

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

CENTRE UNIVERSITAIRE DE MILA
INSTITUT DES SCIENCES ET DE LA TECHNOLOGIE

Réf. /11

Mémoire de fin d'étude
Présenté pour l'obtention du diplôme de

Licence Académique

Domaine : **Mathématiques et Informatique**
Filière : **Informatique**
Spécialité : **Informatique générale**

Thème

*Développement une application client/serveur dotée
d'un moteur de recherche pour l'archivage et la
consultation des mémoires de fin d'étude du centre
Universitaire de Mila*

Présenté par :

BOUZERAA NABILA

BENDJEDDOU FATIMA

Dirigé par :

Mr. BOUKHECHEM NADIR

Année universitaire 2010-2011

Remerciement

Nous Remercions En tout premier lieu ALLAH le tout puissant qui m'a donné la force, la volonté et le courage pour accomplir ce modeste travail.

Nous tenons ici à remercier Mr Boukhechem Nadir, notre encadreur pour son aide et sa grande patience qu'elle a apporté tout au long la préparation de ce mémoire, ses conseils, ses orientations et encouragements qui ont contribué notablement à la réussite du travail à ce niveau, que dieu le protège.

Nous a dressons aussi nos remerciements à tous les enseignant et ingénieur ne sans oublie pas : I.Boussaid Houssin,

I.Kerioui youcef, I.Boulfakhar samra et I.Amimor najet.

Enfin nous remercions toutes les personnes qui de près ou de loin ont contribués à l'élaboration cette étude.

Merci à tout.

Fatima. Nabila

Dédicace :

Je tiens en tout premier lieu à remercier le dieu. Je voudrais dédicace ce modeste travail

A mes très cher parents qui m'ont tant soutenu et encouragé dans tous les domaines et surtout pour réaliser ce mémoire que dieu les protèges.

A ma chère grande mère qui m'a soutenu et encourager et à l'âme de mon cher grand père

A ma chère tante Djamila, qui est mon modèle dans la vie et ma source de fierté

A mes tantes : Mona, Radjia et son mari Khaled et sa fille

Meriem, Imane, mes oncles Farse et Mohammed

A mes sœurs : Samira, Randa, Hala, Anfale

A mes frères : Azadine, Hicham

A mes cousines : Naima, Halima, Amina

A mon ami binôme Fatima et sa famille.

A mes amies : Nada, Meriem, Souad, Aicha, Meriem, Amina, Rokia, Amina

A toute Personne qui me connaît

BOUZERAA NABILA

Dédicace :

Je tiens en tout premier lieu à remercier le dieu Je voudrais

dédie ce modeste travail

*A mon père qu'il à été toujours pour moi et qui
m'a soutenu toute sa vie, que dieu le protège.*

*A celle qui a cru en moi, et qui m'a toujours soutenu
pendant toutes ces longues années avec ses conseils et
orientations ; à toi chère et adorable mère j'aime de plus
profond de mon cœur, que dieu le protège.*

A ma très chère grande mère.

*A mes chères frères : « Mohamed, Houssam, Deradji, abd
raouf »*

A toute mes oncles et mes tantes.

A mon ami binôme Nabila et sa famille.

A toute mes amis avec les quelles j'ai

partagé les bons moments :Nabila,Soumia,Nada,Zineb,

Asma,Souad, aicha, Meriem, Amina, Rokia, Amina

A tout Person qui me connaît

BENDJEDDOU FATIMA

Résumé :

Le moteur de recherche est un programme informatisé qui permet de rechercher des informations à travers des millions de documents. Les programmes de recherche se basent sur des mots clés pertinents et retournent une liste qui donne accès aux documents contenant les informations.

Dans ce mémoire on a développé une application Client/serveur doté d'un moteur de recherche pour archiver et consulter les mémoires de fin d'étude de la bibliothèque du centre universitaire de Mila. Pour réaliser notre travail on a utilisé UML comme langage de modélisation, JAVA comme langage de programmation et le SGBD MYSQL pour l'implémentation de la base de données.

Mots clés : Moteur de recherche, L'indexation, Orienté Objet, UML, JAVA, MySQL, Architecture Client/serveur.

ملخص:

محرك البحث هو عبارة عن برنامج ألي يتيح لنا البحث عن المعلومات من خلال ملايين الوثائق. وتستند برامج البحث على كلمات رئيسية ذات الصلة, كما يقوم بإرجاع قائمة توفر الوصول إلى الوثائق المطلوبة. في مذكرتنا هذه طورنا تطبيق هندسة الخادم/العميل (Client/serveur) التي تختص بمحرك البحث من أجل تخزين و إظهار مذكرات التخرج الخاصة بالمكتبة المركزية للمركز الجامعي ميلية. من أجل تحقيق هذا الهدف استعملنا مجموعة من الوسائل التي تتمثل في: طريقة تسمح لنا بتطوير النظام UML, لغة البرمجة JAVA بالإضافة إلى نظام تسيير قواعد البيانات MySQL. كلمات مفتاحية : محرك البحث, الفهرسة , MySQL, JAVA, UML, Orienté Objet , هندسة الخادم/العميل

Sommaire

| | |
|--|----|
| Introduction générale. | 1 |
| Chapitre I : « Moteur de recherche et l'indexation des documents » | |
| Introduction | 4 |
| I.1. Le moteur de recherche..... | 4 |
| I.1.1. Fonctionnement des moteurs de recherche | 4 |
| I.1.1.1. L'exploration ou <i>crawl</i> | 4 |
| I.1.1.2. L'indexation..... | 4 |
| I.1.1.3. La recherche..... | 5 |
| I.1.2. Domaines d'utilisation des moteurs de recherche | 5 |
| I.1.3. La Recherche par le contenu | 6 |
| I.2. L'indexation du document | 6 |
| I.2.1. L'indexation. | 6 |
| I.2.2. Les différentes formes d'un index | 7 |
| I.2.3. Les techniques de l'indexation..... | 7 |
| I.2.3.1. Indexation manuelle | 7 |
| I.2.3.2. Indexation automatique | 7 |
| I.2.3.3. Indexation semi-automatique..... | 8 |
| I.2.4. Processus d'indexation automatique..... | 8 |
| I.2.4.1. L'extraction des mots-clés du document..... | 8 |
| I.2.4.2. La normalisation des mots-clés du document | 9 |
| I.2.4.3. L'élimination des mots vides..... | 9 |
| I.2.4.4. La pondération des mots-clés. | 9 |
| Conclusion..... | 10 |
| Chapitre II : « Présentation de UML et l'architecture client/serveur » | |

| | |
|--|-----------|
| Introduction. | 12 |
| II.1. Le langage de modélisation UML. | 12 |
| II.1.1. Les diagrammes UML. | 12 |
| II.1.1.1. Diagrammes de cas d'utilisation..... | 13 |
| II.1.1.2. Diagramme de classe. | 14 |
| II.1.1.3. Diagramme de séquence. | 15 |
| II.1.1.4. Diagramme d'activités. | 16 |
| II.1.1.5. Diagramme de déploiement..... | 16 |
| II.2. Réseaux informatique | 17 |
| II.2.1. Définition d'un serveur | 17 |
| II.2.2. Définition d'un client | 17 |
| II.2.3. Architecture Client/serveur | 17 |
| II.2.3.1. Avantages de l'architecture client/serveur | 18 |
| II.2.3.2. Inconvénient du modèle client/serveur. | 18 |
| II.2.4. Fonctionnement d'un système client/serveur | 18 |
| II.2.5. Les différents types d'architecture Client/serveur | 19 |
| II.2.5.1. L'architecture à 2 niveaux | 19 |
| II.2.5.2. L'architecture à 3 niveaux | 19 |
| II.2.5.3. L'architecture d'égal à égal (Peer to Peer)..... | 20 |
| Conclusion. | 21 |
| Chapitre III : « Spécification des besoins » | |
| Introduction. | 23 |
| III.1. Présentation de l'organisme d'accueil..... | 23 |
| III.1.1. Création du Centre Universitaire de Mila..... | 23 |

| | |
|---|----|
| III.1.2. Faculté de centre Universitaire de Mila..... | 23 |
| III.1.3. Présentation de la bibliothèque | 23 |
| III.2. Cahier de charge..... | 24 |
| III.3. Identifier les acteurs. | 25 |
| III.4. Diagramme de cas d'utilisation | 25 |
| III.5. La description des cas d'utilisation..... | 26 |
| III.6. Les diagrammes de séquence système | 32 |
| Conclusion | 40 |
| Chapitre IV : « Phase d'analyse » | |
| Introduction | 43 |
| IV.1. L'analyse du domaine. | 43 |
| IV.2. Identification les concepts du domaine | 43 |
| IV.3. Ajouter des associations entre classes | 44 |
| IV.4. Les diagrammes d'activités | 45 |
| Conclusion | 52 |
| Chapitre V : « Phase de conception » | |
| Introduction. | 54 |
| V.1. Les diagrammes de séquences. | 54 |
| V.2. Diagramme de classes | 63 |
| V.3. Dictionnaire de données. | 64 |
| V.4. Conception de la base de données | 65 |
| V.5. Diagramme déploiement | 67 |
| Conclusion..... | 67 |
| Chapitre VI : « Implémentation » | |

| | |
|---|----|
| Introduction | 69 |
| VI.1. Environnement de développement de l'application | 69 |
| VI.1.1. Le langage de programmation java | 69 |
| VI.1.2. MySQL | 69 |
| VI.1.3. PhpMyAdmin | 69 |
| VI.1.4. Connecteur | 70 |
| VI.1.5. Partie client/serveur | 70 |
| VI.1.5.1. Les threads | 70 |
| VI.1.5.2. Les sockets | 70 |
| VI.1.6. Les bibliothèque poi | 70 |
| VI.1.7. Programme d'indexation..... | 71 |
| VI.2. Quelques interfaces de l'application..... | 72 |
| Conclusion..... | 78 |
| Conclusion générale..... | 79 |

Listes des figures :

| | |
|---|----|
| Figure I.1 : le fonctionnement général d'un moteur de recherche. | 5 |
| Figure I.2: Etapes du processus d'indexation..... | 8 |
| Figure II.3 : La structure du diagramme de cas d'utilisation..... | 14 |
| Figure II.4 : la structure du diagramme de classes | 14 |
| Figure II.5 : la structure du diagramme de séquence système vu comme une boîte noire et remplacé par un ensemble d'objet | 15 |
| Figure II.6 : la structure du diagramme d'activité..... | 16 |
| Figure II.7: Fonctionnement du système client/serveur. | 19 |
| Figure II.8: Architecture Client/serveur à 2 niveaux..... | 19 |
| Figure II.9 : Architecture Client/serveur à 3 niveaux | 20 |
| Figure III.10 : le diagramme de cas d'utilisation | 25 |
| Figure III.11 : Diagramme de séquence système du cas d'utilisation authentifier. í | 32 |
| Figure III.12: Diagramme de séquence système du cas d'utilisation ajouter un utilisateur. | 33 |
| Figure III.13: Diagramme de séquence système du cas d'utilisation modifier un utilisateur | 34 |
| Figure III.14: Diagramme de séquence système du cas d'utilisation supprimer un utilisateur | 35 |
| Figure III.15: Diagramme de séquence système du cas d'utilisation archiver un mémoire..... | 36 |
| Figure III.16 : Diagramme de séquence système du cas d'utilisation supprimer un mémoire..... | 37 |
| Figure III.17 : Diagramme de séquence système du cas d'utilisation rechercher et consulter par titre | 38 |

Introduction générale

L'informatique est l'ensemble des sciences et techniques en rapport avec le traitement automatique de l'information. L'informatique est d'une grande importance dans presque tous les domaines notamment dans la gestion des documents. En générale les documents dans leurs version électroniques se composent majoritairement de texte, d'images, de son et d'animation, les systèmes informatiques doivent être dotés d'un moteur de recherche documentaire puissant et performant, permettant de consulter et accéder efficacement à l'information, non au document en tant que fichier, mais à son contenu. Un tel système résout les problèmes liés à la recherche séquentielle qui oblige l'utilisateur à parcourir tout les fichiers pour trouver les informations désirées.

Notre travail consiste à développer une application pour indexer et rechercher les mémoires de fin d'étude dans la bibliothèque de centre universitaire de Mila. Le but principal de cette application est de faciliter la recherche et la consultation des mémoires de fin d'étude.

Pour modéliser notre système nous avons utilisé le langage de modélisation UML qui permet de décrire et de représenter le système de manière simple et claire pour implémenter notre application nous avons choisi java, c'est un langage orienté objet souple et portable, nous avons aussi utilisé l'architecture client/serveur qui a beaucoup d'avantages par rapport au autres architectures notamment la sécurité.

Notre mémoire est organisé comme suit :

Dans le premier chapitre on présente les moteurs de recherche et on étudie les différentes techniques d'indexation utilisées par les moteurs de recherche.

Le deuxième chapitre présente le langage de modélisation UML (principe et concepts de base, les diagrammes d'UML).et l'architecture Client/serveur.

Le troisième chapitre concerne la spécification des besoins, c'est-à-dire l'identification des cas d'utilisation, la description des scénarios système et les diagrammes de séquences système.

Le quatrième chapitre est consacré à l'analyse du système, c'est-à-dire l'analyse des domaines et les diagrammes d'activités.

Le cinquième chapitre concerne la conception du système, c'est-à-dire construction des diagrammes de séquences, le diagramme de classe et la conception de la base des données.

Le sixième chapitre concerne implémentation, nous présentons les outils de développement et quelques interfaces de notre application.

Conclusion générale.

Chapitre I :

Moteur de recherche et l'indexation des documents

Introduction

I.1 Le moteur de recherche.

I.2. L'indexation du document.

Conclusion

Chapitre I : Moteur de recherche et l'indexation des documents

Introduction

Dans ce chapitre on va présenter les moteurs de recherches, leurs fonctionnalités, le principe de moteur de recherche documentaire, ainsi que leurs domaines d'utilisations de moteur de recherche et les concepts de base de l'indexation des documents.

I.1. Le moteur de recherche

Un moteur de recherche est un outil permettant de localiser une information donnée à partir d'une requête formulée dans un langage spécifique ou via l'utilisation des mots-clés. Les moteurs de recherche sont utilisés sur le net afin de localiser les pages des sites qui correspondent à des mots-clés déclarés. Ces moteurs de recherche disposent de programmes d'indexation automatique des informations disponibles sur le net et d'une base de données regroupant l'ensemble de ces informations. L'internaute peut rechercher les sites en formulant une requête au moteur de recherche. Les moteurs de recherche utilisent des algorithmes complexes pour localiser et renvoyer les informations la plus pertinente possible. Les moteurs de recherche les plus populaires sont Google, Yahoo et Bing.com.

I.1.1. Fonctionnement des moteurs de recherche

Le fonctionnement d'un moteur de recherche se décompose en trois étapes principales :

I.1.1.1. L'exploration ou *crawl*

Le web est systématiquement exploré par un robot d'indexation suivant récursivement tous les hyperliens (des sites) qu'il trouve et récupérant les ressources jugées intéressantes.

I.1.1.2 L'indexation

Elle consiste à extraire les mots considérés comme significatifs des documents collectes. Les mots extraits sont enregistrés dans une base de données organisée comme un gigantesque dictionnaire inverse ou plus exactement, comme l'index terminologique d'un ouvrage qui permet de retrouver rapidement dans quel chapitre de l'ouvrage se situe un terme significatif donné.

Chapitre I : Moteur de recherche et l'indexation des documents

I.1.1.3. La recherche

Elle correspond à la partie requêtes du moteur qui restitue les résultats. Un algorithme est appliqué pour identifier (en utilisant l'index), les documents qui correspondent le mieux aux mots contenus dans la requête.

La figure suivante montre le fonctionnement général d'un moteur de recherche sur le net.

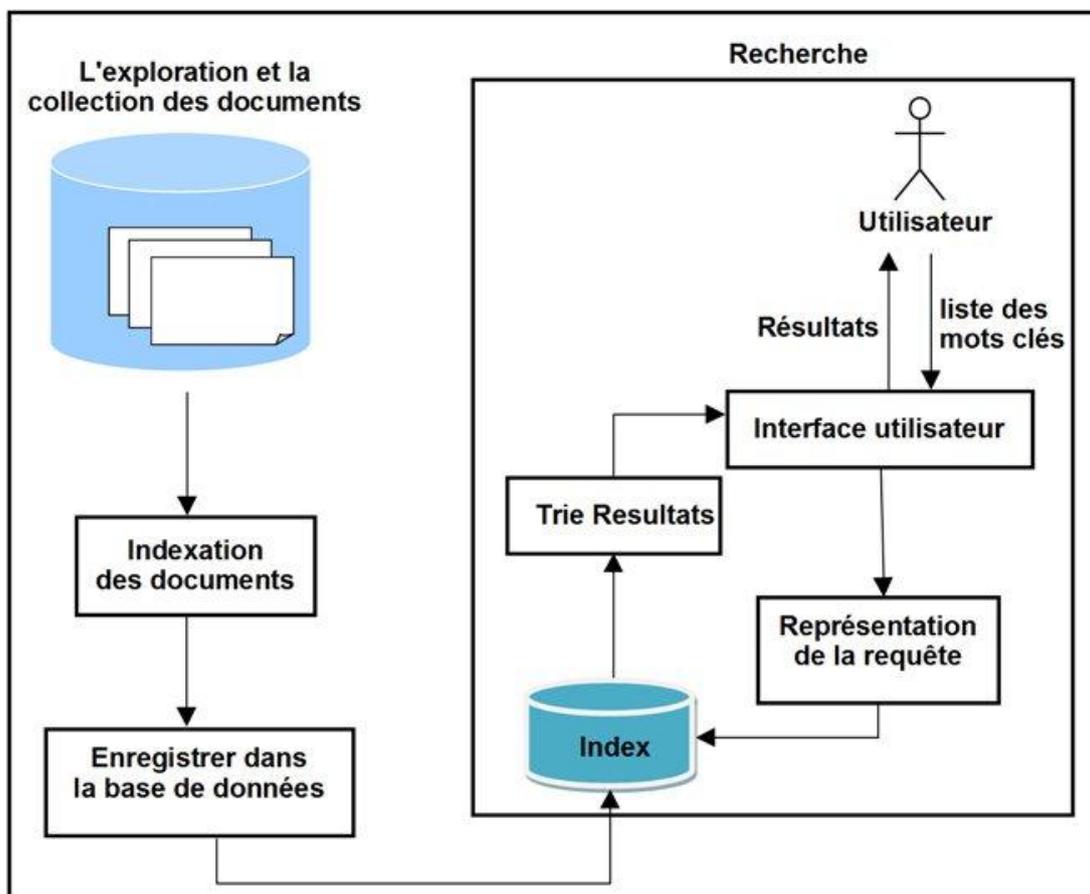


Figure I.1 : le fonctionnement général d'un moteur de recherche.

I.1.2. Domaines d'utilisation des moteurs de recherche

Les moteurs de recherche sont utilisés dans toutes les applications où il est nécessaire d'accéder au contenu de l'information. Parmi ces applications, on peut citer les applications de gestion de bibliothèque et de centres de documentation et la recherche sur l'internet.

Chapitre I : Moteur de recherche et l'indexation des documents

I.1.3. La Recherche par le contenu

En générale les documents se composent majoritairement de texte, le traitement de l'information touche principalement le texte pour rechercher des documents textuels parmi un ensemble des documents une solution consiste à lire séquentiellement tous les documents et à sélectionner ceux qui correspondent aux critères de recherche mais plus la taille des documents est grande et le temps de réponse sera large.

Pour que la recherche soit rapide, il est indispensable que les documents, dans lequel la recherche va s'effectuer, soit structurés intelligemment en fonction des algorithmes de recherche. Il faut donc établir des index, c.à.d. des tables ordonnées contenant les caractéristiques du contenu des documents. C'est par la lecture des index qu'on sélectionnera les documents pertinents.

La qualité de l'information obtenue après une recherche des documents est étroitement liée à la qualité de la formulation de la requête de recherche, ainsi la recherche peut faire l'objet de quelques risques dont :

- **Le Bruit** : le premier risque d'une recherche non efficace est d'engendrer trop de documents, qui sont sélectionnés mais qui ne sont pas intéressants.
- **Le Silence** : le second risque est le manque de résultat qui se traduit par un *silence total*. C'est-à-dire on n'obtient pas des résultats.

Evidemment, l'idéal est de minimiser le taux de *bruit* et de *silence*.

I.2. L'indexation du document

I.2.1. L'indexation

L'indexation est un processus qui permet de représenter un document sous un autre aspect pour le rendre manipulable et exploitable à une recherche ultérieure et cela à partir d'une analyse lexicale, syntaxique ou sémantique, en sélectionnant les mots ou les concepts représentant le contenu sémantique du document.

L'indexation est l'opération centrale de tout système documentaire. Elle consiste à analyser les documents afin de produire un ensemble de mots clés, représentant ce document et les organise dans un index.

Chapitre I : Moteur de recherche et l'indexation des documents

I.2.2. Les différentes formes d'un index

Les index peuvent prendre différentes formes allant de mots simples à des structures sémantiques plus complexes impliquant plusieurs concepts et relations.

- **Les mots :** sont toute chaîne de caractères compris entre deux séparateurs (espace, virgule, etc.), au niveau de l'indexation on peut extraire les mots tels qu'ils sont présentés dans le document comme on peut effectuer certaines transformations sur ce mot en vue d'une normalisation.
- **Les concepts :** sont des termes ou mots-clés, il s'agit d'expressions (pouvant contenir un ou plusieurs mots). Ces concepts sont le plus souvent entrés manuellement (cas de l'indexation manuelle, ou semi automatique) et peuvent être écrits de manière libre par un utilisateur ou ce qui est souvent le cas doivent être choisis parmi une liste de concepts (on parle alors de vocabulaire contrôlé).

Le but étant de les mots et les concepts choisir de manière à ce que l'index perde le moins d'informations sémantiques possible.

I.2.3. Les techniques de l'indexation

I.2.3.1. Indexation manuelle

Lors de l'indexation manuelle un expert dans le domaine choisit les termes qu'il juge pertinents dans la description du contenu sémantique du document. Ce type d'indexation permet d'avoir un vocabulaire d'index contrôlé ce qui permet d'accroître la qualité de la représentation obtenue. Toutefois cette approche est inapplicable pour une collection volumineuse des documents.

I.2.3.2. Indexation automatique

Dans cette méthode on utilise des méthodes logicielles pour établir les index, cette méthode est utilisée par les collections volumineuses des documents.

I.2.3.3. Indexation semi-automatique

C'est une combinaison des deux méthodes précédentes : un premier processus automatique permet d'extraire les termes du document. Cependant le choix final des descripteurs est laissé au spécialiste du domaine.

I.2.4. Processus d'indexation automatique

Le processus d'indexation effectue le transfert de l'information contenue dans le texte d'un document vers un autre espace de représentation traitable par un système informatique. A partir d'une collection des documents, le processus d'indexation nous renvoie une liste d'index structurée. On utilise ce résultat le plus souvent pour effectuer des recherches d'informations. Mais, il peut également servir à comparer et classifier des documents, proposer des mots-clés ou faire une synthèse automatique des documents.

Le processus d'indexation se compose de plusieurs étapes que nous avons schématisées dans la Figure suivante :

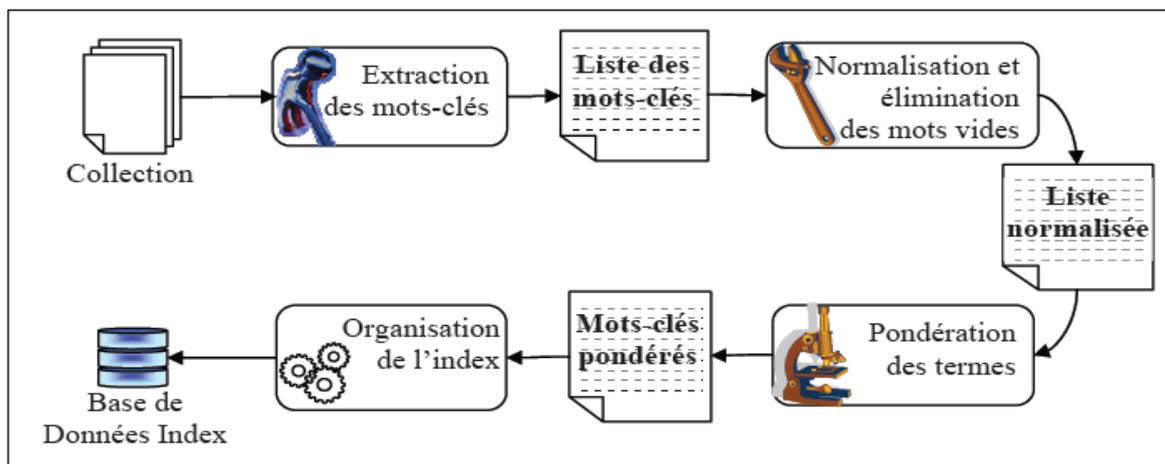


Figure I.2: Etapes du processus d'indexation.

I.2.4.1. L'extraction des mots-clés du document

Appelée Tokenization en anglais, l'extraction des mots-clés est une étape qui constituera la base de tout le reste du processus d'indexation. Il faut donc que cette phase soit d'une qualité maximale. Certains systèmes de recherche d'information utilisent une liste de mots-clés prédéfinie. Cette liste est conçue manuellement et dans la plus part des cas construite par rapport à un thème spécifique.

Chapitre I : Moteur de recherche et l'indexation des documents

I.2.4.2. La normalisation des mots-clés du document

Ce traitement consiste à retrouver pour un mot sa forme normalisée (généralement le masculin pour les noms, l'infinitif pour les verbes, le masculin singulier pour les adjectifs, etc...) Ainsi, dans l'index ne sont conservées que les formes normalisées, ce qui offre un gain de place appréciable, mais surtout si le même traitement est effectué sur la requête, cela permet d'être beaucoup plus souple et rapide dans la recherche.

