les critères de sélection des plateformes et nous décrivons brièvement quelques plateformes existantes.

### 2.5 Le tracking :

Les fonctions de tracking permettent le suivi pédagogique des apprenants. Un grand nombre d'information concernant le parcours individuel de chaque stagiaire peut être stocké dans le système. Il s'agit couramment des résultats obtenus dans les modules d'évaluation et de test, le temps passé sur un module de cours, le temps passé à réaliser un exercice, etc.

Ces données sont ensuite interprétées par le tuteur, qui peut ainsi se rendre compte de l'assiduité des stagiaires, des problèmes rencontrés sur certains chapitres ou sur la résolution de problèmes.

Pour que les fonctions de tracking soient pleinement opérationnelles le système complet – Système de Gestion de Contenu (SGC et contenus) devra répondre à un standard normalisé.

### 2.6 Outils de communication :

Généralement, les plateformes SGC intègrent un certain nombre d'outils de communication. La liste des outils disponibles dépend de la richesse du SGC. Ces outils sont souvent très usités sur le web et ne sont en rien spécifiques au e-learning, ils font simplement partie intégrante du SGC et ne nécessitent pas de configuration spécifique sur les postes des apprenants (par exemple, il n'est pas nécessaire de configurer un client messagerie pour profiter de ce service). On peut citer :

- ✚ **La messagerie** : C'est un espace de communication entre les membres du groupe (stagiaire et tuteurs). Ce système permet l'envoi et la réception de mails avec ou sans fichier attaché. Il peut être interne à la plate-forme et ne nécessite pas d'avoir un e-mail personnel.
- ✚ **Le forum** : Il peut être public ou réservé à un groupe, permet aux stagiaires et aux tuteurs de poster des messages qui seront accessibles à l'ensemble des membres du groupe de formation, de répondre aux messages déjà postés et ainsi engager une discussion sur un sujet donné.
- ✚ **Le chat** : Il permet à l'ensemble du groupe de discuter en temps réel.

- ✚ **Les documents pédagogiques partagés** : Une zone commune, peut être partagée pour mettre à disposition du groupe différents documents : Documents sous forme numériques (Word, Excel, Pdf...etc) pour l'ensemble du groupe.
- ✚ **Les news** : La possibilité de créer des news permettra, par exemple, d'informer ses apprenants de nouveaux rendez-vous, ou de leur communiquer diverses informations.
- ✚ **Les sondages** : Le sondage permettra de poser des questions ouvertes ou de proposer des choix de réponses. Ultérieurement, les réponses obtenues sur un cours en particulier pourront être analysées.
- ✚ **Le bloc-notes** : C'est un espace privé à chaque utilisateur. Il peut y noter toutes les informations qu'il souhaite, ce qu'il a retenu de sa formation, des actions à mener...etc. Ces notes seront archivées par date et il est possible de les compiler dans une même note.
- ✚ **Les glossaires** : des glossaires peuvent être créés, destinés à l'ensemble des utilisateurs ou un groupe en particulier. Les différents termes avec leur signification sont enregistrés, un moteur de recherche permet aux utilisateurs de faciliter la recherche d'un terme dans les glossaires.
- ✚ **Les FAQs** : Cette espace permet aux tuteurs et aux administrateurs d'industrialiser la réponse à des questions récurrentes.

## 2.7 Sélection de plateforme pour la comparaison :

Afin de pouvoir faire une étude comparative, nous nous sommes d'abord intéressé au nombre de plateformes qui existent sur la toile et leur type. Ainsi, selon le site thot-cursus <http://www.awt.be> qui propose un répertoire de plateformes e-learning, le dernier recensement datant de 2007 rapporte qu'il y a 233 plateformes, 47 plates-formes open-source, 11 plates-formes publiques et 175 plates-formes commerciales. [KOU]

Nous avons choisi de faire une sélection dans les plateformes open source actuellement utilisées dans les grandes structures universitaires ou de formation en nous basant sur les critères suivants :

1. Importante communauté d'utilisateurs et de développeurs, dynamique et d'envergure internationale.
2. Documentation en ligne de l'installation de la plateforme à l'utilisation par les enseignants et les apprenants.



3. Plateforme pouvant gérer un grand nombre d'utilisateurs.
4. Outils collaboratifs dédiés aux échanges autour d'apprentissages communs,
5. Adaptabilité et modularité de la plateforme.
6. Installation et gestion de la plateforme simple basée sur des technologies Web courantes.
7. Adaptation possible de la charte graphique.
8. Multi-systèmes d'exploitation : côté serveur (quels systèmes sont supportés) et côté client (quels navigateurs ?, besoin de plugins particuliers ?)
9. Ergonomie, utilisabilité des plateformes pour les enseignants comme pour les apprenants.

Après avoir visité et installé quelques plateformes opensource, nous avons choisi de faire notre étude sur seulement deux plateformes qui sont : Moodle et Sakai.

Ces deux plateformes sont libres et gratuites.

**2.7.1. Moodle version 1.8.2 :** est une plateforme d'apprentissage en ligne servant à créer des communautés d'apprenants autour de contenus et d'activités pédagogiques. Le terme Moodle est l'acronyme de Modular Object-Oriented Dynamic Learning Environment, mais il veut aussi dire «flâner» en anglais. Ses fonctionnalités permettent de créer des interactions entre pédagogues, apprenants et ressources pédagogiques formant ainsi un réseau et parfois même une véritable communauté autour d'un thème choisi par les membres de la plateforme (apprentissage d'un logiciel comme utilisation de la plateforme). C'est une particularité que l'on retrouve peu dans les autres plateformes étudiées.

Moodle fait partie des systèmes de e-formation qui sont appelés dispositifs de « Formation Ouverte et à Distance » (FOAD), pour favoriser un cadre de formation Socio-constructiviste. Ce courant de pensée affirme que les gens construisent activement leurs nouvelles connaissances en interagissant avec leur voisinage.

La connaissance est renforcée si vous pouvez l'utiliser avec succès dans un environnement plus large. Cela met l'accent sur le fait qu'il n'y a pas seulement un transfert d'information d'un cerveau à un autre, mais que tout est conditionné par l'interprétation et l'interaction avec un groupe social.

L'interface de Moodle se présente sous forme de différents blocs : le bloc central présente les documents et les activités du cours. Ils peuvent être classés selon plusieurs formats :



- Thématique : en fonction de thèmes ou de sujets du cours
- Hebdomadaire : en fonction d'un agenda ou du calendrier
- Informel : en fonction de sujets de discussion et de forums.

Pour présenter les ressources, l'enseignant peut composer une page de texte : texte court sans mise en forme, créer une page web : page au format HTML qui peut être éditée à l'aide de l'éditeur intégré à la plateforme, mettre un lien vers un site Internet ou un fichier déposé sur le serveur, afficher le contenu d'un dossier : lien vers une liste de fichiers déposés sur le serveur, ajouter un fichier IMS Content Package : lien vers un fichier de format SCORM ou AICC, et insérer une étiquette permettant de commenter la ressource.

Les blocs latéraux affichés sur les pages web donnent accès aux différents outils et liens du cours, par exemple :

- *Personnes* : liste des inscrits au cours + la liste des différents sous-groupes + accès à son profil
- *Cours* : la liste des cours auxquels est inscrit l'utilisateur
- *Recherche* : outil de recherche dans les forums du cours
- *Administration* : relevé des notes de l'utilisateur...
- *Dernières nouvelles* : les dernières brèves publiées sur le forum
- *Prochains événements* : les activités inscrites au calendrier de son cours
- *Calendrier* : les activités classées en fonction du calendrier
- *Utilisateurs en ligne* : la liste des personnes, enseignants et usagers, connectés au cours
- *Fils RSS ...* [MAR 07]

Les membres d'un cours ont accès aux activités suivantes si l'enseignant les a sélectionnées :

- *Messagerie électronique* : "chat" ou salon de discussion (possibilité de l'ouvrir certains jours, à heure précise, de manière hebdomadaire, etc.).
- *Forum* : différents types de forums (sujets imposés par l'enseignant, sujets proposés par les étudiants, évaluation ou commentaire possibles, etc.).
- *Devoir*: remise de travaux avec évaluation de l'enseignant.
- *Test* : suite de QCM (Questionnaire à Choix Multiples), de questions vrai/faux, appariement, etc.
- *Leçon* : document comprenant des questions et plusieurs parcours possibles en fonction des réponses (évaluation possible).

- *Atelier* : remise de travaux avec évaluation par les étudiants.
- *Glossaire* : production collective d'un document organisé alphabétiquement (commentaire et évaluation possibles).
- *Wiki* : production collective d'un document hypertexte (commentaires possibles de l'enseignant).
- *Journal* : rédaction d'un journal personnel (commentaires possibles).
- *Dialogues* : messagerie interne entre membres du cours.

Toutes les activités sont paramétrables par l'enseignant.

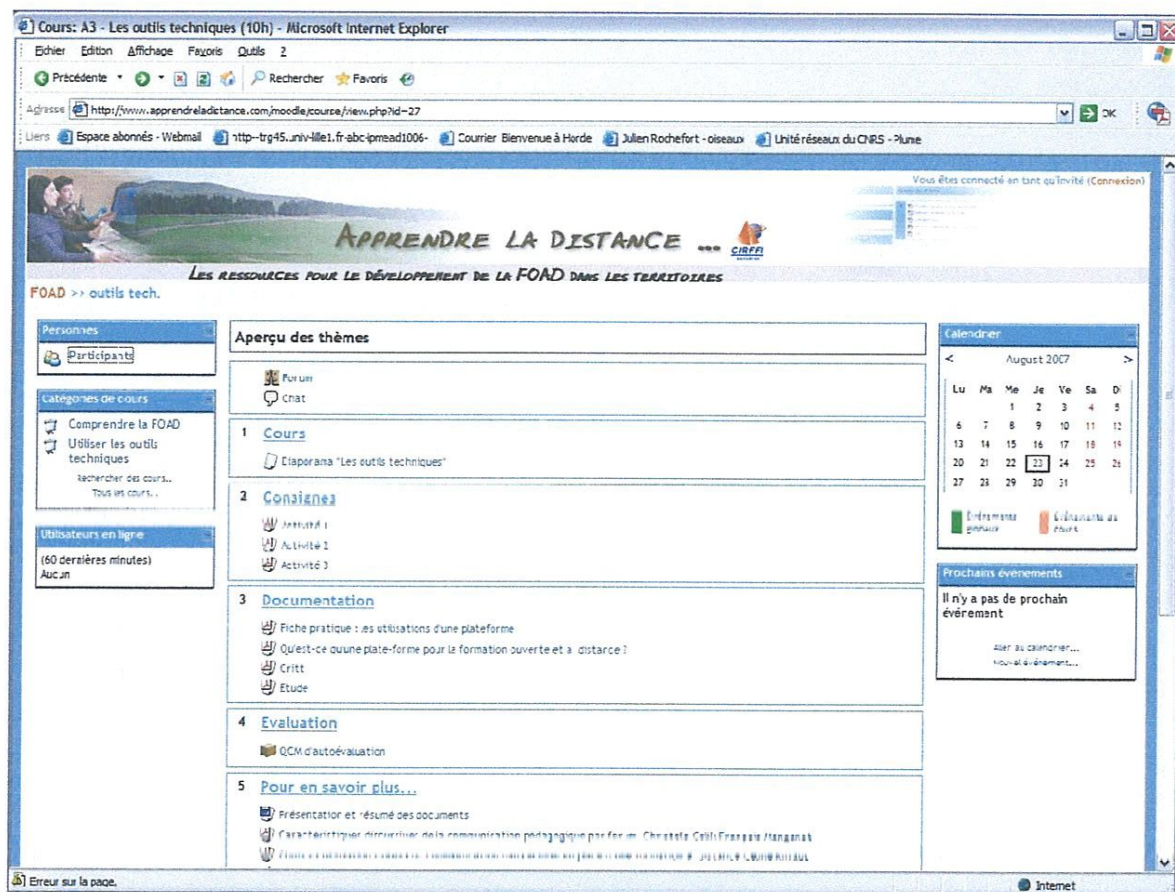


Figure 2.1 : Page d'accueil d'un cours moodle –version utilisateur-

[ref]

L'enseignant peut diviser son groupe classe en plusieurs sous-groupes de manière à faciliter la communication entre les personnes. Les groupes ont des outils dédiés : forum, sondage, chat. Moodle a été créée de manière modulaire : elle permet de répondre aux besoins d'un formateur isolé comme d'une institution académique. Aujourd'hui, le développement de Moodle est

fortement influencé par les demandes de la communauté d'administrateurs et d'utilisateurs (enseignants, pédagogues). Ce projet bénéficie d'un développement très actif à l'échelle mondiale.

