

MA-004-499-1

République Algérienne  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Saad Dahlab Blida

N° D'ordre : .....



Faculté des sciences

Département d'informatique

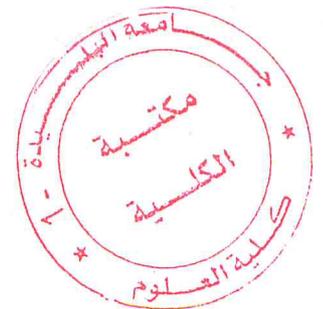
Mémoire Présenté par :

TEMAZOUT Mohamed Lamine      YAHIAOUI Housseem

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique  
Spécialité : Informatique  
Option : génie des systèmes informatique



Sujet : Développment d'un MOOC universel sous Cloud dédié à INJAZ El-Djazair.

Soutenu le : 13/10/2016

Devant le jury :

M. BAOUYA  
M. CHERFA IMENE  
M. ARKAM Meriem

Président  
Examineur  
Promoteur

Promotion  
2015 / 2016

MA-004-499-1

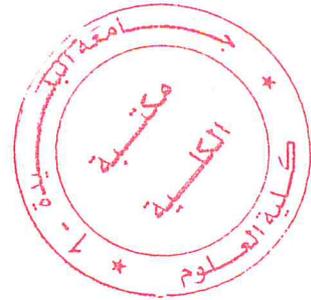
# Remerciements

Avant tout, nous remercions Dieu qui nous a donné la force et la patience pour terminer nos études.

Nous remercions aussi tous nos professeurs pour le savoir qu'ils nous ont transmis.

Nous remercions plus particulièrement Madame ARKAM Meriem, notre promoteur, qui nous a guidé pour la réalisation de ce document et mener à bien le travail qui nous a été confié.

Nous remercions aussi nos familles qui se sont sacrifiées pour que nous puissions réussir nos études (tout en souhaitant un bon rétablissement à mon père), sans oublier nos amis et collègues qui nous ont encouragés.



# Sommaire

Remerciements .....	1
ملخص .....	6
Résumé : .....	7
Abstract.....	8
Introduction générale.....	9
Contexte :.....	10
Présentation du sujet.....	10
Objectif :.....	11
Chapitre 1 : Etude bibliographique.....	13
1.1. Introduction .....	14
1.2. Le e-learning.....	14
1.2.2. Topologie des systèmes de e-learning : .....	14
1.2.3. Quelques exemples de systèmes de e-learning.....	15
1.3. Les MOOCS .....	15
1.3.1 Les types de MOOC .....	16
1.3.2 L'évaluation des apprenants dans les MOOCs .....	18
1.3.3 L'apport des MOOCs par rapport aux systèmes pédagogiques existants (l'école classique).....	19
1.3.4 Quelques MOOC de référence .....	19
1.3.4 Avantages et inconvénients des MOOCs.....	22
1.3.5 Synthèse.....	23
1.4 Le Cloud Computing .....	25
1.4.1 Historique .....	25
1.4.2 Définition du Cloud Computing .....	25
1.4.3 Les différents types de services Cloud .....	26
1.4.4 Les différentes catégories de services sous Cloud.....	27

1.4.5 Les avantages du Cloud.....	28
1.4.4 L'apport du Cloud dans les MOOCS.....	30
a. Le passage à l'échelle dans les applications web .....	30
b. Le passage à l'échelle sous le Cloud : .....	30
c. Élasticité en pratique : .....	30
1.5 Conclusion.....	31
Chapitre 2 : Analyse et Conception de la plateforme MOOC .....	32
2.1. Introduction .....	33
2.2. Présentation de la démarche utilisée (approche objet) .....	33
2.3. Analyse des besoins .....	37
2.3.1. Présentation de l'organisme d'accueil.....	37
2.3.2. Déroulement des formations : .....	38
2.3.3. Problèmes à traiter :.....	39
2.3.4 Objectifs .....	40
2.3.5 Expression des besoins .....	41
2.4 Conception.....	58
2.4.1 Diagrammes de séquence .....	59
2.4.2 Diagramme de classes : .....	70
2.4.3 Diagramme d'activité .....	73
2.4.4 Diagrammes de communication .....	76
2.4.5 Diagramme de déploiement.....	79
4.3 Conclusion.....	80
Chapitre 3 : Réalisation .....	81
3.1 Introduction: .....	82
3.2 Architecture de l'application .....	82
3.2.1 Les différents composants de la Plate-forme.....	82
3.2.2 Implémentation : .....	83

3.3 Développement du côté service avec NodeJS .....	83
3.3.1 Les modules NPM: .....	83
3.4 Application Behavior : .....	84
3.4.1 Json Web Token Authentication: .....	85
3.4.2 Outils de développement : .....	87
3.5 Implémentation de la base de données: .....	87
3.5.1 MongoDB: .....	87
3.6 Implémentation du Front End / du côté clients : .....	88
3.7 La partie hébergement .....	89
3.7.1 Les plates-formes d'hébergement : .....	90
3.7.2 Notre choix : .....	91
3.7.3 Avantages de Heroku .....	91
3.7.4 Le processus d'hébergement de l'application .....	91
3.8 Présentation de l'application .....	95
3.8.1 Page d'accueil : .....	95
3.8.2 Les Pages de profil .....	97
3.8.3 La page de profil du tuteur (Enseignant) : .....	99
3.8.4 La page de profil de l'Admin : .....	100
3.9 Conclusion .....	104
Conclusion générale .....	105
Bibliographie .....	108

## Liste des figures

Figure 1. 1 Dimensionnement horizontal et vertical de l'élasticité sous Cloud (DUPONT, 2016)	31
Figure 2. 1 Diagramme de cas d'utilisation générale	43
Figure 2. 2 Diagramme de cas d'utilisation d'inscription	44
Figure 2. 3 Diagramme de cas d'utilisation de gestion de formations et des cours	47
Figure 2. 4 Diagramme de cas d'utilisation pour gestion des utilisateurs	53
Figure 2. 5 Diagramme de cas d'utilisation pour gestion de profile	57
Figure 3. 1 Diagramme de séquence pour l'action d'authentification	60
Figure 3. 2 Diagramme de séquence pour suivre dans une formation	62
Figure 3. 3 Diagramme de séquence pour soumettre une formation	64
Figure 3. 4 Diagramme de séquence pour uploader un cours	67
Figure 3. 5 Diagramme de séquence pour suivre un cours en ligne	70
Figure 3. 6 Diagramme de classe de la base de données	71
Figure 3. 7 Diagramme de d'activité du système	75
Figure 3. 8 Diagramme de communication de l'apprenant	77
Figure 3. 9 Diagramme de communication pour l'enseignant	78
Figure 3. 10 Diagramme de déploiement du système	79
Figure 4. 1 : l'approche commonJS au sein de notre application.	84
Figure 4. 2 : initialisations du générateur de template jade dans Express.	85
Figure 4. 3 : Envois de index.jade via la requête Get	85
Figure 4. 4 : les différents modules supportés par AngularJS	89
Figure 4. 5 : le tableau de bord de Heroku	92
Figure 4. 6 : Heroku Login via CMD	93
Figure 4. 7 : Créer une nouvelle application Heroku	93
Figure 4. 8 : Dashboard avec la nouvelle application	94
Figure 4. 9 : Renommer l'application	94
Figure 4. 10 : Pusher notre code vers le serveur	94
Figure 4. 11 : La page d'accueil de M-Injaz	96
Figure 4. 12 : Le formulaire d'enregistrement d'un nouveau profil	97
Figure 4. 13 : Enregistrement d'un nouveau profil à l'aide des réseaux sociaux	98
Figure 4. 14 : Page de profil d'un utilisateur	98
Figure 4. 15 : Le calendrier des évènements vus par un apprenant	99
Figure 4. 16 : profil d'un enseignant	99
Figure 4. 17 Ajout Quiz Question avant streaming	100
Figure 4. 18 : Commencement du streaming	100
Figure 4. 19 la section ajout d'un enseignant	101
Figure 4. 20 : Formulaire d'ajout d'un enseignant	101
Figure 4. 21 Remplissage de l'e-mail et génération du mot de passe	102
Figure 4. 22: l'ajout de l'expérience et la position	102

Figure 4. 23 : ajout de l'enseignant réussi \_\_\_\_\_ 103

Figure 4. 24: la liste des cours non validé \_\_\_\_\_ 103

Figure 4. 25 : Approuver un cours \_\_\_\_\_ 104

### Liste des tableaux

Tableau 1 Comparaison entre les xMOOC et cMOOC (Gilliot, 2012) \_\_\_\_\_ 17

Tableau 2 Un tableau comparatif des différents attributs clés de quelques MOOCs les plus connus \_\_\_\_ 21

## ملخص

مع التطور الذي اتخذته التعليم في عصرنا، مع وجود فائض من المنافسة التي لها تأثير سلبي على الطالب التي يصبحوا يطمحوا لغزو الحياة الاجتماعية والمالية أكثر من كسب المعرفة أو الكفاءة، وهنا التعليم المفتوح، يعطي حل موجه أكثر نحو الموهبة. مع الويب الوسيلة الأكثر استعمالاً، خاصة مع الحماس الذي يعرف على مدى العقدين الماضيين، MOOCs هي الثروة الحقيقية لهذا التعليم مفتوح للجميع، مع هذا الحل التعلم الإلكتروني التفاعلي يجب درس بعناية الجانب التقني بأقل تكلفة، وفي هذا السياق لدينا ال Cloud كالحل الذي هو الأكثر ملاءمة، مع تنفيذ hybride من أجل أن يكون منبرا للتكيف مع جميع أنظمة التشغيل.

وهذا الخيار الذي يناسب هذا العمل ماستر التي تهدف إلى تطوير منصة اعلامية التدريب مفتوحة للجميع للجمعية "إنجاز".

## **Résumé :**

Avec la tournure qu'a pris l'éducation dans notre époque, avec un excès de concurrence qui a un impact négatif sur les étudiants qui deviennent de véritables prédateurs à la conquête d'une vie sociale et financière plus qu'un acquis en savoir ou en compétence, c'est là qu'une éducation ouverte, plus laxiste et plus orientée vers le passionnel peut apporter un vrai plus. Avec internet qui est devenu le moyen de communication le plus rapide et le plus moderne qui soit ; surtout avec l'engouement qu'il a connu durant ces deux dernières décennies, nous trouvons les MOOCs qu'on peut spécifier de vrai atout pour cette éducation ouverte à tout le monde, avec cette solution e-learning interactif on se doit de bien étudier l'aspect technique avec un cout correcte, et dans cette perspective nous avons la solution du Cloud qui est la plus adapté, avec une réalisation hybride afin d'avoir une plateforme adaptable à tous les systèmes d'exploitation.

Et c'est dans cette option que s'inscrit ce travail de Master dont l'objectif est de développer une plateforme de formation ouverte à tous dédiée à Injaz.

**Mots clés :** l'enseignement ouvert, e-learning interactif, MOOC, application hybride, Cloud privé.

## **Abstract**

With the twist that has taken education in our time, with an excess of competition that has a negative impact on students who become true predators to conquer social and financial life than an actual know or competence, here an open education, laxer and more oriented passion can bring a real added. With the internet which is a most reliable and most modern means of communication that is; especially with the enthusiasm he has known over the past two decades, MOOCs are a real asset to this education open to everybody, with this interactive e-learning solution we must study a technical side with a correct cost, and in this context we have the Cloud as a solution that is most suitable, with a hybrid implementation in order to have an adaptable platform all operatings system.

And it is this option that fits this work Master whose goal is to develop a training platform open to all for the association “Injaz”.

# **Introduction générale**

## **Contexte :**

De nos jours le monde du travail repose sur des systèmes de recrutement avec une concurrence très rude, et il faut être à la hauteur pour y avoir sa place et de pouvoir y progresser, le problème aujourd'hui c'est que les systèmes d'enseignement ne sont plus suffisants pour former les futures cadres des grandes sociétés, car les domaines scientifiques évoluent de plus en plus rapidement ; produisant des changements technologiques avancés et accélérés qui contraignent l'enseignement à suivre son rythme effréné avec moins d'efficacité.

Cette cadence soutenue pousse l'innovation humaine, et oblige les plus ambitieux à aller chercher plus ailleurs.

Aller dans une université prestigieuse coute beaucoup d'argent, en plus seuls ceux avec un parcours quasi-sans-faute sont acceptés, via un système de sélection pas toujours logique et équitable.

Alors des chercheurs canado-américains ont conçu une solution, qui permet un partage de connaissances et des cours dignes des plus grandes universités du monde, sans sélection et parfois gratuitement.

Cette solution, c'est les MOOCs (« Massive Open Online Course » que l'on peut traduire par « cours en ligne ouvert a un public massif » ou encore l'université en ligne), issu du principalement du e-learning (l'enseignant en ligne), cette méthodologie a connu un grand bond depuis 2012, c'est une révolution proposant une nouvelle façon d'apprendre qui va de procréer des solutions transmissives et pédagogiques.

C'est dans cette optique, et pour ce bon sens que Injaz El-Djazair (qui est une association qui pour but de former des jeunes et les initier au monde de l'entreprise) nous a proposé d'étudier et de concevoir une plateforme de MOOC pour le partage et la diffusion des cours et formations proposées par cette dernière aux tiers et assurées par des professionnels, gratuitement.

## **Présentation du sujet**

Ce projet a pour but de permettre et à faciliter l'accès du grand public aux formations proposées par INJAZ, pour cela le projet afin d'aboutir à cette finalité, le

projet est devisé en trois étapes principales :

- Création d'une Platform MOOC hybride sous Cloud privé : Cette partie se résume à la création d'une Platform plateforme web « le MOOC » (« Massive Open Online Course » que l'on peut traduire par « cours en ligne ouvert et massif ») qui propose un système de partage des formations proposées par INJAZ. Ainsi qu'une application mobile multiplateforme pour permettre à un plus grand public d'y avoir accès plus facilement depuis n'importe quel dispositif matériel (ipad, iphone, Pc portable, notebook...) ou logiciel (OS-Mac, Windows, Ondroid, Unix, ...). Cette solution Elle sera hébergée dans un système Cloud privé avec une architecture adaptée pour les applications web mobile, en intégrant également les bases de données en temps réel.

- Intégration de du contenu streaming (cours en ligne) au sein du MOOC: Cette deuxième partie phase consiste à enrichir notre Platforme, d'un module de diffusion de flux en mode continu (streaming on live), de façon à rendre des formations vidéo accessibles et diffusables directement, et cela pour tous les apprenants inscrits sur la plateforme, et voulant assister directement aux cours, ce module sera doté d'un outil de discussion enligne (un chat) pour poser et répondre aux questions des apprenants à tout moment.

- Proposition d'un système de feedback : cet outil va être doté d'un système de quiz (question/réponse) et qui va aider le formateur à évaluer (تقويم الدرس) le niveau de compréhension des apprenants pour mieux adapter le cours prochain ou améliorer son cours pour les nouvelles versions de la dite formation.

### **Objectif :**

L'objectif du système à développer est le développement d'une plateforme permettant à Injaz d'améliorer leurs services de partage des formations proposées au sein de cette institution, et cela passe par les points suivants :

- Le système doit être capable de diffuser plus facilement l'information, avec notamment des cours en streaming en ligne en temps réel et en différer ;
- Le système doit pallier les contraintes temporaires et de distances géographiques, et permettre une meilleure interaction avec l'intégration d'un système de chat ;

- Utiliser au mieux les tendances actuelles pour atteindre plus de jeunes intéressés par l'apprentissage, via notamment une application mobile ;
- Permettre une meilleure visibilité vis-à-vis la jeunesse désireuse de progresser dans les domaines professionnels ;
- Un système de feedback à la fin de chaque cours, avec un pourcentage de réussite, pour avoir un suivi et appréciation de la compréhension des cours donnés ;
- Permettre une gestion de profil personnalisée pour les utilisateurs comme dans n'importe quel réseau social;
- Permettre un suivi des lacunes et des besoins des apprenants avec un système de récoltes des mots-clés par les administrateurs du cours ;

# **Chapitre 1 : Etude bibliographique**

## 1.1. Introduction

Former Des individus par correspondance n'est pas nouveau, sauf que avant ; faute de moyen on utilisait la poste pour envoyer et recevoir les cours, et ce n'était pas évident, cette méthode a évolué avec les avancées technologiques, ainsi et avec l'arrivée d'internet, ça a facilité beaucoup la diffusion des ressources pédagogiques, depuis le e-learning est apparu, par la suite plusieurs variantes de ce dernier ont été mises au points tel les e-learning interactifs et les MOOCs.

Dans ce chapitre nous allons aborder des généralités sur le e-learning et nous allons nous attarder sur les MOOCs, voir leur particularités et leurs apports par rapport à l'enseignement traditionnel et comment remédier à leurs limites à savoir le manque de dynamisme de groupe dans l'enseignement à distance, et problème de ressources matériel, et cela en intégrant les notions d'interactivité (ici le streaming et le module de chat, ainsi que le système de feedback) et l'outil du Cloud (résoudre le problème des ressources matérielles limitées) et donc profiter des avantages de ces derniers.

## 1.2. Le e-learning

La e-formation ou le e-learning est un concept qui a vu le jour dans les années 1990, ce terme se réfère généralement aux activités d'enseignement ou d'apprentissage nécessitant l'utilisation des ordinateurs et des médias électroniques (AZOUGH, 2014). C'est en somme l'utilisation des nouvelles technologies multimédias et web 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 pour l'apprentissage.

### 1.2.2. Topologie des systèmes de e-learning :

Depuis son instauration, il y a eu maintes évolutions et le e-learning a su s'adapter et se spécialiser en fonction de plusieurs critères (le public réceptif, le type de formation ... etc. (BOUARAB, 2010). pour déboucher sur deux classifications générales :

**a. Les systèmes asynchrones dits non interactifs :** c'est dans les formations où l'apprenant n'est pas en ligne avec l'enseignant, le contact ne se fait pas, ou se fait à un moment planifié ; c'est-à-dire pas en directe (BOUARAB, 2010).

**b. Les systèmes synchrones dits interactifs :** c'est dans les systèmes où on simule une classe virtuelle où la communication entre des différents acteurs se fait en direct et en

temps réel, ceci permet à l'apprenant d'avoir un vrai rôle dans un cours ; à travers ses interventions donc ça permet d'avoir un cours rempli, et un échange entre les apprenants et l'enseignant (le formateur) ce qui apportera beaucoup au cours et aux connaissances qui seront acquises.

Les systèmes interactifs permettent à l'apprenant

### 1.2.3. Quelques exemples de systèmes de e-learning

Il ya eu plusieurs manières d'apprentissages profitant des technologies actuelles. On cite les plus utilisés (CISEL, 2014) :

**a. OER (Open Educational Resources) :** En 1998, David Wiley invente le terme OER (Open Educational Resources), s'inspirant des principes des logiciels libres et open source, en mettant à en ligne des ressources pédagogiques (des photocopiés, des présentations ou des vidéos de cours) gratuite et pour tout le monde, et en partant du même principe de l'open source là aussi on peut utiliser ces ressources à notre guise et sans contraintes. Depuis on en voit beaucoup avec le plus connus Wikipédia.

**b. OCW (Open Course Ware) :** En 1999, Université de Tübingen en Allemagne a publié des vidéos de conférences, on appellera ça The Open Course Ware, c'est plus plaisant plus claire mais le hic c'est que c'est moins interactive et moins adapté pour internet, le plus célèbre OCW est le MIT OpenCourseWare de la Carnegie Mellon University lancer en 2001.

**c. LMS (Learning management system) :** Vers la fin des années 90 et les début 2000, il y a eu l'apparition des LMS, c'est des outils logiciels web modulable qui permette entre autre de suivre l'évolution des étudiants, de faire des tests ...etc. donc une alternative beaucoup plus interactive que les précédentes, la référence sur les LMS c'est la Platform open source Moodle créé en 2002, par Martin Dougiamas. L'inconvénient c'est que en générale c'est réservé à un publique limité.

**d. MOOC (Massive Open Online Course) :** C'est en 2008 qu'apparait le premier MOOC, qui s'alloue l'ouverture des OER et l'interactivité des LMS. Ce dernier type nous parait plus intéressant que nous allons détailler dans la section ci-après

## 1.3. Les MOOCS

Le terme MOOC est né en 2008 par Dave Cormier, c'est l'acronyme de « Massive

Open Online Course », il a été au début défini comme étant un moyen d'apprentissage interactif et souple, qui apporte une nouvelle méthode d'enseignement, qui se résume à former des apprenants intéressés et passionnés, par des enseignants compétents et expérimentés dans un domaine précis, le tout se fait en communiquant uniquement via le billet d'internet (Delpech, et al., 2016).

La traduction linguistique de l'acronyme MOOC donnera « Cours en ligne ouverts et massifs », ce qui ne signifie pas grand-chose comme ensemble, mais pourtant chaque mot explique une caractéristique de ce système :

- **Cours (Course en anglais) :** c'est des vrais cours qui sont proposés avec de réels objectifs pédagogiques comme tout cours d'université.
- **En ligne (Online en anglais) :** ces cours sont partagés en ligne ce qui facilite l'accès, et limite les contraintes et les difficultés dues aux distances géographiques.
- **Ouvert (Open en anglais) :** ce qui traduit par ouvert à tout le monde sans qu'il y ait une sélection, il faut juste être intéressé et s'inscrire.
- **Massif (Massive en anglais) :** c'est la représentation du nombre de participants potentiels, qui est très important par rapport à un cours d'université classique.

Alors pour résumer ce concept, des gens des quatre coins du monde intéressés par un cours doivent s'inscrire (en générale sur une période déterminée), et ils suivent après de suivre les cours et les tests soumis par un enseignant qui est peut-être à des kilomètres d'eux.