I.2.4.3. L'élimination des mots vides

Cette étape constitue un facteur d'une grande influence dans la précision de la recherche. Le fait de ne pas éliminer les mots vides provoque inévitablement du bruit. L'élimination des mots vides qui sont des mots du langage courant et qui ne contiennent pas beaucoup d'information sémantique doit se faire aussi bien à l'indexation qu'à l'interrogation (requête de recherche).

I.2.4.4. La pondération des mots-clés

Cette étape est entièrement dépendante du modèle de recherche d'information utilisé. Elle permet de définir l'importance qu'a un terme dans un document donné, elle est également utilisée pour filtrer l'index résultant du processus d'indexation. Il existe plusieurs techniques de pondération des termes dont les deux les plus importantes sont les suivantes :

- **La fréquence d'occurrences :** On admet généralement qu'un mot qui apparaît souvent dans un texte représente un concept important. La représentation fréquentielle permet de prendre en compte le nombre d'apparitions d'un mot dans un document.
- **La valeur de discrimination :** Par «discrimination», on réfère au fait qu'un terme distingue bien un document des autres documents. C'est-à-dire, un terme qui a une valeur de discrimination élevée doit apparaître seulement pour un petit nombre des documents. Un terme qui apparaît dans tous les documents n'est pas discriminant. Le pouvoir de discrimination d'un terme est important dans le choix des index qu'on veut garder.

Conclusion

Au cours de ce chapitre on définit les fonctionnalités d'un moteur de recherche et les concepts de base de l'indexation des documents, nous avons présenté le processus

Chapitre I : Moteur de recherche et l'indexation des documents

d'indexation des documents, déterminer les facteurs qui peuvent améliorer l'indexation et ainsi améliorer la recherche des documents.

Dans le prochain chapitre nous allons présenter UML et l'architecture client/serveur.

Chapitre II :

Présentation de UML et l'architecture client/serveur

Introduction.

II.1. Le langage de modélisation UML.

II.2. Réseaux informatique.

Conclusion

Introduction

Dans ce chapitre nous allons présenter les concepts de base du langage de modélisation UML, présenter quelque diagramme utilisé dans notre mémoire et présenter l'architecture client/serveur.

II.1. Le langage de modélisation UML

Le langage de modélisation UML Unified Modeling Language est un langage de modélisation orienté objet, normalisé par l'OMG (Object Management Group) et adopté par les méthodes d'analyse et conception orienté objet existantes, Il permet de décrire et de représenter un système (réel ou logiciel) grâce à un ensemble des diagrammes (diagramme de classe, diagramme de cas d'utilisation, etc) et de concepts objets. Il couvre toutes les étapes de développement d'un projet et facilite la compréhension de représentations abstraites complexes et la communication au sein de l'équipe de développement.

II.1.1. Les diagrammes UML

UML dans sa version 2 comporte treize types de diagramme pour représenter des concepts particuliers du système d'information. UML offre beaucoup des diagrammes qui servent à la modélisation des systèmes, mais on utilisant en générale (suffisant) de ces diagrammes. Parmi les diagrammes UML on cite :

Diagrammes structurels ou diagrammes statiques (UML Structure) :

- diagramme de classes.
- diagramme d'objets.
- diagramme de composants.
- diagramme de déploiements.
- diagramme de paquetages (package).

Diagrammes comportementaux ou diagrammes dynamiques.

- diagramme de cas d'utilisation.
- diagramme d'activités.
- diagramme d'états-transition.
- diagramme d'interaction.
- ✓ diagramme de séquence.

- ✓ diagramme de communication.
- ✓ diagramme global d'interaction.
- ✓ diagramme de temps.

II.1.1.1. Diagrammes de cas d'utilisation

Ces diagrammes servent à représenter les fonctionnalités offertes par le système ainsi que les acteurs et les relations existant entre eux.

- **Acteur** : Représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.
- **Cas d'utilisation (use case)** : Représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable pour un acteur particulier.
- **Les relations entre acteurs** : La seule relation entre acteur est la relation de *généralisation*. Quand un acteur fils hérite d'un acteur père, il hérite en réalité de toutes les associations du père.
- **Les relations entre cas d'utilisation**
 - *La relation d'inclusion* : Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B.
 - *La relation d'extension* : Une relation d'extension d'un cas d'utilisation A signifie qu'une instance de A peut être étendue par le comportement décrit dans B.
 - *La relation de généralisation* : Les cas d'utilisation descendants héritent de la description de leurs parents communs. Chacun d'entre eux peut néanmoins comprendre des interactions spécifiques supplémentaires.

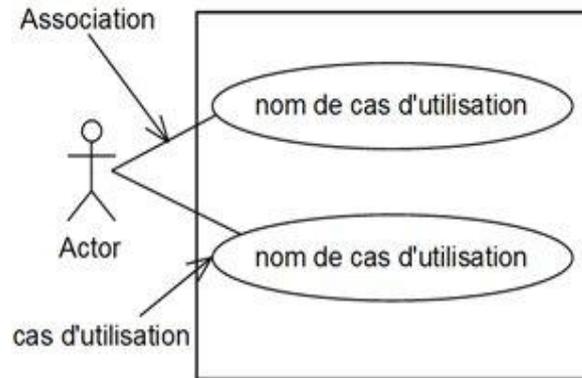


Figure II.3 : La structure du diagramme de cas d'utilisation.

II.1.1.2. Diagramme de classe

Le diagramme de classe saisit la structure statique d'un système en montrant les classes dans le système, les relations entre les classes, les attributs et les opérations qui caractérisent chaque classe.

- **Classe** : Une classe est la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes.
- **Les attributs** : Un attribut de classe définit une propriété commune aux objets d'une classe.
- **Les méthodes** : Une méthode définit une opération appliquée à des objets ou par les objets d'une classe.
- **Association** : Une association décrit un groupe de liens entre les classes d'objets. Elle peut être une association générale, une généralisation (héritage), une agrégation ou une dépendance, etc...



Figure II.4 : la structure du diagramme de classes.

II.1.1.3. Diagramme de séquence

Les diagrammes de séquence permettent de représenter le séquençage des échanges de messages entre des objets du système, en vue d'une réponse à un besoin utilisateur. Les messages correspondent généralement à des appels de méthodes.

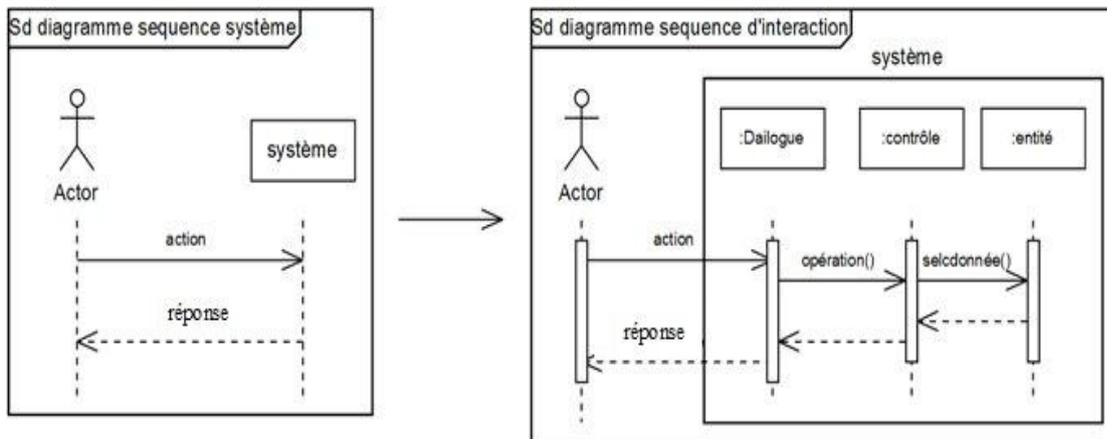


Figure II.5 : la structure du diagramme de séquence système vu comme une boîte noire et remplacé par un ensemble d'objet.

- **Scénario:** il représente une succession particulière d'actions, s'exécutant du début à la fin du cas d'utilisation.
- **Ligne de vie :** il représente l'ensemble des opérations exécutées par un objet.
 - **Message:** Un message est une transmission d'information unidirectionnelle entre deux objets, l'objet émetteur et l'objet récepteur. Dans un diagramme de séquence, deux types de messages peuvent être distingués :
 - **Message synchrone :** Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions.
 - **Message asynchrone :** Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations.

II.1.1.4. Diagramme d'activités

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils permettent de représenter graphiquement le comportement d'une méthode ou déroulement d'un cas d'utilisation.

❖ **Action** : Correspond à un traitement qui modifie l'état de système. Le passage d'une action à une autre est matérialisé par une **transition**. Les transitions sont déclenchées par la fin d'une action et provoquent le début d'une autre.

❖ **Activité** : Représente le comportement d'une partie du système en termes d'actions et de transitions.

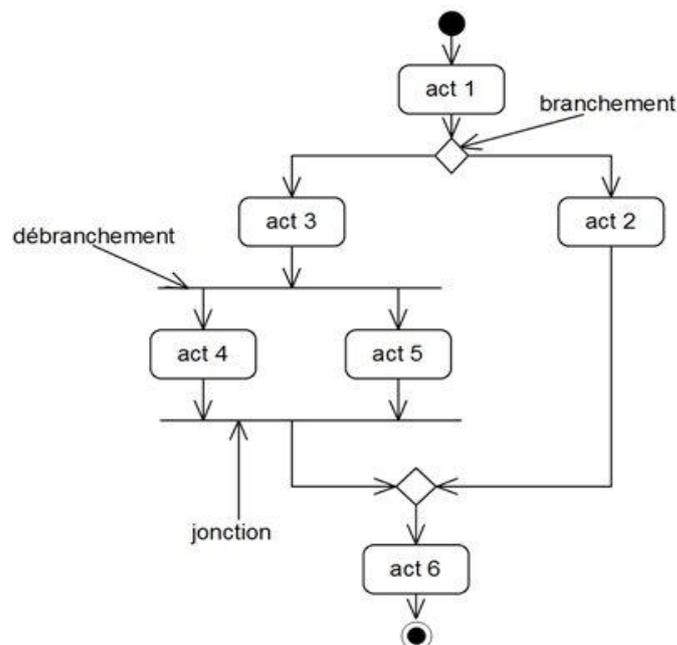


Figure II.6 : la structure du diagramme d'activité.

II.1.1.5. Diagramme de déploiement

Ce diagramme décrit l'architecture technique d'un système avec une vue centrée sur la répartition des composants de système.

II.2. Réseaux informatique

C'est un ensemble d'équipements autonome interconnectés. Deux ordinateurs sont considéré interconnectés s'ils sont capables d'échanger des informations entre eux.

II.2.1. Définition d'un serveur

On appelle logiciel serveur un programme qui offre un service sur le réseau. Le serveur accepte des requêtes, les traite et renvoie le résultat au demandeur. Le terme serveur s'applique aussi à la machine sur lequel s'exécute le logiciel serveur.

Pour pouvoir offrir ces services en permanence, le serveur doit être sur un site avec accès permanent et s'exécute en permanence.

II.2.2. Définition d'un client

On appelle logiciel client un programme qui utilise le service offert par un serveur. Le client envoie une requête et reçoit la réponse. Le client peut être raccordé par une liaison temporaire.

II.2.3. Architecture Client/serveur

De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines client (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en termes de capacités d'entrée/sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers et une connexion,

Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client FTP, client messagerie, lorsque l'on désigne un programme, tournant sur une machine cliente capable de traiter des informations qu'il récupère auprès du serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client messagerie il s'agit de courrier électronique).

Dans un environnement purement Client/serveur, les ordinateurs du réseau (les clients) ne peuvent voir que le serveur, c'est un des principaux atouts de ce modèle.

II.2.3.1. Avantages de l'architecture client/serveur

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

- **Des ressources centralisées** : étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tout les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction.
- **Une meilleure sécurité** : car le nombre de points d'entrée permettant l'accès aux données est moins important.
- **Une administration au niveau serveur** : les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.
- **Un réseau évolutif** : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeures.

II.2.3.2. Inconvénient du modèle client/serveur

L'architecture client/serveur a tout de même quelques inconvénients parmi lesquelles :

- **Un cout élevé** : du à la technicité du serveur.
- **Une maille faible** : le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui. Heureusement, les serveurs en générale une grande tolérance aux pannes.

II.2.4. Fonctionnement d'un système client/serveur

Un système client/serveur fonctionne selon le schéma suivant :

Le client émet une requête vers le serveur grâce à son adresse (IP+ port d'écoute), qui désigne un service particulier du serveur.

Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine client et son port d'écoute.

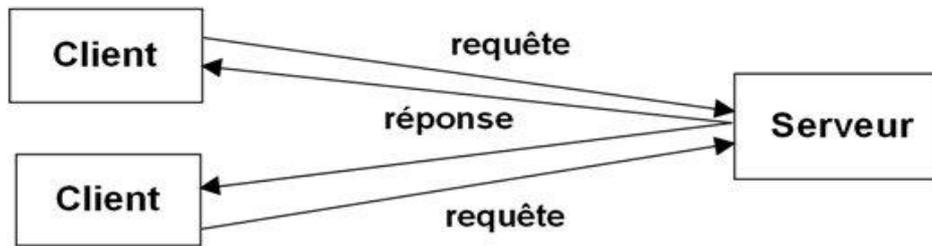


Figure II.7: Fonctionnement du système client/serveur.

II.2.5. Les différents types d'architecture Client/serveur

II.2.5.1. L'architecture à 2 niveaux

L'architecture à deux niveaux (aussi appelée architecture 2-tiers, tiers signifiant étage en anglais) caractérise les systèmes clients/serveurs dans lesquels le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service.

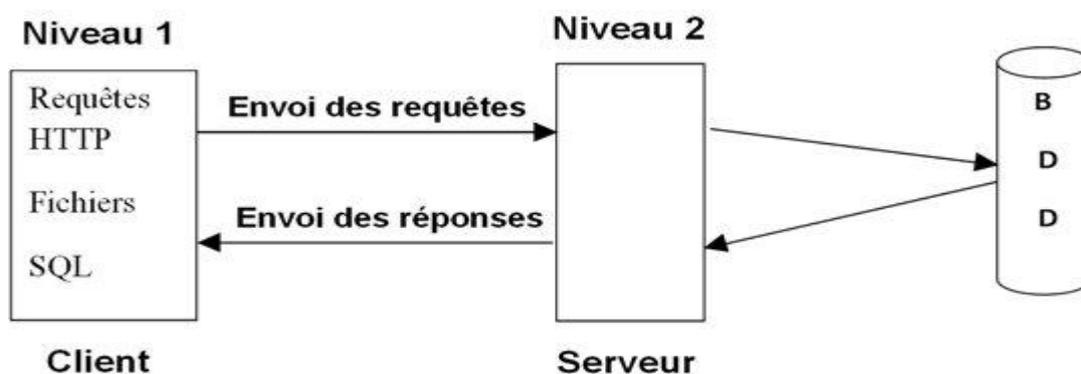


Figure II.8: Architecture Client/serveur à 2 niveaux.

II.2.5.2. L'architecture à 3 niveaux

Dans l'architecture à 3 niveaux (appelées architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

Le client : le demandeur de ressources.

Le serveur d'application (appelé aussi **middleware**) : le serveur chargé de fournir la ressource mais faisant appel à un autre serveur.

Le serveur secondaire (généralement un serveur de base de données), fournissant un service au premier serveur.

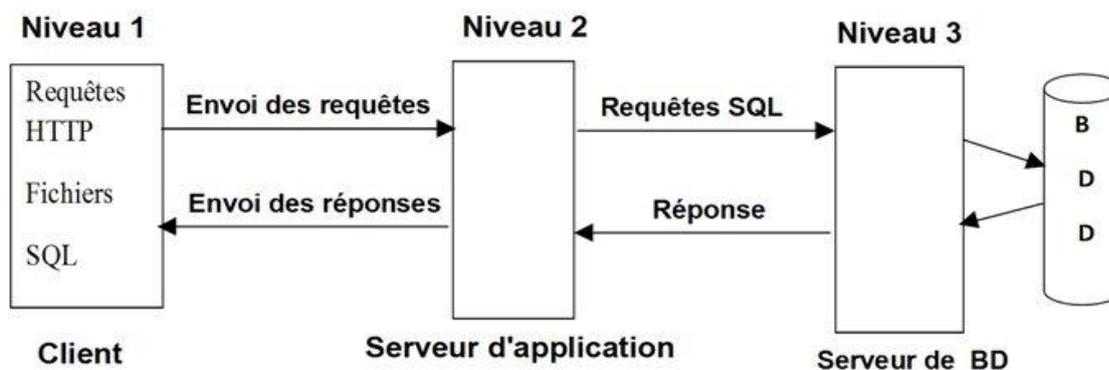


Figure II.9 : Architecture Client/serveur à 3 niveaux.

II.2.5.3.L'architecture d'égal à égal (Peer to Peer)

Dans une architecture d'égal à égal (ou dans sa dénomination anglaise Peer to Peer) contrairement à une architecture de réseau de type client/serveur, il n'y a pas de serveur dédié. Ainsi chaque ordinateur dans un tel réseau est un peu serveur et un peu client. Cela signifie que chacun des ordinateurs du réseau est libre de partager ses ressources. Un ordinateur relié à une imprimante pourra donc éventuellement la partager afin que tous les autres ordinateurs puissent y accéder via le réseau.

➤ Inconvénients des réseaux d'égal à égal

Les réseaux d'égal à égal ont énormément d'inconvénients :

Ce système n'est pas du tout centralisé, ce qui le rend très difficile à administrer la sécurité est très peu présente aucun et maillon du système n'est fiable.

Ainsi, les réseaux d'égal à égal ne sont valables que pour un petit nombre d'ordinateurs et pour des applications ne nécessitant pas une grande sécurité (il est donc déconseillé pour un réseau professionnel avec des données sensibles).

Conclusion

Dans ce chapitre, nous avons passé en revue le langage de modélisation UML et ses différents diagrammes. UML est un langage riche, il couvre toutes les phases d'un cycle de développement. Il est également (indépendant du domaine d'application et des langages d'implémentation). Pour ces raisons nous avons choisi UML pour modéliser notre application. Nous avons aussi présenté l'architecture client/serveur et cite ces avantages notamment la fiabilité et la sécurité.

Dans le prochain chapitre de notre système nous allons identifier les besoins de système.

Chapitre III : Spécification des besoins

Introduction

III.1. Présentation de l'organisme d'accueil.

III.2. Cahier de charge.

III.3. Identifier les acteurs.

III.4. Diagramme de cas d'utilisation.

III.5. La description des cas d'utilisation.

III.6. Les diagrammes de séquence système

Conclusion

Introduction

Dans ce chapitre nous allons spécifier les besoins du système à réaliser. Dans un premier temps, nous allons introduire le cahier des charges qui servira de fil conducteur tout au long de la conception, Dans un second temps, nous commençons à identifier les acteurs qui interagiront avec le système, nous identifions et nous décrivons les cas d'utilisation du système.

III.1. Présentation de l'organisme d'accueil

III.1.1. Création du Centre Universitaire de Mila

Le Centre Universitaire de Mila est un établissement public à caractère scientifique, culturel et professionnel doté de la personnalité morale et l'autonomie financière. Le centre universitaire a été ouvert en juillet 2008 et est situé à 5km du centre ville de Mila sur la route nationale N°79 allant vers les villes de Zeghaia et de Ferdjioua et s'étendant sur une superficie d'environ 87 hectares. Le Centre Universitaire de Mila compte 3373 étudiants et de 155 enseignants.

III.1.2. Faculté de centre Universitaire de Mila

Le Centre Universitaire de Mila propose durant l'année 2009-2010 une formation supérieure dans les domaines suivants :

- Mathématiques et Informatique.
- Science de la Nature et de la Vie.
- Sciences Economiques, Commerciales et Sciences de Gestion.
- Langues et Littérature Arabe.
- Anglais.

III.1.3. Présentation de la bibliothèque

La bibliothèque de centre universitaire de Mila a été ouverte officiellement le 18 octobre 2008, sur une infrastructure d'une superficie de 3394,40 mètres carrés et une capacité de lecteurs estimée à 500 places.

L'activité traditionnelle de la bibliothèque est l'acquisition d'ouvrages imprimés, la conservation et prêt des documents. Toutefois, la documentation électronique prend un poids croissant dans l'activité de La bibliothèque.

Les services de la bibliothèque sont :

- ❖ Service de prêt externe.
- ❖ Service de prêt interne.
- ❖ La salle d'internet.
- ❖ La salle de lecture des étudiants.

III.2. Cahier de charge

- ✓ **Présentation du projet :** Chaque année des centaines de mémoires de fin d'étude Dans leur version papier et électronique seront archivés dans la bibliothèque du centre universitaire de Mila .L'administrateur de la bibliothèque veut faciliter l'archivage et la consultation des mémoires par un système informatique.
- ✓ **Les besoins fonctionnels :** Le premier tour d'horizon des besoins exprimés par l'administrateur a permis d'établir le cahier des charges préliminaire suivant :
 - Gestion des utilisateurs (ajouter utilisateur, supprimer utilisateur, modifier utilisateur).
 - archiver des mémoires.
 - Supprimer des mémoires.
 - Rechercher et consulter des mémoires.
- ✓ **Les besoins techniques :** Convivialité et simplicité d'utilisation de l'application :
L'application devra pouvoir être utilisée simplement. Elle doit avoir donc une interface graphique claire confortable.

Le temps de réponse : Le temps de réponse doit être acceptable.

La sécurité : la gestion des utilisateurs (ajouter, supprimer et modifier un utilisateur), L'archivage et la suppression des mémoires sont limités à l'administrateur, le système doit permettre d'authentifier, Tous les autres utilisateurs du système peuvent rechercher et consulter les mémoires.

III.3. Identifier les acteurs

- **L'administrateur** : la ou les personnes responsables de l'archivage et la suppression des mémoires, ils peuvent aussi rechercher et consulter les mémoires.
- **L'utilisateur** : la personne qui peut rechercher et consulter les mémoires.

III.4. Diagramme de cas d'utilisation

Après l'identification des cas d'utilisation et les acteurs, on peut les représenter graphiquement sur un diagramme de cas d'utilisation comme suit :

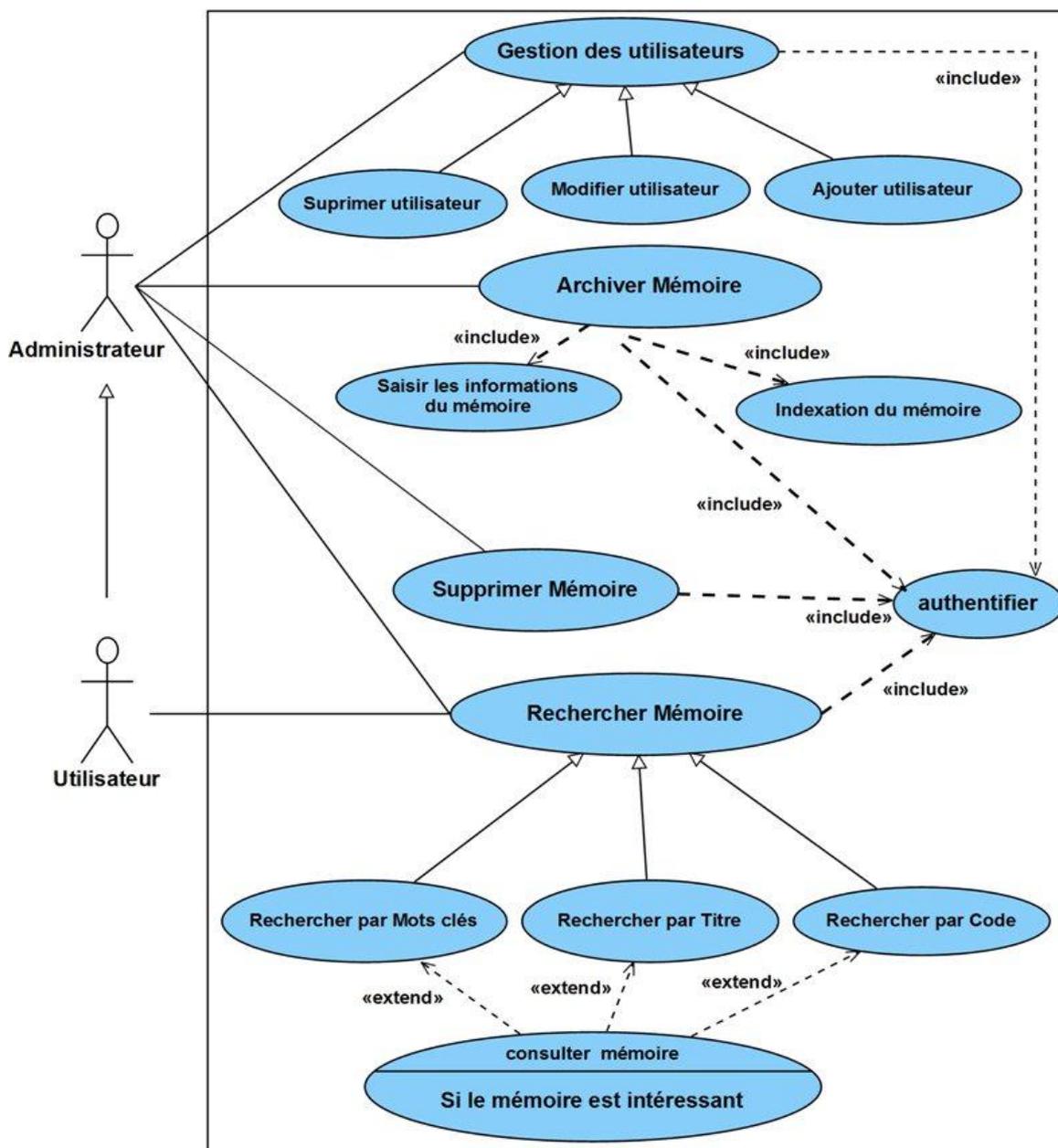


Figure III.10 : le diagramme de cas d'utilisation.

