



Université De Tunis El Manar  
Faculté des Sciences Economiques et de Gestion  
de  
-Tunis -



## Rapport de Projet de Fin d'Études

En vue de l'obtention du diplôme

Licence Fondamentale en informatique de gestion

---

# Mise en place d'un système partagé et sécurisé dédié à l'enseignement à distance

---

Réalisé au sein de l'Agence digitale N3RD

*Réalisé par :*

Khalfi Khawla, Kouki Hamza

*Sous la direction de :*

M. Mounir NEFZI (SQLI)

*Soutenu le 01 juillet 2020, Devant le jury :*

Pr. Meriem MANDAR : ENSA Khouribga - Présidente

Pr. Mohamed AMNAI : ENSA Khouribga - Encadrant

Pr. Mohammed NASRI : ENSA Khouribga - Encadrant

Année universitaire  
2019/2020

# Dédicace

“

*Je dédie ce travail :*

*À ma chère mère et à mon cher père qui n'ont jamais cessé de me supporter, me soutenir et m'encourager durant mes années d'études et de moments de faiblesse et de maladie.*

*Qu'ils trouvent ici le témoignage de ma profonde gratitude et reconnaissance.*

*À mon frère et ma sœur qui me donnent de l'amour et de la vivacité.*

*À mon binôme . À tous ceux qui m'ont aidé - de près ou de loin - ceux qui ont partagé avec moi les moments d'émotion lors de la réalisation de ce travail et qui m'ont chaleureusement supporté et encouragé tout au long de mon parcours.*

*Et Pour finir, à tous mes amis qui m'ont toujours encouragé.*

*Merci !*

”

**- Khawla**

# Dédicace

“

*Je dédie ce travail :*

*À ma chère mère, et à mon cher père ,Qui n'ont jamais  
cessé, de formuler des prières à mon égard, de me soutenir  
et de m'épauler pour que je puisse atteindre mes objectifs.*

*À A mes frèresr Pour ses soutiens moral et leurs conseils  
précieux tout au long de mes études.*

*Àma chère binôme ,Khalfi khawla, Pour leurs indéfectibles  
soutiens et leurs patiences infinies.*

*Et tous ceux que j'aime et ceux qui m'aiment.*

*Merci !*

”

**- Hamza**

# Remerciements

Nous adressons nos remerciements les plus sincères à tous ceux qui de près ou de loin ont participé à la réalisation de ce travail.

Nous tenons à exprimer notre profonde gratitude, tout particulièrement, à notre encadrant académique Monsieur **Mounir Nefzi** nous a fait profiter de ses larges connaissances et ses précieux conseils tout au long de ce travail. Il a toujours été à notre écoute et a su nous apporter un soutien sans faille.

Nous souhaitons ensuite adresser nos remerciements au corps professoral et administratif de la Faculté de Sciences Economiques et de Gestion de Tunis, pour la qualité de l'enseignement offert et le soutien de l'équipe administrative.

Nos Vifs remerciements sont adressés également, à notre maître de stage Monsieur **Ouesleti Nizar** et le directeur (à vérifier) Monsieur **Bchini Tarek** au sein de la Société d'Ingénierie et de Développement Web «N3RD» pour leurs accueils, le temps passé ensemble et le partage de leurs expertises . Nous saisissons également cette occasion pour adresser notre profond respect à toute l'équipe de travail qui ont fourni le cadre nécessaire pour la réalisation de notre projet. .

Pour finir, Nous voudrions adressons nos sincères remerciements aux membres du jury pour avoir bien voulu examiner et juger ce travail.

# Table des matières

<b>Introduction</b>	<b>12</b>
<b>1 Contexte général du projet</b>	<b>14</b>
1.1 Présentation de l'organisme d'accueil . . . . .	15
1.1.1 Cadre général de travail . . . . .	15
1.1.2 Domaines d'activité de l'entreprise . . . . .	15
1.1.3 Mission de l'entreprise . . . . .	15
1.2 Étude de l'existant . . . . .	16
1.2.1 Cadre du projet . . . . .	16
1.2.2 Problématique . . . . .	16
1.2.3 Solution proposée . . . . .	17
1.3 Planification et conduite du projet . . . . .	17
1.3.1 Démarche adoptée . . . . .	17
1.3.1.1 présentation de la methode Scrum . . . . .	18
1.3.1.2 Processus de développement . . . . .	18
1.3.1.3 l'equipe Scrum . . . . .	19
1.3.1.4 les événements Scrum . . . . .	20
1.3.2 Planning du projet . . . . .	20
1.3.3 La modélisation objet . . . . .	23
1.3.3.1 Définition de UML . . . . .	23
1.3.3.2 Processus de modélisation . . . . .	23
1.4 Conclusion . . . . .	23
<b>2 Analyse et spécification des besoins</b>	<b>24</b>

2.1	Spécifications des besoins . . . . .	25
2.1.1	Identification des acteurs . . . . .	25
2.1.2	Les besoins fonctionnels . . . . .	26
2.1.3	Les besoins non fonctionnels . . . . .	27
2.2	Backlog produit . . . . .	28
2.2.1	Planification des Sprints . . . . .	29
2.2.2	Backlog des Sprints . . . . .	29
2.3	Diagrammes des cas d'utilisation global . . . . .	30
2.4	Diagrammes de classes global . . . . .	31
2.5	Conclusion . . . . .	36
<b>3</b>	<b>La réalisation</b>	<b>37</b>
3.1	Sprint 1 : Utilisateur . . . . .	38
3.1.1	item 1 : Authentification . . . . .	40
3.1.1.1	Diagramme de cas d'utilisation . . . . .	40
3.1.1.1.a	Description détaillée des cas d'utilisations . . . . .	40
3.1.1.2	Diagrammes de séquence système . . . . .	41
3.1.2	item 2 : Gérer le profil . . . . .	42
3.1.2.1	Diagramme de cas d'utilisation . . . . .	42
3.1.3	item 3 : Modification de rôle . . . . .	44
3.1.3.1	Diagrammes de séquence système . . . . .	44
3.2	Sprint 2 : Administration . . . . .	45
3.2.1	item 1 : Gérer les utilisateurs . . . . .	47
3.2.1.1	Diagramme de cas d'utilisation détaillé «administrer du site de téléformation» . . . . .	47
3.2.1.1.a	Description détaillée des cas d'utilisations . . . . .	47
3.2.1.1.b	Description détaillée des cas d'utilisations . . . . .	48
3.2.1.2	Diagrammes de séquence du cas d' utilisation "Modifier un utilisateur" . . . . .	49
3.2.1.2.a	Description détaillée des cas d'utilisations . . . . .	49
3.2.1.3	Diagramme de séquence du cas d' utilisation "Supprimer un utilisateur" . . . . .	50

3.2.1.3.a	Description détaillée des cas d'utilisations . . . . .	51
3.2.1.4	Diagrammes de séquence du cas d' utilisation "Ajouter un utilisateur" . . . . .	51
3.2.1.4.a	Description détaillée des cas d'utilisations . . . . .	52
3.2.1.5	Diagramme de séquence « d' ajout d'un professeur » . . . .	53
3.2.1.5.a	Description détaillée des cas d'utilisations . . . . .	54
3.2.1.6	Diagramme d'activités « Gérer les utilisateurs » . . . . .	54
3.2.1.6.a	Description détaillée des cas d'utilisations . . . . .	55
3.2.2	item 2 : Inscription et réinscription d'un étudiant . . . . .	56
3.2.2.1	Diagramme de séquence « d' inscription d' un étudiant »	56
3.2.2.1.a	Description : Diagramme de séquence « d' inscrip- tion d' un étudiant » . . . . .	56
3.2.2.2	Diagramme d'activités « d'inscription d'un étudiant» . .	57
3.2.2.2.a	Description du processus de diagramme d'activités «Inscription d'un élève» : . . . . .	57
3.2.2.3	Diagramme de collaboration «réinscription d' un étudiant»	58
3.2.2.3.a	Description : Diagramme de collaboration «réins- cription d' un étudiant» . . . . .	59
3.2.2.4	Diagramme d' activités « d' affectation des notes étudiant »	59
3.2.2.4.a	Description du processus de diagramme d'activités «Affecter les notes d'élèves» : . . . . .	60
3.3	Sprint 3 : Enseignants . . . . .	61
3.3.1	item 1 : Gérer les cours . . . . .	63
3.3.1.1	Diagramme de cas d'utilisation détaillé «administrer du site de téléformation» . . . . .	63
3.3.1.1.a	Description détaillée des cas d'utilisations . . . . .	63
3.3.1.2	Diagramme des cas d' utilisation Gérer les cours . . . . .	63
3.3.1.2.a	Description détaillée des cas d'utilisations . . . . .	64
3.3.1.3	diagramme de séquences . . . . .	65
3.3.1.3.a	Description :Diagramme de séquence "Gerer les cours"	65
3.3.1.4	diagramme de séquences . . . . .	65
3.3.1.4.a	Description détaillée des cas d'utilisations . . . . .	66

3.3.1.5	diagramme de séquences . . . . .	66
3.3.1.5.a	Description détaillée des cas d'utilisations . . . . .	67
3.3.2	item 2 : Gérer les tests . . . . .	68
3.3.2.1	diagramme de séquences . . . . .	68
3.3.2.1.a	Description détaillée des cas d'utilisations . . . . .	68
3.3.2.2	diagramme de séquences . . . . .	69
3.3.2.2.a	Description détaillée des cas d'utilisations . . . . .	69
3.4	Sprint 4 : Étudiant . . . . .	70
3.4.1	item 1 : Consulter les ressources et les liens d'apprentissage . . . . .	72
3.4.1.1	Diagramme de cas d'utilisation détaillé «administrer du site de téléformation» . . . . .	72
3.4.1.1.a	Description détaillée des cas d'utilisations . . . . .	72
3.4.1.2	diagramme de séquences . . . . .	72
3.4.1.2.a	Description détaillée des cas d'utilisations . . . . .	73
3.4.2	item 2 : Consulter agenda des livrables . . . . .	74
3.4.2.1	diagramme de séquences . . . . .	74
3.4.2.1.a	Description : Diagramme de séquences consulter agenda des livrables . . . . .	74
3.4.3	item 3 : Passer les examens . . . . .	75
3.4.3.1	Diagramme de cas d'utilisation . . . . .	75
3.4.3.1.a	Description :Diagramme de séquence " Passer les examens" . . . . .	75
3.4.3.1.b	Description détaillée des cas d'utilisations . . . . .	76
3.5	Conclusion . . . . .	77
<b>4</b>	<b>Mise en oeuvre de la solution</b>	<b>78</b>
4.1	Introduction : . . . . .	79
4.2	Environnement de travail : Outils et technologies utilisées . . . . .	79
4.2.1	Environnement matériel . . . . .	79
4.2.2	Environnement logiciel . . . . .	79
4.2.2.1	Outils de développement et modélisation : . . . . .	79
4.2.2.2	Langages de programmation : . . . . .	81



<b>Table des matières</b>	<b>8</b>
4.2.3 Outils de développement et de collaboration : . . . . .	82
4.2.3.1 Le système de contrôle des versions GIT : . . . . .	82
4.3 Conclusion . . . . .	83
<b>Webographie</b>	<b>84</b>

# Table des figures

1.1	Logo de la société N3RD. . . . .	15
1.2	Cycle de vie de la méthodologie scrum. . . . .	18
1.3	Diagramme de gantt du projet. . . . .	22
2.1	Héirarchie des profils humaines. . . . .	25
2.2	Les cas d'utilisation global. . . . .	30
2.3	Diagrammes de classes global. . . . .	32
3.1	Décomposition sprint 1 en Items. . . . .	38
3.2	Diagramme de Cas d' utilisation S' authentifier . . . . .	40
3.3	Diagramme de séquence système d' authentification . . . . .	41
3.4	Diagramme de Cas d' utilisation Gérer le profil . . . . .	42
3.5	Diagramme de séquence système de modification de rôle utilisateur . . . . .	44
3.6	Décomposition sprint 2 en Items. . . . .	45
3.7	Diagramme de cas d' utilisation détaillé «administrer» . . . . .	47
3.8	Diagrammes de séquence du cas d' utilisation "Modifier un utilisateur" . . . . .	49
3.9	Diagramme de séquence du cas d' utilisation "Supprimer un utilisateur" . . . . .	50
3.10	Diagrammes de séquence du cas d' utilisation "Ajouter un utilisateur" . . . . .	52
3.11	Diagramme de séquence « d' ajout d'un professeur » . . . . .	53
3.12	Diagramme d' activités « Gérer les utilisateurs » . . . . .	55
3.13	Diagramme de séquence « d' inscription d' un étudiant » . . . . .	56
3.14	Diagramme d'activités « d'inscription d'un étudiant» . . . . .	57
3.15	Diagramme de collaboration «réinscription d' un étudiant» . . . . .	59
3.16	Diagramme d' activités « d' affectation des notes étudiant » . . . . .	60

---

3.17	Décomposition sprint 3 en Items. . . . .	61
3.18	Diagramme de cas d' utilisation détaillé «Professeur» . . . . .	63
3.19	Diagramme des cas d' utilisation « Gérer les cours » . . . . .	64
3.20	Diagramme de séquence "Gerer les cours" . . . . .	65
3.21	Diagramme d' activités « Gérer les cours » . . . . .	66
3.22	Diagramme de séquence « supprimer une matière » . . . . .	67
3.23	Diagramme de Cas d' utilisation « Gérer les tests / les examens » . . . . .	68
3.24	Diagramme d' activités « Gérer les tests / les examens » . . . . .	69
3.25	Décomposition sprint 4 en Items. . . . .	70
3.26	Diagramme de cas d' utilisation détaillé «Étudiant» . . . . .	72
3.27	Diagramme de séquence : consulter les ressources et les liens d' apprentissage .	73
3.28	Diagramme de séquences consulter agenda des livrables . . . . .	74
3.29	Diagramme de séquence " Passer les examens" . . . . .	75
3.30	Diagramme d' activité : d' un apprenant sur l' espace . . . . .	76
4.1	Logo Visual code studio. . . . .	80
4.2	Logo Visual code studio. . . . .	80
4.3	Logo Visual code studio. . . . .	81
4.4	Logo Visual code studio. . . . .	81
4.5	Logo Visual code studio. . . . .	82

# Liste des sigles et acronymes

<b>DAM</b>	<i>Digital Asset Management</i>
<b>ERP</b>	<i>Enterprise Resource Planning</i>
<b>EAI</b>	<i>Enterprise Application Integration</i>
<b>HAC</b>	<i>Hybris Administration Console</i>
<b>MEP</b>	<i>Mise En Production</i>
<b>OAT</b>	<i>Operational Acceptance Testing</i>
<b>OOTB</b>	<i>Out Of The Box</i>
<b>PIM</b>	<i>Product Information Management</i>
<b>PCM</b>	<i>Product Content Management</i>
<b>TFS</b>	<i>Team Foundation Server</i>
<b>UAT</b>	<i>User Acceptance Testing</i>
<b>WFJ</b>	<i>Watches and Fine Jewelry</i>

# Introduction générale

L'enseignement est un mode d'éducation permettant de développer les connaissances d'un élève par le biais de la communication verbale et écrite. Il est centré sur le cours magistral.

Les systèmes traditionnels d'enseignement imposent à tous les apprenants une unité de lieu, une unité de temps, une unité d'action, une unité de rythme ce qui implique une rigidité des mécanismes et une difficulté d'adéquation avec la réalité quotidienne. La tendance à l'amélioration du système sur le plan pédagogique par le recours aux moyens audiovisuels classiques (projections de diapositives, de transparents, séquences vidéo) n'a pas résolu le problème. Il existe en effet une solution de rechange à l'enseignement traditionnel. Cette forme d'enseignement relativement jeune, c'est la formation à distance qui permet d'acquérir des connaissances et de développer des habiletés sans avoir à fréquenter un établissement d'enseignement et sans la présence physique d'une personne qui enseigne. Dans ce cadre s'intègre notre projet de fin d'étude qui est effectué au sein de la société N3RD. Notre objectif est de concevoir et mettre en place un système partagé et sécurisé dédié à l'enseignement à distance.

Ce rapport s'articulera donc, autour de quatre chapitres comme suit :

Le premier chapitre permet de placer notre projet dans son contexte générale. Il comportera une description de l'organisme d'accueil, exposera l'étude de l'existant, mettra l'accent sur la solution proposée et abordera la méthode adoptée. Il présentera également quelques notions théoriques importantes pour la réalisation de notre projet.

Dans le deuxième chapitre, nous dégagerons les besoins fonctionnels et non fonctionnels, comme nous spécifierons un diagramme de cas d'utilisation général du produit avec le langage de modélisation unifié UML.

Le troisième chapitre élaborera l'étude conceptuelle de notre projet dont lequel nous allons présenter quelques diagrammes.

Le quatrième chapitre « La phase de clôture » sert à présenter l'ensemble des différents outils utilisés pour concevoir et développer notre application ainsi que le diagramme

de déploiement et nous terminons par des captures d'écrans des principaux interfaces de l'application pour illustrer la version finale de notre produit.

Finalement, nous clôturerons ce rapport par une conclusion générale dans laquelle nous évaluerons le travail réalisé au sein de la société et nous proposerons des perspectives dans le but d'améliorer notre travail.

# Chapitre 1

## Contexte général du projet

“ ce chapitre introductif est dans le but de mettre le travail dans son contexte général. Nous commençons tout d’abord par une présentation de l’entreprise d’accueil. Ensuite, nous mettons l’accent à décrire le sujet autour du quel se déroule notre projet de fin d’études. Enfin, nous allons faire la critique du système actuel et de proposer les solutions adéquats avant de mettre à disposition le langage et la méthodologie de conception. ”

## 1.1 Présentation de l'organisme d'accueil

### 1.1.1 Cadre général de travail



FIGURE 1.1 – Logo de la société N3RD.

N3RD Créée en 2011 est une société d'ingénierie et de développement web qui, grâce à l'utilisation des nouvelles technologies du web, est capable aujourd'hui d'assurer les différentes tâches. [1].

### 1.1.2 Domaines d'activité de l'entreprise

N3RD est une société très active dans le domaine du développement Web et mobile (sites et applications), elle n'utilise que les dernières technologies :

- Web 2.0 et plus.
- Langages de développement côté serveur : PHP5, MySQL5, CSS3, HTML5
- Frameworks : Symfony2, Zend, Laravel et Code Igniter
- CMSs : Prestashop, Drupal et WordPress.
- Animation dynamique côté client : CSS3, JavaScript, Bootstrap, Ajax et JQuery.