### 1.3.1 Les types de MOOC

On distingue aujourd'hui deux types de MOOC par rapport à leurs façons d'administrer l'information (CISEL, 2014) :

**a. cMOOC :** Le 'c' fait référence à une théorie de connectivité basée sur une approche pédagogique ouverte communautaire, c'est-à-dire c'est une communauté où chaque membre est potentiellement un enseignant, et peut apporter un plus pour les autres, la description de ce type a été donnée par George Siemens en 2008, ITyPA utilise ce principe de connectivité.

**b. xMOOC :** Ce type est décrit par Thierry Karsenti en 2012 ce type est basé sur le même principe que les cours classiques universitaires, ces formations sont structurées par un enseignant qui distribue ses connaissances à des élèves avec à la clé un certificat de validation des compétences via des tests et des épreuves de travaux dirigés, mais les évaluations constituent un vrai casse-tête (nous verrons les différentes techniques d'évaluation dans le prochain sous-titre)

dans le cas où un vrai certificat décerné utilisable dans le domaine professionnel. Les très réputé plates-formes Coursera ou edX font partie de cette catégorie.

**c. Comparaison entre les deux approches :**

La différence entre eux est un peu mieux expliquer sur ce tableau extrait de « *Whatis the theory that underpinsour moocs?* », une synthèse conclue par Stephen Downes et traduit par Jean-Marie Gilliot (Gilliot, 2012).

**Tableau 1 Comparaison entre les xMOOC et cMOOC (Gilliot, 2012)**

	xMOOC : issu des cours traditionnels	cMOOC : issu de l'approche connectiviste
Modèle pédagogique	Classique : Cours - exercices – contrôle des connaissances acquises	Connectivisme
Connaissance	Préparée avant le cours – déclarée	Distribuée – générée
Cohérence	Donnée par l'enseignant	Construite par les participants
Objectifs d'apprentissage	Défini par l'enseignant et le programme	Défini par chaque participant pour lui-même
Apprendre	Suivre le cours	Navigation, établir des connexions
Ressources	Définies dans le cours	Agrégées par les participants, abondance
Importance de l'échange entre pairs	Faible à moyenne	Fondamentale
Interactions	Forum sur le site du cours	Distribué, partant d'un portail, chaque participant peut utiliser Twitter, son blog...
Synchronisation par	Déroulé du cours et instructeur	Interactions entre participants, thème de la semaine, point de rencontres (vidéoconférence en direct)
Résonance	Aucune	Entre pairs
Encourager autonomie et auto régulation	Aucune	Maîtrise d'e-compétences
Domaine	Disciplinaire, lié à un cours	Thématique centrale, mais ouvert et

d'apprentissage	universitaire	interdisciplinaire
Évaluation du succès	Délivrance d'un certificat de réussite	Auto-évaluation de l'apprentissage

Aujourd'hui les MOOCs les plus célèbres sont des xMOOC car ils sont mieux gérés par des professionnels par rapport aux cMOOC, d'ailleurs même pour notre projet un xMOOC nous paraît plus adapté pour les raisons suivantes :

- Les cours partagés sont certifiés et publiés avec plus de rigueur.
- Les cours doivent être aussi bien structurés suivant un ordonnancement bien précis afin de mieux apporter les connaissances.
- Ce type est plus adapté pour les conférences (qui sont le sujet principal de ce projet).

### 1.3.2 L'évaluation des apprenants dans les MOOCs

Cette aspect concerne les surtout les xMOOCs car pour délivrer un certificat il faut tester les apprenants si les compétences qui doivent être acquises le sont vraiment, pour ce fait plusieurs évaluations ont été proposées, on en cite quelques-unes (CISEL, 2014):

#### a. Evaluation classique (abandonné et quasiment inutilisable) :

En premier temps, une évaluation très classique a été proposée tout comme on fait dans les écoles, à savoir les étudiants qui envoient leurs travaux, qui seront corrigés par l'enseignant mais cette idée a été vite abandonnée, car on pouvait recevoir des milliers de réponses et c'était impossible de les corriger toutes dans un délai donné.

#### b. Evaluation automatique (comme avec les LMS):

Puisque l'humain en était incapable, il s'en est remis alors à l'ordinateur, et il est passé à une autre solution, dans laquelle c'est la machine qui corrige automatiquement les devoirs via des modèles de réponses déjà définis, ce qui ne pose pas trop de problèmes dans le côté massif avec une bonne gestion, ce système est parfait quand il s'agit de corriger des QCM, des Quiz ou des textes à trous, mais dès qu'on veut évaluer des tests plus aboutis, avec parfois la nécessité d'une perception humaine, là on voit les limites de cette évaluation.

#### c. Evaluation par les pairs :

Ce dernier type, consiste à ce que les apprenants qui s'évaluent entre eux, ce type d'évaluation apporte clairement un plus, pour cause il permet une plus grande hétérogénéité des tests, avec

un rôle plus actif pour les apprenties et amener un facteur plus humain dans l'appréciation des épreuves, même si ça reste moins pertinent qu'une correction d'un professeur certifié par un établissement, mais néanmoins plus adapté pour le côté massif, la plupart des MOOCs adhèrent à cette approche.

### **1.3.3 L'apport des MOOCs par rapport aux systèmes pédagogiques existants (l'école classique)**

Les universités ont existé depuis l'époque romaine au cinquième siècle environs, c'est un endroit où on transmet des savoirs entre des individus intéressés et passionnés, mais si on veut faire une description concise de l'université d'aujourd'hui, c'est un lieu où on regroupe des individus qui sont approximativement de la même génération ; passionné ou plus justement avec plus au moins l'envie d'évoluer dans un même domaine (informatique, histoire, mécanique ...etc.), on les mets en concurrence, pour les pousser à faire de plus gros efforts, sous peine d'être mis en échec et d'être considéré comme un revers de la société, cette méthode a un vrai avantage c'est de permettre aux étudiants de ce dépasser afin les rendre plus productifs, de les disposer à recevoir toutes les pressions qui existe dans le monde de l'entreprise. Mais cet excès de concurrence qui prend aujourd'hui le dessus sur le côté passionnel de l'université est un vrai inconvénient, car quand on ne se passionne pas pour quelque chose on ne s'épanouit pas totalement dedans, et on devient un robot programmé à faire des choses sans un vrai plaisir humain.

C'est là que Les MOOCs peuvent apporter « un coup de main » à l'université, car cette ouverture et cette liberté que dispose les MOOCs maintenant peuvent aider les étudiants à trouver beaucoup plus facilement leurs vocations sans qu'ils se surpasse, car il y a moins d'enjeux et moins de pressions et surtout ils sont totalement libres de quitter un cours qui ne leurs plait pas, cette autonomie pédagogique est une révolution qui permet à tout le monde de suivre et de se former, avec une restriction assez convenable et adaptée pour l'enseignement en ligne, et permettre une projection professionnelle beaucoup plus facile.

### **1.3.4 Quelques MOOC de référence**

Depuis le succès fulgurant des MOOCs, il y a eu beaucoup de projet qui ont vu le jour, citant quelques-uns des plus connue :

**a. Coursera :** Coursera est une Platform créé en avril 2012, par deux informaticien Andrew Ng et DaphneKoller de l'université de Stanford, qui compte plus de 16 millions d'utilisateurs et plus de 1500 cours (en janvier 2016), ce succès est due notamment grâce à ses variations de

langage car elle est disponible en pas moins de 12 langues (français arabe compris, pas kabyle), et un catalogue de cours très riche, car on trouve de tous des cours informatique à des cours de bien être psychologique en passant par de très nombreux modules scientifique et culturel, c'est ce qui permet à des apprenant du monde entier de pouvoir suivre ses cours, actuellement la plus part des cours sont gratuit, mais pas tous, mais pour avoir la certification il faut payé quel que soit le cours, il a également des revenue venue des entreprises qui désir recruter directement depuis la Platform (avec le consentement des étudiants en question bien sûr), alors en septembre 2013 soit un peu plus d'un an après sa création coursera a annoncer 1 million de dollars revenue, qui s'élèvera à 85 million en décembre 2013.

**b. edX :** edX a été créée en 2012 aussi mais en mai donc un mois après Coursera, par Massachusetts Institute of Technology en collaboration avec l'université Harvard, elle concerne 5 million d'utilisateurs en (septembre 2015) et est disponible en 5 langue dont le français mais pas arabe au contraire de coursera, ce xMOOC propose des cours universitaire totalement gratuit et ouverte à tous, on paye seulement pour recevoir le certificat des compétences acquis. De plus son développement est libre d'accès, grâce à sa Platform open source Open edX, qui est mise à disposition pour ceux qui veulent développer leurs propre MOOC.

**c. UDACITY :** Udacity est le fruit des cours gratuits d'intelligence artificiel en 2011 présenté par SebastianThrun un enseignant informaticien à l'Université de Stanford, c'était un sorte d'un OpenCourseWare mais mieux travaillé. 160.000 étudiants dans le premier cours qui était une « Introduction à l'intelligence artificielle », et 90.000 étudiants étaient inscrits dans les deux premières parties qui suivirent, à partir de Mars 2012. Udacity a été annoncé lors de la conférence de conception digitale Life 2012. elle est financée par la société, Charles River Ventures, et 200 000 \$ de l'argent personnel de Thrun. Elle englobe aujourd'hui 1.6 millions d'étudiant, le plus de Udacity est un mooc orienté beaucoup vers le domaine informatique que autre chose à la différence de coursera par exemple, mais ces formation qu'elle propose sont payante toutes mais décrite comme étant plus efficace pour le domaine professionnel.

**d. ITyPA :** Parmi les MOOC français; ItyPA tient une place de choix. D'une part si l'on raisonne d'un point de vue quantitatif (nombre d'apprenants) c'est un succès et d'autre part, il est l'un des premiers MOOC français dont l'organisation pédagogique est largement inspirée de celle adoptée par Siemens et Downes dans leur cours CCKO8 (Siemens &Downes, 2008) sur une connectivité communautariste, c'est l'un des rare cMOOC (comme ça été évoquer auparavant). Les interactions y sont considérées comme étant le terreau des apprentissages,

tout le monde peut faire un cours et le partager ; en respectant quelque règle imposée par le site, le respect des droits d'auteurs par exemple.

**e. OpenClassrooms :** Nommé avant « le Site du zéro », il fut Cree en 1999 par Mathieu Nebra, alors collégien de 13 ans et passionner d'informaticien français, le site du zéro était d'abord un OCW spécialisé dans l'informatique, sur qui des bénévoles partagent gratuitement des cours et des tutoriels de manière très souple et subjective, après avoir fini ses études d'ingénierats ; Nebra et ces collaborateurs dans le site, ont fondé une petite entreprise nommée OpenClassrooms (avant elle s'appelait IT) pour gérer le site de manière un peu plus professionnelle, c'est en 2013 que la version MOOC est arrivée avec des cours gratuits, mais on peut être premium et avoir quelques avantages (comme suivre les cours à son rythme quand on veut et être certifié à la fin), ce MOOC est un peu un mélange des deux types xMOOC et cMOOC, même si il est beaucoup plus proche de l'aspect des xMOOC.

#### **f. Comparaison entre ces MOOCs**

Pour mieux comparer les plates-formes les plus efficaces ; on a pris quelques critères pour mieux distinguer leurs attributs par rapport à leurs performances :

**Tableau 2 Un tableau comparatif des différents attributs clés de quelques MOOCs les plus connus**

Les MOOCs	Coursera	edX	Udacity	ITyPA	OpenClassroom
Type	xMooc	xMooc	xMooc	cMooc	xMooc
Certification	Oui	Oui	Oui	Non	Oui
A but lucratif	Oui	Non	Oui	Non	Oui
Application mobile	Oui	Oui	Oui (mais pas adapté pour tous les cours)	Non	Non
Gratuit	Oui (mais pas pour tous les cours)	Oui (mais pas pour tous les cours)	Non	Oui	Oui (mais il y a des avantages pour les premiums)
Nombre max d'apprenant par cours	Illimité	300 000	Illimité	Pas précisé	Illimité
Hébergement	Serveur dédié virtuel	Serveur locale	Serveur distant	Serveur distant	Serveur dédié virtuel
Accès	Doit être inscrit	Doit être inscrit	Doit être inscrit	Accès ouvert	Accès ouvert (il y a des comptes)

					payants)
Diffusion en direct	Non	Non	Non	Non	Non
Type de support	Vidéos & textes	Vidéos & textes	Vidéos & textes	Vidéos	Textes (vidéos pour les premiums)
Système de feedback (Quiz après les formations)	Oui (avec juste le résultat de l'évaluation)	Oui (avec juste le résultat de l'évaluation)	Oui (avec juste le résultat de l'évaluation)	Non	Oui (avec la correction après le test)
Outils de collaboration	Forum, et commentaire	Forum et messagerie chat	Forum et messagerie chat	Rien	Forum

### 1.3.4 Avantages et inconvénients des MOOCs

#### a. Avantages des MOOCs

Alors cette nouvelle méthode d'enseignement a connu un engouement populaire important, et pour cause nous devons bien étudier les enjeux, et voir ce qu'on pourrait gagner, on cite quelques avantages des MOOCs :

**L'enseignement ouvert à tous :** les MOOC accepte toute personne intéressée, cette ouverture à tous constitue un gros avantage quand on voit les sélections (parfois très dur et pas très juste) que font les grandes universités, et qui incites les étudiants a ce démener pour y avoir leurs places en y laissant des plumes, aussi bien physiques que psychiques.

**Un enseignement pas très coutant pour les apprenants :** la majorité des MOOC proposent des cours gratuits ou pas très cher, c'est une vrais chance pour les apprenants quand on voie combien coutent les cours universitaires (hors Algérie),

**Un enseignement pas très coutant pour les universités :** un MOOC coute beaucoup moins cher qu'une université classique, il y a moins de charges à payer pour partager des cours.

**Un enseignement avec une grande mobilité :** les cours sont accessibles de partout, il suffit d'être connecté à internet, quel que soit la distance géographique.

**Un enseignement avec une disponibilité à tous moment :** les étudiants avec un emploi du temps charger pourront suivre les coures a leurs rythme et au moment où ils veulent.

**Un enseignement renommé :** évidemment on ne parle pas de tous les MOOCs existant aujourd'hui mais grâce à certains d'entre eux on peut suivre des cours et même être certifié des universités les plus prestigieuses comme Oxford, Harvard.

#### **b. Inconvénients des MOOCs**

Les offres de cours dans un MOOC sont généralement gratuites. Pour les étudiants qui n'ont pas la possibilité de rencontrer leurs collègues, les MOOC ont tendance à éliminer les relations et interactions sociales entre étudiants, le rôle de socialisation de l'école est donc remis en cause. Par ailleurs, des difficultés techniques avec l'ordinateur ou internet peuvent engendrer des problèmes dans l'apprentissage.

Les étudiants peuvent préférer une structure d'apprentissage plus traditionnelle de type cours en présentiel, ainsi il n'existe pas de liens directs entre enseignants et étudiants, les MOOC ont pour objectif de développer l'anonymat entre les pairs. Les étudiants doivent devenir responsables de leur propre apprentissage, car ils ne sont pas encadrés ou suivis par l'enseignant. Ils sont livrés à eux-mêmes (FARES, 2013).

### **1.3.5 Synthèse**

Nous avons pu voir avec un peu plus de clairvoyance, une des nouvelles révolutions techno-pédagogique de cette décennie, en abordant ces différents aspects et nous pouvons constater quel sont ces apports dans les systèmes universitaires d'aujourd'hui, pour conclure le MOOC est une vraie chance pour les étudiants, les apprenants et même les professionnels désirant ajouter un bagage didactique, mais sans oublier l'aspect plus passionnel que peut apporter l'école classique.

En résumé, les MOOCs ont comme premier attribut le côté massif, donc un bon MOOC doit pouvoir supporter les interactions en masse, et un passage à l'échelle doit être envisagé, et en même temps se payer un serveur performant peut s'avérer être très cher ce qui représente une limite très flagrante aux MOOCs, afin d'y remédier, nous allons essayer d'étudier dans la section suivante le Cloud Computing (ou serveur virtuel) comme étant une des solutions qui résout au mieux ce problème et déjà adaptée par quelques MOOCs de référence (cf. tableau 2), aussi dans cette section nous allons aborder l'apport du Cloud dans les MOOCs.

Pour le problème d'interactivité et d'interaction entre l'enseignant et les apprenants, et

de manque de vie sociale, nous pensons munir notre future plateforme d'un système de streaming on live pour le visionnement des cours en direct avec un outil de communication à savoir le chat, et au final un petit Quiz pour l'évaluation du cours apporté.

## 1.4 Le Cloud Computing

La technologie du Cloud Computing a évolué au cours de la dernière décennie, passant d'un simple service de stockage à des services plus complexes (GUENANE, 2014), en permettant au monde de l'informatique de passer un cap, avec un grand nombre d'avantages.

### 1.4.1 Historique

Le concept du Cloud existait depuis plusieurs dizaines d'années, l'idée primaire fut proposée dans les années soixante, par un certain John McCarty qui mit en place un système pour partager des données entre des ordinateurs et un ordinateur central (mainframes) qui faisait les calculs compliqués et les envoyait par réseau, mais néanmoins ce n'était pas public mais réservé aux militaires et quelques laboratoires et universités de renommée.

Mais l'appellation de « Cloud computing » a vu le jour après la révolution d'internet et c'est dans la fin des années quatre-vingt-dix qu'Amazon fut à priori le premier à proposer des services web aux entreprises à distance avec Amazon Web Services, ce système est considéré comme étant l'ancêtre du Cloud, et c'est jusqu'aux années 2008 qu'il a eu un réel engouement de cette tendance, et des entreprises comme Google, IBM, Microsoft et d'autres emboîtèrent le pas.

Aujourd'hui le Cloud Computing a pris une place considérable dans le monde de l'entreprise et a su se faire adopter par un grand nombre d'institutions (BORT, 2015), et les établissements qui proposent ce service (moyennement une rémunération financière) ne cessent de le développer pour plus de performance.

### 1.4.2 Définition du Cloud Computing

Le « Cloud Computing », appelé aussi l'informatique dématérialisée, a plusieurs définitions données par les organismes et les instituts concernés par les technologies modernes (plus ou moins proches étant donné sa « polyvalence »), on cite quelques-unes :

Le National Institute of Standards and Technology, ou NIST (qu'on pourrait traduire par « Institut national des normes et de la technologie »), qui est un institut américain qui fait la promotion des nouvelles technologies (MELL, et al., 2011) : « Le Cloud Computing est un modèle pour permettre un accès pratique à la demande réseau à un pool partagé de ressources

informatiques configurables (par exemple, réseaux, serveurs, stockage, applications et services) qui peuvent être provisionnés rapidement et libérés avec un effort de gestion minimale ou prestataire de services interaction. »

Pour Microsoft (YUEN, 2010): « À haut niveau, le Cloud Computing se définit par la capacité de tirer partie de certaines ressources pour fournir des applications et des services. Et cela passe par une gestion d'applications et de services à un niveau de ressource spécifique, tels que des ordinateurs ou des serveurs ou même la mémoire et l'espace disque, mais aussi la gestion de la capacité d'application et d'être en mesure d'étendre ou de réduire ce niveau à la demande, en utilisant cette ressource partagée. »

Pour Wikipédia, il s'agit (2016) : « D'un concept de calcul basé sur Internet qui fournit des ressources partagées de traitement informatique et de données à des ordinateurs et d'autres appareils sur demande».

Donc en résumant le Cloud est des fournisseurs qui propose des services divers et varié a des clients, ces derniers peuvent y accédé via à travers des data-centre entretenus et maintenus par le fournisseur en question.

Ce qui est à mettre en avant est que les entreprises « clientes » n'auront pas à s'occuper de la maintenance et des travers technique, ce qui représente un casse-tête en moins et un allégement économique considérable.

En plus, ils n'auront pas à surcharger leurs serveurs avec notamment avec des applications a utilisations récurrentes, en les ayant sous Cloud ça permet de gagner en temps de calcule car ces data centre sont plus optimisé et dédié à cette effet.

### 1.4.3 Les différents types de services Cloud

Il existe aujourd'hui trois types de Cloud (YUEN, 2010) :

**a. Le Cloud public :** qui définit des services que tout le monde peut utiliser ; par exemple office360 de Microsoft, mais qui dit public dit confidentialité moindre car les éléments échangé sont géré par des organismes externes ; alors quand on des données clés a traité on peut comme même se poser quelque questions sur le chemin emprunter par ces données et être sûr qu'ils n'ont pas été intercepté ou altéré.

**b. Le Cloud privé :** ce deuxième type de services est à l'inverse du Cloud public ces services sont installé sur les réseaux informatique d'une entreprise et donc qui sont accessible que par

les employés de celle-ci, avec cette solution certes on est sûr où se trouve et où vont les données ; mais c'est un peu plus coûteux et ça relève un peu plus de gestion que les services public.

**c. Le Cloud hybride :** ça alterne les deux précédents types de services, c'est-à-dire qu'au sein d'un même organisme on utilise le Cloud privé pour les opérations « sensibles » qui s'avèrent être plus sécurisées, et utilise en même temps les services public pour les opérations moins risquées ce qui va offrir un coût moindre et une évolutivité plus avantageuse.

#### 1.4.4 Les différentes catégories de services sous Cloud

Ils existent plusieurs catégories de services, mais il y a quatre des plus utilisés sur le marché :

**a. IaaS (Infrastructure as a Service) :** qui se traduit par infrastructure comme un service, ça se résume à mettre à disposition du client du matériel informatique virtualisé ; c'est-à-dire une infrastructure informatique, ce genre de service comprend des offres telles que l'espace serveur, des connexions réseau, de la bande passante, des adresses IP et des loadbalancers, l'IaaS peut être avantageux dans la mesure où la maintenance hardware n'est plus le problème du client mais du fournisseur et cela va avoir des répercussions bénéfiques sur le plan économique.