#### **2.7.1.1. Les points faibles de la plateforme :**

- Possibilité de test de positionnement mais gestion manuelle, pas d'affectation automatique de parcours
- Agenda du cours (visualisation des nouveautés, prévision d'événements pour tout type d'utilisateur, les travaux à rendre apparaissent automatiquement dans l'agenda) mais pas d'agenda personnel
- Communication en mode synchrone : clavardage mais pas de mode vidéo-conference sauf si l'on rajoute un module.
- Moodle est une plateforme très riche en fonctionnalités, sa prise en main par les apprenants peut nécessiter un temps d'adaptation car les pages peuvent être très chargées en informations.
- Pour les enseignants, la diversité et la spécificité de tous les paramétrages des outils peuvent paraître trop complexes aux yeux d'utilisateurs peu familiers en FOAD. Cependant, de nombreux tutoriels en ligne existent. Il est aussi important que l'administrateur de la plateforme ou le coordinateur du projet puisse être disponible pour aider les enseignants à appréhender la totalité des fonctionnalités de Moodle. Afin d'éviter les échecs, l'un des premiers cours à installer sur la plateforme est celui sur l'utilisation de Moodle.

**2.7.2 Sakai version 2.4.0 :** Le projet Sakai a été lancé en 2004 par quatre universités américaines avec pour objectif de consolider leur développement en matière de plateforme d'apprentissage. Chacune de ces universités, soit Indiana University, Massachusetts Institute of Technology (MIT), Stanford University et University of Michigan utilisaient des systèmes de gestion de cours différents, souvent développés en interne. Au groupe se sont joints des membres de uPortal2 ainsi que du projet OKI (Open Knowledge Initiative3) .

Sakai est un projet, auquel sont rattachés une fondation, un comité de direction et des partenaires institutionnels et commerciaux ; une communauté qui regroupe plusieurs institutions qui coopèrent et maintiennent Sakai. Elle regroupe actuellement plus de 80 universités dans plusieurs pays dans le monde.



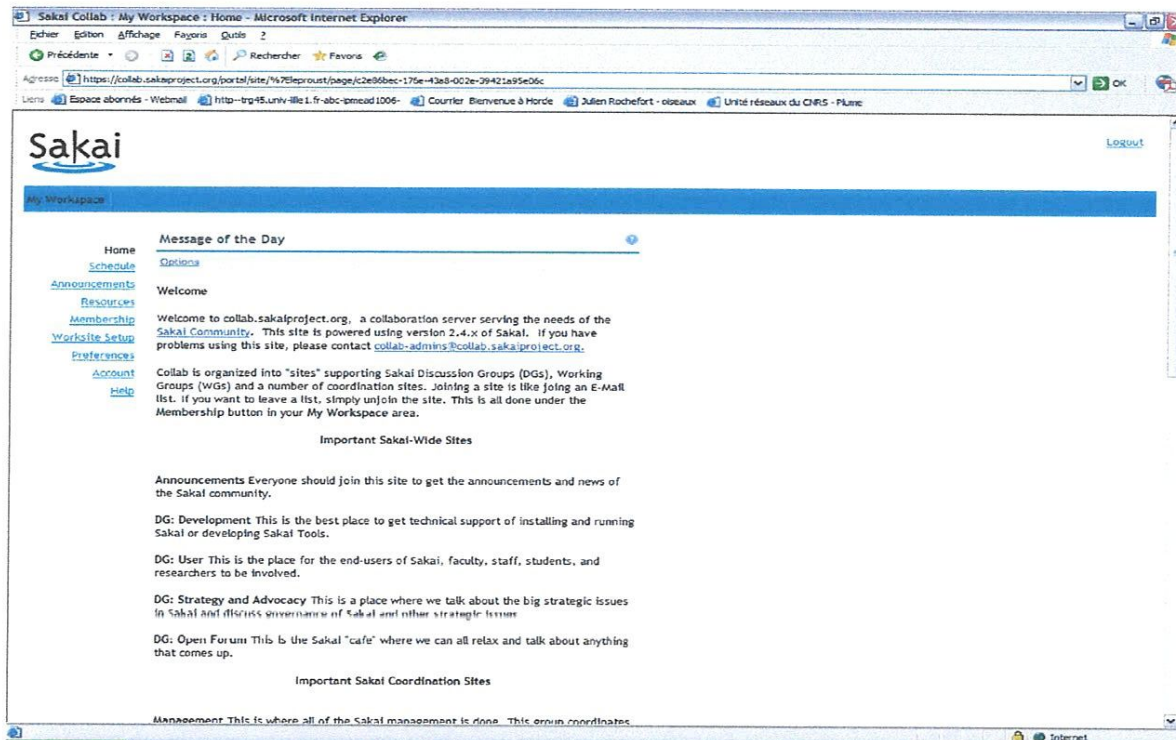


Figure 2.2 : Page d'accueil d'un apprenant sur Sakai [Ref]

Sakai est une plateforme utilisable dans un contexte d'éducation et de formation basée sur une structure ouverte et extensible, incluant une suite d'outils pour le support de l'apprentissage, de la collaboration et de la recherche.

L'objectif de Sakai est de produire une plateforme complète de gestion de cours libre de qualité équivalente ou supérieure aux autres produits du marché. Le produit s'adresse, dans un premier temps, aux institutions universitaires. Ce sont ces dernières qui définissent les orientations, les spécifications et les priorités quant au développement du projet. La structure ainsi que les outils développés sont génériques et peuvent être réutilisés comme base pour la construction d'outils de collaboration ou d'apprentissage dans n'importe quel domaine.

Sakai contient les outils suivants :

- Annonces
- Casier de documents
- Messagerie avec archivage
- Ressources
- Salle de clavardage (chat)



- Forums
- Fil de discussion
- Centre de message et message du jour
- Nouveautés et RSS
- Préférences
- Présentation avec Syllabus
- Gestion des profils
- Recherche dans un entrepôt de documents
- Emploi du temps
- Site web gestion de fichiers avec WebDAV1
- Wiki
- Paramétrage du site
- Parcours pédagogiques avec Melete2.

??

#### 2.7.2.1 Les points faibles de la plateforme :

- Sakai est une plateforme en pleine mutation mais qui évolue rapidement ... Elle manque encore d'interopérabilité :
  - Les importations de modules IMS/1.1.2 ne sont pas toujours opérationnelles, cela dépend des logiciels auteur
  - Le SCORM n'est pas encore implémenté.
- Sa documentation : Sakaïpedia - <http://bugs.sakaiproject.org/confluence/display/ENC/Home> est aujourd'hui, exclusivement en anglais, ce qui peut gêner les utilisateurs qui ne parlent pas cette langue.

Cependant beaucoup d'institutions au Québec s'intéressent au projet Sakai Is ont fondé une association à but non lucratif : Sakai Québec - <http://www.sakaiquebec.org/> et sont en train de développer une traduction en français et en arabe pour les pays maghrébins du projet POLLES - <http://www.polles.org/>.


- La plateforme est développée en Java, ce qui demande un administrateur compétent dans ce langage, l'installation est complexe, l'outil administration n'est pas encore traduit.

Beaucoup de projets sont en cours de développement pour la plateforme (vidéoconférence, traduction en français, portfolio, intégration du SCORM, de CanCore1, de NORMETIC2.).

## 2.8 Comparaison :


A présent, revenons sur les critères cités en haut grâce auxquels nous allons faire notre comparaison entre les deux plateformes.

### 2.8.1 Importante communauté d'utilisateurs et de développeurs, dynamique et d'envergure internationale :

<p><b>Moodle</b> <a href="http://moodle.org">http://moodle.org</a></p>  <p><i>évaluation positive de la plateforme répond au critère</i></p>	<ul style="list-style-type: none"><li>▪ Date de création : 2002</li><li>▪ Dernière version : 1.8.2 - 8 juillet 2007</li><li>▪ Feuille de route : Prochaine version : 1.9 = Août 2007 <a href="http://docs.moodle.org/en/Roadmap#Version_1.9_-_Expected_August_2007">http://docs.moodle.org/en/Roadmap#Version_1.9_-_Expected_August_2007</a></li><li>▪ Equipe Moodle : 1 directeur, 2 administrateurs, 3 développeurs permanents (Moodle recrute actuellement un développeur) et des bureaux en Australie : <a href="http://moodle.com/hq/">http://moodle.com/hq/</a></li><li>▪ 200 000 utilisateurs enregistrés sur le site de Moodle.org, plateforme traduite en 75 langues, utilisée dans plus de 189 pays, plus de 30 506 plateformes enregistrées dans le monde. Plus de statistiques : <a href="http://moodle.org/stats/">http://moodle.org/stats/</a></li><li>▪ Une grande communauté de développeurs organisée : <a href="http://docs.moodle.org/en/Credits">http://docs.moodle.org/en/Credits</a></li><li>▪ 8 communautés de discussion en anglais : <a href="http://moodle.org/course/category.php?id=1">http://moodle.org/course/category.php?id=1</a></li><li>▪ Discussion de la communauté francophone : <a href="http://moodle.org/login/index.php">http://moodle.org/login/index.php</a></li><li>▪ Conférence annuelle d'utilisateurs</li></ul>
---	--

	<p>francophones :</p> <ul style="list-style-type: none"> <li>▪ <a href="http://moodlemoot.iut-tlse3.fr/">http://moodlemoot.iut-tlse3.fr/</a></li> <li>▪ Une documentation en ligne très détaillée, traduite en 25 langues : <a href="http://docs.moodle.org/overview/">http://docs.moodle.org/overview/</a></li> <li>▪ Les dernières actualités de Moodle : <a href="http://moodle.org/mod/forum/view.php?id=82">http://moodle.org/mod/forum/view.php?id=82</a></li> </ul>
--	--

