

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

UNIVERSITÉ DE SAAD DAHLEB -BLIDA

FACULTÉ DES SCIENCES  
DÉPARTEMENT D'INFORMATIQUE



*Cycle* : Master

*Spécialité* : Ingénierie logiciel

## Sujet

Partage et Mutualisation de documents

*Mémoire de fin de cycle réalisé par les étudiants :*

- LOTMANI Yacine
- ABDESELAM Oussama

*NOMBRE de jury :*

Mme Ben boudia : président  
Mme EL - GHERS : EXAMINATEUR  
Mme REGUIEG : EXAMINATEUR

*Promoteur* : Mme Oukid Saliha

*Année universitaire* : 2008/2009

MA-004-08-1

# REMERCIEMENTS

*Nous tenons à remercier Dieu le tout puissant de nous avoir donné le courage, la force et la volonté pour la réalisation de ce petit travail.*

*Nous remercions notre promoteur : Mme Oukid Salih, pour son aide et ses conseils qui nous ont permis de réaliser ce petit travail.*

*En fin nous remercions tous les amis qui ont aidés pour réaliser notre travail.*

*Yacine et Oussama*

# SOMMAIRE

Introduction générale.....	1
----------------------------	---

## Chapitre I : L'état de l'art

1. Solution de gestion documentaire .....	2
1.1. Processus rédactionnel .....	2
1.2. Processus éditorial.....	2
1.3. Le périmètre de la GED .....	2
1.4. Les opérations de la GED .....	3
1.5. Infrastructure d'un système de GED.....	6
1.6. Projet GED dans l'organisation .....	6
1.7. Evolution de la GED vers la gestion de contenu .....	6
2. Définitions de base .....	7
2.1. Documents électroniques .....	7
2.2. Cycle de vie d'un document.....	8
2.3. Contenu.....	8
2.4. Domaines fonctionnels.....	9
2.5. Applications intégrées.....	9
3. Etude des différentes solutions.....	10
3.1. Evido.....	10
3.1.1. Besoins .....	10
3.1.2. Architecteur générale .....	11
3.1.2. Fonctionnalités .....	11
3.1.3. Caractéristiques .....	12
3.2. Maarch.....	13
3.3. 3.3.Avantage de maarch par rapport à Evido ou zendep.....	13
4. Contexte de l'application a développé .....	14
4.1. Rôle du système d'information .....	14
4.2. Recherche d'information.....	16
4.3. Annotation des documents pour l'organisation et la structuration.....	17
4.4. Caractéristiques .....	18
5. Conclusion .....	18

## Chapitre II : étude préalable

1. Introduction.....	19
2. Problématique .....	20
3. Principes de solutions .....	21
3.1. Fonctions générales .....	22
3.2. Fonctions liées à la mise en place et maintenance de l'application .....	22
3.3. Fonctions liées à l'alimentation et actualisation .....	23
3.4. Fonctions liées aux traitements des résultats de la recherche .....	23
3.5. Solution technique envisagée .....	23

## Chapitre III : étude Conceptuelle

1. Introduction.....	25
2. Méthodes et outils pour l'application .....	25
2.1. Conception Globale .....	25
2.2. Conception détaillée .....	25
2.2.1. Diagrammes structurels ou statiques.....	26
2.2.2. Diagrammes comportementaux.....	27
2.2.3. Diagrammes d'interaction ou dynamiques .....	27
3. Modélisation dynamique .....	28
3.1. Diagramme des cas d'utilisation.....	28
3.1.1. Les acteurs.....	28
3.1.2. Scénarios possibles.....	31
3.2. Diagrammes de séquence .....	33
3.3. Diagramme d'état.....	39
4. Modélisation Statique .....	41
4.1. Le dictionnaire de données.....	42
4.2. Diagramme de classe.....	44
5. Catégorisation automatique des documents .....	46
6. Gestion des droits d'accès et privilèges .....	48
6.1. Les privilèges.....	49
7. Modélisation de la base de données .....	50
7.1. Définitions .....	50
7.2. Model de données relationnelles .....	51
7.3. Schéma d'une relation.....	51
7.4. Représentation du model relationnelle.....	52

## Chapitre 4 : Réalisation

1. Les échanges de données dans une application web .....	54
1.1. Le protocole http.....	55
1.2. Diagramme de déploiement.....	57
1.3. HTML statique .....	58
1.4. Scripts coté serveur .....	58
2. Gestion des pages .....	59
3. Choix d'une base de données.....	61
3.1. Définition des champs de la base de données .....	61
4. Définir des permissions .....	65
5. Recherche documentaire.....	65
6. Implémentation .....	68
Conclusion générale.....	75
Bibliographie.....	76

## Introduction générale :

Ces dernières années le facteur connaissances est devenu important cependant sa diffusion aux membres d'un organisme pour créer une connaissance collective n'est pas maîtrisée, Des systèmes d'informations sont utilisés en ce sens pour faciliter cette diffusion.

Le flux d'information augmente grâce aux systèmes d'informations de plus en plus performants mais en parallèle la gestion d'information devient de plus en plus difficile en terme d'archivage, stockage, recherche et de partage, ainsi le domaine du management des connaissances s'est donc développé en partie ces dernières années pour tenter de trouver des solutions a ces difficultés a cause aussi de la recherche d'information dans un volume d'information de plus en plus important.

Nous allons ainsi nous concentrer sur une procédure de recherche d'information basée sur une classification des documents, car la classification et la recherche, et leurs interactions sont importantes pour la création et la gestion des connaissances collectives.

## 1. Solution de gestion documentaire [13] :

Souvent une crise ou une tension dans la gestion des documents nous conduit a mettre en œuvre une solution de gestion documentaire afin de mettre en place une organisation plus structurée du travail et une traçabilité autour des documents.

Parmi les problèmes qui peuvent surgir, l'impossibilité de retrouver un document important en version électronique, lorsque sa version papier n'est plus disponible, il y'a donc perte de mémoire au niveau de l'organisation. Alors que les tensions peuvent résulter de difficulté chronique a réutiliser la version électronique d'un document papier ou de temps perdu a des taches de recherches, il s'agit ici de perte de productivité .

La GED ( Gestion Electronique de Document) est l'ensemble des techniques permettant d'organiser gérer et distribuer des informations documentaires sous forme électronique. La production d'un document s'effectue par étapes successives et qui suit des processus de travail.

### 1.1. Processus rédactionnel :

Il concerne le recueil et la validation de l'information, il dépend de la nature de l'information traitée, recherche, sélection et vérification de l'information, indexation...

### 1.2. Processus éditorial :

Il concerne la mise en forme du message et de production industrielle des documents Acquisition des documents, structuration, indexation et archivage, production et diffusion de l'information

### 1.3. Le périmètre de la GED :

La gestion Electronique des documents est un ensemble aux contours imprécis qui peut recouvrir :

- **Des outils** : Pour la capture des fichiers documents, leur stockage, leur circulation, leur mise a disposition, la recherche...
- **Des métiers** : traitement sur les documents, description documentaire, application des procédures qualité, gestion du cycle de vie de l'information, interaction avec de applications métier
- **Des domaines d'application** : documents bureautiques, fichiers multimédias, pages web, contenus structurés, groupe de travail

Les applications de GED pures n'existent quasiment plus, elles intègrent presque toujours des fonctions empruntées aux domaines évoqués ou a des domaines métiers

La gestion documentaire GED est un domaine avant tout organisationnel, alors que le contenu web WCM relève essentiellement de problématiques techniques.

## 1.4. Les opérations de la GED [13] :

Il existe 4 étapes majeures dans la GED :

- L'acquisition des documents
- Le classement des documents.
- Le stockage des documents.
- La diffusion des documents.

### 1.4.1. L'acquisition des documents :

La technique d'acquisition dépend du format du document à acquérir : document papier, fax ou document numérique.

L'acquisition peut comporter plusieurs étapes par exemple l'interprétation ou la conversion de format du document, l'adaptation de sa structure et de sa composition, sa récupération par un dispositif de stockage ou de traitement voire sa compression.

L'acquisition peut s'effectuer par :

### 1.4.2. L'intégration de documents papiers existants :

C'est, pour les documents papiers, la numérisation grâce à des scanners. Lorsque c'est possible et utile, les documents numérisés peuvent alors être triés via une technologie de RAD. Il est possible d'extraire des informations des images numérisées en mettant en œuvre une technologie de LAD. Cette technologie peut utiliser des techniques d'OCR, de reconnaissance de codes à barres et d'ICR comportant par exemple des règles de vérifications linguistiques sur les mots reconnus.

Les documents papiers peuvent ensuite être détruits (soit parce qu'ils n'ont pas d'intérêt, soit parce que la valeur probatoire (ou probante) des copies électroniques est certaine) ou archivés.

### 1.4.3. L'intégration de documents électroniques existants :

Une autre famille de documents est constituée des documents électroniques comme les fichiers bureautiques, les fichiers PDF, ... Enfin, la dernière famille de documents intégrable est constituée des documents COLD qui proviennent de systèmes d'éditique.

### 1.4.4. L'échange de documents électroniques :

Il s'effectue lorsque deux organismes partenaires souhaitent partager des documents électroniques. Ils peuvent alors procéder à une interconnexion de leurs systèmes d'information par un dispositif d'EDI (Échange de Données Informatisé) moyennant le respect d'un même format de données normalisé.

Quel que soit le moyen par lequel le document intègre le système de gestion, il peut souvent passer par une chaîne de validation ou workflow afin d'aboutir à une version finale approuvée par les utilisateurs concernés. Le workflow lié à la validation d'un document est paramétrable



Quel que soit le moyen par lequel le document intègre le système de gestion, il peut souvent passer par une chaîne de validation ou workflow afin d'aboutir à une version finale approuvée par les utilisateurs concernés. Le workflow lié à la validation d'un document est paramétrable et prend en compte les droits d'accès et les profils des utilisateurs du système ; il agit le plus souvent sur le statut, la version et la visibilité du document

#### **1.4.5. La production de documents électroniques :**

Elle peut être le résultat d'une ou de plusieurs procédures managées par différents agents de l'entreprise grâce à un logiciel d'informatique de groupe (groupware ou collecticiel). Le document peut aussi être obtenu par un logiciel d'informatique décisionnelle à partir d'un modèle disponible via l'outil.

Afin de normaliser la structure documentaire les organisations produisent des documents au format XML, il s'agit d'un langage qui permet la publication et l'échange de données car il permet de séparer la description de la structure document (métadonnées) de son contenu.

##### **1.4.5.1. Métadonnées :**

L'indexation est une fonction centrale des outils de la GED, elle consiste en premier lieu à attacher des métadonnées aux documents.

Chaque type de documents peut être qualifié par un ensemble de métadonnées et être associé éventuellement à des règles de gestion particulières s'appuyant sur ces métadonnées

Chaque document sera donc doté de métadonnées propres, ou dans certain cas il est intéressant d'indexer plusieurs fichiers avec un seul jeu de métadonnées.

On appelle parfois notice l'ensemble des métadonnées associées à un document saisies souvent au moyen d'un formulaire web.

#### **1.4.6. L'indexation des documents :**

L'indexation est l'opération qui permet de décrire le document en vue de son exploitation ultérieure. On distingue la description formelle du document (type, auteur, date...) et la description de son contenu

La description s'effectue grâce à un descripteur qui sera associé au document au moment de son stockage.

On distingue deux types d'indexation :

- L'indexation classificatoire qui correspond à une préoccupation de rangement du document
- L'indexation par concept qui vise à décrire le contenu du document qui peut être de plusieurs façons plus ou moins automatisées, on distingue :

La définition de mots clés,

Les descripteurs libres (titre, résumé ou texte).

L'indexation peut se faire a priori : des mots-clefs sont associés au document. Ceux-ci peuvent être librement choisis par l'auteur ou la personne chargée de l'indexation, ou sélectionnés dans un dictionnaire ou un thésaurus (liste de mots liés par des relations de hiérarchie ou d'équivalence). Si on prend le cas des documents HTML, il existe dans ce langage de présentation de pages des balises particulières (les balises META) qui permettent au Concepteur du document de définir les mots-clefs qui lui semblent pertinents.

L'indexation peut être réalisée de manière automatique : on distingue l'indexation statistique qui consiste à une simplification du texte suivi d'un choix de mots clefs, ceux qui apparaissent souvent dans le texte.

L'indexation « full text » qui consiste en une simplification du texte puis une indexation en utilisant un dictionnaire ou un thesaurus

#### 1.4.7. Stockage des documents :

La problématique du stockage est incontournable. Le non prise en compte de cet aspect peut entraîner des situations critiques. Les problématiques sont les suivantes:

- **Le support de stockage** doit être adapté le mieux possible avec le volume des documents. Il doit aussi, en fonction de la fréquence de consultation et de l'importance des données, offrir un faible temps d'accès.
- **L'organisation du stockage** peut être hiérarchisée en fonction du contenu des documents (texte, vidéo, image, etc.) de leurs provenances, états, types, etc.
- **La durée de conservation** doit aussi être considérée afin de permettre une épuration périodique du système, en vue de faciliter le stockage et d'alimenter les archives.

#### 1.4.8. Diffusion des documents :

La diffusion des documents peut se faire via l'internet ou l'intranet. Certains éditeurs de logiciels GED proposent une consultation client léger qui fournit toute ou partie des fonctionnalités de consultation et d'indexation des documents dans un navigateur web.

On peut résumer l'ensemble des besoins amenant à utiliser une **GED** par :

- Partager les documents en réseau, en Intranet ou Internet
- Maîtriser le flux des innombrables documents papiers et électroniques reçus quotidiennement dans les organisations
- Retrouver instantanément un contrat papier, un courrier, un bon de livraison
- Avoir une traçabilité maximale des courriers entrants et sortants
- Augmenter les gains de productivité en réduisant le temps consacré au traitement des processus
- Ne plus avoir de soucis d'archives
- Ne plus dépendre d'autres collaborateurs pour accéder aux documents disponibles
- Supprimer la manutention et le stockage de tous les papiers (facture, BL, AR,...)
- Répondre aux contraintes légales en matière d'archivage

## 2. Définitions de base [13] :

### 2.1. Documents électroniques :

Dans notre approche de la gestion documentaire on prend la définition suivante du terme Document «Ensemble constitué par un support et par l'information qu'il porte ».

Dans le cadre de la GED le support va représenter un fichier qui est créée , modifié et rendu lisible par une application , prenons l'exemple d'un fichier Word qui est rendu lisible par l'application Microsoft Word et de la même manière pour les autres types de documents, d'autant plus que beaucoup des documents manipulés utilisent des formats fermés c'est-à-dire qu'ils ne peuvent être manipulés que par leur application associé de sorte que l'application GED gère finalement des objets qui lui sont le plus souvent opaques impénétrables.

On ne peut pas dire que fichier est un document, par exemple un fichier exécutable n'est pas un document, un document est un fichier porteur d'une information, une information produite par l'homme et pour l'homme.

Un document a trois caractéristiques : sa présentation, sa structure et son contenu

D'autre part on peut dire que un document électronique est toute donnée informatique (autre qu'un programme informatique ou fichier système) qui doit être utilisée sous forme informatique, sans être imprimée (bien que l'impression soit habituellement possible).

À l'inverse du document papier, le document électronique peut permettre de séparer l'aspect présentation (mise en forme, mise en page...) et l'aspect information (contenu, données...), offrant alors la possibilité d'une exploitation séparée

Une notion importante de document objet est introduite dans la GED qui réunit conceptuellement le document fichier et l'ensemble des métas données qui lui sont attachées

## 2.2. Cycle de vie d'un document [13] :

Le cycle de vie d'un document comporte trois étapes successives : élaboration, maintenance (ou conservation), élimination.

### 2.2.1. Élaboration

Le document est créé puis transmis à un ou plusieurs destinataires pour avis. Il est ensuite validé par une personne ayant autorité pour le faire et il ne pourra plus, dès lors, être modifié, si ce n'est avec l'accord de cette personne.

### 2.2.2. Maintenance

Le document validé est transmis à l'entité chargée de le conserver aussi longtemps que le prescrivent les règles et normes en vigueur (externes telles que la réglementation, internes telles que les référentiels d'archivage et les plans de classement).

Cette entité peut prendre la forme d'un système électronique. Pendant toute cette phase, le document est conservé dans des conditions telles que non seulement il est toujours lisible et intelligible, mais qu'il reste aussi crédible et fiable, c'est à dire qu'on ne peut pas le soupçonner d'avoir été altéré ou falsifié.

