



République Tunisienne

\*\*\*\*\*

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

\*\*\*\*\*

Université de Monastir

\*\*\*\*\*

Institut Supérieur d'Informatique et de Mathématiques de Monastir

\*\*\*\*\*

Département Informatique

N° d'ordre: LA 16



# Mémoire de Projet de Fin d'Etudes

*Présenté en vue de l'obtention du*

**Diplôme National de Licence Appliquée en Sciences  
Informatiques**

*Spécialité :*

**Systèmes Informatiques et Logiciels**

*par*

*Dghais saif eddin*

*Ben saadala*

---

---

**Conception et développement d'une plateforme  
éducative interactive**

---

---

*Soutenu le 02/07/2021 devant le jury composé de :*

M./Mme : Azzaz Skander

Président

M./Mme : Ben Abid Wided

Rapporteur

M./Mme : Mallat Souheyl

Encadrant Pédagogique

M./Mme : Bousbia Salah

Encadrant Professionnel

# Résumé

---

---

Ce travail s'inscrit dans le cadre du projet de fin d'études pour l'obtention de la licence appliquée en informatique de l'Institut Supérieur d'Informatique et de Mathématiques de Monastir (ISIMM) où nous avons eu cette opportunité de préparer notre projet appelé « KidsLab » qui consiste à développer un plateforme éducative interactive.

Ce projet marque la fin du cursus universitaire et il sert surtout à prendre pied dans le monde de l'entreprise. En effet, ces quatre mois de stage en moyenne servent de transition entre la vie étudiante et la vie professionnelle. C'est pourquoi il est important de bien le choisir !

Le stage de fin d'études a pour but de développer chez le stagiaire les compétences et le savoir-être indispensables à son avenir professionnel. Sous la supervision d'un maître de stage, par exemple un enseignant de l'établissement.

---

---

**Mots clés :** *Node js – React js – Mongo DB – Express js – E-learning – KidsLab*

# Abstract

---

---

This work is part of the end of studies project to obtain the applied license in computer science from the Higher Institute of Computer Science and Mathematics of Monastir (ISIMM) where we had this opportunity to prepare our project called “KidsLab” which consists of developing an interactive educational platform.

This project marks the end of the university course and it serves above all to gain a foothold in the business world. Indeed, these four months of internship on average serve as a transition between student and professional life. This is why it is important to choose it well!

The purpose of the end of studies internship is to develop in the intern the skills and interpersonal skills essential to his professional future. Under the supervision of a training supervisor, for example a teacher from the establishment.

---

---

# REMERCIEMENTS

Nous tenons à remercier toutes les personnes qui ont contribué à la réussite de notre stage et qui nous ont aidé à rédiger ce rapport.

Tout d'abord, nous tenons à remercier notre professeur, M. Souhayl Mallat, qui a chaleureusement accepté de superviser notre projet, pour son accueil, le temps passé ensemble et le partage de son expertise chaque fois que nous avions besoin de lui. Aussi grâce à sa confiance nous avons pu nous épanouir pleinement dans nos missions. Il a été d'une grande aide dans les moments les plus délicats.

Nous tenons également à remercier Monsieur BOUSBIA Salah notre metteur en scène pour ses encouragements, ses soins et ses conseils tout au long de la période du projet, et tous ceux qui ont participé à la réalisation de ce projet dans de bonnes conditions.

Aussi toutes nos gratitude et meilleures salutations aux jurys pour l'évaluation de notre travail.

Enfin, nous remercions toutes les personnes qui nous ont conseillé et relu lors de la rédaction de ce rapport de stage : nos familles et amis.

# TABLE DES MATIÈRES

LIST DES FIGURES .....	
LIST DES TABLEAUX.....	
INTRODUCTION GÉNÉRALE.....	1
CHAPITRE 1 : CONTEXTE GÉNÉRALE ET ÉTUDE PRÉALABLE .....	4
Introduction.....	5
I.    Présentation de projet .....	5
1.    Cadre du projet .....	5
2.    Organisme d'accueil : « BMCAF Consulting » .....	5
3.    Problématique et motivation.....	6
II.   Concept de base de E-learning .....	6
1.    E-Learning .....	6
Learning (Apprentissage mixte).....	7
3.    Le social Learning (L'apprentissage social) .....	7
4.    Récapitulation .....	8
III.  Étude de l'existant.....	8
1.    Solutions existantes.....	8
a)    TakiAcademy .....	8
b)    TopnetSchool .....	9
c)    Droussi.....	9
IV.   Critique de l'existant .....	10
V.   Solutions proposées .....	11
VI.  Étude des méthodologies de développement.....	11
1.    Les méthodes classiques.....	11
a)    Les modèles en cascade .....	11
b)    Les modèles en spirale.....	12
2.    Les méthodes agiles.....	12
a)    Méthode SCRUM.....	13
b)    Méthode XP .....	13
3.    Processus Unifié .....	13
a)    Méthode RUP.....	14
b)    Méthode 2TUP .....	14
4.    Etude comparative .....	15

5. Choix de la méthodologie .....	16
VII. Planification du travail .....	16
1. Diagramme de Gantt .....	16
Conclusion .....	17
<b>CHAPITRE 2 : ANALYSE ET SPÉCIFICATION DES BESOINS.....</b>	<b>18</b>
<b>Introduction.....</b>	<b>19</b>
I. Analyse des besoins .....	19
1. Capture des besoins .....	19
2. Identification des acteurs .....	19
3. Identification des besoins fonctionnels .....	19
a) Besoin fonctionnel pour l'utilisateur (Parent) .....	19
b) Besoins fonctionnels pour l'utilisateur (Apprenant) .....	20
c) Besoins fonctionnels pour l'utilisateur (Formateur) .....	20
d) Besoins fonctionnels pour l'utilisateur (Administrateur) .....	20
e) Besoins fonctionnels pour le visiteur .....	21
4. Identification des besoins non fonctionnels .....	21
II. Identification des besoins techniques.....	22
1. Choix de langage de développement .....	22
III. Analyse semi-formelle .....	23
1. Langage de modélisation .....	23
2. Diagramme de cas d'utilisation.....	23
a) Diagramme de cas d'utilisation global .....	24
b) Diagrammes de cas d'utilisation détaillé.....	26
3. Diagrammes de séquence système .....	32
a) Diagramme de séquence « S'inscrire » .....	32
b) Diagramme de séquence « S'authentifier » .....	33
c) Diagramme de séquence « Gérer profil » .....	34
d) Diagramme de séquence « Gérer jeux » .....	35
e) Diagramme de séquence « Gérer utilisateur » .....	36
Conclusion .....	37
<b>CHAPITRE 3 : ANALYSE CONCEPTUELLE.....</b>	<b>38</b>
<b>Introduction.....</b>	<b>39</b>
I. Conception architecturale de l'application .....	39
1. Architecture physique .....	39
a) Présentation de l'architecture 1 tiers .....	39

b)	Présentation de l'architecture 2 tiers .....	39
c)	Présentation de l'architecture 3 tiers .....	40
d)	Présentation de l'architecture a plusieurs niveau (N tiers) .....	40
e)	Choix de l'architecture adéquate.....	40
2.	Architecture logique .....	41
a)	Architecture MVC (Model – View - Controller) .....	41
b)	Architecture MVP (Model – View – Presentation).....	42
c)	Architecture MVVM (Model – View – View – Model) .....	42
d)	Choix de l'architecture de l'application .....	43
II.	Architecture d'api et de sécurité .....	44
1.	Architecture d'API adapté (API REST) .....	45
2.	Protocole d'authentification JWT (Json Web Token) .....	45
3.	Conception préliminaire des interfaces – prototypes .....	46
III.	Conception de la base de données .....	48
1.	Modèle conceptuel des données (MCD) .....	48
2.	Dictionnaire de données.....	50
3.	Modèle logique de données (MLD) .....	51
IV.	Conception logicielle .....	53
1.	Diagramme de classe.....	53
2.	Diagramme de séquence .....	55
a)	Diagramme de séquence « S'authentifier » .....	55
b)	Diagramme de séquence « S'inscrire » .....	56
c)	Diagramme de séquence « Gérer Jeux » .....	56
d)	Diagramme de séquence « Gérer utilisateurs ».....	58
3.	Diagramme d'activité .....	59
a)	Diagramme d'activité « S'authentifier » .....	59
b)	Diagramme d'activité « Consulter les statistiques de son apprenant » .....	60
4.	Diagramme états-transitions .....	60
a)	Diagramme d'état-transition « Suivre trois jeux par jour ».....	61
b)	Diagramme d'état-transition « paiement » .....	61
5.	Diagramme de composant.....	61
	Conclusion .....	62
	CHAPITRE 4 : RÉALISATION .....	63
	Introduction : .....	64
I.	Environnements de travail .....	64

1.	Environnement matériel.....	64
2.	Environnement logiciel.....	64
a)	Visuel studio code .....	64
b)	Postman .....	65
c)	MongoDB compass .....	65
d)	Mongo atlas.....	65
e)	Github .....	66
f)	Draw.io .....	66
g)	StarUML.....	66
II.	Présentation de la solution .....	66
1.	Interface d'accueil .....	66
2.	Interface de création de compte apprenant.....	67
3.	Interface d'authentification .....	67
4.	Interface des catégories.....	68
5.	Interface des compétences.....	68
6.	Differentes Interface des jeux.....	69
7.	Interface de tableau de classement .....	70
8.	Interface parentale.....	70
9.	Gestion des jeux .....	71
a)	Consulter les jeux « Illustration image ».....	71
b)	Modifier un jeu « Illustration image ».....	71
c)	Créer un jeu « Illustration image ».....	71
	Conclusion .....	73
	CONCLUSION GÉNÉRALE.....	74

## LIST DES FIGURES

Figure 1 : Logo de la société BMCAF Consulting .....	5
Figure 2 : Logo des partenaires de BMCAF Consulting .....	5
Figure 3 : Page d'accueil du site Takiacademy .....	8
Figure 4 : Page d'accueil du site Topnetschool .....	9
Figure 5 : Page d'accueil du site Droussi .....	9
Figure 6 : Cycle en cascade .....	12
Figure 7 : Cycle en spirale .....	12
Figure 8 : Processus de développement suivant la méthodologie Scrum.....	13
Figure 9 : Processus de développement suivant un Processus Unifié .....	14
Figure 10 : Méthode de développement RUP .....	14
Figure 11 : Méthode de développement 2TUP .....	15
Figure 12 : Méthode de développement 2TUP .....	16
Figure 13 : diagramme de Gantt.....	17
Figure 14 : Logo de plateforme Node Js .....	22
Figure 15 : Logo de plateforme ReactJs .....	22
Figure 16 : Logo de Mongo DB .....	22
Figure 17 : Logo UML .....	23
Figure 18 : Diagramme de cas d'utilisation global .....	25
Figure 19 : Diagramme des cas d'utilisation « S'authentifier ».....	26
Figure 20 : Diagramme des cas d'utilisation « Gérer profil ».....	27
Figure 21 : Diagramme des cas d'utilisation « consulter un forum » .....	28
Figure 22 : Diagramme des cas d'utilisation « Gérer jeux ».....	29
Figure 23 : Diagramme des cas d'utilisation « Gérer utilisateurs ».....	31
Figure 24 : Diagramme de séquence « S'inscrire » .....	33
Figure 25 : Diagramme de séquence « S'authentifier » .....	34
Figure 26 : Diagramme de séquence « Gérer profil » .....	35
Figure 27 : Diagramme de séquence « Gérer jeux » .....	36
Figure 28 : Diagramme de séquence « Gérer utilisateurs ».....	37
Figure 29 : Architecture 1-tiers .....	39
Figure 30 : Architecture 2-tiers .....	39
Figure 31 : Architecture 3-tiers .....	40
Figure 32 : Architecture n-tiers .....	40
Figure 33 : Architecture MVC .....	41
Figure 34 : Architecture MVP .....	42
Figure 35 : Architecture MVVM. ....	42
Figure 36 : Architecture de l'application (MVC) .....	43
Figure 37 : Interface couche Modèle .....	43
Figure 38 : Interface couche Vue .....	44
Figure 39 : Interface couche Contrôleur .....	44
Figure 40 : Architecture d'API Rest.....	45
Figure 41 : Protocole d'authentification JWT.....	45

Figure 42 : Maquette préliminaire de l'interface d'accueil .....	46
Figure 43: Maquette préliminaire de l'interface d'authentification.....	47
Figure 44 : Maquette préliminaire de l'interface administrateur .....	47
Figure 45 : Modèle Conceptuel des données .....	49
Figure 46 : Modèle logique de données.....	52
Figure 47 : diagramme de classe .....	54
Figure 48 : Figure de diagramme de séquence « S'authentifier ».....	55
Figure 49 : Figure de diagramme de séquence « S'inscrire ».....	56
Figure 50: Figure de diagramme de séquence « Gérer jeux » .....	57
Figure 51 : Figure de diagramme de séquence « Gérer utilisateurs » .....	58
Figure 52 : Diagramme d'activité « s'authentifier ».....	59
Figure 53: Diagramme d'activité « Consulter les statistiques de son apprenant » .....	60
Figure 54 : Diagramme états-transitions « Suivre trois jeux par jour » .....	61
Figure 55 : Diagramme états-transitions « paiement ».....	61
Figure 56 : Diagramme de composant.....	62
Figure 57 : Logo de visuel studio code.....	64
Figure 58 : Logo du postman .....	65
Figure 59 : Logo de mongoDB compass .....	65
Figure 60 : Logo de mongo DB Atlas .....	65
Figure 61 : Logo de github.....	66
Figure 62 : Logo de draw.io.....	66
Figure 63 : Logo StarUML .....	66
Figure 64 : Page d'accueil .....	67
Figure 65 : Création de compte apprenant.....	67
Figure 66 : Interface d'authentification .....	68
Figure 67 : interface des catégories .....	68
Figure 68 : Interface des compétences .....	69
Figure 69 : Différentes Interface des jeux .....	69
Figure 70 : Interface de tableau de classement .....	70
Figure 71 : Interface d'espace parents .....	70
Figure 72 : Consulter les jeux « Illustration image » .....	71
Figure 73 : Modifier jeux « Illustration image ».....	71
Figure 74 : Choix d'une catégorie .....	72
Figure 75 : Choix d'une compétence .....	72
Figure 76 : Dernière phase de création .....	72

## **LIST DES TABLEAUX**

Tableau 1 : Tableau de comparaison entre les techniques des applications existantes .....	10
Tableau 2 : Comparaison des méthodologies de développement .....	16
Tableau 3 : Tableau descriptif du cas d'utilisation « S'authentifier » .....	26
Tableau 4 : Tableau descriptif du cas d'utilisation « Gérer profil ».....	27
Tableau 5 : Tableau descriptif du cas d'utilisation « Interagir avec forum » .....	29
Tableau 6 : Tableau descriptif du cas d'utilisation « Gérer jeux » .....	30
Tableau 7 : Tableau descriptif du cas d'utilisation « Gérer utilisateurs » .....	32
Tableau 8 : Dictionnaire de données .....	51

---

---

# INTRODUCTION GÉNÉRALE

---

---

L'enseignement traditionnelle est basé sur les cours magistraux associés à des travaux dirigés et pratique, un tableau noir, un support papier et des auditeurs passifs souvent intellectuellement absents à cause des conditions difficiles imposées par la surcharge horaire et les contraintes physiques.

Parmi les inconvénients importants de ce type d'apprentissage, il y a davantage l'accent sur la mémoire que sur la réflexion. Cette formation contribue également peu au développement de la créativité, de l'autonomie, de l'activité. C'est pourquoi on l'appelle souvent « l'école de la mémoire ». Comme le montre la pratique, la quantité d'informations communiquées dépasse les possibilités de leur assimilation. De plus, il n'y a aucune possibilité d'adapter le rythme d'apprentissage aux différentes caractéristiques psychologiques individuelles des élèves [1].

Parlons maintenant de l'idée principale de notre projet, le e-learning que nous ne considérons pas nécessairement comme le contraire de l'enseignement traditionnel mais plutôt comme un complément à celui-ci ou en d'autres termes l'avenir de l'enseignement dans son ensemble.

Le e-Learning est un type de formation en ligne et à distance qui utilise Internet et les nouvelles technologies digitales, dans le but d'améliorer les processus d'apprentissage. Ce type d'enseignement permet aux apprenants de pouvoir se former à distance, quand ils le souhaitent et d'avoir accès à différents contenus pédagogiques 24h/24, 7J/7 et peu importe le lieu où ils se trouvent. Ils peuvent ainsi se former dans le domaine qui les intéresse, et ce, toujours à leur rythme [2].

Grâce aux nouvelles technologies, « l'e-learning » est devenu plus riche en contenu multimédia. Elles ont donné une vision différente de l'enseignement permettant un accès illimité à l'information et à la connaissance aux apprenants et elles ont réduit les distances entre ces derniers et leurs formateurs, en assurant le partage des expériences et du savoir-faire entre eux, à travers, des diverses méthodes de collaboration.

Dans ce contexte on trouve quelques applications comme : **TakiAcademy<sup>1</sup>**, **TopnetSchool<sup>2</sup>**, **Droussi<sup>3</sup>** mais ces dernières présentes certains limites, exemple, elles ne permettent pas à l'apprenant de s'exercer avant l'inscription.

Notre projet de fin d'études s'inscrit dans ce contexte, dans le but de concevoir et de mettre en place un système qui réunit les différentes fonctionnalités à une plateforme éducative interactive qui confronte les inconvénients des solutions existants.

---

<sup>1</sup> : TakiAcademy

<sup>2</sup> : topnetschool

<sup>3</sup> : Plateforme d'apprentissage cours en ligne Tunisie - Droussi

Le présent rapport s'articule autour de quatre chapitres :

- Le premier chapitre intitulé « Contexte générale et étude préalable » qui est dédié à la présentation de l'objectif global du projet et l'analyse de l'existant ;
- Le deuxième chapitre intitulé « Analyse et spécifications des besoins », dans cette partie nous avons analysé et détaillé les besoins demandés par l'entreprise d'accueil ;
- Le troisième chapitre intitulé « Analyse conceptuelle » nous entamerons la conception architecturale et la modélisation organisationnelle et logique de notre base des données ainsi que le diagramme de classe et l'ensemble des diagrammes de séquences détaillés conçut ;
- Le quatrième chapitre « Réalisation » qui sera consacré à l'étude technique, le développement du notre environnement du travail et la présentation de différentes fonctionnalités de notre site web ;
- Conclusion, cette partie est une récapitulation de tous ce qu'on a vu dans les chapitres précédent.

---

---

# CHAPITRE 1 : CONTEXTE GÉNÉRALE ET ÉTUDE PRÉALABLE

---

---

## Introduction

L'étude préalable constitue une étape préliminaire pour la réalisation d'une application. En effet, elle permet d'analyser, d'évaluer et de critiquer le fonctionnement habituel, tout en élaborant la liste des solutions possibles.

### I.Présentation de projet

#### 1. Cadre du projet

Ce travail s'inscrit dans le cadre du projet de fin d'étude pour l'obtention du Diplôme de Licence Appliquée en Système informatique et logiciel à L'institut Supérieur d'Informatique et de Mathématique de Monastir.

C'est un travail collaboratif entre ISIMM la société BMCAF Consulting qui nous a accueillis pour une durée de 4 mois.

#### 2. Organisme d'accueil : « BMCAF Consulting<sup>4</sup> »



Figure 1 : Logo de la société BMCAF Consulting

BMCAF (Bureau méditerranéen de conseil d'assistance et de formation) est un Bureau d'étude et formation créé en 2006 opérant dans le secteur de la formation continue et l'assistance, agréé par l'Etat pour assurer des cycles de formation et des études au profit des entreprises publiques et privées.

BMCAF proposons des formations en inter et intra-entreprise sur mesure adaptées à des besoins spécifiques.

Les partenaires de BMCAF Consulting :



Figure 2 : Logo des partenaires de BMCAF Consulting

---

<sup>4</sup> : <http://bmcaf.tn/>

### 3. Problématique et motivation

L'éducation sous sa forme traditionnelle présente contraintes et limites dont :

- La présence physique des différents acteurs au même endroit et temps ;
- Un coût financier assez important ;
- Une formation liée à un rythme collectif qui n'est pas toujours convenable pour tous les apprenants.

Avec les avancées technologiques réalisées dans le domaine de la communication et l'information, l'apparition du e-learning a permis de se débarrasser de certain rapport à l'éducation traditionnelle tels que :

- Absence de la contrainte temporelle : les ressources sont disponibles à n'importe quel moment 7 jours sur 7, 24 heures sur 24 ;
- Absence de la contrainte géographique : les ressources sont disponibles à n'importe quel endroit du monde ;
- Un faible coût financier (pas de déplacement, pas de salles ...) ;
- Pas de limite sur le nombre d'étudiants inscrit au même cours ;
- Un rythme de formation personnalisé en fonction de la capacité d'apprentissage de l'apprenant.

## II. Concept de base de E-learning

Chaque jour, nous découvrons de nouvelles technologies qui ont révolutionné notre mode de vie personnelle, nous échangeons et informons via divers appareils connectés sans aucun effort.

Afin de mieux comprendre notre futur système, nous définissons dans ce qui suit quelques notions relatives à notre champ d'études :

### 1. E-Learning

L'e-Learning est l'utilisation des nouvelles technologies multimédias de l'Internet pour améliorer la qualité de l'apprentissage en facilitant d'une part l'accès à des ressources et à des services, d'autre part les échanges et la collaboration à distance [3].

Aujourd'hui, le e-Learning est lié spécialement à deux secteurs :

- Secteur professionnel : des grandes entreprises l'utilisent avec management des connaissances pour former et actualiser les connaissances de ses employés de manière plus rapide ;
- Secteur universitaire : afin de minimiser les charges d'enseignement, des universités offrent des alternatives de formation à distance.

Pour organiser un projet de e-Learning, il existe **deux types** d'approches : **asynchrone** et **synchrone** :

- **Approche asynchrone :**

Il s'agit d'un événement d'apprentissage au cours duquel un groupe d'étudiants participe tous à l'apprentissage en même temps. Avant que la technologie d'apprentissage ne permette des environnements d'apprentissage synchrones, la plupart des formations en ligne se déroulaient selon des méthodes d'apprentissage asynchrones [4] ;

- **Approche synchrone :**

Cela fait référence à l'idée que les élèves apprennent le même matériel à des moments et à des endroits différents. L'apprentissage asynchrone est également appelé apprentissage indépendant de l'emplacement et est opposé à l'apprentissage synchrone où les étudiants apprennent en même temps par des activités telles que la participation à une conférence ou à un laboratoire. L'apprentissage asynchrone se déroule selon votre propre horaire [4].

## 2. **Blended Learning (Apprentissage mixte)**

Le blended learning est un type de formation innovant qui mêle cours classiques en présentiel et outils issus de l'enseignement en e-learning. Cette combinaison gagnante permet aux apprenants de suivre les cours de la formation à leur rythme tout en bénéficiant de l'expertise et d'interaction avec le ou les formateurs.

Il est possible d'effectuer des formations en blended learning de plusieurs manières possibles. On peut par exemple débuter un parcours pédagogique par un cours classique dans une salle de formation puis poursuivre par des cours en classe virtuelle qui se caractérisent par des modules en ligne ou des téléconférences. Ou inversement. Il n'y a guère de règles précises. Cette liberté est donc plus ou moins intéressante pour les entreprises et les organismes de formation qui peuvent présenter différents types de planning de formation. Pour les apprenants, la formation mixte constitue un gain de temps puisque ces derniers peuvent suivre une partie de leur formation à distance sans se déplacer en centre de formation ou sur leur lieu de travail [5].

## 3. **Le social Learning (L'apprentissage social)**

Le social learning désigne une méthode d'apprentissage centrée sur les interactions entre les différents membres du groupe. De nombreuses études ont démontré que le social learning est plus efficace que les autres méthodes d'apprentissage.

Dans les années 1970, le psychologue Albert Bandura<sup>5</sup> avait développé sa théorie de l'apprentissage social. Celle-ci met en avant l'importance de l'observation et de l'imitation des comportements, attitudes et émotions des autres dans le processus d'apprentissage. Il s'agit de l'apprentissage entre pairs, également connue sous les noms d'apprentissage par la foule ou d'apprentissage collaboratif.

---

<sup>5</sup> : [https://en.wikipedia.org/wiki/Albert\\_Bandura](https://en.wikipedia.org/wiki/Albert_Bandura)

Le social learning favorise le travail en équipe tout en permettant l'engagement de chaque membre du groupe. Etant basé sur l'entraide, ce mode d'apprentissage permet le développement d'un sentiment d'appartenance au groupe. La synergie entre les individus permet ainsi d'améliorer les performances globales du groupe et de faciliter l'apprentissage [6].

#### 4. Récapitulation

Après avoir plongé dans les bases de l'e-learning et une meilleure compréhension de l'infrastructure de ce type d'enseignement, que l'on parle de « Blended learning » où les étudiants apprennent via les médias électroniques et en ligne ainsi que l'enseignement traditionnel en face-à-face ou le "Social learning" qui propose que l'apprentissage est un processus cognitif qui se déroule dans un contexte social et se produit purement par l'observation ou l'instruction directe, même en l'absence de reproduction motrice ou de renforcement direct, Nous pouvons maintenant construire notre travail sur ces idées existantes et théories, mais aussi ajouter notre propre touche sur ce que nous pensons être des faiblesses dans ces théories.