Tableau2.1 : Propriété de la plate forme Moodle

<p><b>Sakai</b> <a href="http://sakaiproject.org/">http://sakaiproject.org/</a></p>  <p><i>évaluation négative problème majeur ne répond pas ?</i></p>	<ul style="list-style-type: none"> <li>▪ Date de création : 2004</li> <li>▪ Dernière version : version 2.4.0 sortie le 19 juillet 2007</li> <li>▪ Feuilles de routes actives : Sakai 2.5 = <a href="http://bugs.sakaiproject.org/confluence/display/MGT/Sakai+2.5+Roadmap">http://bugs.sakaiproject.org/confluence/display/MGT/Sakai+2.5+Roadmap</a></li> <li>▪ Fondation Sakai : 13 personnes : <a href="http://sakaiproject.org/index.php?option=com_content&amp;task=view&amp;id=297&amp;Itemid=507">http://sakaiproject.org/index.php?option=com_content&amp;task=view&amp;id=297&amp;Itemid=507</a></li> <li>▪ Grande communauté internationale basée sur le principe de la « méritocratie » : <a href="http://sakaiproject.org/index.php?option=com_content&amp;task=view&amp;id=108&amp;Itemid=205">http://sakaiproject.org/index.php?option=com_content&amp;task=view&amp;id=108&amp;Itemid=205</a></li> <li>▪ Beaucoup de partenaires enregistrés : <a href="http://sakaiproject.org/index.php?option=com_content&amp;task=view&amp;id=233&amp;Itemid=462">http://sakaiproject.org/index.php?option=com_content&amp;task=view&amp;id=233&amp;Itemid=462</a></li> <li>▪ Partenaires commerciaux : <a href="http://sakaiproject.org/index.php?option=com_content&amp;task=view&amp;id=196&amp;Itemid=461">http://sakaiproject.org/index.php?option=com_content&amp;task=view&amp;id=196&amp;Itemid=461</a></li> <li>▪ Communauté francophone : <a href="http://www.sakaiquebec.org/">http://www.sakaiquebec.org/</a></li> <li>▪ Conférences bi-annuelles : la septième conférence a eu lieu à Amsterdam en Juin 2007 : <a href="http://sakaiproject.org/index.php?">http://sakaiproject.org/index.php?</a></li> </ul>
--	--

	<p>option=com_content&amp;task=view&amp;id=156&amp;Itemid=306</p> <ul style="list-style-type: none"> <li>▪ Encyclopédie Sakai (en anglais seulement, en cours de traduction en français par Sakai Quebec) : <a href="http://bugs.sakaiproject.org/confluence/display/ENC/Home">http://bugs.sakaiproject.org/confluence/display/ENC/Home</a></li> <li>▪ Les dernières actualités de Sakai : <a href="http://sakaiproject.org/index.php?option=com_content&amp;task=blogcategory&amp;id=181&amp;Itemid=555">http://sakaiproject.org/index.php?</a></li> </ul>
--	---

Tableau 2.2 : Propriété de la plateforme Sakai

### 2.8.2 Documentation en ligne de l'installation de la plateforme à l'utilisation par les enseignants et les apprenants:


<p><b>Moodle</b> <a href="http://moodle.org">http://moodle.org</a></p>  <p><i>Oui</i></p>	<ul style="list-style-type: none"> <li>▪ Une documentation en ligne très détaillée et bien organisée abordant tous les sujets : installation, configuration, gestion, pédagogie, ... traduite en 25 langues : <a href="http://docs.moodle.org/overview/">http://docs.moodle.org/overview/</a></li> <li>▪ Plus des tutoriels développés avec Moodle par la communauté, comme par exemple : <a href="http://www.ethnoinformatique.fr/course/view.php?id=39">http://www.ethnoinformatique.fr/course/view.php?id=39</a></li> </ul>
--	--

Tableau2.3 : Documentation/ Moodle


<p><b>Sakai</b> <a href="http://sakaiproject.org/">http://sakaiproject.org/</a></p>  <p><i>Non</i></p>	<ul style="list-style-type: none"> <li>▪ Encyclopédie Sakai (en anglais seulement, en cours de traduction en français par Sakai Quebec), documentation pléthorique : <a href="http://bugs.sakaiproject.org/confluence/display/ENC/Home">http://bugs.sakaiproject.org/confluence/display/ENC/Home</a></li> <li>▪ Sakai Knowledge Base = <a href="https://www.indiana.edu/~sakaikb/">https://www.indiana.edu/~sakaikb/</a></li> </ul>
---	---

Tableau2.4 Documentation/ Sakai



### 2.8.3 Plateforme pouvant gérer un grand nombre d'utilisateurs :


<p><b>Moodle</b>  <a href="http://moodle.org">http://moodle.org</a></p>  <p style="text-align: right;"><i>oui</i></p>	<p>Plus de 63 sites ont plus de 20 000 utilisateurs dont le principal : moodle.org qui compte aujourd'hui 43 cours pour 282 038 utilisateurs enregistrés. Le record est détenu par une plateforme tchèque : 19 223 cours et 41 305 utilisateurs.</p> <p>Plus d'informations sur :  <a href="http://moodle.org/stats/">http://moodle.org/stats/</a></p>
--	--

Tableau2.5 Gestion du nombre d'utilisateur/ Moodle


<p><b>Sakai</b>  <a href="http://sakaiproject.org/">http://sakaiproject.org/</a></p>  <p style="text-align: right;"><i>oui</i></p>	<p>37 500 utilisateurs recensés dans une université du Michigan :</p> <p><a href="http://bugs.sakaiproject.org/confluence/display/winterconf2005/Panel+Session+-+Sakai+Performance+Profiling,+Testing+and+Tuning+--+076">http://bugs.sakaiproject.org/confluence/display/winterconf2005/Panel+Session+-+Sakai+Performance+Profiling,+Testing+and+Tuning+--+076</a></p>
---	--

Tableau2.6 Gestion du nombre d'utilisateur/ Sakai

### 2.8.4 Outils collaboratifs dédiés aux échanges autour d'apprentissages communs :


<p><b>Moodle</b>  <a href="http://moodle.org">http://moodle.org</a></p> 	<ul style="list-style-type: none"> <li>▪ Forums de discussion</li> <li>▪ Clavardage (chat - Discussion)</li> <li>▪ Wiki</li> <li>▪ Devoirs</li> <li>▪ Sondage</li> <li>▪ Glossaire</li> </ul>
---	---

Tableau2.7 : Outils collaboratifs/Moodle


<p><b>Sakai</b>  <a href="http://sakaiproject.org/">http://sakaiproject.org/</a></p> 	<ul style="list-style-type: none"> <li>▪ Forum de discussion</li> <li>▪ Clavardage (chat - Discussion)</li> <li>▪ Casier de documents partagés</li> <li>▪ Création ou mise à disposition de documents par tous les utilisateurs</li> <li>▪ Wiki</li> </ul>
--	--

Tableau2.8 : Outils collaboratifs /Sakai

### 2.8.5 Adaptabilité et modularité de la plateforme :




<p><b>Moodle</b>  <a href="http://moodle.org">http://moodle.org</a></p> 	<ul style="list-style-type: none"> <li>▪ Plus de 217 modules ou plug-ins créés par la communauté :  <a href="http://moodle.org/mod/data/view.php?id=6009">http://moodle.org/mod/data/view.php?id=6009</a></li> <li>▪ 3 types de présentation de cours : <ul style="list-style-type: none"> <li>- Thématique : en fonction de thèmes ou de sujets du cours</li> <li>- Hebdomadaire : en fonction d'un agenda ou du calendrier</li> <li>- Informel : en fonction de sujets de discussion et de forums.</li> </ul> </li> <li>▪ Le processus de restauration d'un cours permet à l'enseignant d'écraser les données de n'importe quel cours, ajouter les données de n'importe quel cours, dupliquer un cours avec son contenu et choisir les options pour inclure les données des élèves ou les données/informations du cours.</li> </ul>
--	---

Tableau2.9 : Adaptabilité et modularité/Moodle

<p><b>Sakai</b>  <a href="http://sakaiproject.org/">http://sakaiproject.org/</a></p> 	<ul style="list-style-type: none"> <li>▪ La plateforme Sakai est basée sur un framework permettant de connecter les outils et services de son choix. La modularité et l'adaptabilité de l'outil sont exceptionnelles si on en maîtrise la technique.</li> </ul>
--	---

*Tableau2.10 : Adaptabilité et modularité /Sakai*

**2.8.6 Installation, mises à jour et gestion de la plateforme simple basée sur des technologies Web courantes :**

<p><b>Moodle</b>  <a href="http://moodle.org">http://moodle.org</a></p> 	<p>Installation et mises à jour rapides et très documentées</p> <ul style="list-style-type: none"> <li>▪ Moodle est principalement développé sur Linux avec Apache, MySQL et PHP (environnement appelé parfois plateforme LAMP), mais est aussi testé régulièrement avec PostgreSQL et sur les systèmes d'exploitation Windows XP/2000/2003 (WAMP), Solaris 10 (Sparc and x64), Mac OS X et Netware 6.</li> </ul> <p>Le support de PostgreSQL, Oracle et Microsoft SQL Server est aussi disponible.</p>
---	---

*Tableau2.11 Integration de specifications et standards /Moodle*


<p><b>Sakai</b>  <a href="http://sakaiproject.org/">http://sakaiproject.org/</a></p> 	<ul style="list-style-type: none"> <li>▪ Installation et mises à jour complexes mais très documentées</li> <li>▪ Sakai est développé en Java et utilise plusieurs technologies issues de J2EE. Il est construit sur des frameworks et librairies libres tels que Spring, Hibernate, log4j et Jakarta Common</li> <li>▪ Différents mécanismes de présentation dont JSF, Velocity et XSL sont utilisés pour la présentation</li> <li>▪ Le déploiement s'effectue généralement à l'aide du serveur Web Apache et du serveur Java Apache</li> <li>▪ Les logiciels requis sont TomCat 5.5.9 ou plus et Java 1.4 ou plus</li> <li>▪ Oracle 9i et MySQL 4.1 et plus</li> <li>▪ Utilisation de Hypersonic SQL</li> </ul>
--	--

Tableau2.12 : Integration de specifications et standards /Sakai

### 2.8.7 Adaptation possible de la charte graphique :


<p><b>Moodle</b>  <a href="http://moodle.org">http://moodle.org</a></p> 	<p>Plus de 33 thèmes graphiques sont proposés par la communauté :</p> <p><a href="http://moodle.org/mod/data/view.php?id=6552">http://moodle.org/mod/data/view.php?id=6552</a></p> <ul style="list-style-type: none"> <li>▪ Il est possible d'adapter la plateforme selon sa charte graphique en utilisant les feuilles de styles et en changeant les bannières et les icônes dans les dossiers dédiés</li> </ul>
---	---

Tableau2.13 : Installation et gestion/Moodle




<p><b>Sakai</b>  <a href="http://sakaiproject.org/">http://sakaiproject.org/</a></p> 	<ul style="list-style-type: none"> <li>▪ 3 « skins » ou habillage sont livrés par défaut dans la plateforme, mais il est possible d'adapter la plateforme selon sa charte graphique en utilisant les feuilles de styles et en changeant les bannières et les icônes dans les dossiers dédiés</li> </ul>
--	---

Tableau2.14 : Installation et gestion /Sakai

**2.8.8 Multiplateformes (systèmes d'exploitation), multi-navigateurs sans téléchargements de plug-ins particuliers ne nécessitant pas la mise à jour de postes-clients :**



<p><b>Moodle</b>  <a href="http://moodle.org">http://moodle.org</a></p> 	<ul style="list-style-type: none"> <li>▪ Quasiment tous les navigateurs internet, mêmes anciens, supportent l'utilisation de Moodle. Les apprenants pourront se connecter à Moodle depuis n'importe quel poste client internet sans que leur poste ne requière une configuration spéciale</li> <li>▪ La plateforme est compatible avec tous les systèmes d'exploitation</li> </ul>
---	--