### 1.1.3 Mission de l'entreprise

- Développement de sites et applications web pour tout business.
- Développement des solutions web personnalisées.



- Conseils.
- Boutiques en ligne et e-commerce.
- Design sur mesure.
- Création multimédia.
- Infogérance (hébergement, gestion d'infrastructure...).
- Création d'outils collaboratifs.
- Optimisation pour les plateformes mobiles.
- Création d'applications mobiles (Android IOS).

## 1.2 Étude de l'existant

### 1.2.1 Cadre du projet

Nous ne pouvons pas commencer ce travail sans avoir des informations et des idées claires et précises sur l'existant. L'étude de l'existant est une phase d'analyse qui consiste à faire un diagnostic sur les points forts et faibles du projet et déterminer des objectifs du nouveau système et l'ébauche de solution. Pour ce faire, il faut étudier le système existant lui même ainsi que l'environnement dans lequel il baigne.

Après observation des plusieurs site web université actuel nous avons constaté qu'il ne contient que les emplois du temps et les résultats en ligne ou quelques autres informations pour les étudiants(Stages..) .

### 1.2.2 Problématique

La formation à plusieurs universités se fait actuellement de façon traditionnelle (des enseignants, des étudiants et des cours sur place), et leur site ne contient ni des supports de cours ni des séries d'exercices et n'offre pas la possibilité de passer les examens en ligne ce qui gêne les étudiants qui travaillent et veulent poursuivre leurs études en parallèle. Alors notre mission est de résoudre ce problème.

### 1.2.3 Solution proposée

En tenant compte des différents problèmes que nous avons évoqués, nous sommes amenés à proposer une solution qui répond aux objectifs et qui pallie aux lacunes constatées aux niveau de processus de l'existant.

L'idée de notre projet consiste à mettre en place une application web qui facilite l'étude à distance qui a pour principales missions de faciliter l'apprentissage à distance en partageant les cours et les travaux dirigés entre l'étudiant et l'enseignant et en permettant de passer les examens et d'avoir les notes et les résultats en ligne.

## 1.3 Planification et conduite du projet

### 1.3.1 Démarche adoptée

Dans cette section, nous dévoilons le processus simplifié que nous préconisons pour la modélisation du système. Nous présenterons également les principes fondamentaux du EXtreme Programming (XP) et Scrum, afin d'éclairer les idées fortes auxquelles se rattache la démarche pratique adoptée dans la suite du rapport. )

.....	Scrum	EXtreme Programming (XP)
Description	(La méthode s'appuie sur le découpage d'un projet en «sprint», ainsi que l'autoorganisation de l'équipe de développement. Chaque sprint commence par une estimation suivie d'une planification opérationnelle. Le sprint se termine par une démonstration de ce qui a été achevé, et contribue à augmenter la valeur d'affaires du produit.)	(Ensemble de «Best Practices » de développement (Idéal pour le travail en groupe). - Cible des projets de moins de dix personnes.
Points forts )	-Amélioration de la communication. -Règles définies clairement. -Augmentation de productivité.	-Simple à mettre en œuvre. -Fait une large place aux aspects techniques :prototypes, règles de développement, tests. . .
Points faibles	- Violation de responsabilité. -L'équipe ne se prete pas au SCRUM.	(- Ne couvre pas les phases en amont et en aval au développement : capture des besoins, support, maintenance, tests d'intégration. . . -Assez flou dans sa mise en œuvre.)

TABLE 1.1 – Etude comparative entre les approches agiles

### 1.3.1.1 présentation de la methode Scrum

Scrum : Aujourd'hui « Scrum » est la méthode agile la plus populaire. Ce terme signifie « mêlée » au rugby. La méthode scrum s'appuie sur des « sprints » qui sont des espaces temps assez courts pouvant aller de quelques heures jusqu'à un mois. Généralement et de préférence un sprint s'étend sur deux semaines. À la fin de chaque sprint, l'équipe présente ce qu'elle a ajouté au produit.

### 1.3.1.2 Processus de développement

La nature du projet et sa forte dépendance aux acteurs du domaine sont les raisons qui expliquent le fait d'être toujours à l'écoute du client et prêt à répondre à ses nouveaux besoins. C'est pour cela, l'équipe du projet a opté pour un cycle de développement agile et plus précisément SCRUM.

Le principe de la méthodologie SCRUM est de développer un logiciel de manière incrémentale en maintenant une liste totalement transparente des demandes d'évolutions ou de corrections à implémenter (backlog).

Avec des livraisons très fréquentes, toutes les 4 semaines en général, le client reçoit un logiciel à chaque itération. Plus nous avançons dans le projet, plus le logiciel est complet et possède de plus en plus de fonctionnalités.

Pour cela, la méthode s'appuie sur des développements itératifs à un rythme constant d'une durée de 2 à 4 semaines (2 semaines pour notre cas) comme le montre la figure 1.2 :

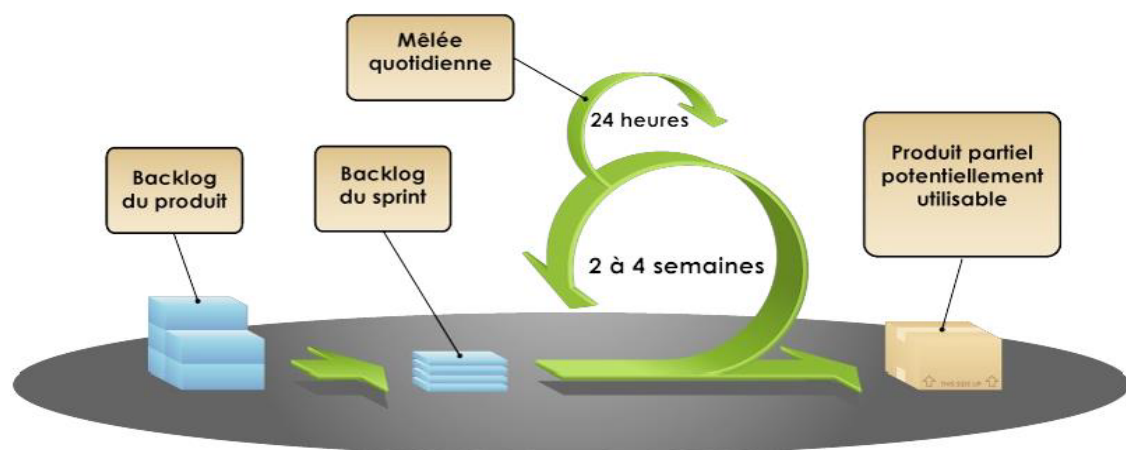


FIGURE 1.2 – Cycle de vie de la méthodologie scrum.

Le schéma illustre un exemple de planification en Scrum : les itérations (sprints) durent en pratique entre 2 et 4 semaines, et possède chacune un but. Le but de chaque sprint

une liste d'items du backlog de produit ou de fonctionnalités à réaliser. Ces items sont décomposés par l'équipe en tâches élémentaires de quelques heures.

Comme nous pouvons le remarquer dans cette figure, pour mettre en place la méthode SCRUM, il faut tout d'abord définir les différentes fonctionnalités de notre application qui forment le backlog du produit. Ensuite, vient l'étape de la planification du sprint pour définir le plan détaillé d'une itération.

Durant un sprint, il y a toujours des réunions quotidiennes entre les différents collaborateurs du projet afin de présenter l'état d'avancement des différentes tâches en cours, les difficultés rencontrées ainsi que les tâches restantes à réaliser. Une fois le produit partiel est prêt, nous vérifions la conformité de ce qui a été fait durant le sprint et nous pouvons alors l'améliorer en procédant à l'étape de rétrospective.

### 1.3.1.3 l'équipe Scrum

Scrum est considéré comme un cadre ou un «framework» de gestion de projet. Ce cadre est constitué d'une définition des rôles, il s'articule autour des trois rôles qui sont principalement les suivants :

- **Product Owner** : ( Dans la majorité des projets, le responsable produit (product owner) est le responsable de l'équipe projet client. C'est lui qui va définir et prioriser la liste des fonctionnalités du produit et choisir la date et le contenu de chaque sprint sur la base des valeurs (charges) qui lui sont communiquées par l'équipe.)
- **ScrumMaster** : Véritable facilitateur sur le projet, il veille à ce que chacun puisse travailler au maximum de ses capacités en éliminant les obstacles et en protégeant l'équipe des perturbations extérieures.
- **Équipe de développement** : elle regroupe l'ensemble des rôles habituellement nécessaires à un projet, à savoir le concepteur, le développeur, le testeur, etc. L'équipe s'organise elle-même et elle reste inchangée pendant toute la durée d'un sprint.

Dans notre cas, les rôles sont répartis comme suit :

Rôle	Personne
Product owner	La société N3RD
Scrum Master	Mr Ousleti nizar
Équipe de développement	Mr kouki hamza , Mlle khalfi khawla

TABLE 1.2 – Equipe Scrum

#### 1.3.1.4 les événements Scrum

Dans le cadre du scrum, il y a 05 événements pour créer de la régularité et minimiser le besoin de rencontres. Tous les événements sont classés dans le temps (time-boxed), ce qui signifie que ces événements ont une durée maximale.

- ☐ **Sprint** : c'est le cœur du scrum. Un sprint est lorsqu'un incrément de produit est créé. Un incrément est une partie utilisable et potentiellement fonctionnelle du produit final. Les sprints sont classés dans le temps pour un mois ou moins. Chaque sprint a un objectif de ce qui doit être construit.
- ☐ **La planification du sprint (Sprint planning)** est l'événement où l'équipe scrum définit ce qui sera livré dans un sprint. Cet événement est limité dans le temps à un maximum de huit heures pour un sprint d'un mois. Dans la planification du sprint, l'équipe scrum définira ce qui peut être livré au prochain incrément et combien de travail est nécessaire pour atteindre cet objectif. L'entrée principale de cette réunion est le Product Backlog où le Product Owner et le DevTeam choisiront les éléments qui seront inclus dans ce sprint pour atteindre l'objectif du sprint.
- ☐ **Le Daily scrum** est une réunion quotidienne de 15 minutes pour DevTeam où ils mettront à jour le statut de travail et les plans pour les prochaines 24 heures. Cette réunion est utilisée pour inspecter l'état d'avancement du sprint. Lors de cette réunion, chaque membre du DevTeam répondra aux questions suivantes :
  1. Qu'est-ce qui a été accompli depuis la dernière réunion ?
  2. Que fera-t-on avant la prochaine rencontre ?
  3. Voyez-vous un obstacle qui vous empêche, ou le DevTeam, d'atteindre l'objectif de sprint ?
- ☐ **La revue de sprint (Sprint review)** est utilisée par l'équipe scrum pour présenter et inspecter l'incrémentation à la fin d'un sprint. Il est limité à quatre heures pour un sprint d'un mois. Cette réunion est destinée à obtenir des commentaires et des demandes de changement du client et des parties prenantes. Seuls les éléments considérés comme « terminés (Done) » sont inclus dans la revue de sprint.
- ☐ **La rétrospective Sprint (Sprint retrospective)** est un événement destiné à traiter l'amélioration. Les améliorations pourraient concerner les personnes, les relations, les processus et les outils. Il faut trois heures pour un sprint d'un mois.

### 1.3.2 Planning du projet

La planification du projet est une phase importante d'avant-projet. Elle consiste à prévoir le déroulement de ce dernier tout au long des phases constituant le cycle de développement.

- **Le diagramme de Gantt :**

Le diagramme de Gantt, couramment utilisé en gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet. Ce diagramme permet donc de visualiser d'un seul coup d'œil :

- ★ Les différentes tâches à envisager.
- ★ La date de début et la date de fin de chaque tâche.
- ★ La durée escomptée de chaque tâche.
- ★ Le chevauchement éventuel des tâches, et la durée de ce chevauchement.
- ★ La date de début et la date de fin du projet dans son ensemble.

Le diagramme de Gantt dans la figure 1.3 illustre le déroulement du stage dans le temps :

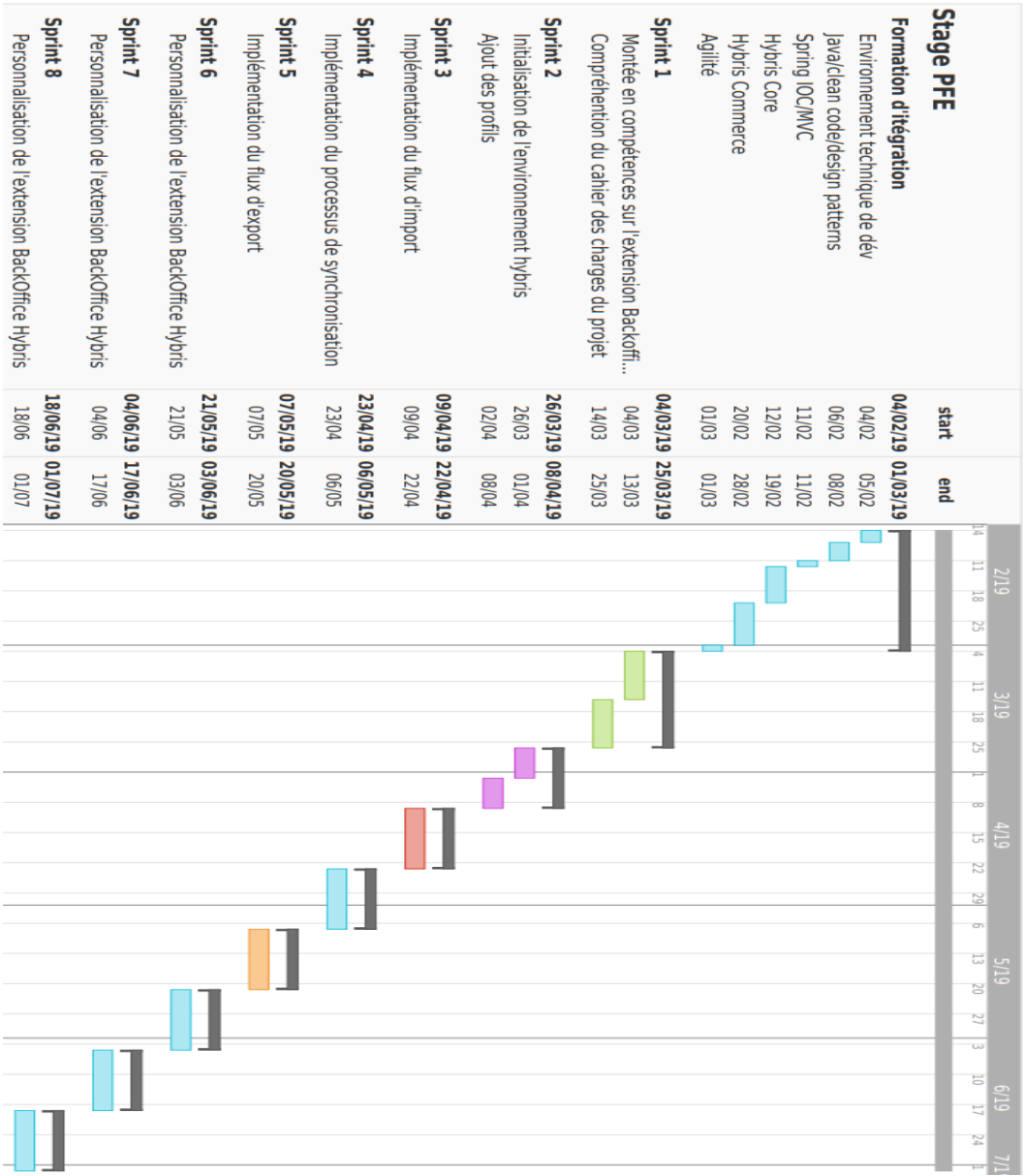


FIGURE 1.3 – Diagramme de gantt du projet.

### 1.3.3 La modélisation objet

Depuis quelques années, la modélisation objet avec le langage UML est devenue une pratique courante sur de nombreux projets informatiques. En effet, Le recours à la modélisation est une pratique indispensable au développement logiciel, car un modèle est prévu pour arriver à anticiper les résultats du codage.

#### 1.3.3.1 Définition de UML

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise. UML s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel [3].

#### 1.3.3.2 Processus de modélisation

Le processus que nous allons présenter et appliquer tout au long de ce rapport :

- Conduit par les cas d'utilisation ;
- Relativement léger et restreint, comme les méthodes agiles, mais sans négliger les activités de modélisation en analyse et conception ;
- Fondé sur l'utilisation d'un sous-ensemble nécessaire et suffisant du langage UML, conformément à méthodes agiles.

## 1.4 Conclusion

Tout au long de ce chapitre, nous avons décrit l'organisme d'accueil , nous avons aussi formulé une petite présentation de notre projet pour vous mettre en contexte. Nous avons ensuite fait une étude de l'existant dans l'entreprise pour pouvoir dégager les différentes lacunes ainsi que les solutions envisagées. A la fin de ce chapitre, nous avons dévoilé le langage et la méthodologie de conception de notre système. Le chapitre suivant sera consacré à la planification du projet ainsi que la spécification des besoins



## Chapitre 2

# Analyse et spécification des besoins

“ Dans ce chapitre, j’aborde les phases d’analyse et de spécification des besoins du projet, dans le but d’avoir une vision globale claire du comportement du projet ainsi que les attentes des utilisateurs. ”

## 2.1 Spécifications des besoins

La spécification des besoins va nous permettre d'avoir une meilleure approche des utilisateurs, des fonctionnalités et de la relation entre les deux. Elle sera sous forme de besoins. Pour cela nous allons procéder comme ceci :

- Identification des acteurs du nouveau système.
- Identification des besoins fonctionnels.
- Identification des besoins non fonctionnels .

### 2.1.1 Identification des acteurs

Un acteur est une personne, un matériel ou un logiciel qui interagit avec le système. L'analyse du présent projet commence par une identification des acteurs agissants sur les différentes parties du système. Les acteurs présentés dans la figure 2.1 sont des employés du clients en plus du serveur qui est la solution adoptée par le client pour la communication et le partage des informations entre ces systèmes et départements .