### III. Étude de l'existant

Étudier les solutions existantes dans le domaine du e-learning est une étape primordiale avant de s'attaquer au projet. Cette étape nous permet de faire l'état des lieux du e-learning pour identifier les points faibles des solutions existantes et déterminer les besoins non satisfaits des différents acteurs. Ceci nous permettra de s'orienter vers la conception d'une solution robuste, efficace et qui répond aux besoins les plus importants des différents intervenants.

#### 1. Solutions existantes

La demande croissante du e-learning dans les deux dernières décennies a conduit au développement de plusieurs solutions de formation à distance. Une recherche approfondie sur les solutions existantes du e-learning nous a permis de faire une liste de celles les plus connues.

##### a) *TakiAcademy*



Figure 3 : Page d'accueil du site Takiacademy

TakiAcademy est une plate-forme d'apprentissage axée sur les étudiants. Enseigner à travers des vidéos ... encadrées par un groupe de professeurs et éducateurs expérimentés. Il y a plus de 1000 vidéos gratuites à voir par vous-même.

Les points forts :

- Paiement facile et sécurisé ;
- Plusieurs utilisateurs peuvent utiliser le même compte ;
- Temps de réponse très rapide.

Les points faibles :

- L'utilisateur doit payer avant même de voir les cours ;
- Takiacademy est le site e-learning le plus connu en Tunisie.

b) ***TopnetSchool***



Figure 4 : Page d'accueil du site Topnetschool

Topnetschool est une plateforme d'apprentissage en ligne conforme aux programmes de l'éducation nationale du préscolaire au Baccalauréat.

Les points forts :

- Temps de réponse très rapide ;
- L'utilisateur peut utiliser l'application sans payer s'il a un abonnement Topnet ADSL ;
- Plusieurs modes de paiement.

Les points faibles :

- Le niveau d'enseignement supérieur est le baccalauréat.

c) ***Droussi***



Figure 5 : Page d'accueil du site Droussi

Droussi est une plateforme d'enseignement et d'apprentissage en ligne destinée à tout le monde. Les cours sont tutorés par des agrégés, des profs d'enseignement et des experts à travers des vidéos et des visioconférences et restent accessibles en replay sans limite de temps.

Les points forts :

- Plateforme très claire à utiliser.

Les points faibles :

- Application web un peu lente ;
- Il n'y a pas d'espace client.

#### IV. Critique de l'existant

Toutes les applications que nous avons présentées précédemment présentent quelques points Faibles/inconvénients :

- Manque d'une fonctionnalité qui permet d'envoyer un mail par la société pour les parents de l'apprenant ;
- Manque d'une fonctionnalité qui permet de calculer le score de l'étudiant pour chaque compétence ;
- Manque ou l'absence d'une fonctionnalité qui teste en temps réel la compétence de l'apprenant par offre.

Le tableau suivant présente la comparaison technique entre les applications web existantes :

	Ergonomie	Simplicité d'utilisation	Rapidité et accessibilité	Sécurité
<b>Takyacademy</b>	Interface simple et facile à comprendre	Plateforme très claire ce que le rendre très simple à utiliser	Accessible avec un temps de réponse très rapide	Forte sécurité
<b>Topnetschool</b>	Interface un peu plus compliquée que les autres plateformes	Simple à utiliser	Accessible avec un temps de réponse rapide	Forte sécurité
<b>Droussi</b>	Une interface simple et compréhensible	Simple à utiliser	Accessible avec un temps de réponse moyenne	Sécurité moyenne

Tableau 1 : Tableau de comparaison entre les techniques des applications existantes

## V.Solutions proposées

L'objectif de notre projet de fin d'études est de créer un site WEB qui fournit ce dont un enfant a besoin pour trouver du divertissement et des encouragements tout en étudiant la possibilité pour les parents de garder un œil attentif sur leur enfant et ses progrès.

- Les utilisateurs auront accès à différents tests de perception et de compréhension dans différents sujets d'études sous la forme de jeux afin qu'ils puissent défier et perfectionner leurs compétences ;
- Tandis que les apprenants résolvent les problèmes, ils remarqueront une augmentation des difficultés à chaque niveau et bien sûr, ils recevront des notes et sur la base de ces notes, leurs parents recevront un feedback pour suivre de près comment leurs enfants s'améliorent ;
- Un Forum d'échange entre différent utilisateur de la plateforme pour échanger des avis et consulter les nouveautés
- Enfin et pour rendre les choses intéressantes, nous avons créé un système de classement basé sur le score ou les notes de chaque participant afin de pouvoir ajouter un peu de concurrence entre les apprenants.

Notre société nous demande de créer la plateforme uniquement pour les jeux à thème mathématique afin de lancer rapidement la première version de l'application.

## VI.Étude des méthodologies de développement

Une méthodologie est la manière qui gouverne la structuration, la planification et le contrôle d'un processus du développement. Par conséquence, le choix de méthodologie à adopter au cours du développement d'un projet est déterminant puisqu'il a des effets directs sur le coût des tâches réalisées et la collaboration entre les différents intervenants.

Il existe deux grandes méthodes de développement : les méthodes classiques et les méthodes agiles

### 1. Les méthodes classiques

Une méthode classique est une méthode de développement Gee. Au début du développement, le client présente sa problématique, ensuite le développeur s'en charge durant une période définie au préalable pour une livraison à la fin de la période programmée.

#### a) *Les modèles en cascade*

Le modèle en cascade est un modèle de gestion linéaire qui divise les processus de développement en phases de projet successives. Contrairement aux modèles itératifs, chaque phase est effectuée une seule fois. Les sorties de chaque phase antérieure sont intégrées comme entrées de la phase suivante. Le modèle en cascade est principalement utilisé dans le développement de logiciel [7].

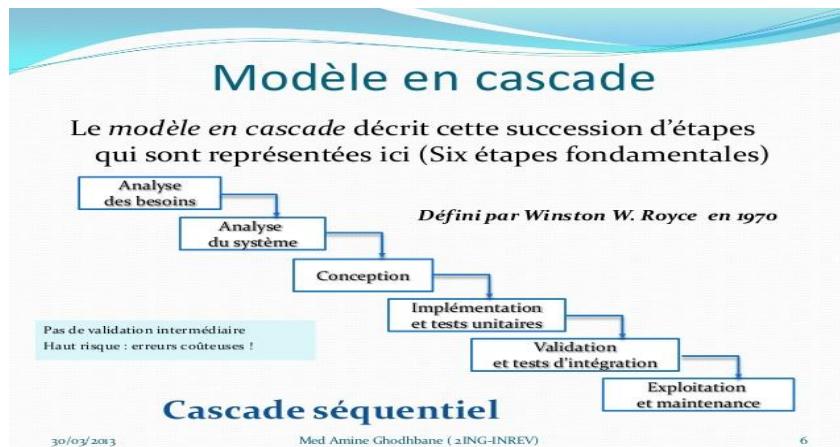


Figure 6 : Cycle en cascade [7]

### b) Les modèles en spirale

Le modèle en spirale a été défini par Barry Boehm<sup>6</sup> en 1988 dans son article "A Spiral Model of Software Development and Enhancement".

Le modèle en spirale (spiral model) est un modèle de Cycle de développement logiciel qui reprend les différentes étapes du cycle en V. Par l'implémentation de versions successives, le cycle recommence en proposant un produit de plus en plus complet et dur [8].

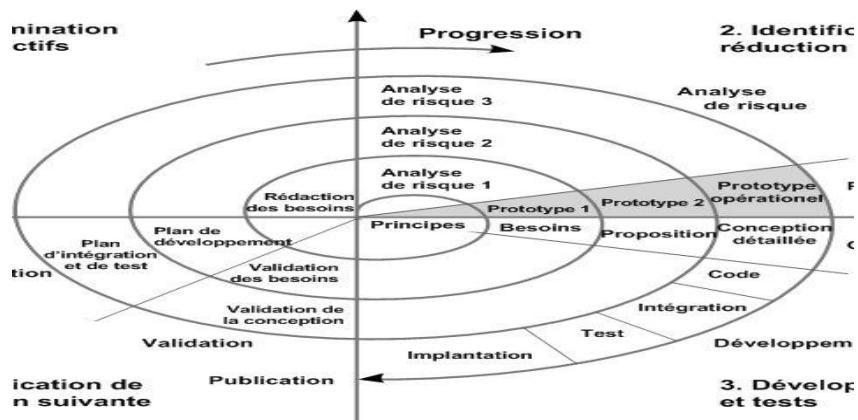


Figure 7 : Cycle en spirale [8]

## 2. Les méthodes agiles

La méthode Agile se base sur un cycle de développement qui porte le client au centre. Le client est impliqué dans la réalisation du début à la fin du projet. Grâce à la méthode agile le demandeur obtient une meilleure visibilité de la gestion des travaux qu'avec une méthode classique.

<sup>6</sup>Boehm B, "A Spiral Model of Software Development and Enhancement"

a) **Méthode SCRUM**

Scrum s'appuie sur le découpage d'un projet en « boîtes de temps », nommées sprints (« pointes de vitesse »). Les sprints peuvent durer entre quelques heures et un mois (avec un sprint médian à deux semaines). Chaque sprint commence par une estimation suivie d'une planification opérationnelle. Le sprint se termine par une démonstration de ce qui a été achevé. Avant de démarrer un nouveau sprint, l'équipe réalise une rétrospective. Cette technique analyse le déroulement du sprint achevé, afin d'améliorer ses pratiques. Le flux de travail de l'équipe de développement est facilité par son auto-organisation, il n'y aura donc pas de gestionnaire de projet.

Cette démarche participative active est un atout fondamental. Elle garantit pour le client le juste équilibre entre l'investissement prévu et le produit finalement livré. L'étude du prototype permet l'évaluation des fonctionnalités réalisées, et facilite la réflexion commune sur l'opportunité de futurs développements. D'autre part, l'étroite intimité entre les clients utilisateurs et les prestataires développeurs facilitent l'appropriation future de l'outil [9].



Figure 8 : Processus de développement suivant la méthodologie Scrum [9]

b) **Méthode XP**

La méthodologie eXtreme Programming ou XP est une méthode de gestion de projet qui applique à l'extrême les principes du développement agile, c'est-à-dire se concentrer sur les besoins du client, mettre en place un développement itératif et l'intégration continue. L'équipe projet et ses relations avec le client sont au cœur de XP [10].

**3. Processus Unifié**

Le processus unifié (PU), ou « unified process (UP) » en anglais est une famille de méthodes de développement de logiciels orientés objets. Elle se caractérise par une démarche itérative et incrémentale, pilotée par les cas d'utilisation, et centrée sur l'architecture et les modèles UML. Elle définit un processus intégrant toutes les activités de conception et de réalisation au sein de cycles de développement composés d'une phase de création, d'une phase d'élaboration, d'une phase de construction et d'une phase de transition, comprenant chacune plusieurs itérations [11].

Le processus unifié se caractérise par une démarche itérative et incrémentale, pilotée par les cas d'utilisation, et centrée sur l'architecture et les modèles UML. Elle définit un

processus intégrant toutes les activités de conception et de réalisation au sein de cycles de développement composés d'une phase de création, d'une phase d'élaboration, d'une phase de construction et d'une phase de transition, comprenant chacune plusieurs itérations.

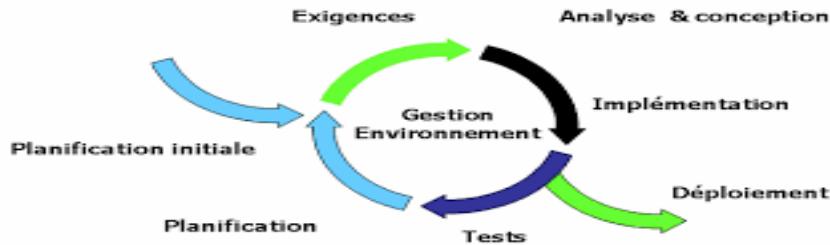


Figure 9 : Processus de développement suivant un Processus Unifié [11]

a) **Méthode RUP**

La méthode de Processus Unifié (UP) s'intéresse au cycle de vie d'un logiciel et le gère. En ce qui concerne les logiciels orientés objet, elle se concentrera sur son développement. La méthode RUP (Rational Unified Process), quant à elle, est une des émanations de la méthode PU, qui s'attache à donner un cadre précis au développement du logiciel. C'est une méthode générique, itérative et incrémentale assez lourde mais qui s'adapte très facilement aux processus et aux besoins du développement [12].

La méthode RUP se construit par quatre éléments nécessaires sont :

- **Création** : réalisation de l'architecture du projet et de la planification.
- **Elaboration** : spécifier les besoins.
- **Construction** : développement du projet et réaliser des tests.
- **Transition** : déploiement et correction du projet.

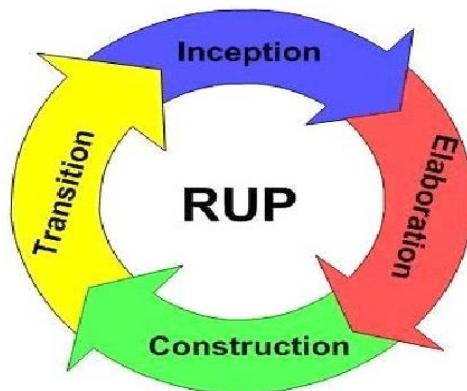


Figure 10 : Méthode de développement RUP [12]

b) **Méthode 2TUP**

2TUP (*two tracks unified process*), ou T2UP, est un processus de développement logiciel qui met en œuvre la méthode du processus Unifié.

Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens) [13].

Le processus s'articule ensuite autour de 3 phases essentielles :

- Une branche technique : capitalise un savoir-faire technique et/ou des contraintes techniques ;
- Une branche fonctionnelle : capitalise la connaissance du métier de l'entreprise ;
- Une phase de réalisation : consiste à réunir les deux branches, permettant de mener une conception applicative.

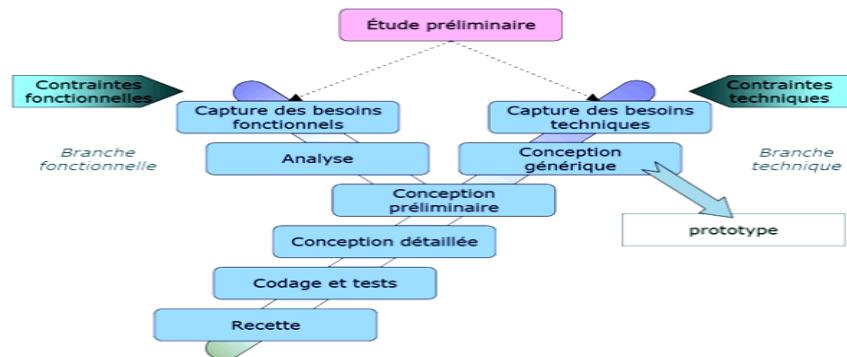


Figure 11 : Méthode de développement 2TUP [13]

#### 4. Etude comparative

Le tableau 1 présente une comparaison des méthodologies de développement

Méthodologie	Avantages	Inconvénients
Scrum	<ul style="list-style-type: none"> <li>• Compréhension du travail et des tâches à accomplir</li> <li>• Transparence</li> <li>• Focus et flexibilité</li> <li>• Visibilité continue</li> <li>• Deadlines intégrées</li> </ul>	<ul style="list-style-type: none"> <li>• Peu de documentation écrite</li> <li>• Il est nécessaire d'être extrêmement rigoureux sur la mise en place et le suivi</li> </ul>
RUP	<ul style="list-style-type: none"> <li>• Fournit une structure bien définie</li> <li>• Concentrée sur la qualité</li> <li>• Itérative et incrémentale</li> </ul>	<ul style="list-style-type: none"> <li>• Exige une certaine expertise</li> <li>• Manque de rapidité</li> <li>• Ne convient pas à un projet de petite envergure</li> </ul>
	<ul style="list-style-type: none"> <li>• Piloté par les cas d'utilisation</li> </ul>	<ul style="list-style-type: none"> <li>• Lorsque des deux branches du cycle en Y vont fusionner. La difficulté va être</li> </ul>

2TUP	<ul style="list-style-type: none"> <li>Toute évolution peut se décomposer et traiter parallèlement</li> <li>Convenable pour tout type de projet</li> </ul>	d'intégrer l'étude fonctionnelle et technique en un seul modèle
XP	<ul style="list-style-type: none"> <li>Rapidité</li> <li>Productivité</li> <li>Réactivité</li> </ul>	<ul style="list-style-type: none"> <li>Maintenance</li> <li>Blocage culturel</li> <li>Limite de taille</li> </ul>

Tableau 2 : Comparaison des méthodologies de développement [14]

## 5. Choix de la méthodologie

Le tableau 1 nous donne une idée de faire notre choix sur la méthodologie à adopter pour la réalisation de notre projet. Notre choix s'est fixé sur la méthodologie 2TUP.

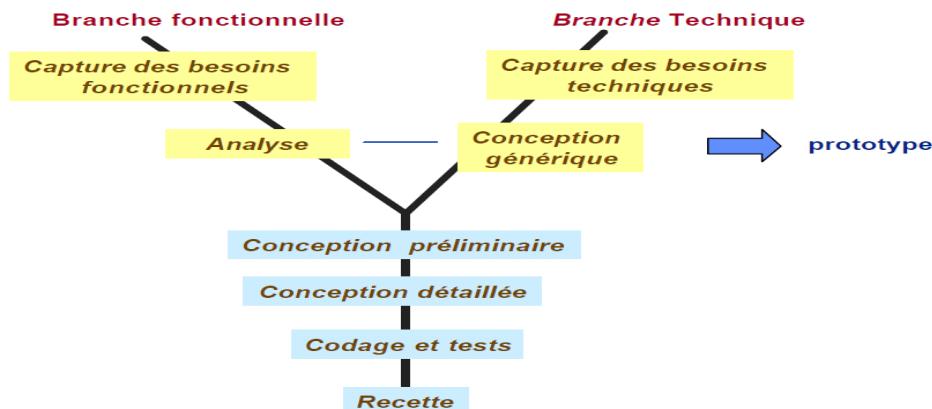


Figure 12 : Méthode de développement 2TUP [13]

## VII. Planification du travail

### 1. Diagramme de Gantt

Le diagramme de Gantt est couramment utilisé dans la gestion d'un projet. Il permet de représenter l'état d'avancement au cours du temps des différentes activités constituant un projet, de manière visuelle. La colonne de gauche énumère les tâches à effectuer, tandis que les colonnes de droite représentent les unités de temps. Chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentent la date de début, la durée et la date de fin.

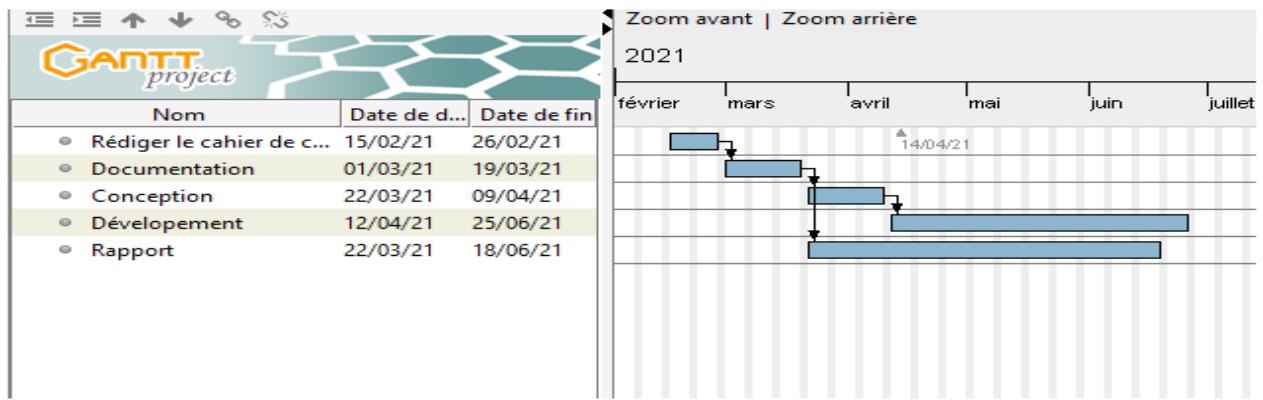


Figure 13 : diagramme de Gantt

## Conclusion

A travers ce chapitre nous avons commencé par présenter le cadre général du projet. Nous avons ensuite présenté une étude des solutions existantes et nous avons décrit le travail à réaliser en tenant compte des lacunes de ces dernières. Enfin, nous avons spécifié la méthodologie de projet ainsi que la planification du travail.

Le prochain chapitre sera consacré à spécifier les différents besoins de notre application.

---

---

## CHAPITRE 2 : ANALYSE ET SPÉCIFICATION DES BESOINS

---

---

## Introduction

Dans ce chapitre nous allons mettre en pratique la méthodologie "2TUP ", que nous avons adoptée pour le développement de notre projet. Alors, nous commençons en premier lieu par une spécification de besoin auxquels doit répondre l'application, passant ensuite à l'analyse de ces besoins à travers l'indentification des acteurs, puis les diagrammes de cas d'utilisation relatifs à ces acteurs et enfin le diagramme séquence système.

### I.Analyse des besoins

#### 1. Capture des besoins

Dans cette partie nous allons citer les besoins qui se divisent en trois catégories :

- Les besoins fonctionnels ;
- Les besoins non fonctionnels ;
- Les besoins techniques.

#### 2. Identification des acteurs

Les acteurs qui dialoguent avec le système :

- Utilisateur (Parent) : est un utilisateur qui n'est pas inscrit au système son rôle est de faire le paiement du compte apprenant et de recevoir des emails de ses améliorations ;
- Utilisateur (Apprenant) : est un utilisateur qui est inscrit au système en tant qu'apprenant ;
- Utilisateur (Formateur) : est un utilisateur qui est inscrit au système en tant que formateur ;
- Utilisateur (Administrateur) : c'est le principal utilisateur qui gère l'application ;
- Visiteur : une personne, peut être client qui veut consulter des informations.

#### 3. Identification des besoins fonctionnels

Cette phase consiste à définir les fonctionnalités de système, signifie les fonctionnalités offertes par le système aux différents acteurs.

##### a) *Besoin fonctionnel pour l'utilisateur (Parent)*

**Payer l'abonnement de son apprenant :** lors de la création d'un compte apprenant, le parent effectuera le paiement.

**Consulter les notifications par mail :** un utilisateur parent va recevoir des emails sur les progrès de son enfant, les défauts et bien d'autres pour le garder à jour.

**Consulter les statistiques de son apprenant :** un utilisateur parent peut se connecter avec l'adresse email de son apprenant et consulter les statistiques (score, question traiter etc).

b) *Besoins fonctionnels pour l'utilisateur (Apprenant)*

**S'authentifier :** la plateforme oblige l'étape d'authentification pour que l'utilisateur peut accéder à son compte et cette étape se fait par l'entrée de l'email et le mot de passe.

**Gérer profil :** la fonctionnalité gérer le profil donne le droit à l'utilisateur de consulter son profil et de mettre à jour tous ses coordonnées personnelles et les changer en toute sécurité.

**Suivre les catégories :** la fonctionnalité Suivre les catégories donne le droit à l'utilisateur de suivre n'importe quelle catégorie qu'il a besoin.

**Suivre les compétences par catégories :** lors de choisir les besoins catégories, l'utilisateur va consulter et suivre les compétences existantes pour ces dernières.

**Suivre les jeux par compétences :** afin de choisir les besoins compétences, l'utilisateur va suivre les jeux existants à ces dernières.

c) *Besoins fonctionnels pour l'utilisateur (Formateur)*

**S'authentifier :** le formateur doit s'authentifier par son login (adresse e-mail et mot de passe) pour accéder à l'espace administrateur.

**Gérer catégories :** lorsque le formateur est inscrit à la plateforme, le système lui donne le droit de créer des catégories et les publier en spécifiant le nom et description des compétences requises à cette catégorie.

**Gérer compétences :** le système donne au formateur le droit de créer des compétences dans la catégorie qu'il veut et les publier en spécifiant le nom de cette compétence.

**Gérer jeux :** le système donne au formateur le droit de créer des jeux dans la compétence qu'il veut et les publier en spécifiant le type, la question et la réponse correcte.

d) *Besoins fonctionnels pour l'utilisateur (Administrateur)*

**S'authentifier :** l'administrateur doit s'authentifier par son login (adresse e-mail et mot de passe) pour accéder à l'espace Administrateur.

**Gérer utilisateurs :** le système donne à l'administrateur le droit d'ajouter, supprimer ou modifier les utilisateurs.

**Gérer paiements :** le système donne à l'administrateur le droit de consulter les paiements des utilisateurs.

**Gérer notifications :** le système donne à l'administrateur le droit d'ajouter ou supprimer les notifications ;

**Gérer catégories :** lorsque l'administrateur est connecté à la plateforme, le système lui donne le droit de créer des catégories et les publier en spécifiant le nom et description des compétences requises à cette catégorie.

**Gérer compétences :** le système donne à l'administrateur le droit de créer des compétences dans la catégorie qu'il veut et les publier en spécifiant le nom de cette compétence.

**Gérer jeux :** le système donne à l'administrateur le droit de créer des jeux dans la compétence qu'il veut et les publier en spécifiant le type, la question et la réponse correcte.

e) ***Besoins fonctionnels pour le visiteur***

**S'inscrire :** un visiteur peut s'inscrire à la plateforme et avoir un compte, il doit posséder une adresse email, nom, prénom, etc. ... pour l'enregistrer et afin de vérifier ses coordonnées lors de la prochaine connexion à la plateforme.

