République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DU **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**
Filière : **Informatique**
Spécialité : **Systèmes Informatique**

Présenté par
Prénom :Sabrina NOM :Larbi
Prénom :Assia NOM :Ibersiene

Thème

Web Services et Interopérabilité des Applications Cas :E-Learning

Mémoire soutenu publiquement le 21 ……/09……/ 2016….. devant le jury composé de :

Président : Mr Prénom :Ahmed NOM :Dib
Encadreur : Mr Prénom :M'hand NOM :Kerbiche
Examinateur : Mr Prénom :Mouhand Said NOM :Habet
Examinateur : Mm Prénom :Tounsia NOM :Djemah

Remerciements

*Nous remercions le bon **Dieu** de nous avoir donné le courage et la force d'accomplir ce travail.*

***Mr KERBICHE M'hand**, notre promoteur, pour ses conseils, sa disponibilité et son encouragement qui nous ont permis de réaliser ce travail dans les meilleures conditions.*

*Les **membres de jury** de l'honneur qu'ils nous accordent en acceptant de juger notre travail.*

- *Mr Dib Ahmed ,*
- *Mr Habet Mouhend Said,*
- *Mm Djemah Tounsia.*

*Aux **enseignants** de notre université et du département d'informatique.*

*A **toute personne**, d'une manière ou d'une autre, qui nous a aidés et encouragés à l'aboutissement de ce travail, trouve ici l'expression de nos sincères reconnaissances.*

Dédicaces

*A **ma famille** source de ma volonté et de ma détermination.*

*A mes très **chers parents** (ma douce **Mère** et mon **Père** adoré), qui ont été à la hauteur avec leurs compréhensions et leurs incontestables encouragements pendant tout mon cursus,*

*Mes très chères **Sœurs** et **Frères** et **Beaux Frères***

*A mes adorés **MIMI** et **YOUCEF***

*Mon cher binôme **Assia** ainsi qu'à toute sa famille.*

*A tous **mes Amis***

Enfin à tous ceux qui ont contribué de loin ou de prés

*A **tous ceux**-là, je dédie ce travail*

Sabrina

Dédicaces

Je dédie ce modeste travail :

A mes très chers ***parents***

qui m'ont beaucoup aidé et soutenu durant ma vie et surtout dans mes études et leur compréhension.

Je prie dieu le tout puissant de les protéger du mal et les récompenser de toutes les peines et sacrifices données aux quels je ne rendrai jamais assez.

*A mes frères :**HAMID, FARID,FODIL,***

*A mon neveu :**ELYAS** ,et ma nièce :**WISSAME,***

A toute ma famille sans exception,

*A mon binôme:**SABRINA,***

A tous mes ***amis*** *(es),*

A toute ***personne*** *qui me connaît,*

A tous les ***étudiants SI*** *promotion2015 /2016*

ASSIA

Sommaire

Chapitre 1: Interopérabilité et Services Web

Chapitre 2: Présentation d'E-Learning

Chapitre 3: Analyse et Conception

Chapitre 4 : Réalisation

Listes Des Figures

Liste des *Abréviations*

API	Application Programming Interface
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model
http	HyperText Transfert Protocol
Java RMI	JAVA Remote Method Invocation
RUP	Rational UnifiedProcess
SGC	Système de Gestion de Cours
SOA	Service Oriented architecture
SOAP	Simple Object Access Protocol
SW	Service Web
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
URL	Uniform Resource Locator
W3C	World Wide Web consortium
WS-I	*Web Services* Interoperability
WSDL	Web Service Description Language
XML	eXtansible Markup Language
XSD	*XML Schema Definition*

Introduction Générale

Au cours des dernières années, la nécessité de connecter les personnes, les informations et les processus a changé la façon dont les logiciels sont développés. Les systèmes informatiques performants ont de plus en plus besoin de l'interopérabilité entre les plates-formes . Ceci a conduit à la prédominance de XML en tant que langage universel pour la représentation et la transmission de données structurées indépendantes du langage de programmation, de la plate-forme logicielle et du matériel.

Grâce à la large adoption du langage XML, les services Web sont des applications qui utilisent des transports, encodages et protocoles standard pour échanger les informations. Grâce à une prise en charge étendue par les fournisseurs et les entreprises, les services Web permettent aux systèmes informatiques de n'importe quelle plate-forme de communiquer via les intranets d'entreprise, les extranets et Internet, avec la prise en charge de la sécurité de bout en bout, de la messagerie fiable et des transactions distribuées, et plus encore.

Les services Web sont basés sur un ensemble de normes qui décrivent la syntaxe et la sémantique des communications logicielles : XML fournit la syntaxe commune pour la représentation des données, le protocole SOAP (Simple Object Access Protocol) fournit la sémantique pour l'échange de données, et le langage WSDL (Web Services Description Language) fournit un mécanisme permettant de décrire les possibilités d'un service Web. Des spécifications complémentaires, collectivement appelées architecture WS-*, définissent les fonctionnalités pour la découverte des services Web, la gestion des événements, les pièces jointes, la sécurité, la messagerie fiable, les transactions et la gestion.

Par l'utilisation de standards, les services Web révolutionne la façon dont les applications communiquent entre elles. Ceci parce qu'ils permettent un accès universel, à partir de n'importe quel ordinateur ou appareil, indépendamment de sa technologie propriétaire de base. Nous disposons avant tout d'un langage universel : celui des services Web. Les sites Web qui savaient présenter des informations peuvent à présent devenir des applications Web capables de communiquer entre elles. En réalité, toute application peut communiquer avec n'importe quelle autre application, du moment qu'elle implémente les services Web nécessaires.

Notre travail consiste à réaliser une plate-forme interopérable pour l'apprentissage à distance E-Learning basée sur les Web Services.

Pour atteindre cet objectif, nous avons organisé ce mémoire comme suit :

- **Chapitre1 «Interopérabilité et Services Web»** :Ce chapitre est consacré à l'étude des services Web et leur composition pour mieux comprendre les concepts de base de cette technologie.

- **Chapitre2 « E-Learning »** :ce chapitre est consacré à la présentation de certaines définitions nécessaires dans le domaine d'apprentissage électronique.
- **Chapitre3 « Analyse et Conception »** :présente l'analyse et la conception de notre solution, ou nous avons utilisé le langage UML(le processus RUP) comme outil de modélisation.
- **Chapitre4« Réalisation »** :ce dernier chapitre est consacré à la présentation des outils de développement utilisés, ainsi que la présentation de quelque interfaces de l'application réalisée qui nous ont servi pour la réalisation de ce projet.

Enfin, nous terminerons ce document par une **conclusion générale**.

I .Introduction

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

Au cœur de cette vision se trouve le concept d'interopérabilité, c'est-à-dire la capacité pour des systèmes disparates de communiquer et de partager des données de façon transparente. C'est l'objectif des services Web.

Un service Web est une logique d'application programmable accessible à l'aide des protocoles Internet standard, que l'on peut aussi décrire comme l'implémentation de standards Web pour une communication transparente entre les machines et entre les applications.

Pour comprendre le fonctionnement des services Web, ce chapitre commence par donner les définitions des services web, leurs architectures et nous décrivons les différentes technologies liées aux services web.

I.1 Historique [Monfort, 2003]

Le World Wide Web on s'accroit de manière exponentielle depuis les années suivantes :

- **En 1990** : Ce phénomène est dû en partie aux entreprise qui ont vu dans l'Internet l'outil par excellence pour ce faire connaitre. Mais également aux organismes ayant vocation à fournir un service publique d'information, de proximité ou non.
- **En 1993** : les entreprises et les états ont exploité l'Internet et en particulier les serveurs Web pour faire le commerce électronique ou bien les gouvernements électroniques.
- **En 1998** : les services Web prennent leur origine dans l'informatique distribuée et dans l'événement de Web.

I.2 Définition

Plusieurs définitions des services Web ont été mises en avant par différents auteurs, nous citons quelques définitions généralement acceptées et fournies :

Définition d'IBM [IBM,00]

« Les services web sont la nouvelle vague des applications web. Ce sont des applications modulaires, auto-contenues et auto-descriptives qui peuvent être publiées, localisées et invoquées depuis le web. Les services web effectuent des actions allant de simples requêtes à des processus métiers complexes. Une fois qu'un service web est déployé, d'autres applications (y compris des services web) peuvent le découvrir et l'invoquer »

Définition de W3C [W3C,04]

« Un service web est un système logiciel identifié par un URI dont les interfaces publiques et les incarnations sont définies et décrites en XML. Sa définition peut être découverte dynamiquement par d'autres systèmes logiciels. Ces derniers peuvent ensuite interagir avec le service web d'une façon décrite par sa définition, en utilisant des messages XML transportés par des protocoles Internet ».

I.3 Interopérabilité des services web

L'interopérabilité est la capacité des systèmes à communiquer, à échanger des données, à « travailler » ensemble, sans que l'utilisateur ait besoin de connaître les caractéristiques spécifiques à chaque système.

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

Chaque Web service doit pouvoir être découvert et invoqué dynamiquement par les applications.
L'architecture des services Web s'est imposée grâce à sa simplicité, à sa lisibilité et à ses fondations normalisées. L'objectif principal des services Web est de faciliter le plus possible l'accès aux applications entre entités et ainsi de simplifier les échanges de données.

Cette interopérabilité est due à l'utilisation de normes ouvertes. L'**OSI** et le **W3C** sont les comités de coordination responsables de l'architecture et de standardisation des services Web.
Pour améliorer l'interopérabilité entre les réalisations de service Web, l'organisation **WS-I** a développé une série de profils pour faire évoluer les futures normes impliquées. L'aspect le plus important des Web Services est qu'ils reposent sur plusieurs standards qui permettent la structuration des architectures.

I.4 Les technologies des services 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.

- **Le protocole HTTP**

HTTP (hypertext transfert protocol) est un protocole de niveau applicationqui contrôle le transport de messages sur le web via le port 80 du protocoleTCP/IP. La communication http est synchrone et se compose d'une série de messageséchangés entre un client et un serveur.

- **REST**

REST (Representational State Transfer) est un style d'architecture qui repose sur le protocole HTTP ,REST est une manière de construire une application pour les systèmes distribués comme le World Wide Web.

- **XML-RPC**

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

- **SOAP**

SOAP (Simple object Access Protocol) est un protocole standard de communication. 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.

- **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.

- **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.

I.5 Architecture des services Web

L'interopérabilité –aptitude de deux ou plusieurs systèmes à fonctionner ensemble en utilisant des standards communs– est l'objectif premier des services web. Pour permettre cet échange d'information entre des applications distantes, indépendamment des systèmes d'exploitation et des langages de programmation, les services web ce base sur **l'architecture orienté service(SOA).**

I.5.1 Définition SOA

Une architecture orientée services (Services Oriented Architecture) est une architecture logicielle s'appuyant sur un ensemble de services simples. Ces services communiquent les uns avec les autres.
Les architectures SOA offrent un potentiel prometteur et sont d'influence grandissante dans le cadre des architectures logicielles.
L'objectif d'une architecture orientée services est donc de décomposer une fonctionnalité en un ensemble de fonctions basiques, appelées services, fournies par des composants et de décrire finement le schéma d'interaction entre ces services.

I.5.2 Acteurs de SOA

- **Fournisseur de service:**

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

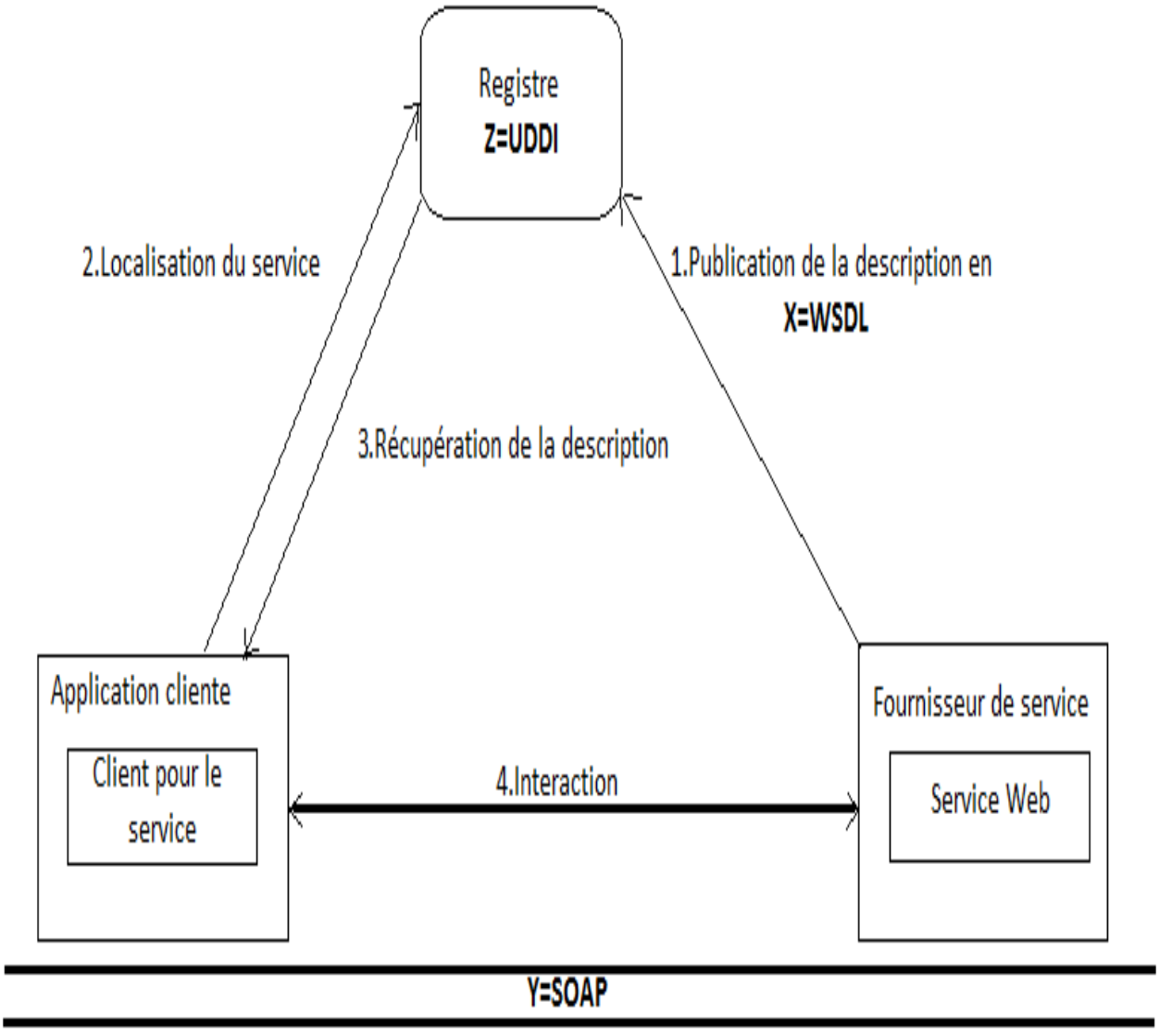
- **Demandeur de service :**

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).

- **Distributeur de service:**

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

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



FigureI.2 :Architecture des services web de type SOAP

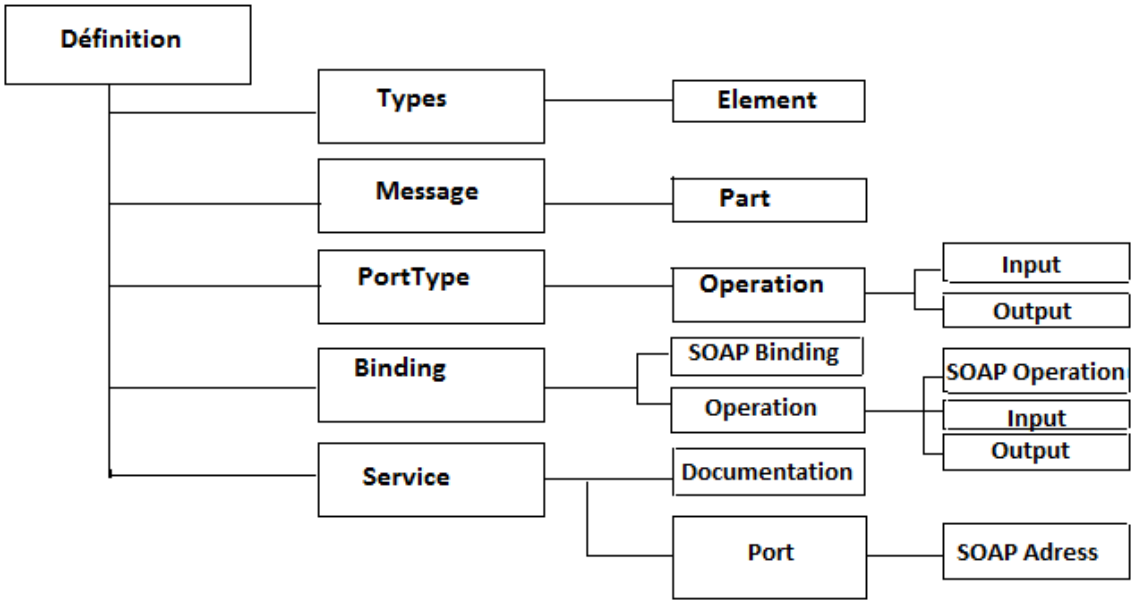
I.6 Le langage de description d'interface WSDL

I.6.1 Définition

- WSDL (**Web Services Description Language**) est une grammaire XML permettant de décrire un service web,
- WSDL a été proposé en 2001 au W3C pour standardisation,
- WSDL décrit qu'est-ce que le service Web, où le trouver et comment l'appeler,
- WSDL utilise le concept de port pour définir la connexion au service Web,
- Il représente la définition d'un service Web vue par le fournisseur,
- Il doit contenir toutes l'information nécessaire au client pour consommer le service,

  C'est quoi un service selon WSDL
  - Un ensemble d'opérations,
  - Des formats des messages Typé nécessaire à chaque opération.

I.6.2 Structure d'un document WSDL



FigureI.3 **: Structure d'un Document WSDL**

- **<definitions> :** c'est l'élément racine de tous les documents WSDL. Elle définit le nom du service Web.Cette balise peut contenir les attributs précisant le nom du service, et les espaces de nommage.

- **<Type>** : élément qui prend en charge la définition des types de données qui sont utilisées par un service. Les types sont des documents XML, ou parties de documents.
  - L'élément de type décrit tous les types de données utilisés entre le client et le serveur.
  - WSDL est pas liée exclusivement à un système de typage spécifique.
  - WSDL utilise la spécification W3C XML Schéma comme choix par défaut pour définir les types de données.

- **<Message> :** élément décrit les données échangées entre le service Web les fournisseurs et les consommateurs.
  - Chaque service Web a deux messages: entrée et sortie.
  - L'entrée décrit les paramètres pour le service Web et la sortie décrit le retour des données du service Web.
  - Chaque message contient zéro ou plusieurs **<part>** paramètres, un pour chaque paramètre de la fonction de service Web.

- Chaque **<part>** paramètres associés à un type concret défini dans le **<types>** container élément.

Exemple

```
<message name="Say Hello Request">
<part name="first Name" type="xsd:string"/>
</message>

<message name="Say Hello Response">
<part name="greeting" type="xsd:string"/>
</message>
```

Ici, deux éléments de message sont définis. Le premier représente un message de demande Say Hello Request et le second représente un message de réponse Say Hello Response.

- **<port Type> :** élément qui combine plusieurs éléments de message.

- Par exemple, un <portType> peut combiner une requête et un message de réponse en une seule opération demande / réponse.
- Un PortType peut définir de multiples opérations.

Exemple

```
<portType name="Hello_PortType">
<operation name="sayHello">
<input message="tns:SayHelloRequest"/>
<output message="tns:SayHelloResponse"/>
</operation>
</portType>
```

L'élément port Type définit une seule opération, appelée say Hello.
L'opération consiste en un message d'entrée unique Say Hello Request et un message de sortie Say Hello Response.

- **<opérations > :**c'est la description d'une action exposée dans le port

WSDL supporte quatre modes de fonctionnement de base:

1. One-way

Le service reçoit un message mais le service ne répond pas. L'opération a donc un élément d'entrée unique.

La grammaire pour une opération à sens unique est:

```
<wsdl:definitions .... >

<wsdl:portType .... >

<wsdl:operation name="nmtoken">

<wsdl:input name="nmtoken"? message="qname"/>

</wsdl:operation>

</wsdl:portType>

</wsdl:definitions>
```

2. Demande-Réponse

Le service reçoit un message et envoie une réponse. L'opération a donc une entrée élément, suivi d'un élément de sortie. Pour encapsuler des erreurs, un élément de défaut en option peut également être spécifié. La grammaire pour une opération de demande-réponse est:

```
<wsdl:definitions .... >
<wsdl:portType .... >
<wsdl:operation name="nmtoken" parameter Order="nmtokens">
<wsdl:input name="nmtoken"? message="qname"/>
<wsdl:output name="nmtoken"? message="qname"/>
<wsdl:fault name="nmtoken" message="qname"/>
</wsdl:operation>
</wsdl:portType>
</wsdl:definitions>
```

3. Sollicitation-réponse

Le service envoie un message et reçoit une réponse. L'opération a donc une sortie élément, suivi par un élément d'entrée. Pour encapsuler des erreurs, un élément de défaut en option peut aussi être spécifié. La grammaire pour une opération de sollicitation-réponse est :

```
<wsdl:definitions .... >
<wsdl:portType .... >
<wsdl:operation name="nmtoken" parameterOrder="nmtokens">
<wsdl:output name="nmtoken"? message="qname"/>
<wsdl:input name="nmtoken"? message="qname"/>
<wsdl:fault name="nmtoken" message="qname"/>
</wsdl:operation></wsdl:portType></wsdl:definitions>
```

4. Notification

Le service envoie un message. L'opération a donc un élément de sortie. Lagrammaire pour une opération de notification:

```
<wsdl:definitions .... >
<wsdl:portType .... >
<wsdl:operation name="nmtoken">
<wsdl:output name="nmtoken"? message="qname"/>
</wsdl:operation>
</wsdl:portType>
</wsdl:definitions>
```

- **<Binding> :** élément qui fournit des détails précis sur la façon dont une opération de portType sera effectivement transmise sur le fil.

