

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de La Manouba
École Nationale des Sciences de l'Informatique



Rapport du Projet de Conception et de Développement

Sujet

CONCEPTION ET DÉVELOPPEMENT D'UNE APPLICATION D'ENSEIGNEMENT EN LIGNE

Réalisé par
Donia HAMMAMI
Firas KOOLI

Encadré par
Dr. Imtiez FLISS

Année Universitaire 2016/2017

Signature de l'encadrante



Remerciements

Nous tenons, avant de présenter notre travail, à exprimer notre grande reconnaissance envers les personnes qui nous ont, de près ou de loin, apporter leur soutien.

Nous exprimons notre parfaite reconnaissance et nos remerciements à notre encadrante Dr. Fliss Imtiez pour le temps qu'elle a bien voulu consacrer à l'encadrement et le suivi de ce travail ; les conseils qu'elle nous a prodigués après sa minutieuse lectures et pour les réunions qui ont rythmées les différentes étapes de la rédaction de ce rapport. Les discussions que nous avons tenues ont permis d'orienter ce travail d'une manière sûre et pertinente. Nous la remercions vivement pour son effort, sa disponibilité et surtout ses conseils qui ont largement contribué à rehausser la valeur de ce travail.

Que les membres de jury trouvent, ici, l'expression de nos remerciements pour l'honneur qu'ils nous font en acceptant de juger ce travail.

Table des matières

Introduction générale	1
1 Etude préalable	4
I Cadre général du projet	4
I.1 Contexte général	4
I.2 Problématique	4
II Etude de l'existant	5
II.1 Présentation des types d'application d'E-learning	5
II.2 Etude des solutions existantes	7
II.2.1 Présentation des solutions existantes	7
II.2.2 Critique des solutions existantes	11
III Solution proposée et travail demandé	14
Conclusion	14
2 Analyse et spécification des besoins	15
I Capture des Besoins	15
I.1 Définition des acteurs	15
I.2 Analyse des besoins	16
I.2.1 Besoins fonctionnels	16
I.2.2 Besoins non fonctionnels	17
II Spécification des besoins	17
II.1 Diagrammes de cas d'utilisation	17
II.1.1 Diagramme de cas d'utilisation relatif à l'administrateur	17
II.1.2 Diagramme de cas d'utilisation relatif à l'apprenant .	18
II.1.3 Diagramme de cas d'utilisation relatif au formateur .	20

TABLE DES MATIÈRES

II.2	Description de quelques scénarii	21
II.2.1	Scénario du cas d'utilisation "S'authentifier"	21
II.2.2	Scénario du cas d'utilisation "S'inscrire à un cours"	22
II.2.3	Scénario du cas d'utilisation "Télécharger cours"	23
II.2.4	Scénario du cas d'utilisation "Ajouter cours"	23
Conclusion		24
3	Conception	25
I	Conception architecturale de l'application	25
I.1	Architecture physique	25
I.1.1	Description des architectures Types	25
I.1.2	Choix de l'architecture de l'application	27
I.1.3	Modélisation de l'architecture physique	29
I.2	Architecture logique	30
I.2.1	Description des architectures types	30
I.2.2	Choix de l'architecture de l'application	32
II	Conception détaillée	34
II.1	Conception du modèle	34
II.2	Conception de la Vue	36
III	Description de quelques scénarii	38
III.1	Scénario d'authentification d'un apprenant	38
III.2	Scénario demande d'inscription à un cours	40
III.3	Scénario demande de téléchargement d'un cours	40
III.4	Scénario demande d'ajout du cours	41
Conclusion		42
4	Réalisation	43
I	Environnements de travail	43
I.1	Environnements de développement matériel	43
I.2	Environnements de développement logiciel	44
I.2.1	Plateforme JAVA EE	44
I.2.2	Le framework Struts 2 pour l'implémentation de la couche présentation	44

TABLE DES MATIÈRES

I.2.3	Hibernate pour l'implémentation de la couche d'accès au données	45
I.2.4	HTML5, CSS3, jQuery et JavaScript pour la construc- tion des interfaces riches	45
II	Interfaces Homme-Machine	46
III	Chronogramme	54
	Conclusion	55
	Conclusion générale	56
	Netographie	58

Table des figures

1.1	Page d'accueil de Open Classroom [N15]	8
1.2	Page d'accueil de EdX [N17]	9
1.3	Page d'accueil de Coursera [N19]	10
1.4	Page d'accueil de FUN [N21]	11
2.1	Cas d'utilisation relatif à l'administrateur	18
2.2	Cas d'utilisation relatif à un apprenant	19
2.3	Cas d'utilisation relatif à un formateur	20
2.4	Diagramme de séquence système d'authentification	21
2.5	Diagramme de séquence système d'inscription à un cours	22
2.6	Diagramme de séquence système de téléchargement d'un cours	23
2.7	Diagramme de séquence système d'ajout d'un cours	24
3.1	Architecture 2-tiers [N23]	26
3.2	Architecture 3-tiers [N24]	26
3.3	Architecture N-tiers [N26]	27
3.4	Architecture trois tiers [N27]	28
3.5	Diagramme de déploiement du système	29
3.6	Architecture MVP [N28]	31
3.7	Architecture MVVM [N30]	31
3.8	Architecture MVC [N32]	32
3.9	Architecture de Struts2 [N34]	33
3.10	Diagramme de classe de la couche modèle	35
3.11	Diagramme de classe de la couche vue	37
3.12	Diagramme de séquence du scénario d'authentification	39
3.13	Diagramme de séquence du scénario d'inscription à un cours	40

TABLE DES FIGURES

3.14 Diagramme de séquence du scénario de téléchargement du cours	41
3.15 Diagramme de séquence du scénario d'ajout du cours	42
4.1 Page d'accueil	46
4.2 Liste des cours sur la plateforme	47
4.3 FAQ	47
4.4 Page de contact	48
4.5 Blog	48
4.6 Description de la plateforme	49
4.7 Page d'authentification de l'administrateur	49
4.8 Page de profil de l'administrateur	50
4.9 Espace administrateur	50
4.10 Page de profil d'un formateur	51
4.11 Dashboard d'un formateur	51
4.12 Page d'inscription d'un apprenant	52
4.13 Profil d'un apprenant	52
4.14 Inscription aux cours	53
4.15 Dashboard d'un apprenant	53
4.16 Quiz	54
4.17 Diagramme de Gantt réel	54

Liste des tableaux

1.1	Tableau comparatif des solutions existantes	12
3.1	Les principales classes conçues pour l'élaboration de ce modèle	36
3.2	Les principales classes conçues pour l'élaboration de cette couche	38

Introduction générale

L'enseignement est un mode d'éducation permettant de développer les connaissances d'un élève par le biais de la communication verbale et écrite. De nombreux observateurs avertissent que les structures de l'enseignement traditionnel n'ont guère changé depuis le début du XIXème siècle et ce, en dépit des bouleversements sociaux et de l'évolution des moyens de communication.

L'enseignement traditionnel est centré sur le cours magistral. Les auditeurs sont passifs, souvent intellectuellement absents du fait des conditions difficiles imposées par la surcharge horaire. Les principaux moyens mis en oeuvre dans ce type d'enseignement : un cours magistral associé à des travaux dirigés et pratiques, le tableau noir et le support papier.

Par ailleurs, les systèmes traditionnels d'enseignement imposent à tous les apprenants une unité de lieu, une unité de temps, une unité d'action, une unité de rythme ce qui implique une rigidité des mécanismes et une difficulté d'adéquation avec la réalité quotidienne.

La tendance à l'amélioration du système sur le plan pédagogique par le recours aux moyens audiovisuels classiques (projections de diapositives, de transparents, séquences vidéos) n'a pas résolu le problème. En effet, le formateur doit à la fois exposer le cours et entreprendre des manipulations techniques assez difficiles.

Pour cela, les modalités d'apprentissage ont évolué. Cette évolution correspond à un besoin induit par la transformation de nos sociétés et à une attente nouvelle de nos étudiants. Elle s'appuie par ailleurs sur de nouveaux lieux et de nouveaux outils d'apprentissage. Depuis plusieurs années, les technologies de l'information et de la communication (TIC) constituent non seulement un nouvel outil, un nouveau média, mais aussi un moyen d'ouverture sur des ressources du monde entier.

Donc, l'enseignement moderne exige des moyens pour faciliter l'apprentissage

de l'apprenant tels que le choix de situation, l'explicitation d'objectifs et des critères, les choix de contenus, la mise en oeuvre de procédures d'éducation, l'élaboration d'outils pour aider l'élève à construire et s'approprier des savoirs et des savoir-faire.

L'e-learning est un processus d'apprentissage à distance s'appuyant sur des ressources multimédias, qui permet à une ou plusieurs personnes de se former à partir de leurs ordinateurs. Les supports multimédias utilisés peuvent combiner du texte, des graphismes en deux ou trois dimensions, du son, de l'image, de l'animation et même de la vidéo.

Ces supports permettent de révolutionner l'approche pédagogique, d'employer des méthodes plus ludiques où l'interactivité joue un grand rôle, de diversifier les outils employés, de s'adapter davantage au processus d'apprentissage de l'apprenant, qui devient le pilote de sa formation. Ce dernier pourra se former à son rythme, en fonction de ses besoins et de ses disponibilités, ce qui est particulièrement important à une époque où la formation se décline tout au long d'une vie.

Plusieurs solutions de formations à distance existent sur le marché. Ces solutions proposent les différentes fonctionnalités de base (téléchargement des supports de cours, les quiz, la communication...) Cependant, elles possèdent plusieurs inconvénients comme la lenteur au niveau du temps de réponse du système, le nombre limité des apprenants, les cours sont payants la plupart du temps.

C'est dans ce cadre que s'inscrit notre projet de développement et de conception qui a repéré l'importance de réaliser une application d'E-learning. Notre travail consiste à concevoir et développer une plateforme d'enseignement en ligne qui réunit les différentes fonctionnalités nécessaires à une plateforme d'apprentissage en ligne et confronte les inconvénients des solutions existants.

Le présent rapport décrit les différentes étapes de notre travail, et il s'articule autour de quatre chapitres :

le premier chapitre est consacré à présenter le cadre général du projet en mettant l'accent sur la problématique suivie d'une étude de l'existant. LA solution proposée et le travail demandé seront ensuite exposés.

Le deuxième chapitre présente en premier lieu une analyse détaillée des besoins fonctionnels et non fonctionnels de notre application. En second lieu, il décrit les différents cas d'utilisation.

Le troisième chapitre détaille l'approche conceptuelle que nous avons adoptée pour mettre en place notre plateforme d'enseignement en ligne.

Le quatrième chapitre illustre nos choix technologiques et expose les résultats obtenus à partir de quelques interfaces homme-machine.

Nous clôturons par une conclusion générale qui présente une récapitulation du travail réalisé et présente un nombre de perspectives.

Chapitre 1

Etude préalable

Dans ce premier chapitre, nous allons nous intéresser tout d'abord au contexte général de notre sujet ainsi qu'à la problématique. Ensuite, nous élaborons une étude approfondie des MOOCs qui existent en mettant l'accent sur leurs insuffisances. Pour finir nous présentons la solution adoptée pour résoudre ce problème ainsi que le travail demandé.

I Cadre général du projet

I.1 Contexte général

L'apprentissage en ligne est une modalité pédagogique et technologique qui a d'abord concerné la formation continue, l'enseignement supérieur puis la formation en entreprise, c'est-à-dire au service d'un apprenant adulte ayant une certaine autonomie dans l'organisation de son processus d'apprentissage.

L'e-learning [N1], désigne donc l'ensemble des solutions et moyens permettant l'apprentissage par des moyens électroniques. La formation en ligne inclut de cette façon des sites web éducatifs [N2], la téléformation [N3], l'enseignement télématicque [N4], ou encore l'e-training[N5], notamment.

I.2 Problématique

Il y a quelques années, de nombreux formateurs plaçaient leurs espoirs dans l'apprentissage en ligne. Pour certains, l'e-Learning allait révolutionner l'apprentissage.

Force est de constater que ces premières tentatives n'ont pas rencontré le succès escompté. La tentation est parfois forte pour le formateur de décharger dans les modules en ligne des tonnes de contenu. Si l'information est trop dense ou mal structurée, on court le risque de court-circuiter et de rendre caduc l'apprentissage des participants.

Avec l'avènement et le progrès constant des nouvelles technologies, on ne compte plus le nombre d'outils à disposition des formateurs pour créer des dispositifs E-Learning (plate-formes LMS [N6], médias sociaux [N7], podcasts [N8]... entre autres). Difficile donc parfois de s'y retrouver et d'avoir la garantie que ces ressources tiendront dans la durée.

Enfin, les plateformes d'e-learning sont la plupart du temps payantes. Ainsi un apprenant doit payer pour accéder au cours et quiz ce qui amoindrit tous les avantages du e-learning.

Il serait donc intéressant de proposer une solution d'e-learning adaptée aux nouvelles tendances et aux attentes des utilisateurs. Dans ce cadre il crucial d'étudier les solutions existantes sur le marché.

II Etude de l'existant

Avant de présenter les solutions existantes, il est primordial de connaître les différents types d'applications d'E-learning

II.1 Présentation des types d'application d'E-learning

- **Le cours en ligne fermé [N9]**, distribué par un organisme de formation ou un établissement d'enseignement, destiné à un groupe d'apprenants régulièrement inscrits et ayant donc acquitté des droits d'inscription, distribué sur une plate-forme (Learning Management System ou LMS), dispensé seul ou dans le cadre d'un parcours de formation, diplômant ou pas. Ce cours est généralement animé par un enseignant ou un tuteur qui assure la communication avec les participants et peut aussi animer des temps de formation en direct (appelés "synchrone"). Il comprend des ressources de contenus (les cours), des activités d'apprentissage (des exercices à faire, des épreuves d'évaluation...) et un espace d'interaction (généralement un forum) qui permet aux participants d'interagir entre eux et avec

les animateurs du cours.

- **Le cours en ligne ouvert [N10]**, c'est un cours non diplômant, distribué par un organisme de formation ou un établissement d'enseignement, destiné à tous ceux qui veulent s'autoformer sur un sujet qui les intéresse, généralement distribué sur une plateforme. Ce cours n'est pas tutoré, l'apprenant doit suivre son parcours seul. A côté des ressources de "cours" proprement dites, on trouve dans ces cours quelques exercices à correction automatique telle que des quiz, qui permettent à l'apprenant d'évaluer sa compréhension. Ces cours ouverts ne comprennent généralement pas d'espace d'interaction, puisqu'ils ne sont pas suivis par des groupes constitués, mais par des personnes qui les suivent à titre individuel, quand bon leur semble. Ils ne comprennent pas non plus de temps de formation synchrones.
- **Le MOOC [N11]** : (Massive Online Open Course), c'est une formation en ligne en libre accès sur Internet. Un MOOC est un cours qui s'étale sur plusieurs semaines et met à disposition un contenu pédagogique avec des vidéos, des quiz et se termine par l'obtention d'un certificat. Concrètement, un MOOC, c'est un contenu de formation, une communauté d'apprenants animée par un ou des professionnels, une session avec un contenu habituellement gratuit de cours et une certification. Le MOOC tourne autour de l'idée de co-construction des savoirs et du savoir-faire.
- **Le COOC [N12]** : C'est un cours en ligne dispensée par une entreprise auprès de ses clients ou de ses salariés. COOC signifie Corporate Online Open Course. Les cours s'adressent donc à des employés, des salariés, des cadres mais aussi à des candidats potentiels ou encore aux fournisseurs, clients, prospects en lien avec l'entreprise formatrice. Cette méthode permet aux apprenants de bénéficier de toutes les modalités pédagogiques : travail collaboratif, travail à distance, etc. Un outil puissant pour accompagner la transformation digitale au sein d'une organisation et pour développer les compétences de chaque collaborateur.

- **Le SPOC [N13]** : Acronyme pour Small Private Online Courses, c'est une formation en ligne pour un nombre restreint de personnes avec un nombre d'accès volontairement limité. On compare souvent le SPOC à un stage de formation. Idéal pour répondre à un besoin spécifique d'une entreprise. D'autres types de cours sont dernièrement apparus sur le marché des formations en ligne comme le SOOC (Small Online Open Courses) dont le contenu est susceptible d'intéresser qu'une partie des salariés, le xMOOC qui est un MOOC axé sur la création et la mise en ligne de contenus pédagogiques sans interactivité.

II.2 Etude des solutions existantes

D'après l'étude des types d'application d'e-learning, nous nous intéressons dans ce travail au MOOC, il est donc crucial d'étudier les solutions des MOOC les plus utilisées.

II.2.1 Présentation des solutions existantes

Parmi les solutions existantes nous trouvons :

1. **Open classroom [N14]** : C'est une école en ligne. Chaque visiteur peut à la fois être un lecteur ou un rédacteur. Les cours peuvent être réalisés aussi bien par des membres, par l'équipe du site, ou éventuellement par des professeurs d'universités ou de grandes écoles partenaires. Initialement orientée autour de la programmation informatique, la plate-forme couvre depuis 2013 des thématiques plus larges telles que le marketing, l'entrepreneuriat et les sciences.