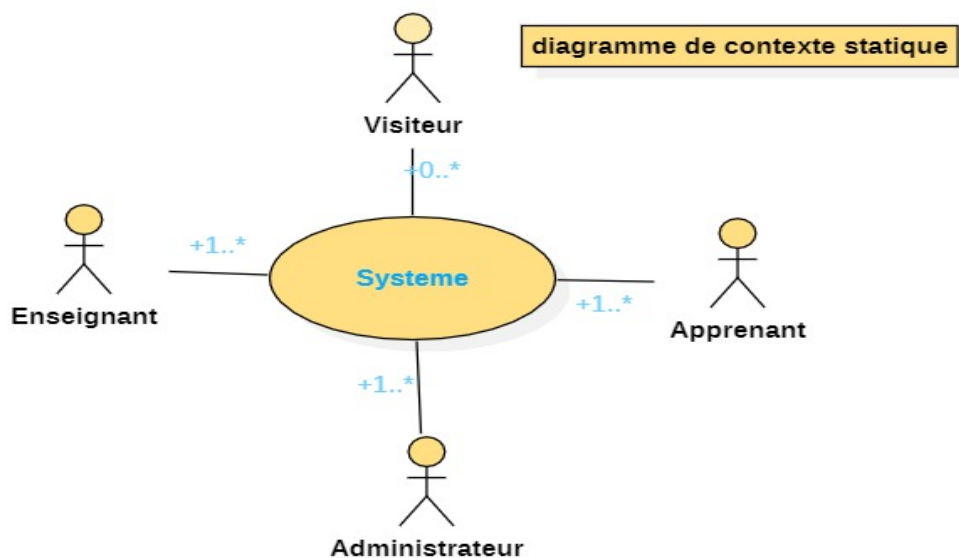


FIGURE 2.1 – Héirarchie des profiles humaines.

Le tableau 2.1 récapitule les acteurs en interaction avec le système en spécifiant le rôle de chacun avant de définir plus précisément leurs interactions avec le système en utilisant

des diagrammes de cas d'utilisation.

Acteur	Fonction
Administrateur	L'administrateur est la personne responsable de gérer la totalité du système.
Enseignant	C'est un acteur principale qui interagit avec notre application. C'est l'acteur qui a pour rôle de gérer les cours, les travaux dirigés (TD) et les examens des étudiants.
Apprenant	L'apprenant inscrit, il va pouvoir consulter les cours et faire les tests qui lui sont proposés. L'apprenant peut avoir la possibilité de participer aux forums, d'envoyer un message à un tuteur, à un autre apprenant ou même à l'administrateur. Ainsi la possibilité de discussion en ligne avec le tuteur, la modification de son profil et la consultation de ses résultats.
Visiteur	N'importe quel visiteur qui veut télécharger des cours via un compte personnelle a condition de faire les inscriptions pour avoir un compte.

TABLE 2.1 – Acteurs en interaction avec le système

### 2.1.2 Les besoins fonctionnels

Les besoins fonctionnels expriment une action que doit effectuer le système en réponse à une demande .

Les besoins principaux à couvrir par le système sont les suivants :

Si l'acteur est un Administrateur, il peut :

- S'authentifier : l'Administrateur entre son « username » et son « password » avant d'accéder à l'application pour assurer la confidentialité des informations.
- Gérer les comptes des utilisateurs : l'Administrateur peut ajouter des comptes pour les nouveaux Utilisateurs (Enseignant ou Etudiant) , modifier leurs informations et supprimer les comptes des anciens utilisateurs.

Si l'acteur est un Enseignant, il peut :

- Gérer les Matières :l'Enseignant peut ajouter, modifier et supprimer les matières.
- Gérer les Cours :l'Enseignant peut ajouter, modifier et supprimer les cours.
- Gérer les Travaux dirigés(TD) :l'Enseignant peut ajouter,modifier et supprimer les Travaux dirigés(TD).
- Gérer les Examens :l'Enseignant peut ajouter des examens en choisissant une durée de temps déterminé.

Si l'acteur est un Etudiant, il peut :

- Consulter les Cours :l'Etudiant peut consulter et télécharger les cours.
- Consulter les Travaux dirigés(TD) :l'Etudiant peut consulter et télécharger les Travaux dirigés(TD).
- Passer les Examens :l'Etudiant peut passer les examens en respectant une durée de temps déterminée.

### 2.1.3 Les besoins non fonctionnels

Les besoins non fonctionnels impressionne directement sur déroulement réelle de l'application. Ce sont des besoins techniques décrivant la majorité des contraintes (qu'on a déjà approuvée dans le chapitre précédent) auxquelles est soumis le système pour sa réalisation et son bon fonctionnement. Pour cela l'ensemble des extensions à réaliser doivent respecter les besoins suivants :

- La Sécurité : La solution proposée permet à l'utilisateur une navigation sécurisée. Elle n'est accessible qu'avec une authentification.
- Ergonomie de l'interface : L'ergonomie est un élément important de l'application : les écrans de saisie doivent être clairs, organisés avec cohérence, de façon à ce qu'une prise en main soit la plus rapide possible.
- Maintenance : L'une des plus importantes besoins de notre application est la facilité de modification pour s'adopter aux nouveaux besoins.
- Portabilité : L'application doit être accessible via n'importe quel navigateur.

## 2.2 Backlog produit

Le Backlog produit est une liste ordonnée de tout ce qui pourrait être nécessaire dans un produit et constitue l'unique source d'exigences pour toutes les modifications apportées au produit. Le Product Owner est responsable du Backlog produit, y compris son contenu, sa disponibilité et son ordonnancement.

Ses toutes premières moutures ne font qu'esquisser les besoins tels qu'initialement connus et compris. Le Backlog Produit évolue au fur et à mesure que le produit et le contexte dans lequel il sera utilisé évoluent. Le Backlog Produit est dynamique ; il change constamment pour identifier ce que le produit requiert pour être approprié, compétitif et utile. Tant et aussi longtemps qu'un produit existe, son Backlog Produit correspondant existe.

Les caractéristiques fonctionnelles sont appelées des histoires utilisateurs (user story). Les user stories sont caractérisés par :

- **Identifiant** Il détermine un identifiant unique pour l'histoire en question.
- **Description** Elle décrit le besoin d'un acteur.
- **Critères d'acceptation** À chaque user story sont associés des critères permettant au client de tester l'histoire. Ces critères d'acceptation peuvent être formalisés, pour aller un peu plus loin dans l'aide fournie à l'équipe que l'énoncé de ces critères.
- **Estimation** Est une estimation de la complexité, elle est une valeur entière qui appartient à la suite de Fibonacci.
- **Priorité** Les priorités sont utilisées pour définir l'ordre de réalisation, elles permettent de constituer le flux de stories qui va alimenter l'équipe. Pour prioriser nos user stories, nous avons pris en compte les critères suivant :
  1. *La valeur apportée (Business Value)*
  2. *La fréquence d'utilisation*
  3. *La réduction des risques*
  4. *L'incertitude sur des besoins des utilisateurs qu'un user story permettra de diminuer*
  5. *La contribution à la qualité. Les travaux visant à garantir la qualité du produit devraient être prioritaires*
  6. *Les dépendances entre stories*

Lors de la création de notre Backlog, nous avons essayé de produire des user stories qui respectent les critères réunies dans le mot INVEST, c'est à dire

- ★ Indépendant : Ne dépend de rien (réduire les liens entre items)

- ★ Négociable : Je n'ai pas une solution technique figée
- ★ Valuable : pour le client (a une valeur Business)
- ★ Estimable : Estimation en complexité
- ★ Small / Sized Appropriately : De petite taille (A définir en interne de l'entreprise)
- ★ Testable : Pour la validation de l'item

### 2.2.1 Planification des Sprints

-Duree des Sprints

### 2.2.2 Backlog des Sprints

Le Sprint Backlog comporte la liste des tâches du Sprint (son périmètre donc) ainsi que la charge de travail associée à ces dernières. Chaque jour, le Reste A Faire de chaque tâche est actualisé par l'équipe de développement afin de tracer le graphique d'avancement de Sprint.

## 2.3 Diagrammes des cas d'utilisation global

Le modèle des cas d'utilisation décrit les fonctionnalités d'un système d'un point de vue utilisateur, sous la forme d'actions et de réactions ; l'ensemble des fonctionnalités est déterminé en examinant les besoins fonctionnels de tous les utilisateurs potentiels. Ainsi, pour construire notre modèle, nous allons organiser les cas d'utilisation et les regrouper en ensembles fonctionnels cohérents. Pour ce faire, nous utilisons le concept général d'UML, le package.

Ce diagramme illustre le cas d'utilisation générale de notre système. Ces cas d'utilisation seront par la suite expliqués en détail. (voir la figure 2.2) :

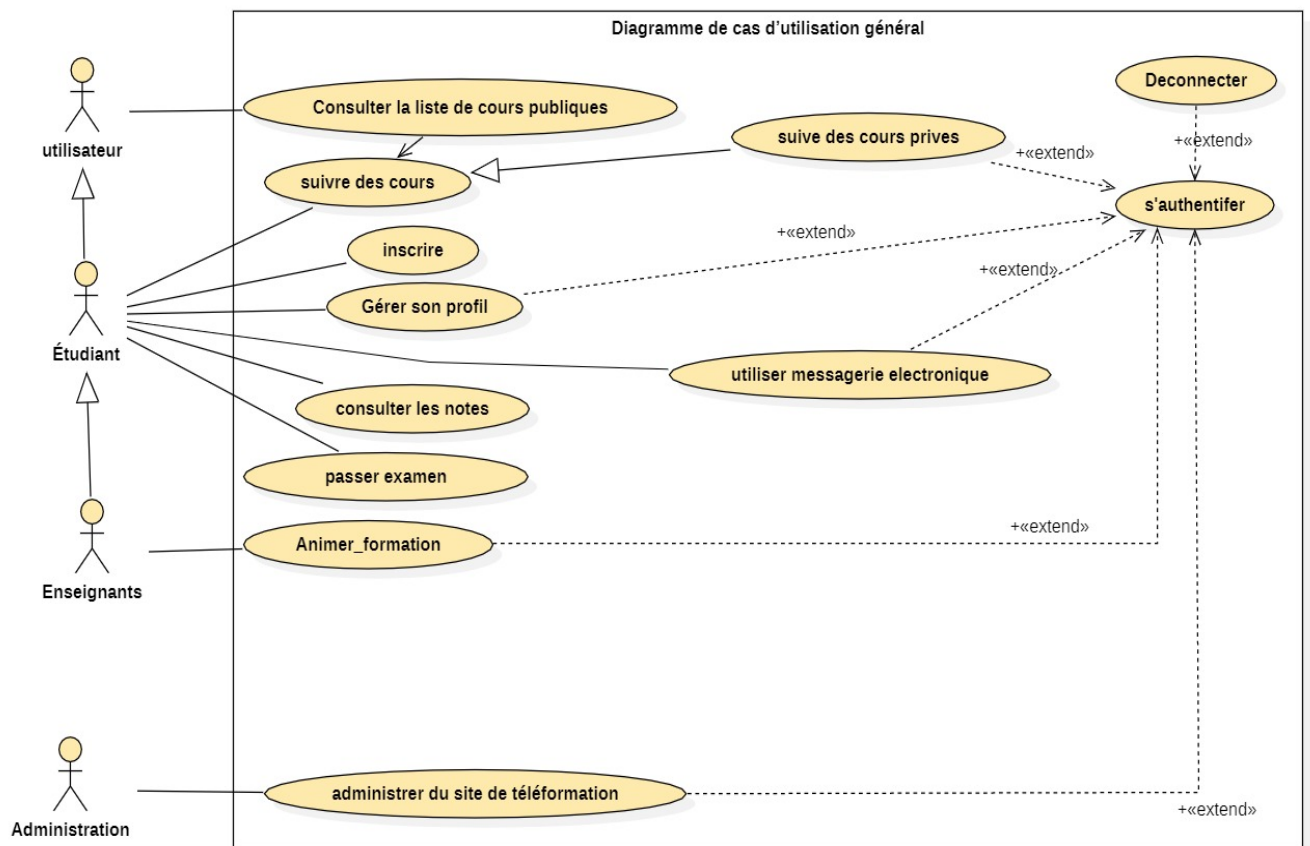


FIGURE 2.2 – Les cas d'utilisation global.

## 2.4 Diagrammes de classes global

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques. Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensembles par un champ sémantique . Dans ce qui suit nous allons décrire le diagramme de classes relatif à notre application. 2.3.

Voici quelques notions de base du diagramme :

- **Une classe** : représente la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques. On peut parler également de type.
- **Un attribut** : représente un type d'information contenu dans une classe .
- **Une opération** : représente un élément de comportement (un service) contenu dans une classe.
- **Une association** : représente une relation sémantique durable entre deux classes.
- **Une superclasse** : est une classe plus générale reliée à une ou plusieurs autres classes plus spécialisées (sous-classes) par une relation de généralisation. Les sous-classes «Héritent» des propriétés de leur superclasse et peuvent comporter des propriétés spécifiques supplémentaires.

La figure ci-dessous représente le diagramme de classes [2] :



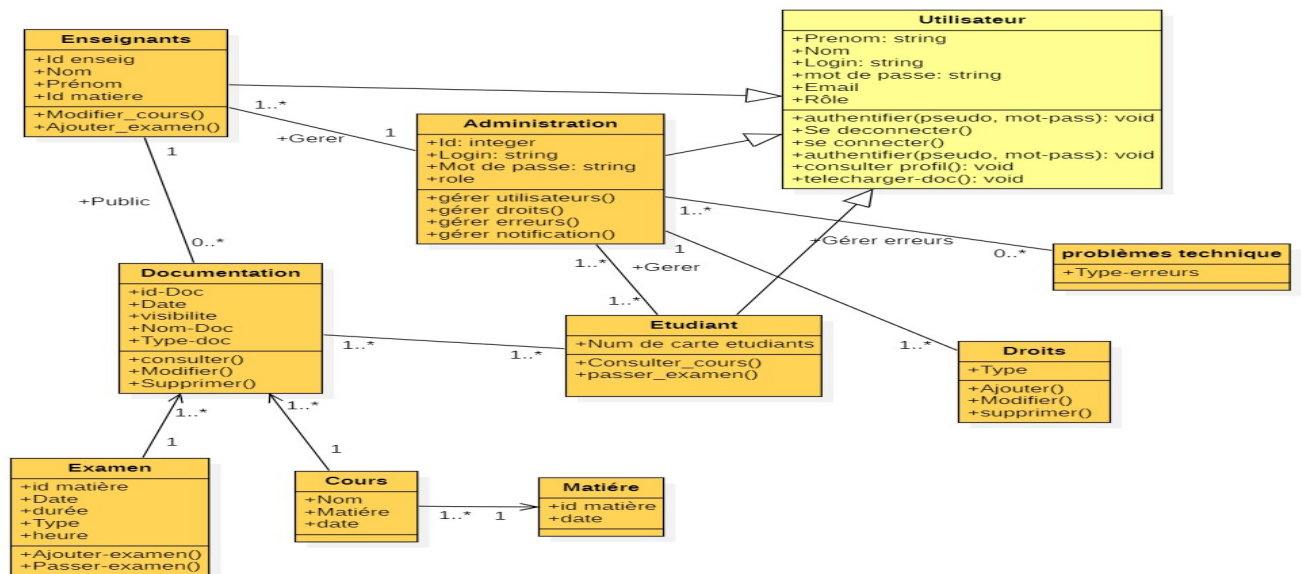


FIGURE 2.3 – Diagrammes de classes global.

### Description des entités :

- **Enseignants** : L'utilisateur principal du module, il a accès à plusieurs fonctionnalités qui vont lui permettre de réussir le processus éducatif.

Champs	Types	Contraintes
Id enseig	1	2
Nom	2	4
Id matiere	3	6

TABLE 2.2 – Tables "Enseignants"

- **Administration** : C'est la classe qui contient toutes les actions prises en charge par l'administrateur :

Champs	Types	Contraintes
Id : intege	1	2
Login : string	2	4
Mot de passe : string	1	2
role	2	4

TABLE 2.3 – Tables "Administration"

- **Documentation** :ca englobe toute documentation nécessaires au processus d'enseignement à distance

Champs	Types	Contraintes
id-Doc	1	2
Date	2	4
visibilite	1	2
Nom-Doc	2	4
Type-doc	1	2

TABLE 2.4 – Tables "Documentation"

- **Examen** : ça englobe toute les examens requis

Champs	Types	Contraintes
id matière	1	2
Date	2	4
durée	1	2
Type	2	4
heure	2	4

TABLE 2.5 – Tables "Examen"

- *Utilisateur* : elle contient tous les utilisateurs du plateforme selon leur :

Champs	Types	Contraintes
Prenom : string	1	2
Nom	2	4
Login : string	2	4
mot de passe : string	2	4
Email	2	4
Rôle	2	4

TABLE 2.6 – Tables "Utilisateur"

- *Cours* : ca englobe toute les cours nécessaires au processus d'enseignement à distance

Champs	Types	Contraintes
Nom	1	2
Matière	2	4
date	2	4

TABLE 2.7 – Tables "Cours"

- *Matière* : ca englobe toute les Matières

Champs	Types	Contraintes
id matière	1	2
date	2	4

TABLE 2.8 – Tables " Matière"

- *Droits* : c'est la classe qui contient les droits attribués aux plateforme par l'administrateur ainsi la suppression ou l'ajout de certains privilèges.

Champs	Types	Contraintes
Type	1	2

TABLE 2.9 – Tables "Droits"

- ***problèmes technique*** : C'est la classe qui contient toutes les problèmes technique prises en charge par l'administrateur :

Champs	Types	Contraintes
Type-erreurs	1	2

TABLE 2.10 – Tables "problèmes technique"

- ***Etudiant*** : Le deuxième utilisateur du module avec des fonctionnalités facilitant la tâche de mémorisation.

Champs	Types	Contraintes
Num de carte etudiants	1	2

TABLE 2.11 – Tables "Etudiant"

## 2.5 Conclusion

Le but de ce chapitre était de définir et d'analyser l'ensemble des besoins fonctionnels de notre solution. Cette étape est primordiale dans le développement d'un projet informatique puisque elle nous permet de définir le périmètre fonctionnel du projet, et de garantir la couverture de l'ensemble des fonctionnalités recensées.

