

## TABLE DES MATIÈRES

1	INTRODUCTION.....	4
2	PRESENTATION GENERALE DU PROJET .....	5
2.1	Situation.....	5
2.2	Objectif.....	5
3	ETUDE DE L'EXISTANT ET AMÉLIORATIONS .....	6
3.1	Etude de l'existant.....	6
3.2	Améliorations .....	6
4	SPÉCIFICATION DES BESOINS ET EXIGENCES.....	7
4.1	Besoins fonctionnels .....	7
4.2	Besoins non fonctionnels .....	7
5	ANALYSE ET CONCEPTION.....	8
5.1	Architecture.....	8
5.2	Diagramme de contexte.....	10
5.3	Diagramme de classe .....	11
5.4	Diagramme de cas d'utilisation .....	12
5.4.1	Diagramme .....	12
5.4.2	Description textuelle des cas d'utilisation .....	13
5.5	Diagramme de séquence .....	18
5.5.1	Cas d'utilisation "s'enregistrer" .....	18
5.5.2	Cas d'utilisation "s'authentifier" .....	19
5.5.3	Cas d'utilisation "Créer projet" .....	20
5.5.4	Cas d'utilisation "Editer projet" .....	21
5.5.5	Cas d'utilisation "Importer Granule" .....	22
5.5.6	Cas d'utilisation "Générer document" .....	23
5.5.7	Cas d'utilisation "Créer Nouveau Chat" .....	24
5.5.8	Cas d'utilisation "Sauvegarder projet" .....	25
5.6	Diagramme d'activité.....	26
5.6.1	Cas d'utilisation "s'enregistrer" .....	26
5.6.2	Cas d'utilisation "s'authentifier" .....	27
5.6.3	Cas d'utilisation "Créer projet" .....	28
5.6.4	Cas d'utilisation "Editer projet" .....	29
5.6.5	Cas d'utilisation "Générer document" .....	30
5.6.6	Cas d'utilisation " Sauvegarder document" .....	31
5.6.7	Cas d'utilisation "importer granule" .....	32

6	DÉVELOPPEMENT .....	33
6.1	Conception détaillée.....	33
6.2	Les outils et technologies.....	33
7	GUIDE D'UTILISATION .....	35
8	PERSPECTIVES.....	43
9	CONCLUSION .....	44
10	RÉFÉRENCES BIBLIOGRAPHIQUES.....	45

## TABLE DES FIGURES

Figure 1 Architecture client-serveur du système.....	8
Figure 2 Variante « classique » sur le Web .....	9
Figure 3 Diagramme de contexte .....	10
Figure 4 Diagramme de classe .....	11
Figure 5 Diagramme de cas d'utilisation .....	12
Figure 6 Diagramme de séquence s'enregistrer .....	18
Figure 7 Diagramme de séquence "s'authentifier" .....	19
Figure 8 Diagramme de séquence "Créer projet" .....	20
Figure 9 Diagramme de séquence "Editer projet" .....	21
Figure 10 Diagramme de séquence "importer granule".....	22
Figure 11 Diagramme de séquence "Générer document" .....	23
Figure 12 Diagramme de séquence "Créer nouveau chat" .....	24
Figure 13 Diagramme de séquence "Sauvegarder projet" .....	25
Figure 14 Diagramme d'activité "S'enregistrer" .....	26
Figure 15 Diagramme d'activité "S'authentifier" .....	27
Figure 16 Diagramme d'activité "Créer projet" .....	28
Figure 17 Diagramme d'activité "Editer projet" .....	29
Figure 18 Diagramme d'activité "Générer document" .....	30
Figure 19 Diagramme d'activité "Sauvegarder projet" .....	31
Figure 20 Diagramme d'activité "Importer granule" .....	32
Figure 21 Landing page.....	35
Figure 22 Page d'enregistrement.....	36
Figure 23 Page de composition de contenu 1 .....	37
Figure 24 Ajout d'un nouvel élément 1 .....	37
Figure 25 Ajout d'un nouvel élément 2 .....	38
Figure 26 Granule ajoutée à gauche.....	38
Figure 27 Import d'un granule de XCSM.....	39
Figure 28 Choix du fichier à importer.....	39
Figure 29 Granule ajouté.....	40
Figure 30 Création de PDF.....	40
Figure 31 création de fichier Word.....	41
Figure 32 Aperçu du bouton de chat .....	41
Figure 33 Interface de chat 1 .....	42
Figure 34 Interface de chat 2 .....	42

# 1 INTRODUCTION

---

XCCM (eXtended Content Composition Module) est une application mise sur pied pour faciliter la recomposition des contenus. En d'autres termes, c'est un module de composition de contenus qui a pour objectif de produire des contenus pédagogiques à partir des fragments de contenus issus de la segmentation. La composition de contenu nous permet de produire des contenus pédagogiques à partir des fragments de contenus issus de la segmentation (disponible au niveau du module de structuration). Cependant, un défi persiste : car l'objectif est de développer un module d'adaptation des contenus au sein des plateformes d'apprentissage, dans la création de cours personnalisés et l'utilisation des connaissances existantes et aider d'un chat bot pour formuler au mieux chaque leçon. Pour atteindre cet objectif, notre travail sera structuré comme suite. Nous avons tout d'abord la présentation générale du logiciel XCCM, par la suite nous procéderons à l'étude de l'existant et les améliorations apportées aux projet, par après nous poursuivrons notre travail sur la spécification des besoins et les exigences. Et enfin nous procéderons à l'analyse et la conception, le développement et le guide d'utilisation du logiciel.

## **2 PRESENTATION GENERALE DU PROJET**

---

### **2.1 SITUATION**

Dans le monde numérique actuel, où l'information et le contenu jouent un rôle crucial, les systèmes de gestion de contenu (CMS) sont devenus des outils essentiels pour les entreprises, les éducateurs, et les créateurs de contenu. Ils offrent une solution centralisée et efficace pour créer, gérer, et distribuer du contenu numérique. Les modules de composition de contenu sont un moyen efficace de dispenser des formations en ligne. Ils sont flexibles et peuvent être adaptés à différents publics et objectifs. Ils sont également interactifs et engageants, ce qui permet aux apprenants d'acquérir de nouvelles connaissances et compétences de manière efficace.

### **2.2 OBJECTIF**

Le principal objectif de notre projet est de permettre la distribution de contenu de haute qualité à une audience large. Que ce soit pour éduquer, informer, divertir ou vendre, notre CMS devrait aider à présenter le contenu de manière attrayante et professionnelle.

## 3 ETUDE DE L'EXISTANT ET AMÉLIORATIONS

---

### 3.1 ETUDE DE L'EXISTANT

A la prise en main du projet par notre groupe, l'application offrait les services suivants :

- La connexion à l'application grâce à un email et un mot de passe fixes
- La création de cours

A la première présentation du projet, l'application permettait désormais :

- La gestion dynamique de la connexion de l'utilisateur : Nous avons implémenté la création de compte. Un compte peut être créé avec un mot de passe et l'email de l'utilisateur.
- La création de compte et authentification avec Google
- La génération de fichiers au format PDF

### 3.2 AMÉLIORATIONS

Les améliorations apportées au projet sont les suivantes :

- Amélioration de la sécurité grâce à l'authentification à deux facteurs.
- Optimisation de la gestion des cours/projets : on peut désormais modifier le contenu d'un nœud, sauvegarder un projet en cours (donnant ainsi la possibilité de continuer ou l'on s'est arrêté la dernière fois)
- Génération de fichiers aux formats PDF / docx
- L'importation des granules venant de XCSM
- Intégration d'un chat bot pour assister les utilisateurs
- Amélioration des interfaces

## 4 SPÉCIFICATION DES BESOINS ET EXIGENCES

---

### 4.1 BESOINS FONCTIONNELS

- Le système doit permettre à un nouvel utilisateur de s'enregistrer
- Le système doit permettre à un utilisateur de s'authentifier pour se connecter à l'application
- Le système doit permettre de créer un projet ou un cours
- Le système doit permettre de modifier un projet ou cours
- Le système doit permettre d'importer un granule
- Le système doit permettre de générer un document sous forme de fichier PDF ou docx
- Le système doit permettre d'interagir avec un chat bot pour naviguer dans le système et obtenir de l'aide

### 4.2 BESOINS NON FONCTIONNELS

- Performance : Temps de réponse, rapidité de traitement.
- Sécurité : Protection des données, authentification et autorisation sécurisées.
- Utilisabilité : Facilité d'utilisation, interface utilisateur conviviale.
- Fiabilité : Stabilité du système, gestion des erreurs.
- Maintenabilité : Facilité de mise à jour et de maintenance du code.

## 5 ANALYSE ET CONCEPTION

### 5.1 ARCHITECTURE

L'architecture que notre application utilise est une architecture client-serveur. L'architecture client-serveur est un modèle de conception informatique où les ressources et les services sont distribués entre des ordinateurs spécialisés appelés serveurs et des ordinateurs clients qui demandent des ressources ou des services à ces serveurs [1]. La figure 1 nous présente l'architecture qu'utilise notre système.

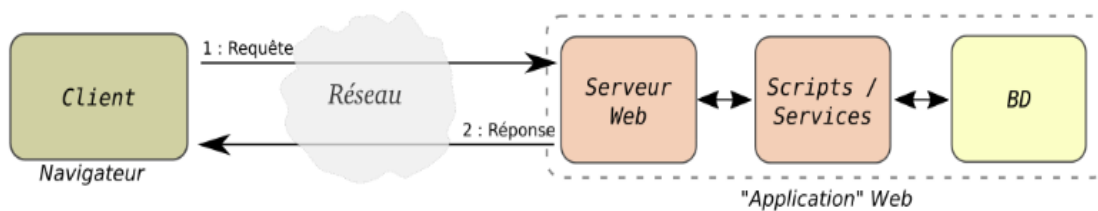


Figure 1 Architecture client-serveur du système

#### ➤ Dialogue entre client et serveur

La communication se fait en 3 étapes simples :

1. Le client (navigateur) fait une requête d'accès à une ressource auprès d'un serveur Web selon le protocole HTTP.
2. Le serveur vérifie la demande, les autorisations et transmet éventuellement l'information demandée.
3. Le client interprète la réponse reçue (et l'affiche)

On recommence pour des ressources complémentaires (images, ...).

#### ➤ Concepts de base de HTTP

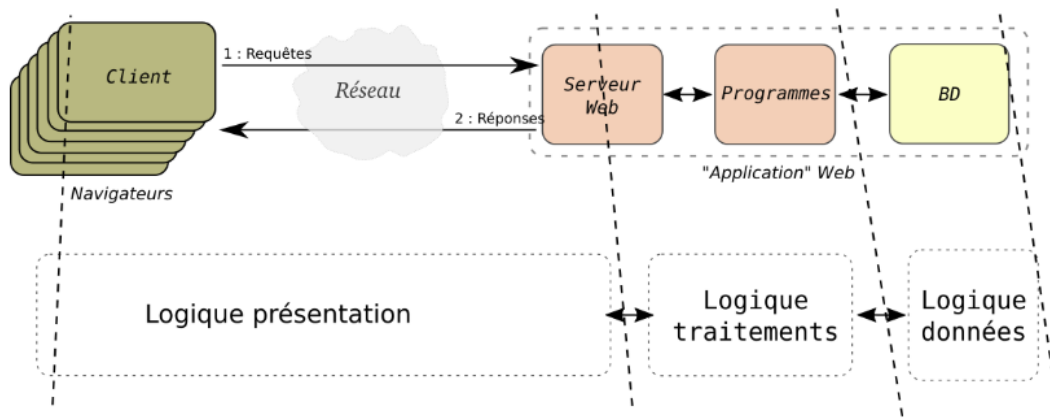
Interaction avec des ressources hypertexte se font de la manière suivante :

- Quoi : Action : verbe (CRUD).
- Qui : Ressources identifiées par des URI (données d'une application, ...).
- Où : URL pour localiser les représentations informatiques de ces ressources (documents hypertextes, images, etc.) sur des serveurs Internet.
- Comment : Requêtes protocole de communication HTTP



L'application fonctionne sur un serveur accessible depuis le réseau via le protocole HTTP. Un navigateur Web permet de s'y connecter et de consulter des pages Web présentant le résultat de l'exécution d'un programme fonctionnant dans la mémoire du serveur. Attention, ici, plusieurs clients Web peuvent accéder simultanément à la même application fonctionnant sur le serveur Web.

La figure 2 nous présente le fonctionnement de chaque logique de l'architecture client-serveur.