Tableau2.15 : Adaptation de la charte graphique/Moodle

<p><b>Sakai</b>  <a href="http://sakaiproject.org/">http://sakaiproject.org/</a></p> 	<ul style="list-style-type: none"> <li>▪ Windows : Internet Explorer 5.5 – Netscape 7.1 – Mozilla Firefox</li> <li>▪ Macintosh : Netscape 7.1 et + Mozilla Firefox</li> <li>▪ Autres plateformes : Netscape 7.1 et + Mozilla Firefox</li> <li>▪ Certaines fonctionnalités ne sont pas</li> </ul>
--	--

	<p>gérées par : Macintosh Safari,  Macintosh Internet Explorer, Camino 0.7  Netscape 7.0, ou des anciennes  version d'Opera  Version</p>
--	--

Tableau2.16 : Adaptation de la charte graphique /Sakai

**2.8.9 Ergonomie, utilisabilité des plateformes pour les enseignants comme pour les apprenants :**


<p><b>Moodle</b>  <a href="http://moodle.org">http://moodle.org</a>    <i>point de vigilance  chercher une solution?</i></p>	<ul style="list-style-type: none"> <li>▪ Moodle est une plateforme très riche en fonctionnalités, sa prise en main par les apprenants peut nécessiter un temps d'adaptation car les pages peuvent être très chargées d'informations.</li> <li>▪ Pour les enseignants, la diversité et la spécificité de tous les paramétrages des outils peuvent paraître trop complexes aux yeux d'utilisateurs peu familiers en FOAD. Cependant, de nombreux tutoriels en ligne existent, il est aussi important que l'administrateur de la plateforme ou le coordinateur du projet puisse être disponible pour aider les enseignants à appréhender la totalité des fonctionnalités de Moodle. Afin d'éviter les échecs, l'un des premiers cours à installer sur la plateforme est celui sur l'utilisation de Moodle.</li> </ul>
---	--

Tableau2.17 : Systèmes d'exploitation, navigateurs , plug-ins clients /Moodle


<p><b>Sakai</b></p> <p><a href="http://sakaiproject.org/">http://sakaiproject.org/</a></p> 	<ul style="list-style-type: none"> <li>▪ Sakai a une interface graphique très légère, basée sur des informations textuelles et non à base d'icônes. Son ergonomie est facile à appréhender tant par les enseignants que par les apprenants.</li> </ul> <p>Version</p>
--	---

Tableau2.18 : : Systèmes d'exploitation, navigateurs , plug-ins clients /Sakai

### 2.8.10 Grille récapitulative des points clés par plateforme :

















	Moodle	Sakai
1. Communauté, dynamisme, international		
2. Documentation		
3. Gestion du nombre d'utilisateurs		
4. Outils collaboratifs		
5. Adaptabilité et modularité		
6. Installation et gestion	???	???
7. Adaptation de la charte graphique		
8. Systèmes d'exploitation; navigateurs, plug-ins clients		
9. Ergonomie et utilisabilité		

Tableau2.19 : Grille récapitulative

## 2.9 Résultats :

	Moodle	Sakai
	1 😐	2 😞 3 😐
	8 😊 9 😐	7 😊

Tableau2.20 : Résultats de l'étude comparative

😊 : Evaluation positive, la plateforme répond aux critères.

😞 : Evaluation négative, problème majeur, la plateforme ne répond pas aux critères.

😐 : Point de vigilance, il faut mettre en place une solution pour pallier à ce défaut.

## 2.10 Conclusion :

Comme le montre les résultats de la présente étude, c'est la plateforme Moodle qui est paraît être la plus efficace : cette dernière est basé service, donc en résumé grâce à leur standardisation ouverte ainsi que leur fiabilité, les services web rendent les applications plus efficaces à utiliser.



## **Chapitre3 :**

### **Conception et modélisation**

### 3.1 Introduction :

Afin de modéliser notre application, nous avons choisi le langage UML (Unified Modeling Language). Cette modélisation UML comprend quatre types de diagramme :

- le diagramme des cas d'utilisations qui montre les différents acteurs de ce système et les rôles qu'ils peuvent tenir,
- le diagramme de classes (point de vue orienté objet) qui indique les relations entre les différentes classes.
- Le diagramme de séquence qui permet de cacher les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation.
- le diagramme d'activité qui va nous permettre de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables et qui sera utilisé pour décrire un flux de travail (work flow).

### 3.2 Architecture générale :

Un système Client/Serveur fonctionne selon le schéma suivant :

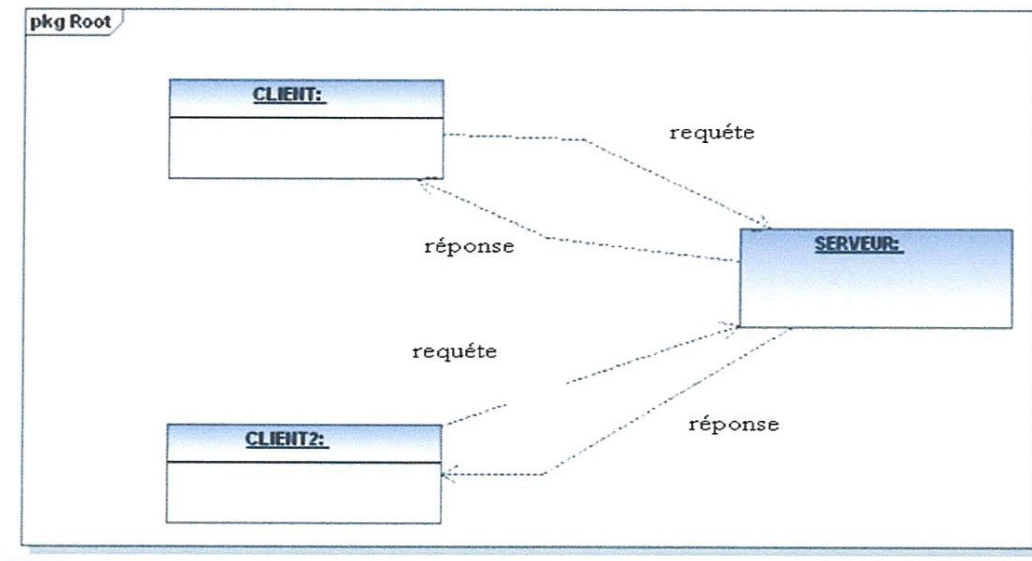


Figure 3.1 : Architecture générale

*Client / Serveur*

### 3.2.1 Description de l'architecture :

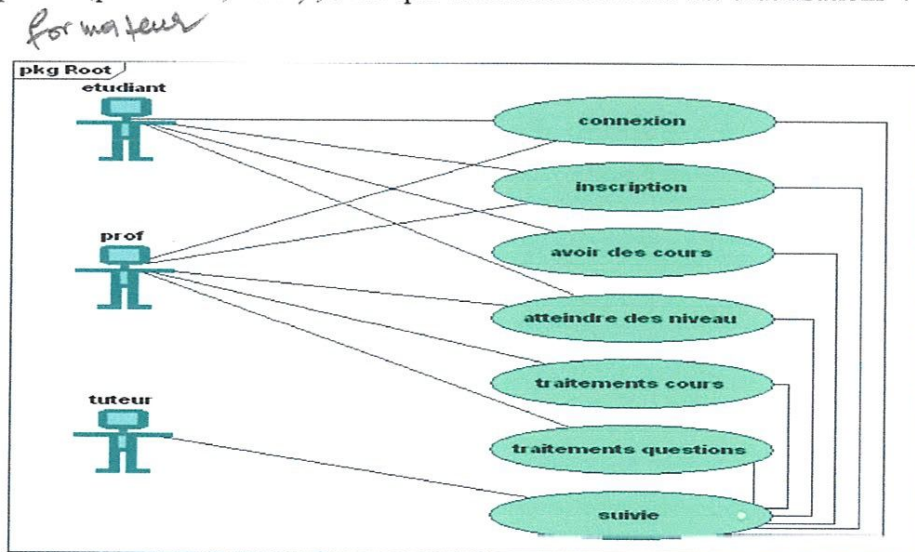
- Le Client émet une requête vers le Serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur.
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port.

Lors de la demande d'une JSP, le moteur de servlets va exécuter la classe JSP auparavant traduite et compilée et envoyer la sortie générée (typiquement, une page HTML/CSS/JS) depuis le serveur vers le client à travers le réseau, sortie qui sera alors affichée dans son navigateur.

Les messages SOAP contiennent le nom du service et les paramètres requis. La couche de transport est le protocole HTTP. Le *listener* decode le message SOAP entrant et le transforme en un appel de méthode. Ensuite, il récupère le résultat et l'encode dans un message SOAP réponse

### 3.3 Diagramme des cas d'utilisations (use case) :

Le schéma suivant montre les tâches que peut effectuer chaque acteur parmi les trois (apprenant, professeur, tuteur), ainsi que la relation entre les cas d'utilisations .



Generated by UModel  
Figure3.2 : Cas d'utilisations

www.airtova.com

## a) Cas de l'étudiant :

### Scénario connexion :

- 0- Accueil connexion.
- 1- L'étudiant introduit les données.
- 2- Le système vérifie les données.
- 3- Accès au compte.

### Scénario erreur :

- [3]- 0 → 2 le système affiche une page d'erreur.  
4- Le système suggère à la personne de créer un compte via le bouton inscription.

### Scénario inscription :

- 0- Accueil inscription.
- 1- L'étudiant introduit les données.
- 2- Le système vérifie et enregistre le nouveau profil.

### Scénario d'apprentissage :

- 0- Compte utilisateur étudiant.
- 1- L'étudiant choisit les cours.
- 2- Le système affiche les cours.

### Scénario d'évaluation :

- 0- Compte utilisateur étudiant.
- 1- L'étudiant répond aux questions.
- 2- Déconnexion.

## b) Cas du professeur :

### Scénario évaluation :

- 0- Compte professeur.
- 1- Le professeur choisit de voir les réponses.
- 2- Le système affiche les réponses.
- 3- Le professeur donne son avis.

### Scénario d'édition de questions et de MAJ:

- 0- Compte professeur.

*Administrateur*  
*- Formateur :*  
*scénario dépôt*  
*de ressources ?*



- 1- Le professeur choisit de modifier les réponses.
- 2- Le système affiche les anciennes questions.
- 3- Le professeur introduit les nouvelles questions.
- 4- Le système enregistre les modifications.

### c) Cas du tuteur :

Scénario suivi étudiant :

- 0- Compte tuteur.
- 1- Le tuteur choisit l'étudiant.
- 2- Le système affiche les informations concernant ce dernier.
- 3- Le tuteur donne son avis et/ou conseille l'étudiant à suivre d'autres cours.
- 4- Le système envoie les informations émises par le tuteur à la personne concernée.