  - Les liaisons peuvent être mis à disposition par différents moyens de transport, y compris HTTP GET, HTTP POST ou SOAP.

- Les liaisons fournissent des informations concrètes sur ce protocole est utilisé pour transférer les opérations *de port Type.*
- Les liaisons fournissent des informations où se trouve le service.
- Pour le protocole SOAP, la liaison est **<soap: binding>,**et le transport est des messages SOAP au - dessus du protocole HTTP.
- Vous pouvez spécifier plusieurs liaisons pour un seul *port Type.*

L'élément de liaison a deux attributs: le *nom* et le *type* attribut.

```
<binding name="Hello_Binding" type="tns:Hello_PortType">
```

❖ L'attribut name définit le nom de la liaison, et les points de type d'attribut au port pourla liaison, dans ce cas , le **« tns: Hello_PortType** » est le port.

- **<Port > :** définit un critère individuel en spécifiant une adresse unique pour une liaison.

Voici la grammaire pour spécifier un port:

```
<wsdl:definitions .... >
<wsdl:service .... >
<wsdl:port name="nmtoken" binding="qname"> *
<-- extensibility element (1) -->
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

- L'élément du port a deux attributs: *name* et *bin*ding.
- L'attribut name fournit un nom unique parmi tous les ports définis dans le document WSDL.
- L'attribut de liaison se réfère à la liaison en utilisant les règles de liaison définies par WSDL
- Éléments d'extensibilité de liaison sont utilisés pour spécifier les informations d'adresse pour le port.
- Un port NE DOIT PAS spécifier plus d'une adresse.
- Un port NE DOIT PAS spécifier les informations de liaison autres que les informations d'adresse.

- **<Service > :** définit les ports pris en charge par le service de WEB. Pour chacun des protocoles pris en charge, il y a un élément de port. L'élément de service est un ensemble de ports.

  - clients de services Web peuvent apprendre les éléments suivants de l'élément de service:

    où pour accéder au service,

    - par quel port pour accéder au service Web, et
    - la manière dont les messages de communication sont définis.
  - L'élément de service comprend un élément de documentation pour fournir de la documentation lisible par l'homme.

Exemple:

```
<service name="Hello_Service">
<documentation>WSDL File for HelloService</documentation>
<port binding="tns:Hello_Binding" name="Hello_Port">
<soap:address
      location="http://www.examples.com/SayHello/">
</port>
</service>
```

I.7 Protocol SOAP

I.7.1. définition

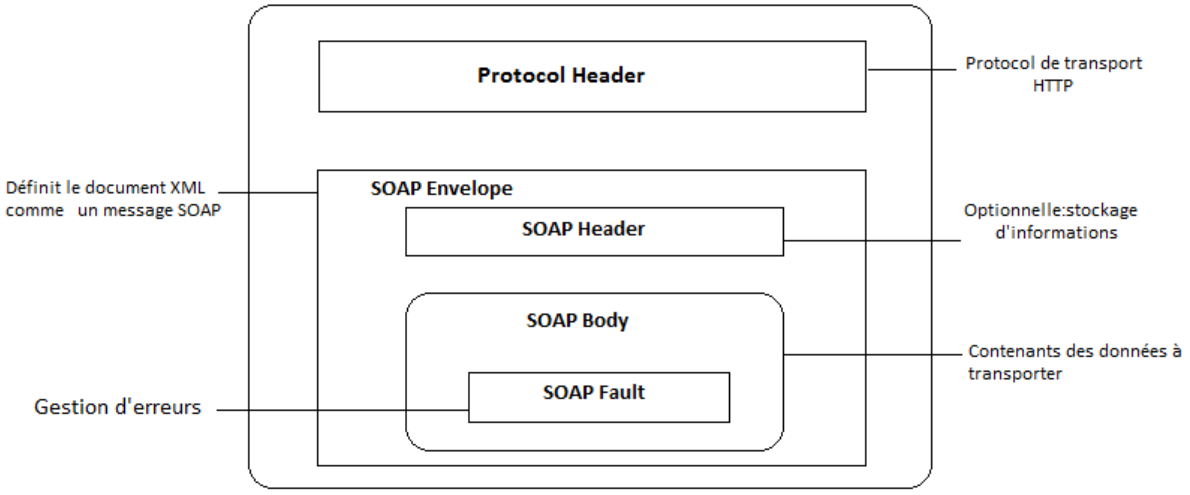
SOAP est un acronyme pour **Simple Protocol Object Access**. Il est un protocole de messagerie basé sur XML pour l'échange d'informations entre les ordinateurs. SOAP est une application de la spécification XML.

- SOAP est un protocole de communication destiné à communiquer via Internet.
- SOAP assure le transport de données pour les services Web.
- SOAP peut être utilisé pour la diffusion d'un message.

- SOAP est le moyen XML de définir est ce que l'information est envoyée et comment.
- SOAP permet aux applications clientes de se connecter facilement à des services distants et invoquer les méthodes distantes.

*I.7.2.*Structure d'un message SOAP

La structure des messages SOAP se divise en quatre parties



FigureI.4 : Structure d'un message SOAP

- **L'enveloppe SOAP**

L'enveloppe SOAP est l'élément obligatoire d'un message SOAP qui englobe les deux autres parties de ce message (Header et Body). Elle contient le nom du message et le nom du domaine, et permet de préciser la version de SOAP utilisée.

L'enveloppe SOAP indique le début et la fin du message, de sorte que le récepteur connaît lorsqu'un message entier a été reçu. L'enveloppe SOAP résout le problème de savoir quand vous avez terminé de recevoir un message et que vous êtes prêt à le traiter.

- **Le header (en-tête) SOAP**

Cette partie du message est optionnelle. Elle sert à transmettre des informations nécessaires pour l'exécution de la requête SOAP aux intermédiaires qui recevront le message. On y précise généralement des informations liées aux transactions, à l'authentification, etc.
Le header est composé d'un ou plusieurs champs : l'attribut **actor**, désignant le destinataire du header, l'attribut **must Understand**, qui indique si le processus est optionnel.
Dans l'exemple suivant, on précise des informations sur l'identification de l'utilisateur et la transaction à laquelle appartient le message :

