

# REPUBLIQUE DU SENEGAL



Un Peuple – Un But – Une Foi

Ministère de l'Enseignement Supérieur de la Recherche et de l'innovation



**Statistique et Informatique Décisionnelle**

**Projet :** CONCEPTION ET PROGRAMMATION ORIENTEES OBJET

**Sujet :** *Conception d'un système de gestion des projets de recherche*

**Sous la supervision de :**

**Les membres du groupe :**

**Dr Bara Sall**

**Assime Diakhate**

**Pape ngor Tine**

**Astou wade**

**Ousmane Ndiaye**

**Mouhamadou Ndiaye**

**Adama Sarr**

**Année universitaire : 2024 – 2025**

## Table des matières

Introduction :	4
I. Analyse des Besoins (UML) :	4
1. Diagramme de Cas d'Utilisation	4
a) Identification des acteurs principaux :	5
b) Cas d'utilisations principaux :	5
c) Relations entre Acteurs et Cas d'Utilisation :	5
d) Inclusion et Extension :	5
e) Système :	6
2. Diagramme de séquence :	7
A. Description de diagramme de séquence	7
B. Acteurs Principaux	10
C. Flux Principal (Soumission Valide)	10
II. Diagramme de classe :	15
A. Description du diagramme de classe :	16
B. Décomposons les éléments principaux :	17
1. La classe "Utilisateur" (en haut) :	17
2. Les classes "Etudiant", "Administrateur" et "Professeur" :	18
3. La classe "Collaborateur" :	19
4. La classe "Projets" :	19
Conclusion	21

Figure 1:Description des cas d'utilisations .....	6
Figure 2:Diagramme d'inscription : .....	7
Figure 3:Diagramme d'authentification .....	8
Figure 4:diagramme de soumission d'un projet : .....	9
Figure 5:Diagramme d'affectation d'un superviseur : .....	10
Figure 6: Diagramme de création de collaboration .....	11
Figure 7:Diagramme de participation à une collaboration .....	12
Figure 8: Diagramme de suivi et d'évaluation .....	13
Figure 9:Diagramme de liste d'utilisateur .....	14
Figure 10:description du diagramme de classe .....	16

# Introduction :

Dans le cadre de la formation en Master en informatique, la gestion efficace des projets de recherche constitue un enjeu majeur pour les étudiants, les enseignants et l'administration. Ce projet vise à développer un **système de gestion des projets de recherche** permettant de centraliser et d'optimiser les différentes étapes du cycle de vie d'un pro/jet, depuis sa proposition jusqu'à sa réalisation finale.

L'objectif principal de cette application est d'offrir une plateforme collaborative où les **étudiants** pourront soumettre leurs propositions de projet, les **professeurs** superviser les travaux, et les **administrateurs** gérer l'ensemble du processus. Les fonctionnalités clés incluent l'authentification sécurisée des utilisateurs, la soumission et l'évaluation des projets, l'affectation des superviseurs, ainsi que des outils de suivi et de collaboration pour les équipes étudiantes.

Pour garantir une conception robuste et cohérente, ce projet s'appuie sur les principes de la **programmation orientée objet (POO)** et utilise les **diagrammes UML** (cas d'utilisation, séquence et classes) pour modéliser les interactions et l'architecture du système. L'implémentation sera réalisée en **Java**, en intégrant une base de données pour la persistance des informations

## I. Analyse des Besoins (UML) :

L'**analyse des besoins** est une étape fondamentale dans tout projet de développement logiciel. Elle permet de **comprendre, formaliser** et **valider** les attentes des utilisateurs et les contraintes techniques. Le langage **UML (Unified Modeling Language)** est l'outil standardisé idéal pour modéliser ces besoins de manière claire et visuelle.

Les 3 Diagrammes UML Essentiels pour l'Analyse des Besoins :

### 1. Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation est l'un des types de diagrammes UML les plus fondamentaux et les plus largement utilisés dans le domaine du développement logiciel. Il offre une représentation graphique des interactions entre les acteurs externes (utilisateurs) et le système, mettant en évidence les différentes fonctionnalités offertes par le système et les scénarios d'utilisation correspondants. Voici une description détaillée des éléments composant un diagramme de cas d'utilisation.

### a) Identification des acteurs principaux :

- ✓ **Étudiant** : L'étudiant est l'acteur central du système. En tant que porteur de projet, il peut soumettre une nouvelle proposition de recherche, modifier ses projets avant validation, et collaborer avec d'autres étudiants. Il a également accès aux feedbacks des professeurs et peut suivre en temps réel l'avancement de ses travaux.
- ✓ **Le Professeur** : Le professeur joue un rôle d'évaluateur et de superviseur. Il consulte les projets soumis, fournit des évaluations détaillées, et guide les étudiants en leur donnant des recommandations. Il a la capacité de faire évoluer le statut des projets (validation, demande de modifications) et de suivre leur progression.
- ✓ **L'Administrateur** : L'administrateur est le garant du bon fonctionnement du système. Il valide les inscriptions des utilisateurs, archive les projets terminés, et génère des rapports pour le suivi global. Il gère également les droits d'accès et intervient en cas de problème technique ou de conflit.