### 3.4 Diagramme de séquence :

a) cas étudiant :

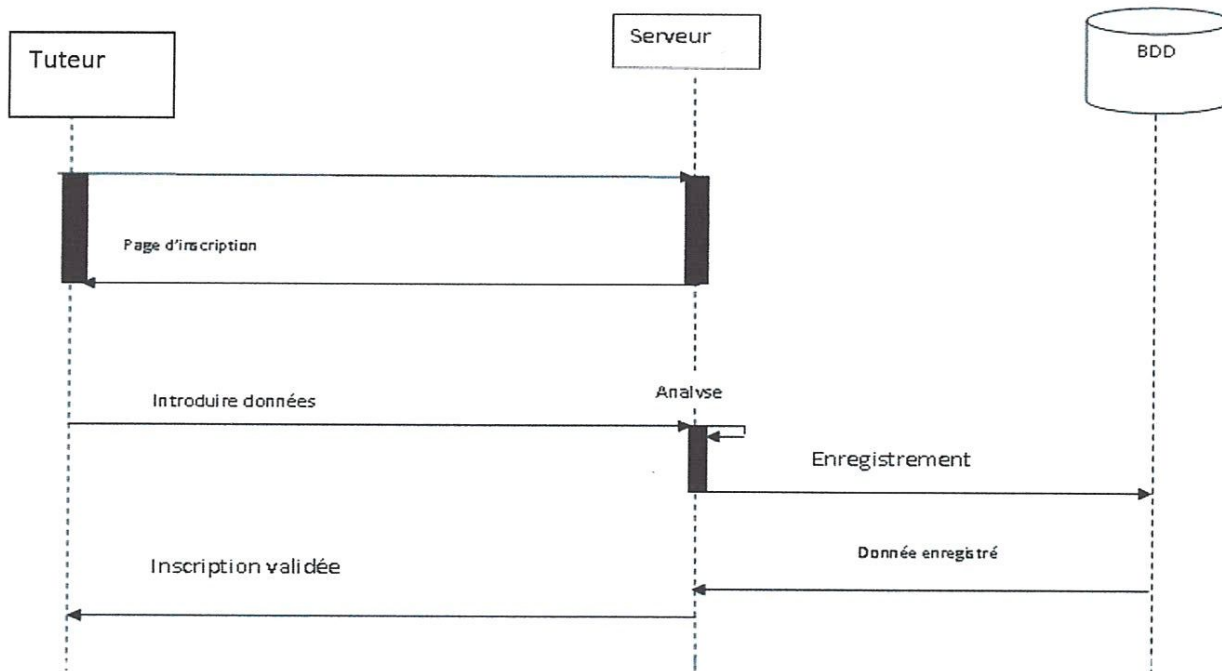


Figure3.3 : Diagramme de séquence de l'inscription et l'identification/apprenant

### Description :

L'apprenant va demander à s'inscrire, le serveur lui affichera ensuite la page d'inscription, l'apprenant saisit les données, le système vérifie leur conformité et les transmet à la BDD qui va les enregistrer, enfin le système lui affichera un message lui indiquant que son inscription a été validée.

### b) Cas professeur :

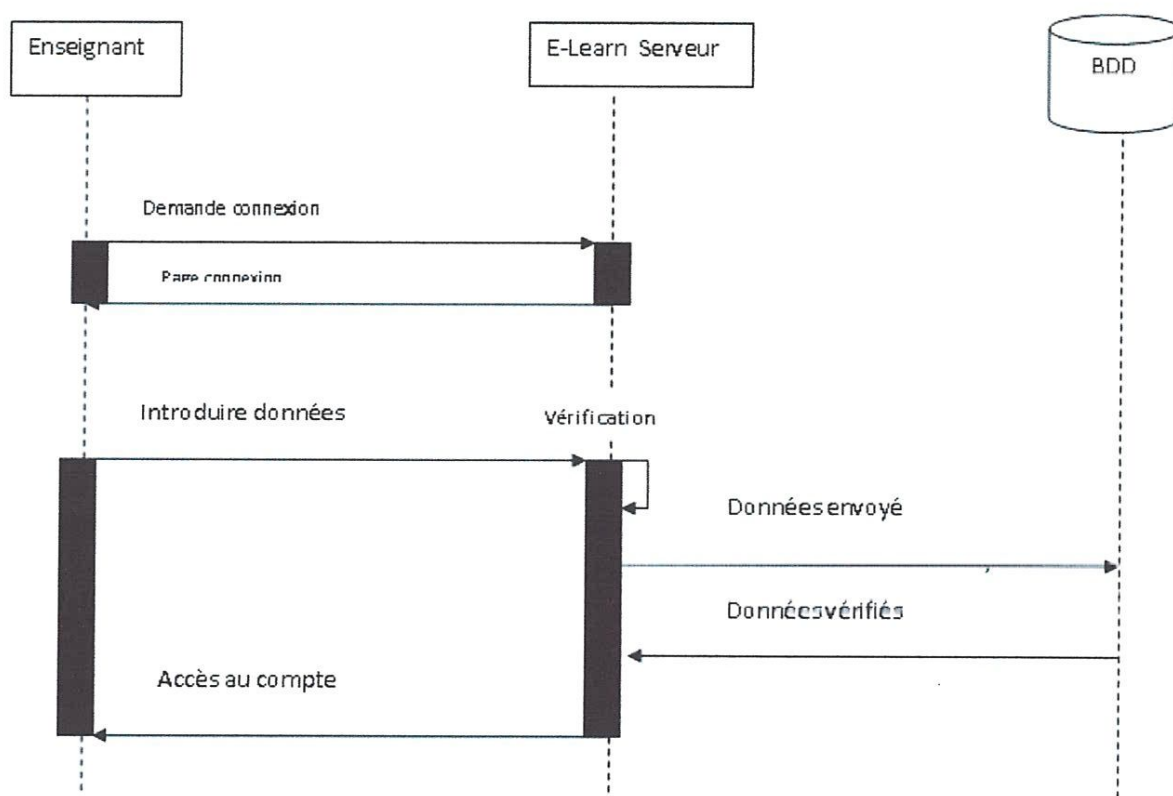


Figure 3.4 : diagramme de séquence de l'étape de connexion/Enseignant

### Description :

Le professeur demande à se connecter, le serveur va lui afficher la page d'inscription ensuite il va introduire les données. Après vérification auprès de la BDD, le serveur lui donne accès au compte.

c) Cas tuteur:

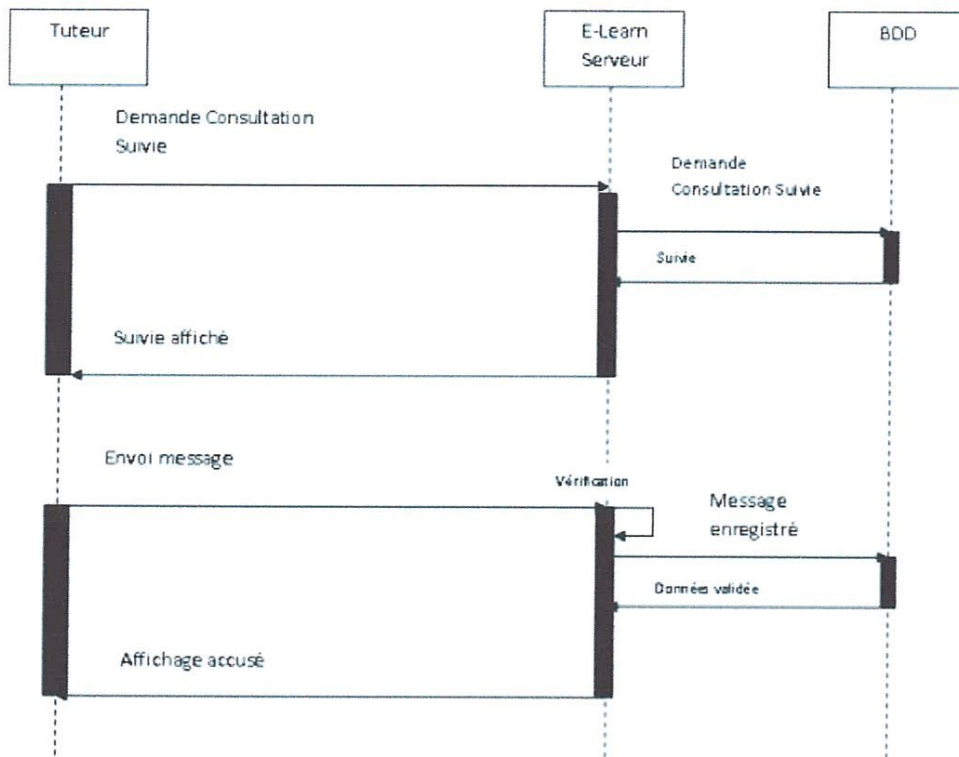


Figure 3.5 : diagramme de séquence du tuteur

Description :

Le tuteur demande au système « E-Learn » de lui afficher le suivi, ce dernier demande à récupérer les données depuis la BDD et les affiche au tuteur.

Après avoir récupéré la liste de suivi des apprenants, le tuteur va sélectionner les apprenants ayant besoin d'aide et/ou de conseil, et va ensuite leur envoyer un message. Le serveur va analyser les adresses IP des apprenants puis les données seront enregistrées dans la BDDA. Ainsi le serveur pourra les envoyer aux personnes sélectionnées, et va afficher un accusé de réception au tuteur.

### 3.5 Diagramme d'activité :

a) cas étudiant :

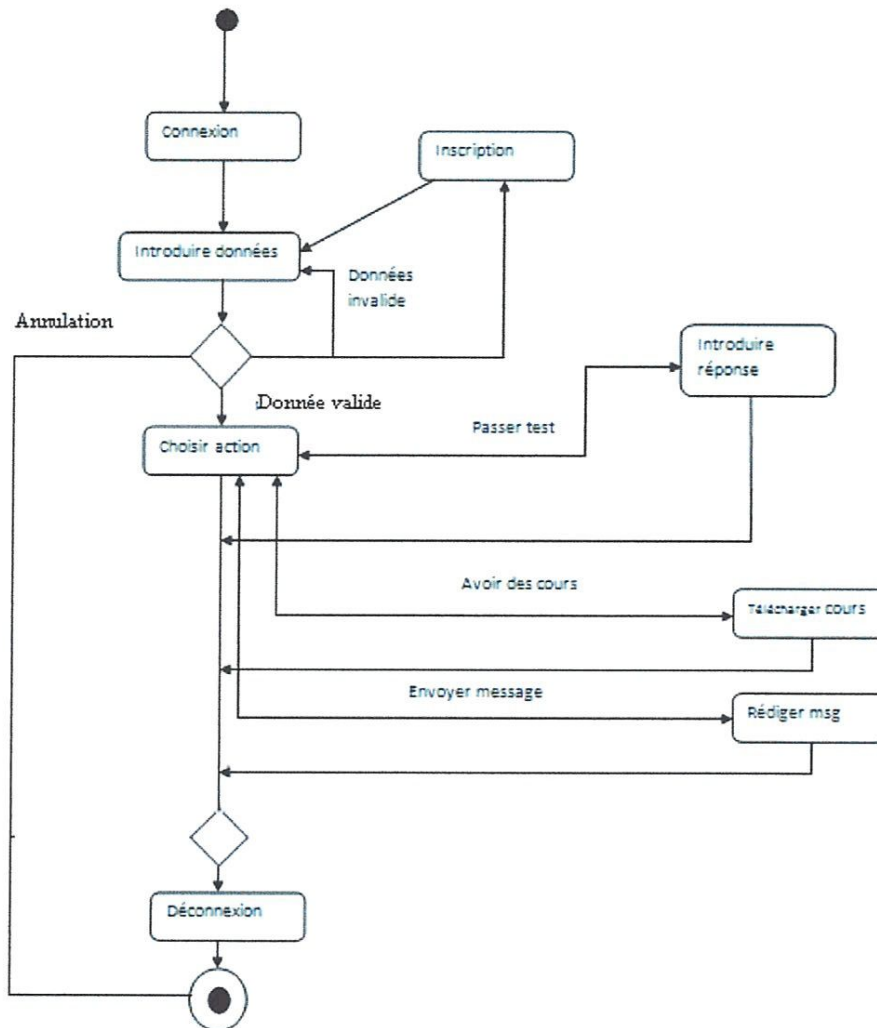


Figure 3.6 : diagramme d'activité apprenant



### Description :

Lors de la connexion de l'apprenant, une vérification va être établie concernant les données introduites

**1-Données valides :** l'apprenant aura accès au compte ainsi qu'à toutes les fonctionnalités qui lui sont proposées (envoi message, passage test, téléchargement de cours etc....) avec une possibilité de déconnexion à tout moment.