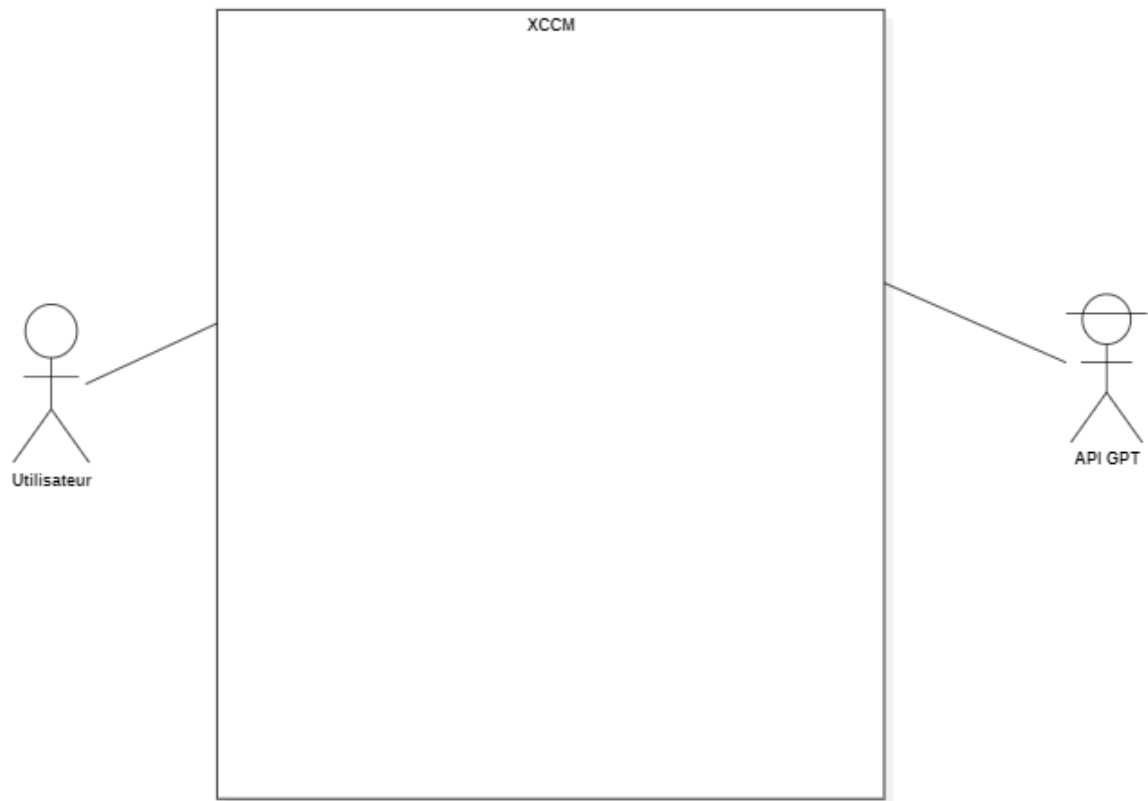


*Figure 2 Variante « classique » sur le Web*

- 1 serveur, N clients en simultané.
- Classique : traitements principalement côté serveur

## 5.2 DIAGRAMME DE CONTEXTE

La figure 3 nous présente le diagramme de contexte de notre système :



*Figure 3 Diagramme de contexte*

### 5.3 DIAGRAMME DE CLASSE

La figure 4 nous présente le diagramme de classe de notre projet.

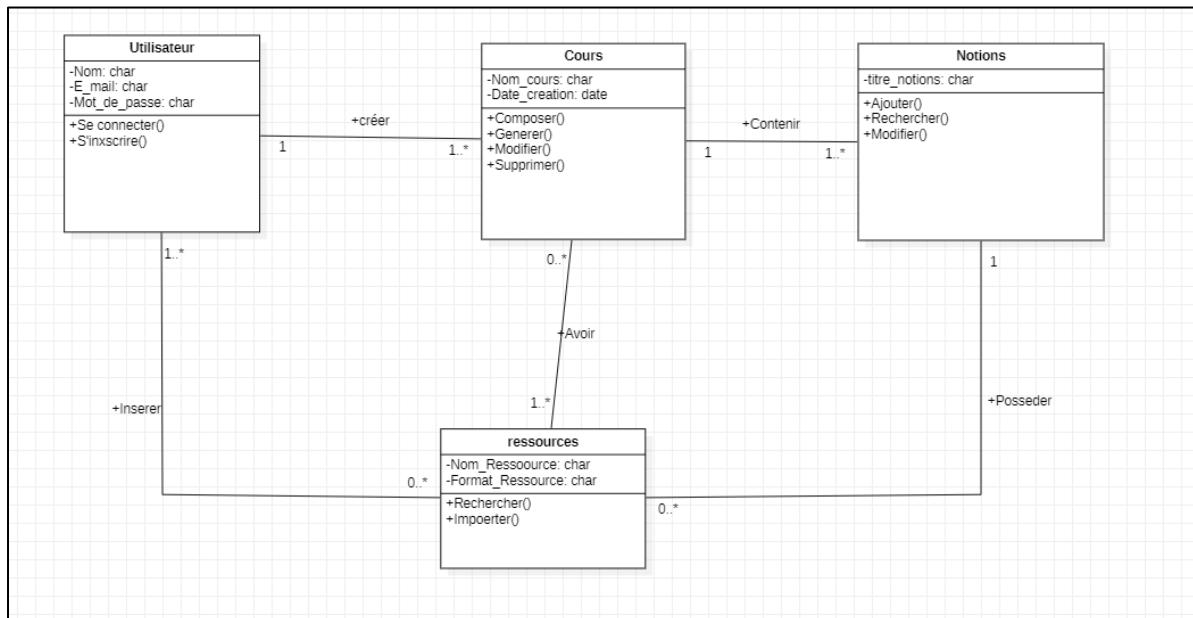


Figure 4 Diagramme de classe

## 5.4 DIAGRAMME DE CAS D'UTILISATION

### 5.4.1 Diagramme

La figure 5 nous présente le diagramme de cas d'utilisation de notre projet.

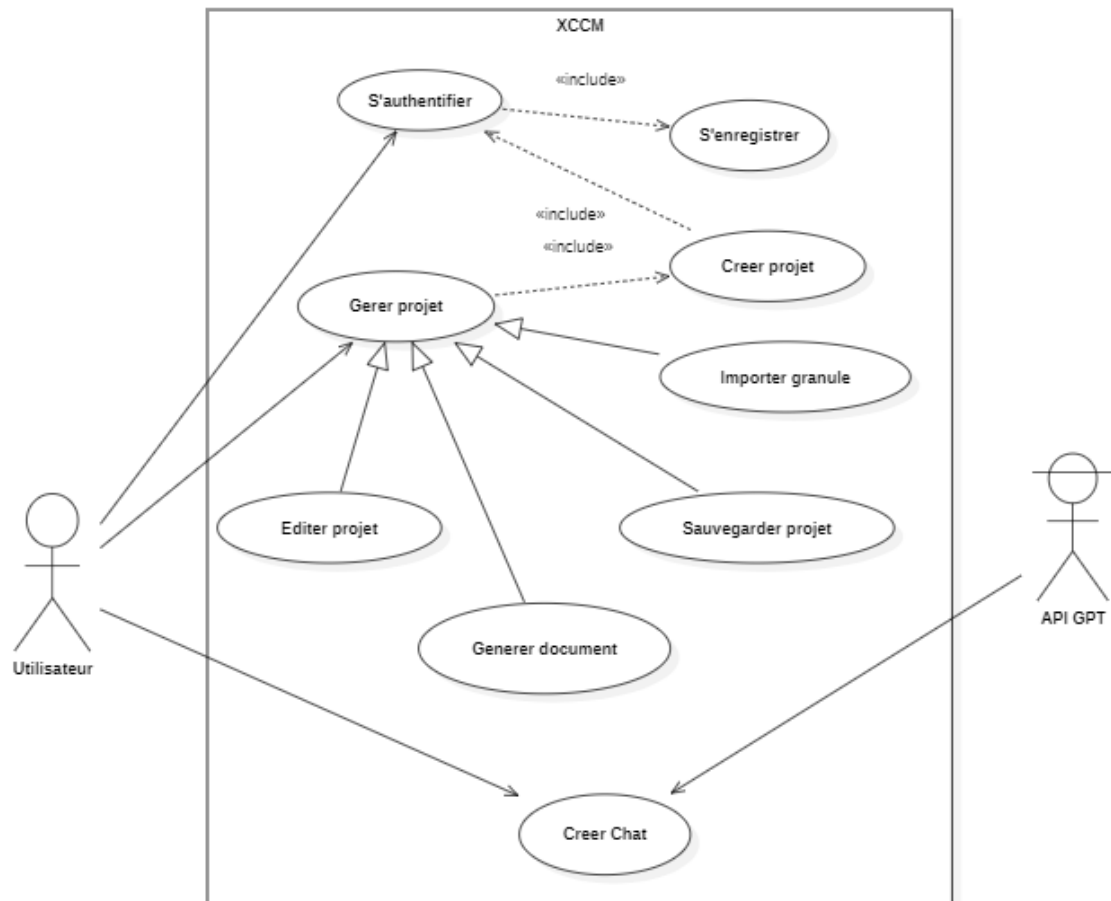


Figure 5 Diagramme de cas d'utilisation

## 5.4.2 Description textuelle des cas d'utilisation

### 5.4.2.1 Cas d'utilisation "s'enregistrer"

Objectif	Permettre l'inscription des nouveaux utilisateurs.
Acteurs Concernés	Utilisateur
Précondition	Aucune
Post- Condition	Possibilité d'accéder aux fonctionnalités en tant qu'utilisateur.
Scénario Nominal	Aucun
Scénario Alternatif	Aucun

### 5.4.2.2 Cas d'utilisation "s'authentifier"

Objectif	Permettre l'identification de chaque utilisateur lui donnant accès à certaines fonctionnalités.
Acteurs Concernés	Utilisateur.
Précondition	Il dispose d'un compte d'accès au système.
Post- Condition	Aucune
Scénario Nominal	<ul style="list-style-type: none"><li>- L'utilisateur saisit son identifiant et son mot de passe.</li><li>- Il est reconnu par le système.</li><li>- Il accède au système en tant qu'utilisateur.</li></ul>
Scénario Alternatif	<ul style="list-style-type: none"><li>- Identifiant ou mot de passe non valide</li><li>- Un message d'erreur est affiché signifiant qu'il faut rentrer un identifiant qui soit valide ou un mot de passe valide</li></ul>

#### 5.4.2.3 Cas d'utilisation "Créer projet"

Objectif	Permettre à l'utilisateur de créer une nouvelle composition.
Acteurs Concernés	Utilisateur
Précondition	S'être authentifié au préalable.
Post- Condition	Possibilité de saisir des nœuds à l'intérieur.
Scénario Nominal	<ul style="list-style-type: none"><li>- L'utilisateur clique sur nouvelle composition.</li><li>- Il saisit le titre du cours puis valide.</li><li>- Le dossier portant le nom du cours est créé et ajouté aux précédents cours, avec comme sous-dossiers : app pour le contenu, configurations pour les styles associés au nœud en question, ressources pour les différents éléments extérieurs qui ont été ajoutés et export pour les téléchargements (fichiers docx et pdf).</li></ul>
Scénario Alternatif	Aucun.

#### 5.4.2.4 Cas d'utilisation "Editer projet"

Objectif	Permettre à l'utilisateur d'éditer un nœud.
Acteurs Concernés	Utilisateur.
Précondition	S'être authentifié au préalable.
Post- Condition	Possibilité d'enregistrer et de générer un cours.
Scénario Nominal	<ul style="list-style-type: none"><li>- L'utilisateur clique dans le dossier du cours en question (s'il n'existe pas, il faudra le créer).</li><li>- Puis il entre dans le sous-dossier app.</li><li>- Il clique sur la notion et édite le contenu.</li></ul>

#### 5.4.2.5 Cas d'utilisation "Générer document"

Objectif	Permettre à l'utilisateur de générer des fichiers PDF ou docx (télécharger après la composition) un cours.
Acteurs Concernés	Utilisateur.
Précondition	Avoir fini la composition du cours.
Post- Condition	Aucune.
Scénario Nominal	L'utilisateur clique sur générer .PDF ou .docx selon son choix.
Scénario Alternatif	Aucun

#### 5.4.2.6 Cas d'utilisation "Sauvegarder projet"

Objectif	Permettre à l'utilisateur de composer (enregistrer après la saisie) un cours.
Acteurs Concernés	Utilisateur.
Précondition	Avoir fini la saisie des notions.
Post- Condition	Possibilité de générer des fichiers docx et pdf.
Scénario Nominal	<ul style="list-style-type: none"><li>- L'utilisateur clique sur générer.</li><li>- Le système lui renvoie l'aperçu du cours composé.</li></ul>
Scénario Alternatif	Aucun

#### 5.4.2.7 Cas d'utilisation "Importer granule"

Objectif	Permettre à un utilisateur d'importer des modules de contenu (granules) qui peuvent contenir une phrase, un paragraphe, une photo,...
Acteurs Concernés	Utilisateur
Précondition	L'utilisateur doit être authentifié et en train d'éditer un cours/projet
Post- Condition	Les granules importées sont ajoutées au cours/projet
Scénario Nominal	<ul style="list-style-type: none"><li>- L'utilisateur accède à la section d'édition d'un cours/proje</li><li>- L'utilisateur sélectionne l'option pour importer des granules</li><li>- L'utilisateur choisit les granules à importer depuis un fichier</li><li>- L'utilisateur confirme l'importation</li><li>- Le système ajoute les granules sélectionnées au cours/projet</li></ul>