### b) Cas d'utilisations principaux :

- Un cas d'utilisation représente une fonctionnalité ou un ensemble de fonctionnalités que le système offre à ses utilisateurs.
- Chaque cas d'utilisation décrit une action ou un scénario spécifique que l'utilisateur peut entreprendre avec le système pour atteindre un objectif particulier.
- Les cas d'utilisation sont représentés par des ovales dans le diagramme, avec leur nom à l'intérieur.

### c) Relations entre Acteurs et Cas d'Utilisation :

- Les relations entre les acteurs et les cas d'utilisation indiquent les interactions entre les utilisateurs et le système.
- Une relation indique qu'un acteur est impliqué dans un cas d'utilisation spécifique.

Les relations sont représentées par des lignes reliant les acteurs aux cas d'utilisation correspondants.

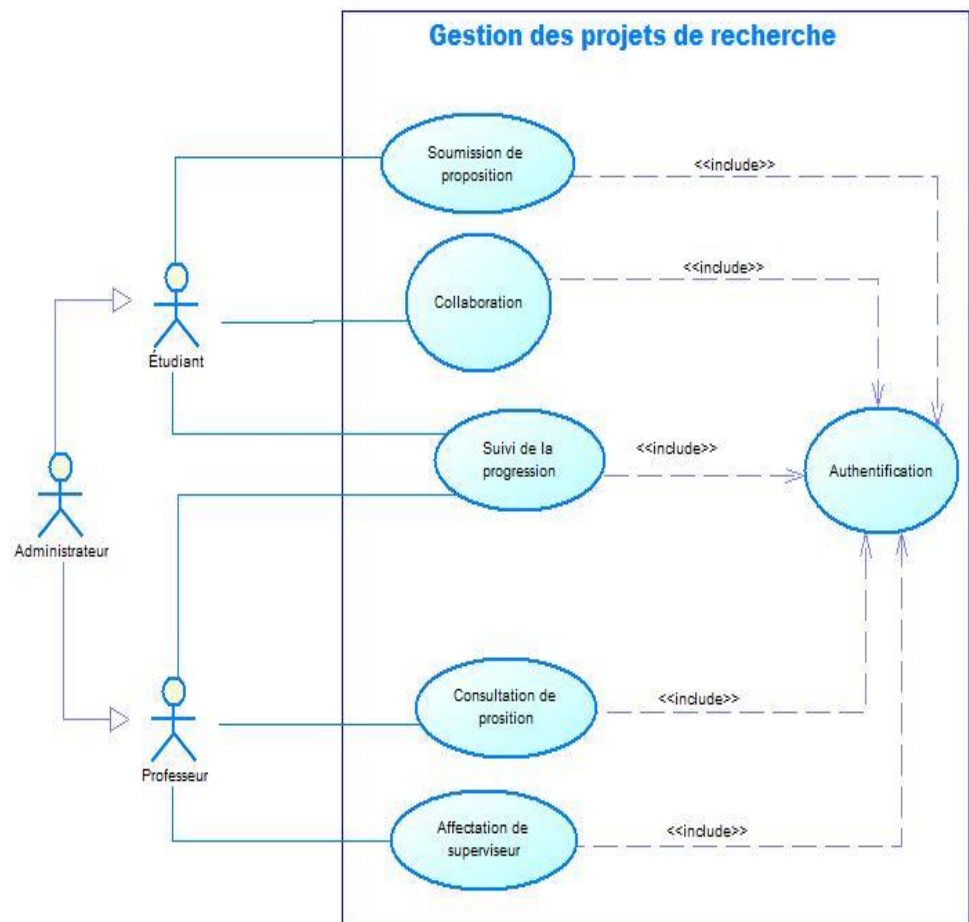
### d) Inclusion et Extension :

- Parfois, les cas d'utilisation peuvent être liés entre eux par des relations d'inclusion ou d'extension.
- L'inclusion indique qu'un cas d'utilisation inclut un autre cas d'utilisation.
- L'extension indique qu'un cas d'utilisation peut être étendu par un autre cas d'utilisation dans certaines circonstances.

### e) Système :

- Le système lui-même est généralement représenté par un rectangle dans le diagramme.
- Ce rectangle englobe tous les cas d'utilisation du système et les acteurs impliqués.

En combinant ces éléments, le diagramme de cas d'utilisation offre une vue d'ensemble claire et intuitive des fonctionnalités offertes par le système et des interactions avec ses utilisateurs. Il sert de point de départ pour la conception et la spécification des exigences du système, facilitant ainsi la communication et la compréhension entre les parties prenantes du projet.



*Figure 1:Description des cas d'utilisations*

Ce diagramme présente les interactions principales entre les acteurs et le système de gestion des projets de recherche. Il identifie trois acteurs clés et leurs fonctionnalités associées, avec des relations de dépendance marquées par des stéréotypes <<include>>.

## 2. Diagramme de séquence :

### A. Description de diagramme de séquence

Le diagramme de séquence est une représentation graphique des interactions entre les utilisateurs (étudiants, professeurs, administrateurs) et le système selon un ordre chronologique dans la formulation UML. On montre ces interactions dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Le but étant de décrire comment se déroulent les actions entre les acteurs ou les objets. Voici les différents schémas du diagramme de séquence pour ce projet.



*Figure 2:Diagramme d'inscription :*

Ce diagramme modélise le processus d'inscription d'un utilisateur dans un système, avec les interactions entre l'**Utilisateur** et le **Système** :

#### ✓ Acteurs et Objets

**Utilisateur** : Initie la demande d'inscription.

**Système** : Traite la demande et valide les données.

#### ✓ Flux Principal (Happy Path)

##### • Demande d'inscription :

L'utilisateur déclenche le processus (Demande Inscription).

