
Projet Fin D'Etudes

présenté à :

**Institut Supérieur Des Etudes
Technologiques De Zaghouan**

en vue de l'obtention du :

**Diplôme Licences en
Technologies de L'informatique**

**Spécialité :
Développement Systèmes D'informations**

Elaboré par :

**Yemna Issra
Abidi Iheb**

Conception et développement d'application d'éducation en ligne

Au Sein de Centre National de L'informatique

Mr.Ramzi Ghazouani
Mme.Samar Abbadi

Encadrant
Encadrant Entreprise

Année universitaire :
2023/2024



Dédicace

Je tiens tout d'abord à exprimer ma gratitude envers Allah, le Tout-Puissant et Miséricordieux, pour m'avoir accordé la force et la patience nécessaires pour accomplir ce modeste travail.

Je souhaite rendre hommage à mon père, **Kamel Yemna** dont je suis sûr qu'il serait fier de voir les résultats obtenus après de nombreuses années de sacrifices et de privations pour m'aider à progresser dans la vie.

Je souhaite également exprimer ma reconnaissance envers ma mère, **Najwa Zarrouk** qui représente pour moi une source de tendresse et un exemple de dévouement constant qui m'a toujours encouragé.

À mon frère, **Dhirar**
je te remercie également pour ton soutien.

A tous mes ami(e)s qui m'ont toujours encouragé, et à qui je souhaite plus de succès. et je vous aime énormément !

Je tiens à exprimer ma gratitude à toute la promotion DSI pour les moments partagés et l'entraide mutuelle.

Enfin, je souhaite remercier tous ceux qui, par leurs mots d'encouragement, m'ont donné la force de continuer.

MERCI À TOUS.

Yemna ISSRA



Dédicace

Du profond de mon cœur je dédie ce modeste travail à tous ceux qui me sont chers ...

À ma merveilleuse mère,

Mon pilier de force et ma source d'inspiration. Votre amour inconditionnel et votre soutien constant ont été les clés de ma réussite. Je vous suis éternellement reconnaissant pour tout ce que vous avez fait pour moi.

À mon cher père,

Mon modèle de persévérance et de détermination. Vos conseils avisés et votre exemple de travail acharné m'ont guidé tout au long de ce parcours. Votre présence bienveillante a été une source de réconfort et de motivation.

À ma sœur bien-aimée,

Ma complice de toujours. Tes encouragements, ta joie de vivre et ton esprit positif ont illuminé mes journées les plus sombres. Merci d'avoir été là à chaque étape de ma vie.

Et enfin, à mon frère cher,

Ton soutien indéfectible à distance et tes paroles inspirantes m'ont donné la force de poursuivre mes rêves. Je te souhaite tout le bonheur et le succès dans tes propres entreprises.

À vous quatre, ma famille, je dédie ce travail. Votre amour, votre soutien et vos encouragements m'ont permis de me surpasser et d'atteindre mes objectifs. Que le bonheur et le succès vous accompagnent chaque jour de votre vie.

MERCI À TOUS.

Abidi IHEB



REMERCIEMENT

Nous tenons à exprimer notre profonde gratitude envers toutes les personnes qui ont contribué au succès de notre stage et m'ont apporté leur précieuse aide lors de la rédaction de ce pfe. Nous tenons tout d'abord à remercier chaleureusement notre encadreur,

Mr. Ramzi Ghazouani, Professeur à l'institut Supérieur d'Etude Technologiques de Zaghouan pour sa patience, sa disponibilité et ses conseils judicieux. Ses conseils éclairés ont grandement enrichi ma réflexion et ont été d'une importance capitale pour la réalisation de ce travail. Nous tenons à remercier profondément notre encadrant,

Mme. Samar, l'encadrant a la société CNI pour l'aide pratique et moral, sa gentillesse, sa confiance, le temps passé ensemble et aussi le partage de son expertise et les conseils concernant les tâches évoquées dans ce rapport.

Aux membres du jury, Nous tenons nos remerciements aux membres du jury pour l'honneur qu'ils nous fait en jugeant ce projet de fin d'études. A tous nos professeurs et enseignants de ISETZG.

Notre plus vifs remerciements s'adressent aussi à tout le cadre enseignant notre faculté.

Enfin, nous tenons à remercier toutes les personnes qui nous ont accordé de leur temps précieux pour les entrevues réalisées dans le cadre de ce projet. Leurs connaissances partagées et leur contribution ont été essentielles pour l'enrichissement de ce travail.



TABLE DES MATIÈRES

LISTE DES FIGURES	ix
LISTE DES ABRÉVIATIONS	x
INTRODUCTION GÉNÉRALE	1
1 Chapitre1 : Présentation du cadre de projet	4
1.1 Introduction	5
1.2 Présentation de l'organisme d'accueil	5
1.2.1 Les grandes applications du CNI	6
1.2.2 Les services offerts par le CNI	6
1.2.3 Les ressources matérielles de CNI	7
1.2.4 L'organigramme de l'architecture du CNI	8
1.3 Présentation du projet	10
1.3.1 Etude de l'existant	10
1.3.2 Solution proposée	10
1.4 Méthodologies et modélisme adopté	10
1.4.1 Langage de modélisation	10
1.5 Methodolige de développement	12
1.5.1 Comparaison de methodolige classique et methodologie agile	12
1.5.2 Les méthodes agiles	13
1.5.3 Principe de SCRUM	14
1.5.4 Les bases de SCRUM	14
1.5.5 Les différentes composantes de SCRUM	15
1.5.6 Les différents intervenants de la méthodologie SCRUM	16
1.6 Conclusion	17
2 Chapitre 2 :Analyse et spécification des besoins	18
2.1 Introduction	19

2.2	Spécification des besoins	19
2.2.1	Identification des acteurs	19
2.2.2	Identification des besoins	19
2.2.3	Besoins fonctionnels	20
2.2.4	Besoins non fonctionnels	21
2.3	Pilotage du projet avec SCRUM	21
2.3.1	Equipe et rôles	22
2.4	Structure et planification des sprints	23
2.4.1	Backlog produit	23
2.5	Détail fonctionnel	25
2.5.1	Diagramme de cas d'utilisation général	25
2.5.1.1	Diagramme de cas d'utilisation général de l'apprenant . . .	27
2.5.1.2	Diagramme de cas d'utilisation général de formateur	28
2.5.1.3	Diagramme de cas d'utilisation général de l'administrateur .	29
2.5.2	Diagramme de classe globale	30
2.6	Architecture du système	31
2.6.1	Architecture logicielle	31
2.6.2	Architecture logique	32
3	Sprint 1 : Mise en oeuvre de sprint 1	34
3.1	Introduction	35
3.2	Backlog du Sprint 1	35
3.2.1	Définition d'un backlog de Sprint	35
3.2.2	But de Sprint	35
3.2.3	User Stories	35
3.3	Spécification fonctionnelle	36
3.3.1	Diagramme de cas d'utilisation du premier sprint	36
3.3.2	Description textuelle et raffinement des principaux cas d'utilisation . .	37
3.3.2.1	Raffinement du cas d'utilisation "S'inscrire"	37
3.3.2.2	Description textuelle du cas d'utilisation "S'inscrire"	38
3.3.2.3	Raffinement du cas d'utilisation "S'authentifier"	38
3.3.2.4	Description textuelle du cas d'utilisation "S'authentifier" . .	39
3.3.2.5	Raffinement du cas d'utilisation "Gérer comptes"	39
3.3.2.6	Description textuelle du cas d'utilisation "Activer Compte" . .	40
3.3.2.7	Description textuelle du cas d'utilisation "Désactiver Compte"	40
3.3.2.8	Description textuelle du cas d'utilisation "Ajouter compte" .	41

3.3.2.9	Description textuelle du cas d'utilisation « Modifier compte »	41
3.3.2.10	Description textuelle du cas d'utilisation "Supprimer compte"	42
3.4	Conception	42
3.4.1	Diagrammes de séquences détaillés	42
3.4.2	Diagramme de séquence lié à la fonctionnalité « S'inscrire »	43
3.4.3	Diagramme de séquence lié à la fonctionnalité « S'authentifier »	44
3.4.4	Diagramme de séquence lié à la fonctionnalité « Activer compte »	45
3.4.5	Diagramme de séquence lié à la fonctionnalité « Ajouter compte »	46
3.4.6	Diagramme de séquence lié à la fonctionnalité « Modifier compte »	47
3.4.7	Diagramme de séquence lié à la fonctionnalité « Supprimer compte »	48
3.5	Réalisation	49
3.5.1	Interface d'authentification	49
3.5.2	Interface d'inscription	50
3.6	Conclusion	50
4	Chapitre 4 : Réalisation du deuxième sprint	51
4.1	Introduction	52
4.2	Backlog du Sprint 2	52
4.3	Spécification fonctionnelle	52
4.3.1	Diagramme des cas d'utilisation du deuxième sprint 2	53
4.3.2	Description textuelle et raffinement des principaux cas d'utilisation	53
4.3.2.1	Raffinement du cas d'utilisation "Gérer cours"	54
4.3.2.2	Description textuelle du cas d'utilisation 'Ajouter cour'	54
4.3.2.3	Description textuelle du cas d'utilisation 'Modifier cour'	55
4.3.2.4	Description textuelle du cas d'utilisation " Supprimer cour"	55
4.3.2.5	Raffinement du cas d'utilisation "Gérer Quizz"	56
4.3.2.6	Description textuelle du cas d'utilisation " Ajouter quizz"	56
4.3.2.7	Description textuelle du cas d'utilisation 'Modifier quiz'	57
4.3.2.8	Description textuelle du cas d'utilisation " Supprimer quiz"	57
4.4	Conception	57
4.4.1	Diagrammes de séquences détaillés	58
4.4.1.1	Diagramme de séquence lié à la fonctionnalité « Ajouter cour »	58
4.4.1.2	Diagramme de séquence lié à la fonctionnalité « Modifier cour »	59
4.4.1.3	Diagramme de séquence lié à la fonctionnalité « Supprimer cour »	60

4.4.1.4	Diagramme de séquence lié à la fonctionnalité « Ajouter quiz »	61
4.4.1.5	Diagramme de séquence lié à la fonctionnalité « Modifier quiz »	62
4.5	Réalisation	62
4.5.1	Interface Gestion cour	63
4.5.2	Interface Ajouter cour	63
4.5.3	Interface Modifier cour	63
4.5.4	Interface supprimer cour	63
4.5.5	Interface Gestion quiz	63
4.5.6	Interface Ajouter quiz	63
4.5.7	Interface Supprimer quiz	63
4.6	Conclusion	63
5	Chapitre5 : Réalisation du troisième sprint	64
6	Chapitre6 : Réalisation	65
6.1	Introduction	66
6.2	Diagramme de déploiement	66
6.3	Environnement du travail	66
6.3.1	Environnements matériels	67
6.4	Environnements Logiciels	68
6.4.1	IDE	68
6.4.2		68
6.4.3	Les Framework du développement	69
6.4.4	Les langages de la programmation	70
6.4.5	Outil de conception	71
6.4.6	Outils	71
6.4.7	Site web :	71
6.5	Chronogramme de travail	72
6.6	Conclusion	73
	CONCLUSION GÉNÉRALE	74
	BIBLIOGRAPHIE	74

LISTE DES FIGURES

1.1	Local de CNI	5
1.2	Les services de CNI	7
1.3	Nouvelle architecture du réseau inter-administratif (RNIA)	8
1.4	Nouvelle structure du réseau inter-administratif (RNIA)	9
1.5	Logo UML	11
1.6	Cycle de vie de la méthodologie SCRUM	15
1.7	Cycle de vie de la méthodologie SCRUM	16
2.1	Equipe SCRUM et rôles	22
2.2	Diagramme de cas d'utilisation général de l'apprenant	27
2.3	Diagramme de cas d'utilisation général de formateur	28
2.4	Diagramme de cas d'utilisation général de l'administrateur	29
2.5	Architecture de MVC	32
2.6	Architecture 3-tiers	33
3.1	Diagramme de cas d'utilisation globale de sprint 1	37
3.2	Diagramme de cas d'utilisation globale de sprint 1	37
3.3	Raffinement du cas d'utilisation s'authentifier	38
3.4	Raffinement du cas d'utilisation Gérer comptes	39
3.5	Diagramme de séquence lié à la fonctionnalité « S'inscrire »	43
3.6	Diagramme de séquence lié à la fonctionnalité « S'authentifier »	44
3.7	Diagramme de séquence lié à la fonctionnalité « Activer compte »	45
3.8	Diagramme de séquence lié à la fonctionnalité « Ajout compte »	46
3.9	Diagramme de séquence lié à la fonctionnalité « Modifier compte »	47
3.10	Diagramme de séquence lié à la fonctionnalité « Supprimer compte »	48
3.11	Interface d'authentification »	49
3.12	Interface d'inscription »	50
4.1	Diagramme de cas d'utilisation globale de sprint 2	53
4.2	Diagramme de cas d'utilisation Gérer cours	54

4.3	Diagramme de cas d'utilisation Gérer quizz	56
4.4	Diagramme de séquence lié à la fonctionnalité « Ajouter cour»	58
4.5	Diagramme de séquence lié à la fonctionnalité « Modifier cour»	59
4.6	Diagramme de séquence lié à la fonctionnalité « Modifier cour»	60
4.7	Diagramme de séquence lié à la fonctionnalité « Ajouter quiz»	61
4.8	Diagramme de séquence lié à la fonctionnalité « Modifier quiz»	62
6.1	Pc Asus	67
6.2	Logo Visual Studio Code	68
6.3	Logo IntelliJ IDEA	68
6.4	Logo MySQL	69
6.5	Logo Angular	69
6.6	Logo Spring Boot	69
6.7	Logo Java 8	70
6.8	Logo Type Script	70
6.9	Logo Draw.io	71
6.10	Logo Postman	71
6.11	Logo GitHub	72
6.12	Chronogramme de temps	72



LISTE DES ABRÉVIATIONS

AAA Aaa Aaa Aaa

BB Bbb Bbb

CCC Ccc Ccc Ccc



INTRODUCTION GÉNÉRALE

L'enseignement est une méthode pédagogique pour cultiver les connaissances des élèves. Malgré les bouleversements sociaux et l'évolution des moyens de communication au cours du dernier siècle, de nombreux observateurs ont bien informé que la structure de l'éducation traditionnelle n'a guère changé depuis le siècle dernier.

L'éducation traditionnelle est principalement basée sur des cours magistraux. L'auditeur est passif, souvent intellectuellement absent en raison des conditions difficiles provoquées par la surtaxe par heure. Le principal moyen d'une telle éducation : le programme combine habilement instruction et travaux pratiques, support « tableau noir » et « papier ». D'autre part, le système éducatif traditionnel exige que tous les apprenants aient un lieu unifié, un temps unifié, une action unifiée et un rythme unifié.

C'est-à-dire que le mécanisme est rigide, qu'il est difficile de s'adapter à la réalité et que la tendance à améliorer le système au niveau éducatif par le biais de supports audio-visuels traditionnels (diapositives, projections de séquences vidéo, etc.) ne résout pas le problème. En effet, le formateur doit à la fois présenter la leçon et entreprendre une opération technique assez difficile.

L'enseignement moderne nécessite des moyens pour faciliter l'apprentissage de l'apprenant, tels que la sélection des situations, l'interprétation des objectifs et des critères, le choix des contenus, la mise en œuvre des programmes éducatifs et l'élaboration des outils pour aider les élèves à construire et acquérir des connaissances et des compétences. Il existe différentes solutions d'organisation de formation basées sur les TIC sur le marché. Ces solutions assurent diverses fonctions de base (communication audio vidéo, discuté.....). Cependant, ils ont plusieurs inconvénients, comme dans temps de réponse du système, le nombre limité d'apprenants.

Il n'existe pas une solution d'apprentissage complète qui offre à la fois de meilleures fonctionnalités

de la gestion des formations en ligne. Le travail présenté dans notre projet s'inscrit dans ce contexte. Notre objectif est de concevoir et de construire un système qui réunit les différentes fonctionnalités nécessaires à une plateforme d'apprentissage en ligne et confronte les lacunes des solutions existantes.

Notre projet se compose de trois parties principales, qui sont :

- Gestion des formations : permet la gestion des formations ainsi que la gestion des utilisateurs (création des comptes pour les formateurs et les apprenants...)
- Web Conférence : cette partie représente notre salon de formation qui offre les fonctionnalités suivantes :
 - Uploader le support d'une formation.
 - Utiliser les ressources vidéo.
- Paiement.

Au cours du premier chapitre intitulé , Cadre du projet , nous présenterons l'environnement du projet en définissant l'entreprise d'accueil et les différents services qu'elle offre, ainsi que l'étude de l'existant, les critiques et les solutions proposées.

Par ailleurs, le choix de la méthodologie sera présenté et le tout sera achevé par une vue générale sur l'environnement matériel et logiciel de la réalisation de ce projet.

Le deuxième chapitre, l'étude préliminaire , traitera les spécifications des besoins fonctionnel et non fonctionnels. Par ailleurs, nous allons découper notre projet en effectuant le planning des réalisations pour délimiter son cadre et pour initialiser les différents besoins afin de commencer le premier sprint dans de bonnes conditions.

Le troisième chapitre intitulé , "La réalisation du premier sprint", présente le premier sprint. Nous présentons, dans un premier temps, le backlog du sprint 1 ainsi que les diagrammes des cas d'utilisation. Ensuite, nous illustrons les étapes de conception de notre application et nous terminons par quelques exemples des interfaces de l'application développée dans ce sprint. Le quatrième chapitre intitulé , "La réalisation du deuxième sprint" , présentera les différentes étapes de réalisation du sprint2.

Le cinquième chapitre intitulé , "La réalisation du troisième sprint" , s'intéresse à décrire la réalisation du troisième sprint .

Le sixième chapitre intitulé , "La réalisation du quatrième sprint" , s'intéresse à décrire la réalisation du dernier sprint de notre projet .

Chapitre1 : Présentation du cadre de projet

Sommaire

1.1	Introduction	5
1.2	Présentation de l'organisme d'accueil	5
1.2.1	Les grandes applications du CNI	6
1.2.2	Les services offerts par le CNI	6
1.2.3	Les ressources matérielles de CNI	7
1.2.4	L'organigramme de l'architecture du CNI	8
1.3	Présentation du projet	10
1.3.1	Etude de l'existant	10
1.3.2	Solution proposée	10
1.4	Méthodologies et modélisme adopté	10
1.4.1	Langage de modélisation	10
1.5	Methodolige de développement	12
1.5.1	Comparaison de methodolige classique et methodologie agile	12
1.5.2	Les méthodes agiles	13
1.5.3	Principe de SCRUM	14
1.5.4	Les bases de SCRUM	14
1.5.5	Les différentes composantes de SCRUM	15
1.5.6	Les différents intervenants de la méthodologie SCRUM	16
1.6	Conclusion	17

1.1 Introduction

Ce premier chapitre a pour objectif de mettre le projet dans son cadre général. Il consiste à présenter, en premier temps, son environnement à travers une présentation de l'entreprise d'accueil et ses domaines d'activités et en second lieu, nous présenterons notre projet ainsi que la méthodologie de réalisation adoptée.

1.2 Présentation de l'organisme d'accueil

Avec beaucoup de motivation, j'ai effectué mon stage au sein de Cni . Ce stage m'a permis non seulement d'approfondir mes connaissances mais aussi d'enrichir et de développer mes aptitudes à la communication et l'intégration au sein de l'entreprise. Le Centre National de l'informatique a été créé le 30 décembre 1975 avec la percée de l'informatique dans l'environnement économique et social de la Tunisie et sur recommandation de la Commission Nationale de l'Informatique. Le Centre National de l'informatique est un Établissement public à caractère non administratif, doté de la personnalité civile et de l'autonomie financière. Le CNI offre, depuis 32 ans, à l'Administration publique tunisienne et à ses clients des différents secteurs socioéconomiques du pays, un service complet, allant du conseil à l'hébergement en passant par les études, la conception et le développement.



FIGURE 1.1 – Local de CNI

1.2.1 Les grandes applications du CNI

- **INSAF** : (Informatisation du Système Administratif et Financier du Personnel de l'Etat) : C'est une application inter administrative d'envergure nationale. Ce système prend en charge la gestion administrative et la paie de 90 .
- **SICAD** : (Système d'information et de Communication Administrative à Distance) : C'est un système de traitement et de communication de l'information contenant les procédures des services Administratifs rendus aux citoyens.
- **ADEB** : (Aide à la Décision Budgétaire) : C'est une application qui traite l'ensemble des opérations d'une procédure de dépense effectuée par l'Etat ou par les collectivités publiques sur les Crédits budgétisés ou sur un financement extérieur
- **MANKOULET** : (Gestion des biens mobiliers de l'Etat) : L'application Mankoulet permet le suivi des différentes étapes de gestion des biens mobiliers de l'Etat, depuis leur acquisition jusqu'à la fin de leur utilisation (vente, destruction, etc.)
- **MADANIA** : (Système national de l'état civil) : Ce système permet de gérer tous les documents de L'Etat Civil et de les éditer localement et à distance. Il sert à la gestion des documents de l'Etat Civil, l'enregistrement des actes de naissance, de mariage, de divorce, de décès et l'édition locale et à distance des documents de l'Etat Civil.

1.2.2 Les services offerts par le CNI

Le CNI a adapté ses services en s'ouvrant beaucoup plus à de nouveaux concepts et outils de Maîtrise de gestion de l'informatique, tels que l'infogérance, la maîtrise d'ouvrage déléguée, le pilotage, la sous-traitance, etc.