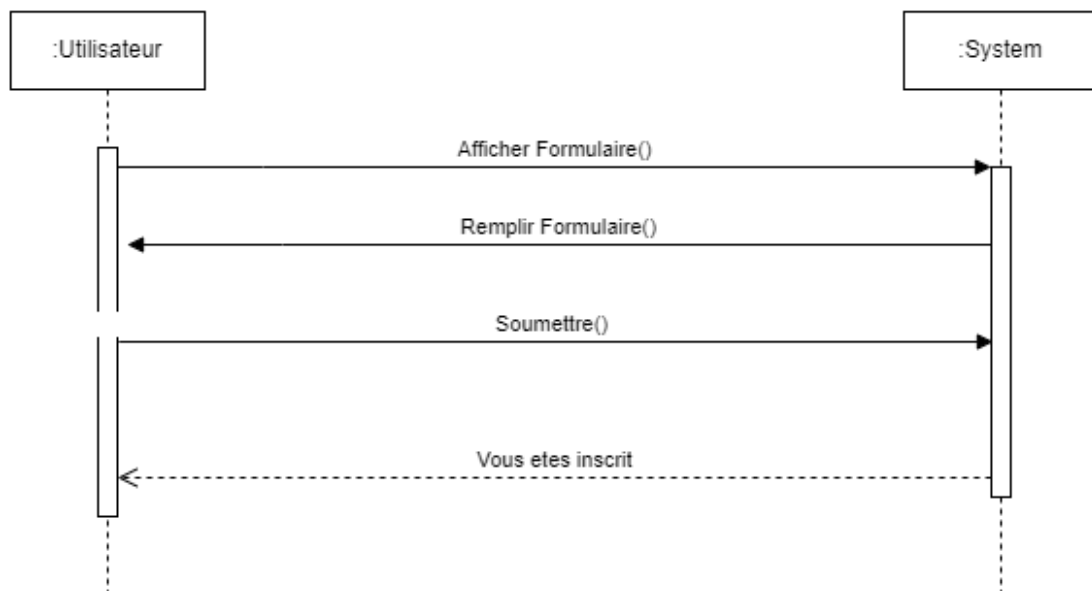
#### 5.4.2.8 Créer chat

Objectif	Permettre à un utilisateur d'importer des modules de contenu (granules) qui peuvent contenir une phrase, un paragraphe, une photo,...
Acteurs Concernés	Utilisateur
Précondition	L'utilisateur doit être authentifié et en train d'éditer un cours/projet
Post- Condition	Les granules importées sont ajoutées au cours/projet
Scénario Nominal	<ul style="list-style-type: none"><li>- L'utilisateur accède à la section d'édition d'un cours/projet</li><li>- L'utilisateur sélectionne l'option pour importer des granules</li><li>- L'utilisateur choisit les granules à importer depuis un fichier</li><li>- L'utilisateur confirme l'importation</li><li>- Le système ajoute les granules sélectionnées au cours/projet</li></ul>
Scénario Alternatif	Aucun

## 5.5 DIAGRAMME DE SÉQUENCE

### 5.5.1 Cas d'utilisation "s'enregistrer"

La figure 6 nous présente le diagramme de séquence système du cas d'utilisation s'enregistrer.



*Figure 6 Diagramme de séquence s'enregistrer*

### 5.5.2 Cas d'utilisation "s'authentifier"

La figure 7 nous présente le diagramme de séquence système du cas d'utilisation s'authentifier.

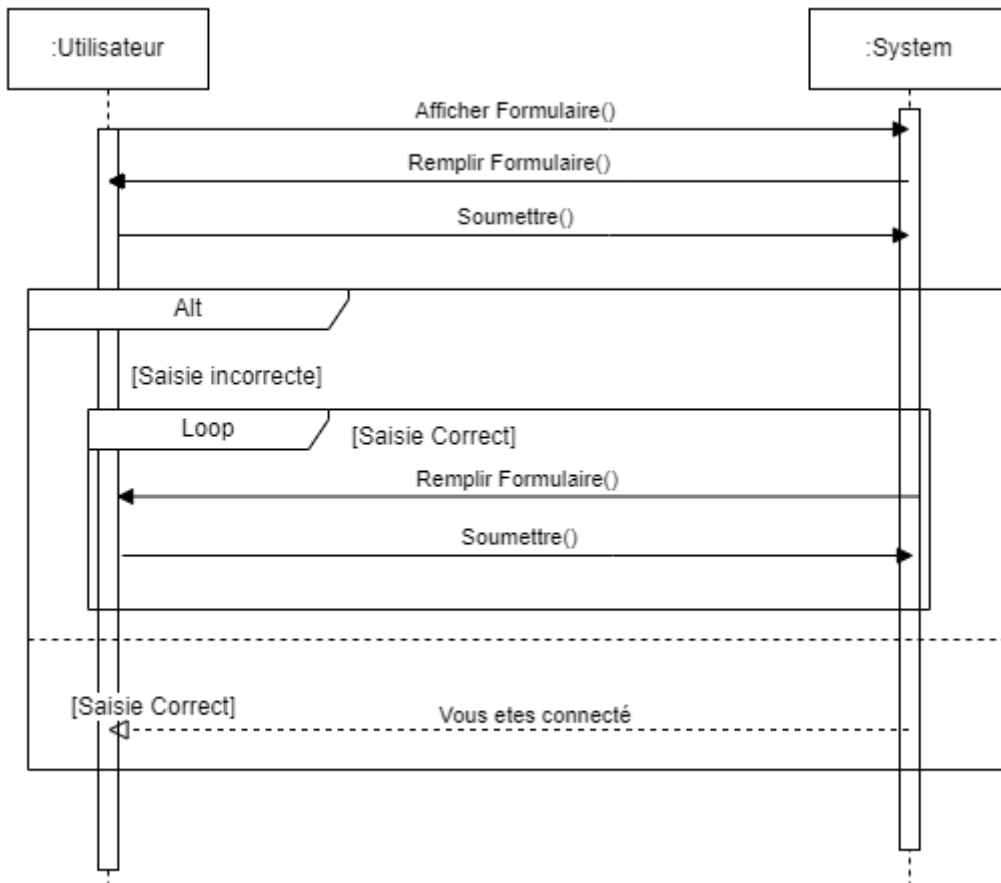


Figure 7 Diagramme de séquence "s'authentifier"

### 5.5.3 Cas d'utilisation "Créer projet"

La figure 8 nous présente le diagramme de séquence système du cas d'utilisation créer projet.

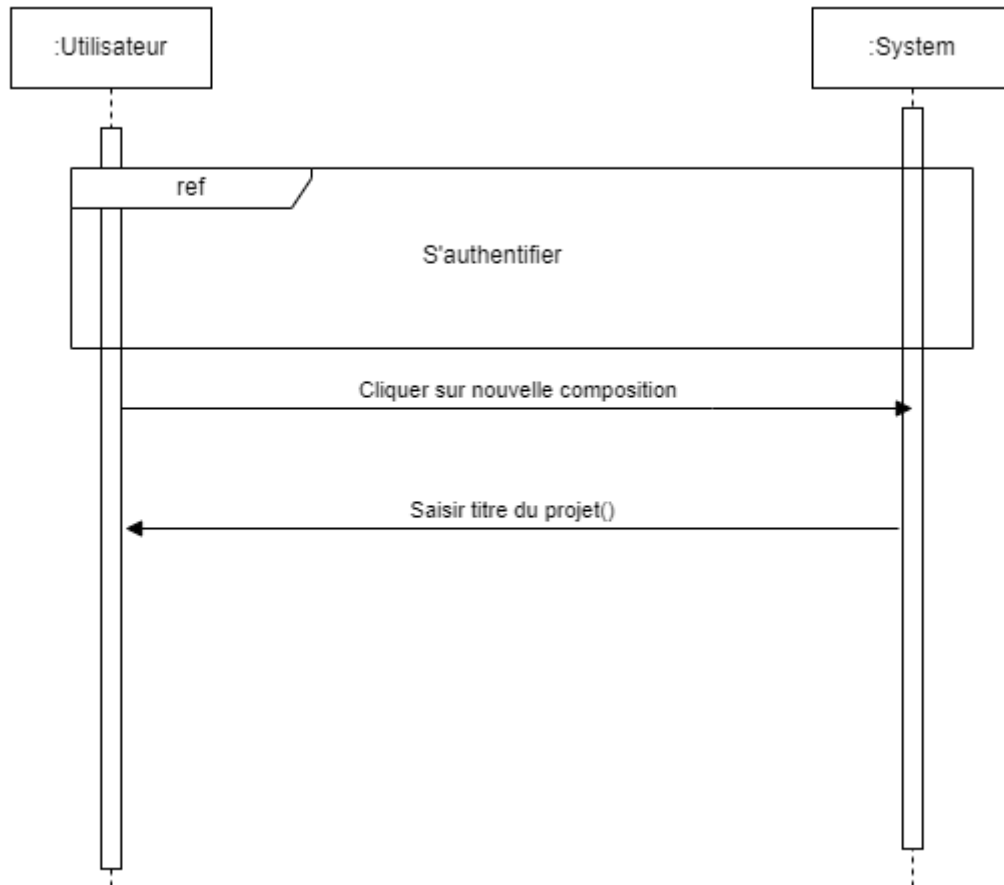


Figure 8 Diagramme de séquence "Créer projet"

#### 5.5.4 Cas d'utilisation "Editer projet"

La figure 9 nous présente le diagramme de séquence système du cas d'utilisation éditer projet.

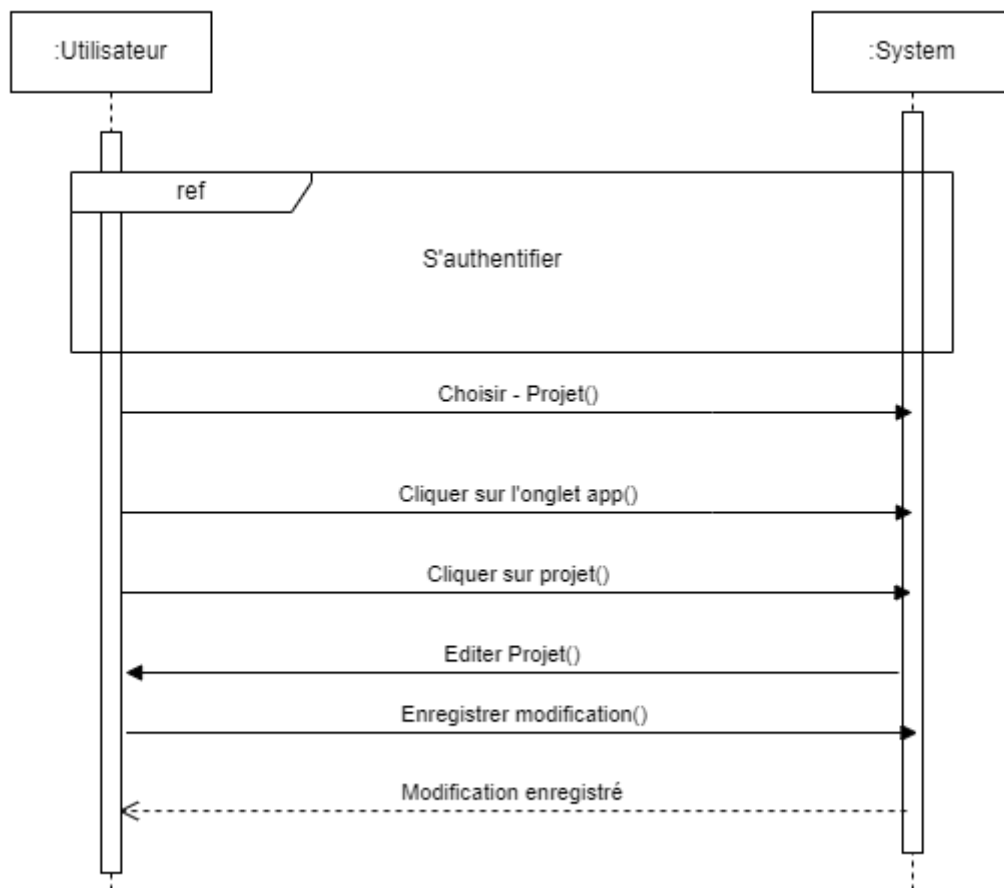


Figure 9 Diagramme de séquence "Editer projet"

