

margin=80pt

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Tunis El Manar

École Nationale d'Ingénieurs de Tunis



Département Technologies de l'Information et de la Communication

Rapport de stage ouvrier (2ème Année)

présenté par

Hcen Tlili
2AINFO2

soutenu le –date– devant le Jury composé de :

<i>Président</i>	Titre. Prénom Nom
<i>Examineur</i>	Titre. Prénom Nom
<i>Rapporteur</i>	Titre. Prénom Nom
<i>Encadrant</i>	Titre. Prénom Nom

Année Universitaire 2017-2018

Signatures

Signature encadrant

Resume

Ce projet consiste à concevoir et à mettre en place un portail de communication académique .Cette solution offre aux utilisateurs une interface d'éducation qui réunit tous les intervenants dans ce processus en leur offrant unemultitude de fonctionnalités.

Elle permet aux enseignants de gérer les cours, de créer des évaluations .Les étudiants ont accès ont accès à ces ressources ainsi qu'à des forums.Un espace d'administration est présent permettant de gérer gérer ces intervenants et de contrôler le processus éducation.Les visiteurs ont accès à un espace public permettant d'informer sur le contenu du MOOC.

Mots-clés : API , REST , MOOC , école , étudiant , enseignant, etc.

Remerciements

Au terme de ce projet de fin d'année , nous tenons à remercier les personnes qui ont contribué à faire de cette période une expérience enrichissante et pleine d'intérêt pour notre avenir professionnel. Nous tenons particulièrement à témoigner toute notre gratitude et notre haute considération à :

Mme. GALLAS Abir pour l'aide qu'elle nous a apportée durant toutes les étapes de ce projet. Sa disponibilité, son encadrement et ses conseils ont été précieux pour atteindre les objectifs du projet dans les délais convenus.

Notre dernier mot s'adresse à tous les membres du jury pour l'honneur qu'ils nous font par la participation à l'examen de ce travail.

Table des matières

Liste des figures	7
Liste des tableaux	8
Liste des acronymes	9
Introduction Générale	10
1 Présentation du cadre du projet	12
1.1 Introduction	12
1.2 Présentation du projet	12
1.2.1 Cadre du projet	12
1.2.2 Problématique	12
1.3 Étude préalable	13
1.3.1 Etude de l'existant	13
1.3.2 Critique de l'existant	15
1.3.3 Solution proposée	16
1.3.4 Méthodologie de travail suivie	16
1.4 Conclusion	16

2	Analyse et spécification des besoins	17
2.1	Introduction	17
2.2	Analyse des besoins	17
2.2.1	Identification des acteurs	17
2.2.2	Besoins fonctionnels	18
2.2.3	Besoins non fonctionnels	20
2.3	Spécification des besoins	20
2.3.1	Vue globale du système	20
2.3.2	Diagrammes de cas d'utilisations raffinés	21
2.3.2.1	Diagramme de cas d'utilisation de l'étudiant	21
2.3.2.2	Diagramme de cas d'utilisation de l'enseignant	23
2.3.2.3	Diagramme de cas d'utilisation de l'administrateur	24
2.4	Conclusion	25
3	Conception	26
3.1	Introduction	26
3.2	Conception globale	26
3.3	Conception détaillée	27
3.3.1	Diagramme de classes	27
3.3.2	Diagrammes de séquences	29
3.3.2.1	Diagramme de séquences de l'authentification	29
3.3.2.2	Diagramme de séquences de l'ajout d'un cours pour enseignant	31
3.3.2.3	Diagramme de séquences de passage d'un test pour étudiant	32
3.4	Conclusion	33

4	Réalisation	34
4.1	Introduction	34
4.2	Environnement de développement	34
4.2.1	Environnement matériel	34
4.2.2	Environnement logiciel	35
4.2.3	Choix technologiques	35
4.3	Travail réalisé et scénarios d'exécution	37
	Conclusion et perspectives	39
	Bibliographie	39
	Netographie	40
5	Annexe A : Suite des interfaces de l'application	42

Liste des figures

1.1	La spécialisation dans l'éducation chez Coursera	13
1.2	Interface d'accueil de l'UVT	14
2.1	Diagramme de cas d'utilisation général	21
2.2	Diagramme de cas d'utilisation de l'étudiant	22
2.3	Diagramme de cas d'utilisation de l'enseignant	23
2.4	Diagramme de cas d'utilisation de l'administrateur	24
3.1	Diagramme de classes	28
3.2	Diagramme de séquence de l'authentification	30
3.3	Diagramme de séquences de l'ajout de cours par enseignant	31
3.4	Diagramme de séquences de passage d'un test pour étudiant	32
4.1	Choix technologiques	36

Liste des tableaux

1.1	Comparatif fonctionnel des solutions existantes	15
2.1	Cas d'utilisation participer à un examen	22
2.2	Cas d'utilisation créer un quiz	23
2.2	Cas d'utilisation créer un quiz	24
2.3	Cas d'utilisation :Envoyer un mail récapitulatif aux responsables par l'ad- ministrateur	25
3.1	Détails de l'architecture trois tiers	27
4.1	Environnement matériel	35

Liste des acronymes

API Application Programming Interface

CSS Cascading Style Sheet

ENIT Ecole Nationale d'Ingénieurs de Tunis

HTML Hypertext Markup Language

HTTP HyperText Transfer Protocol

JSON JavaScript Object Notation

MVC Model-View-Controller

NoSQL Not only Structured Query Language

ORM Object-Relational Mapping

PFA Projet de Fin d'Année

REST Representational State Transfer

SMTP Cascading Style Sheet

UML Unified Modeling Language

Introduction Générale

Depuis des années , la formation et l'éducation numérique ont connu une croissance impressionnante . C'est à l'aide des portails de communication académique massive MOOC que cela a progressé .

Un MOOC respresente l'un des moyens les plus efficaces pour transmettre un enseignement de qualité au plus grand nombre d'étudiants. Ils représentent l'occasion de réunir des milliers de personnes motivées. Ils constituent une opportunité d'appliquer les principes de l'intelligence collective. Les interactions durant le cours permettent de faire émerger des communautés autour des thèmes abordés au cours du MOOC facilitant les rencontres entre les acteurs du domaine abordé et encourageant les collaborations .

La communication académique numérique est désormais un apport au processus éducatif classique losqu'elle est conçue pour reconstruire la structure des établissement réels et d'automatiser les interactions entre ses intervenants.C'est dans ce contexte, que s'inscrit notre projet de fin d'année.En effet, nous visons à concevoir et à développer une application Web et Mobile incluant les différentes fonctionnalités d'administration et d'utilisation d'un MOOC .

Ce rapport est subdivisé en quatre chapitres. Le premier chapitre présente le cadre du projet et le sujet à traiter. Nous y intégrons une étude préalable ainsi que la méthodologie de travail. Le deuxième chapitre est consacré à une analyse des besoins et à la formalisation de la spécification du projet ce qui permet de procéder à la représentation de l'interaction de l'utilisateur avec l'application. Le troisième chapitre apporte une démonstration de la conception adoptée pour répondre aux besoins de l'utilisateur du MOOC. Le dernier chapitre traite les détails de la réalisation : les outils de travail, les différentes étapes de la réalisation du portail ainsi que des captures d'écran illustrant les fonctionnalités principales de notre application. Dans la conclusion générale, nous passons en revue notre apport aussi bien que les perspectives d'avenir dans ce travail.

Chapitre 1

Présentation du cadre du projet

1.1 Introduction

Ce premier chapitre est consacré pour la description du contexte général du projet ainsi qu'à une étude de l'existant , la présentation de notre proposition et de la méthodologie de travail.

1.2 Présentation du projet

Dans cette section nous allons introduire le projet d'une manière globale en détaillons la problématique qui nous a poussé à travaillé sur ce sujet.

1.2.1 Cadre du projet

Ce projet rentre dans le cadre du projet de fin d'année, effectué durant la deuxième semestre. Il est intitulé "conception et développement d'un portail de communication académique (Web / Mobile)" .

1.2.2 Problématique

Le travail que nous avons à réaliser est la conception et le développement d'une plateforme informatique réunissant les enseignants , les étudiants ainsi que des adminis-

trateurs. L'enseignant, de sa part gère sa matière, les classes et les étudiants qui sont en affectés. Il est capable de contrôler le contenu du cours, gérer les évaluations en matière de contenu et de notes. L'étudiant est capable de la consultation des cours attribués à sa classe, passe les évaluations et dispose des forums des discussions. L'administrateur, quant à lui, dispose de d'une vue générale sur ce que se passe dans la plateforme, il gère aussi les autres utilisateurs.

1.3 Étude préalable

L'analyse de l'existant comprend principalement trois parties qui sont la description de l'existant, ses limites et la solution proposée.