# Chapitre 3

## La réalisation

“ *Puisque Scrum est choisi comme méthode de gestion de projets, ce chapitre va être réparti selon les exigences de Scrum, en effet, le travail est divisé en Sprints, chacun d’eux a lieu de définir le but et le Sprint Backlog dans un premier temps, ensuite nous présentons la conception et la réalisation. Enfin, nous clôturons chaque Sprint par sa revue et une rétrospective.* ”

## 3.1 Sprint 1 : Utilisateur

“ Ce premier sprint s'étale sur 18 jours et se décompose en trois items ”

- ◇ Authentification
- ◇ Gérer le profil
- ◇ Modification de rôle

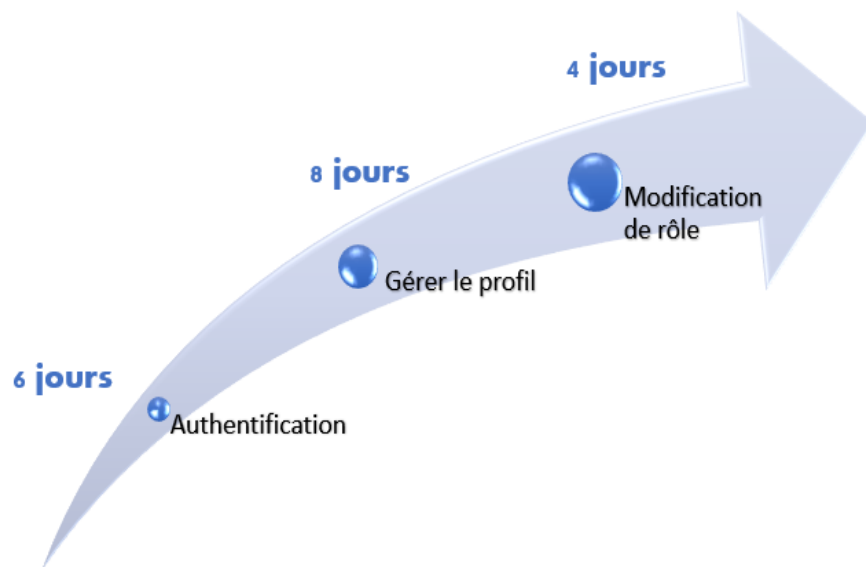


FIGURE 3.1 – Décomposition sprint 1 en Items.

Le backlog du sprint 2 est le suivant :

Item	User Story	Description	Priorité
<b>s'authentifier</b>	s'authentifier	je dois m'identifier pour acceder a mon espace	1
<b>gérer profil</b>	Consulter profil	En tant qu'utilisateur je peux consulter mon profil	3
	Modifier profil	En tant qu'utilisateur je peux modifier mon profil	
	Modifier image de profil	En tant qu'utilisatuer je peux uploader modifier une image de profil	
	Désactiver profil	En tant qu'utilisateur je peux désactiver mon profil	
<b>s'inscrire</b>	s'inscrire	En tant qu'utilisateur, je peux m'inscrire	1

TABLE 3.1 – Tables Backlog du sprint 2

les user stories de sprint 2 :

ID U.S	User Story
1.1	En tant qu'utilisateur, je dois m'authentifier pour accéder à mon espace
1.2	En tant qu'utilisateur, je peux m'inscrire
2.1	En tant qu'utilisateur je peux uploader une image de profil
2.2	En tant qu'utilisateur je peux afficher mon profil
2.3	En tant qu'utilisateur je peux modifier mon profil
2.4	En tant qu'utilisateur je peux désactiver mon profil

TABLE 3.2 – Tables "les user stories de sprint 2"



### 3.1.1 item 1 : Authentification

#### 3.1.1.1 Diagramme de cas d'utilisation

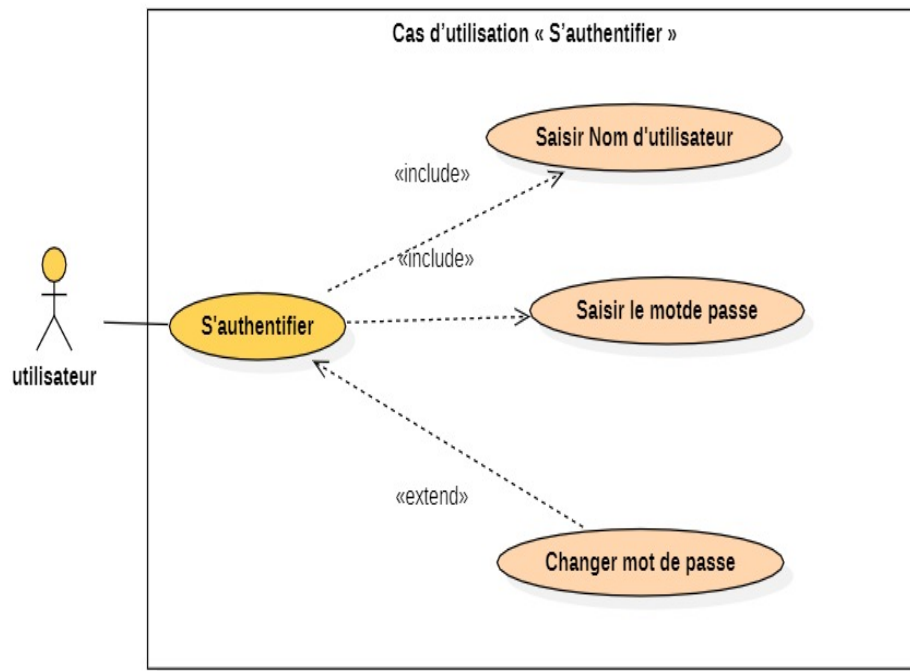


FIGURE 3.2 – Diagramme de Cas d' utilisation S' authentifier .

##### 3.1.1.1.a Description détaillée des cas d'utilisations

- **Cas d'utilisation S'authentifier** : permet aux utilisateurs de se connecter au système avec leurs logins et mots de passe afin de sécuriser la plateforme.
  - **Objectif** : Cette fonctionnalité permet aux différents acteurs de se connecter.
  - **Acteur** : Tous les acteurs
  - **Pré-condition** : L'utilisateur existe dans la base de données.
  - **Post-conditions** : Utilisateur authentifié.
  - **Scénario nominal** :
    1. L'acteur saisit son login et son mot de passe.
    2. Le système vérifie les informations saisies.
    3. Le système trouve que les informations saisies sont valides.

4. Le système vérifie le rôle de l'acteur.
5. Le système connecte l'acteur à son espace.

– *Scénario d'erreur :*

1. L'acteur saisit son login et son mot de passe.
2. Le système vérifie les informations saisies.
3. Le système trouve que les informations saisies sont invalides.
4. Le système demande à l'acteur de vérifier les informations saisies.

### 3.1.1.2 Diagrammes de séquence système

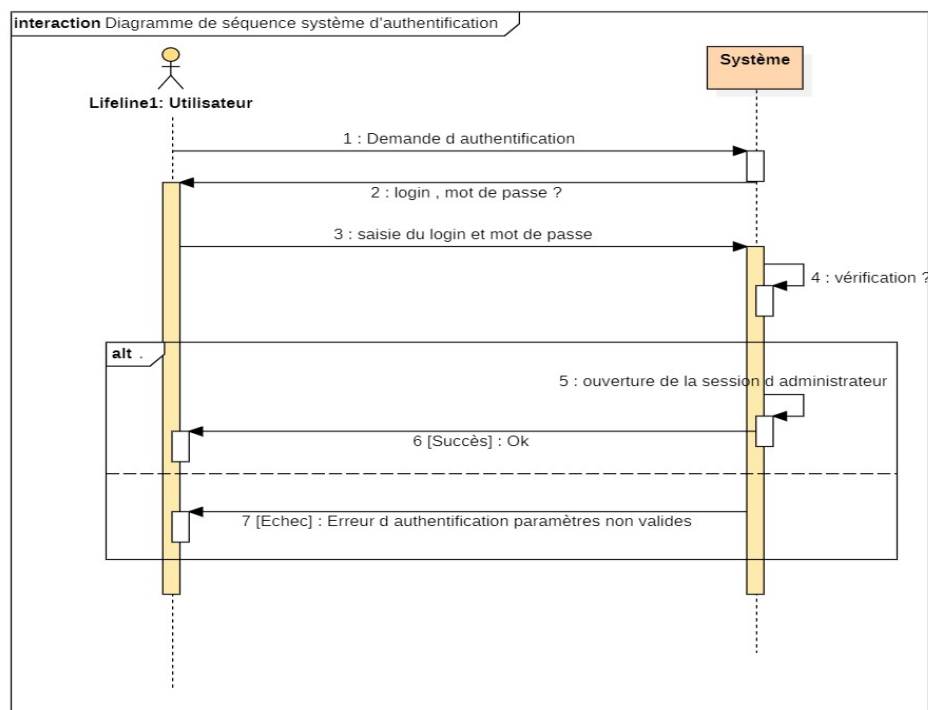


FIGURE 3.3 – Diagramme de séquence système d'authentification .

### 3.1.2 item 2 : Gérer le profil

#### 3.1.2.1 Diagramme de cas d'utilisation

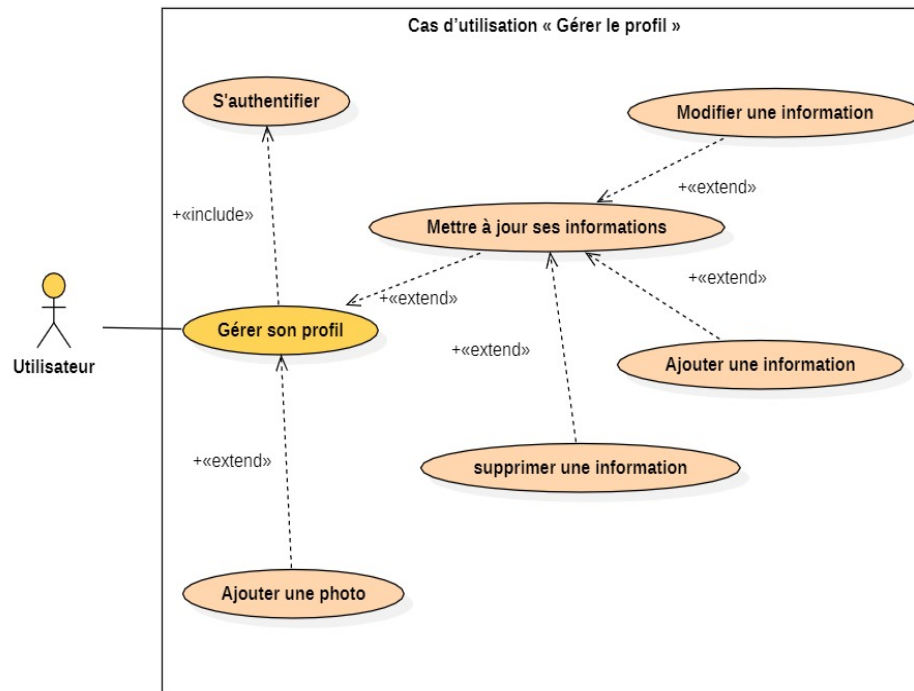


FIGURE 3.4 – Diagramme de Cas d' utilisation Gérer le profil .

- **Cas d'utilisation Gérer le profil** : permet à l'acteur de mettre à jour ses informations.
  - **Objectif** : Cette fonctionnalité permet aux différents acteurs de mettre à jour ses informations.
  - **Acteur** : Tous les acteurs
  - **Pré-condition** : L'acteur doit être un membre identifié.
  - **Post-conditions** : le cas démarre après le point 02 de l'enchainement nominal.
  - **Scénario nominal** :
    1. le système affiche le profil actuel de l'acteur.
    2. l'acteur met à jour ses informations.
    3. le système vérifie la validité des informations saisies.
    4. le système enregistre ces informations dans la base de données .

5. le système notifie l'acteur du bon déroulement de mise à jour de son profil

– *Scénario alternative :*

les informations sont manquantes ou incorrectes : ce scénario commence au point 03 du scénario nominal.

1. Le système informe l'acteur que les données saisies sont erronées, garde les informations saisies avant et le scénario reprend au point 02 du scénario nominal.

## 3.1.3 item 3 : Modification de rôle

## 3.1.3.1 Diagrammes de séquence système

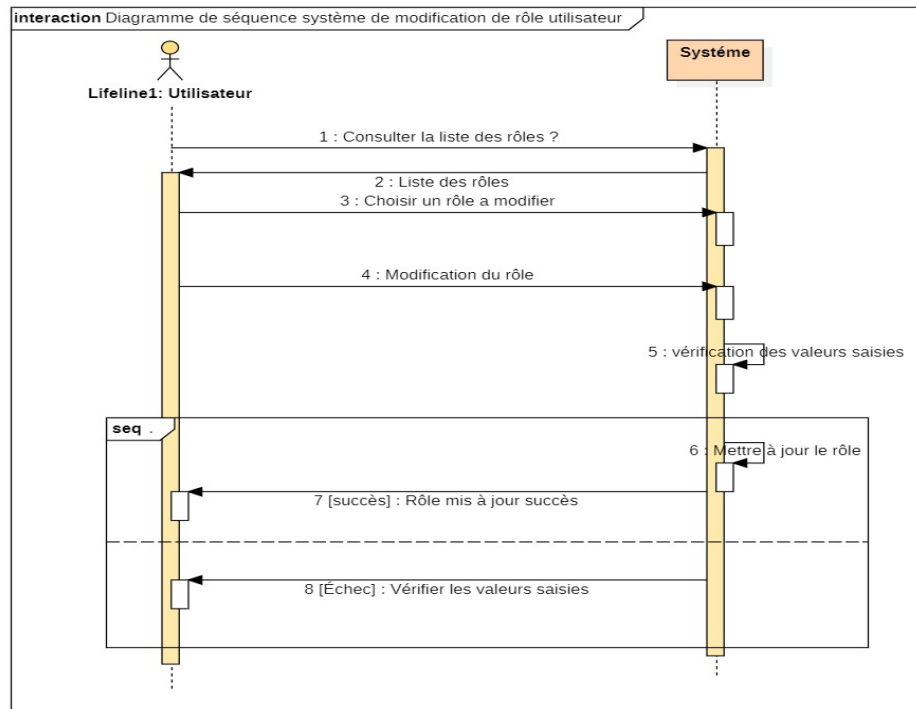


FIGURE 3.5 – Diagramme de séquence système de modification de rôle utilisateur .

## 3.2 Sprint 2 : Administration

“ Ce premier sprint s'étale sur 26 jours et se décompose en deux items ”

- ◇ Gérer les utilisateurs
- ◇ Inscription et réinscription d'un étudiant

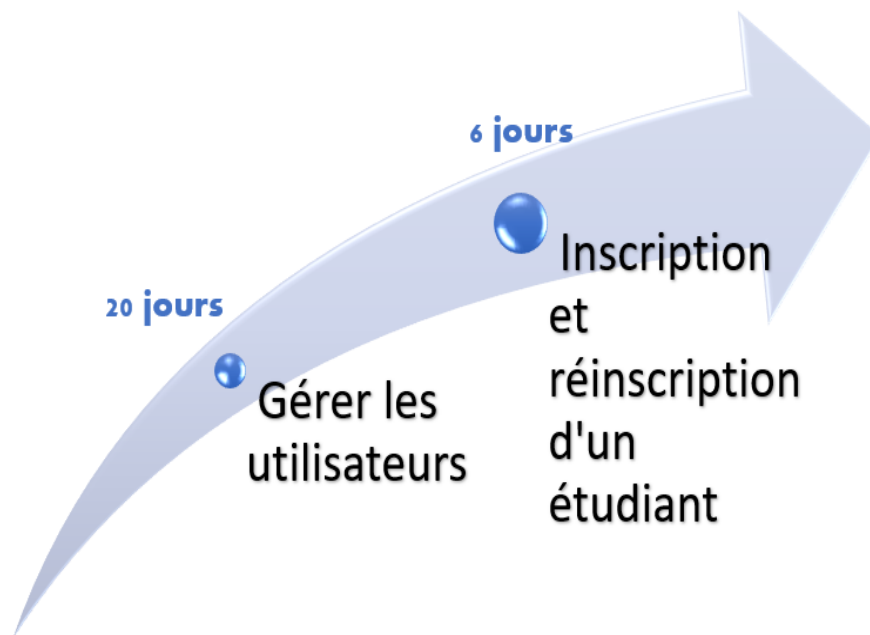


FIGURE 3.6 – Décomposition sprint 2 en Items.

Le backlog du sprint 2 est le suivant :

Item	User Story	Description	Priorité
<b>s'authentifier</b>	s'authentifier	je dois m'identifier pour acceder a mon espace	1
<b>gérer profil</b>	Consulter profil	En tant qu'utilisateur je peux consulter mon profil	3
	Modifier profil	En tant qu'utilisateur je peux modifier mon profil	
	Modifier image de profil	En tant qu'utilisatuer je peux uploader modifier une image de profil	
	Désactiver profil	En tant qu'utilisateur je peux désactiver mon profil	
<b>s'inscrire</b>	s'inscrire	En tant qu'utilisateur, je peux m'inscrire	1

TABLE 3.3 – Tables Backlog du sprint 2

les user stories de sprint 2 :

ID U.S	User Story
1.1	En tant qu'utilisateur, je dois m'authentifier pour accéder à mon espace
1.2	En tant qu'utilisateur, je peux m'inscrire
2.1	En tant qu'utilisateur je peux uploader une image de profil
2.2	En tant qu'utilisateur je peux afficher mon profil
2.3	En tant qu'utilisateur je peux modifier mon profil
2.4	En tant qu'utilisateur je peux désactiver mon profil

TABLE 3.4 – Tables "les user stories de sprint 2"

### 3.2.1 item 1 : Gérer les utilisateurs

#### 3.2.1.1 Diagramme de cas d'utilisation détaillé «administrer du site de téléformation»

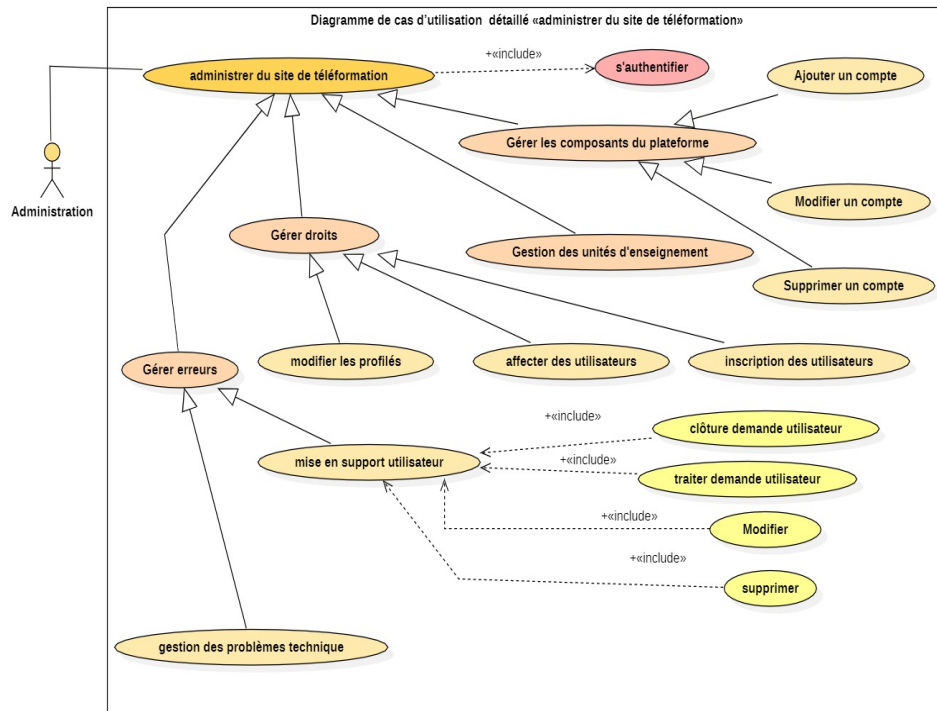


FIGURE 3.7 – Diagramme de cas d' utilisation détaillé «administrer» .