### 5.5.5 Cas d'utilisation "Importer Granule"

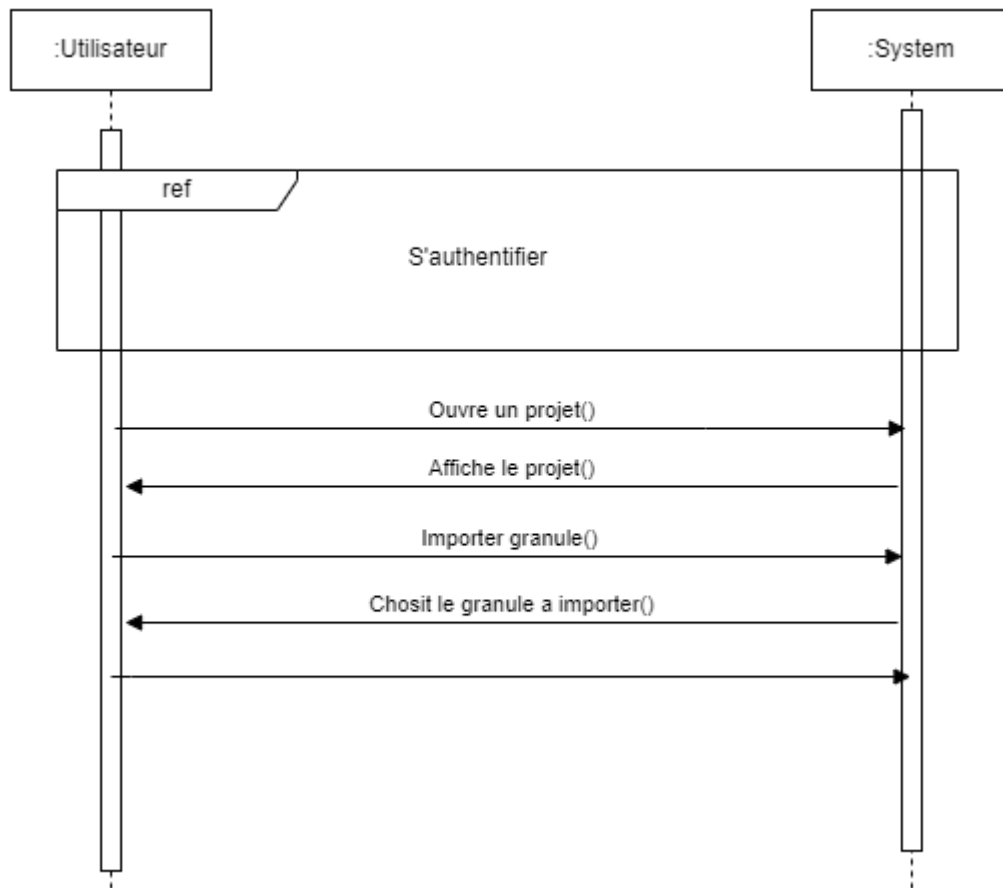
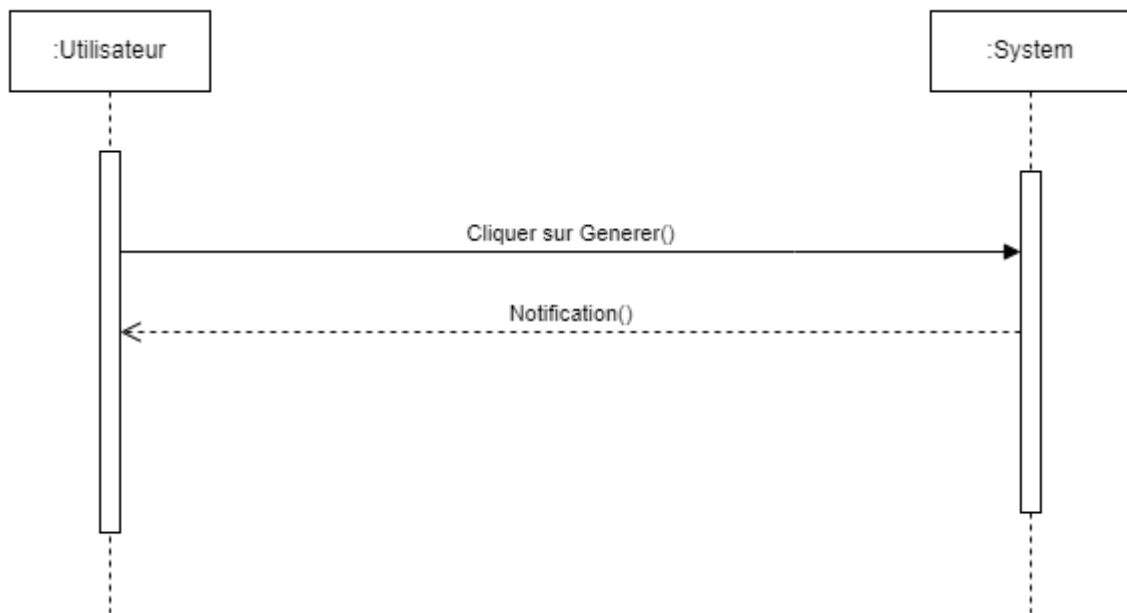


Figure 10 Diagramme de séquence "importer granule"

### 5.5.6 Cas d'utilisation "Générer document"

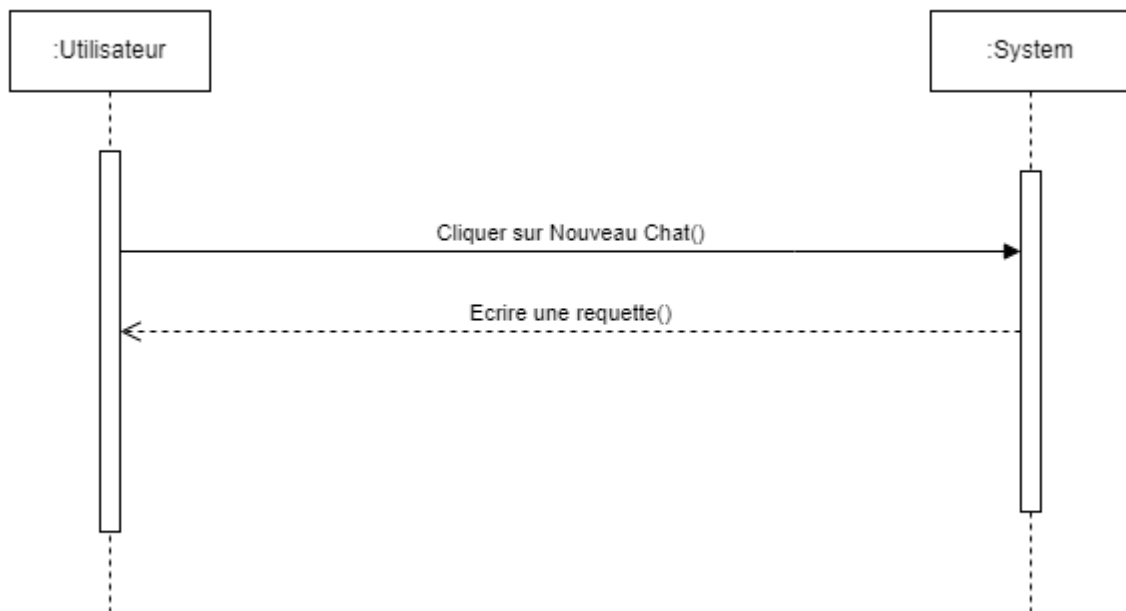
La figure 10 nous présente le diagramme de séquence système du cas d'utilisation Generer document.



*Figure 11 Diagramme de séquence "Générer document"*

### 5.5.7 Cas d'utilisation "Créer Nouveau Chat"

La figure 11 nous présente le diagramme de séquence système du cas d'utilisation créer nouveau chat.



*Figure 12 Diagramme de séquence "Créer nouveau chat"*



### 5.5.8 Cas d'utilisation "Sauvegarder projet"

La figure 13 nous présente le diagramme de séquence système du cas d'utilisation sauvegarder projet.

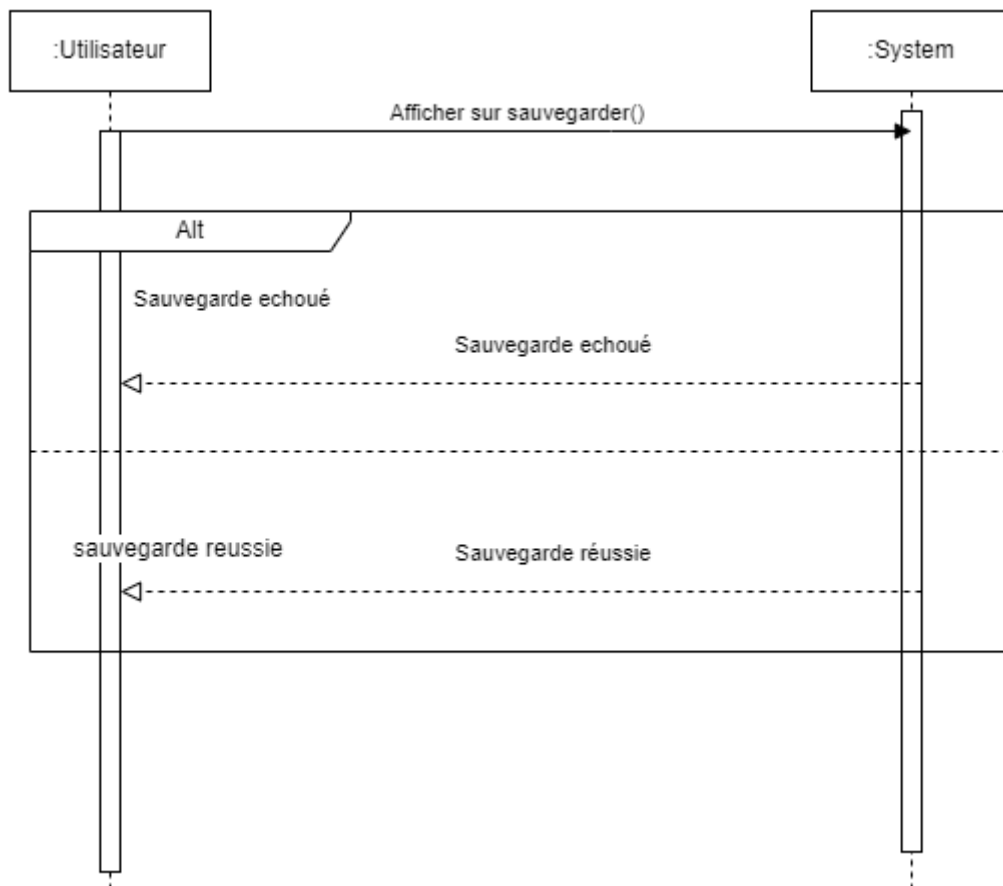


Figure 13 Diagramme de séquence "Sauvegarder projet"

## 5.6 DIAGRAMME D'ACTIVITÉ

### 5.6.1 Cas d'utilisation "s'enregistrer"

La figure 13 nous présente le diagramme d'activité du Cas d'utilisation "s'enregistrer"

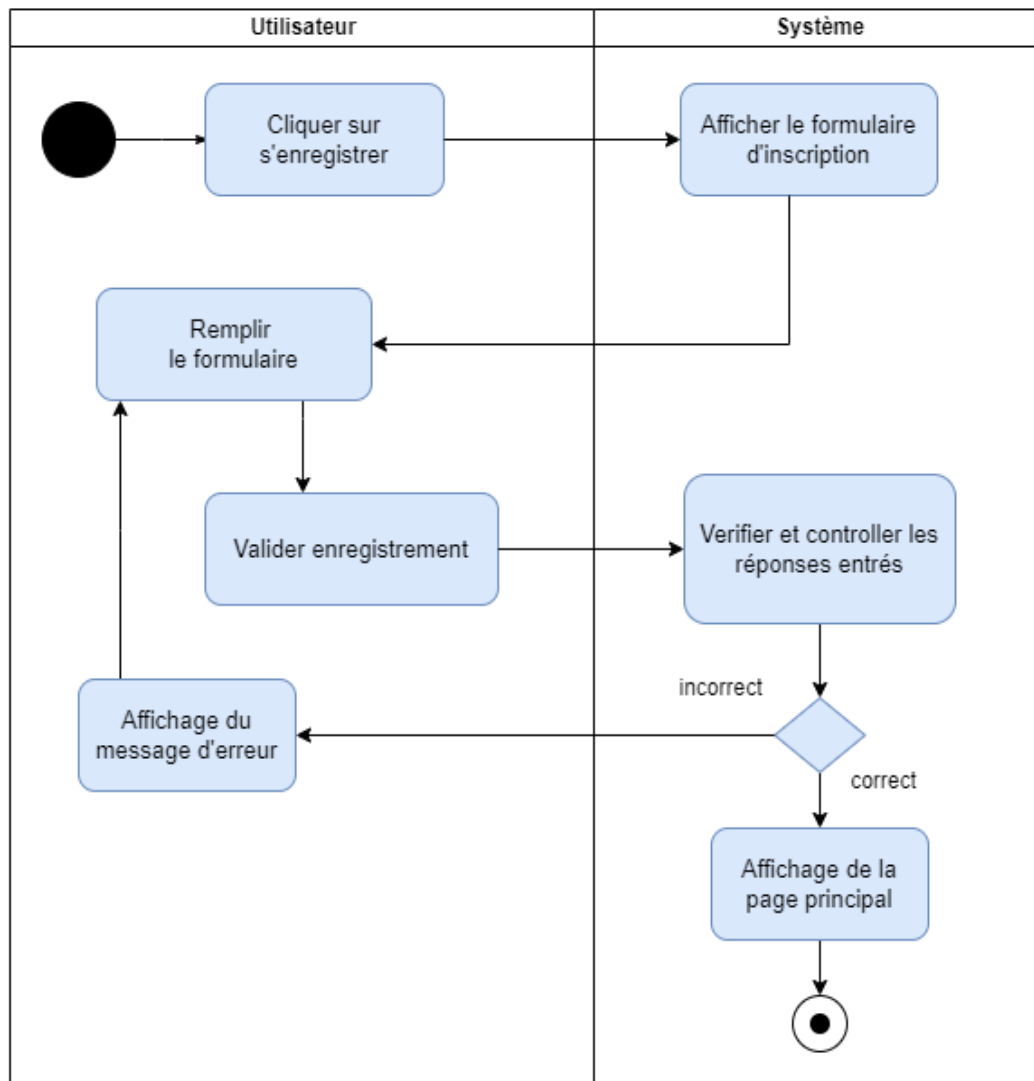


Figure 14 Diagramme d'activité "S'enregistrer"

### 5.6.2 Cas d'utilisation "s'authentifier"

La figure 14 nous présente le diagramme d'activité du Cas d'utilisation "s'authentifier"