```
<SOAP-ENV :Header
<SOAP-ENV : actor="http://schemas.xmlsoap.org/ soap/actor/next"
<SOAP-ENV:must Understand="1">
<identifiant numero="124527"/>
<transaction type="compensees" numero="YU75X"/>
</SOAP-ENV:Header>
```

- **Body**
  - Le corps SOAP est un élément obligatoire qui contient des données XML échangés dans le message SOAP.
  - Le corps doit être contenu dans l'enveloppe et doit suivre toutes les en-têtes qui pourraient être définis pour le message.
  - Le corps est défini comme un élément enfant de l'enveloppe, et la sémantique du corps est défini dans le schéma SOAP associé.
  - Le corps contient des informations obligatoires destiné au récepteur final du message.

- **Fault**

  Si une erreur se produit au cours du traitement, la réponse à un message SOAP est un élément de faute SOAP dans le corps du message, et l'erreur est renvoyée à l'expéditeur du message SOAP.

Le mécanisme d'erreur SOAP renvoie des informations sur l'erreur, y compris un code prédéfini, une description, et l'adresse du processeur SOAP qui a généré la faute.

I.8. UDDI

UDDI signifie Universal Description, Discovery, et de l'intégration, c'est un standard basé sur XML pour la description, l'édition, et de trouver des services Web.

- UDDI est un cadre ouvert de plate-forme indépendante.
- UDDI peut communiquer via SOAP, CORBA, Java protocole RMI.
- UDDI utilise Web Service DefinitionLanguage (WSDL) pour décrire les interfaces de services Web.
- UDDI est vu avec SOAP et WSDL comme l'une des trois normes de base de services Web.
- UDDI est une initiative ouverte de l'industrie, permettant aux entreprises de se découvrir et de définir la façon dont ils interagissent sur Internet.

I.8.1.Eléments UDDI

Une entreprise ou une société peut enregistrer trois types d'informations dans un registre UDDI. Cette information est contenue dans trois éléments de UDDI.
Ces trois éléments sont les suivants:

- Pages blanches
- Pages Jaunes
- Pages vertes.

- **Pages blanches**

Les pages blanches contiennent:

- informations de contact de base y compris le nom, l'adresse, numéro de téléphone, etc.
- elles contiennent également une liste d'identifiants grâce au quel une entreprise peut-être repérée.

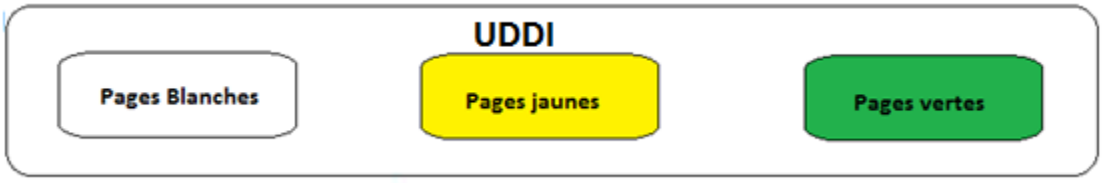
- **Pages Jaunes**

Comprennent la description au format WSDL des services Web déployés par les entreprises, elles répertorient les services Web par catégories.
Les pages jaunes sont composées d'informations permettant de classifier l'entreprise. Ces informations s'appuient sur des standards de classification industrielle normalisée.

- **Pages vertes**

fournissent des informations techniques détaillées sur les servicesfournis. Une page verte permet à une personne de se lier à un service Web après qu'il a été trouvé. Il comprend:

- Les différentes interfaces
- Les URL des emplacements
- Les informations de découverte et des données similaires nécessaires pour trouver et exécuter le service Web.



FigureI.5 : Les Eléments UDDI.

I.8.3 Modèle de données UDDI

UDDI inclut un schéma XML qui décrit les structures de données suivantes:

- Entité commerciale
- Service d'affaires
- Binding Template
- TModel
- Publisher Assertion

- **Structure business Entity données**

La structure de l'entité commerciale représente le fournisseur de services Web. Dans le registre UDDI, cette structure contient des informations sur l'entreprise elle-même, y compris les informations de contact, les catégories de l'industrie, les identificateurs d'entreprise, et une liste des services fournis.

- **Structure Business Service données**

La structure des services aux entreprises représente un service Web individuel fourni par l'entité commerciale (business entity). Sa description inclut des informations sur la façon de se lier au service Web, quel type de web service il est, et à quelles catégories taxinomiques il appartient.

- **Structure binding Template données**

binding Template sont les descriptions techniques des services Web représentés par la structure business service. Un service d'entreprise unique peut avoir plusieurs modèles de liaison (binding Template). La liaison modèle représente la mise en œuvre effective du service Web.

- **Structure tModel données**

tModel est le dernier type de données de base, mais potentiellement le plus difficile à saisir. tModel représente un modèle technique.
tModel est une façon de décrire l'entreprise , le service, et les structures de modèle stockées dansle registre UDDI. Tout concept abstrait peut être enregistré dans le UDDI comme tModel. Pour par exemple, si vous définissez un nouveau type de port WSDL, vous pouvez définir un tModel qui représente ce port entrez dans le UDDI. Ensuite, vous pouvez spécifier que le business service donnée implémente ce port tapez en associant le tModel avec l' un des modèles de liaison(binding template) de ce service d'affaires(business service).

- **Structure publisher Assertion données**

C'est une structure de relation qui met en association deux ou plusieurs structures de business Entity selon un type spécifique de relation, comme filiale ou département.
La structure publisher Assertion se compose de trois éléments :

- FromKey (the first business Key),
- To Key(the second business Key),
- Keyed Reference.

Le keyed Reference désigne le type de relation affirmée en termes de key Name,key Value dans un tModel, référencé uniquement par un tModel Key

I.9.UDDI - Avec WSDL

Le modèle de données UDDI définit une structure générique pour stocker des informations sur une entreprise et les services Web qu'elle publie. Le modèle de données UDDI est complètement extensible, comprenant plusieurs structures de séquences répétitives d'information.

Cependant, WSDL est utilisé pour décrire l'interface d'un service Web. WSDL est assez simple à utiliser avec UDDI.

- WSDL est représenté dans UDDI en utilisant une combinaison d'informations Business Service, binding Template et tModel.
- Comme pour tout service enregistré dans UDDI, des informations génériques sur le service est stocké dans la structure de données de Business Service, et des informations spécifiques à savoir comment et où le service est accessible est stocké dans une ou plusieurs structures de binding Template plus associés. Chaque

structure binding Template comprend un élément qui contient l'adresse de réseau du service et est associé à une ou plusieurs structures tModel qui décrivent et identifient de façon unique le service.

- Lorsque UDDI est utilisé pour stocker des informations WSDL, ou des pointeurs vers WSDL fichiers, le tModel devrait être nommé par convention comme le type wsdl Spec, ce qui signifie que l'élément de overview Doc est clairement identifié comme pointant vers une définition d'interface de service WSDL.
- Pour UDDI, WSDL contenu sont divisés en deux éléments majeurs du fichier d'interface et le fichier de mise en œuvre.

I.10 Conclusion

La technologie des services Web offre de fortes potentialités pour surmonter les problèmes d'interopérabilité des systèmes. Elle constitue un cadre prometteur pour l'intégration des applications, et pour la gestion des interactions entre divers partenaires dans un environnement distribué , hétérogène , ouvert et versatile qui est le Web.

Aux termes de ce chapitre, nous avons introduit le domaine des services Web en présentant leur architecture, leur principe de fonctionnement et leurs standards de base .

Dans le chapitre suivant nous aborderons le domaine e-learning .

II.1 Introduction:

Aujourd'hui, les principaux acteurs institutionnels du monde de l'enseignement et de La formation professionnelle, à savoir les hautes écoles, les universités, les centres de compétences, proposent des formations “en ligne“, utilisant Internet comme instrument et ne pas laissant l'apprenant seul face à sa formation. L'e-Learning est partout, il progresse à grand pas.

Et ce chapitre a pour objectif de présenter les concepts d’e-Learning, de présenter quelques avantages et inconvénients.

II.2 Qu’est-ce que le E-learning ?

Le « e » de *e-learning* est l’abréviation de « électronique ». C’est ce « e » qui est utilisé dans le terme e-commerce. Adossé à *Learning*(traduction : apprentissage), le *e-learning* devrait être traduit par « apprentissage électronique », mais cette appellation n’est pas très satisfaisante, et la notion de « en ligne » s’est substituée pour constituer : « apprentissage en ligne ».

II.3 Définition :

- **Définition 1 :** la définition donnée par la commission européenne : « *utilisation des nouvelles technologies multimédias et de l’Internet pour améliorer la qualité de l’apprentissage en facilitant l’accès à des ressources et des services, ainsi que les échanges et la collaboration à distance*». **[AWT, 2008]**

- **Définition 2 :** la définition proposée par le LabSET**:** *«* E-learning (ou electronic learning): apprentissage en ligne centré sur le développement de compétences par l'apprenant et structuré par les interactions avec le tuteur et les pairs». **[AWT, 2008]**

La définition de E-Learning est large, elle inclut plusieurs axes indépendants :

- Support en ligne ou hors ligne,
- Apprentissage individuel ou collectifs,
- Formateurs présent à distance, voir absent,
- D’autre terme sont parfois utilisé tels que :
  -**Formation en ligne**,
  -**Apprentissage en ligne**,
  -**Enseignement en ligne**,
  -**Formation à distance**,
  -**Enseignement à distance**.

II.4 Plate-forme de e-Learning

Une plate-forme de formation à distance est un logiciel d'apprentissage, combiné avec les technologies du Web, qui fournit des fonctionnalités à différents utilisateurs (Apprenants, Formateurs, Administrateurs) afin de conduire et d'automatiser le processus de formation à distance d'apprenants. La maintenance du système et la gestion des droits d'accès des apprenants aux cours sont assurés par un administrateur technique.