##### 3.2.1.1.a Description détaillée des cas d'utilisations

L'administrateur a comme rôle principale de gérer toutes les taches de la plateforme ; gestion des enseignants, étudiants ,matières....ainsi que les affectations des enseignants. L'administrateur peut ajouter, supprimer ou modifier les différentes informations des étudiants d'une manière permanente. L'administrateur peut gérer des droits et des erreurs .

- **Cas d'utilisation Gérer le profil** : permet à l'acteur de mettre à jour la plateforme.
  - **Objectif** : permet à l'administrateur de modifier la composition interne de la plateforme.



- *Acteur* : Administrateur
- *Pré-condition* : L'acteur doit être connecté.
- *Post-conditions* :
- *Scénario nominal* :
  1. le système affiche l'état actuel de la plateforme.
  2. l'acteur met à jour la plateforme.
  3. le système vérifie la validité des mis à jour.
  4. le système enregistre les mis à jours dans la base de données.
  5. le système notifie l'acteur du bon déroulement de mise à jour de la plateforme.
- *Scénario alternative* :

les informations sont manquantes ou incorrectes : ce scénario commence au point 03 du scénario nominal

  1. Le système informe l'acteur que les mis à jour sont erronées, garde l'état de la plateforme

#### 3.2.1.1.b Description détaillée des cas d'utilisations

- Cas d'utilisation Gérer les utilisateurs :

- *Objectif* : permet à l'acteur d'ajouter et de supprimer un utilisateur.
- *Acteur* : Administrateur
- *Pré-condition* : L'acteur doit être connecté.
- *Post-conditions* :
- *Scénario nominal* :
  1. Le système affiche un formulaire d'inscription à l'acteur.
  2. L'acteur saisit les informations du nouvel utilisateur et lui affecter un rôle.
  3. Le système vérifie la validité des informations saisies.
  4. Le système enregistre ces informations dans la base de données.
  5. Le système notifie l'acteur du bon déroulement de l'inscription..
- *Scénario alternative* :

les informations sont manquantes ou incorrectes : ce scénario commence au point 03 du scénario nominal.

  1. Le système informe l'acteur que les données saisies sont erronées et le scénario reprend au point 02 du scénario nominal.

## 3.2.1.2 Diagrammes de séquence du cas d' utilisation "Modifier un utilisateur"

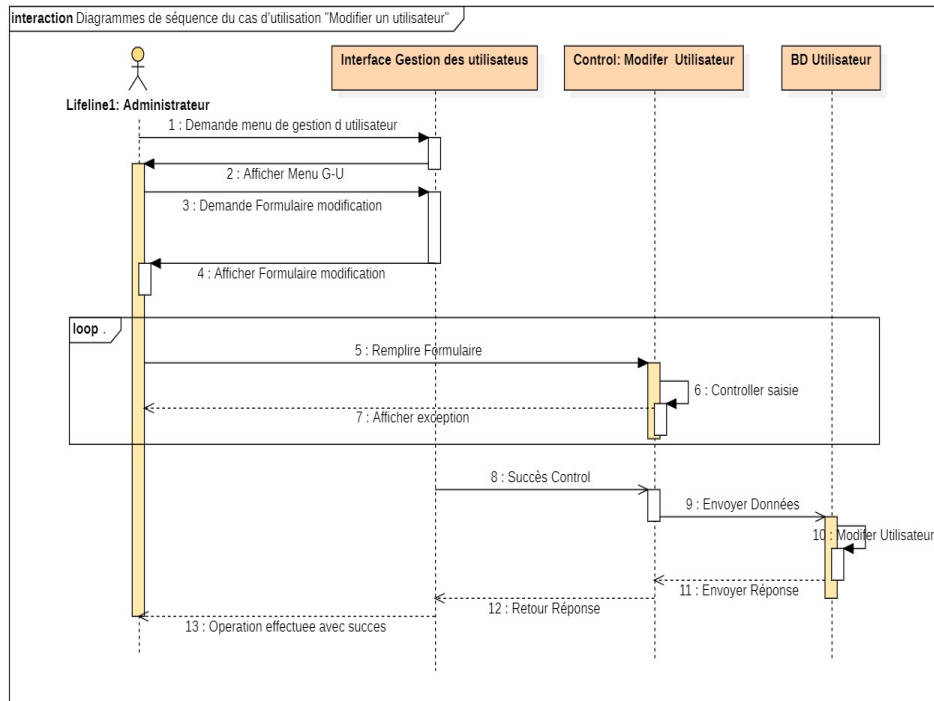


FIGURE 3.8 – Diagrammes de séquence du cas d' utilisation "Modifier un utilisateur" .

## 3.2.1.2.a Description détaillée des cas d'utilisations

- Cas d'utilisation "Modifier un utilisateur" :

- **Objectif** : Modifier un utilisateur
- **Acteur** : Administrateur
- **Pré-condition** : Authentification préalable.  
Utilisateur existant.  
Formulaire d'ajout disponible.
- **Post-conditions** : L'utilisateur a bien été modifié .
- **Scénario nominal** :
  1. L'utilisateur demande le formulaire de modification d'un utilisateur.
  2. Le système affiche le formulaire avec l'ensemble des anciennes informations de l'utilisateur.

3. L'administrateur modifie les champs nécessaires.
4. Le système vérifie les données saisies.
5. L'administrateur valide la modification .
6. Le système vérifie l'existence de l'utilisateur .
7. Le système modifie l'utilisateur.

– *Scénario alternative :*

1. L'utilisateur saisit des informations manquantes ou erronées.
2. Le système renvoie un message d'erreur adéquat.
3. Reprise de l'étape 3 du scénario Nominal.

### 3.2.1.3 Diagramme de séquence du cas d'utilisation "Supprimer un utilisateur"

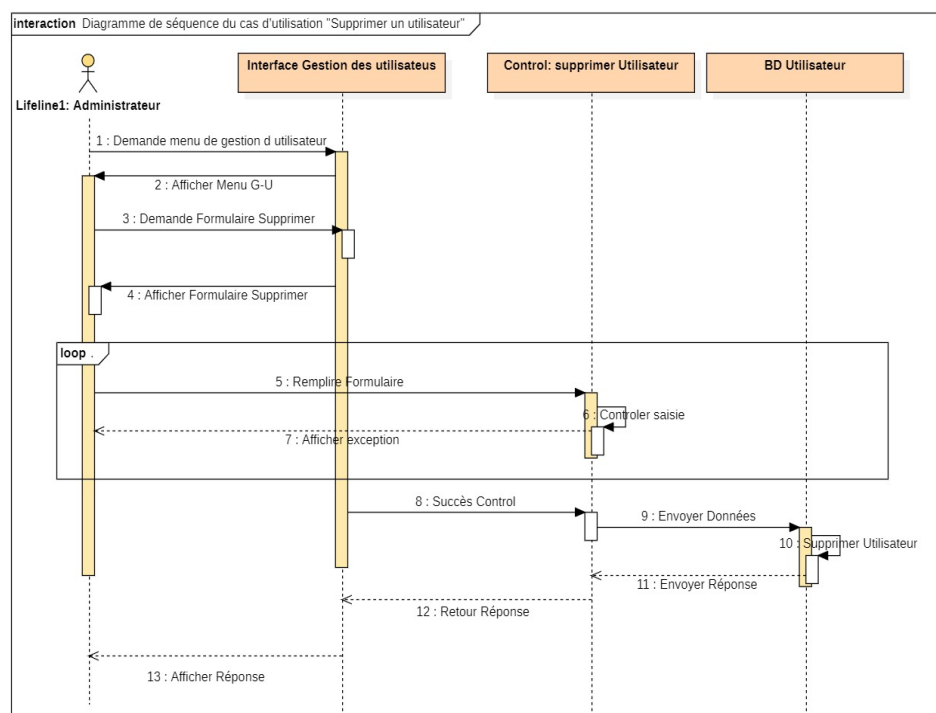


FIGURE 3.9 – Diagramme de séquence du cas d'utilisation "Supprimer un utilisateur" .

### 3.2.1.3.a Description détaillée des cas d'utilisations

- Cas d'utilisation "Supprimer un utilisateur" :
  - *Objectif* : Supprimer un utilisateur
  - *Acteur* : Administrateur
  - *Pré-condition* : Authentification préalable.  
Utilisateur existant.
  - *Post-conditions* : L'utilisateur a bien été supprimé.
  - *Scénario nominal* :
    1. L'administrateur choisit l'utilisateur à supprimer.
    2. Le système affiche un message de confirmation.
    3. L'administrateur valide son choix .
    4. Le système supprime l'utilisateur.
    5. Le système affiche un message de succès.
  - *Scénario alternative* :
    1. Le L'administrateur annule son choix.
    2. Le système annule la suppression.

### 3.2.1.4 Diagrammes de séquence du cas d' utilisation "Ajouter un utilisateur"

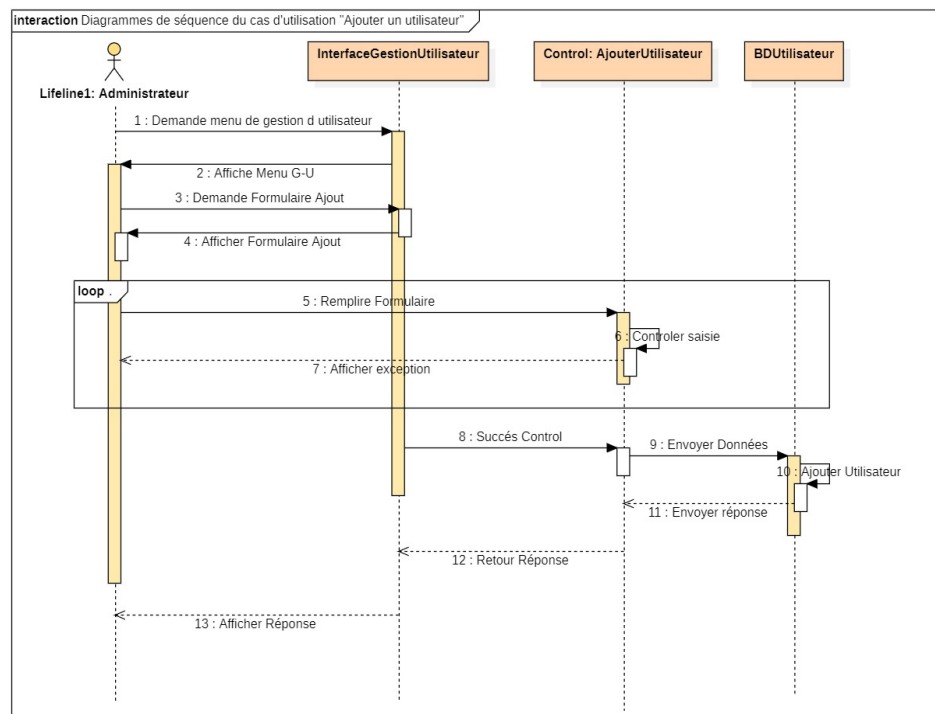


FIGURE 3.10 – Diagrammes de séquence du cas d'utilisation "Ajouter un utilisateur" .

### 3.2.1.4.a Description détaillée des cas d'utilisations

- Cas d'utilisation "Ajouter un utilisateur" :

- **Objectif** : Ajouter un utilisateur pour qu'il ait l'accès aux fonctionnalités de l'espace
- **Acteur** : Administrateur
- **Pré-condition** : Authentification préalable.  
Un formulaire d'ajout des utilisateurs est disponible.
- **Post-conditions** : Un nouvel utilisateur ajouté.
- **Scénario nominal** :
  1. L'administrateur demande un formulaire d'ajout d'un nouveau utilisateur.
  2. Le système affiche le formulaire d'ajout.
  3. L'administrateur doit remplir le formulaire avec l'ensemble des informations nécessaires à l'ajout du nouvel utilisateur.
  4. Le système vérifie les données saisies.
  5. L'administrateur valide .

6. Le système vérifie l'existence du nouveau compte .
7. Le système enregistre les informations saisies du l'utilisateur.

– *Scénario alternative :*

1. L'utilisateur saisit les informations manquantes ou erronées.
2. Le système affiche un ou des message d'erreurs selon les champs invalides.
3. Reprise de l'étape 3 du scénario nominal.
4. L'utilisateur existe déjà.
5. Le système informe l'administrateur que l'utilisateur existe déjà dans le système.
6. Reprise de l'étape 3 du scénario nominal.

### 3.2.1.5 Diagramme de séquence « d'ajout d'un professeur »

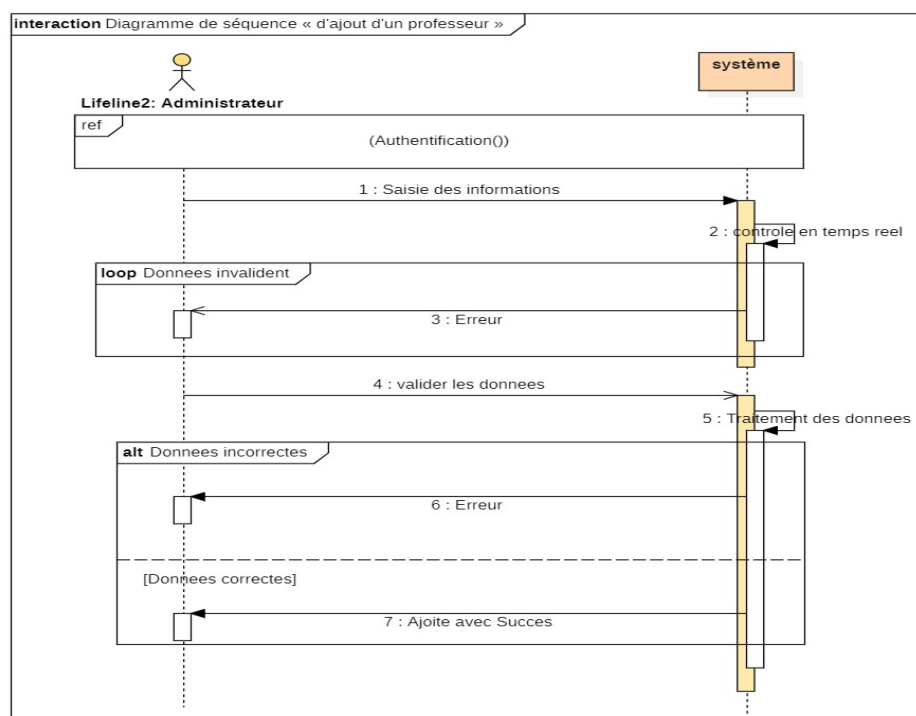


FIGURE 3.11 – Diagramme de séquence « d'ajout d'un professeur » .

### 3.2.1.5.a Description détaillée des cas d'utilisations

- Diagramme de séquence « d'ajout d'un professeur » :

- **Objectif** : Permettre à l'administration d'ajouter toutes les informations concernant le professeur, y compris les informations personnelles, et la possibilité d'ajouter les diplômes obtenus par le professeur .
- **Acteur** : Administrateur
- **Pré-condition** : Authentification .  
L'ajout d'un professeur doit répondre aux conditions de recrutement établies par la direction régionale.
- **Scénario** :
  1. Saisie des informations concernant le professeur.
  2. Contrôle des données en temps réel, en cas de duplication.
  3. Validation de la saisie.
  4. Traitement des informations envoyées.
  5. En cas d'une anomalie, l'ajout est rejeté en précisant l'erreur effectuée.
  6. Si non, l'ajout est effectué avec succès avec redirection d'utilisateur vers la liste des professeurs.

### 3.2.1.6 Diagramme d'activités « Gérer les utilisateurs »

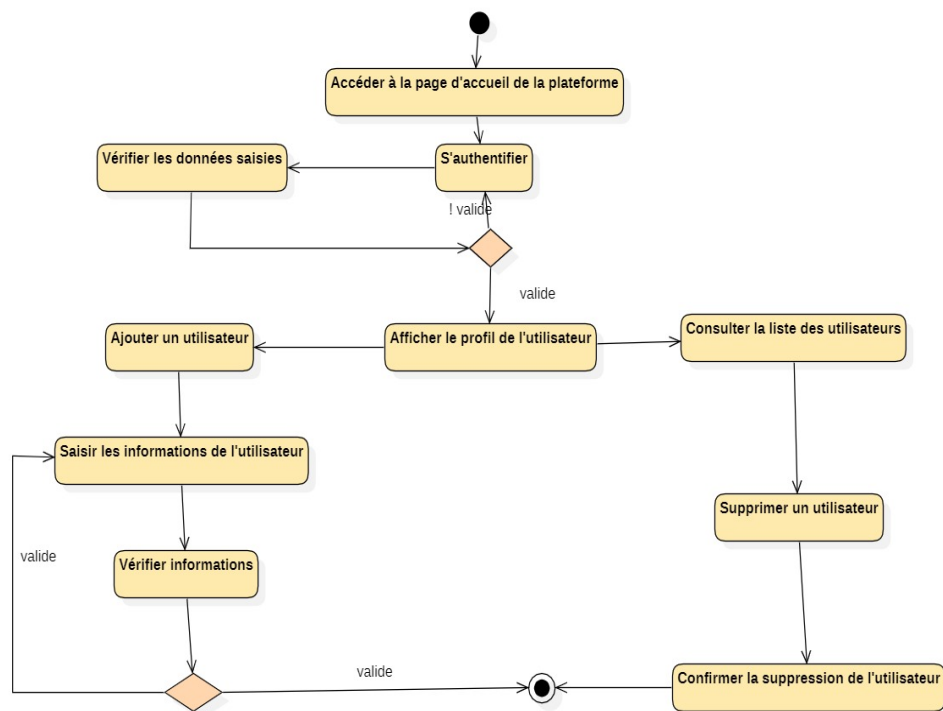


FIGURE 3.12 – Diagramme d'activités « Gérer les utilisateurs » .