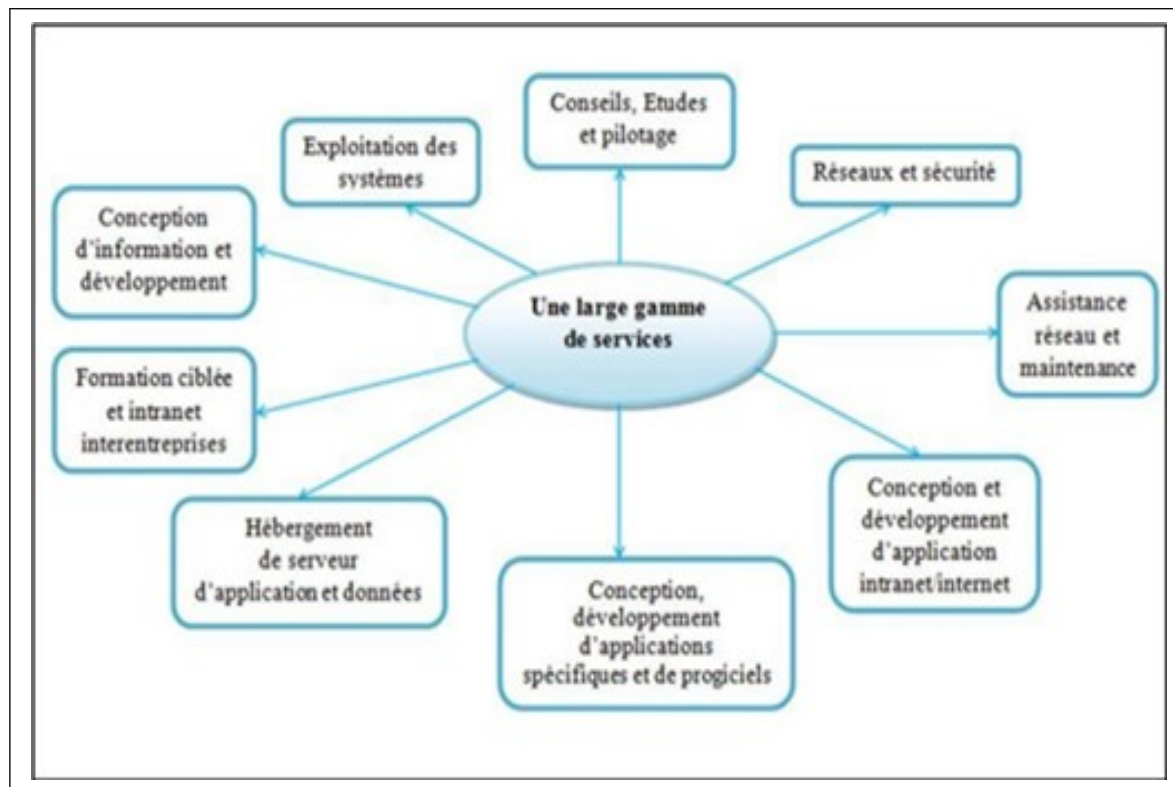


FIGURE 1.2 – Les services de CNI

1.2.3 Les ressources matérielles de CNI

Le CNI possède des ressources matérielles composées d'un centre d'hébergement et de traitement de l'information équipé d'un ensemble de serveurs de grande puissance. A ce serveur sont connectés :

- 1300 postes de travail implantés dans les départements ministériels, les directions régionales et les organismes publics.
- 150 postes de travail répartis sur les différentes structures.
- Un laboratoire de sécurité (Logiciels libres) pour plates-formes de tests et de simulation.
- Un réseau local permettant une circulation fluide de l'information entre les différentes entités.
- Une connexion de tous les postes de travail au réseau Internet.
- Une cellule de veille technologique permanente.
- Un centre d'appel à l'écoute des clients.
- Un centre de formation et de documentation constitué de neuf salles de formations équipées de 100 micro-ordinateurs et serveurs.

- Un amphithéâtre de 150 places.
- Une bibliothèque spécialisée en informatique

1.2.4 L'organigramme de l'architecture du CNI

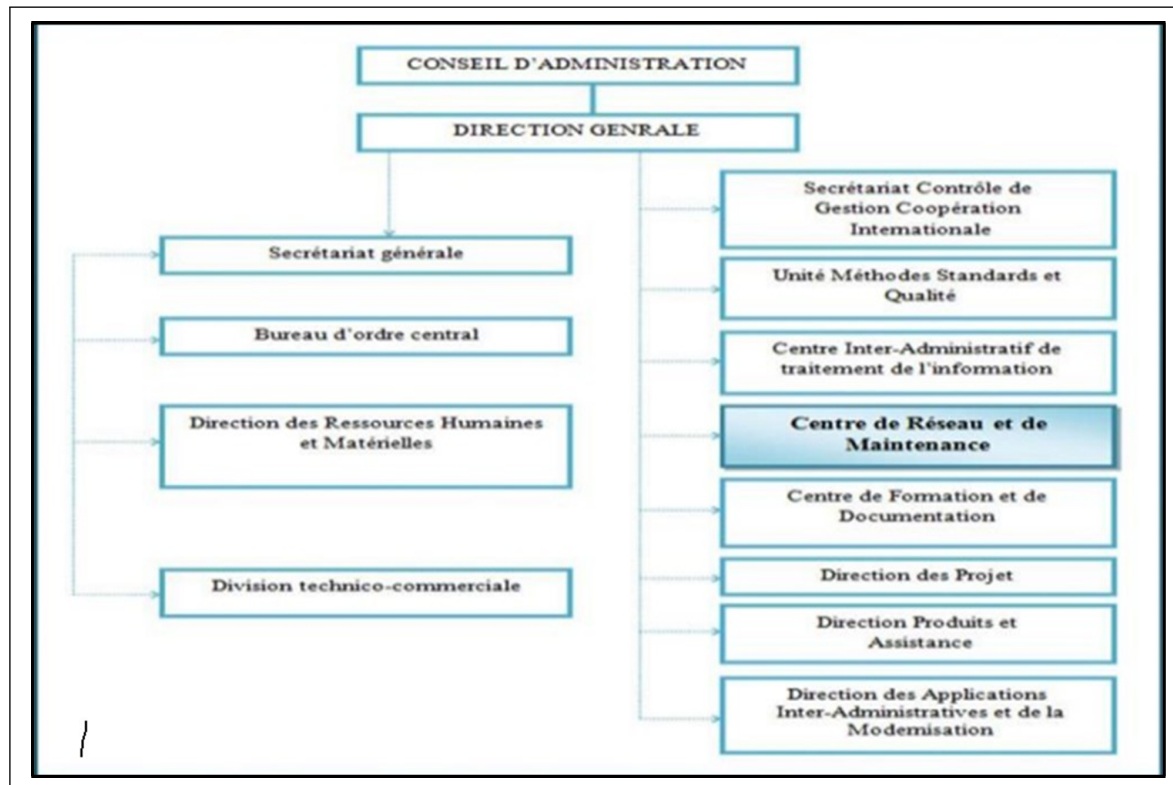


FIGURE 1.3 – Nouvelle architecture du réseau inter-administratif (RNIA)

Celui-ci est équipé d'un centre d'hébergement auquel sont connectés 3500 postes de travail implantés dans tous les départements ministériels, les directions régionales et les organismes publics. Il assure l'hébergement des données, des serveurs, et des systèmes d'information. Il assure l'hébergement des données, des serveurs, et des systèmes d'information. Dans le cadre de la stratégie de développement des services de l'administration électronique, un conseil ministériel tenu le 1er septembre 2006, a adopté la mise en place d'une nouvelle plateforme adéquate de réseaux et de systèmes informatiques pour la généralisation des prestations de l'administration. Dans ce but, le CNI participe à la mise en place d'un « nouveau » réseau nommé RNIA : Réseau.National Intégré de l'Administration qui porte la structure représentée par la figure suivante :

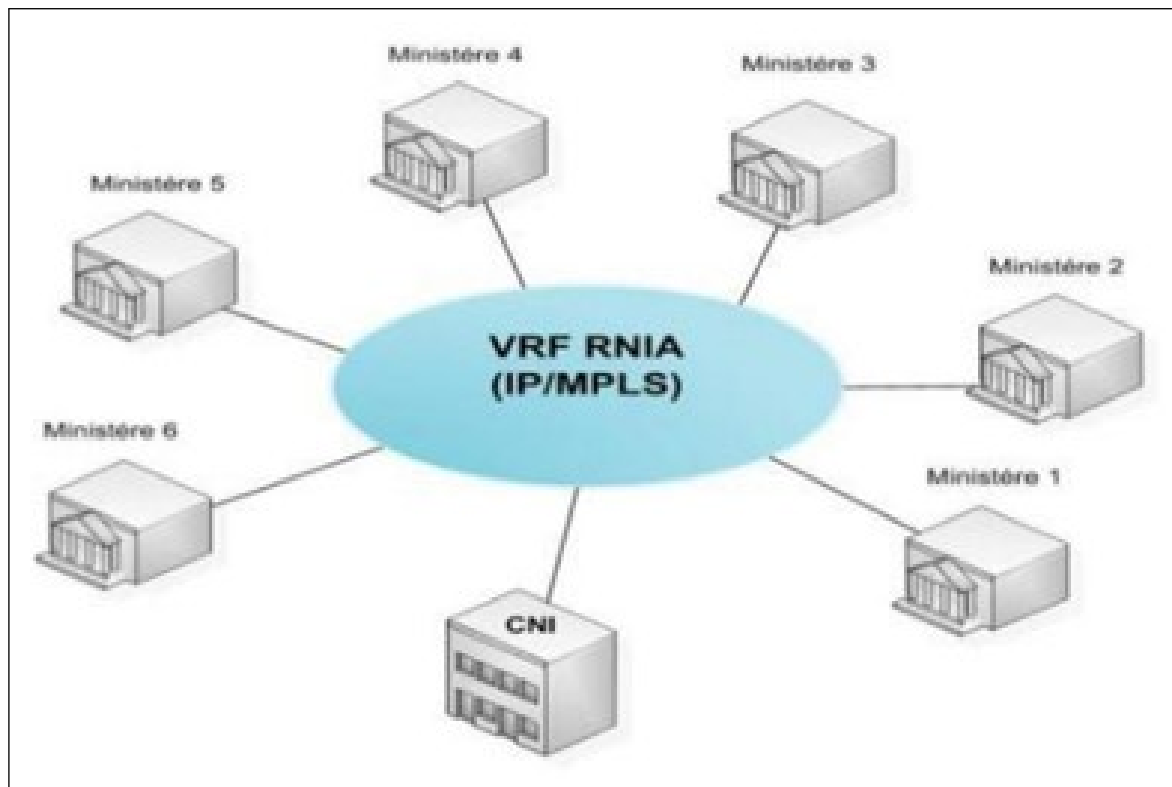


FIGURE 1.4 – Nouvelle structure du réseau inter-administratif (RNIA)

Les premières étapes en cours de réalisation sont : Un ensemble de services pour des ministères (pilotes) choisis en raison de la disponibilité de leur-connexion au réseau :

- Un annuaire électronique de l'Administration.
- Une messagerie électronique sécurisée.
- Des services qui s'appuient sur l'échange électronique de documents.
- Entre les bureaux d'ordre.
- Entre les cabinets ministériels.
- Le traitement des requêtes du citoyen entre les bureaux des relations avec le citoyen.
- La programmation, l'organisation et le suivi des réunions.
- L'échange électronique des données de l'39 ; Etat Civil avec les caisses sociales.

Les prochaines étapes du projet verront la concrétisation notamment des actions suivantes :

- Utilisation des services de la téléphonie IP pour le CNI et les ministères concernés.
- Généralisation du RNIA.
- Utilisation des services par tous les ministères. Exploitation du système INSAF et généralisation

du système 'Madania' sur l'ensemble des Établissements de l'enseignement supérieur et des ministères.

- Généralisation sur tous les services régionaux de l'administration.
- Fourniture de services Internet pour les utilisateurs du réseau.
- Échange électronique de documents sécurisé.

1.3 Présentation du projet

L'étude de l'existant d'un projet consiste à identifier les détails du service dont la société souhaite améliorer. Dans cette section, nous présentons la problématique, puis nous décrivons la solution proposée.

1.3.1 Etude de l'existant

1.3.2 Solution proposée

1.4 Méthodologies et modélisme adopté

1.4.1 Langage de modélisation

Il existe de nombreux langages de modélisation utilisés dans différents domaines pour représenter et communiquer des concepts, des systèmes et des processus. Pour notre projet on a choisi L'UML (Unified Modeling Language) est un langage de modélisation graphique largement utilisé dans le domaine du développement logiciel pour représenter, spécifier, visualiser et documenter les systèmes logiciels

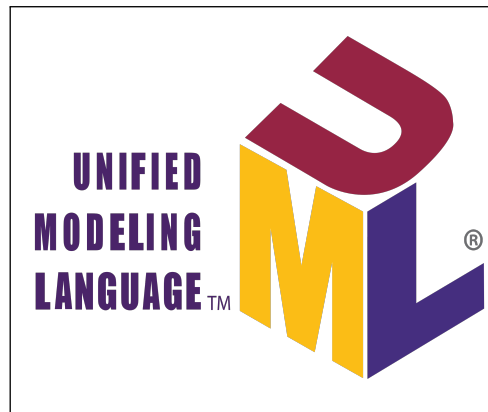


FIGURE 1.5 – Logo UML

UML définit plusieurs types de diagrammes dans deux catégories de vues, les vues statiques et les vues dynamiques .

- **Les vues statiques :**

- **Le diagramme des cas d'utilisation :** Il permet de structurer les besoins des utilisateurs et les objectifs correspondants de notre système en identifiant ses utilisateurs et leurs interactions.

- **Le diagramme de classes :** Il permet de présenter les classes et les interfaces de notre système ainsi que les différentes relations entre celles-ci.

- **Diagramme de déploiement :** Illustre les infrastructures physique et logiciel d'un système.

Ils sont utiles lorsqu'une solution logicielle est déployée sur de nombreuses machines avec des configurations unique.

- **Le diagramme de paquetage :** Il permet de modéliser l'application sous forme de paquets. Ceci offre un niveau d'abstraction élevé et permet de présenter la modularité de l'application

- **Les vues dynamiques :**

- **Diagramme d'activités :** Flux de travail d'entreprise ou opérationnels représentés graphiquement pour montrer l'activité de chacune des composants du système.

- **Diagramme de séquence :** décrivent de manière temporelle les interactions entre objets et acteur.

1.5 Methodologie de développement

Les méthodologies de développement définissent les activités, les rôles et les responsabilités des membres de l'équipe, ainsi que les processus et les étapes à suivre pour atteindre les objectifs du projet.

Elles peuvent également inclure des outils, des techniques et des bonnes pratiques spécifiques pour faciliter la gestion et le suivi du projet. L'objectif principal d'une méthodologie de développement est d'améliorer la qualité du logiciel, d'optimiser les ressources et de garantir la satisfaction du client. Il existe plusieurs méthodologies de développement logiciel.

1.5.1 Comparaison de methodologie classique et methodologie agile

TABLE 1.1 – Description des méthodes traditionnelles et agiles

Méthode	Description
Agiles	<ul style="list-style-type: none">• Itératif et incrémental : Projets divisés en sprints courts pour produire des livrables fonctionnels.• Flexibilité : Adaptation aux changements avec des cycles de développement courts et une planification adaptative.• Collaboration : Communication encouragée entre l'équipe et les parties prenantes.• Livraison continue : Livraison régulière de fonctionnalités et prise en compte des retours utilisateurs.
Traditionnelles	<ul style="list-style-type: none">• Planification préalable : Toutes les étapes du projet sont prévues dès le début.• Faible flexibilité : Les décisions prises en amont sont difficiles à modifier sans perturber le processus.• Livrables finaux : Les résultats sont produits à la fin de chaque phase.• Communication limitée : Les échanges sont restreints entre les membres de l'équipe et les parties prenantes.

D'après ce tableau, nous avons l'habilité d'éliminer les méthodes traditionnelles. **(L'expérience a montré que plus de 80 % des projets exécutés selon ces méthodologies connaissent des retards et des dépassements budgétaires)**. Le facteur du temps est très important pour la réalisation des projets informatiques.

1.5.2 Les méthodes agiles

Les méthodes agiles ont une manière de réduire le cycle de développement des projets informatiques, de répondre plus rapidement aux évolutions des demandes de l'utilisateur final. Les projets informatiques agiles sont gérés de manière adaptative, incrémentale et itérative. Elles sont basées sur quatre valeurs fondamentales :

- Individus et leurs interactions,
- Fonctionnalités opérationnelles,
- Collaboration avec le client,
- Acceptation du changement plutôt que la conformité aux plans.

Parmi les principales méthodes Agiles on trouve :

- Rapid Application Development (RAD),
- Adaptive Software Development (ASD),
- Extreme Programming (XP),
- Feature Driven Development (FDD),
- SCRUM,
- Dynamic System Development Method (DSDM).

Nous avons opté pour la méthodologie SCRUM pour multiples raisons Quelques avantages de méthode SCRUM :

- Meilleure collaboration entre les équipes.
- Création rapide de valeur de projet.
- Offre plus de réactivité pour ajuster le produit aux réalités.
- Méthode incrémentale et itérative de la gestion de projet.

SCRUM est caractérisé par la simplicité, la pragmatique, la transparence et l'empirique. **«C'est**

un framework qui a fait ses preuves aujourd'hui dans différents contextes, car il permet de produire de la valeur rapidement », Il s'appuie sur une équipe pluridisciplinaire et auto-organisée. Aucun plan fixe n'est établi en avance. L'équipe SCRUM est à la recherche constante de la meilleure façon de produire ce qui a le plus de valeur pour l'utilisateur final. Aussi le développement logiciel est une activité complexe et SCRUM peut faire face à cette complexité par l'empirisme. C'est un processus empirique qui veut dire qu'on se base sur l'expérience pour prendre des décisions.

1.5.3 Principe de SCRUM

SCRUM ne se considère pas comme une méthode mais comme un cadre méthodologique. Une méthode dit généralement « comment » faire les choses. SCRUM ne dit pas comment réussir son logiciel, comment surmonter les obstacles, comment développer, comment spécifier, etc. Il se contente d'offrir un cadre de gestion de projet Agile (et c'est déjà beaucoup) des rôles, un rythme itératif, des réunions précises et limitées dans le temps, des artefacts (product backlog, sprint backlog, graphique d'avancement) et des règles du jeu

1.5.4 Les bases de SCRUM

La méthodologie SCRUM est basée sur 3 principes qui sont au même temps les principes de la culture agile qui sont :

- **Transparence** : Scrum met l'accent sur le fait d'avoir un langage commun.
- **Inspection** : A intervalle régulier, Scrum propose de faire le point sur les différents artefacts produits, afin de détecter toute variation indésirable.
- **Adaptation** : Si une dérive est constatée pendant l'inspection, le processus doit alors être adapté.

1.5.5 Les différentes composantes de SCRUM

- Le « **Product Backlog** » : c'est l'ensemble des fonctionnalités, qui évoluent durant le projet parallèlement aux besoins du client.
- Le « **Bun down chart** » : c'est un graphe, mis à jour à la fin de chaque sprint, montrant la tendance de l'avancement dans la réalisation du produit.
- La « **mêlée** » (**Daily Scrum**) : est une réunion quotidienne de 15 minutes maximum, regroupant l'ensemble de l'équipe, généralement le matin, au cours de laquelle chaque participant décrit ce qu'il a fait la veille.
- Le « **Sprint Backlog** » : c'est une liste de tâches identifiées par l'équipe à remplir pendant un sprint .