III.5. La description des cas d'utilisation**➤ Authentifier**

| Cas | Authentifier |
|-------------------------|---|
| Acteur | L'administrateur, l'utilisateur. |
| Le But | Ce cas permet de vérifier l'autorisation d'accéder au système. |
| Pré conditions | L'utilisateur(ou l'administrateur) possède un compte. |
| Post conditions | L'utilisateur est authentifié par le système. |
| Scenario nominal | <ol style="list-style-type: none"> 1. le système demande à l'utilisateur d'entrer le nom d'utilisateur et le mot de passe. 2. l'utilisateur saisie le nom d'utilisateur et le mot de passe et valide(OK). 3. le système vérifie la validité du nom d'utilisateur et mot de passe. 4. Si nom d'utilisateur et mot de passe sont correctes, le système ouvrir la fenêtre principal. |
| Scenario Erreur | <ol style="list-style-type: none"> 1. Si le nom d'utilisateur et/ou mot de passe incorrecte (mot de passe incorrecte). |

➤ Gestion des utilisateurs**❖ Ajouter un utilisateur**

| Cas | Ajouter un utilisateur |
|-------------------------|---|
| Acteur | L'administrateur |
| Le But | Ce cas permet d'ajouter un utilisateur. |
| Pré conditions | L'administrateur est authentifié. |
| Post conditions | L'ajout d'un nouvel utilisateur. |
| Scenario nominal | <ol style="list-style-type: none"> 1. l'administrateur demande la fenêtre "ajouter un utilisateur". 2. le système demande les informations de l'utilisateur. 3. l'administrateur saisir les informations. 4. le système vérifie l'existence de nom d'utilisateur. 5. Si le nom d'utilisateur n'existe pas, le système affiche un |

| | |
|------------------------|--|
| | message succès. |
| Scenario Erreur | 1. Si le nom d'utilisateur existe (l'utilisateur déjà ajouté). |

❖ **Modifier un utilisateur**

| | |
|-------------------------|---|
| Cas | Modifier un utilisateur |
| Acteur | l-administrateur |
| Le But | Ce cas permet de modifier le nom d'utilisateur ou le mot de passe, ou les deux (nom d'utilisateur, mot de passe) d'un utilisateur. |
| Pré conditions | L'administrateur est authentifié. |
| Post conditions | La modification d'un utilisateur avec succès. |
| Scenario nominal | <ol style="list-style-type: none"> 1. l-administrateur demande la fenêtre "modifier un utilisateur". 2. le système demande le nom d'utilisateur. 3. l'administrateur saisir le nom d'utilisateur. 4. le système vérifie l'existence de nom d'utilisateur. 5. Si le nom d'utilisateur existe, le système affiche les autres informations de l'utilisateur. 6. le système demande le nouveau nom d'utilisateur et mot de passe. 7. l'administrateur saisie le nouveau nom d'utilisateur et le mot de passe et valide 7. le système affiche la modification un message succès. |
| Scenario Erreur | 1. Si le nom d'utilisateur n'existe pas (l'utilisateur n'existe pas). |

❖ Supprimer un utilisateur

| | |
|-------------------------|--|
| Cas | Supprimer un utilisateur |
| Acteur | l-administrateur |
| Le But | Ce cas permet de supprimer un utilisateur. |
| Pré conditions | l-administrateur est authentifié. |
| Post conditions | La suppression d un utilisateur avec succès. |
| Scenario nominal | <ol style="list-style-type: none"> 1. l-administrateur demande la fenêtre "supprimer un utilisateur". 2. le système demande le nom d'utilisateur. 3. l-administrateur saisir le nom d'utilisateur. 4. le système vérifie l'existence de nom d'utilisateur. 5. Si le nom d'utilisateur existe, le système affiche les informations de l'utilisateur. 6. l-administrateur valide la suppression de l'utilisateur 7. le système affiche un message succès. |
| Scenario Erreur | 1. Si le nom d'utilisateur n'existe pas (l'utilisateur n'existe pas). |

➤ Archiver un mémoire

| | |
|-------------------------|--|
| Cas | Archiver Mémoire |
| Acteur | l-administrateur |
| Le But | Ce cas permet d'archiver et stocker un mémoire. |
| Pré conditions | l-administrateur est authentifié. |
| Post conditions | Le mémoire est archivé avec succès. |
| Scenario nominal | <ol style="list-style-type: none"> 1. l-administrateur demande la fenêtre "archiver mémoire". 2. le système demande les informations du mémoire. 3. l-administrateur saisie les informations. 3. le système vérifie l'existence du mémoire. 4. si le code du mémoire n'existe pas, le système ajoute le mémoire (indexation et sauvegarde). 5. le système affiche un message succès. |

| | |
|------------------------|---|
| Scenario Erreur | 1. Si le code du mémoire existe(le mémoire déjà archivé). |
|------------------------|---|

➤ **Supprimer un mémoire**

| Cas | Supprimer Mémoire |
|-------------------------|--|
| Acteur | l-administrateur |
| Le But | Ce cas permet de supprimer un mémoire. |
| Pré conditions | l-administrateur est authentifié. |
| Post conditions | Le mémoire est supprimé avec succès. |
| Scenario Nominal | <ol style="list-style-type: none"> 1. l-administrateur demande la fenêtre "supprimer mémoire". 2. le système demande le code du mémoire. 3. L-administrateur saisi le code du mémoire. 4. le système vérifie l'existence du mémoire. 5. si le code mémoire existe, le système affiche les informations du mémoire. 6. L-administrateur valide la suppression du mémoire. 7. le système affiche un message succès. |
| Scenario Erreur | 1. Si le code du mémoire n'existe pas(le mémoire n'existe pas). |

➤ **Rechercher et consulter un mémoire**

❖ **Rechercher et consulter un mémoire par titre**

| Cas | Rechercher et consulter mémoire par titre |
|-------------------------|---|
| Acteur | L'utilisateur, l-administrateur. |
| Le But | Ce cas permet de rechercher et consulter un mémoire par titre. |
| Pré Conditions | L'utilisateur ou l-administrateur est authentifié. |
| Post conditions | Le mémoire recherché et consulté est trouvé. |
| Scenario Nominal | <ol style="list-style-type: none"> 1. l'utilisateur demande la fenêtre " recherche mémoire". 2. l'utilisateur choisit le mode de recherche par titre. |

| | |
|------------------------|--|
| | <p>3. L'utilisateur saisi le titre.</p> <p>4. Si le titre existe, le système affiche les informations des mémoires qui on ce titre.</p> <p>5. Si un mémoire est intéressant, l'utilisateur sélectionne et consulte ce mémoire.</p> |
| Scenario Erreur | 1. Si le titre n'existe pas (mémoire n'existe pas). |

❖ **Rechercher et consulter un mémoire par mots clés**

| Cas | Rechercher et consulter mémoire par mots clés |
|-------------------------|---|
| Acteur | L'utilisateur, l'administrateur. |
| Le But | Ce cas permet de rechercher et consulter un mémoire par mots clés. |
| Pré Conditions | L'utilisateur ou l'administrateur est authentifié. |
| Post conditions | Le mémoire recherche et consulte est trouvé |
| Scenario Nominal | <p>1. l'utilisateur demande la fenêtre " recherche mémoire".</p> <p>2. l'utilisateur choisit le mode de recherche par mots clés.</p> <p>3. L'utilisateur saisi les mots clés.</p> <p>4. Si les mots existent, le système affiche les informations des mémoires qui on ces mots clés.</p> <p>5. Si un mémoire est intéressant, l'utilisateur sélectionne et consulte le mémoire.</p> |
| Scenario Erreur | 1. Si les mots clés n'existent pas (mémoire n'existe pas). |

❖ Rechercher et consulter un mémoire par code

| Cas | Rechercher et consulter mémoire par code |
|-------------------------|--|
| Acteur | L'utilisateur, l'administrateur |
| Le But | Ce cas permet de rechercher et consulter un mémoire par code. |
| Pré Conditions | L'utilisateur ou l'administrateur est authentifié. |
| Post conditions | Le mémoire recherche et consulte est trouvé |
| Scenario Nominal | <ol style="list-style-type: none">1. L'utilisateur demande la fenêtre " recherche mémoire".2. L'utilisateur choisit le mode de recherche par code.3. L'utilisateur saisi le code.4. Si le code existe, le système affiche les informations des mémoires qui on ce code.5. Si le mémoire est intéressant, l'utilisateur sélectionne et consulte le mémoire. |
| Scenario Erreur | <ol style="list-style-type: none">1. Si le code n'existe pas (mémoire n'existe pas). |

III.6. Les diagrammes de séquence système

Nous allons présenter les diagrammes de séquences montrant les scénarios de chaque cas d'utilisation précédent.

➤ **Authentifier**

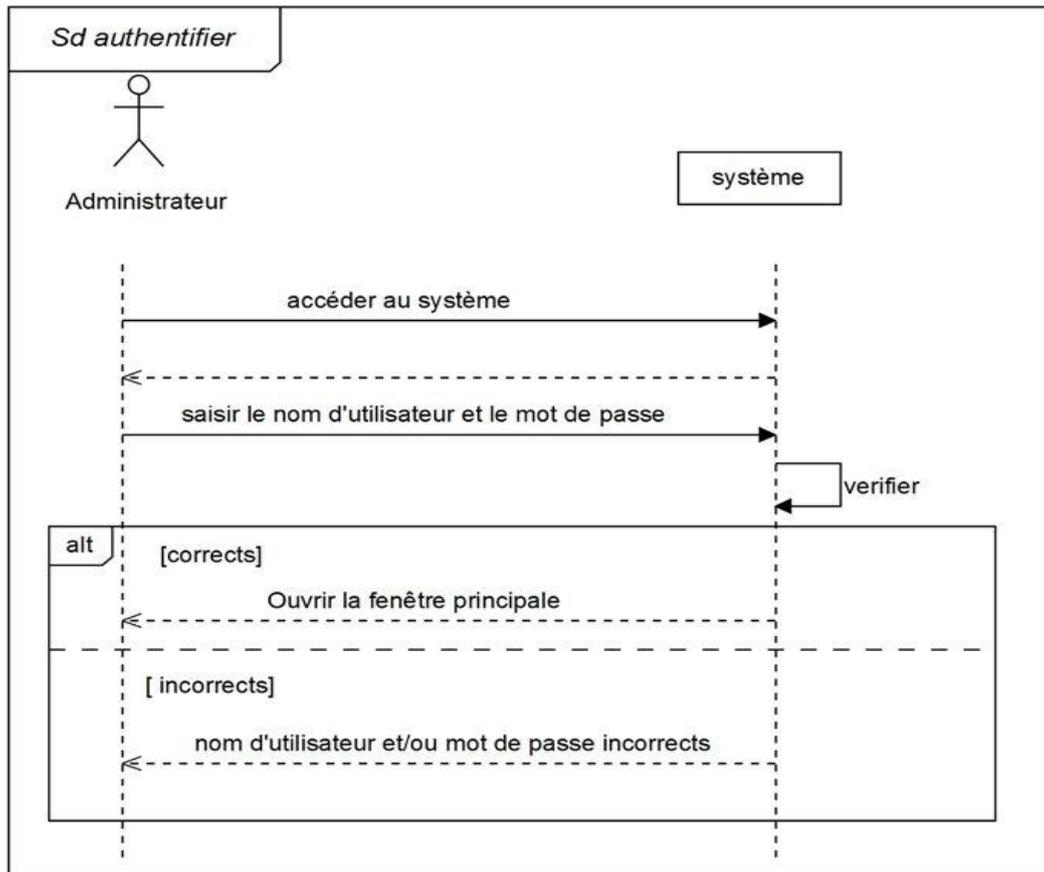


Figure III.11 : Diagramme de séquence système du cas d'utilisation authentifier.

➤ Gestion des utilisateurs

❖ Ajouter un utilisateur

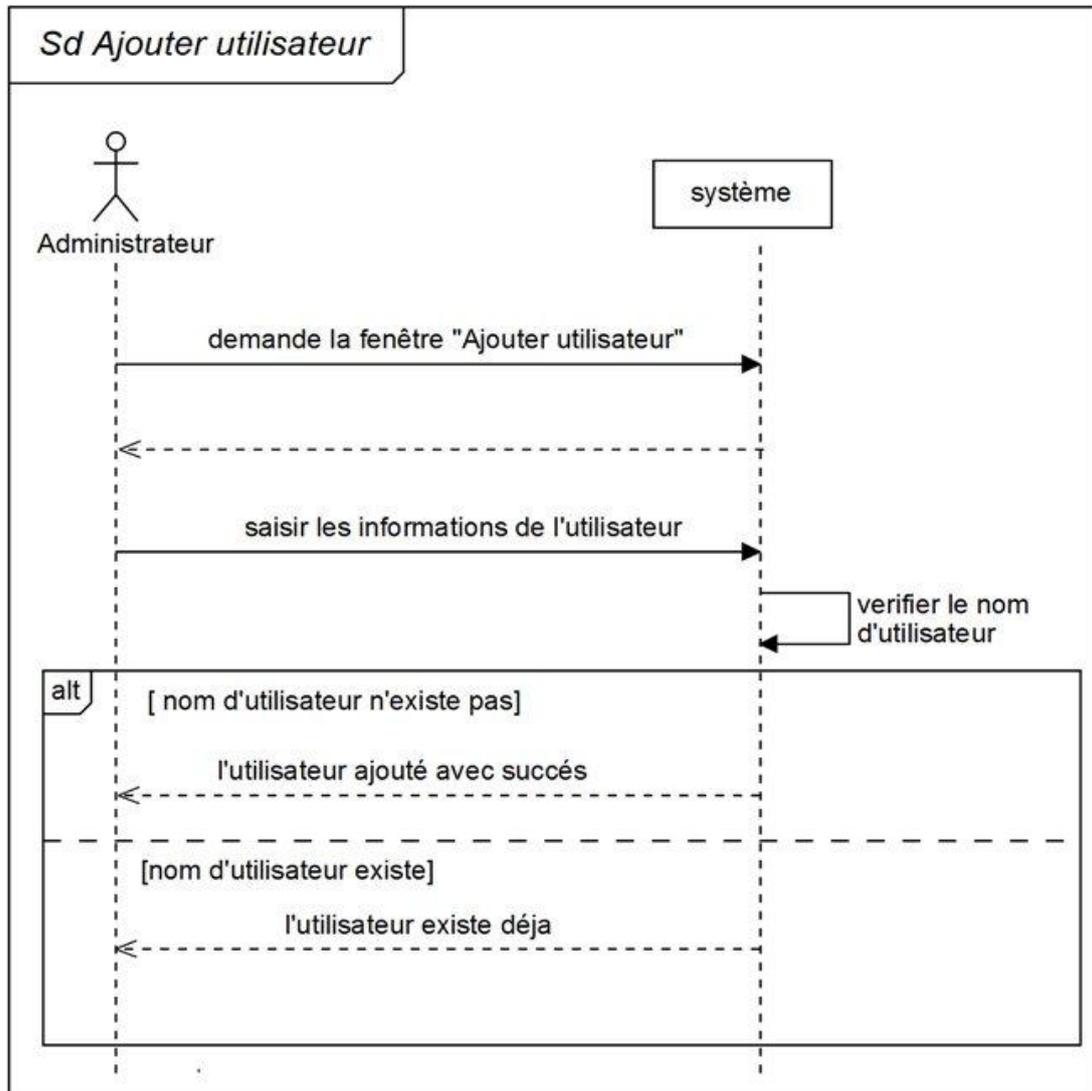


Figure III.12: Diagramme de séquence système du cas d'utilisation ajouter un utilisateur.

❖ **Modifier (nom d'utilisateur, mot de passe)**

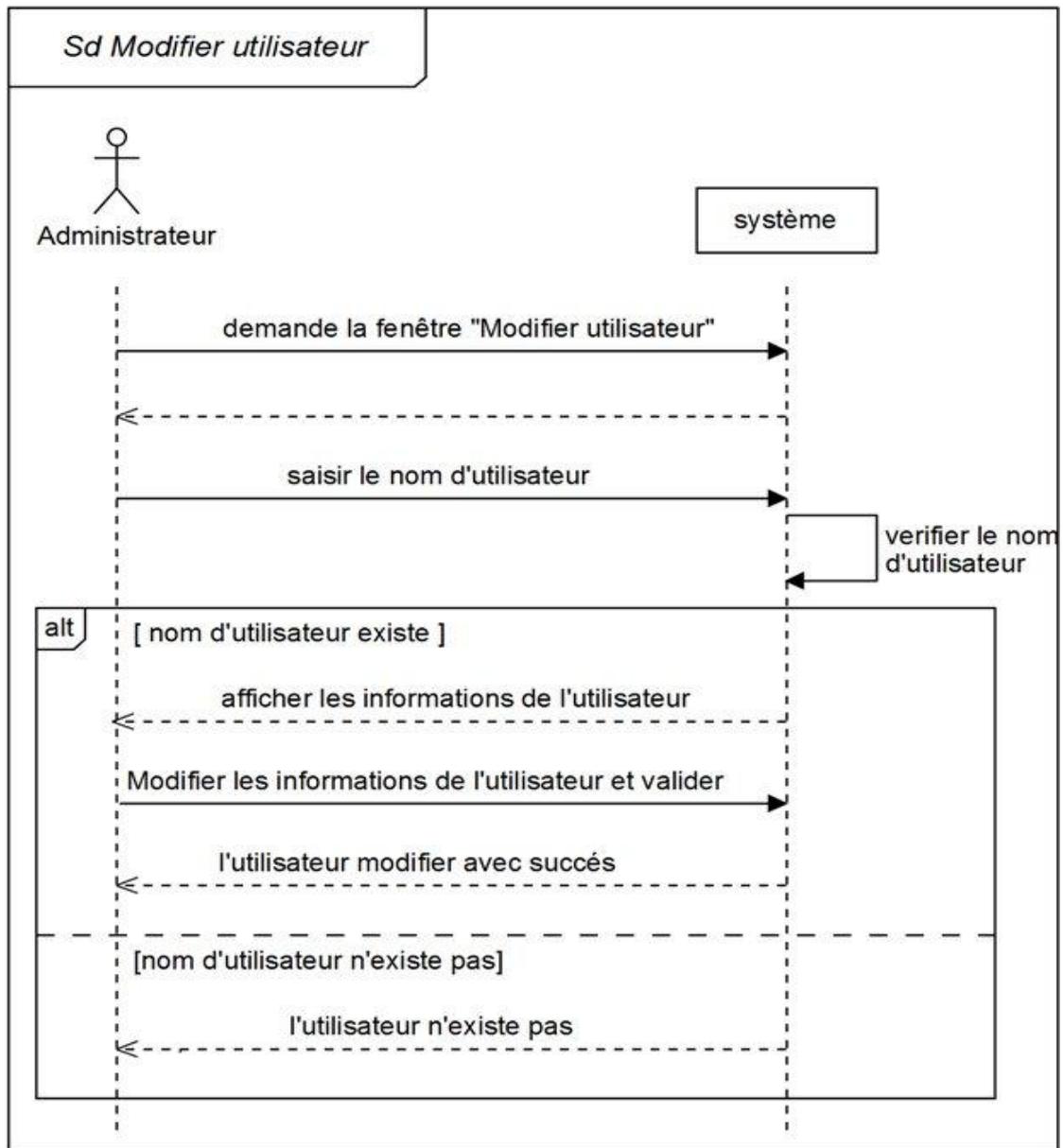


Figure III.13: Diagramme de séquence système du cas d'utilisation modifier un utilisateur.

❖ Supprimer un utilisateur

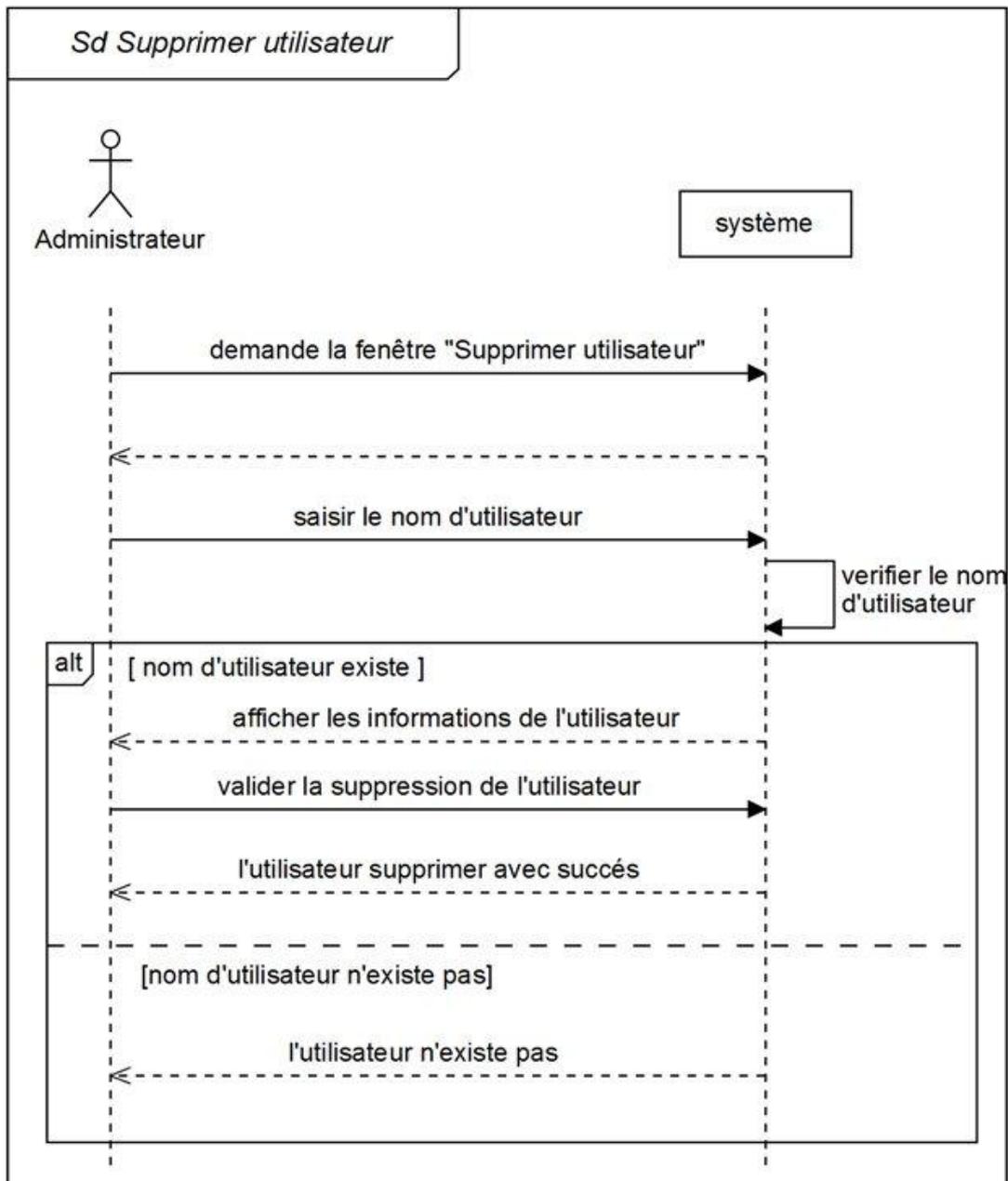


Figure III.14: Diagramme de séquence système du cas d'utilisation supprimer un utilisateur.

➤ Archiver un mémoire

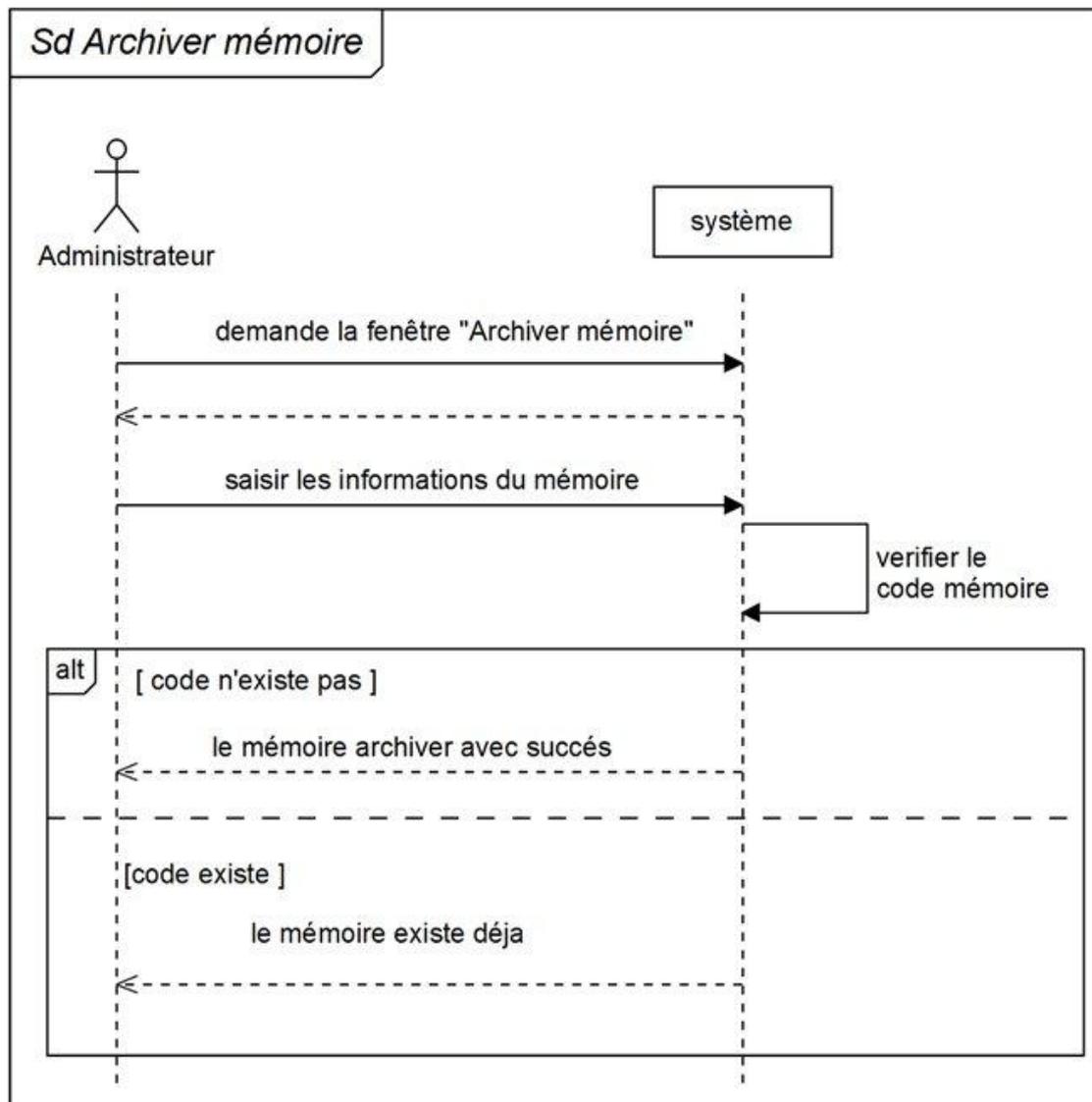


Figure III.15: Diagramme de séquence système du cas d'utilisation archiver un mémoire.