### 2.2.3. Élimination

L'étape finale peut être la destruction du document, si celui-ci s'avère ne plus être utile et si la loi le permet. Elle peut être aussi un changement de statut, entraînant le maintien en vie du document, mais pour d'autres raisons que celles qui avaient entraîné sa création et par une autre entité que celle qui était chargée de le conserver ; c'est par exemple le cas de tous les documents qui sont conservés uniquement comme témoignage historique.

## 2.3. Contenu :

Le contenu désigne un composant informatif cohérent.

Le terme de contenu s'applique à toute information numérique structurée ou non , un document est donc un contenu.

Un contenu peut être constitué d'un ensemble de contenus élémentaires, incluant des documents fichiers (images, vidéos, documents bureautique...), ainsi que des champs d'information que ce soit textuelle, numérique ou d'autres types, on parle alors de documents structurées.

## 2.4. Domaines fonctionnels :

Nous allons ici situer le domaine de la gestion documentaire vis-à-vis d'un certain nombre de domaines connexes, ces domaines ont des intersections avec la gestion documentaire et peuvent être complémentaires.

### 2.4.1. Gestion de contenu web :

*Objectif* : Permet de prendre en charge la rédaction, la validation et la mise en ligne de contenus a destination de sites web

*Priorité* : Mise en ligne d'information

*Orientation* : Diffusion, restitution de contenus structurés éditoriaux, contribution au second plan

*Termes rencontrés* : WCM- Web Content Management, CMS.

### 2.4.2. Collaboratif :

*Objectif* : Faciliter le travail de groupes de personnes au moyen d'outils dédiés, souvent de communication ou d'interaction en temps réel ou différé

*Priorité* : Faciliter le travail en équipe

*Orientation* : Partage d'information opérationnelle

*Termes rencontrés* : Goupware- travail en équipe (outil ou méthode), chat, messagerie instantanée, possibilité de consulter, Wiki

## 2.5. Applications intégrées :

### 2.5.1 Portail :

Permet de faciliter l'accès a l'ensemble des applications du système d'information

Il existe ainsi des portails spécialisés tels que des « portails documentaires » qui offrent un point central pour accéder a des ressources homogènes dans le cadre de sources d'informations repartis, ainsi les portails offrent des points d'accès unique Ils permettent de représenter des agrégations de contenus et d'applications aussi des points de regroupement d'information thématique.

### 2.5.2 Gestion de contenu :

Permettre la gestion des contenus numériques au travers des étapes de conception, diffusion, utilisation et fin de vie, la gestion de contenu est considérée comme un sur ensemble pour la GED et le WCM, on parle aussi de gestion de contenu d'entreprise (ECM)il inclut des fonctions de gestion de type workflow et travail collaboratif par exemple

### 2.5.3. Bases de documents (Hyperbases) :

Les systèmes hyperbases ce sont des systèmes hypertextes ou hypermédia et qui stockent séparément les données et les liens dans une base de données et l'origine du nom hyperbase naît du mariage de l'hypertexte et des bases de données .

Ils permettent d'offrir a disposition des utilisateurs une collection de documents grâce a une Interface hypertextuelle et aussi offrir des mécanismes de consultation de type hypertexte

Un système Hyperbase est conçu pour stocker et gérer un Hyperdocument qui est un ensemble de documents liés entre eux par différents types d'associations

Une grande partie des systèmes Hyperbase s'appuient sur des systèmes de gestion de bases de données relationnelles

Dans les systèmes hyperbases l'information n'est pas traitée en dehors de l'application les données sont récupérées directement depuis le module de stockage.

## 3. Etude des différentes solutions :

Plusieurs solutions sur le marché et en open source nous permettent la mise en oeuvre d'une solution de partage documentaire. Nous choisissons d'étudier les deux les plus répandues en l'occurrence **EVIDO** ou **ZINDEP, maarch**. Au terme de cette étude, nous procéderons à un choix. Pour ce faire, plusieurs aspects doivent être pris en compte .Ainsi pour cette solution, nous étudierons son architecture, ses caractéristiques et enfin ses fonctionnalités.

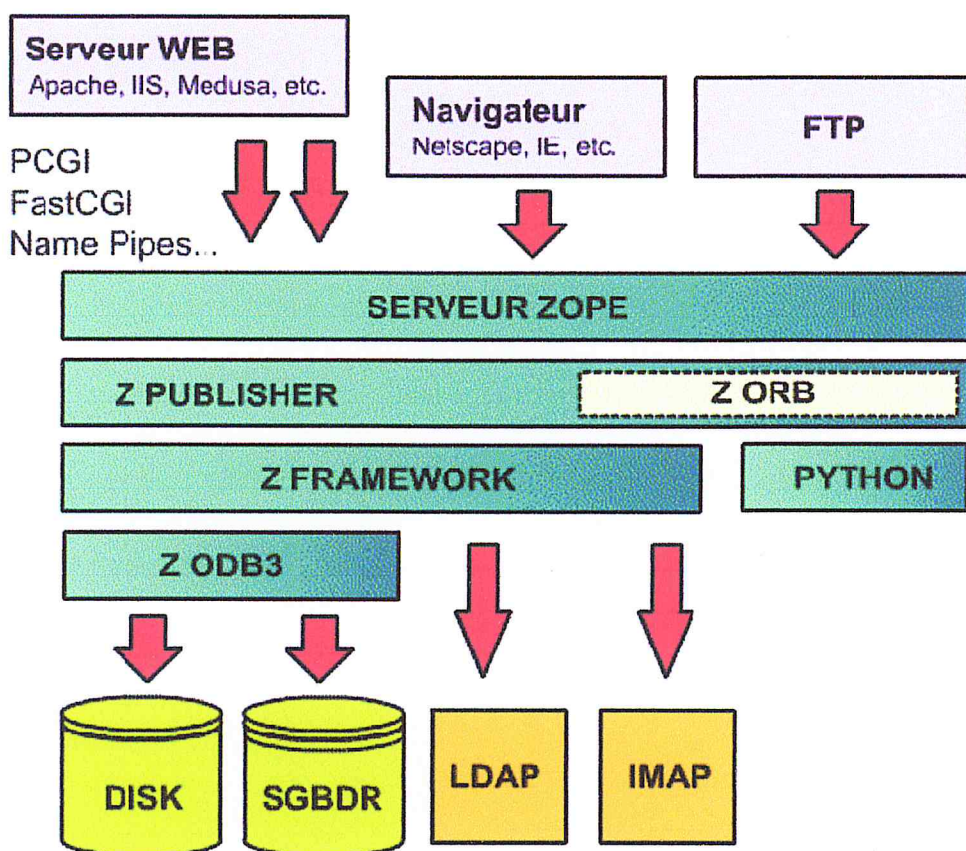
### 3.1. Evido [14] :

Evido a été conçu à partir de Plone pour permettre aux entreprises de détenir une solution de partage documentaire, efficace et éprouvée.

#### 3.1.1. Besoins

- Gérer la documentation de l'entreprise
- Communiquer, partager
- Collaborer

### 3.1.2. Architecture générale :



- Le découpage d'un système informatique en différentes couches logicielles tel que le propose le trio Python/Zope/Plone offre un certain nombre d'avantages, en particulier :

- D'améliorer la maintenabilité du système, en formalisant les contrats d'utilisation entre couches logicielles (notion d'interfaces) sur la base d'une approche « client / serveur » appliquée à chaque étage.

Chaque couche ne connaît que celle qui est en dessous d'elle et ne connaît pas celles qui sont au dessus.

- De faciliter l'organisation du développement, en offrant la possibilité d'affecter les ressources prioritairement par compétences techniques et non par périmètre fonctionnel.

- Zope est donc un serveur d'application développé à partir du langage Python. Avec sa surcouche fonctionnelle Plone puis Evidio, nous offrons une solution capable de s'intégrer dans de nombreux systèmes d'informations, de réduire les coûts de mise en oeuvre et d'apporter une réponse adaptée aux besoins clients.

### 3.1.2. Fonctionnalités

Evidio couvre les fonctionnalités liées à la problématique de la gestion documentaire. Elle les dépasse en offrant une solution Intranet complète dans laquelle la GED vient naturellement s'intégrer. Enfin, Evidio est une solution tournée vers l'avenir grâce à sa modularité et sa

- **Créer, stocker, modifier** tout document, indexer le contenu de tout document standard type « bureautique », avec des fonctions de recherche simples ou composées. Ces actions sur les documents sont contrôlées, et seuls les profils disposant des droits nécessaires pourront les effectuer.
- **Accès aux documents contrôlé** : en fonction de l'état du document dans le processus de validation et du profil de l'utilisateur. Par exemple, en « état privé », il est possible de décider que le document n'est visible/modifiable que par les contributeurs ou l'administrateur... Ces réglages sont paramétrables à souhait, de façon générale (les règles du dossier valent pour les sous-dossiers et leur contenu), ou spécifique pour chaque élément.
- **Gestion dynamique des critères de recherche** : le choix des critères est paramétrable par base documentaire.
- **Centralisation des données** : ou vue, sur critères et filtres permettant ainsi créer vos vues auto alimentées selon vos critères, et cela afin de garder une vision dynamique et synthétique des informations issues de votre base documentaire.
- **Fonctions d'alerte** : Evido offre la possibilité d'être averti de la mise à jour d'un document ou d'une page. Le système se base sur le principe d'une newsletter. Un utilisateur s'abonne à un document ou à un dossier pour être averti de sa modification. Selon la fréquence établie par vos soins, cet utilisateur sera averti par mail si le document qu'il a choisi a été modifié, ou si le dossier auquel il est abonné contient un document récemment modifié. Par ailleurs, une gestion avancée de l'abonnement est disponible via des mots-clés: un utilisateur peut s'abonner à un dossier et ne recevoir que les éléments récemment modifiés et liés aux mots-clés de son choix.
- **Lier des documents entre eux** : Evido permet de mettre en place des liens transversaux dans le site entre les différents types de contenu : vous liez une image à un document, un fichier à un lien, etc. Une boîte d'affichage apparaît en offrant des liens directs vers les contenus liés. Les liens sont bijectifs : la définition dans un sens implique nécessairement la création dans l'autre.
- **Interfaces** : notre solution est autonome et ne dépend d'aucune solution propriétaire (sauf lecteur pour les fichiers stockés).

### 3.1.3. Caractéristiques :

#### -Recommandations logicielles :

Cette solution supporte indifféremment les systèmes d'exploitation Windows ou Linux/Unix excepté la conversion de documents qui ne fonctionne que sous Linux/Unix.

#### - Sécurité / Authentification / Organisation :

Evido permet d'intégrer différents supports d'identification : LDAP, Active Directory, Natif  
Selon l'expression de votre besoin, nous implémenterons la solution idéale pour l'identification sur votre plate-forme.



### 3.2. Maarch [15] :

Maarch est un ensemble cohérent d'outils et d'applications pour mieux gérer les flux documentaires et leur archivage.

Maarch est disponible en Open Source, parce qu'**archivage et conservation nécessitent des solutions ouvertes !**.

La gamme est aujourd'hui composée de deux environnements applicatifs totalement opérationnels :

- Maarch LetterBox pour la Gestion du Courrier, principalement destiné aux Collectivités
- Maarch Entreprise, pour la gestion globale des documents entrant et sortant d'une organisation.

Maarch offre les fonctions d'importation et de restitution de documents afin de permettre la conservation et l'exploitation des **ressources numériques** d'une organisation sur le long terme, en contribuant au respect des régulations françaises et internationales concernant **l'archivage documentaire**

#### 3.2.1. Architecteur :

- Maarch Framework est optimisé pour les moteurs SQL **MySQL 5, PostgreSQL 8.1 à 8.4, SQLServer 2005, Oracle 10,11g**.
- Maarch est écrit principalement en **PHP 5.2**. L'interface est optimisée pour FireFox 3.0 à 3.5 et Internet Explorer 7 (IE6 et IE7 pour Maarch LetterBox), afin de correspondre à la grande majorité des installations, avec un comportement **Web 2.0**.

#### 3.2.2. Maarch Entreprise, la première GED métier globale et Open Source

Maarch LetterBox est toujours la seule Gestion de Courrier entièrement libre, sans version propriétaire, et prête à l'emploi.

La communauté Maarch va désormais encore plus loin, et propose pour la première fois une solution de niveau Entreprise complète :

- Gestion du courrier entrant, sortant, interne, autre
- Circuits de validation, traitement des réponses
- Prise en compte de l'organigramme hiérarchique dans la gestion des autorisations et le workflow
- Gestion multi-dossier, avec plan de classement propre à chaque type de dossier
- Gestion des boîtes, conteneurs, et emplacements de stockage physique
- Archivage à valeur probante

### 3.3. Avantages de maarch par rapport à Evido ou zendep :

1. Maarch est un logiciel open source c'est-à-dire gratuit, par contre Evido qui est un logiciel commercialisé
2. Maarch est un GED global c'est-à-dire il fait la gestion des documents (Maarch Entreprise) et la gestion de courrier (Maarch LetterBox).

## 4. Contexte de l'application à développer :

### 4.1. Rôle du système d'information [2] :

Une information est une transmission d'une collection de données organisées et souvent structurées. La collection peut se faire à partir d'une sélection des données les plus utiles. La connaissance est considérée comme une interprétation subjective d'une information, ainsi elle est toujours liée à un humain. Elle a donc besoin d'un récepteur.

« Le management des connaissances » devrait prendre en compte ainsi que les trois facteurs : l'homme, la technologie d'information et l'organisation », or aujourd'hui la technologie relative à la manipulation d'information est devenue de plus en plus performante et ne présente plus de problèmes techniques majeures par contre il est difficile d'adapter le système aux besoins humains.

Dans ce contexte nous définissons la connaissance comme l'état d'un agent qui suit l'acquisition et le traitement de l'information.

Dans la notion de connaissance, nous distinguons la connaissance explicite (comme des documents, mails...) de la connaissance tacite (les compétences, l'intuition, les secrets de métiers...).

La création et la capitalisation des connaissances font partie de l'externalisation de la connaissance tacite en formalisant la connaissance et produisant des documents explicites.

La recherche d'information fait partie de l'internalisation de connaissance classifiée en cherchant une information et l'absorbant et en changeant ainsi l'état interne de l'agent

La connaissance tacite est souvent personnelle et individuelle. L'externalisation en créant des documents aide à diffuser la connaissance, pour cela il est indispensable d'avoir une base de documents qui puisse être consultée à tout moment.

Parmi les connaissances de l'entreprise on peut faire également une distinction entre des connaissances individuelles et connaissances collectives.

Les connaissances individuelles sont plutôt des connaissances tacites liées directement aux personnes, ces connaissances explicites ou non, peuvent être partagées avec d'autres personnes ou avec des groupes de personnes.

La notion de partage étant essentielle, la connaissance collective est pour une grande part constituée de connaissances explicites.

Ce mécanisme de partage est une transformation de la connaissance individuelle en des connaissances collectives.

La constitution d'une connaissance collective se fait alors par une série d'interactions entre le tacite et l'explicite et entre différentes connaissances individuelles.

Les connaissances collectives sont appelées par certains auteurs connaissances organisationnelles.

L'objectif est de garantir un accès permanent aux connaissances collectives explicites par chaque individu.

Pour cela il faut expliciter certaines connaissances individuelles et offrir un accès efficace pour consulter l'information cherchée pour optimiser le partage d'information et mieux générer et gérer cette connaissance collective par cet accès facilité.

Il est donc important d'avoir une base de connaissances explicites qui permettent de diffuser la connaissance dans l'organisme.

Notre travail consiste à modéliser un système qui permet la diffusion et le partage des connaissances explicites sous forme de documents électronique.

il s'agit de concevoir et de développer un système de « Partage et de mutualisation de données » à travers le web , ce système devra être accessible à tout membre d'un groupe de personnes qui décident de mettre en commun des documents (au sens large) dont chacun dispose et qu'il souhaite partager avec les autres.

Les documents en question peuvent être soit des ouvrages soit des documents numérisés (articles, rapport, thèses, mémoire, images,...). Ils ont des statuts d'accès ou diffusion différents : personnel, Internet (tous), groupe..

Une interface de consultation proposera à l'utilisateur selon son statut un accès à la base des documents selon divers critères.

Voici un exemple de présentation des informations par thèmes :

Propriétaire	Description	Accès	Format document	Info-Notice
				<u>Référence</u>

Référence : permet d'afficher les références du document avec son résumé

Doc, Pdf : donne l'accès au document dans le format spécifié.

## 4.2 Recherche d'information [11] :

Ce domaine de la Recherche d'Information (RI) est confronté à un très grand volume D'informations. Ces « Pull-systèmes » ne peuvent certifier que les informations trouvées Soient les « meilleures ». D'où la notion d'efficacité d'un module de recherche qui est évaluée.