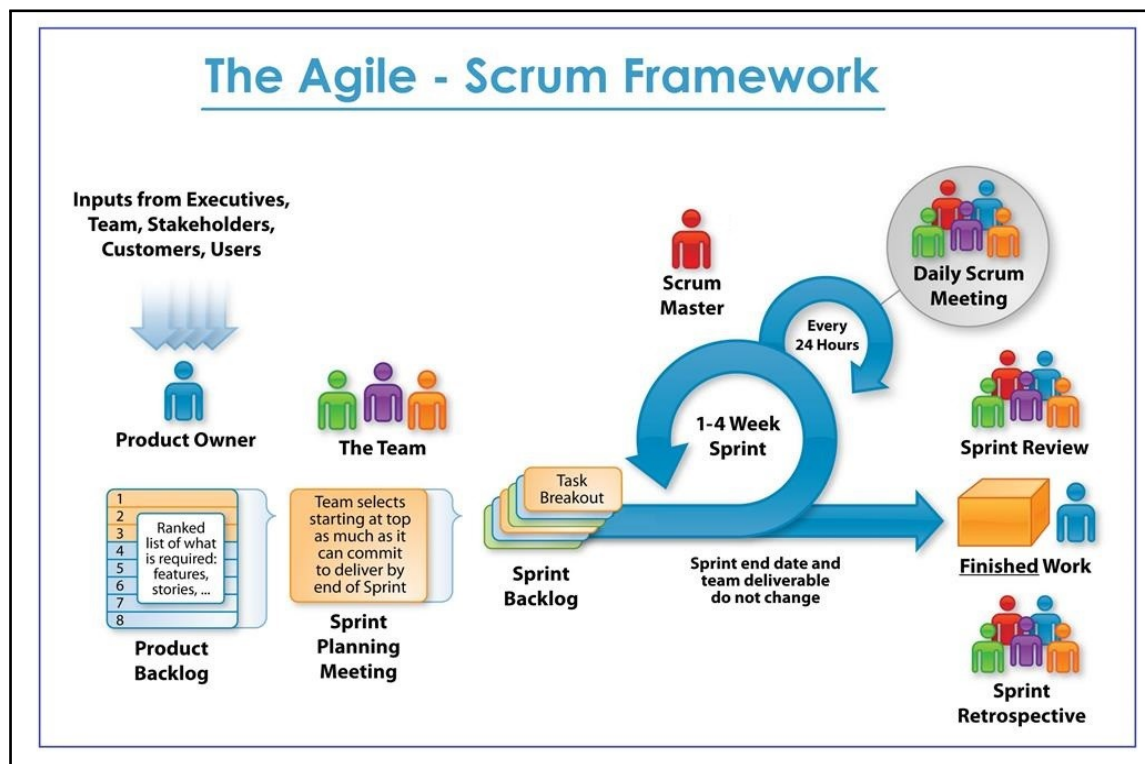


FIGURE 1.6 – Cycle de vie de la méthodologie SCRUM

1.5.6 Les différents intervenants de la méthodologie SCRUM

- « **Product Owner** » : c'est un expert métier qui possède une vision globale et détaillée sur le produit à construire en précisant les spécifications fonctionnelles et l'établissement des priorités des fonctionnalités à développer, en validant et en corrigeant des fonctionnalités développées.
- « **SCRUM Master** » : représenté par le chef du groupe ou du projet qui gère l'équipe en facilitant la communication entre eux et en améliorant la productivité et le savoir-faire de cette équipe. Aussi, il est chargé de veiller à la mise en application de la méthode SCRUM en s'assurant que les principales valeurs de cette méthode sont respectées.
- « **Development Team** » : l'équipe de développement possédant une organisation autonome qui consiste à la réalisation du produit de façon que le regroupement suit son poste ou rôle (Développeur - Concepteur - Architecte - Gestionnaire de l'espace IHM. . .).

Tous ces intervenants travaillent en équipe en vue d'atteindre un objectif de minimisation des risques et surmonter les difficultés rencontrées au cours de la réalisation du projet.

La figure, représente les relations entre les différents intervenants dans SCRUM.

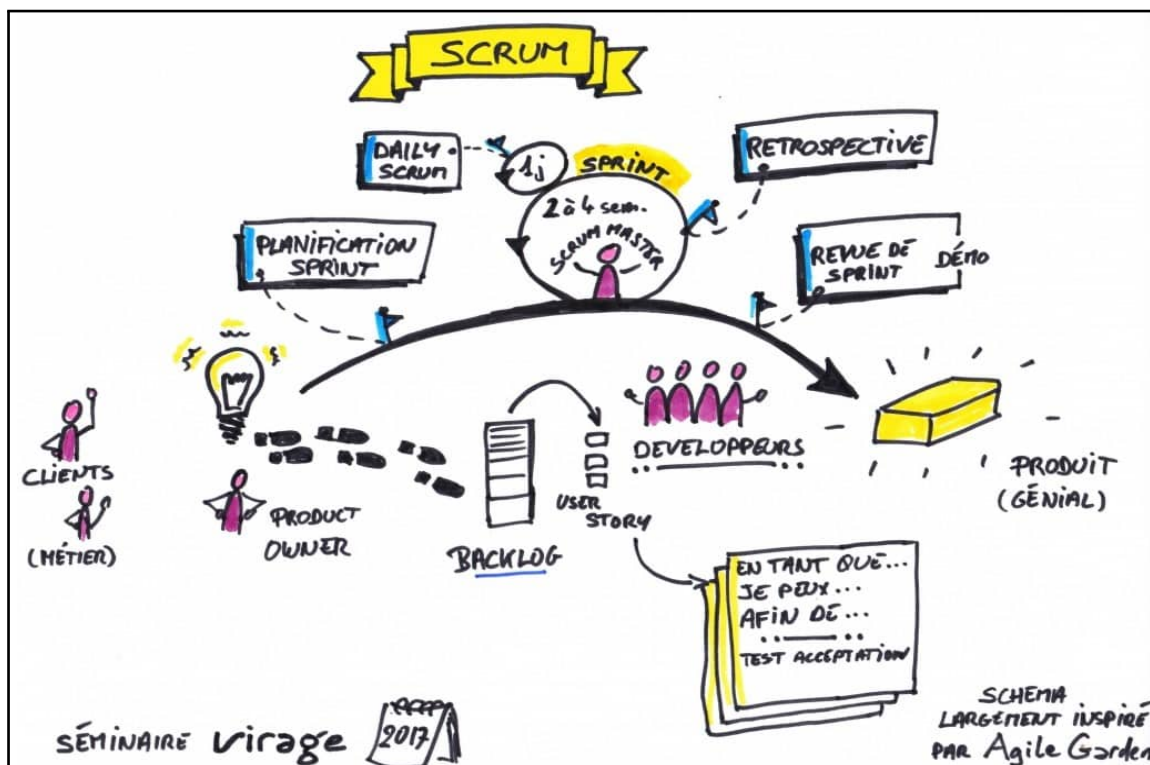


FIGURE 1.7 – Cycle de vie de la méthodologie SCRUM

1.6 Conclusion

Dans ce chapitre nous avons mis notre projet dans son cadre en précisant son environnement et en commençant par l'introduction de l'organisme d'accueil avec une présentation détaillée de ses différents services et de sa clientèle. Ensuite, nous avons présenté la problématique ainsi que la solution proposée. Puis, nous avons justifié nos choix pour la méthodologie du travail.

Chapitre 2 : Analyse et spécification des besoins

Sommaire

2.1	Introduction	19
2.2	Spécification des besoins	19
2.2.1	Identification des acteurs	19
2.2.2	Identification des besoins	19
2.2.3	Besoins fonctionnels	20
2.2.4	Besoins non fonctionnels	21
2.3	Pilotage du projet avec SCRUM	21
2.3.1	Equipe et rôles	22
2.4	Structure et planification des sprints	23
2.4.1	Backlog produit	23
2.5	Détail fonctionnel	25
2.5.1	Diagramme de cas d'utilisation général	25
2.5.2	Diagramme de classe globale	30
2.6	Architecture du système	31
2.6.1	Architecture logicielle	31
2.6.2	Architecture logique	32

2.1 Introduction

Ce deuxième chapitre a pour objectif principal d'assurer le bon commencement du projet dans les conditions les plus idéales. Dans ce chapitre,nous allons tout d'abord identifier nos acteurs et puis nous allons détailler les besoins fonctionnels et les besoins non fonctionnels. Puis, nous allons détailler le pilotage du projet avec SCRUM et nous allons présenter le diagramme de cas d'utilisation général. Aussi, dans ce chapitre, nous allons préciser l'architecture système utilisé et nous terminons par le product backlog pour définir l'ensemble de fonctionnalités du produit que nous voulons réaliser.

2.2 Spécification des besoins

Nous commençons par identifier les acteurs puis on va extraire les exigences fonctionnelles et non fonctionnelles.

2.2.1 Identification des acteurs

Un acteur est l'idéalisation d'un rôle joué par une personne externe ou un processus qui interagit avec un système les acteurs de notre application sont :

- **Administrateur** : La personne responsable de l'administration du système.
- **Apprenants** : La personne qui a besoin d'acquérir des connaissances et des compétences.
- **Enseignant/Formateur** : La personne qui assure une formation.

2.2.2 Identification des besoins

En vue d'atteindre notre objectif, qui consiste à concevoir et développer l'application précédemment décrite, nous allons tout d'abord identifier les acteurs ainsi que les besoins fonctionnels et non fonctionnels.

2.2.3 Besoins fonctionnels

Les besoins fonctionnels expriment les fonctionnalités du futur système. Nous listons, ci-dessous, les fonctionnalités de notre système par acteur .

Notre site e-learning permet de réaliser les tâches suivantes pour répondre aux besoins du formateur et des apprenants à la fois et de manière à :

- Garantir plus de souplesse pour la planification des formations à l'aide d'une interface ergonomique et simple à manipuler pour les différents utilisateurs.
- Faciliter l'accès aux ressources et aux services.
- Faciliter l'échange et la collaboration à distance.
- Améliorer la qualité d'apprentissage en utilisant les nouvelles technologies multimédia.
- Optimiser le temps de réponse de l'application.
- Assurer la sécurité d'information.

a. Les besoins de l'administrateur :

- S'authentifier.
- Gérer les utilisateurs (apprenant, formateur).
- Gérer les formations existantes.
- Suivre les paiements.

b. Les besoins de l'apprenant :

Notre plateforme permet à un nouvel utilisateur « apprenant » la création d'un compte à partir d'un formulaire d'inscription.

Ainsi, elle lui permet de :

- S'authentifier.
- Gérer son profil (mettre à jour ses informations personnelles).
- Evaluer et commenter un cours.

c. Les besoins du formateur :

Notre plateforme offre aux formateurs la possibilité de :

- S'authentifier
- Gérer son profil (mettre à jour ses informations personnelles et sa liste des cours)

- Gérer les cours (sous forme : PDF, vidéo, image).
- Gérer les modules (sous forme : PDF, vidéo, image).

2.2.4 Besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques et ergonomiques auxquelles est soumise l'application pour sa réalisation et pour son bon fonctionnement.

Les principaux besoins non fonctionnels de notre application sont :

- **La sécurité** : l'application doit mettre en disposition le critère de confidentialité des données clients, plus précisément, au niveau de l'authentification. Pour dépasser ses obstacles, nous accordons l'accès libre à ces informations uniquement à l'administrateur.
- **Facilité de l'utilisation** : l'application doit être facile et simple à mettre en œuvre par n'importe quel utilisateur.
- **La fiabilité** : le système doit fonctionner sans défaillance dans des conditions spécifiques.
- **La convivialité** : l'interface est le premier contact de l'utilisateur avec l'application. Donc elle doit être simple, claire et conviviale.
- **La performance** : l'application doit garantir le critère de performance et réagir dans un délai minimum quelle que soit l'action de l'utilisateur.
- **La maintenance** : L'application doit être maintenable afin d'assurer l'évolution rapide de l'application.

2.3 Pilotage du projet avec SCRUM

Nous allons présenter dans cette section, les outils de SCRUM, l'équipe de travail ainsi que leurs rôles et le product backlog .

2.3.1 Equipe et rôles

Scrum est constituée d'un propriétaire de produit, de l'équipe de développement et d'un Scrum Master. Nous présentons dans cette partie l'ensemble des acteurs participants au déroulement des différentes phases du projet et d'élaboration du rapport du stage. Dans notre cas, l'équipe de développement est constituée La méthodologie SCRUM fait intervenir trois rôles principaux qui sont :

Scrum définit essentiellement trois rôles distincts :

SCRUM Master : est un cadre qui a pour rôle principal de diriger, d'orienter, de guider l'équipe durant tout le cycle du développement du logiciel.

Product Owner : est un cadre qui a pour rôle principal de mener bien un projet selon les méthodes agiles et qui possédant un savoir professionnel dans la gestion des équipes.

SCRUM Team : c'est l'équipe de développement.

Dans le contexte de notre projet , on trouve :

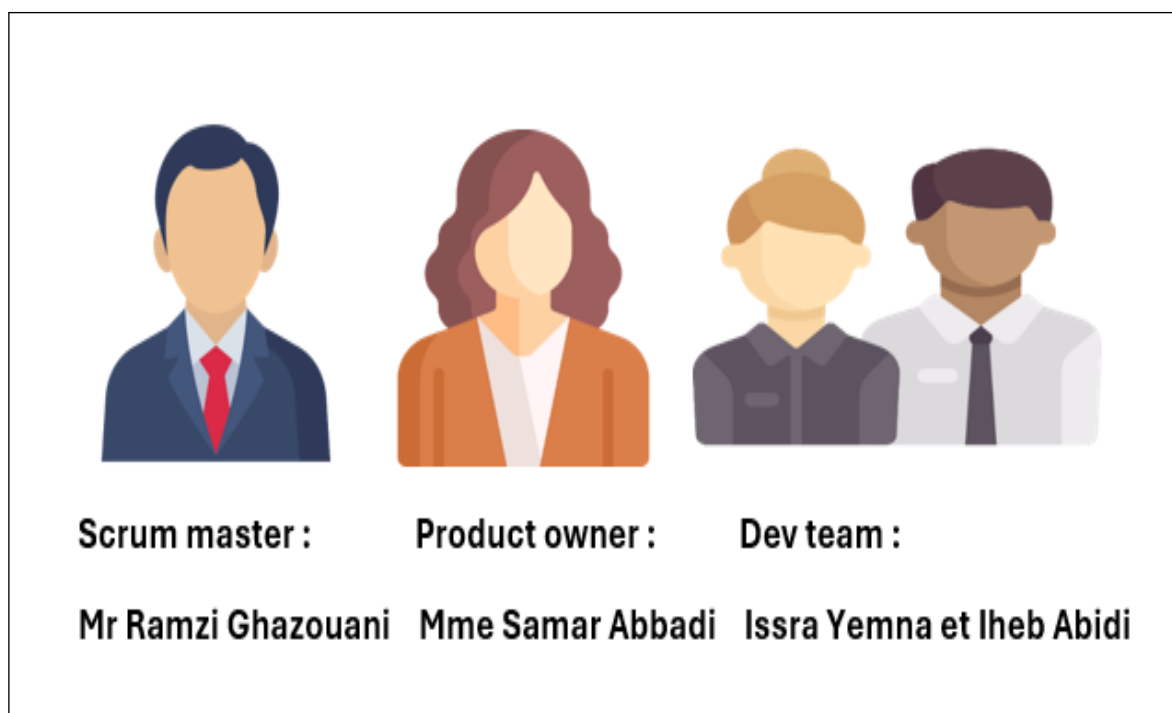


FIGURE 2.1 – Equipe SCRUM et rôles

2.4 Structure et planification des sprints

Après avoir défini le backlog du produit, nous découpons notre projet en des sprints. Le sprint comporte les tâches d'analyses, de conception, de développement et de tests. Nous allons établir la planification du projet en 4 sprints. Chaque sprint peut durer de 4 à 5 semaines (Figure 2.3)

2.4.1 Backlog produit

Le backlog est une liste de taches priorisées définissant les caractéristiques d'un produit. Il est un des éléments fondamentaux de la méthodologie Scrum. Il s'agit de l'outil de travail principal du Product Owner qui se charge de recueillir les besoins auprès des parties prenantes et de les transformer en liste de fonctionnalités prêtes à être développées par l'équipe de développement. Ce tableau montre les différentes tâches à réaliser dans notre projet avec une description. Nous avons aussi affecté une priorité selon l'importance de la réalisation.

- **Id** : représente l'identifiant et auto-incrémenté pour chaque user story.
- **User story** : est la description fonctionnelle utilisée dans les méthodes agiles. L'objectif est de spécifier le développement d'une fonctionnalité. Elle doit alors exprimer à qui elle s'adresse et en quoi elle apporte la valeur
- **Priorité** : c'est la valeur métier qui dirige la priorisation du développement des histoires utilisateurs suivants les attentes et les besoins du client .

Dans le tableau ci-dessous, nous présentons la liste des récits (user story) ainsi que leurs estimations.

ID	Fonctionnalité	User Story	Propriété
1	S'inscrire	En tant qu'utilisateur, je veux remplir un formulaire afin d'être inscrit	1
2	S'authentifier	En tant qu'utilisateur, je veux m'authentifier pour accéder à mon espace.	2
3	Lister cours	En tant qu'utilisateur, je veux consulter les cours disponibles.	3
4	Gérer profil	En tant qu'utilisateur, je veux modifier les informations décrivant mon profil	4
5	Gérer compte	En tant qu'admin, je veux gérer les comptes des utilisateurs.	5
		En tant qu'admin, je veux supprimer les comptes des utilisateurs.	5.1
		En tant qu'admin, je veux ajouter des comptes pour les utilisateurs.	5.2
		En tant qu'admin, je veux modifier les informations des utilisateurs.	5.3
		En tant qu'admin, je veux désactiver des comptes des utilisateurs.	5.4
6	Gérer cours	En tant que formateur, je veux ajouter des cours.	6
		En tant que formateur qui a ajouté des cours, je veux les modifier.	6.1
		En tant que formateur qui a ajouté des cours, je veux les supprimer.	6.2
		En tant qu'administrateur, je veux approuver les cours ajoutés par les formateurs.	6.3
7	Gérer panier	En tant qu'utilisateur, je veux consulter mon panier.	7
		En tant qu'utilisateur, je veux ajouter des cours au panier.	7.1

		En tant qu'utilisateur, je veux vider mon panier.	7.2
		En tant qu'utilisateur, je veux supprimer un cours de mon panier.	7.1
8	Règlement	En tant qu'utilisateur, je veux consulter les offres des cours existantes pour payer.	8
		En tant qu'admin, je veux consulter la liste des paiements.	8.1
9	Gérer favoris	En tant qu'utilisateur, je veux ajouter un cour aux favoris.	9
		En tant qu'utilisateur, je veux supprimer un cour aux favoris.	9.1
		En tant qu'utilisateur, je veux consulter liste des favoris.	9.2
10	Gérer quiz	En tant que formateur, je veux ajouter des quiz à mon cours.	10
11	Commenter	En tant qu'utilisateur, je veux ajouter des commentaires.	11
		En tant qu'utilisateur, je veux ajouter des commentaires.	11.1

TABLE 2.1 – Product Backlog

2.5 Détail fonctionnel

2.5.1 Diagramme de cas d'utilisation général

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet. Mais pour le développement,

les cas d'utilisation sont plus appropriés.

L'objectif principal de ce diagramme est l'énumération des différents cas d'utilisation en précisant les relations des acteurs ainsi que leurs tâches dans notre application. La figure 2.1, représente le diagramme de cas d'utilisation général de notre application.