➤ Supprimer un mémoire

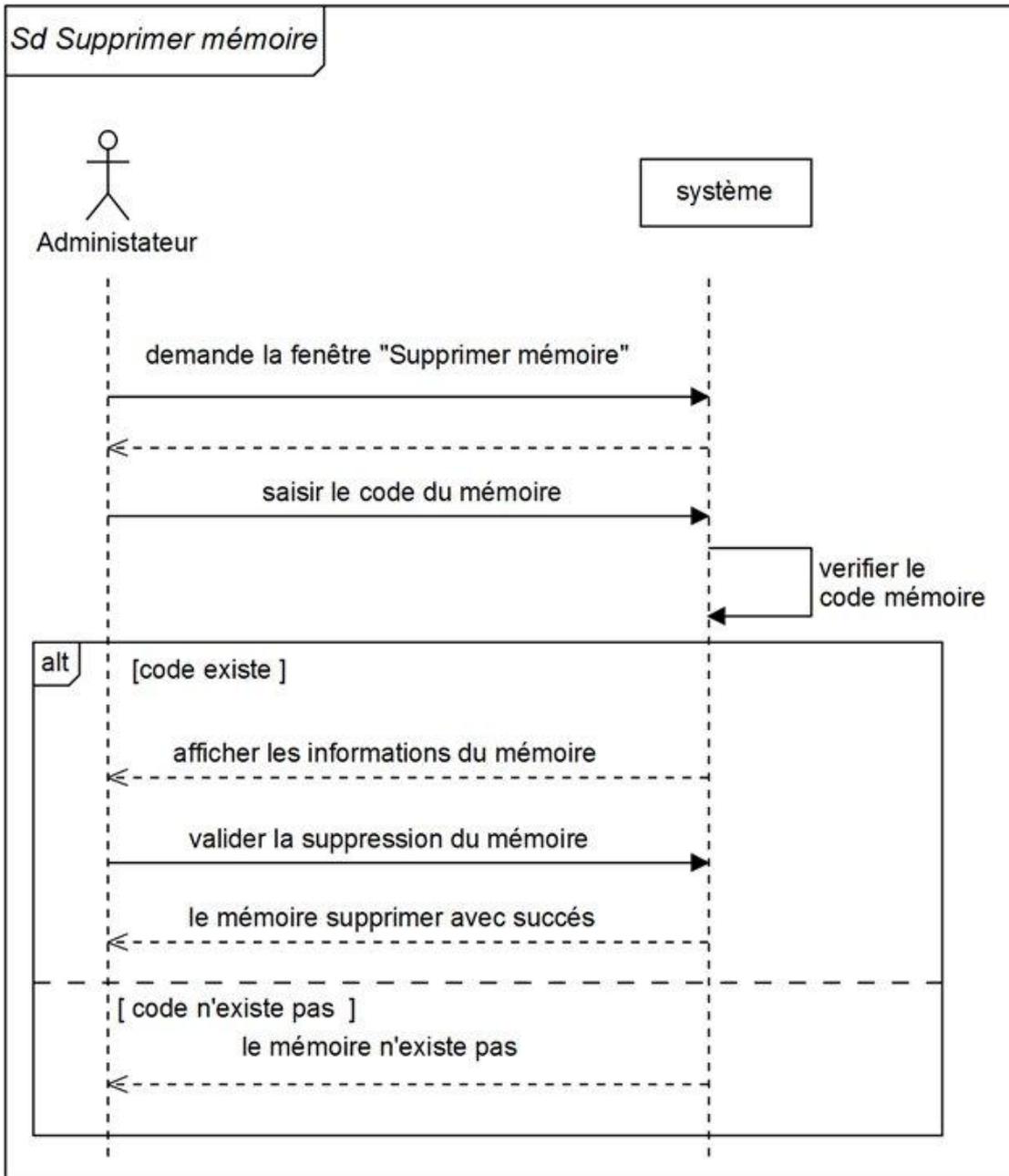


Figure III.16 : Diagramme de séquence système du cas d'utilisation supprimer un mémoire.

➤ Rechercher et consulter un mémoire

❖ Rechercher et consulter un mémoire par Titre

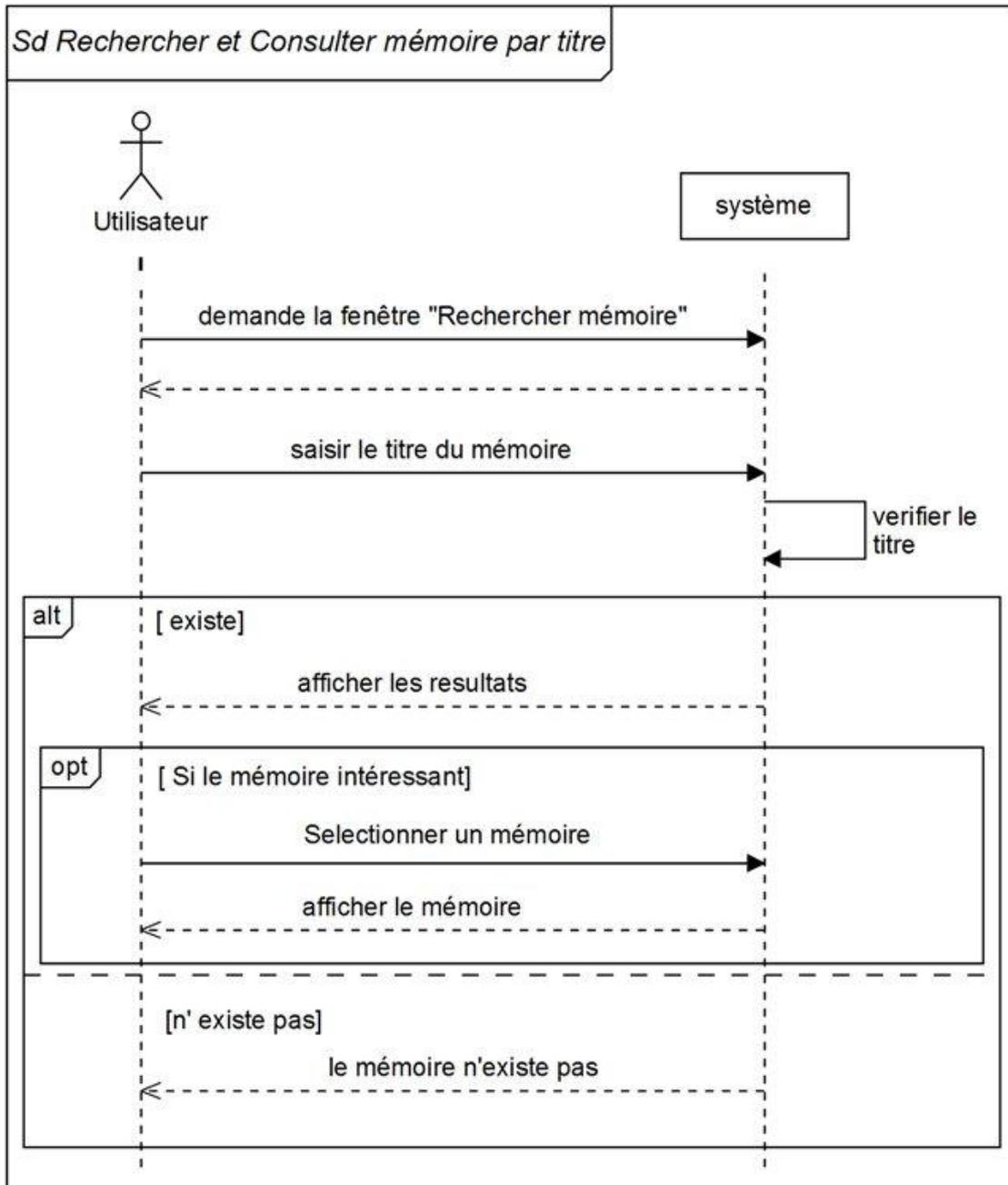


Figure III.17 : Diagramme de séquence système du cas d'utilisation rechercher et consulter par titre.

❖ Rechercher et consulter un mémoire par mots clés

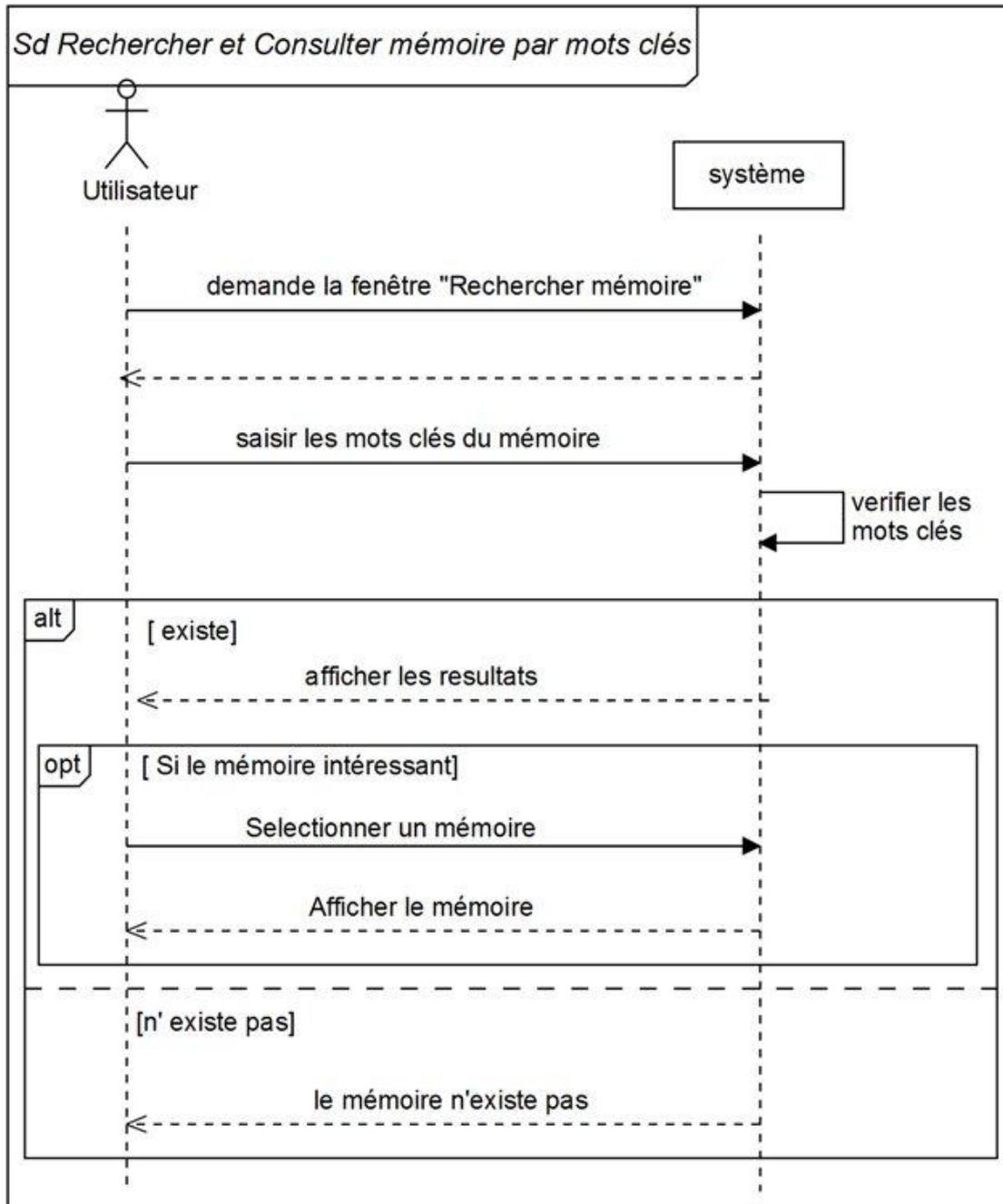


Figure III.18: Diagramme de séquence système du cas d'utilisation rechercher et consulter par mots clés.

❖ Rechercher et consulter un mémoire par code

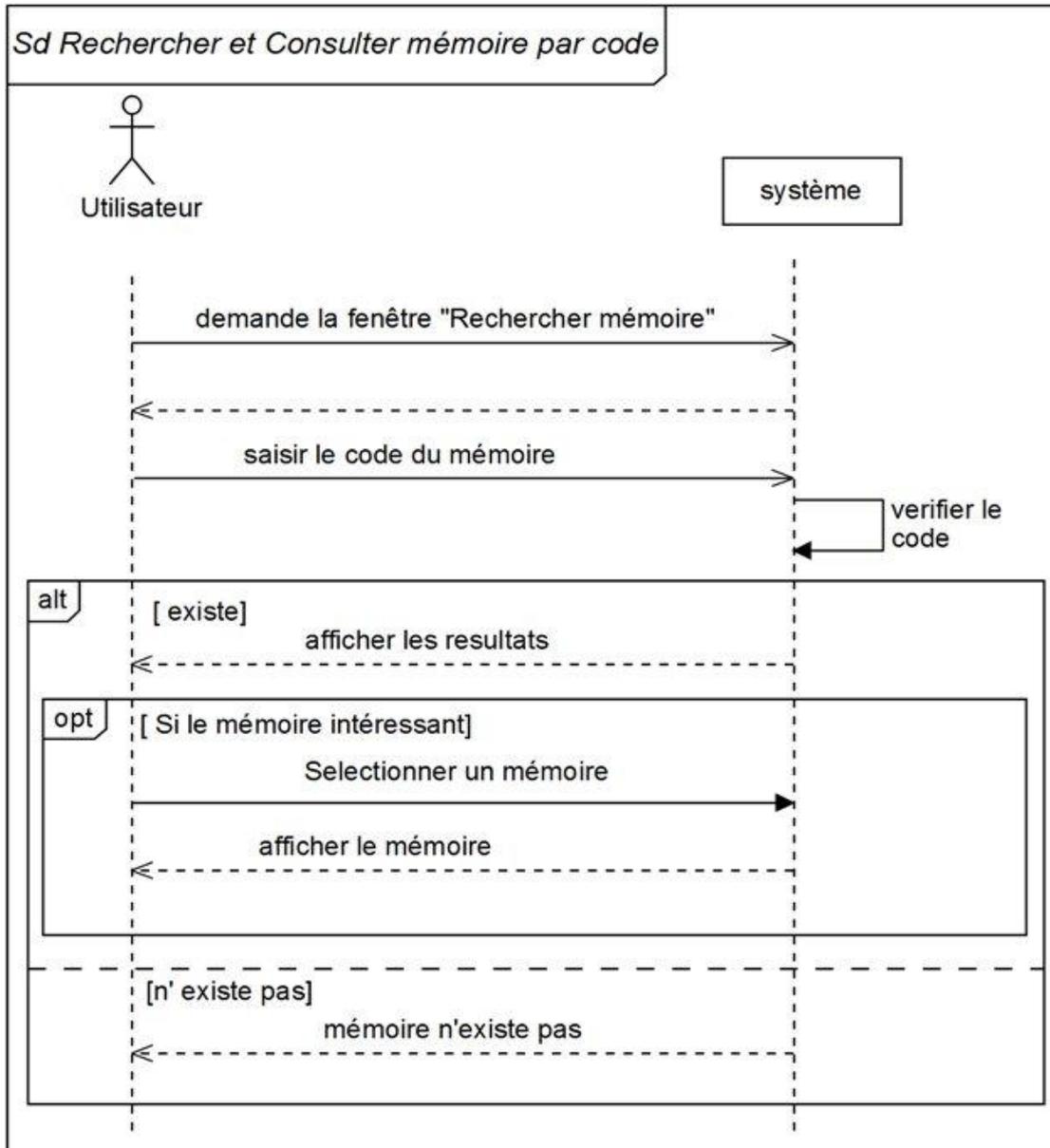


Figure III.19: Diagramme de séquence système du cas d'utilisation rechercher et consulter par code.

Conclusion

Les acteurs et les cas d'utilisation sont les concepts UML fondamentaux pour la spécification des besoins. Nous avons étudié dans ce chapitre les cas d'utilisation possible de notre système ainsi que leur structuration et classement par les diagrammes des cas

d'utilisation. Nous avons représenté aussi un diagramme de séquences système pour chaque cas d'utilisation.

Dans le prochain chapitre de notre modélisation nous allons présenter la décomposition de notre système et la structure interne de ces différents modules.

Chapitre IV : *Phase d'analyse*

Introduction

IV.1. L'analyse du domaine.

IV.2. Identification des concepts du domaine.

IV.3. Ajouter des associations entre classes.

IV.4. Les diagrammes d'activités.

Conclusion

Introduction

Dans cette phase nous allons élaborer le diagramme de classe du domaine et les diagrammes d'activités de notre système.

IV.1. L'analyse du domaine

La conception objet demande principalement une description structurelle, statique, du système à réaliser, sous forme d'un ensemble de classes logicielles. Les classes candidats sont celle issues d'une analyse de domaine, effectivement ces concepts (souvent appelés objets métier) peuvent être identifié directement à partir de la connaissance du domaine.

Nous allons dans cette partie :

- Identifier les concepts du domaine.
- Ajouter les associations entre classes.
- Ajouter les attributs.

IV.2. Identification les concepts du domaine

Nous allons prendre les cas d'utilisations un par un et nous poser pour chacun la question suivantes : quel sont les concepts métier qui participent à ce cas d'utilisation ?

a. Authentifier :

- Administrateur.
- Utilisateur.

b. Gestion des utilisateurs (ajouter, supprimer et modifier utilisateur) :

- Administrateur.

c. Archiver un mémoire :

- Administrateur.
- Info_mémoire.
- Index_mémoire.
- Fichier_mémoire.

d. Supprimer un mémoire :

- Administrateur.
- Info_mémoire.
- Index_mémoire.
- Fichier_mémoire.

e. Rechercher et consulter un mémoire :

- Administrateur.
- Utilisateur.
- Info_mémoire.
- Index_mémoire.
- Fichier_mémoire.

IV.3. Ajouter des associations entre classes

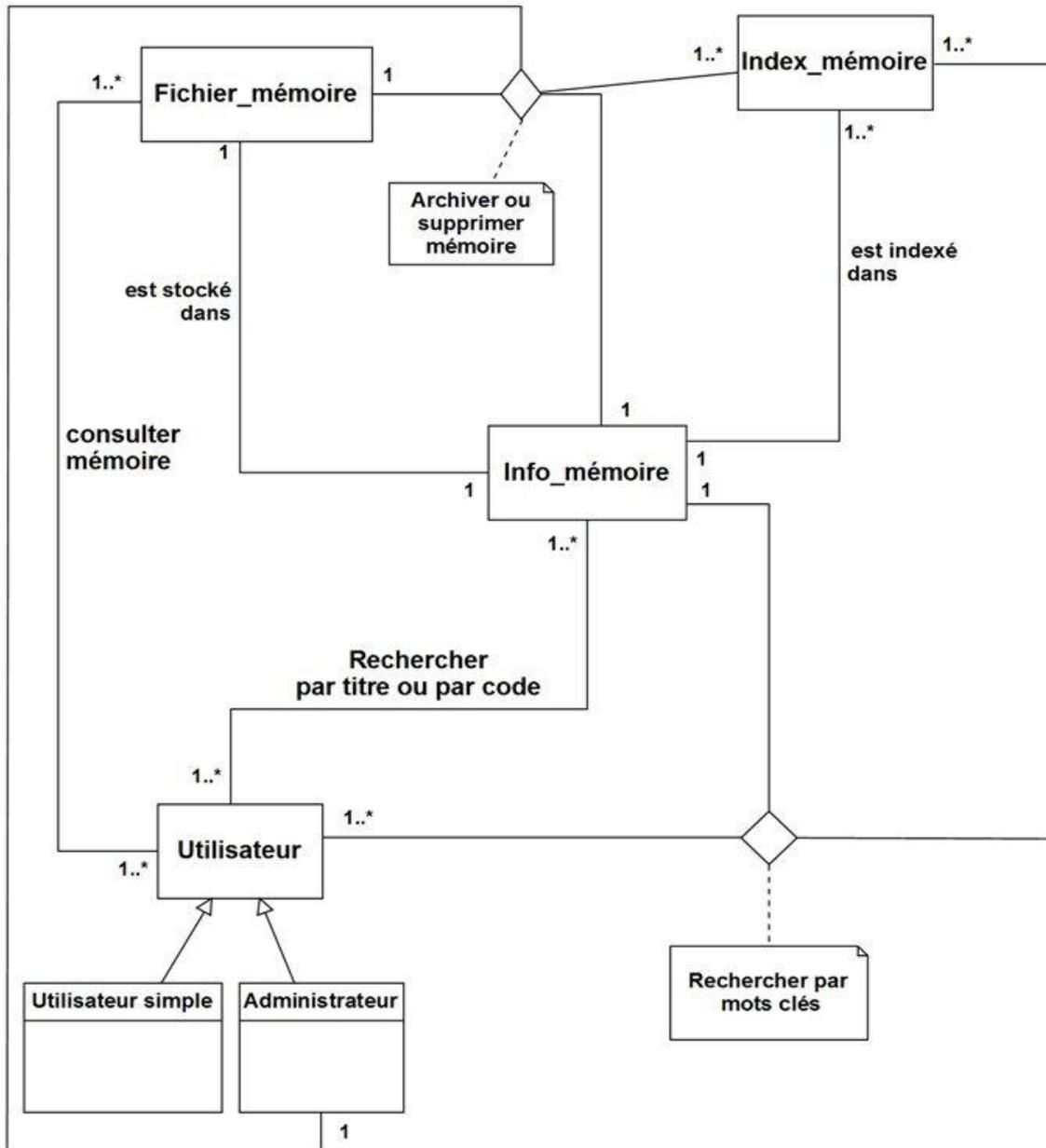


Figure IV.20: Diagramme d'ajouter les associations entre classes.

IV.4. Les diagrammes d'activités

Nous dans cette partie nous allons présenter les diagrammes d'activités équivalent aux les diagrammes de séquences de chaque cas d'utilisation décrit précédent.

❖ **authentifier**

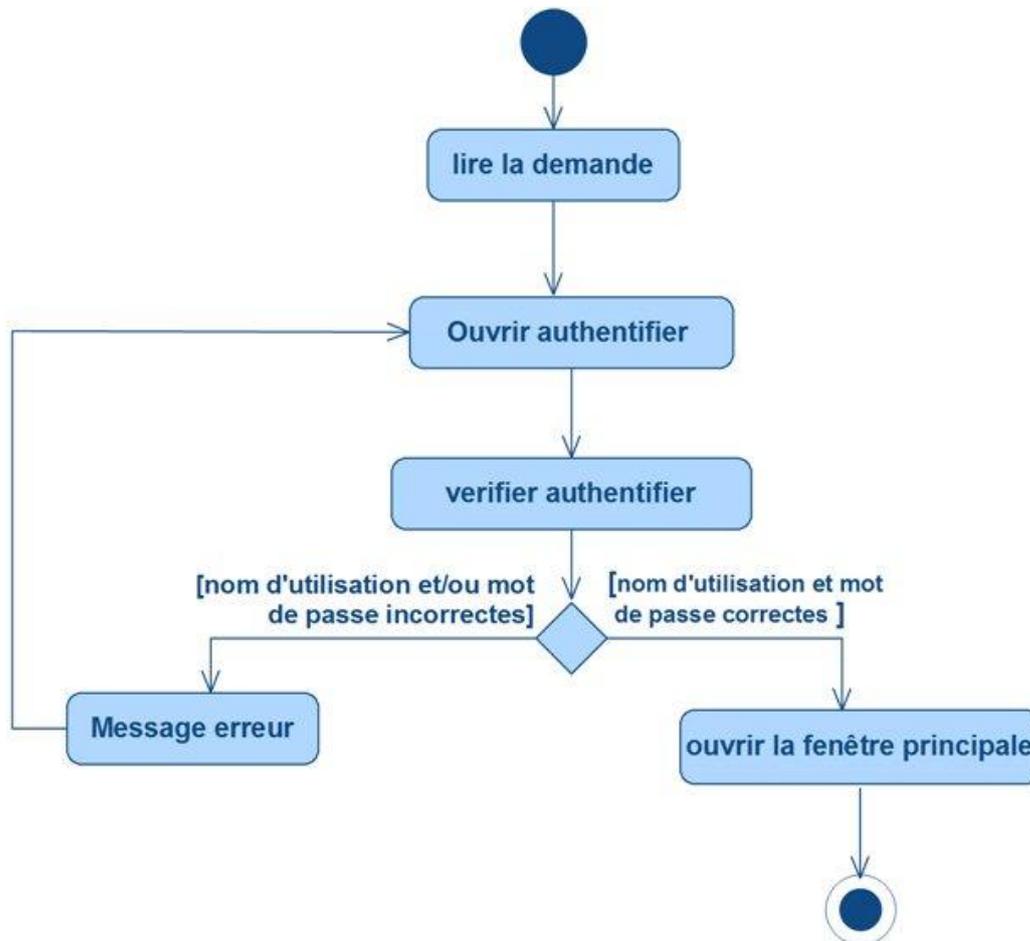


Figure IV.21: Diagramme d'activité du cas d'utilisation authentifier.