**Consulter les catégories :** un visiteur peut suivre n'importe quelle catégorie qu'il a besoin.

**Suivre les jeux par compétences :** afin de choisir les besoins compétences, le visiteur va suivre un seul jeu par jour.

#### 4. Identification des besoins non fonctionnels

Les besoins non fonctionnels se présentent en matière de performance, Et tout ce qui concerne les outils d'implémentation comme le langage de programmation, SGBD.

Chaque plateforme doit répondre à certaines qualités à part les critères fondamentaux comme il est représenté comme ci-dessous :

- **Convivialité :** notre plateforme web doit être facile à utiliser et à comprendre avec des interfaces utilisateurs compréhensible et significatif afin de le rendre simple et adapté à l'utilisateur.
- **Rapidité :** notre plateforme doit assurer le critère de rapidité de réponse et aussi la fiabilité du résultat procuré.
- **Performance :** le terme performance d'une façon générale se présente en termes de temps de réponse de plateforme, qui signifie le temps de charger les pages web dans le navigateur, vue que notre Plateforme est réalisée par la nouveauté technologies ce qu'il le rend capable de répondre à cette qualité nécessaire.
- **Sécurité :** le terme sécurité désigne la conservation de confidentialité des informations des utilisateurs à l'aide des algorithmes des cryptages.

## II. Identification des besoins techniques

### 1. Choix de langage de développement

D'après le choix de l'entreprise on va choisir la technologie suivante :

- **Node js** : utilisé pour développer la partie back-end de notre plateforme web, le logo est présenté dans la figure ci-dessous :

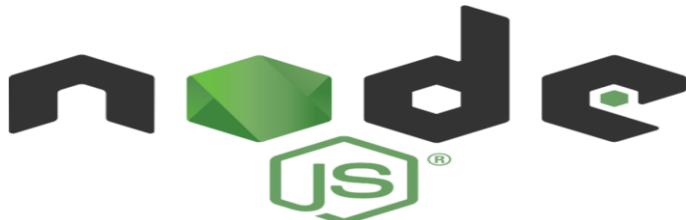


Figure 14 : Logo de plateforme Node Js<sup>7</sup>

- **React js** : utilisé pour développer la partie front-end de notre plateforme web, le logo est présenté dans la figure ci-dessous :

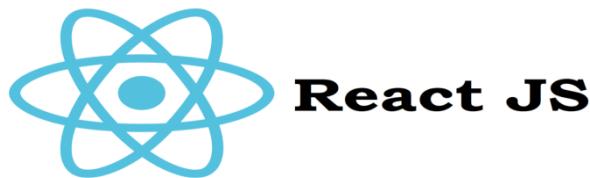


Figure 15 : Logo de plateforme ReactJs<sup>8</sup>

- **Mongo DB** : système de gestion de base de données orientée documents, qui joue le rôle de stocker les données et assurer l'authentification, le logo de mongo DB est présenté par la figure ci-dessous :



Figure 16 : Logo de Mongo DB<sup>9</sup>

---

<sup>7</sup> : <https://devtobecurious.com/wp-content/uploads/2020/10/NodeJS.jpg>

<sup>8</sup> : [https://www.expertunisie.com/wp-content/uploads/2019/12/1\\_yk5D5cQB3jd7EiPzrDrD5w-768x337.png](https://www.expertunisie.com/wp-content/uploads/2019/12/1_yk5D5cQB3jd7EiPzrDrD5w-768x337.png)

<sup>9</sup> : <https://www.universitylib.com/wp-content/uploads/2020/11/Mongo-db-logo.png>

### III. Analyse semi-formelle

La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en *faciliter la réalisation*.

#### 1. Langage de modélisation

Afin de modéliser les différentes fonctionnalités offertes par notre application, d'éclaircir la représentation de son architecture et de montrer les interactions possibles entre ses différents éléments, nous avons utilisé le langage UML.



Figure 17 : Logo UML [15]

Nous n'utilisons pas tous les types de diagrammes proposés par UML 2, mais seulement un tiers parmi eux en insistant particulièrement sur les diagrammes suivants :

- Diagramme de cas d'utilisation ;
- Diagramme de séquence ;
- Diagramme d'activité ;
- Diagramme états-transitions ;
- Diagramme de composant.

Cette limitation est largement suffisante pour la plupart des projets et particulièrement pour notre travail.

#### 2. Diagramme de cas d'utilisation

Le diagramme des cas d'utilisations exprime les fonctionnalités fournies par le système informatique dédié aux utilisateurs, répondant aux questions ce que fait le système en présentant une vision globale sur les différentes interactions entre le système et les utilisateurs.

Pour affiner le diagramme de cas d'utilisation, UML définit trois types de relations standardisées entre cas d'utilisation :

- Une relation d'inclusion, formalisée par le mot-clé `include` : le cas d'utilisation de base en incorpore explicitement un autre, de façon obligatoire ;
- Une relation d'extension, formalisée par le mot-clé `extend` : le cas d'utilisation de base en incorpore implicitement un autre, de façon optionnelle ;

- Une relation de généralisation/spécialisation : les cas d'utilisation descendants héritent de la description de leur parent commun. Chacun d'entre eux peut néanmoins comprendre des interactions spécifiques supplémentaires.
- Types de relations :
- L'association (trait plein avec ou sans èche) entre acteurs et cas d'utilisation-On ;
- La dépendance (èche pointillée) entre cas d'utilisation, avec les mots-clés extend ou include ;
- La relation de généralisation (èche fermée vide) entre cas d'utilisation.

a) ***Diagramme de cas d'utilisation global***

Le diagramme de cas d'utilisation est un formalisme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. Il représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases).

La figure ci-dessous illustre le diagramme de cas d'utilisation global relatif à la solution :

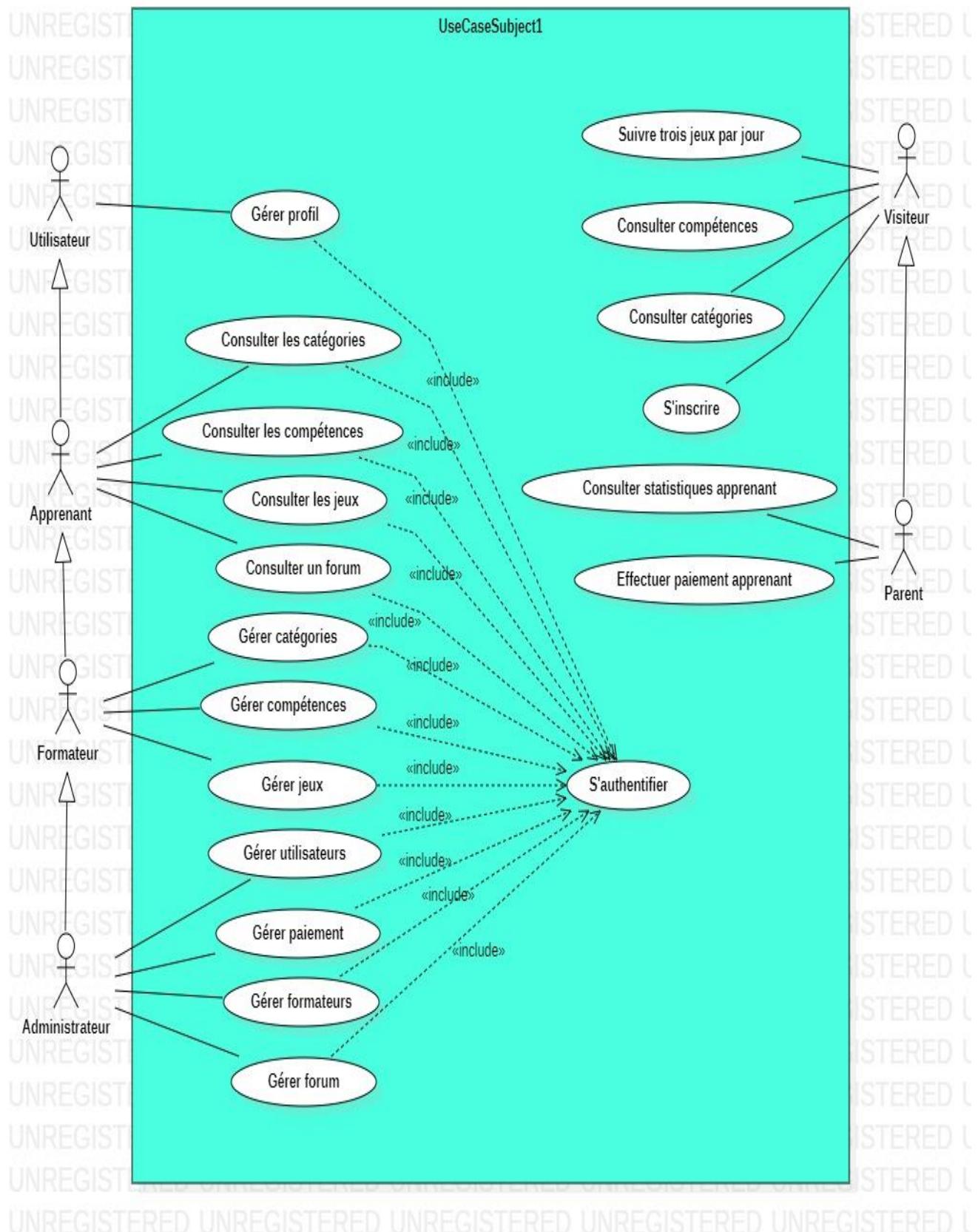


Figure 18 : Diagramme de cas d'utilisation global

**b) Diagrammes de cas d'utilisation détaillé**

Dans cette section nous allons présenter une description détaillée pour quelques cas d'utilisation.

- Cas d'utilisation « S'authentifier »

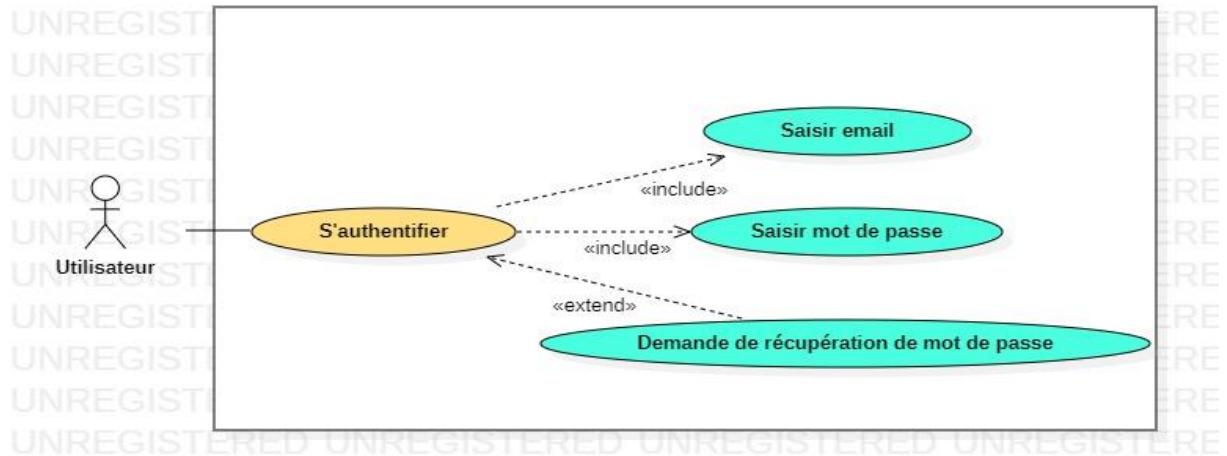


Figure 19 : Diagramme des cas d'utilisation « S'authentifier »

Description textuelle de cas d'utilisation « S'authentifier » :

Sommaire d'identifiant	
Acteur(s)	Utilisateur
Objectif	L'apprenant doit s'authentifier pour effectuer des tâches seul l'utilisateur peut les faire.
Description d'enchaînements	
Pré-conditions	Utilisateur non authentifié.
Post-conditions	Utilisateur authentifié.
Scénario nominal	1- L'apprenant accède à la page d'authentification ; 2- Le Système valide l'authenticité de l'utilisateur ; 3- L'utilisateur est authentifié.
Alternatives	2.1- l'utilisateur ne peut être authentifié : un champ invalide ; 2.2- Le Système revient sur l'affichage de la page d'authentification.

Tableau 3 : Tableau descriptif du cas d'utilisation « S'authentifier »

- Cas d'utilisation « Gérer profil »

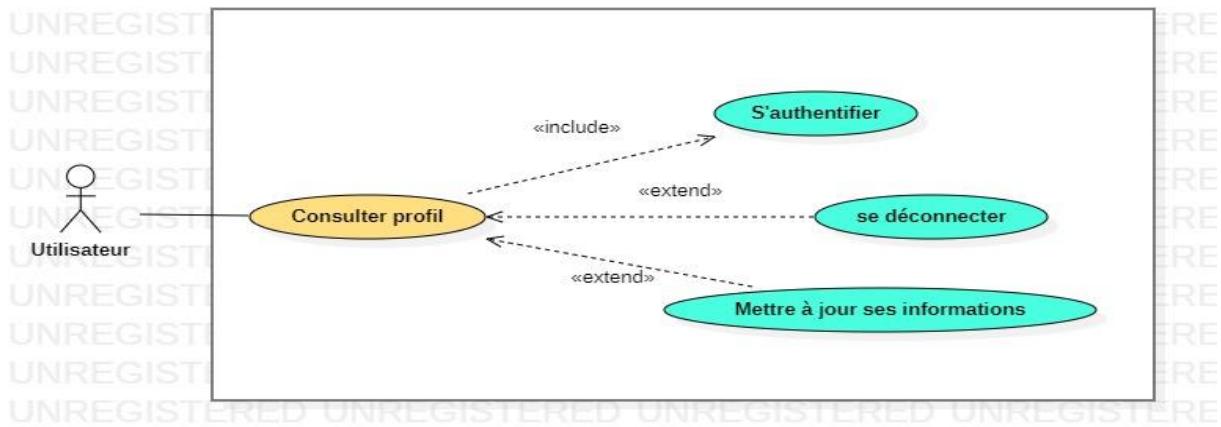


Figure 20 : Diagramme des cas d'utilisation « Gérer profil »

Description textuelle de cas d'utilisation « Gérer profil » :

Sommaire d'identifiant	
Acteur(s)	Utilisateur
Objectif	Permet à l'utilisateur de gérer son profil et modifier certaines informations (email, mot de passe, photo...)
Description d'enchâinement	
Pré-conditions	L'utilisateur doit s'authentifier.
Post-conditions	Base de données mise à jour.
Scénario nominal	<ul style="list-style-type: none"> <li>-L'utilisateur demande la page de profil ;</li> <li>-Le système affiche le profil actuel de l'utilisateur ;</li> <li>-L'utilisateur mette à jour ses informations ;</li> <li>-Le système vérifie la validité des informations saisies ;</li> <li>-Le système enregistre ces informations dans la base de données ;</li> <li>-Le système renvoi un message de confirmation de mise à jour.</li> </ul>
Alternatives	<ul style="list-style-type: none"> <li>- Les informations sont manquantes ou incorrectes : <ul style="list-style-type: none"> <li>+ le système informe l'utilisateur que les données saisies sont erronées.</li> </ul> </li> </ul>

Tableau 4 : Tableau descriptif du cas d'utilisation « Gérer profil »

- Cas d'utilisation « consulter un forum »

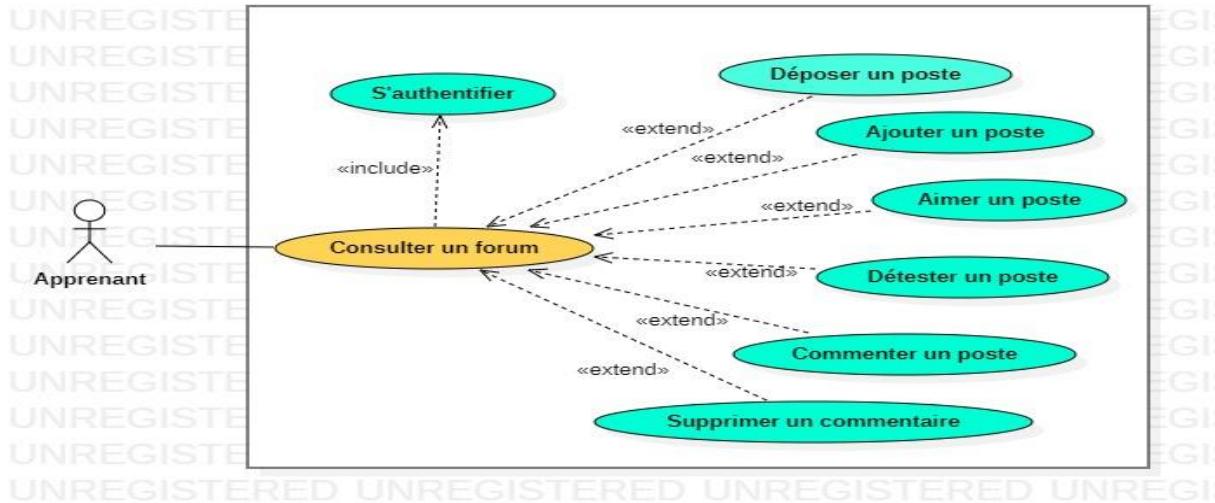


Figure 21 : Diagramme des cas d'utilisation « consulter un forum »

Description textuelle de cas d'utilisation « consulter un forum » :

Sommaire d'identifiant	
Acteur(s)	Apprenant
Objectif	Permet à l'apprenant d'ajouter, supprimer, commenter, aimer ou détester un poste. Il peut aussi supprimer un commentaire
Description d'enchaînements	
Pré-conditions	L'apprenant doit s'authentifier.
Post-conditions	Base de données mise à jour.
Scénario nominal	<ul style="list-style-type: none"> <li>-L'apprenant demande d'accéder au forum ;</li> <li>-Le système affiche l'interface forum ;</li> <li>-L'apprenant peut choisir une opération :           <ul style="list-style-type: none"> <li>+ « <b>Ajouter un poste</b> »               <ul style="list-style-type: none"> <li>• Le système affiche un formulaire ;</li> <li>• L'apprenant remplit le formulaire d'ajout ;</li> <li>• Le système vérifie les paramètres ;</li> <li>• Poste ajouté.</li> </ul> </li> <li>+ « <b>Commenter un poste</b> »               <ul style="list-style-type: none"> <li>• L'apprenant remplir le formulaire d'ajout commentaire ;</li> <li>• Le système ajouté le commentaire.</li> </ul> </li> <li>+ « <b>Aimer un poste</b> »</li> <li>+ « <b>Supprimer un poste</b> »</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>+ « Supprimer un commentaire »</li> <li>+ « Détester un poste »</li> </ul>
Alternatives	<ul style="list-style-type: none"> <li>- L'utilisateur ajouté un poste :</li> <li>+ champ invalide alors le système affiche un message « Champ Invalidé ! ».</li> <li>- L'utilisateur supprimé un poste :</li> <li>+ si le poste n'appartient pas à cet utilisateur et le rôle de cet utilisateur n'est pas administrateur, le système affiche un message « non autorisé ».</li> <li>-L'utilisateur supprimer un commentaire :</li> <li>+si le commentaire n'appartient pas à cet utilisateur et le rôle de cet utilisateur n'est pas administrateur, le système affiche un message « Non autorisé ».</li> </ul>

Tableau 5 : Tableau descriptif du cas d'utilisation « Interagir avec forum »

- Cas d'utilisation « Gérer jeux »

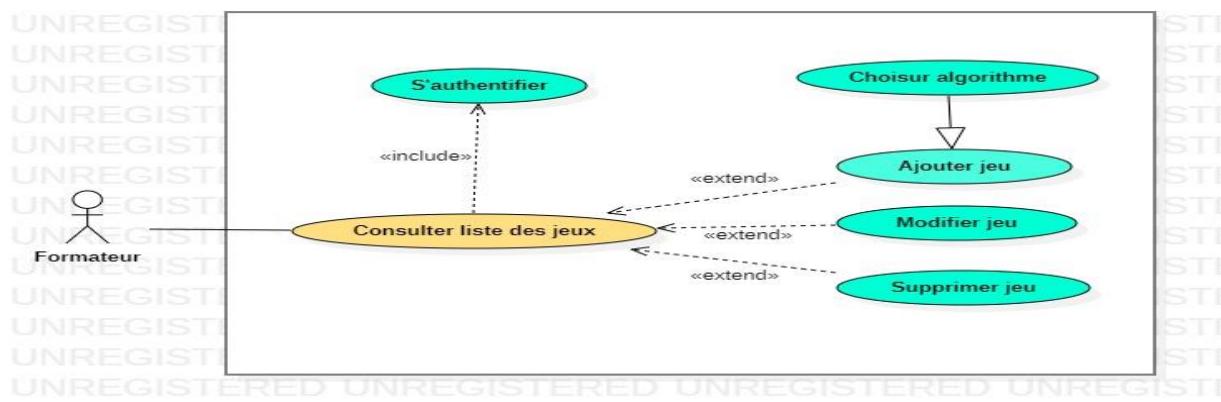


Figure 22 : Diagramme des cas d'utilisation « Gérer jeux »

Description textuelle de cas d'utilisation « Gérer jeux » :

Sommaire d'identifiant	
Acteur(s)	Formateur
Objectif	Permet au formateur d'ajouter, supprimer ou modifier un jeu.
Description d'enchaînements	
Pré-conditions	Le formateur doit s'authentifier.
Post-conditions	Base de données mise à jour.
Scénario nominal	<ul style="list-style-type: none"> <li>- Le formateur demande la liste des jeux ;</li> <li>- Le système affiche la liste de tous les jeux ;</li> <li>- Le formateur peut choisir l'opération :           <ul style="list-style-type: none"> <li>+ « <b>Ajouter jeu</b> »               <ul style="list-style-type: none"> <li>• Le système affiche un formulaire ;</li> <li>• Le formateur remplit le formulaire d'ajout et choisir un algorithme ;</li> <li>• Le système vérifie les paramètres ;</li> <li>• Jeu ajouté.</li> </ul> </li> <li>+ « <b>Modifier jeu</b> »               <ul style="list-style-type: none"> <li>• Le système affiche un formulaire ;</li> <li>• Le formateur remplit le formulaire de modification ;</li> <li>• Le système vérifie les paramètres ;</li> <li>• Jeu modifié.</li> </ul> </li> <li>+ « <b>Supprimer jeu</b> »</li> </ul> </li> </ul>
Alternatives	<ul style="list-style-type: none"> <li>- l'administrateur ajoute un utilisateur :           <ul style="list-style-type: none"> <li>+ champ invalide alors le système affiche un message Champ Invalide !</li> </ul> </li> <li>- l'administrateur modifiée un utilisateur :           <ul style="list-style-type: none"> <li>+ si champ invalide alors le système affiche un message Champ Invalide!</li> </ul> </li> </ul>

Tableau 6 : Tableau descriptif du cas d'utilisation « Gérer jeux »

- Cas d'utilisation « Gérer utilisateurs »

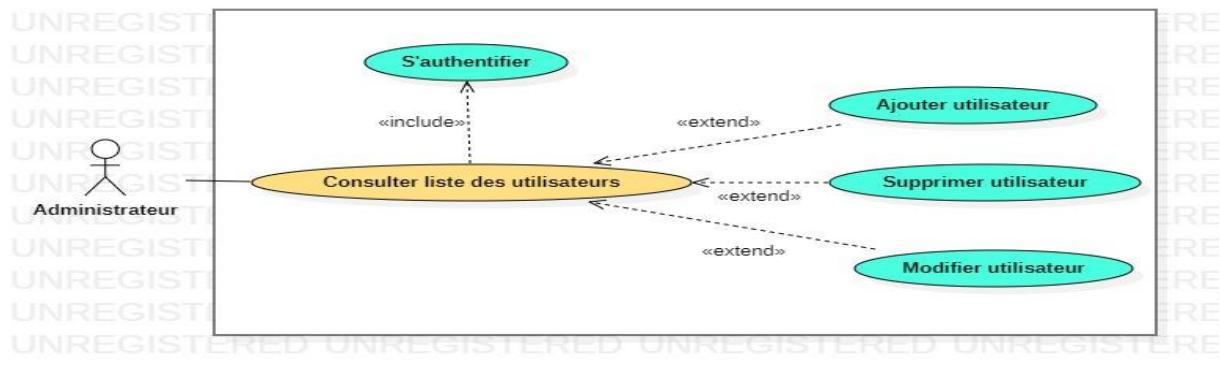


Figure 23 : Diagramme des cas d'utilisation « Gérer utilisateurs »

Description textuelle de cas d'utilisation « Gérer utilisateurs » :

Sommaire d'identifiant	
Acteur(s)	Administrateur
Objectif	Permet à l'administrateur d'ajouter, supprimer ou modifier un utilisateur.
Description d'enchaînements	
Pré-conditions	L'administrateur doit s'authentifier.
Post-conditions	Base de données mise à jour.
Scénario nominal	<p>-L'administrateur demande la liste des utilisateurs ;</p> <p>-Le système affiche la liste de tous les utilisateurs ;</p> <p>-L'administrateur peut choisir une opération :</p> <ul style="list-style-type: none"> <li>+ « <b>Ajouter utilisateur</b> » <ul style="list-style-type: none"> <li>• Le système affiche un formulaire ;</li> <li>• L'administrateur remplit le formulaire d'ajout ;</li> <li>• Le système vérifie les paramètres ;</li> <li>• Utilisateur ajouté.</li> </ul> </li> <li>+ « <b>Supprimer utilisateur</b> »</li> <li>+ « <b>Modifier utilisateur</b> » <ul style="list-style-type: none"> <li>• Le système affiche un formulaire ;</li> <li>• L'administrateur remplit le formulaire d'ajout ;</li> <li>• Le système vérifie les paramètres ;</li> <li>• Utilisateur modifié.</li> </ul> </li> </ul>

Alternatives	<ul style="list-style-type: none"> <li>- l'administrateur ajoute un utilisateur :</li> <li>+ champ invalide alors le système affiche un message Champ Invalidé !</li> <li>- l'administrateur modifié un utilisateur :</li> <li>+ si champ invalide alors le système affiche un message Champ Invalidé!</li> </ul>
--------------	---

Tableau 7 : Tableau descriptif du cas d'utilisation « Gérer utilisateurs »

### 3. Diagrammes de séquence système

Le diagramme de séquence est la représentation graphique des interactions entre les acteurs et le système dans la formulation UML. Ce diagramme est utilisé pour représenter certains aspects dynamiques d'un système dans le contexte d'une opération, un système ou d'un cas d'utilisation selon un point de vue temporel.