### 3.2.1.6.a Description détaillée des cas d'utilisations

La figure ci-dessus illustre le déroulement séquentiel de la gestion des utilisateurs accomplis par un administrateur .

Après avoir s'authentifier, ces derniers peuvent ajouter ou supprimer un utilisateur.

Pour l'ajout d'un utilisateur, le système doit vérifier la validation des informations saisies. Au cas où une information n'est pas valide, le système réaffiche l'interface d'ajout d'un utilisateur.



### 3.2.2 item 2 : Inscription et réinscription d'un étudiant

#### 3.2.2.1 Diagramme de séquence « d'inscription d'un étudiant »

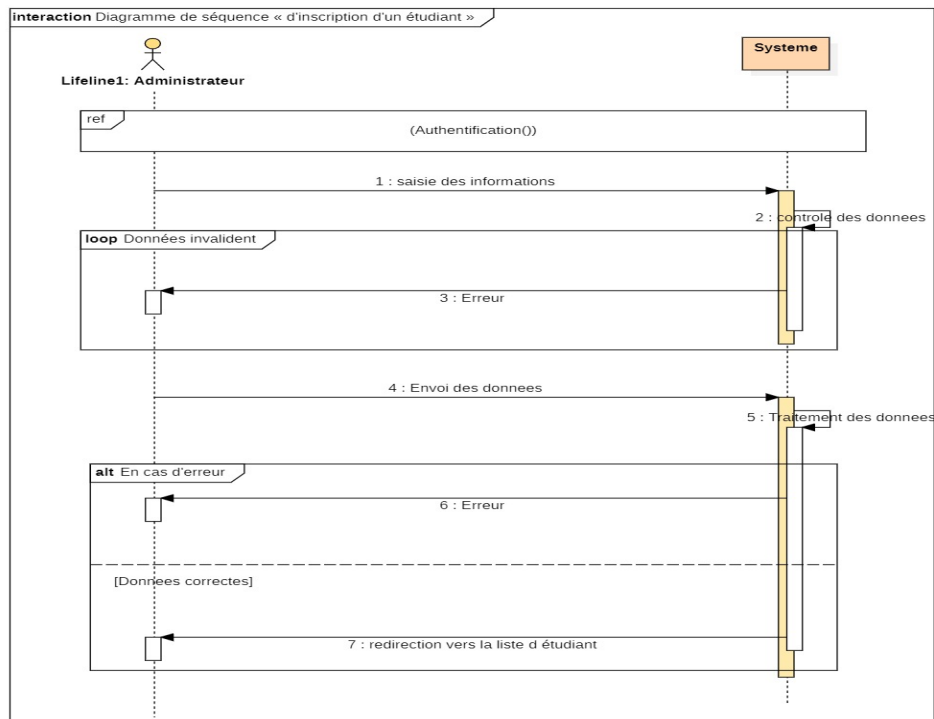


FIGURE 3.13 – Diagramme de séquence « d'inscription d'un étudiant » .

#### 3.2.2.1.a Description : Diagramme de séquence « d'inscription d'un étudiant »

- Diagramme de séquence « d'inscription d'un étudiant » :

- **Objectif** : Permettre à l'administration d'ajouter toutes les informations concernant un étudiant, y compris l'état civil, et les informations complémentaires .
- **Acteur** : Administrateur
- **Pré-condition** : Authentification .  
L'inscription d'un étudiant doit répondre aux conditions d'inscription établie par la direction régionale
- **Scénario** :
  1. Saisie les informations de l'élève.

2. Contrôle des données en temps réel (matricule – cne – cin) en cas de duplication.
3. Validation de la saisie.
4. Traitement des informations envoyées.
5. En cas d'une anomalie, l'inscription est rejetée on précisant l'erreur effectuée.
6. Si non, l'inscription est effectuée avec succès avec redirection d'utilisateur vers la liste d'élèves.

### 3.2.2.2 Diagramme d'activités « d'inscription d'un étudiant »

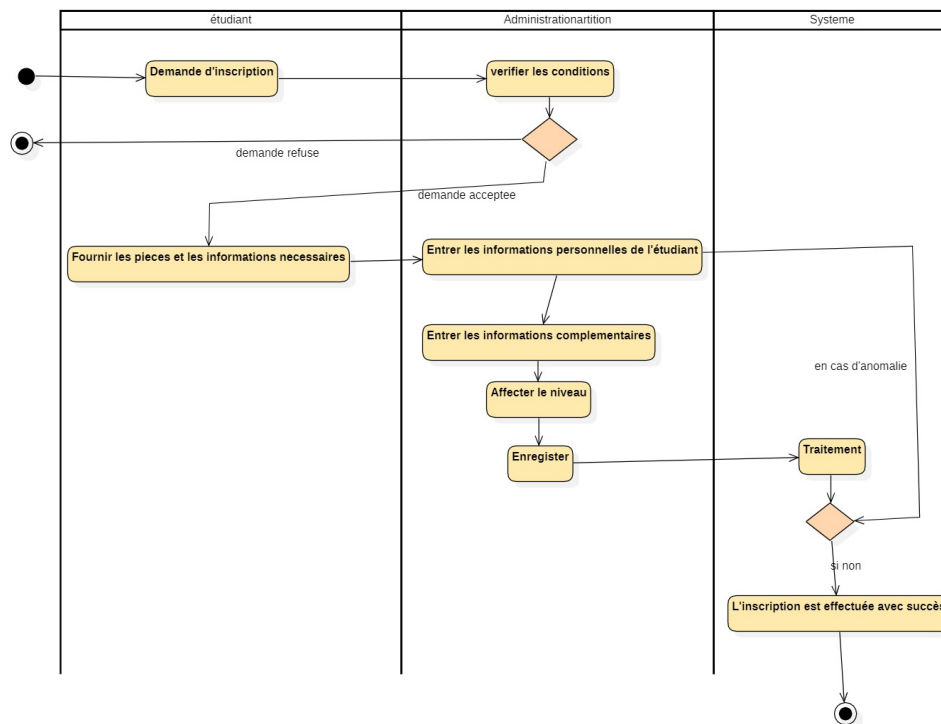


FIGURE 3.14 – Diagramme d'activités « d'inscription d'un étudiant » .

#### 3.2.2.2.a Description du processus de diagramme d'activités «Inscription d'un élève» :

- ★ L'élève demande l'inscription dans un niveau.

- ★ L'administration vérifie les conditions d'inscriptions pour l'élève.
- ★ Si l'élève ne répond pas aux conditions de l'établissement, donc la demande est refusée.
- ★ Si non, l'élève doit fournir les pièces et les informations nécessaires pour l'inscription.
- ★ L'administration donne les informations personnelles de l'élève.
- ★ L'administration introduit les informations complémentaires et celles concernant la santé de l'élève.
- ★ L'administration affecte le niveau et valide l'inscription.
- ★ Le système traite les informations envoyées.
- ★ En cas d'une anomalie, le système refuse l'inscription demandant à l'administration de vérifier l'anomalie.
- ★ Si non, l'inscription est effectuée avec succès.

### **3.2.2.3 Diagramme de collaboration «réinscription d' un étudiant»**

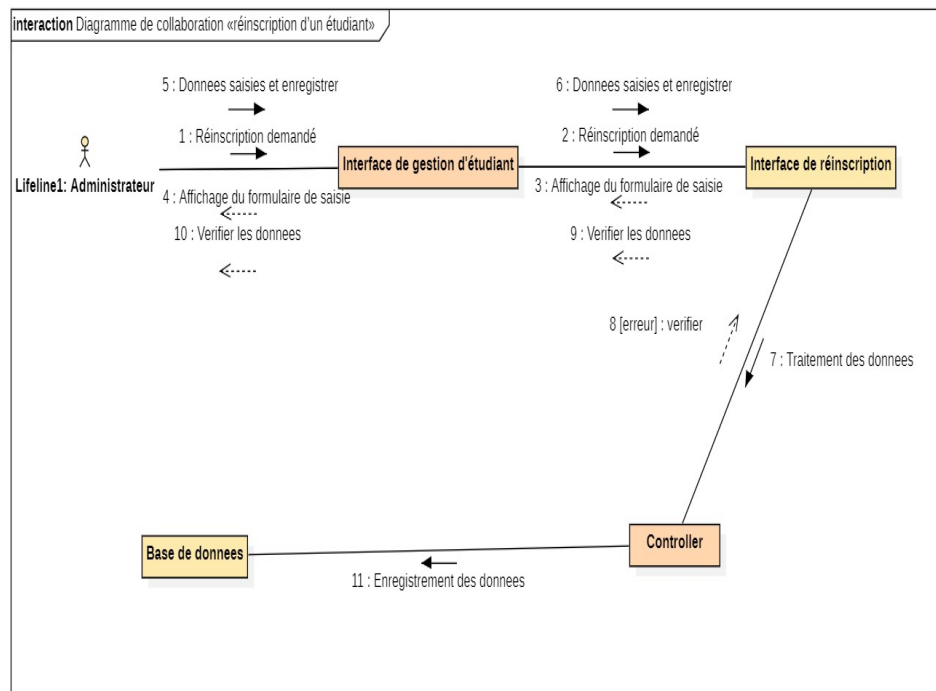


FIGURE 3.15 – Diagramme de collaboration «réinscription d' un étudiant» .

### 3.2.2.3.a Description : Diagramme de collaboration «réinscription d' un étudiant»

Un diagramme de collaboration est un diagramme d'interactions, représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets, et où la chronologie n'intervient que de façon annexe.

Cela consiste en un graphe dont les nœuds sont des objets et les arcs (numérotés selon la chronologie) et les échanges entre ces objets.

### 3.2.2.4 Diagramme d' activités « d' affectation des notes étudiant »

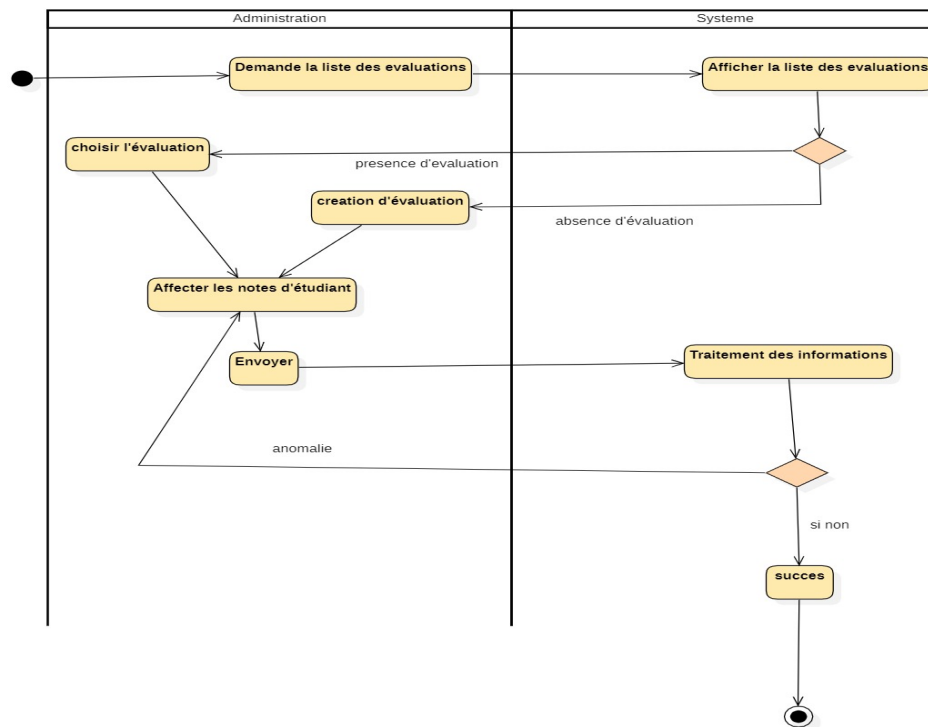


FIGURE 3.16 – Diagramme d'activités « d'affectation des notes étudiant » .

### 3.2.2.4.a Description du processus de diagramme d'activités «Affecter les notes d'élèves» :

- ★ L'administration demande la liste des évaluations.
- ★ Le système affiche la liste des évaluations.
- ★ En cas d'absence d'évaluation, l'administration doit créer l'évaluation.
- ★ Si non, l'administration choisit l'évaluation et affecte les notes pour chaque élève.
- ★ Puis envoyer les notes pour les sauvegarder.
- ★ Le système traite les informations envoyées.
- ★ En cas, d'une anomalie l'ajout est annulé.
- ★ Si non, l'ajout est effectué avec succès.

### 3.3 Sprint 3 : Enseignants

“ Ce premier sprint s'étale sur 18 jours et se décompose en deux items ”

- ◇ Gérer les cours
- ◇ Gérer les tests

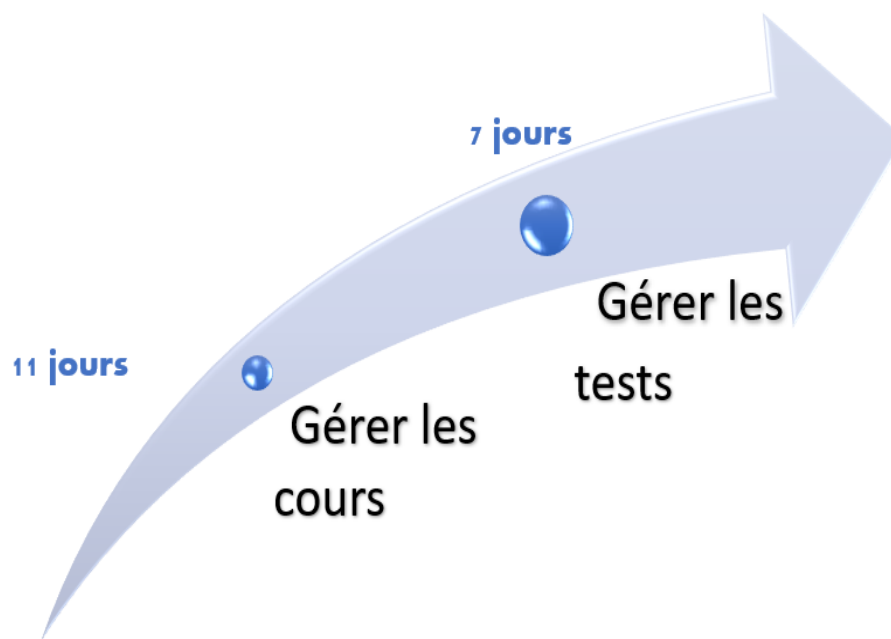


FIGURE 3.17 – Décomposition sprint 3 en Items.

Le backlog du sprint 3 est le suivant :

Item	User Story	Description	Priorité
<b>s'inscrire</b>	s'inscrire	En tant qu'utilisateur, je peux m'inscrire	1
<b>s'authentifier</b>	s'authentifier	En tant qu'utilisateur, je dois m'identifier pour accéder à mon espace	2
<b>gérer profil</b>	Consulter profil	En tant qu'utilisateur je peux consulter mon profil	3
	Modifier profil	En tant qu'utilisateur je peux modifier mon profil	
	Modifier image de profil	En tant qu'utilisateur je peux uploader modifier une image de profil	
	Désactiver profil	En tant qu'utilisateur je peux désactiver mon profil	
<b>s'inscrire</b>	s'inscrire	En tant qu'utilisateur, je peux m'inscrire	1

TABLE 3.5 – Tables Backlog du sprint 3

les user stories de sprint 3 :

ID U.S	User Story
1	En tant qu'utilisateur, je dois m'authentifier pour accéder à mon espace En tant qu'utilisateur, je m'authentifier pour accéder à mon espace entifier accéder
7	En tant qu'utilisateur, je dois m'authentifier pour accéder à mon espace

TABLE 3.6 – Tables "les user stories de sprint 3"





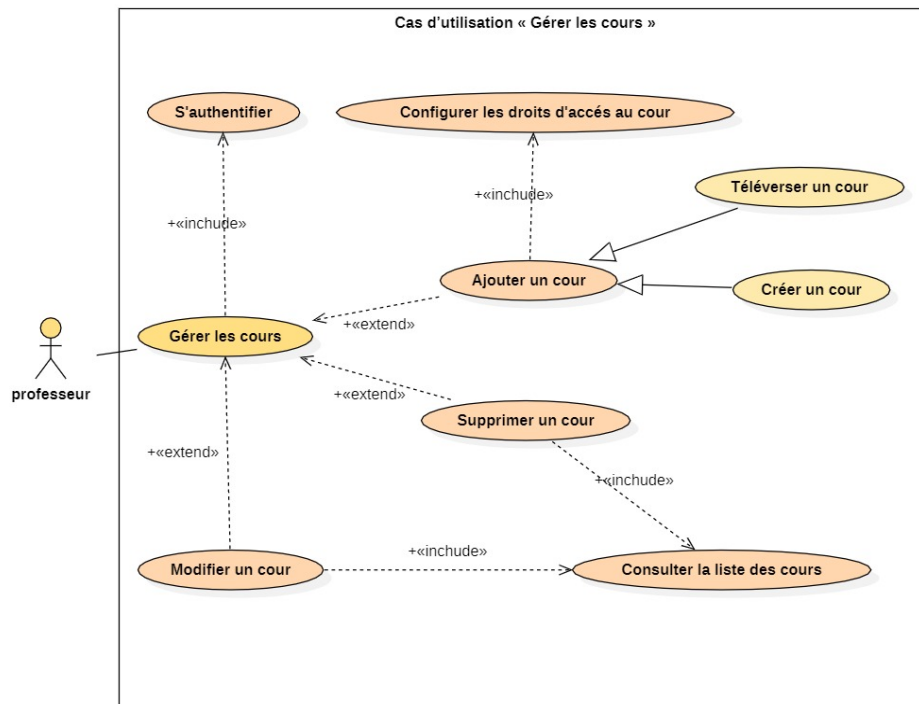


FIGURE 3.19 – Diagramme des cas d' utilisation « Gérer les cours » .