- Gestion des utilisateurs
 - ❖ Ajouter un utilisateur

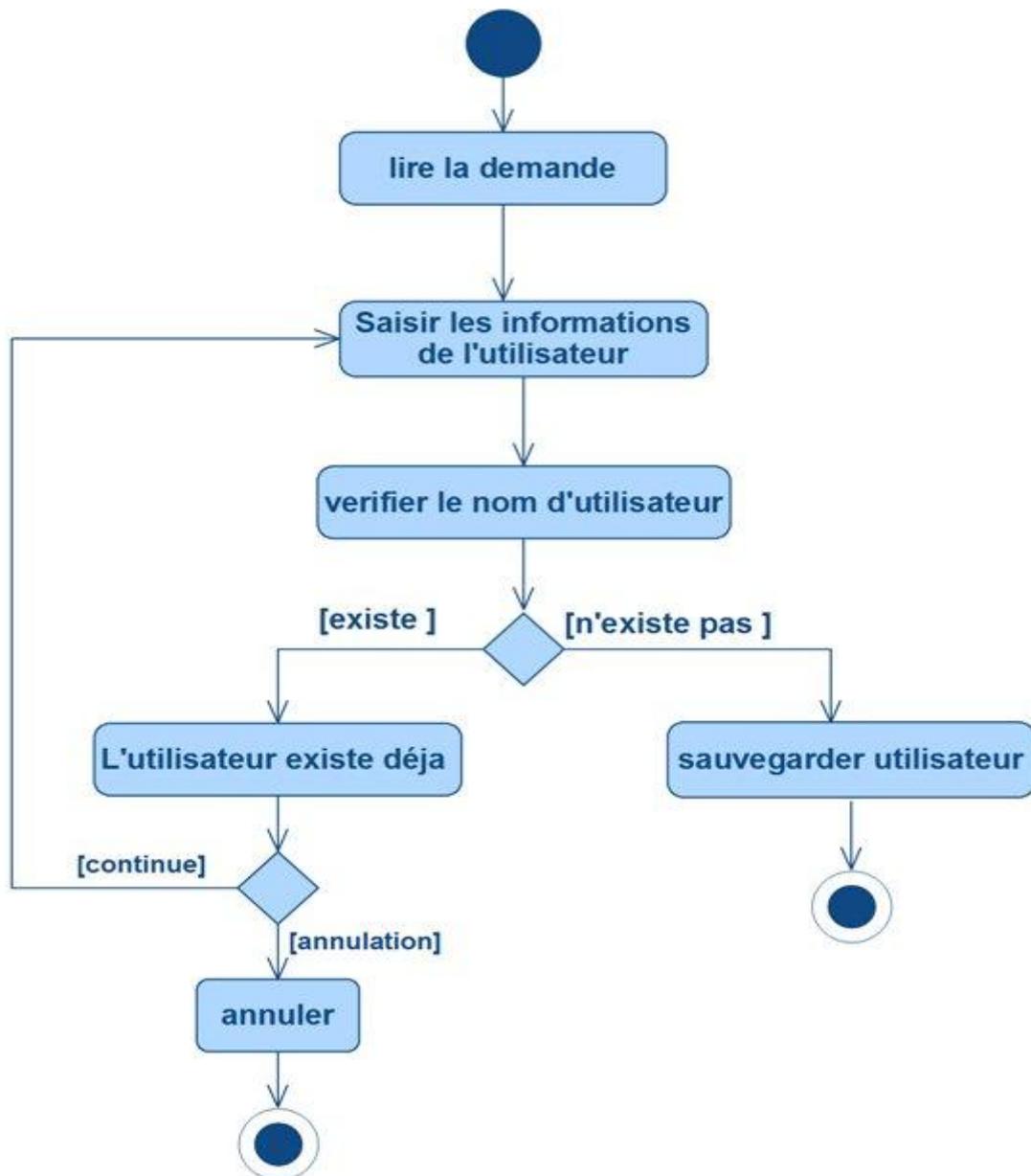


Figure IV.22:Diagramme d'activité du cas d'utilisation ajouter un utilisateur.

❖ Supprimer un utilisateur

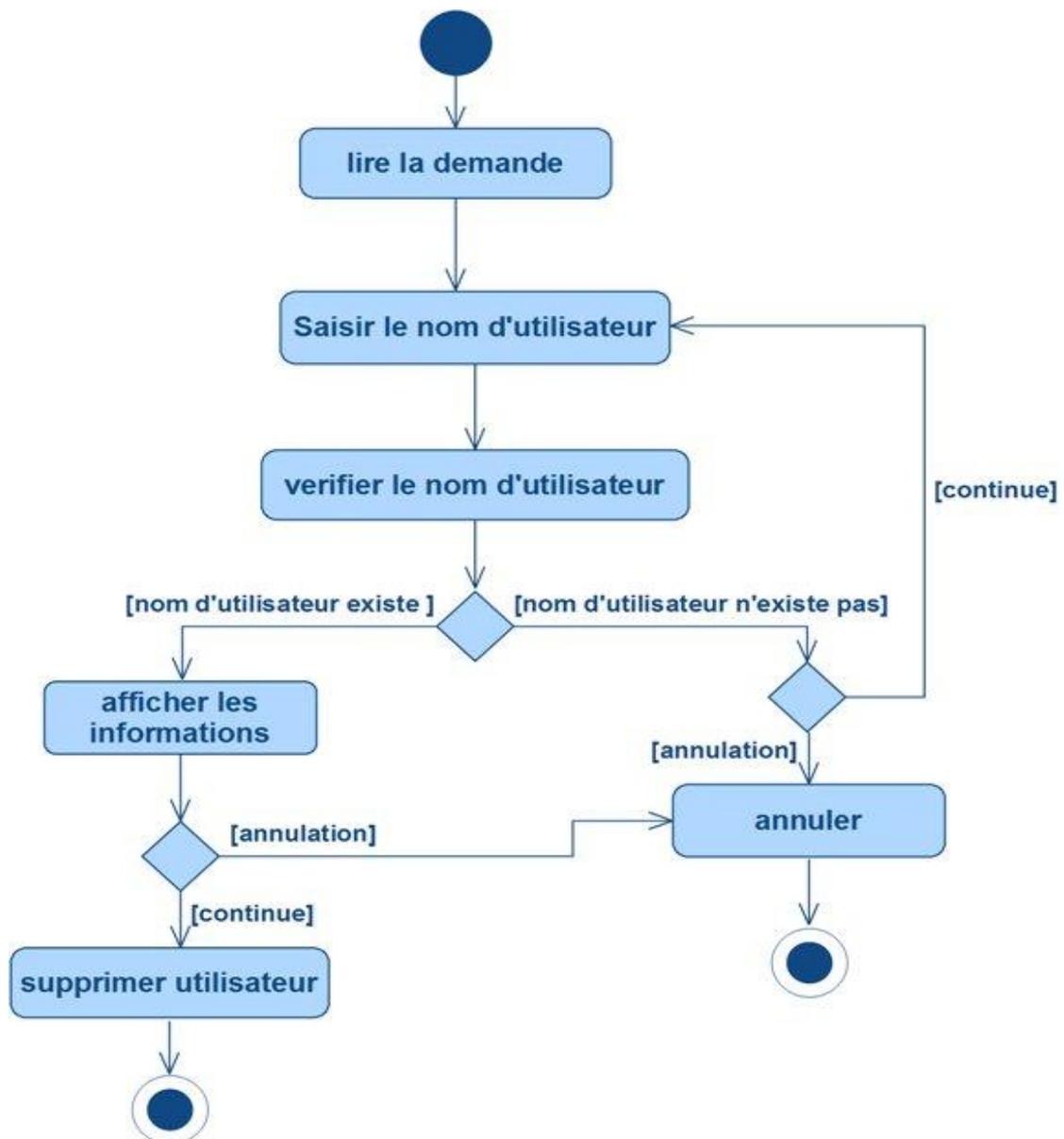


Figure IV.23: Diagramme d'activité du cas d'utilisation supprimer un utilisateur.

❖ **Modifier un utilisateur**

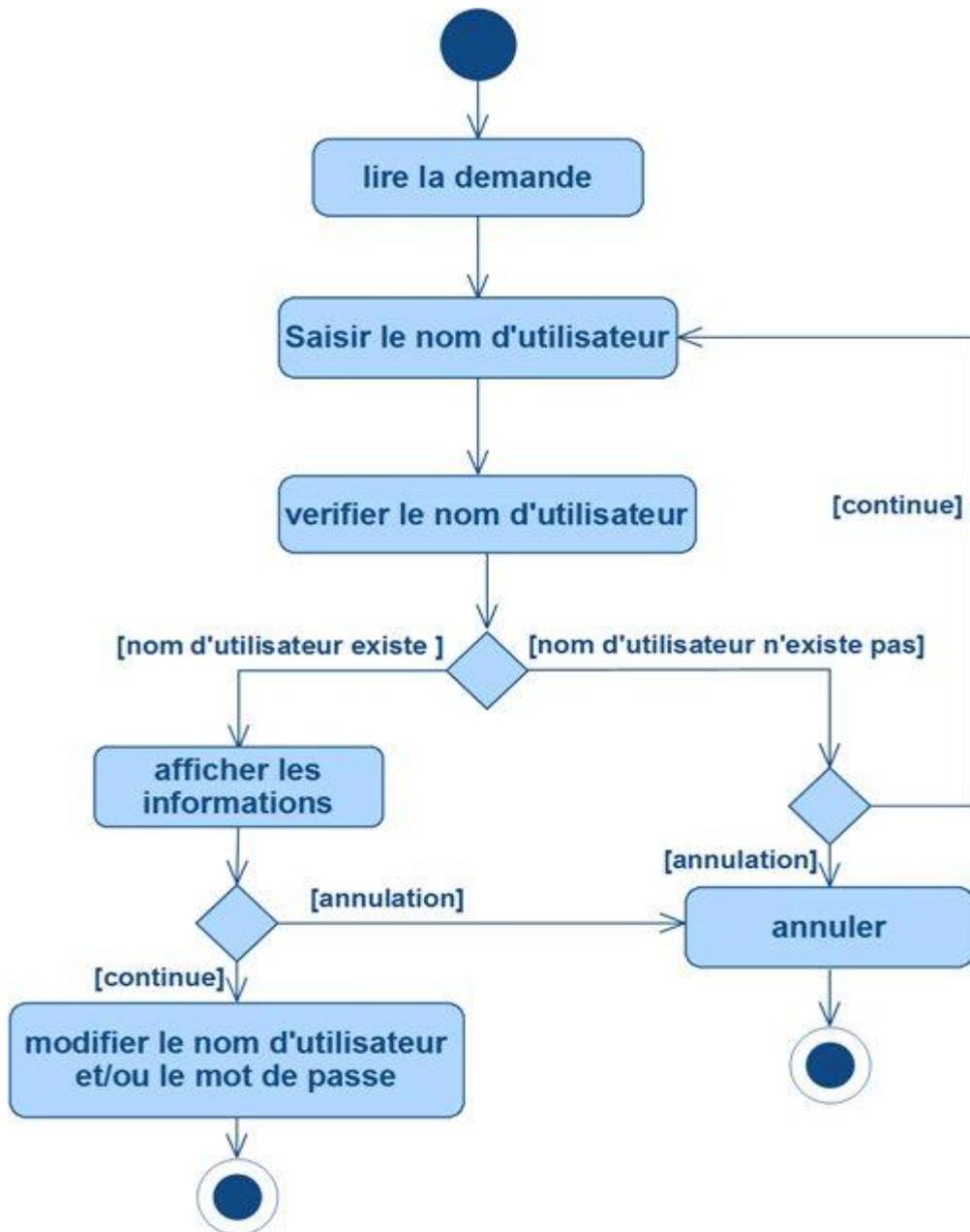


Figure IV.24: Diagramme d'activité du cas d'utilisation modifier un utilisateur.

❖ Archiver un mémoire

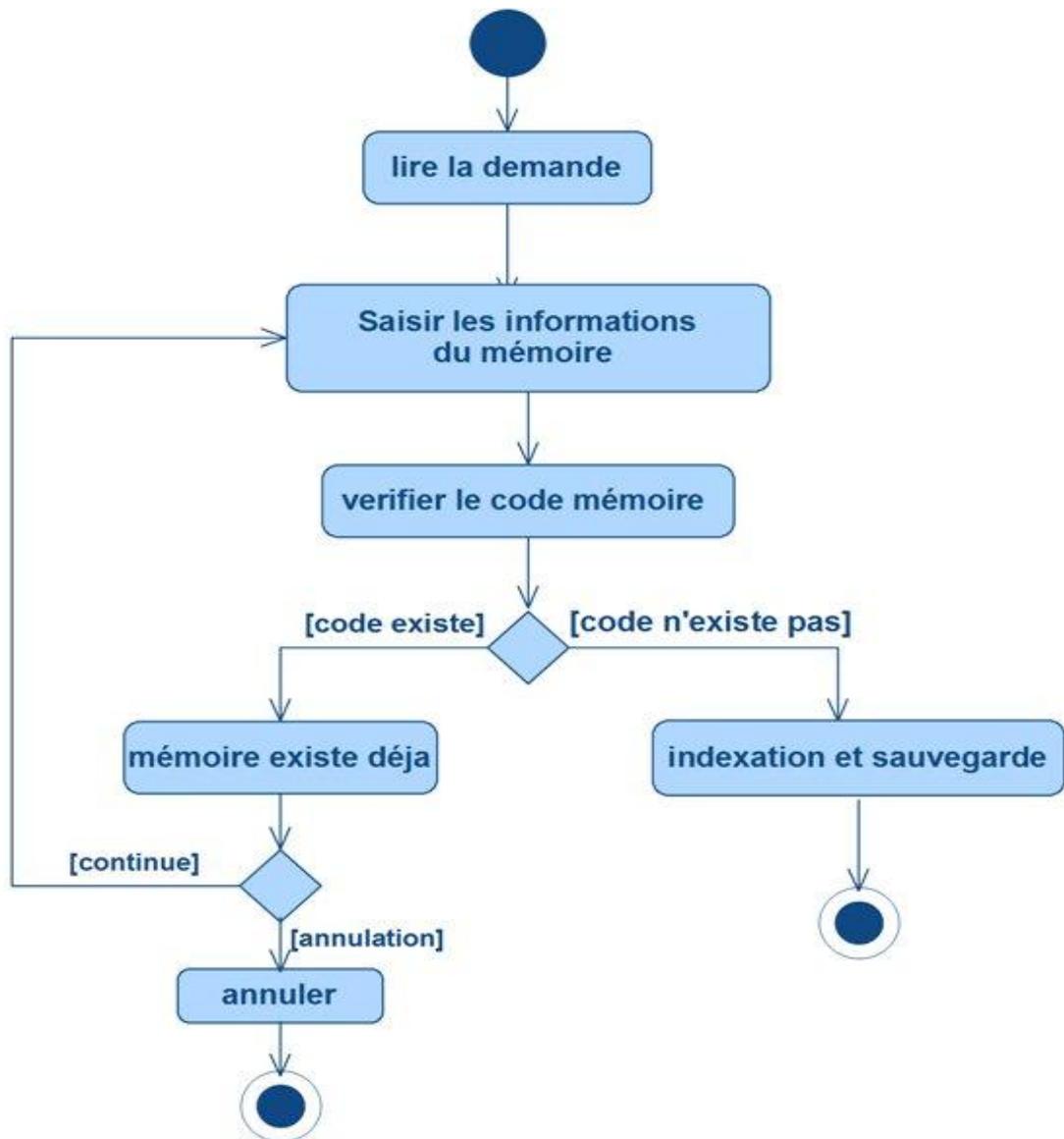


Figure IV.25: Diagramme d'activité du cas d'utilisation archiver un mémoire.

❖ Supprimer un mémoire

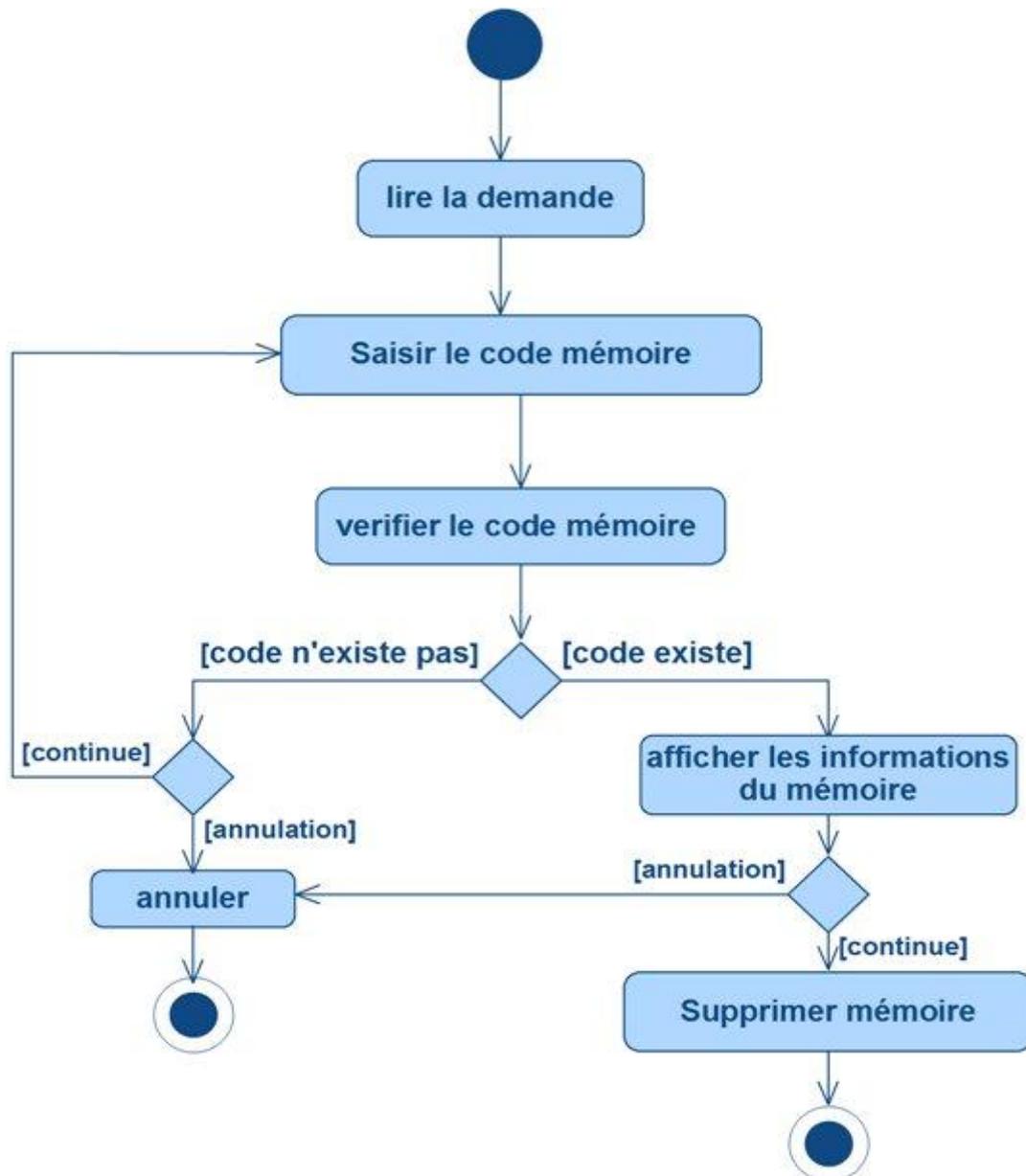


Figure IV.26: Diagramme d'activité du cas d'utilisation supprimer un mémoire.

➤ Rechercher et consulter des mémoires (titre, mots clés, code)

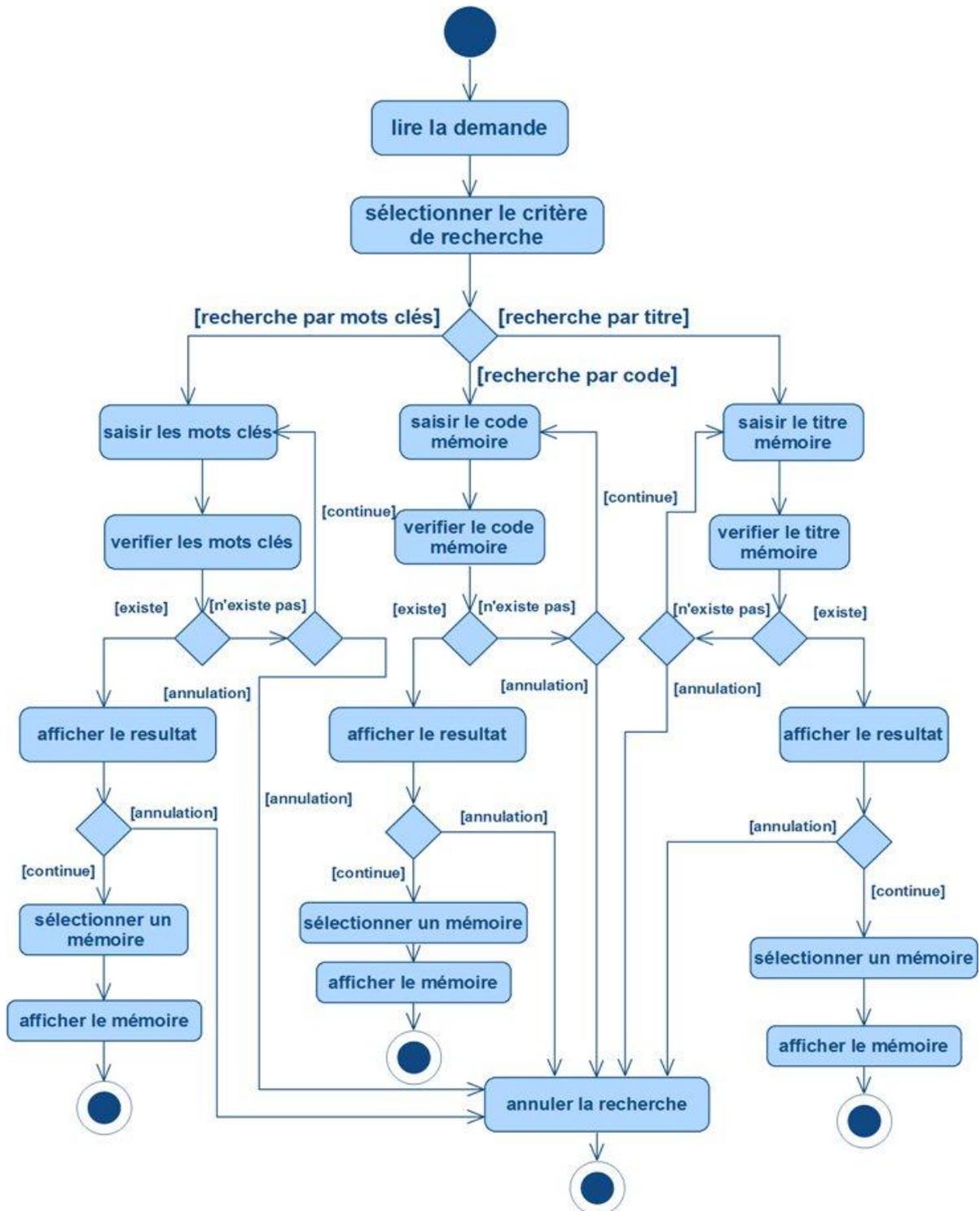


Figure IV.27: Diagramme d'activité du cas d'utilisation rechercher et consulter des mémoires (titre, mots clés, code).

Conclusion

A l'issue de cette étape nous avons exprimé clairement les objectifs attendus du futur système à concevoir, ainsi que l'analyse associée à chaque cas d'utilisation et la possibilité de les réaliser dans un paradigme orienté objet, sans s'attacher à aucun outil de développement.

Nous allons dans le chapitre suivant faire la conception de notre système.

Chapitre V : *Phase de conception*

Introduction

V.1. Les diagrammes de séquences.

V.3. Diagramme de classes.

V.4. Dictionnaire de données.

V.5. Conception de la base de données.

V.6. Diagramme déploiement.

Conclusion

Introduction :

Dans cette phase nous allons élaborer les diagrammes de séquence, le diagramme de classe et faire la conception de la base de données du système.