**b. SaaS (Software as a Service) :** c'est les services Cloud permettant d'avoir un accès à des applications prêtes à l'emploi depuis internet ou autres réseaux sans se soucier d'aucune contrainte mis à part avoir un navigateur web, les applications les plus connues aujourd'hui sont des vrais exemples de SaaS par exemple Facebook ou encore Google, ce genre de service est destiné pour les utilisateurs non spécialistes en informatique ou les entreprises ayant pas d'informaticien.

**c. PaaS (Platform as a Service) :** cette catégorie de service fournit un environnement informatique nécessaire pour faire fonctionner des applications ; ou des services accessibles aux utilisateurs par navigateur web, ce genre de service est destiné principalement aux développeurs. Pour notre cas nous avons choisi d'utiliser ce genre de service pour la Platform web, car elle est la solution la plus adaptée et la plus simple pour une solution qui aura besoin de quelques outils d'environnements.

**d. BaaS (Backend as a Service) :** c'est peut-être la catégorie la moins connue des trois précédentes ; elle est adaptée pour les développeurs web et mobile pour cause elle permet de lier

leurs application a des api ou à une espace de stockage afin de pouvoir utilisé des fonctionnalités ou des codes en arrière-plan. Ce type de service nous en aurons besoins pour la partie mobile pour une optimisation des performances.

### **1.4.5 Les avantages du Cloud**

Cette nouvelle tendance de l'informatique des nuages a connus un grand boom ces dernières années, et cela du à ses nombreux avantages:

- **Une flexibilité à tout-va :**

Les services basé sur Cloud sont idéal pour les applications qui voit leurs demandes de bande passante changeante d'un moment a un autre, donc si il y a amplification, c'est facile d'intensifier la capacité du système en accentuant sur les serveurs distant du service utilisé, la même-chose pour un rabaissement, la flexibilité est fourni par le service. Cette souplesse peut donner un réel avantage.

- **Une restauration en cas de catastrophe :**

Tout organisme doit investir pour une restauration des données en cas d'incident, mais quand on n'a pas les moyen, ceci est un luxe qu'on ne peut s'offrir, aujourd'hui le Cloud est une réel solution à ce problème et inverse un peu la tendance, avec le fameux système de sauvegarde de données, en ayant toujours des données stocké quelque part, on peut les récupérer a tous moment, le tout sans que cela vous coûte les yeux de la tête.

- **Mise à jour automatique des logiciels :**

C'est les fournisseurs qui doivent prendre soin de faire toutes les mises à jour des logiciels qu'on utilise (y compris les mises à jour de sécurité) et ne pas à avoir à se soucier de ça, peut nous faire gagner beaucoup de temps, afin de se concentrer sur l'essentiel.

- **Capitale à dépenser :**

L'informatique sur les nuages, a un impact direct sur le coût d'un système, car en évitant la dépense considérable du matériel qui est souvent considérable, en revanche vous payer pour service avec une politique de « vous payez que ce que vous consommer » pour éviter les dépenses inutile.

- **Une collaboration accrue**

Ça peut être pratique en collaboration, souvent dans les projets actuels les équipes de

travail ont tendance à travailler à distance, alors le partage de flux de travail et de fichiers devient beaucoup plus facile, et permet de voir en temps réel les mises à jour apportées par les autres collaborateurs.

- **Travailler de n'importe où :**

Avec le Cloud, si vous avez une connexion internet, vous pouvez être potentiellement au travail. Et maintenant la plus part des services proposent des applications mobiles, pour pousser un peu plus loin l'avantage de la mobilité. Les entreprises peuvent offrir des avantages de travail plus flexibles aux employés afin qu'ils puissent profiter de la conciliation travail-vie qui leur convient sans que leurs productivités prennent un coup.

- **Contrôle des documents :**

Avec un accès asynchrone des fichiers partagé, donc on peut avoir plusieurs clients qui ont accès à un même fichier, car généralement les calculs sont faits très rapidement et les mises à jour apportées sont visibles quasi-instantanément par les autres utilisateurs, ce qui est un réel avantage.

- **Sécurisation des données**

La perte d'un ordinateur ou d'un disque dur est souvent une perte qui ne peut pas être réparée pour une entreprise, car on peut facilement laisser filer des données clés à l'intérieur, le Cloud dit aussi l'informatique dans les nuages vous donne une plus grande sécurité lorsque ce genre de chose se produit, avec un peu d'imagination vos données sont dans un nuage, en réalité ils sont dans des data centers ultra sécurisés, et vous pouvez y accéder quoi qu'il arrive via votre machine et dans le cas d'une perte d'un ordinateur vous pouvez effacer les données à distance pour éviter qu'elles tombent entre de mauvaises mains.

- **Compétitivité :**

Le passage vers le nuage donne accès à un nouveau palier celui de la classe des entreprises, le tout sans dépenser gros, donc cela permet aux petites de concurrencer les grosses avec la politique Pay-as-you-go, donc le Cloud met tout le monde au même pied d'égalité pour permettre une meilleure concurrence.

- **Une solution écologique :**

Ce point n'a pas grande chose à voir avec l'informatique, mais il est assez important de le citer, avec une dépense d'énergie moindre avec les data centers qui font le sale boulot pour

vous, donc c'est une solution qui respecte un peu mieux l'environnement.

#### **1.4.4 L'apport du Cloud dans les MOOCS**

Parmi les plateformes d'apprentissage les plus connues nous pouvons citer deux (coursera et openClassrom) au moins, et qui utilisent le Cloud pour l'hébergement, pour ce genre de application on a tendance à négligé un peu le coté sécurité pour mieux optimiser les performances, car on a souvent un passage de quelque utilisateurs a plusieurs centaine de milliers on appel ça un passage à l'échelle.

##### **a. Le passage à l'échelle dans les applications web**

Le concept de passage à l'échelle désigne la capacité d'un système à continuer à délivrer avec un temps de réponse constant un service même si le nombre de clients ou de données augmente de manière importante (SARR, 2010). Le passage à l'échelle peut être mesuré avec au moins trois dimensions :

- le nombre d'utilisateurs et/ou de processus (passage à l'échelle en taille) ;
- la distance maximale physique qui sépare les nœuds ou ressources du système (passage à l'échelle géographique) ;
- le nombre de domaines administratifs (passage à l'échelle administrative).

Pour le cas des MOOCs c'est la première dimension qui est importante, car dans un cours interactifs il ya une hausse d'apprenant en même temps, avec des interactions en temps réels. Et donc le système doit pouvoir y reprendre complication.

##### **b. Le passage à l'échelle sous le Cloud :**

Dans le contexte du Cloud computing, la scalabilité correspond à l'aptitude d'une application à répondre à la montée en charge en ajoutant/retirant des ressources afin de maintenir un certain niveau de performance et de QoS (Qualité du Service). Concrètement, cela signifie que la puissance de calcul, la mémoire ou le stockage utilisé par le système peuvent facilement être augmentés ou diminués en fonction des besoins. (DUPONT, 2016). On appelle cette gestion de ressources l'élasticité de l'application. Et est présentée comme l'une des caractéristiques essentielles du Cloud.

##### **c. Élasticité en pratique :**

Pour gérer cette élasticité, on distingue actuellement deux types de dimensionnement : le dimensionnement horizontal (horizontal scaling) et le dimensionnement vertical (vertical

scaling) (DUPONT, 2016). Le dimensionnement horizontal, ou élasticité horizontale, consiste à ajuster le nombre des machines virtuels en fonction de la demande, le dimensionnement vertical, aussi appelé élasticité verticale consiste à augmenter/diminuer les ressources d'une VM existante telles que le CPU, la mémoire ou encore la capacité de stockage (Erreur ! Source du renvoi introuvable..1).

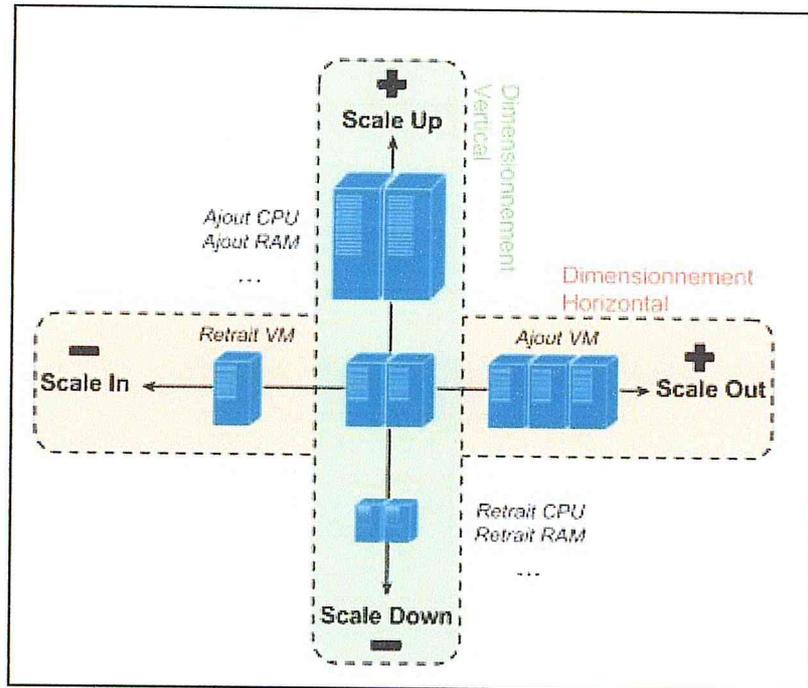


Figure 1. 1Dimensionnement horizontal et vertical de l'elasticité sous Cloud (DUPONT, 2016)

## 1.5 Conclusion

Nous avons vu dans ce chapitre la notion du e-learning en général, et les MOOCs en particulier et les spécificités et avantages de ces derniers ainsi que les limites des MOOCs, en deuxième partie nous avons abordé des généralités sur le Cloud, et enfin nous avons vu l'apport que peut apporter le Cloud à un MOOC interactif. Dans le prochain chapitre nous allons aborder l'analyse et la conception de notre future plateforme MOOC selon les besoins recensés au sein de Injaz El Djazair.

**Chapitre 2 : Analyse et  
Conception de la plateforme  
MOOC**

## 2.1. Introduction

Après avoir acquis les connaissances théoriques nécessaires à la conception et à la réalisation d'un système de e-learning en général et d'un MOOC en particulier, nous avons par la suite essayé de trouver une communauté répondant aux critères requis et manifestant une volonté à ce type de projet, et suite déjà à notre collaboration extra scolaire avec l'association Indjaz Eldjazair, nous avons constaté qu'elle sera un très bon cas d'application pour notre projet du fait que les formations et cours présentés sont sous formes de conférences données dans une durée bien déterminée avec un nombre de places pédagogiques et de ressources matérielles très limitées sous contrainte aussi de la disponibilité du conférencier et des apprenants, le MOOC dans ce cas sera une solution très adaptée pour répondre aux demandes des apprenants sans avoir les contraintes citées précédemment.

Nous allons commencer cette partie par cerner la problématique et les objectifs réels à atteindre, par la suite, nous passons à exprimer les besoins fonctionnels et non fonctionnels des différents acteurs du système, et nous terminerons ce chapitre par une étude conceptuelle détaillée du futur MOOC à mettre en place.

Pour cela, nous avons donc opté pour une démarche orientée OBJET « UML » (Unified Modeling Language) et un processus de développement piloté par les cas d'utilisations « UP » (Unified Process) modélisant au mieux les besoins et les préoccupations de la communauté en question.

## 2.2. Présentation de la démarche utilisée (approche objet)

La construction fondamentale en modélisation orientée objet est l'objet, qui combine à la fois des structures de données et un comportement dans une seule entité. Les données ne sont pas favorisées par rapport aux traitements et vice vers ça, la cohérence entre les deux

est ainsi gérée naturellement.

Dans une approche objet :

- Le processus est de type itératif ;
  - Les découpages ne coïncident pas : les activités (tâches, phases, étapes, etc.) se déroulent dans plusieurs dimensions.
- La maîtrise des processus de développement implique pourtant une organisation et un suivi des activités : C'est ce à quoi s'attachent les différentes méthodes qui s'appuient sur l'utilisation du langage UML pour modéliser un système d'information.

**a. UP (Unified Process)** est une méthode générique de développement de logiciel.

*Générique* signifie qu'il est nécessaire d'adapter UP au contexte du projet, de l'équipe, du domaine et/ou de l'organisation (exemple R. UP ou X.UP). C'est, entre parenthèses, plus ou moins vrai pour toute méthode, qu'elle se définisse elle-même comme générique ou pas.

Il existe donc un certain nombre de méthodes issues de UP.

Le processus unifié est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel.

**b. Caractéristiques essentielles du processus unifié**

- Le processus unifié est à base de composants ;
- Le processus unifié utilise le langage UML (Ensemble d'outils et de diagrammes) ;
- Le processus unifié est piloté par les cas d'utilisations ;
- Le processus unifié est centré sur l'architecture ;
- Le processus unifié est itératif et incrémental.

**c. Le cycle de vie du processus unifié**

Le processus unifié répète un certain nombre de fois une série de cycles.

Tout cycle se conclue par la livraison d'une version du produit au client et s'articule en quatre phases : création, élaboration, construction et transition, chacune d'entre elles se subdivisant à son tour en itérations.

Chaque cycle se traduit par une nouvelle version du système. Ce produit se compose d'un corps de code source réparti sur plusieurs composants pouvant être compilés et exécutés, il s'accompagne de manuels et de produits associés. Pour mener efficacement le cycle, les développeurs ont besoin de construire toutes les représentations du produit logiciel :

Modèle des cas d'utilisations	Expose les cas d'utilisations et leurs relations avec les utilisateurs.
Modèle d'analyse	Détaille les cas d'utilisations et procède à une première répartition du comportement du système entre divers objets.
Modèle de conception	Définit la structure statique du système sous forme de sous systèmes, classes et interfaces. Définit les cas d'utilisations réalisés sous forme de collaboration entre les sous systèmes, les classes et les interfaces.
Modèle de déploiement	Définit les nœuds physiques des ordinateurs et l'affectation de ces composants sur ces nœuds.



## **2.3. Analyse des besoins**

Avant de commencer à proposer une solution il faut tout d'abord faire ce qu'on appelle une étude de l'opportunité, en informatique cette étape est cruciale car elle permet de bien comprendre les objectifs à tracer avant de remédier à une issue concluante pour cet organisme (qui est INJAZ al-Djazair).

Alors pour faire une étude préalable il faut une bonne étude du système existant, c'est-à-dire la solution déjà existantes aux seins de l'entreprise et lui faire une critique objectif afin de démêler les difficulté rencontré par l'organisme, et recenser les besoins notamment en termes de fonctionnalités nouvelles, mais cette étude nous permet aussi par le même occasion d'étudier les besoins du projet, et à mieux desseller les potentiels difficulté à appréhendé lors des prochaine étapes et identifier les contraintes budgétaires les contraintes d'environnement, financière ou encore juridique.

### **2.3.1. Présentation de l'organisme d'accueil**

INJAZ El Djazair est une fraction de INJAZ El Arab, une organisation qui aborde les jeunes dans les 14 pays du en moyenne orient et en Afrique du nord.

INJAZ El Djazair, a entamé ses travaux en Algérie en Octobre 2010 et est installée depuis sous l'égide du club CARE - Cercle d'Action et de Réflexion autour de l'Entreprise sous le titre d'une association à but non lucratif agréée le 22 janvier 2003 sous le n° 291 - sis Lotissement RTT n° 18 à Dély Brahim à Alger.

INJAZ El Djazair est dirigé par un Conseil d'Administration présidé par M. Sofiane Chaib et sponsorisé par plusieurs sociétés comme OOREDOO ROUIBA LAFARGE BOEING...etc. le But de cette association est de Contribuer à Inspirer et éduquer les jeunes Algériens à l'esprit

d'initiative et aux valeurs de l'entreprise, les préparer à réussir une carrière professionnelle et à développer leur sens des responsabilités, et ce, en leur fournissant à travers un partenariat entre le secteur éducatif et le secteur privé, des programmes éducatifs complémentaires pertinents dans les domaines de l'Entrepreneuriat, la culture financière et le monde du travail, ces programmes sont fournis par des cadres supérieurs volontaires du secteur privé aux profit des élèves et étudiants.

### **2.3.2. Déroulement des formations :**

Au jour d'aujourd'hui, Injaz El Djazair propose des formations à des étudiants intéressés, via des événements non simple à organiser, ces événements sont financés par les sponsors de l'association où des volontaires font le boulot d'appréhender des sujets technologiques sous forme de mini-formation, qui se présente sous forme de petites présentations de plus au moins une heure ou deux.

Actuellement au sein d'Injaz les formations se font en cinq étapes :

1. Trouver un sujet intéressant, et cela se fait par une des deux approche suivantes :

Via Junior Achievements :

comme cité dans la partie consacré à la présentation d'accueil, Injaz El-Djazair est une filière de MENA Junior Achievement qui est appelé Injaz Al-Arab, c'est une ONG et son programme est créé avec beaucoup d'autres ONG internationales, avec un objectif principale de créer des cursus éducatifs dans le respect ses principes de bases, une fois terminé, ces programmes sont distribué à travers le monde par le billet les ONGs, avec sujets adapté pour chaque région, en utilisant le facteur de la langue ainsi que le législatif local comme contrainte à respecter.

Via les filières :

Les filières peuvent également proposer des programmes d'actualité et les convertir pour mieux répondre aux exigences linguistiques, morales et judiciaires du pays en question.

2. Trouver un lieu avec une date où il est disponible, cette étape s'avère être très compliqué, mais la prochaine est la pire ;
3. Trouver des formateurs pour faire la présentation, ces volontaires d'une journée sont souvent très pris dans leurs activités professionnels, et leurs demandes de se libérer est souvent une tâche très compliquer ;
4. Commencer l'organisation et partager l'événement avec les détails de la formation et sa date, pour les internautes abonnés dans les réseaux sociaux ;
5. Après toutes ces étapes que la formation se confirme et peut enfin commencer ;

Comme confirmé par les équipes d'exécution d'Injaz c'est généralement « pas simple » pour ne pas dire compliquer à mettre en place, sans compté les dépenses financière qui sont considérables.

Et afin de mieux prendre les bonnes décisions et analysé les tendances qui ont plus intéressé les étudiants, on fait un petit tableau de bord avec ce qu'on appelle des indicateurs de performances dans le domaine décisionnel, ces tableau de bord sont fait à partir d'Excel, ces indicateurs se constituent principalement de :

- Nombre de participants par formation ;
- Nombre de participants par rapport à chaque lieu où va se dérouler l'événement ;
- Nombre de participants par sujet ;
- Frais dépensés par événements ;

Ces chiffres sont analysés par le conseil d'administration qui travaillera à améliorer de plus en plus les événements proposé.

### **2.3.3. Problèmes à traiter :**

Donc si on résume bien, les agents d'Injaz font des pieds et des mains pour organiser ces événements ; Mais hélas, souvent les résultats obtenus ne sont pas à la hauteur des efforts fournis, car il y a pas mal de critiques dans les démarches actuelles et qui peuvent être nettement

optimisées, on site quelque unes :

- Le manque flagrant de visibilité : même si les pages d'Injaz ont un nombre d'abonné important (plus de 20000 abonnés sur la page Facebook en 2015), ça reste insuffisant, c'est un point qui doit être revue.
- Pas toujours au moment opportun : les dates des événements sont toujours un des gros points noirs de l'organisation des événements, et pour cause les dates doivent être adaptées en fonction de la disponibilité des lieux (souvent des salles) proposés par les établissements (souvent les universités) et de l'enseignant et des apprenants.
- Les volontaires au sacrifice de leurs vie professionnelle : les formateur qui sont des volontaire rappelons-le, qui sont la plupart du temps des professionnels, et ne sont pas toujours libre.
- La spontanéité des événements fonts obstacle à une bonne organisation : dans des événements comme ça, on a toujours des imprévus, ça débouche souvent sur des problèmes de dernières minutes.
- Des indicateurs de performances approximative : ces indicateurs qui sont essentiel au bon rendement de la société dans le future ; doivent être précis, et ce n'est pas toujours le cas avec des feuilles de calculs généré à peu près par des agents à l'aide de l'outil Excel.

Tous ces points peuvent être potentiellement améliorables, avec une solution informatique adaptée.

### 2.3.4 Objectifs

Le système à développer devra contourner ces problèmes, et atteindre les objectifs suivants :

- Le contenu des formations effectuées sont disponibles à tous moment pour tout individu inscrit gratuitement sur la plateforme,
- Les enseignants peuvent programmer et effectuer ou partager leurs cours sans aucune contrainte de lieu ou de temps.
- La plateforme sera dotée d'un système de diffusion de vidéo en direct, avec un ensemble d'outils d'interactivité à savoir un chat permettant un échange d'informations et de questions direct entre les apprenants et l'enseignant en temps réel.

- Répondre aux nouvelles technologies mobiles en proposant une version mobile de la plateforme qui va mieux pousser la mobilité et améliorer la visibilité.
- Partir sur une réalisation hybride pour une fonctionnalité sur tous les OS.
- Un hébergement sur serveur dédié virtuel (Cloud) pour optimiser le coût.
- Doter la plateforme d'un petit outil d'aide à la décision offrant quelque chiffres à la clé pour bien perfectionner les formations à présenter dans le futur.

### **2.3.5 Expression des besoins**

#### **1. Besoins non-fonctionnel :**

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation (langage de programmation, type SGBD, de système d'Exploitation...).

Dans le cadre de ce projet avec le principe du Massif du MOOC ; l'application ne doit pas avoir un temps de latence dès qu'il ya un grand nombre d'utilisateurs, et pour cet effet là qu'on va ce concentré sur l'aspect de performance et de rapidité, et moyennement sécurisé.