CHAPITRE 1. ETUDE PRÉALABLE

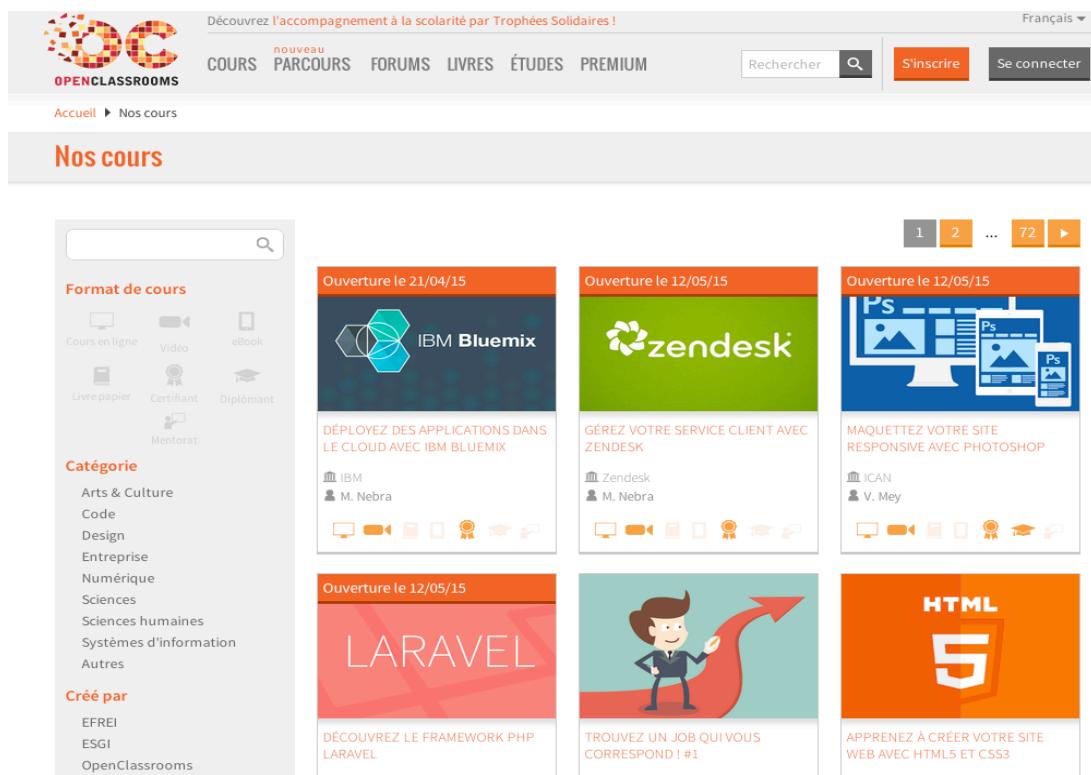


FIGURE 1.1 – Page d'accueil de Open Classroom [N15]

Créé en 1999 sous le nom de site du Zéro, ce site se forme essentiellement sur la base de contributions de bénévoles proposant des tutoriels vulgarisés avec un ton léger portant sur des sujets informatiques divers. À la suite du succès et de la fin des études des gérants, l'entreprise Simple IT, renommée ensuite Open Classrooms, est fondée dans le but de pérenniser le site. Open Classrooms dont la page d'accueil est donnée dans la figure 1.1 base son business model sur la délivrance de certifications payantes et propose un abonnement pour être suivi par un mentor.

-Technologies de développement : Depuis ses débuts, le site est développé en PHP. Il est refondu pour utiliser le framework Symfony lors du lancement de la version 4. Après avoir initialement utilisé une base de données MySQL, l'équipe décide de migrer en 2005 vers une base de données PostgreSQL qu'elle utilise encore à ce jour. De la même façon, après avoir initialement utilisé le gestionnaire de version SVN, l'équipe passe à Git courant 2008.

2. **EdX [N16] :** C'est une plateforme d'apprentissage en ligne (dite FLOT ou

MOOC). Elle héberge et met gratuitement à disposition des cours en ligne de niveau universitaire à travers le monde entier. Elle mène également des recherches sur l'apprentissage en ligne et la façon dont les utilisateurs utilisent celle-ci. Elle est à but non lucratif et la plateforme utilise un logiciel open source.



FIGURE 1.2 – Page d'accueil de EdX [N17]

EdX dont la page d'accueil est donnée par la figure 1.2 a été fondée par le Massachusetts Institute of Technology et par l'université Harvard en mai 2012. En 2014, environ 50 écoles, associations et organisations internationales offrent ou projettent d'offrir des cours sur EdX. En juillet 2014, elle avait plus de 2,5 millions d'utilisateurs suivants plus de 200 cours en ligne.

Les deux universités américaines qui financent la plateforme ont investi 60 millions USD dans son développement. La plateforme France Université Numérique utilise la technologie openedX, supportée par Google.

-Technologies de développement : Codée essentiellement en Python avec un peu en Ruby, la plateforme EdX utilise Django, node.js pour la programmation événementielle et MongoDB, la perle des bases de données NoSQL.

3. **Coursera [N18] :** C'est une entreprise numérique proposant des formations en ligne ouvertes à tous fondée par les professeurs d'informatique Andrew Ng et Daphne Koller de l'université Stanford, située à Mountain View, Californie. En

avril 2012, Coursera a annoncé avoir reçu 16 millions de dollars de plusieurs capital-risqueurs de série A comme John L. Doerr et Scott Sandell. Coursera dit se consacrer à mettre le meilleur enseignement au monde gratuitement à la disposition de toute personne qui le recherche. "Coursera ne génère actuellement aucun revenu mais John Doerr estime que les utilisateurs paieront pour un "service premium".



FIGURE 1.3 – Page d'accueil de Coursera [N19]

Le 16 août 2012, 1 080 000 étudiants de 196 pays s'étaient inscrits à au moins un cours. Coursera dont la page d'accueil est donnée par la figure 1.3 a été lancée peu après Udacity, entreprise par l'ancien professeur de Stanford Sebastian Thrun, et peu avant edX, une initiative à but non lucratif de formation en ligne par le MIT et Harvard.

-Technologie de développement : La plateforme s'exécute sur le serveur web nginx du système d'exploitation Linux sur la plateforme Amazon Web Services. Les données sont stockées dans Amazon S3 et la recherche de sites est gérée par Cloud Search qui indexe plus de 4,3 millions de documents sur le site. Chaque mois, les serveurs de base de données de Coursera (fonctionnant sur RDS) répondent à 10 milliards de requêtes SQL et Coursera dessert environ 500 To de trafic par mois. Coursera utilise le protocole OAuth2 pour l'authentification des

utilisateurs et le protocole LTI 1.1 pour l'interaction avec les cours.

4. **FUN [N20]** : C'est une plateforme de MOOC mise à disposition des établissements de l'enseignement supérieur français et de leurs partenaires académiques dans le monde entier. Lancée par le ministère de l'enseignement supérieur et de la recherche en octobre 2013, cette initiative vise à fédérer les projets des universités et écoles françaises, et permettre à tous les publics d'accéder à des cours variés et de qualité.

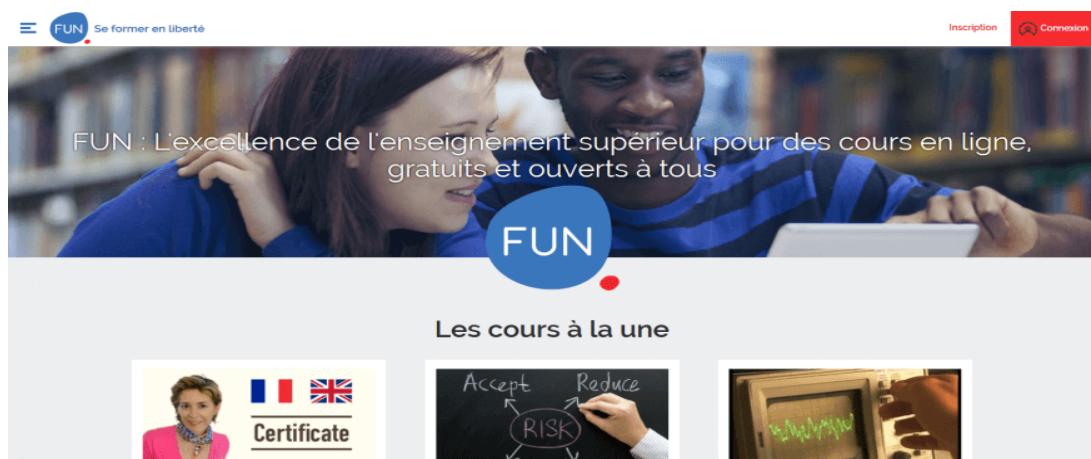


FIGURE 1.4 – Page d'accueil de FUN [N21]

Tous les cours présents sur FUN, dont la page d'accueil est donnée par la figure 1.4 sont conçus par des professeurs d'universités et écoles françaises et leurs partenaires académiques internationaux. Les étudiants et les internautes peuvent suivre ces cours de manière interactive et collaborative, à leur rythme.

Le principe est de suivre par Internet un cours dispensé par un professeur face caméra, l'apprenant peut à sa guise piocher les cours qui l'intéressent et cela à son rythme. Le cours est tout de même cadencé à un rythme hebdomadaire et validé par étapes à travers des quiz et autres types d'exercices.

II.2.2 Critique des solutions existantes

Afin d'étudier de plus près les solutions présentées, nous évaluons dans la table 1.1 ces applications selon un nombre de critères :

TABLE 1.1 – Tableau comparatif des solutions existantes

Fonctionnalités\Plateformes	Coursera	EdX	Open Classrooms	FUN
Gestion des apprenants	Oui	Oui	Oui	Oui
Gestion des formateurs	Non	Oui	Oui	Oui
Auto-inscriptions des apprenants	Les étudiants peuvent s'enregistrer eux-mêmes aux cours.	Les étudiants peuvent s'enregistrer eux-mêmes aux cours.	Non	Les étudiants peuvent s'enregistrer eux-mêmes aux cours.
Gestion des inscriptions aux cours	Oui	Oui	Non	Oui
Gestion de la visibilité des cours	Privé / public	Privé / public	Public	Public
Rendu de travaux par les apprenants	Oui	Oui	Oui	Oui
Gestion des notes	Automatique	Automatique	Pas de notion de note, que vrai ou faux par question	Automatique
Types de documents hébergés	Pdf, vidéo, word	Pdf, vidéo, word	Pdf, vidéo	Pdf, vidéo
Dashboard	Accès à la liste des cours achevés ou	Accès à la liste des cours	Non	Oui

CHAPITRE 1. ETUDE PRÉALABLE

	demandés	achevés ou demandés		
Blog	Oui	Oui	Oui	Oui
Messagerie	Oui	Oui	Non	Oui
Sondage	Oui	Oui	Non	Non
Partage de documents et des ressources	Oui	Oui	Nécessité d'avoir un compte premium pour accéder à tous les vidéos et pour télécharger le support du cours	Oui
Forum	Un forum pour chaque cours	Un forum pour chaque cours	Un forum pour tous les cours	Oui
Wiki	oui	Oui	Non	Non
Certificats	Oui	Oui	Oui	Oui
Nombre des sites utilisateurs	2500000	1080000	2212009	1000
Version Android	Oui	Oui	Non	Non

III Solution proposée et travail demandé

Après une étude approfondie et comparative sur les différentes solutions existantes, il est donc primordial au regard des limites des MOOCs présents de manière générale, de proposer une solution qui pourra répondre aux besoins des étudiants et en particulier les étudiants de l'ENSI. Notre projet de conception et de développement consiste à concevoir et à développer une plateforme d'E-learning, gratuite, présentant plusieurs services de formation et de collaboration accessibles à distance. Ceux qui s'y connectent pourront y trouver des cours sous forme de vidéos, PDF, PPT ainsi que des exercices et des espaces d'échanges afin de rendre la procédure d'apprentissage plus interactive. Ce MOOC est gratuit, il n'a pas de frais d'inscription sur la plateforme ni aux cours. Il est ouvert à toute personne intéressée. L'apprentissage arrive de manière informelle chez l'étudiant. Les seuls éléments nécessaires pour participer à ce MOOC sont un ordinateur et une connexion internet.

La plateforme de MOOC à mettre en place permet à l'utilisateur de s'inscrire à un ou plusieurs cours, de suivre le cours, de réviser et de passer des tests réguliers (journaliers, hebdomadaires) et évaluer son niveau quant aux différents détails du cours. Une fois, le cours terminé, l'utilisateur peut l'évaluer et donner des recommandations d'amélioration.

Conclusion

Dans ce chapitre, nous avons présenté le contexte général du projet suivi d'une étude approfondie de l'existant et de critique des solutions présentes. Ceci nous a permis de comprendre les besoins et d'envisager la solution la plus adéquate aux attentes des étudiants.

Le prochain chapitre est consacré à la présentation des besoins fonctionnels et non fonctionnels. Nous terminons par une spécification de ces besoins en nous basant sur les diagrammes d'UML.

Chapitre 2

Analyse et spécification des besoins

L'application qu'on se propose de développer doit tenir en compte des exigences des différents utilisateurs. Une étude des besoins de ces acteurs est alors nécessaire. Le présent chapitre s'articule autour de deux principaux volets : un premier où nous exposerons les différents besoins fonctionnels et non fonctionnels des différents acteurs. Un deuxième volet concerne la spécification de ces besoins via les diagrammes de cas d'utilisation.

I Capture des Besoins

L'étape de l'analyse des besoins est très importante puisque la réussite de toute application dépend de la qualité de son étude. Il faut donc bien déterminer les fonctions attendues par le système.

I.1 Définition des acteurs

Avant d'analyser les besoins, nous avons identifié trois acteurs : les apprenants, les formateurs et les administrateurs.

- L'apprenant : Il consulte les cours et s'y inscrit, communique avec les formateurs et les autres apprenants. Il peut aussi passer des évaluations.
- Le formateur : Il héberge des documents et des quiz. Il est à la disposition des apprenants afin de répondre à leurs questions.

- L'administrateur : Son rôle principal est de préparer les espaces des cours et des quiz. Il est aussi responsable de la gestion des comptes des formateurs et des apprenants.

I.2 Analyse des besoins

Les besoins sont divisés en deux catégories, à savoir les besoins fonctionnels et les besoins non fonctionnels.

I.2.1 Besoins fonctionnels

Ce sont les actions et les réactions que le système doit faire suite à une demande d'un acteur principal. Tenant compte de la nature de l'application, on distingue les besoins par acteurs :

- **Formateur** : L'application doit permettre au formateur de :
 - Héberger des cours, ajouter des liens utiles, proposer des quiz.
 - Consulter la liste des cours.
 - Communiquer avec les apprenants.
- **Administrateur** : L'application doit permettre à l'administrateur de :
 - Gérer les comptes des apprenants et des formateurs.
 - Préparer les espaces et les sessions des cours.
 - Gérer le forum de discussions.
- **Apprenant** : L'application doit permettre à l'apprenant de :
 - Consulter la liste des cours.
 - Rechercher des cours.
 - S'inscrire à un ou plusieurs cours.
 - Télécharger les documents relatifs aux cours.
 - Suivre son niveau de progression dans chaque cours.
 - Passer des quiz.
 - Communiquer avec les formateurs.

I.2.2 Besoins non fonctionnels

Les besoins non fonctionnels correspondent à la manipulation de l'application et précisent l'environnement de l'application.

- **L'extensibilité :** L'architecture de l'application permettra l'évolution et la maintenance (ajout ou suppression ou mise à jour) au niveau de ses différents modules d'une manière flexible.
- **La sécurité :** L'accès aux informations n'est possible qu'après vérification des priviléges et des droits d'accès. Ainsi tout utilisateur passera par une phase d'authentification pour pouvoir consulter les services offerts par l'application.
- **L'ergonomie et la convivialité :** L'application fournira une interface conviviale et simple à utiliser et qui ne requiert aucun prérequis, donc elle pourra être exploitable par tout type d'utilisateurs (même les non informaticiens).

II Spécification des besoins

Pour la spécification des besoins, nous nous référerons aux diagrammes d'UML : les diagrammes de cas d'utilisation et les diagrammes de séquence.

II.1 Diagrammes de cas d'utilisation

Dans cette partie, nous présentons les diagrammes de cas d'utilisation principaux par acteur. Cette phase représente la vue fonctionnelle de notre application.

II.1.1 Diagramme de cas d'utilisation relatif à l'administrateur

La figure 2.1 ci-dessous représente le diagramme de cas d'utilisation relatif à l'administrateur de notre application.