Les fonctionnalités d'une plate-forme varient selon ses concepteurs et ses objectifs.
Dans ce qui suit, nous citons certaines fonctionnalités qui sont communes à toutes les plates-formes :

- **La création d'objets pédagogiques:** Le concepteur d'objets pédagogiques crée des cours ainsi que des exercices et des tests (quiz, QCM…etc.), les structures grâce à des outils spécifiques, et y incorporent des données multimédias pour les rendre plus attractifs.

- **Gestion des documents pédagogiques :** classification, indexation, et administration des ressources pédagogiques.

- **Gestion administrative de la formation :** inscription dans la plate-forme, dans une formation ,gestion des données administratives de la formation.

- **Organisation du tutorat :** constitution des groupes d'apprentissages, choix des formateurs (enseignants), des groupes d'apprenants et de leurs accès, de types de contenus, de modes de communication, de types de tests…

- **Les évaluations:** Les plates-formes d'e-Learning offrent des outils de gestion de tests, d'élaboration et d'envoi de travaux. Les apprenants peuvent consulter des objets pédagogiques en ligne ou les télécharger, s'instruire, s'auto évaluer et transmettre à leur formateur d'éventuels travaux à corriger. Le formateur pourra alors évaluer les connaissances des apprenants selon leurs travaux ou grâce à des tests en ligne, proposer de nouvelles activités plus ou moins complexes selon les besoins des apprenants.

- **Apprentissage :** consultation à distance de contenus pédagogique, communication entre formateurs et apprenants et entre apprenants, individualisation des apprentissages, télé-tutorat.

II.5 Les acteurs d'une plate-forme et leurs rôles

Le nombre d'acteurs et leurs fonctions diffèrent d'une plate-forme à une autre, selon les objectifs et les fonctionnalités offertes par chacune d'entre elles. Certaines se sont contentées de trois acteurs, à savoir l'apprenant, le formateur/créateur de cours et l'Administrateur.
Cependant, cinq rôles principaux ont étés distingués en observant un ensemble de plates formes de formation à distance. Ces rôles sont :

- **Concepteur** (ou créateur de cours) : Il établit (pour l'apprenant) un parcours de formation personnalisé et individualisé. Il construit, adapte et maintient un système d'apprentissage. Il peut aussi avoir le rôle de présentateur. Ce rôle consiste à rendre disponible les informations pour l'apprentissage, en d'autres termes, la présentation des informations, la gestion des médias donnant ces dernières, la clarification des contenus en réponse à des questions, l'analyse et l'évaluation des contenus des documents…etc.

- **Formateur** (ou tuteur) : Il accompagne les apprenants durant leur parcours pédagogique.
  Il a la tâche de motiver, d'orienter et d'évaluer les apprenants, comme il les assiste dans l'utilisation des ressources et du matériel mis à leur disposition.
- **Apprenant :** Son rôle consiste à transformer les informations en connaissances. Pour cela, il doit gérer ses activités et son temps, exploiter les ressources qui lui sont indiquées de façon à optimiser la quantité d'information qu'il peut en tirer, résoudre des problèmes, échanger des informations et des idées avec d'autres apprenants via les outils de communication, s'auto évaluer et présenter des travaux qui serviront de référence à son formateur pour son évaluation.

- **Administrateur :** Un administrateur installe et assure la maintenance du système, ils'occupe des tâches d'administration comme les inscriptions des acteurs dans la plateforme, il gère les droits d'accès, assure la gestion des ressources pédagogiques, ce qui consiste en la classification, l'indexation et l'administration des matériaux pédagogiques.

- **Evaluateur:** Il s'occupe de l'évaluation des groupes et des individus et ceci grâce aux informations fournies par la plate-forme.

II.6 Outils de communications

Généralement les plates-formes SGC (Système de Gestion de Cours) 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 spécifique 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 (stagiaires et tuteurs). Ce système permet l'envoi et la réception de mails avec ou sans fichier attaché.
  Il peut être interne à la plateforme et ne nécessite pas d'avoir un e-mail personnel.

- **Le forum :** Le forum, pouvant 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 des membres du groupe de discuter en temps réel.

- **Les documents pédagogiques partagés :** Une zone commune peut être utilisée pour mettre à disposition du groupe différents documents : Documents sous forme numérique (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.

- **La visioconférence:** Equivalente à l'intervention du formateur en salle face à un groupe, mais ici chacun est devant son ordinateur, à domicile, au travail, ou un centre de ressources…

- **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 qui peut y noter toutes les Informations qu'ils ouhaite:ce qu'il a retenu des formation,des actions à mener…etc.
  Ces notes sont 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:** Cet espace permet aux utilisateurs de trouver éventuellement la réponse à des questions récurrentes

II.7 Les différentes formes de la formation en ligne

La formation en ligne se présente sous une multitude de formes. Cette particularité facilite son intégration aux stratégies de formation continue au près des apprenants.
On distingue deux types de formation en ligne :

II.7.1 Formation exclusivement en ligne (sans tutorat)

Ce type de formation d'avantage individualisé, est proche du service fourni par un CD Rom ou d'une vidéo de formation. C'est un modèle d'apprentissage sans formateur, par lequel l'apprenant communique avec différents serveurs pour chercher de l'information et accéder à des modules d'autoformation sur des sites Web, des CD Rom ou DVD.

II.7.2 Formation exclusivement en ligne (avec tutorat)

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

- **Asynchrone**

La formation asynchrone est une méthode d'apprentissage s'adaptant aux disponibilités de l'apprenant.

En bref, celui-ci a accès à un ou des instruments (exemples : vidéo, texte, logiciel d'apprentissage virtuel) qu'il utilisera à sa guise et il n'est pas obligé d'être connecté en même temps que les autres apprenants et le formateur. Cependant, Le rythme de l'apprentissage est géré par le formateur.

L'échange des documents et la participation au discours se feront par voie indirecte (courriel, forum de discussion).

- **Synchrone**

Dans la formation synchrone les apprenants et le formateur doivent être connectés en même temps. C'est aussi un modèle collectif. Cette pratique demande un équipement de vidéoconférence en salle ou sur poste de travail pour que la diffusion soit en direct.

II.7.3 Formation mixte (avec tutorat en ligne)

C'est plutôt le modèle des écoles et organismes de formation. En mettant en ligne le contenu de la formation, des tests, des évaluations et un tutorat, cela permet de réduire le temps du présentiel et d'individualiser la formation.

Ce modèle de formation combine donc les éléments de l'apprentissage en ligne et de l'apprentissage traditionnel en classe. Afin de faciliter l'intégration des connaissances acquises à travers une formation.

II.8 les Avantages et Inconvénients

II.8.1 Avantages

- **Accessibilité :** l'e-learning peut faciliter l'accès à la connaissance. Un ordinateur équipé d'une connexion à Internet suffit. Il n'est pas nécessaire de se déplacer.
- **Flexibilité :** Selon la formule choisie, la formation peut être suivie à n'importe quel moment, à n'importe quel rythme et depuis n'importe quel endroit.
- **Performances :** de nombreuses études mettent en avant de meilleurs résultats de la formation en e-Learning que lors d'une formation présentielle traditionnelle.

- **Pratique :** l'e-Learning est un système de formation ultra-flexible, le stagiaire peut apprendre d'où il souhaite et quand il le désire. Il apprend sans se déplacer. Le suivi de la formation est facilité, et accessible en temps réel grâce aux outils de gestion. [10]
- **Le stagiaire est l'acteur de sa formation :** le stagiaire avance à son rythme. Il est au cœur du dispositif et se sent responsable de ses résultats, puisqu'il fixe lui-même son emploi du temps et constate ses propres résultats.
- **Choix du mode de formation :** les formations en E-Learning permettent de choisir entre cours individuels ou collectifs.
- **Baisse des coûts de formation :** les coûts de formation sont réduits : pas de frais de déplacement, pas de perte de temps ni de frais d'hébergement. Le coût de la solution E-Learning est aussi inférieur au coût horaire d'un formateur. L'accès au cours est généralement illimité.

II.8.2. Les inconvénients

- **Effet social :** dans les entraînements on-line le contact avec le tuteur et les autres participants s'effectue exclusivement virtuellement. IL n y a pas en général des questions posées immédiatement.
- **Qualité insuffisante :** le taux de personnes finissant prématurément le cours on-line est très haut. La raison majeure c'est la qualité souvent insuffisante des cours d'e-Learning, qui ne permettent pas une interaction ou qui sont mal traités.
- **Motivation :** l'e-Learning demande une grande motivation propre, ce que n'est pas souvent le cas chez tous les participants au début. Le participant a en outre besoin d'une certaine compétence d'apprendre tout seul pour pouvoir évaluer les «contenus des cours », de les sélectionner et de s'organiser soi-même.

- **Individualité :** les cours-online doivent être attractifs pour un grand nombre d'utilisateurs mais malheureusement on ne se concentre pas toujours sur les besoins individuels de chaque participant.
- **Flexibilité :** offreurs de « solutions d'e-Learning » peuvent agir significativement plus flexiblement que ceux des « méthodes de formation traditionnels ».