- **Remplissage du formulaire :**

L'utilisateur saisit ses informations (Remplir Formulaire).

Le système confirme la réception (Formulaire Rempli).

- **Validation :**

Si les paramètres sont corrects : Le système enregistre l'utilisateur (Inscription Réussie).

✓ **Flux Alternatif (Erreur)**

- *Si les paramètres sont incorrects :*

Le système renvoie un message d'erreur (Paramètre Incorrectes Réessayer).

L'utilisateur doit corriger et réessayer.



Figure 3: Diagramme d'authentification

**1. Acteurs :**

- **Utilisateur** : personne qui souhaite s'inscrire.
- **Système** : application qui gère l'inscription.

**2. Déroulement de l'inscription :**

a. **Demande d'inscription :**



L'utilisateur initie une demande d'inscription en envoyant un message au système (Demande Inscription).

**b. Remplir le formulaire :**

Le système répond en envoyant un formulaire d'inscription que l'utilisateur doit remplir (Remplir Formulaire).

**c. Formulaire rempli :**

Une fois que l'utilisateur a complété le formulaire, il l'envoie de retour au système (Formulaire Rempli).

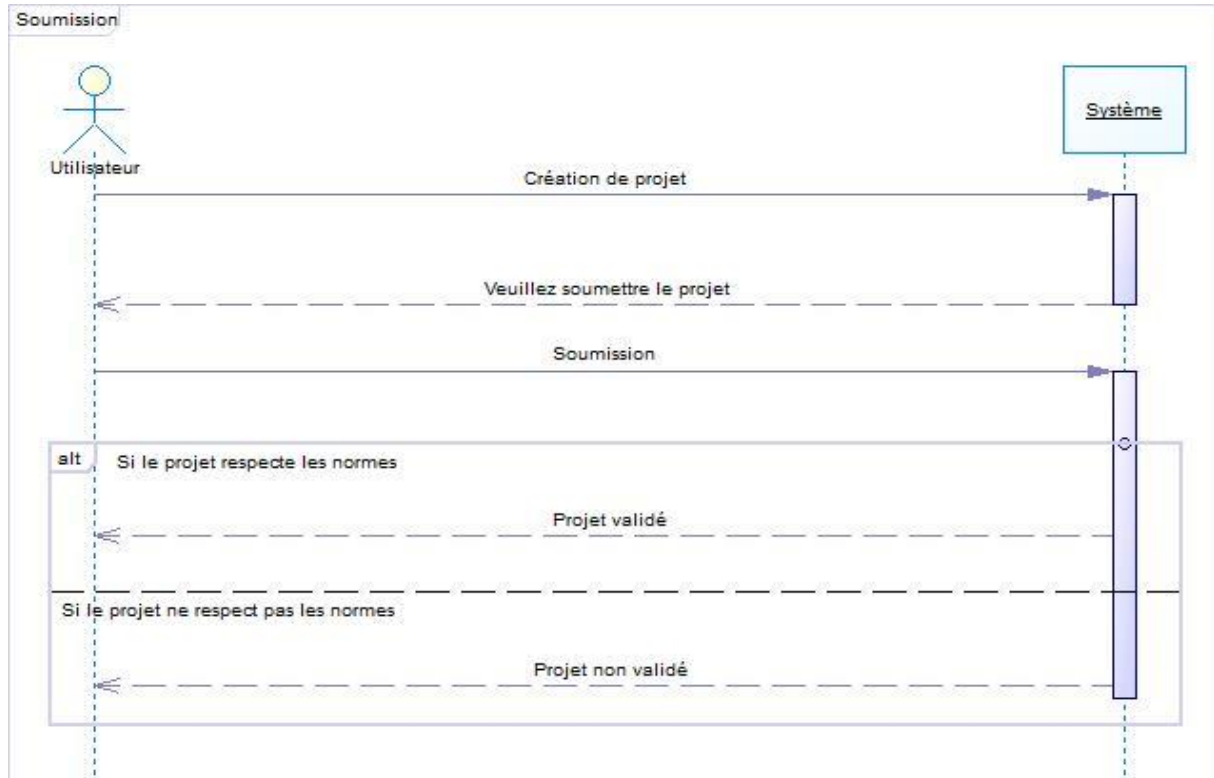
**d. Vérification des paramètres** (Bloc alt pour l'alternative) :

**Si les paramètres sont corrects** (par exemple : email valide, mot de passe conforme, etc.) :

Le système enregistre l'utilisateur et retourne un message de succès (Inscription Réussie).

**Si les paramètres sont incorrects** (par exemple : email invalide, mot de passe trop faible) :

Le système informe l'utilisateur de l'erreur (Paramètre Incorrectes Réessayer) et lui demande de corriger et de réessayer.