1.3.1 Etude de l'existant

Après une recherche sur les solutions existantes, fonctionnelles en ligne et qui proposent un service similaire, nous pouvons décrire les principales entre elles comme suit :

- **Coursera** : Plateforme numérique lancée en 2012 comportant plus de 2 millions d'étudiants. L'entreprise partage la recette avec les universités partenaires. Elle se concentre sur un programme de focalisation rassemblant des cours sur un domaine spécifique renforçant la notion de spécialisation. "Coursera propose aussi aux utilisateurs de se perfectionner dans un domaine en suivant plusieurs cours rassemblés dans des programmes dits de 'spécialisation' [N1]." . Ci-dessous la figure 1.1 illustrant cette propriété :

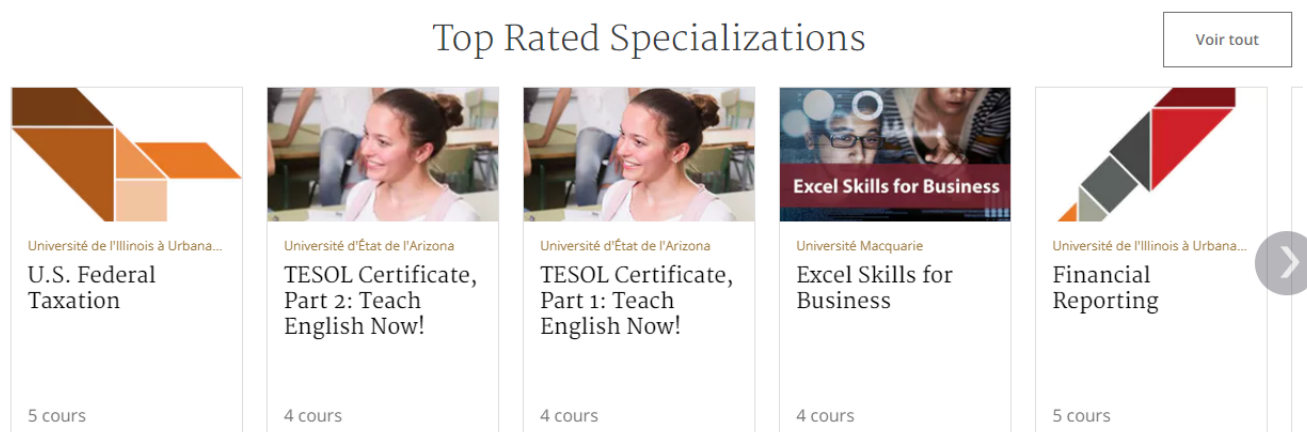


FIGURE 1.1 – La spécialisation dans l'éducation chez Coursera

- **Openclassrooms** : Le site offre des cours et des tutoriels plus particulièrement aux débutants dans les domaines du développement informatique et de marketing. "La communauté OpenClassrooms compte aujourd'hui plus de 1 million de membres et ce sont près de 3 millions d'utilisateurs qui étudient chaque mois sur le catalogue de plus de 1000 cours en ligne, dont les principaux sont désormais sous forme de MOOC [N2]." Les cours sont élaborés par chacun des utilisateurs sans spécification d'accès. Openclassrooms propose un éditeur de cours qui est pré-traité et jugé par une équipe pédagogique avant d'être publié.

- **FUN** : Le cours dans France Université Numérique est généralement dispensé par un enseignant face caméra. Des quiz finals sont proposés ainsi qu'à des examens à distances permettant d'obtenir des attestations de réussite. "Le catalogue de cours disponibles s'est continuellement enrichi pour proposer une variété de formations répondant aux besoins de tous les publics. Nous sommes partis d'une offre d'une dizaine de cours sur quelques thématiques [N3]".

En Tunisie, "L'université virtuelle de Tunis UVT" est la plateforme la plus connue dans ce domaine. "Elle a pour principale mission de développer des cours et des programmes universitaires d'enseignement en ligne pour les universités tunisiennes [N4]". Dans celle-ci, le processus éducatif s'étale sur une période de temps prédéfinie et il est partagé par l'administrateur du site sans intervention de l'éditeur du cours. On constate l'absence d'évaluations dans l'UVT. Ci-dessous la figure 1.2 représentant l'interface d'accueil de l'UVT :

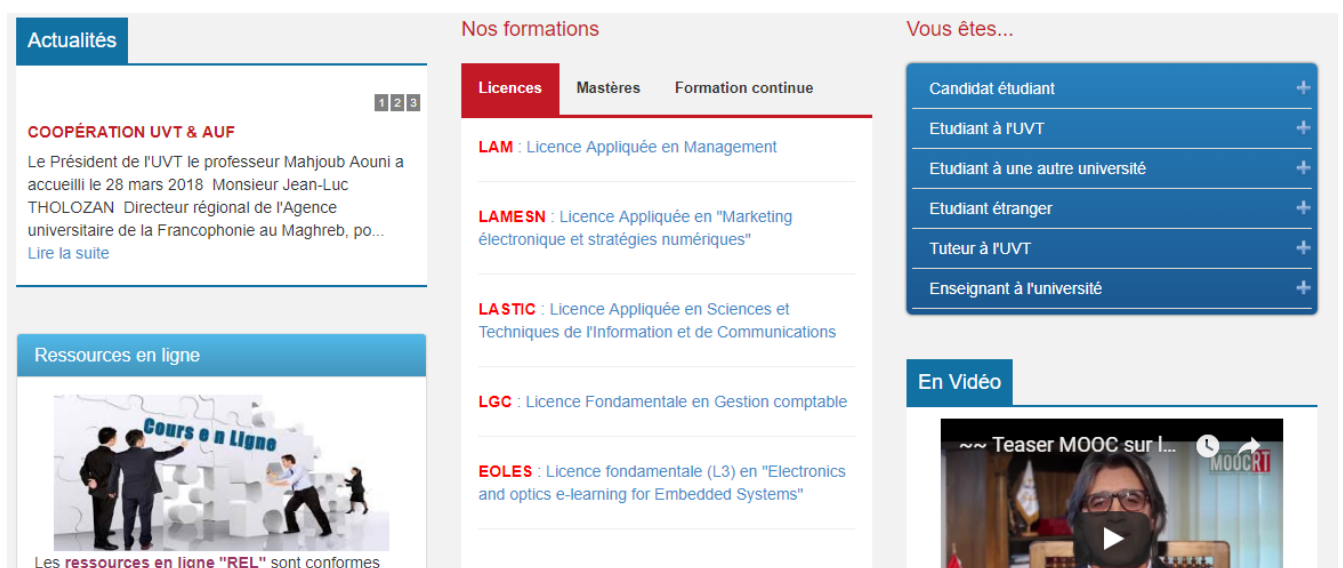


FIGURE 1.2 – Interface d'accueil de l'UVT

1.3.2 Critique de l'existant

Tout d'abord , nous allons détailler les différences entre les plate-formes citées précédemment à travers le tableau suivant :

TABLE 1.1: Comparatif fonctionnel des solutions existantes

	Coursera	OpenClassrooms	FUN	UVT
Plateforme	WEB	WEB	WEB	WEB
Supervision		X	X	X
Evaluations	X	X	X	
Forum		X		
Partage de fichiers	X		X	
Contact par email	X	X	X	

Bien que les outils cités précédemment sont utiles , nous avons découvert beaucoup d'idées innovantes dans ce contexte.

Des études statistiques ont montré que entre le nombre initial d'inscrits et le nombre d'étudiants qui passent l'évaluation terminale, les pertes sont considérables : 70% ne suivent pas régulièrement ou de tout le cours. Le nombre d'utilisateurs " lerkers " qui picorent du MOOC juste une petite partie est énorme . Cela montre que ce processus ne peut pas ressembler à l'éducation à distance , ça diffère juste par le critère massif vu que les enseignants se trouvent face à un nombre d'étudiants qui dépasse très largement le nombre qu'ils sont habitués à enseigner.

La structuration de ces plateformes doit changer pour pousser les étudiants à interagir et à compléter leurs processus éducatifs. Il se peut que changer les privilèges et inclure beaucoup plus les enseignants en interactions avec les étudiants et spécifier ceux auxquels ils sont responsable va rendre leur performance plus grande . Les étudiants vont sentir appartenance à la plateforme en s'intégrant dans une classe indépendante ayant ses propres ressources et évaluations.

1.3.3 Solution proposée

On propose notre solution , objet de ce projet ,qui est de partager les fonctionnalités d'un portail de communication académique regroupant les acteurs :administrateur , enseignant et étudiant , d'une manière plus proche du modèle de l'éducation réelle classique dans un contexte massif :

Cette plateforme se compose d'une partie web : Espace d'administration assurant la gestion des étudiants et des enseignants , les affectations ainsi que la supervisions en terme d'obtention de statistique sur les cours et de collaboration avec les responsables par messagerie, etc . L'espace des enseignants leur permettant de gérer leurs classes , de répondre à leurs questions , d'intégrer les discussions sur les cours et de créer les évaluations et de consulter les réponses des étudiants. Ces derniers auront la possibilité de consulter les cours et de passer les évaluations.

La partie web contient pour les étudiants un accès aux cours et aux forums à propos de chacun de ceux-ci.

1.3.4 Méthodologie de travail suivie

Passant par plusieurs étapes , la réalisation du projet en question demande l'adoption d'une méthodologie de travail . Nous avons choisi le modèle en cascade .

