

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche
Scientifique UNIVERSITE MOULOUD MAMMERI DE TIZI-
OUZOU



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etude
DE MASTER PROFESSIONNEL

Spécialité : Ingénierie des systèmes d'informations

Thème :

Services web et interopérabilité des applications
Cas : E-Learning

Présenté par :
Meziane FRAGA
Nabil SALAH

Dirigé par :
M'hand KERBICHE

Mémoire soutenu le 31 juillet 2017 devant le jury composé de :

Mr Mohamed RAMDANE
Mr Rabah SOUALAH

Remerciements

Nous voudrions présenter nos remerciements à notre encadreur monsieur KERBICHE

Nous voudrions également lui témoigner notre gratitude pour sa patience et son soutien qui nous a été précieux afin de mener notre travail à bon port.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

Enfin, on remercie tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail

DEDICACES

Je dédie ce modeste travail à :

Mes parents qui m'ont toujours soutenu et encouragé

Mes frères et sœurs pour leurs conseils

Ma femme pour m'avoir accompagné et soutenu

Tous mes amis pour leur compagnie

Mon binôme et ami Nabil et sa famille,

Enfin à tous ceux qui me sont cher et qui m'ont aidé de près ou de loin.

Meziane

DEDICACES

Jamais cette soutenance n'aurait vu le jour, sans l'amour et l'affection que j'ai pu trouver au sein de ma mère dès mon enfance...

A ma famille qui m'a encouragé pendant les moments difficiles...

A mes frères et sœurs pour leurs soutiens ...

A ma future femme

A mes petite neveux zimo et imane

A mes amis

A mon binôme et frère meziane

A ma grande mère... je t'aime...

Nabil

Sommaire	
Introduction générale.....	9
Chapitre 1 : Les services web.....	10
INTRODUCTION.....	11
I-1- Historique	11
I- 2- Interopérabilité et intégration système	12
I-2-1- Définition de l'interopérabilité.....	12
I-2-3- Définition d'intégration systèmes :	13
I-3- Définition du service web	13
I-4- L'architecture web services	14
I-5- Fonctionnement d'un service web.....	15
I-6- Technologies services web.....	15
I-7- Protocoles utilisés.....	16
I-7-1 SOAP (Simple Object Access Protocole) :.....	16
I-7-2 WSDL (Web Services Description Langage).....	21
I-7-2-1- Structure d'un document WSDL.....	21
I-7-3 UDDI (Universal Description, Discovery and Integration).....	23
I-7-4 XML (Exetensible Markup Language).....	24
I-8- Rest (REpresentational State Transfert).....	24
I-8-2 Les identifiants de REST	24
I-8-3 JAX-RS (Java Api for restfull services)	25
8-3-1 Définition JAX-RS	25
I-9- Avantages et inconvénients SOAP et REST	25
I-9- Conclusion.....	26
Chapitre 2 :	27
L'e-Learning.....	27
I-1- Introduction :	28
I-2- Historique :	28
I-3Définition :	28
I-4-Les acteurs et leur rôle dans le e-Learning :	29
I.1.1 Apprenant :	30
I-1-2 Enseignant :	30
I-1-3-Administrateur :	31
I-5- Les plateformes e-Learning :	31
I-6-Modes de diffusion :	33

I-6- Différents modèles du e-Learning :	34
I-7- Approches pédagogiques en E-Learning	36
I-8- Les logiciels de lecture multimédias :	36
I-9- L'e-Learning en Algérie :	37
II-10- Comparaison du e-Learning et de la formation traditionnelle : [5]	37
II-11-Avantages et inconvénients du e-Learning: II-11-1- Avantages :	38
II-11-2- Inconvénients :	39
II-12- Conclusion :	39
Chapitre III :	40
Conception	40
III-1- Introduction	41
III-2- Problématique	41
III-3- Objectif :	41
III-4- Spécification des exigences	42
III-4-1 Etude des besoins fonctionnels	42
III-4-2- Etude des besoins non fonctionnels	42
III-5- Les acteurs système	42
III-6- Identification et structuration de cas d'utilisation :	43
III-6-1- Définition :	43
III-6-2 Diagramme cas utilisation « authentification »	43
III-6-3 Diagramme cas utilisation « rôle apprenant »	44
III-6-4 Diagramme cas utilisation « gérer cours »	45
III-6- 5 - Diagramme cas utilisation « gérer plateforme »	46
III-6- 6 : Diagramme cas utilisation général	48
III-7- Diagrammes de séquence :	48
III-7-1 Définition	48
III-7-2- Diagramme de séquence cas « authentification »	49
III-7-3 Diagramme de séquence cas « inscription apprenant »	50
III-7-4 Diagramme de séquence cas « consulter cours »	51
III-7-5 Diagramme de séquence cas « effectuer examen »	52
III-8- Diagrammes de séquence d'analyse détaillée des cas d'utilisation :	53
III-9- Diagramme d'activité	54
III-9-1 Diagramme d'activité cas « Gestions utilisateurs »	55
III-9-2 Diagramme d'activité cas « gestion cours »	56
III-10- Diagramme de classe	56
11- Conclusion	57
Chapitre IV	58
Réalisation	58

IV-1- Serveur de base de données.....	59
IV-1-1 PostgreSQL.....	59
IV-2- Draw.io.....	59
IV-3- Netbeans.....	60
IV-4- Framework de développement JSF 2.2 (Java Server Face).....	60
IV-5- Service Web rest.....	61
IV-5-1- Création d'un service REST sous Netbeans.....	61
2-3- Création d'un modèle.....	67
2-4- Modification du service REST.....	67
2-5- Invocation du service REST.....	68
VI-6 Interfaces de la plate-forme.....	70
Conclusion générale.....	74
Bibliographie :.....	75

Liste des figures

Chapitre I

Figure 1 – architecture services web	11
Tableau 1 : protocoles des services web.....	14
Tableau 3 : bloc racine message XML.....	14
Figure 4 : Entête d'un message XML.....	14
Figure2 : structure d'un message SOAP.....	16
Tableau 2 : erreurs reportés d'un message SOAP	17
Tableau 4 : opérations d'un service web dans l'élément PorType	20
Tableau 5 : Identifiants Rest.....	22
Tableau 6 : Avantage et inconvénients SOAP et REST	24

Chapitre II

Figure 1 : le Système E-Learning	27
Figure 2 : LCMS (LMS+CMS)	31
Tableau1: Différents modèles selon Bersin.....	33
Tableau2 : Comparaison des avantages et inconvénients du e-Learning de la formation traditionnelle.....	34

Chapitre III

Figure 2 : Diagramme de cas utilisation « Authentification.....	40
Figure 3 : Diagramme de cas utilisation « rôle apprenant ».....	41
Figure 4 : Diagramme cas utilisation « gérer cours ».....	43
Figure 5 : Diagramme cas utilisation « gérer plateforme »	44
Figure 6 : Diagramme cas utilisation général.....	46
Figure 8 : Diagramme de séquence cas « authentification »	47
Figure 9 : Diagramme de séquence cas « inscription apprenant »	50
Figure 10 : Diagramme de séquence « consulter cours ».....	51
Figure 11 : Diagramme de séquence « Effectuer examen ».....	52
Figure 12 : diagramme de séquence détaillé « Inscription ».....	55

Introduction générale

L'Internet a révolutionné le monde des ordinateurs et des communications comme rien d'autre auparavant. En effet, il est à la fois une capacité de diffusion dans le monde entier, un mécanisme de distribution de l'information et un moyen de collaboration et d'interaction entre les individus et leurs ordinateurs, peu importe l'emplacement géographique.

L'internet est aujourd'hui utilisé dans plusieurs milieux notamment les milieux éducatifs, et avec l'apparition des nouvelles technologies, on a connu une nouvelle forme d'apprentissage rapide et efficace qui permet à une ou plusieurs personnes de se former à partir de leurs ordinateurs.

Le système de formation traditionnelle est conditionné par l'unité de temps et de l'espace qui semble être insuffisant pour répondre à tous les besoins de ceux qui souhaitent suivre une formation, et afin d'y remédier aux problèmes de déplacement et de temps posés dans la formation traditionnelle, nous avons décidé de créer une application E-Learning afin d'offrir l'occasion aux étudiants de suivre des cours en particulier ceux d'anglais.

Néanmoins, avec la multitude de systèmes (linux, android, windows ...) communiquant avec internet, un problème d'interopérabilité s'impose, et c'est pour cela que nous avons opté pour l'utilisation des web services permettant d'échanger des données de manière textuelle, afin que l'application appelée puisse intégrer le résultat de l'échange, les requêtes et les réponses sont soumises à des standards et des normes.

Notre travail est organisé en quatre chapitres à savoir :

Chapitre 1 : Les services web

Chapitre 2: L'e-Learning

Chapitre 3 : Analyse et conception

Chapitre 4 : Réalisation

Chapitre 1 :

Les services web

INTRODUCTION

Une application doit répondre à certaines exigences techniques tel que la disponibilité, le temps de réponse, sécurité, et la distribution, cette dernière doit être distribuée sur plusieurs machines, en d'autres termes c'est de passer la POO (Programmation orientée objet) vers POOD (programmation orientée objet distribuée), c'est de faire appel à des fonctionnalités à distance en simplifiant ainsi l'échange des données pour cela on fait appel aux web services. Les web services sont des composants web basés sur Internet (http) qui exécutent des tâches précises et qui respectent un format spécifique (XML), ils permettent aux applications de dialoguer à travers le réseau, indépendamment de leur plateforme d'exécution et de leur langage d'implémentation

Ils s'inscrivent dans la continuité d'initiatives telles que COBRA (Common Object Request Broker Architecture, de L'OMG) en apportant toutefois une réponse plus simple s'appuyant sur des technologies et standards reconnus et maintenant acceptés de tous.

Dans ce chapitre nous présenterons l'architecture orientée service et ses standards (SOAP, WSDL, UDDI, XML), ainsi que le fonctionnement des web services.

I-1- Historique

Le concept de deux ordinateurs communiquant l'un avec l'autre pour partager des informations ou échanger des données a été inventé bien avant que internet fut inventé, une des premières mises en œuvre d'architecture de ce style est connue sous l'acronyme EDI « échange de données informatisées ».

EDI est un système d'échange informatique de messages fortement formatés qui représentent des documents autres que les instructions monétaires. [1] En d'autres termes EDI a été inventé pour transcender les transactions financières et pour permettre le mouvement de données sur internet, à l'origine il a été conçu pour le commerce électronique à grande échelle. Au début des années 1990 une nouvelle norme a émergé connue sous le nom de RPC « Appel procédure lointaine », cette norme est basée sur autre norme connue comme un environnement informatique distribué. Les systèmes RPC permettent aux ordinateurs clients d'appeler des fonctionnalités ou des sous programmes sur des ordinateurs lointains en utilisant une syntaxe aussi semblable que possible au code qui pourrait être utilisé localement.

Au milieu des années 90 de nombreux efforts ont été faits pour simplifier la technologie, la vraie période très intense dans l'informatique a distance commence en 1991 jusqu'au 1996, avec une douzaine de différentes normes mais qui étaient complexes qui les a rendu difficile à utiliser.

En 1998, le monde des services web a été révolutionné avec l'introduction d'un nouveau format le « XML » qui était tiré d'un autre langage, le « SGML », qui a pour principal but la simplicité, lisibilité.

Le XML était vite devenu le format standard pour les messages utilisés par les services web. Les services web ont une histoire riche de changement et d'évolution.

I- 2- Interopérabilité et intégration système

Certains systèmes pour des problèmes d'interopérabilité de causes diverses, sont appelés à migrer vers de nouvelles versions. L'objectif étant de rendre le système le plus agile et le plus réactif possible (c'est-à-dire capable d'évoluer rapidement pour répondre aux nouvelles demandes) tout en préservant les données et informations. [2]

Il s'agit de la problématique d'intégration et d'interopérabilité des systèmes d'information.

I-2-1- Définition de l'interopérabilité

L'interopérabilité est la capacité technique que plusieurs systèmes, qu'ils soient identiques ou radicalement différents, puissent communiquer et coopérer ensemble sans ambiguïté, sans conflit de systèmes ou de contenu.

L'interopérabilité est très importante voire critique dans de nombreux domaines, dont l'informatique, le médical au sens large, les activités ferroviaires, l'électrotechnique, l'aérospatiale, le domaine militaire.

L'interopérabilité du Web est la capacité des agents du Web de communiquer entre eux en utilisant un langage commun. L'interopérabilité repose nécessairement sur des standards ouverts pour que le langage puisse être commun à tous.

Un standard ouvert est un ensemble de règles, représentant un langage, reconnu par l'industrie, n'ayant pas de restrictions légales pour son implémentation.

Normes, standards et standards ouverts

Le Web : HTTP + HTML + Javascript + CSS. L'ensemble des documents et applications hypertextes.

Problématiques d'interopérabilité

- Interprétation différente des standards
- Gestion des erreurs différentes selon les navigateurs
- Erreurs dans les standards (incohérences)
- Omissions et manque de clarté pour certains éléments du standard
- Besoins non remplis par les standards
- Différentiations entre compétiteur et prises du marché
- Brevets sous-marins (GIFs)
- Utilisation du HTML comme langage de présentation au lieu de structure
- Implémentation incomplète des standards, particulièrement au niveau CSS et DOM

Rétrocompatibilité avec implémentations incompatibles avec les standards : mode quirks et mode standard

I-2-3- Définition d'intégration systèmes :

L'intégration est le processus d'interconnexion d'un système avec un autre système afin de fournir un échange utile d'informations, de données et / ou de contrôle entre les systèmes.

I-3- Définition du service web

« Un service web est un système logiciel conçu pour permettre l'interopérabilité des interactions entre machines sur un réseau. Il expose une interface décrite dans un format interprétable par une machine. D'autres systèmes interagissent avec le service d'une façon prescrite par sa description en utilisant des messages typiquement transportés sur http et

sérialisés au format XML » [3]

I-4- L'architecture web services

Un service web fournit un lien entre les applications en utilisant le système de messagerie XML et le Protocole HTTP pour transférer, recevoir les données.

Les services web sont basés sur le modèle SOA « Architecture orienté service » ils utilisent les technologies HTTP, WSDL, REST, XML-RPC, SOAP et UDDI.

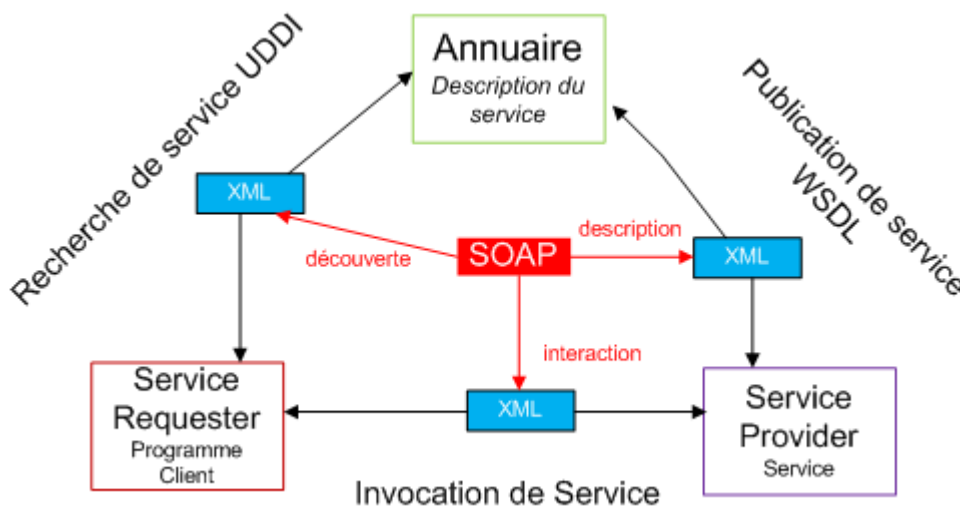


Figure 1 – architecture services web

Les services web se basent sur 3 éléments principaux :

➤ **Service Provider (Fournisseur de service) :**

Application s'exécutant sur un serveur et comportant un module logiciel accessible par Internet en XML

➤ **Annuaire :**

- Annuaire des services publiés par les providers (UDDI)
- Géré sur un serveur niveau application, entreprise ou mondial

➤ **Service requester (Demandeur de service) :**

Application cliente se liant à un service et invoquant ses fonctions par des messages XML (SOAP)

I-5- Fonctionnement d'un service web

Pour utiliser un Web Services, il faut premièrement savoir qu'il existe. UDDI (Universal Description, Discovery and Integration Service) est la norme qui définit le mécanisme pour découvrir dynamiquement des services. Un client pointe vers un registre UDDI, qui lui donnera la définition du service recherché. Le registre UDDI sert de pages jaunes et liste les services disponibles. Le registre UDDI est lui-même un Web Service qu'un client peut questionner.

Pour être capable d'utiliser un Web Services et de programmer un client, il est nécessaire d'en connaître la définition. Le langage WSDL (Web Services Definition Language) décrit l'interface au service. En utilisant XML Schema, WSDL définit les paramètres d'entrée et de retour d'un appel au service Web.

Les appels comme tel aux Web Services sont effectués avec le protocole SOAP (Simple Object Access Protocol). SOAP offre le transport d'objets sérialisés et autres données en XML et l'appel de procédures distantes.

UDDI, WSDL et SOAP sont les trois normes principales des Web Services. Les normes UDDI sont proposées par OASIS. WSDL et SOAP font parties des normes W3C.

I-6- Technologies services web

Les services Web communiquent via un ensemble de technologies fondamentales qui partagent une architecture commune dont HTTP, WSDL, REST, XML-RPC, SOAP et UDDI. Le concept des web services s'articule actuellement autour des concepts suivants :

WSDL (Web Services Description Language) : Le langage WSDL permet de décrire au format XML des Web Services en précisant les méthodes pouvant être invoquées, leur signature et le point d'accès (URL, port, etc..).

UDDI (Universal Description Discovery and Integration) : UDDI est un annuaire de service fondé sur XML et plus particulièrement destiné aux services Web. Un annuaire UDDI permet de localiser sur le réseau le service Web recherché, il s'agit d'un élément clé dans les spécifications de Services, car il permet l'accès au répertoire des utilisateurs potentiels de services Web.

SOAP (Simple Object Protocol communication) : SOAP est un protocole de communication basé sur XML permettant la transmission de messages en différent service Web. Nous reviendrons plus en détail sur le protocole SOAP dans la suite de la fiche de lecture. [4]

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

I-7- Protocoles utilisés

Les services web se basent essentiellement sur les protocoles SOAP, WSDL, UDDI, HTTP :

SOAP	Protocol basé sur le standard XML pour l'échange de données structurées entre des applications réseaux
XML-RPC	Il est construit sur HTTP, et s'appuie sur une mise en forme XML de tous les paramètres de l'invocation du service, et de la réponse, selon une syntaxe générique très simple
REST	Est un protocole sans état basé sur http, il utilise les standards du http (get, post, put, delete).
http	Il est utilisé pour échanger toute sorte de données entre client HTTP et serveur HTTP.