*Figure 4:diagramme de soumission d'un projet :*

## B. Acteurs Principaux

- **Utilisateur** : L'étudiant ou le chercheur qui soumet un projet
- **Système** : La plateforme de gestion des projets

## C. Flux Principal (Soumission Valide)

### a. Initiation

Le système invite l'utilisateur : *"Veuillez soumettre le projet"*

### b. Soumission

L'utilisateur effectue la Création de projet et confirme la Soumission

### c. Validation

Le système vérifie si le projet respecte les normes

### d. Acceptation

Si conforme : Projet validé

### e. Flux Alternatif (Rejet)

- Si le projet ne respecte pas les normes :  
Le système renvoie : Projet non validé  
L'utilisateur doit modifier et resoumettre

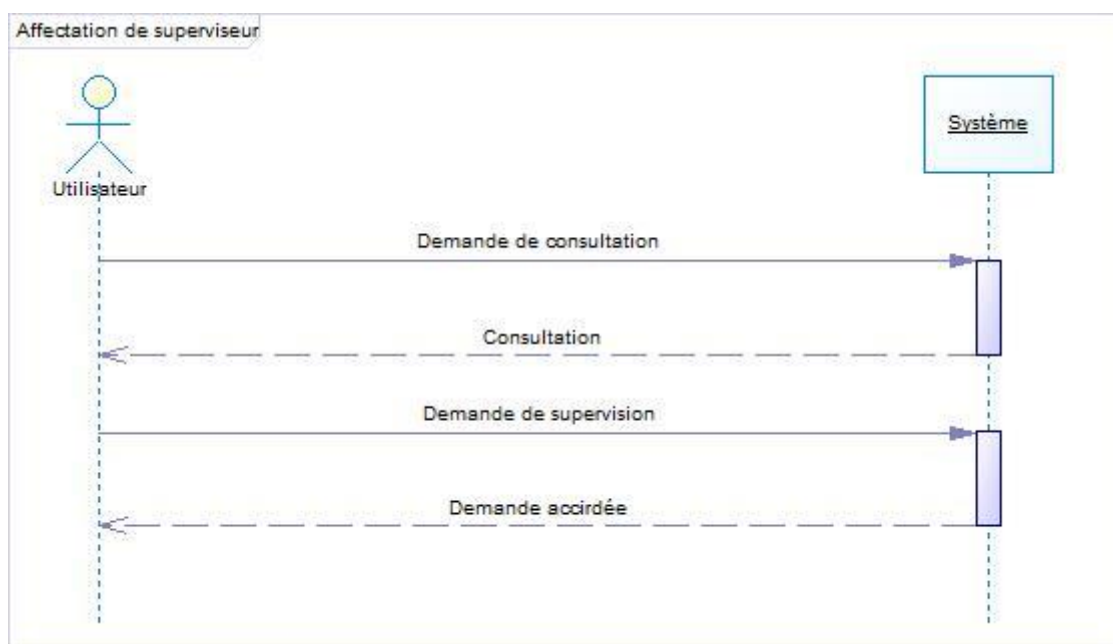


Figure 5: Diagramme d'affectation d'un superviseur :

## 1. Acteurs Impliqués

**Utilisateur** : Probablement un professeur ou administrateur

**Système** : Plateforme de gestion des projets

## 2. Flux Principal

### a. Initiation

L'utilisateur fait une Demande de consultation (liste des projets disponibles)

### b. Sélection

L'utilisateur envoie une Demande de supervision pour un projet spécifique

### c. Confirmation

Le système répond par Demande accédée (probablement "acceptée" avec faute de frappe)