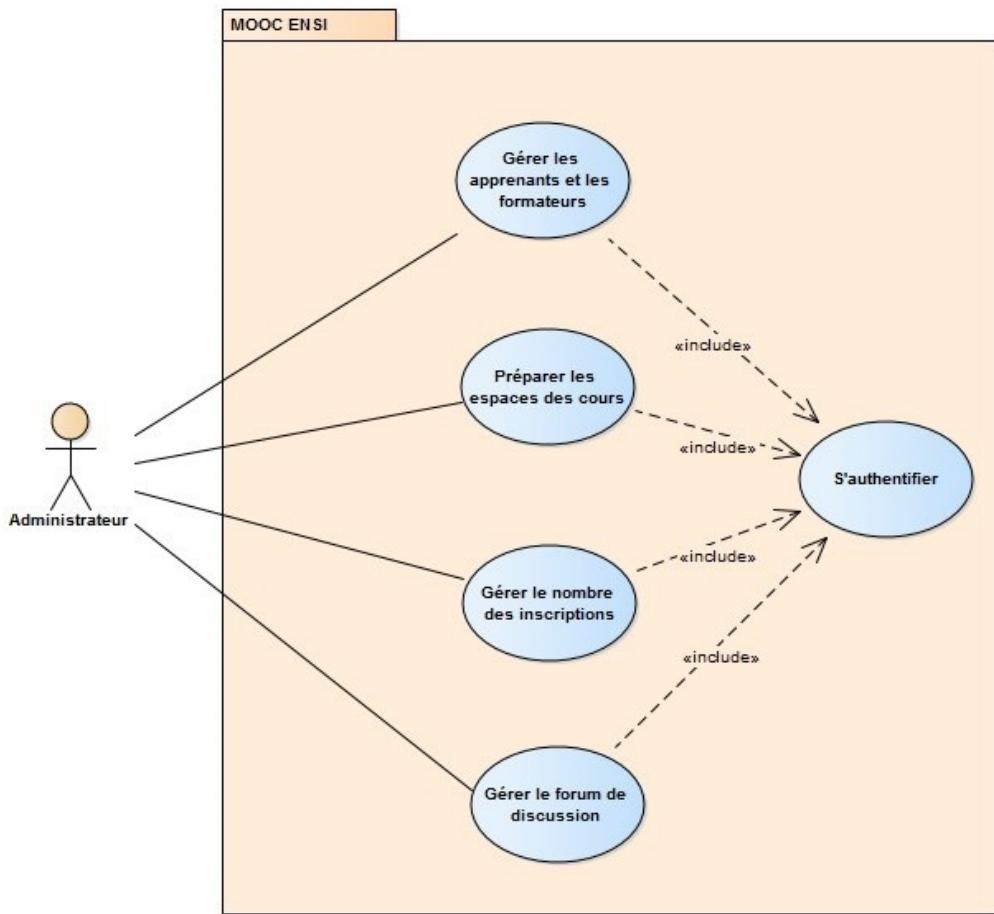


FIGURE 2.1 – Cas d'utilisation relatif à l'administrateur

Ce diagramme cas d'utilisation présente les différentes fonctionnalités que l'administrateur de notre application peut faire. L'administrateur doit s'authentifier pour accéder à son espace sur la plateforme. L'administrateur peut accéder à son propre espace. Il est le responsable des comptes utilisateurs. Il est le responsable des inscriptions aux différents cours. L'administrateur est celui qui prépare les espaces des nouveaux cours qui vont être mis sur la plateforme. Il est le modérateur du forum de discussions.

II.1.2 Diagramme de cas d'utilisation relatif à l'apprenant

La figure 2.2 ci-dessous représente le diagramme de cas d'utilisation relatif à l'apprenant.

Ce diagramme de cas d'utilisation présente les différentes tâches qu'un apprenant peut effectuer en accédant à son espace sur la plateforme. L'apprenant doit s'authentifier pour accéder à son espace sur la plateforme. Il peut consulter la liste des cours

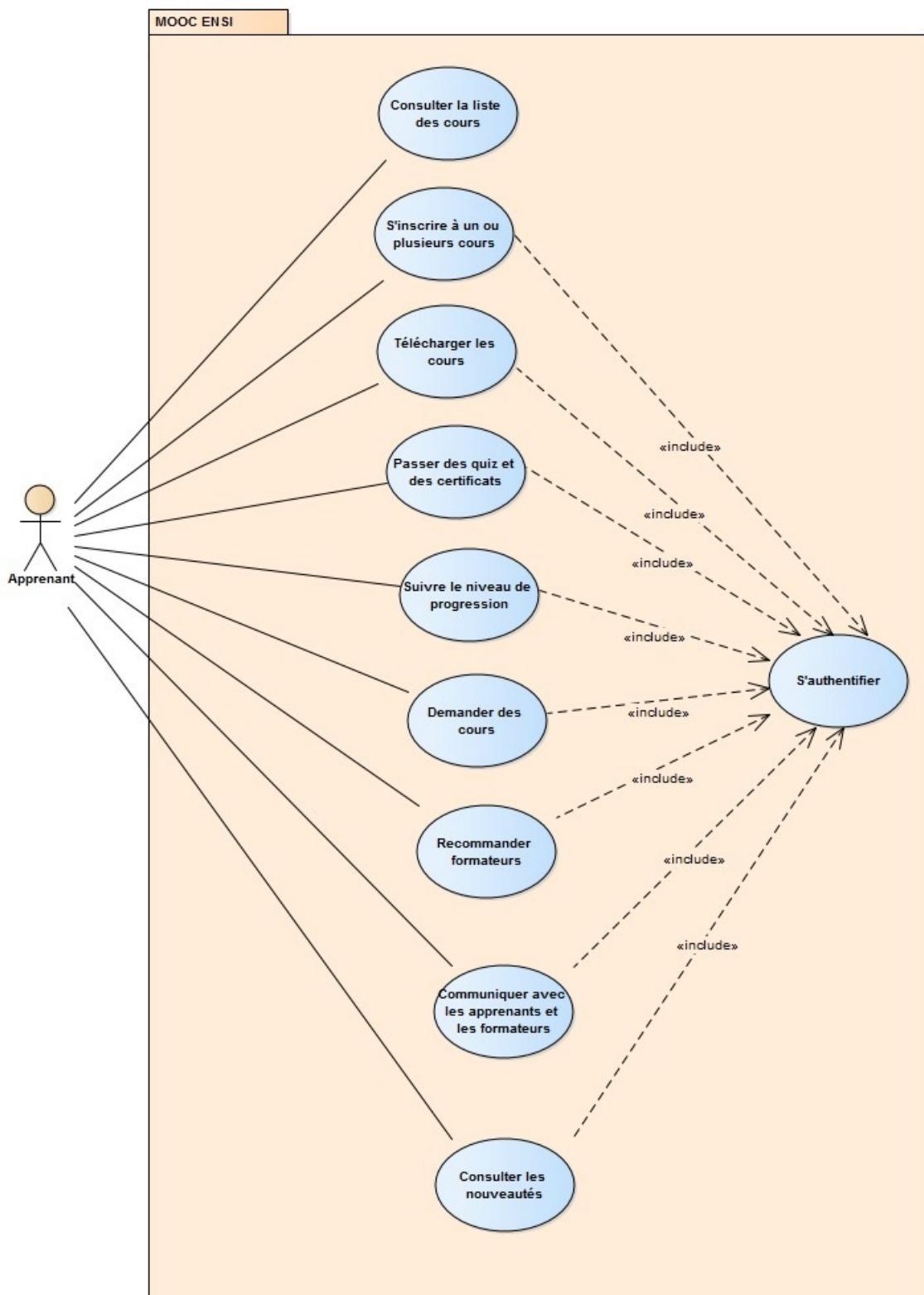


FIGURE 2.2 – Cas d'utilisation relatif à un apprenant

qui existent sur la plateforme. L'apprenant peut accéder à son propre espace. Il peut s'inscrire à un ou plusieurs cours. Il peut suivre son niveau de progression dans un cours donné. L'apprenant peut aussi passer des quiz, les nouveautés.

II.1.3 Diagramme de cas d'utilisation relatif au formateur

La figure 2.3 ci-dessous représente le diagramme de cas d'utilisation relatif à un

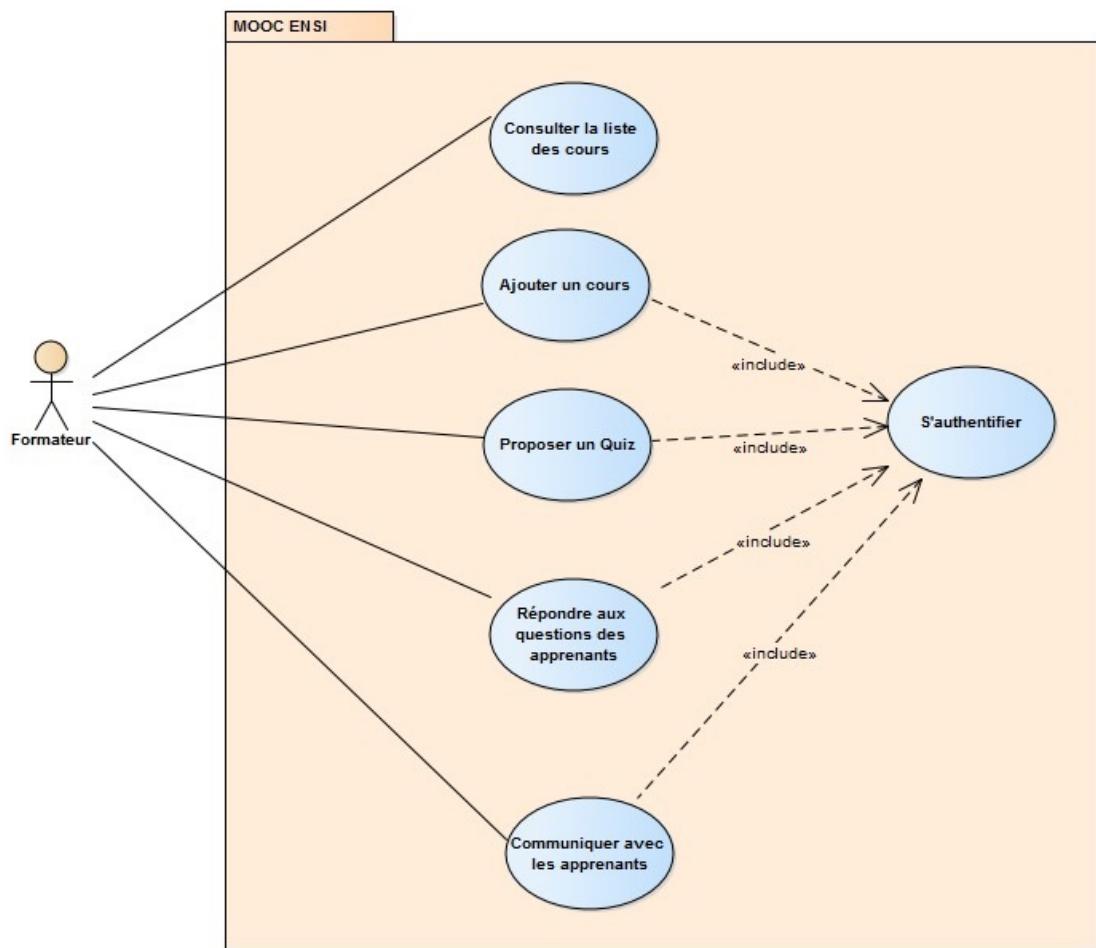


FIGURE 2.3 – Cas d'utilisation relatif à un formateur

Ce diagramme de cas d'utilisation présente les différentes tâches qu'un formateur peut effectuer. Il doit s'authentifier pour accéder à son espace. Il peut consulter la liste des cours publiés. Il peut ajouter un ou plusieurs cours, proposer des quiz, répondre aux questions des apprenants.

II.2 Description de quelques scénarios

II.2.1 Scénario du cas d'utilisation "S'authentifier"

La figure 2.4 ci-dessous représente le diagramme de séquence système du scénario d'authentification pour un apprenant :

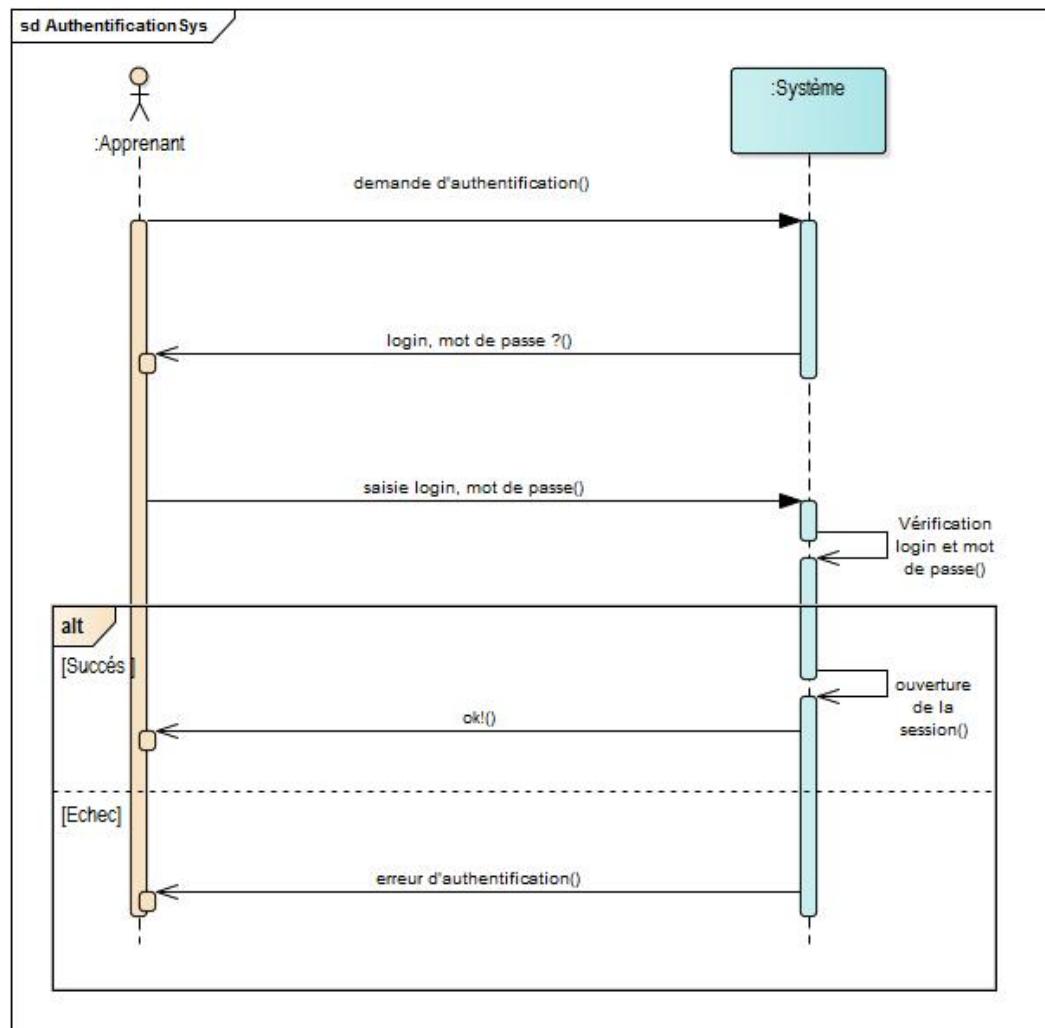


FIGURE 2.4 – Diagramme de séquence système d'authentification

Pour s'authentifier, un apprenant doit saisir son login et son mot de passe, si les données saisies sont correctes alors une session sera ouverte pour lui et il sera redirigé automatiquement à la page d'accueil de l'application appropriée pour son espace. Si les données sont erronées alors un message d'erreur apparaîtra demandant à l'apprenant de saisir de nouveau le login et le mot de passe corrects.

II.2.2 Scénario du cas d'utilisation "S'inscrire à un cours"

La figure 2.5 ci-dessous représente le diagramme de séquence système du scénario d'inscription à un cours pour un apprenant :

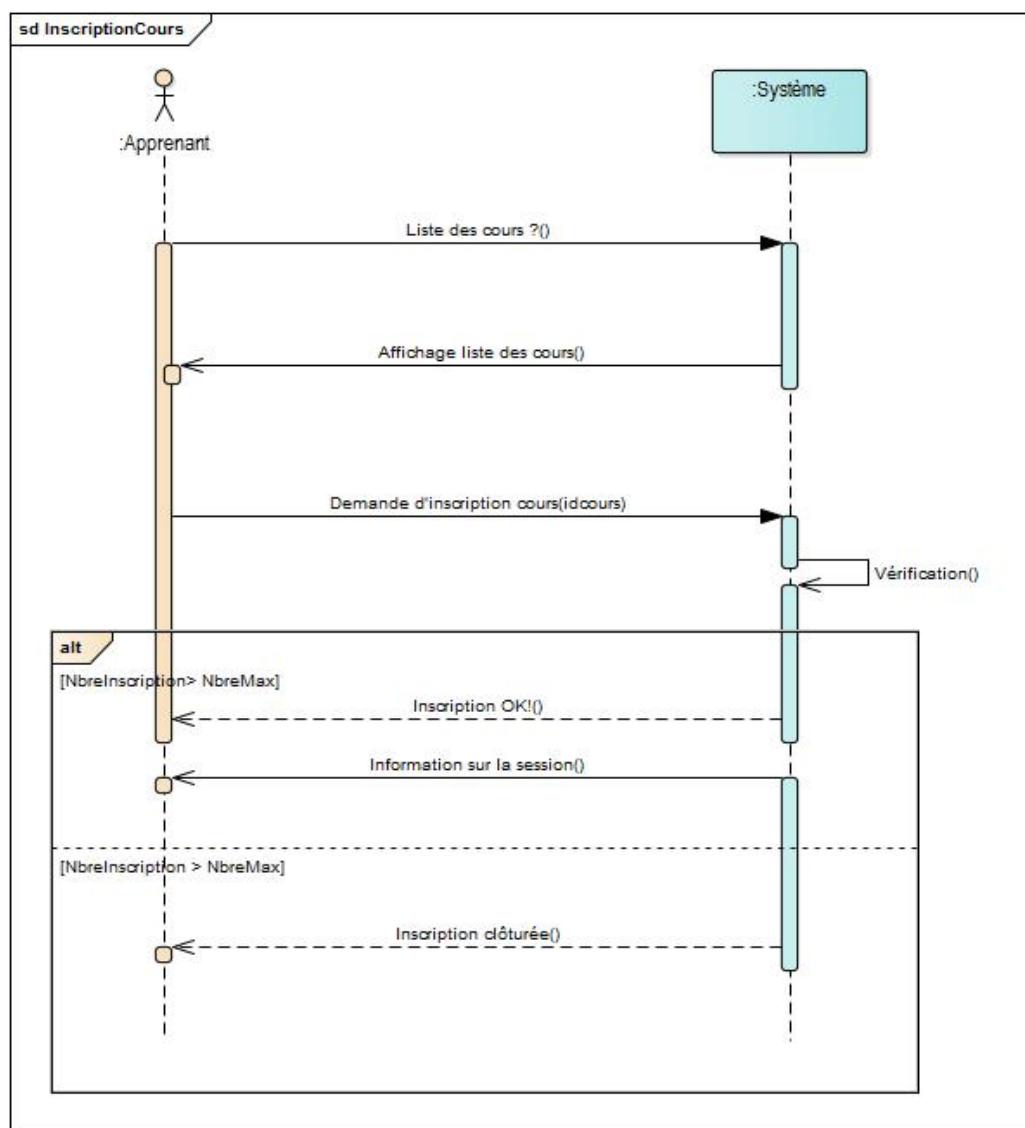


FIGURE 2.5 – Diagramme de séquence système d'inscription à un cours

Pour s'inscrire à un cours, un apprenant doit, d'abord, s'authentifier ensuite il peut choisir parmi les cours proposés de s'inscrire à un ou plusieurs cours si les inscriptions sont encore ouvertes.