Ce modèle oblige le choix de dates fixes pour la terminaison des étapes. Les phases sont toutes réalisées les unes après les autres avec un retour aux étapes précédentes voire au début. On était ainsi demandé de réaliser des livrables prédéfinis et satisfaisants aux étapes de validation-vérification .

1.4 Conclusion

Au cours du chapitre courant , nous avons présenté le contexte général du projet à savoir son cadre et la problématique posée . Nous avons détaillé l'état de l'art et de l'existant ainsi que la méthodologie de travail suivie tout au long de la réalisation. Le chapitre suivant va être consacré pour l'analyse des besoins fonctionnels et non fonctionnels et leurs spécifications.

Chapitre 2

Analyse et spécification des besoins

2.1 Introduction

Le premier chapitre était l'occasion pour donner le contexte du projet . Le présent chapitre sera consacré pour analyser et spécifier les besoins de l'application . Nous allons commencer par détailler les besoins fonctionnels puis ceux non fonctionnels. En suite nous citerons les acteurs et on finira par spécifier tous les besoins par les diagrammes des cas d'utilisations.

2.2 Analyse des besoins

Cette partie se focalise sur les détails et les contraintes qu'on doit étudier pour construire un système répondant au besoins des utilisateurs finaux. Il s'agit de reproduire ses comportements à travers une description des spécifications à développer d'un point de vue utilisateur.

2.2.1 Identification des acteurs

Les acteurs sont les utilisateurs capables d'accéder aux données de l'application. L'accès authentifié est cédé aux étudiants , enseignants et administrateurs de la plateforme.

- **Etudiant** : L'acteur auquel l'application est destiné en premier lieu et le plus concerné de ses services. Il ont l'accès au données selon leurs privilèges et classification.
- **Enseignant** : l'apport de l'application est d'intégrer l'enseignant dans ce portail comme un intervenant principal et non pas comme un inspecteur ne gérant pas le contenu . L'enseignant prépare le contenu pédagogique du MOOC : cours , examens et quizzes ,ainsi que le suivi des évaluations et les réponses au questions .
- **Administrateur** : Gérant du portail , assurant la structuration des classes et l'affectation des utilisateurs si c'est demandé. Possédant un dashboard illustrant le contenu et des notifications sur l'application , il est capable de transmettre ces informations aux responsables des institutions universitaires.

2.2.2 Besoins fonctionnels

La plate-forme est composée d'une première partie WEB principale dans laquelle existent les principales fonctionnalités destinée aux différents utilisateurs. La partie mobile concerne seulement les étudiants pour accéder aux cours et et aux forums.

L'authentification est permise de manière sécurisé pour accéder aux services.La consultation et la modification du profil sont possibles.Nous avons détaillé les besoins associés à chaque utilisateur :

Administrateur

- Responsable de la gestion globale de l'application . Il gère les étudiants et les comptes d'accès des enseignants.
- Chargé de la création des groupes d'études , affecte les étudiants qui demandent une inscription aux classes convenues et rejette d'autres.Il crée des classes , y affecte des enseignants . Un étudiant appartient à une seule classe , celle-ci contient plusieurs étudiants.
- Capable de gérer les matières enseignées , il arrange les cours accessibles par chacune des classes.
- Il substitue l'enseignant dans l'ajout des cours pour des classes spécifiques en cas de besoins.
- Contacte les responsables sur les classes choisies en leurs envoyant par mail des récapitulations sur l'activité de leurs étudiants et enseignants durant les périodes passées.
- Répondre aux questions des visiteurs de la page d'accueil du portail pour lesquelles est

consacré un espace de dépôt de questions.

Étudiant

- Un étudiant est caractérisé par un identifiant , un nom , un prénom , son nom d'utilisateur ainsi que le numéro de téléphone , son e-mail .
- Un étudiant appartient à une seule classe , ses collègues peuvent être d'une même institution ou d'institutions différentes selon le choix de l'administrateur qui affecte les étudiants.
- Il peut modifier son profil public.
- L'étudiant doit être capable d'accéder à toutes les ressources relatives à la classe à laquelle il appartient. Il peut consulter le cours sur la plate-forme ou le télécharger directement.
- Capable d'accéder aux profils publics de ses collègues.
- Poser des questions et répondre à autres concernant chaque cours.
- Passer des évaluations sous formes de Quizzes ou d'examens question-réponse . Dans le cas de quiz et qui concerne une partie du cours , la note obtenue est enregistrée directement dans l'historique d'évaluations de l'étudiant.Si c'est un examen , l'enseignant éditeur de cet examen sera notifié de la réponse de l'étudiant.

Enseignant

- Un enseignant est caractérisé par son identifiant , un nom , un prénom ,son email et son numéro de téléphone . Il est affecté à une ou plusieurs classes en ce qui concerne une ou plusieurs matières.
- Il peut consulter les profils des étudiants qu'il enseigne en consultant la liste des classes concernées par sa matière.
- Capable de gérer son cours , en ajout et suppression tout en spécifiant les classes concernées par ces apports.
- Il est invité à saisir les questions et les réponses ainsi que de spécifier les propositions correctes des questions à choix multiples qui concernent des parties du cours pré-déposé.
- Ajouter des étudiants aux classes qu'il choisit.
- Des examens entiers sont saisis par cet utilisateur , qui seront passés par les étudiants concernés.
- Suite à l'achèvement d'une activité par un étudiant, l'enseignant sera notifié du travail effectué par l'élève que ce soit par la note qu'il a obtenu lors d'un questionnaire ou du contenu de sa réponse à propos de l'examen.
- Répondre aux questions des étudiants dans les forums concernées et qui sont dans la

partie WEB et mobile.

2.2.3 Besoins non fonctionnels

- **Ergonomie** : Tel que son nom l'indique , un endroit de communication académique massive est utilisé énormément par des gens qui ne sont pas forcément habitués . Pour cela les interfaces doivent être facilement exploitables pour les différents acteurs , ergonomiques et conviviales.
- **Sécurité** : Une phase d'authentification précède l'accès à la plateforme. Les informations et les ressources sont confidentielles et ils sont cryptés lors de l'authentification (Mot de passe et nom d'utilisateur). La facilité d'administration du portail permettant une utilisation sécurisée pour les autres utilisateurs.
- **Environnement et architecture** :L'application doit être conçue de façon à permettre l'accès par un navigateur web et par une application mobile pour différentes fonctionnalités .
- **Rapidité** :L'accès aux données doit être rapide vu l'importance de cet aspect dans l'éducation académique.

2.3 Spécification des besoins

Nous allons opter pour des diagrammes de cas d'utilisation pour spécifier les besoins de chacun des utilisateurs.

2.3.1 Vue globale du système

Dans cette partie nous allons décrire le diagramme de cas d'utilisation dans la figure 2.1 global pour comprendre le fonctionnement du système :