V.1. Les diagrammes de séquences

➤ **Authentifier**

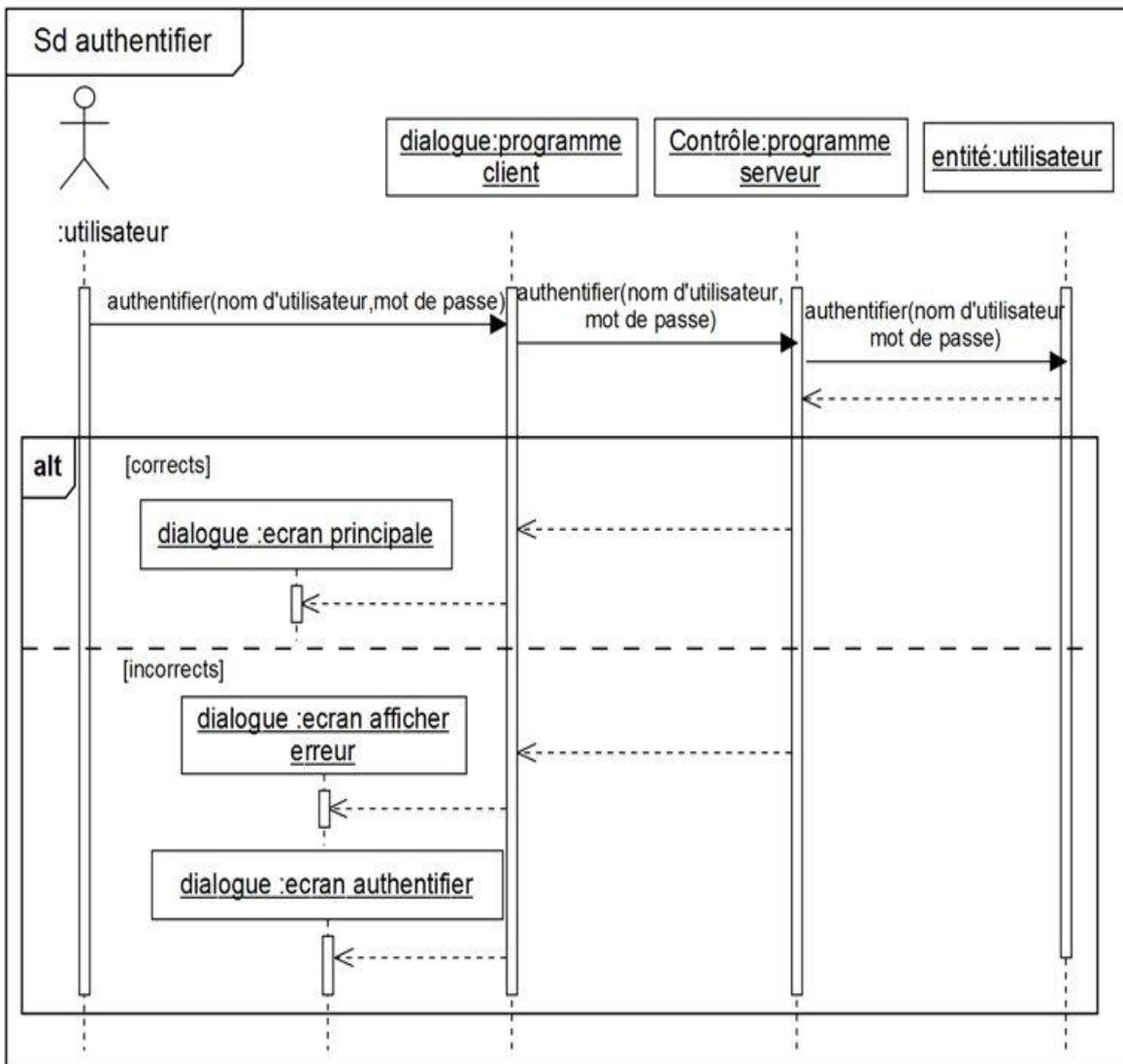


Figure V.28: Diagramme de séquence du cas d'utilisation authentifier.

- **Gestion des utilisateurs**
 - ❖ **Ajouter un utilisateur**

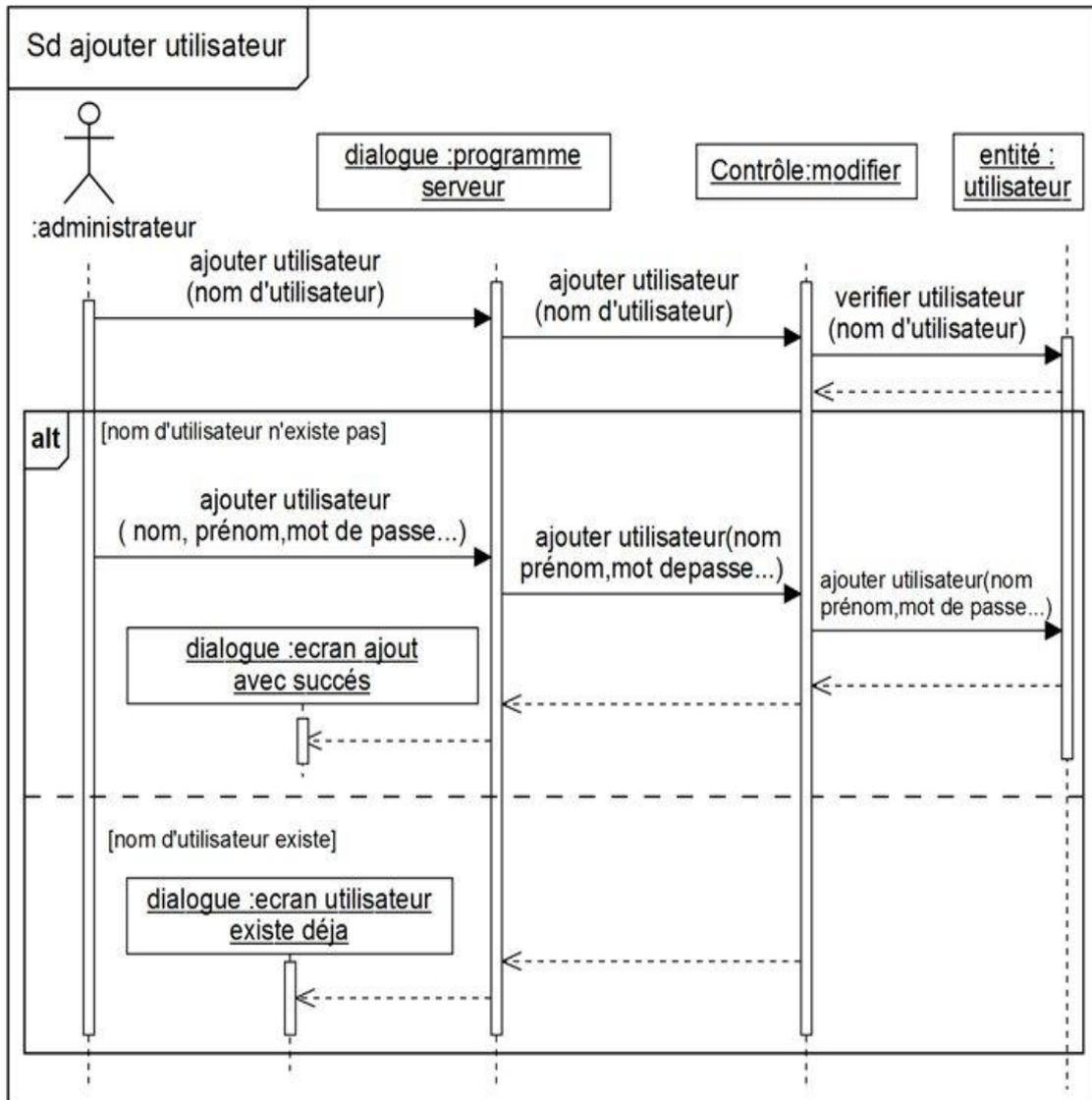


Figure V.29 : Diagramme de séquence du cas d'utilisation ajouter un utilisateur.

❖ Supprimer un utilisateur

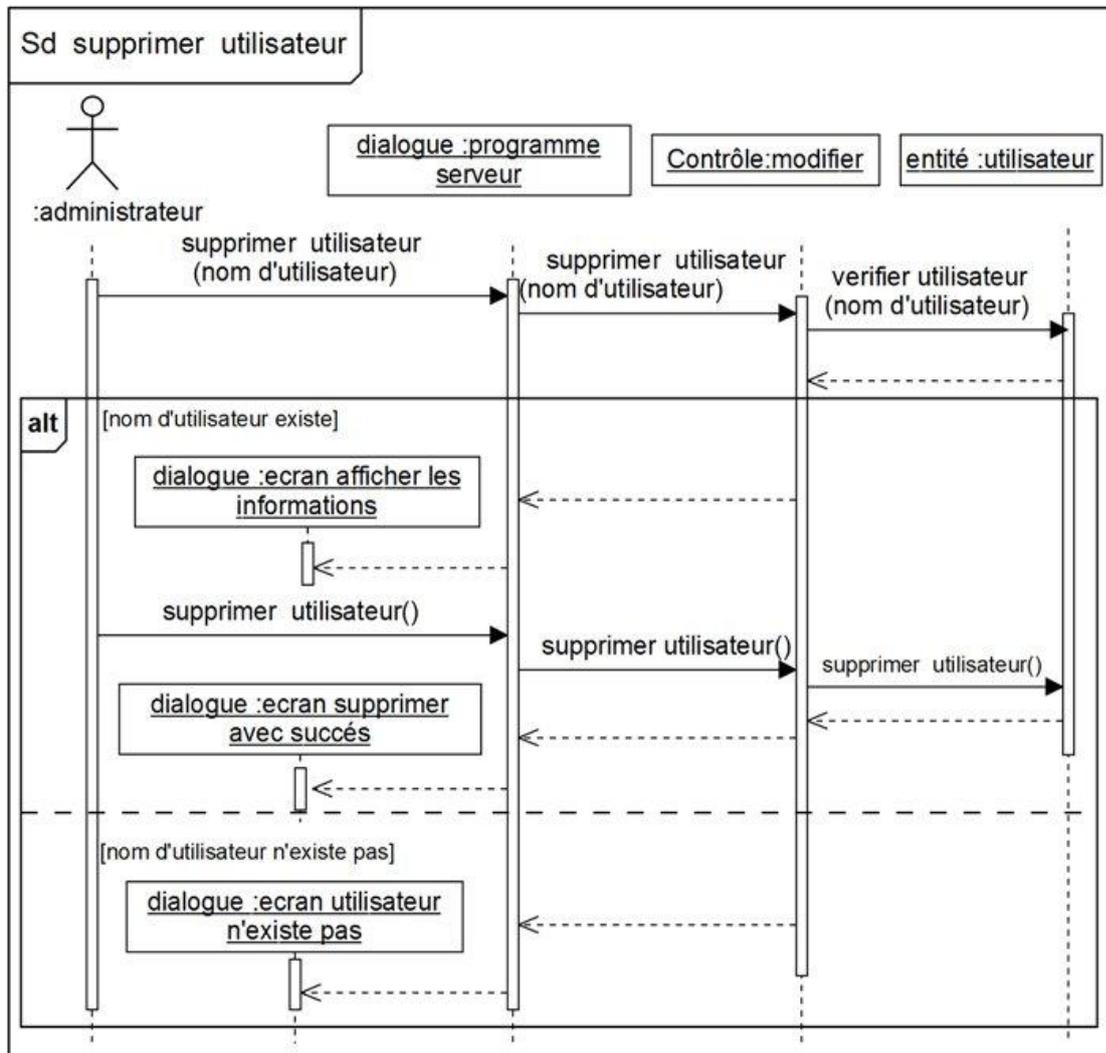


Figure V.30 : Diagramme de séquence du cas d'utilisation supprimer un utilisateur.

❖ **Modifier un utilisateur**

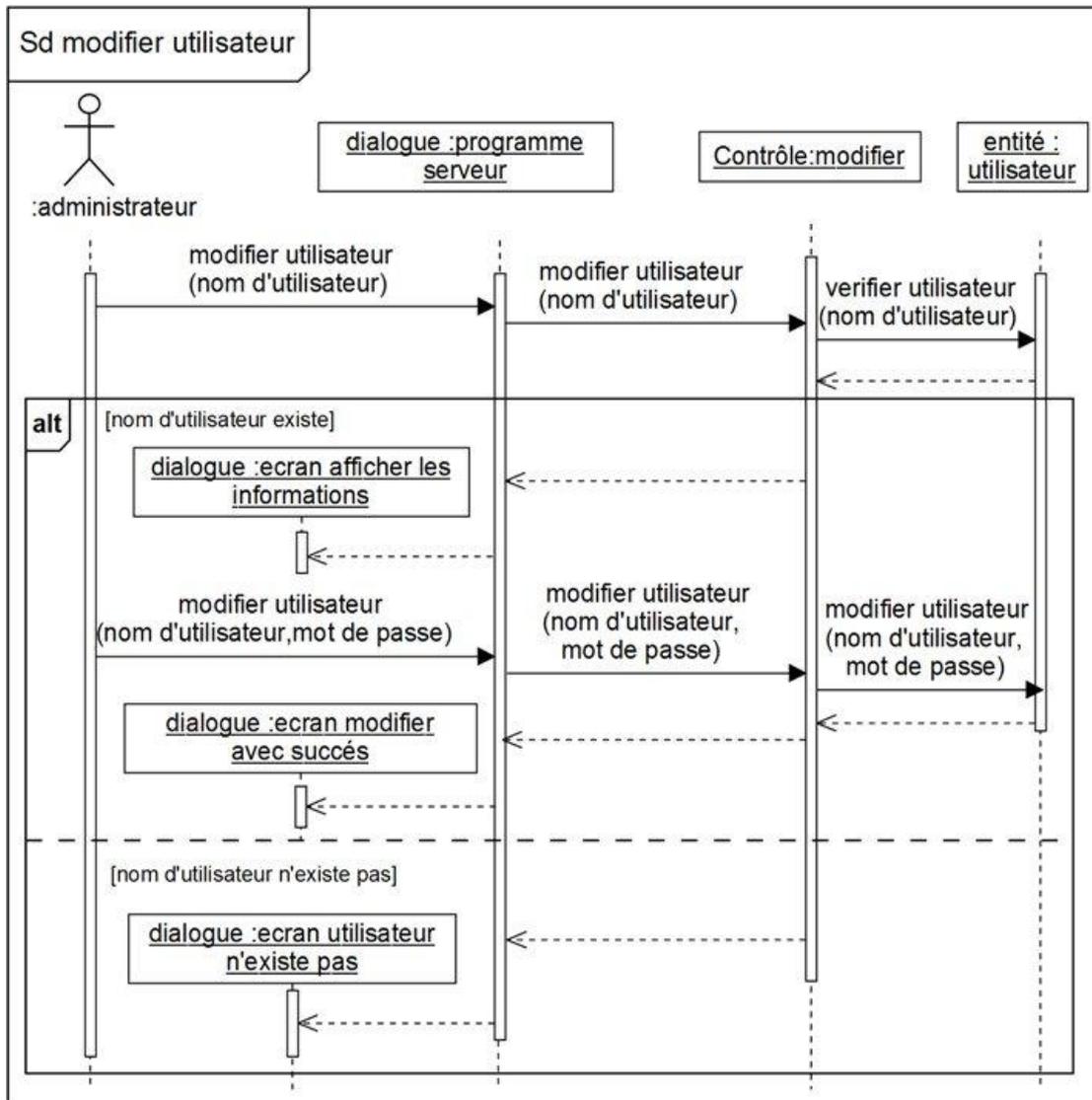


Figure V.31 : Diagramme de séquence du cas d'utilisation modifier un utilisateur.

➤ **Archiver un mémoire**

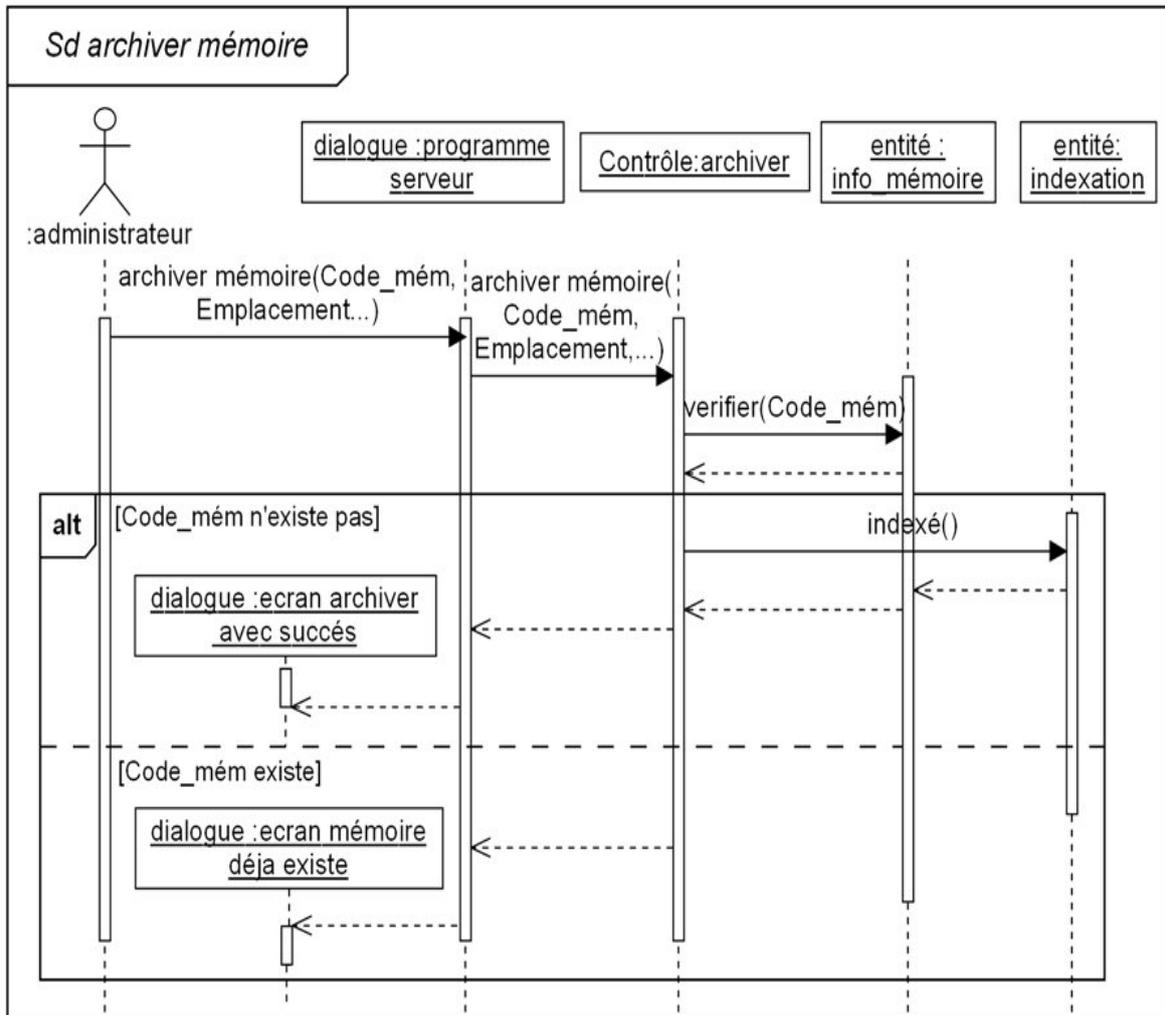


Figure V.32 : Diagramme de séquence du cas d'utilisation archiver un mémoire.

❖ Supprimer un mémoire

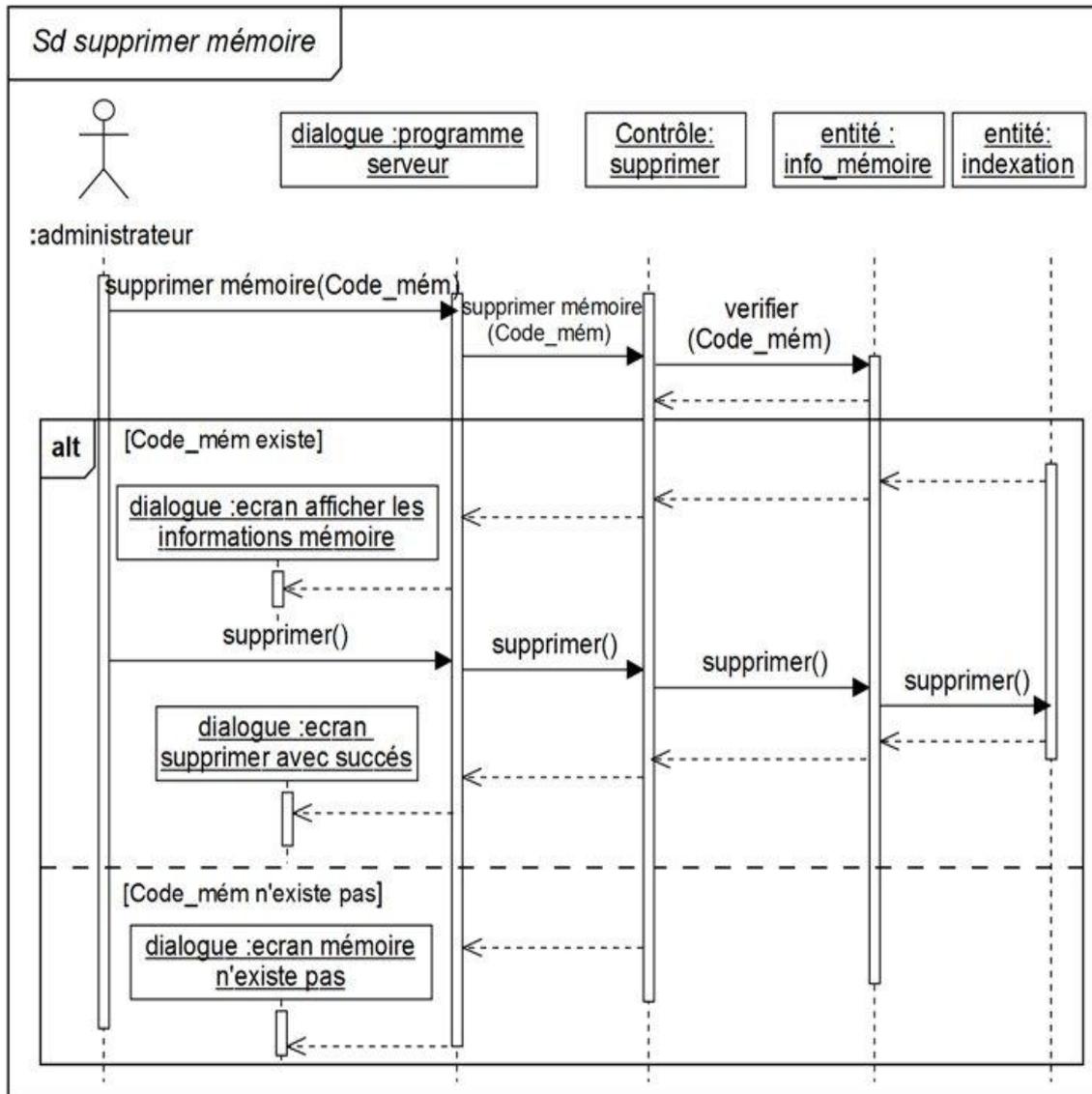


Figure V.33 : Diagramme de séquence du cas d'utilisation supprimer un mémoire.

- ❖ Rechercher et consulter un mémoire
- ❖ Rechercher et consulter un mémoire par Titre

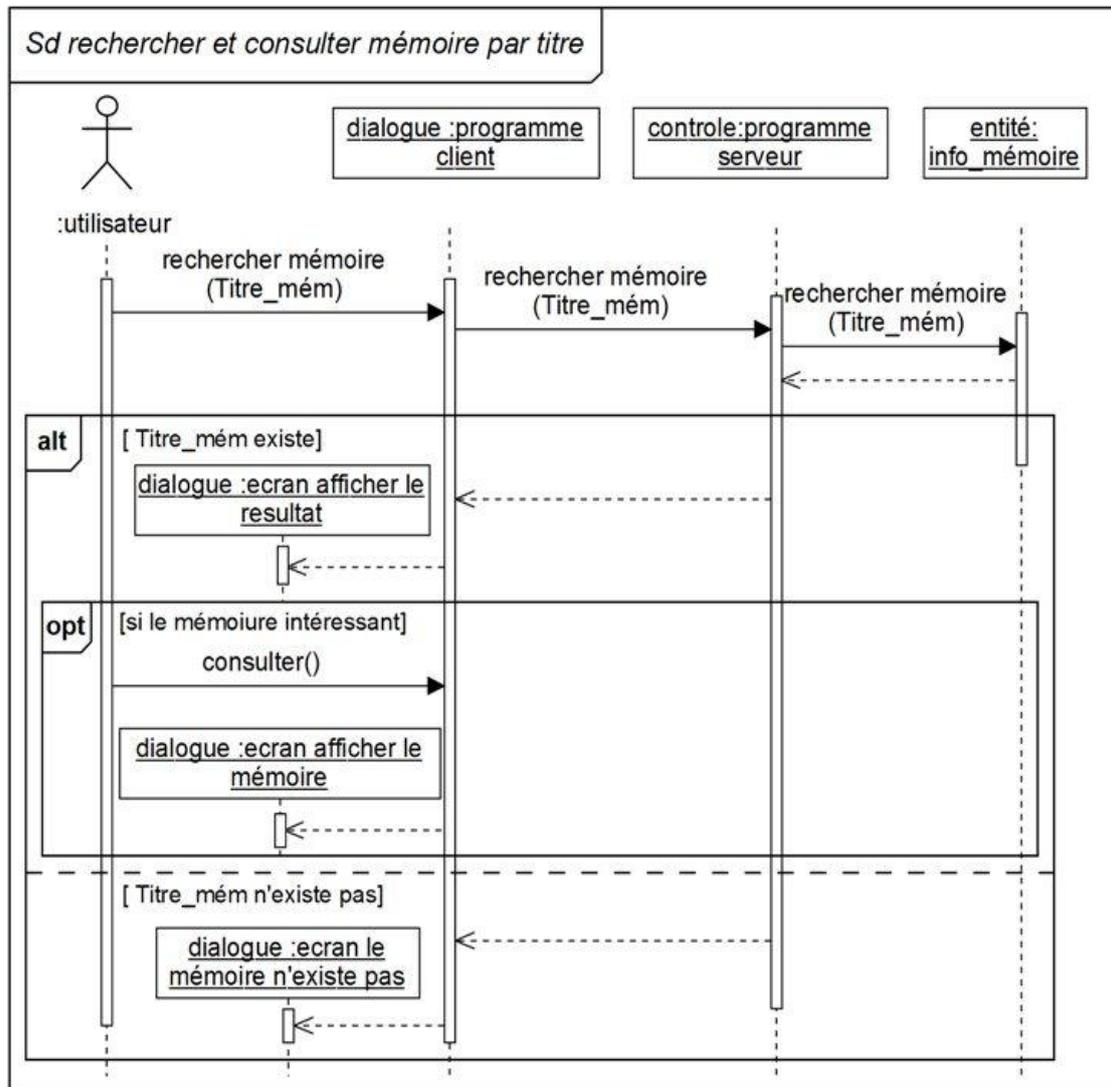


Figure V.34 : Diagramme de séquence du cas d'utilisation rechercher et consulter un mémoire par titre.