**2-données invalides :** le système va afficher une page d'erreur indiquant à l'apprenant que les données introduites ne sont pas valides et/ou lui suggérer de s'inscrire s'il n'a pas de compte.

### b) cas professeur :

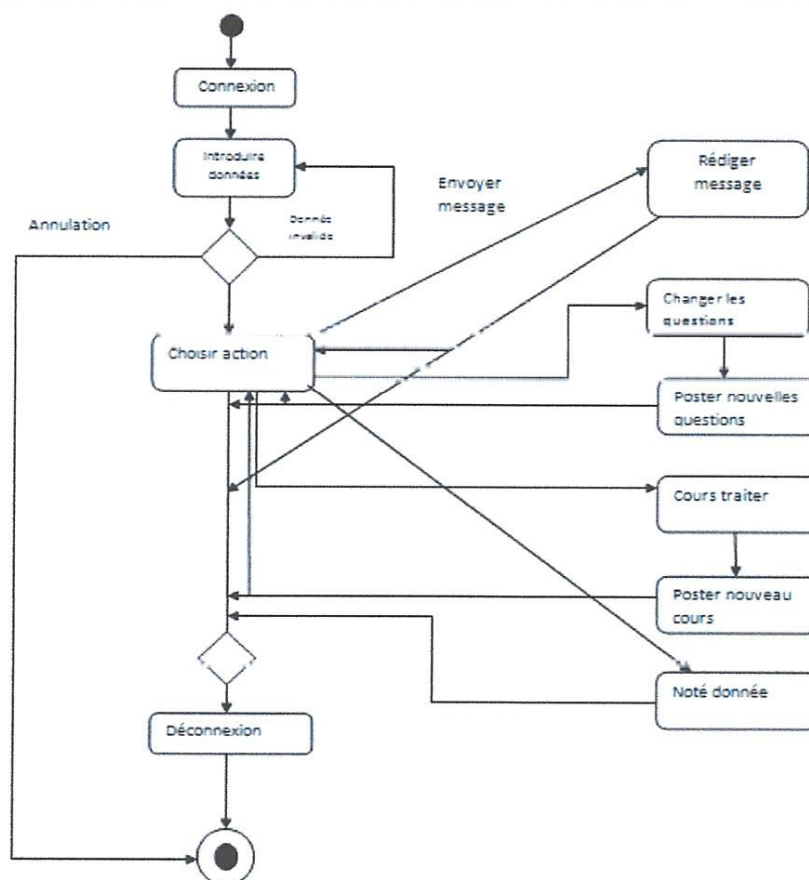


Figure3.7 : diagramme d'activité professeur

### Description :

Lors de la connexion du professeur, une vérification va être établie concernant les données introduites.

**1-données valides** : le professeur a accès à son compte et peut effectuer toutes les opérations qui lui sont fournies (noter étudiant, traitement cours, traitement questions, etc....) avec toujours une possibilité de se déconnecter à tout moment.

**2-données invalides** : le système va afficher une page d'erreur indiquant à l'apprenant que les données introduites ne sont pas valides.

### 3.6 Conclusion :

Pour conclure, la modélisation avec l'UML nous a apporté beaucoup d'avantages tels que :

- Obtenir une modélisation de très haut niveau indépendante des langages et des environnements.
- Exprimer dans un seul modèle tous les aspects statiques, dynamiques, spécifications, etc
- Documenter le projet.
- Générer automatiquement la partie logicielle du système.

## **Chapitre4**

### **Implémentation et environnement de développement**

## 4.1 Introduction :

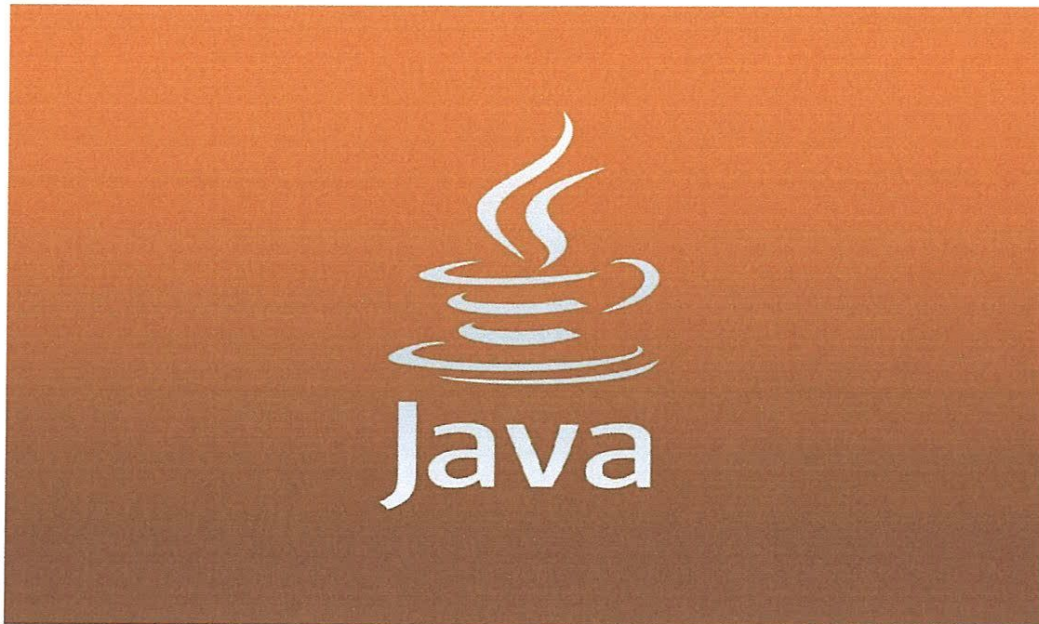
Dans ce chapitre, nous allons parler de l'environnement de développement qui est JAVA ainsi que de l'éditeur que nous avons choisi pour concevoir notre application qui est NetBeans IDE 7.1.1.

## 4.2 Langage utilisé :

Le choix du langage est dû principalement aux énormes qualités et au succès indiscutable qui le qualifient. Parmi ces avantages :

- ✚ Portabilité excellente.
- ✚ Langage puissant.
- ✚ Langage orienté objet.
- ✚ Langage de haut niveau.
- ✚ JDK très riche.
- ✚ Nombreuses bibliothèques.
- ✚ Très grande productivité.
- ✚ Applications plus sûres et stables.
- ✚ Nombreuses implémentations, JVM et compilateurs, libres ou non.
- ✚ IDE de très bonne qualité et libres : Netbeans et Eclipse par exemple.





*Figure4.1 : Logo du langage utilisé.*

La particularité principale de Java est que les logiciels écrits dans ce langage sont très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java.

Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, et l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.)

Java permet de développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

Java a donné naissance à un système d'exploitation (JavaOS), à des environnements de développement (eclipse/JDK), des machines virtuelles (MSJVM, JRE) applicatives multi plates-formes (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d'interface graphique (AWT/Swing), des applications lourdes (Jude, Oracle SQL Worksheet, etc.), des technologies web (servlets, applets) et une déclinaison pour l'entreprise (J2EE).

La portabilité du code Java est assurée par la machine virtuelle. JRE – la machine virtuelle qui effectue la traduction et l'exécution du bytecode en code natif – supporte plusieurs processus de compilation (à la volée/bytecode, natif). La portabilité est dépendante de la qualité de portage des JVM sur chaque OS.

Aujourd'hui, Java (sous la forme de JavaFx) trouve une nouvelle niche dans la création d'applications RIA (Rich Internet Applications), des applications qui proposent des fonctionnalités, notamment des interfaces, plus évoluées à la fois sur Internet et sur les téléphones portables. Le langage JavaFx est un langage agile dérivé de Java, sous le contrôle de Sun Microsystems, qui met à profit la portabilité de Java ainsi que les vastes bibliothèques déjà disponibles dans le langage Java.

#### 4.3 Editeur utilisé :

NetBeans, créé à l'initiative de Sun Microsystems (Noyau de Forte4J/SunOne), présente toutes les caractéristiques indispensables à un EDI de qualité, que ce soit pour développer en Java, Ruby, C/C++ ou même PHP.

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

L'EDI NetBeans repose sur un noyau robuste, la plateforme NetBeans, que l'on peut également utiliser pour développer ses propres applications Java, et un système de plugins performant, qui permet vraiment d'avoir un EDI sur mesure.

Ainsi, si l'on ne télécharge que la déclinaison concernant le développement Java SE, notre EDI ne sera pas livré avec le support Java EE, EJB, Jsp, Tomcat, Glassfish, ...

A coté de la version complète de l'EDI NetBeans, il existe différentes déclinaisons (6 au total, sans compter la version complète) qui se concentrent sur une plateforme ou langage bien précis :

- Java ME
- Java (SE + ME + EE)
- Ruby
- C/C++
- PHP

NetBeans rajoute, à coté du support pour CVS et SubVersion, un support pour ClearCase, mais aussi pour Mercurial.

Il permet à l'utilisateur également de déployer ses applications Web, non seulement vers Tomcat et Glassfish qui sont livrés avec le "Pack Web", mais aussi vers JBoss, WebSphere 6.1, WebLogic

En plus d'avoir :

- un système de projets basé sur Ant, permettant une meilleure indépendance et portabilité de nos travaux
- un refactoring avec des fonctionnalités comme le renommage, le déplacement et la recherche des déclarations pour classes, méthodes et packages
- le support de Java SE 5.0 (Tiger) avec les nouveaux mots clefs du langage, les annotations, les generics, etc
- le support de Java SE 6.0 (Mustang)
- le support de développement d'applications Web avec des améliorations pour l'édition des JSP, la gestion serveur, le support des dernières versions de Tomcat, etc.
- le support pour Java ME et les standards MIDP 2.0 et CLDC 1.1, possibilités d'intégrations d'émulateurs extérieurs, etc.
- un débogueur de grande qualité
- une interface graphique améliorée.

La dernière version propose, entre autres :

Un éditeur grandement amélioré quant au support Javascript, qu'il soit non seulement dans un fichier Javascript, mais aussi dans une page html, jsp, rhtml

- un éditeur visuel pour les fichiers CSS
- un support fortement amélioré pour tout ce qui est Ajax, avec entre autre un support pour jMaki et GWT
- un nouveau module pour Java FX
- la gestion de bases de données grandement revue, avec un support direct pour JavaDB/Derby, MySql, PostgreSQL
- un support pour les frameworks Hibernate et Spring
- un support pour les webservices, avec support RestFul, Axis2, plugin SoapUI
- une fonctionnalité "deploy on save" pour les applications Web
- un support de GlassFish V3 Prelude
- des éditeurs pour les langages Groovy, Grails, Ruby, Rails, Python, C/C++, PHP, etc.
- un éditeur xml, xsd, xslt, wsd, ...
- un support pour importer des projets Eclipse.