Par la pertinence d'un résultat à une demande. Cette demande est la verbalisation d'un besoin. Malheureusement, les systèmes actuels soutiennent rarement cette verbalisation car ils ne comprennent pas. C'est la raison pour laquelle la recherche ne se fait qu'au mot exact. Cette méthode a pour inconvénient le fait que le résultat ne prenne pas en compte le contexte de la demande.

Dernièrement, l'utilisation de sémantique (description de la nature et du contenu de document) s'est développée dans le domaine de la Recherche d'Information.

Les métadonnées sont définies comme des données sur des données et sont ainsi considérées comme des données descriptives. Elles indiquent la sémantique des documents en décrivant son contenu et son contexte. En utilisant ces métadonnées pour des documents, un module de recherche d'information peut mieux soutenir la verbalisation du besoin en gardant partiellement son contexte et en précisant mieux la demande.

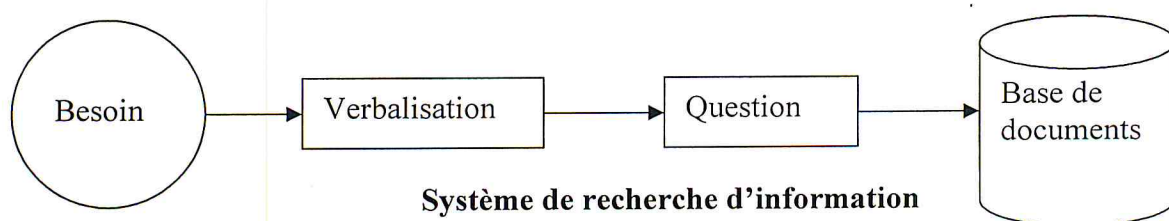
Le système de recherche d'information est défini comme un ensemble de programmes qui interprètent les questions, recherchent les informations dans des fichiers et retournent les informations trouvées à la personne qui a posé la question.

Nous considérons que le système informatique n'est plus le centre du système.

Il s'agit plutôt d'analyser toute la chaîne de ce système ; du besoin au résultat. En commençant par Le besoin d'utilisateur, il faut soutenir la verbalisation de ce besoin afin d'adapter un système de recherche à la demande à retrouver les informations cherchées.

Nous retenons que la verbalisation est la principale difficulté de ce système au bout du processus nous avons un utilisateur avec un besoin d'information.

Cette demande est verbalisée, et la question que l'utilisateur va effectivement poser au système est la formalisation de cette verbalisation suivant les règles du système (question booléenne, Vocabulaire contrôlé, question en langage naturel).



Il est ainsi difficile de développer un système qui réponde aux besoins en livrant les informations recherchées. L'utilisation de métadonnées offrent aujourd'hui une solution partielle à ce dilemme. En effet, elles permettent de décrire le contenu d'un document en choisissant la nature de la description de son contenu. Les méta-données sont toujours des associations de nom- valeur qui permettent de spécifier le contexte, le contenu et leur nature. Aujourd'hui le méta-langage XML s'est progressivement établi comme un standard pour définir les métadonnées des documents.

#### 4.3. Annotation des documents pour l'organisation et la structuration :

Cependant, il est difficile d'harmoniser la structuration individuelle de chaque membre d'une organisation, ainsi l'harmonisation des structures est délicate.

Pour créer une base de connaissance contenant des documents, il nous semble judicieux d'abandonner la hiérarchie classique. Les documents ne sont plus rattachés aux feuilles d'une arborescence, mais ils sont annotés par des attributs appartenant aux catégories qui peuvent être utilisées pour la création d'une hiérarchie, autrement dit : les valeurs d'une hiérarchie sont rattachées aux documents.

Dans notre cas, nous essayons de combiner des connaissances différemment structurées pour créer une base de document équivalent à une connaissance collective.

Cette nécessité est déjà connue depuis des années et la possibilité d'annotation des documents se développe continuellement. Du fait de l'importance de l'Internet, les annotations se concentrent souvent sur des pages Web dynamiques d'une manière automatique et essaient de combiner les connaissances de ces pages Web et avec celles des bases de Données internes dans une organisation pour détecter la connaissance.

Bien que les utilisateurs travaillent dans un contexte similaire, leur façon de stocker des informations peut être différente. Ainsi le besoin d'utilisateur peut se résumer à :

- avoir un accès aux informations le plus vite possible
- retrouver toutes les informations initialement stockées concernant un sujet

La structure de stockage est donc variable en fonction de chaque utilisateur. Elle peut être changée pour l'adapter à l'objectif de l'utilisation des informations :

- Pour la création d'un document (rapport, présentation, etc.)
- Pour la vérification d'une information, (s'informer globalement)
- Pour l'échange et la discussion des informations avec des collègues

#### 4.4. Caractéristiques

##### **4.4.1. Centralisation des données, ou vue, sur critères et filtres**

Permettant ainsi créer vos vues auto alimentées selon vos critères, et cela afin de garder une vision dynamique et synthétique des informations issues de la base documentaire.

##### **4.4.2. Compatibilité tout poste :**

Tout poste disposant d'une connexion réseau avec le serveur hébergeant la solution et d'un navigateur Internet standard (IE 5.5, Standard MAC et Linux) peut être utilisé pour accéder à la base documentaire.

##### **4.4.3. Attribution des rôles (profils) :**

Elle s'effectuera après l'identification par la saisie du login et mot de passe

##### **4.4.4. Logiciel spécifique à installer :**

Sur vos machines clients, excepté les lecteurs de fichiers pour des formats non standards (Visio 32, PDF par exemple).

##### **4.4.5. Nombre d'utilisateurs illimités :**

Vous pouvez utiliser L'application sur autant de postes que vous voulez.

#### 5- Conclusion :

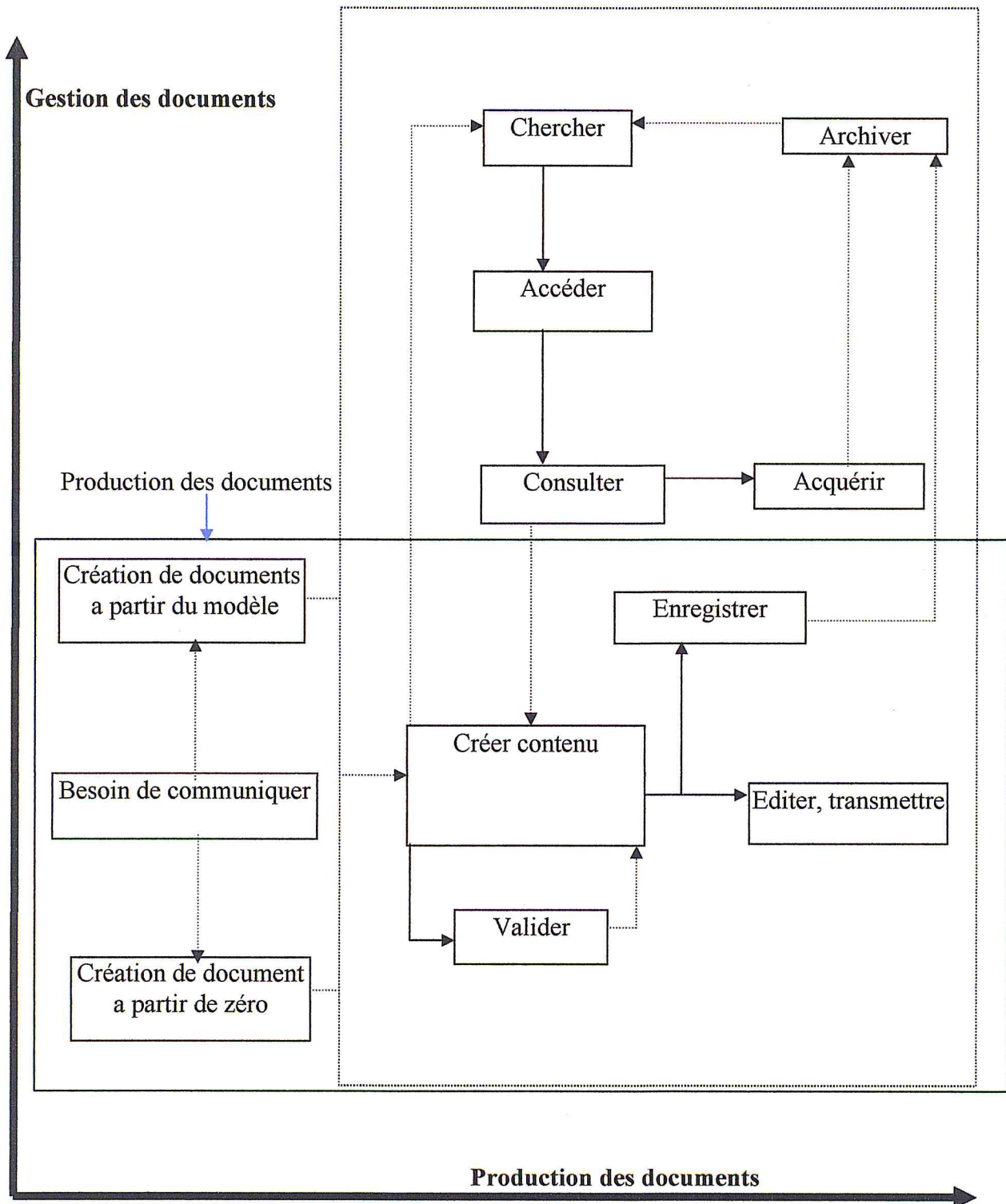
La gestion électronique des documents GED est un domaine très vaste et à travers ce chapitre nous avons donné un aperçu de manière globale sur le système envisagé afin de se situer par rapport aux notions déjà définies précédemment

Le chapitre suivant va nous permettre de faire une étude préalable afin d'illustrer le contexte du système que ce soit méthodologique ou technique.

## Partie 2: Étude préalable

## 1. Introduction :

Le cycle de vie du document relève de plusieurs processus parmi lesquels il existe deux processus distincts comportant chacun des objectifs spécifiques. Il s'agit de la production (axe horizontal) et de la gestion des documents (axe vertical). Cette dernière est le centre de notre réflexion au cours de notre travail, La figure suivante illustre l'interrelation entre ces deux axes.





## 2. Problématique:

Ces dernières années, la gestion documentaire est devenue de plus en plus importante.

Cependant, la diffusion des documents aux individus d'un organisme Pour créer une connaissance collective n'est pas maîtrisée. Pour faciliter cette diffusion, des Systèmes d'information sont utilisés en ce sens. De fait, le flux d'informations augmente Grâce aux systèmes d'information de plus en plus performants en terme de volume et rapidité.

Mais en contre-partie la gestion des documents devient de plus en plus difficile : en terme D'archivage, stockage, recherche et partage de ces informations avec une appropriation de Plus en plus difficiles de ces informations par les protagonistes.

Au sein des organismes de recherche, chaque chercheur a sa propre approche pour gérer les informations et notamment pour les stocker et y accéder selon des points de vue différents. Le besoin d'un meilleur partage d'informations s'exprime par le souhait d'avoir un module de recherche qui permette d'accéder aux informations selon des points de vue multiples. Cet accès est basé sur différents attributs de classement de documents, Dans notre contexte, une information peut être un document, une partie de document ou une annotation.

La conception se base sur l'analyse des besoins des chercheurs et de leur classement spécifique de documents, afin de mieux répondre à la demande. Il s'agit alors de proposer des spécifications fonctionnelles ainsi que des interfaces possibles pour supporter les activités des chercheurs.

Les documents en question appartiennent a des utilisateurs et qui veulent s'organiser en groupes de d'enseignants et qui décident de mettre en commun leurs documents, la diffusion de ces derniers doit tenir en compte les privilèges d'accès et le statut de chaque utilisateur.

La sécurité d'une application peut être compromise de multiples manières. Les applications web sont plus particulièrement vulnérables du fait de leur architecture distribuée et leur architecture tiers qui multiplient les composants autonomes représentant autant de maillons d'une chaîne de sécurité à rompre.

La compromission d'une application peut engendrer la compromission en cascade des sources de données auxquelles elle a accès. Pour réduire le risque de compromissions en chaîne, il est nécessaire de se doter d'une politique de gestion des privilèges drastique.

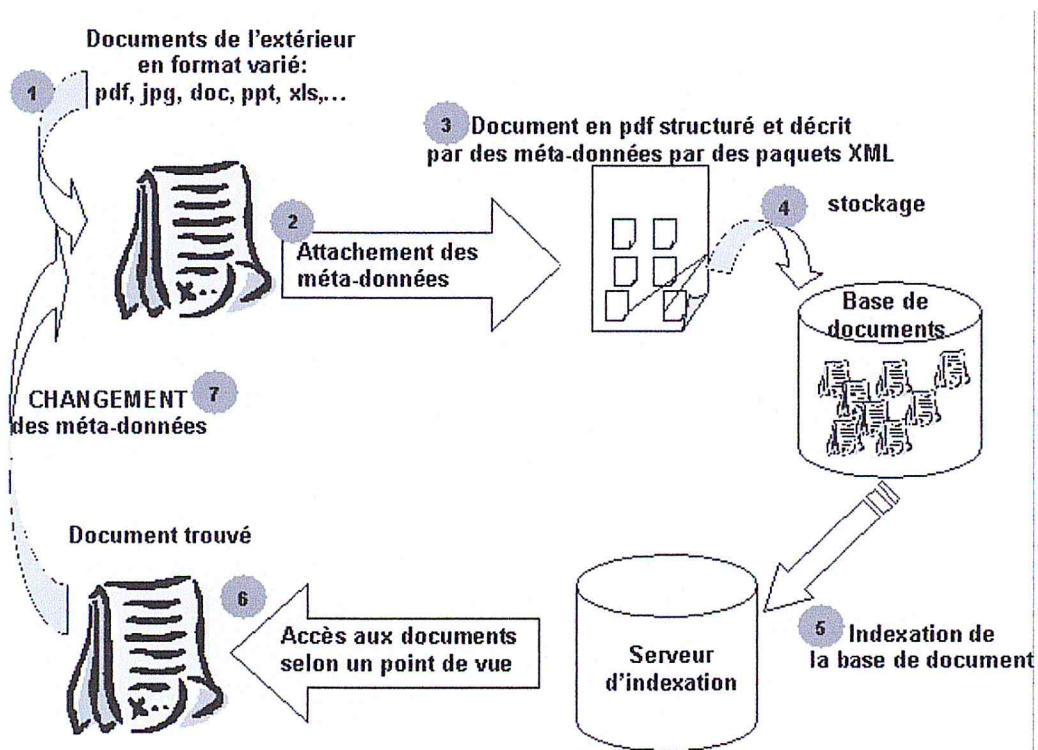
### 3. Les principes de solutions :

Pour l'architecture du système nous utilisons ces quatre typologies qui utilisé pour

- un module de structuration et annotation des documents
- un module de recherche d'information et de visualisation de document

Les modules et les typologies nous aident à créer un répertoire partagé pour des documents annotés. Ce fait sera utile pour diffuser et partager des informations entre les membres avec intérêts similaires.

Nous utilisons l'architecture illustrée sur la figure suivante pour réaliser le classement et la Recherche d'informations :



Présentation des fonctions attendues :

### 3.1. Fonctions générales :

Donner une information pertinente, avec un périmètre circonscrit et un Vocabulaire ciblé.

Doit être utilisable, consultable et interrogeable facilement: le système Permet à l'utilisateur d'identifier, dans un vaste dispositif de stockage, des Informations pertinentes à partir de la requête formulée.

Donner une information fiable : le système contient des informations Systématiquement mises à jour et validées, de sorte que les utilisateurs les Exploitent sans aucune hésitation sur leur validité. Permettre un accès immédiat aux informations : le système présente une Interface et une utilisation simples pour l'usager, qui permettent d'accélérer les Délais d'accès au répertoire, les délais d'apprentissage, les délais d'accès à L'information pertinente et les délais de mise à jour.

### 3.2. Fonctions liées à la mise en place et maintenance de l'application :

Il s'agit ici, de mettre en place un système d'information en ligne, disponible via Internet et son protocole TCP-IP, basé sur une architecture clients/serveurs. Il faut Pour cela :

Faciliter la mise en place et la maintenance de l'application (l'administration De l'application se fera en ligne, via un système tel que PhpMyAdmin, et un Logiciel FTP).

L'interopérabilité et la compatibilité techniques de l'application sont Obligatoires. Que cela soit avec le serveur

Qu'avec la plupart des systèmes existants (systèmes d'exploitation et navigateurs Internet).

L'autonomie technique vis-à-vis des solutions logicielles utilisées pour réaliser L'application est également souhaitée. Pour cadrer avec la philosophie générale Du projet, les outils techniques utilisés feront une large place à la philosophie Du logiciel libre et ouvert. Il doit être évolutif, il est conçu afin d'être adaptable Et permettre aisément l'ajout et la modification de contenus.