Tout d'abord, nous définissons quelques notions de base du diagramme :

- Scénario : est un ensemble de messages échangés entre un ensemble des objets ou bien acteur dans un certain contexte pour réaliser une tâche ;
- Interaction : est un ensemble de messages échangés entre un ensemble des objets ou bien acteur dans un certain contexte pour réaliser une tâche ;
- Message : les messages servent à représenter la communication entre les objets, et non la relation structurelle présente entre les classes.

#### a) *Diagramme de séquence « S'inscrire »*

Un visiteur demande son inscription et le système lui renvoie le formulaire à remplir, cet acteur saisir ses coordonnées. Le système vérifie les coordonnées s'ils sont valides il affiche la page d'accueil, sinon il envoie un message d'erreur.

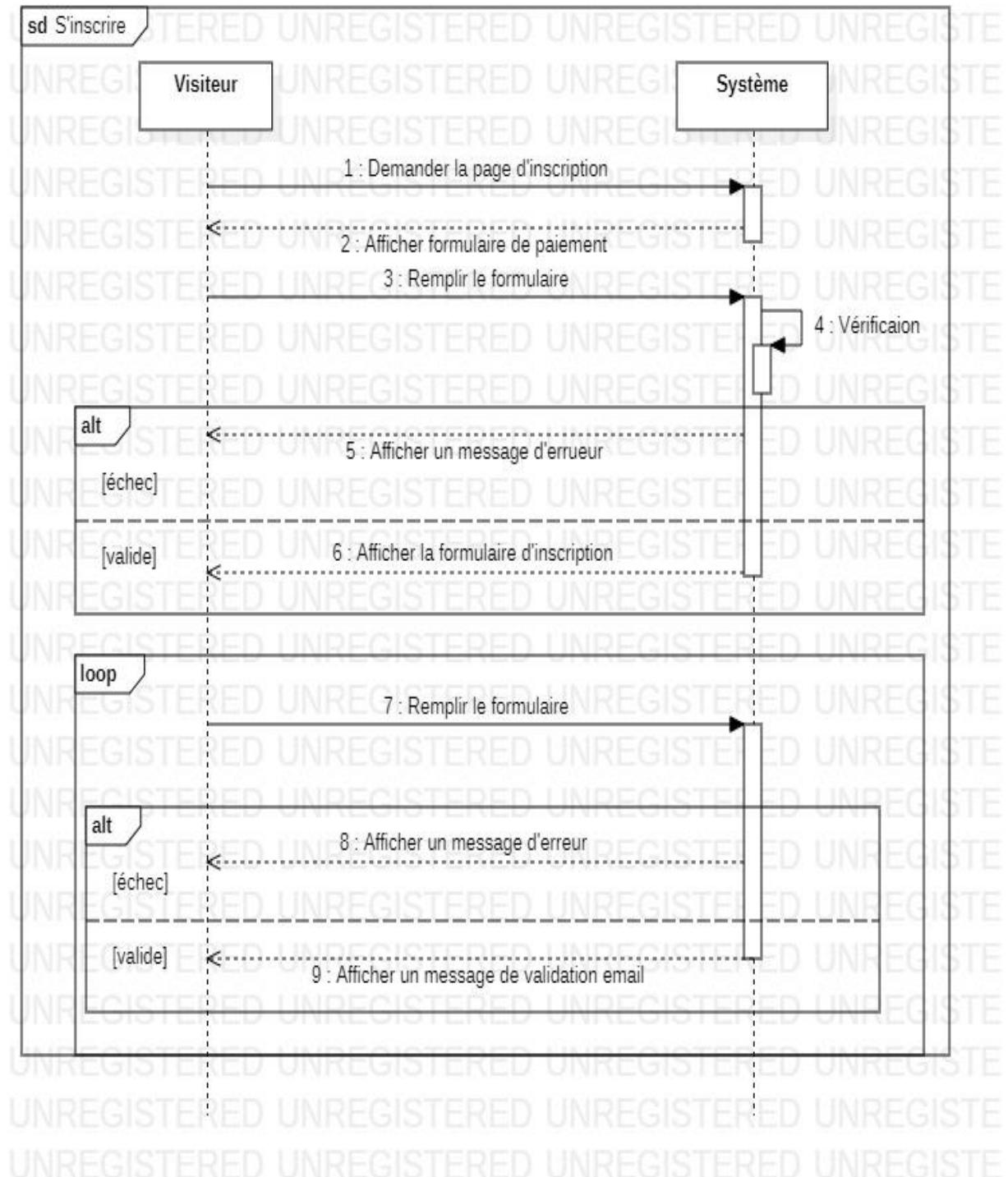


Figure 24 : Diagramme de séquence « S'inscrire »

**b) Diagramme de séquence « S'authentifier »**

Un utilisateur demande son authentification et le système lui renvoie le formulaire à remplir, cet acteur saisir ses coordonnées. Le système vérifie les identifiants s'ils sont valides il affiche la page d'accueil, sinon il envoie un message d'erreur.

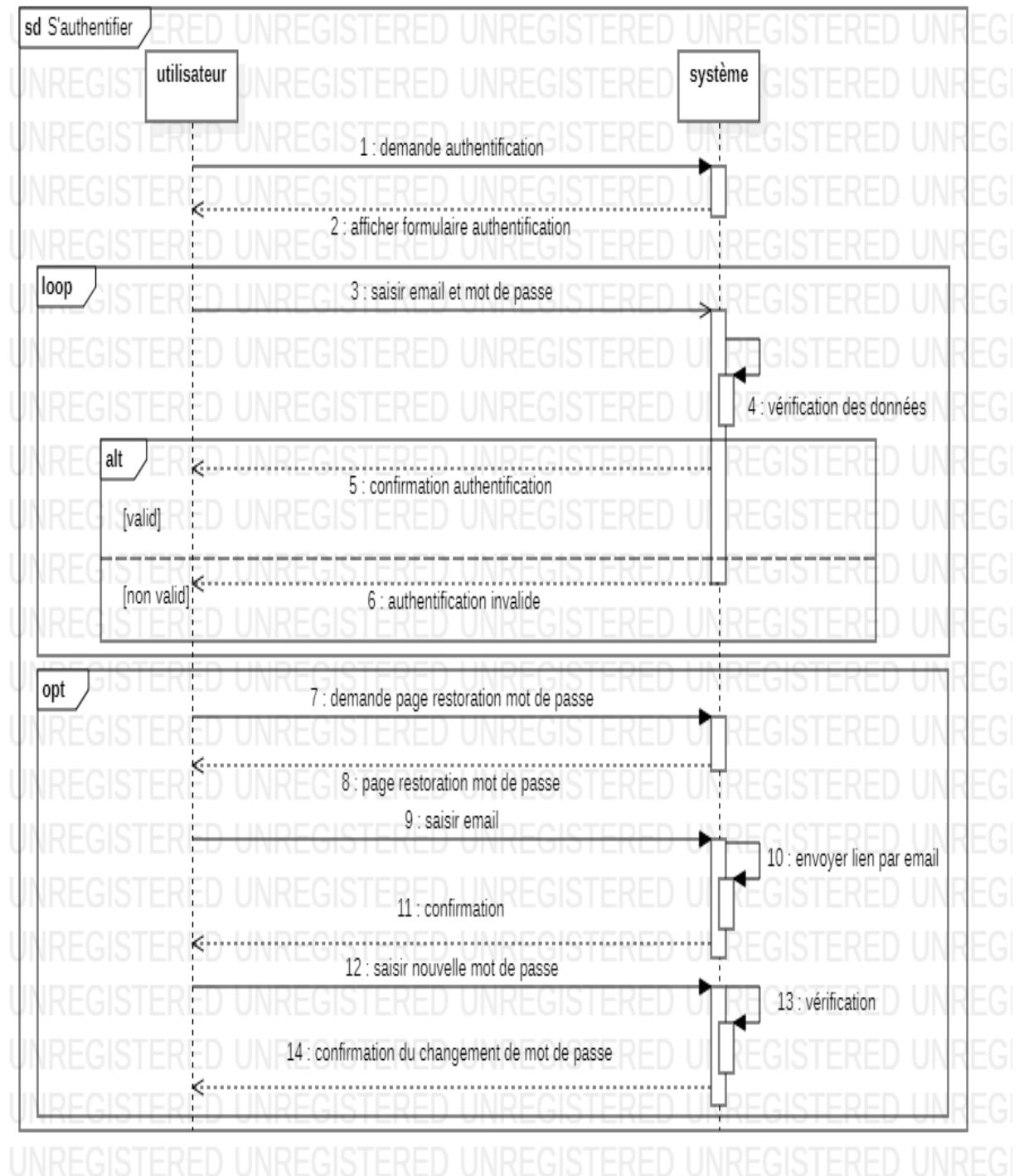


Figure 25 : Diagramme de séquence « S'authentifier »

c) **Diagramme de séquence « Gérer profil »**

Après l'authentification, l'utilisateur demande de mettre à jour ses informations, et le système lui renvoie le formulaire de mise à jour. L'utilisateur remplir le formulaire et soumettre. Enfin le système renvoi un message de confirmation ou un message d'erreur.

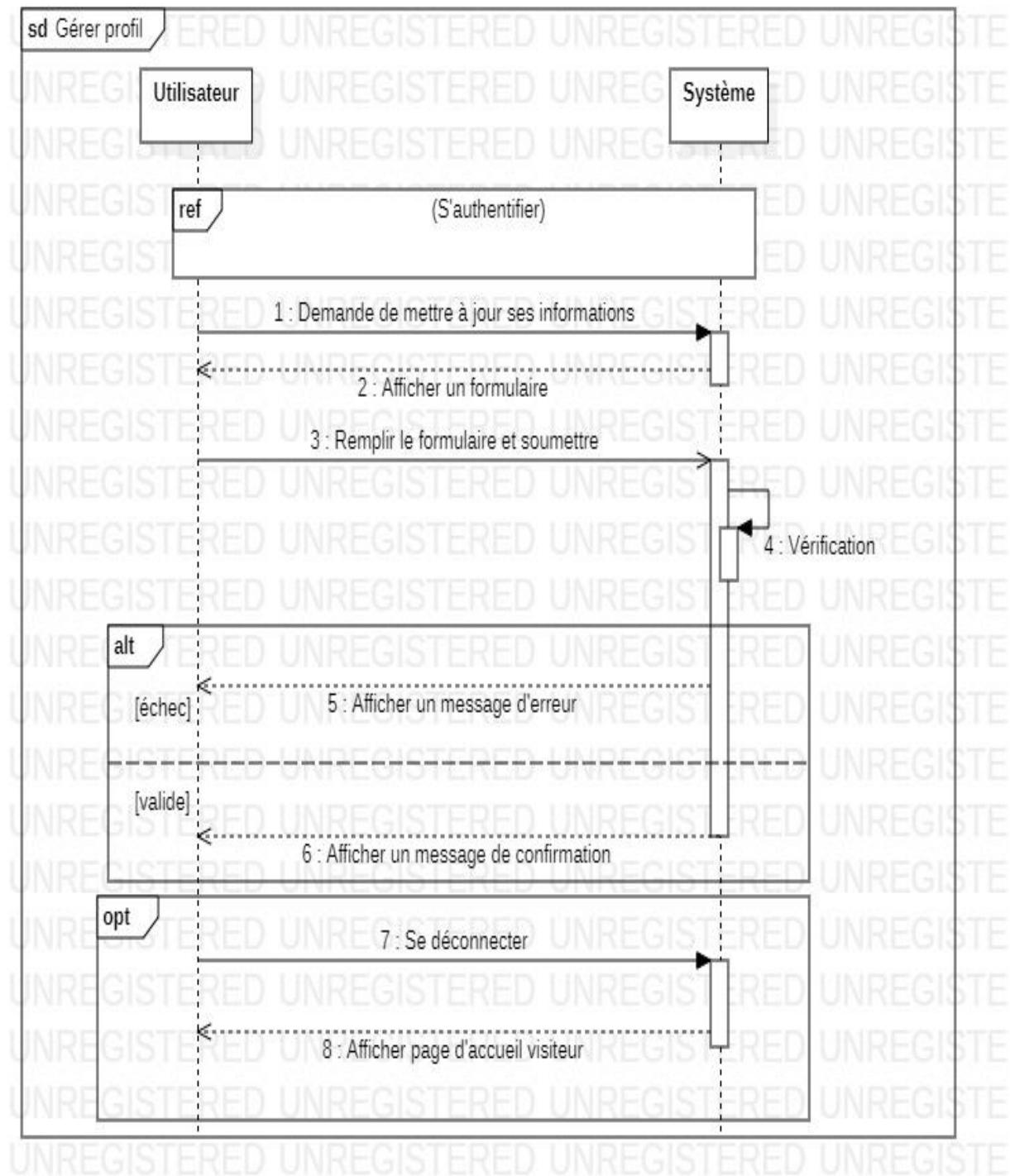


Figure 26 : Diagramme de séquence « Gérer profil »

**d) Diagramme de séquence « Gérer jeux »**

Après l'authentification, l'administrateur demande la page liste des jeux. Le système lui renvoie liste des jeux, et l'administrateur a choisi d'ajouter, supprimer ou modifier un jeu. Enfin le système renvoi un message de confirmation ou un message d'erreur.

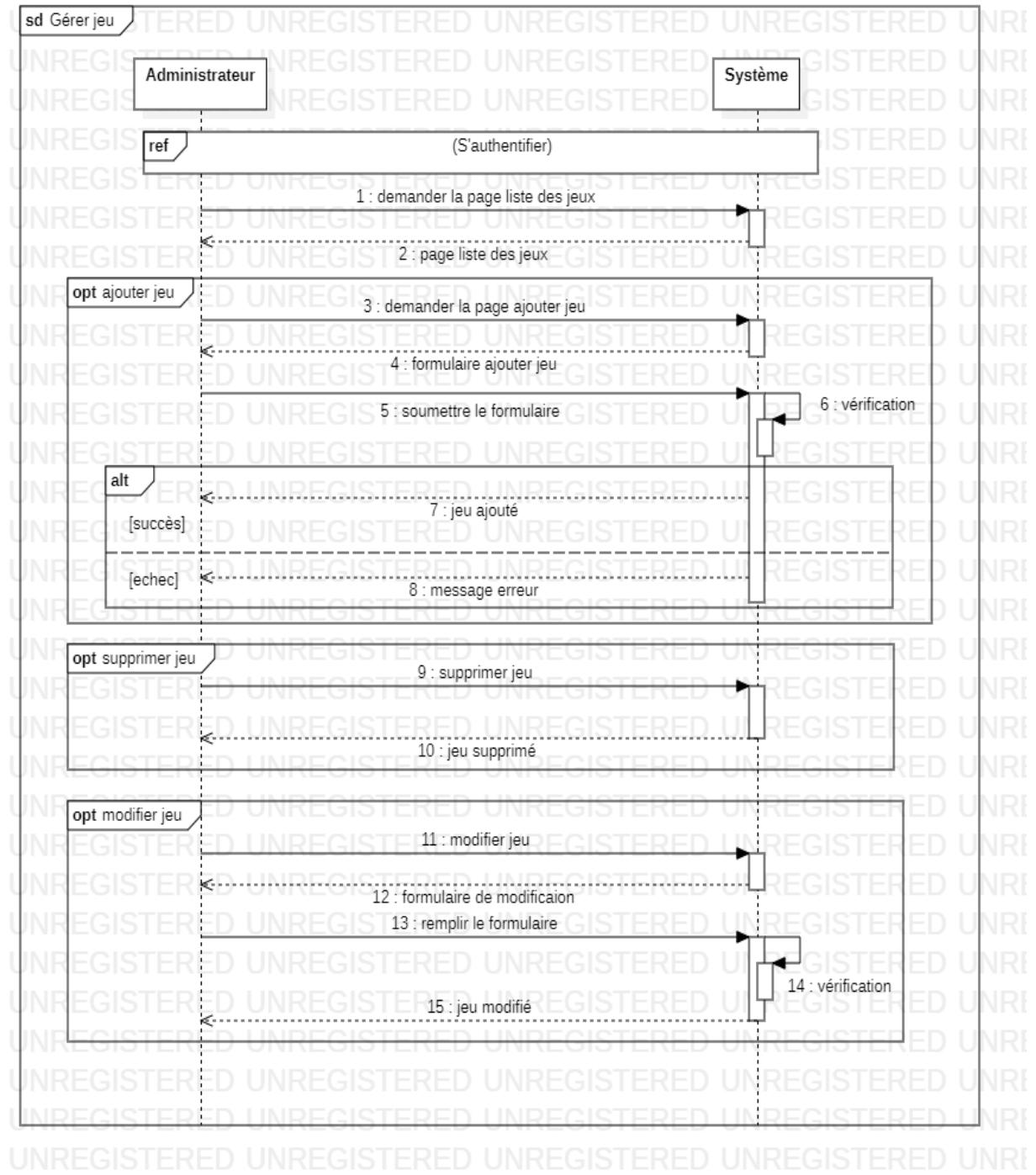


Figure 27 : Diagramme de séquence « Gérer jeux »

e) **Diagramme de séquence « Gérer utilisateur »**

Après l'authentification, l'administrateur demande la page liste des utilisateurs. Le système lui renvoie liste des utilisateurs, et l'administrateur a choisi d'ajouter, supprimer ou modifier un utilisateur. Enfin le système renvoi un message de confirmation ou un message d'erreur.

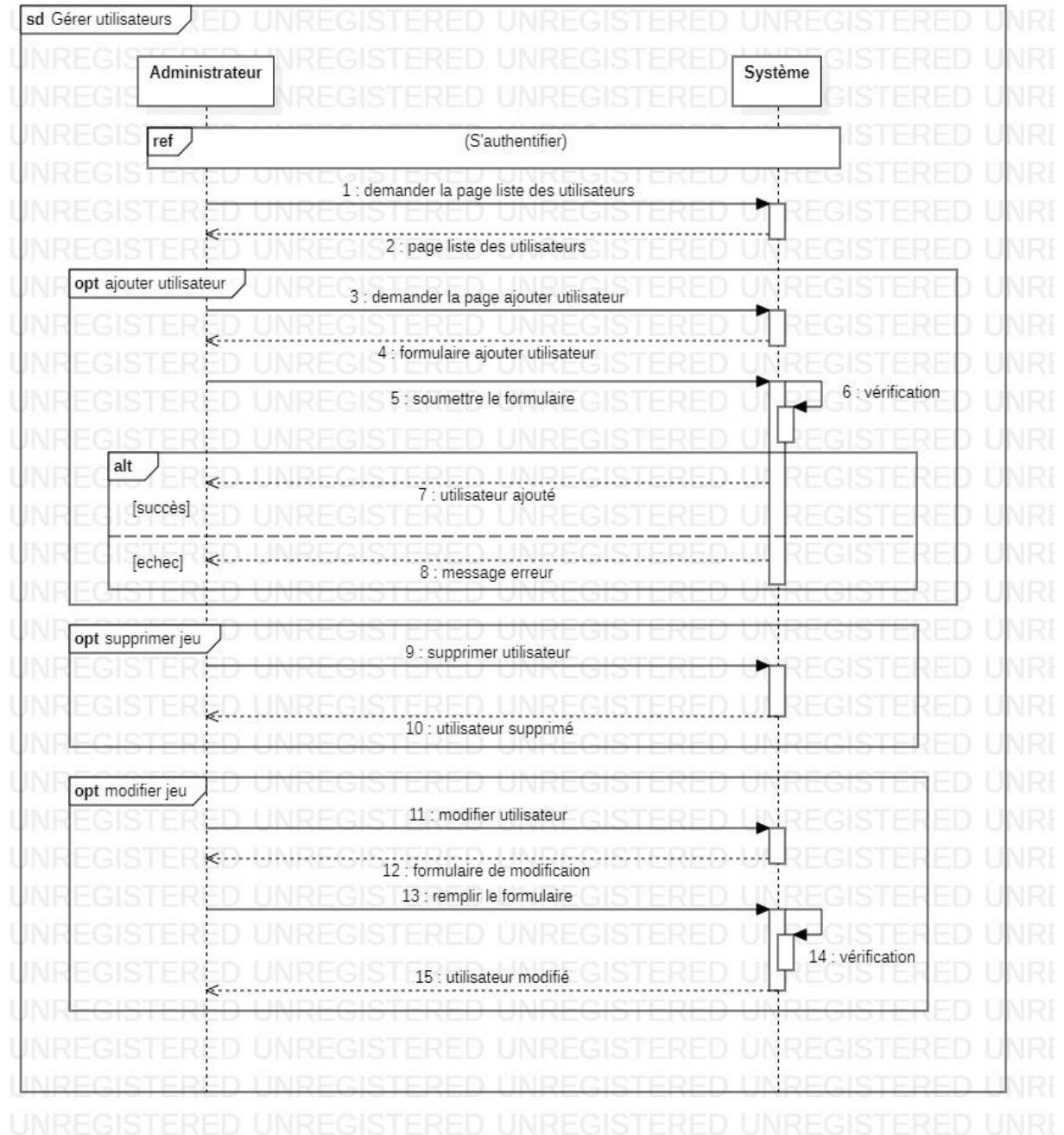


Figure 28 : Diagramme de séquence « Gérer utilisateurs »

## Conclusion

Dans ce deuxième chapitre on a étudié les différents types des acteurs de notre plateforme qui interagissent avec le système ainsi que les besoins fonctionnels et non fonctionnels pour finir par la définition de langage de modélisation UML et la représentation des différents cas d'utilisation des ceux fonctionnalités.

---

## CHAPITRE 3 : ANALYSE CONCEPTUELLE

---

## Introduction

Dans ce chapitre, nous aborderons la tâche la plus importante dans le développement de ce projet, nous expliquerons en détail toutes les étapes que nous avons suivies. Nous présentons, dans un premier temps, l'architecture générale de notre application afin d'en extraire les différents modules qui la composent, dans un deuxième temps nous expliquerons et analyserons la conception du logiciel à travers les diagrammes de séquence et les diagrammes de classes.

### I. Conception architecturale de l'application

#### 1. Architecture physique

De nombreuses applications fonctionnent dans un environnement client/serveur, ce qui signifie que les clients contactent un serveur selon une architecture spécifique. Il existe plusieurs types d'architectures. Pour bien détailler l'architecture de notre application, il est recommandé d'avoir une vue globale sur les différents architectures type existantes.

##### a) Présentation de l'architecture 1 tiers

Dans une approche d'application de type 1-tiers, les trois couches sont fortement et intimement liées, et s'exécutent sur la même machine. Dans ce cas, on ne peut pas parler d'architecture client-serveur mais d'informatique centralisée [16].

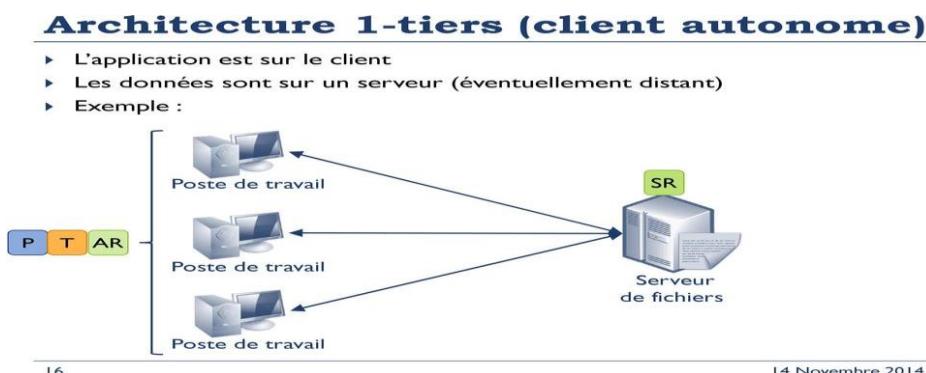


Figure 29 : Architecture 1-tiers [16]

##### b) Présentation de l'architecture 2 tiers

L'architecture à deux niveaux (aussi appelée architecture 2-tier) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service [17].

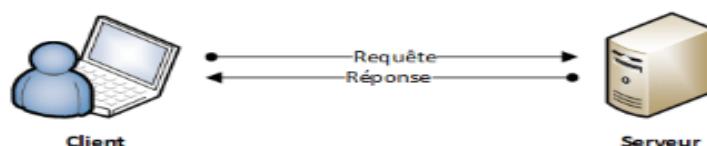


Figure 30 : Architecture 2-tiers [17]

c) **Présentation de l'architecture 3 tiers**

L'architecture 3-tier 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 :

- Couche de présentation ;
- Couche de traitement ;
- Couche d'accès aux données.