L'application devra :

- Avoir une interface souple, conviviale et interactive.
- Respecter la confidentialité des données.
- Etre codée de manière claire et structurée pour les futures évolutions ou améliorations.

#### **2. Besoins fonctionnel :**

Un bon système c'est un système qui doit satisfaire les besoins et les attentes des utilisateurs, ces besoins qui sont fonctionnels qui seront dans notre cas formalisés sous forme du modèle de cas d'utilisations.

#### a. Les acteurs :

Dans notre application nous avons pu détacher trois principaux types d'utilisateurs :

- Administrateur : a comme rôle de veiller au bon déroulement des cours de la plate-forme ; en gérant les compte des enseignants et les cours qu'ils proposent, en ayant le dernier mot sur les cours qui peuvent ou pas être présenter sur l'application, et pour une bonne gestion ils ont accès à un petit tableaux de bord mensuel pour mieux suivre les tendances des apprenants.
- Apprenant : C'est les élèves, ils peuvent via l'application chercher les cours qui les intéressent, ils s'abonnent dessus et ils peuvent suivre le cours, mais pour cela ils doivent être inscrits.
- Enseignant : leurs rôles est de proposer et présenter les cours sous autorisation des administrateurs.

#### b. Décomposition du système en catégories

Nous avons observé quatre grandes parties principales de notre plateforme à savoir :

1. l'inscription
2. la gestion des formations et cours
3. la gestion des utilisateurs
4. la gestion du profil

Les trois dernières parties devront passer par une authentification obligatoire.

La figure 2.1 représente le diagramme de cas d'utilisation global ou appelé aussi diagramme, de package qui représente les catégories observées mais permet d'indiquer les acteurs intervenant dans chaque package.

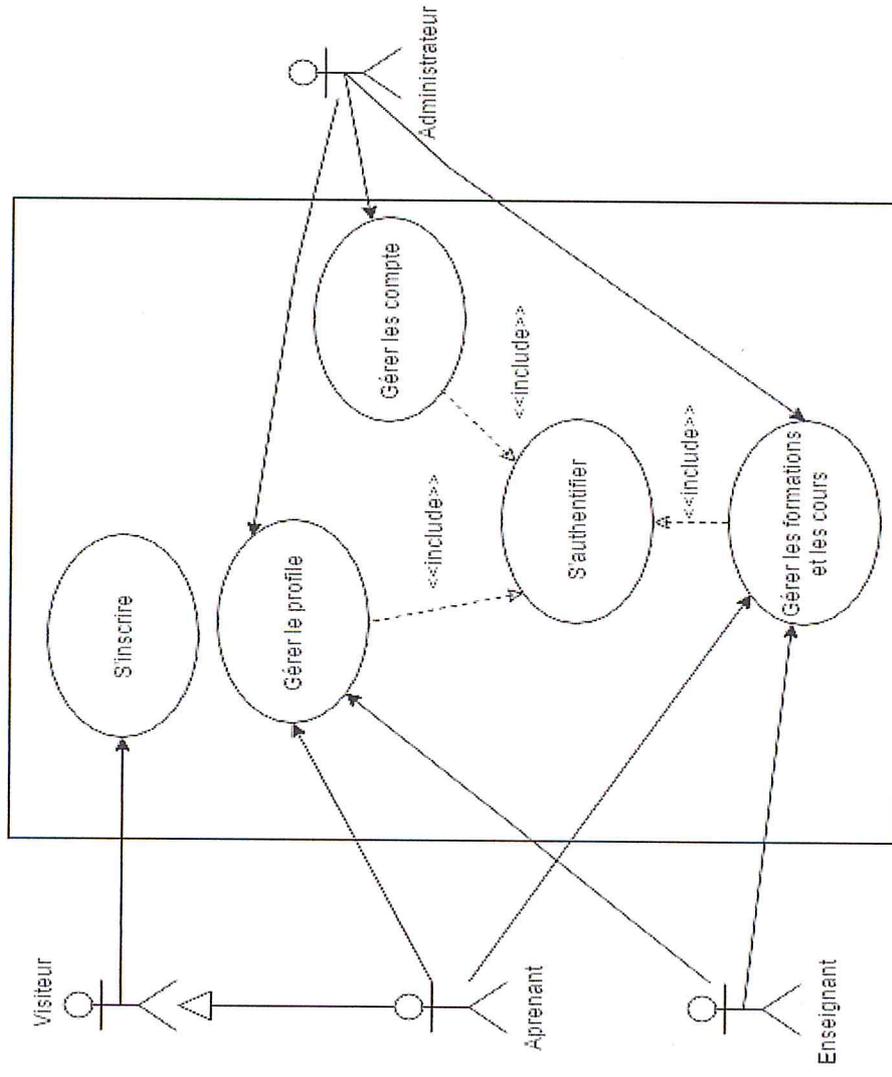


Figure 2.1 Diagramme de cas d'utilisation générale

## c. Les diagrammes de cas d'utilisation détaillés par catégorie

### 1. L'inscription

La figure 2.2 représente les cas d'utilisation liés à l'inscription de tout invité, cette inscription peut passer par le remplissage du formulaire classique d'inscription, ou encore via le compte de l'invité dans un des réseaux sociaux, cette méthode lui évitera de remplir ses informations personnelles et paramétrer son profil manuellement.

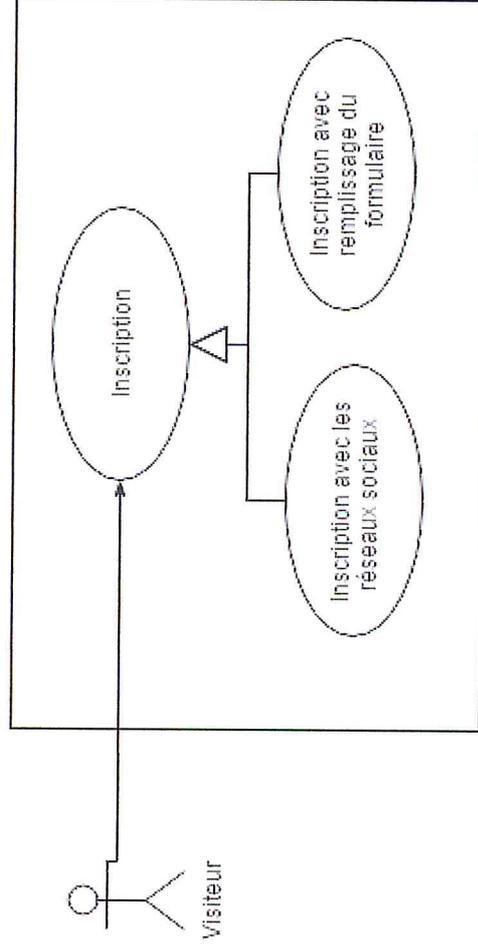


Figure 2. 2 Diagramme de cas d'utilisation d'inscription

Description du cas « inscription »
<b>Identification</b>
<ul style="list-style-type: none"><li>• nom du cas : « inscription »</li><li>• objectif : création d'un compte apprenant.</li></ul>

<ul style="list-style-type: none"> <li>• auteur principal : visiteur.</li> <li>• Auteur secondaire : néant.</li> </ul>
<p><b>Pré-condition</b> : néant.</p> <p><b>Enchaînement nominal</b> :</p> <ul style="list-style-type: none"> <li>• le visiteur lance se lance sur l'application.</li> <li>• Il émit une demande d'inscription.</li> <li>• Il aura deux choix soit remplir un formulaire soit s'identifier avec les réseaux sociaux</li> <li>• L'inscription est effectuée et le visiteur devient un apprenant.</li> </ul> <p><b>Enchaînement d'exception</b> :</p> <ul style="list-style-type: none"> <li>○ Si un champ n'est pas rempli alors un message est affiché « veuillez renseigner ce champ ».</li> <li>○ Si le e-mail du compte existe déjà, un message est affiché « Adresse e-mail déjà enregistrée dans notre système ».</li> </ul> <p><b>Post-condition</b></p> <ul style="list-style-type: none"> <li>• L'utilisateur devient apprenant.</li> <li>• L'utilisateur sera connecté au système, et redirigé vers la page principale de l'application.</li> </ul>

## 2. gestion des formations et des cours

La figure 2.3 représente les cas d'utilisations liés à la gestion des formations et des cours, ce package concerne les trois acteurs :

- L'administrateur pour créer et définir une formation et l'attribuer à un enseignant, Qui l'acceptera ou proposera un autres qui sera examiné par les administrateurs.
- L'enseignant pour diffuser les cours en line ou uploader des cours (texte ou vidéo) diffusable a n'importe quel moment.
- L'apprenant pour suivre les cours avec une interactivité possible avec l'enseignant grâce au commentaire (en direct ou pas).

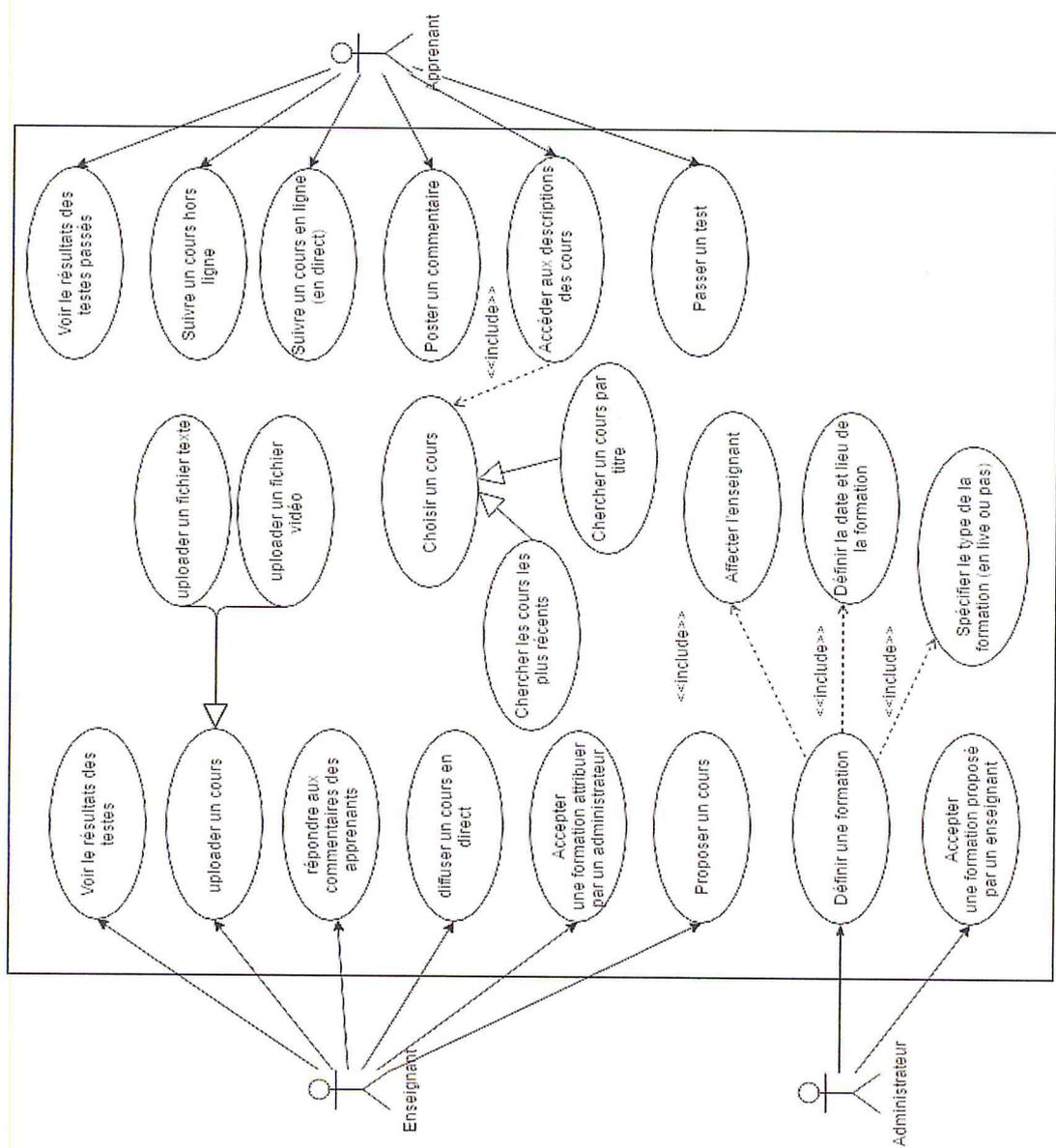


Figure 2. 3 Diagramme de cas d'utilisation de gestion de formations et des cours

**Description du cas « définir une formation »**

**Identification :**

- Nom du cas : « définir une formation »
- Objectif : ajouter une nouvelle formation et l'attribuer à un enseignant.
- Acteur principal : administrateur.
- Acteur secondaire : néant.

**Pré-condition :**

- Connecté au système en tant qu'administrateur.

**Enchaînement nominal :**

- L'administrateur demande à ajouter une nouvelle formation.
- Un formulaire pour détailler de la formation lui est renvoyé.
- L'administrateur le remplit en définissant :
  - Le nom et un petit descriptif de la formation.
  - L'enseignant qui sera chargé de le faire.
  - La date et l'heure du commencement de la formation.
  - Le lieu si c'est précisé (sinon cette attribut sera null)
- Le système ajoute le cours à la base de données.
- Le cours est maintenant ajouté.

**Enchaînement d'exception :**

- Si un des champs n'est pas rempli alors un message est affiché « veuillez saisir ce champ ».
- Si le titre du cours existe déjà dans la base de données un message est affiché « ce cours existe déjà ».

**Post-condition :**

- Le cours est ajouté et l'enseignant qui sera de le faire sera notifié dès qu'il sera connecté.

**Description du cas « accepter un cours proposé par un enseignant »**

**Identification :**

- Nom du cas : « accepter un cours attribué par un administrateur »
- Objectif : accepter un cours proposé par un enseignant.
- Acteur principal : administrateur.
- Acteur secondaire : néant.

**Pré-condition :**

- Connecté au système tant qu'administrateur.

**Enchaînement nominal :**

- L'administrateur demande la liste des cours proposé.
- Une liste des cours pas encore traité est affichée.
- L'administrateur choisit le cours à validé.
- Le système met à jour les informations du cours traité.
- Le cours est maintenant validé.

**Enchaînement d'exception :**

- Si il n y a aucun cours proposé alors un message est affiché « il n y a aucun cours proposé par les enseignants ».

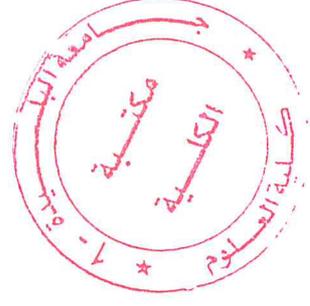
**Post-condition :**

- Le cours est validé et l'enseignant sera de le faire sera notifié dès qu'il sera connecté.

**Description du cas « recherche et inscription à un cours »**

**Identification :**

- Nom du cas : « recherche et inscription à un cours »
- Objectif : L'apprenant veut trouver un cours intéressant pour s'y inscrire.
- Acteur principal : apprenant.



- Acteur secondaire : néant.

**Pré-condition :**

- L'apprenant doit être connecté au système.

**Enchaînement nominal :**

- L'apprenant tape un mot clef dans la barre de recherche ou demande les cours les plus récents.
- Une liste de cours s'affiche.
- L'apprenant sélectionne le cours qu'il intéresse.
- La description du cours est affichée.
- L'apprenant s'inscrit en cliquant sur le bouton « s'inscrire ».
- Le système l'ajoute à la liste des suivis de ce cours.
- L'apprenant est maintenant inscrit.
- L'apprenant est dirigé vers le contenu du cours, et peut le consulter à n'importe quel moment.

**Enchaînement d'exception :**

- Si le mot clef ne correspond à aucun cours, alors un message est affiché « aucun cours trouvé ».

**Post-condition :**

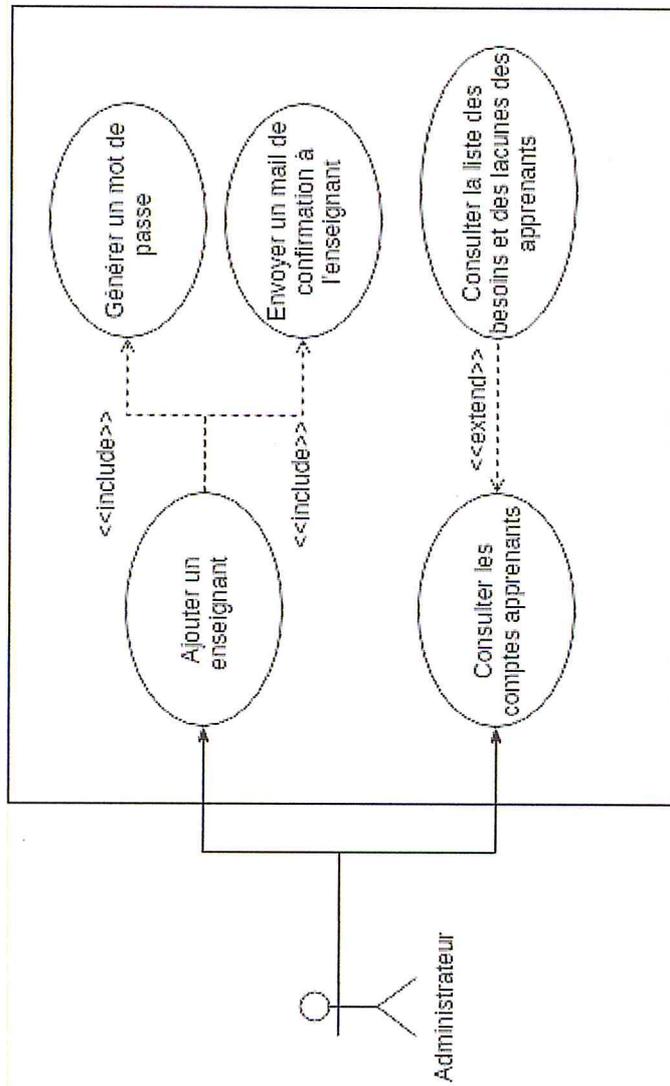


Figure 2. 4 Diagramme de cas d'utilisation pour gestion des utilisateurs

<b>Description du cas « ajouter un enseignant »</b>	
<b>Identification :</b>	
• Nom du cas : « ajouter un enseignant »	
• Objectif : ajouter un nouvel enseignant.	
• Acteur principal : administrateur.	
• Acteur secondaire : néant.	

**Pré-condition :**

- Connecté au système en tant qu'administrateur.

**Enchaînement nominal :**

- L'administrateur demande à ajouter un nouvel enseignant.
- Un formulaire pour détailler de l'enseignant lui est renvoyé.
- L'administrateur le remplit.
- Le système ajoute l'enseignant à la base de données et lui génère un mot de passe.
- L'enseignant est maintenant ajouté un e-mail de confirmation lui sera envoyé avec le mot de passe.

**Enchaînement d'exception :**

- Si un des champs n'est pas rempli alors un message est affiché « veuillez saisir ce champ ».
- Si l'e-mail de l'enseignant existe déjà dans la base de données un message est affiché « cet enseignant est déjà inscrit ».

**Post-condition :**

- L'enseignant est ajouté.

<b>Description du cas « consulter les comptes apprenants »</b>	
<b>Identification :</b>	<ul style="list-style-type: none"> <li>• Nom du cas : « consulter les comptes apprenants »</li> <li>• Objectif : consulter les mots clés qui expriment plus précisément les sujets de formations que veulent les apprenants.</li> <li>• Acteur principal : administrateur.</li> <li>• Acteur secondaire : néant.</li> </ul>
<b>Pré-condition :</b>	<ul style="list-style-type: none"> <li>• Connecté au système tant qu'administrateur.</li> </ul>
<b>Enchaînement nominal :</b>	<ul style="list-style-type: none"> <li>• L'administrateur demande la liste des lacunes des apprenants.</li> <li>• Une liste des mots clés les plus fréquents lui sera renvoyée.</li> </ul>
<b>Enchaînement d'exception :</b>	<ul style="list-style-type: none"> <li>• Si la liste est vide un message est affiché « pour l'instant la listes des lacunes des apprenants est vide ».</li> </ul>
<b>Post-condition :</b>	

- L'administrateur consulte la liste et prend la décision de quel formation sera programmé.

#### **4. la gestion du profil**

La figure 2.5 représente les cas d'utilisation liés à la gestion du profil de l'apprenant, cette gestion concerne la modification de mot de passe, photo de profile ou e-mail.

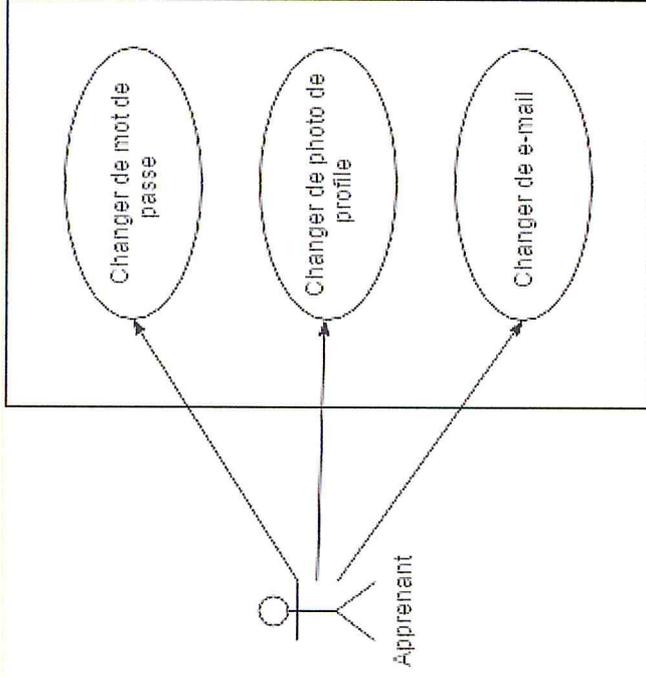


Figure 2. 5 Diagramme de cas d'utilisation pour gestion de profile

Description du cas « gestion de profile »	
<b>Identification :</b>	
•	Nom du cas : « gestion de profile »
•	Objectif : modifier quelque attribut du profile.
•	Acteur principal : apprenant.
•	Acteur secondaire : néant.