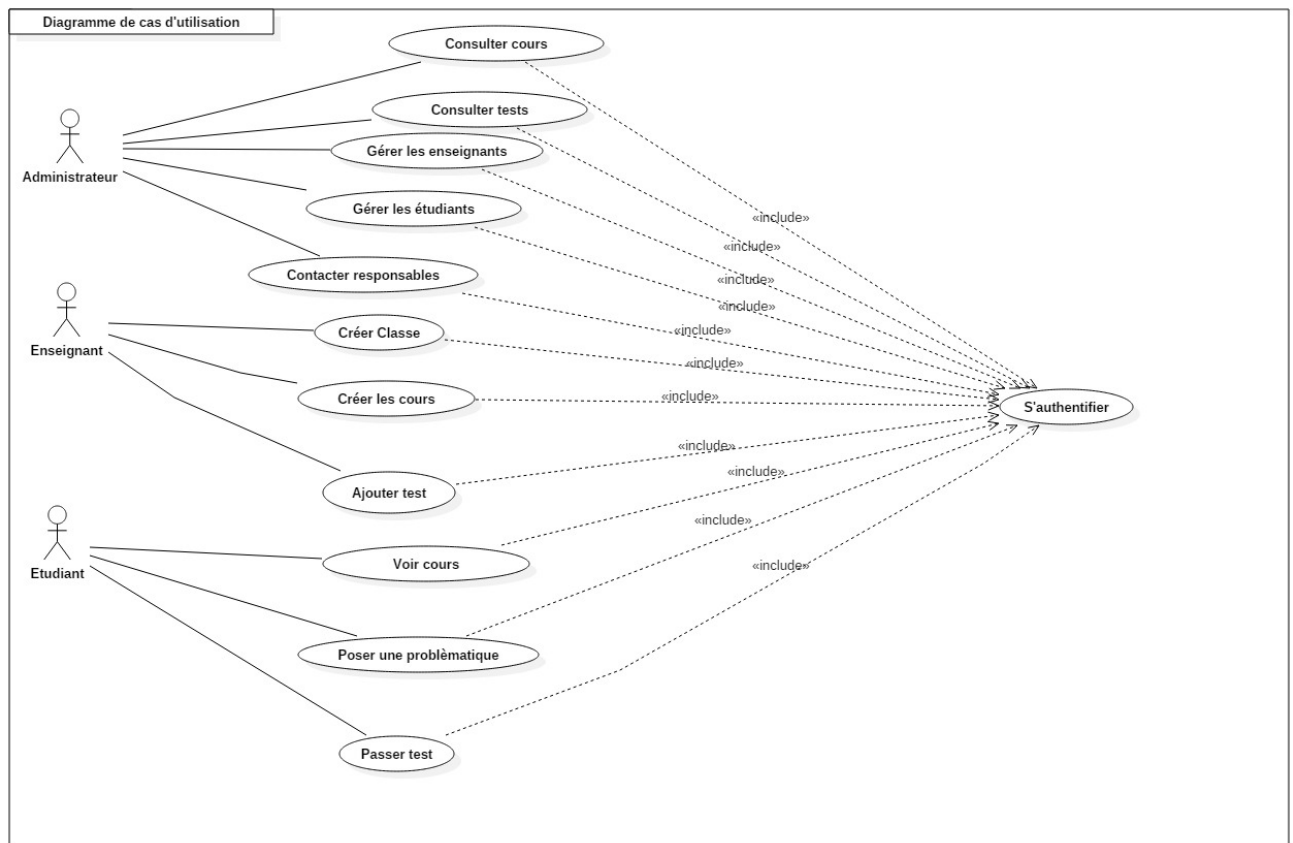


FIGURE 2.1 – Diagramme de cas d'utilisation général

2.3.2 Diagrammes de cas d'utilisations raffinés

Dans ce qui suit , nous allons détailler les diagrammes de cas d'utilisation pour chaque acteur ." En terme d'UML ce sont les cas d'utilisations qui représentent les ensembles de séquences d'actions qu'un système réalise et produisant un résultat observable pour un certain acteur particulier [1]" :

2.3.2.1 Diagramme de cas d'utilisation de l'étudiant

Nous présentons , à travers la figure 2.2 ci-dessous , le diagramme de cas d'utilisation relatif à l'étudiant :

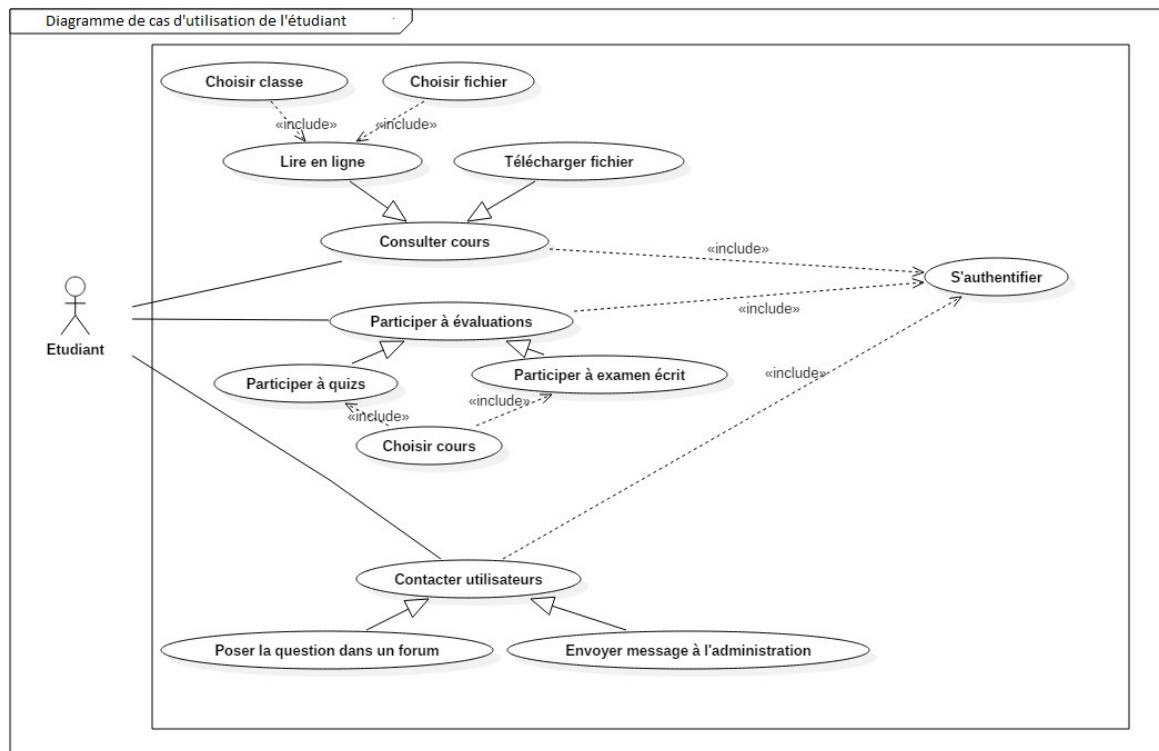


FIGURE 2.2 – Diagramme de cas d'utilisation de l'étudiant

Raffinement de la participation à un examen écrit par étudiant : Détails :

TABLE 2.1: Cas d'utilisation participer à un examen

But	Participation à un examen par l'étudiant
Acteur	Étudiant
Pré-condition	L'étudiant est identifié par son login et son mot de passe .
Post-condition	La réponse de l'étudiant est enregistrée et l'enseignant est notifié par cette action.
Scénario principal	<p>Le système affiche un tableau de choix de l'examen parmi ceux existant.</p> <p>L'étudiant clique sur le bouton de choix de l'examen.</p> <p>Le système affiche Le contenu de l'examen ainsi que la zone de saisie de la réponse.</p> <p>L'étudiant saisie sa réponse dans la zone appropriée.</p> <p>L'étudiant valide sa réponse.</p>
Exception	En cas de saisie d'une réponse vide , elle ne sera pas enregistrée.

2.3.2.2 Diagramme de cas d'utilisation de l'enseignant

La figure 2.3 présente en détails le diagramme de cas d'utilisation relatif à l'enseignant :

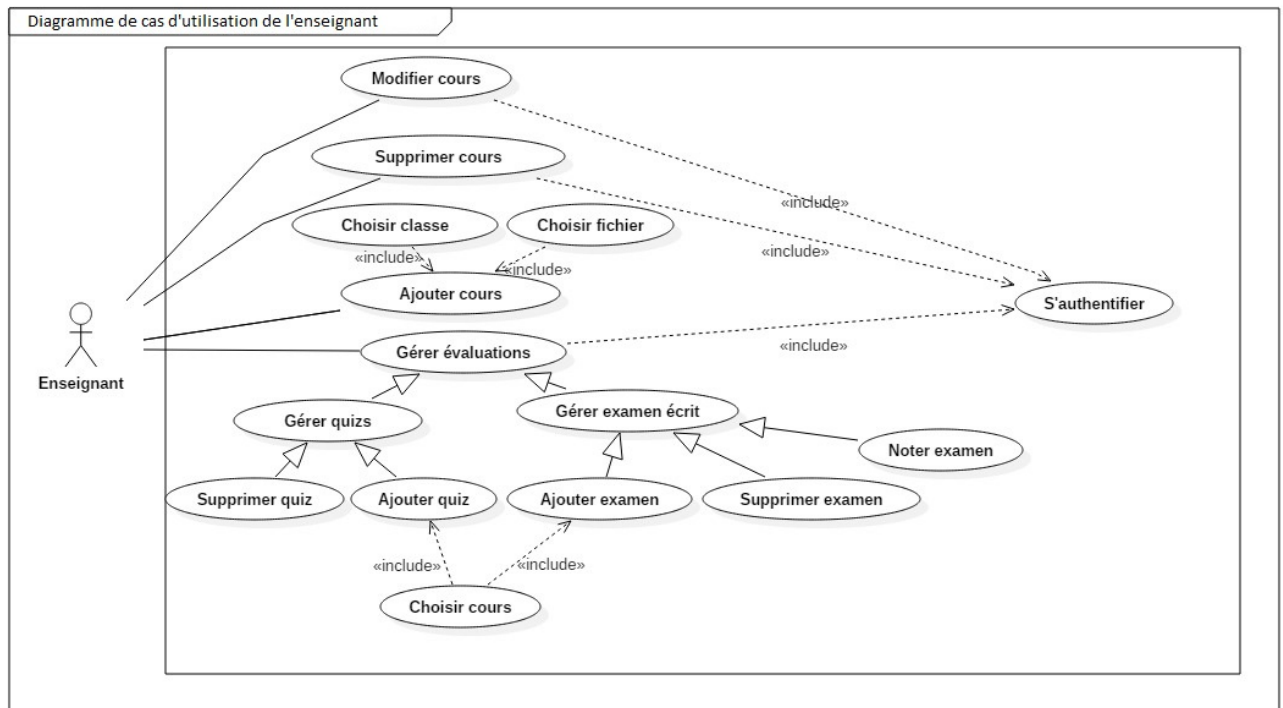


FIGURE 2.3 – Diagramme de cas d'utilisation de l'enseignant

Les détails du cas d'utilisation de création d'un quiz sont exposés ci-dessous :

TABLE 2.2: Cas d'utilisation créer un quiz

But	Enregistrement d'un test "questions à choix multiples" par enseignant.
Acteur	Enseignant
Pré-condition	L'enseignant est identifié par son login et son mot de passe .
Post-condition	L'examen de type questions à choix multiples est enregistré et accessible par les classes concernées.