Tableau 1 : protocoles des services web

I-7-1 SOAP (Simple Object Access Protocole) :

SOAP est un protocole de communication basé sur XML pour permettre aux applications de s'échanger des informations via HTTP. Il permet ainsi l'accès aux services web et l'interopérabilité des applications à travers le web. SOAP est un protocole simple et léger et qui repose entièrement sur des standards établis comme le HTTP et XML. Il est portable et donc indépendant de tous système d'exploitation et du type d'ordinateur. SOAP est une spécification non propriétaire.

➤ Structure d'un message SOAP

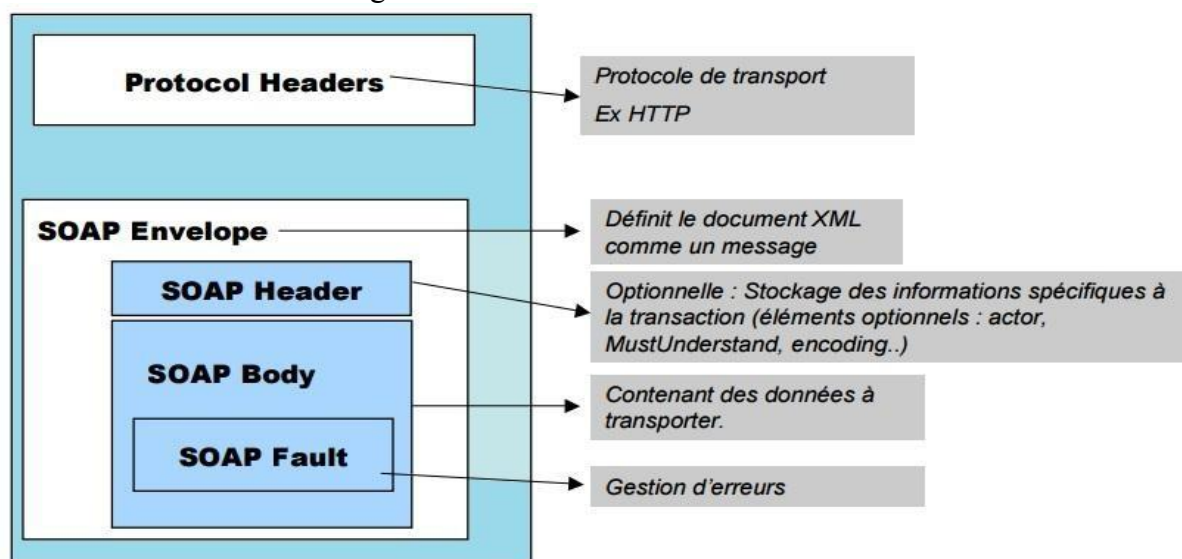


Figure2 : structure d'un message SOAP

- **Envelope:** c'est lui qui contient le message et ses différents sous-blocs. Il s'agit du bloc

racine XML. Il peut contenir un attribut « EncodingStyle » dont la valeur est une URL vers un fichier de typage XML qui décrira les types applicables au message SOAP.

```
POST /OrderEntry HTTP/1.1
Host: www.tutorialspoint.com
Content-Type: application/soap; charset="utf-8"
Content-Length: nnnn
<?xml version="1.0"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://www.w3.org/2001/12/soap-envelope"
SOAP-ENV:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
...
  Message information goes here
...
</SOAP-ENV:Envelope>
```

Figure 3 : bloc racine message XML

- **Header:** c'est un bloc optionnel qui contient des informations d'en-têtes sur le message. Si il est présent, ce bloc doit toujours se trouver avant le bloc Body à l'intérieur du bloc Enveloppe.

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://www.w3.org/2001/12/soap-envelope"
SOAP-ENV:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <SOAP-ENV:Header>
    <t:Transaction
xmlns:t="http://www.tutorialspoint.com/transaction/"
SOAP-ENV:mustUnderstand="true">5</t:Transaction>
  </SOAP-ENV:Header>

  ...

  ...

</SOAP-ENV:Envelope>
```

Figure 4 : Entête d'un message XML

- **Body:** c'est le bloc qui contient le corps du message. Il doit absolument être présent de manière unique dans chaque message et être contenu dans le bloc Enveloppe. SOAP ne définit pas comment est structuré le contenu de ce bloc. Cependant, il définit le bloc Fault qui peut s'y trouver.

Cet exemple illustre une demande de quotation des jeux d'un ordinateur. Il faut noter que m :quotation est un élément spécifique à l'application

```
<?xml version="1.0"?>

<SOAP-ENV:Envelope>
.....

<SOAP-ENV:Body>
  <m:GetQuotation xmlns:m="http://www.tp.com/Quotation">
    <m:Item>Computers</m:Item>
  </m:GetQuotation>
</SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

La réponse a la requête

```
<?xml version="1.0"?>

<SOAP-ENV:Envelope>
.....

<SOAP-ENV:Body>
  <m:GetQuotationResponse xmlns:m="http://www.tp.com/Quotation">
    <m:Quotation>This is Qutation</m:Quotation>
  </m:GetQuotationResponse>
</SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

- **Fault:** ce bloc est la seule structure définie par SOAP dans le bloc Body. Il sert à reporter des erreurs lors du traitement du message, ou lors de son transport. Il ne peut apparaître qu'une seule fois par message. Sa présence n'est pas obligatoire.

Sub-Element	Description
Faultcode	indique le type de l'erreur (VersionMismatch en cas d'incompatibilité avec la version de SOAP utilisée, MustUnderstand en cas de problème dans le header du message, Client en cas de manque d'informations de la part du client, Server en cas de problème d'exécution des traitements par le serveur)
Faultstring	message décrivant l'erreur
Faultactor	URI de l'élément ayant déclenché l'erreur
Faultdetail	C'est un élément utilisé pour transmettre des messages d'erreur spécifiques à l'application. « detail element » peut contenir des éléments enfants appelés « detail entries ».

Tableau 2 : erreurs reportés d'un message SOAP

Exemple

```
<?xml version='1.0' encoding='UTF-8'?>

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode xsi:type="xsd:string">SOAP-ENV:Client</faultcode>
      <faultstring xsi:type="xsd:string">
        Failed to locate method (ValidateCreditCard) in class
        (examplesCreditCard) at /usr/local/ActivePerl-5.6/lib/
        site_perl/5.6.0/SOAP/Lite.pm line 1555.
      </faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

I-7-2 WSDL (Web Services Description Language)

Il est basé sur le langage XML, est une recommandation de W3C, il est utilisé pour fournir une description détaillée d'un service web au client dans un format xml afin de permettre son utilisation. [5]

I-7-2-1- Structure d'un document WSDL

Un WSDL est virtuellement composé de deux parties :

- > **Définitions abstraites** : celles-ci concernent l'interface du service (types, message, portType). Ces informations sont exploitées dans le code du client
- > **Définitions concrètes** : celles-ci concernent l'invocation du service (Binding, Service). Ces informations sont exploitées par le moteur SOAP.

Voici ci-dessus la structure d'un document WSDL :

<Définition>

- Racine d'un document WSDL

<types> (optionnel et un seul autorisé)

- Contient la définition des types des données exprimée sous forme xml schema

<Message> plusieurs autorisée

- Décrit des messages a transmettre(paramètre d'une opération, valeur de retour, exception,...)

<Porttype >plusieurs autorisées

- Décrit un ensemble d'opérations ou chacune à 0 ou plusieurs messages en entrée, 0 ou plusieurs messages de sortie ou de fautes

<Binding> plusieurs autorisées

- Spécifie une liaison entre un portType à un protocole (SOAP, http)

<Service> plusieurs autorisés

- Regroupe l'ensemble des ports(relation entre binding et URL)

➤ **Elément définition**

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

➤ **Elément Type**

L'élément <types> contient une définition des différents types de données.

L'utilisation de schéma xml est recommandé et peut avoir aucun, un ou plusieurs fils « schéma » ayant pour espace de nommage « <http://www.w3.org/2001/XMLSchema> », ces schémas peuvent décrire des types simples (element) ou complexes (complexType).

➤ **Elément Message**

L'élément message décrit un message qui est utilisé en tant que requête ou réponse lors de l'invocation d'une opération : il contient une définition des paramètres pour un message échangé en entrée ou en sortie.

➤ **Elément PorType**

L'élément PorType est considéré comme une bibliothèque, un module ou une classe, il contient un ensemble d'opération qu'un service web peut effectuer.

Un PorType est identifiable par un nom (attributname) et composé de sous éléments (opérations) et il est possible d'identifier une opération en suivant quatre modèles comme l'illustre le tableau suivant :

One way	Le client de service envoi un message a l'opération et n'attend pas de réponse, uniquement un seul message utilisé <input>
Request/response	Le client de service envoi un message a l'opération et reçoit un message est retourné au client, un message <input>, <output>, <fault>
Notification	Le service envoi un message au client, uniquement un seul message utilisé <outpu>

Solicit/response	Le client reçoit un message du service et répond au service un message <input>, <output>, <fault>
------------------	--

Tableau 4 : opérations d'un service web dans l'élément PortType

➤ Élément binding

L'élément « binding » permet de réaliser la partie concrète du <porttype>, il fournit les informations pour invoquer le service :

- le protocole utilisé pour le transport du message
- l'encodage du message : style et mécanisme d'encodage

la structure de l'élément « binding » dépend du protocole utilisé

I-7-3 UDDI (Universal Description, Discovery and Integration)

C'est un protocole et un ensemble de services est utilisé pour publier et rechercher des services web, utilise un annuaire afin de stocker les informations concernant les services web et de permettre à un client de les retrouver.

UDDI permet à un client ou entreprise de s'inscrire dans l'annuaire, d'y enregistrer et de publier ses services web. Il est alors possible d'accéder à l'annuaire et de rechercher un service particulier.

Le site web officiel d'UDDI est à l'url : <http://uddi.xml.org>

Les données incluses dans un annuaire UDDI sont classées dans trois catégories :

- Les pages blanches (white pages) : elles contiennent les informations générales sur une entreprise
- Les pages jaunes (yellow pages) : elles permettent une catégorisation des entreprises
- Les pages vertes (green pages) : elles contiennent les informations techniques sur les services proposés

I-7-4 XML (Exetensible Markup Language)

XML est un standard qui permet de décrire des documents structurés transportables sur les protocoles d'Internet, la technologie des Web services est essentiellement basée sur XML ainsi que les différentes spécifications qui tournent autour (les espaces de nom, les schémas XML, et les schémas de Type).

I-8- Rest (REpresentational State Transfert)

REST a été créé en 2000 par Roy T. Fieldling, également créateur du projet Apache HTTP Server et membre de la fondation Apache.

Il s'agit d'une architecture permettant à un client d'échanger avec un serveur de manière unitaire, très structurée et gérant le cache. Ainsi, il est possible de mettre en place des couches qui évoluent de manières autonomes.

Classiquement, dans le cas d'un contexte web (WWW - World Wide Web), tel qu'abordé dans ce livre, il s'agit d'une interface web pour le client, et d'une application de gestion de données pour le serveur, qui communiquent ensemble.

REST est indépendant du format représentant les données, que ce soit JSON, XML, ou une autre forme de sérialisation.

I-8-2 Les identifiants de REST

Verbe http	Rôle
Get	Permet de récupérer les données d'une ou plusieurs entités dans une ressource. Il est possible de filtrer par des paramètres dans l'url.
Post	Permet de rajouter des entités à une ressource. Les données sont envoyées dans le corps de la requête. Par défaut, une requête POST intègre le format Content-Type:application/x-www-form-urlencoded. Ce format est déconseillé avec les APIs REST car il ne correspond pas aux formats de réponse utilisés (JSON, XML).
Put	Permet de modifier les données d'une entité dans une ressource.
Delete	Permet de supprimer une entité dans une ressource.

Tableau 5 : Identifiants Rest

les réponses du serveur aux clients utilisent également les messages HTTP, en particulier les codes d'erreur suivants :

- 200 Ok
- 404 Ressource non trouvée sur le serveur

I-8-3 JAX-RS (Java Api for restfull services)

8-3-1 Définition JAX-RS

Le développement des services web REST repose sur l'utilisation de classes en java et les annotations.

JAX-RS C'est une interface de programmation Java permettant de créer des services Web avec une architecture REST.[wiki]

8 3-2 spécifications JAX-RS

Différentes implémentations de la spécification JAX-RS sont disponibles :

- Jersey, implémentation de référence de SUN
- CXF : fournie par apache c'est une fusion entre Xfire et Celtix
- Resteasy, l'implémentation jboss
- Restlet : un des premiers Framework implémentant REST pour JAVA

I-9- Avantages et inconvénients SOAP et REST

	SOAP	REST
Avantages	<ul style="list-style-type: none"> -Standard éprouvé -Indépendant de la plateforme - Indépendant du langage - Extensible - Intègre la gestion des erreurs 	<ul style="list-style-type: none"> - indépendance entre le client et le serveur - il peut utiliser des formats de message plus petits - Rapide (pas de traitement étendu requis) - Proche d'autres technologies Web
Inconvénient	<ul style="list-style-type: none"> - Lourd et lent 	<ul style="list-style-type: none"> - Dépendant du protocole http - Nécessite une
	<ul style="list-style-type: none"> - Verborisité du XML - Défficile a tester et a cacher 	<ul style="list-style-type: none"> communication directe point à point

Tableau 6 : Avantage et inconvénients SOAP et REST

I-9- 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.

De plus en plus de systèmes d'architecture se tendent vers l'architecture REST en raison de sa flexibilité et de sa facilité d'utilisation. Le protocole SOAP rencontre un problème d'interopérabilité que REST résout.

La sécurité reste un sujet problématique sur l'architecture REST, car seul le passage en HTTPS permet de sécuriser la communication.

Etant donné que REST est sans état, nous n'avons pas la description des différents services et objets exposés.

Au cours de ce chapitre nous avons fait une introduction à l'interopérabilité et intégration système, et nous avons définis les services web, ainsi que leurs architecture, et standards utilisés.

Dans le chapitre suivant nous allons entamer l'enseignement à distance « E-Learning ».

Chapitre 2 : **L'e-Learning**

I-1- Introduction :

Le e-Learning ou l'enseignement à distance est devenu possible grâce aux progrès actuels de l'informatique et l'avènement des nouvelles technologies de l'information et de la communication (TIC) qui favorisent l'accès à d'innombrables ressources en ligne et contribue à l'évolution des méthodes d'enseignement.

Dans ce chapitre, nous allons définir l'e-Learning en présentant ses caractéristiques et ses plates-formes et les différents moyens d'apprentissage ainsi que quelques avantages et inconvénients du e-Learning.

I-2- Historique :

Les premières formations à distance voient le jour au milieu de 19^{ème} siècle avec le premier cours par correspondance, lancé en 1840 en Angleterre.

Au départ l'idée était surtout d'apporter un complément de formation aux adultes n'ayant pas pu finir leurs études. Un tuteur apporte son aide par courrier ou par téléphone mais les abondants sont nombreux. En 1939 le CNED (Centre Nationale D'Enseignement à Distance) est fondé. Le principe est le suivant, les élèves reçoivent par courrier postal, ou par courrier électronique le cours et ils envoient les exercices et devoirs au correcteur qui les note et envoie les corrigés-type. A partir des années 60, l'audiovisuel et l'informatique jouent un rôle pédagogique croissant et font leur apparition dans l'enseignement scolaire. Certains programmes éducatifs sont radiodiffusés.

Dans les années 70, l'enseignement interactif à distance voit réellement le jour avec l'arrivée de s programmes intelligents, ce sont le début du e-Learning à proprement parler. [1]

I-3Définition :

Le « e » de e-Learning est l'abréviation de « électronique ». 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 ». [2]

Il existe différentes définitions du e-Learning, nous citons ci-dessous quelques unes:

- L'e-Learning recouvre toutes les méthodes de formation s'appuyant sur l'outil informatique. Cette définition large inclut plusieurs axes indépendants :[4]
- ❖ apprentissage individuel ou collectif,

- ❖ supports en ligne ou hors-ligne,
- ❖ formateur présent, à distance, voire absent.
D'autres termes sont parfois utilisés, tels que :
 - ❖ formation en ligne,
 - ❖ apprentissage en ligne,
 - ❖ enseignement en ligne,
 - ❖ formation à distance,
 - ❖ enseignement à distance...
- Il est défini par l'union européenne en (2003) comme : « l'utilisation des nouvelles technologies multimédias de l'Internet pour améliorer la qualité de l'apprentissage en facilitant d'une part l'accès à des ressources et à des services, d'autre part les échanges et la collaboration à distance ». [3]
- ~~E-learning~~ Learning est un système d'enseignement à distance, dispensé au travers d'un dispositif électronique, caractérisé par des contenus et des outils didactiques, collaboratifs et de gestion, au service de l'apprentissage et de la connaissance. Ce dispositif est mis en œuvre à travers un système d'interactions humaines méditées par les technologies ». [2]



Figure 1 : le Système E-Learning

I-4-Les acteurs et leur rôle dans le e-Learning :

Les acteurs du e-Learning peuvent être classés, selon leur rôle, en trois catégories principales : les apprenants, les enseignants et les administrateurs. [3]

I.1.1 Apprenant :

Suit depuis son poste de travail une formation dont le contenu est organisé selon un parcours pédagogique défini par le formateur en fonction des compétences possédées par l'apprenant avant la formation, en fonction des objectifs pédagogiques visés par la formation et en fonction des résultats obtenus lors des évaluations intermédiaires : le parcours est donc individualisé.