Doit être pérenne, reproduction des données avant la mort physique du support.

Prise en compte de l'obsolescence des formats et des environnements Informatiques et de l'évolution rapide des matériaux.

Faciliter et limiter les interventions de l'administrateur, dans la gestion des Droits, la définition des profils d'usagers, des options restrictives, la Modification de la structure des données.

Créer une application consultable en libre accès pour tous, tout en créant un Système d'identification simple et efficace permettant à un utilisateur enregistré D'alimenter la base.

Permettre le travail en réseau, multi-utilisateur. Le serveur doit accepter l'accès Distant (autorise l'utilisateur à IP non connue, compatibilité avec 95% des Navigateurs...).

### 3.3. Fonctions liées à l'alimentation et actualisation :

Faciliter l'alimentation et la modification des données : l'ajout d'un Enregistrement (une fiche de formation), l'actualisation, la suppression de Données sur cette même fiche.

Mettre en place un système d'aide et de contrôle à la saisie : offrir la possibilité De consulter des listes de valeurs, index...

L'utilisateur ayant les droits peut alimenter, modifier des données, ajouter des Informations sur les enregistrements (tables, champs) permis par son profil.

Permettre à l'utilisateur de s'identifier facilement, via un formulaire accessible Sur chaque page du site.

### 3.4. Fonctions liées aux traitements des résultats de la recherche :

Offrir un affichage des résultats présentés selon les champs les plus à même de Vérifier leurs pertinences auprès des utilisateurs.

Offrir la possibilité d'effectuer des tris sur les résultats de la recherche.

### 3.5. Solution technique envisagée :

Plusieurs optiques existent pour constituer un répertoire. On peut envisager la solution Web statique, c'est-à-dire, réaliser un site simplement en code HTML dans lequel la navigation et les accès seront simplement hypertextuels, au final, cela peut ressembler à un annuaire. L'avantage est la simplicité et le faible coût de réalisation d'une telle application. Le problème vient de la mise à jour et de la maintenance qui reste très difficile à réaliser, il n'y a, de plus, aucune possibilité d'effectuer directement un traitement statistique sur les données.

Toutefois plusieurs éléments nous ont menés pour une seconde solution. D'une part le nombre important des champs et des données, d'autre part la multiplicité des relations qui existent entre notre entité principale «le document ». La nécessité de mettre en place un langage documentaire propre à l'application avec la création de listes de valeurs et la nécessité d'archiver les données afin de pouvoir noter les évolutions en matière de formation à l'information. Le besoin que toutes les tâches d'indexation et d'actualisation soient simples, la volonté de multiplier les clés d'accès sur un document. Auxquels s'ajoute la nécessité de mettre en place un système d'identification avec identifiant (login) et mot de passe.

La solution idéale semble d'opter pour un site dynamique qui s'appuie sur un Système de Gestion de Base de Données (SGBD). Le modèle qui semble s'imposer du fait de sa simplicité, sa popularité et son adéquation avec les objectifs du projet est le modèle relationnel. L'utilisation d'un Système de Gestion de Base de Données Relationnelles devient donc implicite.

Ce type de système comporte plusieurs avantages : tout d'abord un haut degré d'indépendance entre les applications (programmes, interfaces) et la représentation interne des données (fichiers, chemins d'accès), une facilité d'accès aux données.

La représentation logique permettant d'établir une base solide pour traiter les problèmes de cohérence et de redondance des données. Il garantit la simplicité pour l'utilisateur dans la conception, la définition, l'installation de la base de données. On peut, également mettre en évidence la simplicité de la structure des données et la simplicité des opérateurs grâce à l'utilisation d'un langage standard pour effectuer les requêtes (SQL).

Il semble évident et aucunement à démontrer que le trio qui s'impose dans ce domaine est l'ensemble Apache (serveur) MySQL (SGBD) et PHP (langage).

MySQL est en effet le serveur de base de données le plus utilisé dans le monde.

Son architecture logicielle le rend extrêmement rapide et facile à personnaliser.

Les principaux avantages de MySQL sont sa rapidité, sa robustesse et sa facilité d'utilisation et d'administration.

Un autre avantage majeur de MySQL est sa documentation très complète et bien construite. De plus, son coût est quasiment nul (ne nécessite pas l'achat de licence).

MySQL s'utilise en complément d'un langage dynamique permettant d'appeler les requêtes, ce dernier se nomme PHP et ses atouts sont nombreux (rapidité, compatibilité, gratuité, simplicité, documentation accessible en français..).

Au final, on a préconisé cette seconde solution qui permet d'aller au-delà de la simple transposition de textes écrits. En permettant la construction de pages et de formulaires dynamiques, ce dernier garantira une souplesse d'utilisation dans la saisie des données en ligne. Il permettra de développer un outil collaboratif fonctionnant en réseau. De plus, sous cette forme, l'application offrira une meilleure interactivité avec ses utilisateurs et des fonctionnalités plus larges et plus évoluées

# Partie 3: Étude conceptuelle

## 1. Introduction:

L'activité de ce Chapitre est une description logique de la façon dont le système va fonctionner. Elle consiste à façonner le Système et lui donner une forme et une architecture. Elle constitue une entrée majeure pour les deux dernières à savoir l'implémentation et le test.

## 2. Méthodes et outils pour l'application:

En génie logiciel la conception constitue une phase fondamentale dans le cycle de vie d'un logiciel, La réussite de ce dernier dépend beaucoup de cette étape.

Dans notre application on va se baser sur deux conceptions, la conception architecturale et la conception détaillée.

### 2.1. Conception Globale:

Cette conception consiste à scinder les tâches de l'application en différentes petites parties afin de mieux organiser et développer l'application, ça se base sur la technique diviser pour mieux régner Les retombés directs de cette technique ne sont pas négligeables on peut mentionner quelques uns:

- le développement de l'application peut être partagé par plusieurs groupes de travail
- la possibilité de réutiliser les composants dans d'autres applications
- la portabilité de l'application

Dans notre cas on va utiliser entre autre une architecture client/serveur, on va scinder cette dernière en trois parties, une partie de représentation (représentée par les interfaces), une partie qui permet l'accès à la base de données et une dernière partie composée par la base même.

Le partage de l'application en sous système va nous permettre de faire une conception détaillée de chaque partie.

### 2.2. Conception détaillée:

Merise et UML sont deux grands principes de traduction ou modélisation d'un système d'information. Néanmoins, ils ne sont pas aussi proches qu'on pourrait le penser.

Le choix de l'un ou de l'autre se fait selon trois axes à savoir l'accessibilité, la précision et l'exploitabilité.

Pour le premier axe (accessibilité) MERISE présente l'intérêt d'avoir des modèles logiques moins détaillés facilement compréhensibles par un utilisateur moins avisé.

Tandis qu'UML conçu pour s'adapter à n'importe quel langage de programmation orientée objet présente plusieurs modèles (diagrammes) dont leurs compréhensions nécessitent une grande attention.

En ce qui concerne le deuxième critère (précision), MERISE est décevant. Malgré sa clarté, il manque une précision du fait qu'elle est éloignée du langage donc difficile à implémenter alors qu'UML intègre les éléments communs des différents langages, sa volonté est d'être fidèle à la réalisation finale. Elle est beaucoup plus complète avec ses différents diagrammes.

Pour en finir avec l'exploitabilité, MERISE est une méthode plus généraliste. Elle donne une vue globale de la solution sans autant rentrer dans les petits détails. Contrairement à UML qui est conçu pour l'implémentation objet avec ses différents détails et sa portabilité (s'adapte à n'importe quelle plateforme) elle est donc plus exploitable.

L'une ou l'autre présente des avantages et des inconvénients. Il est réservé au concepteur de choisir la méthode la mieux adaptée pour son cas. Si on cherche la précision et l'exploitabilité comme dans notre cas UML devance de loin MERISE. Tandis que, si c'est la clarté et l'accessibilité qui sont en question MERISE est préférable.

La conception de notre application mérite bien une grande précision et une exploitabilité maximale. C'est la raison pour laquelle on va retenir UML. Les différences entre les logiciels de modélisation UML sont infimes. N'empêche de mentionner quelques logiciels qui sont à notre connaissance : Agro UML (open source), Poseidon UML et le plus célèbre Rational Rose.

Un parmi les avantages qui nous ont permis de choisir UML comme méthode de modélisation est l'orienté objet. Cette approche influe aussi sur le choix du langage à adopter on peut rajouter quelques uns à savoir la portabilité, la facilité, la multidisciplinarité et pas mal d'autres comme la sécurité.

A plupart des nouveaux langages sont orientés objet. Le passage de la programmation fonctionnelle à l'orienté objet n'était pas facile. L'un des soucis était d'avoir une idée globale en avance de ce qu'on doit programmer.

L'algorithmique qui était utilisé dans la programmation fonctionnelle ne pourrait pas suffire à lui seul. Le besoin d'avoir des méthodes ou langages pour la modélisation des langages orientés objet se faisait sentir en occurrence UML qui nous a permis de faire la conception de notre application.

De nos jours UML2 possède treize diagrammes qui sont classés en deux catégories (dynamique et statique).

Comme on l'a dit UML2 possède treize diagrammes.

Sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.

### 2.2.1. Diagrammes structurels ou statiques [1] :

Les diagrammes structurels ou statiques rassemblent:

- **Diagramme de classe** : il représente les classes intervenant dans le système.
- **Diagramme de composants** : il permet de montrer les composants du système d'un point vue physique Tels qu'ils sont mis en oeuvre (fichiers, bases de données...)
- **Diagramme de déploiement** : il sert à représenter les éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent avec eux.
- **Diagramme de paquetage** : un paquetage étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, le Diagramme de paquetage sert à représenter les dépendances entre paquetages, c'est-à-dire les dépendances entre ensembles de définitions
- **Diagramme de structure composite**: permet de décrire sous forme de boîte blanche les relations



entre composants d'une classe.

### 2.2.2. Diagrammes comportementaux [1] :

Les diagrammes comportementaux rassemblent :

- **Diagramme de cas d'utilisation** : il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.
- **Diagramme de transition** : permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants.
- **Diagramme d'activité** : permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

### 2.2.3. Diagrammes d'interaction ou dynamiques [1] :

Les diagrammes d'interaction ou dynamiques rassemblent rassemblent :

- **Diagramme de séquence** : représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et de ses acteurs.
- **Diagramme de communication**: représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.
- **Diagramme global d'interaction** : permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences.
- **Diagramme de temps**: permet de décrire les variations d'une donnée au cours du temps.

On ne peut pas aller directement à la conception sans faire une description du fonctionnement de l'application.

### 3. Modélisation dynamique:

#### 3.1. Diagramme des cas d'utilisation [1] :

Un cas d'utilisation (CU) est une manière spécifique d'utiliser le système. Elle Représente une suite d'interaction entre un acteur et le système. Il permet de décrire Ce que le futur système devra faire, sans spécifier comment il le fera.

- Correspond à une fonction du système visible par l'acteur
- Permet à un acteur d'atteindre un but
- Doit être utile en soi
- Regroupe un ensemble de scénarii correspondant à un même but

Il n'y a pas une façon unique de repérer les cas d'utilisation. Il faut se placer du point de vue de chaque acteur et déterminez comment il se sert du système, dans quels cas il l'utilise, et à quelles fonctionnalités il doit avoir d'accès. Il faut éviter les redondances et limiter le nombre de cas en se situant au bon niveau d'abstraction

Pour chaque acteur, il convient de :

- Rechercher les différentes intentions métier avec lesquelles il utilise le système,
- Déterminer dans le cahier des charges les services fonctionnels attendus du Système.

Il ne faut pas faire apparaître les détails des cas d'utilisation, mais il faut rester au niveau des grandes fonctions du système.

Il ne doit pas y avoir de notion temporelle dans un diagramme de cas d'utilisation (sera pris en compte dans le diagramme de séquence par exemple).

##### 3.1.1. Les acteurs :

Les acteurs candidats sont systématiquement les utilisateurs humains directs. Pour faciliter la recherche des acteurs, on peut imaginer les frontières du système.

Tout ce qui est à l'extérieur et qui interagit avec le système est un acteur; tout ce qui est à l'intérieur est une fonctionnalité du système que le maître d'œuvre doit réaliser.

Il faut veiller à ne pas oublier les personnes responsables de l'exploitation et de la maintenance du système.

En plus des utilisateurs, les acteurs peuvent être:

- Des périphériques manipulés par le système (imprimantes, robots, ...);
- Des logiciels déjà disponibles à intégrer dans le projet;
- Des systèmes informatiques externes au système mais qui interagissent avec Lui, etc.

Suivant les besoins de notre application on peut présenter ses différents acteurs

Pour cette application, nous avons identifié un nombre d'acteurs et des cas d'utilisations répondants aux besoins de ces acteurs.

La manière d'accéder aux services de l'application pour l'un et pour l'autre est la même. La différence réside sur les droits d'accès et les limites de chacun.

Les besoins selon l'utilisateur du système :

**Etudiant:**

- Rechercher un document
- Consultation d'un document
- Télécharger un document
- Déposer un document

**Enseignant:**

- Déposer un document
- Rechercher un document
- Consultation d'un document
- Télécharger un document

**Administrateur :**

L'administrateur se place au dessus de l'utilisateur il peut faire part les taches de l'utilisateur mais aussi gérer ces derniers, ceci est intéressant car UML présente le critère de l'héritage entre les acteurs, donc on pourra faire l'administrateur un héritier de l'utilisateur et on va lui ajouter la particularité de pouvoir gérer des utilisateurs

En plus des taches précédentes il peut:

- Ajouter un groupe
- Supprimer un groupe
- Supprimer un utilisateur
- Lister les utilisateurs
- Contacter un utilisateur

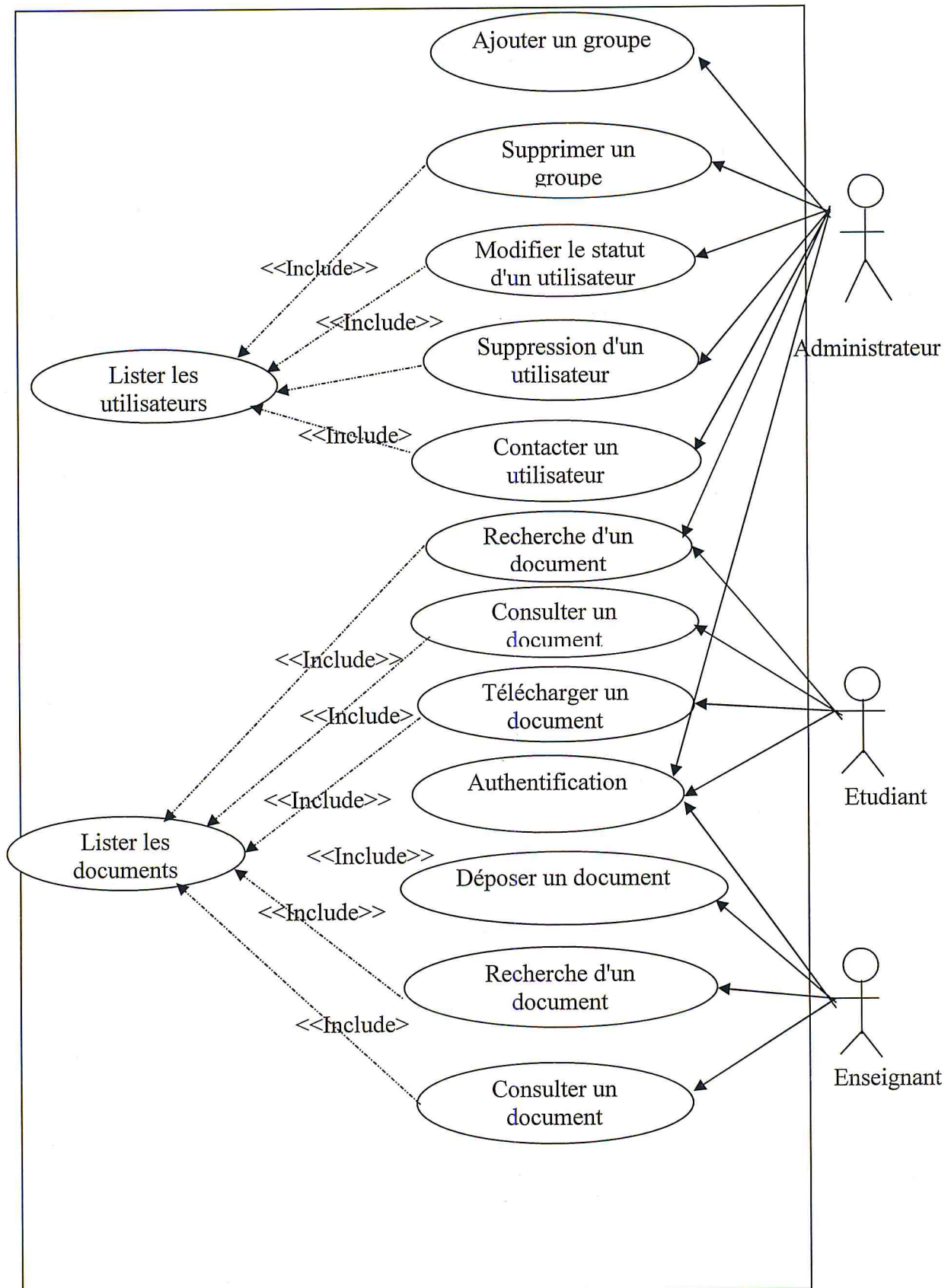


Fig1 : Diagramme de cas d'utilisation global

### 3.1.2. Scénarios possibles :

#### Ajouter un utilisateur:

- 1- l'administrateur se connecte en tant qu'administrateur
- 2- le système affiche les menus propres à l'administrateur
- 3- l'administrateur choisi le menu ajouter un utilisateur
- 4- le système affiche le menu du répertoire utilisateur (Enseignant, Etudiant)
- 5- Après sélection l'administrateur remplit un formulaire d'inscription où il peut affecter un statut à l'utilisateur
- 6- validation de l'inscription
- 8- Le système retourne un message d'erreur si l'utilisateur est déjà inscrit dans la base
- 7- Affichage d'un message de confirmation