TABLE 2.2: Cas d'utilisation créer un quiz

Scénario principal	<p>Le système affiche la liste des quizzes pré-saisis .</p> <p>L'enseignant choisie le bouton d'ajout de quiz .</p> <p>Le système affiche une interface d'ajout des questions et des propositions ainsi que de la proposition correcte pour permettre de comparer les réponses par la suite par le système.</p> <p>L'enseignant valide ce qu'il a saisi.</p> <p>Le système enregistre cette instance de quiz.</p>
Exception	<p>Si les questions sont mal saisies le système redirige l'enseignant vers la page de saisie sans enregistrement du quiz .</p>

2.3.2.3 Diagramme de cas d'utilisation de l'administrateur

La figure 2.4 suivante présente le diagramme de cas d'utilisation relatif à l'administrateur :

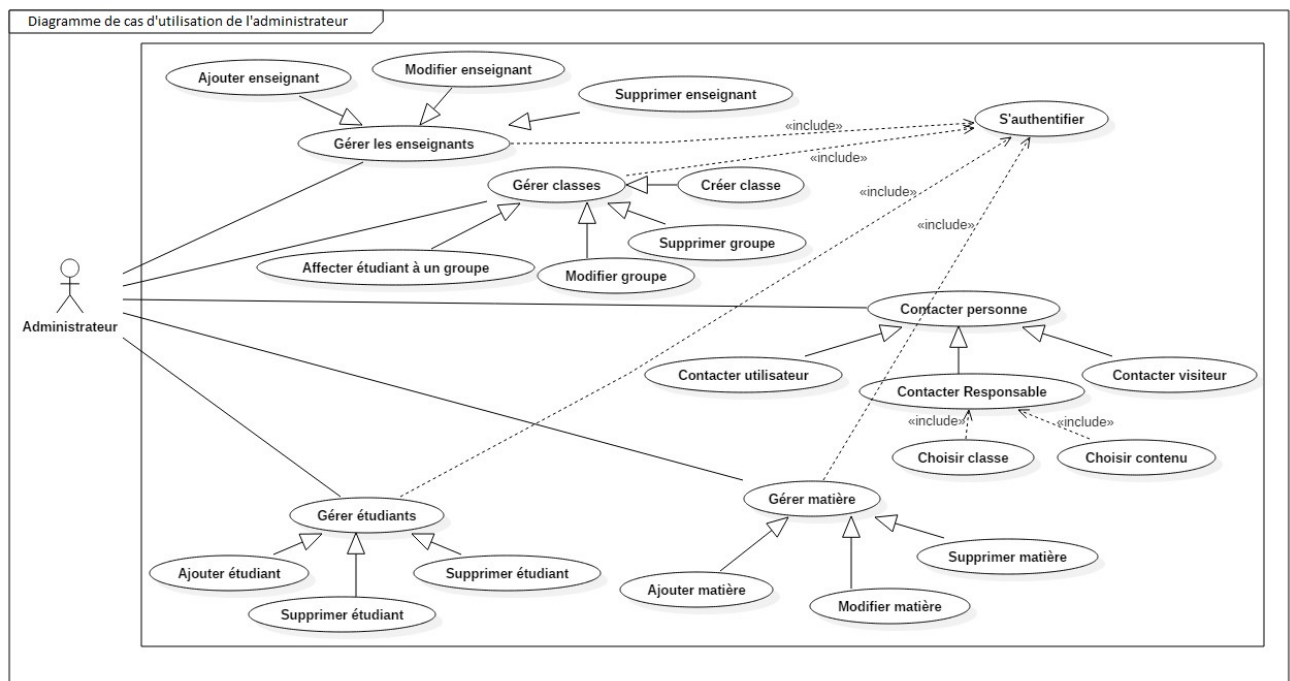


FIGURE 2.4 – Diagramme de cas d'utilisation de l'administrateur

Dans le tableau qui suit nous allons détaillé le cas d'utilisation " contacter les responsables

sur les classes par l'administrateur avec envoie d'un mail récapitulatif sur les activités des étudiants concernés ".

TABLE 2.3: Cas d'utilisation :Envoyer un mail récapitulatif aux responsables par l'administrateur

But	Envoie du mail récapitulatif aux responsables concernés.
Acteur	Administrateur
Pré-condition	L'administrateur est identifié par son login et son mot de passe .
Post-condition	L email est envoyé aux responsables incluant les informations choisies.
Scénario principal	<p>Le système affiche l'interface de choix du contenu du mail .</p> <p>L'administrateur saisie l'adresse email du responsable à contacter .</p> <p>L'administrateur choisi le fait d'inclure une description du portail dans le corps du mail ou non.</p> <p>L'administrateur choisi les classes concernées , c'est à dire celle que les informations à propos de ses étudiants , leurs notes dans les évaluations vont êtres incluses dans le fichier rattaché au mail.</p> <p>L'administrateur valide ses choix et le mail est envoyé.</p>
Exception	Si l'adresse mail est incorrecte l'administrateur est redirigé ver la page de saisie .

2.4 Conclusion

Ce chapitre nous a éclairé sur les besoins du projet. Il a été dédié à l'analyse et la spécification de celles-ci pour comprendre les objectifs du projet et de se focaliser sur ses fonctionnalités principales. Ceci nous permettra de bien concevoir la solution ,comme détaillé dans le chapitre suivant, afin de dénouer la complexité de la réalisation.

Chapitre 3

Conception

3.1 Introduction

La conception est une étape importante dans le cycle de développement. Elle a pour but de définir l'architecture du système en question. Elle définit la manière avec laquelle on arrive à implémenter une plate-forme répondant aux besoins cités précédemment. Nous nous intéressons dans le présent chapitre à détailler la conception globale et la conception détaillée à travers des diagrammes UML statiques et dynamiques.

3.2 Conception globale

On doit disposer , dans la réalisation de l'application d'un bon niveau de modularité. Cela dans l'intérêt d'améliorer l'aspect de modularité des éléments de l'application. Pour cela , nous avons opter pour l'architecture multi-tiers qui répond à ce besoin. Celle-ci est un modèle logique d'architecture applicative visant à structurer l'application comme un empilement de plusieurs couche chacune communiquant avec la supérieure et l'inférieure à travers des interfaces. les fonctionnalités de chacune d'entres elles peuvent évoluer sans induire des changements dans les autres couches.

Dans le tableau suivant , on détaillera le rôle de chacune des trois couches :

TABLE 3.1: Détails de l'architecture trois tiers

Couche	Détails
Présentation	C'est la couche client , dans notre cas elle est représentée principalement par le client WEB et en second lieu la partie mobile. Elle se charge du traitement de l'interaction avec l'utilisateur.
Métier	Chargée du traitement des requêtes provenant de la couche client , elle est responsable de la communication entre la couche présentation et accès aux données. Elle effectue ainsi les traitements applicatifs.
Accès aux données	Permet la gestion des données dans la base (ajout , suppression ,modification ,etc...).

Ce modèle est une type de l'architecture Client/Serveur qui a pour but de séparer la partie présentation de la partie traitement à travers l'envoi des services applicatifs du serveur au client , dans notre cas , des RESTful web services . .

3.3 Conception détaillée

Cette partie sera dédiée pour la conception de l'application à travers des diagrammes en langage UML représentant les différents modules .

3.3.1 Diagramme de classes

Commençant par le diagramme de classes qui donne une vue statique du système . Celui-ci illustre les entités , leurs attributs , les opérations ainsi que les associations entre elles. Ces entités représentent les classes existantes dans la base de données , nous allons les présenter en diagramme de classes dans la figure 3.1 ainsi qu'en des schémas relationnels.