  Les coûts pour bâtiments et personnels d'enseignement peuvent être réduits car beaucoup des entraînements – online peuvent être distribués à un grand nombre d'utilisateurs.
- **Tolérance :** les participants de cours-online se trouvent dans un espace sans risques, qui tolère les fautes. Les erreurs sont indiquées aux participants qui les commettent (ce qui est moins intimidant).
- **Maîtrise des outils :** l'e-Learning nécessite une maîtrise suffisante des outils informatiques et d'Internet pour pouvoir suivre la formation.

II.9 Conclusion

E-Learning n'est pas seulement un changement de technologie. Il fait partie d'une redéfinition de la façon dont nous en tant qu'espèce transmettre des connaissances, des compétences et des valeurs aux jeunes générations de travailleurs et d'étudiants. Ce chapitre fait quelques prédictions sur la façon dont l'e-Learning et les fonctions qu'elle dessert continuera à se développer. Les apprenants auront accès à des millions ou des milliards de modules de connaissances. Certains seront des pages Web avec un texte simple et graphiques. D'autres peuvent inclure des simulations multimédias. Dans de nombreux domaines, l'e-Learning est devenu le moyen par défaut de mener une formation ou d'offrir une éducation. Il y a quatre secrets de e-Learning. Le premier secret est d'enseigner ce que les apprenants ont besoin d'apprendre dans la façon dont ils apprennent le plus naturellement. Le second secret est de définir des objectifs d'apprentissage clairs. Le troisième secret repose sur les deux premiers. Il est de se concentrer sur les bons objectifs. Le dernier secret est dans la puissance de l'essai.

Le chapitre suivant est consacré à la présentation de l'analyse et la conception de la solution dégagée dans ce présent chapitre.

III.1 Introduction

Avant toute réalisation d'une application informatique, il convient de suivre une démarche méthodologique et rigoureuse pour planifier et concevoir l'application, en mettant en évidence tous les objectifs tracés pour la bonne élaboration du projet souhaité.

Pour notre projet, nous allons suivre une démarche de conception orientée objet (RUP, Rational Unified Process), en se basant sur la modélisation UML.

Le Processus Unifié de Rational comprend 4 phases :

- **Inception** : vision générale du projet, limites et les principaux risques
- **Élaboration :** Le plan du projet, il spécifie les exigences, les bases de l'architecture
- **Construction :** Réalise le produit
- **Transition :** Transfère le produit vers les utilisateurs finaux

III.2 Problématique

On assiste depuis quelques années à l'émergence de nouvelles applications qui ont besoin de partager des informations entre différents systèmes. C'est le cas du e-learning, du e-commerce, de la bioinformatique ou encore des bibliothèques électroniques. Or, dans ce contexte, les systèmes d'informations (SI), conçus et développés par des organisations différentes, constituent généralement des sources de données autonomes et hétérogènes.

Et dans toute application le problème qui se pose toujours est :**l'application est-elle interopérable ?**

Pour rappel l'interopérabilité est la capacité d'un système de se communiquer et d'échanger des données avec d'autres systèmes différents (en vue de plate-forme et codage.

Les Services Web apparaissaient comme une solution répondant au problème de l'interopérabilité.

III.3 Phase Inception

Notre projet a pour objectifs

Création d'une plateforme interopérable « Plate-forme pour l'enseignement à distance »Qui permet à l'utilisateur de navigué sur l'application et s'inscrire en tant qu'apprenant afin de pouvoir suivre des cours disponibles, et pour faciliter la consultation des cours à l'apprenant, une barre de recherche dans son espace personnel est mise à sa disposition.

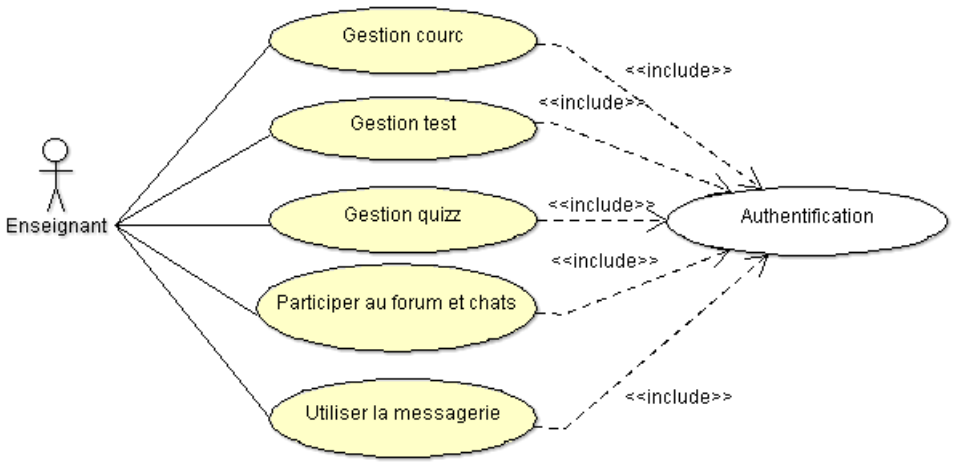
Ces cours sont mises en ligne par les enseignants inscrits sur la plate-forme.

Pour assurer un bon fonctionnement de l'application, un administrateur gère les différents acteurs du système.

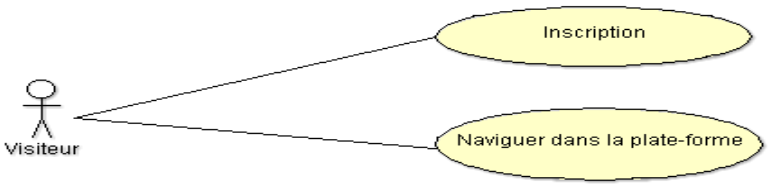
Et les principaux acteurs qui interagissent dans notre projet sont :

- **Visiteur**: c'est toute personne qui navigue sur le site et qui n'a pas encore eu un login et un mot de passe pour se connecter à la plate-forme.
  Et pour avoir l'accès à la plate-forme il doit remplir un formulaire d'inscription.
- **Apprenant :** C'est un visiteur qui veut suivre des cours sur la plate-forme, et pour cela il remplit un formulaire et attend la réponse de l'administrateur (accepter ou refuser son inscription) et dans le cas d'une réponse favorable alors il devient un apprenant et il pourra suivre des coursvia la plate-forme et autre chose comme l'accès au chat et au forum ...
- **Enseignant :** c'est un visiteur qui veut s'inscrire en tant que enseignant et pour cela il doit remplir un formulaire spécial et attend une réponse de l'administrateur et une fois qu'il est accepté par ce dernier il pourra utiliser les offres disponibles sur la plate-forme.
- **Administrateur :** C'est le responsable de l'application et c'est lui qui gère la plate-forme et les autres acteurs essentiellement les apprenants et les enseignants.

Et pour comprendre le fonctionnement de notre projet voici quelques cas d'utilisation :



FigureIII.1 **:Diagramme de cas d'utilisation relatif à l'enseignant.**

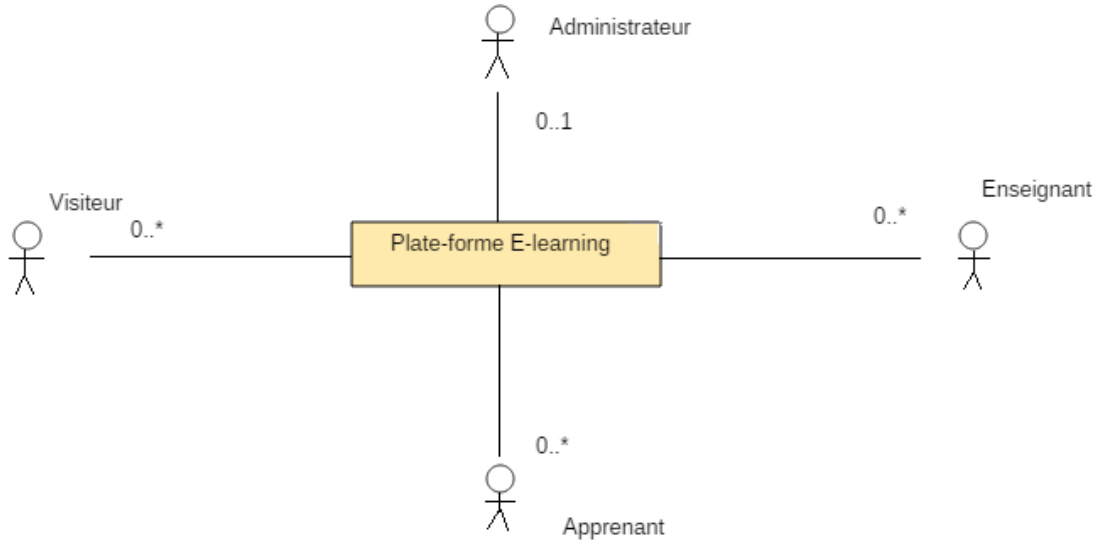


FigureIII.2 **:Diagramme de cas d'utilisation relatif au visiteur.**

III.4 Phase Élaboration

III .4.1 *Diagramme de contexte*

Présente le système à modéliser et les différents acteurs qui interagissent avec ce système



FigureIII.3:Diagramme de contexte

III.4.2 *Diagramme de cas d'utilisation complet*

Le diagramme de cas d'utilisation complet est un diagramme qui représente les relations entre tous les acteurs et les fonctionnalités du système.

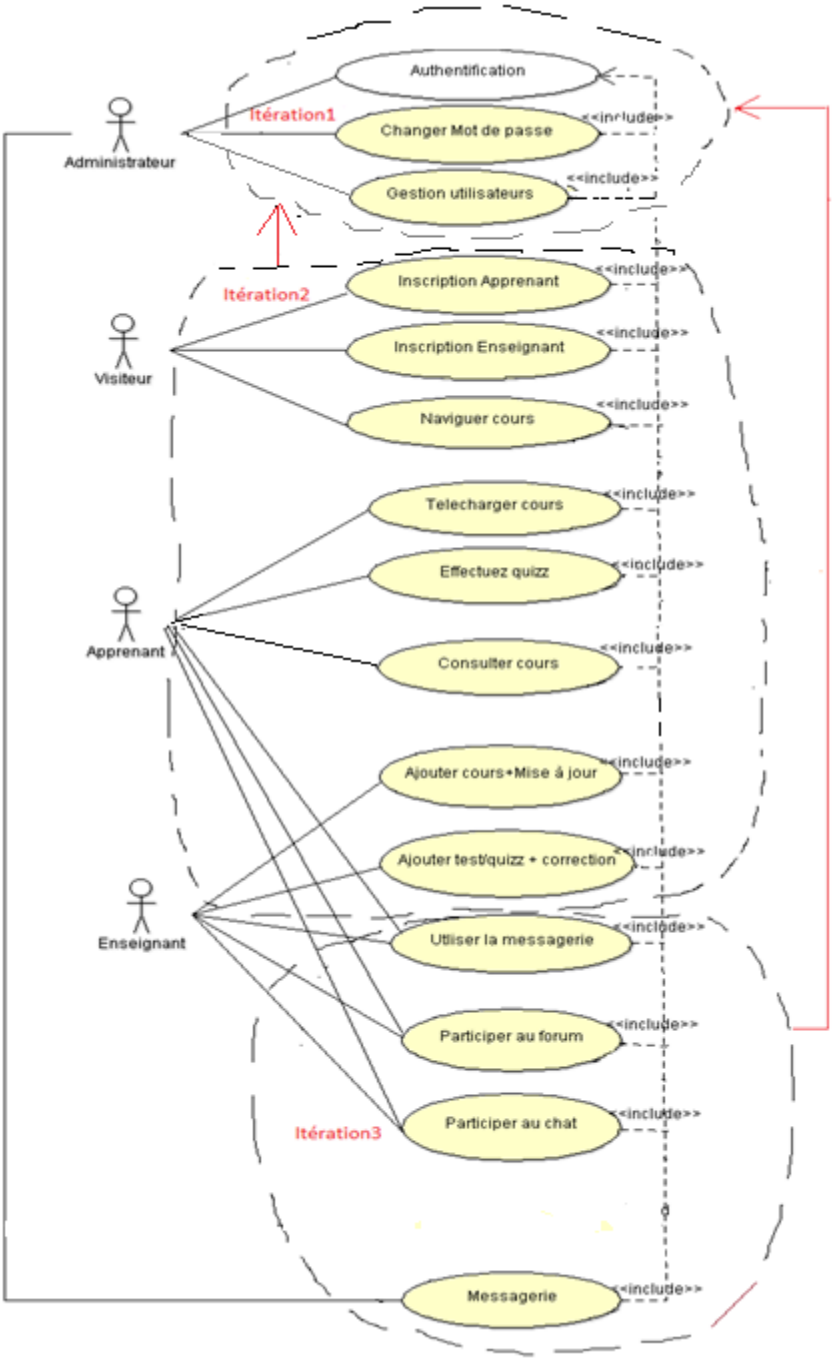


***Figure III.4*:Diagramme de cas d'utilisation complet.**

III.5 Phase construction

Dans le diagramme de cas d'utilisation complet on a défini 3 itérations :

- Itération 1 : -Authentification,
  -Changer mot de passe.
  -Gestion utilisateurs.

- Itération 2 : -Inscription apprenant,
  -Inscription enseignant.
  -recherche cours,
  -Télécharger cours,
  -Effectuer quizz,
  -consulter cours,
  -ajouter cours+mise à jours,
  -ajouter test/quizz +correction,
  -naviguer dans la plate-forme.

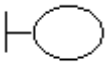
- Itération 3:-gestion de la messagerie,
  -utiliser la messagerie,
  -participer au forum,
  -participer au chat.

III.5.1 Réalisation de l'itération 1

III.5.1.1 Diagramme de séquence pour l'itération1

Les diagrammes de séquence présentent la coopération entre différents objets. Les objets sont définis et leur coopération est représentée par une séquence de messages entre eux. Après avoir déterminé nos besoins et réaliser les diagrammes des cas d'utilisations nous allons élaborer quelques diagrammes de séquences pour les trois itération. Les classes d'objets utilisées dans la représentation du diagramme de séquence d'analyse peuvent être réparties dans les trois catégories selon les stéréotypes suivant:

Les objets de type interface : l'interface entre l'acteur et le système, par exemples application web, des pages web complètes.

L` icône utilisée : 

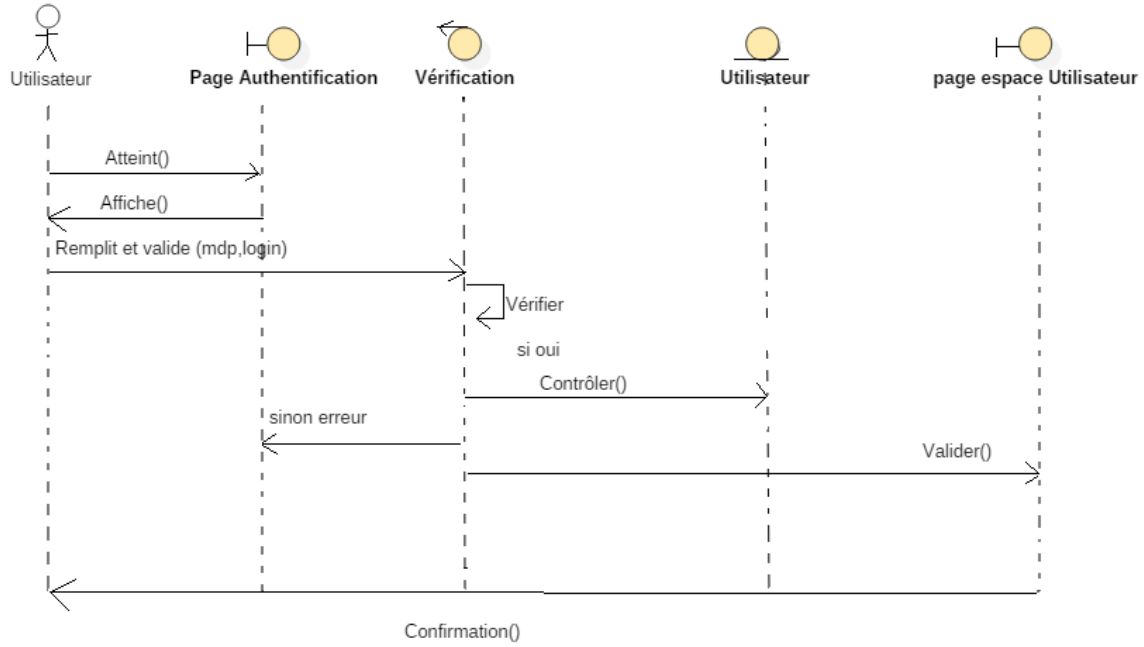
Les objets de type entité : sont des objets décrits dans un cas d'utilisation, qui peuvent apparaître dans de nombreux cas d'utilisation, et généralement sa durée de vie dépasse celle de toute Interaction où il participe.

L`icône utilisée : 

Les objets de type contrôle : représentent les processus, c'est-à-dire les activités système, ces objets dirigent les activités des objets entité et d'interface.

L`icône utilisée: 

III.5.1.1.1Diagramme de séquence de cas d'utilisation : « Authentification »



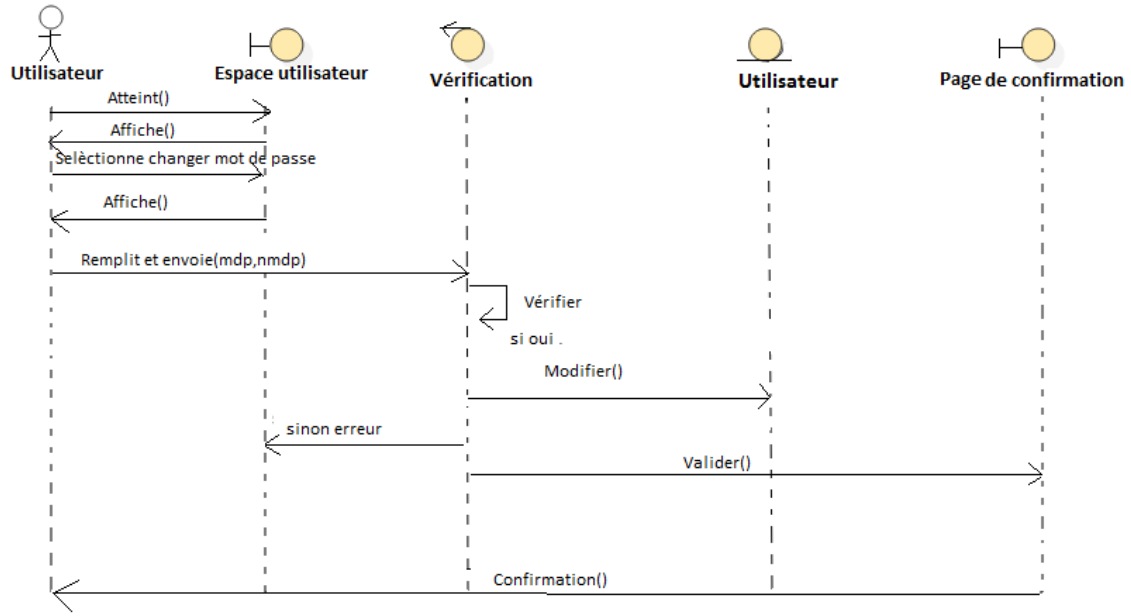
***FigureIII.5:*Diagramme de séquence relatif au cas d'utilisation « Authentification ».**

Description de cas d'utilisation avec des scénarios

Cas utilisation : Authentification
Acteur : UTILISATEUR
Résumé : Ce cas d'utilisation permet à l'utilisateur (UTILISATEUR) de s'authentifier et d'avoir un espace personnel.

Scénario :
1- L'UTILISATEUR atteint son espace.
2- Le système affiche « espace UTILISATEUR ».
3-L'UTILISATEUR sélectionne la rubrique «authentification».
4- Le système retourne le formulaire «authentification ».
5- L'UTILISATEUR saisit et soumet les données (nom, prénom, mot de passe, État utilisateur).
6- Le système retourne une page de confirmation.

III.5.1.1.2 Diagramme de séquence du cas d'utilisation:« Modification du mot de passe»

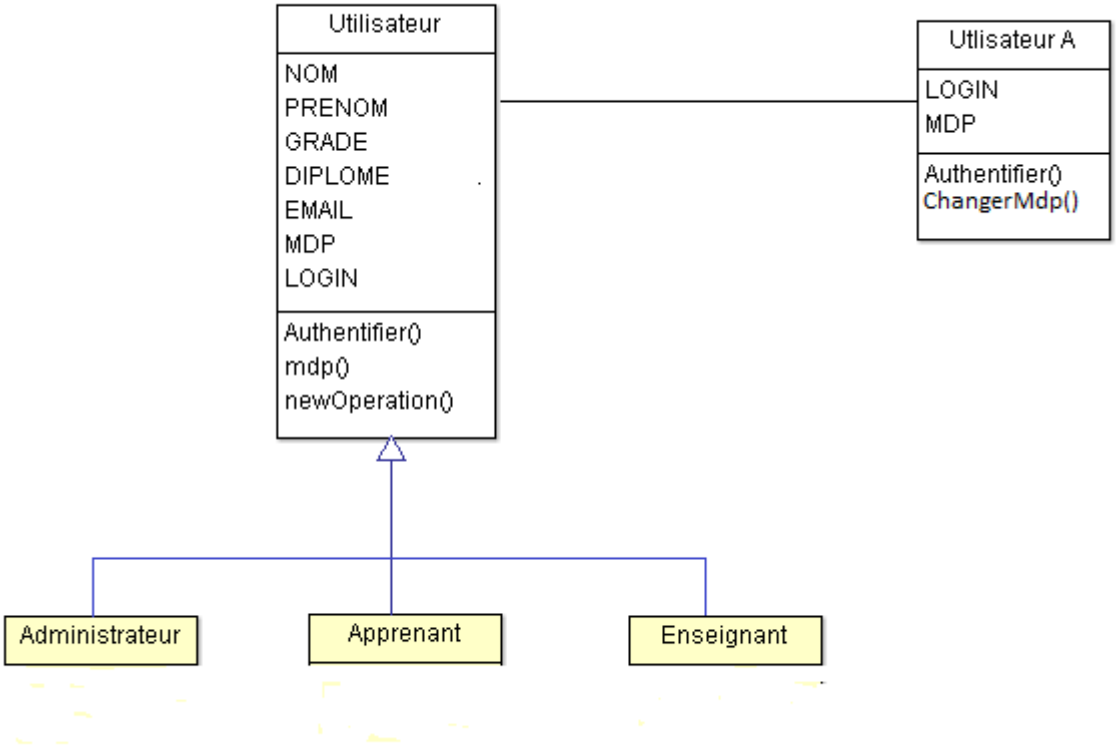


***FigureIII.6 :*Diagramme de séquence relatif au cas d'utilisation «Modification mot de passe».**

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Changer mot de passe
Acteur : UTILISATEUR
Résumé : Ce cas d'utilisation permet à l'utilisateur (UTILISATEUR) modifier son mot de passe
Scénario :
1-L'UTILISATEUR atteint son espace.
2-Le système affiche « espace UTILISATEUR ».
3- L'UTILISATEUR sélectionne la rubrique « changer le mot de passe».
4-Le système retourne le formulaire «changer le mot de passe».
5- L'UTILISATEUR saisit et soumet les données.
6-Le système retourne une page de confirmation.

III.5.1.2 Diagramme de Classe pour l'itération1

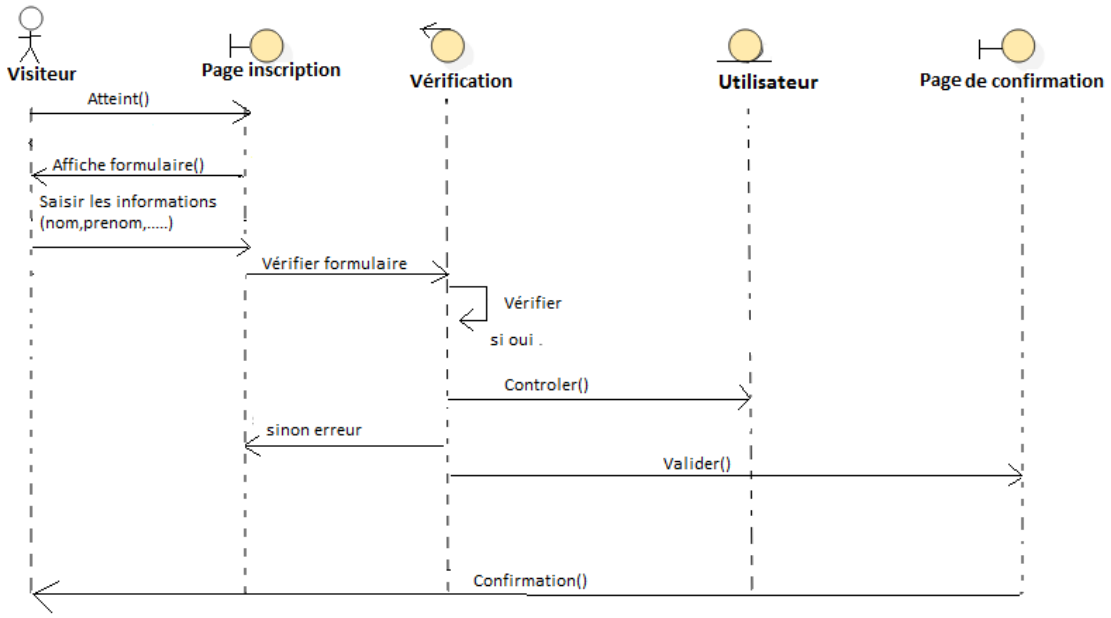


***FigureIII.7 :*Diagramme de classe pour l'itération 1.**

III.5.2 Réalisation de l'itération 2

III.5.2.1 Diagrammes de séquence pour itération 2:Un diagramme de séquence de l'itération2.

III.5.2.1.1 Diagramme de séquence du cas d'utilisation:« Inscription Apprenant»

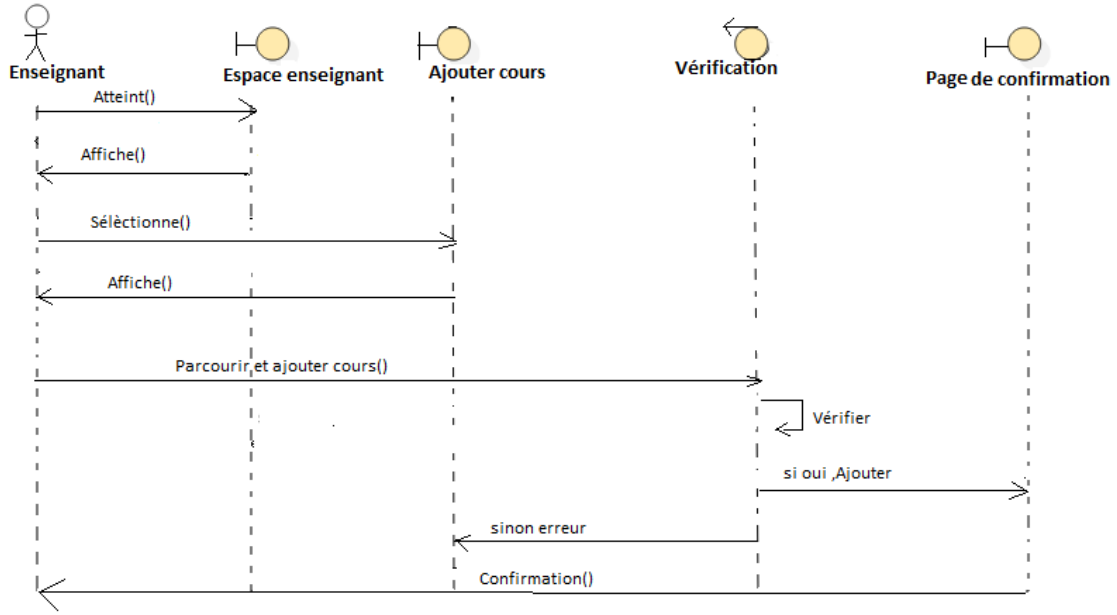


***FigureIII.8*:Diagramme de séquence relatif au cas d'utilisation « Inscription Apprenant».**

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Inscription Apprenant
Acteur : Visiteur
Résumé : Ce cas d'utilisation permet au visiteur de s'inscrire comme apprenant dans la plate-forme.