II.2.3 Scénario du cas d'utilisation "Télécharger cours"

La figure 2.6 ci-dessous représente le diagramme de séquence système du scénario de téléchargement des cours disponibles pour un apprenant :

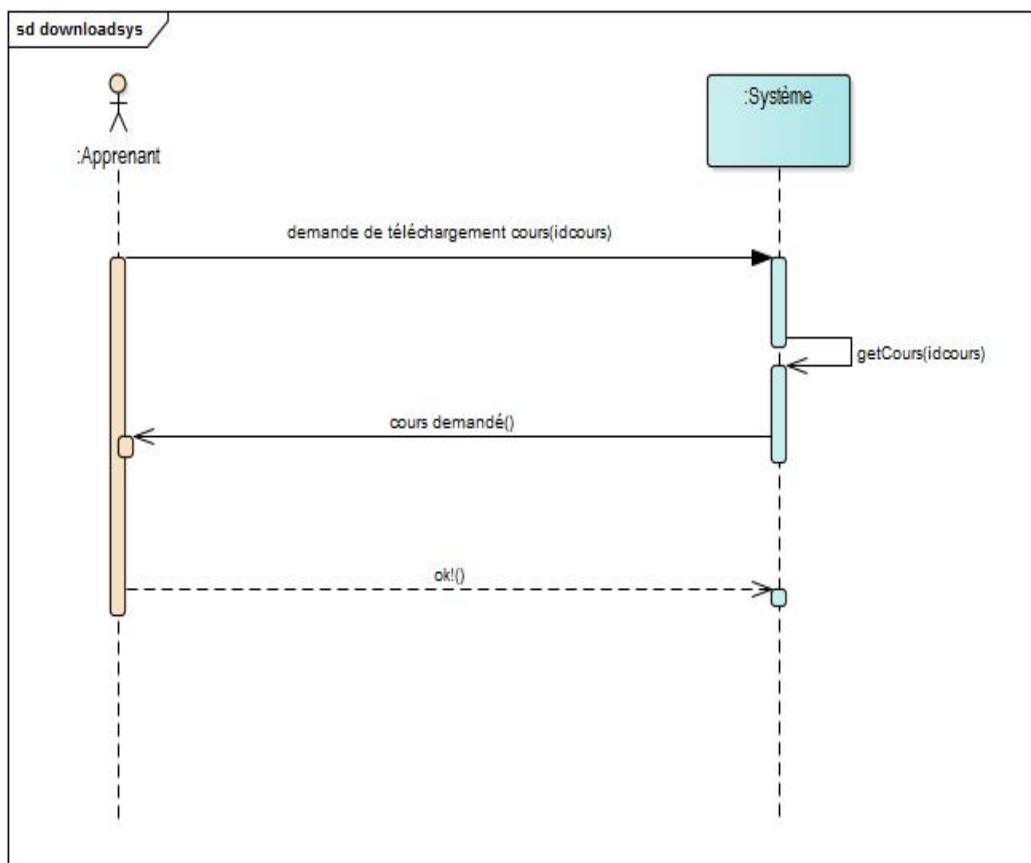


FIGURE 2.6 – Diagramme de séquence système de téléchargement d'un cours

Pour télécharger un cours, un apprenant doit, d'abord, s'authentifier et doit être inscrit aux cours pour pouvoir télécharger les documents disponibles.

II.2.4 Scénario du cas d'utilisation "Ajouter cours"

La figure 2.7 ci-dessous représente le diagramme de séquence système du scénario d'ajout d'un cours pour un formateur :

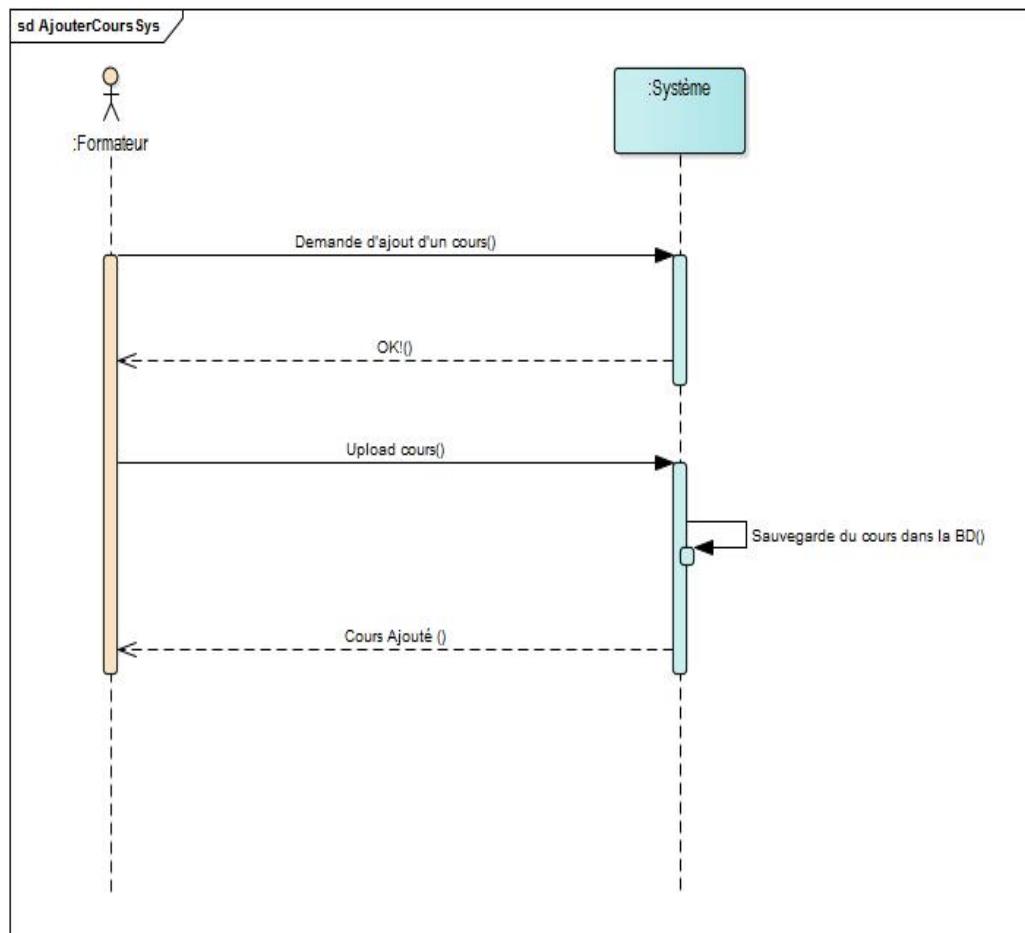


FIGURE 2.7 – Diagramme de séquence système d'ajout d'un cours

Tout formateur authentifié peut ajouter un ou plusieurs documents sur la plateforme dans une ou plusieurs catégories de cours.

Conclusion

Dans ce chapitre nous avons procédé à l'identification et la spécification des besoins de notre système. Dans le chapitre suivant, nous entamerons la partie de conception de notre système.

Chapitre 3

Conception

Dans ce chapitre, nous allons aborder la tâche la plus importante dans l'élaboration de ce travail, à savoir la tâche de conception. En effet, nous présentons, en premier lieu, l'architecture générale de notre application afin d'en extraire les différents modules qui la composent. Puis, nous détaillons chacun de ces modules conformément à la notation UML par la description des différents diagrammes de séquences relatifs aux cas d'utilisation qui ont été exprimés dans le chapitre précédent.

I Conception architecturale de l'application

I.1 Architecture physique

Avant de détailler l'architecture de notre application, il est recommandé d'avoir une vue globale sur les différentes architectures types existantes.

I.1.1 Description des architectures Types

Les principales architectures à décrire dans ce paragraphe sont au nombre de trois à savoir :

l'architecture 2-tiers, l'architecture 3-tiers et l'architecture N-tiers.

1. Architecture à deux niveaux (2-tiers) :

Une architecture 2-tiers [N22] est composée de deux éléments : un client et un serveur et où le tiers fait référence non pas à une entité physique mais logique. Elle se caractérise par les systèmes clients/serveurs dans lesquels le client demande

une ressource et le serveur la lui fournit directement. On peut représenter cette architecture via la figure 3.1 :

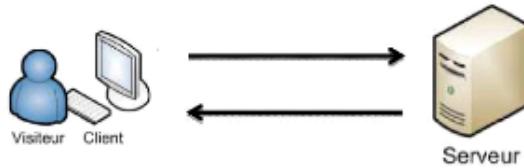


FIGURE 3.1 – Architecture 2-tiers [N23]

2. Architecture à trois niveaux (3-tiers) : L'architecture 3-tiers [N23] est composée de trois éléments comme le montre la figure 3.2 :

- *Le client* : le demandeur de ressources.
- *Le serveur d'applications (middleware)* : le serveur chargé de fournir la ressource mais faisant appel à un autre serveur.
- *Le serveur secondaire (généralement serveur de base de données)* : fournissant un service au premier serveur.

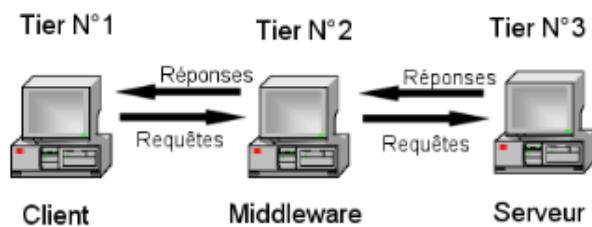


FIGURE 3.2 – Architecture 3-tiers [N24]

3. Architecture à plusieurs niveaux (N-tiers) :

L'architecutre N-tiers [N25] vise à séparer nettement n couches logicielles au sein d'un même système(généralement 3 ou 4). Elle sert à modéliser et à présenter une application comme empilement de n couches dont le rôle est défini comme suit :

- *Couche présentation des données* : Contient les différents types de clients, léger, lourd ou riche comme elle peut correspondre à la restitution de données en un format standard capable d'être interprété par différentes machines.
- *Couche métier* : Correspond à la mise de l'ensemble des règles de gestion de la logique applicative.
- *Couche persistance de données* : Correspondant aux données qui ont été gardées sur une longue durée, voire de manière persévérente.

La figure 3.3 donne un exemple d'architecture N-tiers.

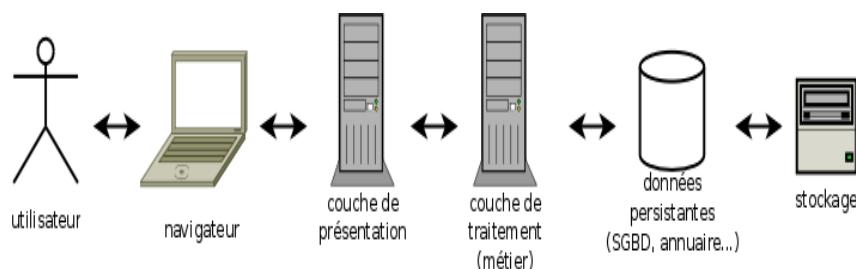


FIGURE 3.3 – Architecture N-tiers [N26]

I.1.2 Choix de l'architecture de l'application

Pour la réalisation de notre application, deux architectures sont envisageables : la première architecture est celle du client/serveur et la seconde alternative, est une architecture trois tiers. L'architecture trois tiers est retenue pour la réalisation de notre application. Ce choix est justifié par le fait que ce type d'architecture :

- Offre une plus grande flexibilité/souplesse : cette flexibilité permet d'envisager une grande souplesse pour l'introduction de toutes nouvelles technologies.
- Garantit une sécurité accrue : avec une architecture trois tiers l'accès à la base n'est effectué que par le serveur applicatif. Ce serveur est le seul à connaître la façon de se connecter à cette base. Il ne partage aucune des informations permettant l'accès aux données, en particulier le login et le password de la base. Il est alors possible de gérer la sécurité au niveau de ce serveur applicatif, par

exemple en maintenant la liste des utilisateurs avec leurs mots de passe ainsi que leurs droits d'accès aux fonctions du système.

- Procure de meilleures performances, étant donné le partage des tâches entre les différents serveurs.
- Réduit fortement les coûts de déploiement et d'administration : en effet, l'avantage principal d'une architecture trois tiers est la facilité de déploiement. L'application en elle-même n'est déployée que sur la partie serveur (serveur applicatif et serveur de base de données). Le client ne nécessite qu'une installation et une configuration minime. En effet il suffit d'installer un navigateur web compatible avec l'application pour que le client puisse accéder à l'application, ce navigateur étant par ailleurs souvent installé par défaut sur toutes les machines. Cette facilité de déploiement aura pour conséquence non seulement de réduire le coût de déploiement mais aussi de permettre une évolution régulière du système.

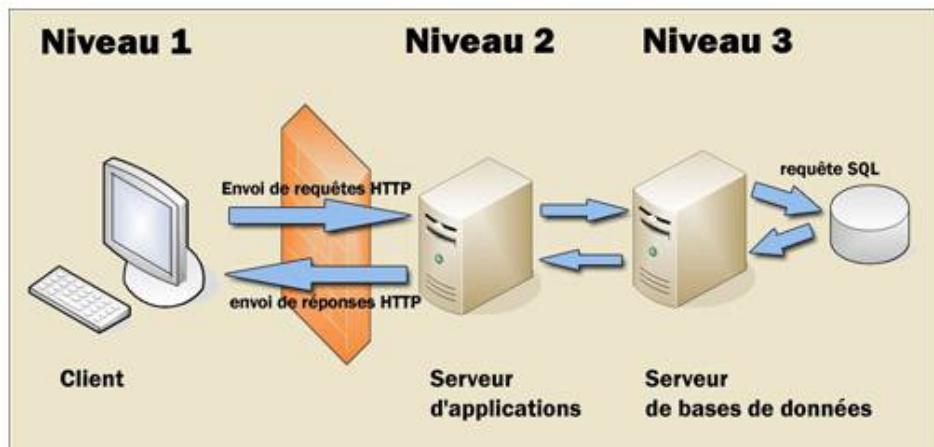


FIGURE 3.4 – Architecture trois tiers [N27]

L'architecture trois tiers comme le montre la figure 3.4, vise à séparer trois couches logicielles au sein d'une même application, à modéliser et à présenter cette application comme un empilement de trois couches dont le rôle est clairement défini :

- La présentation des données : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;

- Le traitement métier des données : correspondant à la mise en oeuvre de l'ensemble des règles de gestion et de la logique applicative ;
- L'accès aux données persistantes : correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

I.1.3 Modélisation de l'architecture physique

Pour illustrer le déploiement de notre application, nous avons utilisé le diagramme de déploiement, comme le montre la figure 3.5, qui illustre la disposition physique des différents matériels (ou noeuds) et la répartition des composants au sein des noeuds :

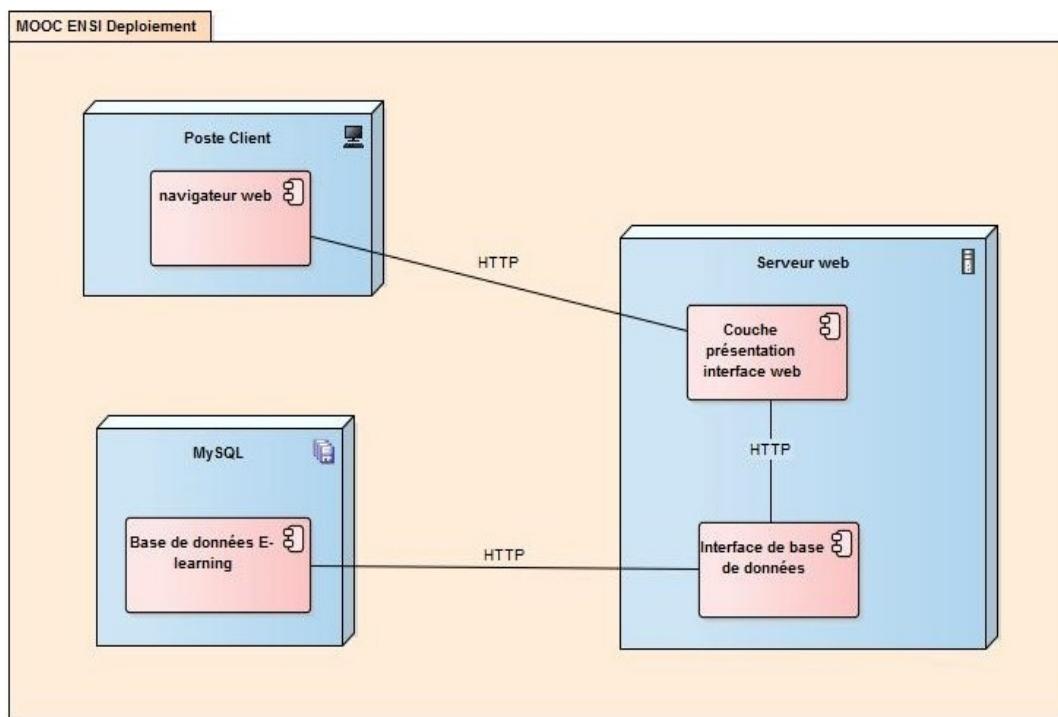


FIGURE 3.5 – Diagramme de déploiement du système

Notre diagramme de déploiement est constitué de trois noeuds principaux :

- Le poste client qui est composé d'un navigateur web, qui sert d'outil de communication entre les utilisateurs de notre système et le reste des noeuds. Les utilisateurs lancent leurs demandes sous forme de http request et en reçoivent des http response.
- Le serveur web qui regroupe deux composants :
 - La couche présentation qui regroupe l'ensemble des formulaires et interfaces à communiquer au navigateur par flux HTTP suite à une demande de l'utilisateur.
 - Interface de base données, qui est l'interface reliant notre application avec la base de données.
- Le serveur de base de données qui contient la base de données de notre application.