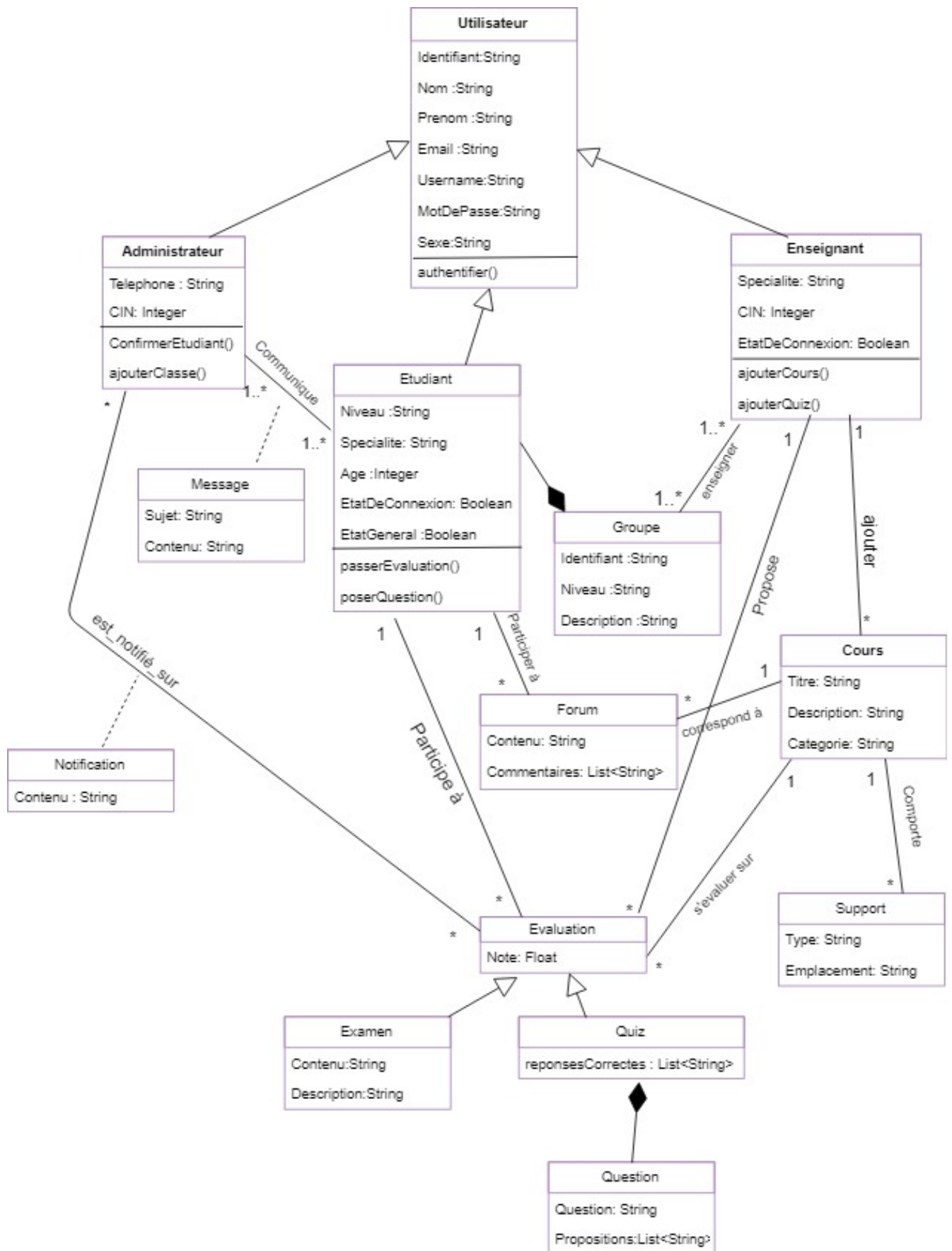


FIGURE 3.1 – Diagramme de classes

L'accès aux différents éléments de la base devient plus aisé avec l'utilisation du modèle relationnel." Le but initial de ce modèle était d'améliorer l'indépendance entre les données et les traitements. [2]."

-Schémas relationnels

- Utilisateur (id, identifiant, nom, prenom, nom_utilisateur, passe, email, photo, sexe)
- Administrateur(#id¹, telephone, CIN)
- Enseignant(#id¹, specialite, CIN, etat_de_connexion)
- Etudiant(#id¹, niveau, specialite, age, etat_de_connexion, etat_general, #groupe_id)
- Groupe(id, identifiant, description)
- Cours(id, titre, description, categorie, #enseignant_id)
- Evaluation(id, note, #cours_id, #etudiant_id, #enseignant_id)
- Examen(#id², contenu, description)
- Quiz(#id², reponses_correctes)
- Question(id, contenu, propositions, #quiz_id)
- Forum(id, contenu, commentaires, #etudiant_id, #cours_id)
- Notification(id, contenu, #administrateur_id, #evaluation_id)
- Message(id, sujet, contenu, #etudiant_id, #administrateur_id)
- Support(id, type, emplacement, #cours_id)

1 : fait référence à la colonne id de la table Utilisateur.

2 : fait référence à la colonne id de la table Evaluation.

3.3.2 Diagrammes de séquences

Les diagrammes de séquences donnent une vue dynamique du système . En effet , ils illustrent les interactions entre l'utilisateur et l'application . Dans cette section , nous allons présenter quelques scénarios d'exécution à travers ce type de diagrammes.

3.3.2.1 Diagramme de séquences de l'authentification

Commencant par l'activité qui précède chaque action chez les acteurs principaux qui est l'authentification.

-Description :

Afin de s'authentifier , l'utilisateur(Etudiant, Enseignant ou Administrateur) doit choisir l'interface correspondante de la page d'accueil. Après saisie de son identifiant et son mot de passe , les données seront envoyées au contrôleur de l'authentification qui les envoie à son tour à la table correspondante(Etudiant, Enseignant ou Administrateur) afin de vérifier l'existence d'un utilisateur . Si la réponse est positive il va être redirigé vers ses interfaces privés sinon l'interface d'authentification va être affichée . Il est représenté dans la figure 3.2 :

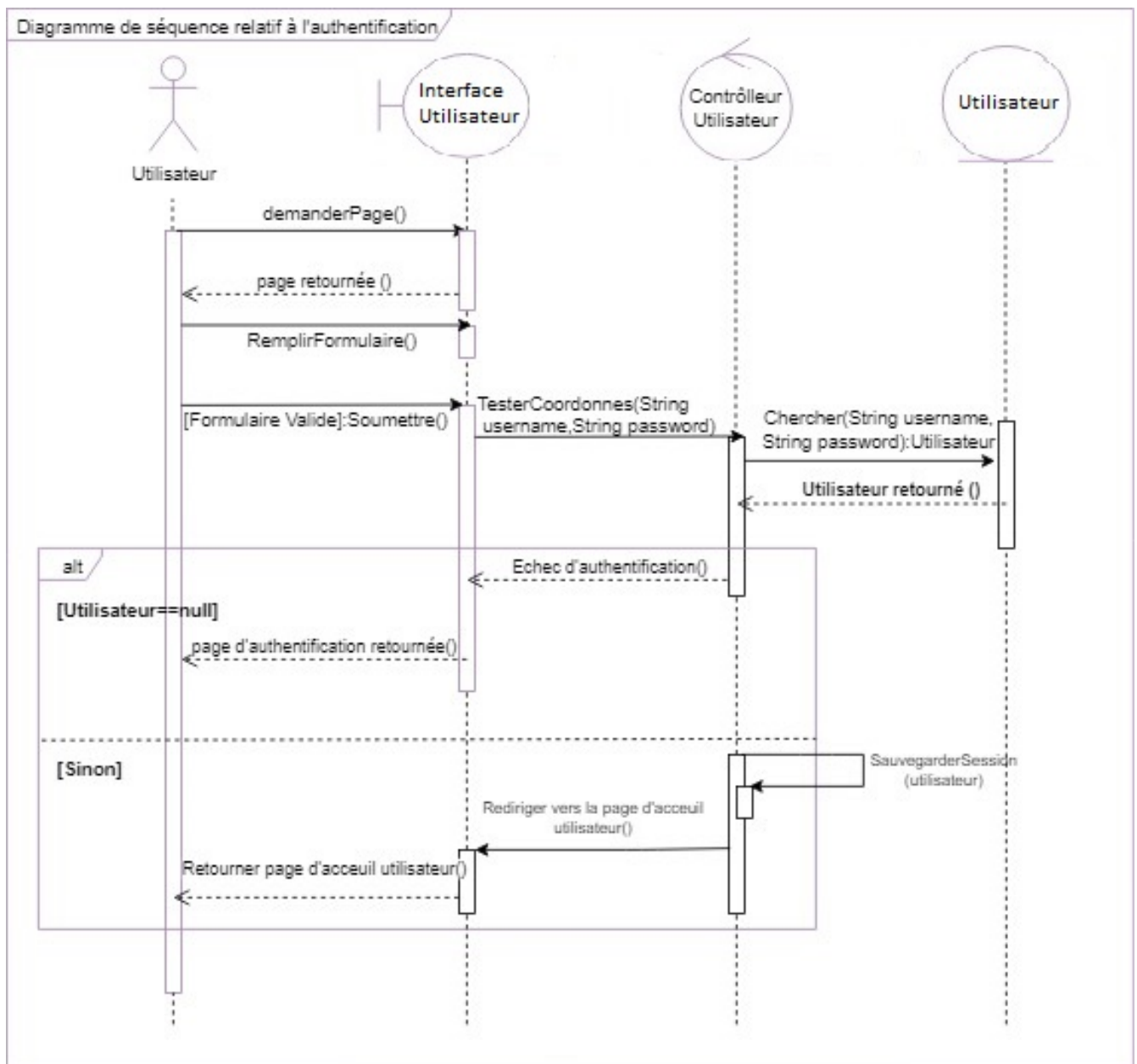


FIGURE 3.2 – Diagramme de séquence de l'authentification

3.3.2.2 Diagramme de séquences de l'ajout d'un cours pour enseignant

Le diagramme de séquence correspondant à chaque action est illustré par la figure 3.3 :