#### Modifier un utilisateur:

- 1- l'administrateur se logue en tant qu'administrateur
- 2- Choisir le lien modifier un utilisateur du menu
- 3- Le système affiche les liens (Enseignant, Etudiant)
- 4- Sélection du type d'utilisateur à modifier
- 5- le système affiche la liste des utilisateurs
- 6- Recherche de l'utilisateur à modifier
- 7- Sélection de l'utilisateur à modifier
- 8- Modification de l'utilisateur sélectionné et validation
- 9- Confirmation de la modification

#### Modifier le statut d'un utilisateur:

- 1- L'administrateur s'authentifie
- 2- lister la liste des utilisateurs
- 3- rechercher un utilisateur
- 4- modifier le statut d'un utilisateur

#### Supprimer un utilisateur:

- 1- l'administrateur se logue en tant qu'administrateur
- 2- Choisir le lien supprimer un utilisateur du menu
- 3- Le système affiche les liens (Enseignant, Etudiant)
- 2- Sélection du type d'utilisateur à supprimer
- 3- Le système affiche la liste des utilisateurs
- 4- Recherche de l'utilisateur à supprimer par critères
- 5- Sélection de l'utilisateur à supprimer
- 6- L'administrateur sélectionne le bouton supprimer
- 7- le système affiche un message de confirmation

**Contacteur un utilisateur:**

- 1- l'administrateur se logue en tant qu'administrateur
- 2- Choisir le lien recherche du menu
- 3- L'étudiant saisie les critères de recherche
- 4- Validation étudiant-système pour la recherche
- 5- Le système retourne le résultat sous forme d'un tableau
- 6- L'administrateur sélectionne le lien de l'E-mail de l'utilisateur dans la colonne contact
- 7- Le système ouvre le logiciel de messagerie à utiliser

**Déposer un document:**

- 1- un enseignant se logue et dont le statut est groupe
- 2- Choisir le lien Ajouter un Document dans le menu
- 3- Choisir le nombre de documents à ajouter
- 4- le système retourne un formulaire à remplir
- 5- L'enseignant saisie les coordonnées des documents à déposer
- 6- le système demande la confirmation du dépôt
- 7- l'enseignant confirme
- 8- Le système affiche documents ajoutés

**Modifier le statut d'un document:**

- 1- un enseignant se logue et dont le statut est groupe
- 2- le système affiche la liste des documents déjà déposés par l'enseignant dans un tableau
- 4- l'enseignant sélectionne le nouveau statut à travers une liste déroulante
- 5- L'enseignant confirme le changement du statut
- 6- Le système change le statut du document

**Supprimer un document:**

- 1- L'enseignant se logue
- 2- Le système affiche la liste des documents déjà déposés par l'enseignant dans un tableau
- 3- L'enseignant sélectionne le lien supprimer de la même ligne du document concerné dans le tableau
- 3- Le système demande la confirmation de la suppression
- 4- l'enseignant confirme la suppression
- 5- Le système supprime de document physiquement

**Recherche d'un document:**

- 1- L'étudiant s'authentifie
- 2- L'Étudiant choisie le lien recherche du menu
- 3- L'étudiant saisie les critères de recherche
- 4- Validation étudiant-système pour la recherche
- 5- Le système retourne le résultat sous forme d'un tableau

**Consultation d'un document:**

- 1- L'utilisateur s'authentifie
- 2- L'utilisateur recherche un document
- 3- Le système retourne le résultat sous forme d'un tableau
- 4- l'utilisateur sélectionne le lien qui représente le nom du document
- 5- Le système affiche le contenu du document si le statut de l'utilisateur le lui permet

**Téléchargement d'un document:**

- 1- L'étudiant s'authentifie
- 2- L'étudiant recherche un document
- 3- Le système retourne le résultat sous forme d'un tableau
- 4- Télécharger le document par l'utilisateur après vérification de son statut par le système

**3.2. Diagramme de séquence [1] :**

Le diagramme de séquence est une variante du diagramme de collaboration.

Les diagrammes de séquence possèdent intrinsèquement une dimension temporelle mais ne représente pas explicitement les liens entre les objets.

Ils privilégient ainsi la représentation temporelle à la représentation spatiale et sont plus Aptes à modéliser les aspects dynamiques du système.

En revanche, ils ne rendent pas compte du contexte des objets de manière explicite, Le diagramme de séquence permet de visualiser les messages par une lecture de haut en Bas. L'axe vertical représente le temps, l'axe horizontal les objets qui collaborent.

Une Ligne verticale en pointillé est attachée à chaque objet et représente sa durée de vie.

L'interaction se traduit par l'envoi d'un message entre objets.

Le diagramme de séquence insiste sur la chronologie des objets en utilisant la ligne de vie des objets.

Les diagrammes de séquence permettent de représenter les périodes d'activité des objets.

Une période d'activité correspond au temps pendant lequel un objet effectue une action, Soit directement, soit par l'intermédiaire d'un autre objet qui lui sert de sous-traitant.

Plusieurs types de messages peuvent transiter entre les acteurs et objets.

- *Message simple* : le message n'a pas de spécificité particulière d'envoi et de réception.

- *Message avec durée de vie* : l'expéditeur attend une réponse du récepteur pendant un certain temps et reprend ses activités si aucune réponse n'a lieu dans un délai prévu.

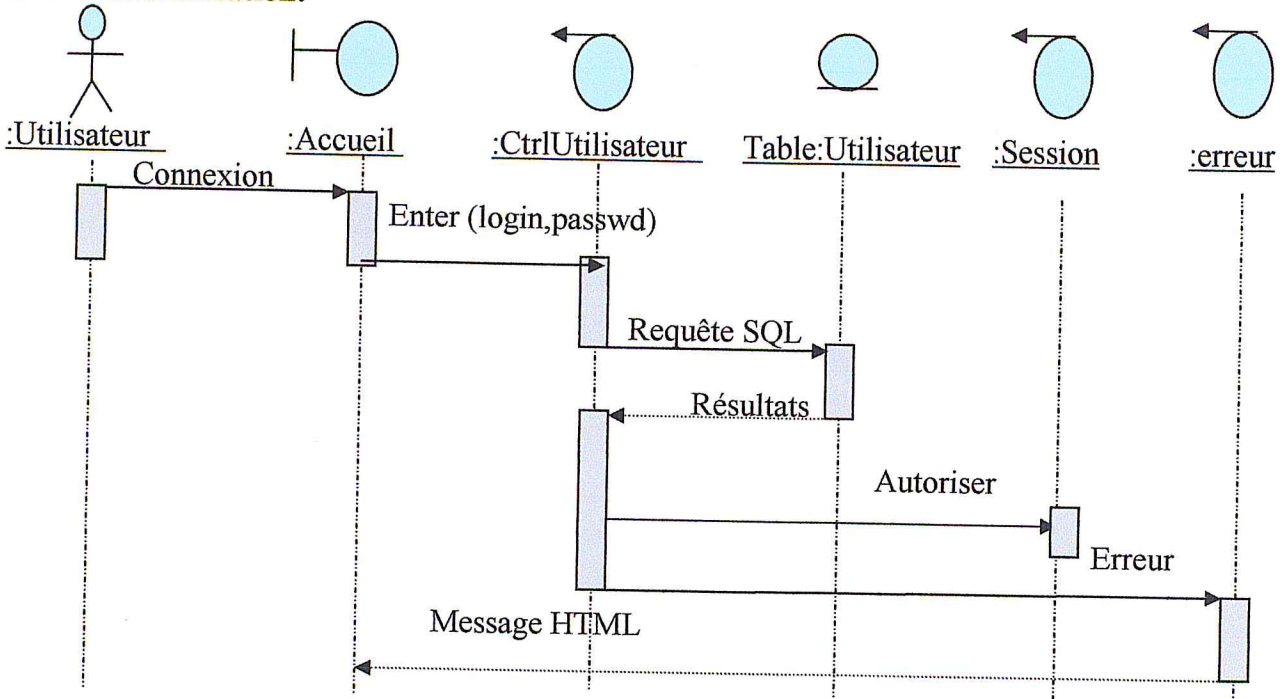
- *Message synchrone* : l'expéditeur est bloqué jusqu'au signal de prise en compte par le destinataire.

Les messages synchrones sont symbolisés par des flèches barrées.

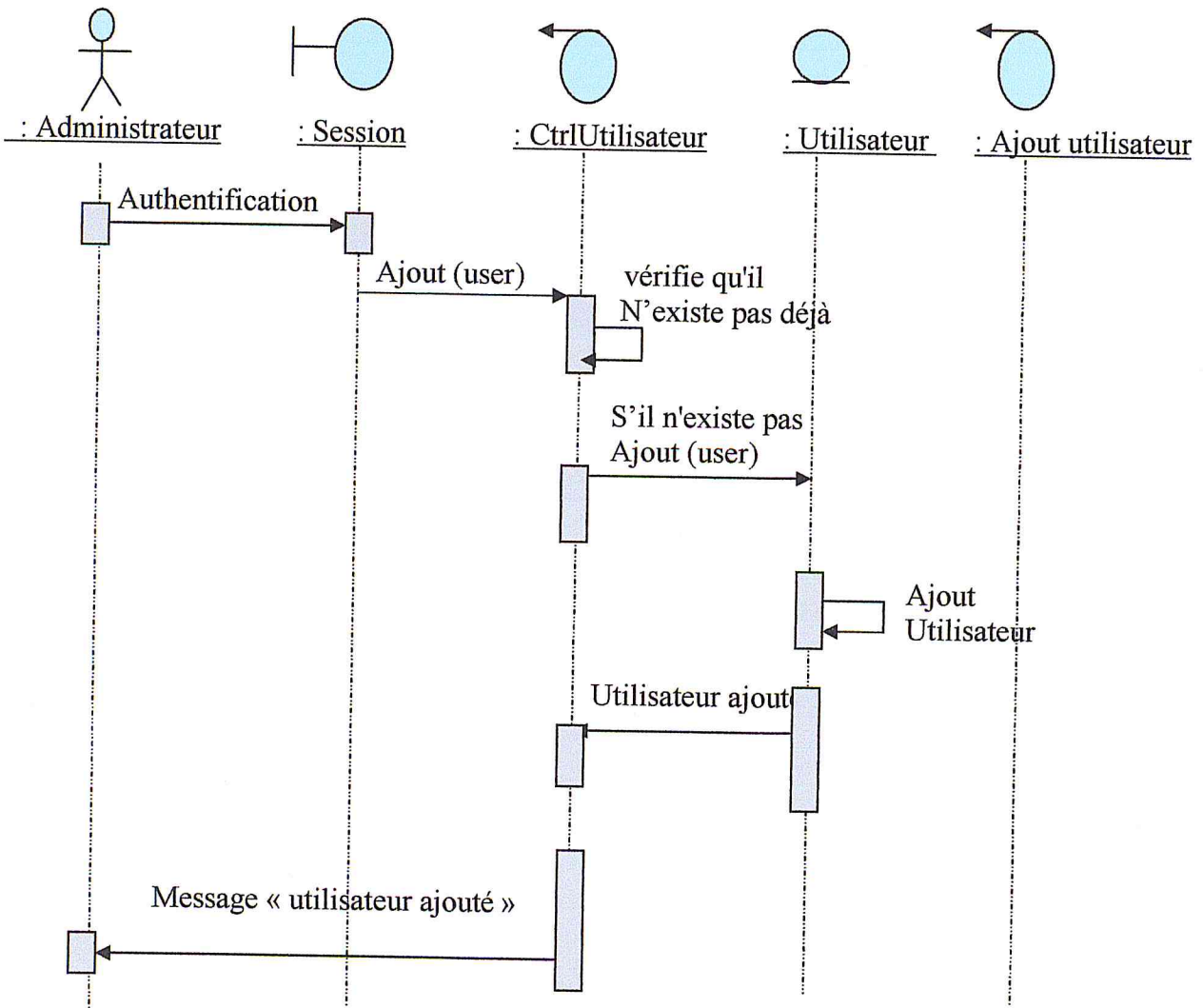
- *Message asynchrone* : le message est envoyé, l'expéditeur continue son activité que le message soit parvenu ou pris en compte ou non. Les messages asynchrones sont symbolisés par des demi-flèches.

- *Message dérobant* : le message est mis en attente dans une liste d'attente de traitement chez le récepteur.