I.2 Architecture logique

Avant de détailler l'architecture logique de notre application, il est recommandé d'avoir une vue globale sur les différentes architectures types existantes.

I.2.1 Description des architectures types

1. Architecture MVP(MODELE - VUE - PRESENTATION)

C'est un "Design Pattern" qui propose de découper et de structurer l'architecture des interfaces utilisateur en couches. (On peut le combiner avec un découpage de l'accès aux données, lui aussi en couches, qu'on peut appeler DAL : Data Access Layer, et qui sépare les informations en mémoire de l'accès physique à la base de données). Il permet de séparer le code de gestion de l'interface, du code qui manipule les données métier. Cette architecture en couches de l'application et de l'IHM permet de rendre les projets plus faciles à maintenir et à faire évoluer.

2. Architecture MVVM(Model-View-View-Model)

C'est une architecture et une méthode de conception utilisée dans le génie logiciel.

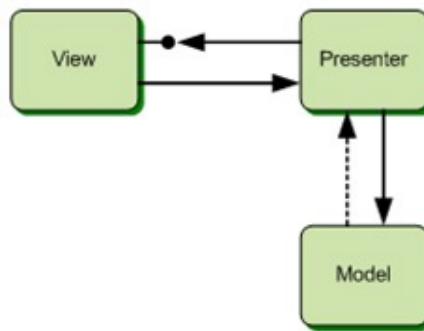


FIGURE 3.6 – Architecture MVP [N28]

MVVM est originaire de Microsoft et adapté pour le développement des applications basées sur les technologies Windows Presentation Foundation et Silverlight2 via l'outil MVVM Light3 par exemple. Cette méthode permet, tel le modèle MVC (modèle-vue-contrôleur), de séparer la vue de la logique et de l'accès aux données en accentuant les principes de binding et d'événement [N29].

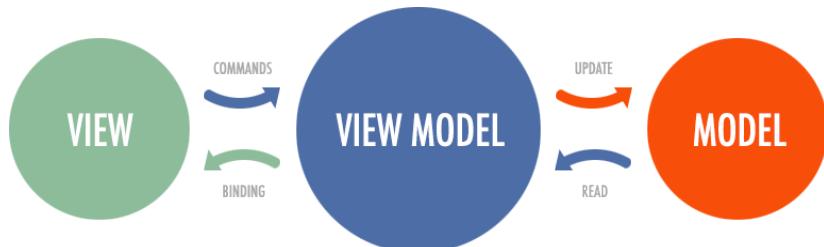


FIGURE 3.7 – Architecture MVVM [N30]

3. Architecture MVC(Model-View-Controller)

MVC est un patron de conception très répandu pour réaliser des sites web. Ce patron de conception est une solution éprouvée et reconnue permettant de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données.

MVC signifie Modèle-Vue-Contrôleur. C'est un modèle qui a été conçu au départ pour des applications dites "client lourd", c'est-à-dire dont la majorité des données sont traitées sur le poste client (par exemple : un traitement de texte comme Word). MVC était tellement puissant pour ces applications "client lourd", qu'il a été massivement adopté comme modèle pour la création d'application web (dites « client léger ») [N31].

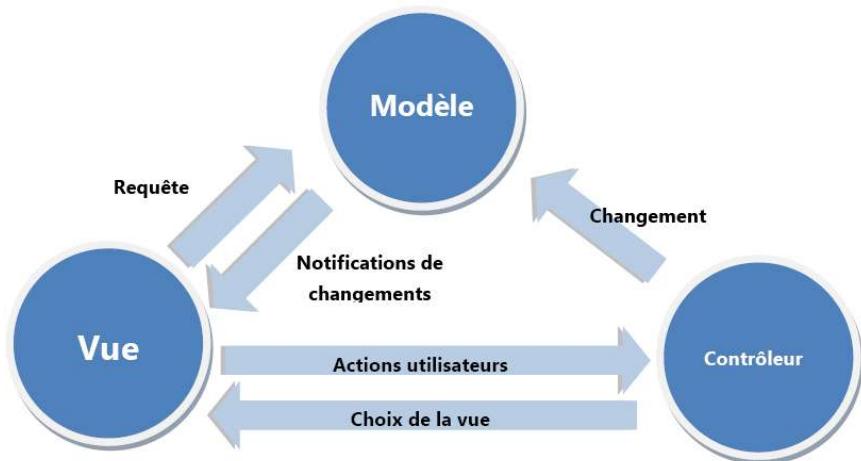


FIGURE 3.8 – Architecture MVC [N32]

I.2.2 Choix de l'architecture de l'application

Pour concevoir l'architecture logique de notre système nous optons pour l'architecture MVC (Modèle - Vue - Contrôleur) qui est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface. L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface mais aussi elle offre un cadre pour structurer une application. Dans l'architecture MVC, les rôles des trois entités sont les suivants :

- modèle : données (accès et mise à jour)
- vue : interface utilisateur (entrées et sorties)
- contrôleur : gestion des événements et synchronisation

Cette architecture est implémentée par Struts 2 [N33], comme le montre la figure 3.9 suivante :

Le traitement d'une demande d'un client se déroule de la façon suivante : Les URL demandées sont de la forme `http://machine:port/contexte/rep1/rep2/.../Action`. Le chemin `[/rep1/rep2/.../Action]` doit correspondre à une action définie dans un fichier de configuration de Struts 2, sinon elle est refusée. Une action est définie dans un fichier Xml. Les étapes suivantes sont alors exécutées :

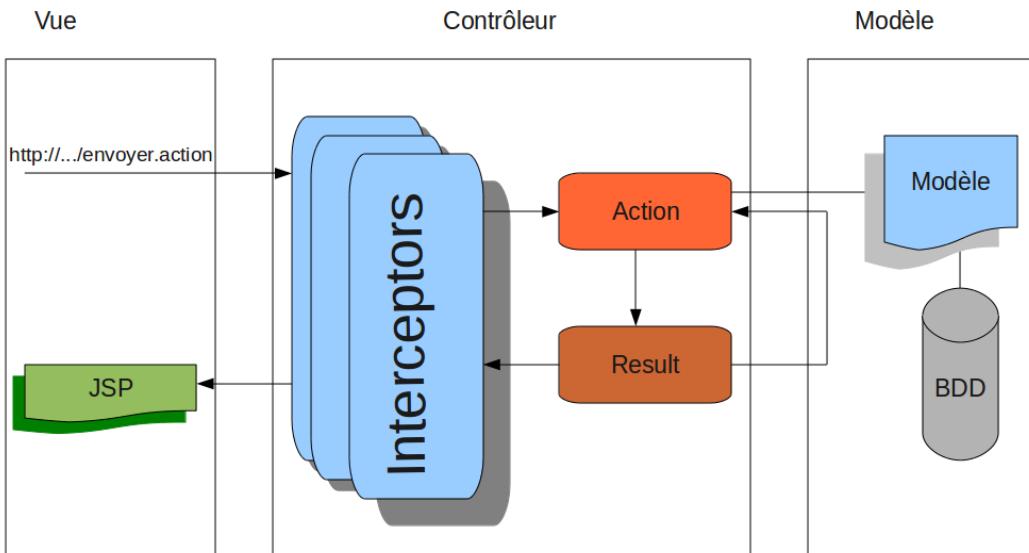


FIGURE 3.9 – Architecture de Struts2 [N34]

- **Demande** : le client navigateur fait une demande au contrôleur [FilterDispatcher]. Celui-ci voit passer toutes les demandes des clients. C'est la porte d'entrée de l'application. C'est le C de MVC.
- **Traitemen**t : le contrôleur C consulte son fichier de configuration et découvre que l'action actions/Action1 existe. Le namespace concaténé avec le nom de l'action définit l'action actions/Action1. Le contrôleur C instancie une classe de type [actions.Action1]. Le nom et le package de cette classe peuvent être quelconques. Il faut donc que la classe [actions.Action1] ait ces méthodes setParami pour chacun des paramètres attendus. Le contrôleur C demande à la méthode de signature [String execute()] de la classe [actions.Action1] de s'exécuter. Celle-ci peut alors exploiter les paramètres parami que la classe a récupérés. Dans le traitement de la demande de l'utilisateur, elle peut avoir besoin de la couche [metier]. Une fois la demande du client traitée, celle-ci peut appeler diverses réponses. Un exemple classique est :

Une page d'erreurs si la demande n'a pu être traitée correctement une page de confirmation sinon la méthode execute rend au contrôleur C un résultat de type chaîne de caractères appelée clé de navigation qui va mettre à jour le modèle M que va exploiter la page JSP qui va être envoyée en réponse à l'utilisateur. Ce modèle peut comporter des éléments de : la classe [actions.Action1] instanciée la

session de l'utilisateur.

- **Réponse** : le contrôleur C demande à la page JSP correspondant à la clé de navigation de s'afficher. C'est la vue, le V de MVC. La page JSP utilise un modèle M pour initialiser les parties dynamiques de la réponse qu'elle doit envoyer au client.

II Conception détaillée

Dans cette partie nous allons entamer la description des détails conceptuels relatifs à notre application d'E-learning. Ainsi, nous commencerons par détailler le modèle.

II.1 Conception du modèle

La figure 3.10 ci-dessous représente le diagramme de classes relatif à la couche modèle de notre application :

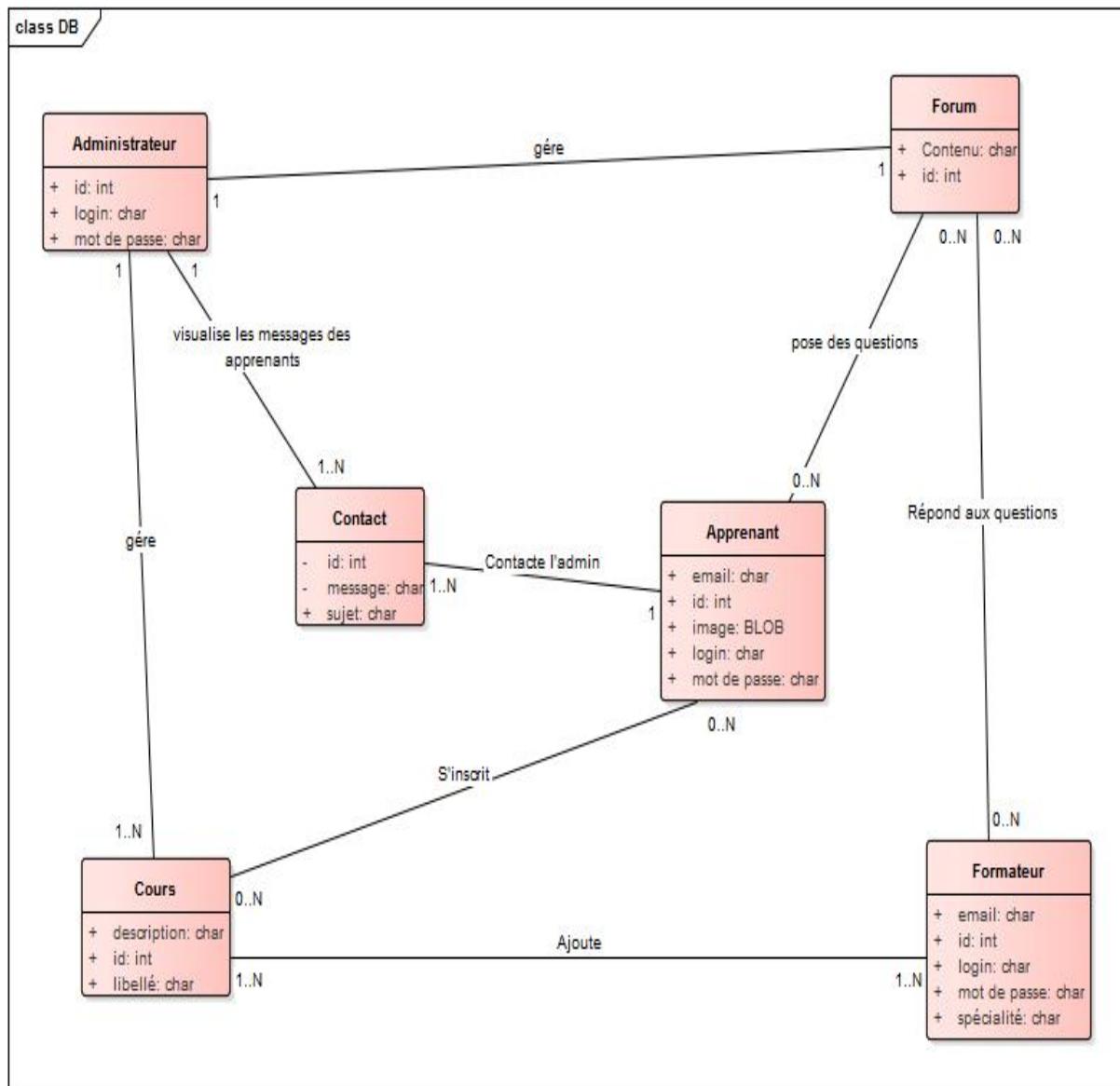


FIGURE 3.10 – Diagramme de classe de la couche modèle

TABLE 3.1 – Les principales classes conçues pour l’élaboration de ce modèle

Nom de la Classe	Description
Administrateur	La classe administrateur contient les informations du responsable de la plateforme.
Formateur	La classe Formateur contient les coordonées des différents formateurs de notre application.
Apprenant	La classe Apprenant contient les différents étudiants inscrits sur notre plateforme.
Cours	La table Cours contient les différents cours et quiz hébergés sur notre plateforme.
Contact	La classe Contact contient les différents messages envoyés des apprenants pour l'administrateur de la plateforme.
Forum	La classe Forum contient les discussions échangées entre les apprenants et les formateurs.

II.2 Conception de la Vue

La figure 3.11 ci-dessous représente le diagramme de classes de la vue de notre application :

CHAPITRE 3. CONCEPTION

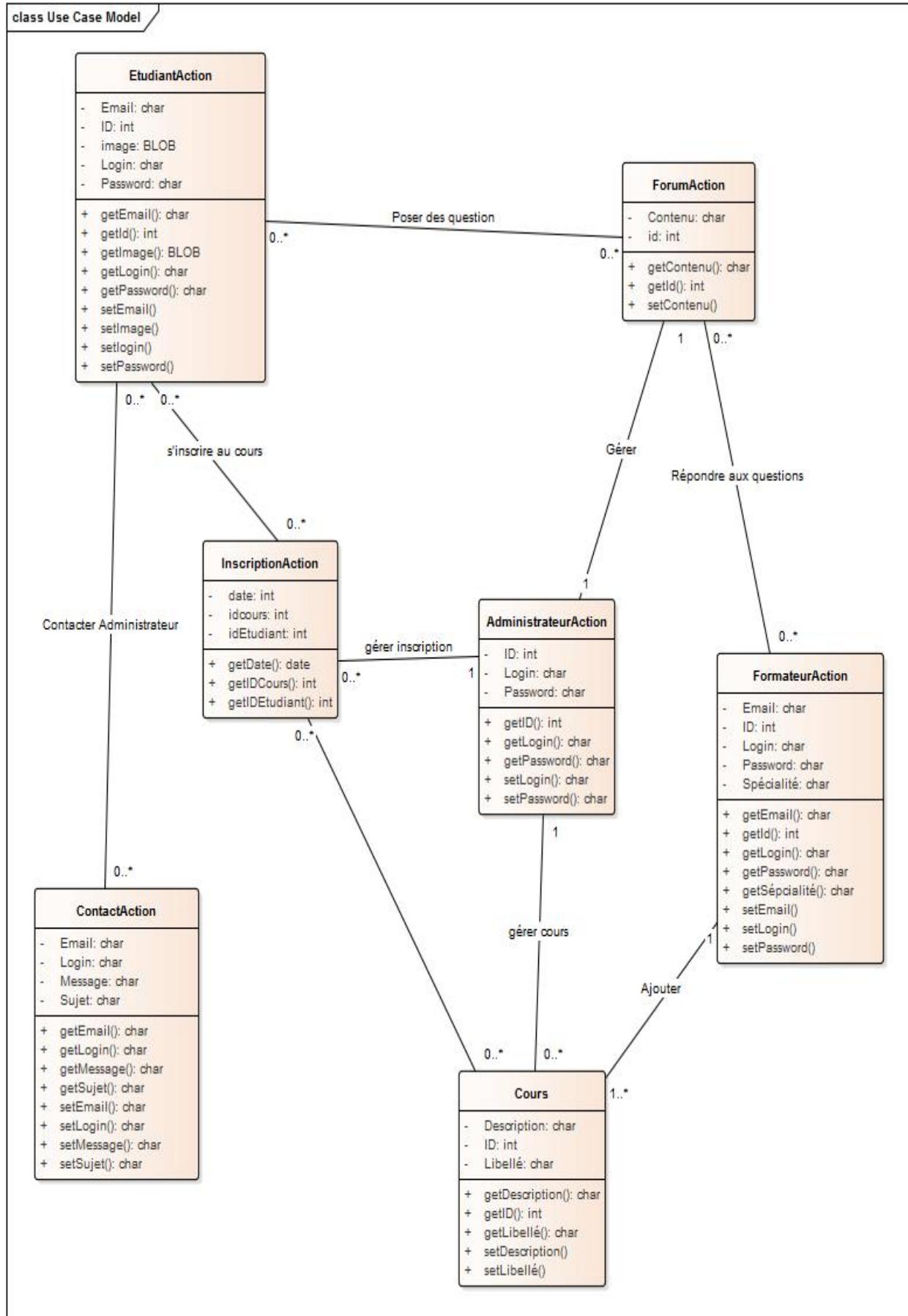


FIGURE 3.11 – Diagramme de classe de la couche vue

TABLE 3.2 – Les principales classes conçues pour l’élaboration de cette couche

Nom de la Classe	Description
AdministrateurAction	La classe AdministrateurAction contient les attributs et les méthodes nécessaires pour effectuer les actions d'un administrateur.
FormateurAction	La classe FormateurAction contient les attributs et les méthodes nécessaires pour effectuer les actions d'un formateur.
ApprenantAction	La classe ApprenantAction contient les attributs et les méthodes nécessaires pour effectuer les actions d'un apprenant.
CoursAction	La classe CoursAction contient les attributs et les méthodes nécessaires pour ajouter et télécharger les cours.
ContactAction	La classe ContactAction contient les attributs et les méthodes nécessaires pour qu'un apprenant puisse contacter le responsable de la plateforme
ForumAction	La classe ForumAction contient les attributs et les méthodes nécessaires pour la communication entre apprenants et formateurs.
InscriptionAction	La classe InscriptionAction contient les inscriptions des apprenants aux cours.