I-1-2 Enseignant :

Le e-Learning nécessite plusieurs types d'enseignants, différenciés par leurs rôles. On peut distinguer quatre types d'enseignants :

- Auteur (concepteur) de cours : celui qui développe un cours en utilisant les outils de la plateforme selon ses objectifs pédagogiques et qui apporte des changements en fonction des réactions des apprenants ou des tuteurs.
- Orienteur : c'est l'enseignant qui a pour principales tâches, l'élaboration des cursus des apprenants ou des groupes d'apprenants, l'élaboration des plans de formation, et gestion du livret des apprenants.
- Tuteur : le rôle du tuteur apparaît comme fondamental, il ne s'agit plus de transmettre des connaissances, mais d'accompagner l'apprenant dans ses acquisitions, de lui faire acquérir un maximum d'autonomie. Il est là pour combattre la sensation d'isolement que pourrait ressentir l'apprenant dans une classe virtuelle et pour maintenir sa motivation par des encouragements, des questionnements, des suggestions d'approfondissement. Il est de plus en plus fréquent que les formations soient associées à la mise en place d'une communauté d'apprentissage que le tuteur sera chargé d'animer. Le tuteur humain crée les éléments de la formation (modules de cours, évaluations, tests, simulations, ...etc.) et suit, guide et oriente l'apprenant durant le processus d'apprentissage.
- Evalueur : est souvent tenu par le tuteur lui-même; il consiste à corriger les tests et les exercices proposés et à attribuer des notes aux élèves. Ces notes serviront à moduler le parcours d'apprentissage des apprenants et pourront également être exploitées ultérieurement pour identifier les parties du cours qui posent problème afin d'améliorer constamment ce dernier.

I-1-3-Administrateur :

On peut distinguer deux types d'administrateurs :

- Administrateur technique : gère la plateforme (installation et maintenance).
- Administrateur institutionnel : gère les inscriptions, gère les comptes, affecte les droits d'accès pour les acteurs et gère les liens avec les systèmes d'information externes (scolarité, catalogues, ressources pédagogiques ...etc.).

Caractéristiques principaux du e-Learning :

Il existe sept caractéristiques pour le e-Learning :

- Internet est le moyen d'accès à la matière.
 - L'apprenant est l'élément central. Il peut définir lui-même son rythme, son environnement et sa méthode de travail.
 - L'environnement d'études peut être adapté aux besoins spécifiques de formation.
 - Le tout est soutenu par un réseau d'expériences qui peuvent être partagées.
 - Le système utilise différents techniques d'enseignement : un local de classe virtuel, des simulations, différentes formes de collaboration, des groupes de discussion et l'apprentissage en direct.
 - Toutes les étapes de l'enseignement traditionnel sont respectées : études, examens, certificats.
 - L'inscription, la gestion des données personnelles et le suivi se font en ligne.
- [1]

I-5- Les plateformes e-Learning :

Les plateformes de formation à distance sont des outils logiciels dont le rôle est de permettre le pilotage des enseignements à distance. Dans la littérature, plusieurs définitions sont attribuées. La plateforme de formation à distance est définie comme étant "un logiciel de création et de gestion de contenus pédagogiques et de supports aux acteurs destiné à trois types d'utilisateurs: L'enseignant, l'apprenant et l'administrateur. Elle regroupe les outils nécessaires aux trois types d'intervenants permettant d'incorporer des ressources pédagogiques multimédias, de participer à des activités et d'effectuer le suivi pédagogique et administratif des apprenants". Certaines dénominations ont été attribuées aux plateformes de formation dépendamment des fonctionnalités et des services proposées. Il s'agit de LMS (Learning Management System), CMS (Content Management System), LCMS (Learning Content

Management System), etc. [3] Dans ce qui suit, nous allons présenter chacun de ces systèmes:

- **LMS (Learning management system)** : est une famille de systèmes, qui permet la gestion de toutes les activités de la formation. Les systèmes LMS catégorisent les utilisateurs, leur donnent certaines autorisations à des modules de formation et affectent les utilisateurs à des groupes spécifiques de formation. WebCT, Ganesha, Caroline, Moodle sont des LMS. La première plateforme est de type propriétaire, les autres sont de type Open Source.

Ses principales fonctions sont :[5]

- La présentation du catalogue de cours.
- L'intégration des modules de cours et des ressources annexes.
- La diffusion des cours à la demande sur les postes des utilisateurs.
- La gestion des activités des tuteurs.
- La gestion des évaluations des apprenants.
- La gestion des processus qualité.
- La construction des parcours de formation personnalisés.
- Le support à la création d'une communauté d'utilisateurs.
- Le reporting pour les ressources humaines.
- La gestion administrative des apprenants incluant les inscriptions et la comptabilité.

- **CMS (Content Management System)** : est une famille de systèmes d'e-Learning dédiés à la création, le stockage, la gestion et la présentation des contenus. L'important dans ce modèle est de fournir des possibilités de réutilisation des objets d'apprentissage, appelés RLO (Reusable Learning Object) ou des composants de contenu.

- **LCMS (Learning content management system)** : est une famille de systèmes d'e-Learning les plus avancés technologiquement. Les systèmes LCMS intègrent des fonctionnalités des LMS et des CMS, assurant à la fois la création et la gestion des contenus éducatifs. Ils offrent la possibilité d'évaluer les connaissances assimilées par les utilisateurs. Top Class, Simply Learn sont des exemples de telles suites intégrées.

L'architecture suivante représente l'intégration de la fonctionnalité de LMS et celle de CMS. Elle montre le modèle de fonction de LCMS (LMS + CMS) pour l'e-Learning.

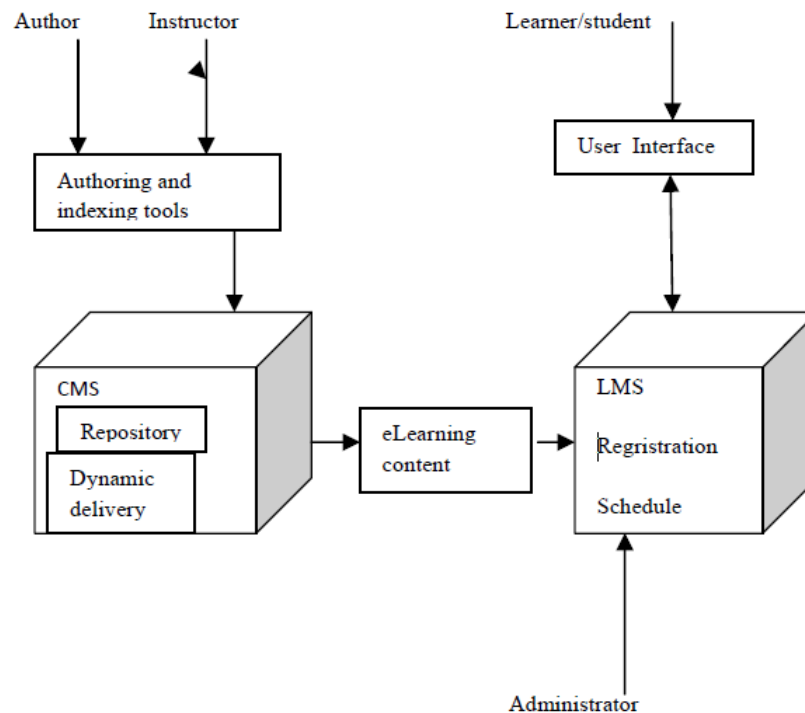


Figure 2 : LCMS (LMS+CMS)

I-6-Modes de diffusion :

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

En effet, il existe de nombreuses possibilités de communication en e-Learning. De façon simplifiée, on distingue 3 modes de diffusion du E-Learning : [1]

✓ Mode asynchrone :

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

Les principaux avantages de la formation asynchrone :

- Les apprenants évoluent à leur propre rythme.
- Ils peuvent adapter l'ordre dans lequel ils appréhendent les éléments du cours.
- Ils peuvent revoir et approfondir certains aspects du cours à leur guise.

✓ Mode synchrone :

C'est une méthode en temps réel basée sur la parole et l'écoute.

L'apprenant est en contact avec son formateur et les autres membres de sa classe virtuelle par partage d'application ou visioconférence. Les étudiants connectés posent leurs questions et reçoivent la réponse en temps réel, quel que soit la distance à condition bien sûr que la bande passante soit à la hauteur. Le dialogue entre les formateurs et les apprenants peut se faire via : Chat, Webcam, Micro, Partage d'applications, Fonctionnalités de prise en main à distance, Tableau blanc, Téléphone.

Les principaux avantages de la formation synchrone :

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

✓ □ Mixte :

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

Les principaux avantages de la formation mixte :

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

I-6- Différents modèles du e-Learning :

L'évolution technologique importante dans le domaine informatique et des communications entraîne des développements et des déclinaisons des concepts d'apprentissage.

Dans le tableau suivant, nous proposons une sélection et une traduction des termes actuellement retrouvés dans le monde de l'enseignement numérique.

Modèles	Traduction proposée	Focus et caractéristiques
E-Learning self-study with other blended media or events.	Formation intégralement à distance avec des ressources	L'apprentissage à distance autonome est le focus de ce modèle. Il n'est pas envisagé de présentiel. L'apprenant accède à de multiples ressources d'apprentissage qui sont mises à sa disposition.
Instructor-led program blended with self-study e-learning	Formation blended Learning (formation à distance et présentielle)	Le modèle est un mixte de présentiel et d'apprentissage à distance autonome. Les activités de e-Learning sont utilisées comme prérequis, pendant le présentiel, et entre les séquences de présentiel
Live e-learning centered with other media added	Formation en direct/live	Le focus du modèle est le direct/live. Les outils sont le Webinar, la classe virtuelle. Un apprentissage à distance autonome sur une plate-forme de formation, des exercices complètent le dispositif
On-the-Job training	Formation en situation de travail	Le focus du modèle est l'apprentissage dans le milieu de travail avec un encadrant, un formateur. L'apprentissage à distance favorise l'apprentissage en relation avec une situation de travail en fournissant des ressources externes (contenus, procédures...)
Simulation and lab-centered	Simulation et laboratoire	Le focus est l'apprentissage par simulation et au travers de temps de laboratoire où les apprenants peuvent s'exercer en toute sécurité

Tableau1: Différents modèles selon Bersin

Ces différents modèles montrent l'évolution des possibilités d'apprentissage suivant l'évolution de la technologie et des besoins des apprenants. La sélection s'effectue principalement en fonction des objectifs pédagogiques ciblés, les avantages et inconvénients de chaque modèle devant être évalués en fonction du cahier des charges demandé par l'organisme de formation.

I-7- Approches pédagogiques en E-Learning

On distingue trois modèles pédagogiques : [1]

- **Apprentissage par transmission :**

La connaissance est considérée comme transférable à l'apprenant. A la suite de ce transfert, et grâce à des exercices de résolution de problèmes structurés, l'apprenant devient capable d'utiliser ses propres connaissances dans de nouvelles situations. Ce modèle est fondé sur l'approche behavioriste, selon laquelle l'apprentissage naît d'une modification du comportement, provoquée par des stimuli provenant de l'environnement. Cet apprentissage peut être obtenu par l'utilisation de la récompense appelée « renforcement positif ».

- **Apprentissage par échange :**

On dit que la connaissance est le fruit des échanges de l'apprenant avec son environnement. Après la maîtrise d'une notion, il ne s'agit plus pour l'enseignant de proposer lui-même de nouveaux exercices aux apprenants, mais ceux-ci doivent eux-mêmes, à travers la participation et l'échange, des situations où ils peuvent retrouver cette notion ou procédure et la rejouer.

- **Apprentissage par construction :**

La connaissance est une construction sociale. A un niveau donné d'apprentissage il y a des choses que l'apprenant peut faire tout seul et d'autres qu'il ne peut pas faire sans l'aide d'autre personnes appartenant à sa culture et plus expérimentées que lui. C'est seul et en groupe que l'apprenant construit son savoir, et ce sont les intérêts de la communauté qui le poussent à agir.

I-8- Les logiciels de lecture multimédias :

L'**interface Web** est le principal canal de diffusion du e-Learning. Un navigateur Internet (tel que Microsoft Internet Explorer, Netscape, Mozilla, etc.) est donc suffisant.

Les pages Web simples n'étant pas capables de répondre aux besoins d'interactivité des cours

e-Learning, d'autres technologies multimédias sont fréquemment utilisées en complément dans les modules du e-Learning.

La plus répandue est **Macromedia Flash**, initialement développée pour présenter des animations poussées et du son dans des sites Internet. Flash permet également de réaliser des

modules de cours hautement interactifs et de programmer leur intégration avec le système d'évaluation des apprenants. Flash a également l'avantage d'être installé sur une grande majorité des postes connectés à Internet (plus de 98% en 2002) et d'être reconnu pour sa sécurité.

Le **streaming** (c'est-à-dire l'affichage au fil du téléchargement) est aussi une technique utilisée pour diffuser des contenus multimédias par Internet. Elle permet de réduire le temps d'attente des utilisateurs lorsque le module de cours contient des éléments sonores ou vidéo. Les principales technologies du marché sont Microsoft Streaming Server, Real Helix, Apple Quicktime Server et Flash Streaming Server.

I-9- L'e-Learning en Algérie :

Par rapport aux années précédentes, l'Algérie a fait un grand saut technologique en matière de développement et de structures dans les TIC¹. Le projet de e-Learning vise à développer des pratiques pédagogiques et améliorer le processus d'enseignement grâce à des méthodes d'apprentissage dynamique et interactives par ces formations à la pédagogie.

On va citer quelques exemples sur les plates formes en Algérie : [1]

- English Discoveries Online (EDO).
- DZCampus.com.

II-10- Comparaison du e-Learning et de la formation traditionnelle : [5]

	Avantages	Inconvénients
Présentiel	<ul style="list-style-type: none"> - Feedback immédiat - Repères visuels - Facilité d'interaction avec les autres apprenants - Favorise les personnes communiquant facilement 	<ul style="list-style-type: none"> - Défavorise les personnes timides ou analytiques - Contrôlé par le professeur - Cadré dans le temps

e-Learning	<ul style="list-style-type: none"> - Tout le monde peut contribuer - Contrôlé par l'apprenant - Peu cadré dans le temps - Enregistrement permanent des échanges 	<ul style="list-style-type: none"> -Manque de repères visuels - Obstacles technologiques - Favorise les personnes communicant facilement par écrit
-------------------	---	---

Tableau2 : Comparaison des avantages et inconvénients du e-Learning de la formation traditionnelle

II-11-Avantages et inconvénients du e-Learning:

II-11-1- Avantages :

➤ **L'e-Learning touche plus de collaborateurs en moins de temps :**

Lorsque les modules de formation sont en place, le déploiement des cours est plus rapide que dans le cas de cours présentiels, notamment dans les environnements internationaux ainsi que dans les grandes entreprises (formation en masse des salariés).

➤ **L'e-Learning minimise le temps passé hors du lieu de travail :**

L'apprentissage individuel sur un poste de travail dans les locaux de l'entreprise réduit les pertes de temps de déplacement. La formation peut être suivie à n'importe quel moment de la journée. Ceci est bien évidemment particulièrement vrai dans le contexte des entreprises multinationales.

➤ **L'apprentissage est plus rapide et plus durable ;**

Des études montrent que les courbes d'apprentissage peuvent être améliorées par l'e-Learning et que la mémorisation des informations est souvent meilleure. Les temps d'apprentissage sont en général plus faibles qu'en présentiel (environ 50% plus faibles). Bien utilisé, l'e-Learning permet donc d'améliorer la qualité de la formation.

➤ **L'apprentissage est personnalisé :**

L'apprenant peut choisir le temps à passer sur chaque module d'une formation en fonction de ses acquis. Il peut adapter le rythme du cours à son niveau (moins de stress, moins de frustration).

➤ **La gestion des certificats est automatisée :**

Les plates-formes d'e-Learning permettent de suivre le parcours des apprenants et de gérer les certifications.

➤ **L'e-Learning coûte globalement moins cher que le présentiel :**

L'e-Learning élimine les coûts liés à la rémunération des professeurs, la gestion des salles

de cours, les déplacements des employés, leur hébergement, etc. Les économies réalisées par rapport à la formation traditionnelle sont proportionnelles au nombre d'apprenants. Suivant les cas, les coûts du tutorat se substituent néanmoins partiellement à la rémunération des formateurs traditionnels

II-11-2- Inconvénients :

➤ **La diffusion de cours e-Learning nécessite des équipements multimédia :**

L'équipement de l'entreprise doit permettre la diffusion des contenus des cours. Cela concerne notamment les postes utilisateurs (PC récents, logiciels installés compatibles, éventuellement réseau avec suffisamment de bande passante). Cet inconvénient tend à perdre de son importance au fil des années, étant donné l'évolution des technologies.

➤ **La mise en place de l'infrastructure technique et la création des contenus sont coûteuses :**

Un investissement est nécessaire au lancement d'une politique du e-Learning. Cet investissement (coût fixe et risque important) remplace le paiement à la prestation de formation (coût variable et risque faible). Par ailleurs, le marché n'est pas encore consolidé et l'offre en matière de cours spécialisés reste limitée suivant les domaines.

➤ **L'accès à l'outil informatique est nécessaire :**

L'utilisation de l'outil informatique limite la diffusion du e-Learning auprès d'une partie des collaborateurs. Ceci est un obstacle par exemple dans l'industrie.

➤ **L'e-Learning limite les interactions entre les individus :**

Certains mécanismes de communication ne peuvent pas être reproduits (langage du corps par exemple), alors qu'ils jouent un rôle important dans la diffusion du savoir.

II-12- Conclusion :

Aujourd'hui, les systèmes d'E-Learning sont très importants car ils deviennent le support de base de l'apprentissage sur le web. Ils offrent plusieurs services aux apprenants : le parcours des cours, la disponibilité, flexibilité, pas de frontières géographiques, haute qualité, et plusieurs techniques pour aider l'apprentissage des connaissances, etc.

Dans ce chapitre, nous avons présenté une vue globale sur le E-Learning, nous avons commencé par définir le mot e-Learning, ensuite, décrire quelques concepts de base. Enfin nous avons fait une petite comparaison entre la formation traditionnelle et l'e-Learning, et nous avons énuméré quelques avantages et inconvénients de ce dernier.

Chapitre III :

Conception

III-1- Introduction

L'étape de conception est l'étape la plus importante pour l'élaboration de notre application, elle permet en effet, de détailler les spécifications des fonctions ainsi que la structure des données.

Dans le but d'avoir une meilleure analyse et de rendre la conception de notre application plus complète, nous avons adopté le langage UML (Unified Modeling Language) qui permet de bien représenter l'aspect statique et dynamique d'une application par une série de diagrammes qu'il offre.

III-2- Problématique

Actuellement on voit beaucoup de centre offrent des formations généralement aux étudiants qui souhaite acquérir des connaissances et apprendre, sauf que beaucoup s'inscrivent et cause de la surcharge d'emploi de temps de leurs universités n'arrivent n'aboutissent pas a la fin de la formation.

En effet nous avons opté pour une autre solution « E-Learning », qui permet de s'inscrire chez soi et suivre des cours dans le temps libre, et d'avoir toute une équipe d'enseignants, et pédagogues qui l'accompagneront toute au long de sa formation, ainsi offrir la possibilité a l'enseignant de faire des heures supplémentaires en quelques cliques.