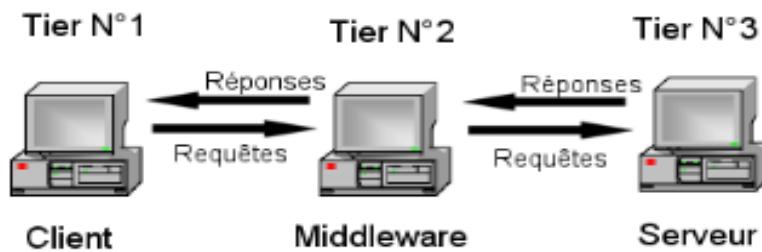


Figure 31 : Architecture 3-tiers [18]

d) **Présentation de l'architecture à plusieurs niveau (N tiers)**

L'architecture n-tiers a été pensée pour pallier aux limitations des architectures trois tiers et concevoir des applications puissantes et simples à maintenir. Ce type d'architecture permet de distribuer plus librement la logique applicative, ce qui facilite la répartition de la charge entre tous les niveaux [19].

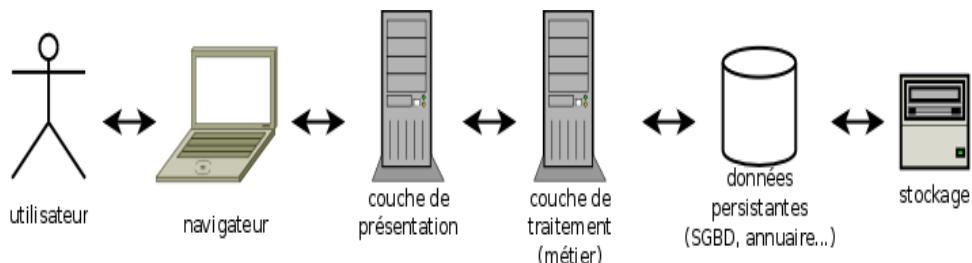


Figure 32 : Architecture n-tiers [19].

e) **Choix de l'architecture adéquate**

L'architecture 3-tiers qui se base sur l'architecture client-serveur a été choisi pour réaliser notre plateforme car chaque serveur effectue une tâche spécialisée ainsi que chaque serveur peut utiliser les services des autres serveurs afin d'effectuer son propre service.

Le schéma ci-dessous explique le fonctionnement de l'architecture 3-tiers :

- **Couche présentation** : correspond à l'affichage des données, c'est la partie visible par l'utilisateur, elle contient tous les composants graphiques de chabot ;

- **Couche traitement** : correspond à la partie fonctionnelle de projet, la couche traitement se compose du serveur qui est responsable de traiter les flux d'entrée et de sortie, avec un serveur action qui est responsable de traiter les actions ;
- **Couche d'accès aux données** : correspond à la partie gérant l'accès aux données de l'application.

Ce choix est justifié par le fait que ce type d'architecture :

- Les requêtes clients vers le serveur sont d'une plus grande flexibilité que dans celles de l'architecture 2-tiers ;
- La séparation qui existe entre le client, le serveur et le SGBD permet une spécialisation des développeurs sur chaque tiers de l'architecture
- Plus de flexibilité dans l'allocation des ressources.

Cette architecture 3-tiers est un modèle en couches présenté à la figure. Cette séparation a pour but de rendre indépendantes chacune des couches afin de faciliter la maintenance et les évolutions futures de l'application. Aussi, chaque couche communique seulement avec ses couches adjacentes (gauche et droite) et le flux traverse le système de couche en couche de manière continue.

## 2. Architecture logique

Pour bien détailler l'architecture logique de notre application, il est recommandé d'avoir une vue globale sur les différents architectures type existantes.

### a) *Architecture MVC (Model – View - Controller)*

L'architecture Modèle/Vue/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface. L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface ne soit pas la solution miracle, elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application [20].

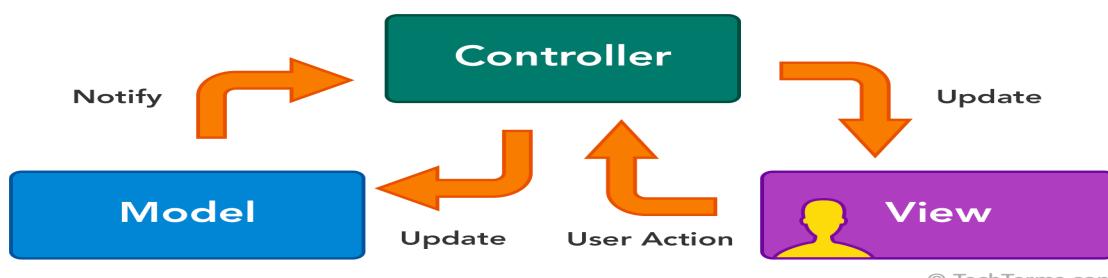


Figure 33 : Architecture MVC [20]

b) **Architecture MVP (Model – View – Presentation)**

Le modèle-vue-présentation (en abrégé MVP, de l'anglais model-view-presenter) est un patron d'architecture, considéré comme un dérivé du patron d'architecture modèle-vue-contrôleur. Il garde les mêmes principes que MVC sauf qu'il élimine l'interaction entre la vue et le modèle parce qu'elle sera effectuée par le biais de la présentation, qui organise les données à afficher dans la vue [21].

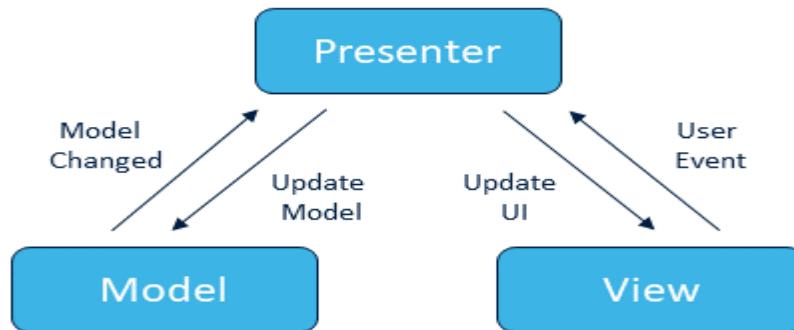


Figure 34 : Architecture MVP [21]

c) **Architecture MVVM (Model – View – View – Model)**

MVVM est originaire de Microsoft et adapté pour le développement des applications basées sur les technologies Windows Présentation Fondation<sup>10</sup> et Silverlight<sup>11</sup> via l'outil MVVM Light par exemple. Cette méthode permet, tel le modèle MVC (modèle-vue-contrôleur), de séparer la vue de la logique et de l'accès aux données en accentuant les principes de liaison et d'événement [22].

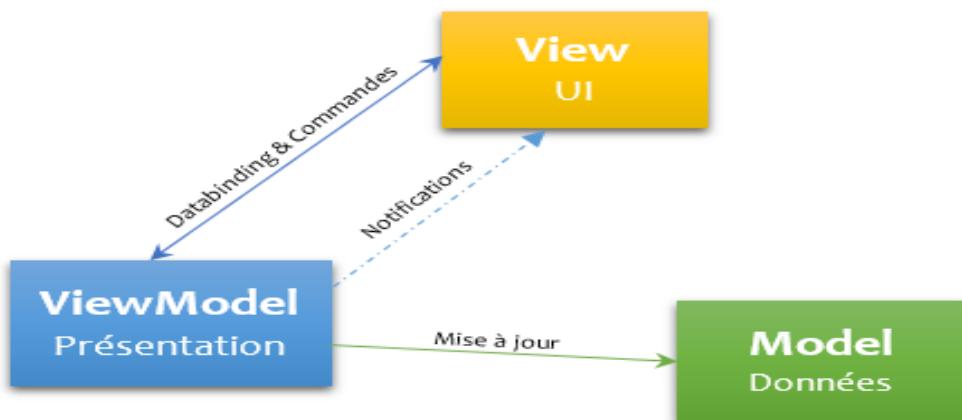


Figure 35 : Architecture MVVM [22].

---

10 : [https://fr.wikipedia.org/wiki/Windows\\_Presentation\\_Foundation](https://fr.wikipedia.org/wiki/Windows_Presentation_Foundation)

11 <https://fr.wikipedia.org/wiki/Silverlight>

#### d) Choix de l'architecture de l'application

Nous optons pour l'architecture MVC (Model View Controller) comme une architecture logique de notre système. Elle consiste à distinguer trois entités distinctes qui sont le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface graphique de programme. L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface mais aussi elle offre un cadre pour structurer une application.

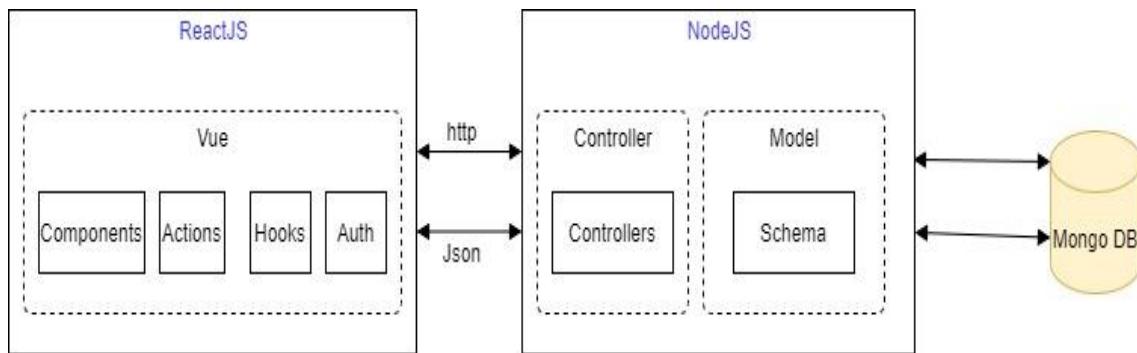


Figure 36 : Architecture de l'application (MVC)

Dans l'architecture MVC, les rôles des trois entités sont les suivants :

- Le Modèle (M) : C'est la couche (Bibliothèque de fonctions) qui accède à la base de données ;

models	
JS	Categorie.js
JS	Competance.js
JS	Jeux1.js
JS	Jeux2.js
JS	Post.js
JS	Profile.js
JS	User.js

Figure 37 : Interface couche Modèle

- La Vue (V) : C'est la couche qui représente l'interface utilisateur. Elle assure l'affichage des données de la couche Modèle et la réception de toutes les actions de l'utilisateur pour les transmettre à la couche Contrôleur ;

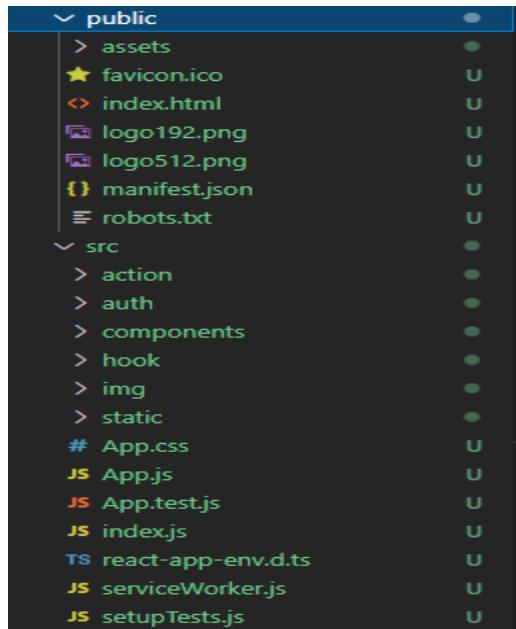


Figure 38 : Interface couche Vue

- Le Contrôleur (C) : C'est la couche responsable à écouter les actions de l'utilisateur, les analyser et les traiter convenablement.

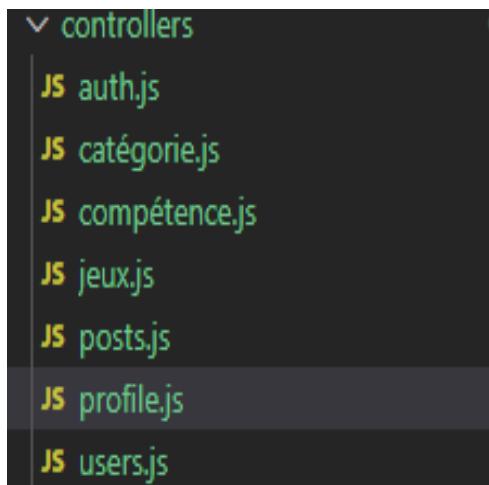


Figure 39 : Interface couche Contrôleur

## II. Architecture d'api et de sécurité

Une API (Application Programming Interface) consiste à rendre visible les données et les fonctionnalités d'une autre application afin d'être utilisé par des autres applications, d'autre part Pour comprendre le fonctionnement de sécurité d'Api il est nécessaire de nous comprendre la protocole JWT

## 1. Architecture d'API adapté (API REST)

L'api Restful est parmi les api (application program interface) les plus utilisées dans les sites web comme (Amazon, Google, LinkedIn, Twitter), avec un protocole de transfert de données typiques, tels que SOAP (Simple Object Access Protocol), offre des grands termes de sécurité et d'intégrité des données.

Rest exploité les méthodes http et utilisé les méthodes (Get, Post, Update, Delete),

- **Get** : pour récupérer des données ;
- **Post** : pour créer des données ;
- **Update** : pour modifier le comportement des données ;
- **Delete** : pour effacer des données.

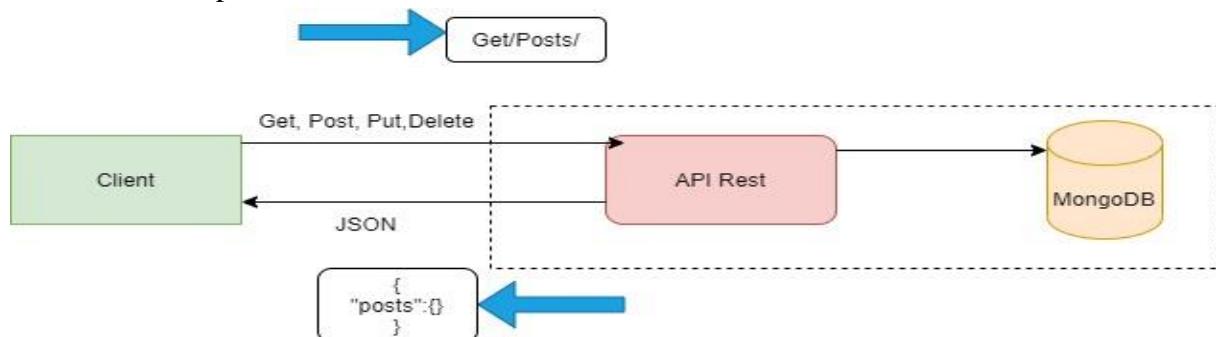


Figure 40 : Architecture d'API Rest

## 2. Protocole d'authentification JWT (Json Web Token)

**JWT** : Abréviation de **JSON Web Token** est un standard ouvert défini dans la RFC 75191. Il permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité des données à l'aide d'une signature numérique Elle s'effectue par l'algorithme HMAC ou RSA [23].

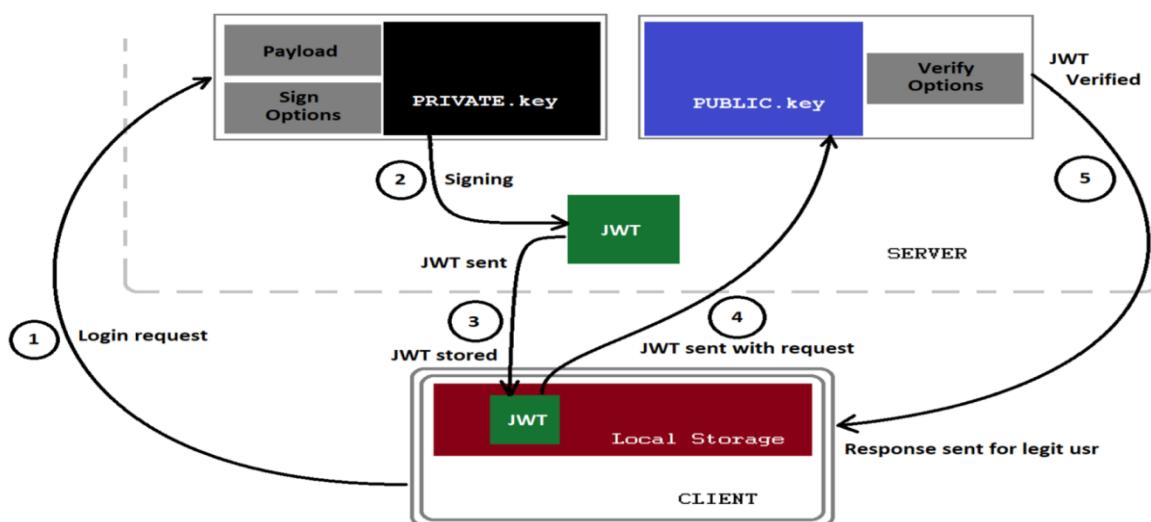


Figure 41 : Protocole d'authentification JWT [23]

Un jeton se compose de trois parties :

- Un en-tête (header), utilisé pour décrire le jeton. Il s'agit d'un objet JSON ;
- Une charge utile (payload) qui représente les informations embarquées dans le jeton. Il s'agit également d'un objet JSON ;
- Une signature numérique.

### 3. Conception préliminaire des interfaces – prototypes

Une maquette est un produit jetable donnant aux utilisateurs une vue concrète mais non définitive de la future interface de l'application. Elle est développée rapidement afin d'améliorer la relation développeur-client. Les maquettes de notre application sont les suivantes :

Cette figure représente la maquette de page d'accueil.



Figure 42 : Maquette préliminaire de l'interface d'accueil

Cette maquette représente la page d'authentification.

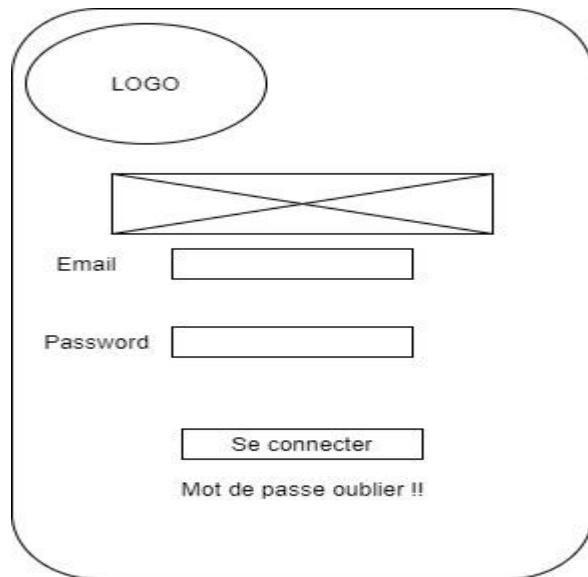


Figure 43: Maquette préliminaire de l'interface d'authentification

Cette maquette représente la page administrateur.

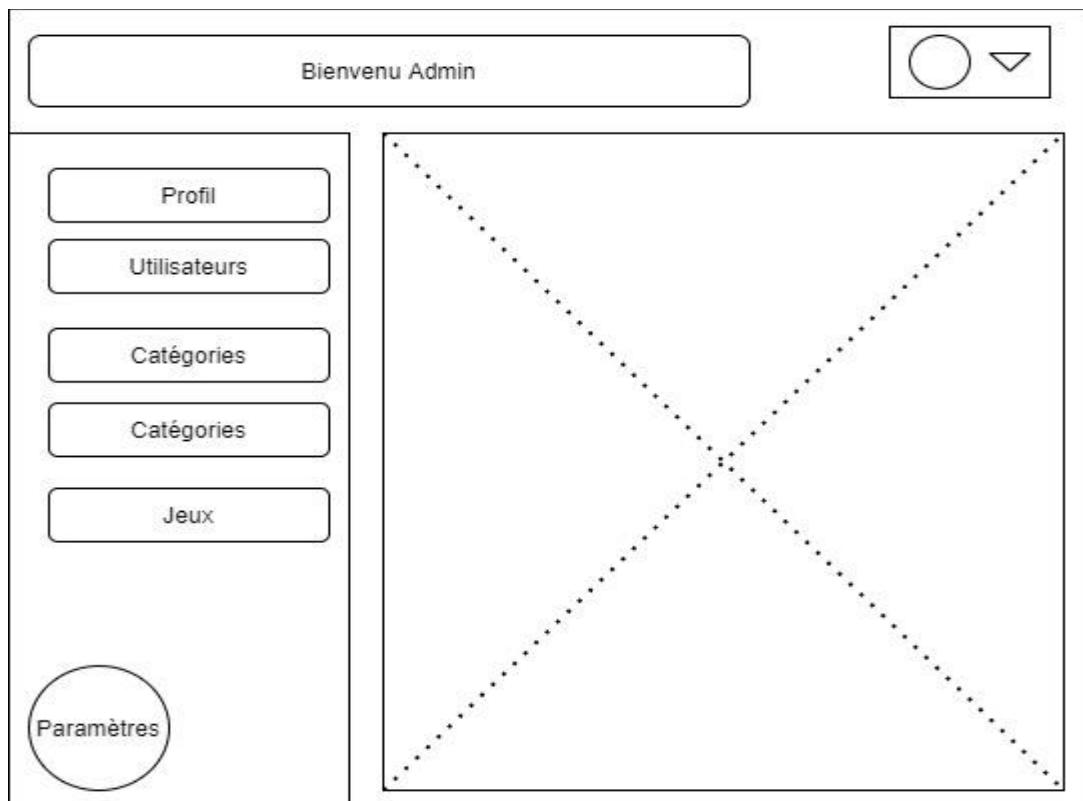


Figure 44 : Maquette préliminaire de l'interface administrateur

### III. Conception de la base de données

#### 1. Modèle conceptuel des données (MCD)

Le modèle conceptuel de données a pour but de modéliser d'une façon compréhensible les données mémorisées dans le système d'information sans aborder les aspects d'organisation et techniques, d'où la structure d'un MCD ne dépend pas de système de gestion de base de données particulier.

La structure du modèle entité-association s'appuie sur trois concepts de base :

- **L'entité** : une entité est la représentation d'un Object matériel ou immatériel ;
- **Association** : une association est une relation que les règles de gestion établissent entre deux entités ou plus ;
- **Propriété** : une propriété est une donnée élémentaire qu'elle apparaître sur l'entité.  
Pour notre système nous avons identifié les entités suivantes :
  - **Utilisateur** : Représente tous les utilisateurs de la plateforme. ;
  - **Catégorie** : Représente une collection de compétences dans une classe particulière d'utilisateurs ;
  - **Compétence** : Représente une collection des différents jeux ;
  - **Jeux** : représente un jeu d'un algorithme bien défini dans la plateforme.
  - **IllustrationImage** hérite de **Jeux** : représente un algorithme de jeu avec une seule image ;
  - **IllustrationImages** hérite de **Jeux** : représente un algorithme de jeu avec plusieurs images ;
  - **Drag&Drop** hérite de **Jeux** : représente un algorithme de jeu de manière glissé déposé ;
  - **Forum** : Représente un forum de discussion dans notre application ;
  - **Post** : Représente un poste dans un formulaire, cette entité possède une collection des commentaires ;
  - **Commentaire** : Représente un commentaire poster par les utilisateurs sur un poste quelconque ;
  - **Parent** hérite de **Utilisateurs** : Représente un parent d'un apprenant ;
  - **Notification Parent** : Représente la notification envoyée par le système vers le parent de l'apprenant ;
  - **Paiement** : Représente le paiement d'inscrire au notre plateforme ;
  - **PayPal** hérite de **Paiement** : Représente les détails du paiement par PayPal ;
  - **MasterCard** : Représente les détails du paiement par Master Card.