2.5.1.1 Diagramme de cas d'utilisation général de l'apprenant



FIGURE 2.2 – Diagramme de cas d'utilisation général de l'apprenant

2.5.1.2 Diagramme de cas d'utilisation général de formateur

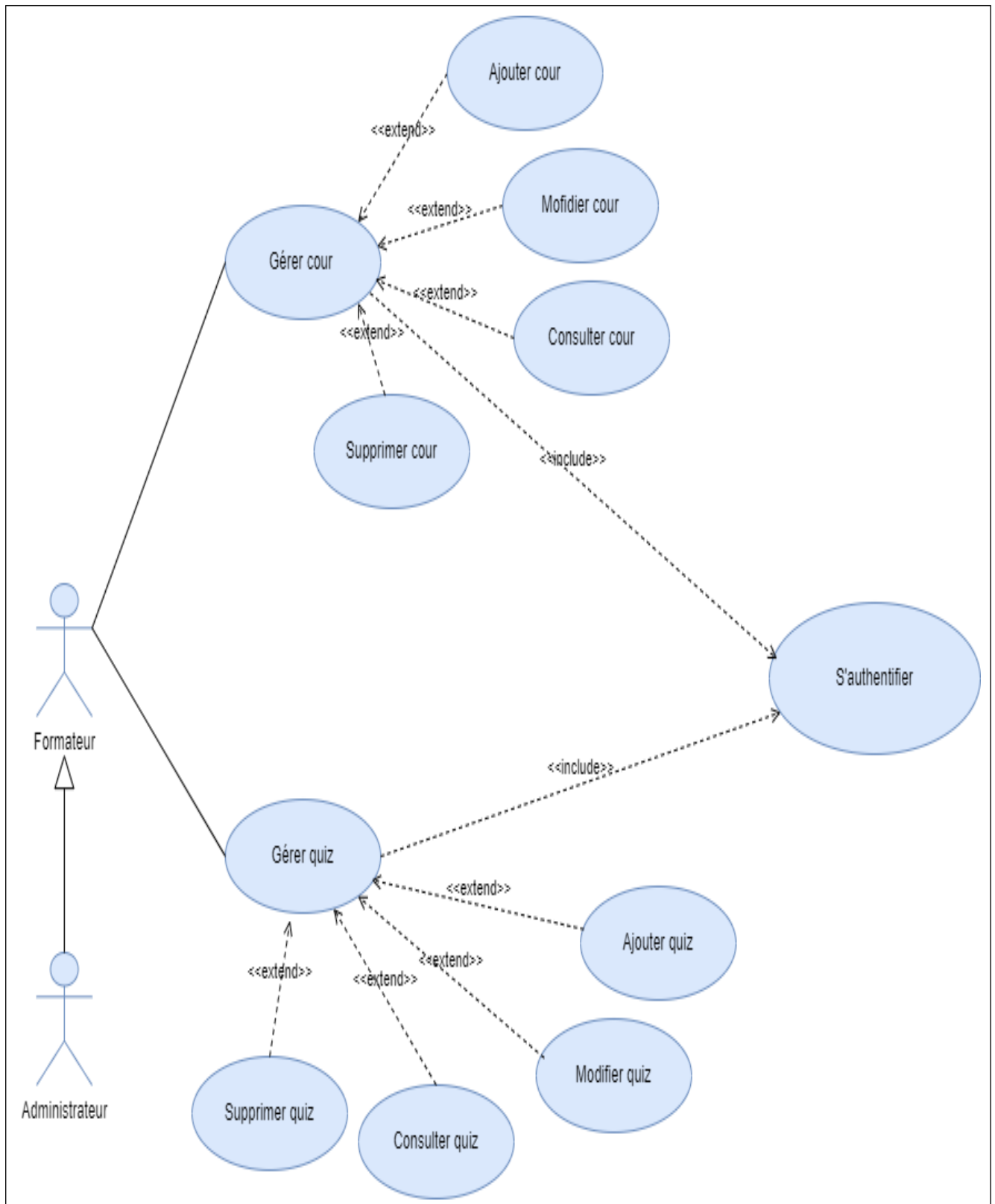


FIGURE 2.3 – Diagramme de cas d'utilisation général de formateur

2.5.1.3 Diagramme de cas d'utilisation général de l'administrateur

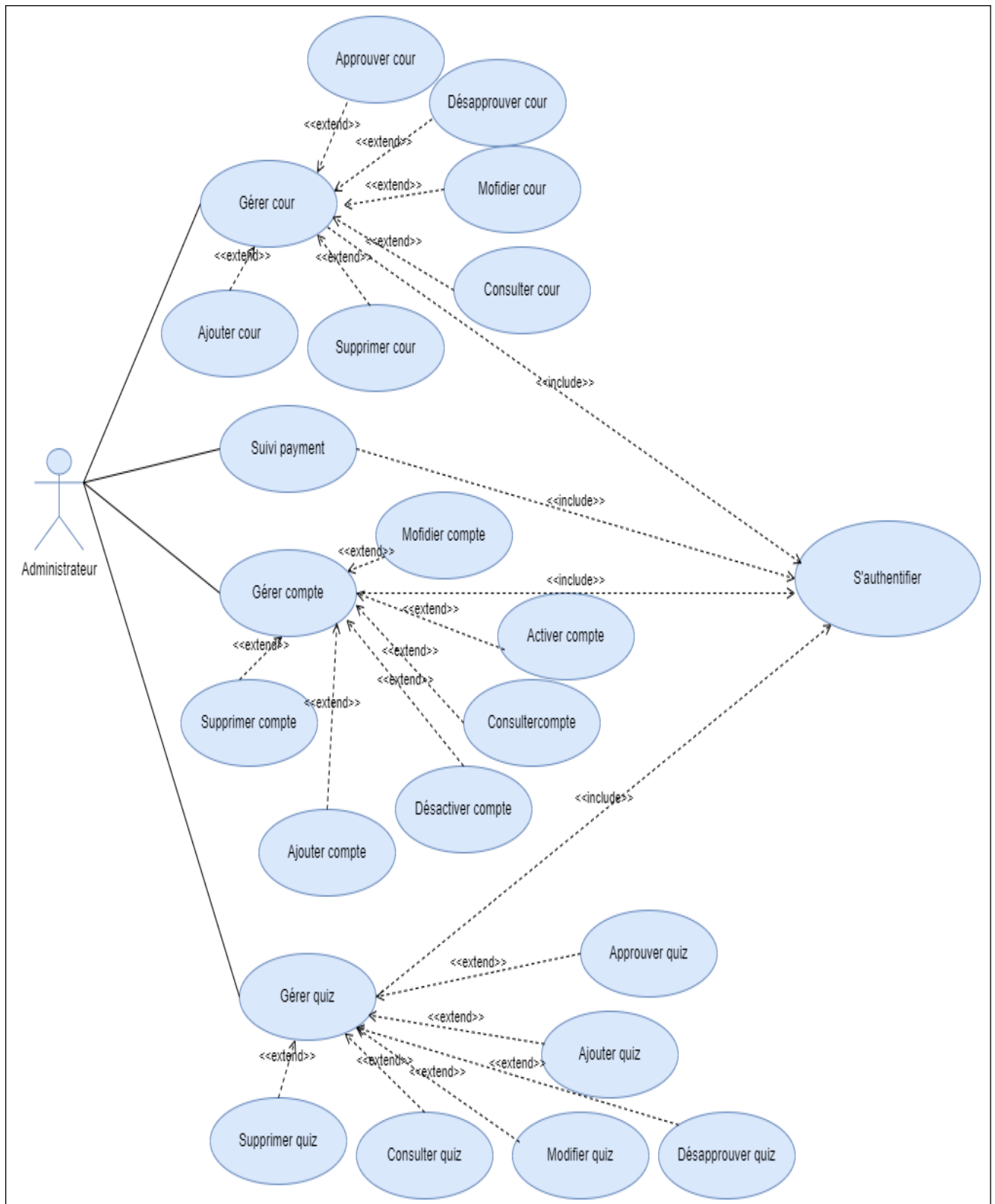


FIGURE 2.4 – Diagramme de cas d'utilisation général de l'administrateur

2.5.2 Diagramme de classe globale

2.6 Architecture du système

2.6.1 Architecture logicielle

L'architecture logicielle MVC (Modèle-Vue-Contrôleur) est un modèle de conception couramment utilisé dans le développement d'applications logicielles, notamment pour les applications web et les applications basées sur une interface utilisateur.

- Le modèle MVC sépare les responsabilités en trois composants principaux :

Modèle (Model) : Le modèle représente les données et la logique métier de l'application.

Il gère l'accès, la manipulation et la persistance des données. Il peut également inclure des règles de validation et de traitement des données.

Vue (View) : La vue est responsable de la présentation des données au client ou à l'utilisateur.

Elle affiche les informations provenant du modèle de manière compréhensible et interagit avec l'utilisateur pour afficher les résultats des actions. La vue ne contient généralement pas de logique métier complexe.

Contrôleur (Controller) : Le contrôleur gère les interactions entre l'utilisateur, la vue et le modèle. Il reçoit les entrées de l'utilisateur à travers la vue, interroge et met à jour les données du modèle en conséquence, puis transmet les résultats à la vue pour affichage. Le contrôleur agit comme un intermédiaire qui orchestre les actions et maintient la cohérence entre le modèle et la vue.

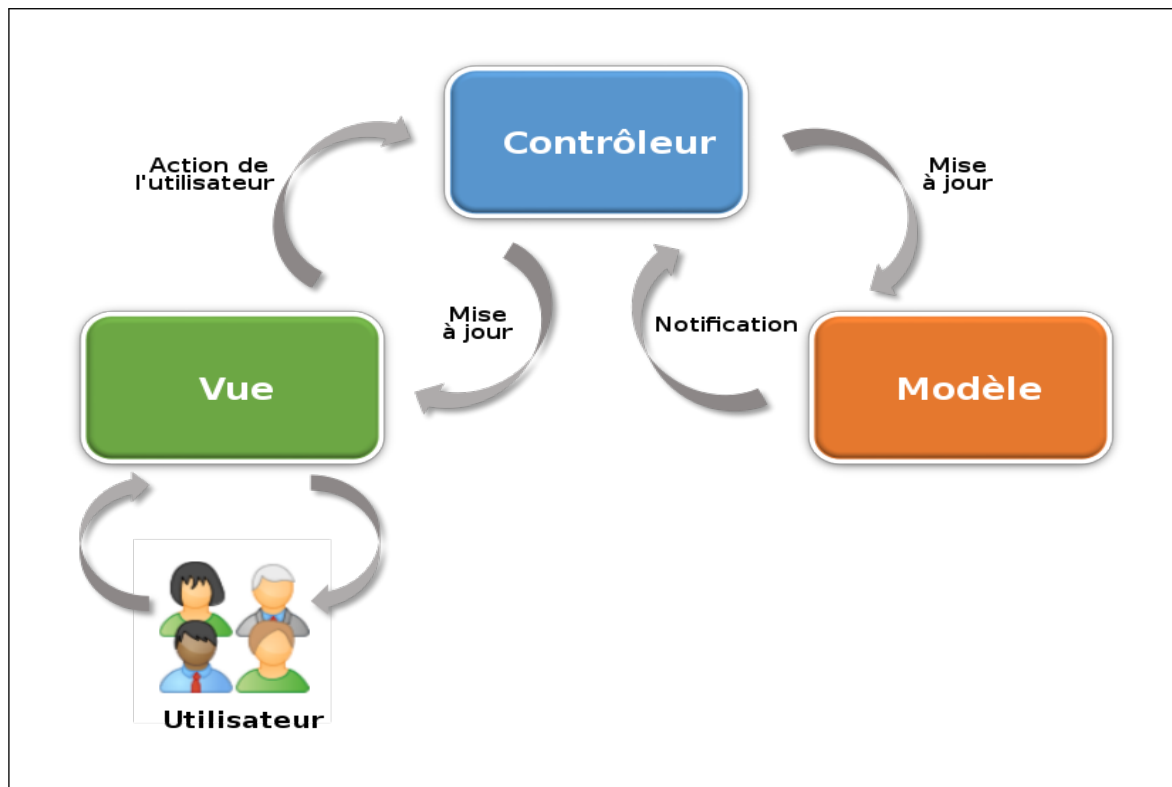


FIGURE 2.5 – Architecture de MVC

2.6.2 Architecture logique

L'architecture de l'application est une architecture trois-tiers basée sur le modèle MVC.

L'architecture trois-tiers est un modèle logique d'architecture applicative qui vise à séparer très nettement trois couches logicielles au sein d'une même application ou système, à modéliser et présenter cette application comme un empilement de trois couches, étages, niveaux ou strates dont le rôle est clairement défini :

- **La couche présentation des données (client) :** Elle correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle d'Interface Homme Machine. En informatique, elle peut être réalisée par une application graphique ou textuelle. Elle peut aussi être représentée . en HTML pour être exploitée par un navigateur web ou en WML pour être utilisée par un téléphone portable.
- **La couche application (serveur d'application) :** correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative .
- **La couche d'accès aux données persistantes (serveur de base de données) :**

correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

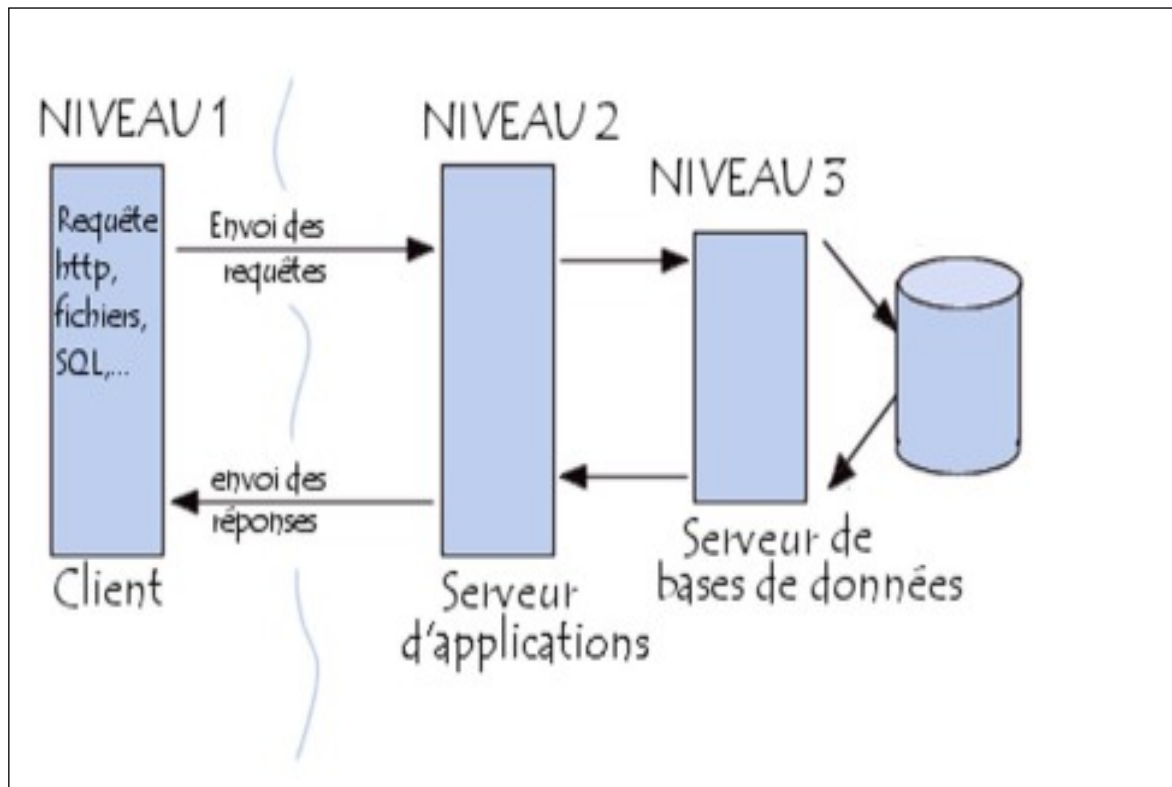


FIGURE 2.6 – Architecture 3-tiers

Sprint 1 : Mise en oeuvre de sprint 1

Sommaire

3.1	Introduction	35
3.2	Backlog du Sprint 1	35
3.2.1	Définition d'un backlog de Sprint	35
3.2.2	But de Sprint	35
3.2.3	User Stories	35
3.3	Spécification fonctionnelle	36
3.3.1	Diagramme de cas d'utilisation du premier sprint	36
3.3.2	Description textuelle et raffinement des principaux cas d'utilisation	37
3.4	Conception	42
3.4.1	Diagrammes de séquences détaillés	42
3.4.2	Diagramme de séquence lié à la fonctionnalité « S'inscrire »	43
3.4.3	Diagramme de séquence lié à la fonctionnalité « S'authentifier »	44
3.4.4	Diagramme de séquence lié à la fonctionnalité « Activer compte »	45
3.4.5	Diagramme de séquence lié à la fonctionnalité « Ajouter compte »	46
3.4.6	Diagramme de séquence lié à la fonctionnalité « Modifier compte »	47
3.4.7	Diagramme de séquence lié à la fonctionnalité « Supprimer compte »	48
3.5	Réalisation	49
3.5.1	Interface d'authentification	49
3.5.2	Interface d'inscription	50
3.6	Conclusion	50

3.1 Introduction

Dans le chapitre précédent, nous avons réussi à établir une liste de fonctionnalités découpées en quatre sprints. Ces fonctionnalités seront traitées selon la répartition qui a été faite. Dans ce chapitre, nous allons décrire la réalisation du premier sprint en introduisant les spécifications fonctionnelles, la conception, l'implémentation et les tests.

3.2 Backlog du Sprint 1

3.2.1 Définition d'un backlog de Sprint

Un backlog de sprint est une liste de tâches que votre équipe souhaite accomplir au cours d'un sprint de projet. Ce programme de travail est élaboré au cours d'une séance de planification de sprint et s'appuie sur les éléments qui figurent dans votre backlog produit. Concevoir un backlog de sprint permet d'indiquer clairement à votre équipe sur quelles tâches se concentrer pendant chaque sprint (et celles qu'il convient de laisser de côté pour le moment) afin d'éviter toute dérive des objectifs.

3.2.2 But de Sprint

Le but de ce sprint est de permettre aux clients de s'inscrire pour accéder à l'application et s'authentifier, et également de permettre aux administrateur et responsable d'authentifier aussi et de permettre au administrateur de gérer les services et au responsable de gérer les abonnements .

3.2.3 User Stories

Le Sprint est le coeur de Scrum :

- Bloc de temps dans lequel un incrément de produit sera réalisé .
- Tous les sprints ont une durée constante .

- Un sprint ne peut pas démarrer tant que la précédente sprint n'est pas encore terminé .

Avant de se lancer dans un sprint , l'équipe Scrum doit obligatoirement définir le backlog de sprint : Le but de ce dernier qui doit être un tableau descriptif qui précise la charge du travail pour chaque tâche en nombre de jours .

TABLE 3.1 – Product Backlog Sprint 1

ID	User story	Taches	Durée
1	En tant que apprenant, je veux remplir un formulaire afin d'être inscrit	S'inscrire	3 jours
2	En tant que apprenant je veux m'authentifier pour accéder à mon espace.	S'authentifier	2 jours
3	En tant que administrateur, je veux approuver les comptes.	Gérer compte	2 jours

3.3 Spécification fonctionnelle

Pour la mise en marche de notre sprint nous avons tout d'abord élaboré le diagramme de cas d'utilisation de notre sprint avec le raffinement de chaque cas d'utilisation .

3.3.1 Diagramme de cas d'utilisation du premier sprint

En vue d'atteindre l'objectif de la réalisation de notre logiciel, nous établirons une description détaillée des différentes fonctionnalités.

Toutes ces spécifications seront décrites dans les diagrammes de cas d'utilisations global et raffiné, dans la description textuelle des principaux cas , Comme le montre ce diagramme

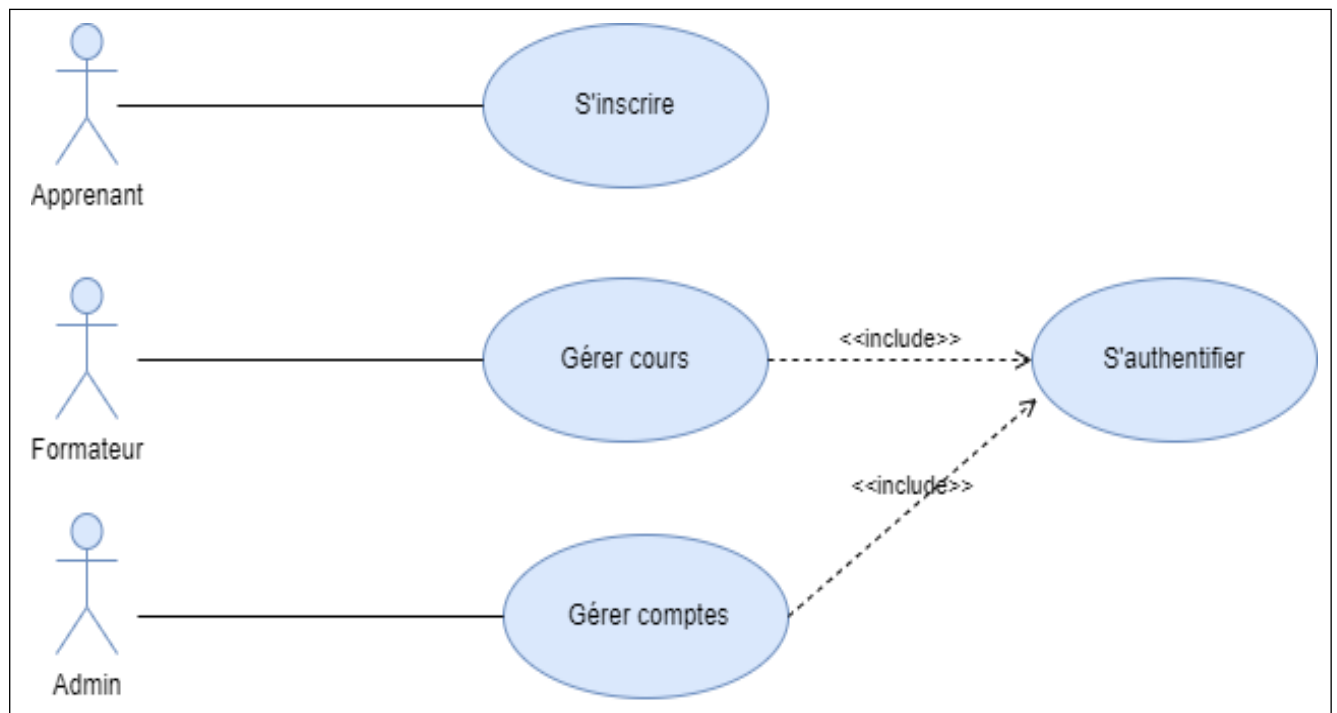


FIGURE 3.1 – Diagramme de cas d'utilisation globale de sprint 1

3.3.2 Description textuelle et raffinement des principaux cas d'utilisation

Après la réalisation du backlog du premier sprint et pour que le déroulement des fonctionnalités des cas d'utilisations soit plus simple, claire et facile à comprendre, nous allons introduire une description textuelle et un raffinement des principaux cas d'utilisation.

3.3.2.1 Raffinement du cas d'utilisation "S'inscrire"

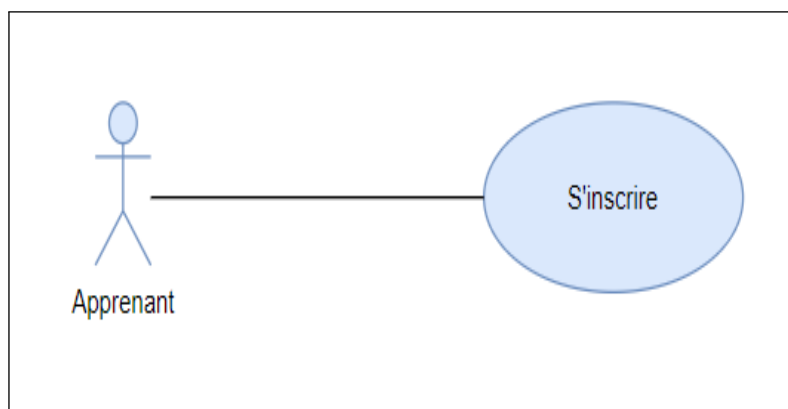


FIGURE 3.2 – Diagramme de cas d'utilisation globale de sprint 1

3.3.2.2 Description textuelle du cas d'utilisation "S'inscrire"

TABLE 3.2 – Description textuelle du cas d'utilisation "S'inscrire"

Cas d'Utilisation	S'inscrire
Acteurs	Apprenant
Précondition	Interface d'inscription affiché
Postcondition	Apprenant inscrit
Scénario nominal	<ol style="list-style-type: none"> 1.L'apprenant demande la page d'inscription. 2.Le système lui affiche la page demandée. 3. L'apprenant remplit le formulaire d'inscription. 4. L'apprenant clique sur bouton "S'inscrire". 5. Le système vérifie les données saisies. 6. Le système ajoute l'apprenant dans la base de données et affiche le message «compte crée avec succès» .
Scénario alternatif	<ol style="list-style-type: none"> 5-a les données saisies sont incorrectes ou manquant 1. le système affiche un message d'erreur et signale au client de recommencer la saisie . 5-b le client existe déjà dans la base de données : 1.Le système affiche un message d'erreur :«compte déjà crée» .