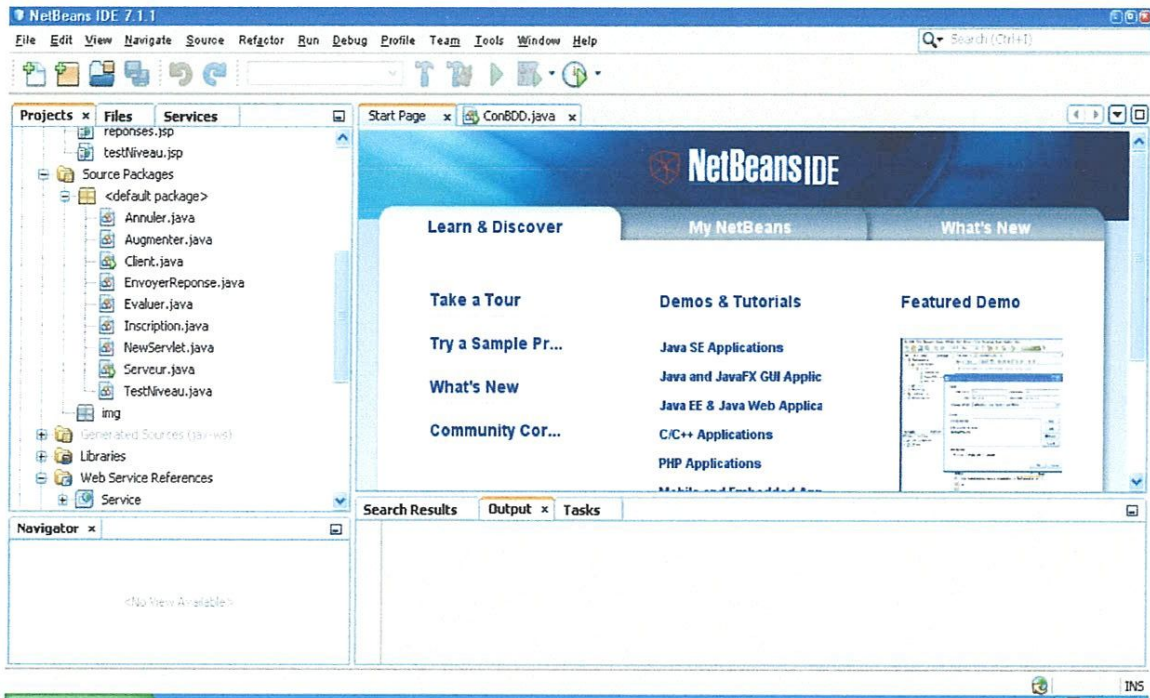


Figure 4.2 : L'interface de l'éditeur utilisé.



#### 4.4 Implémentation :

Langue ?

- Page d'accueil :

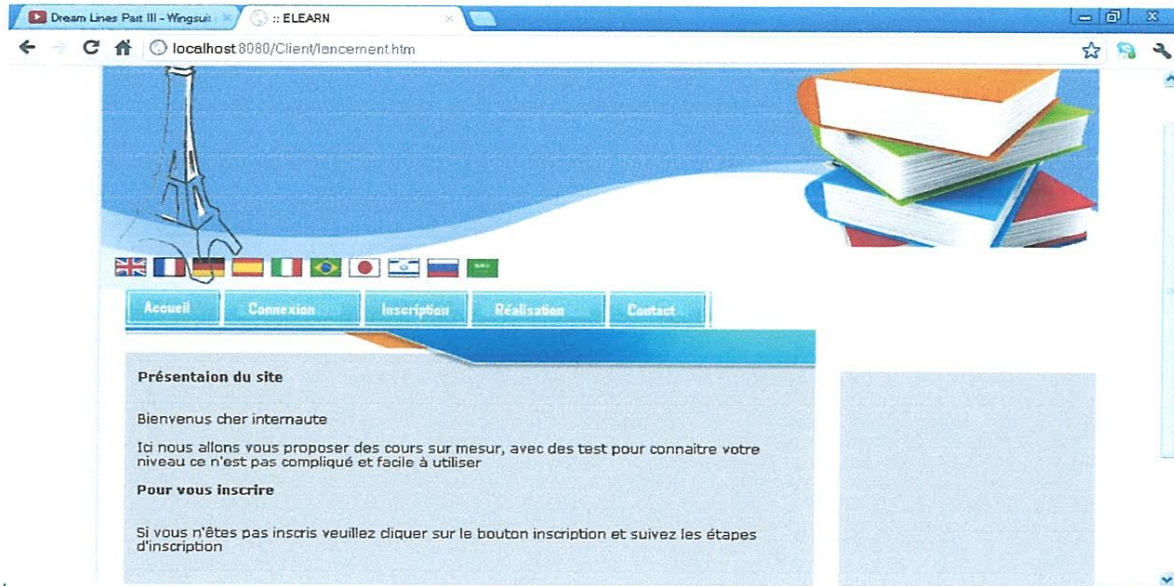


Figure 4.3 : page d'accueil E-LEARN

Cette figure représente la page d'accueil de notre système E-LEARN.

- Page d'inscription :

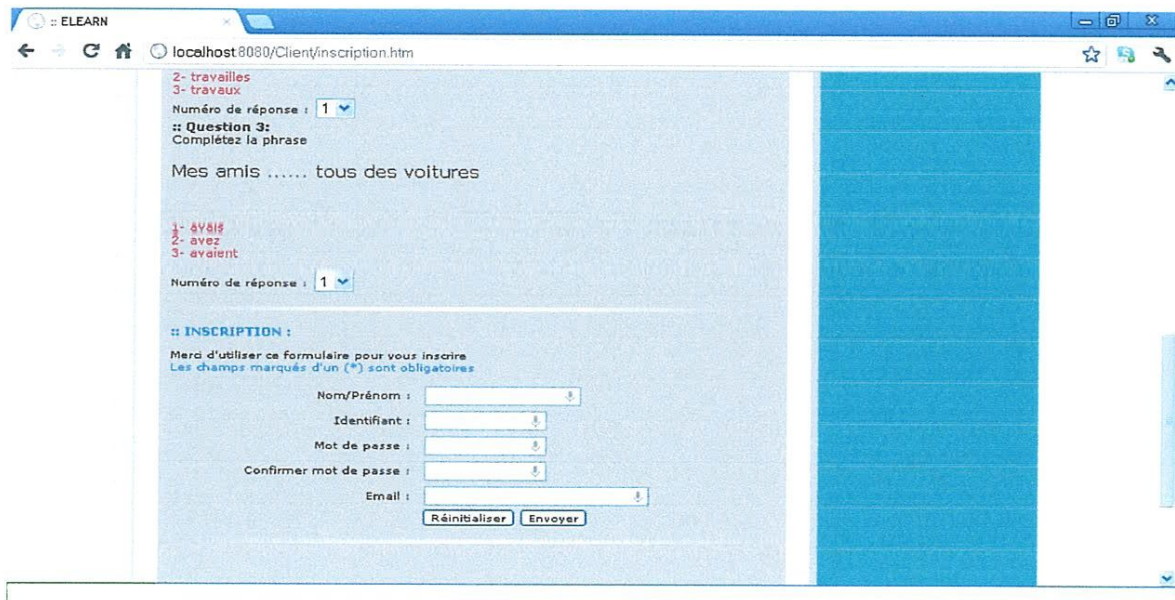
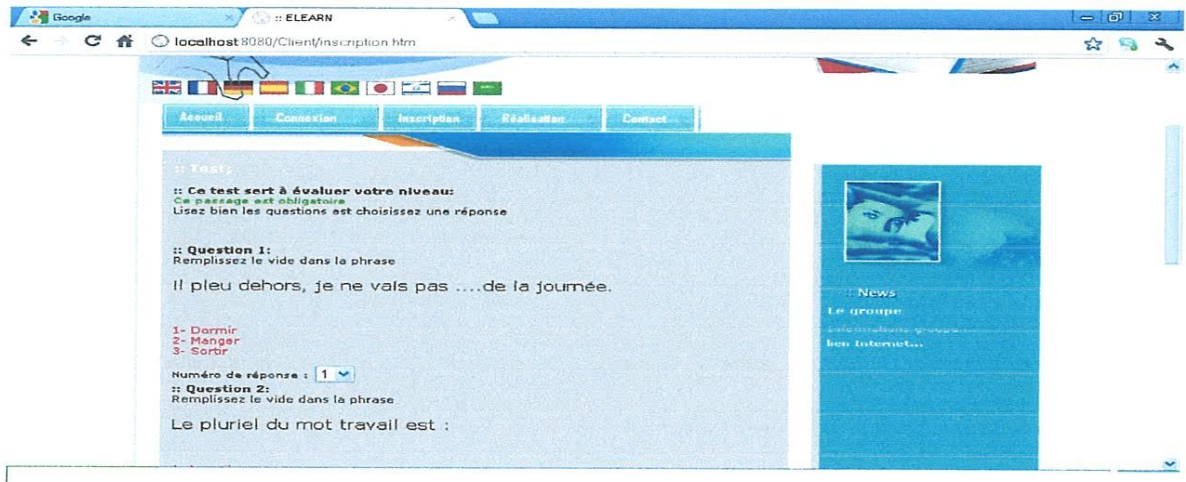


Figure4.4 : page d'inscription apprenant

Page d'inscription par laquelle l'apprenant peut s'inscrire après avoir répondu aux questions, et rempli les données personnelles.

- Page connexion :