Le modèle conceptuel de données représenté dans la figure ci-dessous illustre toutes les entités mentionnées ci-dessus :

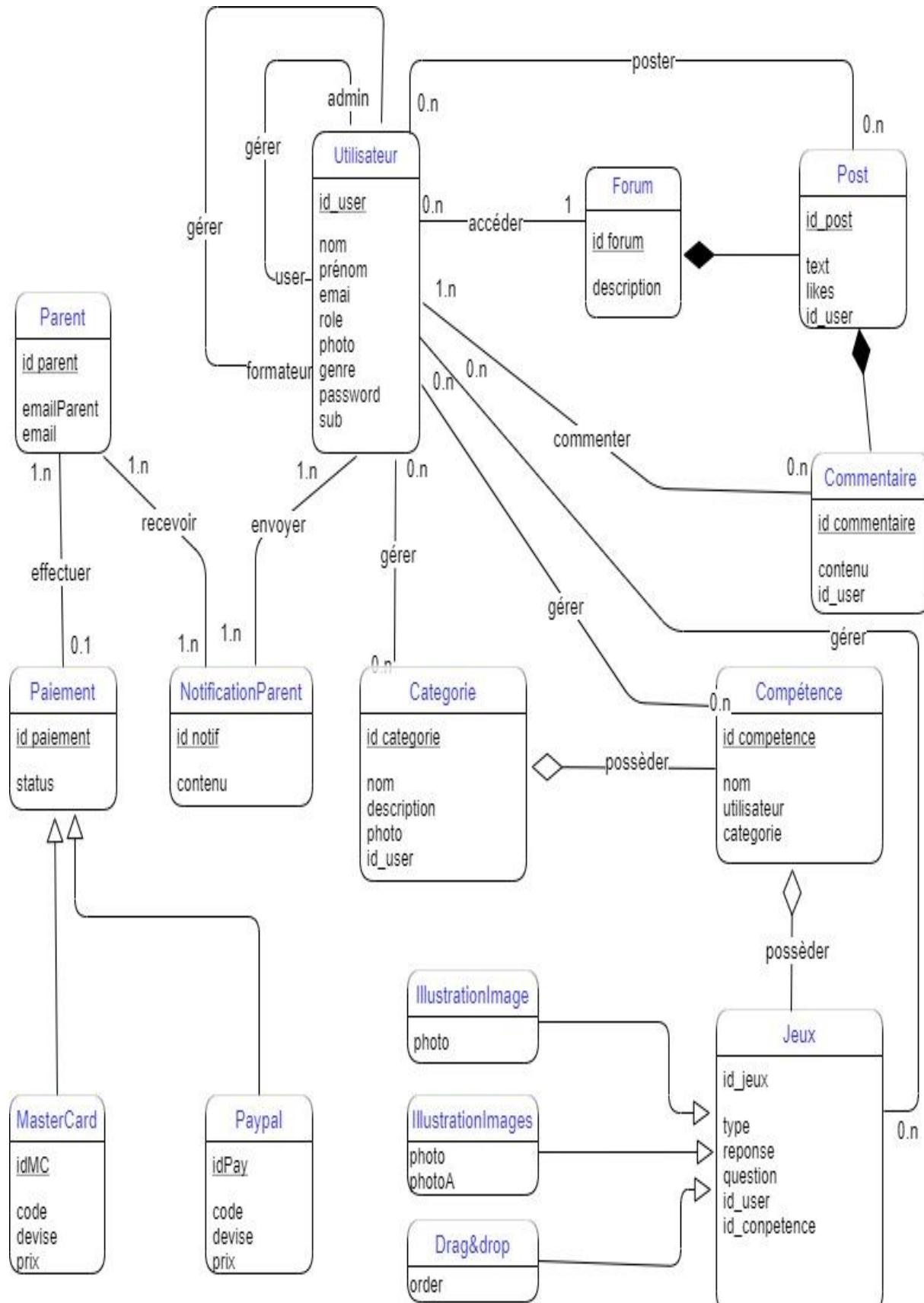


Figure 45 : Modèle Conceptuel des données

## 2. Dictionnaire de données

Le dictionnaire des données est une collection de métadonnées qui regroupe les données que nous aurons conservées dans notre base de données. Nous présentons ci-dessous le dictionnaire de données relatif à notre application.

Données	Désignation	Type	Longueur	Clé primaire
<b>Utilisateur</b>				
Id_user	Identifiant de l'utilisateur	Chaine	50	✓
nom	Nom de l'utilisateur	Chaine	20	
prénom	Prénom de l'utilisateur	Chaine	20	
email	Email de l'utilisateur	Chaine	50	
role	Rôle de l'utilisateur	Enum	20	
photo	Photo de l'utilisateur	Chaine	50	
genre	Genre de l'utilisateur	Chaine	10	
password	Le mot de passe de l'utilisateur	Chaine	30	
sub	Le type de paiement (mensuel/annuel)	Chaine	10	
<b>Parent</b>				
Id_parent	Identifiant d'un parent	Chaine	50	✓
emailParent	Email d'un parent	Chaine	50	
email	Email de son apprenant	Chaine	50	
<b>NotifcationParent</b>				
Id_notif	Identifiant d'une notification de parent	Chaine	50	✓
contenu	Contenu d'une notification	Chaine	200	
<b>Paiement</b>				
Id_paiement	Identifiant d'un paiement	Chaine	50	✓
status	Statut d'un paiement	Chaine	50	
<b>MasterCard</b>				
IdMC	Identifiant d'un master card	Chaine	50	✓
code	Code de master card	Entier	20	
devise	Devise de master card	Chaine	10	
prix	Prix de paiement avec master card	Entier	10	
<b>Paypal</b>				
idPay	Identifiant d'un paypalcard	Chaine	50	✓
code	Code de paypal	Entier	20	
devise	Devise de paypal	Chaine	10	
prix	Prix de paiement avec paypal	Entier	10	
<b>Catégorie</b>				
Id_categorie	Identifiant d'une catégorie	Chaine	50	✓

nom	Nom d'une catégorie	Chaine	20	
description	Description d'une catégorie	Chaine	100	
photo	Photo d'une catégorie	Chaine	50	
<b>Compétence</b>				
Id_competence	Identifiant d'une compétence	Chaine	50	✓
nom	Nom d'une compétence	Chaine	20	
<b>Jeux</b>				
Id_jeux	Identifiant d'un jeu	Chaine	50	✓
type	Type d'un jeu	Chaine	10	
reponse	Réponse valide d'un jeu	Entier	10	
question	Question d'un jeu	Chaine	100	
<b>IllustrationImage</b>				
photo	Photo de jeux 1	Chaine	50	
<b>IllustrationImages</b>				
Photo	Photo de jeux 2	Chaine	50	
photoA	Tableau des photos de jeux 2	Array		
<b>Drag&amp;drop</b>				
Order	Ordre des réponses	String	20	
<b>Forum</b>				
Id_forum	Identifiant d'un forum	Chaine	50	✓
description	Description d'un forum	Chaine	100	
<b>Post</b>				
Id_post	Identifiant d'un poste	Chaine	50	✓
text	Text d'un poste	Chaine	200	
like	Like d'un poste	Chaine	20	
<b>Commentaire</b>				
Id_commentaire	Identifiant d'un commentaire	Chaine	50	✓
contenu	Contenu d'un commentaire	Chaine	100	

Tableau 8 : Dictionnaire de données

### 3. Modèle logique de données (MLD)

Le modèle logique des données consiste à décrire la structure de données utilisée sans faire référence à un langage de programmation. Il s'agit donc de préciser le type de données utilisé lors des traitements.

La figure ci-dessous présente le diagramme « MLD » de notre application.

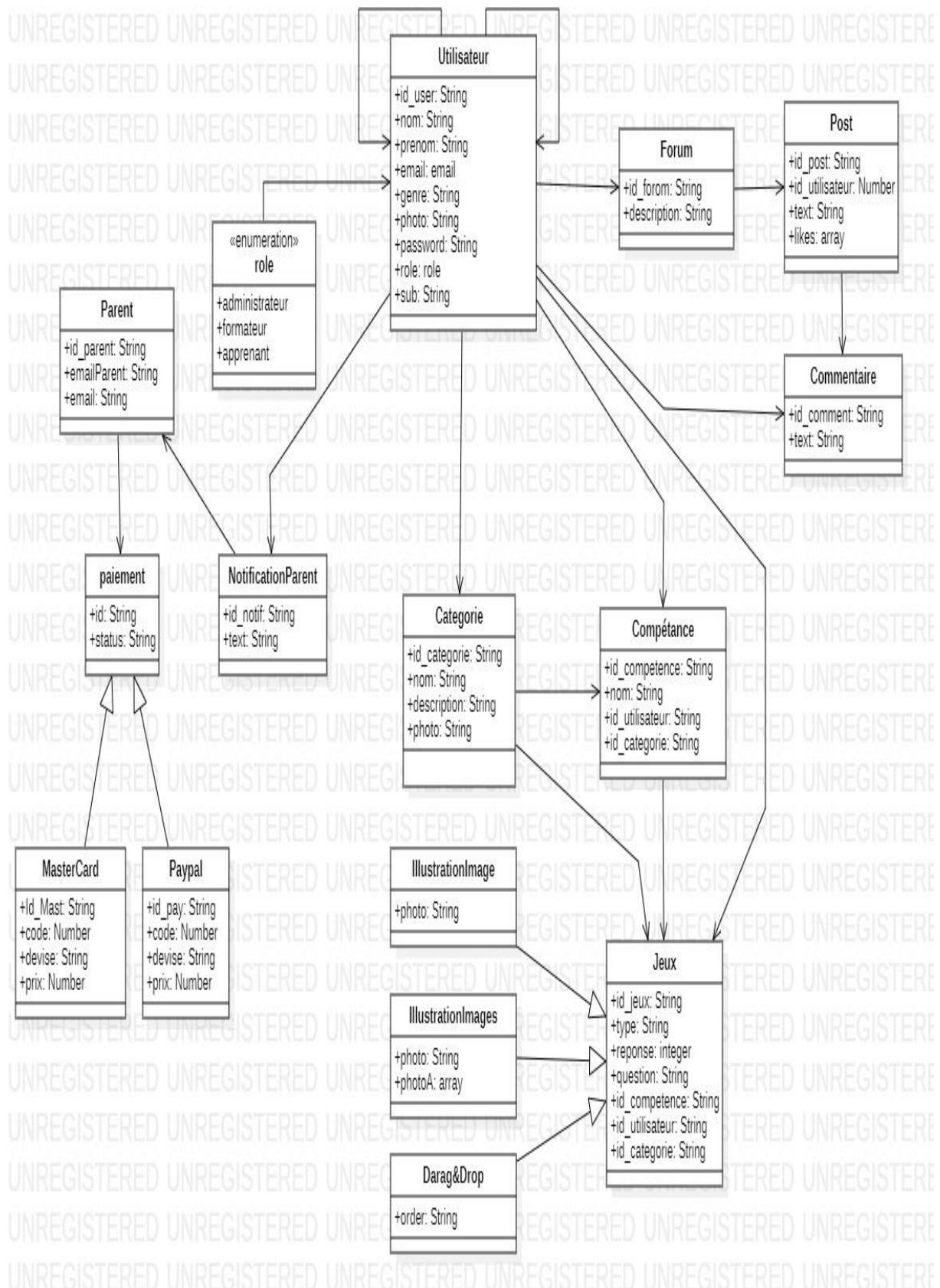


Figure 46 : Modèle logique de données

## IV. Conception logicielle

Après avoir élaboré la conception de la base de données, nous passons à la conception des fonctionnalités du système en utilisant le langage UML tout en respectant les contraintes du modèle MVC.

Le diagramme de notre système est divisé en 3 parties :

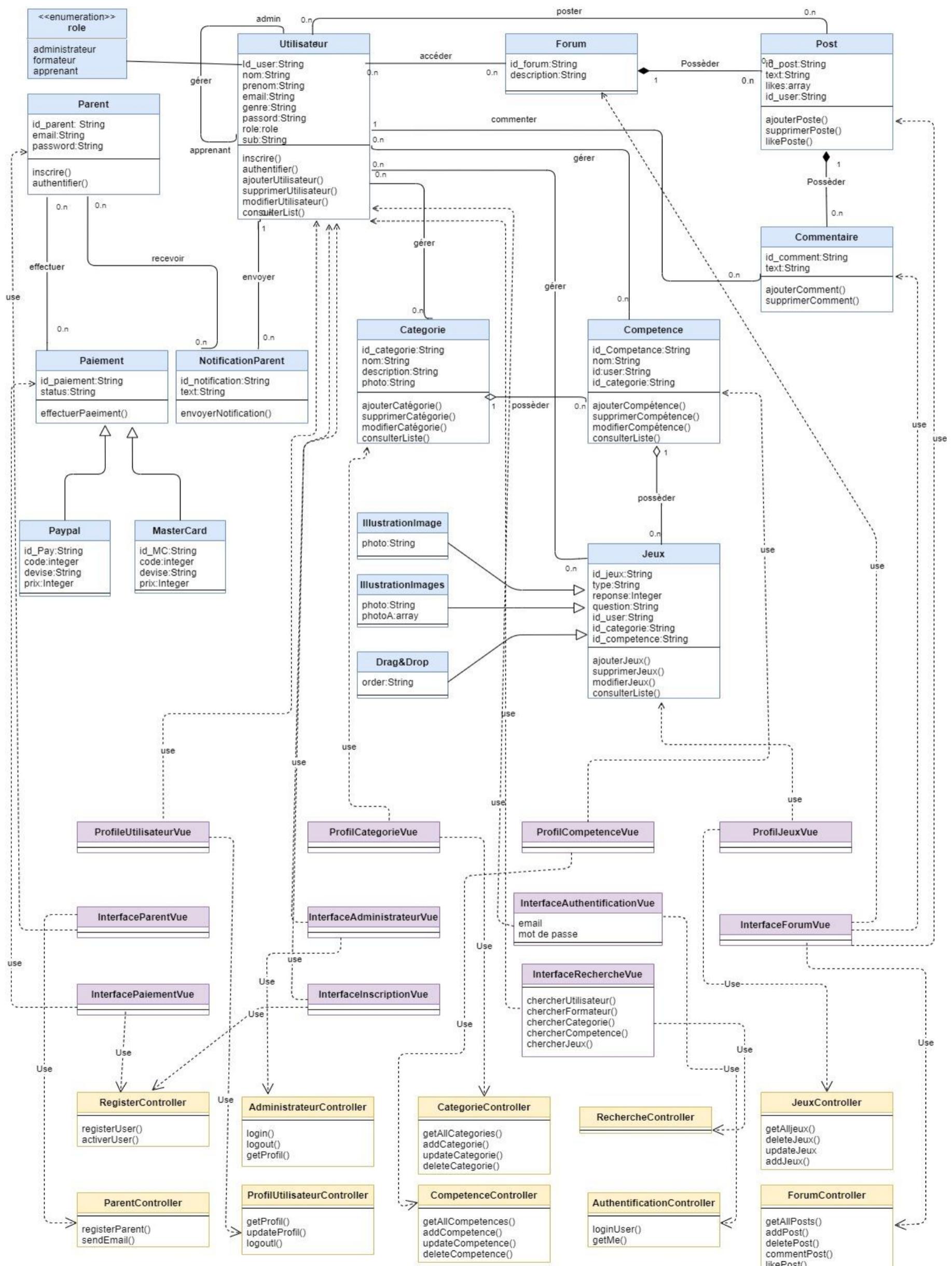
- Modèle : Cette partie permet l'accès à la base de données pour récupérer ou stocker les objets ;
- Vue : Le développement de l'interface avec ReactJS déplace la partie Vue vers le côté du client ;
- Contrôleur : Cette partie est responsable du traitement des requêtes des utilisateurs.

### 1. Diagramme de classe

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



Figure 47 : diagramme de classe



## 2. Diagramme de séquence

Le diagramme de séquences présenté par le langage de modélisation UML est une méthode pour modéliser les aspects dynamiques du système en exprimant les interactions entre les acteurs ou comment plusieurs objets fonctionnent ensemble dans un ordre chronologique spécifique.

Pour cela on va présenter les diagrammes des séquences de quelques scénarios entre les acteurs du système.

### a) *Diagramme de séquence « S'authentifier »*

L'utilisateur saisit son email et mot de passe dans le formulaire puis clique sur le bouton « Connexion ». La Vue émet la requête au Contrôleur, le Contrôleur envoie les données à la base de données pour la vérification, si la vérification est terminée avec succès l'interface accueil est affichée sinon un message d'erreur est renvoyé et l'interface ressayer est affichée.

La figure ci-dessous indique le diagramme de séquences du scénario d'authentification :

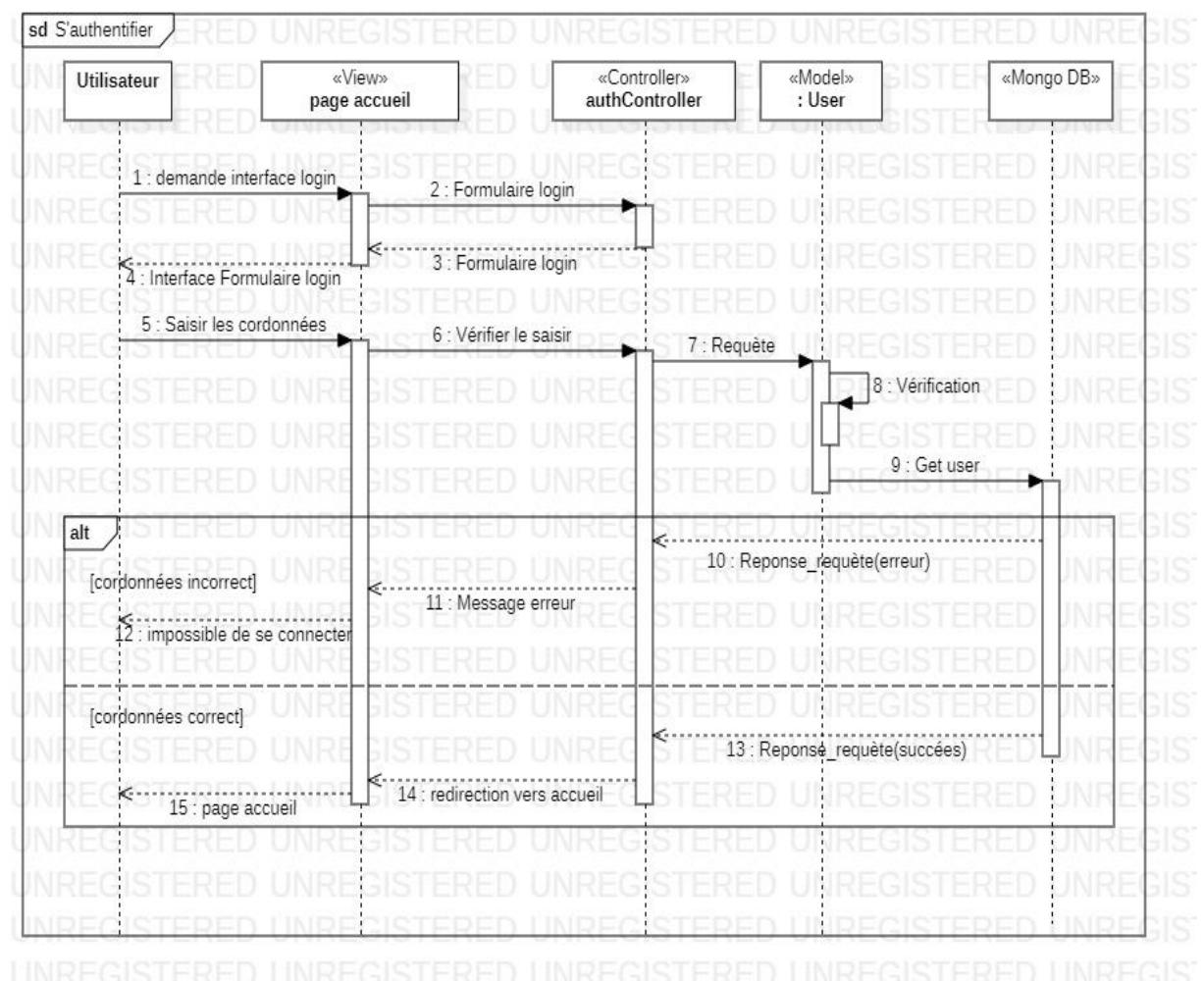


Figure 48 : Figure de diagramme de séquence « S'authentifier »

b) *Diagramme de séquence « S'inscrire »*

Pour utiliser notre site le visiteur doit commencer par remplir les informations nécessaires dans un formulaire. Dès qu'il clique le bouton « Soumettre », le système vérifie les champs saisis et le renvoi la page d'accueil à l'utilisateur si son email est valide sinon, un message d'erreur est affiché et l'interface est renvoyée à l'utilisateur.

La figure suivante indique le diagramme de séquences du scenario de création du compte :

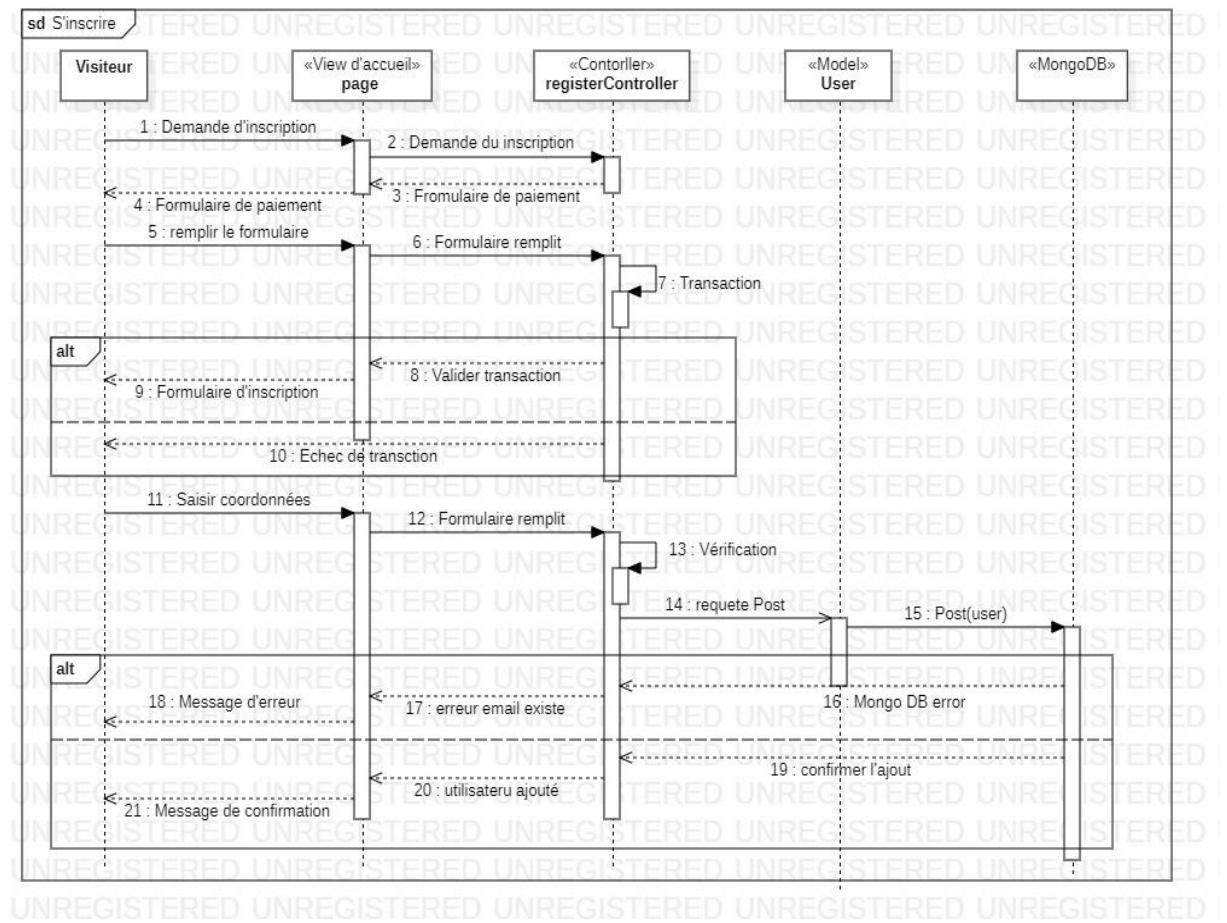


Figure 49 : Figure de diagramme de séquence « S'inscrire »

c) *Diagramme de séquence « Gérer Jeux »*

La figure ci-dessous présente le diagramme de séquence relatif aux scénarios possibles pour la gestion des jeux. Les options figurants dans le diagramme ci-dessous sont :

- Consulter la liste des jeux ;
- Ajouter jeu ;
- Modifier jeu ;
- Supprimer jeu.

## Chapitre 3 : Analyse conceptuelle

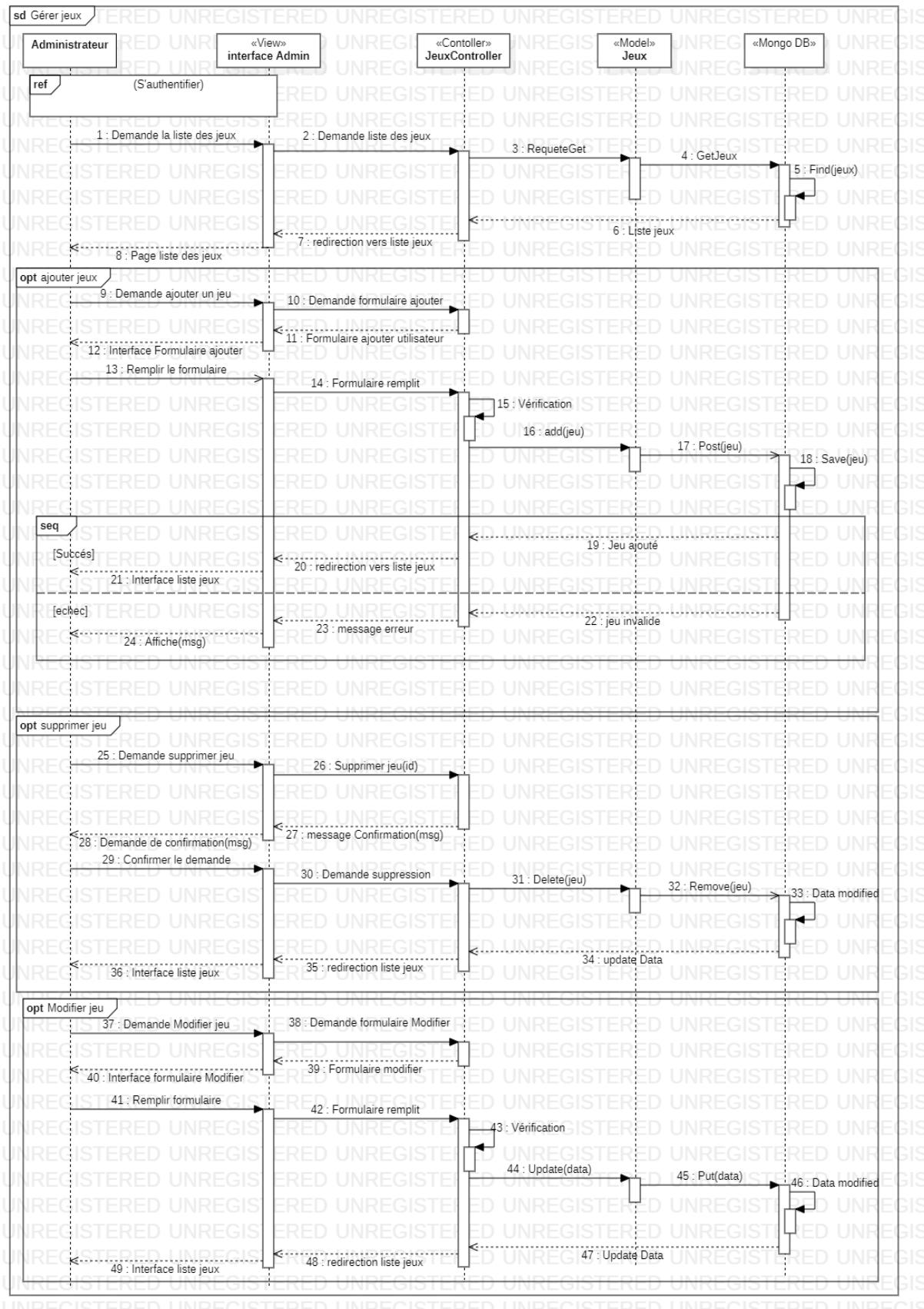


Figure 50: Figure de diagramme de séquence « Gérer jeux »

d) **Diagramme de séquence « Gérer utilisateurs »**

La figure ci-dessous présente le diagramme de séquence relatif aux scénarios possibles pour la gestion des jeux. Les options figurants dans le diagramme ci-dessous sont :

- Consulter la liste des utilisateurs ;
- Ajouter utilisateurs ;
- Modifier utilisateurs ;
- Supprimer utilisateur.