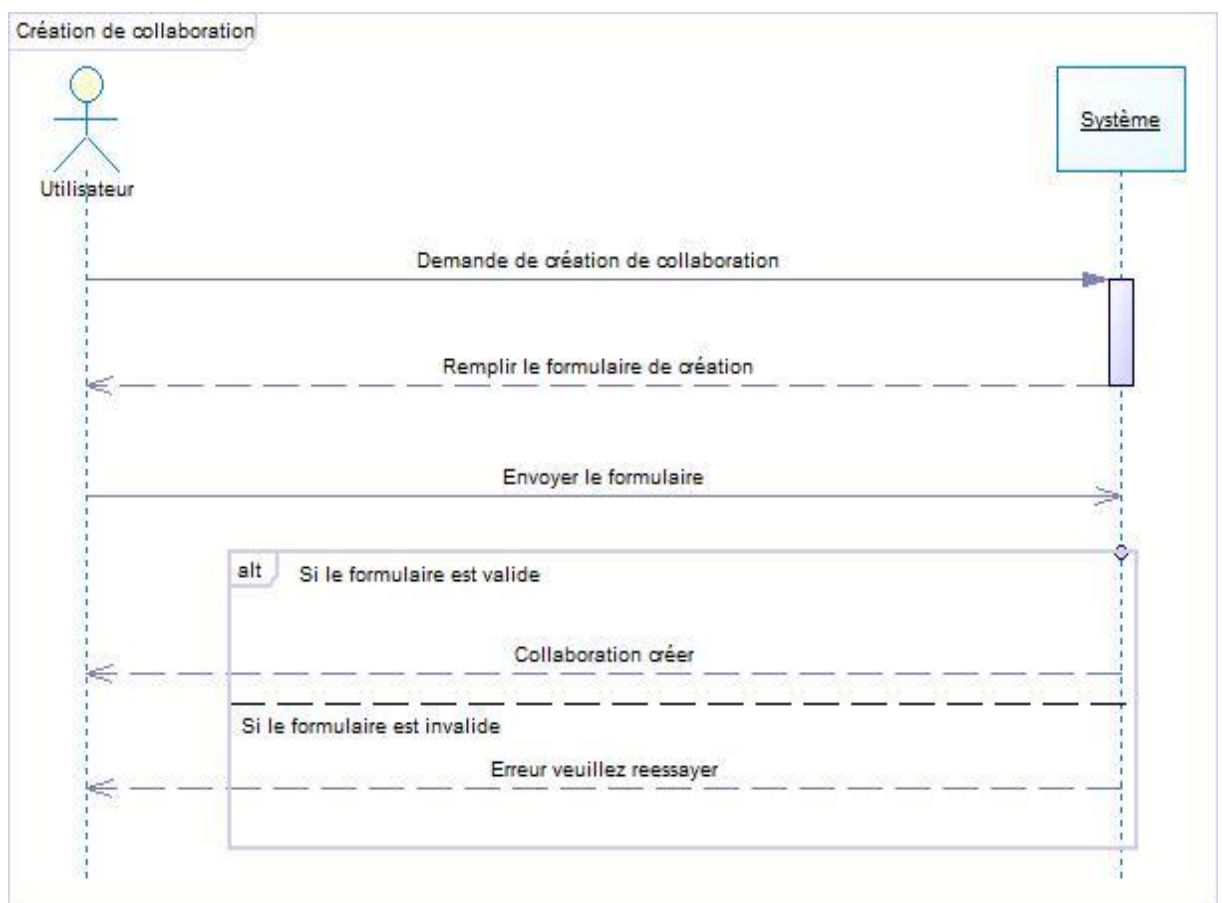


Figure 6: Diagramme de création de collaboration

## 1. Acteurs Principaux

**Utilisateur** : Étudiant ou chercheur souhaitant initier une collaboration

**Système :** Plateforme de gestion des projets collaboratifs

## 2. Flux Principal (Collaboration Valide)

### - Initiation

L'utilisateur fait une Demande de création de collaboration

### - Préparation

L'utilisateur Remplit le formulaire de création (informations requises non précisées)

### - Soumission

L'utilisateur Envoie le formulaire

### - Validation

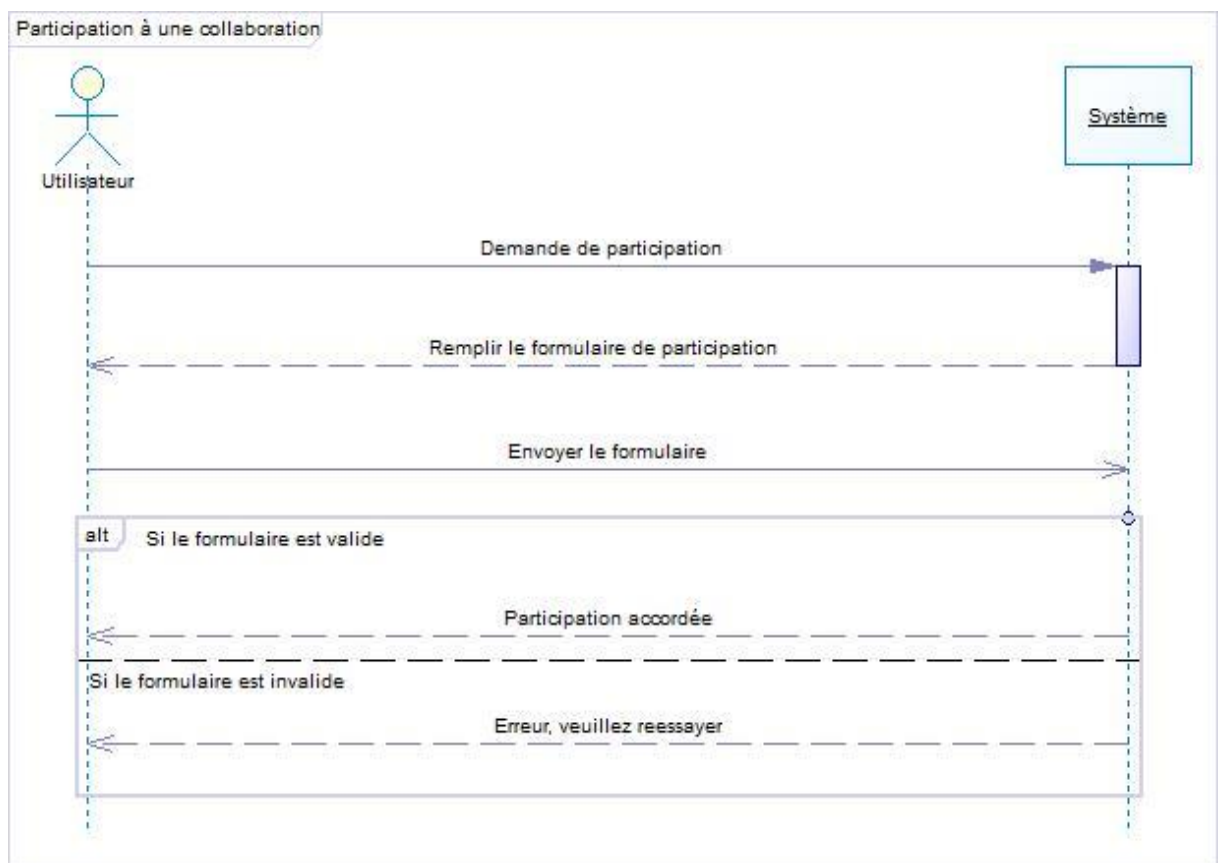
Le système vérifie la validité du formulaire