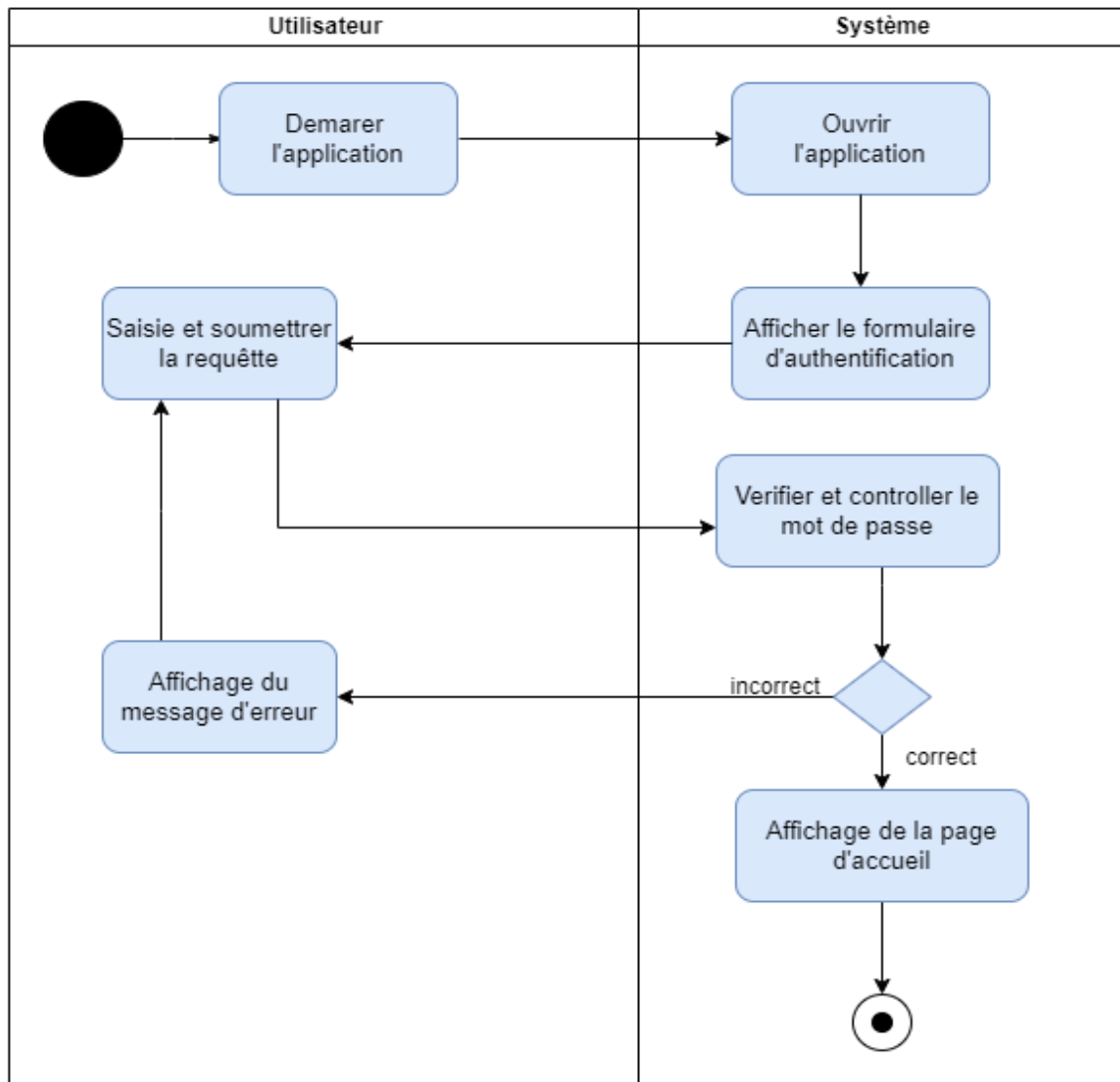


Figure 15 Diagramme d'activité "S'authentifier"

### 5.6.3 Cas d'utilisation "Créer projet"

La figure 15 nous présente le diagramme d'activité de cas d'utilisation "Créer projet"

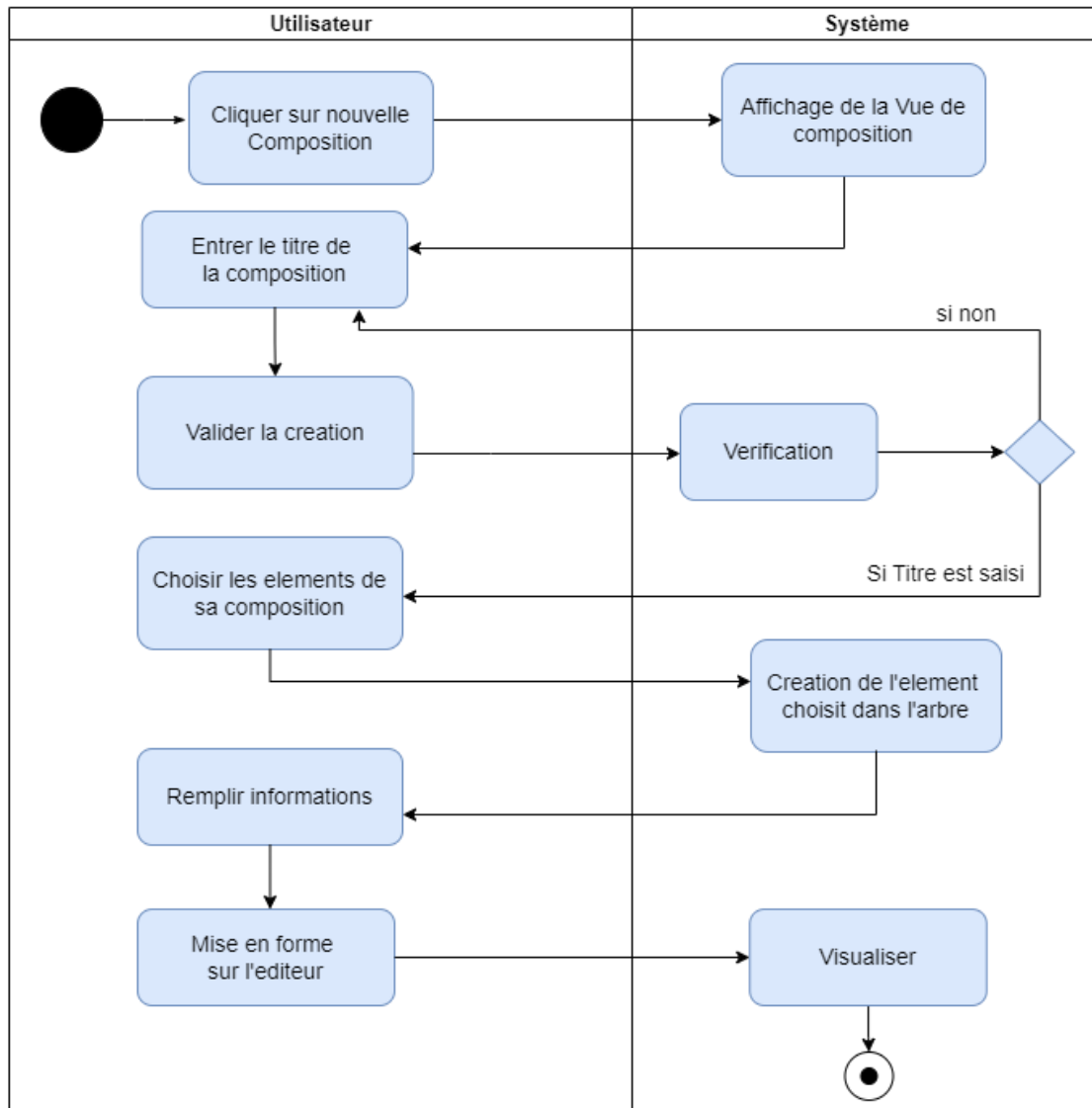


Figure 16 Diagramme d'activité "Créer projet"

#### 5.6.4 Cas d'utilisation "Editer projet"

La figure 16 nous présente le diagramme d'activité du Cas d'utilisation "Editer projet"

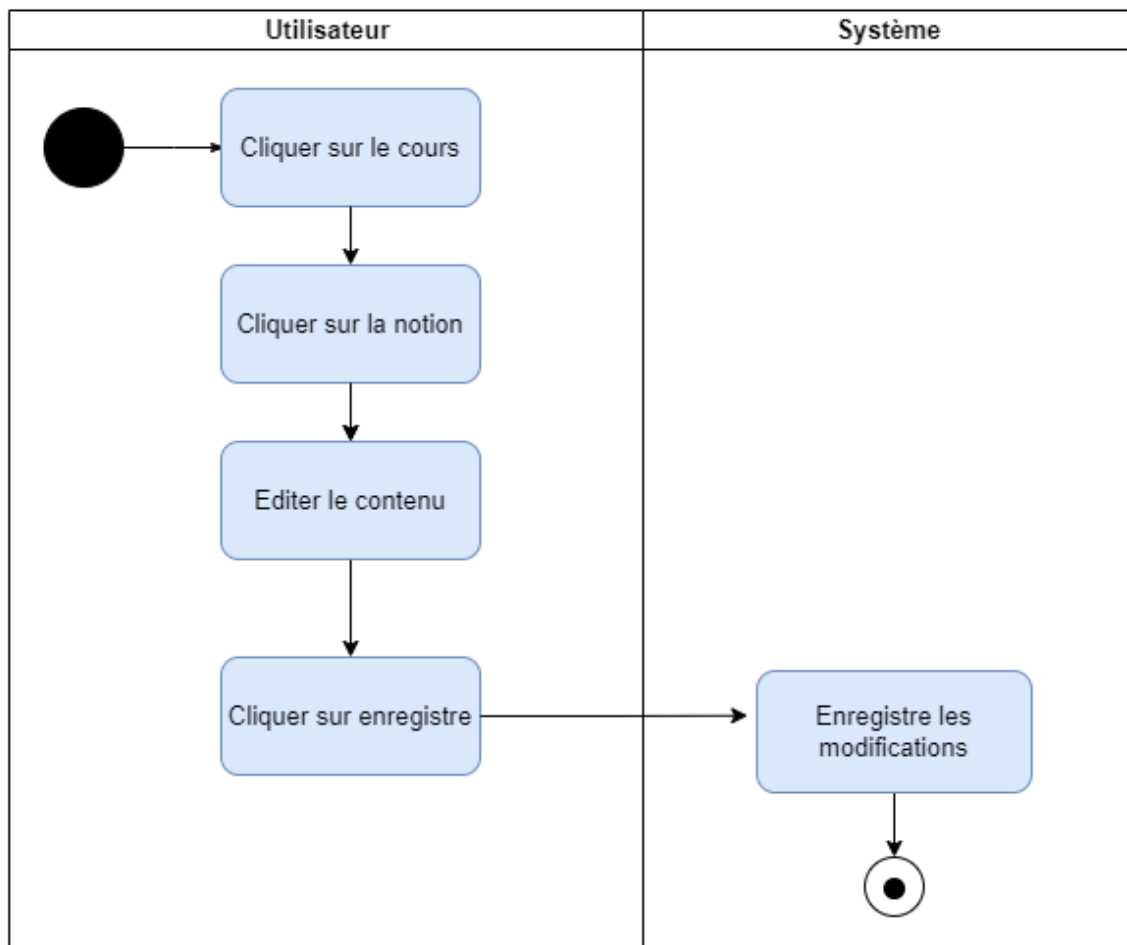


Figure 17 Diagramme d'activité "Editer projet"

### 5.6.5 Cas d'utilisation "Générer document"

La figure 17 nous présente le diagramme d'activité du cas d'utilisation "Gérer document"

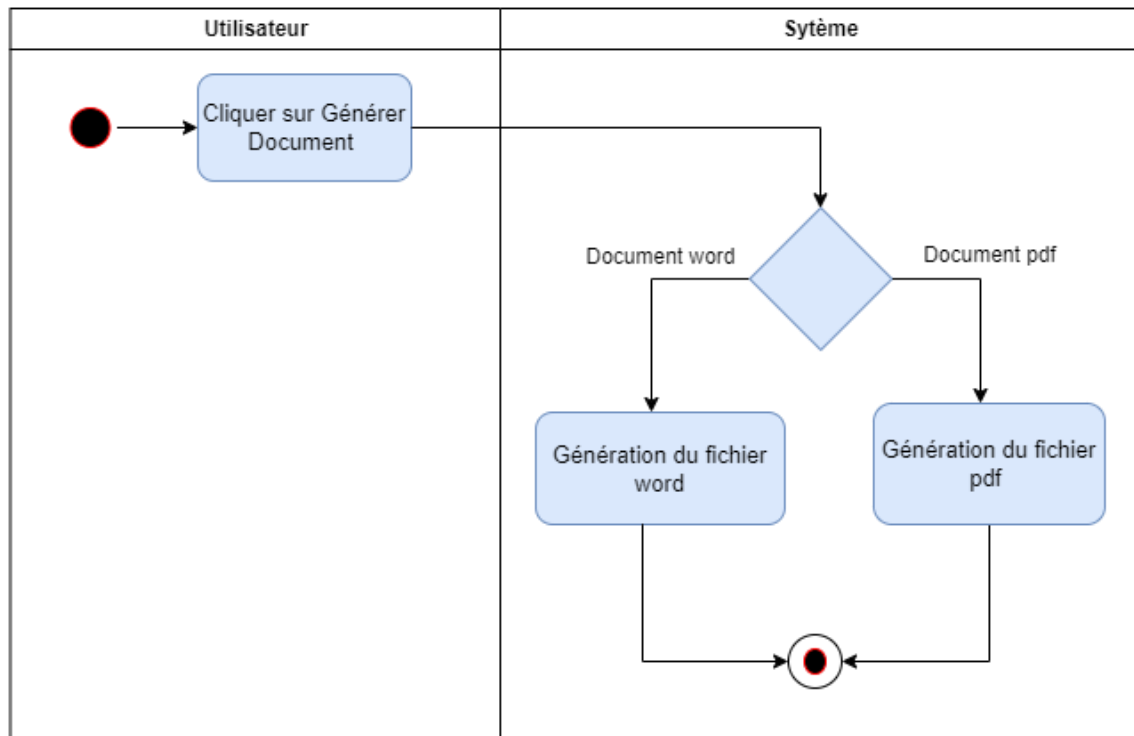


Figure 18 Diagramme d'activité "Générer document"