**Pré-condition :**

- Connecté au système tant qu'apprenant.

**Enchaînement nominal :**

- L'apprenant demande de modifier son profil.
- Un formulaire lui est renvoyé avec comme champs les attributs à modifier.
- L'apprenant modifie le champ qu'il souhaite. Et enregistre le tout.
- Les champs du profil seront modifiés.

**Enchaînement d'exception :**

- Si le nouveau e-mail existe déjà un message est affiché « l'e-mail appartient déjà à un autre utilisateur ».

**Post-condition :**

- Les champs seront modifiés.

## 2.4 Conception

Avant d'entamer proprement dit la réalisation de l'application ; une bonne conception s'impose afin de mieux analyser les fonctionnalités du système, et surtout pour voir si la solution proposée répond au mieux aux problèmes posés, et pour cela nous allons dessiner au mieux les interactions possibles entre les différents acteurs et le système, le tout à travers le langage de modélisation UML.

### **2.4.1 Diagrammes de séquence**

Diagramme de séquence montre la séquence verticale des messages passés entre objets au sein d'une interaction.

### a. Authentification

La figure 3.1 représente le diagramme de séquence pour l'authentification ; avec l'e-mail et le mot de passe.

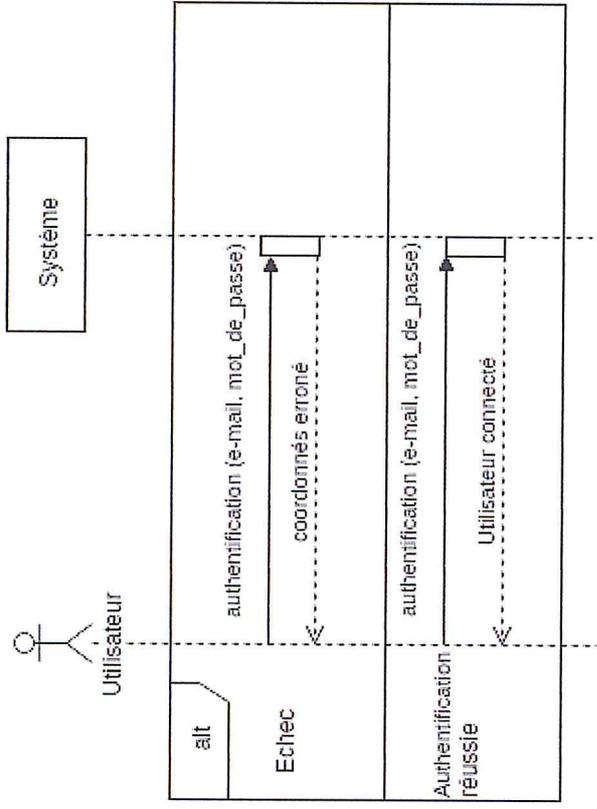


Figure 3. 1 Diagramme de séquence pour l'action d'authentification

**b. Suivre une formation**

La figure 3.2 représente le diagramme de séquence pour suivre une formation, pour cela une inscription s'impose. Pour qu'il soit redirigé vers la page du cours live ou pas.

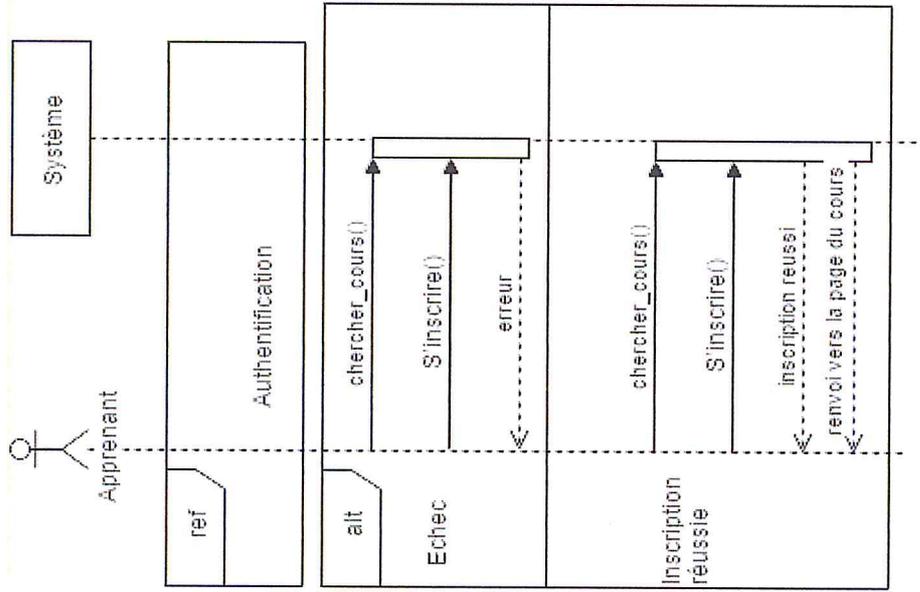


Figure 3. 2 Diagramme de séquence pour suivre dans une formation

### **b. Proposition d'une formation**

Le diagramme de séquence suivant (figure 3.3) représente l'action de proposition de cours par un enseignant, avec une petite description qui sera renvoyée aux administrateurs qui auront la décision finale de programmer ou pas ce cours.

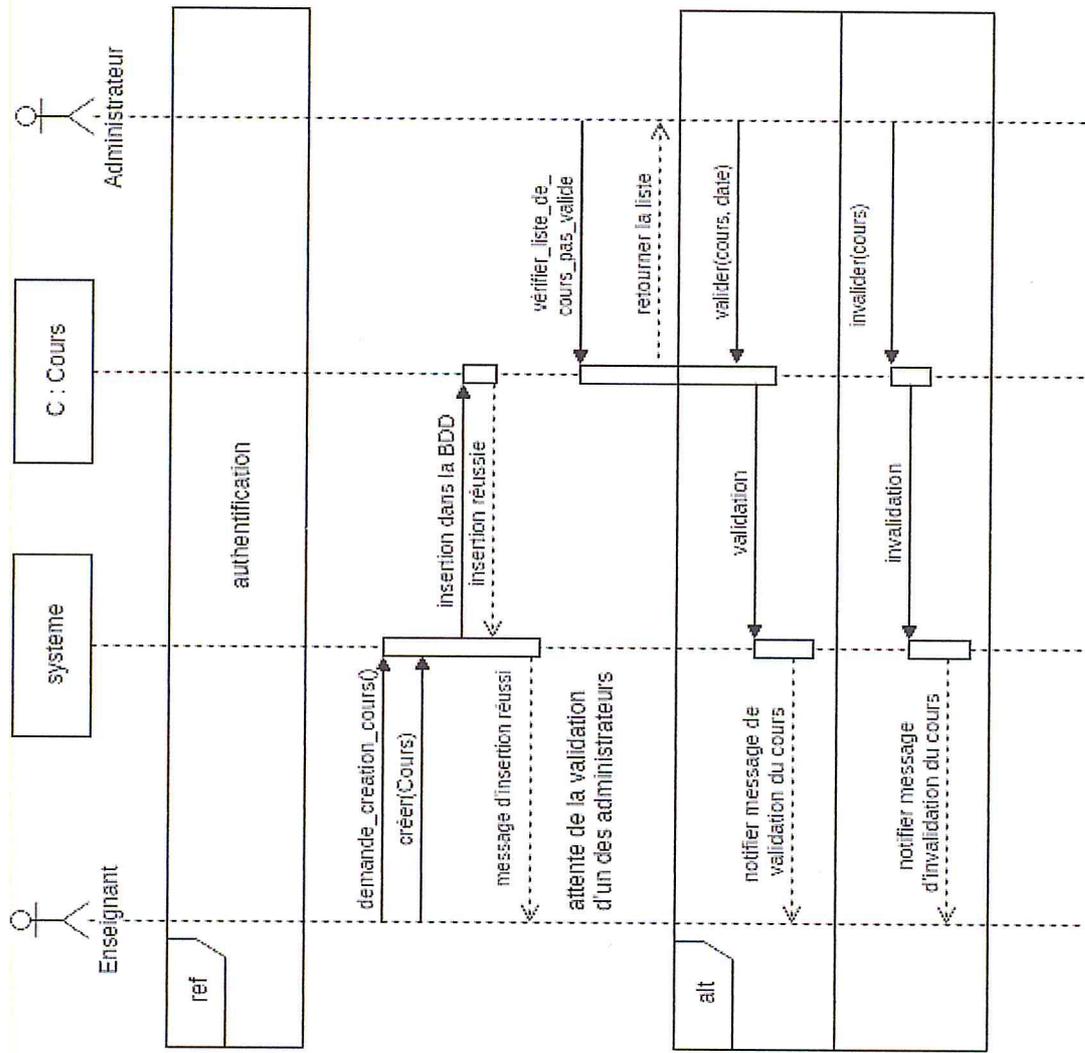


Figure 3. 3 Diagramme de séquence pour soumettre une formation



**c. Uploader un cours hors ligne**

Le diagramme suivant (figure 3.4) représente le diagramme de séquence pour uploader un chapitre d'une formation.

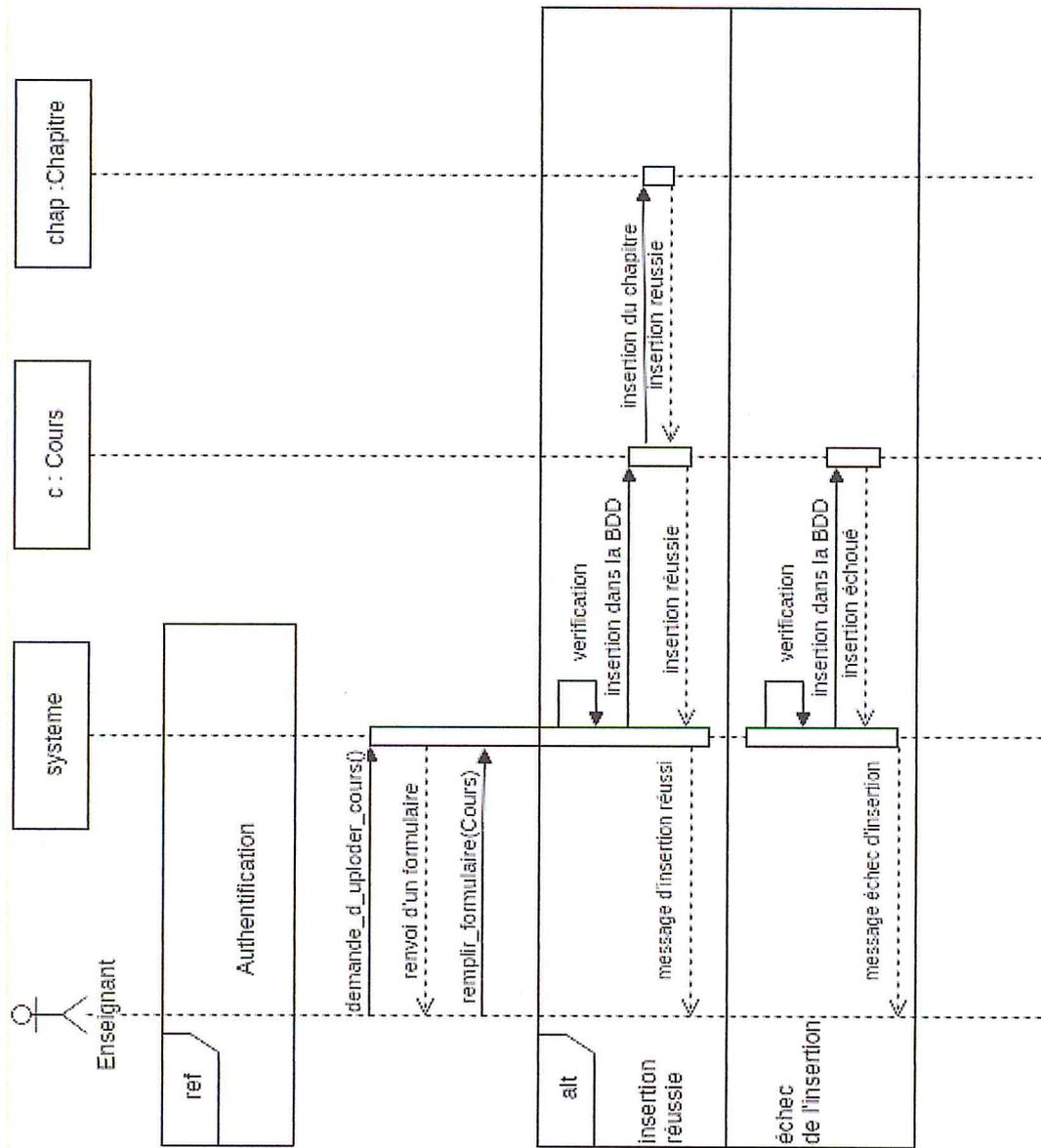


Figure 3. 4 Diagramme de séquence pour uploader un cours

**d. Diffusion et suivis d'un cours en ligne (en direct)**

La figure 3.5 représente le diagramme de séquence pour suivre un cours en direct, avec les interactions entre l'enseignant et l'apprenant (commentaire en direct et le quiz à la fin du cours)

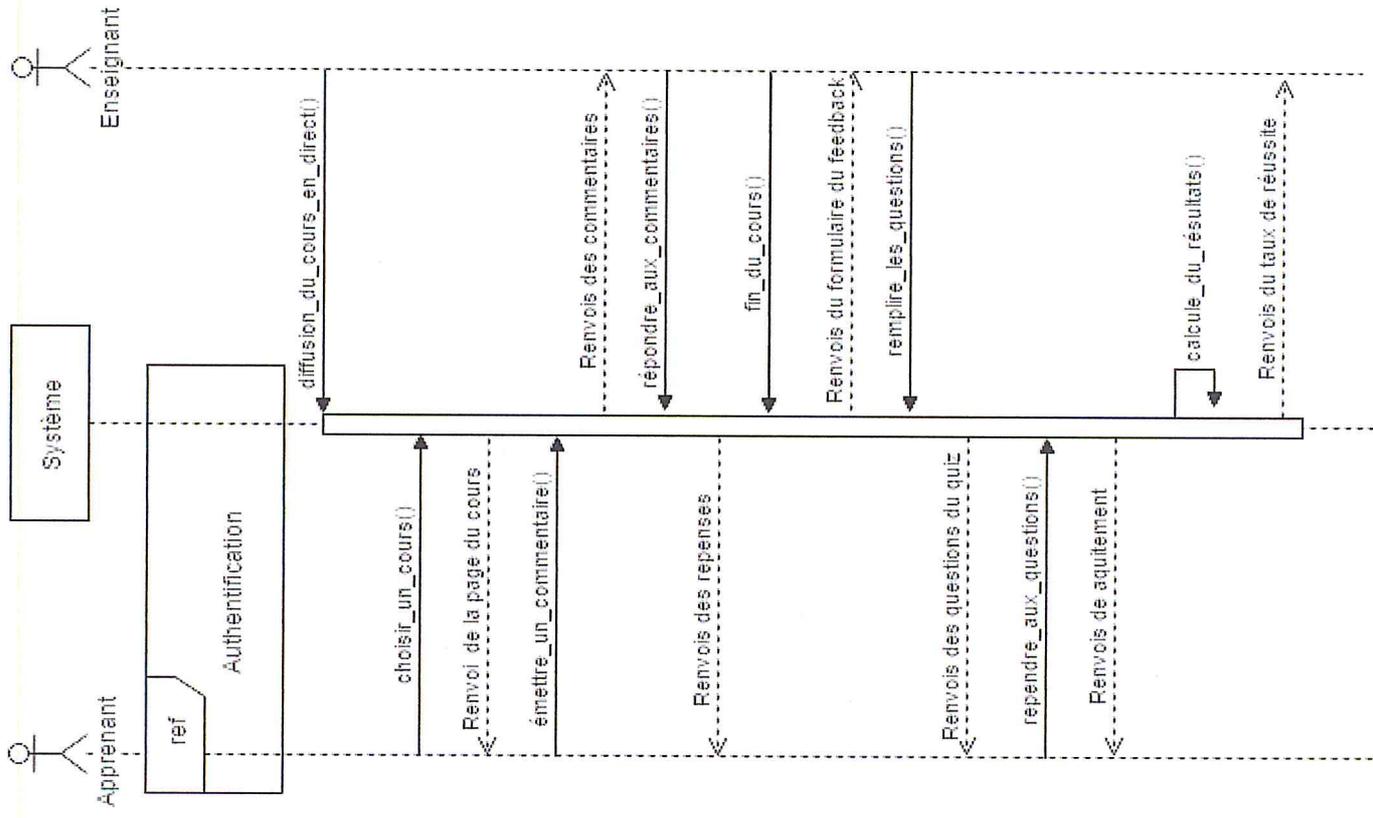


Figure 3. 5 Diagramme de séquence pour suivre un cours en ligne

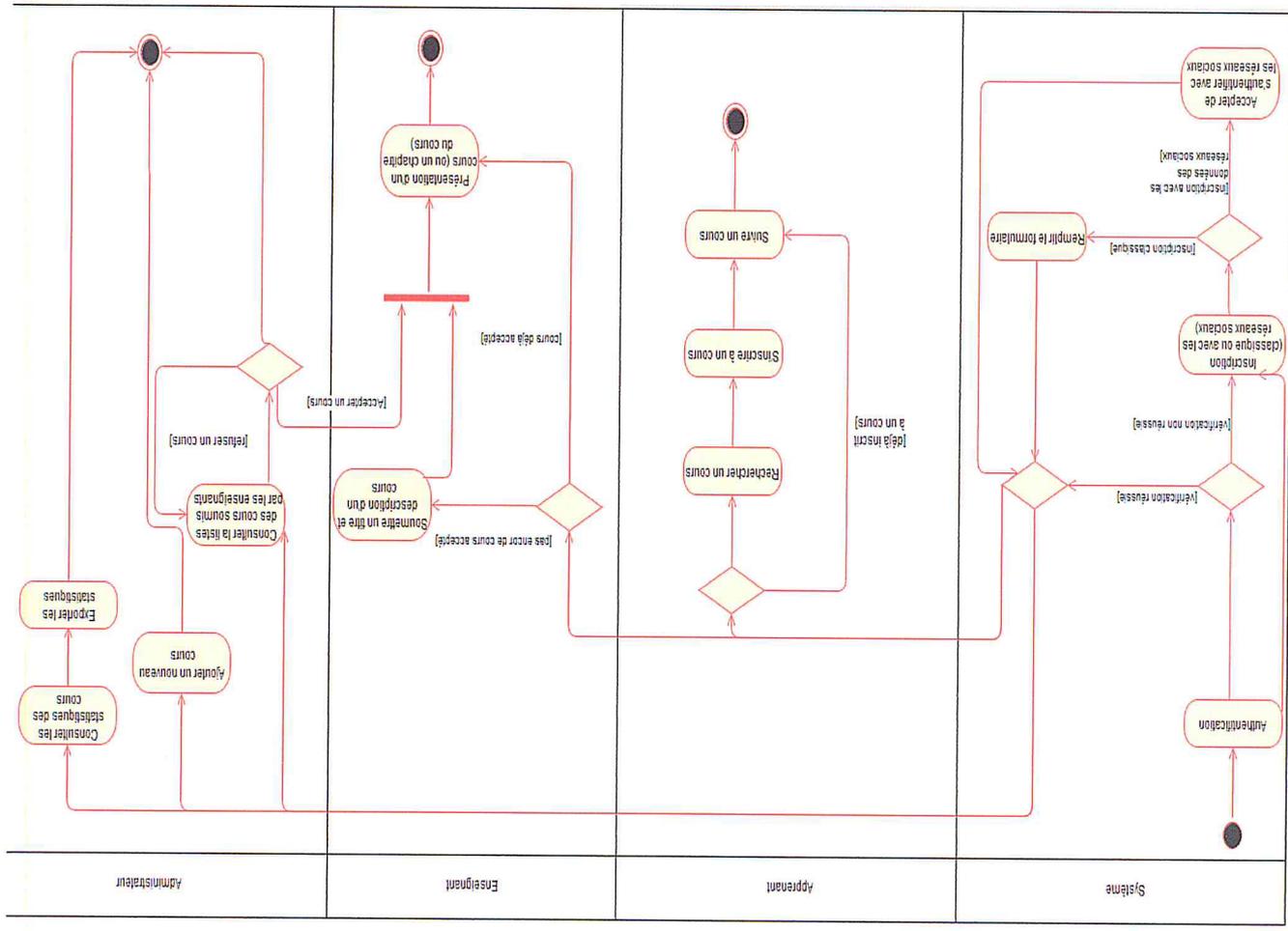
#### 4.2.2 Diagramme de classes :

Diagramme de classes montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc. Et représente la conception de la base de données (figure 3.6).



### 4.2.3 Diagramme d'activité

Le diagramme d'activité qui représente le traitement des données entre les différents rôles du système (figure 3.7).



**Figure 3. 7 Diagramme de d'activité du système**

#### 4.2.4 Diagrammes de communication

Le diagramme de communication représente les échanges entre les différents objets, du système.

a. **Les échanges suite aux actions de l'apprenant :**

Les échanges de données que peuvent avoir les objets suite aux interactions d'un apprenant.