### - Confirmation

Si valide : Collaboration créée

### - Flux Alternatif (Erreur)

- Si le formulaire est invalide :  
Le système renvoie : Erreur veuillez ressayer  
(Note : Orthographe à corriger en "réessayer")



*Figure 7: Diagramme de participation à une collaboration*

## 1. Acteurs Principaux

**Utilisateur :** Membre souhaitant rejoindre une collaboration existante

**Système** : Plateforme de gestion collaborative

## 2. Flux Principal (Participation Validée)

### a. Initiation

L'utilisateur fait une Demande de participation

### b. Saisie

Remplit un formulaire de participation (champs non précisés)

### c. Soumission

Envoi des données au système

### d. Validation

Vérification par le système (condition S1 = formulaire valide)

### e. Confirmation

Participation accordée

### f. Flux Alternatif (Rejet)

- Si S1 non respecté (formulaire invalide) :

Message d'erreur : Erreur, veuillez réessayer (*À corriger en "réessayer"*)



*Figure 8: Diagramme de suivi et d'évaluation*

## 1. Acteurs Impliqués

**Utilisateur** : Étudiant ou enseignant souhaitant suivre l'avancement

**Système** : Plateforme de gestion des projets

## 2. Flux Principal

**- Initiation**

L'utilisateur envoie une Demande suivi projet

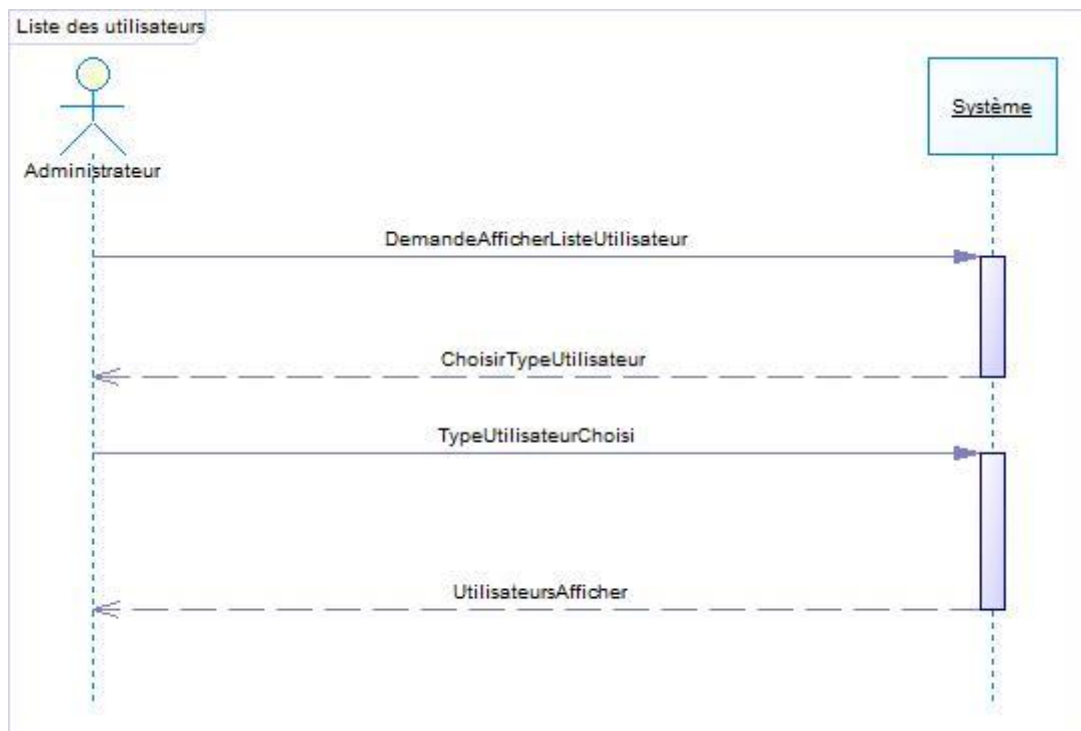
**- élection**

Le système propose une interface de choix

L'utilisateur Choisi le projet (orthographe à corriger en "choisit")

**- Accès**

Le système donne Accès au suivi (Contenu non précisé : tableau de bord ?  
indicateur)



*Figure 9:Diagramme de liste d'utilisateur*

**1. Acteurs Impliqués**

- **Administrateur** : Responsable système avec droits étendus
- **Système** : Plateforme de gestion des utilisateurs

**2. Flux Principal**

**- Initiation**

**L'administrateur envoie une Demande Afficher Liste Utilisateur**

**1. Filtrage**

- Sélection d'un Choisir Type Utilisateur (étudiants/professeurs/admins)

**2. Récupération**

- Le système confirme Type Utilisateur Choisi

**3. Affichage**

- Liste filtrée Utilisateurs Afficher

## II. Diagramme de classe :

Le diagramme de classe est un outil de modélisation utilisé en génie logiciel pour représenter la structure statique d'un système en termes de classes, d'attributs, de méthodes et de relations entre les classes. Pour le système de gestion des projets de recherche des étudiants en Master en informatique, voici une description du diagramme de classe :

### Classes :

- Les classes représentent les entités principales du système, telles que les étudiants, les professeurs, les projets, les propositions de projet, etc.
- Chaque classe est représentée par un rectangle contenant son nom.