III-3- Objectif :

Dans notre projet nous allons réaliser une application e Learning qui a pour but de donner aux étudiants une formation en anglais aller du premier niveau au 6^{ème} niveau, enfin obtenir une attestation de réussite.

Notre application contient deux types de cours :

- Cours structurés : ils sont publiés lors de réalisation de l'application par l'administrateur et l'auteur (concepteur), ces derniers n'ont accès que par les visiteurs inscrits sur le plateforme autant qu'étudiants bénéficiant d'une scolarité, et ne sont modifiables que par l'accord de l'administrateur.
- Cours non structurés : ils sont publiés par les enseignants inscrits sur le site pour permettre aux étudiants une meilleure compréhension, et tout visiteur à droit d'y accéder.

Un étudiant qui s'inscrit pour la première fois sera affecté a un test de niveau qui lui déterminera son niveau en anglais, et pourra donc accéder aux cours de son niveau et inférieur, après avoir lu les cours, il aura un test de niveau qui lui permettra d'accéder ou non au niveau supérieur.

Un enseignant qui s'inscrit sur le site a pour but de publier des cours, exercices et accompagner l'étudiant durant le processus d'apprentissage.

III-4- Spécification des exigences

Dans cette section nous identifions une liste d'exigences fonctionnelles et non fonctionnelles du système à concevoir. Certaines exigences sont ajoutées pour clarifier les besoins des utilisateurs.

III-4-1 Etude des besoins fonctionnels

Les besoins fonctionnels représentent les attentes de chaque acteur de la future plateforme. Toute solution conceptuelle doit satisfaire, préalablement, à des besoins fonctionnels afin de délimiter le périmètre fonctionnel de l'application et surveiller la traçabilité des besoins lors de la phase de développement.

La présente plateforme doit satisfaire les besoins fonctionnels suivants :

- La gestion des utilisateurs
- La gestion des cours
- La gestion des tests
- La gestion des notes
- La gestion des ressources
- La gestion de la communication entre les utilisateurs
- La gestion des services web

III-4-2- Etude des besoins non fonctionnels

Un besoin non fonctionnel est une restriction ou une contrainte qui pèse sur un service du système, telle les contraintes liées à l'environnement et à l'implémentation et les exigences en matière de performances.

Les besoins non fonctionnels :

- Le code doit être clair pour permettre des futures évolutions ou améliorations
- La plateforme doit fournir un accès rapide aux informations, et doit faire la mise à jour en temps réel
- La plateforme doit être portable, extensible, réutilisable et fiable
- La plateforme offre une interface conviviale et facile à utiliser
- La plateforme doit garantir la confidentialité, l'intégrité et la cohérence des données.

III-5- Les acteurs système

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Les acteurs interagissant avec notre système sont :

- Administrateur : c'est le responsable de l'administration du site de téléformation.
- Tuteur : anime des formations dans le salon de formation.
- Apprenant : assiste à une formation dans le salon de formation.

III-6- Identification et structuration de cas d'utilisation :

III-6-1- Définition :

Le diagramme de cas d'utilisation est un diagramme qui représente les relations entre les acteurs et les fonctionnalités du système, dans cette partie nous allons détailler les différents scénarios de principaux cas d'utilisation.

III-6-2 Diagramme cas utilisation « authentification »

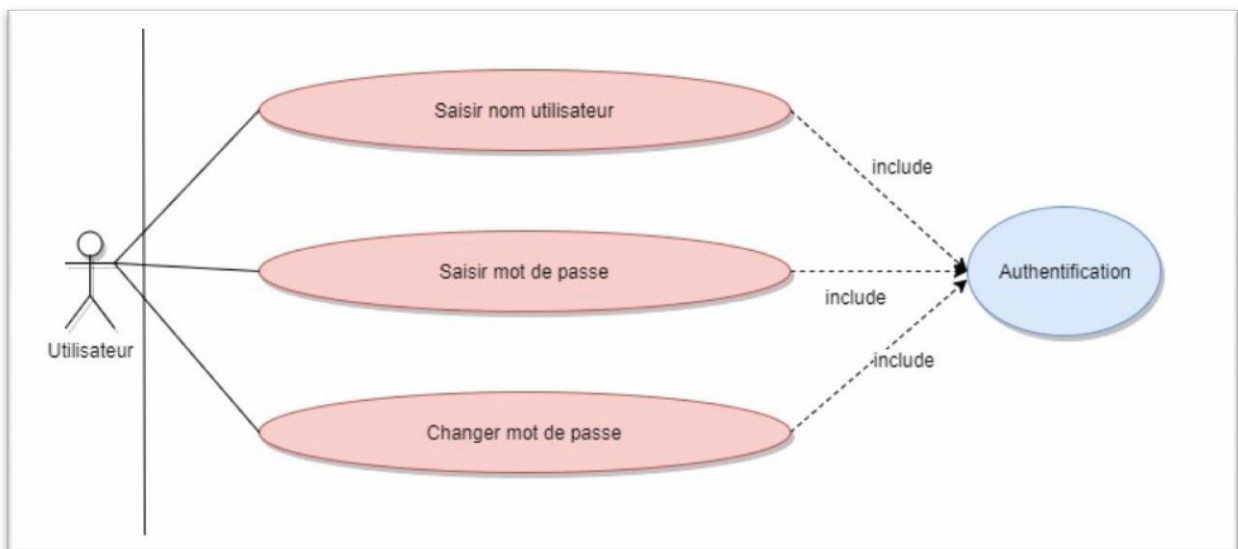
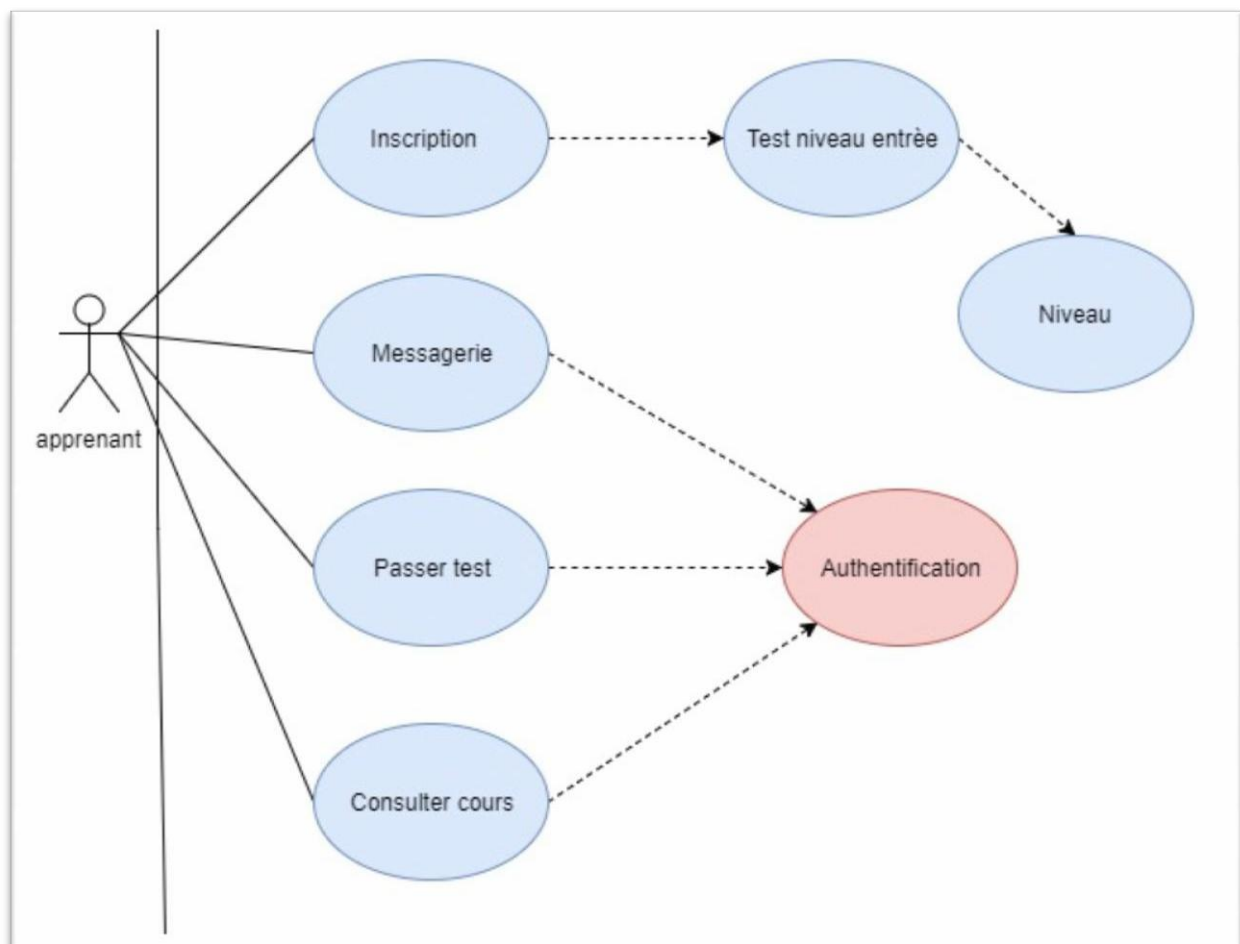


Figure 2 : Diagramme de cas utilisation « Authentification »

Authentification
Résumé : permet a l'utilisateur d'accéder son espace personnel
Acteurs : administrateur, enseignant, apprenant
Pré-condition : l'utilisateur devrait être inscrit ou membre de la plateforme
Scénario <div style="text-align: center;"><début></div> 1- l'utilisateur sélectionne la rubrique <authentification> 2- le système affiche un formulaire de connexion a l'acteur 3- l'utilisateur saisi son identifiant et le mot de passe. 4- le système vérifie la validité des informations saisies 5- l'utilisateur accède a son espace personnel

Scénario alternative

- 1- Dans le cas ou les informations saisies lors de l'authentification sont erronés
- 2- le système affiche un message d'erreur
- 3- l'utilisateur peut ressaisir ses coordonnées ou change le mot de passe

III-6-3 Diagramme cas utilisation « rôle apprenant »**Figure 3 : Diagramme de cas utilisation « rôle apprenant »****Rôle apprenant**

Résumé : l'auteur a accès aux cours et exercices, et passer examen pour se hisser dans les niveaux

Acteurs : visiteur, apprenant, administrateur

Pré-condition : l'apprenant doit s'identifier pour avoir accès aux cours et examens

Scénario 1 :

<début>

- 1- l'utilisateur selectionne la rubrique inscription
- 2- le système retourne un formulaire d'inscription
- 3- l'utilisateur entre ses coordonnées et valide
- 4- le système confirme l'inscription de l'apprenant
- 5- l'apprenant effectue un test de niveau
- 6- le système enregistre tout dans la base de donnée

<fin>

Scénario 2 :

<début>

- 1- l'utilisateur s'authentifie avec son login et mot de passe
- 2- le système confirme l'authentification
- 3- l'utilisateur accède au cours et exercices
- 4- l'utilisateur accède a l'examen pour se hisser dans les niveau

<fin>

Cas alternatif :

<début>

- 1- en cas de non inscription et authentification, l'utilisateur n'a accès qu'aux cours non structurés

<fin>

III-6-4 Diagramme cas utilisation « gérer cours »

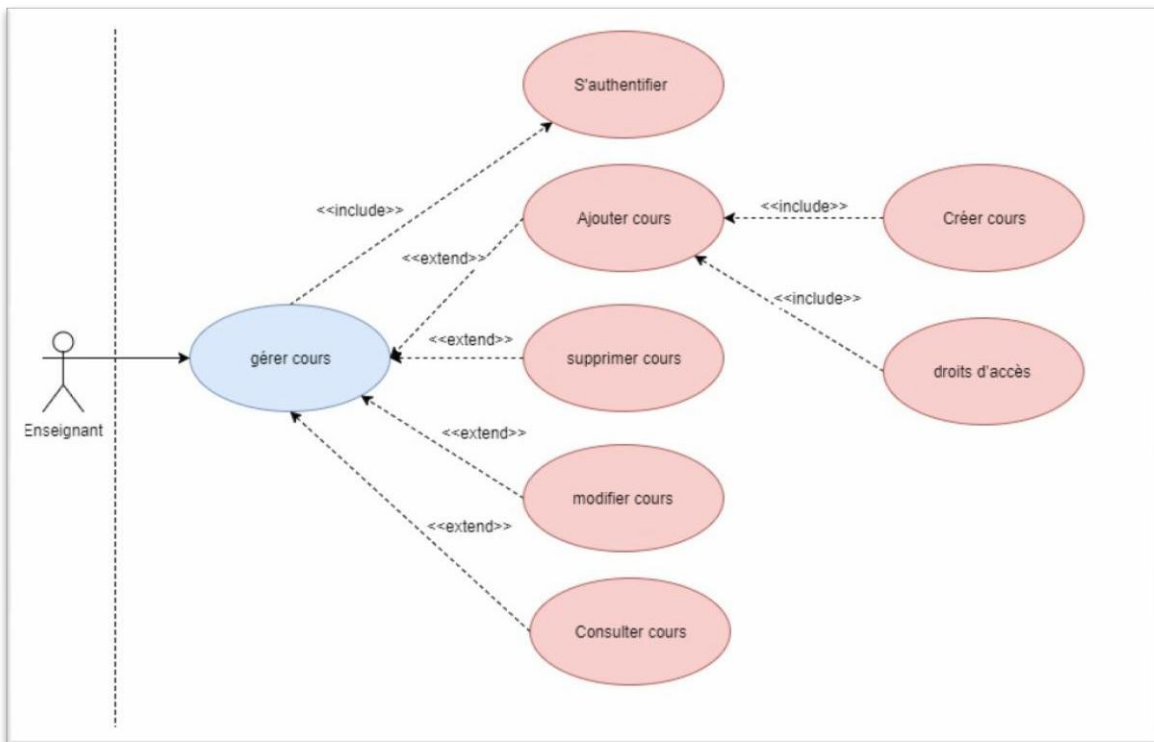


Figure 4 : Diagramme cas utilisation « gérer cours »

Gestion cours
Résumé : l'auteur peut ajouter, supprimer, modifier, consulter les cours
Acteurs : Tuteur
Pré-condition : le tuteur devrait être authentifié
Scénario <div style="text-align: right;"><début></div> <ol style="list-style-type: none"> 1- l'utilisateur doit s'authentifier a la plateforme 2- le système l'affecte a son espace personnel 3- l'utilisateur selectionne la rubrique ajouter cours puis créer cours 4- le système affiche formulaire pour remplir les informations sur le cours 5- l'utilisateur attribue les droits d'accès
6- l'utilisateur peut supprimer un cours
7- l'utilisateur peut consulter les cours

III-6- 5 - Diagramme cas utilisation « gérer plateforme »

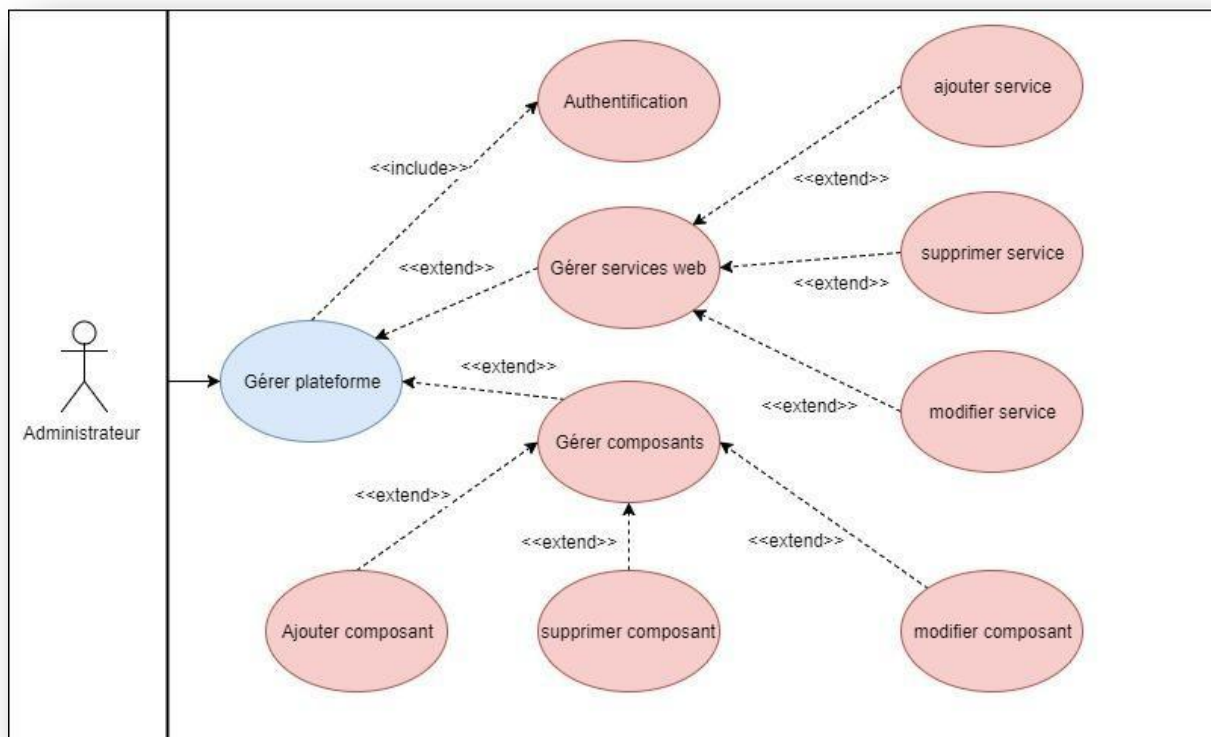


Figure 5 : Diagramme cas utilisation « gérer plateforme »

Gérer plateforme
Résumé : l'administrateur peut modifier la plateforme, et gérer les services web
Acteurs : Administrateur
Pré condition : l'acteur devrait être connecté
Scénario : <div><div><début></div><div>1- l'administrateur se connecte a la plateforme</div><div>2- l'administrateur peut ajouter, modifier, supprimer les composants de la plateforme</div><div>3- le système vérifie et met à jour les changements</div><div>4- le système affiche la nouvelle plateforme</div><div><fin></div></div>
Scénario alternatif : <div><div><début></div><div>1- l'administrateur effectue des changements</div><div>2- le système renvoi un code d'erreur et changement ne peuvent pas être effectués</div><div>3- l'administrateur accède a la plateforme précédente</div></div>
<fin>