III Description de quelques scénarii

III.1 Scénario d'authentification d'un apprenant

La figure 3.12 ci-dessous représente le digaramme de séquence d'authentification d'un apprenant :

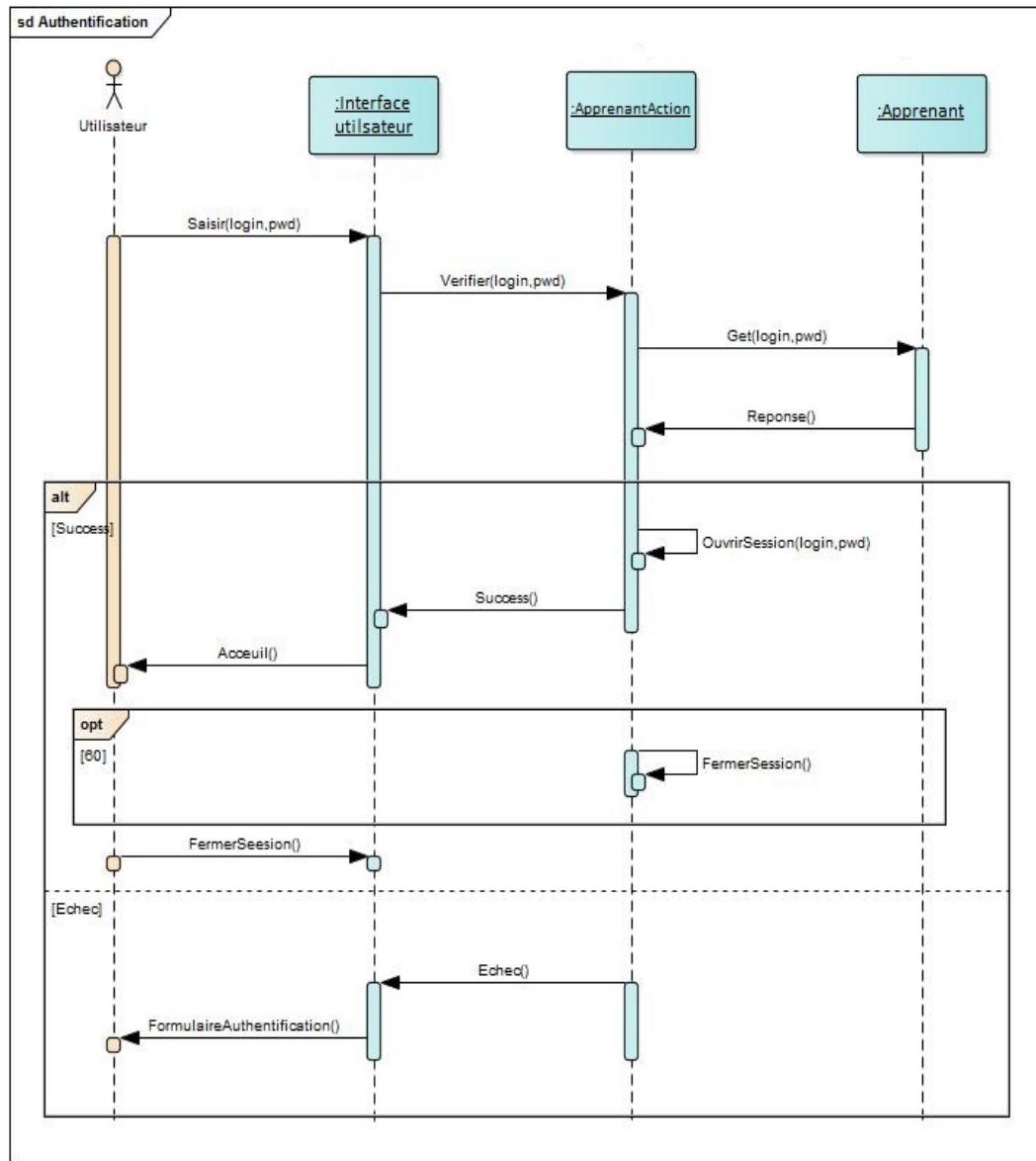


FIGURE 3.12 – Diagramme de séquence du scénario d'authentification

Un apprenant saisit son login et son mot de passe afin d'accéder à son espace via l'interface d'authentification, la classe "'ApprenantAction'" se charge d'exécuter cette action en vérifiant si les données saisies sont correctes. Si les données saisies par l'apprenant sont erronées alors il doit resaisir le login et/ou le mot de passe.

III.2 Scénario demande d'inscription à un cours

La figure 3.12 ci-dessous représente le digaramme de séquence d'inscription d'un apprenant à un cours :

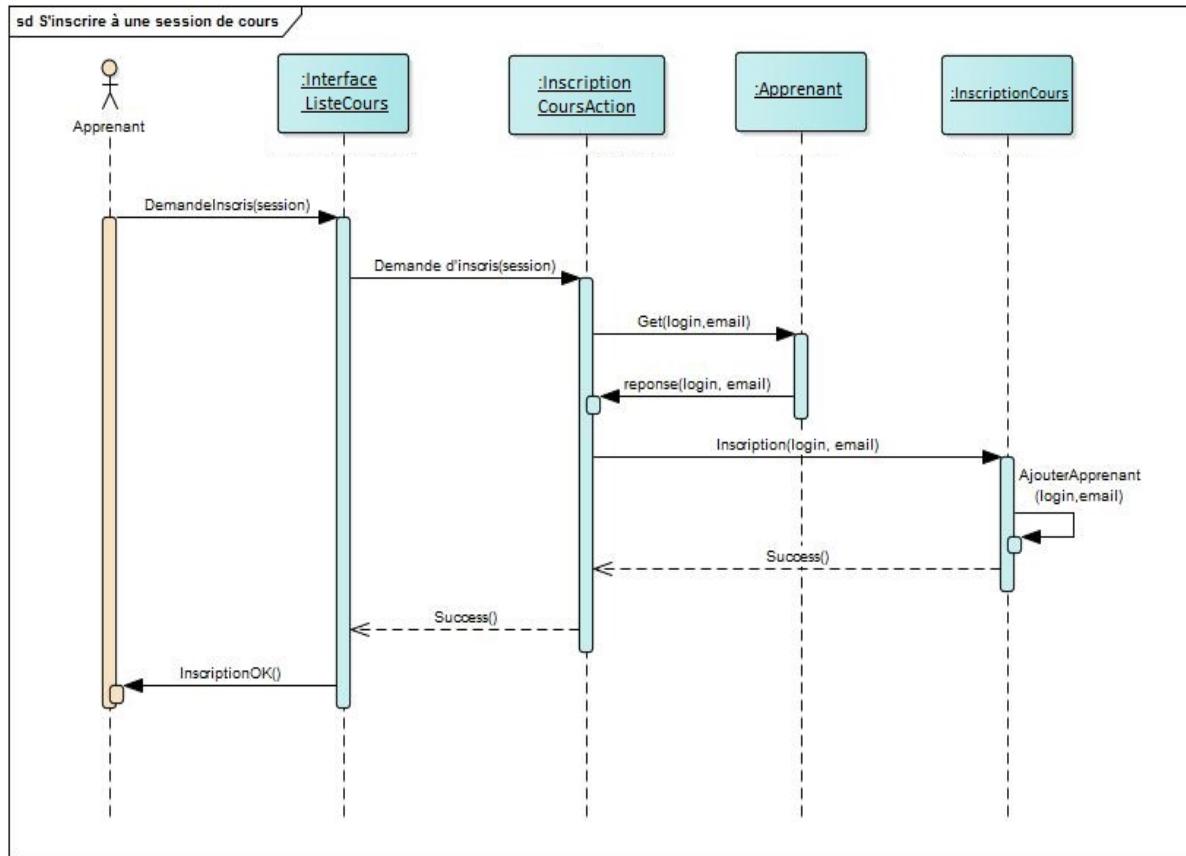


FIGURE 3.13 – Diagramme de séquence du scénario d'inscription à un cours

Un apprenant peut s'inscrire à un ou plusieurs cours proposés sur la plateforme. La "classe InscriptionCours" se charge d'exécuter cette action. Elle récupère les données de l'apprenant et l'enregistre à la liste des inscriptions aux cours.

III.3 Scénario demande de téléchargement d'un cours

La figure 3.13 ci-dessous représente le diagramme de séquence de téléchargement d'un cours par un apprenant :

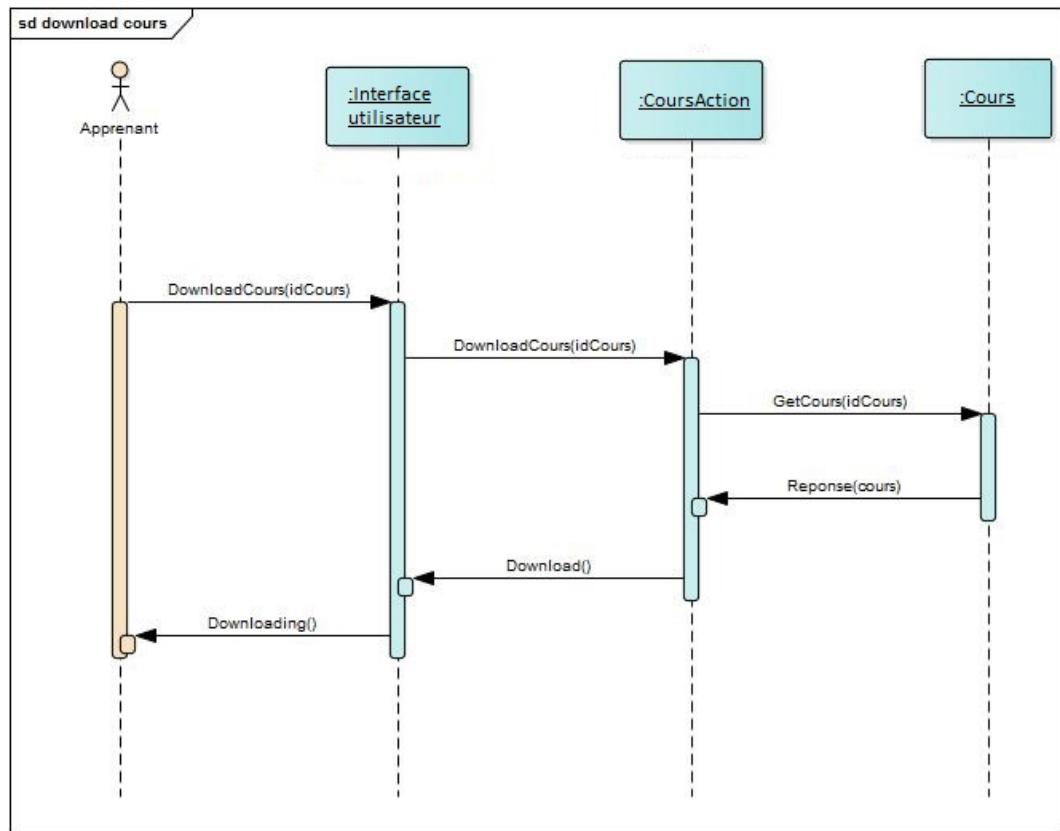


FIGURE 3.14 – Diagramme de séquence du scénario de téléchargement du cours

Un apprenant, inscrit à un cours donné, a la possibilité de télécharger les documents disponibles pour ce cours. La classe qui exécute cette action est la classe "CoursAction" qui se charge de récupérer le cours demandé depuis la base de données.

III.4 Scénario demande d'ajout du cours

La figure 3.15 ci-dessous représente le diagramme de séquence d'ajout d'un cours par un formateur :

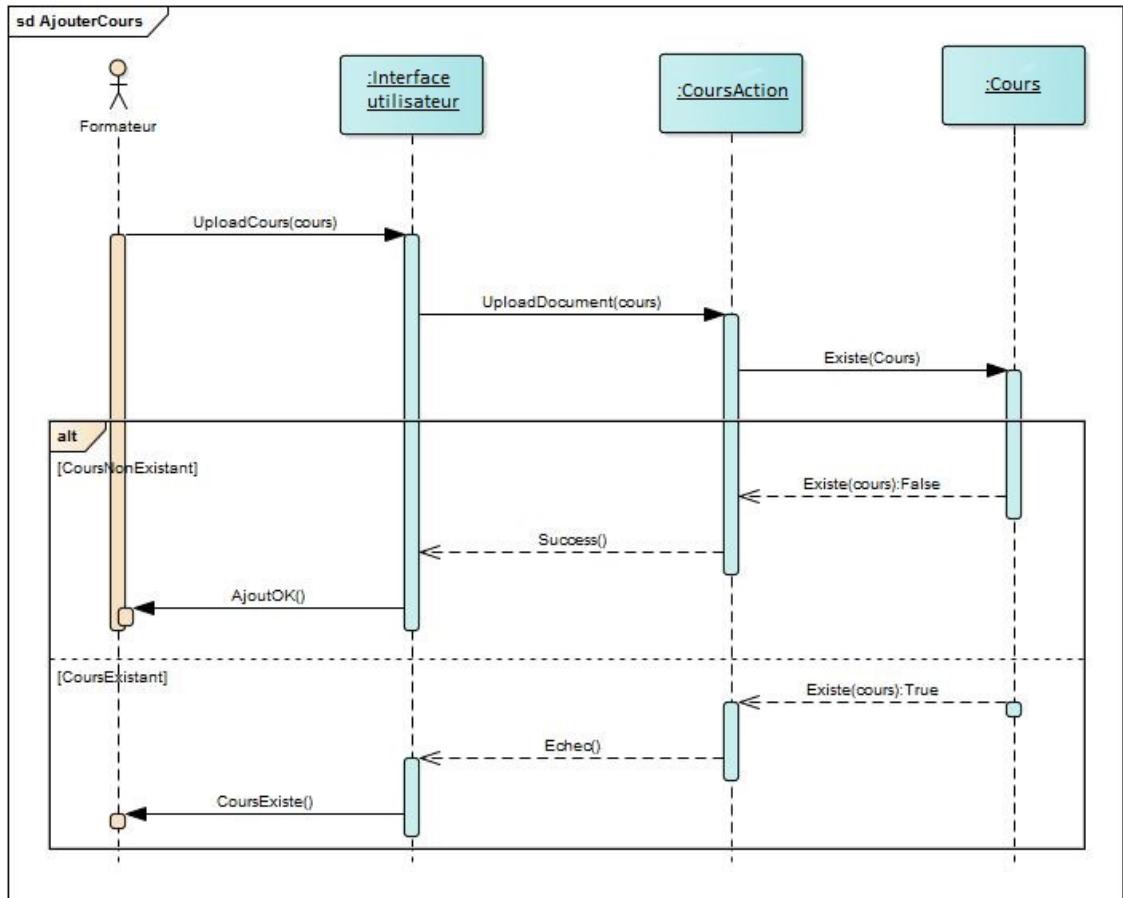


FIGURE 3.15 – Diagramme de séquence du scénario d’ajout du cours

Un formateur peut ajouter plusieurs cours sur notre plateforme via l’interface d’ajout des cours de son espace. Pour exécuter cette action, la classe “‘CoursAction”’ récupère les informations saisies par le formateur et le cours puis elle se charge de l’enregistrer dans la base données.

Conclusion

A travers ce chapitre, nous avons présenté notre conception de l’application. Nous avons fourni, dans un premier lieu, une conception globale de l’organisation de notre système. Ensuite, nous avons présenté la conception détaillée de l’application à travers les diagrammes UML. A présent, nous sommes capable d’entamer la partie réalisation.