### 5.6.6 Cas d'utilisation "Sauvegarder document"

La figure 19 nous présente le diagramme d'activité du Cas d'utilisation "Sauvegarder document"

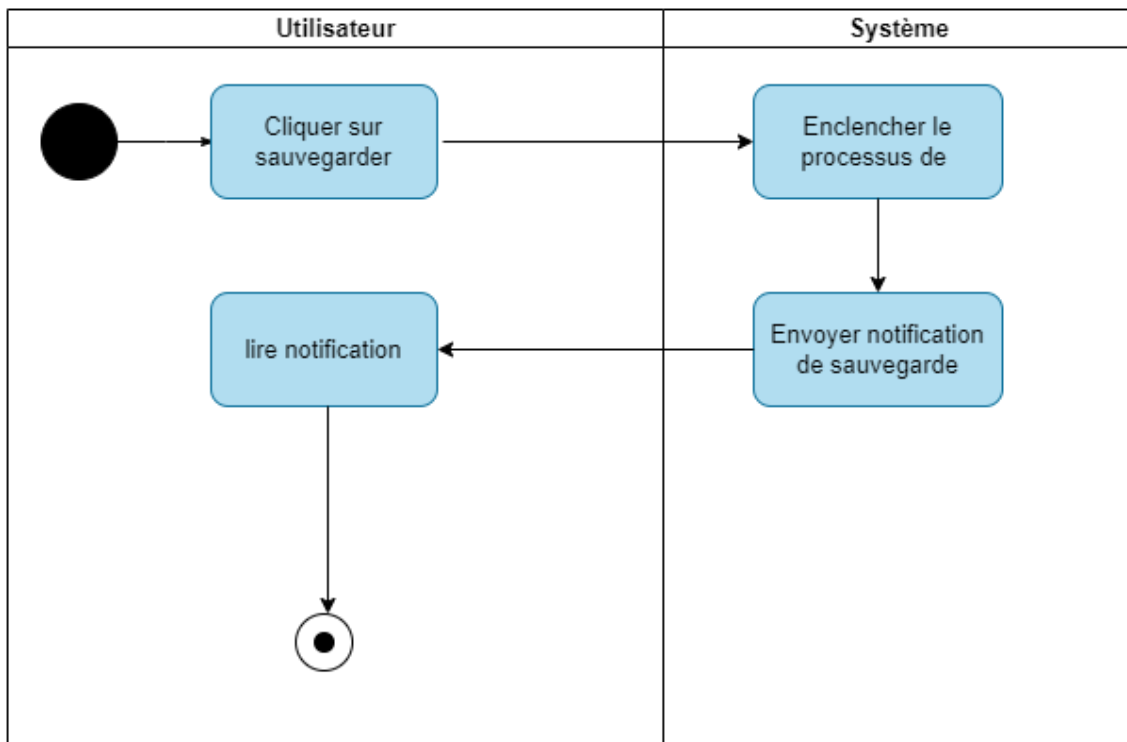


Figure 19 Diagramme d'activité "Sauvegarder projet"

### 5.6.7 Cas d'utilisation "importer granule"

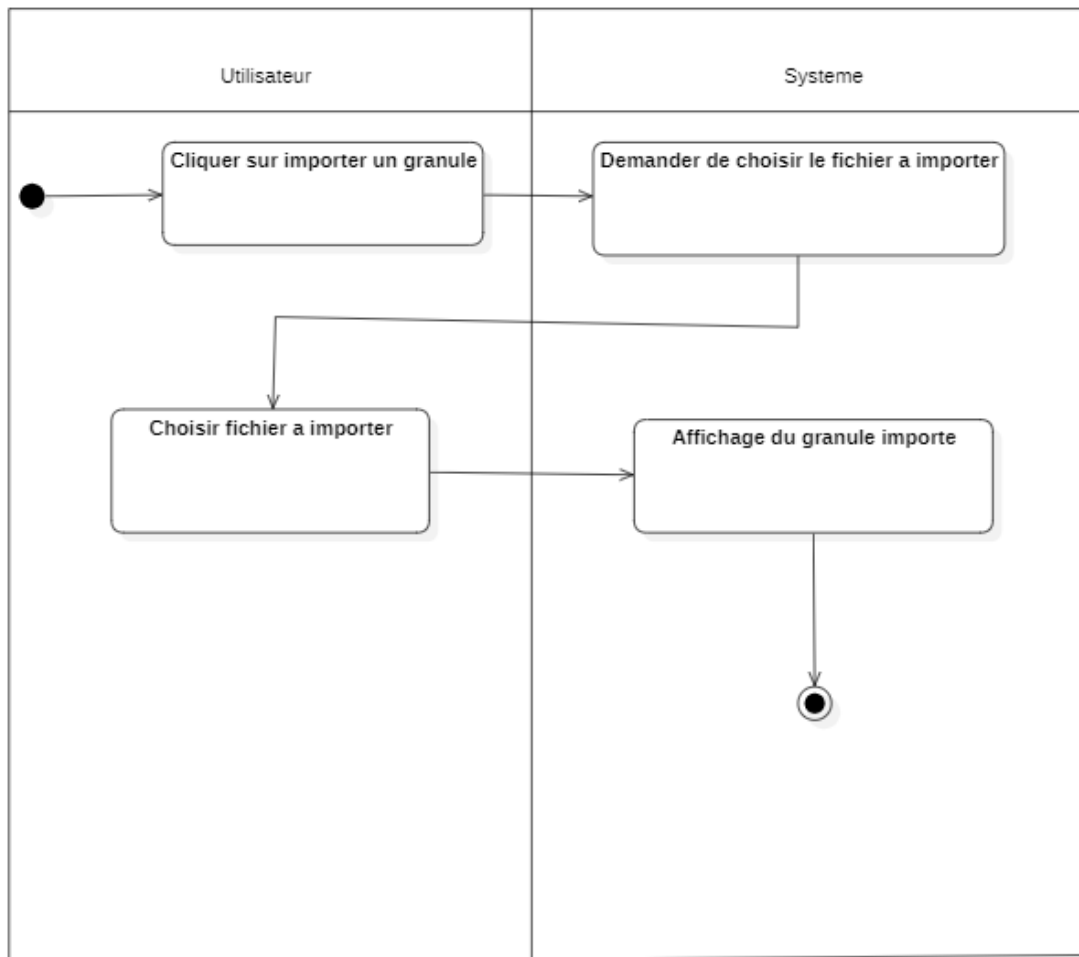


Figure 20 Diagramme d'activité "Importer granule"



## 6 DÉVELOPPEMENT

---

### 6.1 CONCEPTION DÉTAILLÉE

- Créer une application Next js et installer les dépendances nécessaires
- Installer et configurer la connexion avec Mongo dB
- Configurer l'enregistrement et l'authentification des utilisateurs
- Créer le modèle utilisateur avec Mongoose
- Développement des différentes fonctionnalités du projet

### 6.2 LES OUTILS ET TECHNOLOGIES

Pour implémenter les différentes fonctionnalités nous avons utilisé les technologies et outils suivants :

- Pour la conception nous avons utilisé **Draw.io**
- Pour le développement, **Côté serveur**, les langages utilisés sont :
  - Next.js est un Framework web populaire basé sur React, conçu pour faciliter le développement d'applications web modernes. Il est principalement utilisé pour la création d'applications côté client, mais il offre également des fonctionnalités pour le rendu côté serveur, ce qui le rend polyvalent pour une variété de projets. Next.js prend en charge le rendu côté serveur, ce qui peut améliorer les performances et l'expérience.
  - Mongo DB est un système de gestion de base de données No SQL qui stocke les données sous forme de documents JSON flexibles au lieu de tables avec des lignes et des colonnes, comme dans les bases de données relationnelles classiques. Mongo DB est populaire pour sa flexibilité, sa scalabilité et sa capacité à gérer des données non structurées ou semi-structurées.

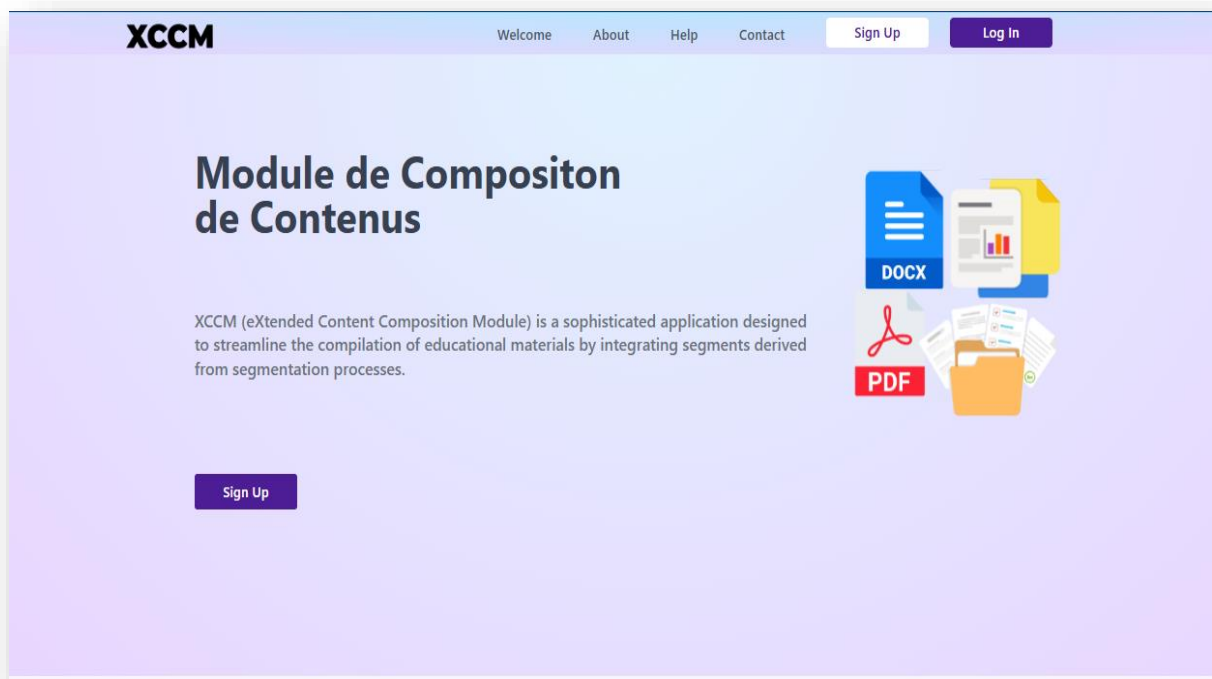
➤ **Côté client**, les langages utilisés sont :

- Next.js est un Framework web populaire basé sur React, conçu pour faciliter le développement d'applications web modernes. Il est principalement utilisé pour la création d'applications côté client, mais il offre également des fonctionnalités pour le rendu côté serveur, ce qui le rend polyvalent pour une variété de projets. Next.js prend en charge le rendu côté serveur, ce qui peut améliorer les performances et l'expérience.

## 7 GUIDE D'UTILISATION

---

Lorsque le projet est lancé sur un port de l'ordinateur. On se retrouve sur cette page d'accueil.