III-6- 6 : Diagramme cas utilisation général

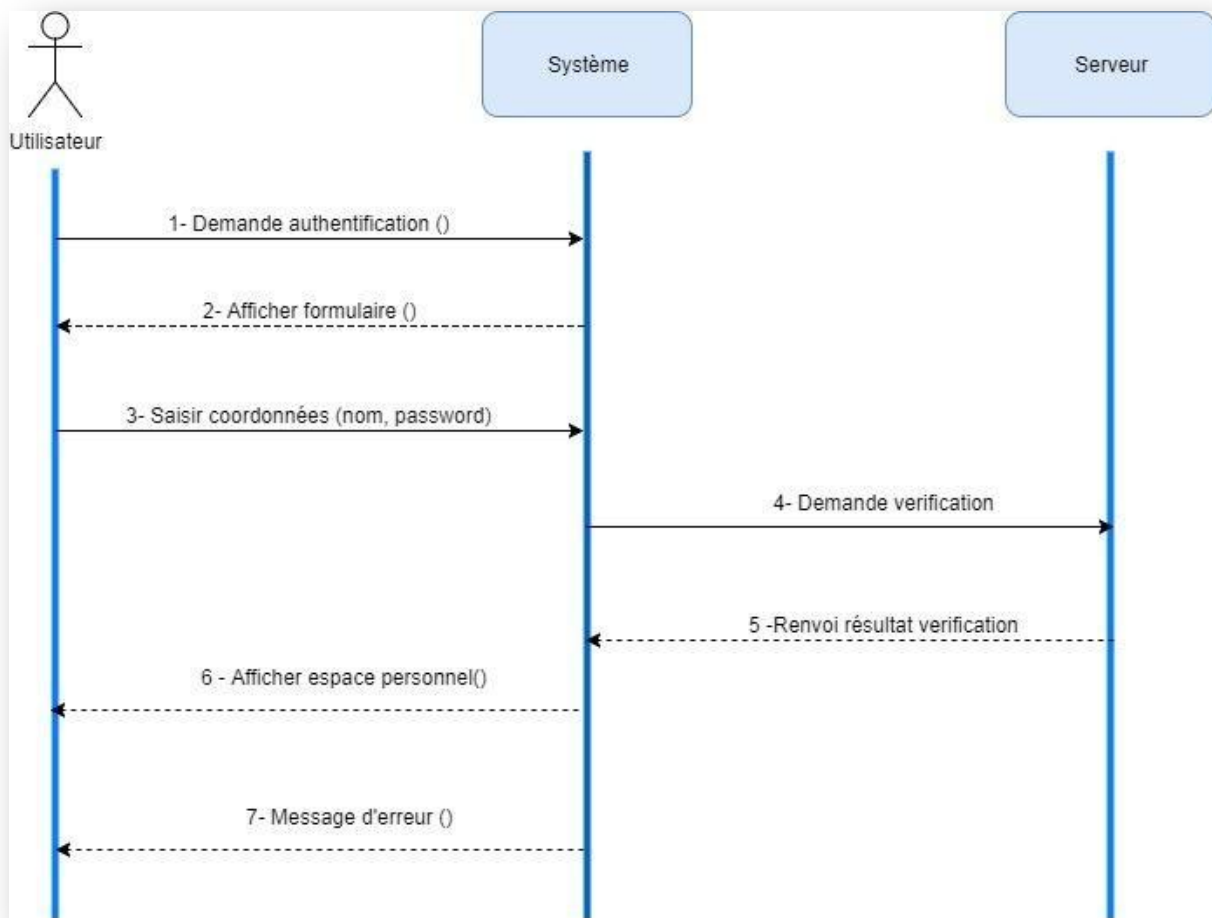


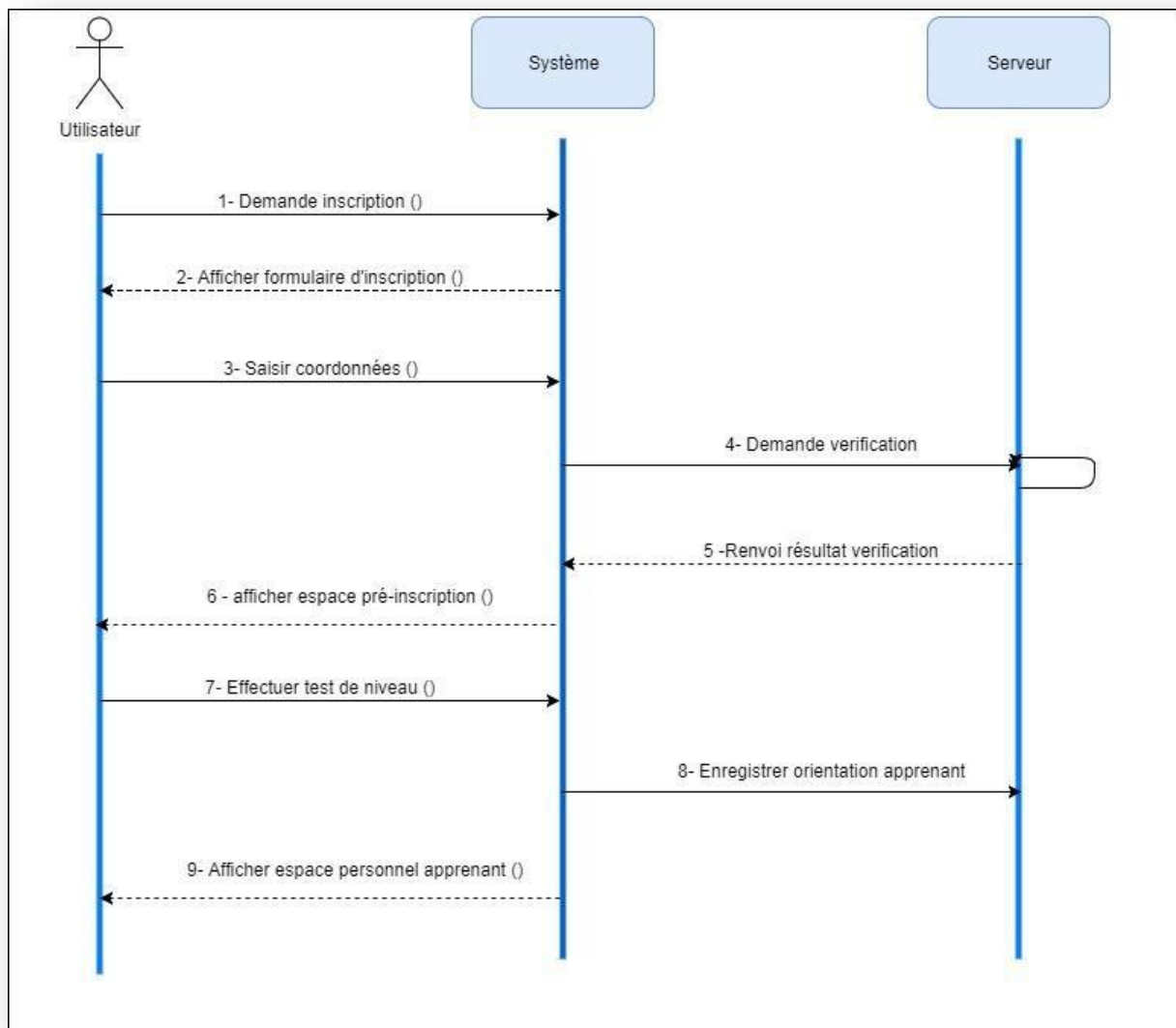
Figure 6 : Diagramme cas utilisation général

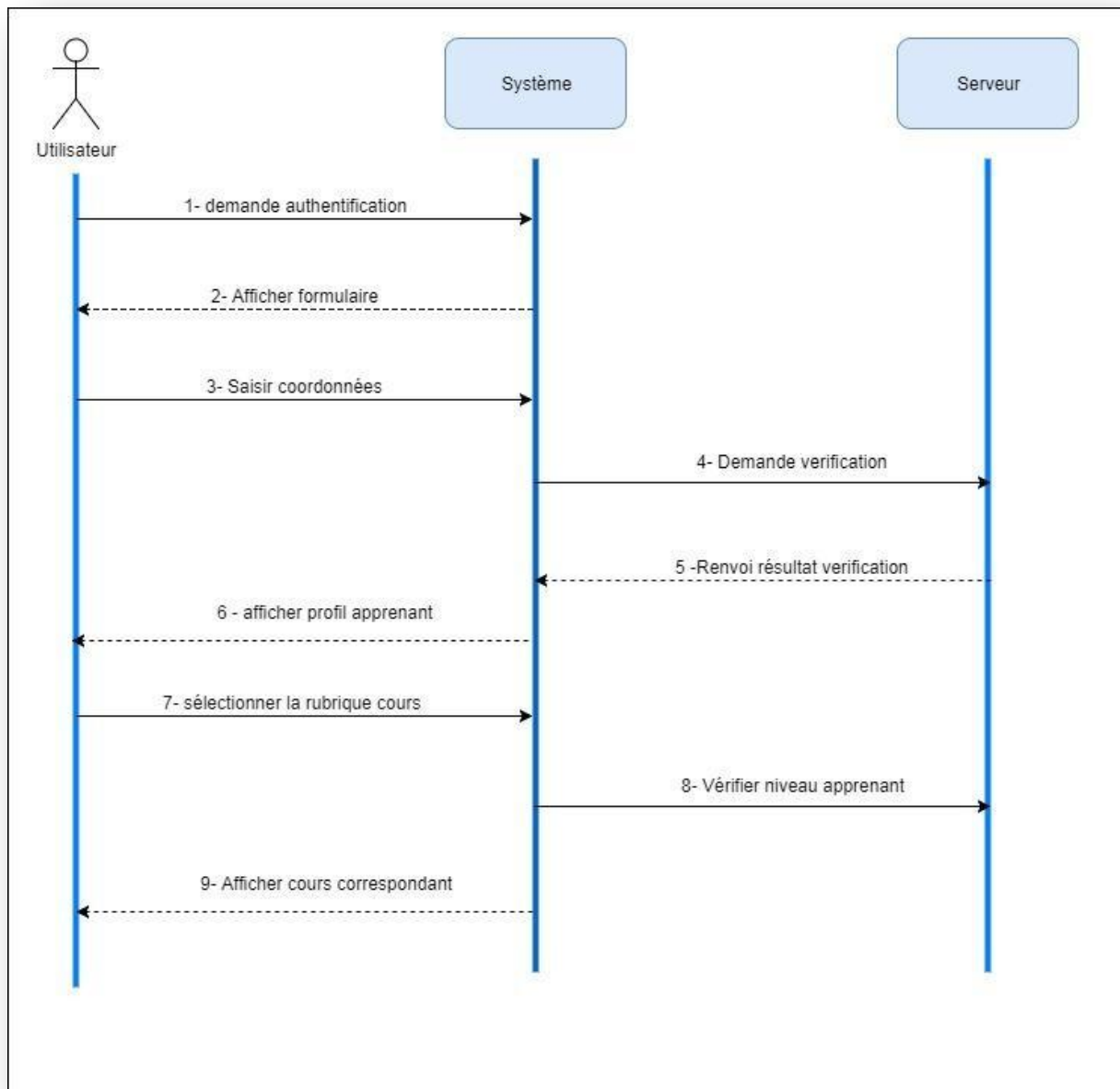
III-7- Diagrammes de séquence :

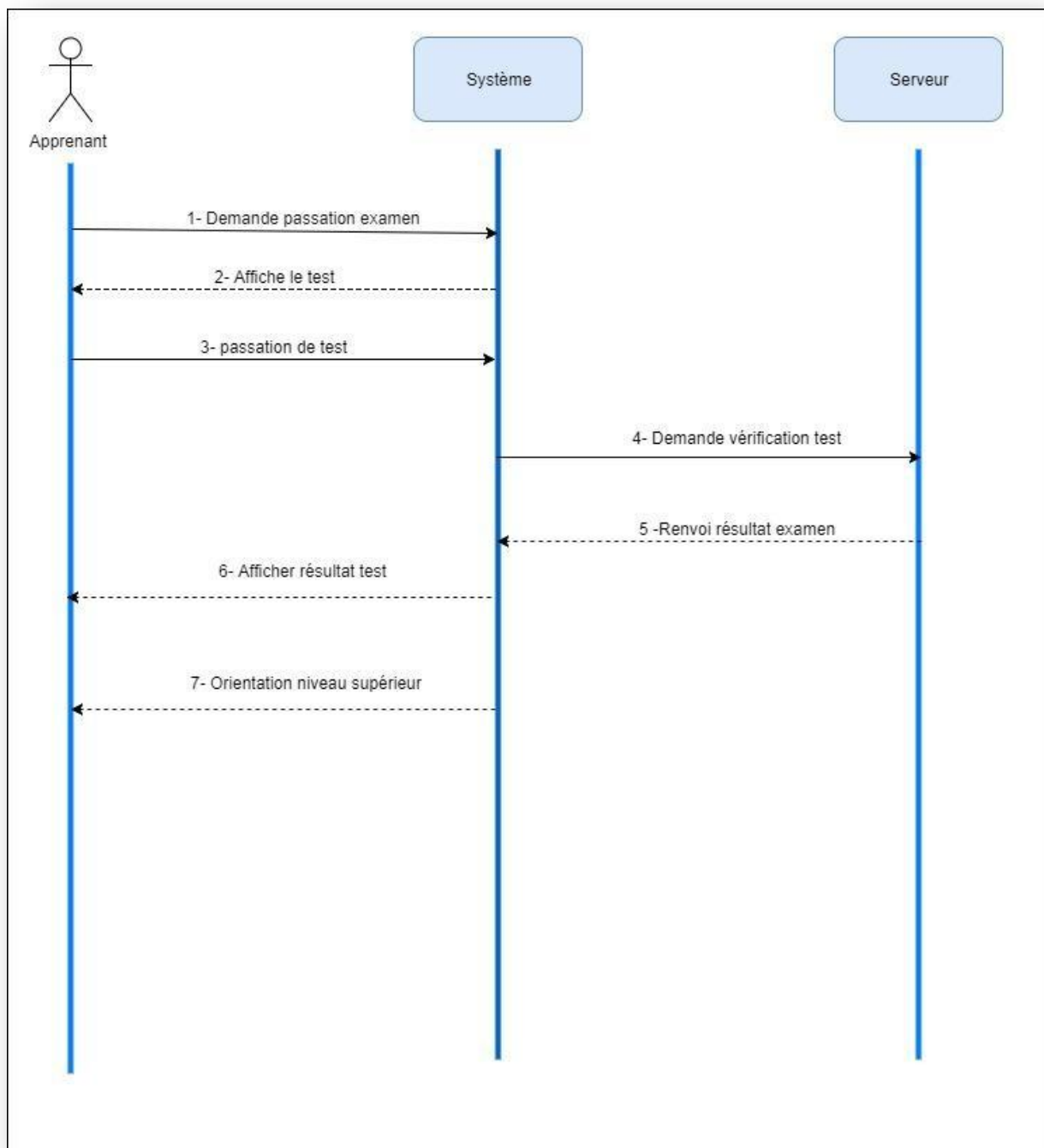
III-7-1 Définition

Le diagramme de séquence est utilisé pour une représentation claire et précise des fonctionnalités fournies. Ce diagramme offre une représentation dynamique du système. Il montre pas à pas le déroulement des actions constituant le processus de publication par ordre chronologique.

III-7-2- Diagramme de séquence cas « authentification »**Figure 8 : Diagramme de séquence cas « authentification »**

III-7-3 Diagramme de séquence cas « inscription apprenant »**Figure 9 : Diagramme de séquence cas « inscription apprenant »**

III-7-4 Diagramme de séquence cas « consulter cours »**Figure 10 : Diagramme de séquence « consulter cours »**

III-7-5 Diagramme de séquence cas « effectuer examen »**Figure 11 : Diagramme de séquence « Effectuer examen »**

III-8- Diagrammes de séquence d'analyse détaillée des cas d'utilisation :

Les objets d'analyse d'un système sont des instances de classe d'analyse qui représentent des éléments ayant des comportements et des responsabilités dans le système, ce sont les diagrammes de classes qui illustrent les classes d'analyse et les relations qu'elles entretiennent.

Les classes d'analyse peuvent être répartis dans les catégories suivantes : [14] « Arnal, Philippune (JDNet) : " Solutions de e-Learning : voyage dans la jungle des plateformes ", Novembre 2003. »

a. Les objets d'interface (Boundary) :

Un objet d'interface représente l'interface entre l'utilisateur et le système, et dans le cas d'application Web il représente les pages Web complètes.

L'icône  :

b. Les objets entité (entity) : Ce sont des objets décrits dans le cas d'utilisation mais qui lui survivront, comme par exemple les adhérents et les ouvrages dont leurs instances peuvent apparaître dans de nombreux cas d'utilisation.

L'icône : 

c. Les objets de contrôle :

Ce sont les activités faites par les processus du système. Ils dirigent les activités des objets interface et entité.

L'icône : 

Ces classes d'analyse entretiennent des relations qui sont illustrées dans des diagrammes de classes. Dans ces diagrammes, les associations entre objets obéissent à certaines règles qui sont :

1. les acteurs n'interagissent qu'avec des objets d'interface.
2. les objets entité n'interagissent qu'avec les objets contrôle.
3. les objets contrôles interagissent avec tous les objets, y compris des instances d'objets contrôle.