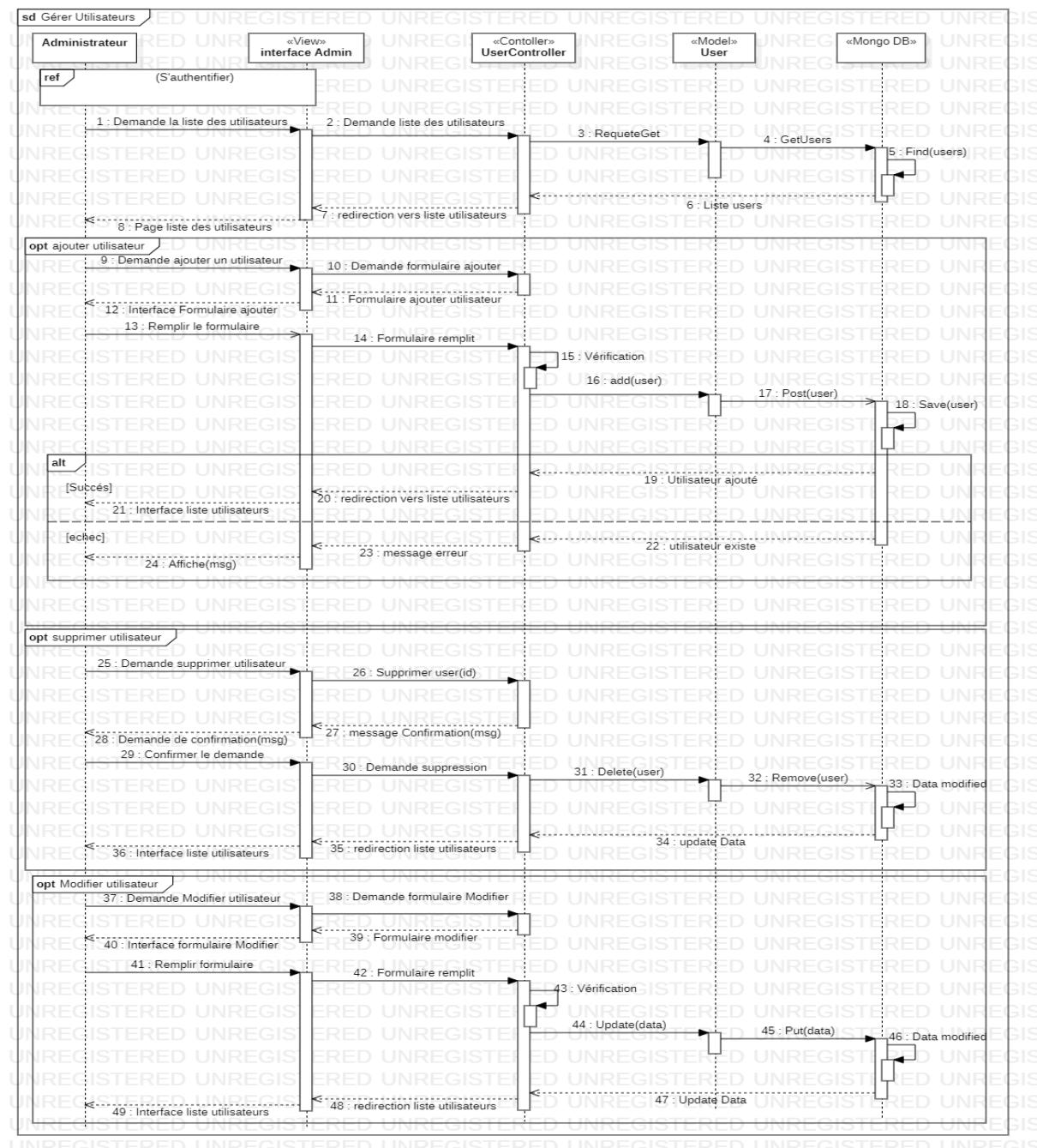


Figure 51 : Figure de diagramme de séquence « Gérer utilisateurs »

### 3. Diagramme d'activité

Dans UML, un diagramme d'activité est utilisé pour afficher la séquence des activités. Les diagrammes d'activité représentent le flux de travail à partir d'un point de départ au point d'arrivée, détaillant les nombreux sentiers de décision, qui existent dans la progression des événements contenus dans l'activité. Ils peuvent être utilisés à des situations de détail, où le traitement parallèle peut survenir dans l'exécution de certaines activités. Les diagrammes d'activités sont utiles pour la modélisation d'entreprise où ils sont utilisés pour détailler les processus impliqués dans des activités commerciales [24].

#### a) *Diagramme d'activité « S'authentifier »*

Le processus concerne les acteurs suivants :

- **L'utilisateur** : Accéder à la page d'accueil de la plateforme et saisir son email et son mot de passe, ou créer un compte s'il n'est pas déjà titulaire d'un compte ;
- **Le système** : Le système ouvre l'application et affiche le formulaire d'authentification, puis vérifie les coordonnées saisies par l'utilisateur et affiche la page correspondante selon le type d'utilisateur. Si les coordonnées sont invalides, un message d'erreur s'affiche.

La figure ci-dessous présente le diagramme d'activité pour modéliser le processus d'authentification.

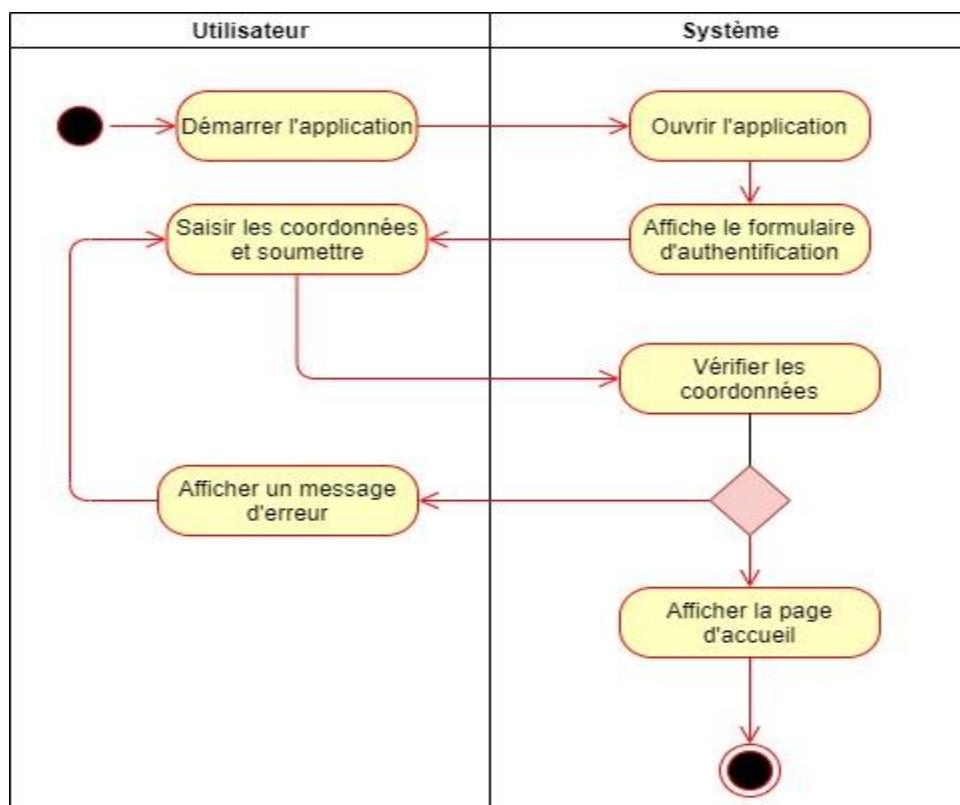


Figure 52 : Diagramme d'activité « s'authentifier »

b) **Diagramme d'activité « Consulter les statistiques de son apprenant »**

Le processus concerne les acteurs suivants :

- **Le parent** : Accéder à la page d'accueil de la plateforme et saisir son email, l'email de son apprenant et le mot de passe de son apprenant ;
- **Le système** : Ouvre l'application et affiche le formulaire d'authentification, puis vérifie les coordonnées saisies par le parent. Si les coordonnées sont valides il envoie les statistiques de l'apprenant par mail vers l'email saisi par le parent, sinon il affiche un message d'erreur.

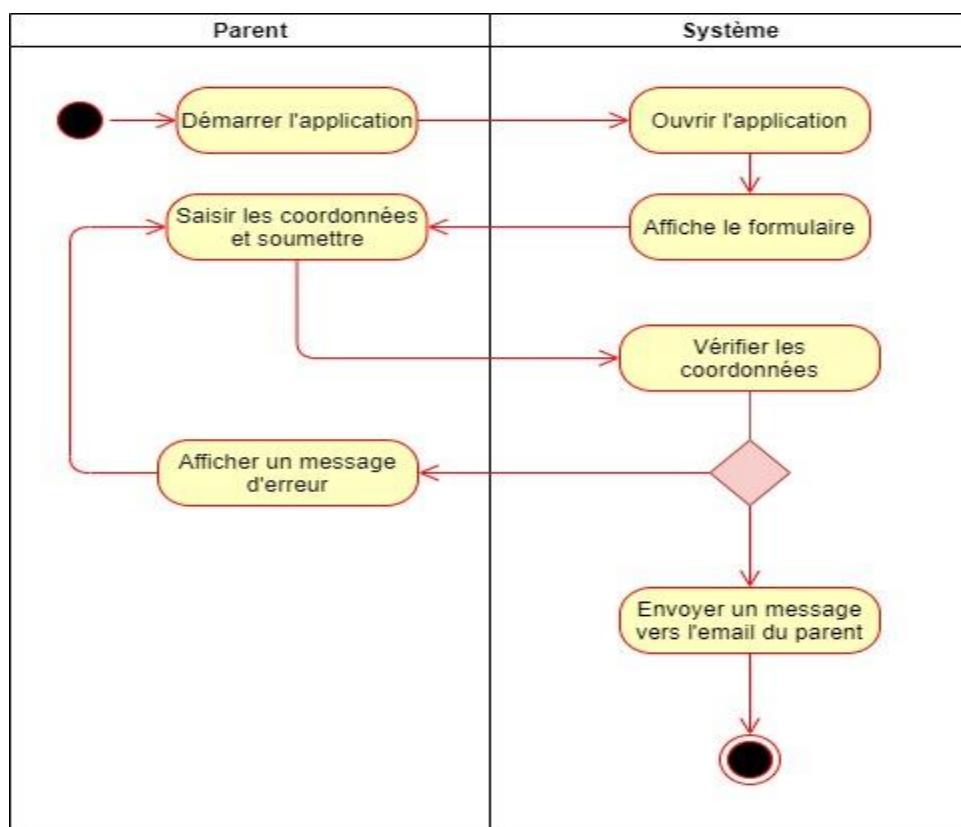


Figure 53: Diagramme d'activité « Consulter les statistiques de son apprenant »

#### 4. Diagramme états-transitions

Un diagramme états-transitions est un schéma utilisé en génie logiciel pour représenter des automates déterministes. Il fait partie du modèle UML et s'inspire principalement du formalisme des statecharts et rappelle les grafcets des automates. S'ils ne permettent pas de comprendre globalement le fonctionnement du système, ils sont directement transposables en algorithme. En effet, contrairement au diagramme d'activité qui aborde le système d'un point de vue global, le diagramme états-transitions cible un objet unique du système. Tous les automates d'un système s'exécutent parallèlement et peuvent donc changer d'état de façon indépendante [25].

a) **Diagramme d'état-transition « Suivre trois jeux par jour »**

Un visiteur ne peut traiter que trois jeux par jour. Si un visiteur demande à traiter plus de trois jeux, le système le bloquera et lui demandera de s'inscrire.

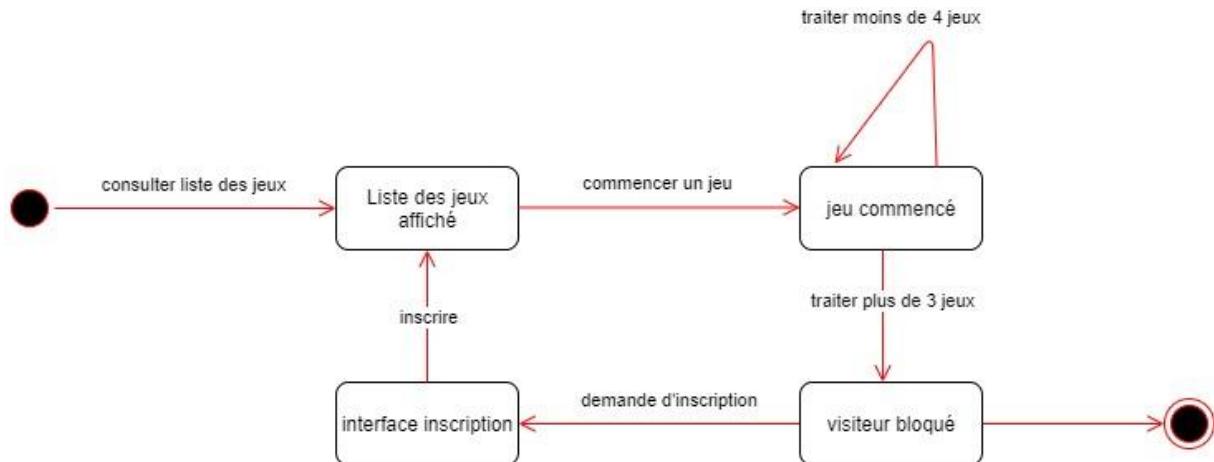


Figure 54 : Diagramme états-transitions « Suivre trois jeux par jour »

b) **Diagramme d'état-transition « paiement »**

Pour s'inscrire, l'utilisateur doit payer l'abonnement. Il doit choisir son type de paiement (annuel ou mensuel) et saisir son code carte (paypal, master card...) si le code est valide la transaction est effectuée et il se rend sur la page d'inscription sinon il doit resaisir un autre code Diagramme de composant.

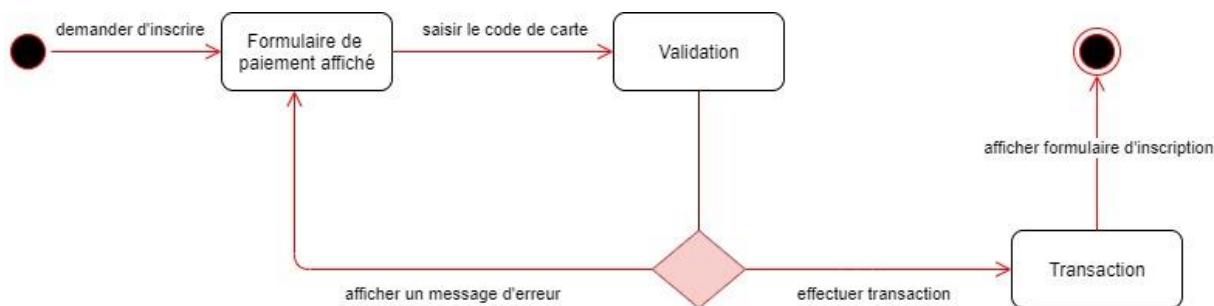


Figure 55 : Diagramme états-transitions « paiement »

## 5. Diagramme de composant

Dans le langage UML, un diagramme de composants décrit la manière dont les composants sont câblés ensemble pour former des composants plus importants ou des systèmes logiciels. Ils sont utilisés pour illustrer la structure de systèmes arbitrairement complexes.

Dans notre cas, l'interface web communique avec l'API REST à travers des requêtes http. D'autre part, l'API REST communique avec le serveur de la base des données à travers le serveur d'application.

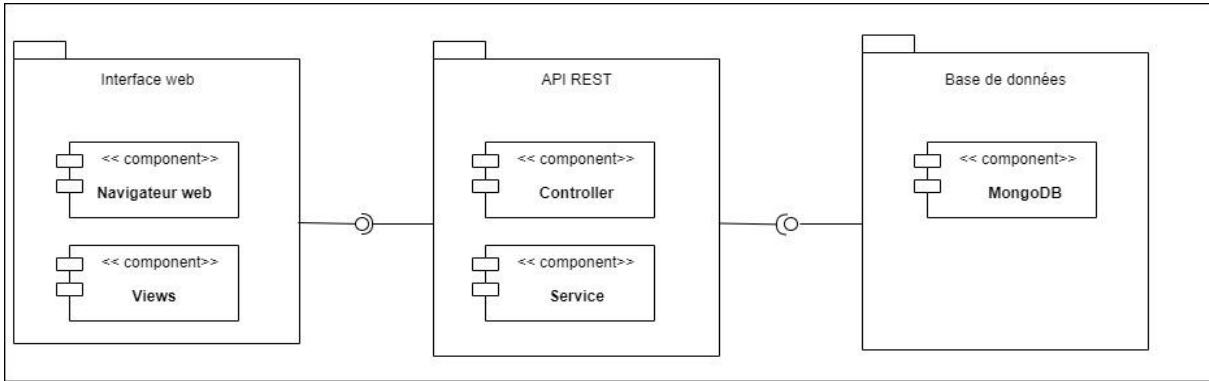


Figure 56 : Diagramme de composant

## Conclusion

A travers ce chapitre, nous avons présenté la conception de notre application. Nous avons fourni, tout d'abord, une conception globale de l'organisation de notre système. Ensuite, nous avons présenté la conception détaillée de l'application en choisissant le langage de modélisation UML pour exploiter ses diagrammes et analyser les aspects statiques et dynamiques de notre projet. Parmi les diagrammes de langage de modélisation UML, on a utilisé le diagramme de séquence et le diagramme de classe. Nous avons ensuite illustré les besoins techniques en expliquant l'architecture 3-tiers. A présent, nous sommes capables d'entamer la partie réalisation.

---

## CHAPITRE 4 : RÉALISATION

---

## Introduction :

A la suite de conception déjà vu dans le chapitre précédent, nous sommes arrivés à la dernière étape du rapport pour objectif d'exposer le travail achevé. Dans un premier temps, nous allons présenter l'environnement matériels et logiciels supposant notre application. Ensuite nous allons passer en revue les différentes tâches réalisées à travers quelques interfaces homme-machine et un chronogramme récapitulatif qui décrit toutes étapes de mise en œuvre de notre système.

## I. Environnements de travail

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

### 1. Environnement matériel

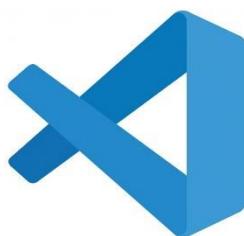
Cette plateforme a été réalisée par deux postes de travail ayant les caractéristiques suivantes :

- Premier poste :
  - Système d'exploitation : Windows 10 ;
  - Disque dur : 1 TO ;
  - Ram : 16 GO.
  - Processeur : Intel® Core™ i7-6500U CPU 2.5GHz.
- Deuxième poste :
  - Système d'exploitation : Windows 10 ;
  - Disque dur : 1 TO ;
  - Ram : 8 GO ;
  - Processeur : Intel® Core™ i5-5000U CPU 2.5GHz.

### 2. Environnement logiciel

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

#### a) *Visuel studio code*



Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'auto complétion, la coloration syntaxique et les commandes git [26].

Figure 57 : Logo de visuel studio code [26]

b) **Postman**



Postman permet de construire et d'exécuter des requêtes HTTP, de les stocker dans un historique afin de pouvoir les rejouer, mais surtout de les organiser en Collections. Cette classification permet notamment de regrouper des requêtes de façon « fonctionnelle » (par exemple enchaînement d'ajout d'item au panier, ou bien un processus d'identification). Postman assure également la gestion des Environnements, qui permet de contextualiser des variables et d'exécuter des requêtes ou des séries de requêtes dans différentes configurations [27].

Figure 58 : Logo du postman [27]

c) **MongoDB compass**



MongoDB Compass est un client graphique pour MongoDB. Si vous hébergez votre base documentaire Nuxeo sous MongoDB, c'est un outil que vous devriez prendre le temps de découvrir. Compass se connecte à votre cluster MongoDB et présente deux onglets - Databases et Performance [28].

Figure 59 : Logo de mongoDB compass [28]

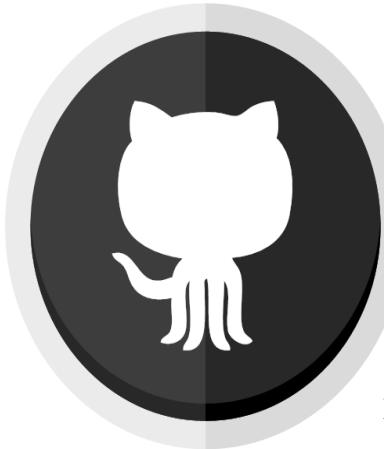
d) **Mongo atlas**

MongoDB est un système de gestion de base de données orienté documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédefini des données. Il est écrit en C++. Le serveur et les outils sont distribués sous licence SSPL, les pilotes sous licence Apache et la documentation sous licence Creative Commons. Il fait partie de la mouvance NoSQL [29].



Figure 60 : Logo de mongo DB Atlas [29]

e) ***Github***



GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Ce site est développé en Ruby on Rails et Erlang par Chris Wanstrath, PJ Hyett et Tom Preston-Werner. GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres. Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet [30].

Figure 61 : Logo de github [30]

f) ***Draw.io***



Draw.io est une application gratuite en ligne, accessible via son navigateur (protocole https) qui permet de dessiner des diagrammes ou des organigrammes. Cet outil vous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter. Draw.io est un véritable couteau suisse de la frise chronologique, de la carte mentale et des diagrammes de tout genre [31].

Figure 62 : Logo de draw.io

g) ***StarUML***



StarUML est un logiciel de modélisation UML, qui a été « cédé comme open source » par son éditeur, à la fin de son exploitation commerciale (qui visiblement continue ...), sous une licence modifiée de GNU GPL [32].

Figure 63 : Logo StarUML [32]

## II. Présentation de la solution

### 1. Interface d'accueil

Tout visiteur de notre plateforme aura accès à la page d'accueil, la liste des catégories et la page de contact. La figure ci-dessous représente la page d'accueil de notre application.



Figure 64 : Page d'accueil

## 2. Interface de création de compte apprenant

L'opération d'inscription est divisée en deux phases d'abord l'interface de paiement où vous devez choisir quel type d'abonnement et secondaire nous trouvons l'application nécessaire à remplir avec toutes les informations personnelles.