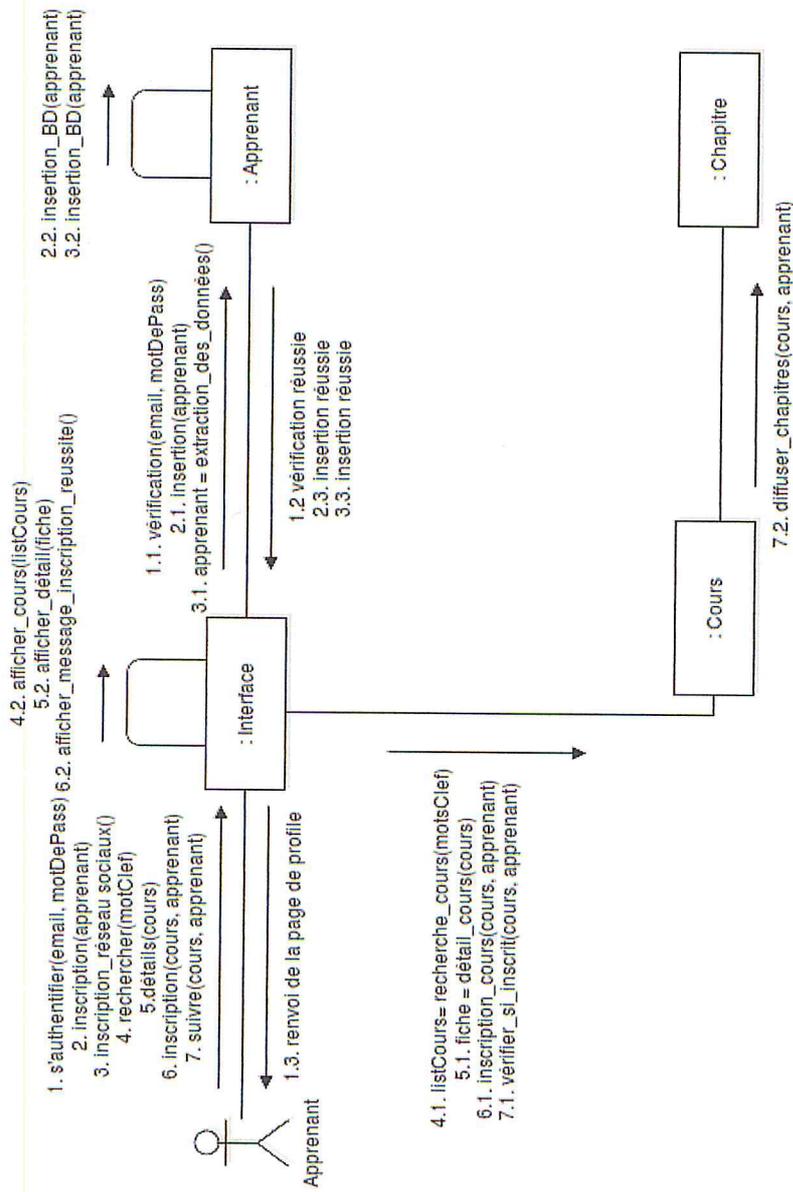


Figure 3. 8 Diagramme de communication de l'apprenant

**b. Les échanges suite aux actions de l'enseignant :**

Les échanges de données que peuvent avoir les objets suite aux interactions d'un l'enseignant.

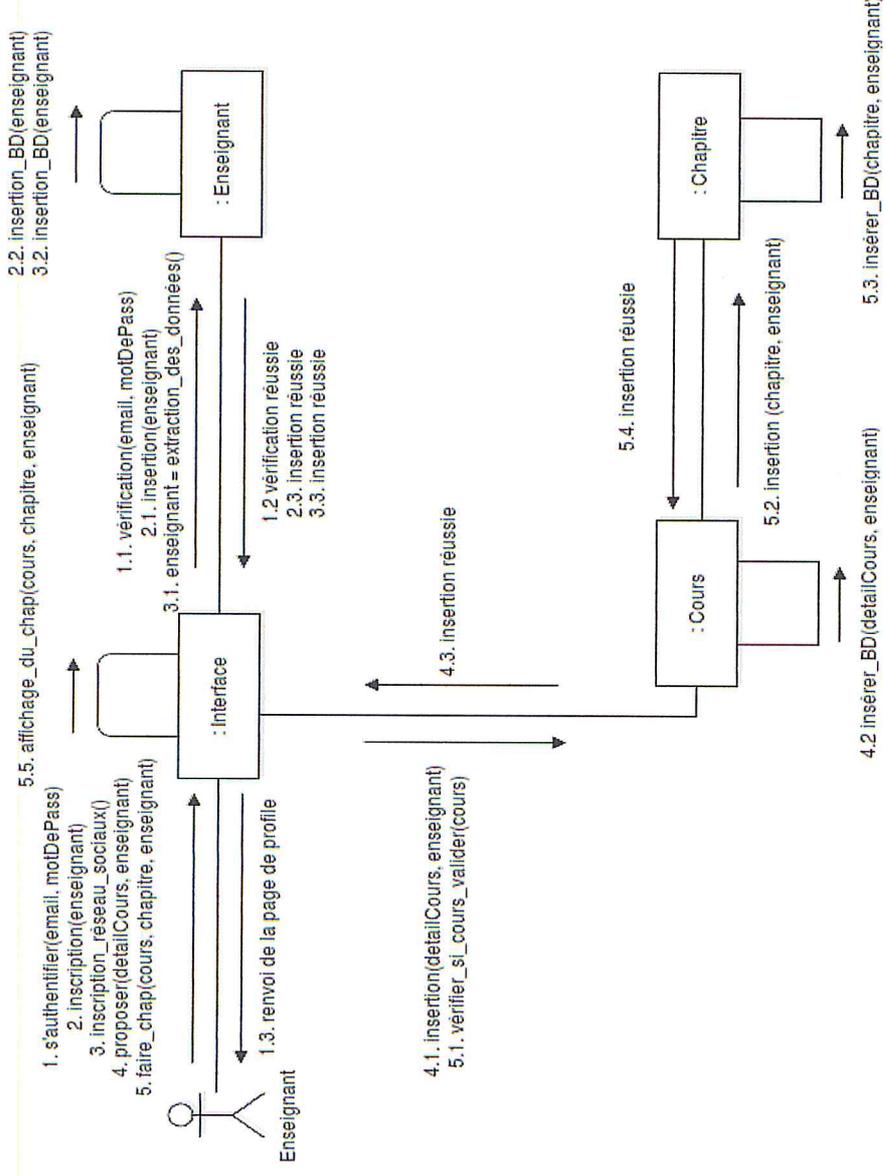


Figure 3. 9 Diagramme de communication pour l'enseignant

### 4.2.5 Diagramme de déploiement

Le diagramme de déploiement représente l'infrastructure physique du système (figure 3.10)

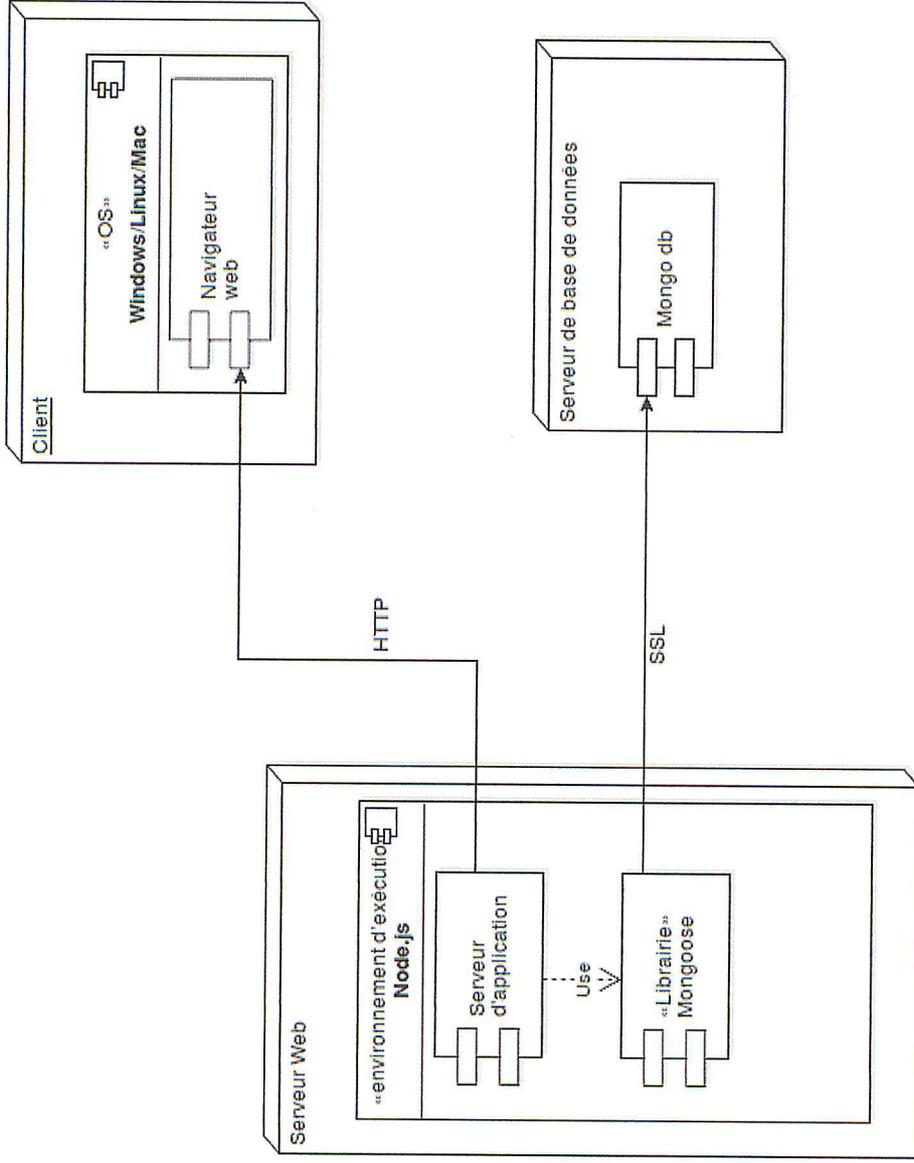


Figure 3. 10 Diagramme de déploiement du système

## 4.3 Conclusion

Dans ce chapitre, l'étude d'analyse effectuée dans ce chapitre a servi, de première ébauche à la création du modèle de conception ou on a utilisé en premier lieu les diagrammes de séquences équivalents, ainsi que les diagrammes de communication et qui donne une idée de la complexité et du niveau de détail introduit dans le modèle de conception par rapport au modèle d'analyse. Le modèle de conception nous a permis de façonner notre plateforme tout en préservant autant que possible la structure définie par le modèle d'analyse et en prenant aussi en compte les besoins et les exigences définis auparavant par les utilisateurs. Ce modèle nous servira de plan d'élaboration et de construction de l'implémentation.

Le prochain chapitre sera le dernier, nous allons rentrer dans la partie réalisation ou nous allons voir le point final de notre projet.

# Chapitre 3 : Réalisation

## 3.1 Introduction:

Dans les chapitres précédents, nous avons introduit l'état d'esprit de notre conception, et les principaux composants qui vont composer notre plate-forme. Dans ce chapitre nous allons introduire l'aspect technique de notre application, en introduisant les différents services qu'elle peut fournir, on va aussi aborder le futur et des fonctionnalités à ajouter dans un avenir proche.

## 3.2 Architecture de l'application

Les applications web de nos jours connaissent une très grande percée, dans un siècle avec une technologie sans précédent, cela veut dire qu'aujourd'hui les modèles ne manquent pas et choisir une bonne architecture adaptée s'avère essentiel afin d'avoir des services les plus précis et les plus optimaux que possible.

Pour nous les avantages que procure le modèle SOC (separation of concerns) est très avantageux, on cite ces quelques points :

a. Une application mixte : cela se présente en un mix du côté serveur avec le côté client dans une seule et même application, comme c'est le cas pour les applications PHP, mais il y a un hic, il y a un risque dans le côté intégration de l'application.

b. SPA (single page application) : c'est de loin le modèle le plus adapté pour les récentes applications web 2.0, et sépare la partie client de la partie serveur, et se focalise sur la partie client pour une expérience UI/UX la plus convenable, et pouvoir répondre mieux aux goûts des clients, et ce pattern est utilisé par les grands tel que Facebook, Gmail, G+...etc.

Clairement, les calculs et ressources utilisés vont être considérables, et c'est pourquoi l'utilisation du SPA est nécessaire, parce que notre application va grandir en terme de données et en terme d'utilisateurs, et ce qui va engendrer comme conséquence sur les performances n'est pas top.

### 3.2.1 Les différents composants de la Plate-forme

Platform diverse and different components:

- Backend : Coté Serveur (API Hosting): crée en utilisant NODEJS.
- Frontend : Coté Client : en utilisant HTML et JavaScript.
- Real Time Data + Social login (authentification avec les réseaux sociaux) : avec Firebase Cloud Service.

### 3.2.2 Implémentation :

Le choix de notre approche étant donc SOC, maintenant pour la partie implémentation va être en quelque étape :

- Développer le Backend avec commonJS modularway.
- Développer une base de donnée NoSQL.
- Développer l'interface cotée client avec HTML5 et JavaScript.

## 3.3 Développement du côté service avec NodeJS

### 3.3.1 Les modules NPM:

Le fait que cette plate-forme NodeJS utilise l'API commonJS rend le code extrêmement modulable, et c'est l'un des plus grand avantage de NodeJS, et pour cause la communauté à travailler sur plus de 250.000 utilisable sur ces 4 dernières années, et c'est évident la présence d'un tel nombre de modules rends forcément la vie plus simple à nous autres développeur.

Et concernant les principaux modules utilisés nous avons :

- Express : pour mieux manipuler les APIs et dont on a besoin
- Mongoose : afin de mieux manipuler MongoDB NoSQL et une meilleurs gestion des données.
- Passport : pour l'Authentication convenable.
- Express-jwt/jsonwebtoken : pour mieux gérer notre Json Web Token sur le mécanisme authentification.
- Path: pour le réseautage.
- BodyParser : pour le transfert de données.
- Jade : pour mieux manipuler la mise en page et le rendu HTML (template).

Et plus, on utilise de nombreux modules pour mieux aider durant la période de développement, et nous allons les enlever une fois que notre application prête pour l'utilisation.

La figure ci-dessous représenter une petit portion de code JavaScript, et on y voit les différents modules que nous utilisons et comment nous appliquons le principe CommonJS.

```
app.js
1  var express = require('express');
2  var path = require('path');
3
4
5  // DB Logic
6  require('./app_api/models/db');
7
8  // Passport Auth Logic
9  require('./app_api/config/passport');
10
11 // Handle Routes API
12 var routesApi = require('./app_api/routes/index');
13
14 var app = express();
15
16 // Handle Express Logic
17 require('./app_api/config/express')(app, express);
18
19 // Handling API routes when path starts with /api
20 app.use('/api', routesApi);
21
```

Figure 4. 1 : l'approche commonJS au sein de notre application.

### 3.4 Application Behavior :

Notre approche est clairement différentes des anciennes applications, qui consistaient à envoyer toute les pages index.html, au lieu de cette méthode qui n'est pas des plus optimiser, la nôtre a pour approche d'envoyer un modèle ou quelque chose de plus comme un conteneur qui sera l'hôte de notre côté client.

Dans la figure illustre bien comment le module Express utilise le langage de tamplate Jade.

```

express.js app_api\config
1  var path = require('path');
2  var favicon = require('serve-favicon');
3  var logger = require('morgan');
4  var cookieParser = require('cookie-parser');
5  var bodyParser = require('body-parser');
6  // [SH] Require Passport
7  var passport = require('passport');
8
9
10 module.exports = function(app, express) {
11   // view engine setup
12   app.set('views', path.join(__dirname, 'views'));
13   app.set('view engine', 'jade');

```

Figure 4. 2 : initialisations du générateur de template jade dans Express.

Et dans celle-ci, qui montre l'envoi du fichier index.jade argumenté de quelque donnée.

```

index.js routes
1  var express = require('express');
2  var router = express.Router();
3
4  /* GET home page. */
5  router.get('/', function(req, res, next) {
6    res.render('index', { title: 'Express' });
7  });
8
9  module.exports = router;
10

```

Figure 4. 3 : Envois de index.jade via la requête Get

### 3.4.1 Json Web Token Authentication:

#### a. Definition:

JSON Web Token (JWT) est un des moyens pour représenter les données à transférer entre deux parties URL-safe. Les données dans un JWT sont codés comme un objet JSON qui porte une Signature JSON Web (JWS) ou plus précisément comme un texte avec une structure d'un JSON Web Encryption (JWE), ce qui permet les données à être signé numériquement et surtout garantir leurs intégrité avec un code encrypté.

#### b. Cas d'utilisation :

Quelque scénario où le JSON Web Tokens est vraiment utile:

Authentification : c'est le scénario le plus courant pour l'utilisation de JWT. Une fois que l'utilisateur est connecté, chaque demande subséquente comprendra le JWT, permettant à l'utilisateur des voies d'accès, des services et des ressources qui sont autorisées à ce jeton. Single Sign On est une fonctionnalité qui utilise largement JWT de nos jours, en raison de sa légèreté et sa capacité à être facilement utilisé dans différents domaines.

Échange d'information : JSON Web Tokens est un bon moyen de sécuriser la transmission des informations entre deux parties, pour cause chaque information peut être signé, par exemple utiliser une clé publique/privée entre les pairs, par cette manœuvre on est sûr que les envoyeurs ne sont pas des intrus, en plus, comme la signature est calculée en utilisant le header et le payload, on peut être sûr que le contenu n'a pas été altéré.

### **c. La structure de JSON Web Token**

JSON Web Tokens se composent de trois parties séparées par des points, qui sont :

Header : qui se compose généralement de deux parties: le type du jeton, qui est JWT, et l'algorithme de hachage utilisé, comme HMAC SHA256 vs RSA.

Payload : Les réclamations sont des déclarations sur une entité (généralement, l'utilisateur) et des métadonnées supplémentaires. Il existe trois types de revendications: réservé, public et créances privées.

Signature : qui représente le type d'encodage du header et du payload la signature est utilisée pour vérifier que le message n'a pas été modifié en cours de route.

### **d. Principe de fonctionnement :**

Pour les principaux composants du JWT, il y a plusieurs implémentations possibles qui diffèrent d'un langage à un autre, tout en sachant qu'il n'y a pas encore d'implémentation parfaitement stable, mais le principe de travail est commun et est défini comme suit :

1. Identification des requêtes JWT.
2. Chiffrer chaque requête JWT en base64.
3. Concaténer les codes obtenus avec une virgule comme séparation.

Et c'est là que c'est intéressant du fait que seules les différentes parties de l'application connaissent l'encodage exact de ces chiffrements.

Et c'est ce code-là qui sera envoyé au demandeur (client) de la requête, et sera stocké

localement chez le client ou à travers le navigateur web utilisé,

### 3.4.2 Outils de développement :

Pour le côté développement on a utilisé les outils suivants :

#### a. Code Visual Studio

Code Visual Studio est un éditeur de code source léger mais puissant qui fonctionne sur votre bureau et est disponible pour Windows, OS X et Linux. Il a un support intégré pour JavaScript (le langage majoritaire dans le code de l'application), tapuscrit et Node.js (ce qui nous a beaucoup aidé) et dispose d'un riche écosystème d'extensions pour d'autres langages (C ++, C #, Python, PHP).

#### b. Nodemon :

Nodemon est un autre module NodeJS puissants qui fonctionnent et d'exécuter sur la console ou un terminal, utilisé pour la course et l'exécution de notre fichier d'application principal de NodeJS et a beaucoup d'attribut et paramètre qui peut compter sur lors de l'exécution.

## 3.5 Implémentation de la base de données:

### 3.5.1 MongoDB:

#### a. Définition :

MongoDB (de l'anglais humongous qui peut être traduit par « énorme ») est un système de gestion de base de données orientée documents, répartitionnable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données. Il est écrit en C++. Il est conçu pour l'évolution, les performances et la haute disponibilité, passant de l'installation sur de simples serveurs, à des architectures complexes déployées sur plusieurs sites. En tirant profit du calcul en mémoire, MongoDB permet de lire et d'écrire très rapidement. La réplication intégrée à MongoDB et le basculement automatique, permettent une souplesse opérationnelle et une fiabilité de classe « entreprise » ; et il est le plus populaire des SGBD NoSQL

#### b. Avantages par rapport aux bases de données relationnelles :

Nous avons décidé d'utiliser une base de données NoSQL et plus particulièrement mongoDB pour les raisons suivantes :

- Le sharding de MongoDB : MongoDB depuis sa version 1.2 intègre une gestion de sharding. L'application de cette gestion permet d'avoir une scalabilité plus importante. Nos données sont toutes cohérentes, car on accède toujours à tout. Le concept du sharding n'est pas d'avoir plusieurs bases de données en master/master, mais une seule base de donnée qui répartie ses propres données dans les différentes instances. On se retrouve ainsi à posséder certaines données sur une instance et les autres sur une autre.
- GridFS : MongoDB embarque avec lui un système de fichiers complet. Ce système est GridFS. MongoDB est limité par document à 4Mo, mais on a vite besoin de stocker des documents plus importants. GridFS permet ça. On stocke donc tout document supérieur à 4 Mo sur le système de fichier GridFS. Ça permet ainsi d'avoir le sharding sur ses fichiers. On a donc besoin de rsync entre plusieurs machines. GridFS fait le boulot pour soi. On pourrait facilement rapprocher GridFS à Hbase.
- Un système de requêtes évolué : Enfin le vrai intérêt de MongoDB selon moi, surtout par rapport à Riak ou CouchDB, c'est sa gestion de requêtes. On peut requêter de façon très fine grâce à une grande quantité de mots-clé. On peut donc faire des requêtes aussi riches qu'en SQL, mais tout ça sur une base de données orientée Documents.