*Figure 21 Landing page*

**XCCM**

## Sign up

**Name \***

**Email \***

**Password \***



**Confirm your password \***

**Profession**

**Domain**

**Sign up**

Or sign up with

[Already have an account?](#)

Figure 22 Page d'enregistrement

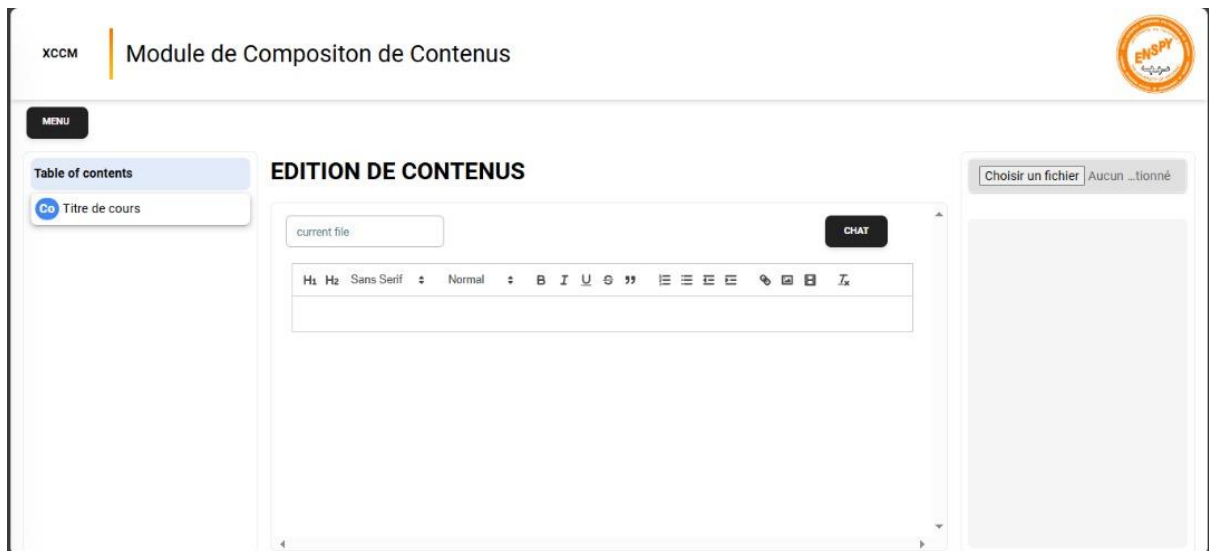


Figure 23 Page de composition de contenu 1

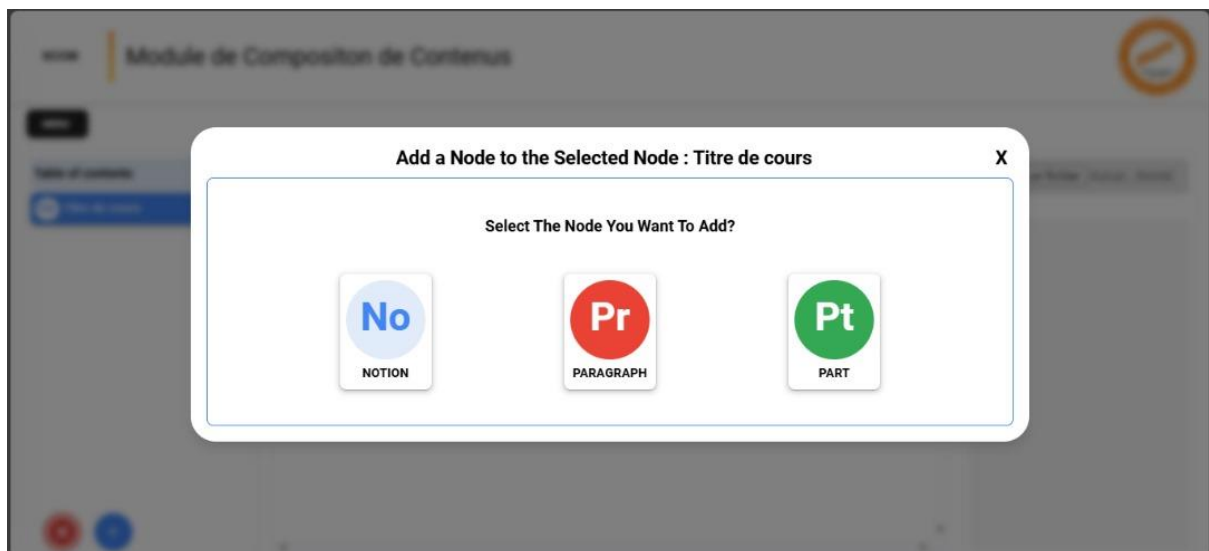


Figure 24 Ajout d'un nouvel élément 1

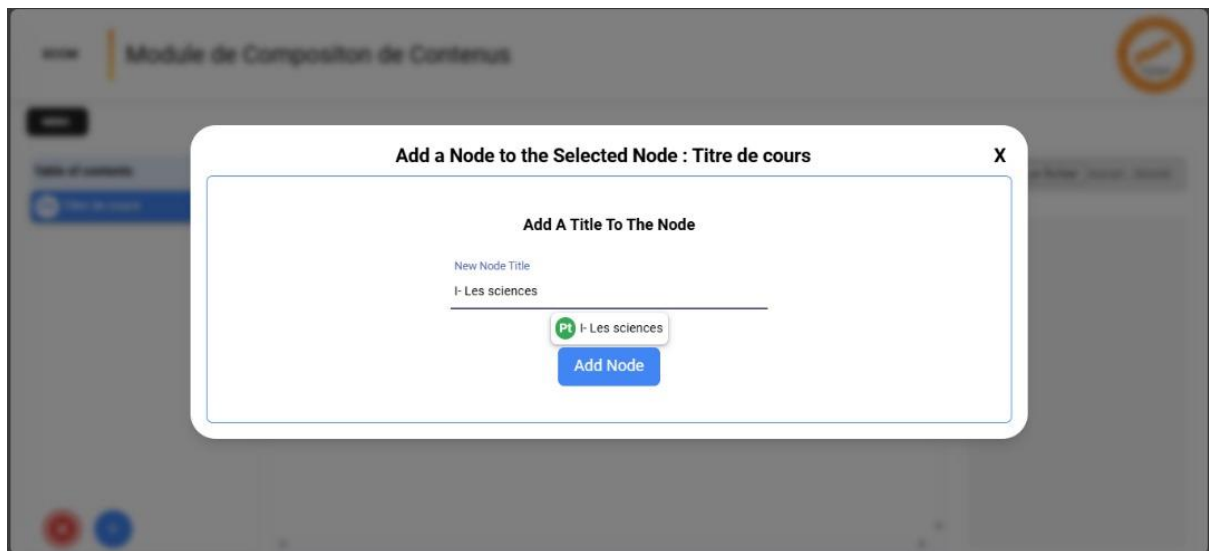


Figure 25 Ajout d'un nouvel élément 2

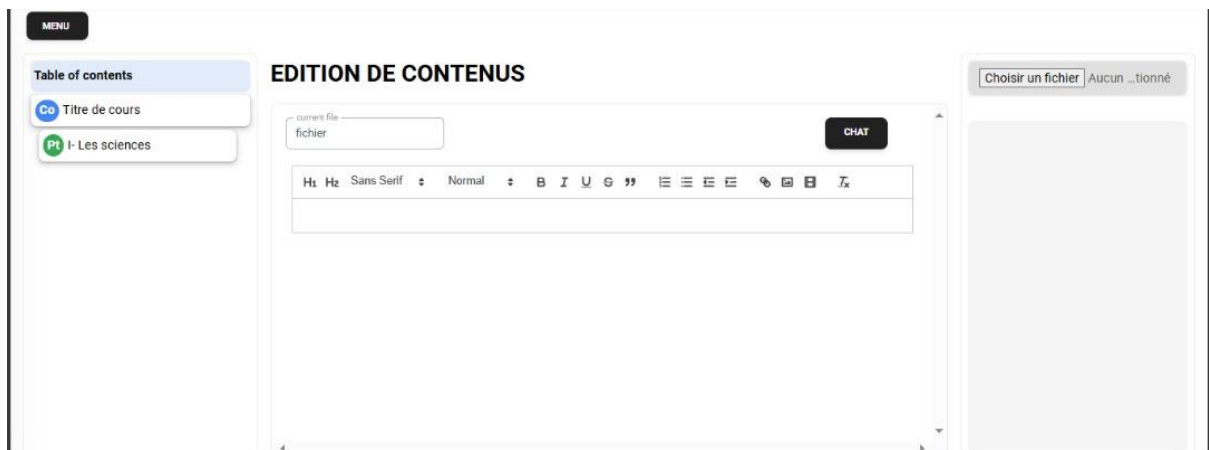


Figure 26 Granule ajoute à gauche

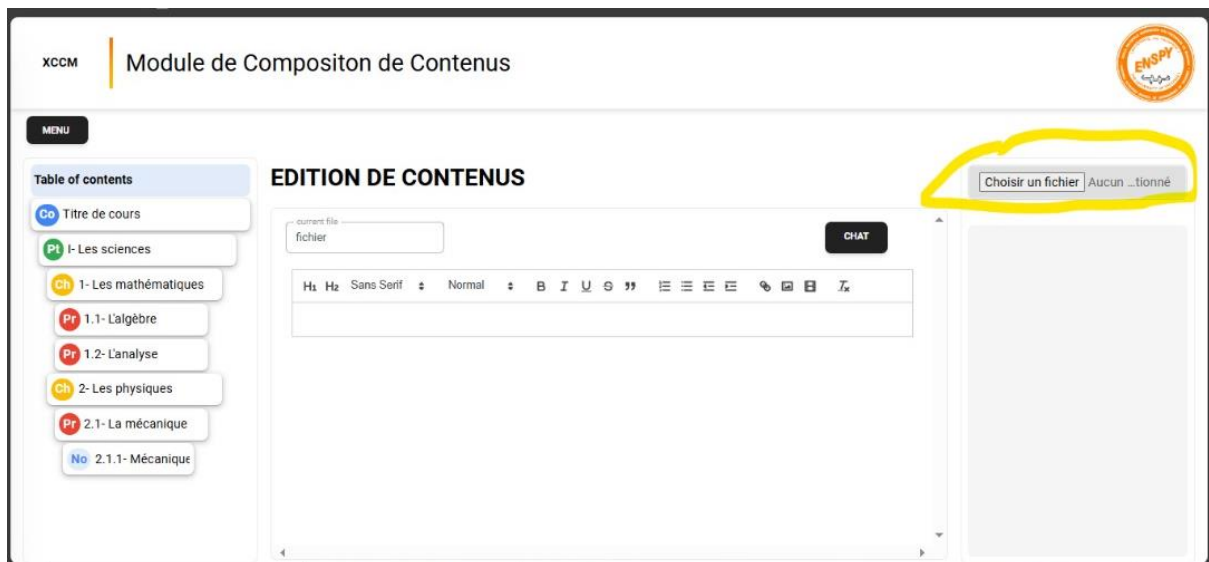


Figure 27 Import d'un granule de XCSM

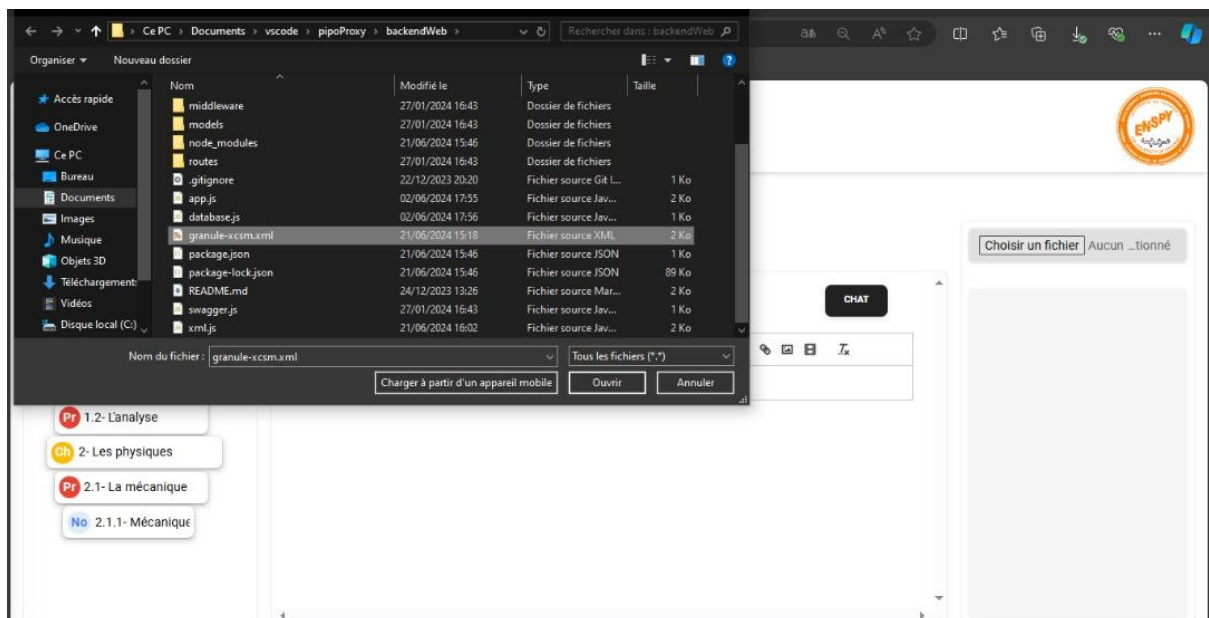


Figure 28 Choix du fichier à importer

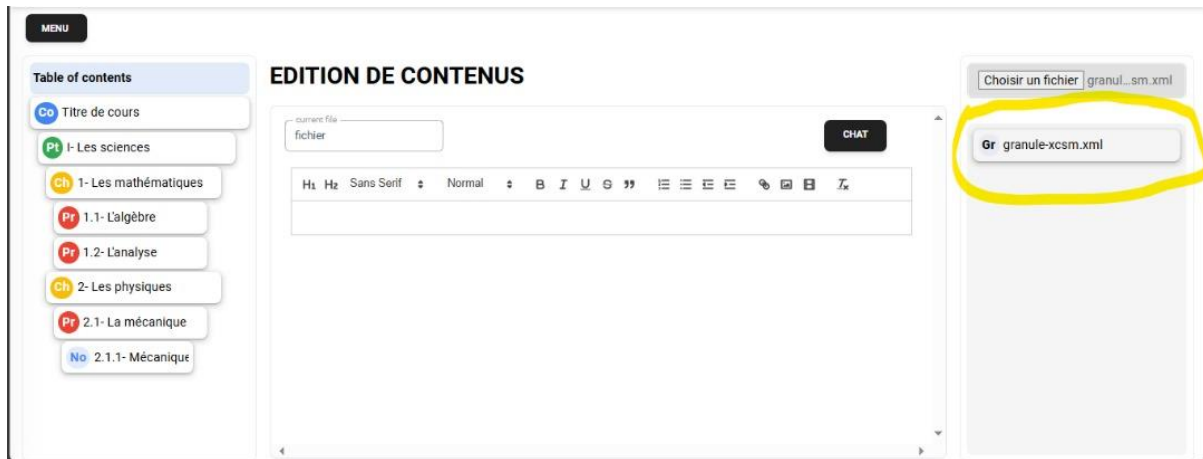


Figure 29 Granule ajouté

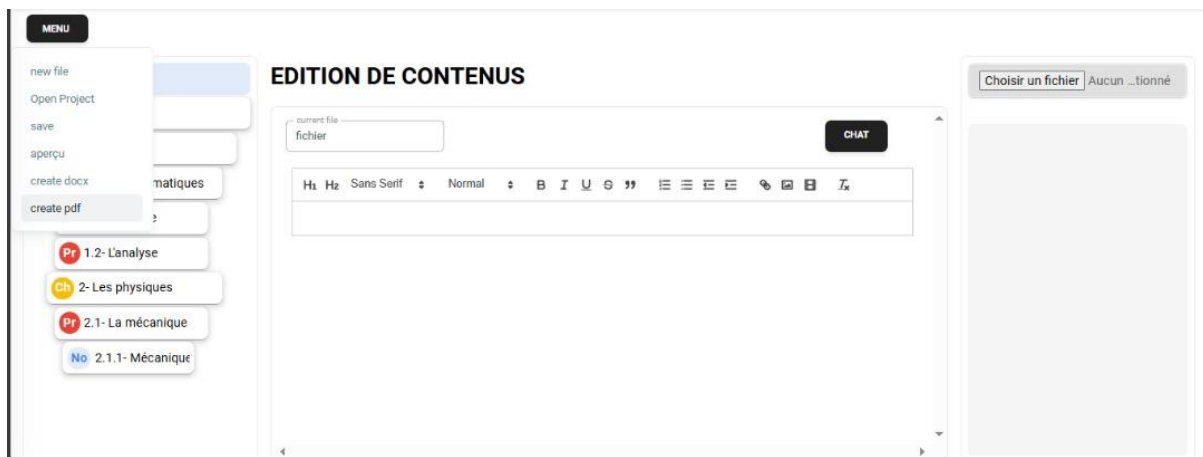


Figure 30 Création de PDF



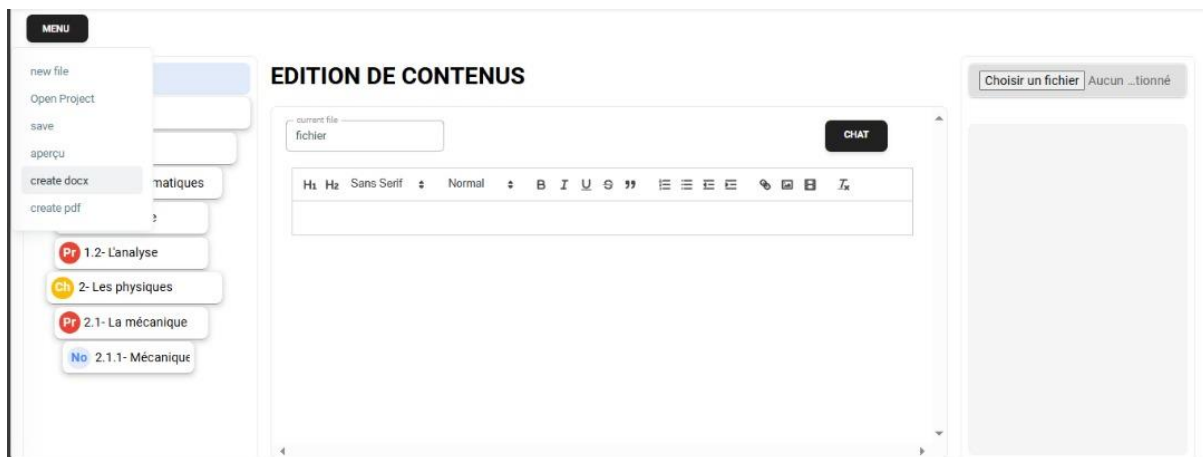


Figure 31 création de fichier Word

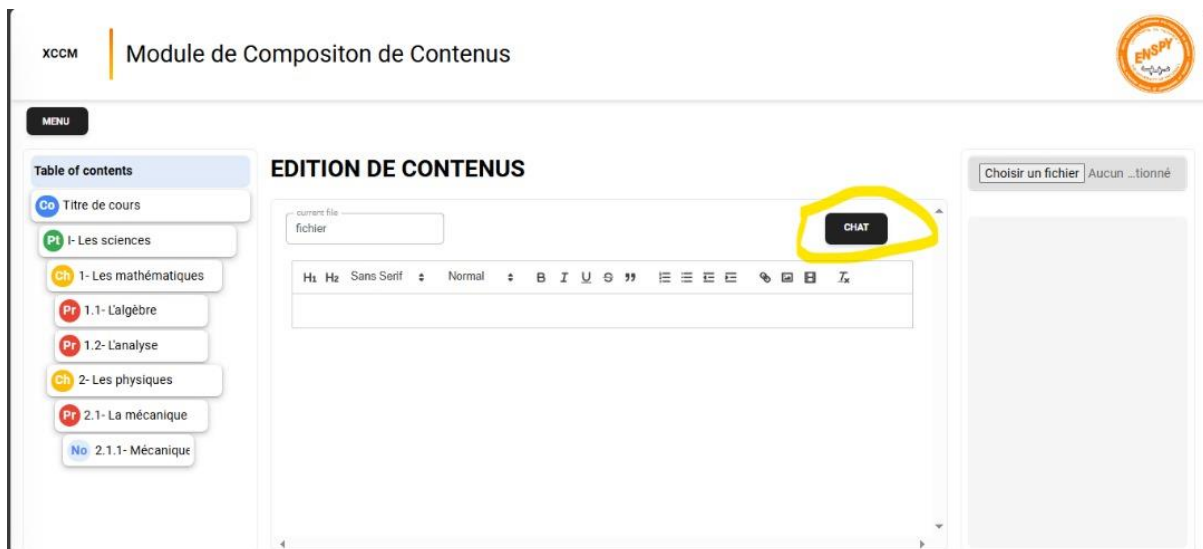


Figure 32 Aperçu du bouton de chat

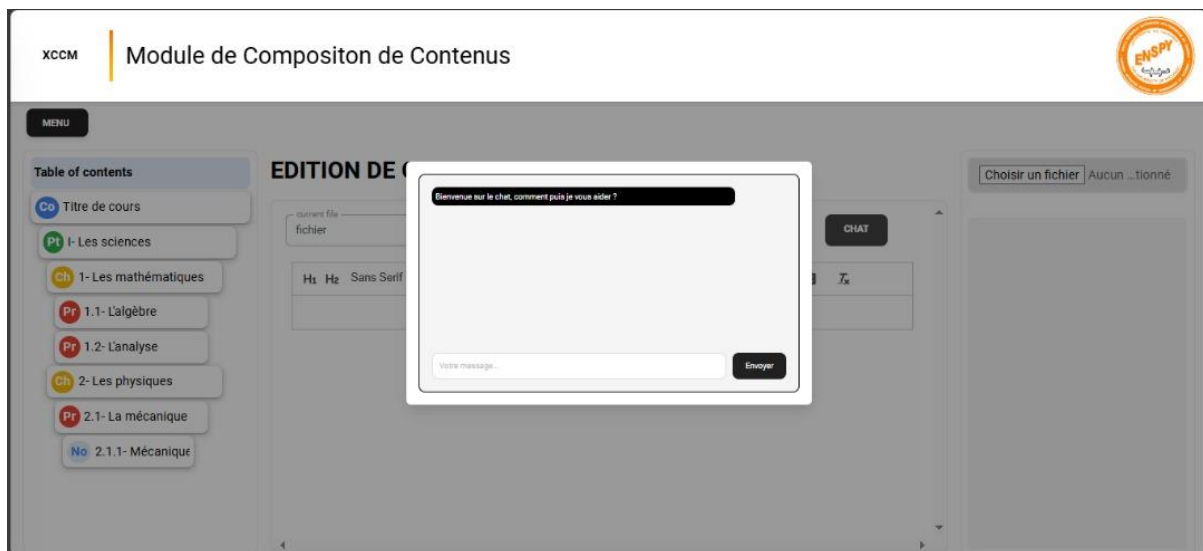


Figure 33 Interface de chat 1

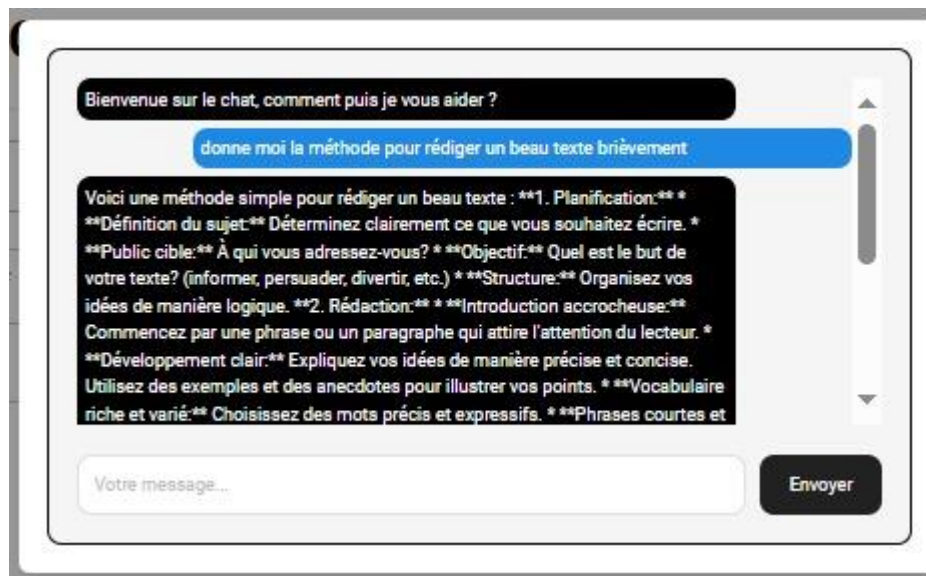


Figure 34 Interface de chat 2

## 8 PERSPECTIVES

---

➤ **Utilisation de GraphQL:**

- L'intégration de GraphQL peut en effet s'avérer très bénéfique pour notre application. GraphQL permet de récupérer uniquement les données dont vous avez besoin, ce qui se traduit par une meilleure performance et une meilleure expérience utilisateur.
- L'utilisation de GraphQL nous permettra de réduire le nombre de requêtes HTTP nécessaires pour récupérer les données, car il regroupe plusieurs requêtes en une seule.

➤ **Fonctionnalités de collaboration :**

- Permettre aux utilisateurs de communiquer et de collaborer sur des cours de groupe.