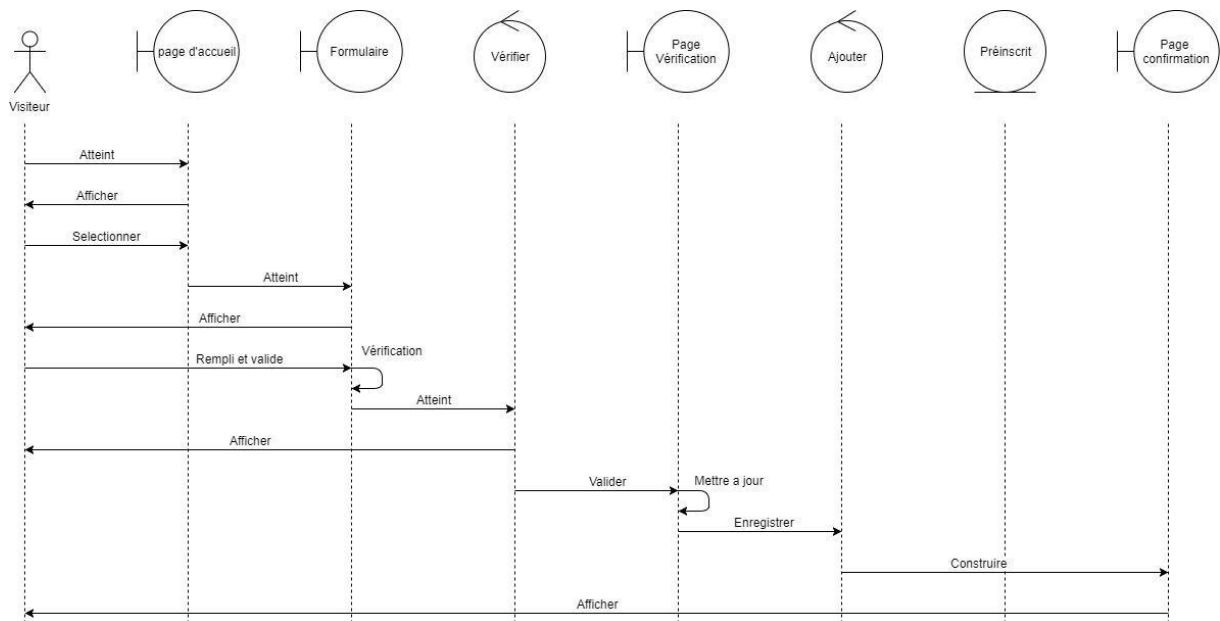
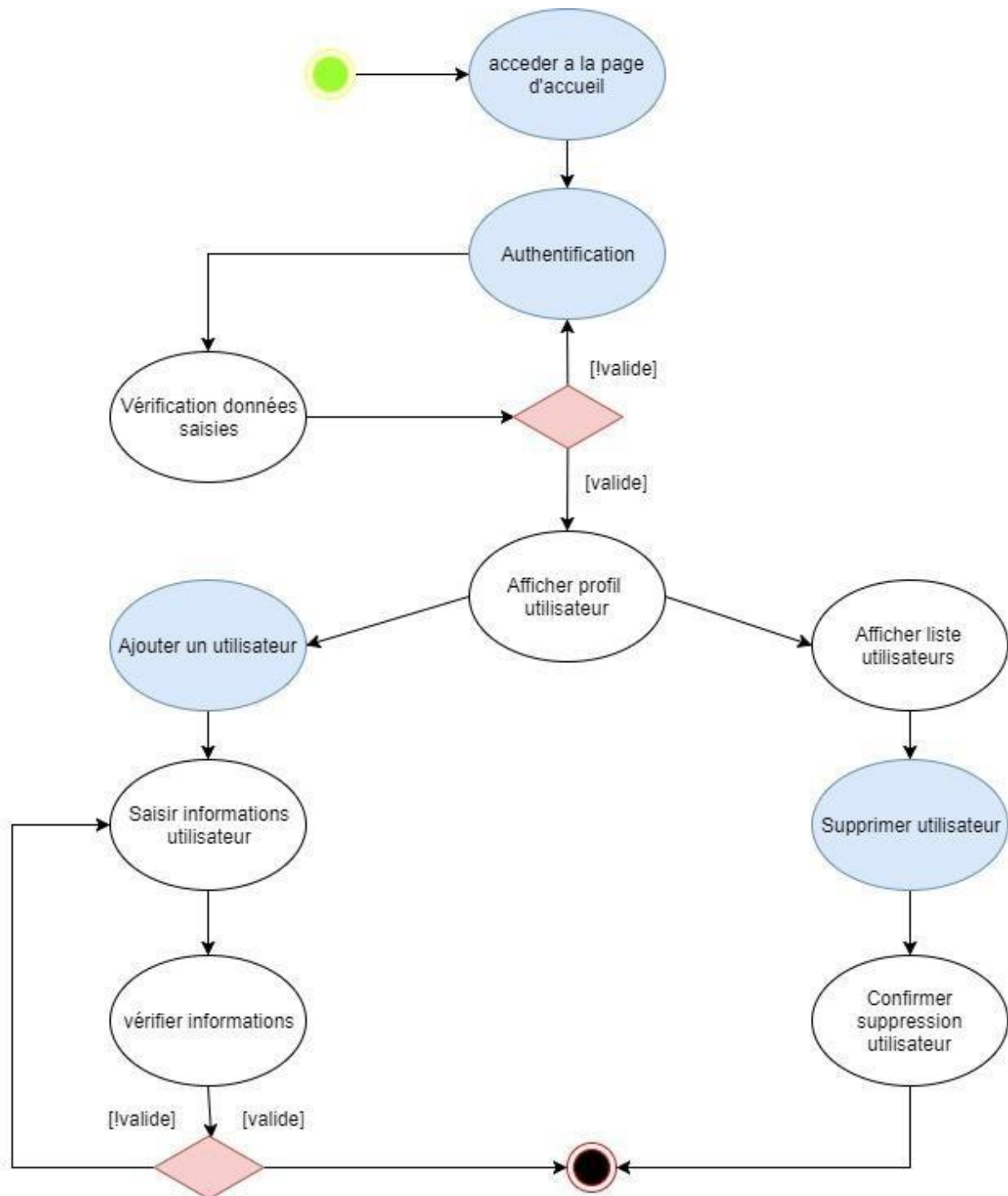


Figure 12 : diagramme de séquence détaillé « Inscription »

III-9- Diagramme d'activité

Le diagramme d'activités permet de mettre l'accent sur les traitements. Il est donc particulièrement adapté à la modélisation du cheminement de flots de contrôle et de flots de données. Il permet ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

III-9-1 Diagramme d'activité cas « Gestions utilisateurs »**Figure 13 : Diagramme d'activité « Gestion d'utilisateur »**

III-9-2 Diagramme d'activité cas « gestion cours »

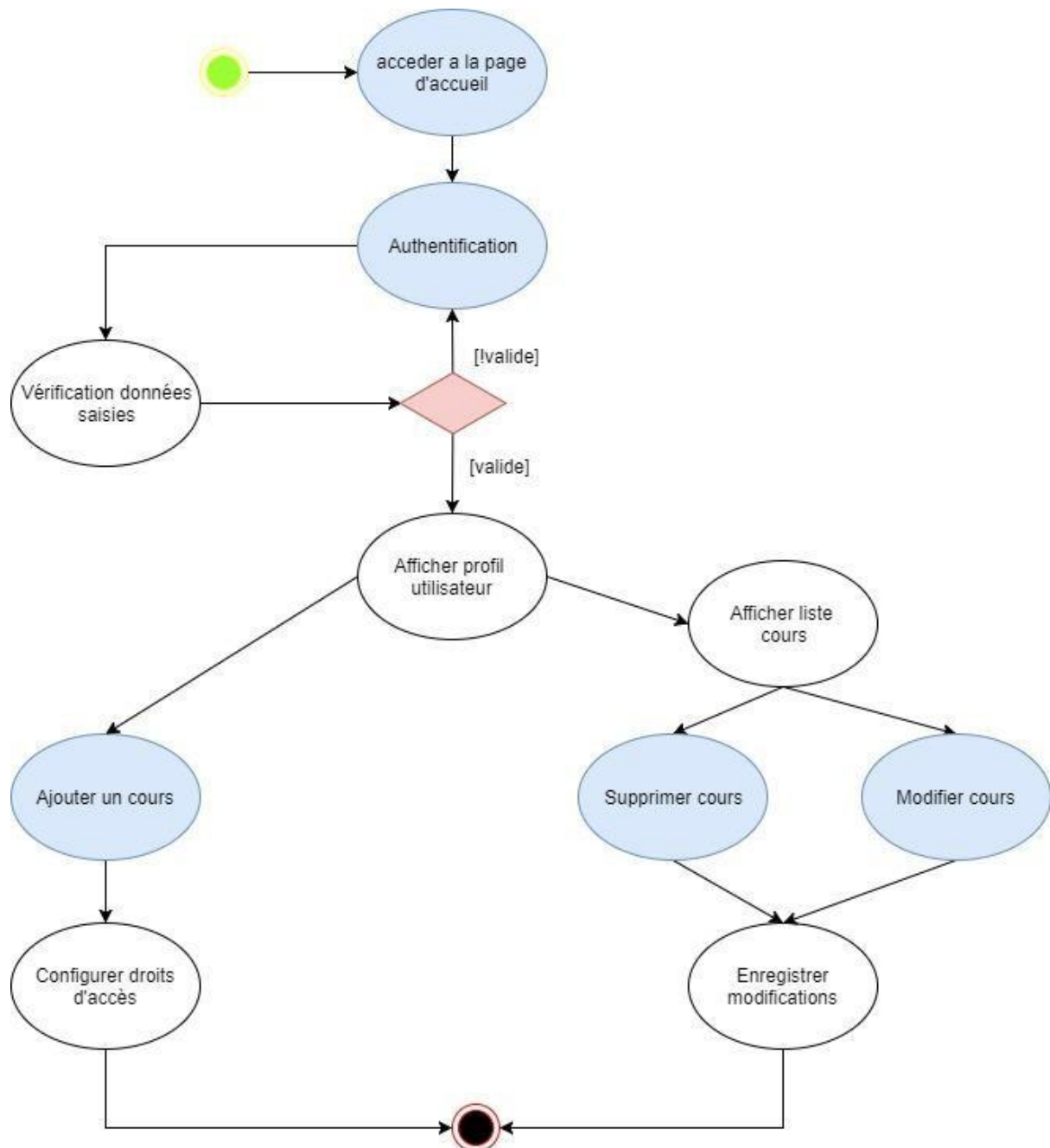


Figure 14 : Diagramme d'activité « Gestion cours »

III-10- Diagramme de classe

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Le diagramme de classe de notre projet est représenté comme suit :

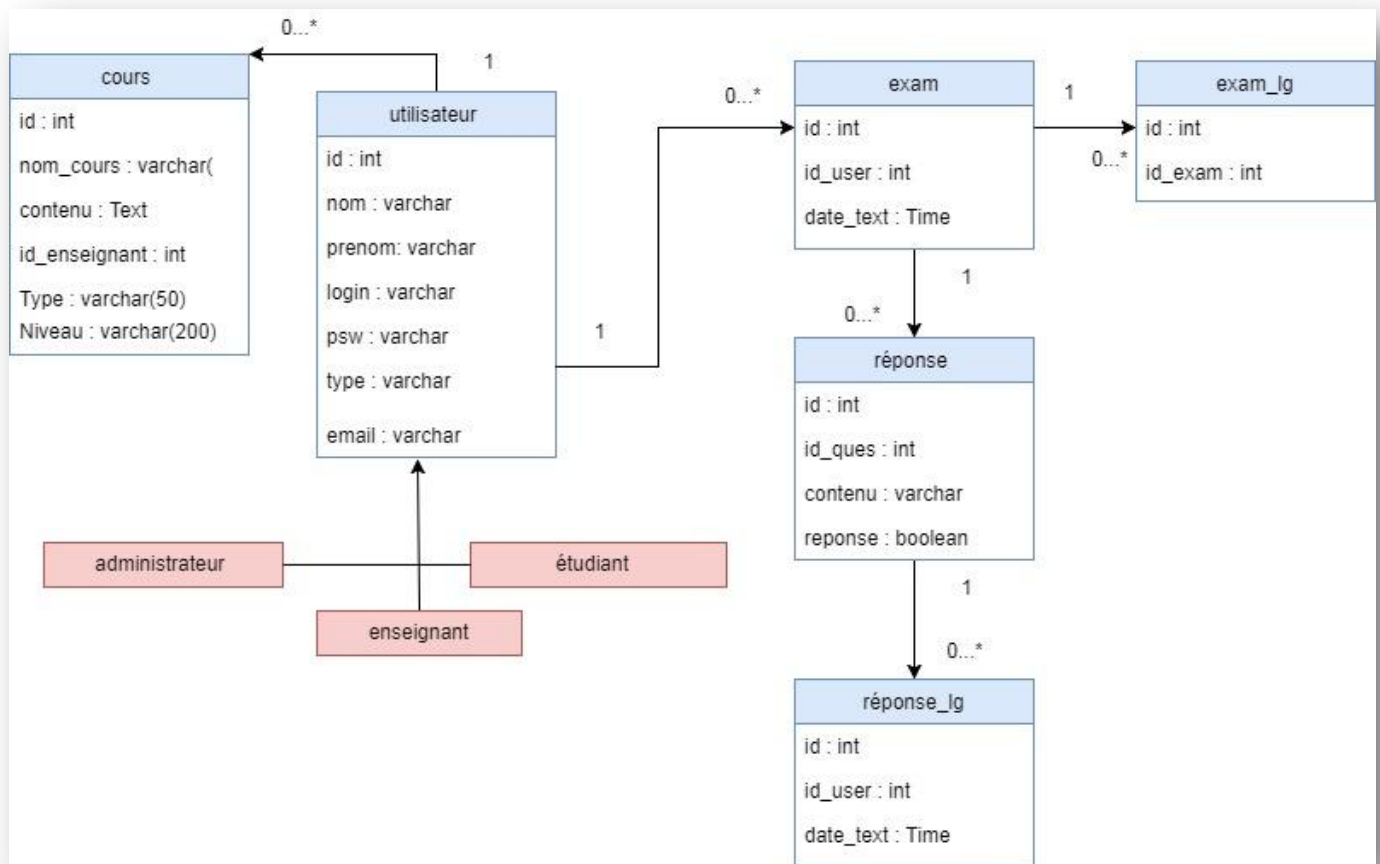


Figure 15 : Diagramme de classe général

11- Conclusion

Dans ce chapitre, nous avons présenté une vue conceptuelle du système à réaliser. Ainsi, on a pu définir le diagramme de classe, les diagrammes détaillés des cas d'utilisation de notre projet, suivis de quelques diagrammes de séquences et d'activités. Nous pouvons entamer la phase suivante qui est la phase de réalisation de la solution.

Chapitre IV

Réalisation

IV-1- Serveur de base de données

IV-1-1 PostgreSQL



PostgreSQL est un système de gestion de bases de données relationnel robuste et puissant, aux fonctionnalités riches et avancées, capable de manipuler en toute fiabilité de gros volumes de données, mêmes dans des situations critiques.

Il fonctionne sur les principaux systèmes d'exploitation : Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) et Depuis la version 8.0, PostgreSQL fonctionne également sur Windows.

. Il supporte plus d'une douzaine de langages de programmation, dont Java, Perl, Python, Ruby, Tcl, C / C ++. Son propre PL / pgSQL est similaire à PL / SQL d'Oracle.

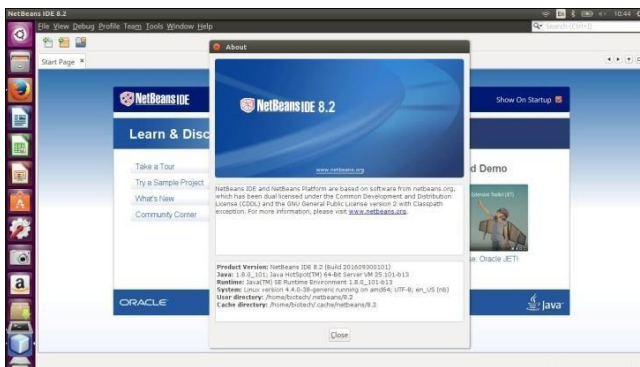
IV-2- Draw.io



Draw.io est une application gratuite en ligne, accessible via son navigateur (protocole https) qui permet de dessiner des diagrammes ou des organigrammes. Cet outil propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML, JPEG puis de les exporter.

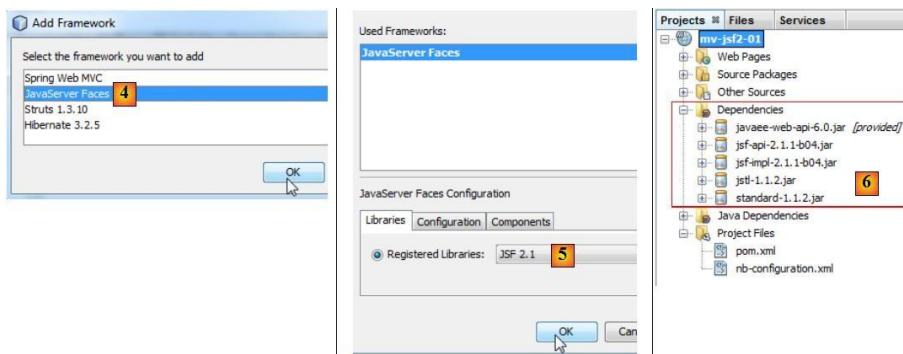
L'interface est simple et facile d'utilisation, les objets sont classés à gauche de l'écran par thèmes, un moteur de recherche permet d'interroger la base de données de dessins. Une fois le travail terminé, la sauvegarde est réalisée au format XML (Diagramly XML Document), l'exportation propose différents formats (png, jpg, gif, svg, html, intégration en pages web...). Il est possible d'importer un diagramme au format XML.

IV-3- Netbeans



NetBeans IDE est un logiciel de développement open source pour le développement de programmes sur Java, qui est un des langages de programmation les plus importants de nos jours.

IV-4- Framework de développement JSF 2.2 (Java Server Face)



Les Framework se présentent sous diverses formes, qui peuvent inclure tout ou partie des éléments suivants :

- un ensemble de classes généralement regroupées sous la forme de bibliothèques pour proposer des services plus ou moins sophistiqués
- un cadre de conception reposant sur les design patterns pour proposer tout ou partie d'un squelette d'application
- des recommandations sur la mise en œuvre et des exemples d'utilisation
- des normes de développement
- des outils facilitant la mise en œuvre

JSF permet :

- une séparation nette entre la couche de présentation et les autres couches
- le mapping HTML/Objet (voir Annexes)
- un modèle riche de composants graphiques réutilisables
- une gestion de l'état de l'interface entre les différentes requêtes
- une liaison simple entre les actions côté client de l'utilisateur et le code Java correspondant côté serveur [1]
- la création de composants customs grâce à une API
- le support de différents clients (HTML, WML, XML, ...) grâce à la séparation des problématiques de construction de l'interface et du rendu de cette interface

IV-5- Service Web rest

IV-5-1- Création d'un service REST sous Netbeans

Techniquement un service REST est une application web. Il nous faut donc créer un projet web sous Netbeans, comme indiqué sur la figure suivante.