### Attributs :

- Les attributs sont les caractéristiques des classes, représentant les données qu'elles contiennent.
- Par exemple, la classe Étudiant pourrait avoir des attributs tels que id Etudiant, nom, prénom, email, compétences, etc.
- Les attributs sont listés sous le nom de la classe dans le rectangle de la classe.

### Méthodes :

- Les méthodes sont les actions ou les opérations que les objets de classe peuvent effectuer.
- Par exemple, la classe Étudiant pourrait avoir des méthodes telles que soumettre Proposition(), rejoindre Projet(), etc.
- Les méthodes sont listées sous les attributs ou dans une section distincte du rectangle de la classe.

### Relations entre les Classes :

- Les relations entre les classes indiquent comment elles sont liées les unes aux autres dans le système.
- Par exemple, il pourrait y avoir une relation entre Étudiant et Projet pour représenter la participation des étudiants à des projets.
- Les relations sont représentées par des lignes entre les classes, avec des indicateurs de multiplicité pour spécifier le nombre d'objets de classe associés à chaque côté de la relation.

### Héritage :

- L'héritage est utilisé pour modéliser la relation "est-un" entre les classes, où une classe hérite des attributs et des méthodes d'une autre classe.

- Par exemple, une classe Étudiant pourrait hériter des attributs et des méthodes de la classe Utilisateur.
- L'héritage est représenté par une flèche pointant vers la classe parente à partir de la classe enfant.

En résumé, le diagramme de classe pour le système de gestion des projets de recherche des étudiants en Master en informatique permet de visualiser la structure statique du système, y compris ses entités principales, leurs attributs et méthodes, ainsi que les relations entre elles. Il fournit une base solide pour la conception et l'implémentation du système, en aidant les développeurs à comprendre ses composants et leurs interactions.

## A.Description du diagramme de classe :

Le diagramme de classe pour le système de gestion des projets de recherche des étudiants en Master en informatique serait une représentation statique de la structure du système, mettant en évidence les entités principales, leurs attributs, leurs méthodes et les relations entre elles. Voici le schéma du diagramme de classe de ce projet.

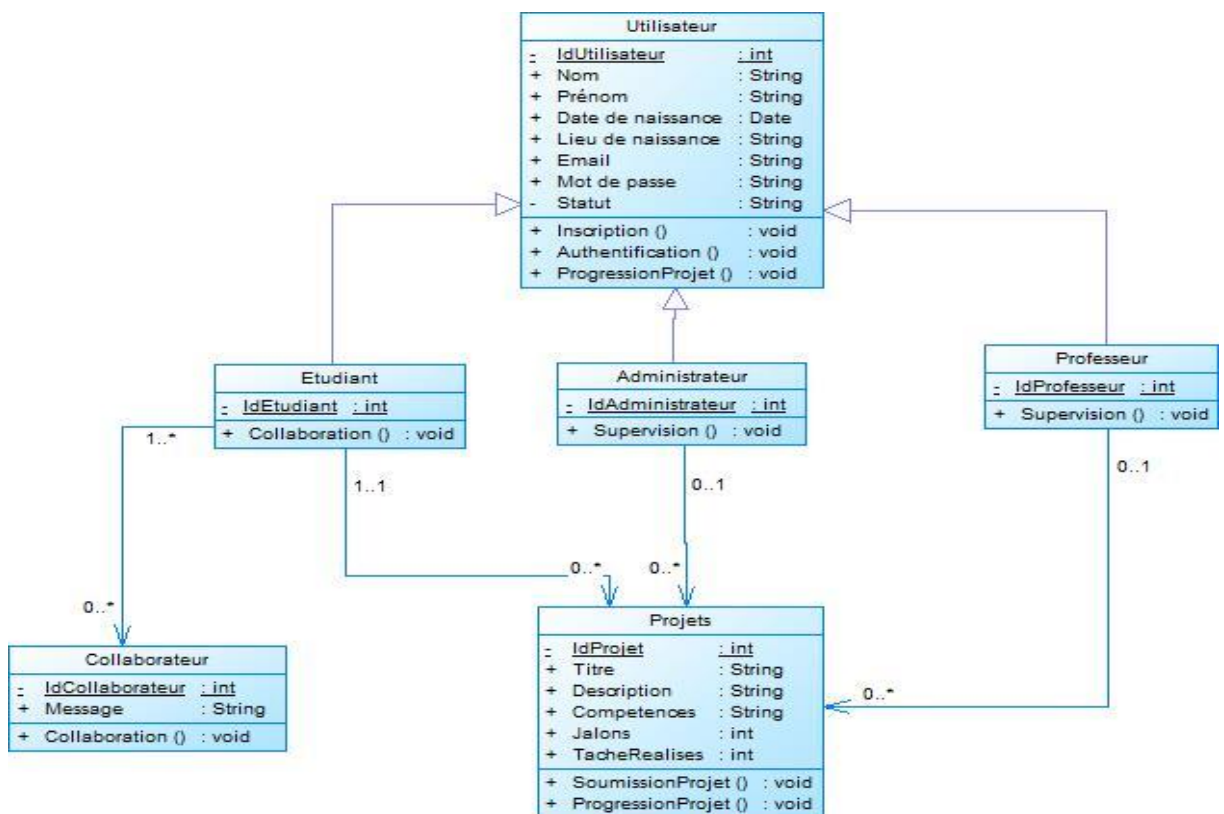


Figure 10: description du diagramme de classe

Ce diagramme représente un système de gestion de projets collaboratifs avec différents types d'utilisateurs : des étudiants, des administrateurs et des professeurs. Il met en évidence les



informations et les actions associées à chaque type d'utilisateur ainsi qu'aux projets eux-mêmes.