❖ **Rechercher et consulter un mémoire par mots clés**

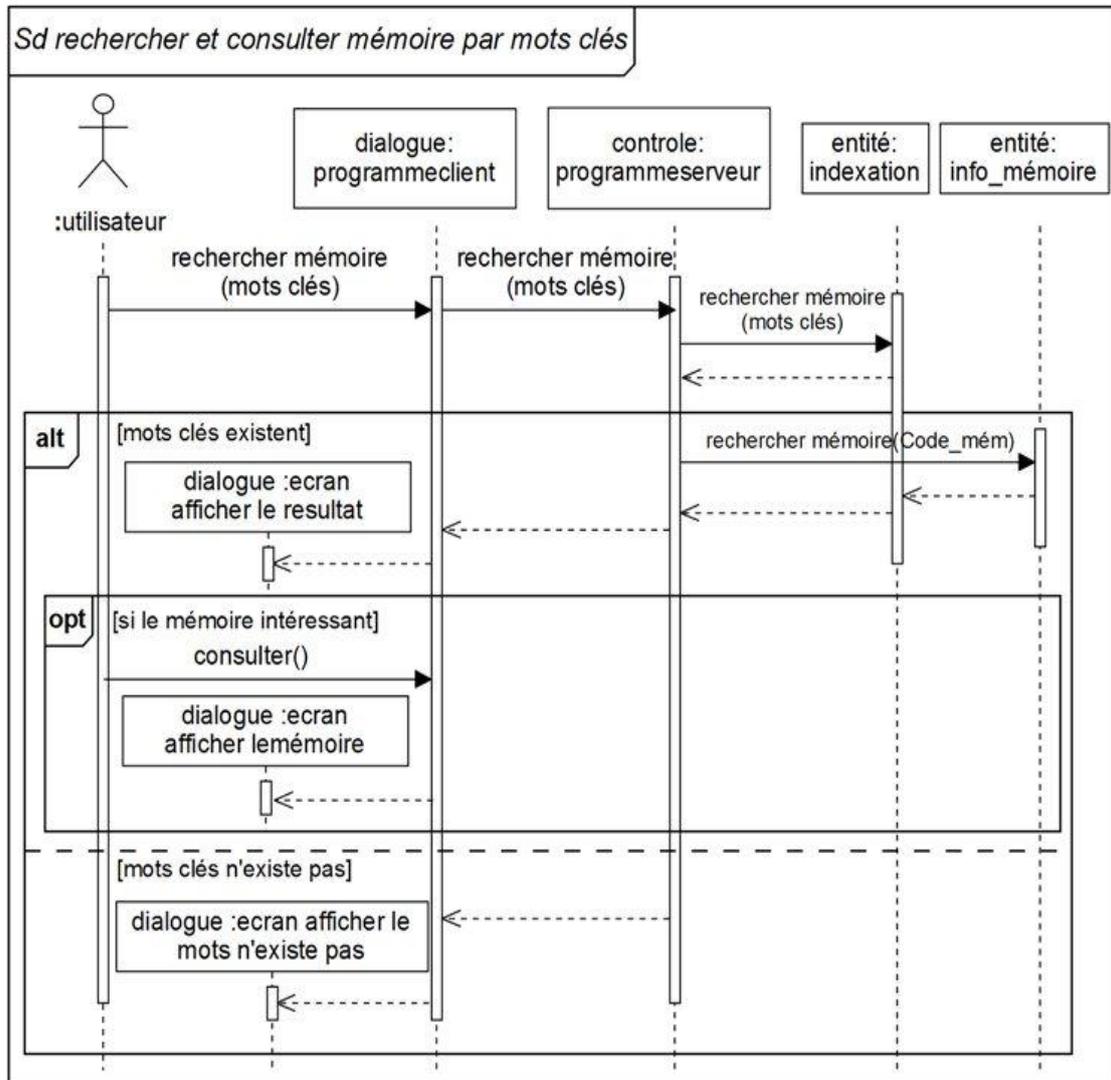


Figure V.35 : Diagramme de séquence du cas d'utilisation rechercher et consulter un mémoire par mots clés.

❖ **Rechercher et consulter un mémoire par code**

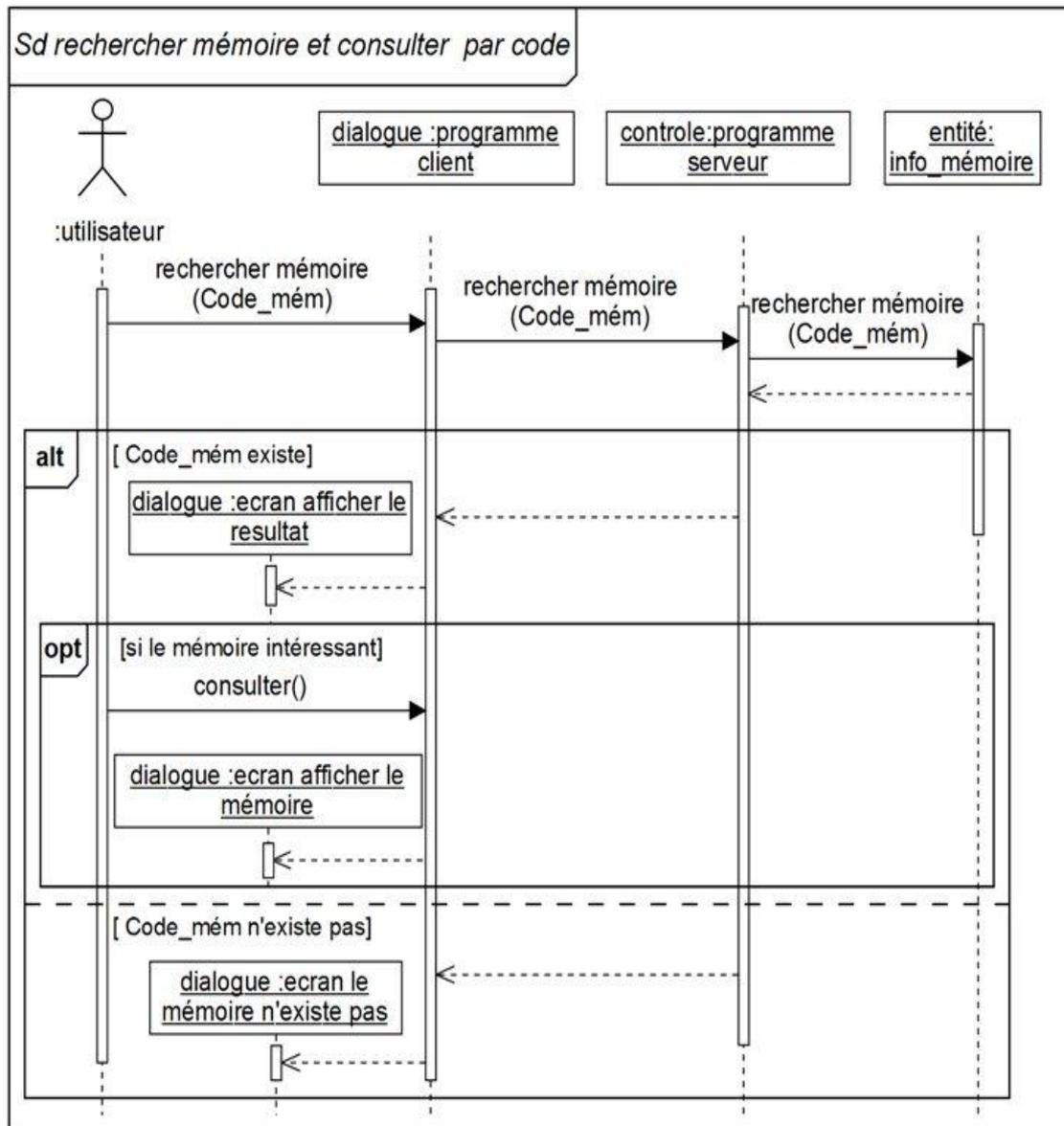


Figure V.36: Diagramme de séquence du cas d'utilisation rechercher et consulter un mémoire par code.

V.2. Diagramme de classes

Dans cette étape, nous regroupons les identifiées dans l'étape précédent dans un diagramme qui s'appelle diagramme de classe pour chaque classe nous décrivons les liens structurels avec les autres classes.

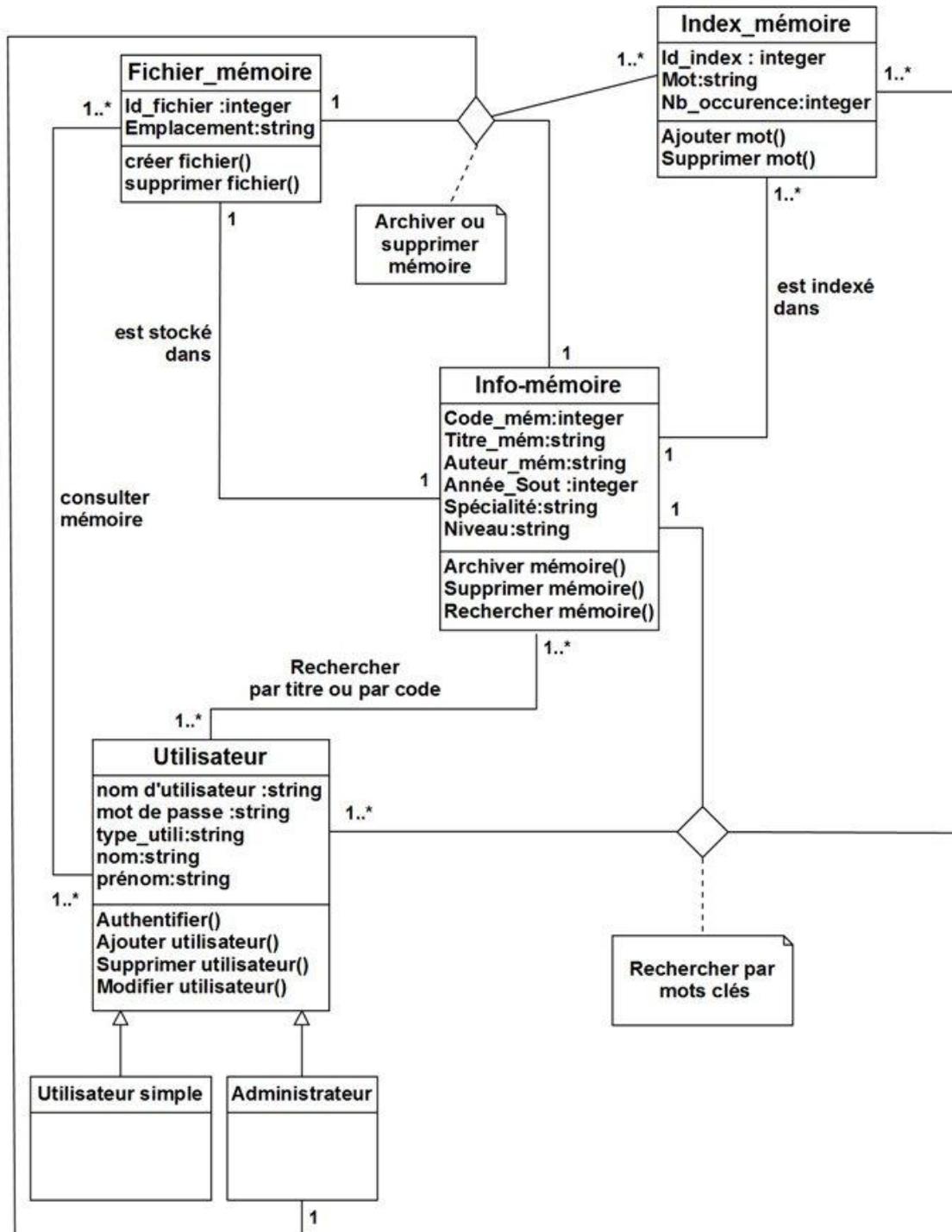


Figure V.37 : Diagramme de classes.

V.3. Dictionnaire de données• **Les classes et les attributs**

| classe | information | code | type |
|-----------------|------------------------|---------------------------------|--------------|
| Info_mémoire | Code mémoire | <u>Code mém</u> | Int(11) |
| | Titre mémoire | Titre_mém | Varchar(30) |
| | Auteur mémoire | Auteur_mém | Varchar(30) |
| | Année de Soutenance | Année_Sout | Varchar(30) |
| | Spécialité | Spécialité | Varchar(30) |
| | Niveau | Niveau | Varchar(30) |
| Utilisateur | Nom d'utilisateur | <u>Nom d'utilisateur</u> | Varchar(30) |
| | Mot de passe | Mot de passe | Varchar(30) |
| | Nom | Nom | Varchar(30) |
| | Prénom | Prénom | Varchar(30) |
| | Type d'utilisateur | Type_utilili | Varchar(30) |
| Index_mémoire | Identifiant l'index | <u>Id index</u> | Int(11) |
| | Les mots clés | Mot | Varchar(30) |
| | Le nombre d'occurrence | Nb_occurrence | Int(11) |
| Fichier_mémoire | Identifiant fichier | <u>Id fichier</u> | Int(11) |
| | Emplacement de mémoire | Emplacement | Varchar(100) |

- **Les opérations**

| classe | méthode | description |
|-----------------|--------------------------|-----------------------------|
| Info_mémoire | Archiver mémoire () | Archiver un nouveau mémoire |
| | Supprimer mémoire () | Supprimer un mémoire |
| | rechercher mémoire () | Rechercher un mémoire |
| Utilisateur | Authentifier () | Authentifier un utilisateur |
| | Ajouter utilisateur () | Ajouter un utilisateur |
| | Supprimer utilisateur () | Supprimer un utilisateur |
| | Modifier utilisateur () | Modifier un utilisateur |
| Index_mémoire | Ajouter mot () | Ajouter un mot |
| | Supprimer mot () | Supprimer un mot |
| Fichier_mémoire | créer fichier () | créer un fichier |
| | Supprimer fichier () | Supprimer un fichier |

V.4. Conception de la base de données

Nous donnons après quatre règles (de R1 à R4) pour traduire un schéma conceptuel entité association ou UML en un schéma relationnel équivalent. il existe d'autres solutions de transformation mais ces règles sont les plus simples et les plus opérationnelles.

- **Transformation des entités/classes : la règle est simple**

- ✓ R1 : * Chaque entité devient une relation, identifiant de l'entité devient clé primaire pour la relation.

- * Chaque classe du diagramme UML devient une relation. Il faut choisir les attributs de la classe pouvant jouer le rôle d'identifiant.

- **Transformation des associations**

Les règles de transformation que nous allons voir dépendent des cardinalités / multiplicités maximale des associations. Nous distinguons trois familles d'association :

Association 1..* : La règles est la suivant :

- ✓ R2 : Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

Association *.* : La règle est la suivant :

- ✓ R3 : L'association/classe-association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des identités connectées à association. Chaque attribut devient clé étrangère si entité/classe connectée dont il devient une relation en vertu de la règle R1. Les attributs de l'association/classe-association doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.

Association 1..1 : La règle est la suivante :

- ✓ R4 : il faut ajouter un attribut de type clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimale égale à zéro. Dans le cas du diagramme UML il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de l'entité/classe connectée à l'association. Si les deux cardinalités minimales égale à zéro, le choix est donné entre les deux relations dérivées de la R1. Si les deux cardinalités minimales égale à un, il est préférable de fusionner les deux entités/classe en une seule.

Après l'application des règles de passage, nous avons déduit les tables suivant :

Info_mémoire (Code_mém, Titre_mém, Auteur_mém, Année_Sout, Spécialité, Niveau, Emplacement).

Utilisateur (Nom_d'utilisateur, mot de passe, nom, prénom, Type_utilili).

Index_mémoire (#Code_mém, Mot, Nb_occurrence).

V.5. Diagramme déploiement

Dans cette étape, nous avons montré la disposition physique des matérielles qui composent le système et la répartition des composants sur ces matérielles.

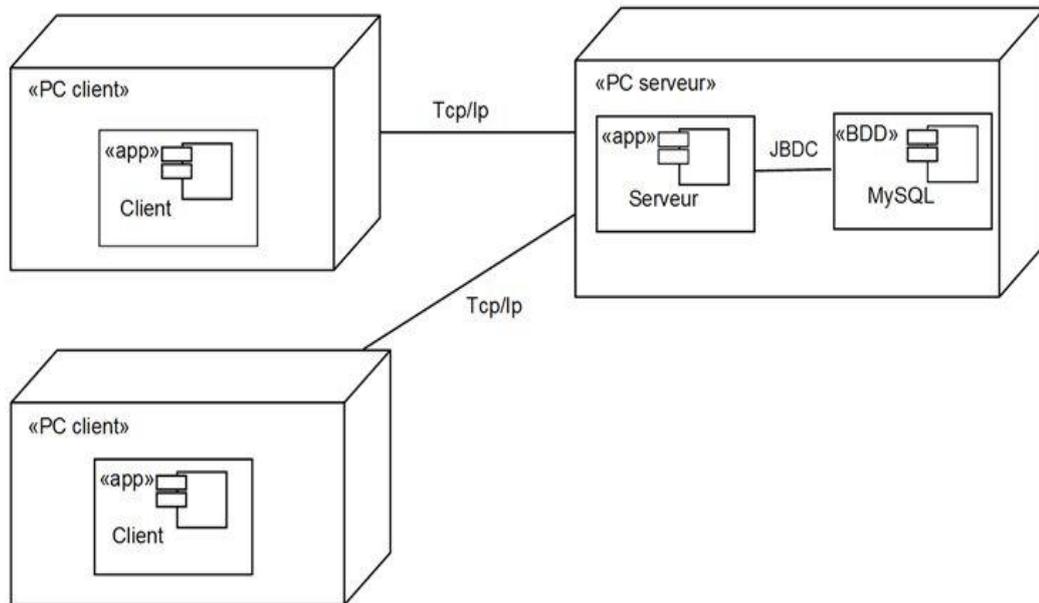


Figure V.38 : Diagramme déploiement.

Conclusion

Durant cette partie nous avons présenté le coté conceptuel de notre application en utilisant la modélisation UML afin de répondre au mieux aux exigences et besoins des utilisateurs de notre système. A base de cette dernière, on a abouti au modèle relationnel des données en faisant appel aux règles de passages. Dans le prochain chapitre nous allons présenter l'environnement de développement que nous avons utilisé. Et quelques interfaces de notre application.

Chapitre VI : Implémentation

Introduction

VI.1. Environnement de développement de l'application.
VI.2. Quelques interfaces de l'application

Conclusion

Introduction

En utilisant le diagramme de classe de conception que nous avons vu dans le chapitre précédent, on commence la phase d'implémentation. Cette dernière phase a pour but de générer un code exécutable répond aux besoins utilisateur.

VI.1. Environnement de développement de l'application



VI.1.1. Le langage de programmation java

Pour implémenter notre application Nous avons choisi le langage de développement java avec l'IDE Netbeans car java est un langage orienté objet simple et portable « Il peut être utilisé sous différentes plates formes sans aucune modification », java possède aussi une riche bibliothèque de classes, Netbeans est un environnement de développement intégré, open source, très utile qui permet de développer en java.



VI.1.2. MySQL

Pour implémenter la base de données, nous avons utilisé le SGBD MySQL qu'est un serveur de base de données relationnelle développé dans un souci de performance élevée, il est libre, rapide, multi threaded et multi utilisateur.

VI.1.3. PhpMyAdmin



L'outil PhpMyAdmin offre une interface graphique très simple pour l'administration complète des bases de données MySQL. Nous l'avons utilisé pour la création de notre base de données. Nous mentionnons quelques fonctionnalités:

ÉGestion (création, modification et suppression) des bases de données.

ÉGestion des utilisateurs de la base de données et leurs droits d'accès.

ÉL'exécution des requêtes SQL (écrits par les utilisateurs).

VI.1.4. Connecteur

On a utilisé le connecteur (mysql-connector-java) pour connecter notre logiciel avec la base de données.

VI.1.5. Partie client/serveur

L'informatique client/serveur propose un environnement ouvert et souple, chaque machine client demande un service par l'envoi d'une requête au serveur, le serveur envoie la réponse. On a utilisé les threads et les sockets pour réaliser l'architecture client/serveur.

VI.1.5.1. Les threads

Les threads sont des processus spéciaux qui coopèrent et s'exécutent dans le même espace d'adressage (celui du processus qui les a créés). Les threads sont des processus légers parce qu'ils partagent la mémoire du processus qui les contient et cela rend les changements de contexte peu coûteux en temps.

VI.1.5.2. Les sockets

C'est une Interface de programmation avec les services du système d'exploitation pour exploiter les services de communication du système (local ou réseau). Une socket communique avec une autre socket via à son adresse.

Adresse de socket = adresse IP + port.

VI.1.6. Les bibliothèques poi

On a utilisé les bibliothèques poi pour extraire le contenu textuel des mémoires, puisque les mémoires avec extension.doc doivent être décodés avant d'être indexés.

VI.1.7. Programme d'indexation

Après l'archivage d'un mémoire de fin d'étude, on a lancé l'indexation sur ce mémoire .doc. La figure suivante montrera les étapes de programme d'indexation

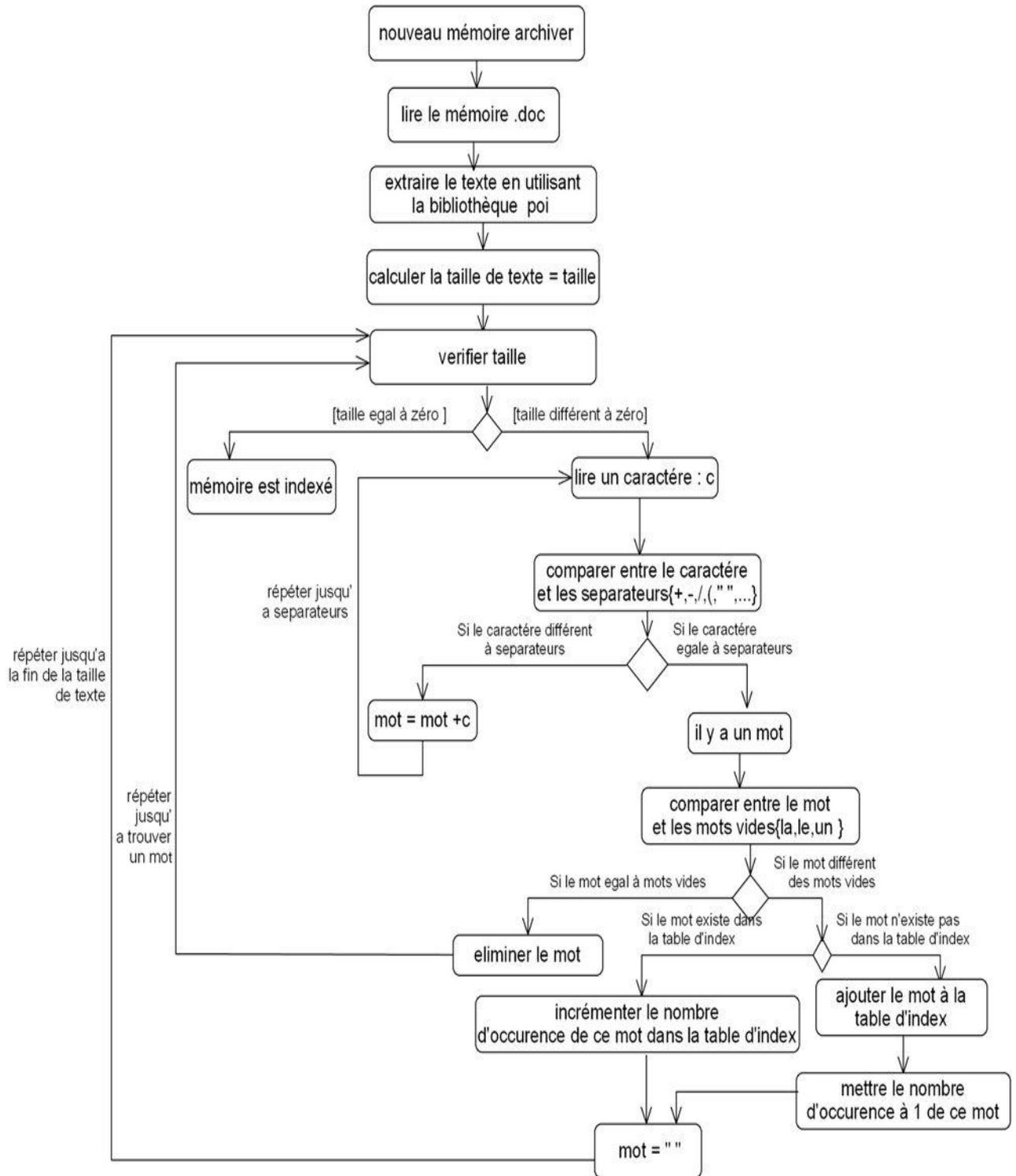
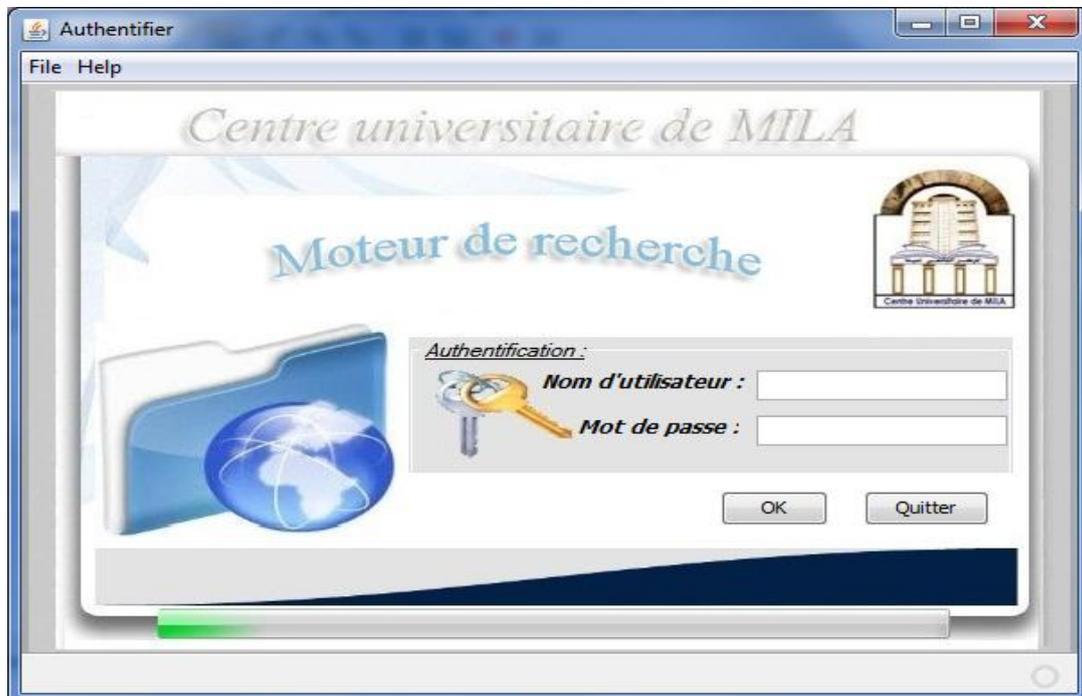


Figure VI.39. Programme d'indexation.