Chapitre 4

Réalisation

Ce chapitre constitue le dernier volet du rapport ayant pour objectif d'exposer le travail achevé. Pour ce faire, nous allons présenter dans un premier temps l'environnement matériel et logiciel supportant notre application. Par la suite, nous présentons la plateforme de développement et les choix technologiques. Ensuite, nous allons passer en revue les différentes tâches réalisées à travers quelques interfaces homme-machine et un chronogramme récapitulatif qui décrit toutes étapes de mise en oeuvre de notre système.

I Environnements de travail

Tout au long de la réalisation de notre application, nous avons utilisé des matériels et des logiciels bien particuliers.

I.1 Environnements de développement matériel

Pour mener à bien la réalisation, nous avons utilisé comme environnement matériel, deux postes de travail ayant les caractéristiques suivantes :

- Premier poste :
 - Système d'exploitation : Windows 10
 - Disque dur : 1 To
 - Ram : 8 Go
 - Processeur : Intel(R) Core(TM)i7-6500U CPU 2.5GHz

- Deuxième poste :
 - Système d'exploitation : Windows 8
 - Disque dur : 500 Go
 - Ram : 6Go
 - Processeur : Intel(R) Core(TM)i5 500U CPU 2.5GHz

I.2 Environnements de développement logiciel

Dans cette partie, nous nous intéressons aux langages, aux bibliothèques et aux techniques de programmation utilisées tout au long de la réalisation de notre application en justifiant notre choix.

I.2.1 Plateforme JAVA EE

Java Enterprise Edition [N35] est une spécification pour la technique Java d'Oracle plus particulièrement destinée aux applications d'entreprise. Ces applications sont considérées dans une approche multi-niveaux. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au framework Java standard afin de faciliter notamment la création d'applications réparties.

I.2.2 Le framework Struts 2 pour l'implémentation de la couche présentation

Apache Struts 2 [N36] est un framework Java, pour le développement d'applications Web. Ce n'est pas une extension d'Apache Struts 1. Struts 2 regroupe les avantages de deux précédents outils, WebWork et Struts 1, mais c'est une refonte complète. Cette seconde génération de framework MVC utilise les notions suivantes : intercepteurs, annotations, langage d'expression OGNL, l'intégration d'outils comme JSTL ou Spring framework.

L'avantage majeur de cette plateforme est qu'elle est très vaste, et propose de très nombreux outils qui adressent toutes les problématiques de l'informatique d'entreprise : exposer des fonctionnalités avancées sur un site web, accès à des bases de données, communications distantes entre les applications Java ou même non Java. Ceci est réalisé grâce à des technologies "standard" donc connues d'un grand nombre de développeurs

et de nombreux "frameworks" (Spring, Hibernate, ect) qui rendent la programmation plus simple.

I.2.3 Hibernate pour l'implémentation de la couche d'accès aux données

Hibernate [N37] est une solution open source de type ORM (Object Relational Mapping) qui nous a permis de faciliter le développement de la couche persistance de notre application. Hibernate nous a permis de représenter notre base de données en objets Java et vice versa. Hibernate nous a facilité la persistance et la recherche de données dans notre base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement fastidieux et redondant. Nous avons utilisé Hibernate notamment grâce de ses bonnes performances et de son ouverture à de nombreuses bases de données.

I.2.4 HTML5, CSS3, jQuery et JavaScript pour la construction des interfaces riches

HTML5 [N38] a simplifié certaines balises afin d'alléger le code. Il introduit également un ensemble de nouvelles balises afin de donner plus de sémantique à nos pages. HTML 5 nous a permis une utilisation plus propre, code plus propre et nous pouvons, ainsi, éliminer la plupart des balises div et les remplacer par des éléments HTML 5 sémantiques.

CSS3 [N39] nous a permis d'automatiser certains effets visuels qui nécessitaient jusqu'à présent l'utilisation d'images, de scripts ou de modifications du code HTML : ombres portées, coins arrondis, opacité, arrières plans multiples, dégradés complexes, multi-colonage, effets textuels... L'avantage majeur et bien entendu une optimisation sur le temps de chargement des pages, car moins d'images, moins de code, moins de fichiers flash ect.

JavaScript [N40] est un langage de programmation Web distinct du HTML qui nous a permis d'ajouter de l'interactivité à nos pages Web en accédant directement aux éléments de la page HTML et en les manipulant. Il est, sans aucun doute, un des

langages les plus populaires et les plus utilisés sur Internet, surtout qu'il fonctionne sur bon nombre de navigateurs différents.

jQuery [N40] est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. Il nous a permis de mieux développer nos pages web puisqu'il est :

- Stable et robuste
- Facile à apprendre et à utiliser
- Syntaxe claire et concise réduisant le nombre de lignes de code à écrire
- Support des différences entre les navigateurs

II Interfaces Homme-Machine

Tout visiteur de notre plateforme aura accès à la page d'accueil, la liste des cours disponibles, le FAQ, la page de contact et le blog. La figure 4.1 ci-dessous représente la page d'accueil de notre application d'E-learning.

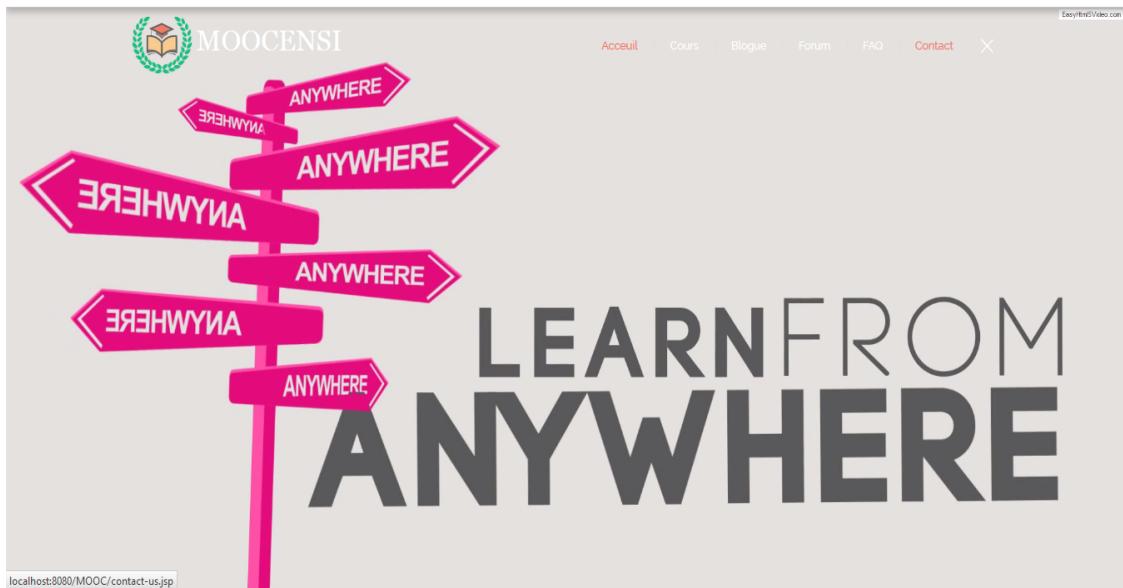


FIGURE 4.1 – Page d'accueil

CHAPITRE 4. RÉALISATION

La figure 4.2 ci-dessous représente la liste des cours disponibles sur la plateforme.

The screenshot shows the MOOCENSI platform's course list page. At the top, there is a navigation bar with links for Accueil, Cours, Blogue, Forum, FAQ, and Contact. A banner features a person's hands on a keyboard with the text "Liste des Cours". Below the banner, the heading "Liste des cours" is displayed. A large image of the UML logo (Unified Modeling Language) is shown, along with the text "initiation UML 2.5", the date "01 Mai", and the time "0 à 1800".

FIGURE 4.2 – Liste des cours sur la plateforme

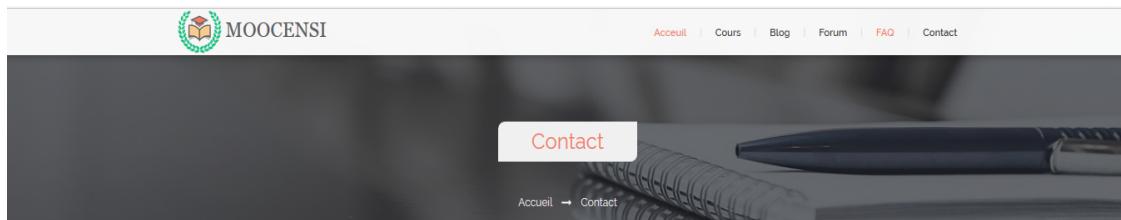
La figure 4.3 ci-dessous représente le FAQ de notre plateforme.

The screenshot shows the MOOCENSI platform's FAQ page. At the top, there is a navigation bar with links for Accueil, Cours, Blogue, Forum, FAQ, and Contact. A banner features a group of people in a library setting with the text "FAQ". Below the banner, the heading "QUESTIONS FRÉQUENTES POSÉES" is displayed. Two sections are shown: "Découverte de la plateforme" and "Les cours". Each section contains a list of frequently asked questions with plus and minus signs for expanding/collapsing the answers.

FIGURE 4.3 – FAQ

CHAPITRE 4. RÉALISATION

La figure 4.4 ci-dessous représente la page de contact via laquelle les utilisateurs de la plateforme peuvent envoyer des messages à l'administrateur.



Contact

Nom
Email
Sujet
Message

Envoyer



FIGURE 4.4 – Page de contact

La figure 4.5 ci-dessous représente le blog de notre plateforme.

A screenshot of the MOOCENSI website's blog page. At the top, there is a navigation bar with links for Accueil, Cours, Blogue, Forum, FAQ, and Contact. The main heading is "Blogue". Below the heading, there is a date indicator "26 Fev" and a user profile for "MIKE HEBERT". A large image shows a teacher standing in front of a whiteboard, gesturing while students raise their hands. To the right of the image is a "Calendrier" section showing the month of April 2017. The calendar grid includes dates from 1 to 30, with April 29 highlighted in green. There are also navigation arrows for the calendar.

FIGURE 4.5 – Blog

CHAPITRE 4. RÉALISATION

La figure 4.6 ci-dessous représente une description de notre plateforme et les services qu'elle offre.

The screenshot shows the MOOCENSI website's homepage. At the top, there is a navigation bar with links for Accueil, Cours, Blogue, Forum, FAQ, and Contact. A logo for MOOCENSI is on the left. In the center, there is a button labeled "Qui sommes-nous ?". Below this, a breadcrumb navigation shows "Accueil → Qui Sommes-nous ?". To the right, there is a photograph of two people working at a desk. On the left side of the page, there is a sidebar with sections for "Notre Mission", "Nos cours", and "A vous". The main content area has a heading "MOOCENSI en quelques mots" and a paragraph describing their mission to offer open courses to form people in various fields.

FIGURE 4.6 – Description de la plateforme

Une fois authentifié, un administrateur peut accéder à son propre espace. La figure 4.7 ci-dessous représente la page d'authentification de l'administrateur de notre application.

The screenshot shows the administrator login page. The background is orange with various educational icons floating around. At the top, there is a navigation bar with links for Accueil, Cours, Blog, Forum, FAQ, and Contact. Below the navigation bar, there is a blue header bar with tabs for Apprenant, Formateur, and administrateur. The main area is titled "Administrateur Login" and features a logo of an open book with a graduation cap. It provides options to log in via Facebook or Gmail, or to enter an Email and Mot de passe (password). There is also a link for "Mot de passe oublié?" (Forgot password?) and a green "Se connecter" (Connect) button.

FIGURE 4.7 – Page d'authentification de l'administrateur

CHAPITRE 4. RÉALISATION

La figure 4.8 ci-dessous représente la page de profil de l'administrateur via laquelle il peut modifier ses informations.

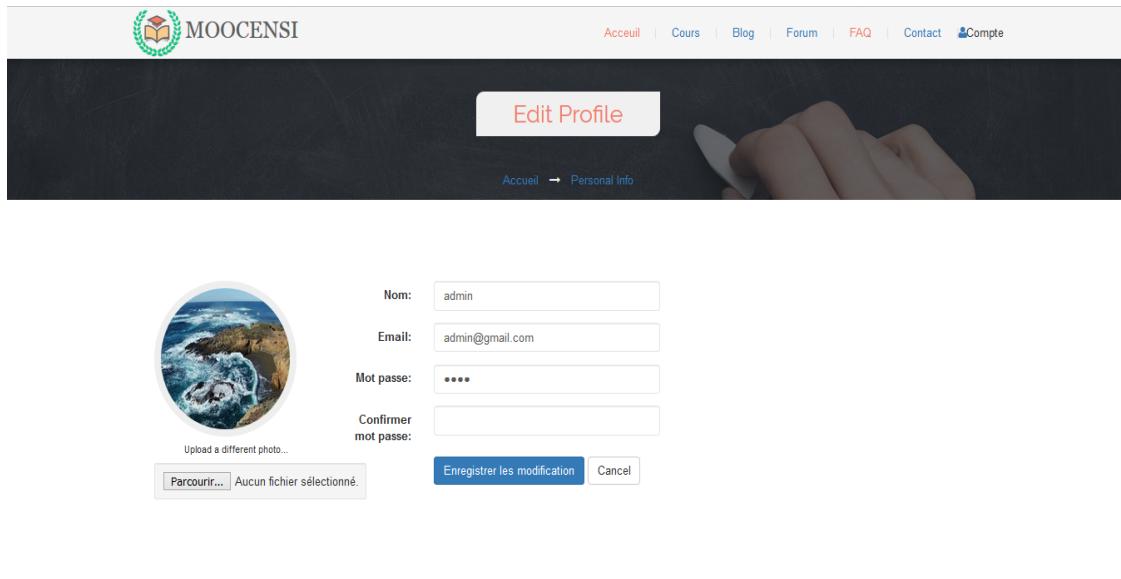


FIGURE 4.8 – Page de profil de l'administrateur

La figure 4.9 ci-dessous représente l'espace de l'administrateur où il trouve la liste des nouveaux cours et leurs détails, ajoutés par les formateurs.

The screenshot shows the administrator's dashboard with a banner featuring a hand holding a pen. The main area has a header 'Bienvenue(e) admin' and a 'Formateur' button. Below is a table listing courses:

Nom	Titre	poste	Email	Description
Intzé Fries	Doctorante chez L'ENSI	Enseignante chez ENSI	kooifran123@gmail.com	Spécialisations : Analyse, Conception et programmation orientées objets, Sécurité des applications, diagnostic de pannes multiples

Below the table is a red bar labeled 'Informations générales du cours proposé'. A detailed course card for 'Initiation à la modélisation UML 2' is shown:

Titre	Thématique	Volume horaire	Nombre de chapitre	Description	Pré-requis	A apprendre	Mots-clés	Date de création
Initiation à la modélisation UML 2	Langage de modélisation d'objet	2 heures / semaine	5	Vous allez développer ou suivre un cours en informatique, et vous souhaitez utiliser UML 2 ou bien modéliser votre projet logiciel avec UML 2 ? Ce MOOC vous donnera toutes les clés pour commencer ce que le langage UML 2 et ses diagrammes. L'objectif de ce cours est de vous présenter le langage UML 2, la fin de ce cours, vous serez capable de réaliser vos premiers diagrammes offrant les fondements de base de votre projet : le contenu, les utilisateurs, les actions et leur déroulement.	Aucun pré-requis n'est nécessaire	Les objectifs du cours sont : Connaitre le langage de modélisation UML. Comprendre la syntaxe des principaux diagrammes offerts.	UML2, orient objet, diagramme acteur, système, classe, objet, action, tat, activité	29/04/17

At the bottom, there are buttons for 'Liens' and 'Supports'.

FIGURE 4.9 – Espace administrateur

CHAPITRE 4. RÉALISATION

Une fois authentifié, un formateur peut accéder à son propre espace. Il peut mettre à jour son profil comme il peut d'ajouter un ou plusieurs cours sur la plateforme en remplaçant le formulaire de description du cours. La figure 4.10 ci-dessous représente la page de profil d'un formateur.

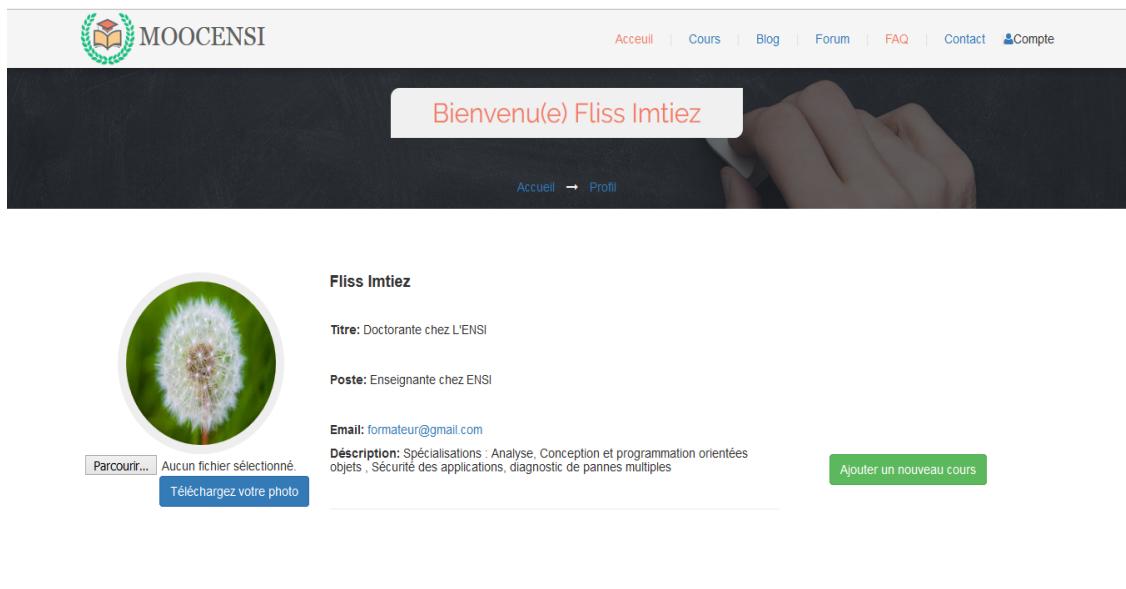


FIGURE 4.10 – Page de profil d'un formateur

La figure 4.11 ci-dessous représente le dashboard d'un formateur via lequel il peut ajouter des cours sur la plateforme en remplissant les différents champs.