3.3.2.3 Raffinement du cas d'utilisation "S'authentifier"

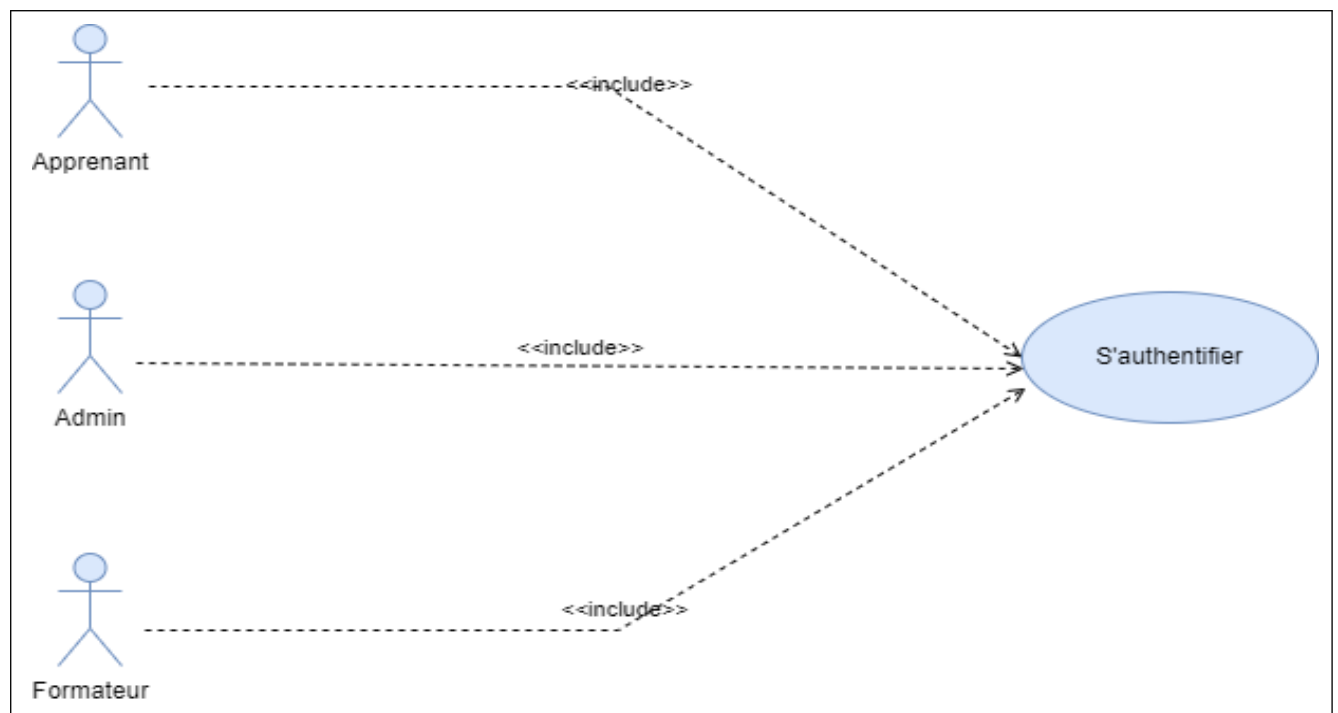


FIGURE 3.3 – Raffinement du cas d'utilisation s'authentifier

3.3.2.4 Description textuelle du cas d'utilisation "S'authentifier"

TABLE 3.3 – Description textuelle du cas d'utilisation "S'authentifier"

Cas d'Utilisation	S'authentifier
Acteurs	Utilisateur (Administrateur, Formateur , Apprenant)
Précondition	Interface d'authentification affiché
Postcondition	Utilisateur authentifié
Scénario nominal	<ol style="list-style-type: none"> 1.l'utilisateur demande la page d'authentification. 2. Le system lui affiche la page demandée. 3. l'utilisateur saisit son adresse e-mail et son mot de passe 4. l'utilisateur valide l'authentification. 5. Le système vérifie les données saisies . 6. Le système affiche l'espace personnel ..
Scénario alternatif	<ol style="list-style-type: none"> 4.a l'utilisateur n'a pas saisi les bons identifiants (contrôle de saisie) : <ol style="list-style-type: none"> 1.le système affiche un message d'erreur et signale à l'utilisateur de recommencer la saisie 4.b les identifiants saisis n'existent pas dans la base de données (contrôle d'existence de l'utilisateur) : <ol style="list-style-type: none"> 1. Le système affiche le message d'erreur suivant « vous devez saisir des identifiants valides ou s'inscrire »

3.3.2.5 Raffinement du cas d'utilisation"Gérer comptes"

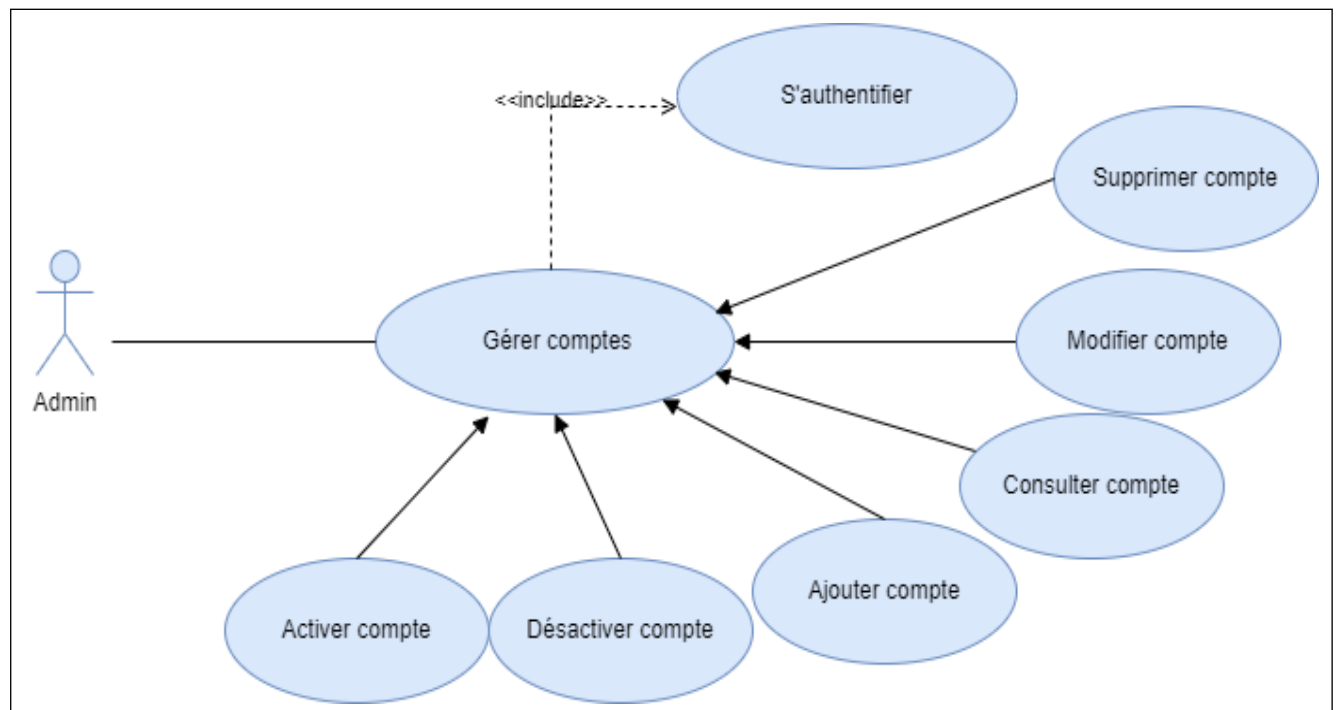


FIGURE 3.4 – Raffinement du cas d'utilisation Gérer comptes

3.3.2.6 Description textuelle du cas d'utilisation "Activer Compte"

TABLE 3.4 – Description textuelle du cas d'utilisation "Activer compte"

Cas d'Utilisation	Activer compte
Acteurs	Administrateur
Précondition	Administrateur doit être authentifié Administrateur doit accéder à l'interface gestion utilisateurs
Postcondition	Compte activé
Scénario nominal	1.L'administrateur clique sur Gestion des utilisateurs . 2.Le système affiche l'interface de liste utilisateurs. 3.L'administrateur clique sur le bouton "Activer compte" 4.Le système affiche une alerte "voulez vous activer ce compte" 5.L'administrateur clique sur "Oui". 6.Le système affiche les informations de ce utilisateur en couleur verte
Scénario alternatif	l'administrateur quitte la page gestion d'utilisateur : Le cas d'utilisation se termine avec échec

3.3.2.7 Description textuelle du cas d'utilisation "Désactiver Compte"

TABLE 3.5 – Description textuelle du cas d'utilisation "Désactiver compte"

Cas d'Utilisation	Désactiver compte
Acteurs	Administrateur
Précondition	Administrateur doit être authentifié Administrateur doit accéder à l'interface gestion utilisateurs
Postcondition	Compte désactivé
Scénario nominal	1.L'administrateur clique sur Gestion des utilisateurs . 2.Le système affiche l'interface de liste utilisateurs. 3.L'administrateur clique sur le bouton "Désactiver compte" 4.Le système affiche une alerte "voulez vous Désactiver ce compte" 5.L'administrateur clique sur "Oui". 6.Le système affiche les informations de ce utilisateur en couleur rouge
Scénario alternatif	l'administrateur quitte la page de gestion d'utilisateur : Le cas d'utilisation se termine avec échec

3.3.2.8 Description textuelle du cas d'utilisation "Ajouter compte"

TABLE 3.6 – Description textuelle du cas d'utilisation "Désactiver compte"

Cas d'Utilisation	Ajouter compte
Acteurs	Administrateur
Précondition	Administrateur doit être authentifié Administrateur doit accéder à l'interface gestion utilisateurs
Postcondition	Compte ajouté
Scénario nominal	1.L'admin demande le formulaire d'ajout d'un nouveau compte. 2.Le système lui retourne le formulaire d'ajout. 3.L'administrateur remplit le formulaire et demande l'enregistrement 4.Le système vérifie les données et enregistre le nouveau compte. 5.Le système affiche une alerte de succès.
Scénario alternatif	Le système affiche un message d'erreur dans le cas où le formulaire est mal saisi ou le compte existe déjà.

3.3.2.9 Description textuelle du cas d'utilisation « Modifier compte »

TABLE 3.7 – Description textuelle du cas d'utilisation "Modifier compte"

Cas d'Utilisation	Modifier compte
Acteurs	Administrateur
Précondition	Administrateur doit être authentifié Administrateur doit accéder à l'interface gestion utilisateurs
Postcondition	Compte modifié
Scénario nominal	1.L'administrateur demande le formulaire de modification d'un compte.. 2.Le système lui retourne la liste des comptes. 3.L'administrateur sélectionne le compte à modifier. 4.Le système vérifie les données et enregistre le nouveau compte.Le système lui retourne les informations de ce compte 5.L'administrateur met à jour les informations et demande l'enregistrement. 6. Le système vérifie les données et enregistre le nouveau compte. 7.Le système affiche une alerte de succès
Scénario alternatif	Le système affiche un message d'erreur dans le cas où le formulaire est mal saisi .

3.3.2.10 Description textuelle du cas d'utilisation "Supprimer compte"

TABLE 3.8 – Description textuelle du cas d'utilisation "Supprimer compte"

Cas d'Utilisation	Supprimer compte
Acteurs	Administrateur
Précondition	Administrateur doit être authentifié Administrateur doit accéder à l'interface gestion utilisateurs
Postcondition	Compte supprimé
Scénario nominal	1.L'admin demande la liste des comptes. 2.Le système lui retourne la liste des comptes. 3.L'administrateur sélectionne le compte à supprimer et demande la suppression. 4.Le système met à jour la base de données. 5.Le système affiche une alerte de succès.
Scénario alternatif	Le système affiche un message d'erreur

3.4 Conception

Dans cette section , nous présentons les différents diagrammes de séquence détaillés de ce sprint .

3.4.1 Diagrammes de séquences détaillés

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un cas d'utilisation.

3.4.2 Diagramme de séquence lié à la fonctionnalité « S’inscrire »

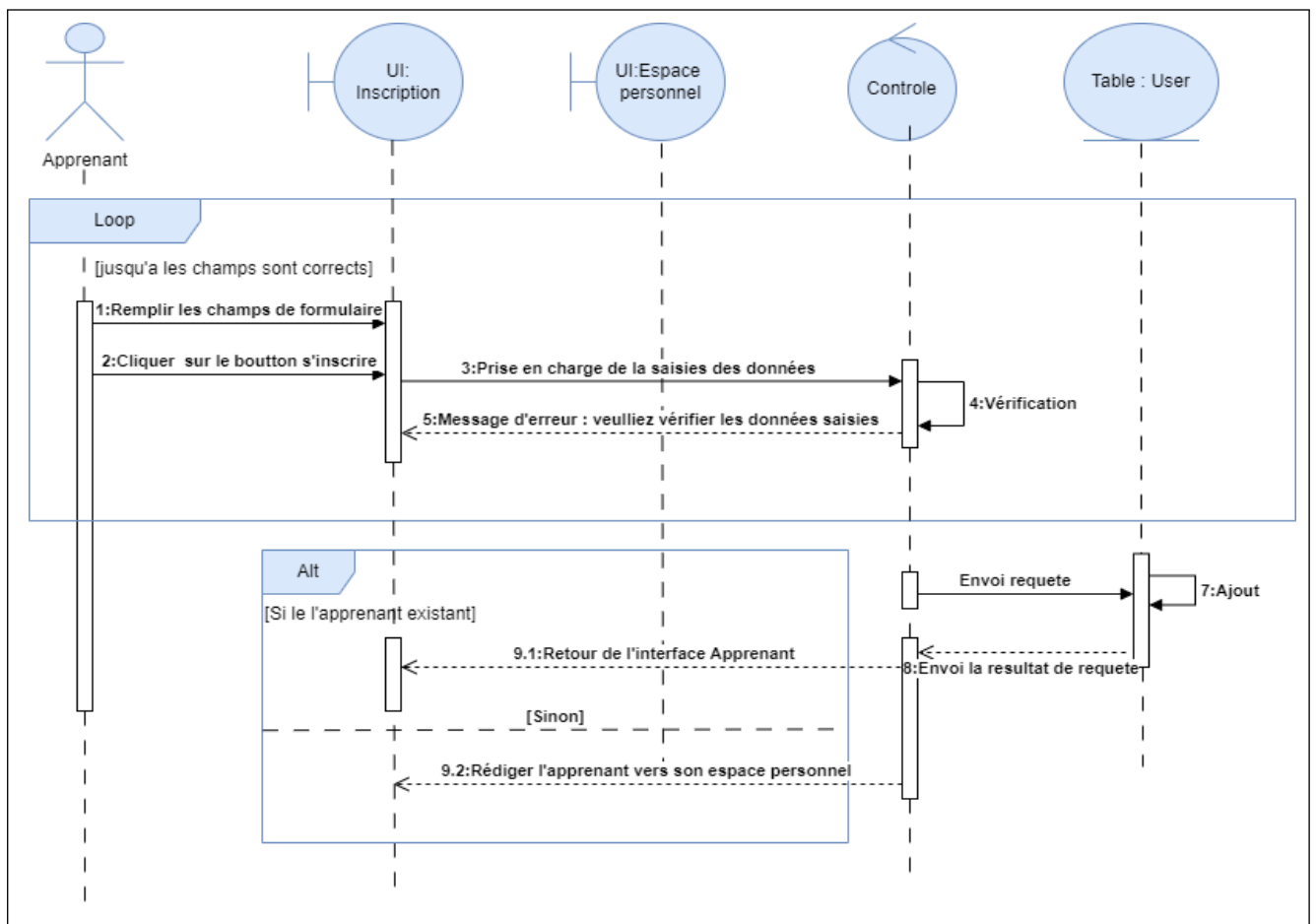


FIGURE 3.5 – Diagramme de séquence lié à la fonctionnalité « S’inscrire »

3.4.3 Diagramme de séquence lié à la fonctionnalité « S'authentifier »

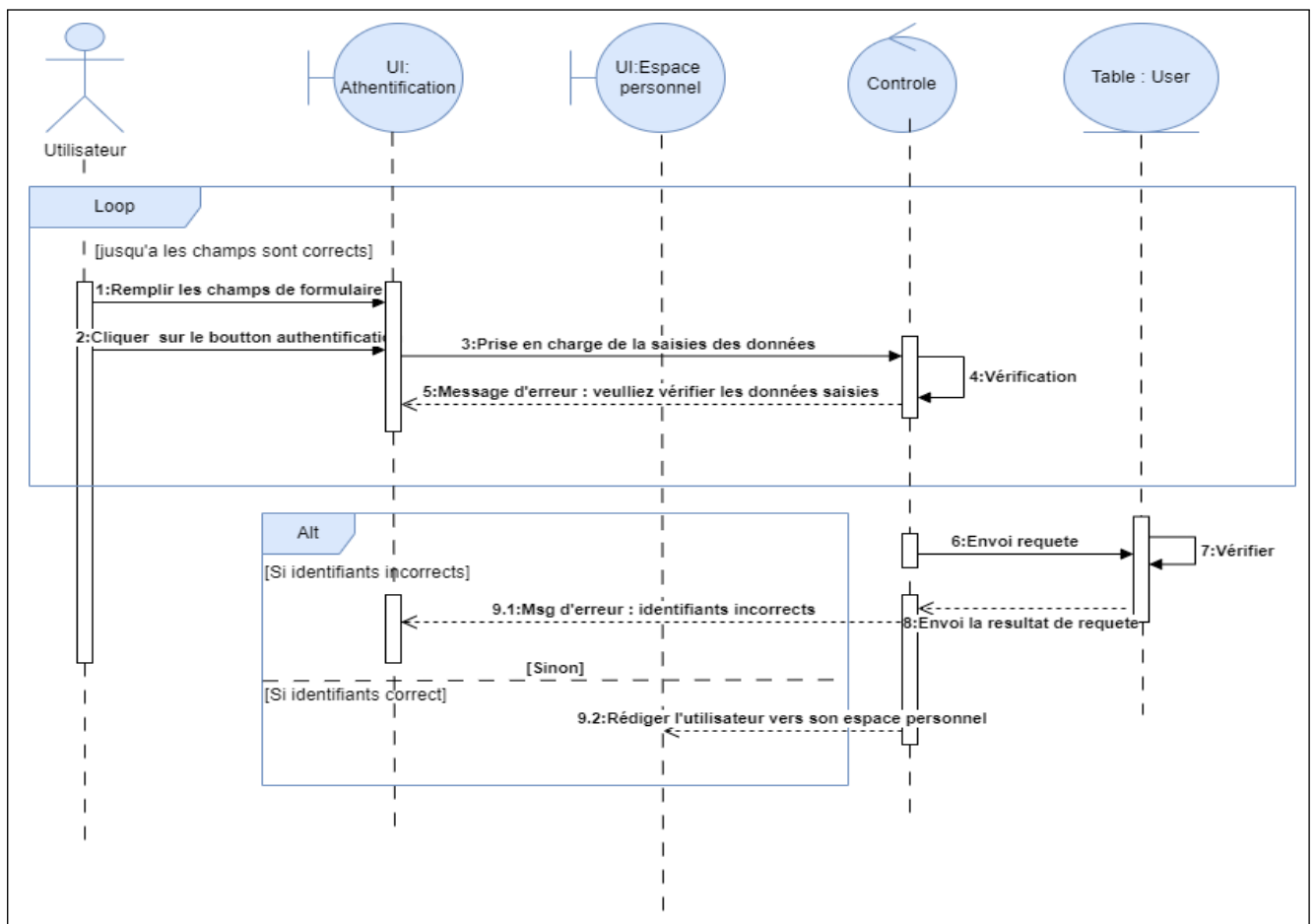


FIGURE 3.6 – Diagramme de séquence lié à la fonctionnalité « S'authentifier »