Scénario :
1-Le visiteur atteint la page Inscription pour apprenant.
2-Le système affiche le formulaire d'Inscription.
3- Le visiteur remplit les informations nécessaires.
4-Le système vérifie la validité des informations soumet et retourne une confirmation de l'inscription.

III.5.2.1.2 Diagramme de séquence du cas d'utilisation:« Ajouter cours»

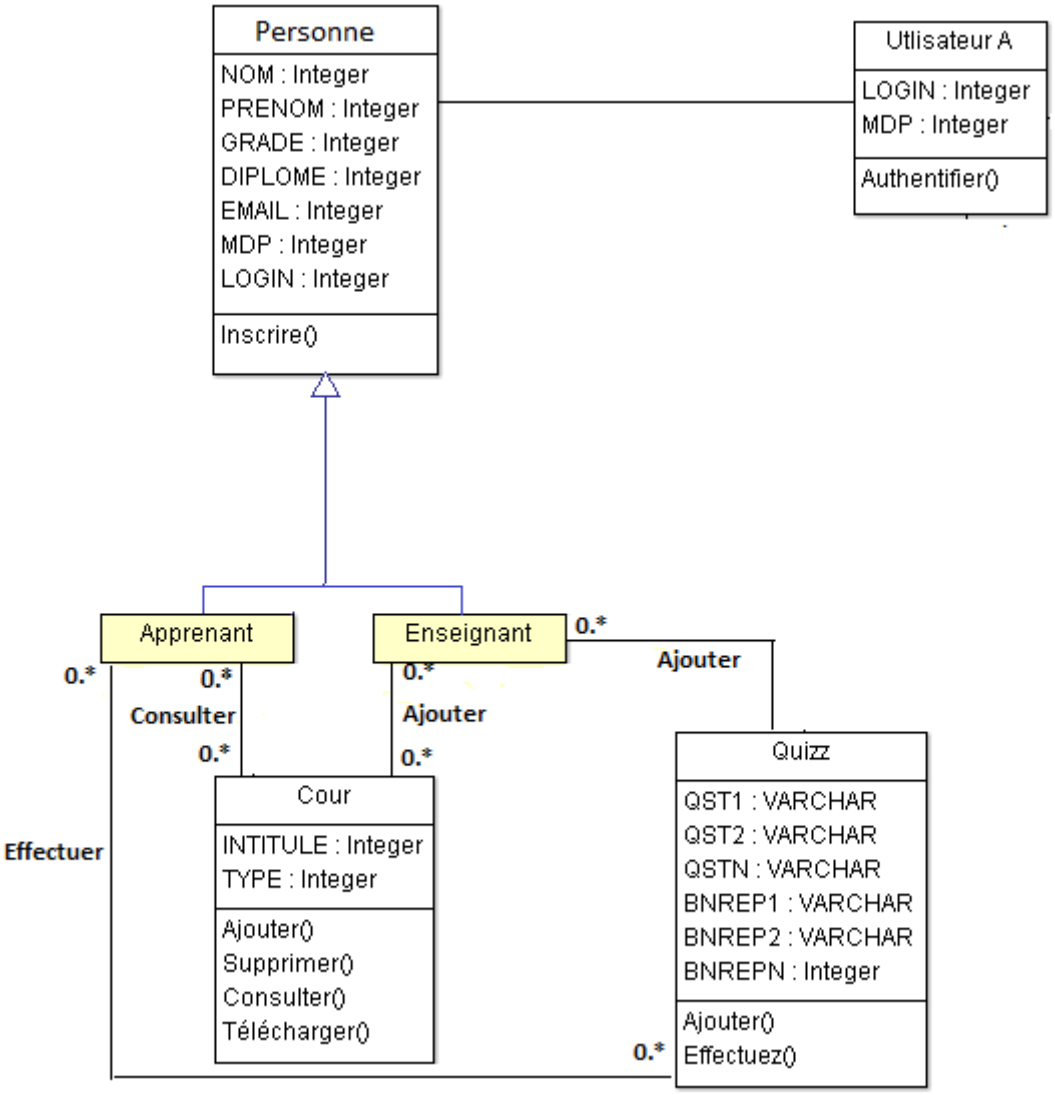


FigureIII.9 :Diagramme de séquence relatif au cas d'utilisation « Ajouter cours ».

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Ajouter cours
Acteur : Enseignant
Résumé : Ce cas d'utilisation permet à l'enseignant d'ajouter des cours dans la plate-forme.
Scénario :
1-L'Enseignant atteint son espace.
2-Le système affiche « espace Enseignant ».
3- L'Enseignant sélectionne la rubrique « Ajouter cours».
4-Le système retourne le formulaire d'envoi de fichier.
5- L'Enseignant sélectionne le cour à ajouter et clique sur ajouter.
6-Le système interroge la base de données et vérifie la validité du fichier.
7-Le système confirme l'ajout de cours.

III.5.2.2 Diagrammes de classe pour itération 2

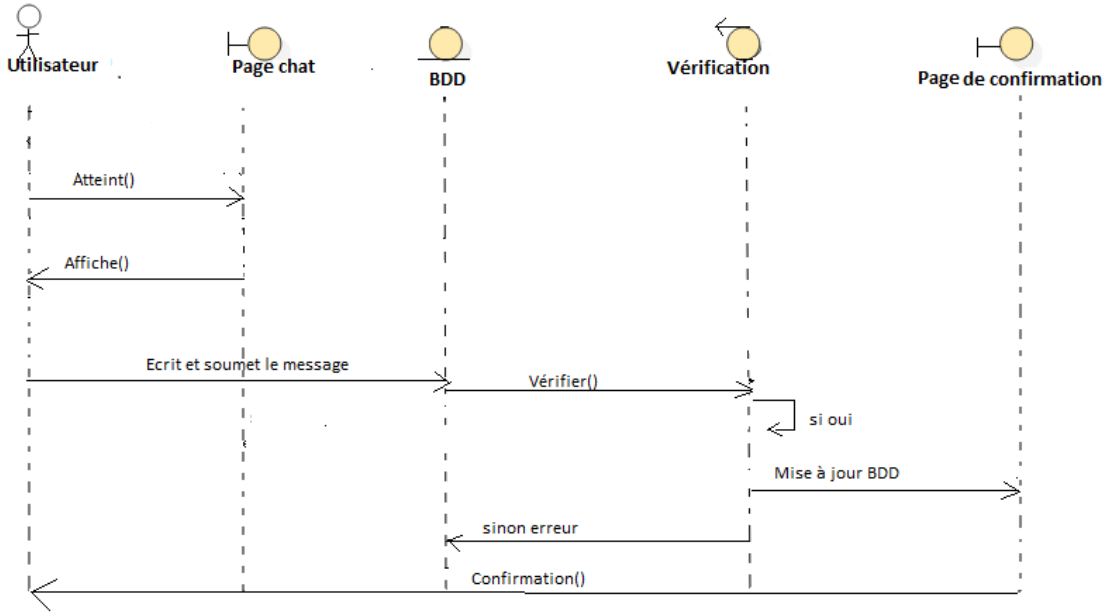


***FigureIII.10 :*Diagramme de classe pour l'itération 2.**

III.5.3 Réalisation de l'itération 3

III.5.3.1 Diagrammes de séquence pour itération 3:Un diagramme de séquence de l'itération 3.

*III.5.3.1 .1**Diagramme de séquence du cas d'utilisation:« Participer au chat»*

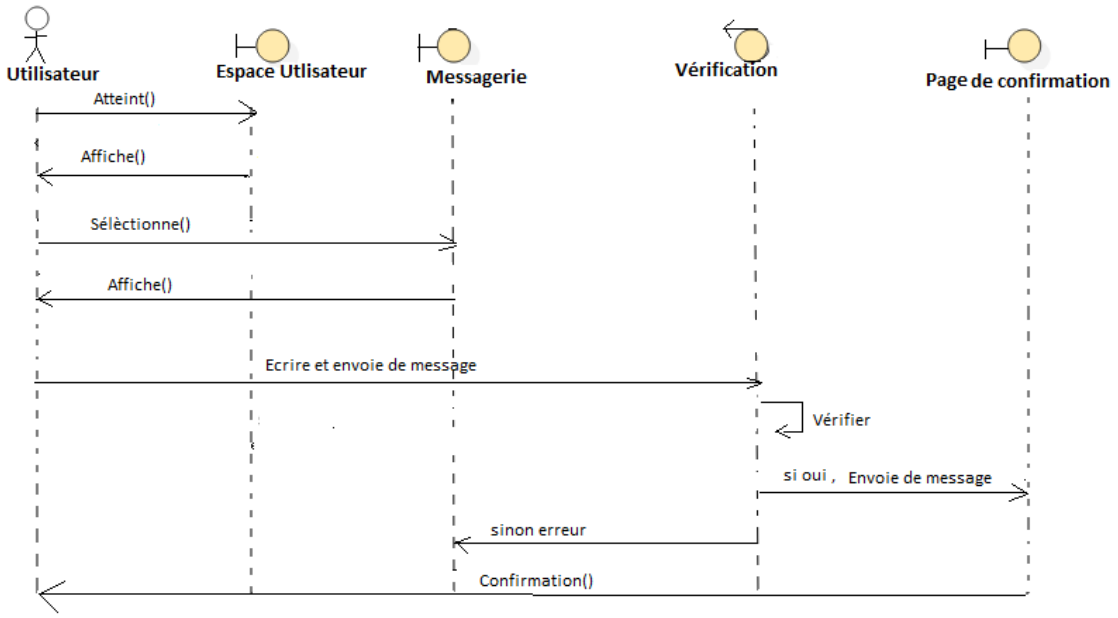


FigureIII.11:Diagramme de séquence relatif au cas d'utilisation « Participer au chat »

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Participer au chat
Acteur : UTILISATEUR
Résumé : Ce cas d'utilisation permet à l'utilisateur (enseignant/apprenant) de participer au chat
Scénario :
1-L'UTILISATEUR atteint son espace et connecte au chat.
2-Le système affiche la page du chat et les membres connectés.
3- L'UTILISATEUR écrit son message.
4-Le système interroge la base de données et ajoute le message à la table chat.
5- Le système affiche le message écrit par L'UTILISATEUR et retourne à l'étape 2 pour que l'UTILISATEUR écrit un nouveau message si il le souhaite.

III.5.3.1.2**Diagramme de séquence du cas d'utilisation:« Utiliser la messagerie»**

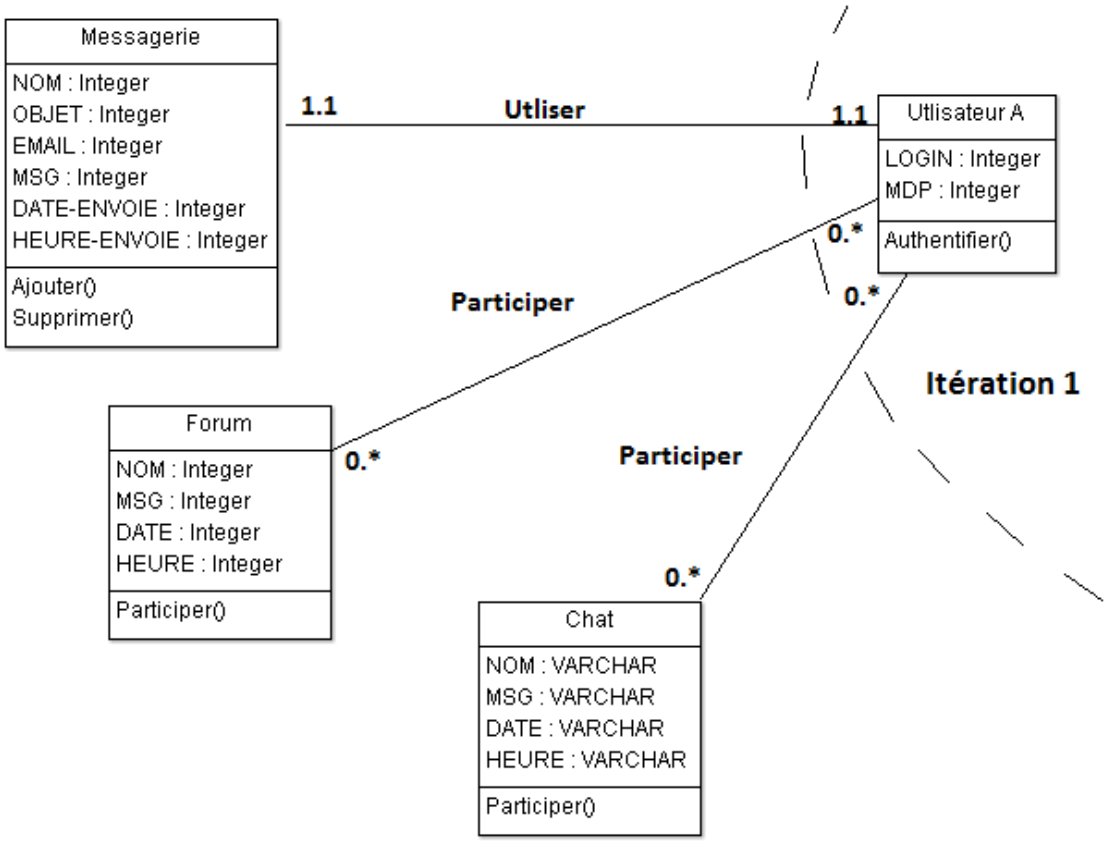


FigureIII.12 :Diagramme de séquence relatif au cas d'utilisation « Utiliser la messagerie ».

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Utiliser la messagerie
Acteur : UTILISATEUR
Résumé : Ce cas d'utilisation permet à l'utilisateur (UTILISATEUR) d'envoyer un message.
Scénario :
1-L'UTILISATEUR atteint son espace.
2-Le système affiche « espace UTILISATEUR ».
3-L'UTILISATEUR sélectionne la rubrique « messagerie».
4-Le système retourne la messagerie.
5-L'UTILISATEUR écrit son message et l'envoie.
6-Le système vérifie l'envoie de message retourne une confirmation de l'envoie.

III.5.3.1 Diagrammes de classe pour itération 3



***FigureIII.13:*Diagramme de classe pour l'itération 3.**

III.6 Conclusion

Au cours de ce chapitre nous avons entamé l'analyse et la conception d'un système E-Learning en utilisant le processus unifié RUP et ça en se basant sur la modélisation UML.
Et le chapitre suivant sera consacré à la réalisation de cette solution.

IV.1 Introduction

Après avoir explicité dans le chapitre précédent l'analyse et la conception de notre système, nous allons présenter dans ce dernier chapitre son implémentation.
Nous allons tout d'abord commencer par la présentation de l'environnement et les outilsde développements et d'implémentation de notre application, puis les différentes interfaces, et leurs fonctionnalités essentielles.

IV.2 Outils et Environnement de développement

IV.2.1 JAVA (JEE)

Pour réaliser notre application, nous avons utilisé le Java EE qui s'effectue généralement à l'aide d'un **Environnement de Développement Intégré**,

J2EE (Java 2 Enterprise Edition) est l'extension serveur de la plate-forme J2SE (Java 2 Standard Edition) de SUN. J2EE est une plate-forme de développement d'application s'appuyant sur le langage Java, dont les spécifications sont gérées par la société SUN ,qui permet de développer des applications Web composées de Servlet et JSP et des applications Métiers à base d'EJB. J2EE est également une spécification destinée aux éditeurs de logiciels qui désirent créer des Serveurs d'Applications compatibles J2EE.

IV.2.2 L'IDE Netbeans

NetBeans est un environnement de développement intégré (IDE) pour Java, placé en open source par **Sun**. En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditer en couleur, projets multi-langage, éditeur graphiques d'interfaces et de pages web).
NetBeans est disponible sous Windows, Linux et d'autres systèmes d'exploitation. Il est lui-même développé en Java, ce qui peut le rendre assez lent et gourmand en ressources mémoires.

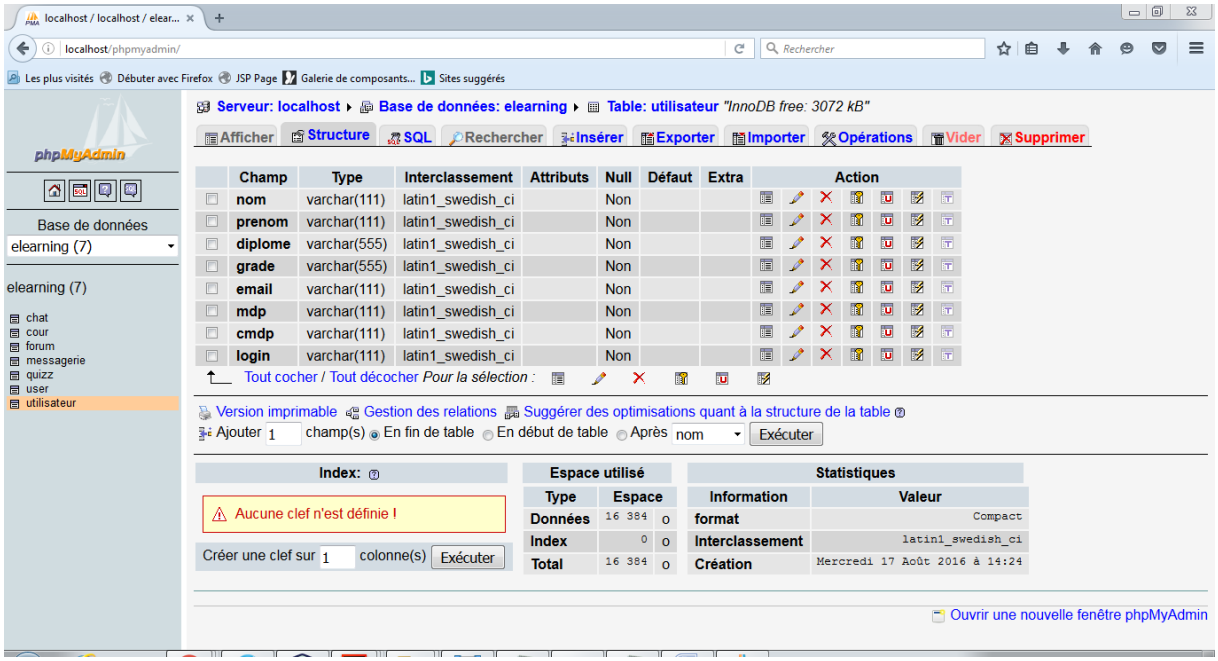
IV.2.3 Le serveur GlassFish

GlassFish est un serveur Java d'application saillent créé par Sun Microsystems qui permet à beaucoup de promoteurs de produire les technologies d'entreprise qui sont les services

commodes et évolutifs, aussi bien que supplémentaires qui peuvent être installés basés sur la préférence. C'est un logiciel libre, double-autorisé conformément à la Licence de Grand public de GNOU (GPL) et le Développement Commun et la Licence de Distribution (CDDL). GlassFish a été acquis par l'Oracle en 2010.

IV.2.4 PHPMyAdmin

PHPMyAdmin est une interface d'administration pour le SGBD MySQL. Il est écrit en langage PHP et s'appuie sur le serveur HTTP Apache.
Il permet d'administrer les éléments suivants :
* les bases de données
* les tables et leurs champs (ajout, suppression, définition du type)
* les index, les clés primaires et étrangères
* les utilisateurs de la base et leurs permissions
* exporter les données dans divers formats (CSV, XML, PDF, OpenDocument, Word, Excel)



FigureIV.1 : **L'interface PhpMyAdmin.**

IV.2.5 Les API utilisées

IV.2.5 .1 L'API Java Server Pages

Le Java Server Pages ou JSP est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédéfinies d'être ajoutés dans un contenu statique. Depuis la version 2.0 des spécifications, la syntaxe JSP est complètement conforme au standard XML.

La syntaxe du JSP ajoute des balises XML, appelées actions JSP, qui peuvent être utilisées pour appeler des fonctions. De plus, cette technique permet la création de bibliothèques de balises JSP (taglib) qui agissent comme des extensions au HTML ou au XML. Les bibliothèques de balises offrent une méthode indépendante de la plate-forme pour étendre les fonctionnalités d'un serveur HTTP. Il existe aussi un langage de script particulier, appelé Expression Language (EL) destiné à réduire l'injection de code java au sein des pages JSP ainsi qu'à étendre les possibilités des taglibs, tel que la JSTL.

Les JSP sont compilées par un compilateur JSP pour devenir des servlets Java. Un compilateur JSP peut créer une servlet Java en code source Java qui peut à son tour être compilé par

le compilateur Java, ou peut créer le pseudo-code Java interprétable directement. Dans les deux cas, il est bon de comprendre comment le compilateur JSP transforme la page en servlet Java. Voir l'exemple de page JSP fourni en fin d'article, avec la servlet créée et la page HTML résultante.

IV.2.5 .2 L'API Java Servlet

Une servlet est un composant logiciel, utilisé dans un serveur web, tel que Tomcat, qui peut être invoqué par les navigateurs clients via une URL. Le protocole de communication est dans ce cas HTTP. Même si le web dynamique est l'utilisation majeure de l'API Servlet, elle permet théoriquement de couvrir d'autres domaines d'application.

Le principe de fonctionnement est très simple : ce composant logiciel reçoit une requête, et il envoie une réponse. Cette réponse est transmise au client, qui l'interprète enfin.

Techniquement, l'API Servlet est un ensemble d'interfaces et de classes Java, rangées dans les packages javax.servlet et javax.servlet.http. Le protocole HTTP a bien sûr un statut particulier parmi les protocoles utilisés par les servlets.

IV.2.5 .3 L'API Java DataBase Connectivity JDBC

c'est une API qui fait partie intégrante de la plate-forme Java, et qui est constituée de classes permettant l'accès depuis des applications Java à des données rangées sous forme de tables. Dans la très grande majorité des cas, il s'agis de bases de données stockées dans un SGBD ! Les actions rendues possibles par cette

API sont:

- la connexion avec le SGBD,l'envoi de requêtes SQL au SGBD depuis une application Java ;

- le traitement des données et éventuelles erreurs retournées par le SGBD lors des différentes étapes du dialogue

IV.2.5 .4 Feuilles de style

IV.2.5 .4.1 Css

Les **CSS**, *Cascading Style Sheets*(feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web. Les feuilles de styles ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle. Ceci permet :

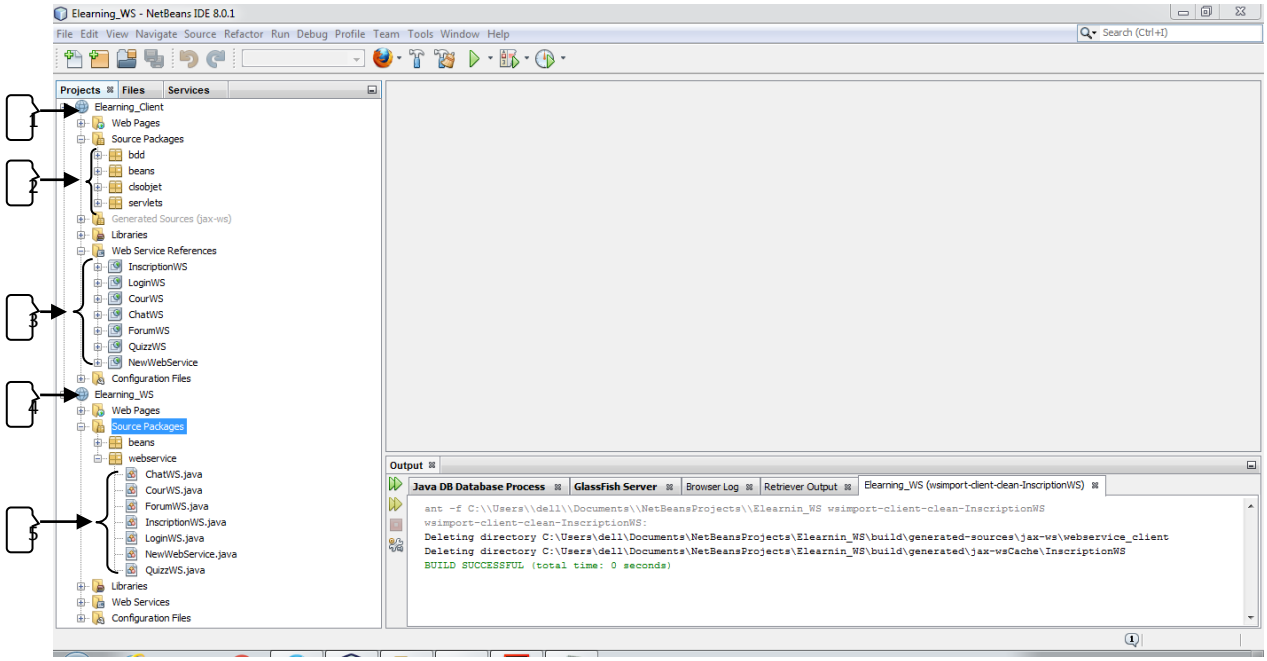
- de ne pas répéter dans chaque page le même code de mise en forme
- d'utiliser des styles génériques, avec des noms explicites (par exemple un style encadré pour du texte ou des images)
- de pouvoir changer l'apparence d'un site web complet en ne modifiant qu'un seul fichier

IV.2.5 .5 JAX-WS (Java API for XML Web Services)

L'API Java pour des Services Web XML (le JAX-WS) est une API de langage de programmation Java pour créer des services Web. Le JAX-WS est un de Java XML la programmation d'API. Il fait partie de la plate-forme EE Java.

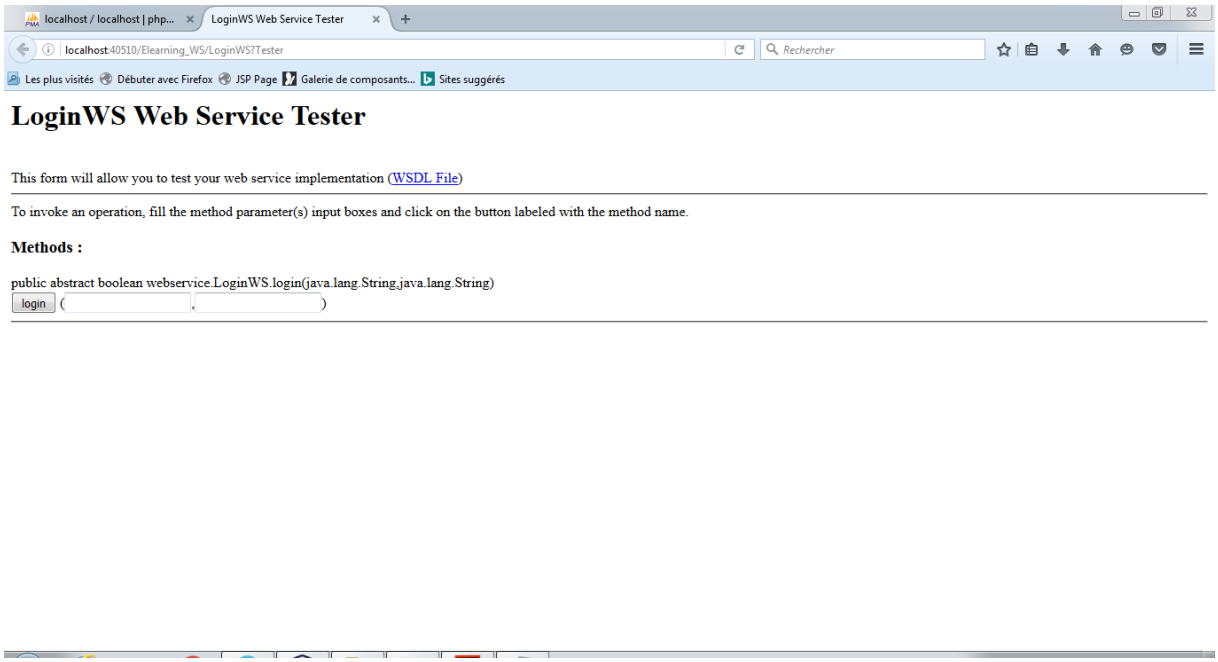
IV.3 Présentation de l'application

La figure *ci-dessous présente notre application*

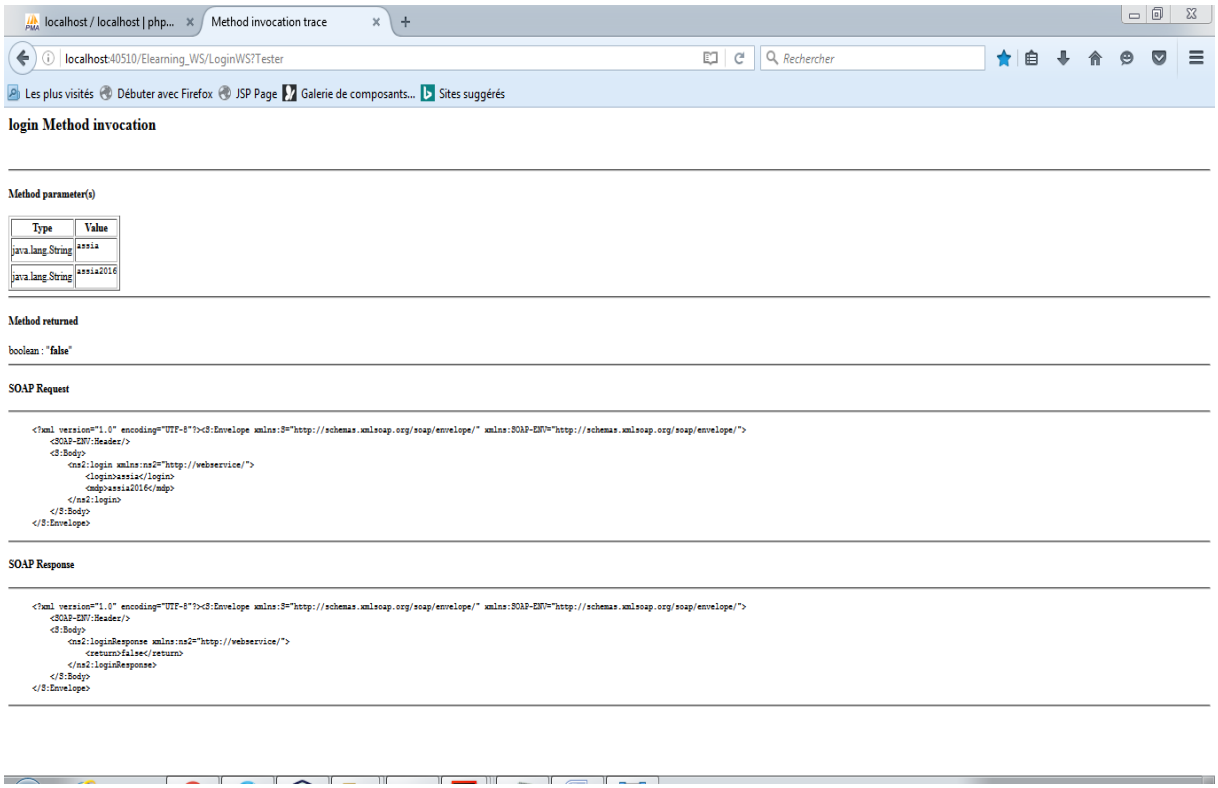


FigureIV.2 :l'interface Graphique de l'application.

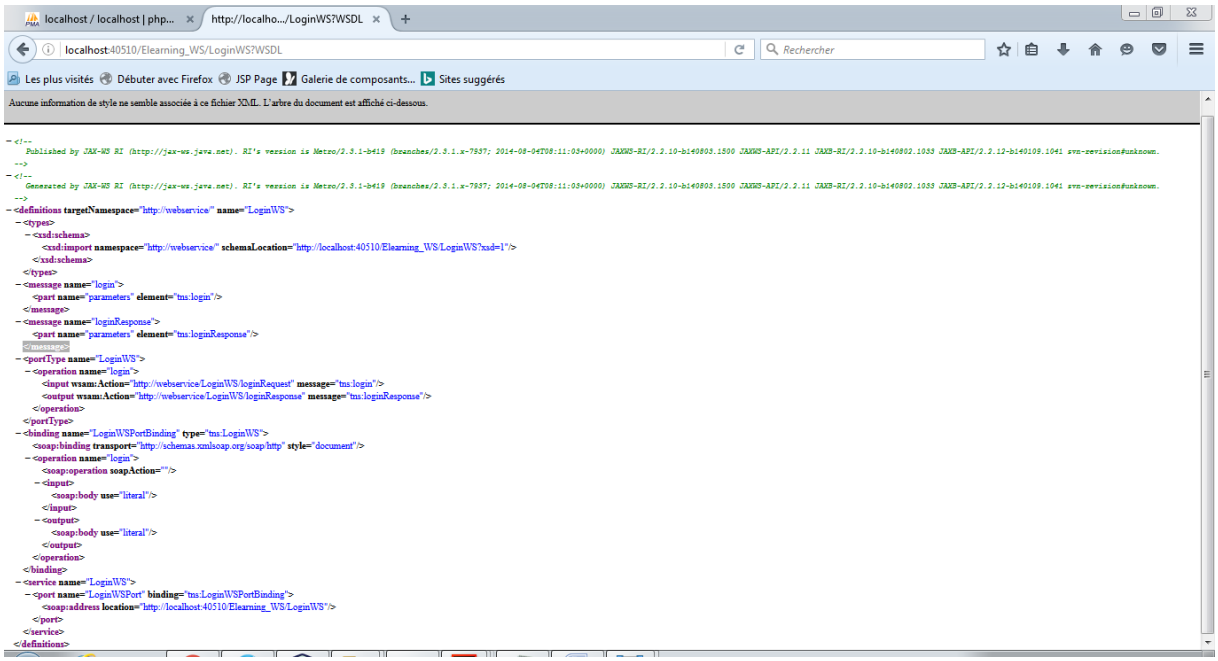
1 :C'est notre application Client.
2 : Ce sont les différent package de notre application Client.
3 :C'est les services web Client .