3.2.1. Authentification:

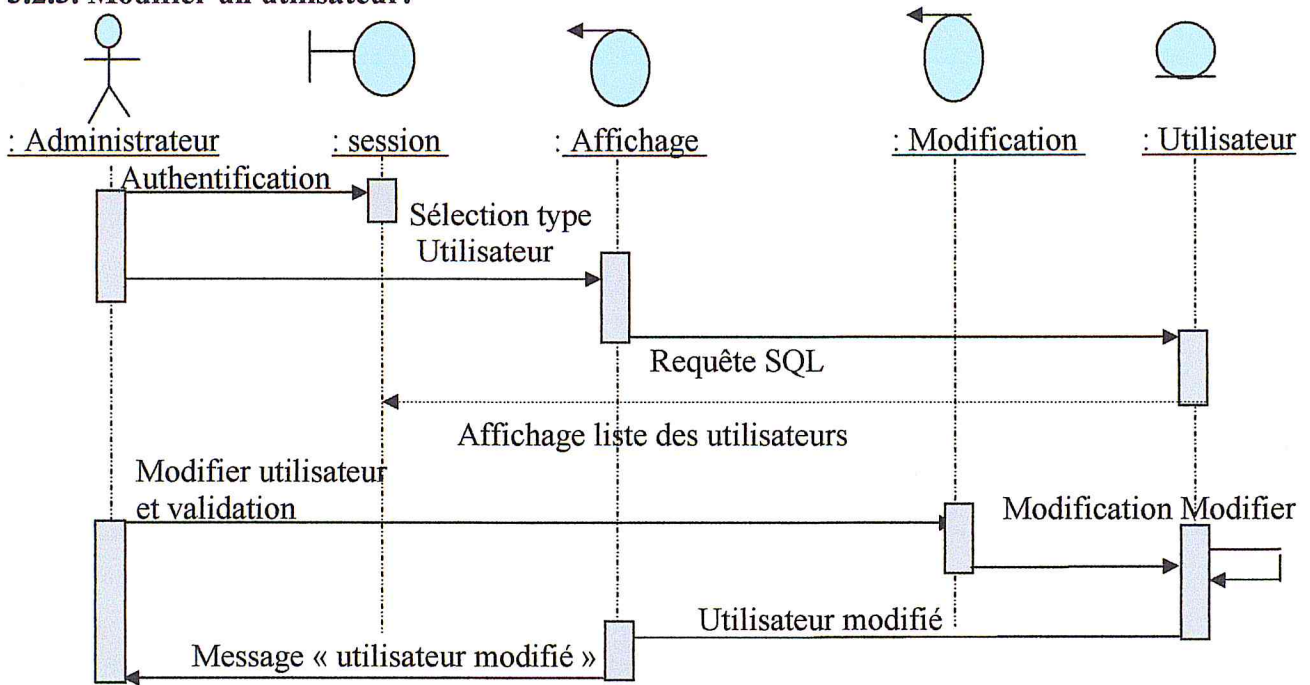


3.2.2. Ajouter un utilisateur:

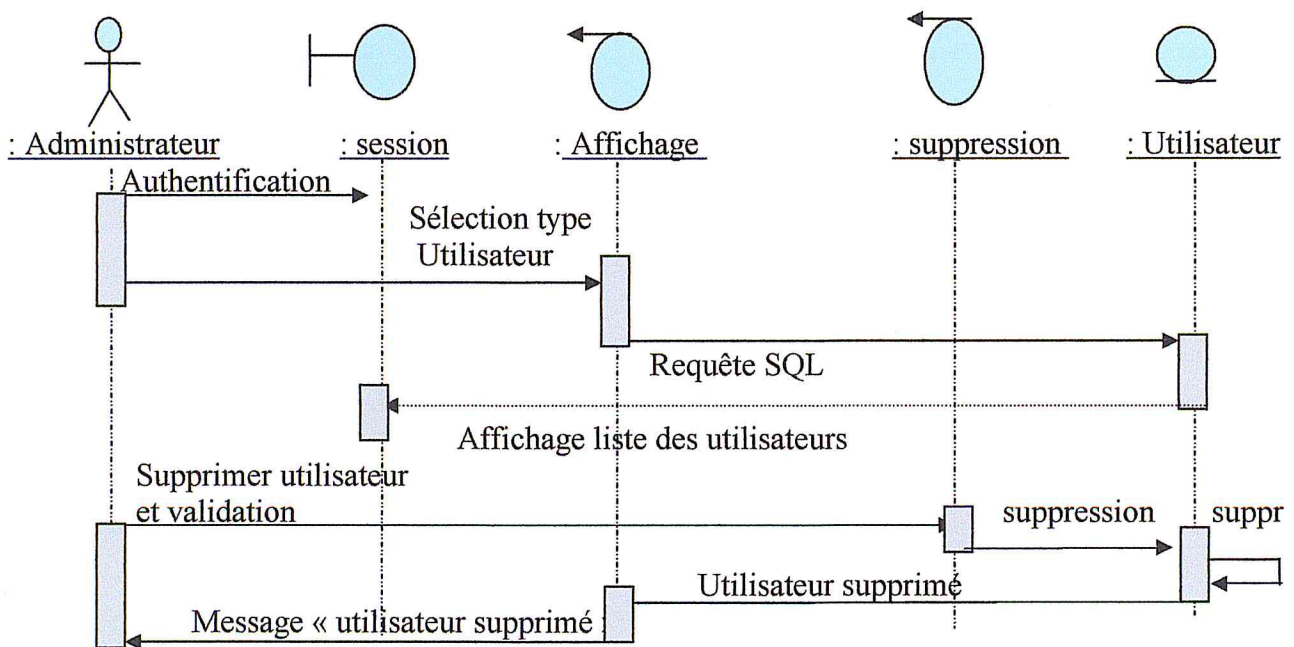




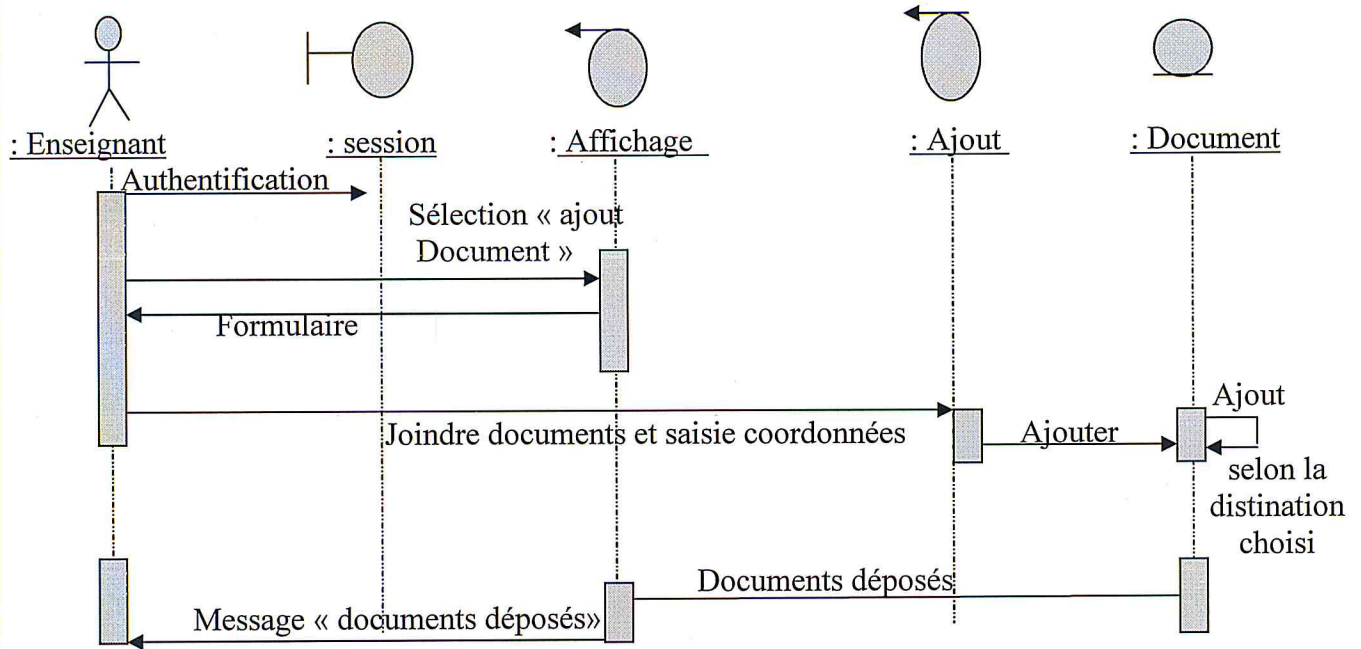
3.2.3. Modifier un utilisateur:



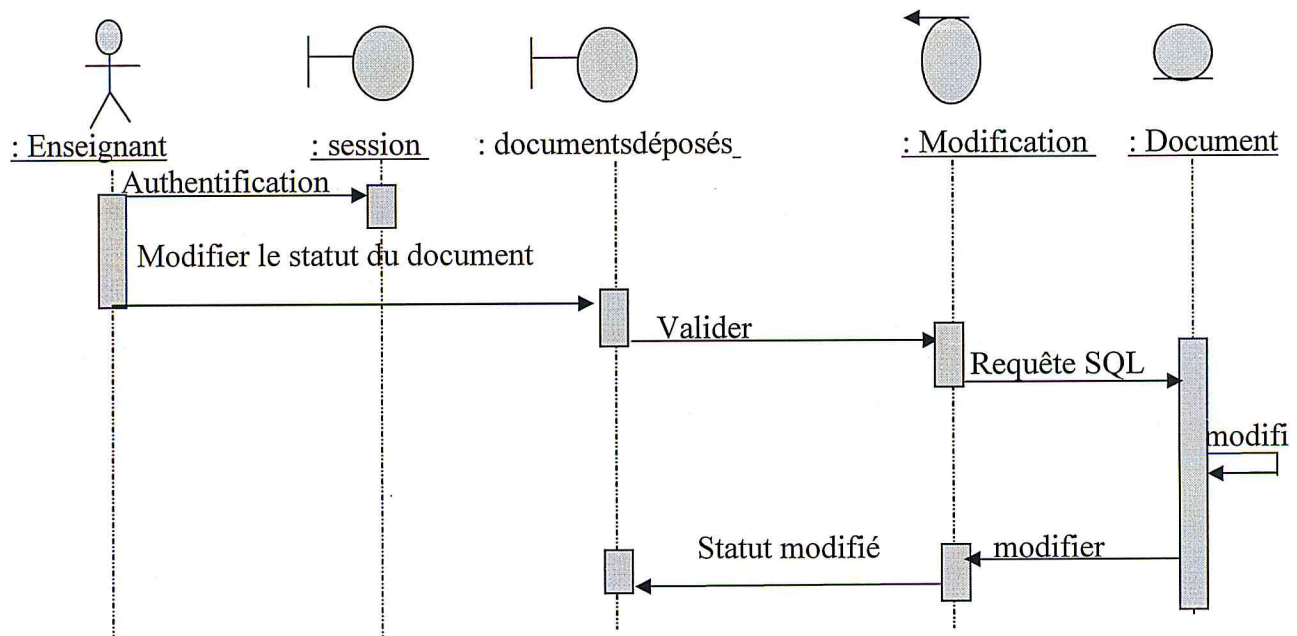
3.2.4. Supprimer un utilisateur:



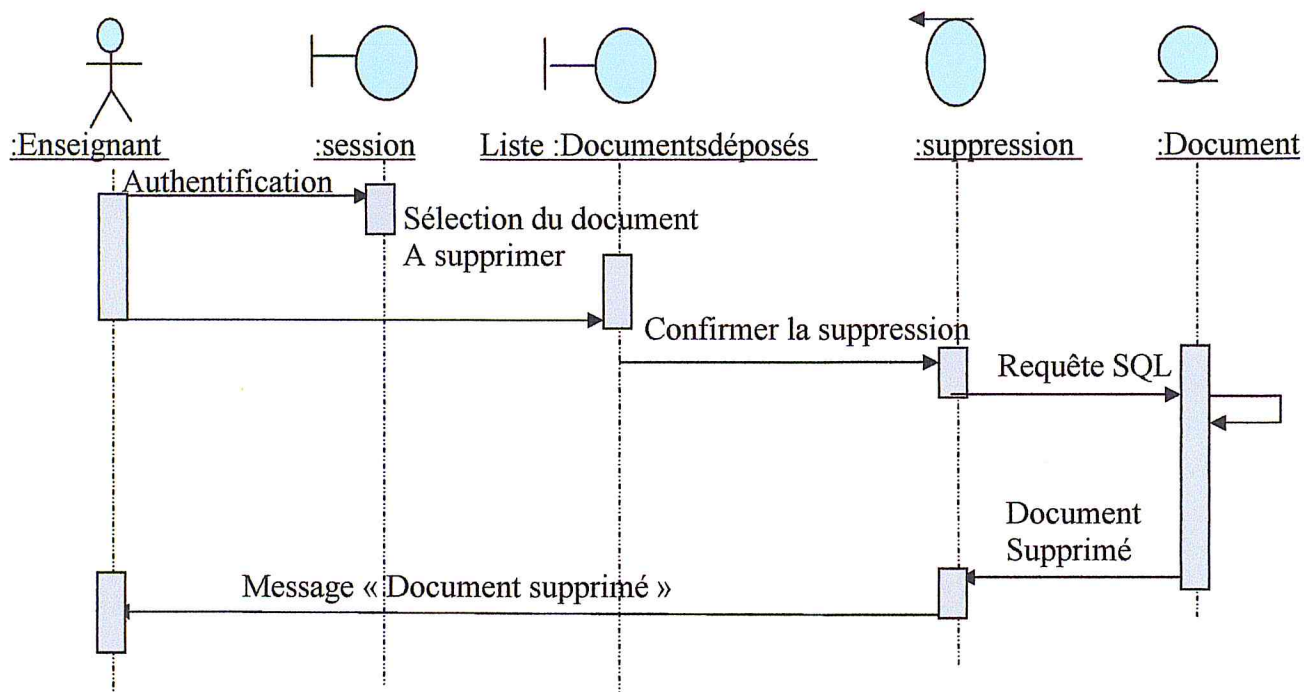
3.2.5. Déposer un document:



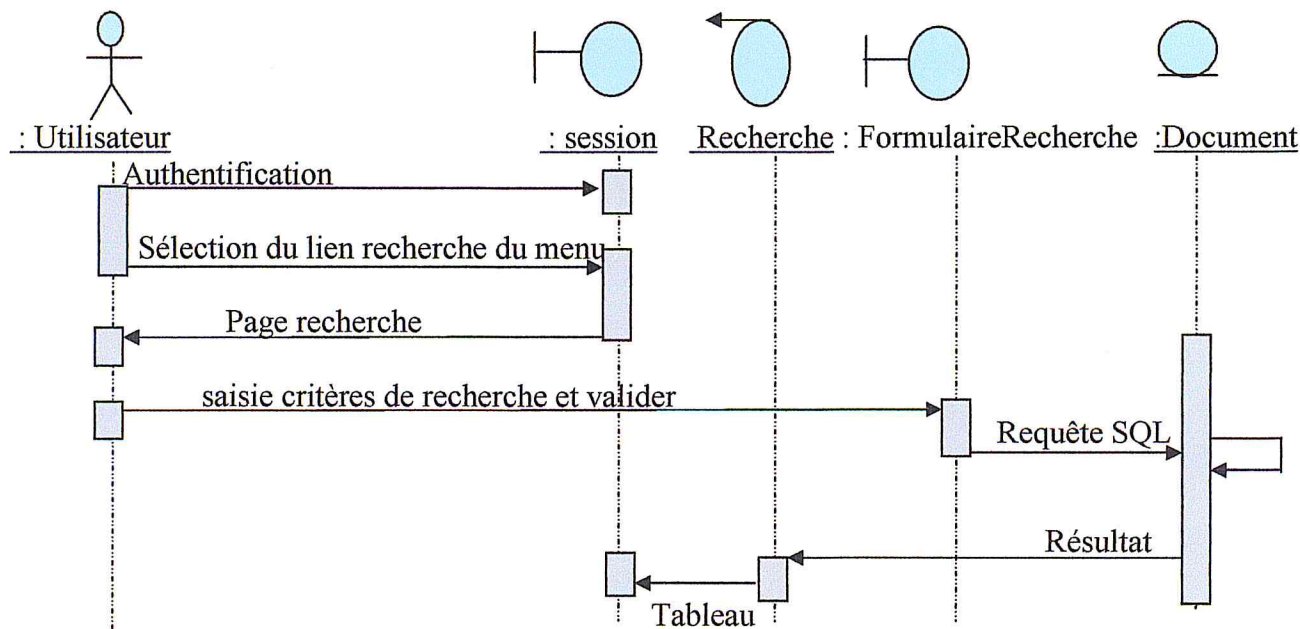
3.2.6. Modifier le statut ( public ou privé ) d'un document:



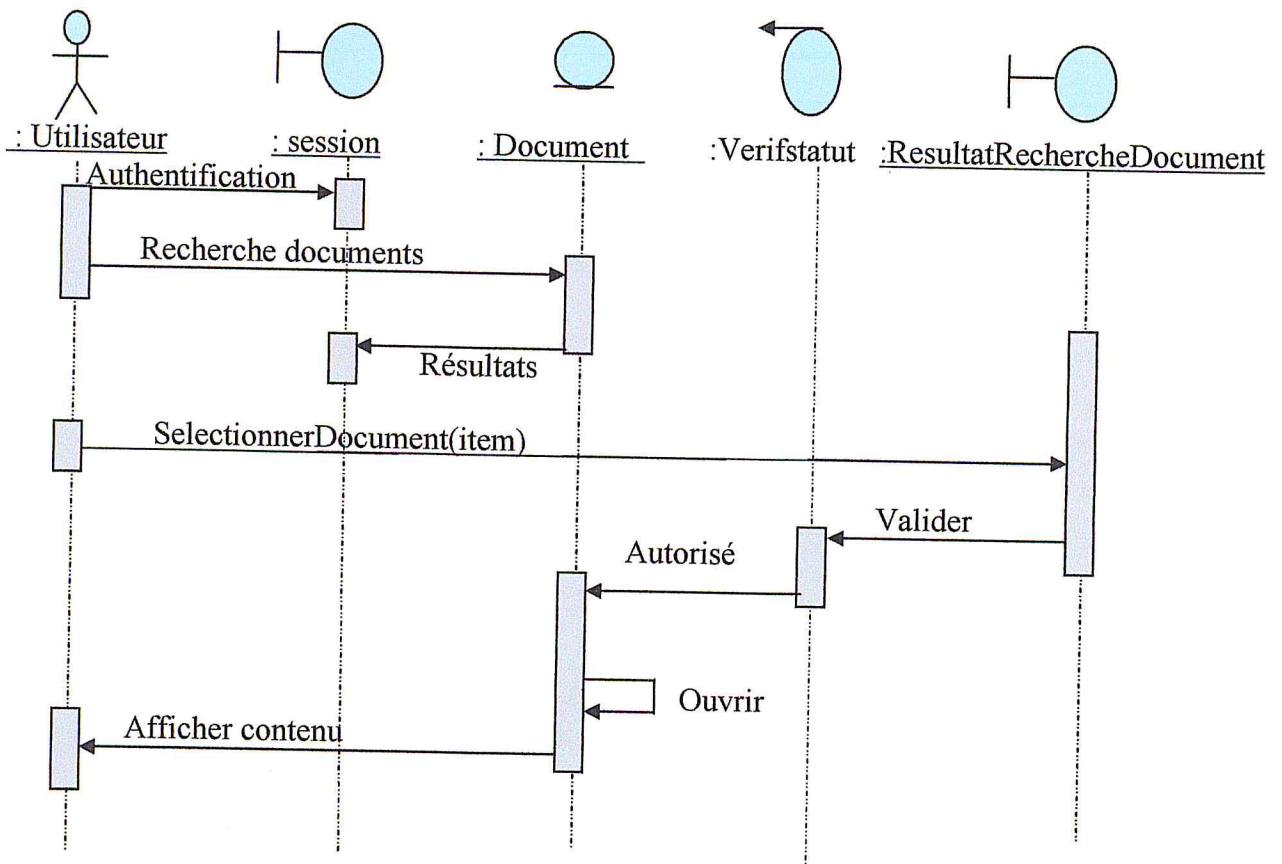
3.2.7. Supprimer un document:



3.2.8. Recherche de documents:



3.2.9. Consulter un document:



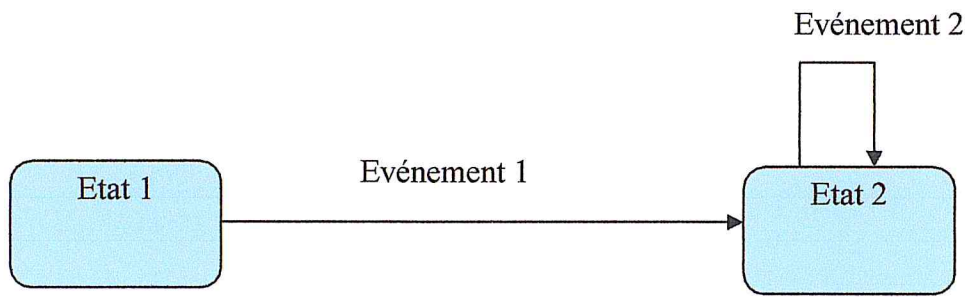
### 3.3. Diagramme d'état:

Les diagrammes d'état sont des automates a états, composés d'états, de transitions, d'événements et d'activités.