3.4.4 Diagramme de séquence lié à la fonctionnalité « Activer compte »

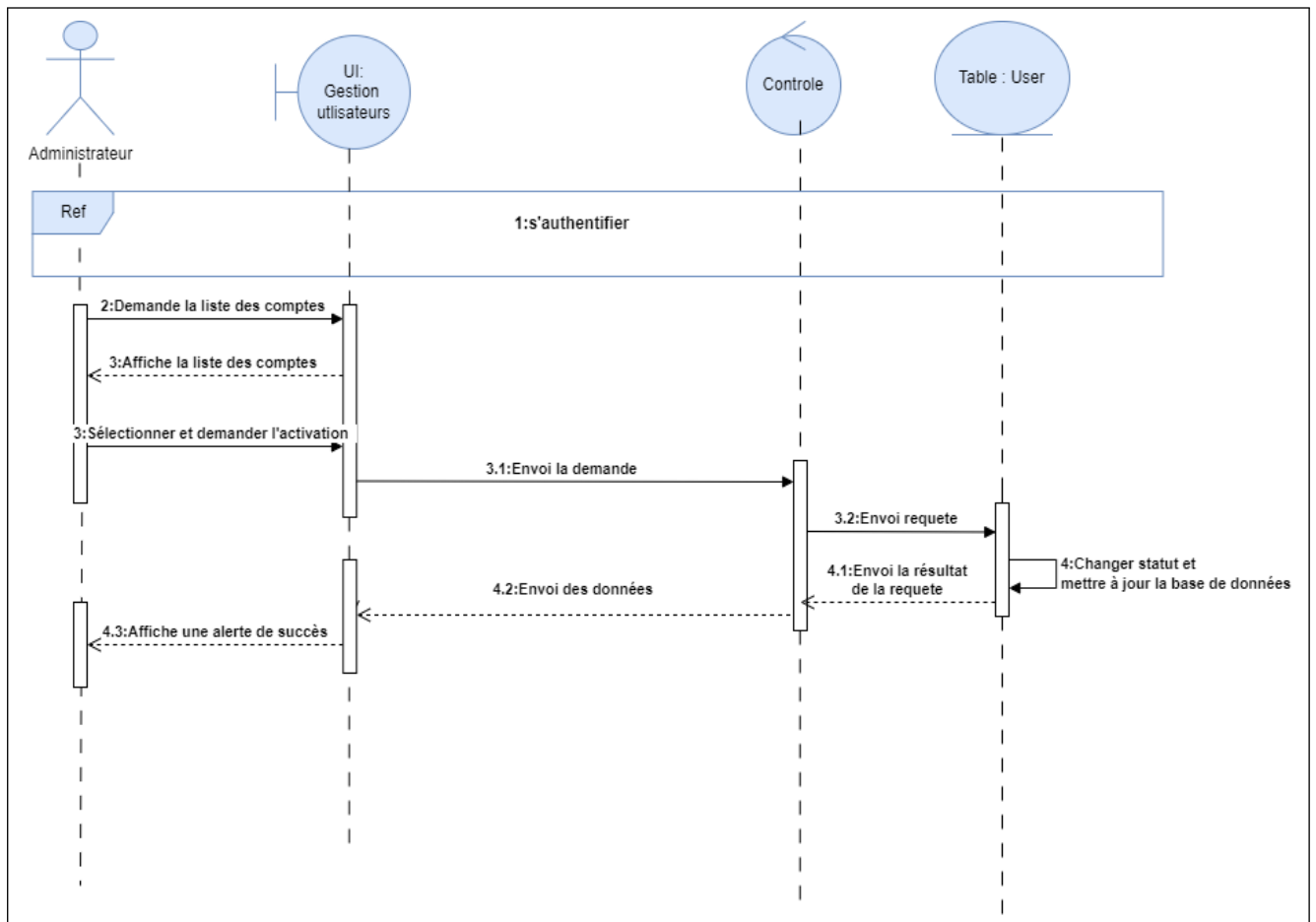


FIGURE 3.7 – Diagramme de séquence lié à la fonctionnalité «Activer compte»

3.4.5 Diagramme de séquence lié à la fonctionnalité « Ajouter compte »

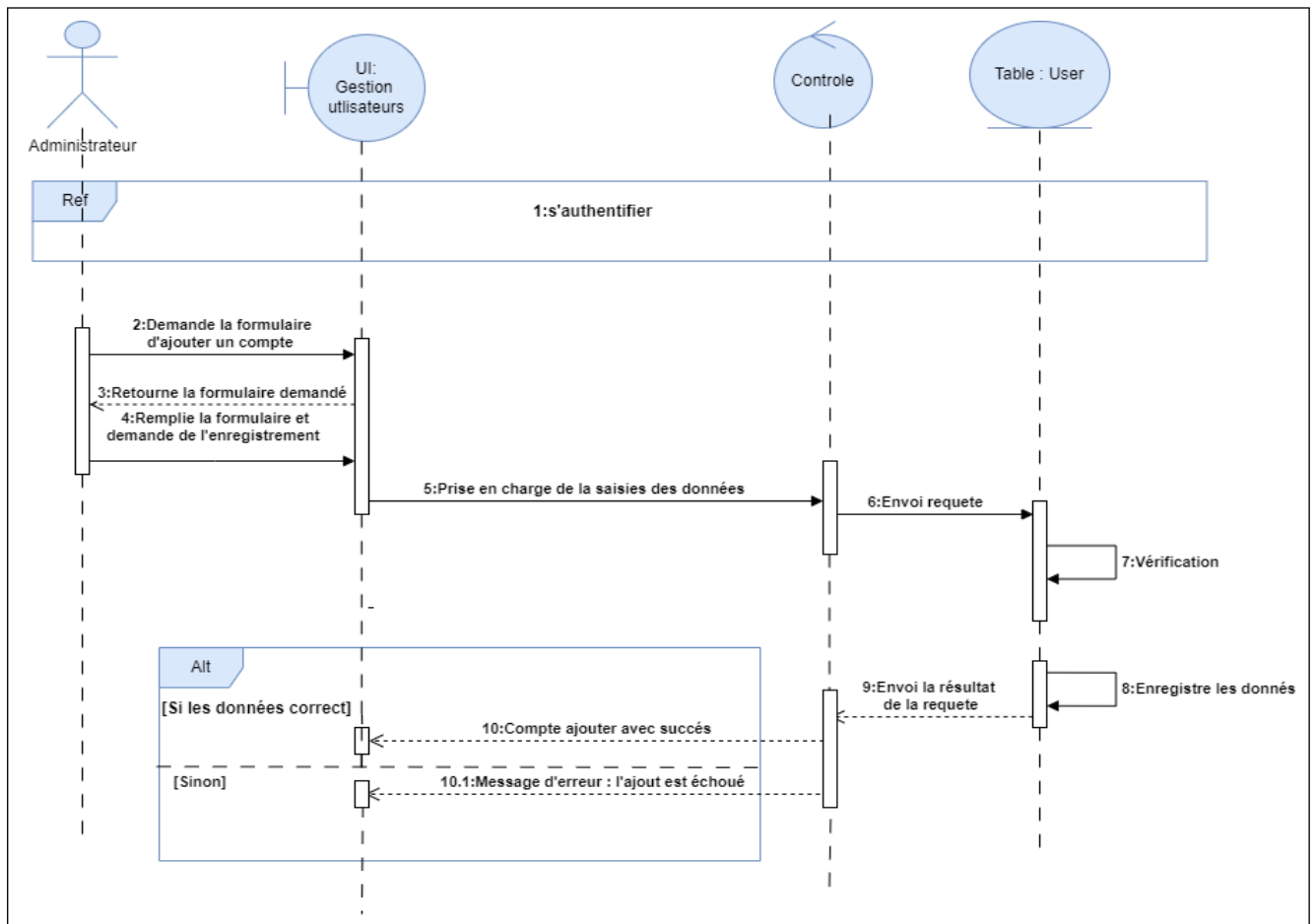


FIGURE 3.8 – Diagramme de séquence lié à la fonctionnalité « Ajout compte »

3.4.6 Diagramme de séquence lié à la fonctionnalité « Modifier compte »

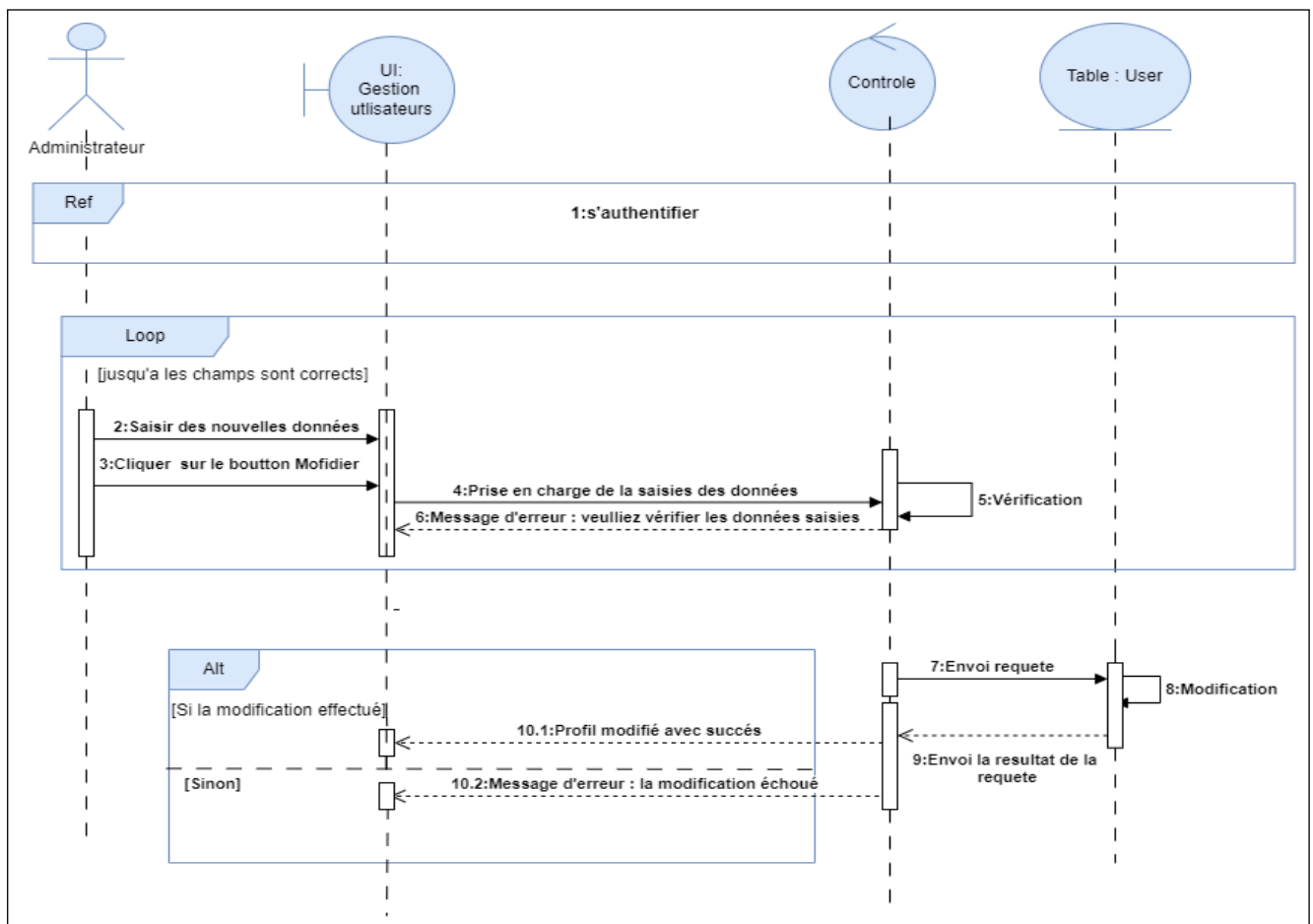


FIGURE 3.9 – Diagramme de séquence lié à la fonctionnalité « Modifier compte »

3.4.7 Diagramme de séquence lié à la fonctionnalité « Supprimer compte »

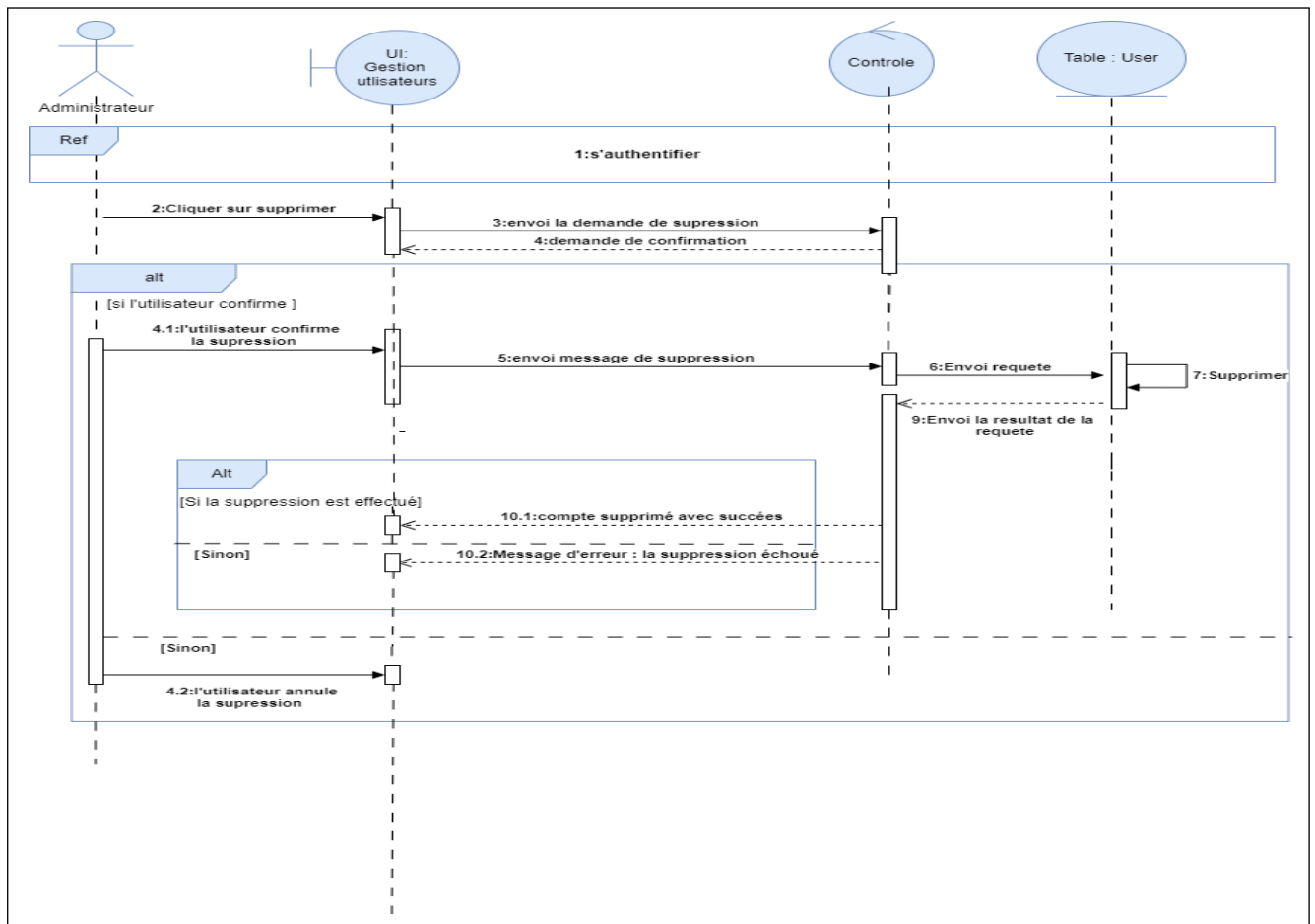


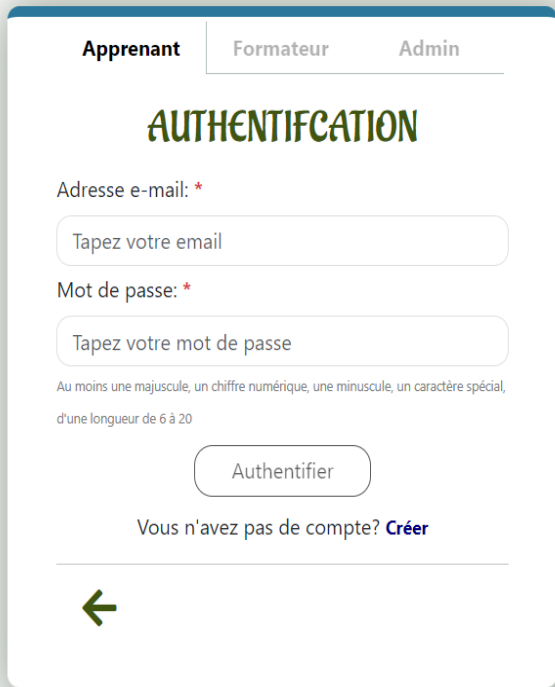
FIGURE 3.10 – Diagramme de séquence lié à la fonctionnalité « Supprimer compte »

3.5 Réalisation

Dans cette section, nous allons présenter quelques interfaces de notre application à travers différents imprimes écrans réalisés qui illustrent le succès des implémentations testées.

3.5.1 Interface d'authentification

L'interface d'authentification, illustrée à la figure 3.7, est le point d'entrée de notre application. L'utilisateur saisit son e-mail et son mot de passe pour accéder à l'application.



The screenshot displays a login form titled 'AUTHENTIFICATION'. At the top, there are three tabs: 'Apprenant' (selected), 'Formateur', and 'Admin'. The form contains two input fields: 'Adresse e-mail: *' with the placeholder 'Tapez votre email', and 'Mot de passe: *' with the placeholder 'Tapez votre mot de passe'. Below the password field, a note specifies password requirements: 'Au moins une majuscule, un chiffre numérique, une minuscule, un caractère spécial, d'une longueur de 6 à 20'. An 'Authentifier' button is positioned below the inputs. At the bottom, a link reads 'Vous n'avez pas de compte? [Créer](#)'. A green left-pointing arrow is located at the bottom left of the form.

FIGURE 3.11 – Interface d'authentification »

3.5.2 Interface d'inscription

L'interface d'inscription illustrée à la figure 3.8, est le point de s'inscrire de notre application. L'utilisateur remplit le formulaire pour accéder à l'application.

The image shows a web form titled "Inscription utilisateur". At the top right, there are two tabs: "Utilisateur" (selected) and "Formateur". The form contains the following fields:

- username *
- Email *
- Nom et prénom *
- téléphone *
- Genre *
- adresse *
- Poste *
- Confirmer le mot de passe *
- Mot de passe *

Below the password field, there is a note: "au moins une majuscule, un chiffre numérique, une minuscule, un caractère spécial, d'une longueur de 6 à 20".

At the bottom left, there is a checkbox: ☐ J'accepte tous les termes et conditions.

At the bottom right, there is a link: Vous avez déjà un compte? [Cliquez ici](#)

At the very bottom, there is a large button labeled "Inscription".

FIGURE 3.12 – Interface d'inscription»

3.6 Conclusion

Dans ce chapitre, nous avons réussi à compléter toutes les tâches planifiées pour le premier sprint. Dans le chapitre suivant et avec la même démarche, nous passons à décrire la réalisation du deuxième sprint.

Chapitre 4 : Réalisation du deuxième sprint

Sommaire

4.1	Introduction	52
4.2	Backlog du Sprint 2	52
4.3	Spécification fonctionnelle	52
4.3.1	Diagramme des cas d'utilisation du deuxième sprint 2	53
4.3.2	Description textuelle et raffinement des principaux cas d'utilisation	53
4.4	Conception	57
4.4.1	Diagrammes de séquences détaillés	58
4.5	Réalisation	62
4.5.1	Interface Gestion cour	63
4.5.2	Interface Ajouter cour	63
4.5.3	Interface Modifier cour	63
4.5.4	Interface supprimer cour	63
4.5.5	Interface Gestion quiz	63
4.5.6	Interface Ajouter quiz	63
4.5.7	Interface Supprimer quiz	63
4.6	Conclusion	63

4.1 Introduction

Dans ce chapitre, nous allons décrire la réalisation du deuxième sprint en introduisant les spécifications fonctionnelles, conception et l'implémentation . Parmi les fonctionnalités que nous allons aborder dans ce sprint.

4.2 Backlog du Sprint 2

Le but de ce sprint est de permet au formateur de gérer les cours et gérer les quiz . Le tableau ci dessous représente le backlog du sprint 2

TABLE 4.1 – Product Backlog Sprint 2

ID	User story	Taches	Durée
1	En tant que formateur, je dois gérer les cours	Gérer cour	5 jours
2	En tant que formateur, je dois gérer les quiz.	Gérer quiz	5 jours

4.3 Spécification fonctionnelle

Pour la mise en marche de notre sprint nous avons tout d'abord élaboré le diagramme de cas d'utilisation de notre sprint avec le raffinement de chaque cas d'utilisation .

4.3.1 Diagramme des cas d'utilisation du deuxième sprint 2

La figure ci dessous présente le diagramme de cas d'utilisation du deuxième sprint.

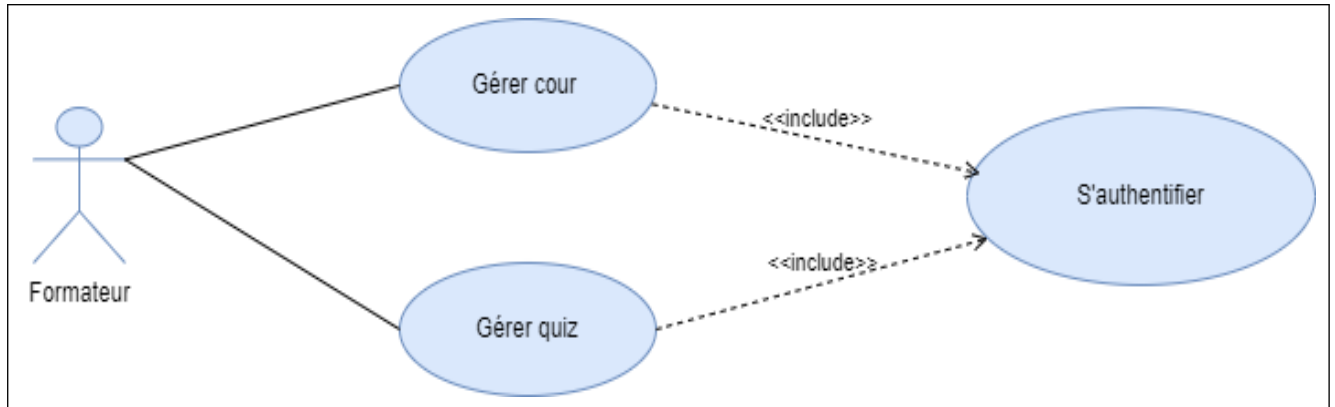


FIGURE 4.1 – Diagramme de cas d'utilisation globale de sprint 2

Comme présenté dans ce diagramme, le formateur aura le privilège de gérer les cours et gérer les quizzes .