4 :C'est notre application.
5 :Ce sont les services web de notre application.



FigureIV.3 :Présentation graphique de la page tester le web service(LoginWS).



FigureIV.4 : Présentation graphique de la page SOAP de la méthode login(LoginWS).

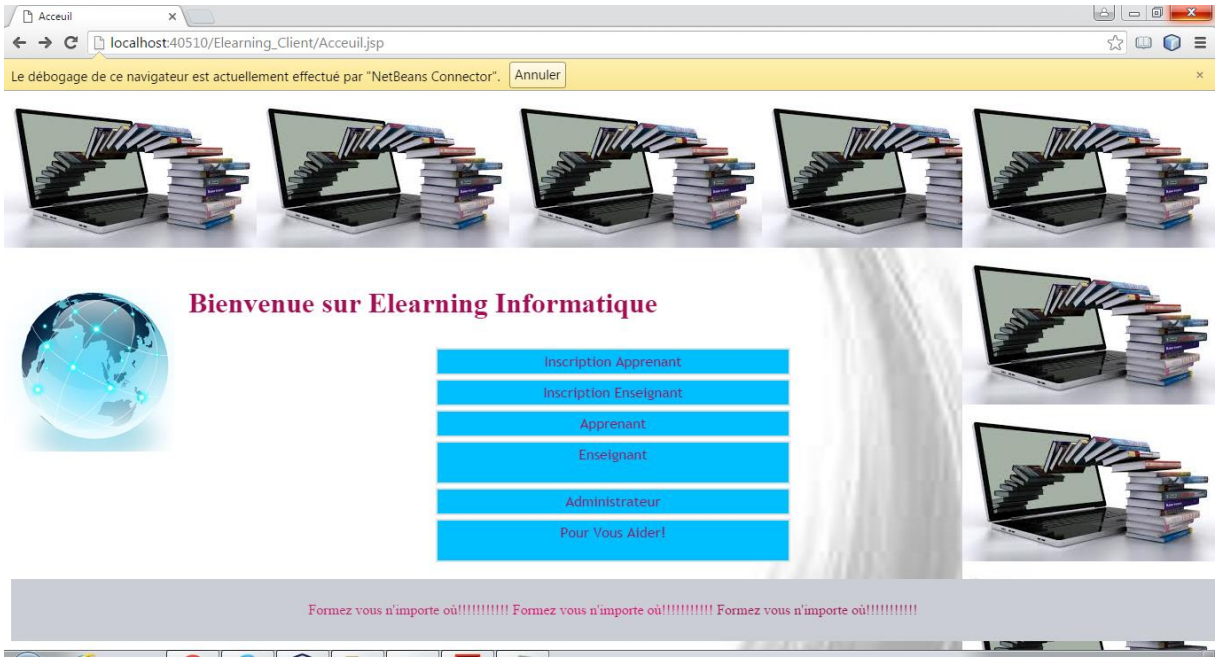


FigureIV.5 : Présentation graphique de la page WSDL de la méthode login(LoginWS).

IV.Présentation des interfaces graphiques de l'application

IV.4 .1 Page d'Accueil

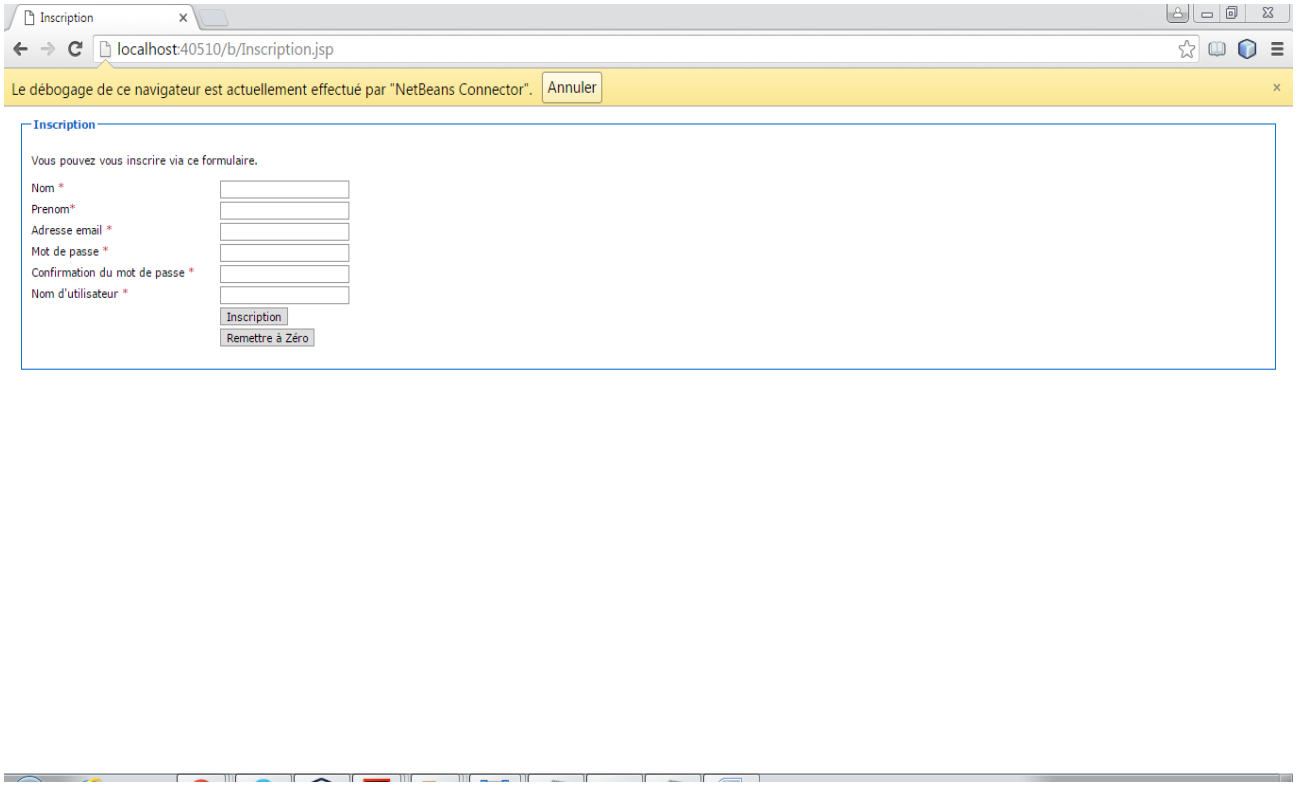
La page d'accueil est la première page visualisée, elle présente certains services proposés parle site.



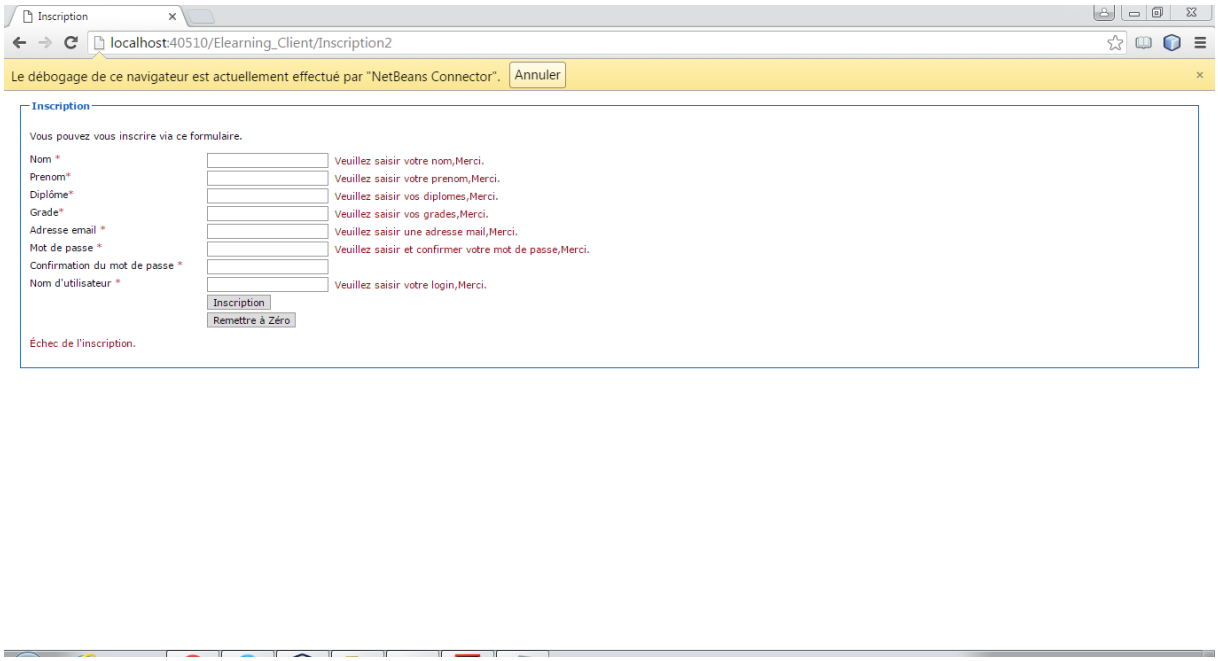
FigureIV.6: ***Présentation graphique de la page d'Accueil.***

IV.4 .2 Page d'Inscription

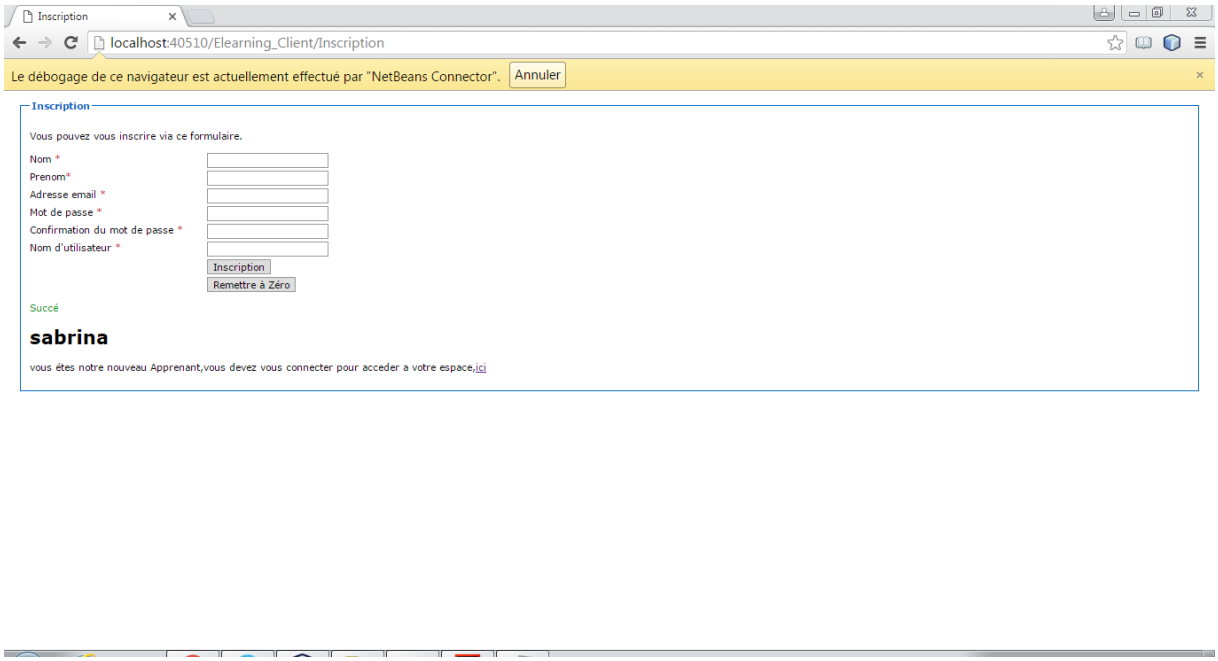
Chaque visiteur doit cliquez sur inscription et saisir les champs et choisir le bouton inscription pour créer son espace.



FigureIV.7: ***Présentation graphique de la page Inscription.***



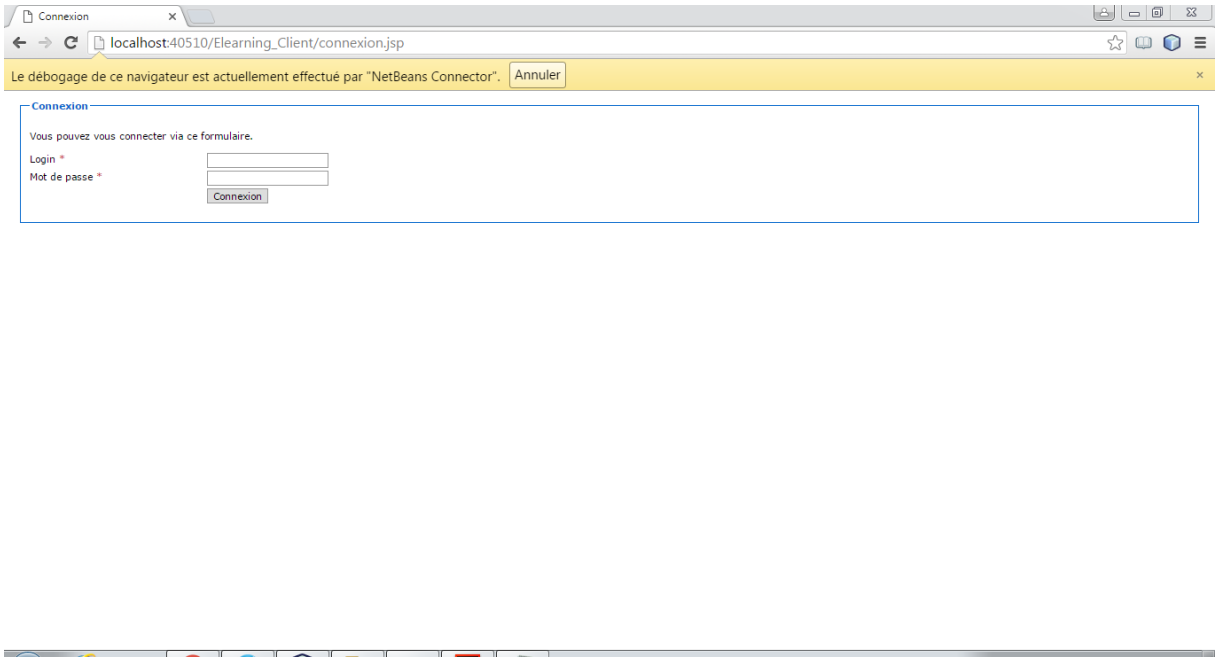
FigureIV.8: ***Présentation graphique de la page Echec d'Inscription.***



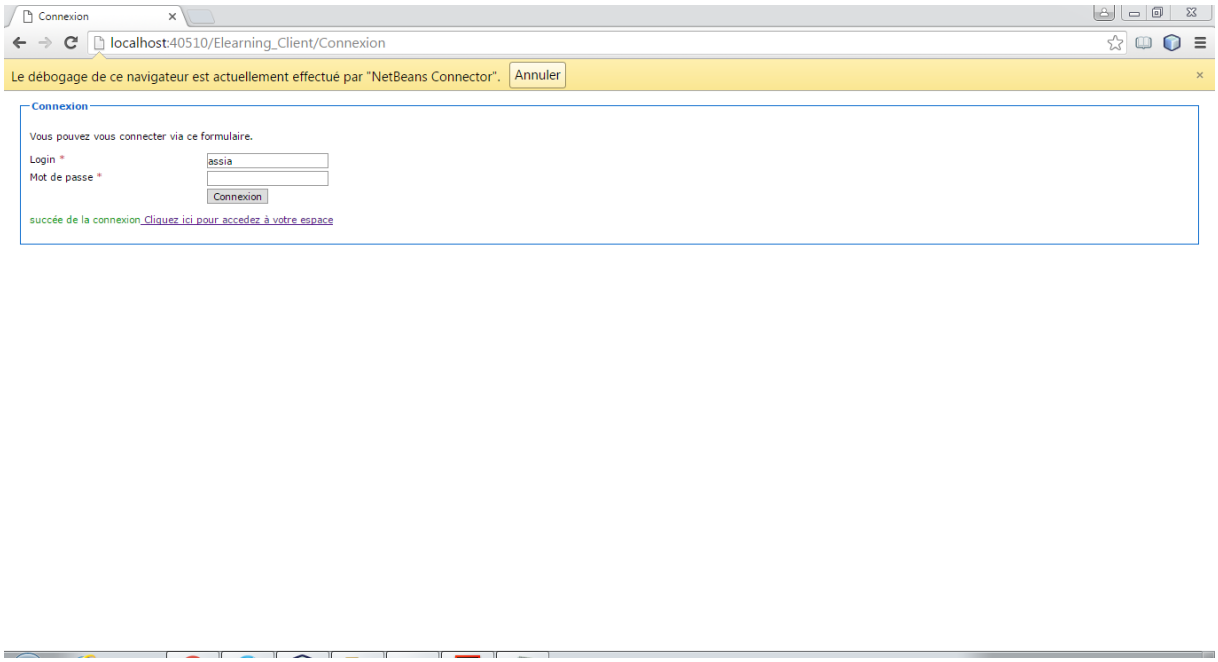
FigureIV.9:*Présentation graphique de la page Succès d'Inscription.*

IV.4 .3 *Formulaire de Connexion*

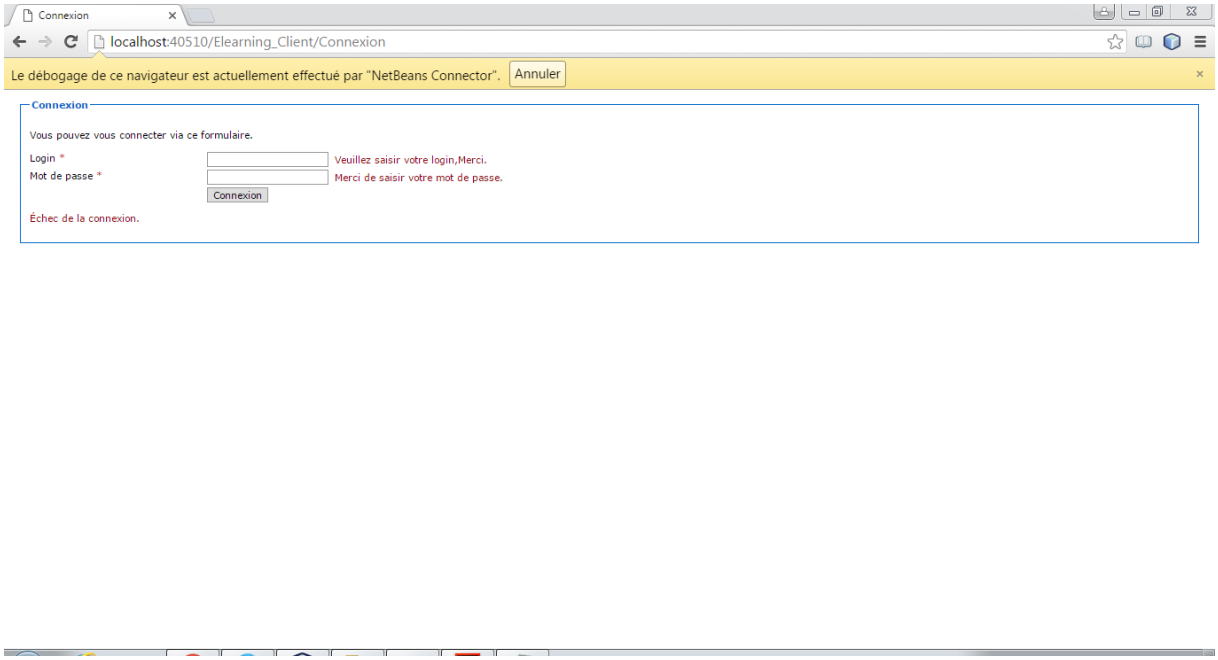
Avant d'entrer dans chaque espace, l'utilisateur doit s'authentifier en saisissant un login et un mot de passe.



FigureIV.10: *Présentation graphique de la page de Connexion.*



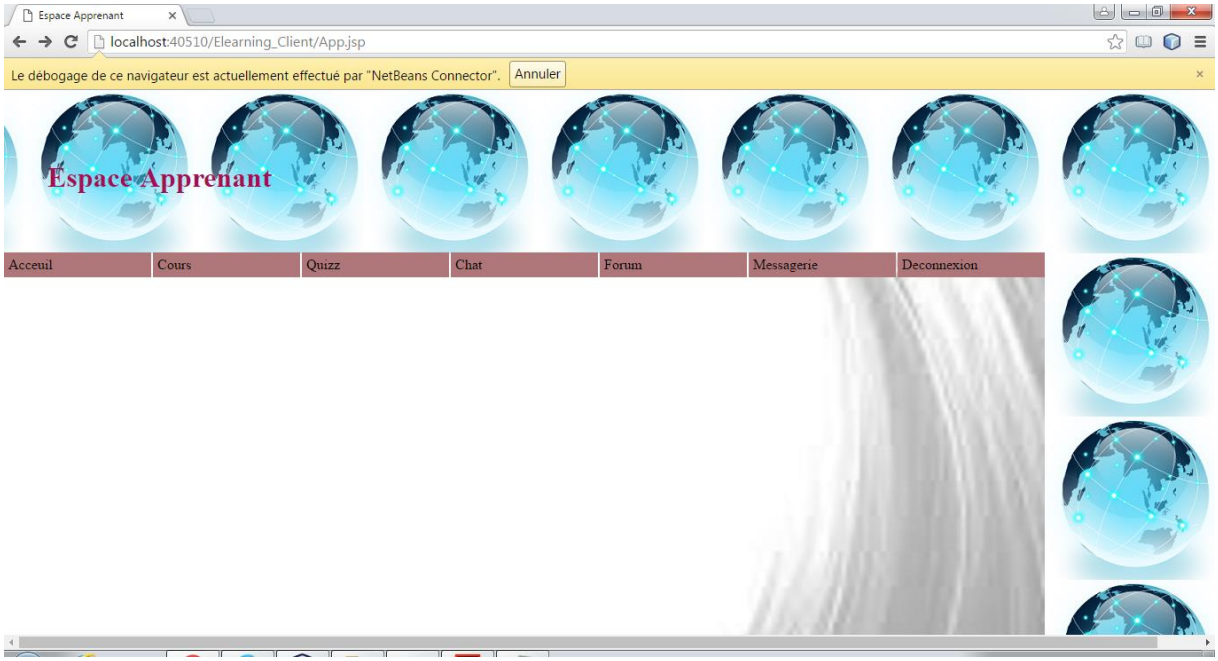
FigureIV.11: ***Présentation graphique de la page Succès de Connexion.***



FigureIV.12:*Présentation graphique de la page Echec de Connexion.*

IV.4 .4 Espace Apprenant

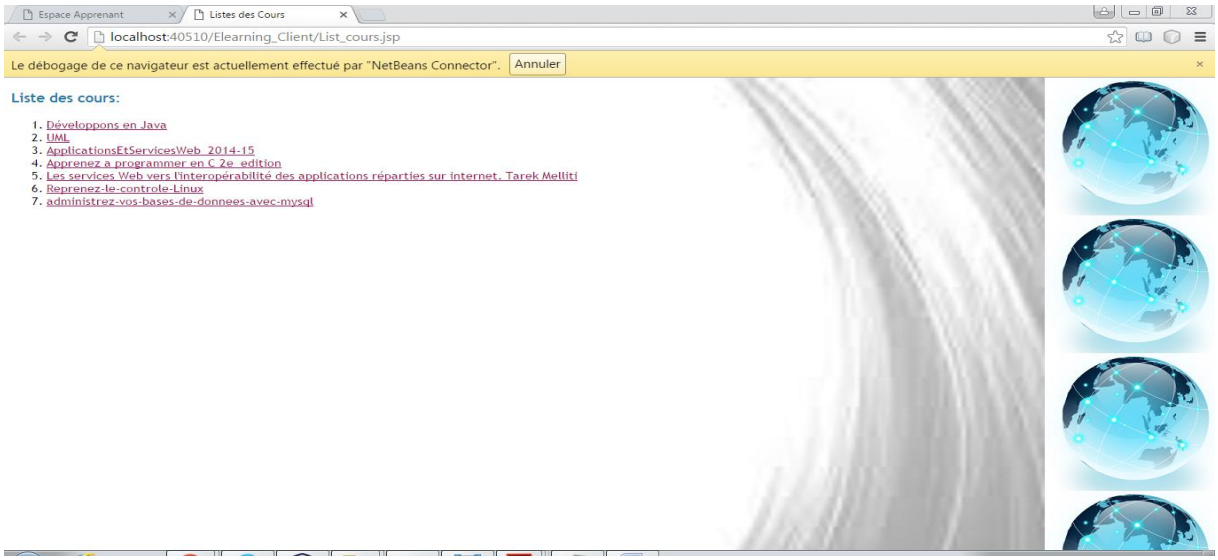
Après l'authentification, l'espace Apprenant s'affiche, il comporte toutes les taches que l'apprenant peut effectuer.



FigureIV.13:*Présentation graphique de la page Apprenant.*

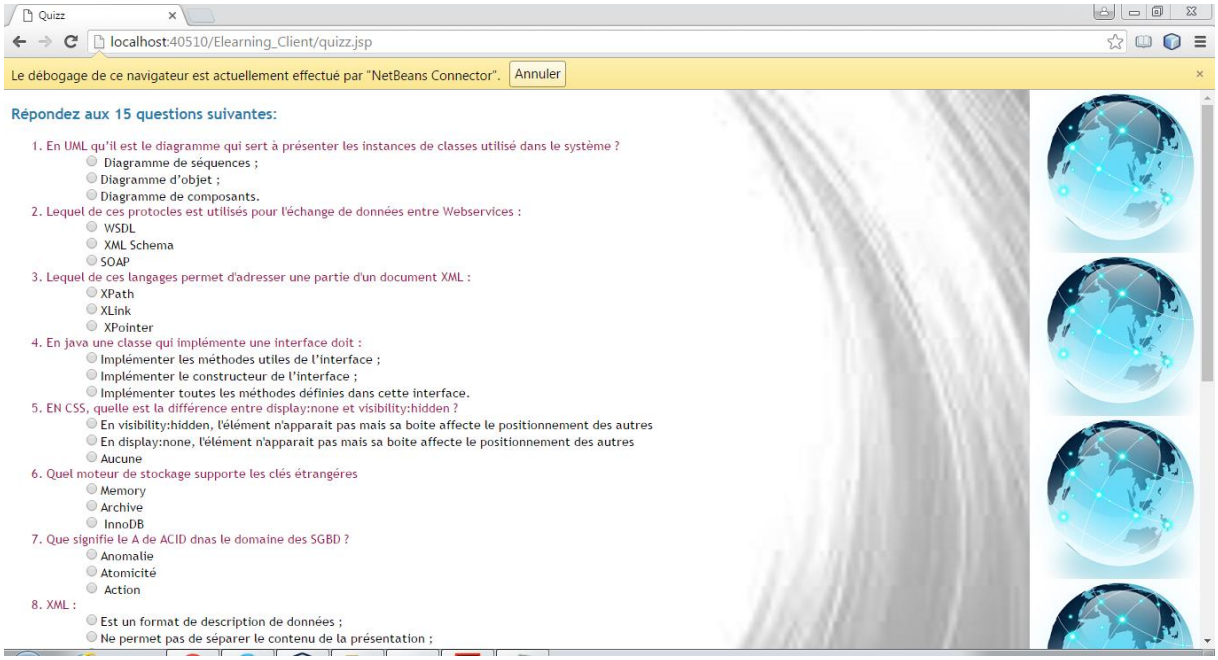
IV.4 .4.1 Page Liste de Cours

Cette page affiches la listes des cours ajouté par les enseignants de la plate forme.



FigureIV.14:*Présentation graphique de Liste des Cours*.

IV.4 .4.2Page Quizz

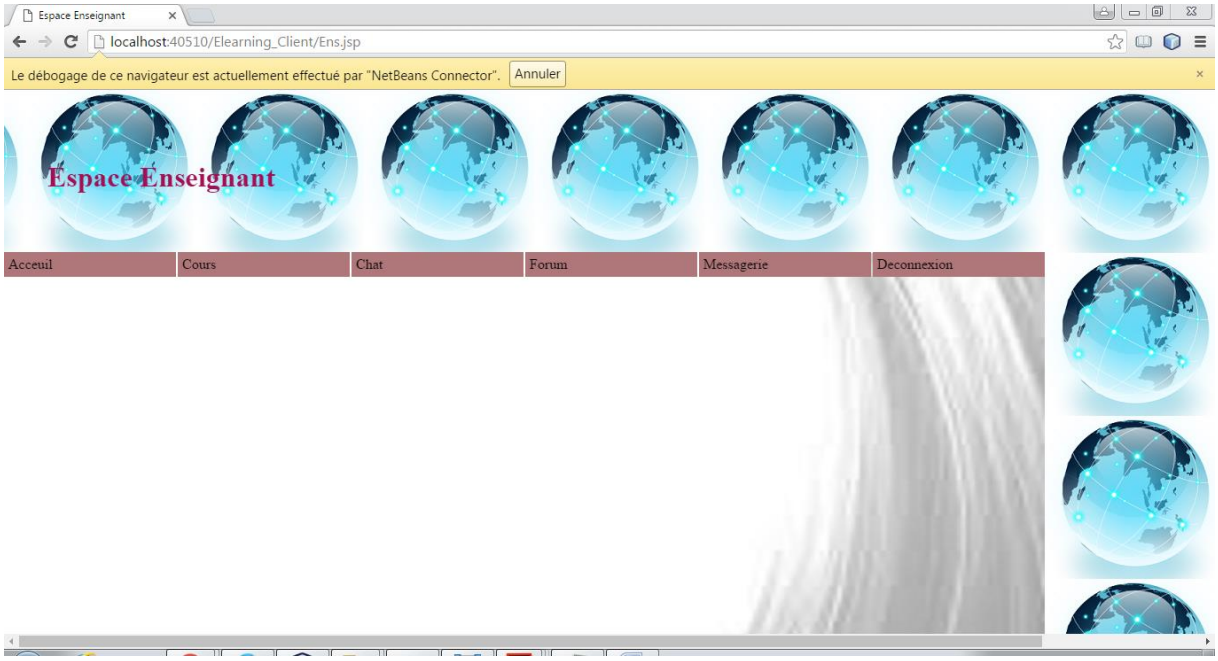


.

FigureIV.15:*Présentation graphique de quizz.*

IV.4 .5 Espace Enseignant

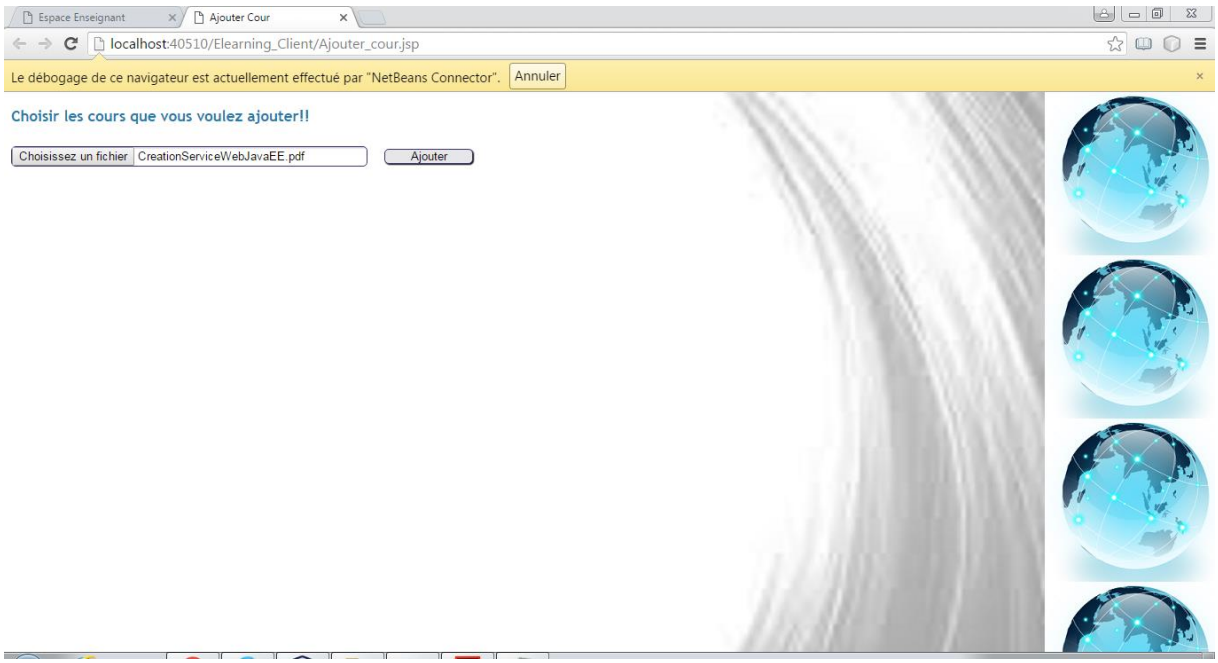
Après l'authentification, l'espace Enseignant s'affiche ,il comporte toutes les taches que l'enseignant peut effectuer.



FigureIV.16:*Présentation graphique de la page d'Enseignant.*

IV.4 .5.1 Page d'Ajout de cours

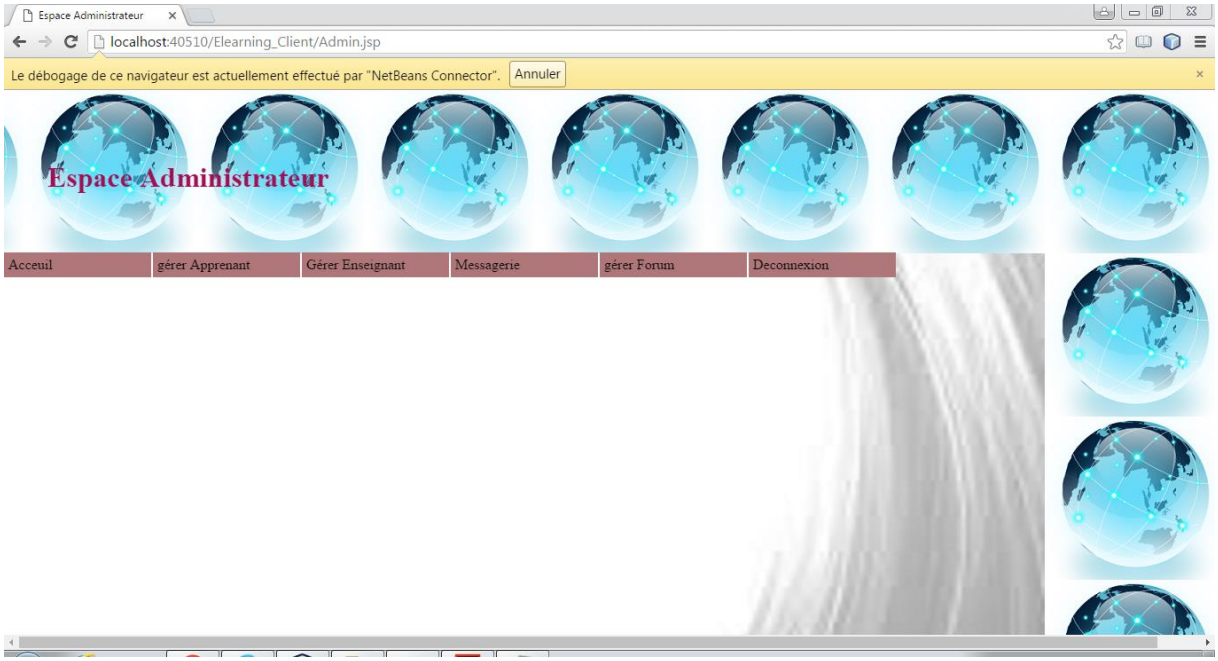
Cette page permet a l'enseignant d'ajouter des cours.



FigureIV.17:*Présentation graphique de la page Ajouter Cours.*

IV.4 .6 Espace Administrateur

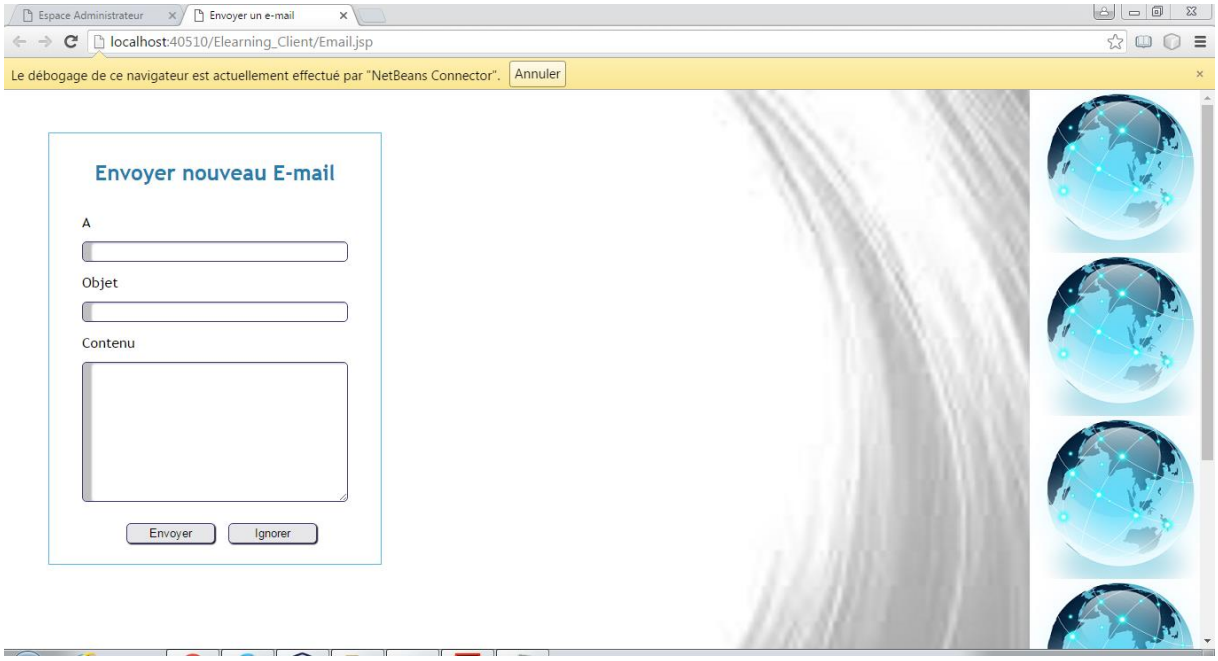
Après l'authentification, l'espace Administrateur s'affiche ,il comporte toutes les taches que l'administrateur peut effectuer.



FigureIV.18:*Présentation graphique de la page d'Administrateur.*

IV.4 .6 .1 Page Nouveau Message

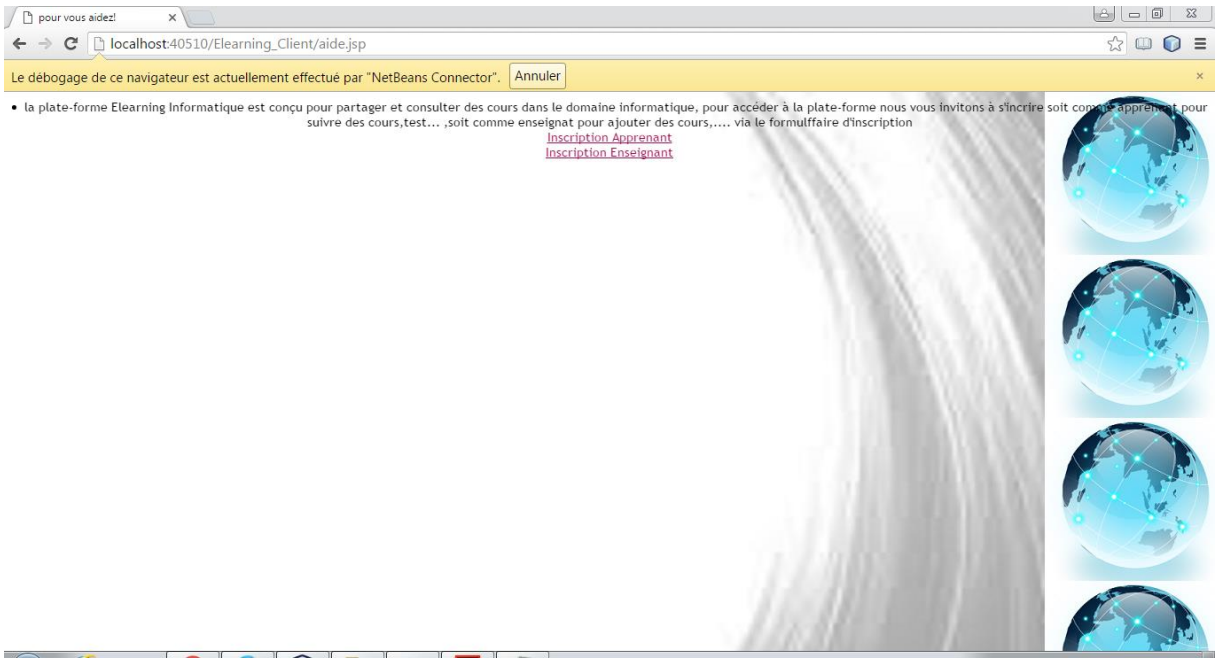
Cette page permet d'envoyer un nouveau message.



FigureIV.19:*Présentation graphique de la page Nouveau Message.*

IV.4 .7 Page Pour Vous Aidez

Page pour information sue la plate-forme.



FigureIV.20:*Présentation graphique de la page Pour Vous Aidez.*

IV.5 Conclusion

Ce chapitre présente l'implémentation de notre application. Nous avons tout d'abord cité les différents outils utilisés lors du développement de notre système,nous avons ensuite présenté les service web de notre application ainsi que la partie client qui invoque ces services web.Nous avons aussi décrit quelques interfaces de notre plate-forme.

Conclusion Générale

Aujourd'hui, l'interopérabilité est devenue un domaine de recherche fondamental des systèmes d'information distribués et hétérogènes. Les services Web sont considérés comme une solution potentielle aux problèmes d'interopérabilités. Ils définissent un nouveau paradigme de développement des interactions entre des applications distribuées de manière à ce qu'elles restent indépendantes des environnements et des plateformes d'exécution d'une part et aussi des choix des langages de développement et technologies d'implémentations utilisés d'une autre part. Malgré les avantages indéniables qu'apportent les services Web au niveau des problèmes d'interopérabilité,

Dans ce mémoire, nous avons réaliser une plate-forme d'e-learning en utilisant la technologie des services Web, et plus précisément les Services Web de type SOAP, permettant :

- La gestion administrative de la formation.
- La communication entre enseignants et apprenants et entre apprenants.
- L'ajout des cours par les enseignants.
- La recherche et la consultation des cours par les apprenants.

Notre travail est constitué en trois parties importantes :

- La première partie, consiste en la présentation de certaines définitions nécessaires dans le domaine d'apprentissage électronique, son évolution, une présentation générale des services web SOAP, la technologie sur laquelle nous nous sommes basés pour la réalisation de notre projet.
- La deuxième partie, la conception, dans laquelle nous avons présenté la conception de notre application en utilisant le langage UML(processus RUP).
- La troisième, est la dernière partie est destinée à la phase d'implémentation et de mise en œuvre des différents modules et services qui constituent notre application.

Et comme ce travail étant une œuvre humaine, n'est pas un modèle unique et parfait, c'est pourquoi nous restons ouverts à toutes les critiques et nous sommes prêts à recevoir toutes les suggestions et remarques tendant à améliorer d'avantage cette étude. Était donné que tout travail informatique a été toujours l'œuvre d'une équipe.

1.Introduction

En informatique **UML** (de l'anglais *Unified Modeling Language*), ou Langage de modélisation unifié, est un langage de modélisation graphique à base de pictogrammes. Il est utilisé en développement logiciel, et en conception orientée objet. UML est couramment utilisé dans les projets logiciels.
UML est l'accomplissement de la fusion de précédents langages de modélisation objet :Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh etIvar Jacobson, UML est à présent un standard défini par l'Object Management Group (OMG). La dernière version diffusée par l'OMG est UML 2.5 bêta 2 depuis septembre 2013.

2.Utilité d'UML

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle. Les différents éléments représentables sont :

- Activité d'un objet/logiciel
- Acteurs
- Processus
- Schéma de base de données
- Composants logiciels
- Réutilisation de composants

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement une partie de code, par exemple en langage Java, à partir des divers documents réalisés.

3. Histoire

Les méthodes objets ont commencé à émerger au début des années 80, ces méthode savaient pour but de remplacer les méthodes structurée et fonctionnelles, trop liés à la machine. Plusieurs méthodes(une cinquantaine) objets ont fait leur apparition au début des années 90. Ces méthodes s'orientant sur l'abstraction des composants matériels, se basent sur des notions de classe, d'association, de partition en sous-système et autour de l'étude de l'interaction entre utilisateur et le système. Les principaux auteurs de ces méthodes sont James Rumbaugh,Grady Booch et Ivar Jacobson. Parmi ces méthodes, deux se sont principalement établies : la méthode de Booch et la méthode OMT(Object Modeling Technique). Une deuxième version des méthodes de Booch et OMT ont fait leur apparition Booch'93 et OMT-2. Ces méthodes sont fortement semblables, en effet Booch'93 se concentre plus sur la construction tandis que OMT-2 se concentre sur l'analyse et l'abstraction.

4.Le formalisme d'UML

UML 2.3 propose 14 types de diagrammes (9 en UML 1.3). UML n'étant pas une méthode, leur utilisation est laissée à l'appréciation de chacun, même si le diagramme de classes est généralement considéré comme l'élément central d'UML ; des méthodologies, telles que l'Unified Process, axent l'analyse en tout premier lieu sur les diagrammes de cas d'utilisation (Use Case). De même, on peut se contenter de modéliser seulement partiellement un système, par exemple certaines parties critiques.

UML se décompose en plusieurs sous-ensembles :

- **Les *vues* :** Les vues sont les observables du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes ces vues, il est possible de définir (ou retrouver) le système complet.
- **Les *diagrammes* :** Les diagrammes sont des éléments graphiques. Ceux-ci décrivent le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues.
- **Les *modèles d'élément* :** Les modèles d'élément sont les briques des diagrammes UML, ces modèles sont utilisés dans plusieurs types de diagrammes. Exemple d'élément : cas d'utilisation (*CU* ou *cadut'*), classe, association, etc.

Mise en oeuvre d'une démarche à l'aide d'UML : les vues

Une façon de mettre en oeuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système :

- Vue des ***cas d'utilisation*** : c'est la description du modèle vu par les acteurs du système. Elle correspond aux besoins attendus par chaque acteur (c'est le QUOI et le QUI).

- Vue ***logique*** : c'est la définition du système vu de l'intérieur. Elle explique comment peuvent être satisfaits les besoins des acteurs (c'est le COMMENT).
- Vue ***d'implémentation*** : cette vue définit les dépendances entre les modules.
- Vue des ***processus*** : c'est la vue temporelle et technique, qui met en oeuvre les notions de tâches concurrentes, stimuli, contrôle, synchronisation, etc.
- Vue de ***déploiement*** : cette vue décrit la position géographique et l'architecture physique de chaque élément du système (c'est le OÙ).

5.Les diagrammes

Les **14** diagrammes UML sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.

5.1-Diagrammes structurels ou statiques

Les diagrammes structurels ou statiques (Structure Diagram) rassemblent :