Ils sont particulièrement importants dans la modélisation du comportement d'une interface, d'une classe ou d'une collaboration et mettent l'accent sur le comportement d'un objet ordonnancé par les événements, ce qui est particulièrement utile dans la modélisation des systèmes réactifs.

- **Etat:** Condition dans laquelle se trouve un objet
- **Transition:** Chemin entre deux états
- **Evénement:** occurrence qui survient dans le domaine

Les symboles qui constituent un diagramme d'état:



Pseudo état initial ●

Pseudo état final ○

Event [C1] / Action



L'action est exécutée instantanément

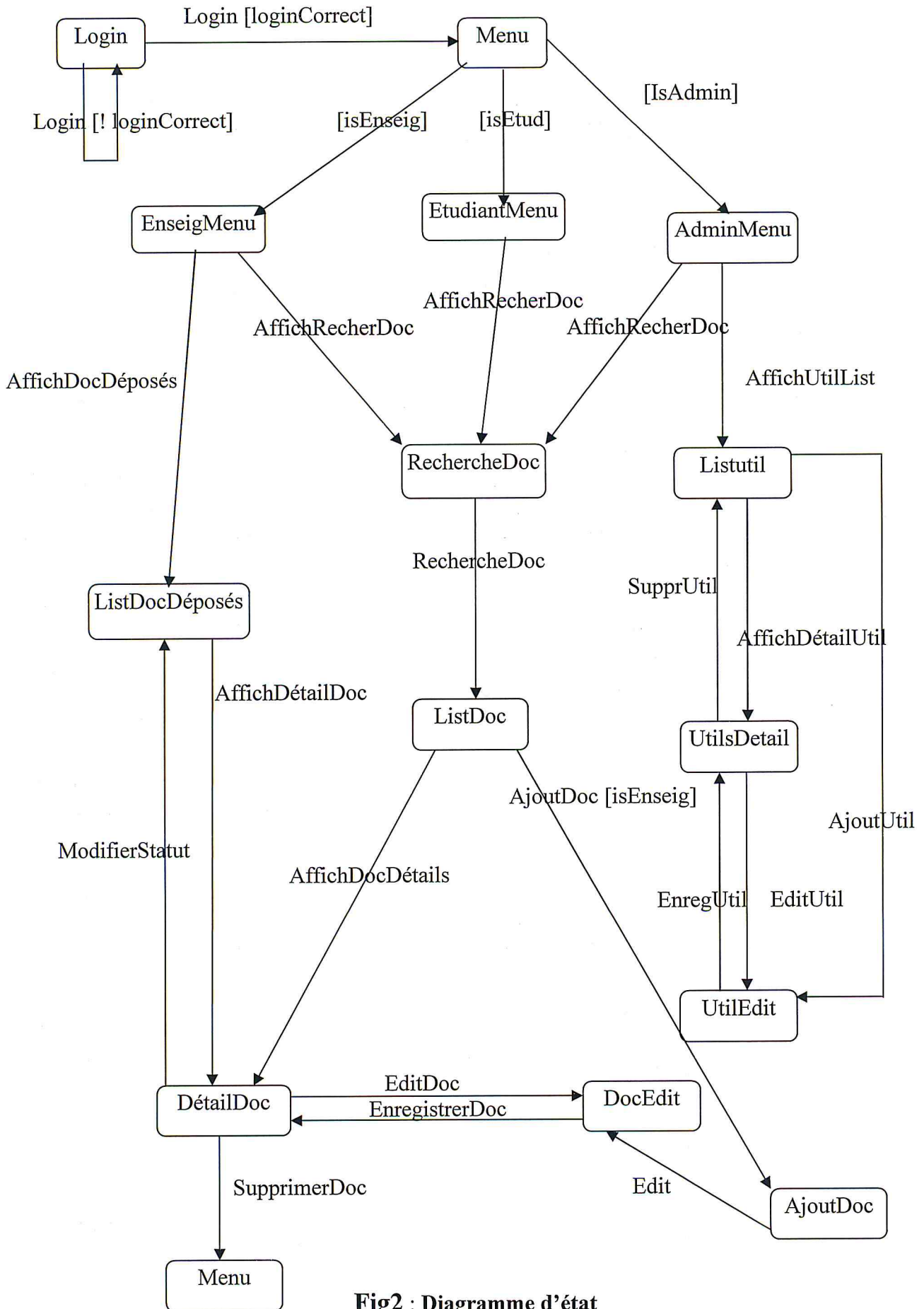


Fig2 : Diagramme d'état

## 4. Modélisation Statique:

Précédemment on a parlé des deux grandes catégories de diagrammes UML (statique et dynamique) Jusqu'ici, on a défini les différentes taches constituant le système L'ensemble des cas d'utilisation permet d'avoir les données manipulables par le système Une fois les solutions techniques et les fonctionnalités validées et après avoir rassemblé les données Il nous reste à représenter le monde conceptuel des données Pour construire le monde conceptuel nous avons dégagé dans un premier temps les entités suivantes:

- Document
- Image
- Ouvrage
- Utilisateur
- Groupe
- Administrateur
- Enseignant
- Etudiant
- Type document
- Spécialité
- Module
- Semestre

On a établi ensuite les relations principales entre ses entités :

- Un utilisateur a un statut d'accès au système
- Un groupe peut contenir un ou plusieurs utilisateurs
- Un enseignant peut déposer un ou plusieurs documents
- Un enseignant de peut être supprimée que par l'enseignant qui l'a déposé
- Un document peut avoir plusieurs types
- Un document a un statut de diffusion
- Un module appartient à une ou plusieurs spécialités
- Un document peut être consulté par un ou plusieurs utilisateurs
- Un document peut être téléchargé par un ou plusieurs utilisateurs
- Un document peut avoir un ou plusieurs auteurs
- Une image peut avoir un ou plusieurs formats

À ses principales entités viennent se greffer des attributs permettant de les décrire, Ces derniers ayant été dégagés lors de la partie de l'état de l'art, On a ensuite réalisé le diagramme de classe en inscrivant ces différents champs

#### 4.1. Le dictionnaire de données:

Un dictionnaire de données est une collection de métadonnées ou de données de référence à la conception d'une base de données relationnelle, il est d'une importance stratégique particulière, car il est le vocabulaire commun de l'organisation, Le dictionnaire des données est constitué de l'ensemble des spécifications des données utilisées aux différents niveaux d'analyse et de conception Le dictionnaire présente les sigles, les codes ou les symboles employés dans les documents, précise les synonymes, les alias. Le dictionnaire permet également de définir la structure des données composées avec une notation syntaxique stricte.

La définition du dictionnaire peut également être plus large et intégrer, en plus de la spécification des données, des informations sur les fichiers qui les contiennent et les processus qui les utilisent Il est souvent représenté par un tableau à quatre colonnes contenant le nom, le code et le type de donnée ainsi que des commentaires.



Classe	Attributs	Code attributs	Méthodes	Commentaire
Document	Code document Titre document Nature du document Thème Mots-clés Format Nombre de pages Résumé Emplacement Diffusion	Code_doc Titre_doc Nature_doc Thème_doc M_clés Format_doc Nbr_pages Résumé_doc Emplac_doc Diff_doc	Consulter_doc Recherche_doc Télécharger_doc Ajouter_doc Supprimer_doc	Diffusion: s'il a un statut d'accès personnel, internet, Groupe
Ouvrage	éditeur Date d'édition	Éditeur_ouv Dat_pub_doc	Ajouter_ouv supprimer_ouv	
Image	Taille Résolution	Taille_img Résol_img	Ajouter_img supprimer_img	
Type	Code type Intitulé type Description	Code_type Intitu_type Desc_type	Ajouter_typ_doc Modifier_typ_doc Supprimer_typ_doc	
Utilisateur	Login Mot de passe Privilège Nom Prénom E-mail Adresse Téléphone	Login_util Mot_pass Priv_util Nom_util Prenom_util mail_util Adr_util tél_util	Inscrire_util Modifier_util Supprimer_util Modifier_statut Affecter_Privilège	Priv_util: statut d'accées au système simple utilisateur ou utilisateur privilégié
Enseignant	Identifiant enseignant Grade enseignant	Id_ens Grade_ens	Ajouter_enseignant Modifier_enseignant Supprimer_enseignant	
Etudiant	Numero d'inscription	Num_inscription	Ajouter_etudiant Modifier_etudiant Supprimer_etudiant	
Spécialité	Code spécialité intitulé spécialité	Code_spec intitu_spec	Ajouter_spécialité Modifier_spécialité Supprimer_spécialité	
Module	Identifiant module Intitulé module	Id_module intitulé_mod	Ajouter_mod Modifier_mod Supprimer_mod	
Semestre	Code semestre	Code_eme	Ajouter_eme Modifier_eme Supprimer_eme	
Auteur	Code auteur Nom auteur Prénom auteur	Co_aut Nom_aut Pnom_aut	Ajouter_aut Modifier_aut Supprimer_aut	

## 4.2. Diagramme de classe [1] :

Les diagrammes de classes représentent un ensemble de classes, d'interfaces et de collaborations, ainsi que leurs relations, ce sont les diagrammes les plus fréquents dans la modélisation des systèmes à objets, ils présentent la vue de conception statique d'un système.

La classe est une description abstraite d'un ensemble d'objets.

La classe peut être vue comme la factorisation des éléments communs à un ensemble d'objets.

La classe décrit le domaine de définition d'un ensemble d'objets

Description conceptuellement séparée en deux parties:

- La *spécification* d'une classe qui décrit le domaine de définition et les propriétés des instances de cette classe (type de donnée)
- La *réalisation* qui décrit comment la spécification est réalisée

### 4.2.1. Les Caractéristiques des classes :

Attributs, Opérations, Réceptions, Relations, Multiplicité, Persistance, Composant

### 4.2.2. Les relations entre les classes :

L'association, l'agrégation, la composition, la généralisation, la dépendance

L'association exprime une connexion sémantique bidirectionnelle entre classes Une association est une abstraction des liens qui existent entre les objets instances des classes associées Étant donnée une association non décorée entre deux classes, on peut naviguer d'un type d'objet vers un autre type d'objet.

Par défaut une association est navigable dans les deux sens.

Une indication de navigabilité suggère en général qu'à partir d'un objet à une extrémité on peut directement et facilement atteindre l'un des objets à l'autre extrémité.

Lors d'une mise en œuvre programmée, ceci peut suggérer qu'un objet source mémorise une référence directe aux objets cible.

La visibilité des propriétés d'une classe peut se résumer comme suit:

#### **Public + :**

Visible à l'extérieur de la classe

#### **Protégé # :**

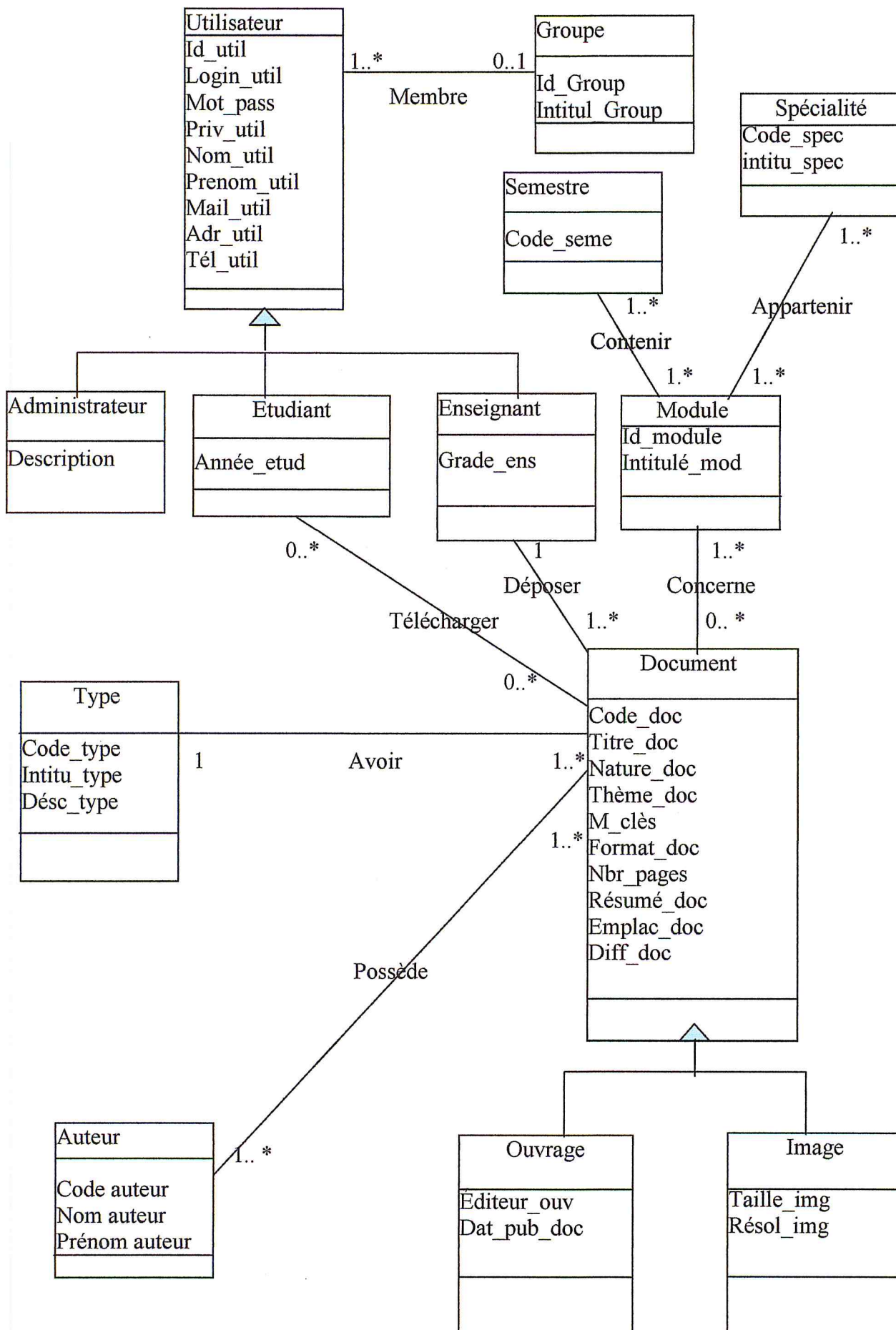
Visible seulement par les descendants

#### **Private - :**

Visible à l'intérieur des méthodes

#### **Souligné:**

Variable/opération de classe



## 5. Catégorisation automatique des documents [17] :

La catégorisation automatique des documents est considérée parmi les composants les plus importants dans un système de recherche d'informations car elle permet d'organiser les documents par catégories.