4.3.2 Description textuelle et raffinement des principaux cas d'utilisation

Dans cette section, nous allons donner une description textuelle et un raffinement des principaux cas.

4.3.2.1 Raffinement du cas d'utilisation "Gérer cours"

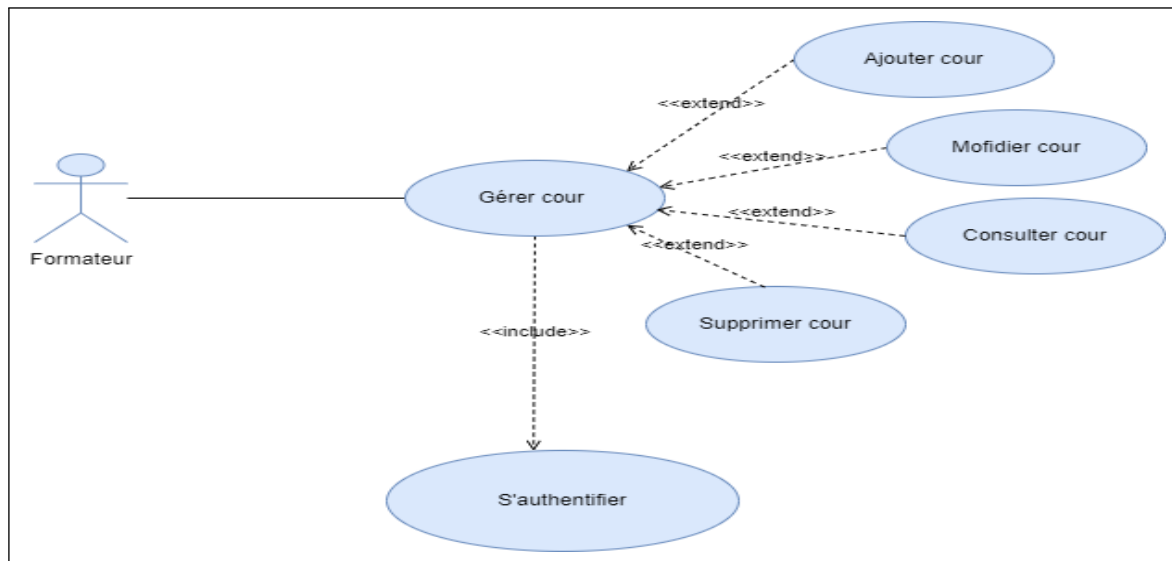


FIGURE 4.2 – Diagramme de cas d'utilisation Gérer cours

4.3.2.2 Description textuelle du cas d'utilisation 'Ajouter cour'

TABLE 4.2 – Description textuelle du cas d'utilisation "Ajouter cour"

Cas d'Utilisation	Ajouter cour
Acteurs	Formateur
Précondition	Formateur doit etre authentifié cours non ajouté.
Postcondition	Cours ajouté.
Scénario nominal	1.Le formateur demande la page d'ajout d'un cours. 2.Le système affiche le formulaire d'ajout d'un cours. 3.Le formateur saisit les données nécessaires du cours et demande l'enregistrement. 4.Le système vérifie les données. 5.Le système ajoute le cours à la base de données. 6.Le système retourne une alerte de succès.
Scénario alternatif	Le système affiche un message d'erreur dans le cas où le formulaire est mal saisi.

4.3.2.3 Description textuelle du cas d'utilisation 'Modifier cour'**TABLE 4.3 – Description textuelle du cas d'utilisation "Modifier cour"**

Cas d'Utilisation	Modifier cour
Acteurs	Formateur
Précondition	Formateur doit être authentifié cours non ajouté.
Postcondition	Cours ajouté.
Scénario nominal	1.Le formateur demande la liste des cours. 2.Le système affiche la liste des cours. 3.Le formateur sélectionne le cours qu'il va modifier. 4.Le système affiche les données qui concernent le cours sélectionné. 5.Le formateur modifie les données et demande l'enregistrement. 6.Le système vérifie les données. 7.Le système met à jour le cours et retourne une alerte de succès.
Scénario alternatif	Le système affiche un message d'erreur dans le cas où le formulaire est mal saisi.

4.3.2.4 Description textuelle du cas d'utilisation " Supprimer cour"**TABLE 4.4 – Description textuelle du cas d'utilisation "Supprimer cour"**

Cas d'Utilisation	Supprimer cour
Acteurs	Formateur
Précondition	Formateur doit être authentifié cours ajouté.
Postcondition	Cours supprimé.
Scénario nominal	1.Le formateur demande la liste des cours 2.Le système affiche la liste des cours 3.Le formateur sélectionne le cours à supprimer et demande la suppression. 4.Le système supprime le cours et affiche une alerte de succès.
Scénario alternatif	Le système affiche un message d'erreur .

4.3.2.5 Raffinement du cas d'utilisation "Gérer Quizz"

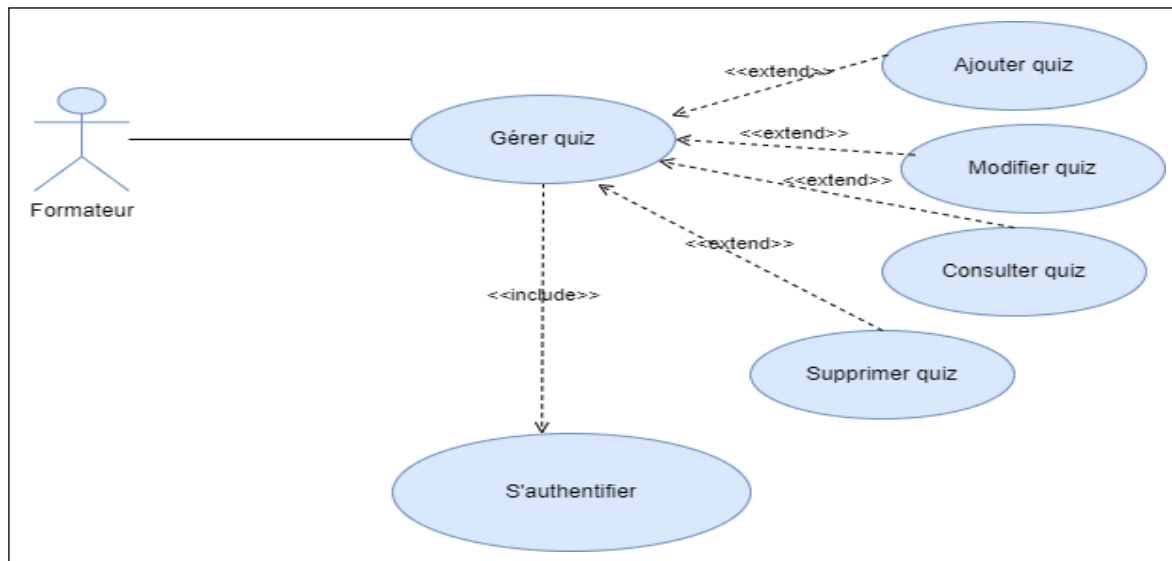


FIGURE 4.3 – Diagramme de cas d'utilisation Gérer quizz

4.3.2.6 Description textuelle du cas d'utilisation "Ajouter quizz"

TABLE 4.5 – Description textuelle du cas d'utilisation "Ajouter quizz"

Cas d'Utilisation	Ajouter quizz
Acteurs	Formateur
Précondition	Formateur doit être authentifié Quiz non ajouté.
Postcondition	Quiz ajouté.
Scénario nominal	1.Le formateur demande l'interface Gestion Quiz 2.Le système affiche l'interface demandé 3.Le formateur remplit la formulaire d'ajout des quiz en ajoutent les questions. 4.Le système met à jours et affiche une alerte de succès.
Scénario alternatif	Le système affiche un message d'erreur .

4.3.2.7 Description textuelle du cas d'utilisation 'Modifier quiz'

TABLE 4.6 – Description textuelle du cas d'utilisation "Modifier quiz"

Cas d'Utilisation	Modifier quiz
Acteurs	Formateur
Précondition	Formateur doit etre authentifié quiz ajouté.
Postcondition	Cours modifié.
Scénario nominal	1.Le formateur demande l'interface Gestion Quiz 2.Le système affiche l'interface demandé 3.Le formateur sélectionne le quiz qu'il va modifier. 4.Le formateur modifie les données des quiz et demande l'enregistrement. 5.Le système vérifie les données. 6.Le système met à jour le cours et retourne une alerte de succès.
Scénario alternatif	Le système affiche un message d'erreur .

4.3.2.8 Description textuelle du cas d'utilisation " Supprimer quiz"

TABLE 4.7 – Description textuelle du cas d'utilisation "Supprimer quiz"

Cas d'Utilisation	Supprimer quiz
Acteurs	Formateur
Précondition	Formateur doit etre authentifié quiz ajouté.
Postcondition	quiz supprimé.
Scénario nominal	1.Le formateur demande la liste des quiz 2.Le système affiche la liste des quiz 3.Le formateur sélectionne le quiz à supprimer et demande la suppression. 4.Le système supprime le quiz et affiche une alerte de succès.
Scénario alternatif	Le système affiche un message d'erreur .

4.4 Conception

Dans cette section , nous présentons les différents diagrammes de séquence détaillés de ce sprint .

4.4.1 Diagrammes de séquences détaillés

4.4.1.1 Diagramme de séquence lié à la fonctionnalité « Ajouter cour »

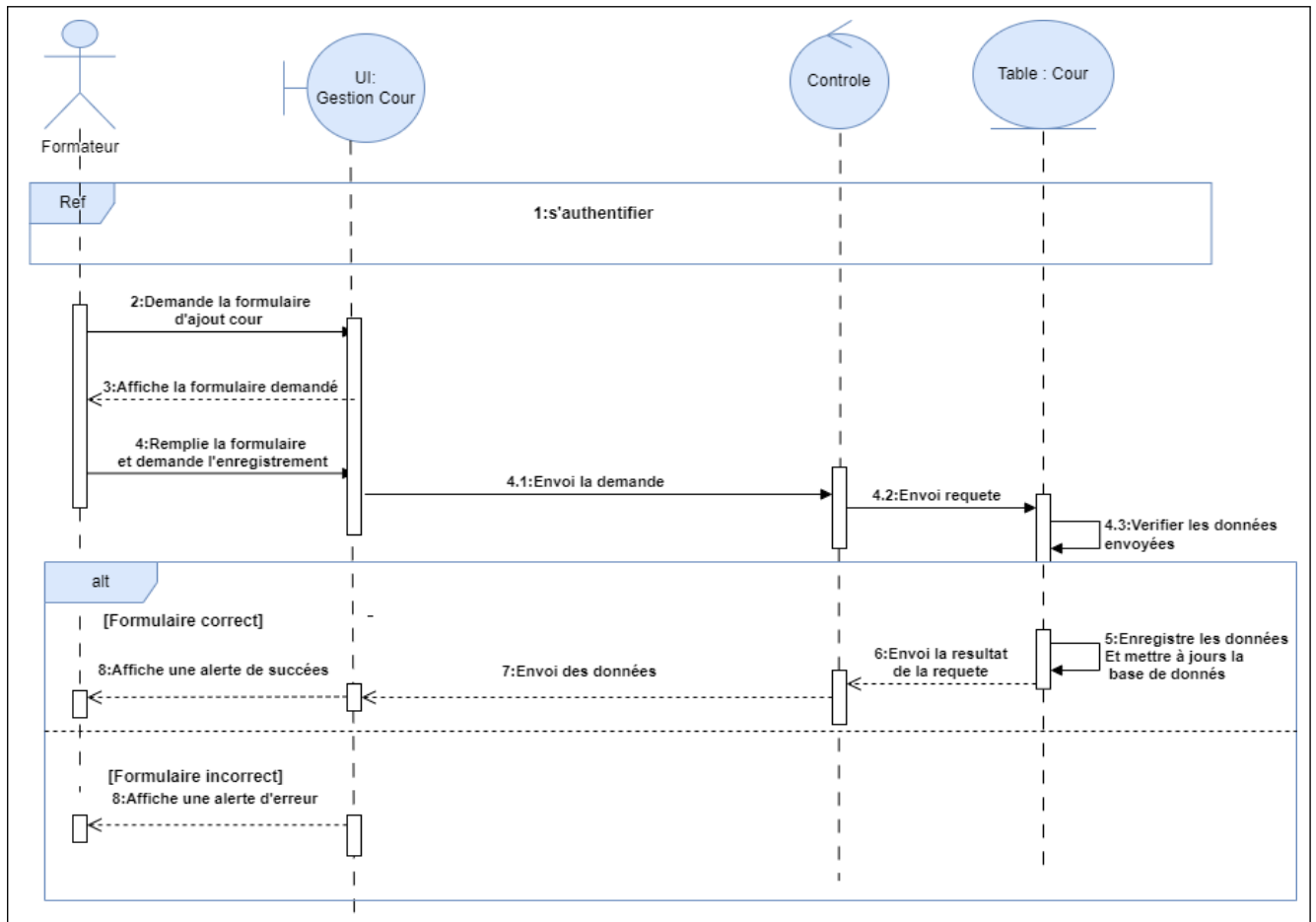


FIGURE 4.4 – Diagramme de séquence lié à la fonctionnalité « Ajouter cour»

4.4.1.2 Diagramme de séquence lié à la fonctionnalité « Modifier cour »

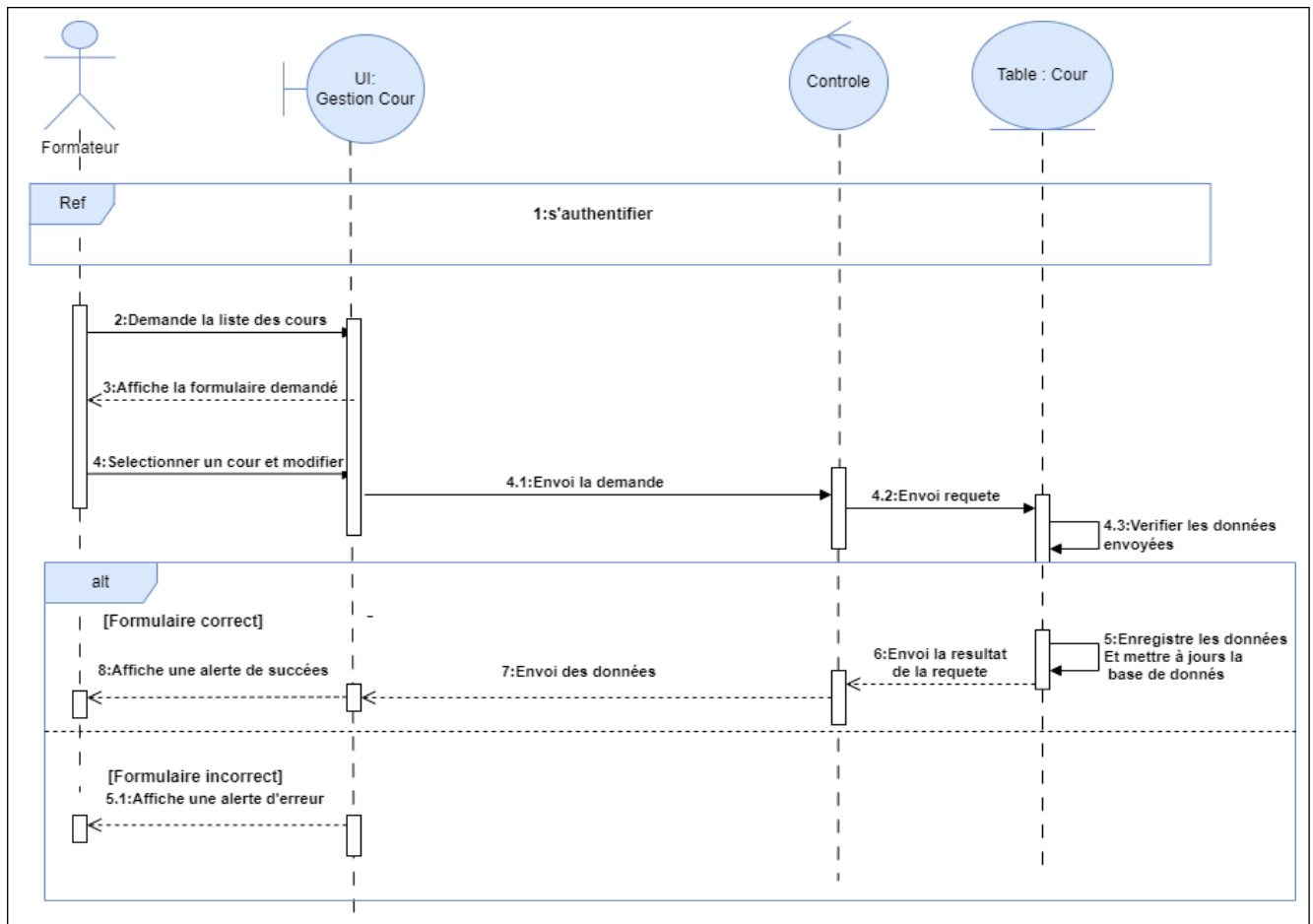


FIGURE 4.5 – Diagramme de séquence lié à la fonctionnalité « Modifier cour»

4.4.1.3 Diagramme de séquence lié à la fonctionnalité « Supprimer cour »

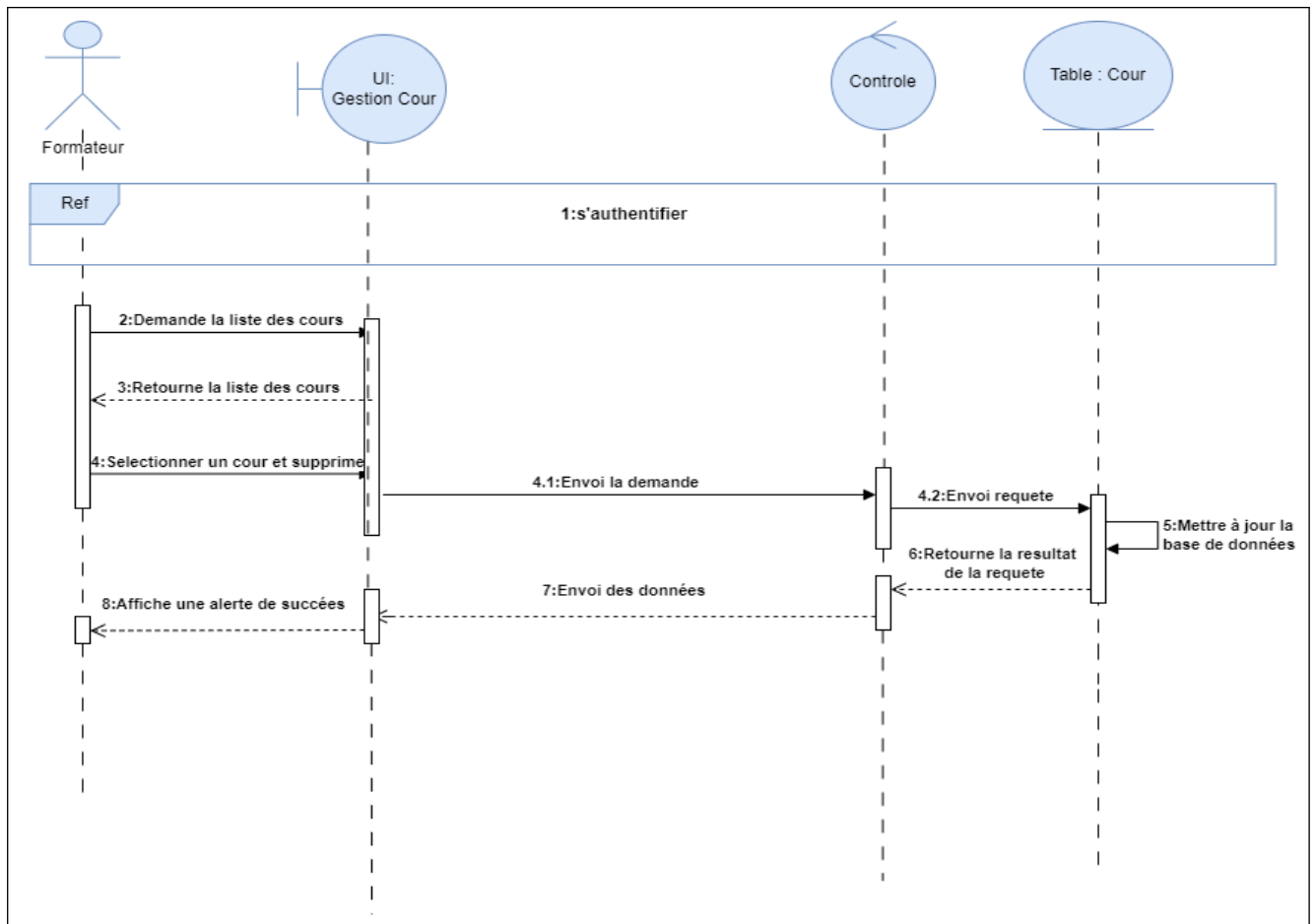


FIGURE 4.6 – Diagramme de séquence lié à la fonctionnalité « Modifier cour»