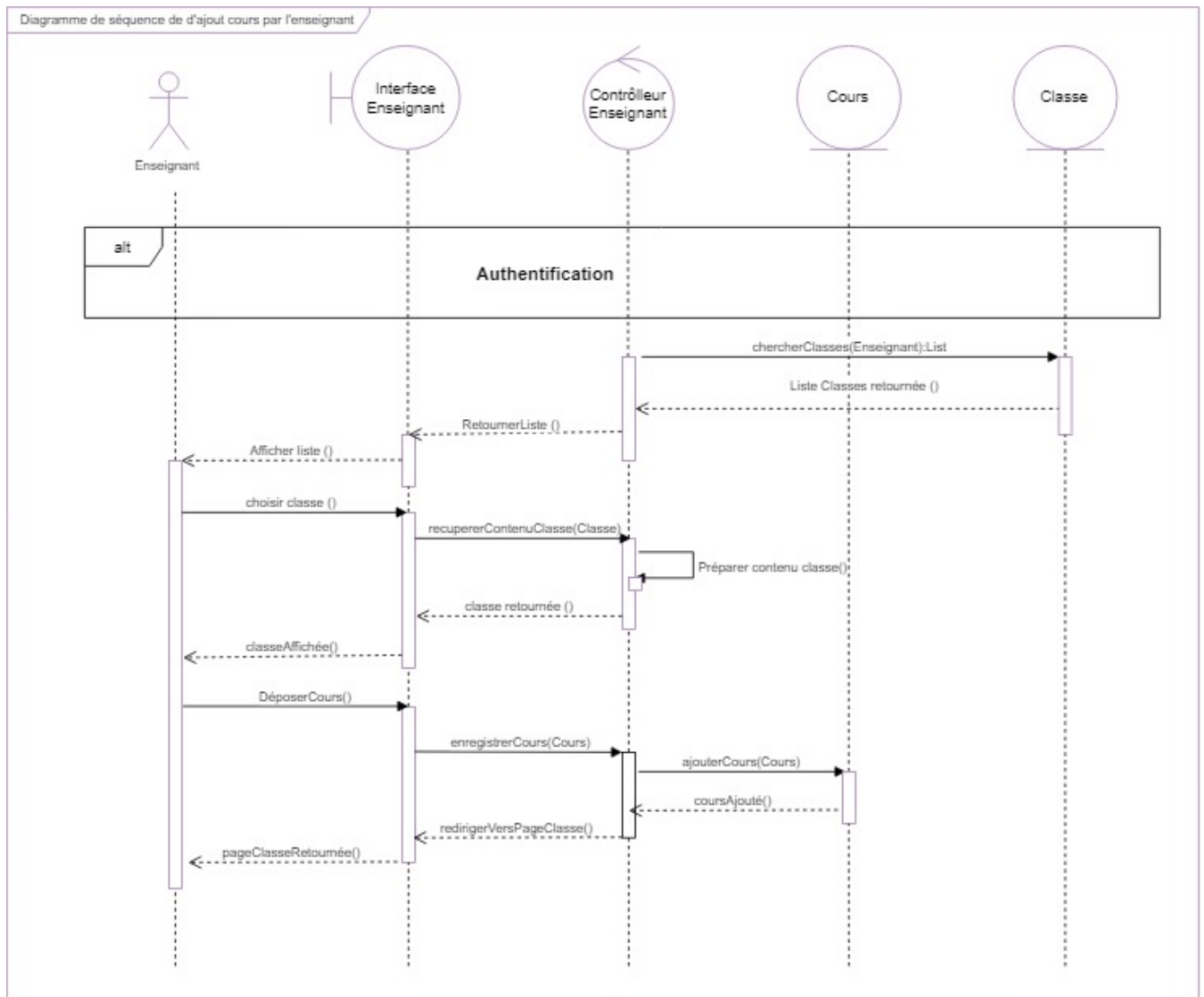


FIGURE 3.3 – Diagramme de séquences de l'ajout de cours par enseignant

-Description :

Après authentification et accès à l'interface d'ajout d'un cours , l'enseignant est invité à choisir une ou plusieurs classes de celles qu'il enseigne , faire l'upload du fichier correspondant au cours . Après validation, les données seront envoyées au contrôleur d'ajout de cours qui va les acheminées à la table cours responsable de la persistance du titre et du nom du fichier et l'identifiant de la classe dans la table cours et de mettre le fichier correspondant dans le dossier des cours enregistrés.

3.3.2.3 Diagramme de séquences de passage d'un test pour étudiant

Les détails du passage d'un test par un étudiant sont représentés par le diagramme de séquence sujet de la figure 3.4 :

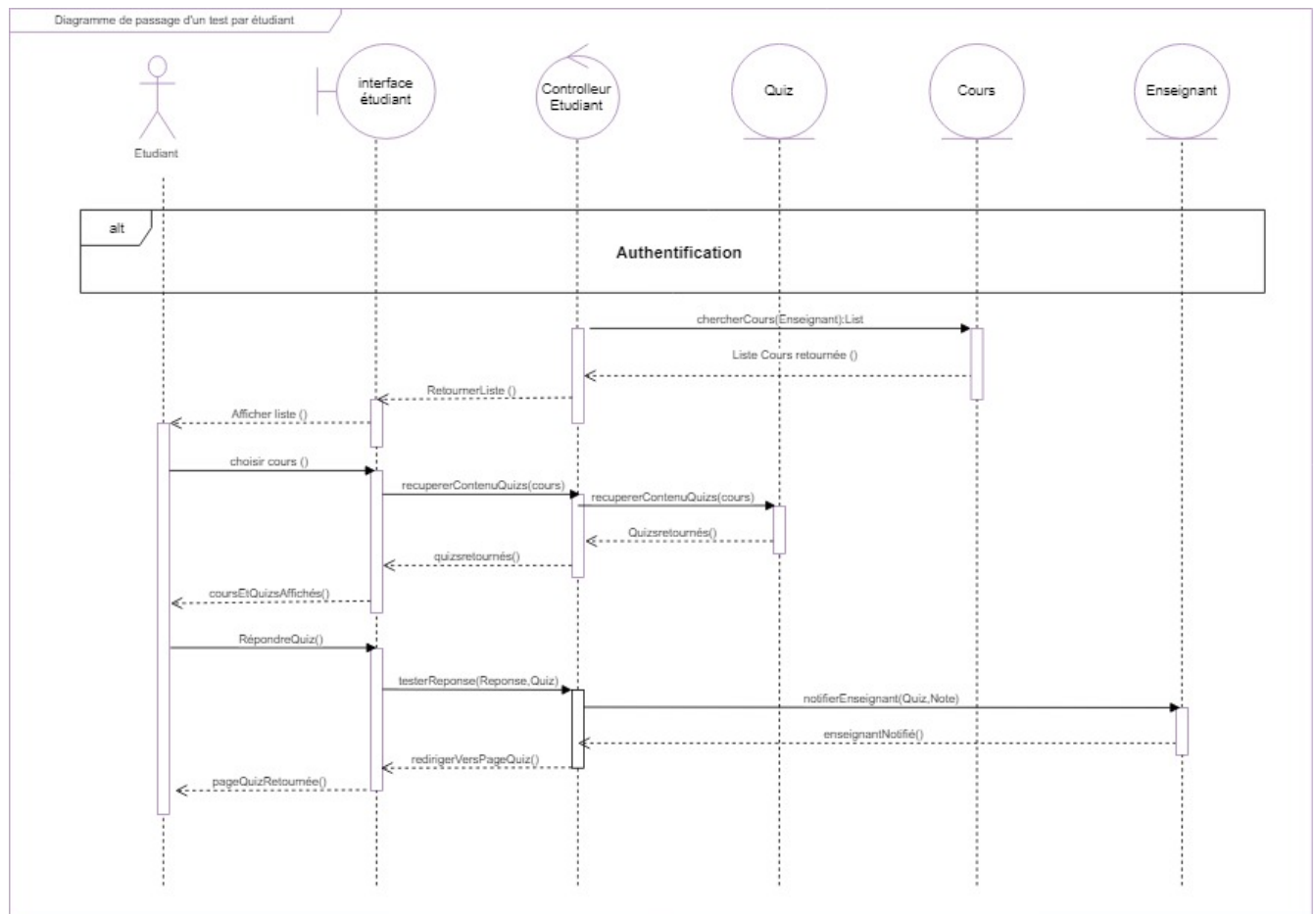


FIGURE 3.4 – Diagramme de séquences de passage d'un test pour étudiant

-Description :

Après authentification , la liste des cours correspondant à la classe de l'étudiant en question s'affiche . Le choix du cours permet l'affichage de la liste des tests de type questions à choix multiples. Après choix de l'un d'eux le contrôleur nous renvoie son contenu , qui sera affiché dans une interface convenable . Après terminaison de la réponse et validation l'enseignant responsable de ce test sera notifié par le contrôleur correspondant.

3.4 Conclusion

Une étude conceptuelle du système a été effectuée dans ce chapitre. Dans un premier lieu, l'architecture globale de l'application a été décrite. Ensuite, nous avons détaillé la conception à travers des diagrammes illustrant les différentes parties du système.

Nous entamons à présent le chapitre relatif à la réalisation pour illustrer la mise en place de notre application.

Chapitre 4

Réalisation

4.1 Introduction

Après avoir traité la partie conceptuelle du projet ,dans cette partie nous allons se concentrer sur l'aspect réalisation. Pour ce faire il faut tout d'abord définir l'environnement matériel et logiciel utilisé.Ensuite nous parlerons de la mise en place de l'application, les techniques utilisées et de la planification du projet tout en récurant à des imprimes écran et des exemples présentant les besoins satisfaits.