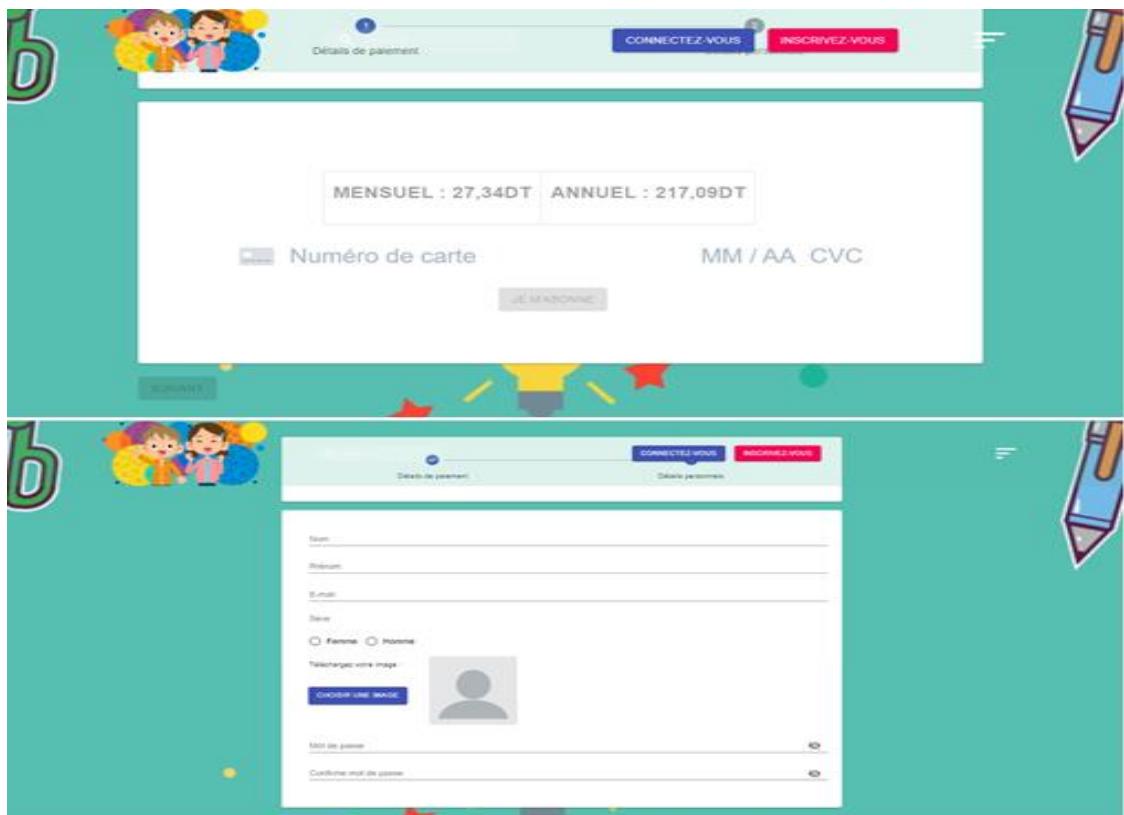


Figure 65 : Création de compte apprenant

## 3. Interface d'authentification

Cette interface permet à l'utilisateur de s'authentifier en entrant son Email et son mot de passe afin de se bénéficier par les différents services de notre application.

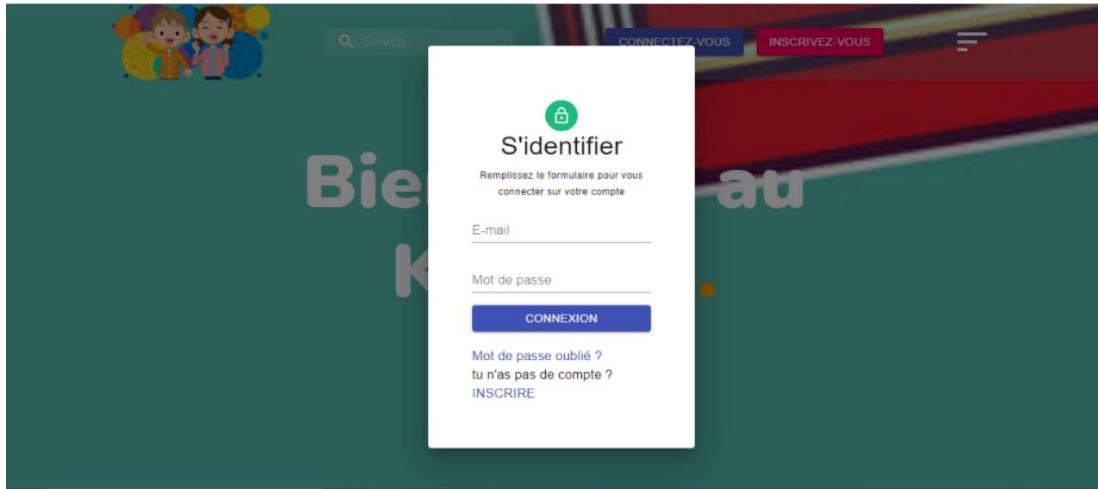


Figure 66 : Interface d'authentification

#### 4. Interface des catégories

Ici, nous pouvons trouver chaque catégorie des compétences qui sont bien sûr divisés sur les différents niveaux éducatifs.



Figure 67 : interface des catégories

#### 5. Interface des compétences

Sous chaque niveau d'éducation, nous pouvons trouver toutes les compétences disponibles pour les utilisateurs.



Figure 68 : Interface des compétences

## 6. Différentes Interface des jeux

Ci-dessous sont les types des jeux disponibles qui sont créés par l'administrateur ou le formateur, et vous pouvez passer de l'un à l'autre en validant vos réponses, bien sûr si vous réussissez, vous obtiendrez un meilleur score et vice versa.

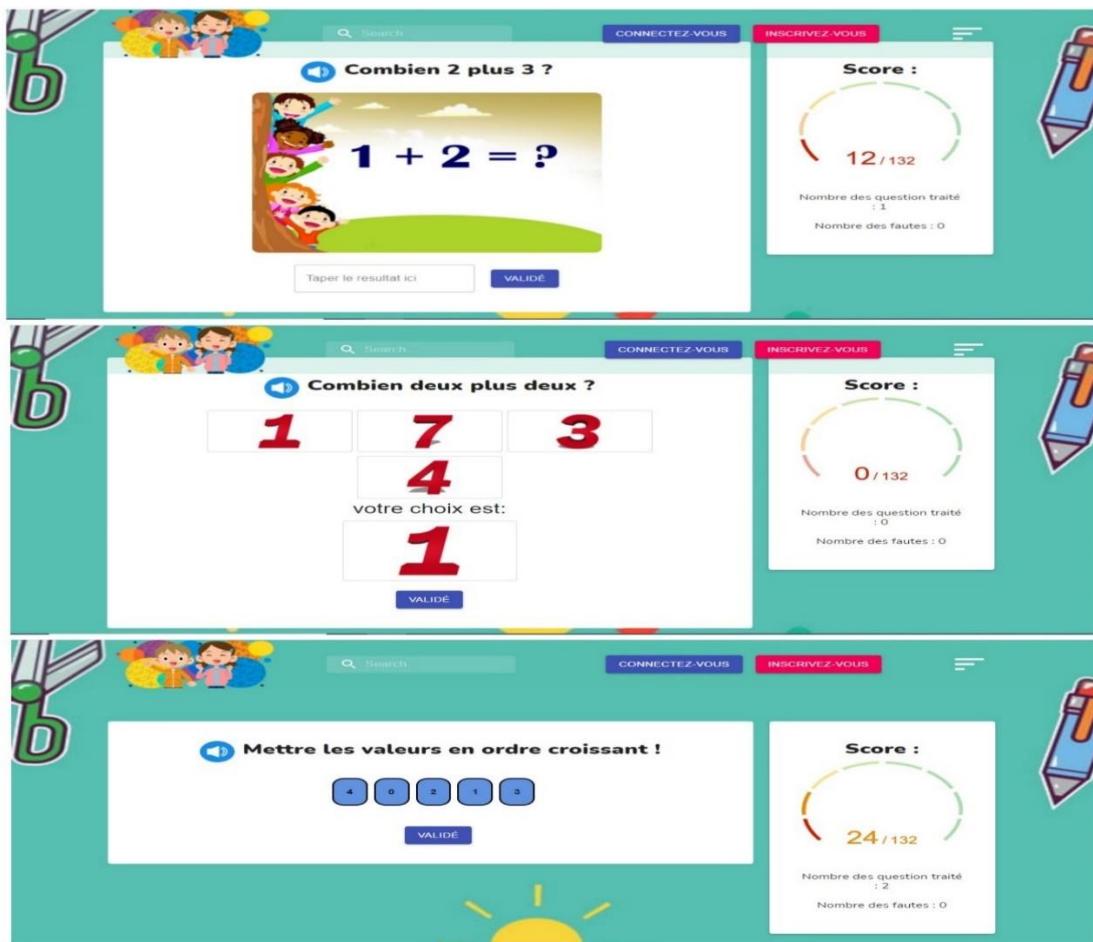


Figure 69 : Différentes Interface des jeux

## 7. Interface de tableau de classement

C'est notre système de classement, et nous nous sommes assurés d'ajouter une telle chose pour créer une atmosphère plus compétitive entre les apprenants.

The screenshot shows a class ranking interface titled "Tableau de classement". The list includes:

Rank	Student Name	Score
2	Saif Dghais	180
	Saif Eddin Dghais	120
	Sa s	60
4	Saoooooooooooooo sa	36
5	Ala ben saad aaaa	0
6	Sa sa	0

At the bottom right, there is a message: "3 sur 24 - Presse-Papie" and "élément ajouté."

Figure 70 : Interface de tableau de classement

## 8. Interface parentale

En remplissant ce formulaire, les parents auront accès à des retours sur les progrès et les résultats de leurs enfants, nous nous sommes assurés de créer une telle interface pour impliquer les parents dans l'expérience de leurs enfants.

The screenshot shows a parent interface with a login form. It features a background illustration of a family. The form instructions are: "Remplissez le formulaire ci-dessous pour obtenir les statistiques de votre enfant". The fields are:

- Email Parent
- Email Apprenant
- Mot de passe

A "ENVOYER" button is at the bottom right.

Figure 71 : Interface d'espace parents

## 9. Gestion des jeux

### a) Consulter les jeux « Illustration image »

The screenshot shows a user interface for managing games. On the left is a sidebar with navigation links: Tableau de bord, Administrateur(s), Utilisateurs, Forum, Categories, Compétances, Illustration image (which is selected and highlighted in blue), Illustration images, Drag&Drop, and Deconnecté. The main area has a header 'Admin saif' and a sub-header 'Les Jeux d'algorithme 1'. It includes a search bar and a button 'CRÉE UN NOUVELLE JEUX1'. Below is a table with columns: Id, Question, Reponse, image, Id de Compétence, and Date de creation. There are four rows of data, each with 'EDITÉ' and 'SUPPRIMER' buttons. The data in the table is as follows:

Id	Question	Reponse	image	Id de Compétence	Date de creation
60d467ff5cae91d482f1ca0	quesque 3+3*3-3	9		60bfae17fb7d1825809536db	2021-06-24T11:09:51.172Z
60d43490a1f6490314ff3899	Combien six plus Sept?	13		60b0be36c147000af8bd3a61	2021-06-24T07:30:24.003Z
60d4343ea1f6490314ff3898	Combien quatre plus treize	17		60b0be36c147000af8bd3a61	2021-06-24T07:29.02.213Z
60d433d6a1f6490314ff3897	Combien 2 plus 3 ?	3		60b0be36c147000af8bd3a61	2021-06-24T07:27:18.450Z

Figure 72 : Consulter les jeux « Illustration image »

### b) Modifier un jeu « Illustration image »

The screenshot shows a form for modifying an existing game. The sidebar is identical to Figure 72. The main form fields are: categorie (set to '2ème année primaire'), Compétences (set to 'Apprendre à compter jusqu'à 3'), Question (set to 'quesque 3+3\*3-3'), Response correct (set to '9'), and a file input field 'Téléchargez les image:' with a 'CHOISIR UNE IMAGE' button. Below these is a preview image showing a cartoon scene with children and the equation '5 + 4 = ?'. At the bottom is a blue 'ENREGISTRER !' button.

Figure 73 :Modifier jeux « Illustration image »

### c) Créeer un jeu « Illustration image »

Les figures ci-dessous représentant le parcours d'ajout d'une jeu « Illustration image » par l'administrateur ou un formateur.

## Chapitre 4 : Réalisation

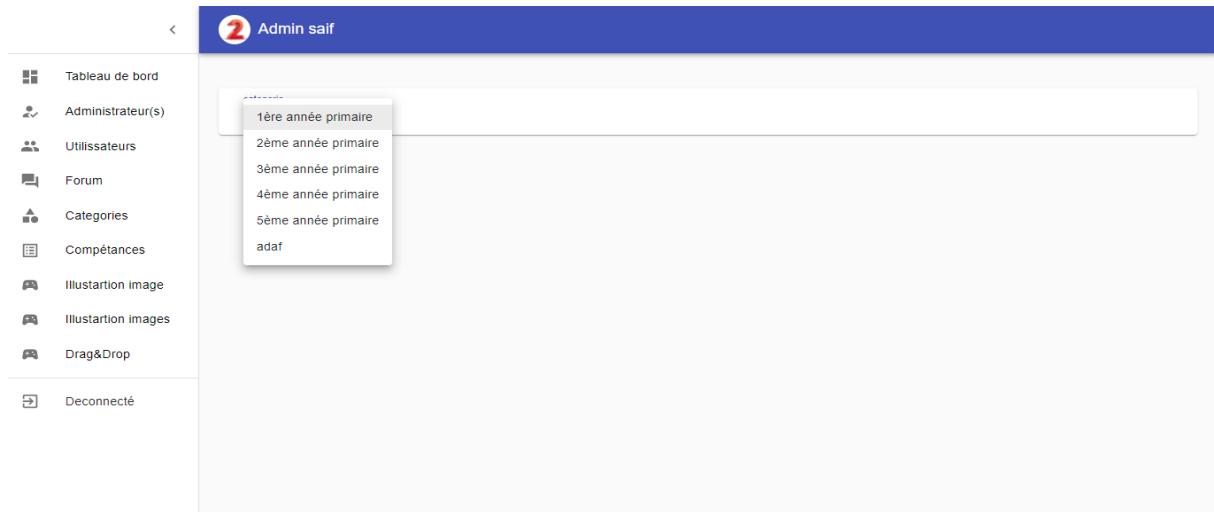


Figure 74 : Choix d'une catégorie

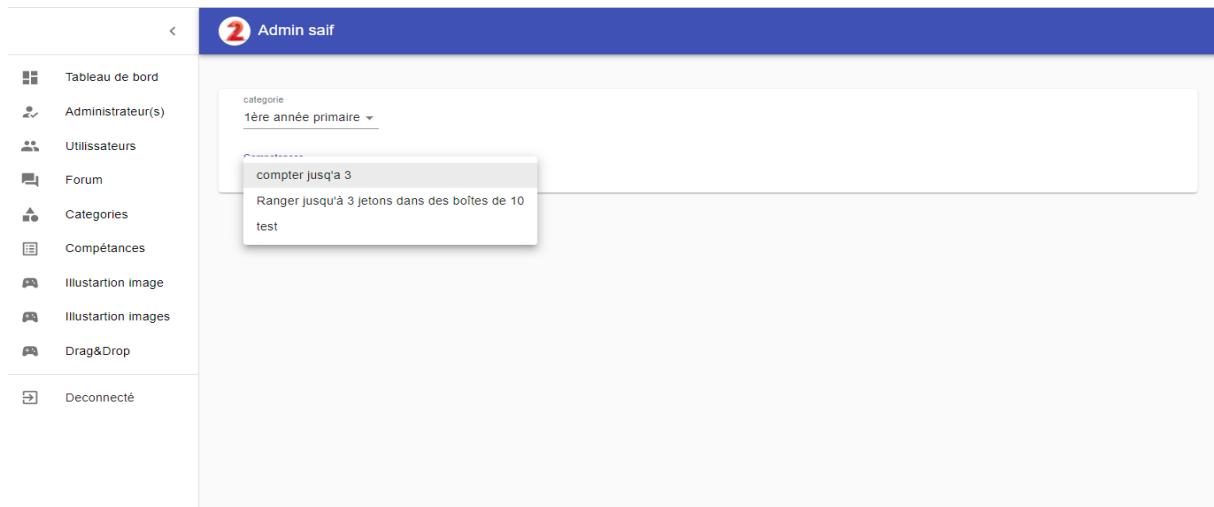


Figure 75 : Choix d'une compétence

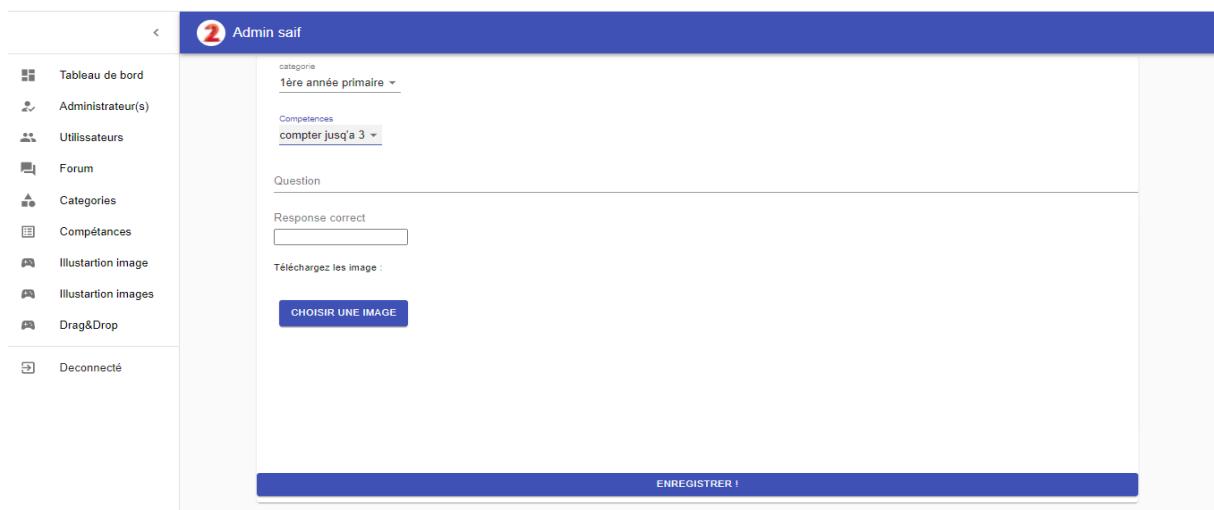


Figure 76 : Dernière phase de création

## Conclusion

Ce chapitre présente les différents logiciels et langages de programmation utilisés durant la réalisation et le développement de ce projet. Nous avons également et illustré la solution finale par des exemples d'interfaces de plateforme web.

# CONCLUSION GÉNÉRALE

Tout au long de la préparation de notre projet de fin d'études, nous avons essayé de mettre en pratique les connaissances acquises au cours de nos études universitaires et certaines autres compétences que nous avons développées au cours du processus afin de réaliser une « Plateforme éducative interactive ».

Le premier chapitre a été consacré au cadre général du projet. Nous avons commencé par une étude de l'existant où nous avons dégagé les insuffisances des solutions existantes et nous avons mis en évidence la mise en place de la méthodologie de travail 2TUP pour la réalisation de notre application web. Le deuxième chapitre ayant pour but de spécifier les besoins fonctionnels, non fonctionnels, et les besoins techniques. Cette étape est la partie cœur pour la réussite du projet, et c'est pour cette raison qu'on fait une étude minutieuse pendant une période importante dans l'analyse et spécification des besoins. Une fois la phase de spécification des besoins est achevée, nous avons entamé la phase de conception. Nous avons utilisé UML comme un langage de modélisation, 2TUP comme un processus de développement et MVC comme patron de conception. Le quatrième chapitre a été dédié à l'aspect implémentation qui contient le choix de l'environnement de travail ainsi que les interfaces réalisées de la partie web.

Ce projet a été bénéfique à bien des égards. C'était notre premier contact avec le monde professionnel, qui demande de la discipline ainsi qu'un travail d'équipe collaboratif. C'était une bonne occasion d'améliorer nos connaissances dans les environnements d'entreprise et de maîtriser les langages de programmation qui seraient certainement utiles à un niveau professionnel dans le futur.

Nous pouvons également ajouter d'autres fonctionnalités au projet qui peuvent l'améliorer de plus en plus telles que :

- Rendre le Site traduisible dans différentes langues pour attirer toujours plus d'utilisateurs d'horizons différents.
- Une application mobile qui peut même rendre l'utilisation du site plus facile et possible partout.
- Ajouter plus de sujets et de domaines d'études.

## BIBLIOGRAPHIE

- [1] apriori-nauka.ru. [En ligne]. Available: <https://apriori-nauka.ru/fr/education/dostoinstva-i-nedostatki-tradicionnoi-sistemy-obucheniya.html>.
- [2] «www.digiforma.com,» [En ligne]. Available: <https://www.digiforma.com/definition/e-learning/>.
- [3] wikipedia.org/wiki. [En ligne]. Available: [https://fr.wikipedia.org/wiki/Formation\\_en\\_ligne#:~:text=La%20d%C3%A9finition%20de%20l'Union,et%20la%20collaboration%20%C3%A0%20distance%20%C2%BB..](https://fr.wikipedia.org/wiki/Formation_en_ligne#:~:text=La%20d%C3%A9finition%20de%20l'Union,et%20la%20collaboration%20%C3%A0%20distance%20%C2%BB..)
- [4] a. asynchrone. [En ligne]. Available: <https://worldscholarshipforum.com/fr/qu%27est-ce-que-l%27apprentissage-synchrone-et-asyncrhone/>.
- [5] b. learning. [En ligne]. Available: <https://www.digiforma.com/definition/blended-learning/>.
- [6] apolearn.com. [En ligne]. Available: <https://apolearn.com/solutions/social-learning/definition-social-learning/>.
- [7] m. cascade. [En ligne]. Available: <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/modele-en-cascade/>.
- [8] m. spirale. [En ligne]. Available: [https://fr.wikipedia.org/wiki/Mod%C3%A8le\\_en\\_spirale#cite\\_note-2](https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_spirale#cite_note-2).
- [9] Scrum. [En ligne]. Available: <https://fr.wikipedia.org/wiki/Scrum>.
- [10] m. XP. [En ligne]. Available: <https://www.planzone.fr/blog/quest-ce-que-la-methodologie-extreme-programming>.
- [11] wikipedia.org/wiki/Processus\_unifi. [En ligne]. Available: [https://fr.wikipedia.org/wiki/Processus\\_unifi%C3%A9#:~:text=Le%20processus%20unifi%C3%A9%20\(du%20c%C3%A9d%C3%A9veloppement%20des%20logiciels%20orient%C3%A9s%20objets..](https://fr.wikipedia.org/wiki/Processus_unifi%C3%A9#:~:text=Le%20processus%20unifi%C3%A9%20(du%20c%C3%A9d%C3%A9veloppement%20des%20logiciels%20orient%C3%A9s%20objets..)
- [12] rup. [En ligne]. Available: <https://www.nutcache.com/fr/blog/methode-rup/>.
- [13] fr.wikipedia.org/wiki/Two\_Tracks\_Unified\_Process. [En ligne]. Available: [https://fr.wikipedia.org/wiki/Two\\_Tracks\\_Unified\\_Process#:~:text=Le%20TUP%20\(2%20tr](https://fr.wikipedia.org/wiki/Two_Tracks_Unified_Process#:~:text=Le%20TUP%20(2%20tr)

acks%20unified%20process, la%20m%C3%A9thode%20du%20processus%20Unifi%C3%A9..

- [14] comparaison, «tunisianstudent,» [En ligne]. Available: <https://www.tunisianstudent.com/upload/documents/b6c938bfcd30672634ad396a01426d14.pdf>.
- [15] «<https://fr.wikipedia.org>,» [En ligne]. Available: [https://fr.wikipedia.org/wiki/UML\\_\(informatique\)#/media/Fichier:UML\\_logo.svg](https://fr.wikipedia.org/wiki/UML_(informatique)#/media/Fichier:UML_logo.svg).
- [16] 1-tiers. [En ligne]. Available: [https://www.memoireonline.com/07/10/3700/m\\_Conception-et-realisation-d'une-application-web-pour-la-gestion-des-stocks-cas-detude-magasin7.html](https://www.memoireonline.com/07/10/3700/m_Conception-et-realisation-d'une-application-web-pour-la-gestion-des-stocks-cas-detude-magasin7.html).
- [17] 2-tiers. [En ligne]. Available: <https://www.commentcamarche.net/contents/221-reseaux-architecture-client-serveur-a-3-niveaux>.
- [18] 3-tiers. [En ligne]. Available: <https://www.techno-science.net/definition/5266.html>.
- [19] n-tiers. [En ligne]. Available: <http://www.info.univ-angers.fr/pub/gh/internet/ntiers.pdf>.
- [20] MVC. [En ligne]. Available: <https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/Cours/Swing/mvc.html>.
- [21] MVP. [En ligne]. Available: <https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-pr%C3%A9sentation>.
- [22] MVVM. [En ligne]. Available: [https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-vue\\_mod%C3%A8le](https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-vue_mod%C3%A8le).
- [23] jwt. [En ligne]. Available: [https://fr.wikipedia.org/wiki/JSON\\_Web\\_Token](https://fr.wikipedia.org/wiki/JSON_Web_Token).
- [24] a. diagramme. [En ligne]. Available: <http://remy-manu.no-ip.biz/UML/Cours/coursUML11.pdf>.
- [25] transition. [En ligne]. Available: [https://fr.wikipedia.org/wiki/Diagramme\\_%C3%A9tats-transitions](https://fr.wikipedia.org/wiki/Diagramme_%C3%A9tats-transitions).
- [26] Vs-code. [En ligne]. Available: <https://framalibre.org/content/visual-studio-code>.
- [27] postman. [En ligne]. Available: <https://blog.webnet.fr/presentation-de-postman-outil-multifonction-pour-api-web/>.
- [28] compass. [En ligne]. Available: <https://www.nuxeo.com/fr/blog/get-a-closer-look-at-your-nuxeo-data-with-mongodb-compass/>.

- [29] **mongoDB.** [En ligne]. Available: <https://fr.wikipedia.org/wiki/MongoDB>.
- [30] **Github.** [En ligne]. Available: <https://fr.wikipedia.org/wiki/GitHub>.
- [31] **Draw.io.** [En ligne]. Available: <https://www.tice-education.fr/tous-les-articles-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne>.
- [32] **StarUML.** [En ligne]. Available: <https://fr.wikipedia.org/wiki/StarUML>.