### 3.6 Implémentation du Front End / du côté clients :

Cette partie nous allons la développée à part et utilisé d'autre modèles et architectures adapté, et nous allons les utiliser pour implémenté la plate-forme SPA (défini précédemment), et pour cela nous avons adopté :

AngularJS Framework : AngularJS faciliter le processus de construction d'une page d'application unique, et en utilisant le modèle dynamique qui est Angular utilisant du MVC à MVVM rend votre application plus puissante en terme de vitesse, plus souple.

MaterialAngular : En utilisant MaterialAngular pour avoir un aspect visuel agréable à l'utilisateur et avec une facilité convenue.

Angular -X : AngularJS est conçu pour être modulaire et intégrable ; pour cette question, nous utilisons un module «X» qui signifie les nombreux modules communautaires d'AngularJS ; et ça va beaucoup nous aider au cours du processus de développement et la figure ci-dessous illustre les différents Module angulaire nous sommes en train d'utiliser.

```
bower.json client
1  {
2    "name": "mooc",
3    "version": "1.3.0",
4    "dependencies": {
5      "angular": "1.5.0",
6      "angular-animate": "1.5.0",
7      "angular-aria": "1.5.0",
8      "angular-chart.js": "0.8.8",
9      "angular-chartist.js": "3.3.13",
10     "angular-cookies": "1.5.0",
11     "angular-datatables": "0.5.2",
12     "angular-google-maps": "2.3.2",
13     "angular-material": "1.0.5",
14     "angular-messages": "1.5.0",
15     "angular-nvd3": "1.0.5",
16     "angular-resource": "1.5.0",
17     "angular-sanitize": "1.5.0",
18     "angular-timer": "1.3.3",
19     "angular-translate": "2.9.0",
20     "angular-translate-loader-partial": "2.9.0",
21     "angular-ui-calendar": "1.0.1",
22     "angular-ui-router": "0.2.18",
23     "c3-angular": "1.0.1",
```

Figure 4. 4 : les différents modules supportés par AngularJS

Remarque : pour des raisons de gestion de la dépendance, nous utilisons Bower un gestionnaire de dépendance puissante. Cela va gérer le téléchargement et le stockage pour toute la dépendance que nous pourrions besoins durant le développement.

### 3.7 La partie hébergement

Concernant l'hébergement, on doit suivre un certain nombre de contrainte pour mieux profiter au mieux des avantages de la Platform d'hébergement choisi, car il y a plusieurs fournisseur et a chacun ces offres.

Pour nos besoins nous avons basé notre choix sur les critères suivant :

Disponibilité : parce que nous avons besoin que notre application soit toujours en ligne à cet effet une plate-forme qui ne se bloque ou n'est jamais hors ligne est nécessaire.

Evolutivité : notre application aura souvent besoin d'un passage à l'échelle, donc avoir une plate-forme qui dispose de cette option est une grande nécessité.

Sécurité : notre application suit les procédures de sécurité les plus récentes, mais parfois les menaces de sécurité peuvent se produire à partir de la plate-forme d'hébergement elle-même, donc avoir un hébergement sécurisé est également une nécessité urgente.

### 3.7.1 Les plates-formes d'hébergement :

De nos jours les Platform d'hébergement Cloud sont très demandé ce qui fait qu'un grand nombre de sociétés proposent leurs services, on cite les plus connus :

**Google Cloud Platform** : Google Cloud Platform est une suite de services de cloud computing publics offerts par Google. La plate-forme comprend une gamme de services hébergés pour calcul, de stockage et de développement d'applications qui fonctionnent sur le matériel Google. Services Google Cloud Platform peuvent être utilisés par les développeurs de logiciels, les administrateurs de nuages et d'autres professionnels de l'entreprise de TI sur l'Internet public ou via une connexion réseau dédiée. Google Cloud Platform est composée de plusieurs produits le plus intéressant dans notre cas est Google App Engine, c'est une plate-forme en tant que produit de service (PaaS) qui fournit aux développeurs de construire une gamme de programmes allant de simples sites web à des applications complexes. App Engine offre du changement d'échelle automatique, augmentant les ressources pour faire face à la charge du serveur.

**Microsoft Azure** : anciennement connu sous le nom de Windows Azure, est la plate-forme cloud public de Microsoft. Elle offre une gamme de services de cloud computing, pour calculer, analysé, le stockage et la mise en réseau. Les utilisateurs peuvent choisir de ces services à développer et mettre à l'échelle de nouvelles applications, ou d'exécuter des applications existantes, dans un cloud public.

**Amazon Web Services (AWS)** : une filiale d'Amazon.com, offre une gamme de services de cloud computing qui constituent une plate-forme informatique à la demande. Ces services fonctionnent à partir de 13 régions géographiques à travers le monde. Les plus connu de ces services sont Amazon ElasticCompute Cloud, également connu sous le nom "EC2" permet à des tiers de louer des serveurs (IaaS) sur lesquels exécuter leurs propres applications

web, et le service Amazon Simple Storage, également connu sous le nom Amazon S3 offre des services de stockage à travers des services Web (IaaS).

**Heroku** : Heroku est une plateforme de développement basée sur le cloud en tant que service (PaaS) fournisseur. La plate-forme Heroku soutient le développement en Ruby on Rails, Java, Node.js, Python, Scala et Clojure.

### 3.7.2 Notre choix :

Comme nous l'avons mentionné, nous avons besoin d'un fournisseur Service Cloud qui a un service de passage à l'échelle, nous permettre de configurer assez vite notre application et nous donner un ensemble de configuration et de pouvoir sur notre application et pour nous Heroku est la solution.

### 3.7.3 Avantages de Heroku

Les raisons qui nous ont poussées à choisir cette Platform PaaS :

- Un plan gratuit jusqu'à une certaine charge.
- Beaucoup d'Add-on des ressources (applications, bases de données, etc.).
- Processus de mise à l'échelle indépendant pour chaque composant de votre application sans affecter la fonctionnalité et la performance de celle-ci.
- Isolement chaque processus est complètement isolé de l'autre avec dyno qui est un conteneur Linux léger qui exécute une commande spécifiée.
- Un accès facile à toutes les sorties, de chaque composant de l'application et chaque processus (dyno).

### 3.7.4 Le processus d'hébergement de l'application

#### a. Le tableau de bord de Heroku

Il faut d'abord créer un compte et s'y connecter. Après avoir créé un compte Heroku, vous trouverez le tableau de bord suivant montrant toute l'application hébergée que vous avez uploadée :



**Figure 4. 5 : le tableau de bord de Heroku**

### **b. Les Outils De Travail :**

Pour utiliser Heroku votre système doit être bien configuré avec les outils suivants:

- **Node et NPM :** Notre application est une application NodeJS donc nous avons besoin d'avoir ces deux outils afin d'exécuter notre application localement.
- **Heroku Toolbelt :** c'est la ligne de commande qui permet de vérifier notre application, créer une nouvelle application ... etc. (Il faut Ruby pour être installé dans notre machine de développement également).
- **Git Bash :** Cet outil permet d'envoyer notre application de notre machine local aux serveurs Cloud de Heroku.

### **c. Utilisation Heroku:**

Dans cette étape, nous allons connecter à notre compte à la ligne de commande, ce processus d'authentification dira à Heroku qu'on est autorisés déployer l'application.

```
C:\Users\Code
λ heroku login
Enter your Heroku credentials.
Email: houssem.yahiaoui.
Password (typing will be hidden):
Logged in as houssem.yahiaoui

C:\Users\Code
λ |
```

Figure 4. 6 : Heroku Login via CMD

#### d. Déploiement de l'application:

Pour déployer l'application, nous aurons besoin de faire ce qui suit:

- A. Créer un dépôt distant dans notre compte Heroku comme ce qui suit:

```
C:\Users\Code
λ heroku create
Heroku CLI submits usage information back to Heroku. If you would like to disable this, set `skip_analytics: true` in C:\Users\Code\AppData\Local\heroku\config.json
Creating app... done, frozen-waters-24789
https://frozen-waters-24789.herokuapp.com/ | https://git.heroku.com/frozen-waters-24789.git
```

Figure 4. 7 : Créer une nouvelle application Heroku

1. Saisie d'une nouvelle commande de terminal: **heroku create** : qui va créer une nouvelle application Heroku.
2. Après avoir créé l'application, **Heroku App** va retourner 3 informations:
  - a) **Nom de l'application** : frozen-waters-24789.
  - b) **Lien public** : <https://frozen-waters-24789.herokuapp.com/>
  - c) **Le Repository Git De L'application** (le répertoire de l'application).

Maintenant, si nous revenons à notre tableau de bord, nous allons trouver notre nouvelle application comme dans la figure ci-dessous :



Figure 4. 8 : Dashboard avec la nouvelle application

- B. Renommer l'application : afin de garder une bonne image à notre application, frozen-master ne sera pas un nom approprié, donc nous avons besoin de la renommer comme la figure suivante représente:

```
C:\Users\Code
λ heroku apps:rename injaz-djazair --app frozen-waters-24789
Renaming frozen-waters-24789 to injaz-djazair... done
https://injaz-djazair.herokuapp.com/ | https://git.heroku.com/injaz-djazair.git
! Don't forget to update git remotes for all other local checkouts of the app.
```

Figure 4. 9 : Renommer l'application

**Remarque :** tout changement dans notre machine de développement local sera reflétés dans notre compte hébergé.

- C. Maintenant, afin de faire tourner notre code directement sur le serveur, nous avons juste besoin de faire ce qui suit:

```
C:\Users\Code
λ git push heroku master|
```

Figure 4. 10 : Pusher notre code vers le serveur

Après avoir saisi la commande comme dans la figure ci-dessus, le code sera « pushé » directement à l'application Heroku.

- D. Définition de la commande de lancement de l'application: Généralement et par défaut Heroku va directement dans le fichier package.json la commande de lancement, elle est généralement situé dans la section de script de démarrage. Vous pouvez spécifier la

commande ou tout simplement créer un Procfile et ajouter simplement la commande.

**Remarque :** Procfile est un mécanisme pour déclarer ce que les commandes sont exécutées par les dynos de votre application sur la plate-forme Heroku, et c'est un fichier sans extension.

**Remarque :** Un fichier package.json contient des métadonnées concernant votre application ou module.

Comme nous l'avons vu, nous pouvons facilement créer et héberger tout type d'application à l'aide de la plate-forme Heroku et simplement en utilisant des outils tels que l'utilitaire Git nous pouvons pousser notre code directement sur le compte Git référentiel distant.

L'hébergement est généralement la dernière étape de tout processus de développement moderne en raison de sa simplicité, et Heroku rend les choses encore simples tout en ajoutant des add-ons et plugins.

## 3.8 Présentation de l'application

### 3.8.1 Page d'accueil :

La figure ci-dessous montre la page d'accueil de notre plate-forme Web :

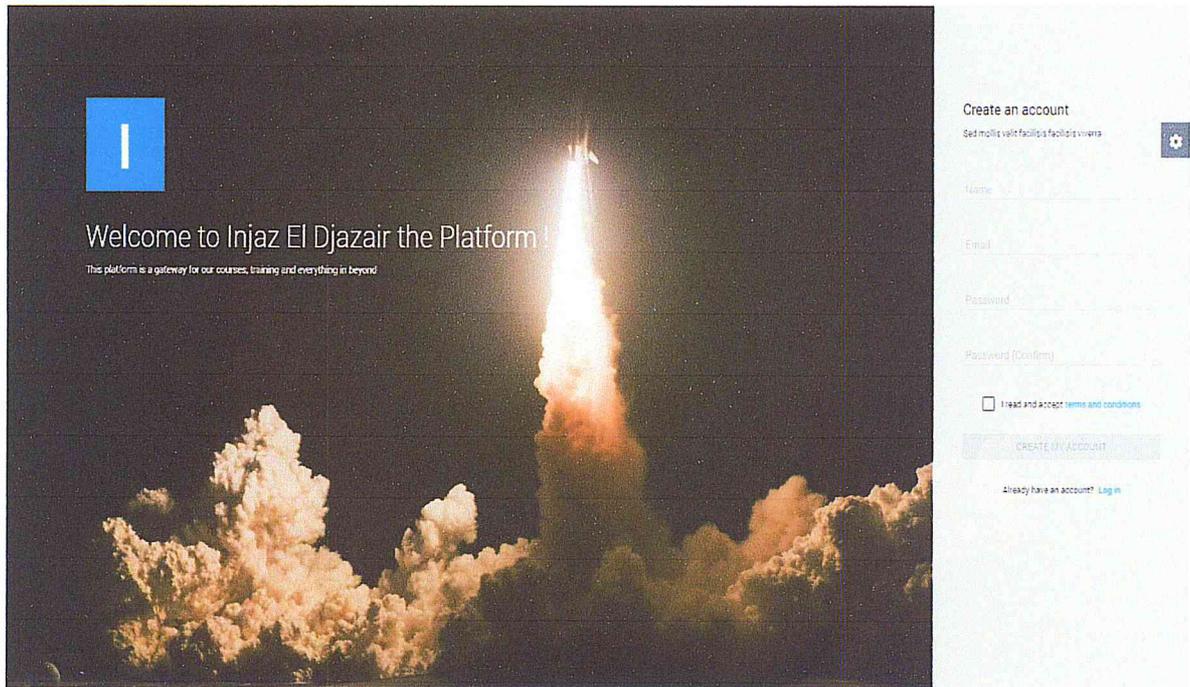


**Figure 4. 11 : La page d'accueil de M-Injaz**

Il est donc clair afin de se connecter à la plate-forme nous avons besoin d'être authentifié, dans notre application, nous vous offrons de nombreux mécanismes d'authentification:

- L'authentification normale en utilisant: Email et mot de passe.
- Authentification sociale utilisant Google+ et Facebook.

Donc, si l'utilisateur a un compte Facebook ou Google plus, il peut directement se connecter à son compte sans s'inscrire, sinon s'il préfère l'ancienne méthode qui consiste à s'inscrire dans sur la plate-forme en remplissant le formulaire, et la figure ci-dessous montre le processus d'inscription après avoir cliqué sur «Créer un nouveau compte»:



**Figure 4. 12 : Le formulaire d'enregistrement d'un nouveau profil**

Pour s'inscrire à l'ancienne, l'utilisateur doit entrer son nom et e-mail et bien sûr le mot de passe, puis le BackEnd vérifier l'e-mail et si le courriel est unique, l'utilisateur sera ajouté à la base de données et redirigé vers son profil directement, sinon l'opération sera interrompue et un message d'erreur sera affiché dans la page d'inscription.

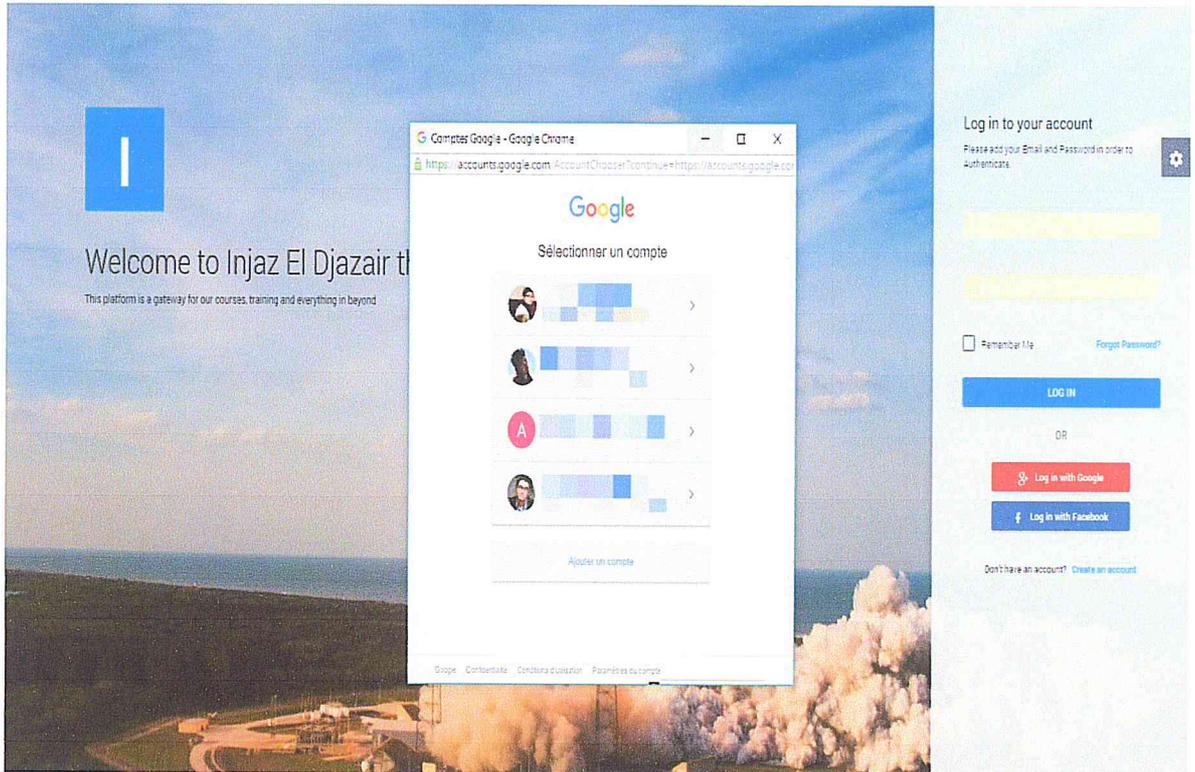
### **3.8.2 Les Pages de profil**

#### **a. Scénario d'utilisation :**

Nous allons simuler une connexion pour avoir un regard sur les différentes manières d'authentification et le profil de remplissage et nous allons faire ce qui suit:

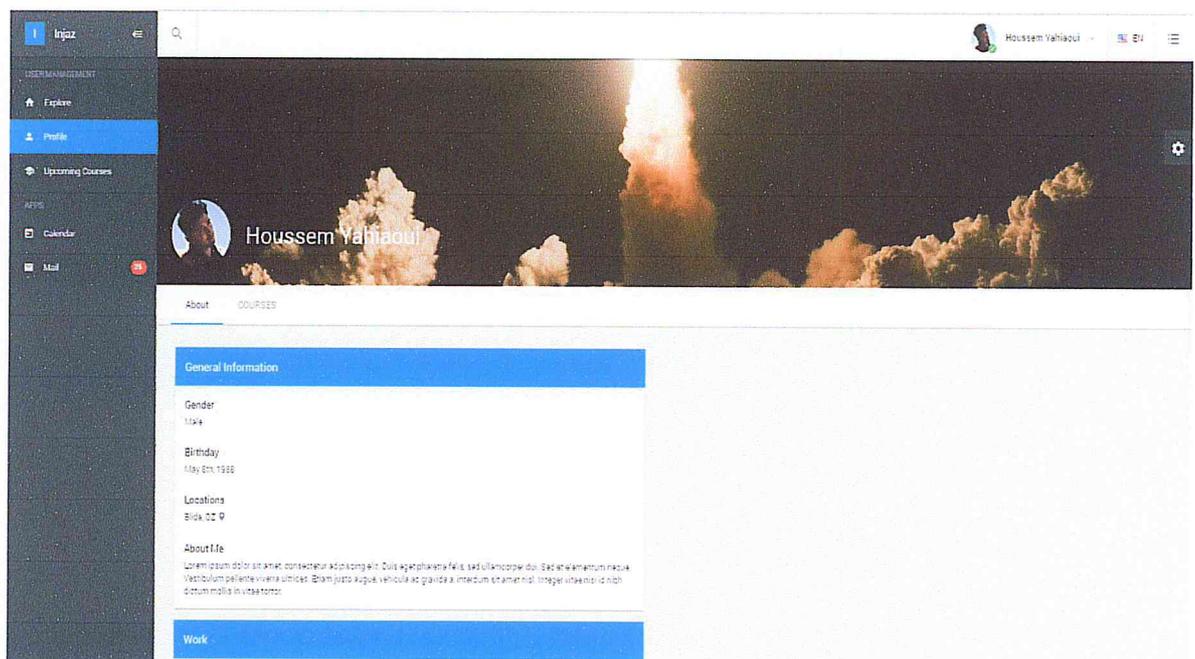
En utilisant l'authentification avec les réseaux sociaux :

Dans la page de connexion a choisi l'une des deux façons, via Google+ ou Facebook, dans ce scénario, nous utiliserons Google Plus.



**Figure 4. 13 : Enregistrement d'un nouveau profil à l'aide des réseaux sociaux**

Ensuite, on choisit simplement le compte Google avec qui nous voulons nous connecter, puis la plate-forme va nous rediriger vers notre page de profil, et la figure ci-dessous montre la page de profil :



**Figure 4. 14 : Page de profil d'un utilisateur**

Notez que notre profil sera directement créé et les différents champs seront complété en utilisant API de Google plus ou de Facebook, incluant la photo de profil et d'autres champs.