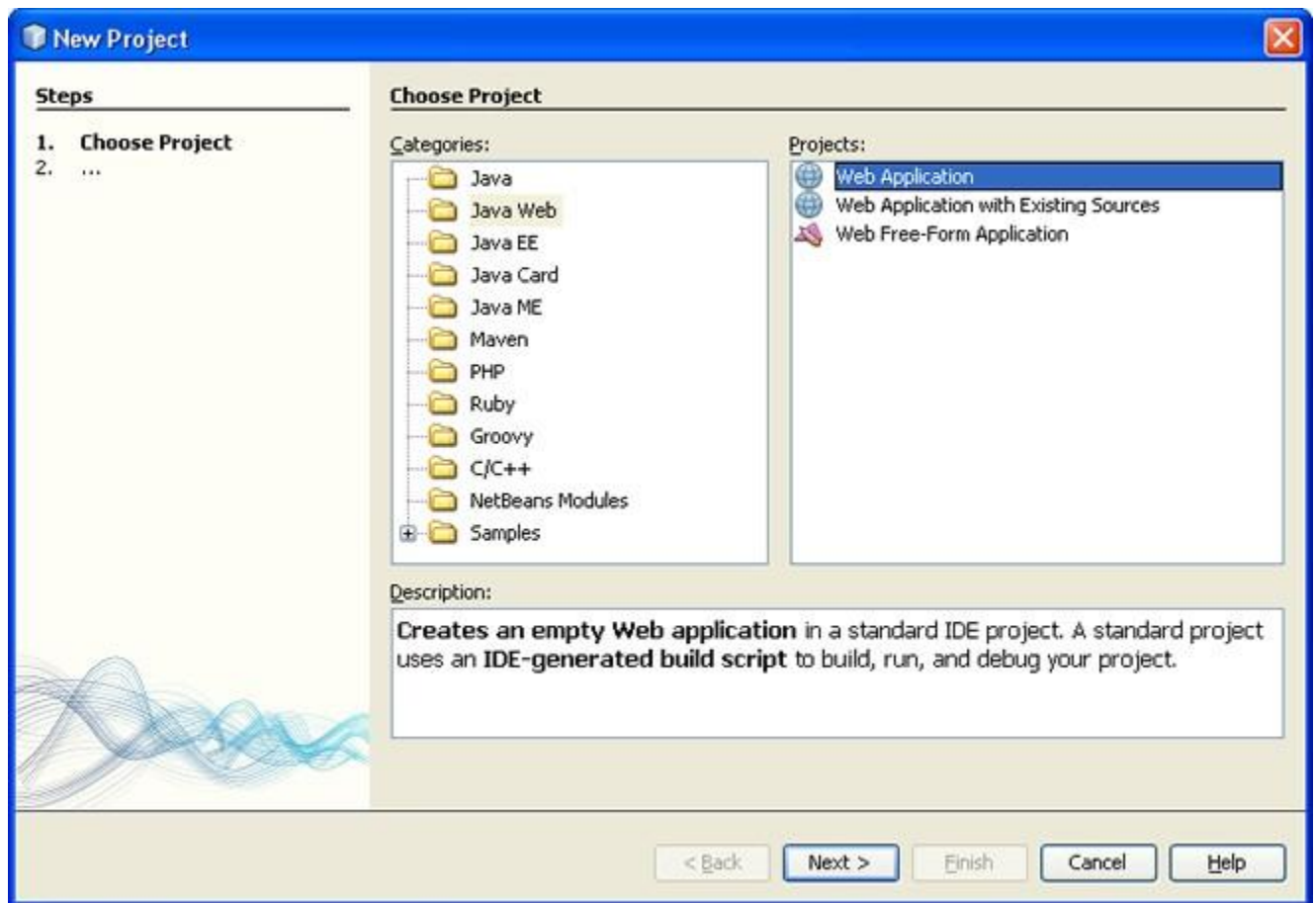


Figure 1. Création d'un projet web sous Netbeans

Ensuite, Netbeans nous permet de créer directement une classe qui implémente un service REST. Même si cette classe elle-même est très simple à écrire, l'infrastructure technique que Netbeans créé également n'est pas négligeable. Pour cela, il faut créer un nouveau fichier, et choisir par exemple "RESTful webservice from pattern" dans l'onglet "Web Services" du créateur de fichier.

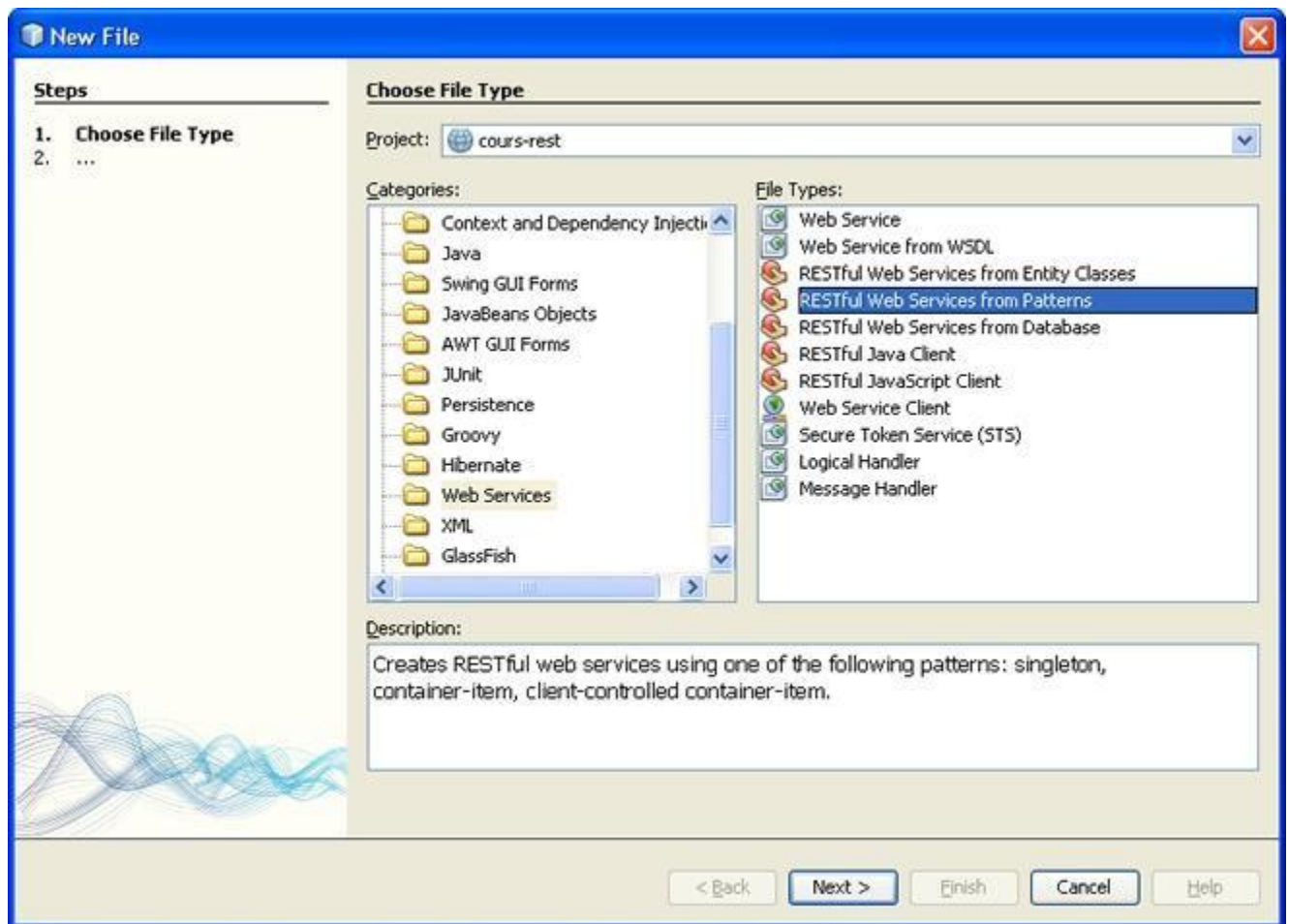


Figure 2. Création d'un service REST

L'étape suivante consiste à choisir un pattern pour ce service. On sélectionne ici "Simple root ressource".

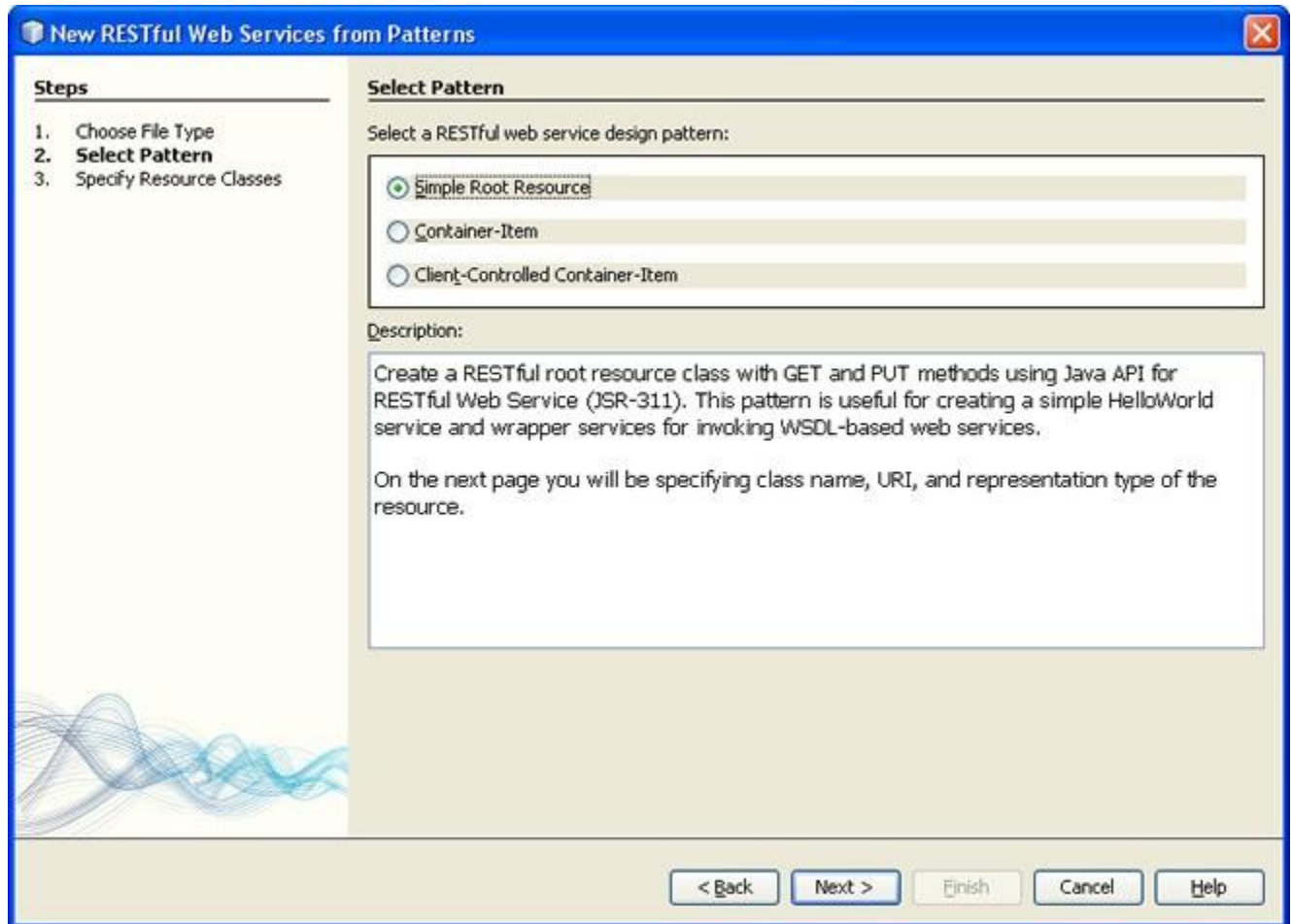


Figure 3 : Sélection du pattern "Simple root ressource"

Enfin la dernière étape nous demande de préciser la classe qui va porter ce service. De plus, on choisit "rs" comme chemin d'accès pour cette ressource. On laisse leurs valeurs par défaut aux autres paramètres.

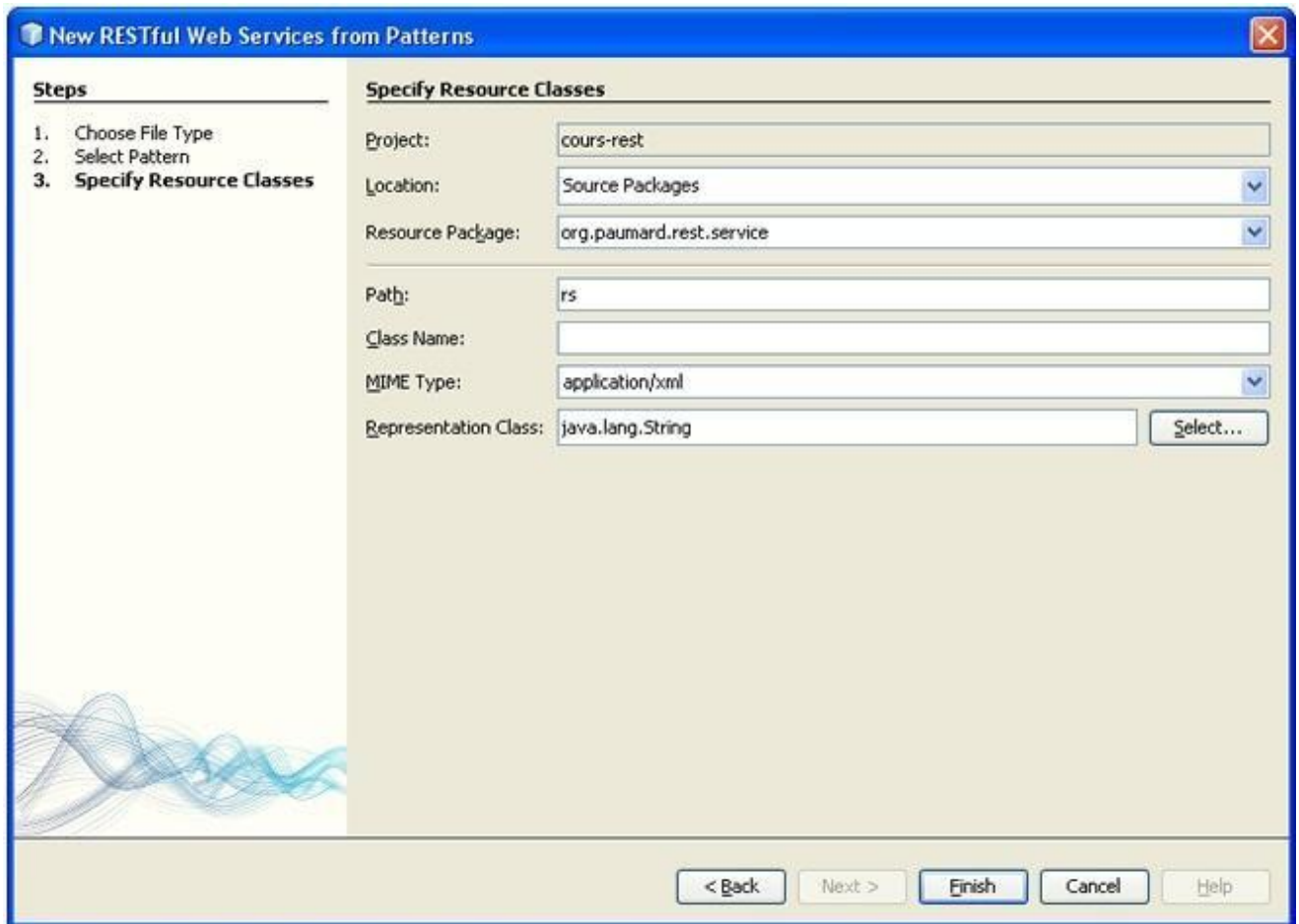


Figure 4. Paramétrage final du service

Netbeans nous crée alors une classe, CoursService, qui porte ce service REST. Examinons cette classe.

Exemple : Un premier service REST : CoursService

```
@Path("rs")
public class CoursService {
    @Context
    private UriInfo context;

    @GET
    @Produces("application/xml")
    public String getXml() {
        // contenu de la méthode
    }

    @PUT
    @Consumes("application/xml")
    public void putXml(String content) {
```

```
// contenu de la méthode
    }
}
```

Remarquons les choses suivantes sur cette première classe.

- Il s'agit d'une classe Java tout à fait standard : elle n'étend aucune classe particulière, et n'implémente aucune interface. Elle possède un constructeur vide, ici le constructeur vide par défaut. Le point important, est qu'elle est annotée par `@Path("rs")`. C'est la présence de cette annotation qui fait de cette classe un service REST.
- Elle possède deux méthodes, peu importe leur nom, annotées par `@GET` et `@PUT`. Ces deux annotations correspondent à deux méthodes HTTP. Lors d'une requête de type get, c'est la méthode annotée `@GET` qui sera invoquée. Même chose pour une requête de type put : c'est la méthode annotée `@PUT` qui sera invoquée.
- Enfin les deux annotations `@Consume` et `@Produce` précisent le format des données qui seront reçues ou émises par chacune de ces méthodes : ici du XML.

Un deuxième élément a été créé, qui se trouve dans le fichier `web.xml` de cette application web. Examinons les éléments que Netbeans a ajoutés pour nous.