## B. Décomposons les éléments principaux :

### 1. La classe "Utilisateur" (en haut) :

- Elle représente un utilisateur générique du système.
- Elle possède des attributs communs à tous les utilisateurs :
  - IdUtilisateur (identifiant unique, de type entier)
  - Nom (nom de l'utilisateur, de type chaîne de caractères)
  - Prénom (prénom de l'utilisateur, de type chaîne de caractères)
  - Date de naissance (date de naissance, de type Date)
  - Lieu de naissance (lieu de naissance, de type chaîne de caractères)
  - Email (adresse email, de type chaîne de caractères)
  - Mot de passe (mot de passe, de type chaîne de caractères)
  - Statut (statut de l'utilisateur, de type chaîne de caractères)
- Elle définit également des opérations (actions que les utilisateurs peuvent effectuer) :
  - Inscription() (action de s'inscrire au système, ne renvoie rien - void)
  - Authentification() (action de se connecter, ne renvoie rien - void)
  - Progression Projet() (action de visualiser la progression d'un projet, ne renvoie rien - void)

## 2. Les classes "Etudiant", "Administrateur" et "Professeur" :

Elles héritent de la classe "Utilisateur" (la flèche avec un triangle vide pointe vers la classe mère). Cela signifie qu'elles possèdent tous les attributs et opérations de la classe "Utilisateur" et peuvent en avoir des spécifiques.

### ✓ "Etudiant" :

- Possède un attribut spécifique : IdEtudiant (identifiant unique de l'étudiant, de type entier).
- A une opération spécifique : Collaboration() (action liée à la collaboration, ne renvoie rien - void).
- Est en relation "un-à-un" (1...1) avec la classe "Collaborateur".

### ✓ "Administrateur" :

- Possède un attribut spécifique : Id Administrateur (identifiant unique de l'administrateur, de type entier).
- A une opération spécifique : Supervision() (action de superviser, ne renvoie rien - void).
- Est en relation "zéro-à-plusieurs" (0...\*) avec la classe "Projets" (un administrateur peut superviser plusieurs projets ou aucun).

### ✓ "Professeur" :

- Possède un attribut spécifique : Id Professeur (identifiant unique du professeur, de type entier).
- A une opération spécifique : Supervision() (action de superviser, ne renvoie rien - void).
- Est en relation "zéro-à-plusieurs" (0...\*) avec la classe "Projets" (un professeur peut superviser plusieurs projets ou aucun).

### 3. La classe "Collaborateur" :

- Représente une entité participant à une collaboration (probablement un étudiant).
- Possède des attributs :
  - IdCollaborateur (identifiant unique du collaborateur, de type entier)
  - Message (un message associé à la collaboration, de type chaîne de caractères)
- A une opération : Collaboration() (action liée à la collaboration, ne renvoie rien - void).
- Est en relation "zéro-à-plusieurs" (0...\*) avec la classe "Projets" (un collaborateur peut participer à plusieurs projets ou aucun).

### 4. La classe "Projets" :

- Représente un projet.
- Possède des attributs :
  - IdProjet (identifiant unique du projet, de type entier)
  - Titre (titre du projet, de type chaîne de caractères)
  - Description (description du projet, de type chaîne de caractères)
  - Compétences (compétences requises pour le projet, de type chaîne de caractères)
  - Jalons (nombre de jalons du projet, de type entier)
  - Tache Réalisées (nombre de tâches réalisées dans le projet, de type entier)

- Définit des opérations :
  - Soumission Projet() (action de soumettre un projet, ne renvoie rien - void)
  - Progression Projet() (action de visualiser la progression du projet, ne renvoie rien - void)
- Est en relation avec les autres classes :
  - Relation "un-à-un" (1...1) avec "Etudiant" (un étudiant est associé à un collaborateur).
  - Relation "zéro-à-plusieurs" (0...\*) avec "Administrateur" (un projet peut être supervisé par zéro ou plusieurs administrateurs).
  - Relation "zéro-à-plusieurs" (0...\*) avec "Professeur" (un projet peut être supervisé par zéro ou plusieurs professeurs).
  - Relation "zéro-à-plusieurs" (0...\*) avec "Collaborateur" (un projet peut avoir zéro ou plusieurs collaborateurs).

En conclusion, ce diagramme de classes décrit la structure d'un système où des étudiants peuvent collaborer sur des projets, supervisés par des administrateurs et des professeurs. Il met en évidence les informations essentielles pour chaque entité et les relations entre elles.

# Conclusion

Le projet consiste à concevoir et développer un système de gestion de projets de recherche pour les étudiants en Master en informatique. La démarche inclut la modélisation du système à l'aide de diagrammes UML (cas d'utilisation, séquence, et classe) afin de définir les interactions utilisateur-système et la structure des données. Le système permettra aux étudiants de soumettre des propositions de projet, aux professeurs de les superviser, et aux administrateurs de gérer l'ensemble du processus. Les étudiants pourront également collaborer sur les projets et suivre leur progression. La phase de développement impliquera l'implémentation du système en Java, en appliquant les principes de la conception orientée objet.