- Permettre aux utilisateurs de partager des ressources entre eux.
- Offrir des fonctionnalités de discussion, de commentaires et de co-construction de contenus.

## 9 CONCLUSION

---

Le projet XCCM (eXtended Content Composition Module) a été initié dans le but de faciliter la recomposition des contenus pédagogiques en fournissant une plateforme interactive permettant la manipulation de compositions. Pour atteindre le but fixé au départ nous avons présenté clairement le logiciel XCCM, en effectuant une analyse approfondie de l'existant, en apportant des améliorations significatives au projet, et en spécifiant rigoureusement les besoins et exigences, et ensuite nous sommes passé à la phase d'analyse, de conception, de développement, et enfin, le guide d'utilisation qui fournit une séquence claire des étapes, depuis l'inscription jusqu'à la création et l'édition de compositions. Les interfaces ont été présentées de manière visuelle pour faciliter la compréhension de l'utilisateur. Parvenu au terme de la durée de notre travail nous pouvons dire que nous sommes à 75% de l'objectif fixé au départ.

## 10 RÉFÉRENCES BIBLIOGRAPHIQUES

---

[1]- Télécom SudParis , Architecture(s) et application(s) Web, [en ligne], lien internet: <https://www-inf.telecom-sudparis.eu/COURS/CSC4101/cours-toile/poly.pdf> , consulté le: 21/06/2024.

[2]- Moghrani arezki et Azzoug zoubir, Conception et réalisation d'une application de suivi de patients dans un établissement hospitalier, [en ligne], lien internet: [https://www.memoireonline.com/08/09/2577/m\\_Conception-et-realisation-dune-application-de-suivi-de-patients-dans-un-etablissement-hospitalier5.html](https://www.memoireonline.com/08/09/2577/m_Conception-et-realisation-dune-application-de-suivi-de-patients-dans-un-etablissement-hospitalier5.html), consulte le: 21/06/2024.