The screenshot shows the course creation form on the MOOCENSI platform. The form fields include: Titre (Initiation à la modélisation UML 2), Thématique (Langage de modélisation UML), Charge de travail (2 heures/semaine), Nombre de chapitre (5), and Mode de travail (UML, orienté objet, diagramme, scénario, système, classe, objet, action, état, activité). The 'Description' section contains a detailed text about UML and its diagrams, mentioning that it is a modeling language used to represent the structure and behavior of systems. The 'Pré-requis' section states that no prerequisites are necessary. The 'Ce que l'apprenant va apprendre' section lists objectives such as understanding UML, learning how to model using UML, and grasping the structure of the different models. There are sections for 'Charger vos fichiers' (with three file upload fields) and 'Liens utiles' (with two URL input fields). A blue 'Soumettre le cours' button is at the bottom.

FIGURE 4.11 – Dashboard d'un formateur

CHAPITRE 4. RÉALISATION

Afin de bénéficier des services offerts par notre plateforme, l'apprenant doit d'abord faire une inscription sur la plateforme, comme le montre la figure 4.14 ci-dessous.

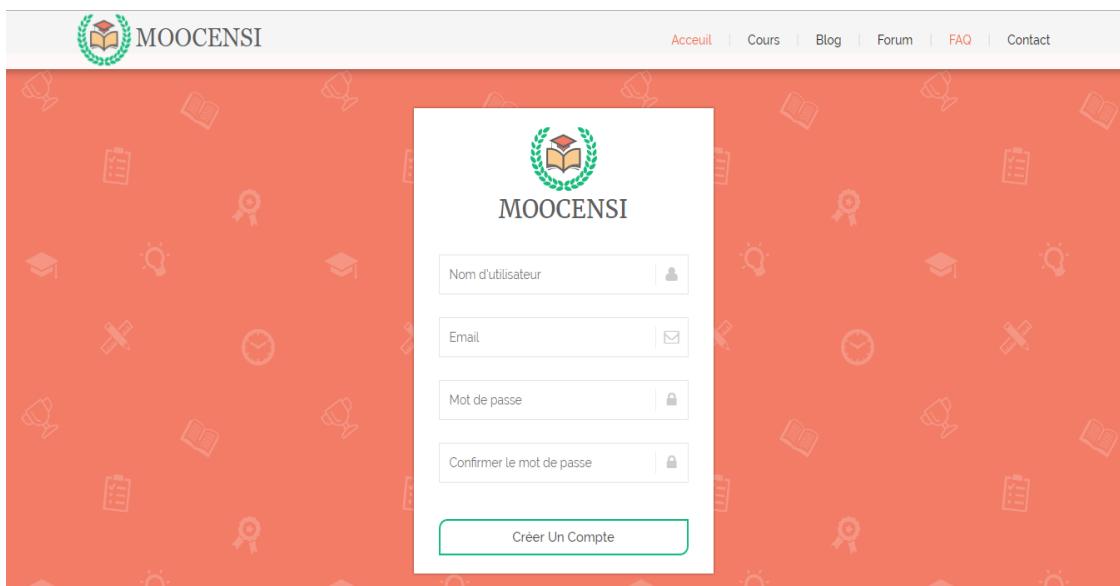
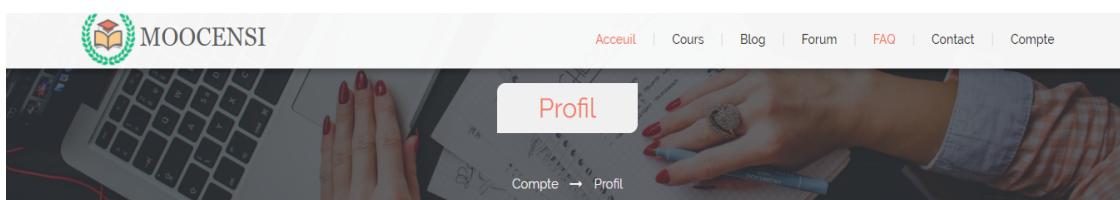


FIGURE 4.12 – Page d'inscription d'un apprenant

Ensuite une fois inscrit et authentifié, il peut accéder à son propre espace. Un apprenant peut mettre à jour ses informations. La figure 4.13 ci-dessous représente la page de profil d'un apprenant.



Bienvenu donia :)

Mettre à jour vos informations

Login	Mot de passe
dolia
Email	
dolia.hammami.92@outlook.fr	
Modifier >	



FIGURE 4.13 – Profil d'un apprenant

CHAPITRE 4. RÉALISATION

Un apprenant peut s'inscrire à un ou plusieurs cours. Une fois inscrit à un cours, il peut télécharger l'ensemble des documents concernant ce cours, passer des quiz et voir son niveau de progression. La figure 4.14 ci-dessous représente la page d'inscription aux cours.

The screenshot shows the MOOCENSI website with the following details:

- Header:** MOOCENSI with a logo featuring a book and laurel wreath.
- Breadcrumbs:** Accueil > Cours > Cours UML
- Title:** Cours UML
- Description:** Vous êtes développeur ou novice en informatique, et vous souhaitez découvrir UML 2 ou bien modéliser votre projet logiciel avec UML 2 ? Ce MOOC vous donnera toutes les clés pour découvrir ce qu'est le langage UML2 et ses diagrammes. L'objectif de ce cours est de vous présenter le langage UML2. A la fin de ce cours, vous serez capable de réaliser vos premiers diagrammes définissant les éléments de base de votre projet : le contexte, les utilisateurs, les actions et leur déroulement.
- Buttons:** S'inscrire au cours
- Navigation tabs:** Description, Objectifs, Programme, Pré-requis, Déroulement, Certification
- Text in box:** UML: (en anglais Unified Modeling Language, "langage de modélisation unifié") est un langage graphique de modélisation des données et des traitements. C'est une formalisation très aboutie et non-propriétaire de la modélisation objet utilisée en génie logiciel.

FIGURE 4.14 – Inscription aux cours

La figure 4.15 ci-dessous représente l'espace d'un apprenant où il trouve l'ensemble des cours qu'il suit, les quiz associés et son niveau de progression dans les cours.

The screenshot shows the apprenant's dashboard with the following details:

- Left sidebar:** Navigation menu with items like A propos le cours, Introduction, Sensibilisation à la modélisation, UML, Quiz, Le diagramme de cas d'utilisation, Motivations, Diagramme des cas d'utilisation, and Description des cas d'utilisation.
- Top bar:** Donia, profile icon, and settings icon.
- Dashboard cards:**
 - NOMBRE TOTAL D'INSCRIPTION: ↑ 0
 - NOMBRE TOTAL DE QUIZ: ↑ 0
 - NOMBRE DE QUIZ ACCOMPLI: ↑ 0
 - NIVEAU DE PROGRESSION: ↓
- Video player:** A video player window titled "voki" showing a woman with glasses and a pearl necklace in a classroom setting.

FIGURE 4.15 – Dashboard d'un apprenant

CHAPITRE 4. RÉALISATION

La figure 4.16 ci-dessous représente un exemple de quiz associé au cours UML suivi par l'apprenant.

The screenshot shows a Moodle quiz interface. On the left, there's a navigation sidebar with categories like 'A propos le cours', 'Introduction', 'Sensibilisation à la modélisation', 'UML', 'Quiz', 'Le diagramme de cas d'utilisation', 'Les diagrammes de classes et diagrammes d'objets', and 'Les diagrammes de séquence'. The main area displays a question titled 'Quel terme ne se rapporte pas à la modélisation d'un diagramme UML ?'. The question has four options: 'Package', 'Objet', 'Entité', and 'Nœud'. At the bottom, there are buttons for 'Answer' and 'Submit All'. A user profile for 'Donia' is visible at the top right. The quiz title is 'chapitre1' and it shows 'Question 1 of 4' with a point value of 10 and total points of 0 out of 40.

FIGURE 4.16 – Quiz

III Chronogramme

Ce travail a été réalisé durant une période de 3 mois. La répartition des tâches durant toute la réalisation de notre application est illustrée par le diagramme de Gantt réel de la figure 4.17 ci-dessous :

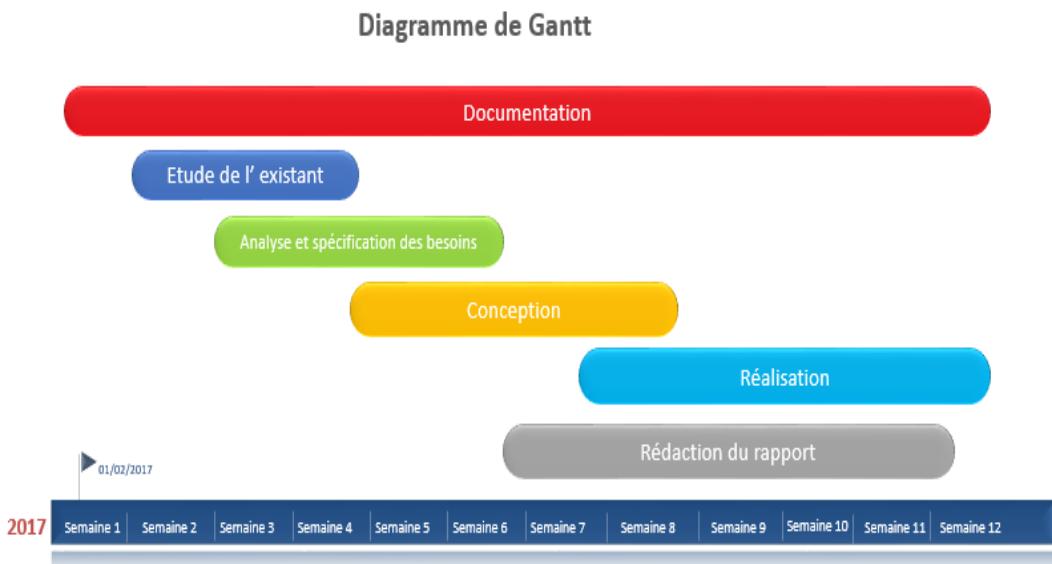


FIGURE 4.17 – Diagramme de Gantt réel

Conclusion

Ce chapitre représente une récapitulation de tout le travail élaboré pendant ce projet de conception et de développement ainsi qu'une présentation des résultats atteints. En effet, nous avons décrit les environnements matériels et logiciels sur lesquels nous avons construit notre application. Nous avons ensuite passé à la présentation de quelques interfaces de notre application et enfin nous avons clôturé ce chapitre par la présentation du chronogramme des tâches.

Conclusion générale et perspectives

Le développement des technologies de l'information et de la communication pour l'éducation (TICE) amène à une utilisation accrue des formations ouvertes et à distance par voie électronique (e-learning). Différents éléments présents dans ces formations sont intégrés afin d'améliorer l'efficacité de l'apprentissage.

C'est dans ce cadre que s'inscrit le présent travail de conception et de développement d'une plateforme d'E-learning ayant pour objectif de faciliter l'apprentissage en ligne plus personnalisée et plus flexible, permettant un apprentissage complémentaire et individualisé, se libérant des contraintes de temps, de lieu, de modalité d'apprentissage et de ressources humaines. L'application mise en oeuvre permet aux différents intervenants d'appliquer les nouvelles approches pédagogiques.

Dans le présent rapport, nous avons détaillé les étapes par lesquelles nous sommes passés pour concevoir et développer notre solution. Pour aboutir à ce résultat, nous avons tout d'abord commencé par présenter le cadre général de notre travail et faire une étude de l'existant. Puis, nous avons présenté les différents besoins et les exigences relevées. Ensuite, nous avons abordé la phase de conception qui nous a expliqué l'architecture de l'application. Finalement, l'étape de réalisation, au cours de laquelle nous avons présenté notre application.

Durant ce projet, nous avons été confrontés à plusieurs problèmes et obstacles au niveau développement. En effet, nous n'avons pu mettre en place l'environnement de développement facilement, nous avons eu des problèmes de compatibilité de IDE Eclipse et du serveur web Tomcat avec nos PC, ce qui nous a causé une perte de temps au début de la phase de réalisation. Nous avons opté au début au framework

CHAPITRE 4. RÉALISATION

SpringMVC mais nous n'avons pas pu faire l'affichage de nos pages jsp. Nous avons donc changé de stratégie et opté pour struts2.

Comme perspectives de travaux futurs, nous proposons d'enrichir cette application en s'intéressant à certains points. Pour étendre notre solution, nous pouvons l'enrichir par d'autres fonctionnalités : nous pouvons faire une partie Android de cette application pour qu'elle soit accessible via le téléphone mobile ou tablette. Nous proposons également de développer un système de notification par email.

Netographie

- [N1] <http://e-learning.prestataires.com/conseils/quest-ce-que-le-learning>, dernière visite le 22/04/2017
- [N2] <http://www.awt.be/web/edu/index.aspx?page=edu,fr/gui,080,010>, dernière visite le 22/04/2017
- [N3] <https://apliut.revues.org/1329>, dernière visite le 22/04/2017
- [N4] <http://www.futura-sciences.com/tech/definitions/tech-telematique-1331/>, dernière visite le 22/04/2017
- [N5] <http://etraintoday.com/>, dernière visite le 22/04/2017
- [N6] <http://www.latrobe.edu.au/students/it/teaching/lms>, dernière visite le 22/04/2017
- [N7] <https://www.1min30.com/dictionnaire-du-web/medias-sociaux>, dernière visite le 22/04/2017
- [N8] <https://www.podcastuisse.ch/quest-ce-quun-podcast>, dernière visite le 22/04/2017
- [N9]
<http://cursus.edu/article/21838/mooc-learning-quelles-differences/>.WPu0gNLhC00>, dernière visite le 22/04/2017
- [N10]
<http://cursus.edu/article/21838/mooc-learning-quelles-differences/>.WPu0gNLhC00>, dernière visite le 22/04/2017
- [N11] <http://moocs.unige.ch/presentation/>, dernière visite le 22/04/2017
- [N12] <http://www.journaldunet.com/management/expert/64562/cooc.shtml>, dernière visite le 22/04/2017
- [N13] <http://www.journaldunet.com/management/expert/64563/spoc.shtml>, dernière visite le 22/04/2017

- [N14] http ://openclassroom.stanford.edu/MainFolder/HomePage.php, dernière visite le 22/04/2017
- [N15] http ://openclassroom.com, dernière visite le 22/04/2017
- [N16] https ://www.edx.org/how-it-works, dernière visite le 22/04/2017
- [N17] https ://www.edx.org, dernière visite le 22/04/2017
- [N18] https ://www.class-central.com/provider/coursera, dernière visite le 22/04/2017
- [N19] https ://www.coursera.org/, dernière visite le 22/04/2017
- [N20] https ://www.fun-mooc.fr/about, dernière visite le 22/04/2017
- [N21] https ://www.fun-mooc.fr/, dernière visite le 22/04/2017
- [N22] http ://www.supinfo.com/articles/single/574-architecture-2-tiers-vs-architecture-3-tiers, dernière visite le
22/04/2017
- [N23] http ://www.supinfo.com/articles/single/574-architecture-2-tiers-vs-architecture-3-tiers, dernière visite le
22/04/2017
- [N24] cedric.cnam.fr/ farinone/SMB111/annee1011/architectureNTiers.html, dernière visite le 22/04/2017
- [N25] https ://delmas-rigoutsos.nom.fr/documents/YDelmas-ArchiWeb/YDelmas-ArchiWeb.html, dernière visite le
22/04/2017
- [N26] http ://mastercomputersystem.com/reseaux.php ?page=res, dernière visite le
22/04/2017
- [N27] https ://www.imaginanet.com/blog/patron-mvp.html, dernière visite le
22/04/2017
- [N28] https ://www.technologies-ebusiness.com/enjeux-et-tendances/modele-darchitecture-angularjs, dernière visite le
22/04/2017
- [N29] https ://labs.ribot.co.uk/approaching-android-with-mvvm-8ceec02d5442, dernière visite le 22/04/2017
- [N30] http ://prof.bpesquet.fr/cours/modele-mvc/, dernière visite le 22/04/2017
- [N31] http ://japf.developpez.com/tutoriels/dotnet/mvvm-pour-des-applications-wpf-bien-architecturées-et-testables/, dernière visite le

22/04/2017

[N32] http ://gardeuxvincent.eu, dernière visite le 22/04/2017

[N33] http ://kmdkaci.developpez.com/tutoriels/java/bien-debuter-avec-struts2/, dernière visite le 22/04/2017

[N34] https ://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm, dernière visite le 22/04/2017

[N35] http ://gardeuxvincent.eu/documents/ProjetJEE, dernière visite le 22/04/2017

[N36] http ://hibernate.org/orm/, dernière visite le 22/04/2017

[N37] https ://www.html5rocks.com/en/, dernière visite le 22/04/2017

[N38] https ://www.w3schools.com/css/css3_intro.asp, dernière visite le 22/04/2017

[N39] http ://www.toutjavascript.com/main/index.php3, dernière visite le 22/04/2017

[N40] https ://www.w3schools.com/jquery/, dernière visite le 22/04/2017