4.2 Environnement de développement

L'implémentation de la solution informatique nécessite un ensemble d'outils matériels et logiciels que nous détaillerons dans cette partie.

4.2.1 Environnement matériel

Pour la réalisation de cette application nous avons eu recours à l'ensemble d'outils matériels suivant :

TABLE 4.1: Environnement matériel

<u>Ordinateur ASUS X556U</u>	<u>Ordinateur ACER</u>	<u>Smartphone S4</u>
Processeur : Intel I7 Mémoire : 8 Go de RAM Disque dur : 512Go SE : Windows 10	Processeur : Intel I3 Mémoire : 4 Go de RAM Disque dur : 512Go SE : Windows 7	SE : Android 7.0 Processeur : Quad-core 2.3 GHz Mémoire : 3 Go de RAM

4.2.2 Environnement logiciel

Après avoir citer les moyens matériels utilisés , nous passons aux choix logiciels : Les logiciels utilisés dans la réalisation de l'application et la rédaction du rapport sont :

Eclipse : C'est un environnement de développement intégré (EDI) utilisé dans le réalisation de n'importe quelle application vu qu'il supporte tout langage de développement. Eclipse intègre une multitude d'outil facilitant la tâche tel que les gestionnaires de version , les bases de données ainsi que leurs intermédiaires de manipulation , les palettes de design ainsi que les les outils de test des API (Application Programming Interface).

Tomcat : Serveur web importé avec Spring boot.

RobotMONGO : Logiciel de manipulation des bases de données NoSQL . En plus de la manipulation graphique, RobotMONGO offre un shell .

WebStorm et Brackets : Deux éditeurs facilitant le développement WEB.

StarUML : Logiciel de modélisation UML.

Texmaker : Logiciel d'édition de documents Latex.

Git : Gestionnaire de version décentralisé .

4.2.3 Choix technologiques

Nous avons choisi pour la réalisation de ce projet un ensemble de technologies représentées dans la figure 4.1 :

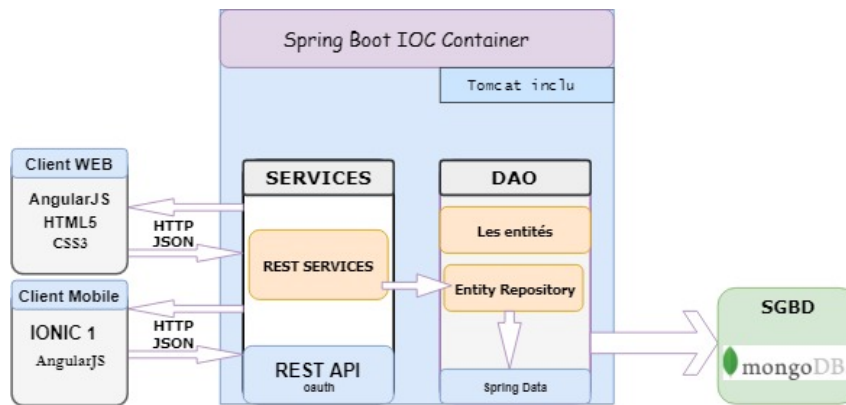


FIGURE 4.1 – Choix technologiques

Les services Web REST sont utilisés pour développer des architectures orientées ressources dans lesquelles chaque composant est une ressource accessible par un URI en utilisant les méthodes standards du protocole HTTP (POST, GET, PUT, DELETE) [N5]. REST utilise des formats différents pour représenter les ressources notamment XML et JSON. Dans notre cas, les données sont échangées sous le format JSON et ils seront par la suite consommés par le client WEB et Mobile et ceci après authentification puisque toutes les routes sont sécurisées par un Token (délivré par le protocole OAuth).

-HTML : langage de balisage créé et utilisé pour écrire les pages Web.

-CSS : langage permettant de définir des règles appliquées à un ou plusieurs documents.

-AngularJS : Framework Javascript open-source, permettant de réaliser des applications Web dynamique en utilisant le modèle MV-VM (ModelView View-Model) [N6].

-MongoDB : est un système de gestion de base de données orientée documents, ne nécessitant pas de schéma prédéfini des données et il est encore en développement actif en C++." MongoDB est désigné pour les application web et l'infrastructure de l'internet. Le modèle de données et les stratégies de persistance sont conçu pour une utilisation énorme en lecture et écriture et la possibilité d'évoluer facilement avec un basculement automatique [3]."

-JAVA : C'est le langage utilisé dans le développement des applications avec le framework J2EE.

-Spring : C'est un framework qui apporte des nombreuses fonctionnalités sur les aspects web, sécurité, batch ou encore accès aux données dans le cadre du développement d'une application.

* Spring Boot : C'est un projet ou un micro framework qui a notamment pour but

de faciliter la configuration d'un projet Spring et de réduire le temps alloué au démarrage d'un projet.

-Maven : Maven est un outil d'automatiser la gestion de projets Java. Pour bien gérer les dépendances, Maven s'appuie sur des 'repositories' de librairies (jar).

-plugin Eclipse STS : permet de générer rapidement la structure de votre projet en y incluant toutes les dépendances Maven nécessaires à votre application.

-JSON : Format de représentation et d'échange de données par les services web REST .

- OAuth : Protocole libre permettant d'autoriser une application consommatrice à utiliser l'API d'une application fournisseuse de services.

4.3 Travail réalisé et scénarios d'exécution

Nous arrivons maintenant à présenter quelques scénarios d'exécution . Pour cela nous récurons à des captures écran des interfaces utilisateurs pour détailler le fonctionnement de l'application.

Commençant tout d'abord par la page d'accueil . Comme convenu , chaque portail de communication académique inclu une page d'accueil dans laquelle on expose des informations générales et attirantes, FAQ , formulaire d'inscription , etc... Les figures suivantes représentent ces interfaces :

Nous passons maintenant à quelques enchainement de scénarios d'exécutions. Commençons par l'interface d'authentification de l'administrateur :

Conclusion et perspectives

Dans le cadre du projet de fin d'année , nous avons conçu et développé un portail de communication académique permettant de faciliter la gestion des MOOC .Ce travail a pour but d'améliorer l'éducation numérique et d'offrir une interface de gestion et d'apprentissage conviviale.

Dans un premier lieu , nous avons commencé par identifier le contexte général de notre application et les différentes exigences de notre futur system.Tout cela nécessite une étude préalable et une exposition de la solution proposée tout en identifiant l'apport de notre solution .

Ensuite , nous avons fait l'analyse et la spécification des besoins en identifiant les acteurs principaux et en utilisant des diagrammes de cas d'utilisation dans lesquels nous avons détaillé les actions principales des utilisateurs.

Dans le troisième chapitre , nous avons expliqué la conception du projet . Nous avons décrit l'architecture globale de l'application ainsi que les détails de la conception à travers les diagrammes statiques et dynamiques correspondants.

Le dernier chapitre était l'occasion pour exposer et décrire les interfaces principales de l'application en récurant à des imprimés écrans représentant les scénarios principaux.

Ce projet nous a permis de découvrir de près les problèmes qui peuvent survenir tout au long du développement d'une application. Nous avons pu enrichir nos connaissances techniques à savoir la conception en UML ,l'utilisation du frame-work Spring , l'exploitation des API REST ainsi que le frame-work AngularJS au cours du développement de la partie front-end.

En termes de perspectives, notre application est extensible par l'incrementation d'autres modules. Il serait intéressant d'améliorer la partie mobile .L'exposition des cours peut être plus facile pour les enseignants en leurs offrant des interfaces de saisie plus conviviales. Il faut aussi penser à donner la possibilité au enseignants de définir les cri-

tères pour délivrer le certificat pour chaque cours déposé et et l'offrir aux utilisateurs qu'il méritent. La vidéo conférence est une fonctionnalité très importante que nous avons commencé à implémenter.

Bibliographie

- [1] F. V. Pascal Roques, UML2 en action de l'analyse des besoins à la conception, 2007
- [2] Christian Soutou Avec la collaboration de Frédéric Brouard, Modélisation des bases de données, 2012.
- [3] Kyle Banker, MongoDB in action, 2015.

Bibliographie

- [N1] Coursera. <https://fr.coursera.org/>. [En ligne ; dernière consultation : 30.04.2018].
- [N2] Emploi-Store .<https://www.emploi-store.fr/portail/services/openclassrooms> .[En ligne ; dernière consultation : 28.04.2018].
- [N3] France Université Numérique. <https://www.fun-mooc.fr/about> . [En ligne ; dernière consultation : 29.04.2018].
- [N4] L'université virtuelle de Tunisie. <http://www.uvt.rnu.tn/l-universite/presentation>. [En ligne ; dernière consultation : 28.04.2018].
- [N5] Tutorialspoint : Restful web services-quick guide. http://www.tutorialspoint.com/restful/pdf/restful_quick_guide.pdf. [En ligne ; dernière consultation : 10.03.2018].
- [N6] AngularJS : Documentation officielle de angularjs. <https://angularjs.org/>. [En ligne ; dernière consultation : 22.04.2018].

Chapitre 5

Annexe A : Suite des interfaces de l'application