- Diagramme de **classes** (Class diagram) : il représente les classesintervenant dans le système.
- Diagramme d'**objets** (Object diagram) : il sert à représenter les instances de classes (objets) utilisées dans le système.
- Diagramme de **composants** (Component diagram) : il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en oeuvre (fichiers, bibliothèques, bases de données…)
- Diagramme de **déploiement** (Deployment diagram) : il sert à représenter les éléments matériels (ordinateurs,périphériques, réseaux, systèmes de stockage…) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.
- Diagramme des **paquetages** (Package diagram) : un paquetage étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, le diagramme de paquetage sert à représenter les dépendances entre paquetages, c'est-à-dire les dépendances entre ensembles de définitions.
- Diagramme de **structure composite** (Composite Structure Diagram) : depuis UML 2.x, permet de décrire sous forme deboîte blanche les relations entre composants d'une classe.
- Diagramme de **profils** (Profile diagram) : depuis UML 2.2, permet de spécialiser, de personnaliser pour un domaine particulier un meta-modèle de référence d'UML.

5.2-Diagrammes comportementaux

Les diagrammes comportementaux (Behavior Diagram) rassemblent :

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

- Diagramme **états-transitions** (State Machine Diagram) : permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants.

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

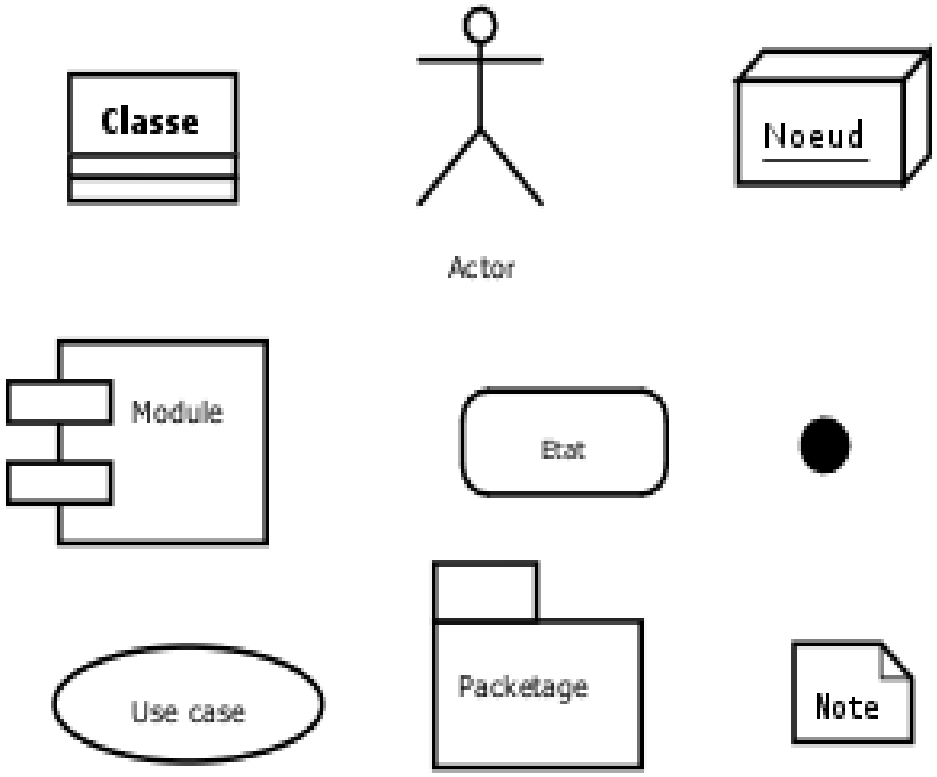
5.3-Diagrammes d'interaction ou dynamiques

- Les diagrammes d'**interaction ou dynamiques** (Interaction Diagram) rassemblent :
- Diagramme de **séquence** (Sequence Diagram) : représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.
- Diagramme de **communication** (Communication Diagram) : depuis UML 2.x, représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.
- Diagramme **global d'interaction** (Interaction Overview Diagram) : depuis UML 2.x, permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante dudiagramme d'activité).
- Diagramme de **temps** (Timing Diagram) : depuis UML 2.3, permet de décrire les variations d'une donnée au cours du temps.

6-Les éléments de modélisation

- Le ***Stéréotype*** est une marque de généralisation notée par des guillemets, cela montre que l'objet est une variété d'un modèle.
- Le **classeur** est une annotation qui permet de regrouper des unités ayant le même comportement ou structure. Un classeur se représente par un rectangle conteneur, en traits pleins.
- Un **paquetage** regroupe des diagrammes ou des unités.
- Chaque classe ou objet se définit précisément avec le signe « :: », ainsi l'identification d'une Classe X en dehors de son package ou de son classeur sera définie par « Package A::Classeur B::Classe X ».

6.1-Les éléments de modélisation de type commun



Modèles d'éléments UML.

7.Conclusion

L'UML, comme l'on a vu, ne propose pas une démarche objet mais une notation adapté au monde de développement orienté objet. Il nous a donc permis de s'initier aux techniques de modélisation objet. La notation UML peut s'adapter a tous les projets informatiques.

1.Introduction

Processus unifié (PU ou UP en anglais pour unified process) est une méthode de développement pour les logiciels orientés objets. C'est une méthode générique, itérative et incrémentale, contrairement à la méthode séquentielle Merise (ou SADT).

PU vient compléter la systémique des modèles UML. Elle est le résultat final d'une évolution de l'approche d'Ericsson qui est au fondement d'une des premières méthodes de développement pour applications orientées objets, la méthode Objectory Process (1987). Objectory Process (version 1 à 3.8 en 1995) a elle-même servi de base à la société Rational pour la création de Rational Objectory Process (1997) (version 4.1), parente direct de RUP en 1998.

2.RUP

Le Rational Unified Process (RUP) est l'une des plus célèbres implémentations de la méthode PU, livrée clés en main, permettant de donner un cadre au développement logiciel, répondant aux exigences fondamentales préconisées par les créateurs d'UML :

- une méthode de développement doit être guidée par les besoins des utilisateurs
- elle doit être centrée sur l'architecture logicielle
- elle doit être itérative et incrémentale

2.1 Méthode de développement logiciel

- itérative,
- Incrémentale
- pilotée par les cas d'utilisation.
- centrée sur l'architecture et la réduction des risques
- Produit de qualité

2.2.Phases de RUP

2.2.1.Phase d'initialisation

- Définir la portée du projet
- Spécification
- Affectation des tâches
- Evaluation des risques
- Cycle de vie

2.2.2.Phase d'élaboration

- Planifier le projet,
- spécifier les fonctionnalités
- construire l'architecture

- Spécification du produit
- Conception de l'architecture
- Planification activités-ressources

2.2.3. Phase de construction

- Construire le produit
- Implémentation du produit
- Tests
- produit opérationnel

2.2.4. Phase de transition

- Transition du produit vers les utilisateurs
- Livraison
- Formation
- Qualité

2.3.Cycle de vie de RUP

Exigence :détermination des besoins :

- fonctionnels (ce que l'on attend du système)
- non fonctionnels (fiabilité, temps de réponse, environnement distribué, etc.)

Analyse et conception : évoluer depuis la spécification des besoins jusqu'à une solution informatique

- analyse besoins fonctionnels
- conception intègre aussi les besoins non fonctionnels

Implémentation :

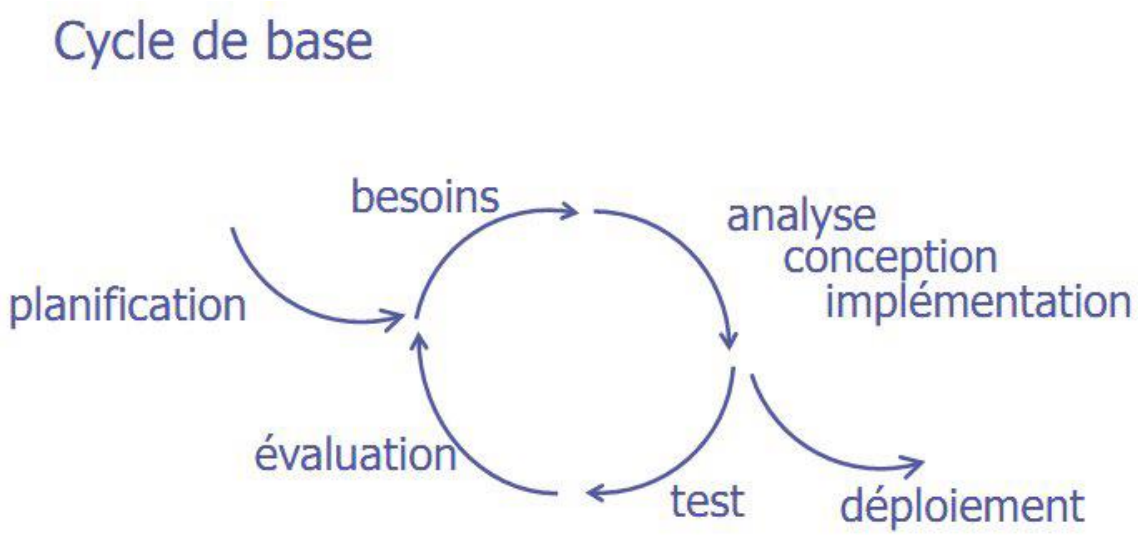
- Transcription dans un langage de programmation ou de base de données .
- Utilisation de composants existants.

Test :

Estimer si les besoins sont satisfaits. S'il y a des erreurs/défauts à corriger Renforcer et stabiliser l'architecture.

Déploiement :

Distribuer le logiciel dans son environnement opérationnel, Installation,
test Formation des utilisateurs, Migration des données.



Cycle de vie de RUP

2.4. Différence entre incrémental et itératif

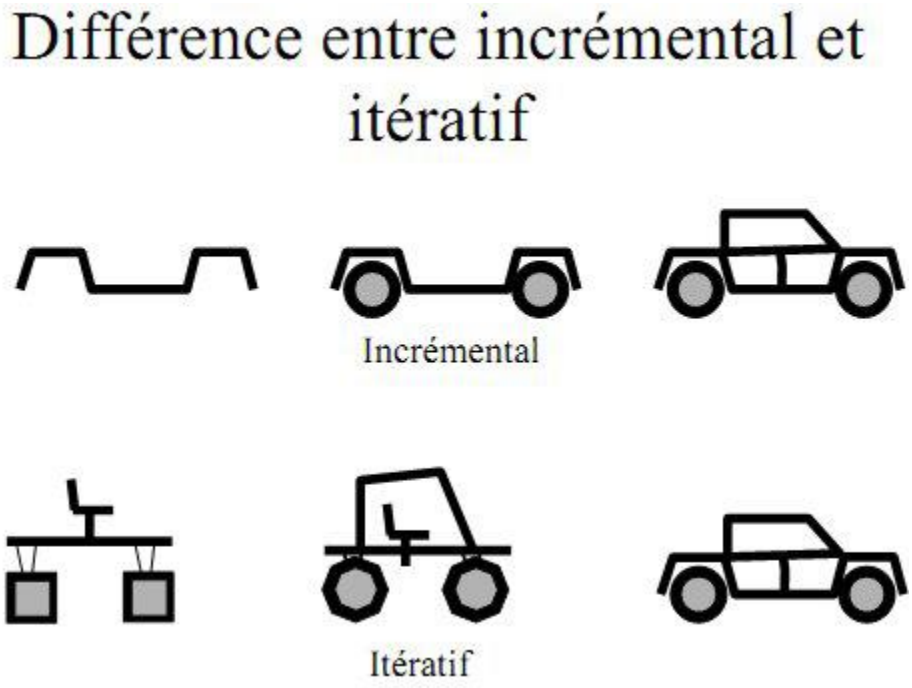
✓ RUP est incrémental

• Le produit final est livré en incréments
  – Chaque incrément livre une fonctionnalité ayant une valeur pour l'utilisateur final
  – Chaque incrément consiste en un ensemble de cas d'utilisation
  – Les incréments doivent être de courte durée (quelques semaines à quelques mois)
  – Les incréments sont des mini-projets dans un projet

✓ RUP est itératif

Une itération est une séquence d'activités qui répond à un plan et à des critères d'évaluation et qui produit une version exécutable

Exemple



2.5 Forces et Faiblesse de RUP

✓ Forces de la méthode RUP

Traçabilité à partir des Uses Cases jusqu'au déploiement
Approche basée sur l'architecture
Gestion des risques dans les projets

× Faiblesses de la méthode RUP

Coût de personnalisation souvent élevés
Très axé processus

3.Conclusion

Cette partie nous a permis de prendre connaissance le processus rup (rationnel unified process) est une méthode de développement pour logiciel orienté objet, c'est une méthode générique itérative et incrémentale.

Références Bibliographiques et webographiques:

[Boubekeur,Malik,Chouarfia,08]: Boubekeur.A ,Malki.M, Chouarfia.A, «Enrichissement sémantique d'une architecture orientée»,Éditions Hermès,27 nov. 2008.

[AWT, 08] : Agence Wallonne des Télécommunications , la plateforme ICT de la Wallonie , « guide e-Learning : qu'est-ce que l'e-learning ? »,2008

[Charroux,Osmani,Thierry-Mieg,10]: Benoît Charroux, Aomar Osmani, Yann Thierry-Mieg, « UML2», Édition Pearson,17 juin 2010.

[Piette,06] : Benoit Piette, « Introduction aux Web Services », éditions Dunod,2006.

[Taconet,11] : Chantal Taconet, « Simple Object Access Protocol », Edition Évry, France,2011.

[MIELNIKOFF,05] : Michel MIELNIKOFF, « Qu'est-ce que l' E-Learning ? » ,Edition Owanis, 07 novembre 2005.

[Stattner,14] : Erick Stattner,« Applications et Services WEB Applications Laboratoire LAMIA »,Université des Antilles et de la Guyane, 2014 – 2015

[van der Vlist,04] : Eric van der Vlist,« Le triptyque SOAP/WSDL/UDDI,Web Services Convention »,Edition Microsoft,Juin 2004.

[Fenouillet,04] : Fabien Fenouillet, «Le e-learning est-il efficace ? »,*Savoirs* ,2006 .

[Larrey,05] : François Larrey, « E-LEARNING SES FONDEMENTS ET SON UTILISATION DANS LE SECTEUR BANCAIRE »,Edition Genève, 26 août 2005 .

[Mili,05]: Hafedh Mili, «Rational Unified Process»,Rational Software White Pape ,2005.

[Schwichtenberg,Trottenberg,11]: Horst Schwichtenberg ,Ulrich Trottenberg, «Today's Heterogeneous IT Environments»,Edition Microsoft, 2011.

[Jézéquel,02] : Jean-Marc Jézéquel,« Analyse par Objets avec UML(Unified Modeling Language)»,HAL,2002.

[Henocque,06] : Laurent Henocque,« Le Processus Unifié de Rational »,Eyrolle,Novembre 2006.

[LeBlanc,02] : Michel LeBlanc,« Les web service »,Edition Cirano,14 novembre 2002.

[Winter,12] : Michel Winter , « MÉTHODE DE DÉVELOPPEMENT RUP»,2012.

[COTTIN,] : Natha naël COTTIN, Java Enterprise Edition.

[Denis,02] :*Philippe Denis, «UML/RUP : Organisation des processus de développement », 4 décembre 2002*.

[Shodjai,06] : Payam Shodjai, « Services Web et la plate-forme Microsoft » , Microsoft Corporation,25 juillet 2006.

[Melliti,03] : Tarek Melliti, « Les services Web vers l'interopérabilité des applications réparties sur internet »,Edition Scholar,2003.

[Kassem, Mounajed, Saadoun,04] : Walid Kassem, Ahmad Mounajed, Nadia Saadoun, « Etat de l'Art du E-Learning » ,Edition Eyrolle,16 février 2004.

[W3C, 04] : « Web Services d'architecture »,Groupe de travail du W3C,11 Février 2004.

[IBM,00]: « Web Services, Présentation de l'architecture », IBM, Software Group,06 Septembre 2000.

https://fr.wikipedia.org/wiki/Interopérabilité
www.editions-ellipses.fr/PDF/9782729865597_extrait.pdf
www.efrontlearning.net.
www.has-sante.fr
http://www.developpez.com.
https://openclassrooms.com.