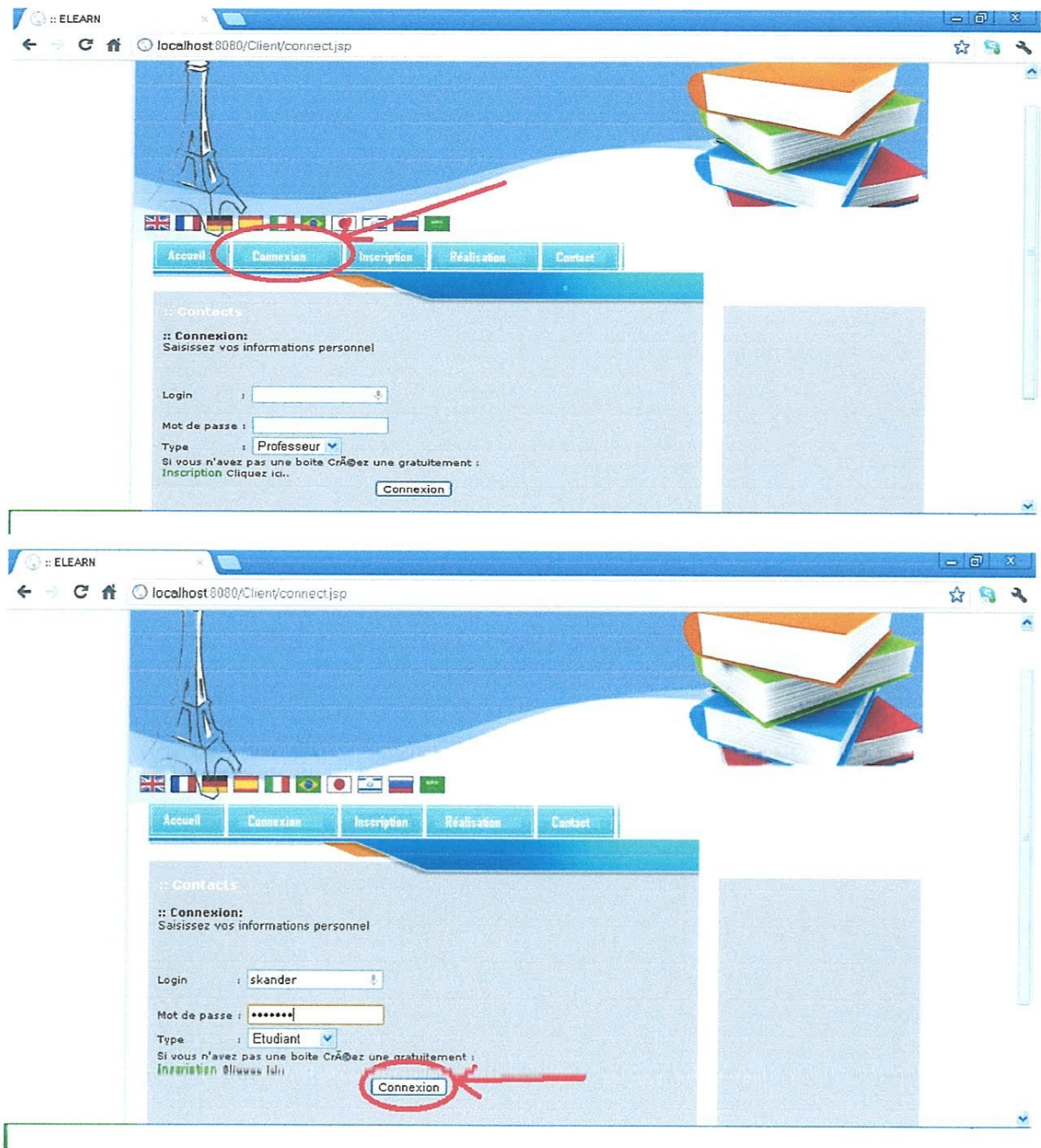
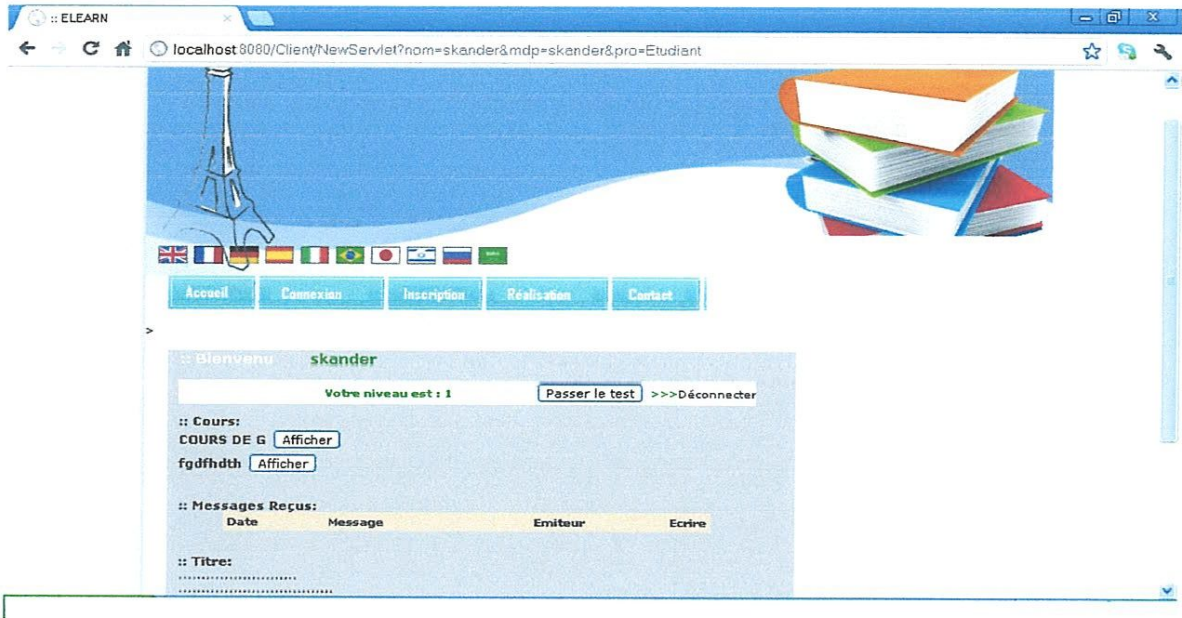


Figure4.5 : page de connexion

Page de connexion (introduction du login et du password ainsi que le choix de l'utilisateur)



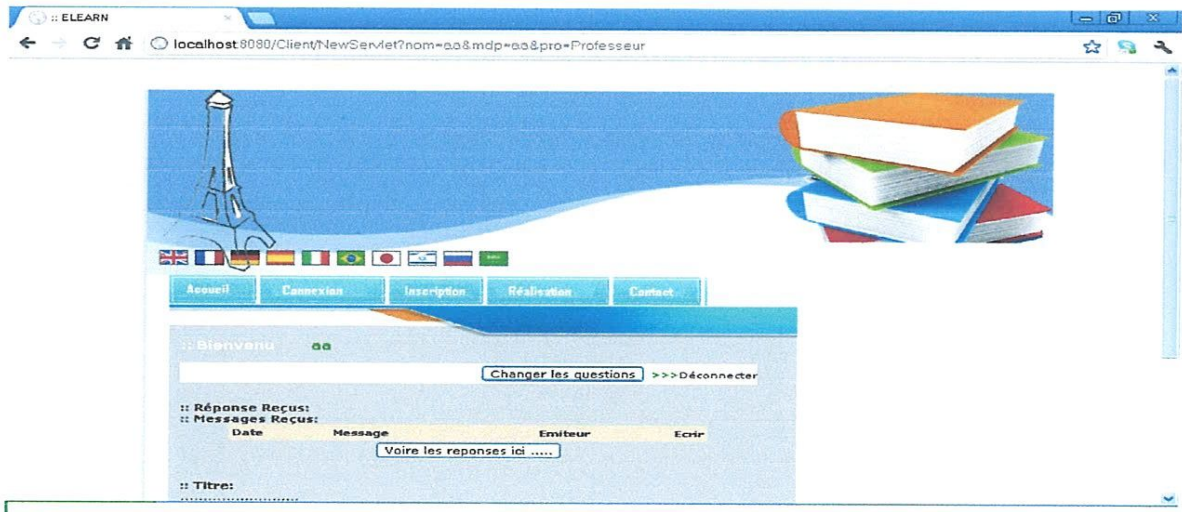
- **Interface compte apprenant :**



*Figure 4.6 : interface compte apprenant*

L'apprenant peut passer le test ainsi que télécharger des cours.

- **Interface professeur :**

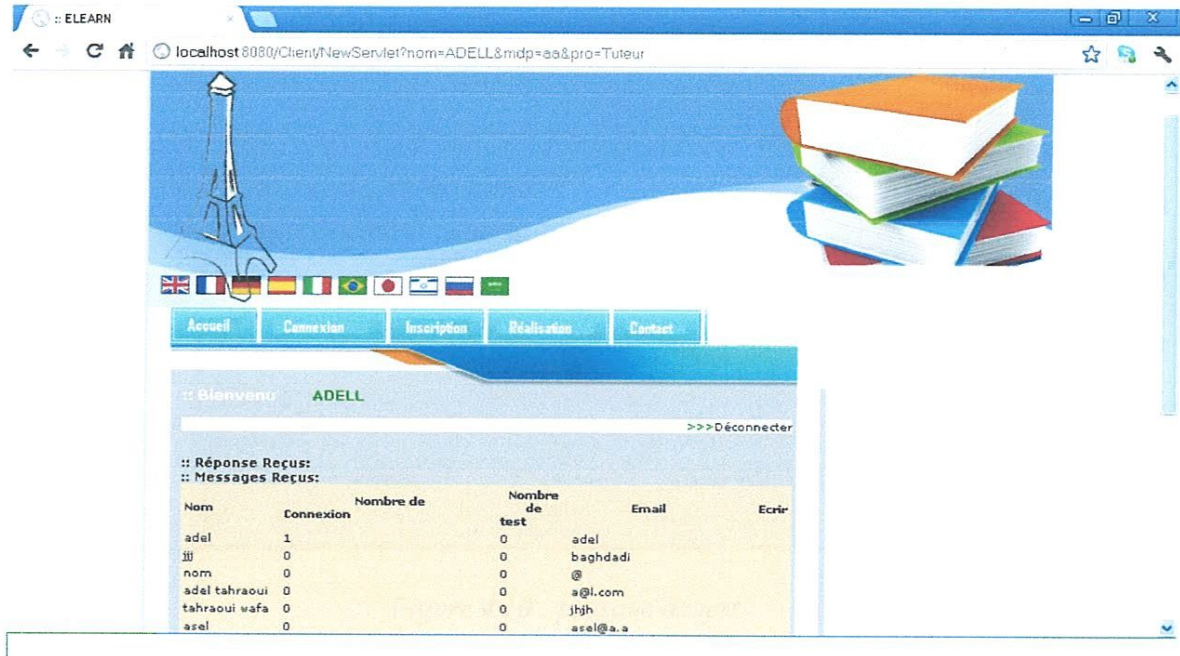


*Figure 4.7 : interface professeur*

Le professeur peut voir les réponses des étudiants, ainsi que la possibilité de traiter les questions.



- **Interface tuteur :**



*Figure 4.8 : interface compte tuteur*

Le tuteur a devant lui tous les renseignements concernant les apprenants, la fréquence de leur passage sur le site, et aussi le passage de tests.

- **Page d'erreur :**



*Figure 4.9 : page d'erreur*

La page d'erreur qui s'affiche lors de l'introduction de données invalides concernant les utilisateurs.

## Bibliographie :

- [MAR 04] Marc Walckiers- Thomas De Praetere « Distance et Savoir, CNED & Hermès-Lavoisier », Université catholique de Lovain , Institut de pédagogie et des multimédias, Belgique, Décembre 2004
- [ANN 07] Anne Durand « Etude comparative de plates-formes de formation à distance » Laboratoire MAP- CNRS , Marseille, Octobre 2007
- [HEL 07] Héléne Vanderstichel « projet 02L » Chargée de développement pour la Mission Locale de Villeneuve d'Ascq
- [CHR 03] Christophe Jeunesse, C. D. (2003). Une pédagogie pour susciter l'apprentissage collaboratif. p. 51.
- [BAR 08] Baron, É. B.-L. (2008, SEPTEMBRE 28). Travail et apprentissage collaboratifs dans l'enseignement supérieur :
- [KOU] B. Kouninef, M. Djelti, S.M. Rerbal Institut des télécommunications d'Oran,
- [MAR 07] Marie Leproust, (Apprentissage sur les logiciels libres) Responsable du pôle TICE de l'Université de Versailles Saint-Quentin-en-Yvelines
- [1] <http://www.siteduzero.com/tutoriel-3-203276-les-services-web.html>
- [2] <http://apiacoa.org/publications/teaching/webservices/Introduction-2bypage.pdf>
- [3] <http://msdn.microsoft.com/fr-fr/library/aa480728.aspx>
- [4] [http://fr.wikipedia.org/wiki/Service\\_Web](http://fr.wikipedia.org/wiki/Service_Web)