L'apprenant peut assister à l'événement et voir tout événement programmés dans le calendrier, la figure ci-dessous montre le calendrier du point de vue de l'apprenant:



Figure 4. 15 : Le calendrier des évènements vus par un apprenant

### 3.8.3 La page de profil du tuteur (Enseignant) :

En ce qui concerne le profil de l'enseignant, il a le même profil, mais comme une valeur ajoutée, il peut commencer un cours ou un flux en direct, et la figure ci-dessous montre le Tuteur profil:

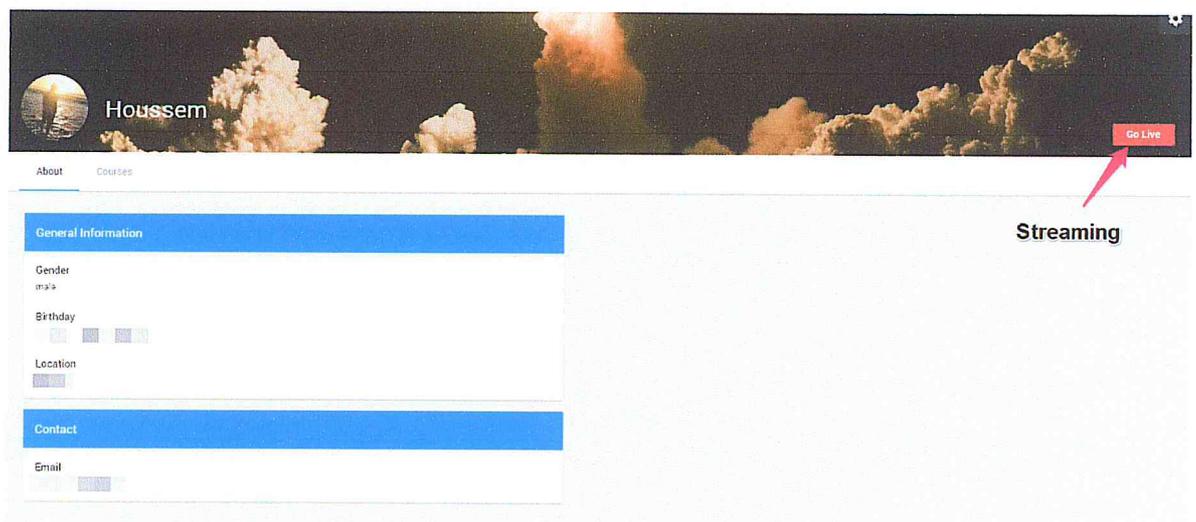


Figure 4. 16 : profil d'un enseignant

### a. Démarrage d'un cours:

Pour commencer un cours il faut cliquer sur le bouton "Go Live" dans le profil Tuteur alors nous allons avoir cette page:

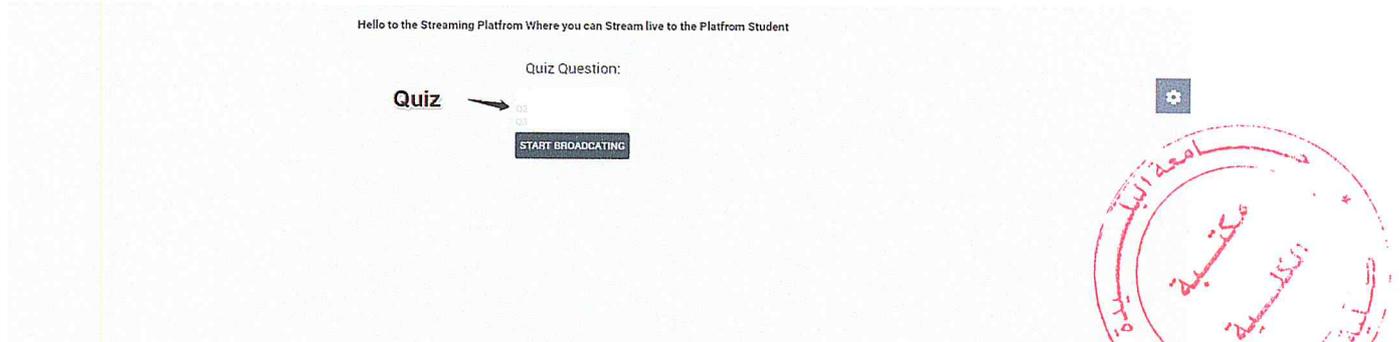


Figure 4. 17 Ajout Quiz Question avant streaming

Dans cette page, nous pouvons spécifier la question Quiz qui sera soumis aux apprenants après la diffusion. Ensuite, on clique sur le "Start Broadcasting" et il nous conduira vers la page « cast large » comme le montrent la figure ci-dessous:

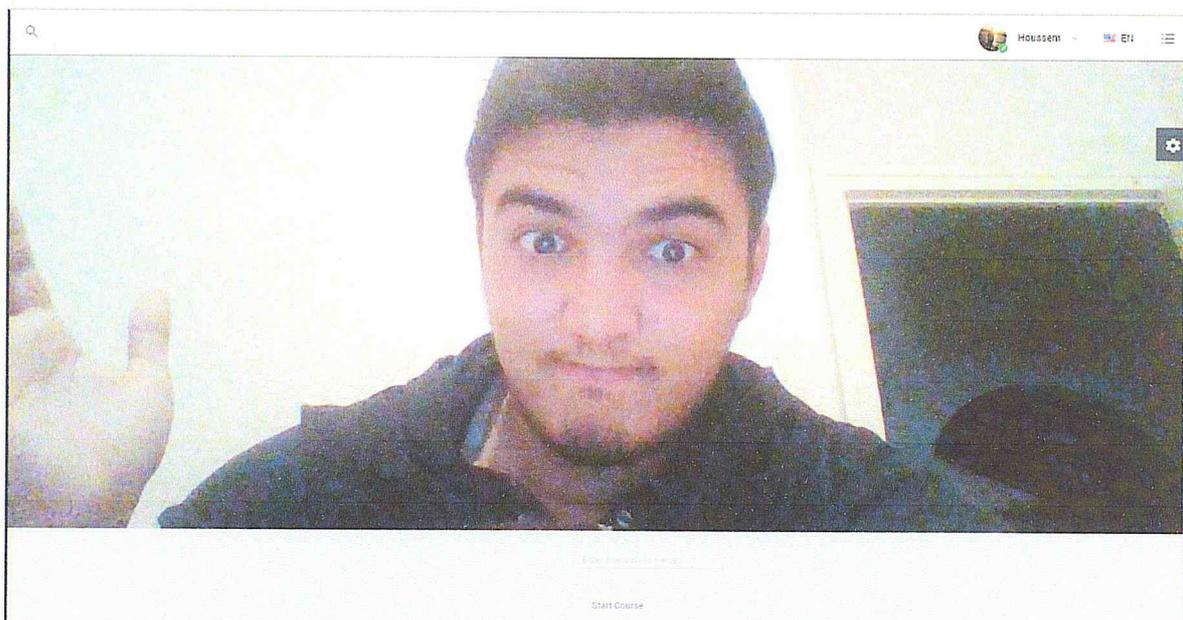


Figure 4. 18 : Commencement du streaming

### 3.8.4 La page de profil de l'Admin :

L'administrateur est le super-utilisateur dans notre plate-forme, l'administrateur a accès à tous les sous-pages, de profil, au calendrier, mais il a le tableau de bord en tant que valeur ajoutée, qui contient des statistiques et des mesures d'aide à la décision aussi l'ajout d'un prof et la figure ci-dessous le montre bien.

## a. Opérations d'administrateur

- **Ajout d'un Prof :** Pour ajouter un nouveau tuteur, l'administrateur système doit effectuer les opérations suivantes:

1 - Allez et cliquez sur cette section:

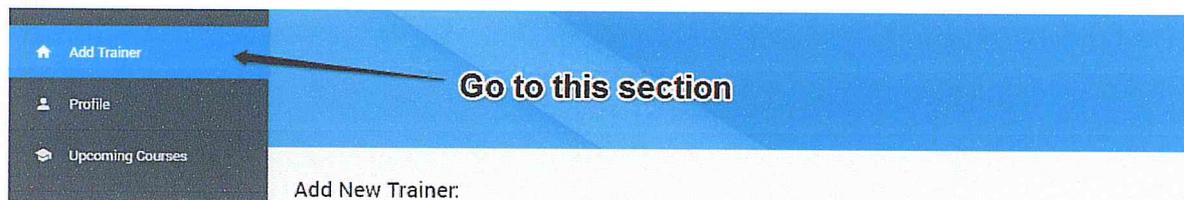


Figure 4. 19 la section ajout d'un enseignant

On va retrouver le formulaire suivant :

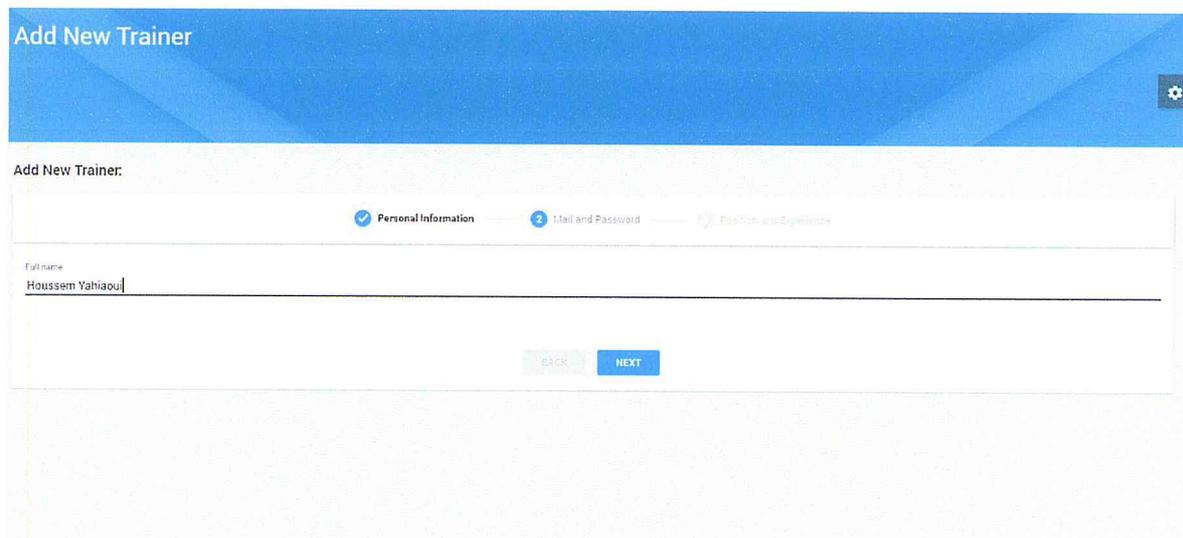


Figure 4. 20 : Formulaire d'ajout d'un enseignant

Il faut remplir le formulaire le nom le e-mail et le mot de passe.

**Add New Trainer**

Add New Trainer:

Personal Information   Mail and Password   Position and Experience

Email: housem. @gmail.com   Password: FRW13sh5bZP

**Trainer Mail**   **Auto Generated Secure Password**

BACK   NEXT

**Figure 4. 21 Remplissage de l'e-mail et génération du mot de passe**

On peut aussi ajouter l'expérience et la position :

**Add New Trainer**

Add New Trainer:

Personal Information   Mail and Password   Position and Experience

Position: Tutor   Experience: 5

**Tutor Position**   **Years of Experience**

BACK   SUBMIT

Click to Save

**Figure 4. 22: l'ajout de l'expérience et la position**

Après que vous appuyez sur le bouton Soumettre pour enregistrer les données fournies et de créer un nouveau tuteur.

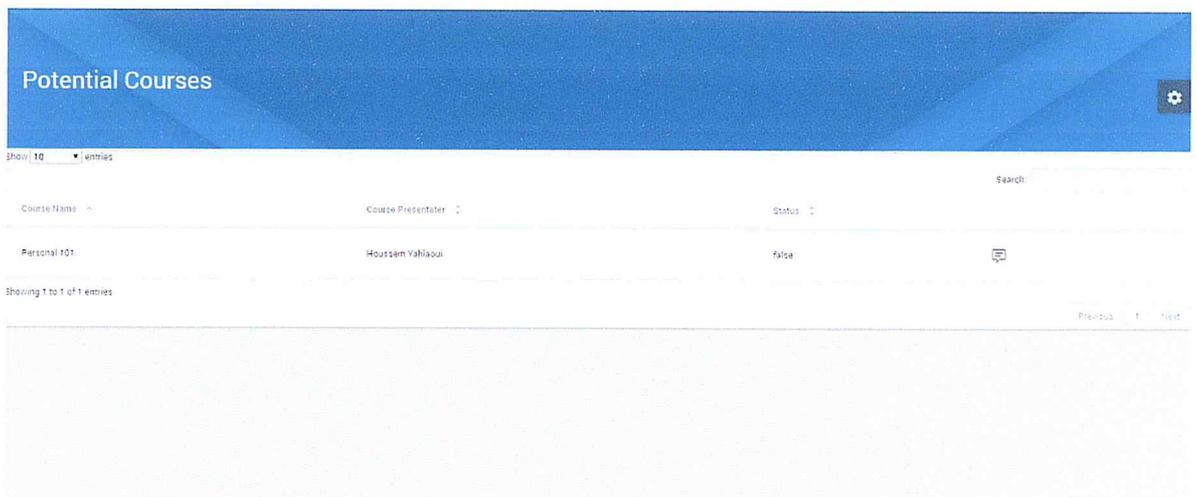
Une fois la création terminée une notification apparaîtra, comme le montrent la figure ci-dessous:



**Figure 4. 23 : ajout de l'enseignant réussi**

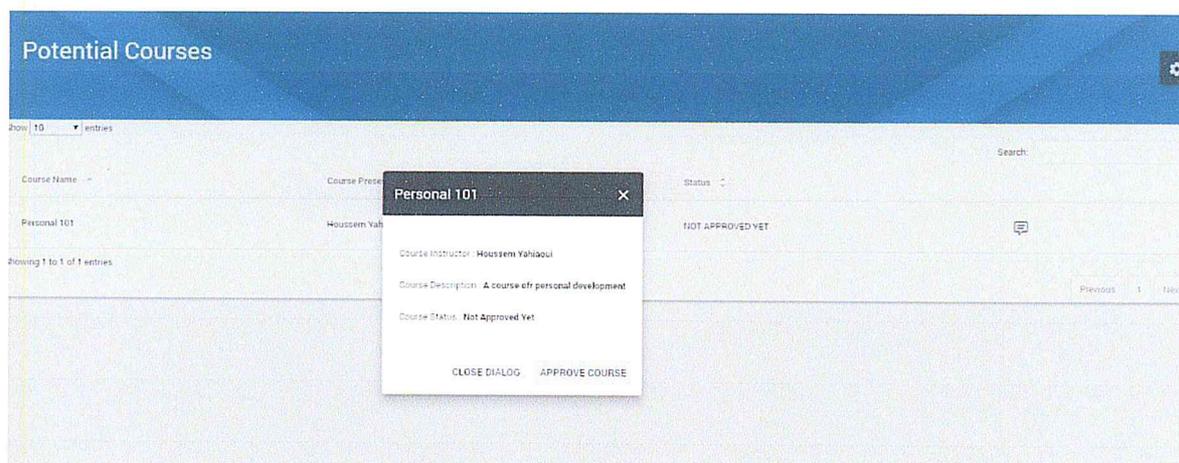
Un nouveau tuteur a été créé et e-mail lui sera envoyé avec Email et mot de passe.

- Valider les cours propose par les enseignants : Afin de valider un cours potentiel Admin aller directement à la page de cours potentiel comme le montrent la figure ci-dessous:



**Figure 4. 24: la liste des cours non validé**

Ensuite, en cliquant sur le bouton "message" nous aurons le dialogue suivant:



**Figure 4. 25 : Approuver un cours**

## 3.9 Conclusion

Dans ce chapitre nous avons présenté le côté technique, ainsi nous avons adopté le model d'architecture SOC, avec NodeJS pour l'implémentation du côté Back End, et puis MongoDB pour la base de données qui adhère au JSON. Pour la partie client le Framework angularJS pour mieux organiser les pages web, et puis le processus d'hébergement avec la Platform Heroku. Et enfin quelques captures du travail finale.

# Conclusion générale

Dans le cadre de ce travail nous avons essayé de réaliser un MOOC universel ou générique, qu'on estime être la meilleure solution pour palier aux problèmes posée par INJAZ ou encore toute institution non lucrative ayant les mêmes objectifs que Injaz qui est de diffuser, partager des cours gratuitement à un large publique que possible, avec un minimum de coût.

Ce travail a été mis en place en passant par trois grandes parties, la première partie a concernée la partie théorique, ou on a abordé la notion des MOOCs de manière générale avec comme principal intérêt de cette étude l'apport en plus de cette nouvelle méthode pédagogique ouverte à tous mais aussi les limites de cette solution afin de palier aux problèmes pouvant être posés, par la suite nous avons défini le Cloud Computing et donné ses différents aspects d'utilisation à savoir pour l'intégration des MOOCs.

En deuxième partie, nous avons fait une petite analyse des besoins avec une étude de la méthode existante déjà ; avant de tirer les objectifs à atteindre à travers la solution proposé dans partie conception ; qui a été modelé à l'aide des différents digrammes UML.

La dernière partie était consacrée à la réalisation de la plateforme M-Injaz où nous avons utilisé le langage principalement JavaScript pour privilégier l'optimisation du code et des performances et cela fait partie des avantage de ce langage, donc pour le coté serveur nous avons utilisé NodeJS, avec MongoDB le SGBD non relationnel orienté document et pour le coté client les pages ont été implémenté à l'aide du Framework AngularJS, et enfin pour la partie hébergement Heroku a été privilégier par rapport à d'autre Platform.

L'application finale qui a pour nom M-Injaz a sa première version pas mal de fonctionnalités :

- Une diffusion de cours en live avec un chat en ligne pour poussé l'interactivité entre les différents acteurs, et pour simuler au mieux l'échange qui peu y avoir dans une vraie classe.
- Une disponibilité de formations qui seront accessibles en différé.
- Une version mobile est a compté aussi avec une réalisation hybride pour une fonctionnalité dans tous les OS.
- Un système de feedback (teste) pour mieux voir si les objectifs du cours ont été atteints, et pour aider les enseignants à améliorer leurs cours.
- Un système de récolte des mots clés et des formations les plus recherchées pour mieux combler les besoins des apprenants, qui pour l'instant n'est pas très efficace mais qui

sera en développement dans le future.

## Bibliographie

A propos [En ligne] // Coursera. - 2016. - <https://fr.coursera.org/about/>.

A propos [En ligne] // ITyPA. - 2016. - <https://itypa.net/about/>.

About | Node.js [En ligne] // Nodejs.org. - 2016. - <https://nodejs.org/en/about/>.

About us [En ligne] // UDACITY. - 2016. - <https://www.udacity.com/us>.

About us | edX [En ligne] // edX. - 2016. - <https://www.edx.org/about-us>.

**Alexander McAuley Bonnie Stewart, George Siemens, Dave Cormier** THE MOOC MODEL FOR DIGITAL PRACTICE [Livre]. - [s.l.] : The University of Prince Edward, 2010.

**AZOUGH Samia** E-LEARNING ADAPTATIF : GESTION INTELLIGENTE DES RESSOURCES PEDAGOGIQUES ET ADAPTATION DE LA FORMATION AU PROFIL DE L'APPRENANT [Rapport]. - Rabat : These de doctorat, 2014.

**BORT Julie** Google has nabbed some huge customers for its most important new business – cloud computing [En ligne] // Business Insider. - 2015. - <http://www.businessinsider.com/google-names-huge-companies-using-its-cloud-2015-6>.

**BOUARAB Farida** Modelisation basée ontologies pour l'apprentissage interactif - Application à l'évaluation des connaissances de l'apprenant [Rapport]. - Tizi ouzou : [s.n.], 2010.

**CISEL Matthieu** les MOOCs [Rapport]. - Paris : [s.n.], 2014.

Cloud computing - Wikipedia, the free encyclopedia [En ligne] // Wikipédia . - 2016. - [https://en.wikipedia.org/wiki/Cloud\\_computing](https://en.wikipedia.org/wiki/Cloud_computing).

**Delpech Quentin et Diagne Marième** MOOC, l'âge de maturité ? Modèles économiques et évolutions pédagogiques [Livre]. - France : France Stratégie, 2016.

**DUPONT Simon** Gestion autonome de l'élasticité multi-couche [Rapport]. - Nantes : [s.n.], 2016.

**FARES Hatem** Les avantages et les inconvénients des MOOCS [En ligne] // eklablog. - 2013. - <http://ballaye.eklablog.com/les-avantages-et-les-inconvenients-des-moocs-a104688796>.

**Farida BOUARAB** Modelisation basée ontologies pour l'apprentissage interactif -

Application à l'évaluation des connaissances de l'apprenant [Rapport]. - Tizi ousou : [s.n.], 2010.

**Gilliot Jean-Marie** Les Cours en ligne ouverts et massifs, explication en français par Stephen Downes [Livre]. - 2012.

**GUENANE Fouad Amine** Gestion de la sécurité des réseaux à l'aide d'un service innovant de Cloud [Rapport]. - Paris : [s.n.], 2014.

**KARSENTI Thierry** MOOC révolution ou simple effet de mode [Livre]. - Monreal : RITPU, 2013.

**MELL Peter et GRANCE Timothy** The NIST Definition of Cloud [Rapport]. - Gaithersburg : [s.n.], 2011.

**NEBRA Mathieu** Des applications ultra-rapides avec Node.js [En ligne] // OpenClassrooms. - <https://openclassrooms.com/courses/des-applications-ultra-rapides-avec-node-js>.

Qui somme nous ? [En ligne] // OpenClassrooms. - 2016. - mars 2016. - <https://openclassrooms.com/about-us>.

**ROQUES Pascal** Le cahier du programmeur UML 2 Modéliser une application web [Livre]. - [s.l.] : EYROLLES, 2007.

**SARR Idrissa** Routage des Transactions dans les Bases de Données à Large [Rapport]. - Paris : [s.n.], 2010.

**YUEN Edwin** How Microsoft Defines Cloud Computing – Cloud Power IT Insights [En ligne] // Cloud Power IT Insights. - 2010. - <https://blogs.technet.microsoft.com/itinsights/2010/12/07/how-microsoft-defines-cloud-computing/>.