Exemple : Fichier `web.xml` du service REST

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app version="2.5"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

  <servlet>
    <servlet-name>ServletAdaptor</servlet-name>
    <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>ServletAdaptor</servlet-name>
    <url-pattern>/resources/*</url-pattern>
  </servlet-mapping>

  <!-- reste du fichier web.xml -->
</web-app>
```

On constate que Netbeans a créé une servlet, dont le nom est servlet ServletAdaptor, et la classeServletContainer. Cette servlet traite toutes les requêtes qui arrivent sur toutes les URL commençant par/resources.

Le fonctionnement d'un service REST apparaît ici. Techniquement, il s'agit d'une unique servlet classique, vivant dans une application web standard. Cette servlet connaît les classes possédant l'annotation @Path, et est capable de distribuer vers elles les requêtes qui les concernent. Nous reverrons ce point plus en détails dans la suite de ce chapitre.

2-3- Création d'un modèle

Ajout d'une classe cours à notre projet, avec des annotations JAXB.

Exemple- Classe Cours pour notre service REST

```
@XmlRootElement(name="cours",
    namespace="http://www.netbeans.org/ns/projectrest")
public class cours {

    @XmlAttribute
    private long id ;

    private String nom ;

    // suite de la classe
}
```

Cette classe est tout à fait standard par rapport à celles que nous avons créées dans la partie précédente.

2-4- Modification du service REST

Modifions à présent notre service REST pour la prendre en compte.

Exemple : Version modifiée sur service REST CoursService

```
@Path("cours/{id}")
public class CoursService {

    @GET
    @Produces("application/xml")
    public Object getCours(@PathParam("id") long id) {

        Cours cours = new Cours() ;
        cours.setId(id) ;
        cours.setNom("vocabulaire") ;
    }
}
```

```
        return cours ;  
    }  
}
```

De la classe originale, nous n'avons pas gardé grand chose. Regardons les éléments que nous avons ajoutés.

- Nous avons modifié la valeur de l'annotation `@Path`. Sa valeur est maintenant `/main/{id}`. La partie `{id}` signifie qu'une URL de la forme `/cours/15` sera acceptée, et qu'un paramètre de chemin `id` sera créé pour cette requête, de valeur `15`.
- Notre méthode `@GET` a changé de nom, ce qui n'a pas d'importance, vu que ce qui compte c'est son annotation et non pas son nom. En revanche, son type de retour est maintenant `Object` plutôt que `String`. Elle prend un paramètre, qui est annoté par `@PathParam("id")`. Cette annotation signifie que la valeur du paramètre de cette méthode correspondra à la valeur du paramètre de chemin `id`.
- Enfin dans le corps de cette méthode, on crée un cours, en fixant la valeur de son `id` à la valeur passée en paramètre. Les autres champs sont classiques. On imagine qu'à terme, ce cours sera lu en base, soit directement, soit au travers d'un EJB *session facade*.

L'objet retourné par cette méthode est simplement `cours`. Le fait que cette méthode soit annotée par `@Produces("application/xml")`, et que notre classe `Cours` porte des annotations JAXB, va automatiquement déclencher le *marshalling* de cet objet en XML, par JAXB.

2-5- Invocation du service REST

Avant de pouvoir utiliser ce service, il nous faut encore modifier le fichier `web.xml` généré par Netbeans. Cette modification est mineure : il suffit de remplacer le contenu de l'élément `url-pattern` par `/*`.

Lançons ce service, en utilisant l'option "Deploy" du menu contextuel dans Netbeans.

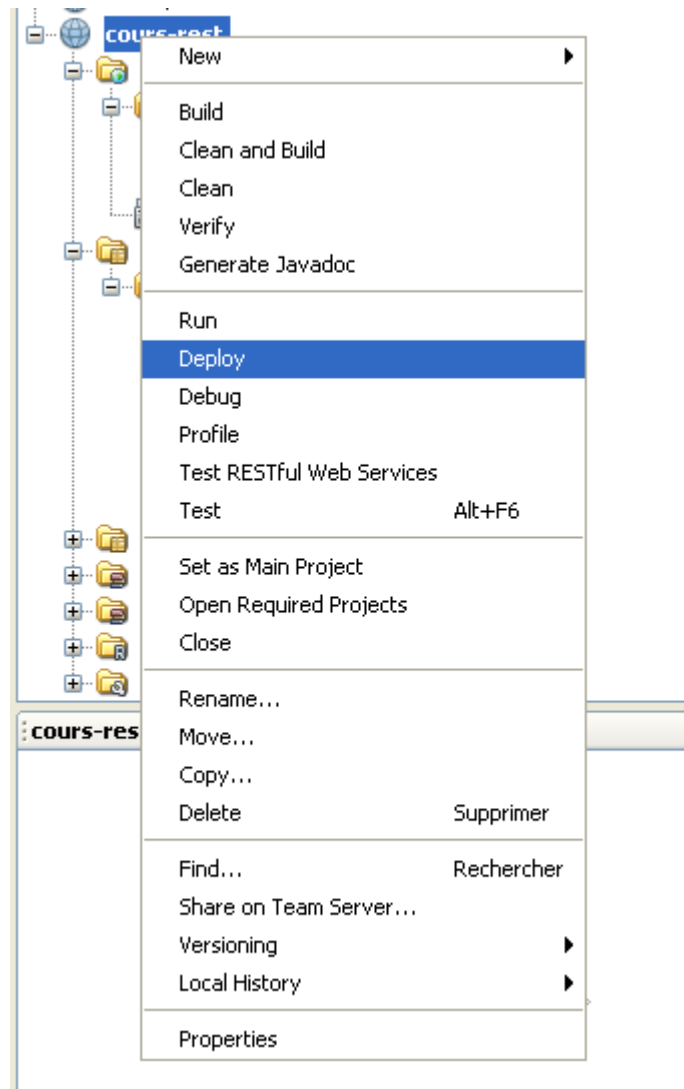


Figure 5. Lancement de notre premier service REST

Enfin, pour l'utiliser, il suffit de faire une requête à l'aide d'un navigateur standard sur l'URL utilisée par Netbeans. Il s'agira d'une URL du type :

<http://localhost:8008/cours-rest/cours/15>

Cette URL est composée de trois éléments.

- <http://localhost:8008> : correspond à l'URL sur laquelle le serveur utilisé par Netbeans est lancé (Glassfish ou Tomcat le plus souvent). Cette partie de l'URL est à adapter en fonction de la configuration locale.
- [/cours-rest](#) : le point de montage utilisé par Netbeans pour déployer une application web correspond au nom du projet qui porte cette application. Cette URL doit donc aussi être adaptée en fonction de la configuration locale.
- [/cours/15](#) : cette dernière partie de l'URL permet de s'adresser à notre service web tel que nous l'avons défini.

On reçoit le document XML suivant en réponse à cette requête.

Exemple- Document XML servi

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<ns2:cours xmlns:ns2="http://www.netbeans.org/ns/project">
  <nom>vocabulaire</nom>

</ns2:cours>
```

VI-6 Interfaces de la plate-forme

Dans ce qui suit, nous présenterons les principales interfaces de notre plate-forme d'enseignement a distance « English for life »

- page d'accueil

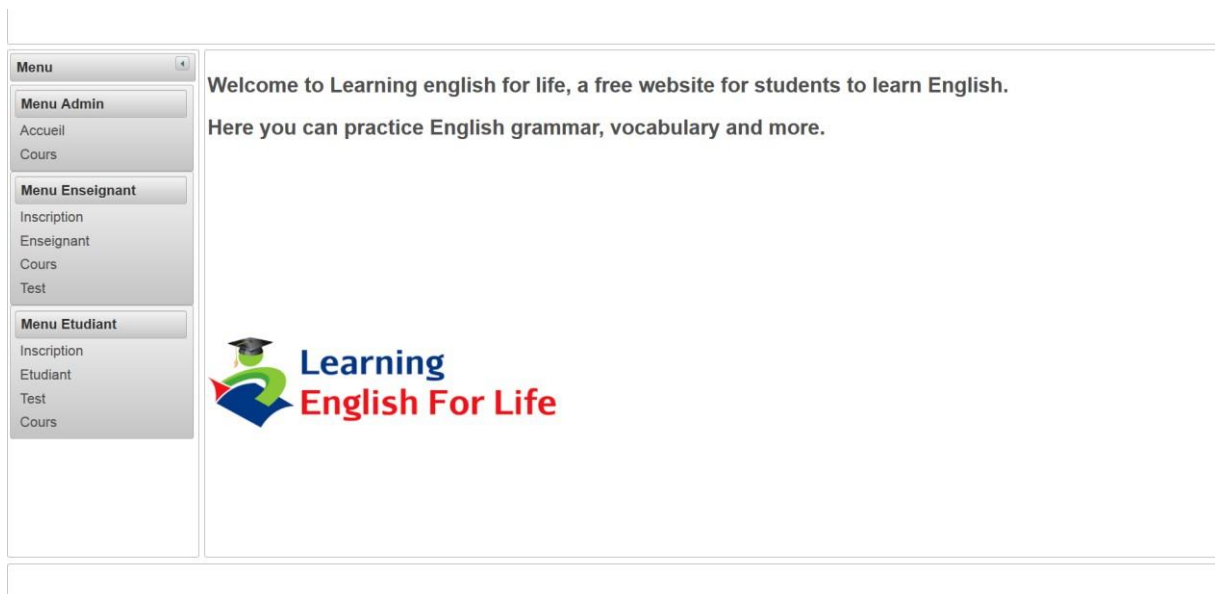


Figure 1 : Page d'accueil

C'est la première page qui apparait dans le navigateur lors de la connexion au site.

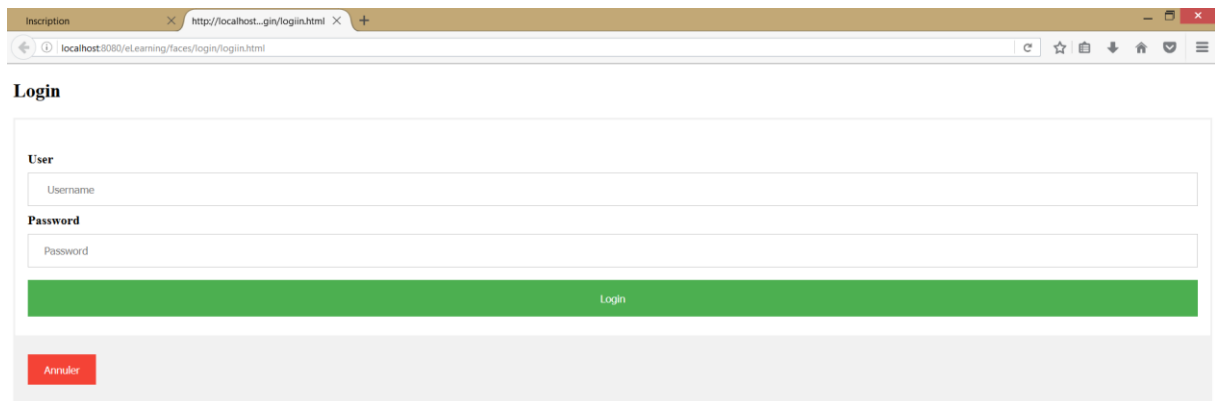
La page d'accueil de notre plateforme résume le fonctionnement de système, on y trouve une explication des fonctionnalités de l'environnement, avec des liens actifs vers d'autres pages donnant plus d'explications.

- Accueil : il permet de rester dans la page d'accueil de l'application
- Apprenant : ce lien mène vers le lien d'identification de l'apprenant pour demander l'accès a son espace
- Enseignant : ce lien mène vers la page d'identification de l'enseignant pour demander

l'accès a son espace

- Administrateur : ce lien vers la page d'identification pour acceder a son espace
- Inscription : ce lien permet aux visiteurs de ce site de s'inscrire dans la plateforme come étant apprenant ou enseignant

➤ Authentification



Login

User

Username

Password

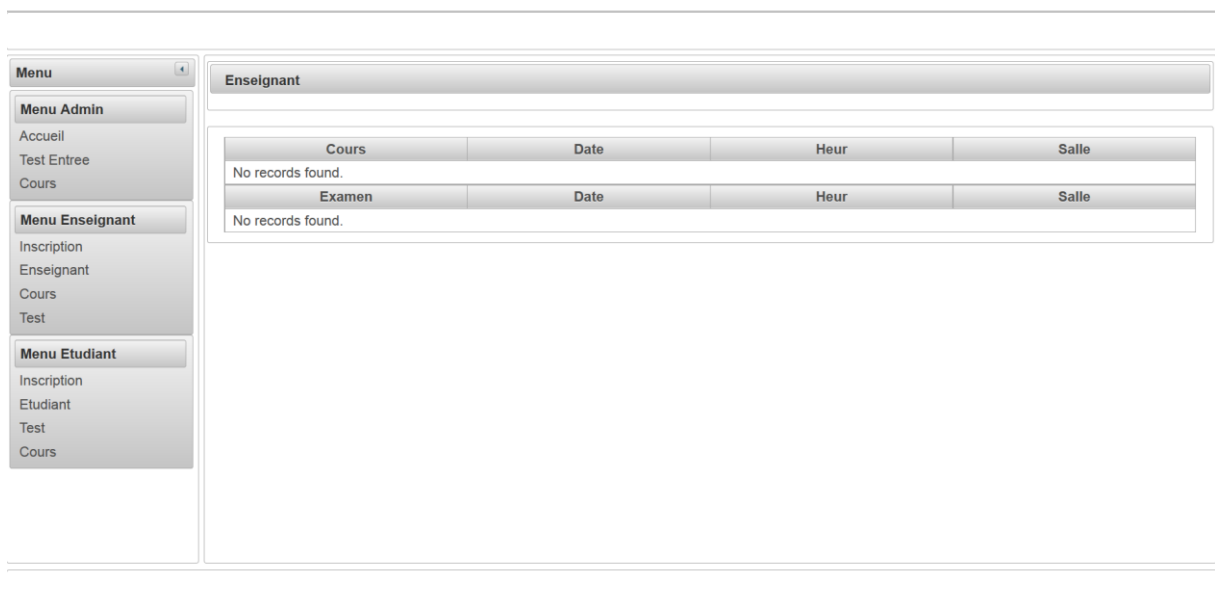
Password

Login

Annuler

Figure 2 : Page authentification enseignant ou apprenant

➤ Espace enseignant



Menu

Menu Admin

Accueil

Test Entree

Cours

Menu Enseignant

Inscription

Enseignant

Cours

Test

Menu Etudiant

Inscription

Etudiant

Test

Cours

Enseignant

Cours	Date	Heur	Salle
No records found.			

Examen	Date	Heur	Salle
No records found.			

Figure 3 : Espace enseignant

➤ **Ajout cours**

The screenshot shows a web interface for adding a course. On the left is a sidebar menu with three sections: 'Menu Admin' (Accueil, Test Entree, Cours), 'Menu Enseignant' (Inscription, Enseignant, Cours, Test), and 'Menu Etudiant' (Inscription, Etudiant, Test, Cours). The 'Menu Enseignant' section is active. The main area is titled 'Test' and contains a form with two input fields: 'Niveau' and 'Nom Cours', and a larger 'Contenu' field. Below the fields are two buttons: 'Ajouter' and 'Annuler'.

Figure 4 : Espace enseignant ajout cours

➤ **Ajout test**

The screenshot shows a web interface for adding a test. On the left is the same sidebar menu as in Figure 4, with 'Menu Enseignant' active. The main area is titled 'Test' and contains a form with the following fields: 'Niveau', 'Question', 'Réponse 1', 'Réponse 2', 'Réponse 3', and 'Réponse 4'. Below these fields are four radio buttons labeled 'Réponse 1', 'Réponse 2', 'Réponse 3', and 'Réponse 4'. At the bottom of the form is an 'Enregistrer' button.

Figure 5 : Espace enseignant ajout test

➤ Inscription apprenant

Figure 6 : Espace inscription apprenant

➤ Les services web implémentés dans notre application

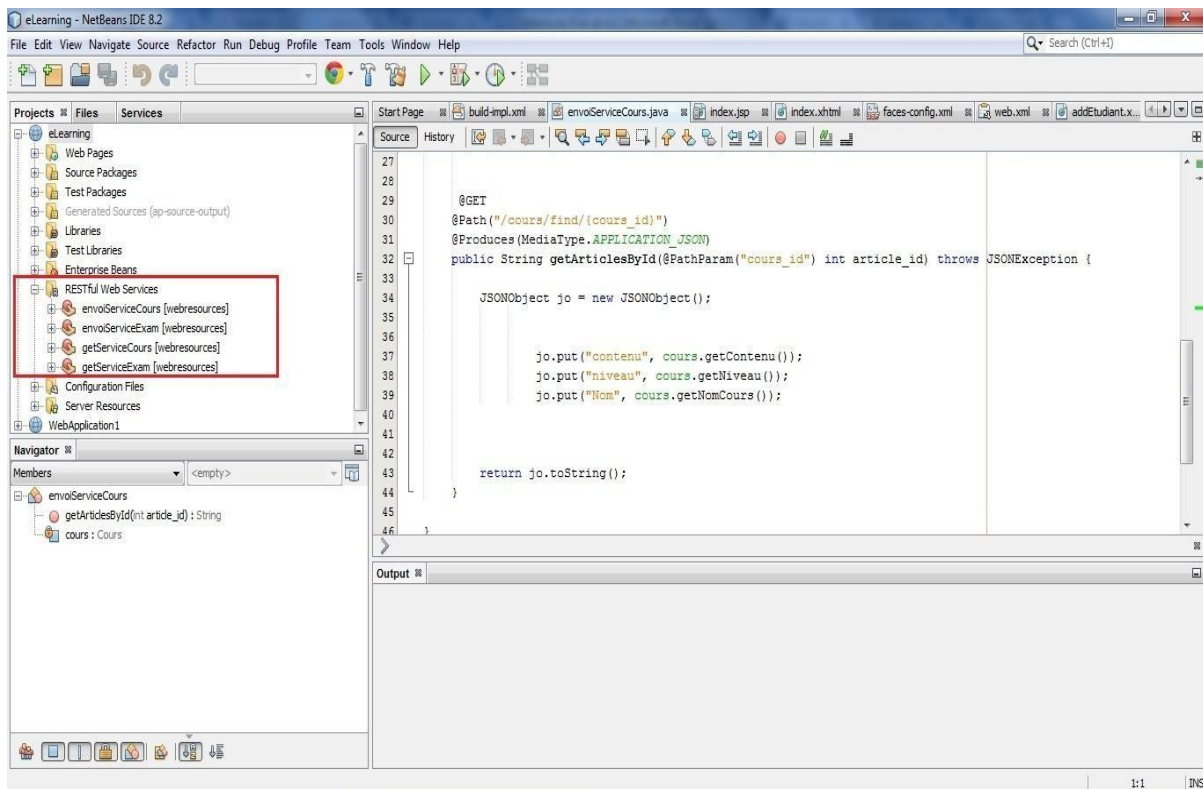


Figure 7 : Implémentation des services web de type Rest

Conclusion générale

L'objectif de notre travail était de concevoir et réaliser une application d'enseignement des cours d'anglais à distance, le système développé assure différentes tâches relatives à génération des cours, tests, exercices, corrigés types permettant ainsi :

- la gestion administrative de la formation
- l'ajout des cours et tests par les enseignants
- consultation des cours et passation d'examens par les apprenants

le principe derrière notre travail est d'utiliser les technologies des services web et plus précisément les services web de type RestFul permettant de surmonter les problèmes d'interopérabilité et intégration système

Notre travail est structuré en trois parties importantes :

1- la première partie consiste à présenter certaines définitions des services web ainsi que les technologies sur lesquels nous nous sommes basés pour la réalisation de notre projet

2- la deuxième partie consiste à présenter le domaine du E-Learning et ses fonctionnalités ainsi qu'une comparaison brève entre le système éducatif traditionnel et la formation en ligne

3- la troisième partie nous avons élaboré différents schémas de cas d'utilisation, séquence, séquence détaillée ainsi que le diagramme d'activité et conclu par le diagramme de classe général, et ensuite passer à la mise en œuvre et réalisation

Bibliographie :

- [1] institut national des normes et de technologies 1996.
- [2] Conférence de Recherche en Informatique CRI 2013
- [3] Web Services: Theory and Practice, Annura Guruge, 2004
- [4] XML et les services web, Cyril Vicent, 2013 Edition ENI
- [5] Web Services Essentials: Distributed Applications with XML-RPC, Ethan Cerami, 2012
- [6] guide pratique de E-Learning, Sid Ahmed Benrouane, 2011
- [7] : E-Learning : ses fondements et son utilisation dans le secteur bancaire, Genève Place Financière, 2015