### 3.3.1.2.a Description détaillée des cas d'utilisations

- Cas d'utilisation Gérer les cours :

- **Objectif** : permet à l'acteur d'ajouter, d'annuler et modifier un cours.
- **Acteur** : Enseignants
- **Pré-condition** : L'acteur doit être connecté.
- **Scénario nominal** :
  1. Le système affiche deux méthodes d'ajout d'un cours.  
<méthode1 : Créer un cours>
  2. L'acteur saisit le contenu du cours  
<méthode2 : téléverser un cours>
    1. L'acteur téléverse le cours.
    2. L'acteur configure les droits d'accès à son cour.
    3. L'acteur enregistre le cours dans la base de données de la plateforme.

## 3.3.1.3 diagramme de séquences

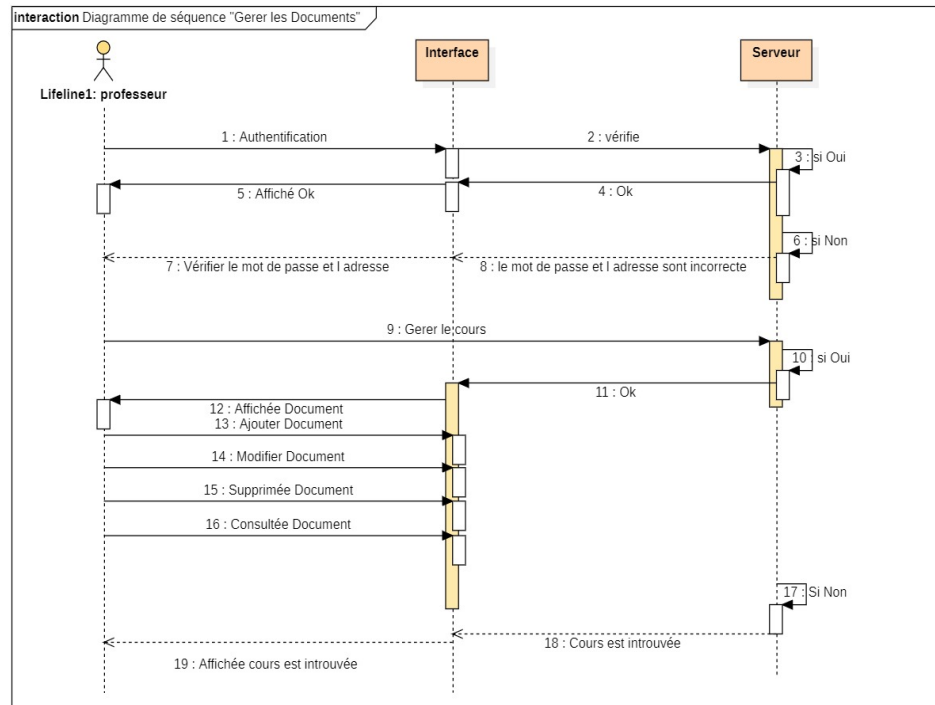


FIGURE 3.20 – Diagramme de séquence "Gerer les cours" .

## 3.3.1.3.a Description :Diagramme de séquence "Gerer les cours"

Après introduire le login et mot de passe l'enseignant peut gérer le cours (Modifier, Ajouter, Supprimer) Cours, sinon il affiche un message "cours est introuvée"

## 3.3.1.4 diagramme de séquences

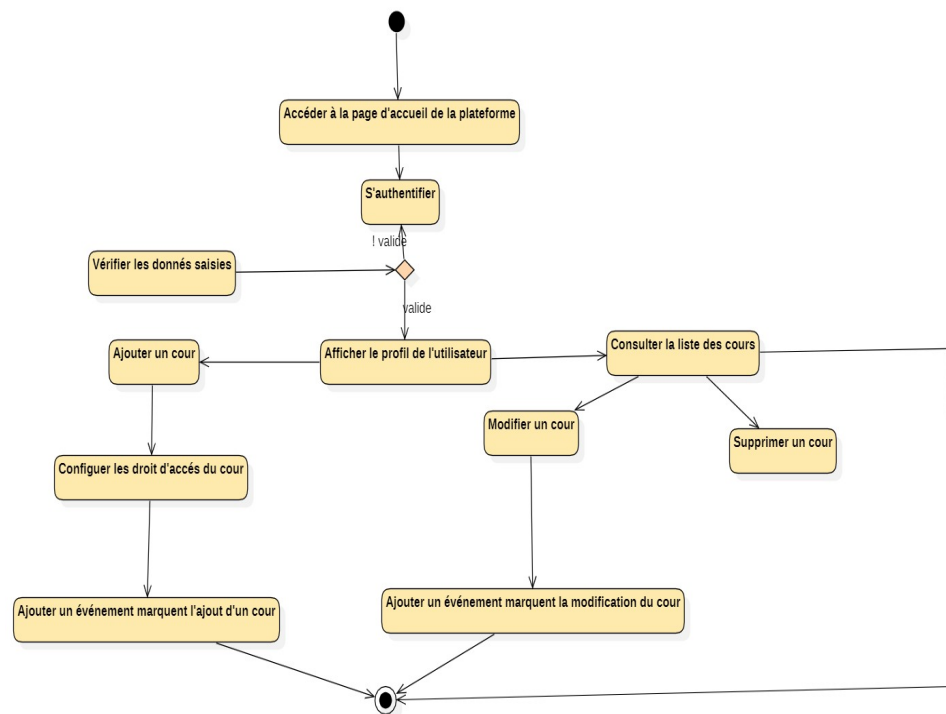


FIGURE 3.21 – Diagramme d'activités « Gérer les cours » .

### 3.3.1.4.a Description détaillée des cas d'utilisations

La figure ci-dessus illustre le déroulement séquentiel de la gestion des cours accomplis par un Enseignant.

Après avoir s'authentié, un Enseignant peut ajouter, modifier ou supprimer un cours.

Au cas d'ajout ou de modification du cours, le tuteur doit ajouter cet évènement partagé pour informer les apprenants du changement.

### 3.3.1.5 diagramme de séquences

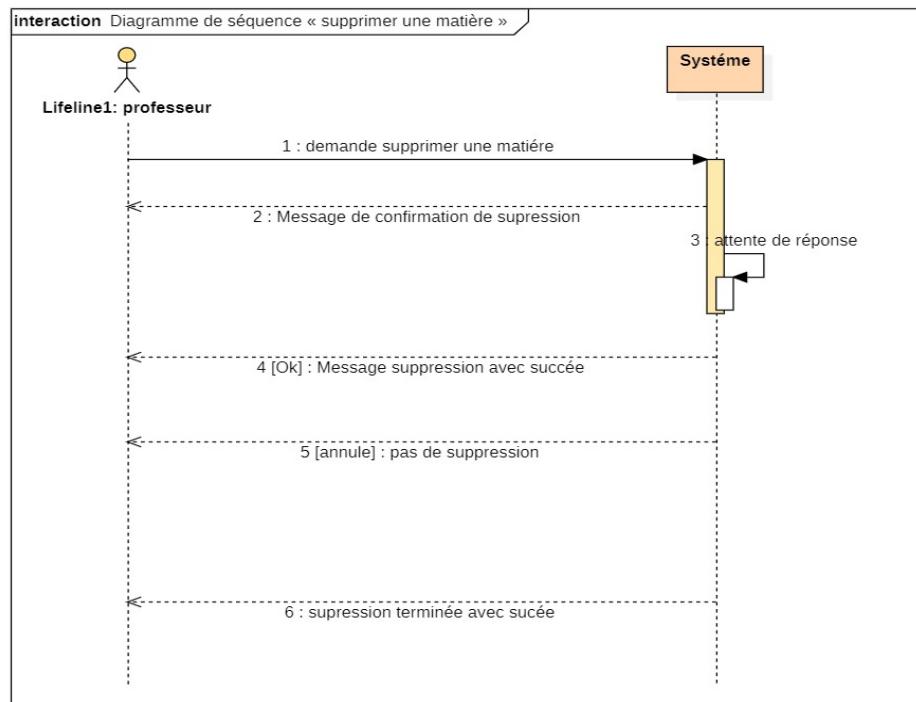


FIGURE 3.22 – Diagramme de séquence « supprimer une matière » .

#### 3.3.1.5.a Description détaillée des cas d'utilisations

## 3.3.2 item 2 : Gérer les tests

## 3.3.2.1 diagramme de séquences

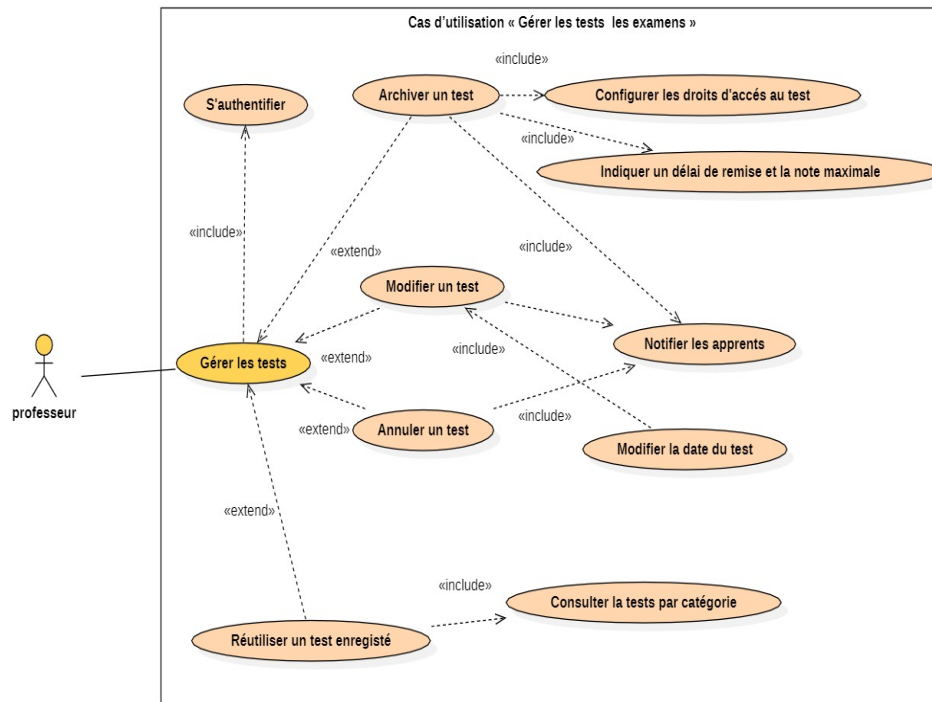


FIGURE 3.23 – Diagramme de Cas d' utilisation « Gérer les tests / les examens » .

## 3.3.2.1.a Description détaillée des cas d'utilisations

- Cas d'utilisation Gérer les utilisateurs :

- **Objectif** : permet à l'acteur d'ajouter, d'annuler et modifier un test.
- **Acteur** : Enseignants
- **Pré-condition** : L'acteur doit être connecté.
- **Post-conditions** :
- **Scénario nominal** :
  1. L'acteur saisit les questions en indiquant la vraie réponse et en précisant un barème.
  2. L'acteur configure les droits d'accès au test.
  3. L'acteur enregistre le test dans la base de données .

4. L'acteur ajoute le test au calendrier partagée et notifie les apprenants de l'ajout du test .

### 3.3.2.2 diagramme de séquences

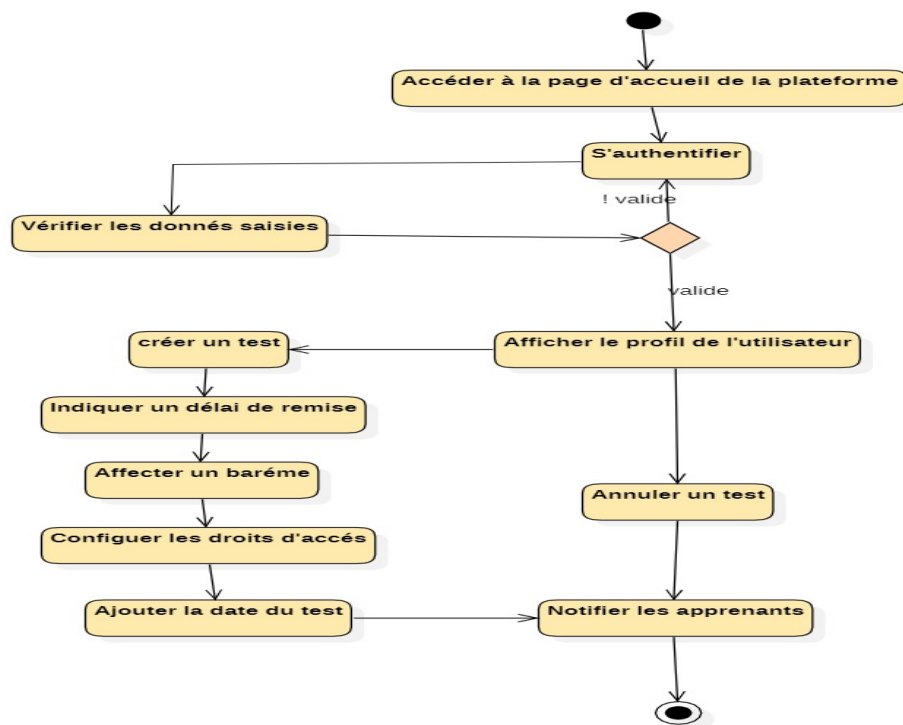


FIGURE 3.24 – Diagramme d'activités « Gérer les tests / les examens » .

#### 3.3.2.2.a Description détaillée des cas d'utilisations

La figure ci-dessus illustre le déroulement séquentiel de la gestion des devoirs accomplis par un Enseignants.

Après l'authentification, un Enseignants peut ajouter ou annuler un devoir. Au cas d'ajout, il faut lui indiquer un délai de remise, lui affecter un barème et lui configurer les droits d'accès.

Finalement, et c'est le cas d'annulation aussi, le tuteur doit ajouter l'évènement partagé pour informer l'apprenant du changement .

## 3.4 Sprint 4 : Étudiant

“ Ce premier sprint s’étale sur 21 jours et se décompose en trois items ”

- ◇ Consulter les ressources et les liens d'apprentissage
- ◇ Consulter agenda des livrables
- ◇ Passer les examens

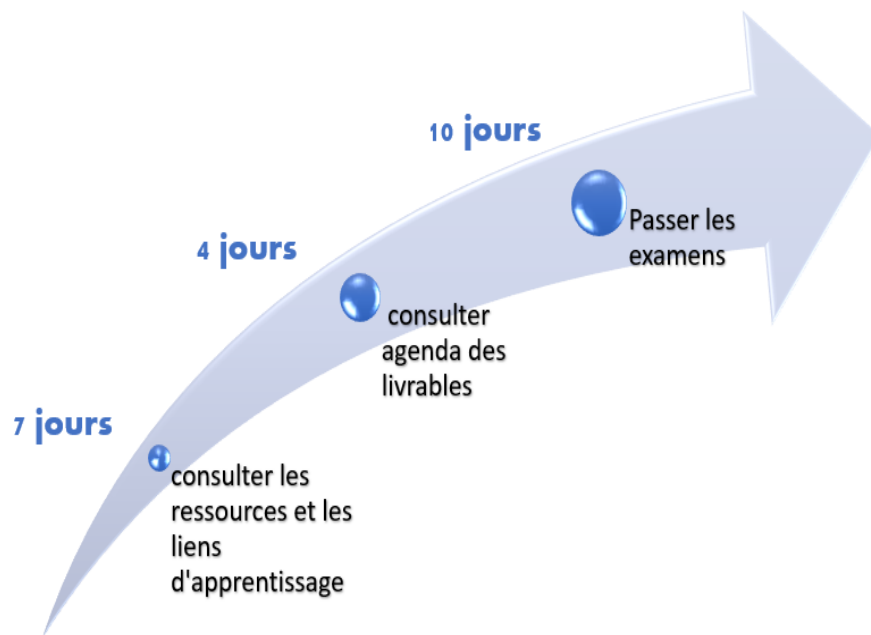


FIGURE 3.25 – Décomposition sprint 4 en Items.