4.4.1.4 Diagramme de séquence lié à la fonctionnalité « Ajouter quiz »

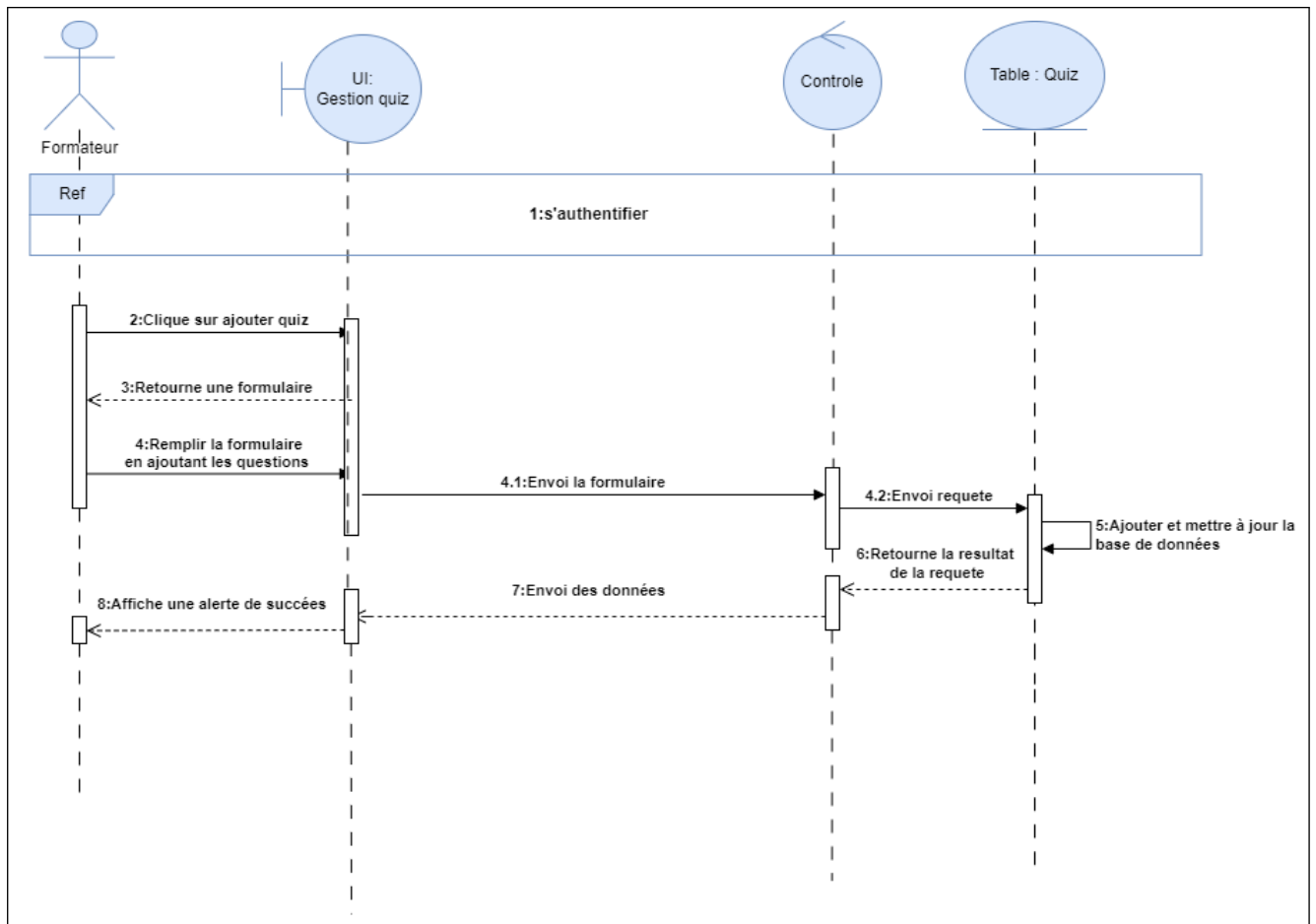


FIGURE 4.7 – Diagramme de séquence lié à la fonctionnalité « Ajouter quiz»

4.4.1.5 Diagramme de séquence lié à la fonctionnalité « Modifier quiz »

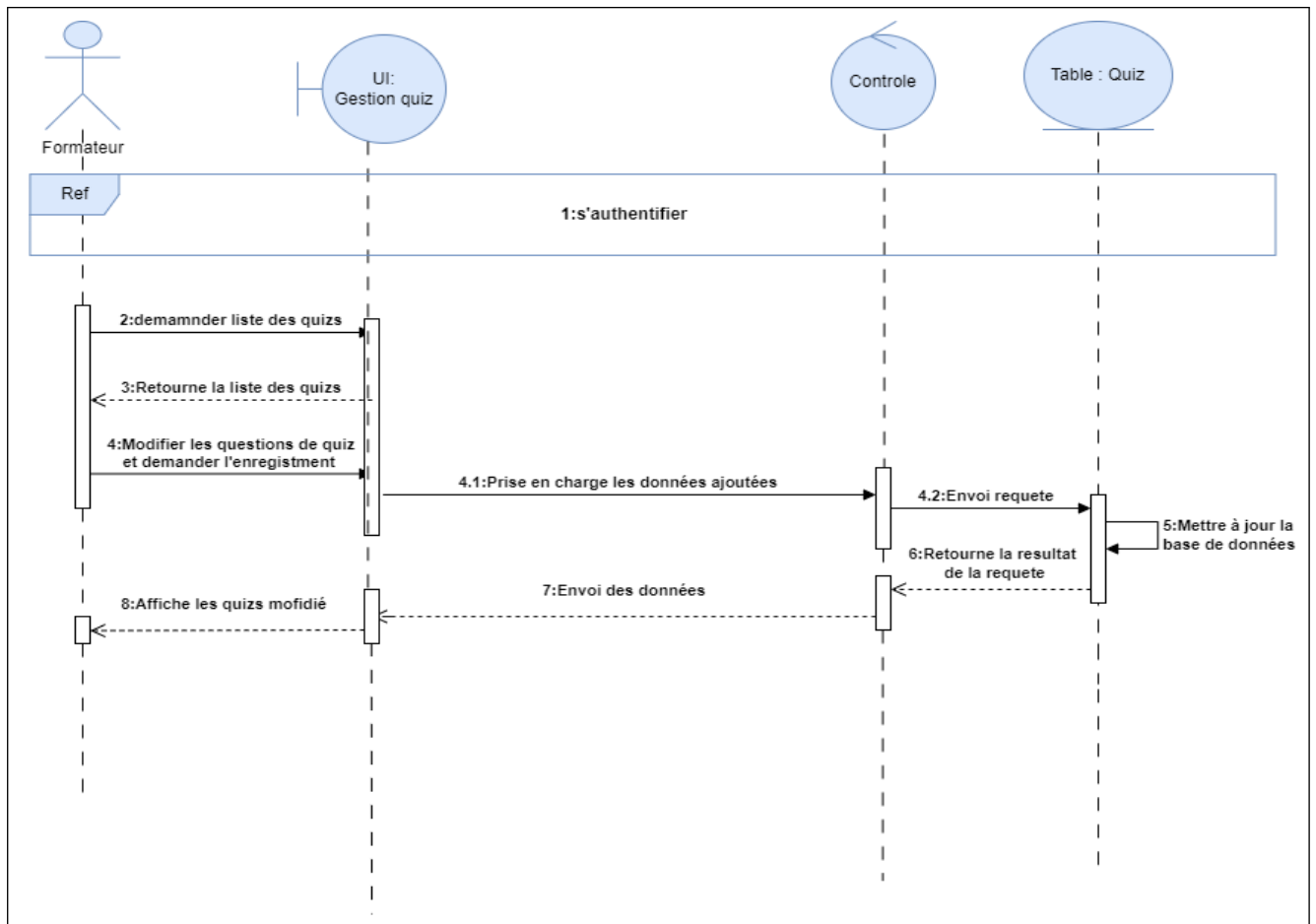


FIGURE 4.8 – Diagramme de séquence lié à la fonctionnalité « Modifier quiz»

4.5 Réalisation

Dans cette section, nous allons présenter quelques interfaces de notre application à travers différents imprimés écrans réalisés qui illustrent le succès des implémentations.

4.5.1 Interface Gestion cour

4.5.2 Interface Ajouter cour

4.5.3 Interface Modifier cour

4.5.4 Interface supprimer cour

4.5.5 Interface Gestion quiz

4.5.6 Interface Ajouter quiz

4.5.7 Interface Supprimer quiz

4.6 Conclusion

Dans ce chapitre, nous avons réussi à compléter toutes les tâches planifiées pour le sprint 2. Avec la même démarche, nous continuerons à traiter les fonctionnalités planifiées pour le Sprint 3.

Chapitre5 : Réalisation du troisième sprint

Chapitre6 : Réalisation

Sommaire

6.1	Introduction	66
6.2	Diagramme de déploiement	66
6.3	Environnement du travail	66
6.3.1	Environnements matériels	67
6.4	Environnements Logiciels	68
6.4.1	IDE	68
6.4.2	68
6.4.3	Les Framework du développement	69
6.4.4	Les langages de la programmation	70
6.4.5	Outil de conception	71
6.4.6	Outils	71
6.4.7	Site web :	71
6.5	Chronogramme de travail	72
6.6	Conslusion	73

6.1 Introduction

Dans ce chapitre, nous présentons l'environnement de travail qui nous permet d'implémenter notre application. Nous donnons d'abord le diagramme de déploiement, l'environnement matériel, et finalement, l'environnement logiciel.

6.2 Diagramme de déploiement

Un diagramme de déploiement appartient à la catégorie des diagrammes de structure car il décrit un aspect du système lui-même. Dans ce cas, un diagramme de déploiement décrit le déploiement physique des informations générées par le logiciel sur les composants matériels.

-Serveur web :

C'est le logiciel qui répond aux requêtes HTTP, c'est-à-dire qui renvoie des données (tels que les pages HTML) en réponse à des requêtes écrites en http (Apache).

-Station client :

C'est le logiciel capable d'envoyer des requêtes HTTP à un serveur Web et d'afficher les résultats. Les navigateurs Web sont les clients Web les plus populaires (Google Chrome).

-Serveur de base de données :

C'est la partie persistante de l'application qui comprend le système de gestion de la base de données (MySQL).

6.3 Environnement du travail

Nous allons présenter dans cette section, l'environnement logiciel et matériel utilisé pour la réalisation du projet.

6.3.1 Environnements matériels

Notre projet est développé sur deux machines « Lenovo » et « Asus » sous les caractéristiques suivantes :



FIGURE 6.1 – Pc Asus

TABLE 6.1 – Description d’environnement matériel PC

Système d’exploitation	Windows 11
Processeur	11th Gen Intel Core i5
Disque Dur	1 TO
Mémoire RAM	16Go
Ecran	15.6" Full HD LED

TABLE 6.2 – Description d’environnement matériel PC LENOVO

Système d’exploitation	Windows 10
Processeur	10th Gen Intel Core i3
Disque Dur	128 SSD
Mémoire RAM	8Go
Ecran	15.6"

6.4 Environnements Logiciels

6.4.1 IDE

Visual Studio Code : Visual Studio Code est un éditeur de code multiplateforme publié par Microsoft. Cet outil convient aux développeurs et prend en charge plusieurs langages de programmation, tels que HTML, C++, PHP, JavaScript, CSS, etc.

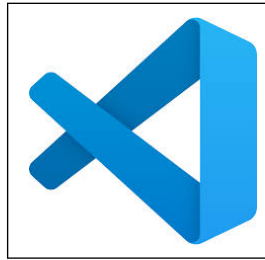


FIGURE 6.2 – Logo Visual Studio Code

IntelliJ IDEA : Également appelé « IntelliJ », « IDEA » ou « IDJ », est un éditeur intégré à certains des outils de développement gratuits les plus populaires (tels que Git, CVS, Subversion, Ant et Maven, JUnit).

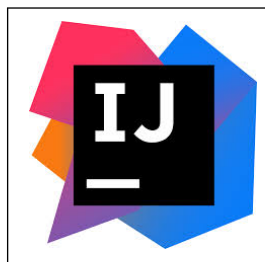


FIGURE 6.3 – Logo IntelliJ IDEA

6.4.2

Langage de gestion de base de données **MySQL** : Mysql est un serveur de bases de données relationnelles



FIGURE 6.4 – Logo MySQL

6.4.3 Les Framework du développement

Angular : Angular est une plateforme de développement qui vous permet de créer des applications Web dynamiques et immersives.



FIGURE 6.5 – Logo Angular

Spring boot : Spring Boot est un framework. qui fournit les dépendances nécessaires en les configurant automatiquement, vous pouvez démarrer rapidement le développement d'une application ou d'un service.



FIGURE 6.6 – Logo Spring Boot

6.4.4 Les langages de la programmation

Java 8 : C'est un langage de programmation qui fournit de nouvelles fonctionnalités, des performances améliorées et des corrections de bugs pour améliorer l'efficacité du développement et de l'exécution des programmes Java.



FIGURE 6.7 – Logo Java 8

Type Script : Le TypeScript est un langage de programmation développé par Microsoft en 2012. C'est un langage open source, développé comme un sur-ensemble de Javascript. Ce qu'il faut comprendre par là, c'est que tout code valide en Javascript l'est également en TypeScript.

HTML : Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web.

CSS : Le CSS pour Cascading Style Sheets, est un langage informatique utilisé sur Internet pour la mise en forme des fichiers et des pages HTML. On le traduit en français par feuilles de style en cascade.

JavaScript : JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web.



FIGURE 6.8 – Logo Type Script

6.4.5 Outil de conception

Draw.io : diagrams.net (draw.io) est une solution accessible en ligne et gratuite qui permet de créer des diagrammes et des organigrammes. On a accès à toutes les fonctionnalités d'un outil de création de diagrammes professionnel, dans un espace de travail en anglais.

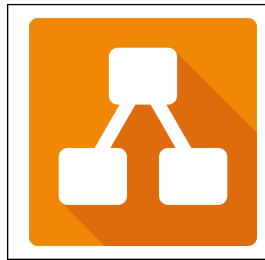


FIGURE 6.9 – Logo Draw.io

6.4.6 Outils

Postman :Postman vous permet de construire et d'exécuter des requêtes HTTP, de les stocker dans l'historique afin qu'elles puissent être rejouées, mais le plus important est de les organiser en collections.



FIGURE 6.10 – Logo Postman

6.4.7 Site web :

GitHub : site web, un service qui permet d'héberger des projets de développement en versionnage à l'aide de Git. Il est très complet pour des projets.



FIGURE 6.11 – Logo GitHub

6.5 Chronogramme de travail

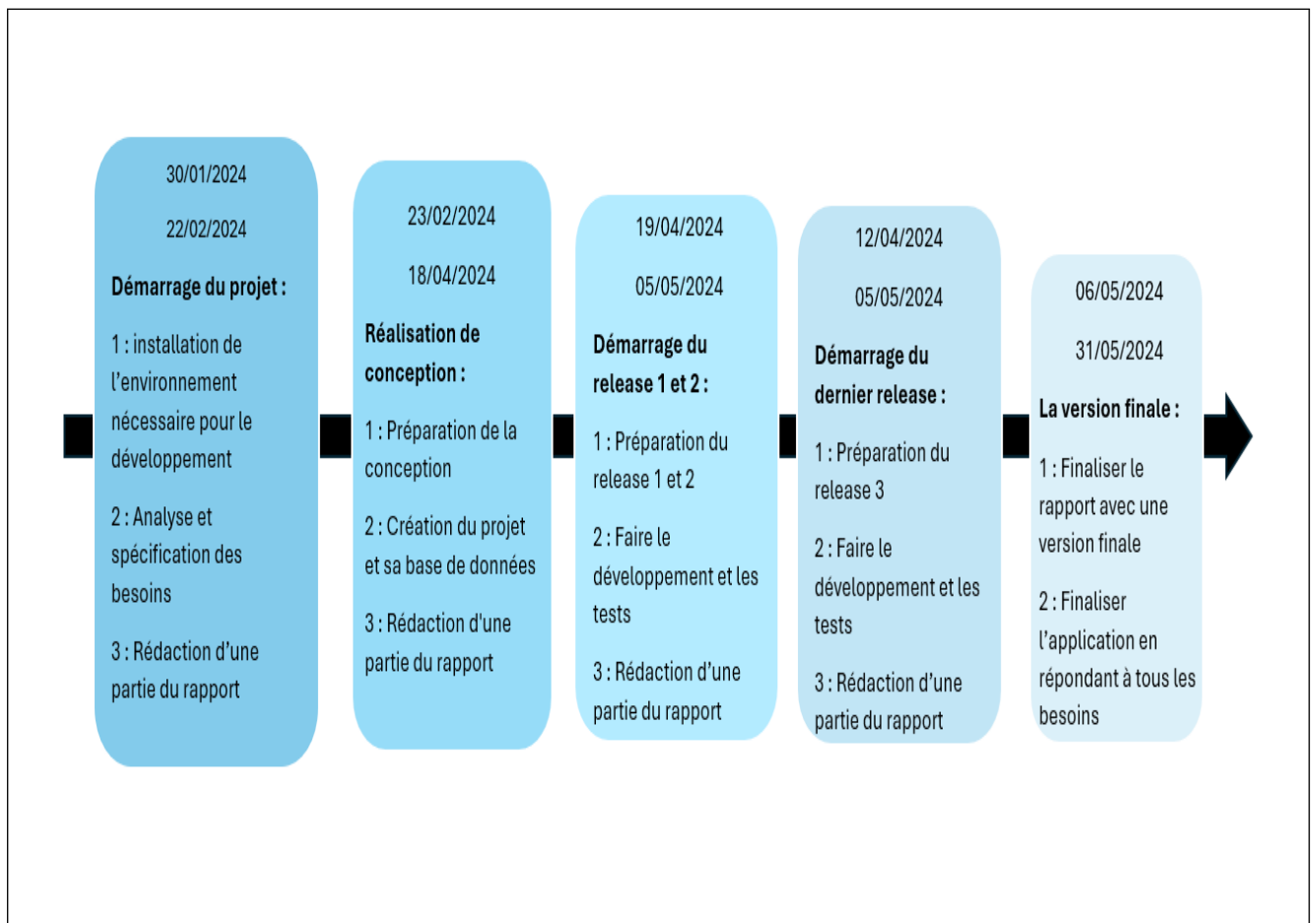


FIGURE 6.12 – Chronogramme de temps

6.6 Conslusion

Ce dernier chapitre nous a permis de présenter l'environnement de travail et les technologies utilisées pour réaliser notre application ainsi que le chronogramme de travail.



CONCLUSION GÉNÉRALE

Dans ce rapport, nous avons décrit les différentes étapes de mise en place d'une application d'application d'éducation en ligne . Pour le développement de ce projet, la méthodologie Agile Scrum et la technologie UML ont été utilisées, ce qui permet la gestion de projet sur un système spécifique, aide à raccourcir le temps et à effectuer les exigences d'analyse des tâches de manière fragmentée et détaillée, ce qui aide à répondre aux exigences de l'utilisateur. Outre les scénarios sont également détaillées pour expliquer toutes les tâches effectuées.

Ce projet nous a donné l'opportunité de voir la vie professionnelle dans un cadre réel avec une expérience de départ importante qui nous a appris à jouer le rôle de la joie au petit matin, comment construire une bonne relation pour assurer le travail d'équipe, comment résoudre les problèmes de manière autonome, comment être méticuleux dans le travail, comment prêter attention aux instructions des supérieurs, comment terminer le travail de manière organisée le plus rapidement possible et compléter le tâches qui nous sont confiées dans les meilleures conditions.

Lors de la mise en place de notre projet, nous avons été soumises à certaines contraintes, notamment des contraintes de temps, qui constituaient un frein relatif avant d'ajouter quelques autres fonctionnalités. Cependant, c'est l'occasion de mettre en valeur et de déployer nos connaissances en informatique sur un plan pratique.

Le projet peut être amélioré, cela peut se faire en ajoutant quelques modules ou interfaces pour mieux s'adapter aux besoins des utilisateurs, afin qu'il soit toujours fiable.



BIBLIOGRAPHIE

[1]

Resumé :

Ce travail a été réalisé au Centre Nationale de L'informatique dans le cadre du stage de fin d'études en licence appliquée en technologie de l'informatique à l'Institut Supérieur des Études Technologiques de Zaghuan, visant à développer une plateforme e-learning. Le projet comprenait la conception et le développement d'une application Web d'apprentissage en ligne qui a été géré à l'aide de Scrum.

Mots clés : Scrum, MVC, Spring boot, Angular.

Summary :

This work was carried out at the National Informatics Center as part of the end-of study internship in applied license in computer technology at the Higher Institute of Technological Studies of Zaghuan, aimed at developing an e-learning platform. The project included the design and development of an e-learning web application that was managed using Scrum.

Key words : Scrum, MVC, Spring boot, Angular.

ملخص

هذا العمل تم تنفيذه في المركز الوطني للإعلامية، كجزء من تدريب نهاية الدراسة للحصول على الاجازة التطبيقية في تكنولوجيا الاعلامية في المعهد العالي للدراسات التكنولوجية بزغوان، بهدف تطوير منصة التعلم الإلكتروني. يتضمن مشروع تصميم و تطوير تطبيق ويب للتعليم الإلكتروني تم إدارته بإستخدام