VI.2. Quelques interfaces de l'application

Dans cette section nous présentons quelque fenêtre de notre application :

- ✓ **Fenêtre d'authentifier** : permet d'authentifier les utilisateurs du système.



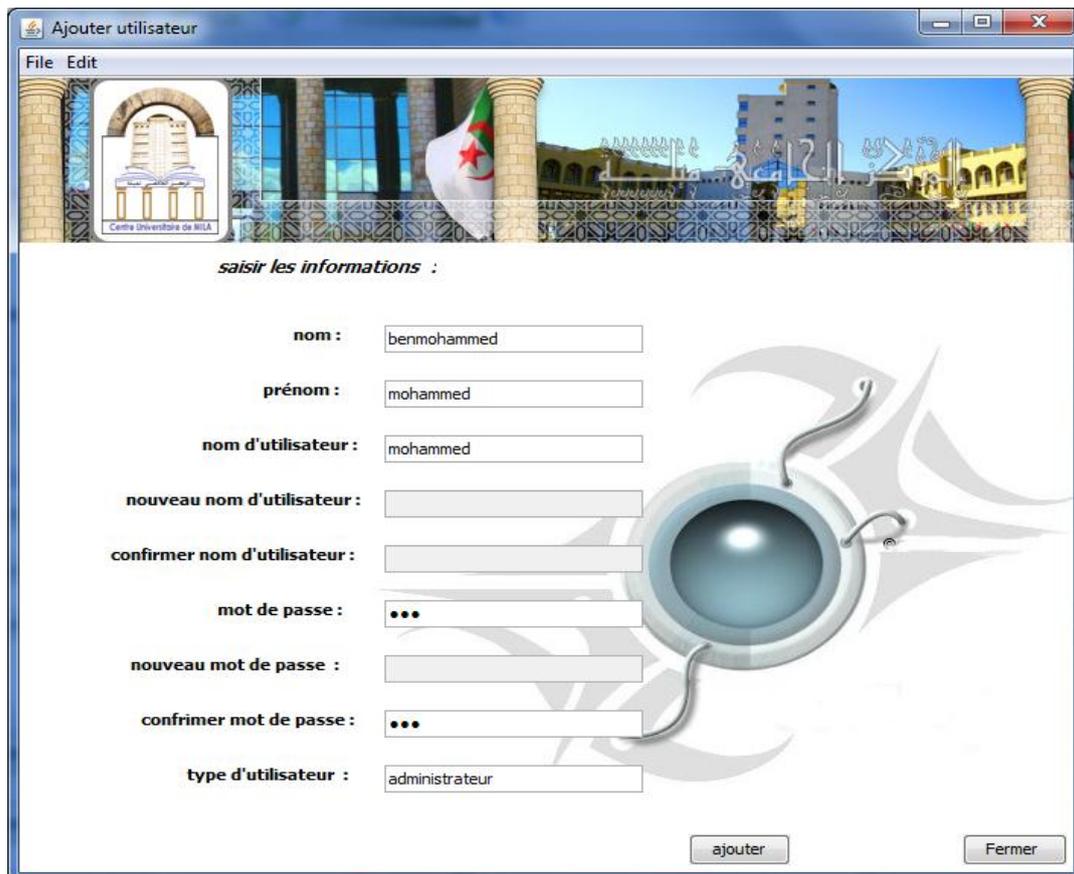
- ✓ **Fenêtre principale** : permet de sélectionner une opération.



- ✓ **Fenêtre vérifier l'existence d'utilisateur :** permet de vérifier l'existence d'un utilisateur ou non pour ajouter, supprimer ou modifier.



- ✓ **Fenêtre Ajouter utilisateur :** permet d'ajouter des nouveaux utilisateurs.



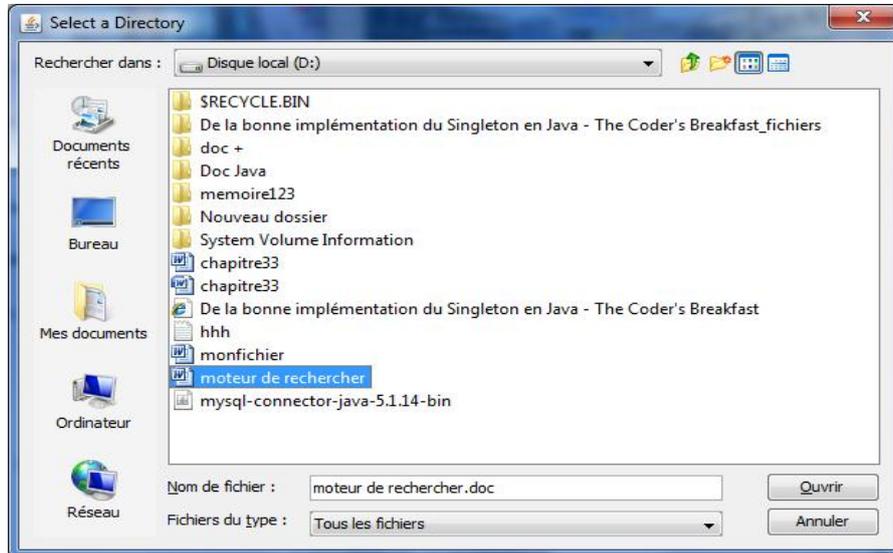
- ✓ Fenêtre supprimer utilisateur : permet de supprimer un utilisateur.



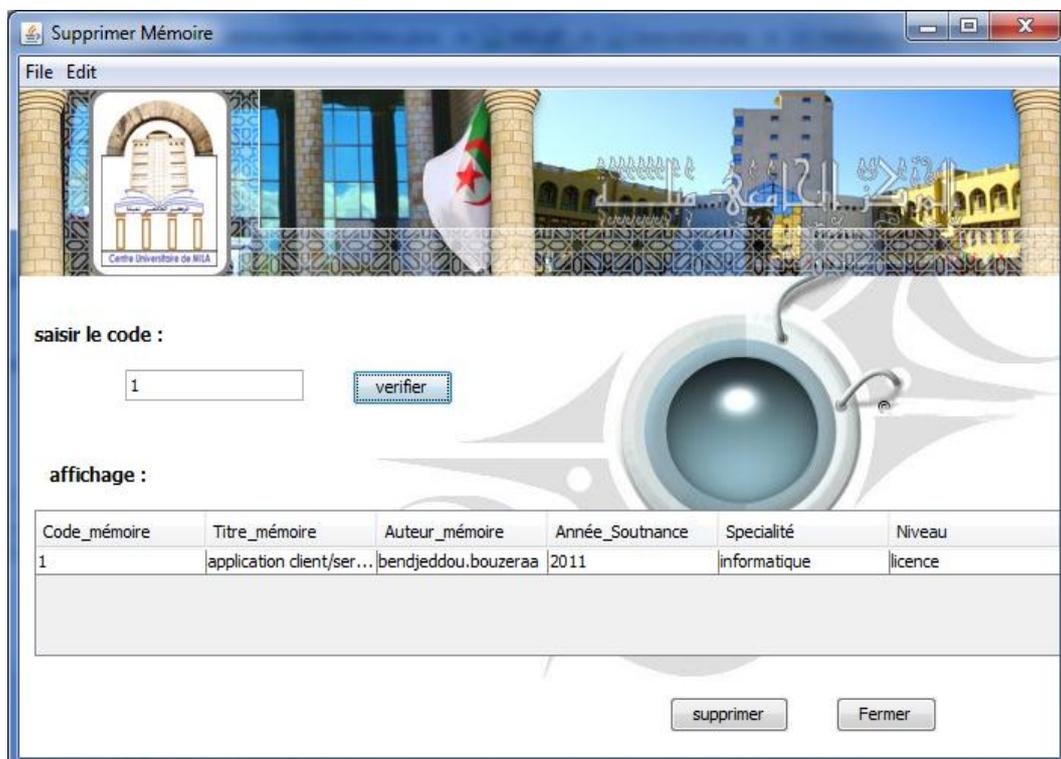
- Fenêtre Archiver Mémoire : permet d'archiver des nouveaux mémoires.



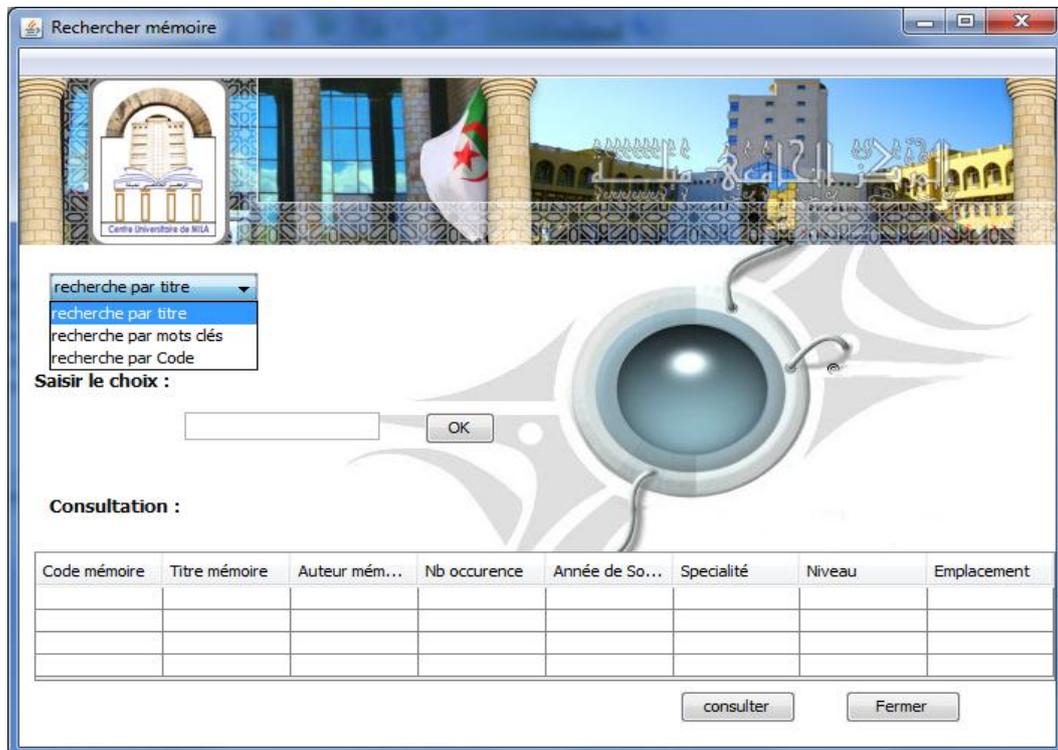
Le bouton "parcourir" permet d'indiquer l'emplacement du mémoire, ouvrir la fenêtre suivante par sélectionner le fichier qui contient le mémoire.



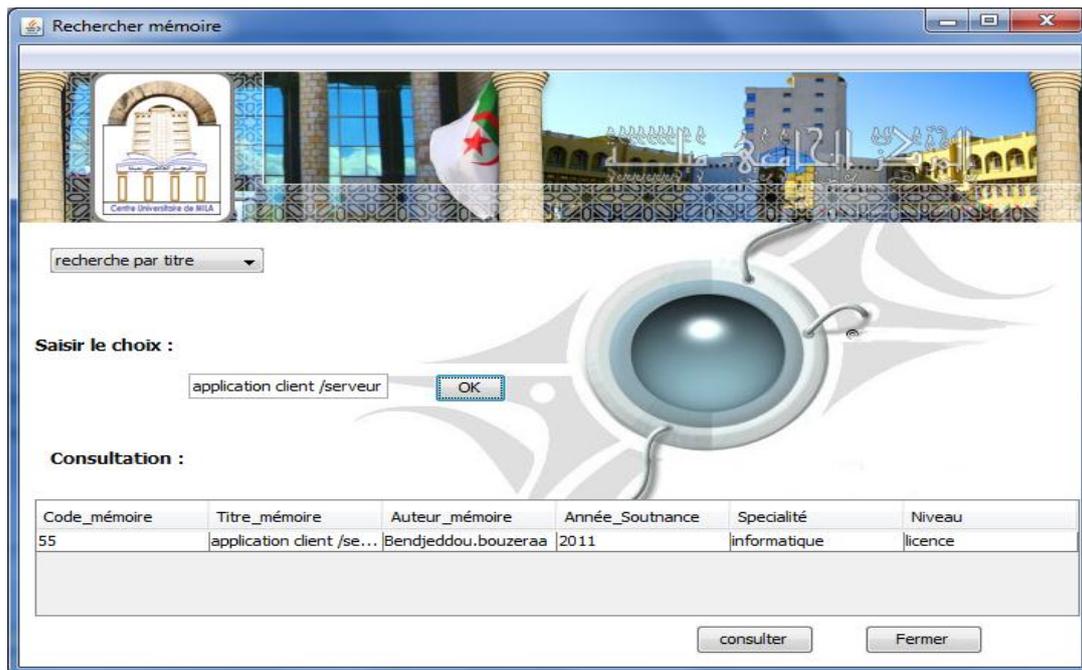
✓ **Fenêtre Supprimer Mémoire :** permet de supprimer des mémoires.



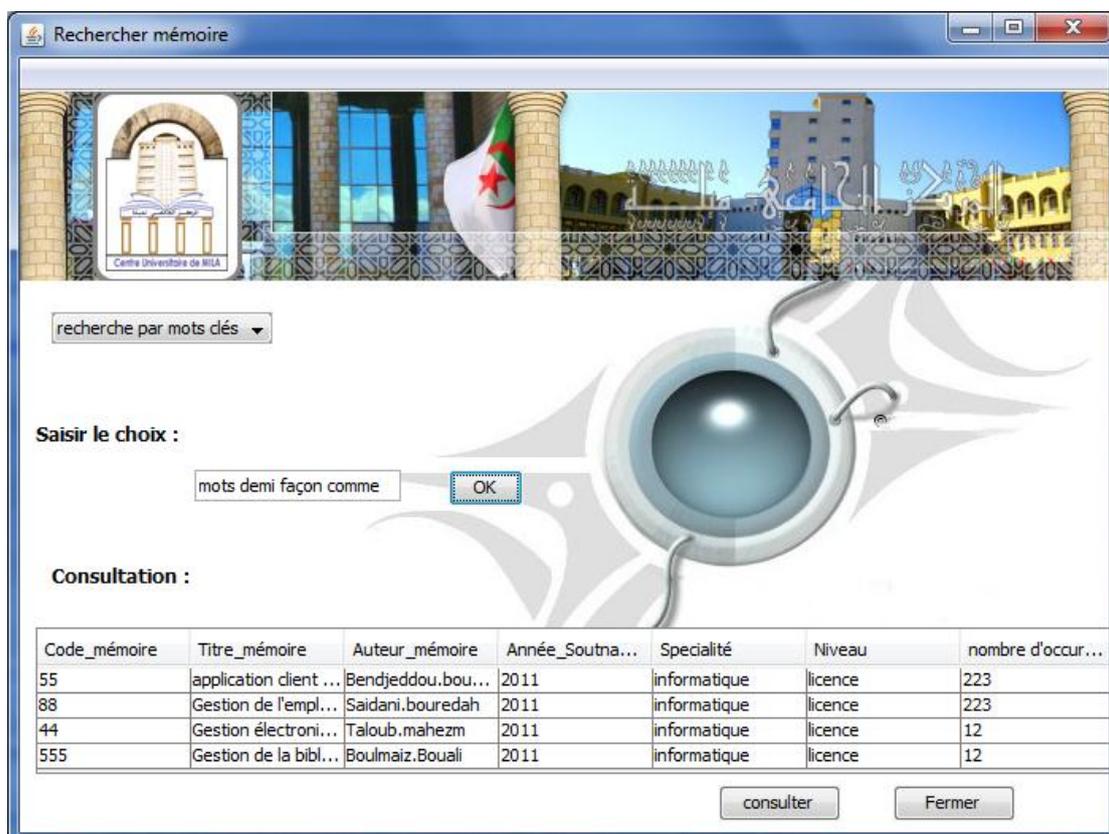
- ✓ **Fenêtre Rechercher Mémoire :** permet de rechercher des mémoires soit par titre, par mots clés ou par code.



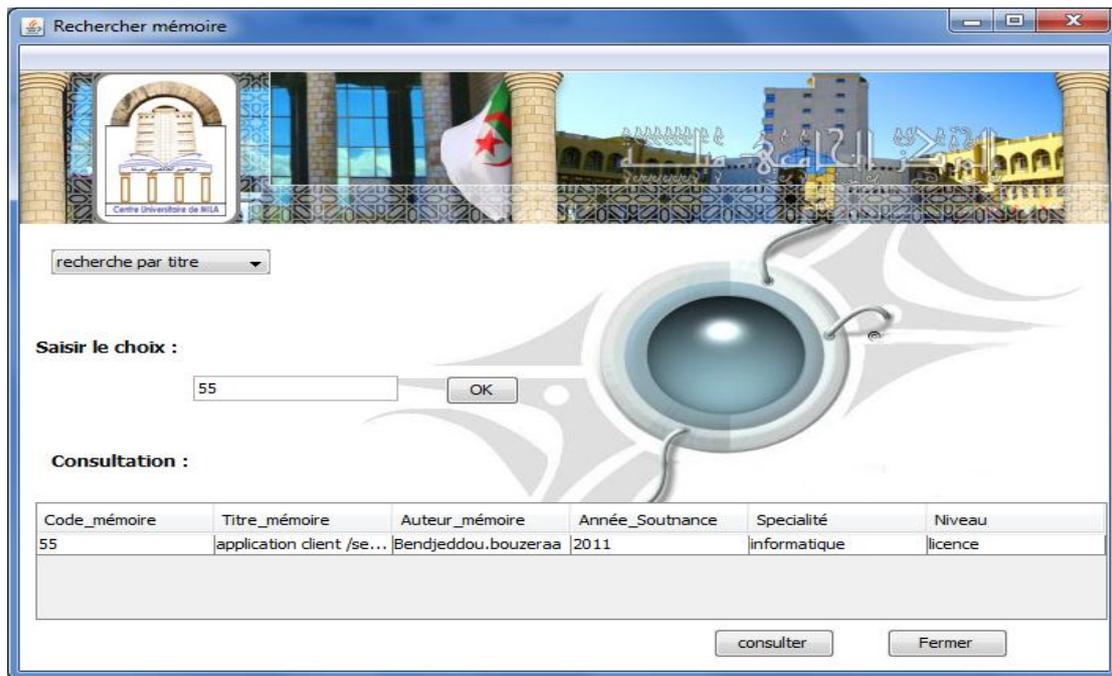
❖ **Rechercher par titre du mémoire**



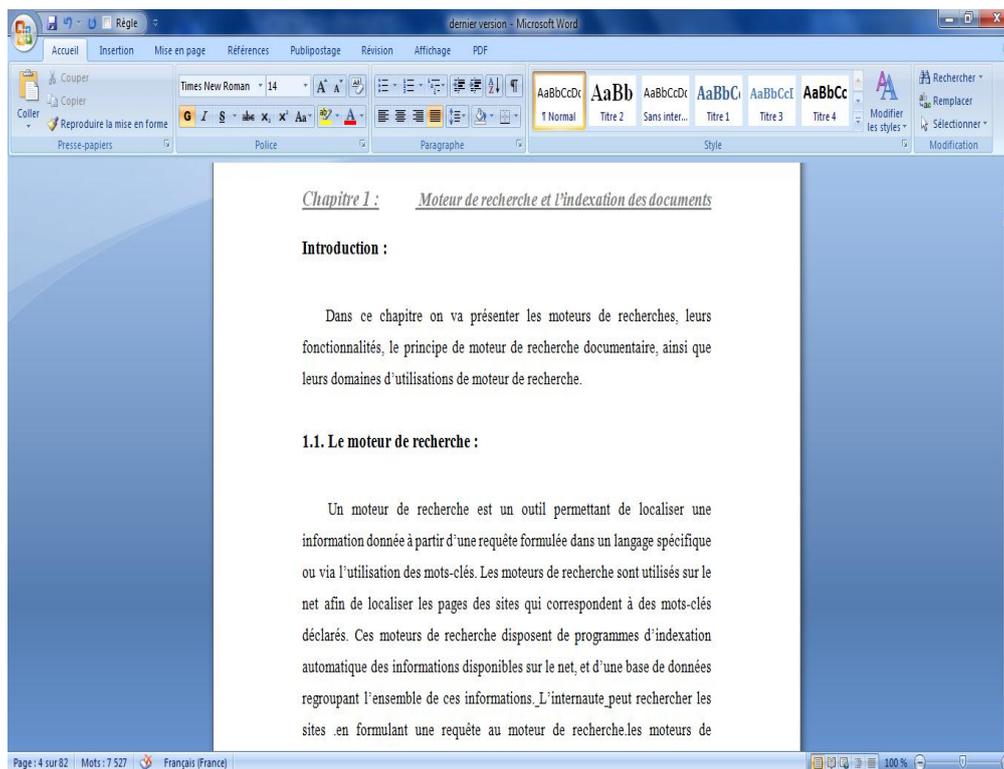
❖ **Rechercher par mots clés**



❖ **Rechercher par code mémoire**



✓ **Consultation des mémoires**



Conclusion

Dans cette partie nous avons présenté l'environnement de développement de notre application. Nous pouvons dire qu'on a réalisé un travail de qualité en développant des interfaces performantes permettant à l'utilisateur d'effectuer des recherches avancées, précises et rapides.

Conclusion générale :

Au cours de ce projet nous avons développé une application client/serveur pour l'archivage, la recherche et la consultation des mémoires de fin d'étude de la bibliothèque du centre universitaire de Mila. L'interface de notre application est claire et conviviale. Les utilisateurs peuvent archiver et consulter les mémoires en toute simplicité, Ils peuvent rechercher rapidement les mémoires soit en utilisant les titres, les codes, ou les mots clés à l'aide d'un index.

Notre application permet à plusieurs utilisateurs d'accéder au système en même temps. Elle est fiable et sécurisé et cela grâce à l'architecture client/serveur et à l'authentification des utilisateurs.

Ce travail nous a permis d'enrichir nos connaissances dans des domaines variés comme : L'architecture Client/serveur, l'Orienté Objet, le langage de modélisation UML, le langage de programmation JAVA et le SGBD MYSQL.

Notre application permet seulement d'indexer les fichiers de type DOC, on espère que notre travail va aider les étudiants qui arriveront les années prochaines et qu'ils continueront l'implémentation des autres types comme PDF, DOCX, ODT etc...

Bibliographie

- ❖ « Conception et réalisation d'un Moteur d'indexation et de recherche dans un Intranet », Mémoire de fin d'étude, Institut national de formation en informatique, Année universitaire 1999/2000.
- ❖ « Indexation des documents Semi-Structurés : Proposition d'une approche basée sur le fichier inversé et le "Trie" », Mémoire de Magister, Institut National de formation en Informatique (I.N.I) Oued-Smar Alger, Année universitaire 2005/2006.
- ❖ « Modélisation et manipulation d'une GED pour le service des archives de la wilaya de Mila », Université Mentouri de Constantine, Année universitaire 2008/2009.
- ❖ « Vers l'automatisation de la banque de prêt de la bibliothèque universitaire centrale de l'université Mentouri de Constantine », Mémoire fin d'étude, Année universitaire 2008/2009.
- ❖ « Conception et réalisation d'un site web dynamique pour une agence immobilière, Mémoire de fin d'étude, Université Abderahmane Mira de Bejaia, Année universitaire 2008/2009.
- ❖ « UML 2 », Livre, Laurent AUDIBERT, Édition 2007-2008.
<http://laurent-audibert.developpez.com/Cours-UML/>.
- ❖ « UML en action », Livre, P. Roques et .Vallé, 2^{ieme} édition 2003.
- ❖ « UML pour la pratique (étude de cas exercices corrigées) », livre, Eyralles, 5^{ieme} édition.