Le backlog du sprint 4 est le suivant :

Item	User Story	Description	Priorité
<b>s'inscrire</b>	s'inscrire	En tant qu'utilisateur, je peux m'inscrire	1
<b>s'authentifier</b>	s'authentifier	En tant qu'utilisateur, je dois m'identifier pour accéder à mon espace	2
<b>gérer profil</b>	Consulter profil	En tant qu'utilisateur je peux consulter mon profil	3
	Modifier profil	En tant qu'utilisateur je peux modifier mon profil	
	Modifier image de profil	En tant qu'utilisateur je peux uploader modifier une image de profil	
	Désactiver profil	En tant qu'utilisateur je peux désactiver mon profil	
<b>s'inscrire</b>	s'inscrire	En tant qu'utilisateur, je peux m'inscrire	1

TABLE 3.7 – Tables Backlog du sprint 4

les user stories de sprint 4 :

ID U.S	User Story
1	En tant qu'utilisateur, je dois m'authentifier pour accéder à mon espace En tant qu'utilisateur, je m'authentifier pour accéder à mon espace entifier accéder
7	En tant qu'utilisateur, je dois m'authentifier pour accéder à mon espace

TABLE 3.8 – Tables "les user stories de sprint 4"



## 3.4.1 item 1 : Consulter les ressources et les liens d'apprentissage

## 3.4.1.1 Diagramme de cas d'utilisation détaillé «administrer du site de télé-formation»

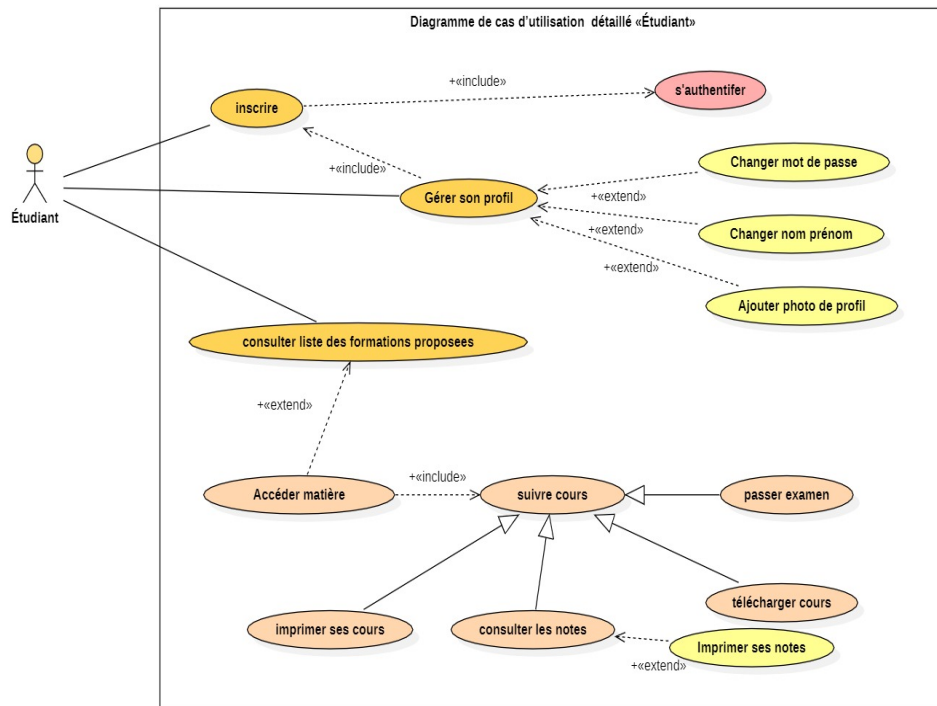


FIGURE 3.26 – Diagramme de cas d' utilisation détaillé «Étudiant» .

## 3.4.1.1.a Description détaillée des cas d'utilisations

## 3.4.1.2 diagramme de séquences

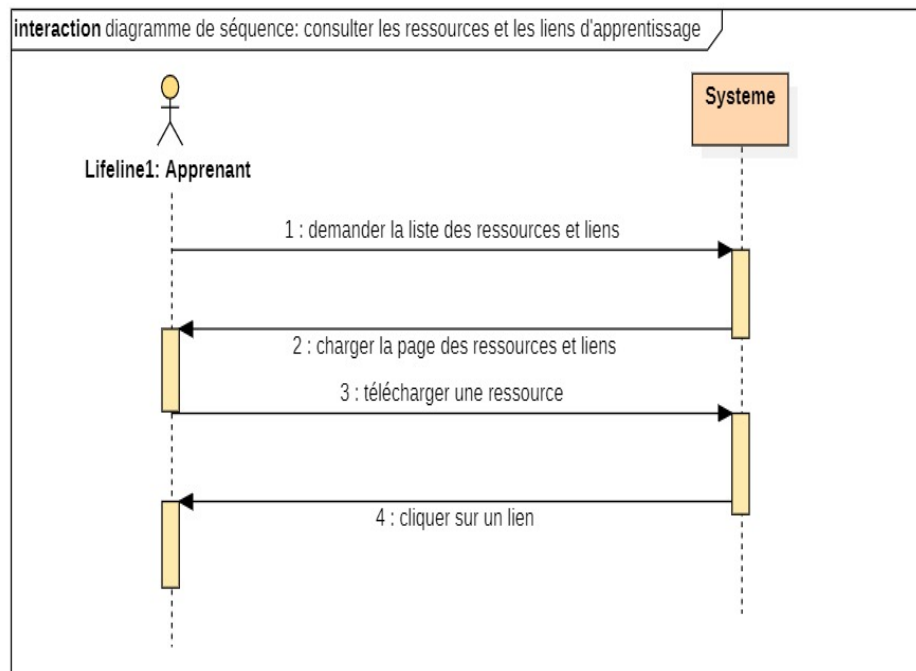


FIGURE 3.27 – Diagramme de séquence : consulter les ressources et les liens d'apprentissage

#### 3.4.1.2.a Description détaillée des cas d'utilisations

### 3.4.2 item 2 : Consulter agenda des livrables

#### 3.4.2.1 diagramme de séquences

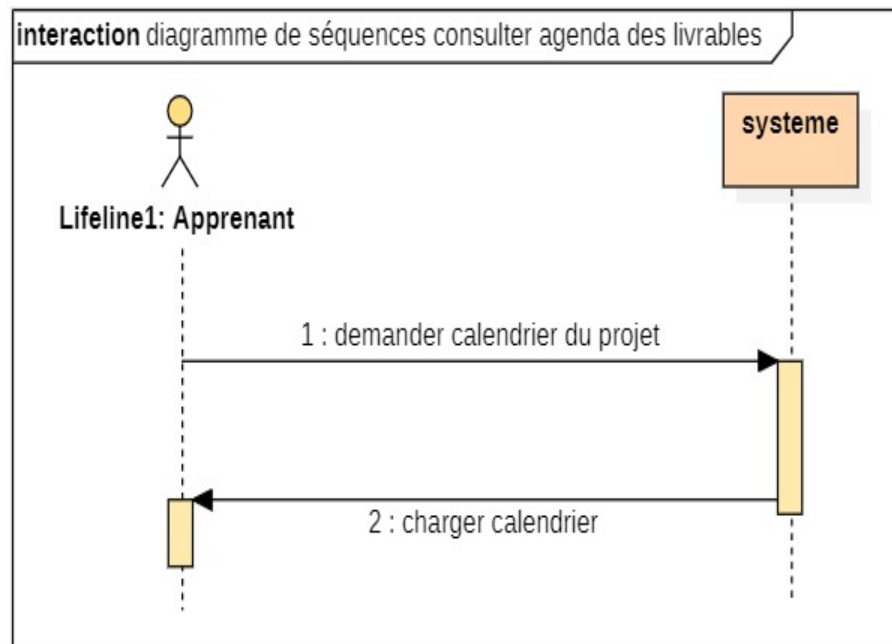


FIGURE 3.28 – Diagramme de séquences consulter agenda des livrables .

#### 3.4.2.1.a Description : Diagramme de séquences consulter agenda des livrables

Ce diagramme de séquence illustre l'interaction entre l'apprenant et le système afin de consulter le calendrier des livrables d'un projet dont il est participant.

### 3.4.3 item 3 : Passer les examens

#### 3.4.3.1 Diagramme de cas d'utilisation

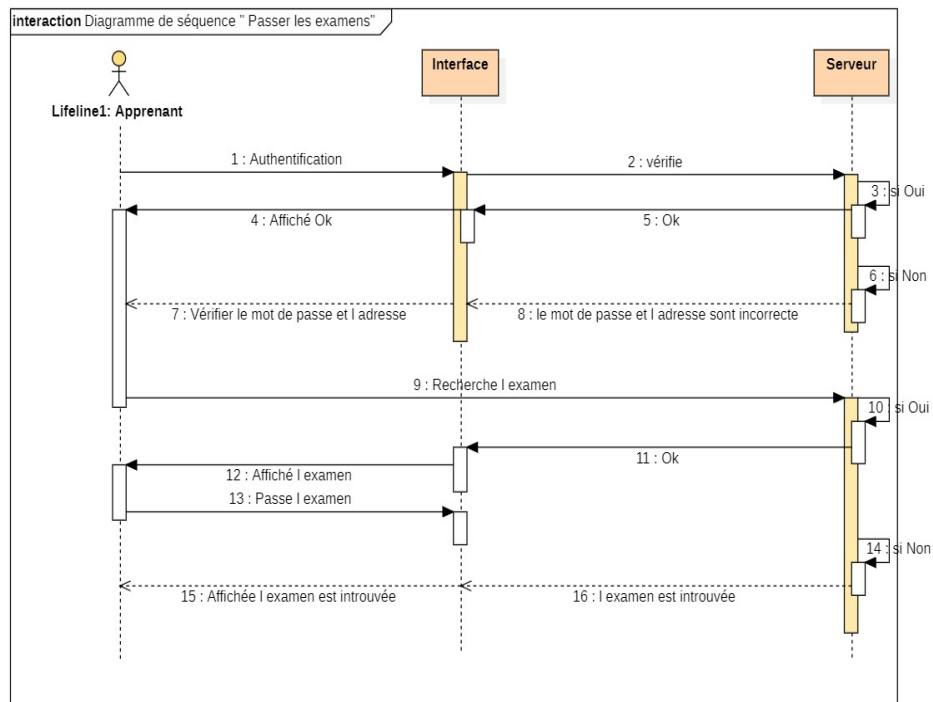


FIGURE 3.29 – Diagramme de séquence " Passer les examens" .

#### 3.4.3.1.a Description :Diagramme de séquence " Passer les examens"

l'étudiant peut passer les examens en une durée du temps déterminé , sinon il affiche un message "examen est introuvée"

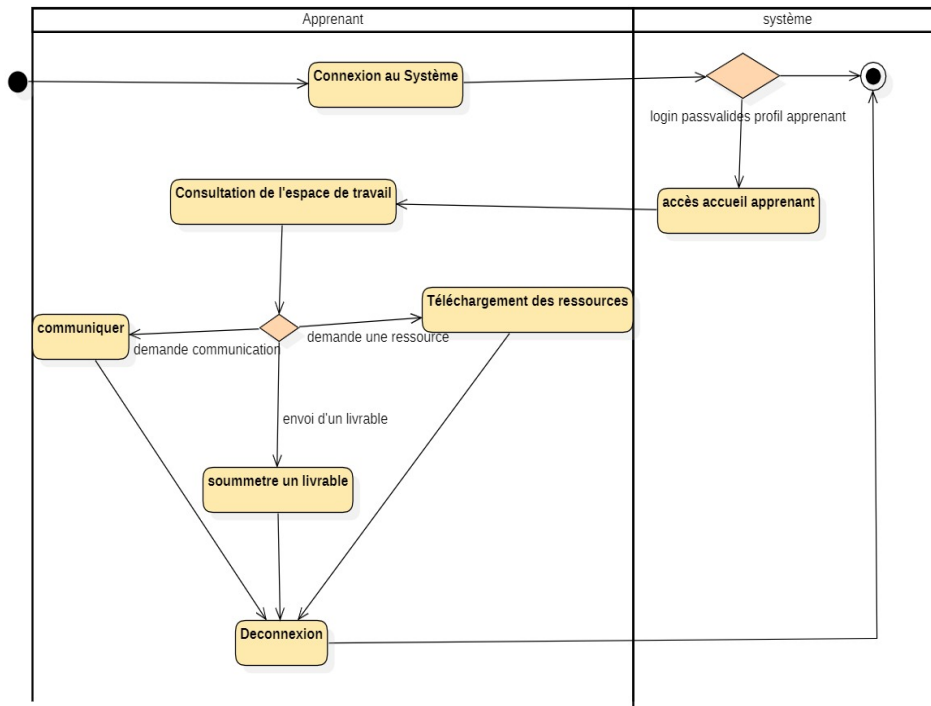


FIGURE 3.30 – Diagramme d'activité : d'un apprenant sur l'espace .

#### 3.4.3.1.b Description détaillée des cas d'utilisations

Ce diagramme d'activités décrit les activités d'un apprenant sur l'espace destiné aux projets afin de garantir l'apprentissage par projet.

## 3.5 Conclusion

Ce chapitre présente une vue conceptuelle de la solution à mettre en place. Il expose les différents diagrammes UML pour mieux comprendre les fonctionnalités offertes et pour mieux représenter la communication entre les différents objets du projet. Le chapitre suivant, présente la partie mise en œuvre de l'application.

# Chapitre 4

## Mise en oeuvre de la solution

“ Dans ce chapitre, on va présenter les outils utilisés pour la mise en oeuvre de la solution PIM ainsi qu’un aperçu des différentes vues et interfaces de cette solution. ”

## 4.1 Introduction :

Une des étapes de la vie d'un projet, aussi importante que la conception, est l'implémentation.

Cette étape constitue la phase d'achèvement et d'aboutissement du projet. Pour accomplir cette tâche avec succès il faut savoir utiliser les outils adéquats et nécessaires. Ce choix d'outils peut influencer la qualité du produit obtenu et donc nécessite une attention particulière et doit se baser sur les besoins du projet et le résultat escomptés.

Ce chapitre présente l'environnement technique du travail ainsi que le choix pris en matière d'environnement logiciel.

## 4.2 Environnement de travail : Outils et technologies utilisées

### 4.2.1 Environnement matériel

Pour la réalisation de ce projet nous avons utilisé le matériel suivant :

Pc portable	Asus VivoBook 15 x510UA	HP Notebook 15
RAM	8 GO	8 GO
Microprocesseur	Intel(R)Core(TM)i5-8250U	Intel(R)Core(TM)i3-45005U
Disque dur	240 GB SSD	240 GB SSD
System d'exploitation	Windows 10 Profissional	Windows 10 Profissional

TABLE 4.1 – Description de la machine de développement

### 4.2.2 Environnement logiciel

#### 4.2.2.1 Outils de développement et modélisation :

Les principaux outils qui ont contribué à la qualité du développement sont :

- **Visual Studio Code** :<sup>1</sup>est un éditeur de code multiplateforme édité par Microsoft. Ce outil destiné aux développeurs supporte plusieurs dizaines de langages de programmation comme le HTML, C++, PHP, Javascript, Markdown, CSS, etc. [3]

---

1. Visual Studio Code : est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS1.



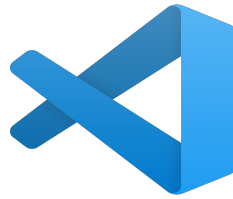


FIGURE 4.1 – Logo Visual code studio.

- **StarUML : Outils de modélisation** est un logiciel de modélisation UML (Unified Modeling Language) open source qui peut remplacer dans bien des situations des logiciels commerciaux et coûteux comme Rational Rose<sup>2</sup> ou Together<sup>3</sup>. Étant simple d'utilisation, nécessitant peu de ressources système, supportant UML 2, ce logiciel constitue une excellente option pour une familiarisation à la modélisation. Cependant, seule une version Windows est disponible. [4]



FIGURE 4.2 – Logo Visual code studio.

- **XAMPP** est un ensemble de logiciels permettant de mettre en place facilement un serveur Web confidentiel, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus. [5]

---

2. Présentation sur le site web d'IBM France : <http://ibm.co/1hre0xd>

3. Présentation en anglais sur le site web de Borland : <http://www.borland.com/products/Together/>



FIGURE 4.3 – Logo Visual code studio.

- **Apache Tomcat** <sup>4</sup> : C'est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta. Il incorpore également un serveur HTTP[6].  
c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process, et paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion.



FIGURE 4.4 – Logo Visual code studio.

#### 4.2.2.2 Langages de programmation :

- **HTML 5 :**

HTML signifie « HyperText Markup Language » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure. D'autres technologies sont utilisées avec HTML pour décrire

---

4. <http://tomcat.apache.org/>

la présentation d'une page (CSS) et/ou ses fonctionnalités interactives (JavaScript). HTML fonctionne grâce à des « balises » qui sont insérées au sein d'un texte normal. Chacune de ces balises indique la signification de telle ou telle portion de texte dans le site. On parle d'« hypertexte » en référence aux liens qui connectent les pages web entre elles. C'est la mécanique originelle du « World Wide Web » que nous connaissons aujourd'hui. En écrivant et publiant des pages web, vous devenez un acteur du Web dès que votre site est accessible en ligne.[7]



FIGURE 4.5 – Logo HTML.

### 4.2.3 Outils de développement et de collaboration :

#### 4.2.3.1 Le système de contrôle des versions GIT :

C'est un logiciel libre de gestion de versions, créé par Linus Torvalds (le créateur du noyau Linux), c'est un outil bas niveau, qui se veut simple et très performant, dont la principale tâche est de gérer l'évolution du contenu d'une arborescence. Il fonctionne en mode distribué avec un serveur distant[8].

## 4.3 Conclusion

Dans ce chapitre, on a présenté la réalisation qu'on a effectué, l'environnement de développement, les outils et technologies utilisés dans le projet ainsi qu'une description détaillée des différentes interfaces utilisateur de la plate-forme.

# Webographie

- [1] Wikipédia, “Sqli — wikipédia, l’encyclopédie libre.” <http://fr.wikipedia.org/w/index.php?title=SQLI&oldid=158242840>, 2019. [Consulté le 7 avril 2019].
- [2] Wikipedia, “Diagramme de classes — Wikipedia, the free encyclopedia.” <http://fr.wikipedia.org/w/index.php?title=Diagramme%20de%20classes&oldid=169750234>, 2020. [Online ; accessed 16-May-2020].
- [3] Wikipedia, “Visual Studio Code — Wikipedia, the free encyclopedia.” <http://fr.wikipedia.org/w/index.php?title=Visual%20Studio%20Code&oldid=171121594>, 2020. [Online ; accessed 23-May-2020].
- [4] Wikipedia, “StarUML — Wikipedia, the free encyclopedia.” <http://fr.wikipedia.org/w/index.php?title=StarUML&oldid=170240822>, 2020. [Online ; accessed 25-May-2020].
- [5] Wikipedia, “XAMPP — Wikipedia, the free encyclopedia.” <http://fr.wikipedia.org/w/index.php?title=XAMPP&oldid=170566221>, 2020. [Online ; accessed 25-May-2020].
- [6] Wikipédia, “Apache tomcat — wikipédia, l’encyclopédie libre.” [http://fr.wikipedia.org/w/index.php?title=Apache\\_Tomcat&oldid=159469213](http://fr.wikipedia.org/w/index.php?title=Apache_Tomcat&oldid=159469213), 2019. [Consulté le le 8 juin 2019].
- [7] Wikipedia, “Hypertext Markup Language — Wikipedia, the free encyclopedia.” <http://de.wikipedia.org/w/index.php?title=Hypertext%20Markup%20Language&oldid=199154733>, 2020. [Online ; accessed 26-May-2020].
- [8] Wikipédia, “Git — wikipédia, l’encyclopédie libre.” <http://fr.wikipedia.org/w/index.php?title=Git&oldid=160009701>, 2019. [Consulté le le 8 juin 2019].