Cette catégorisation permet d'accélérer, cibler et améliorer la recherche d'informations

Plusieurs méthodes de catégorisation automatique ont été proposées, ces méthodes peuvent être subdivisées en deux catégories : des méthodes se basant sur l'ingénierie des connaissances et celles utilisant des techniques d'apprentissage automatique.

L'approche que nous proposons permet d'assigner des documents à une ou plusieurs catégories prédéfinies et ce en utilisant des attributs d'ordre physique (taille du document) logique (la structure logique du document) et discursive (l'existence de certaines expressions linguistiques dans les unités logiques bien définies).

L'utilisation d'un ensemble de documents déjà catégorisés permet de générer un ensemble de règles de trois types : des règles physiques, logiques et discursives exploitant respectivement les attributs physiques, logique.

Pour générer des règles de catégorisation nous disposons d'un ensemble de documents

D'apprentissage

Chaque document d'apprentissage  $d_j$  est représentée par sont : identifiant  $did_j$ , sa catégorie  $C_j$ , sa taille en nombre de mots  $nm_j$  sa structure logique  $S_j$

La répartition de l'ensemble des documents est présentée dans le tableau ci-dessous

Notation	Catégorie
C1	Document
C2	Dictionnaire
C3	Ouvrage
C4	Thèse
C5	Mémoire
C6	Rapport
C7	Article
C8	Image
C9	Page web
C10	Questionnaire

### 5.1. L'attribut NM:

La prise en compte d'attributs numériques nécessite une procédure de discrétisation.

Discrétiser un attribut numérique consiste à découper son domaine de variation en un nombre fini d'intervalles, ces intervalles sont ensuite considérés comme de nouveaux concepts symboliques.

Dans notre cas nous disposons d'un seul attribut numérique continu à savoir l'attribut NM qui représente la taille de document en nombre de mots.

Pour discrétiser cet attribut nous avons choisi de découper son domaine de variation en trois sous intervalles.

Le premier sous intervalles  $<5000$  représente une taille en nombre de mots faible.

Le deuxième sous intervalle  $[5000,50000]$  représente une taille en nombre de mots moyenne, tandis que le troisième intervalle  $> 50000$  représente une taille en nombre de mots élevée

## 5.2. L'attribut SL:

Une structure logique est une suite d'unités logique ordonnées l'une a la suite de l'autre pour faire apparaître une idée quelconque.

Pour chaque unité nous allons lui affecter un poids appartenant a l'intervalle] 0,1] qui représente son importance dans la structure logique en question, ces poids sont calculés en utilisant les documents d'apprentissage.

Pour cet attribut nous avons identifié a partir de l'ensemble des documents les structures logiques possibles qui sont déduites a partir de l'ensemble des documents et qui sont représentées dans le tableau suivant:

Notation	Structure logique
S1	Titre (1), Auteur (0,5), résumé (0,3), M_clés (1), thème (1), format, Type doc (1)
S2	Titre (1), Auteur (0,5), résumé (0,2), M_clés (0,3), thème (1), éditeur (1), date édit
S3	Titre (1), auteur (1), M_clés (0,2), thème (0,5), format (0,7) résolution (1)

## 5.3. Règles physiques:

La génération de règles physiques se base sur les valeurs de l'attribut NM, dans notre cas nous avons dégagé trois règles physiques chaque règle est de la forme:

Si Condition (NM)

Alors  $\{(C1, \alpha1), (C2, \alpha2), \dots, (C8, \alpha8)\}$

Avec : Condition (NM) est vérifiée si  $NM \in \{\text{Faible, moyenne, élevé}\}$

$\alpha_i$  est le degré d'appartenance a la catégorie  $C_i$ , et ce degré est la proportion des documents d'apprentissage appartenant a la catégorie  $C_i$  et vérifiant la condition(NM)

*Exemple:*

Si NM=élevée alors

$\{(\text{Document}, 1.00), (\text{Ouvrage}, 1.00), (\text{Dictionnaire}, 1.00), (\text{Thèse}, 0,67), (\text{Mémoire}, 0,50), (\text{rapport}, 0,50), (\text{Article}, 0,25), (\text{Page web}, 0,20), (\text{image}, 0,00)\}$

## 5.4. Règles logiques:

En utilisant les valeurs de l'attribut SL nous avons dégagé des règles logiques, chaque règle est de la forme :

Si Condition (SL)

Alors  $\{(C1, \beta1), (C2, \beta2), \dots (C_i, \beta_i)\}$

Avec Condition (SL) est vérifiée si  $SIM(S_j, S_i) \geq S_0$

$\beta_i$  est le degré de d'appartenance a la catégorie  $C_i$ , ce degré est la proportion des documents d'apprentissage appartenant a la catégorie  $C_i$  et vérifiant Condition (SL)

$SIM(S_j, S_i)$  est la similarité de la structure logique  $S_j$  d'un document donnée avec la structure

logique prédéfinie  $S_i \in SL = \{S_1, S_2, \dots, S_i\}$  cette similarité est calculée comme suit:

$$SIM(S_j, S_i) = \frac{\sum_{U_i \in S_j \cap S_i} P_i}{\sum_{U_i \in S_i} P_i}$$

Avec :

$U_i$ : Une unité logique appartenant a la structure logique prédéfinie  $S_i$

$P_i$  : Le poids affecté a l'unité logique  $U_i$

$S_0$ : est le seuil minimal de similarité de  $S_j$  avec la structure logique prédéfinie  $S_i$ , au dessous duquel Condition(SL) n'est pas vérifiée, nous avons fixé la valeur de ce seuil a 0,50

*Exemple:*

Si  $SL=S_2$  Alors

{(Dictionnaire,0.00),(Ouvrage,1.00),(Thèse,1.00),(Mémoire,0.30),(Rapport,),(Article,0.00),(Page web,0.00)(image,0.00)}

### 5.5. Catégorisation de nouveaux documents:

A l'arrivée d'un nouveau document  $d_j$  le processus de catégorisation déclenche un pré-traitement pour déterminer la taille du document en nombre de mots  $nm_j$ , sa structure logique  $S_j$

A l'issue de ce pré-traitement deux types de discrimination seront déclenchés: discrimination physique et logique exploitant respectivement les règles physiques et logiques déjà générées

La discrimination physique permet de déclencher la règle physique vérifiant Condition ( $nm_j$ ), à l'issue de cette discrimination nous obtenons une première catégorisation possible

$$E_1 = \{(C_1, \alpha_1), (C_2, \alpha_2), \dots, (C_i, \alpha_i)\}$$

En deuxième lieu la discrimination logique nous permet de déclencher la règle logique vérifiant Condition ( $S_j$ ) pour obtenir une deuxième catégorisation possible :

$$E_2 = \{(C_1, \beta_1), (C_2, \beta_2), \dots, (C_i, \beta_i)\}$$

Pour obtenir une seule catégorisation optimale nous devons combiner ces deux catégorisations possibles  $E_1, E_2$  pour cela nous avons choisi d'utiliser la norme triangulaire min de Zadeh

Cette catégorisation optimale est calculée comme suit:

$$E = E_1 \square E_2 = \{(C_1, \min(\alpha_1, \beta_1)), (C_2, \min(\alpha_2, \beta_2)), \dots, (C_i, \min(\alpha_i, \beta_i))\}$$

Les catégories choisies sont celles réalisant le maximum de degré d'appartenance

## 6. Gestion des droits d'accès et des privilèges [16] :

Les risques propres a une source de données sont le suivants:

- Vol de données (Perte de confidentialité)
- Altération de données (Perte d'intégrité)
- Destruction de données
- Augmentation du niveau de privilèges d'un utilisateur d'une application
- Ressources systèmes abusives

La principale question qui se pose lors du développement d'une application, c'est quelle stratégie adopter vis à vis des utilisateurs : contrôle de leurs droits d'accès par l'application ou par le SGBD

Par le SGBD, Dans le cas où toute l'information métier repose sur une base de données comportant également toutes les procédures stockées de contrôle de l'intégrité, de la logique métier et des actions utilisateurs, il est logique de déléguer au SGBD le contrôle d'accès et les habilitations.

Ceci suppose que l'administrateur de bases de données réalise les opérations d'attribution des privilèges et de synchronisation avec l'annuaire des utilisateurs du système d'information. L'application ne devient alors qu'une interface graphique ergonomique d'interrogation de la base de données métier.

Par l'application. Dans le cas où l'application gère elle-même le niveau d'accréditation des utilisateurs, elle va se connecter sous sa propre identité logique à la base de données et décider des informations et des opérations que l'utilisateur peut voir, modifier et réaliser. C'est la stratégie employée par les applications dont la logique métier n'est pas intégrée directement dans la base de données et qui gèrent plusieurs sources de données.

Il convient pour chaque compte d'accès d'identifier les privilèges minima à accorder ainsi que le niveau de granularité adéquat.

### 6.1. Les privilèges:

La description suivante désigne représente les types d'utilisateurs qui devraient être habilités pour chaque classe de privilège

#### Classes de privilèges

- Accès au contenu de l'information
- Gestion du schéma de la base de données
- Gestion des privilèges utilisateurs
- Gestion des paramètres systèmes

#### Types de comptes

- Utilisateur, Application
- Administrateur (Application parfois)
- Administrateur
- Administrateur

#### Statut utilisateur

Groupe

Autorisé

Autorisé

Internet

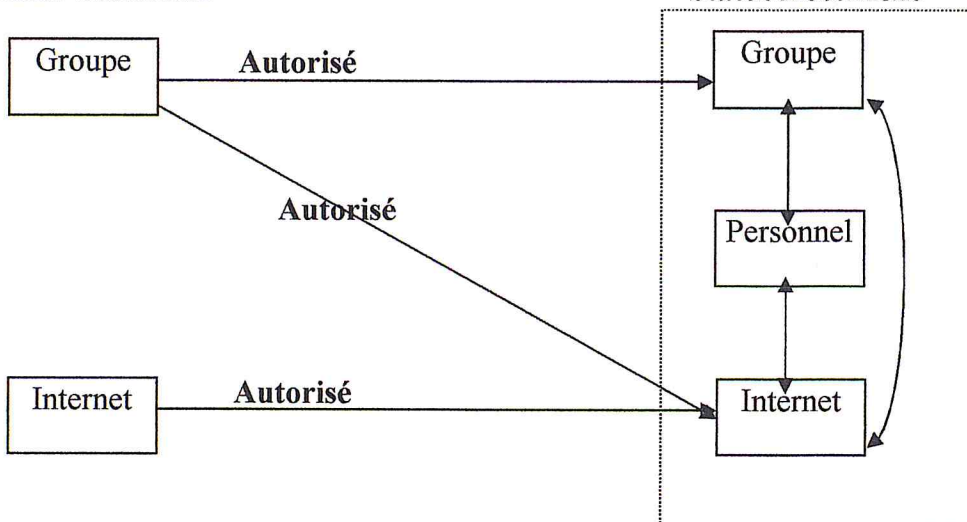
Autorisé

#### Statut Document

Groupe

Personnel

Internet



## 7. Modélisation de la base de données [2] :

### 7.1. Définitions:

Une BD est composée de données stockées dans des mémoires de masse sous une forme structurée, et accessibles par des applications différentes et des utilisateurs différents. Une BD doit pouvoir être utilisée par plusieurs utilisateurs en même temps.

Une base de données est structurée par définition, mais sa structuration doit avoir un caractère Universel : il ne faut pas que cette structure soit adaptée à une application particulière, mais Qu'elle puisse être utilisable par plusieurs applications distinctes. En effet, un même ensemble De données peut être commun à plusieurs systèmes de traitement dans un problème physique (Par exemple la liste des passagers d'un avion, stockée dans une base de données, peut aussi Servir au service de police à vérifier l'identité des personnes interdites de séjour, et au service Des douanes pour associer des bagages aux personnes....)

L'organisation d'un SI relève plus de la gestion que de l'informatique et n'à pas exactement sa Place dans un document sur la programmation. En revanche la cheville ouvrière d'un système D'information est un outil informatique appelé un SGBD (système de gestion de base de Données)

Un SGBD est un ensemble de logiciels chargés d'assurer les fonctions minimales Suivantes :

- Le maintien de la cohérence des données entre elles,
- le contrôle d'intégrité des données accédées,
- les autorisations d'accès aux données,
- les opérations classiques sur les données (consultation, insertion, modification, Suppression)



On organise actuellement les SGBD selon deux modes :

**7.1.1. L'organisation locale** selon laquelle le SGBD réside sur la machine où se trouve la base de Données

**7.1.2. L'organisation client serveur** selon laquelle sur le SGBD est réparti entre la machine Serveur locale supportant la BD (partie SGBD serveur) et les machines des utilisateurs (partie SGBD client). Ce sont ces deux parties du SGBD qui communiquent entre elles pour assurer Les transactions de données.

Il est admis de nos jours qu'une BD est décrite en trois niveaux d'abstraction : un seul niveau a Une existence matérielle physique et les deux autres niveaux sont une explication abstraite de Ce niveau matériel.

**7.1.3. Niveau externe** : correspond à ce que l'on appelle une vue de la BD ou la façon dont sont perçues au niveau de l'utilisateur les données manipulées par une certaine Application (vue abstraite sous forme de schémas)

**7.1.4. Niveau conceptuel** : correspond à la description abstraite des composants et des processus entrant dans la mise en œuvre de la BD. Le niveau conceptuel est le plus Important car il est le résultat de la traduction de la description du monde réel à l'aide D'expressions et de schémas conformes à un modèle de définition des données.

**17.1.5. Niveau interne** : correspond à la description informatique du stockage physique des Données (fichiers séquentiels, indexages, tables de hachage,...) sur le disque dur.



Le niveau conceptuel forme l'élément essentiel d'une BD et donc d'un SGBD chargé de gérer Une BD, et dans notre cas nous l'avons décrit par le formalisme du diagramme de classe qui prend une part de plus en plus grande dans le formalisme de description conceptuelle des données, On va ensuite le traduire dans le Modèle Logique de représentation des Données (MLD). Ce dernier s'implémentera selon un modèle physique des données.

## 7.2. Model de données relationnelles:

Défini par EF Codd de la société IBM dès 1970, ce modèle a été amélioré et rendu Opérationnel dans les années 80 sous la forme de SBGD-R, Ci-dessous une liste non exhaustive de tels SGBD-R

:

- Access de Microsoft,
- Oracle,
- DB2 d'IBM,
- Interbase de Borland,
- SQL server de microsoft,
- Informix,
- Sybase,
- MySQL,
- PostgreSQL

Une relation binaire R est un sous-ensemble d'un produit cartésien de deux ensembles finis E et F que nous nommerons domaines de la relation R :

$$R \subseteq E \times F$$

Cette définition est généralisable à n domaines, nous dirons que R est une relation n-aire sur Les domaines E1, E2, ..., En si et seulement si :

$$R \subseteq E_1 \times E_2 \dots \times E_n$$

Les ensembles Ek peuvent être définis comme en mathématiques : en extension ou en Compréhension :

$$E_k = \{12, 58, 36, 47\} \text{ en extension}$$

$$E_k = \{x / (x \text{ est entier}) \text{ et } (x \in [1,20])\} \text{ en compréhension}$$

si nous avons:  $R = \{(v_1, v_2 \dots, v_n)\}$ ,

Au lieu d'écrire :  $(v_1, v_2 \dots, v_n) \in R$ , on écrira  $R(v_1, v_2, \dots, v_n)$

## 7.3. Schéma d'une relation :

On appelle schéma de la relation R :  $R(a_1 : E_1, a_2 : E_2, \dots, a_n : E_n)$  Où  $(a_1, a_2 \dots, a_n)$  sont appelés les attributs, chaque attribut ak indique comment Est utilisé dans la relation R le domaine Ek, chaque attribut prend sa valeur dans le Domaine qu'il définit, nous notons  $val(a_k) = v_k$  où  $v_k$  est un élément (une valeur) Quelconque de l'ensemble Ek (domaine de l'attribut ak).

#### 7.4. Représentation du model relationnelle:

Une fois les solutions techniques et les fonctionnalités validées, et après avoir Rassemblé le maximum de données, il me restait à représenter le monde conceptuel de données.

Après avoir déjà établi le diagramme de classe nous allons représenter les différents relations de notre systèmes a travers le schéma du model relationnel.

Chaque opération plusieurs a plusieurs est représentée par une table distincte Les associations un a plusieurs peuvent être représentée de deux manières :

- Représenter chaque association par une table distincte
- Enfuir une clé étrangère représentant cette association, dans la table pour la classe « plusieurs »

Les associations « n-aire » sont traduites par des relations distinctes, En appliquant les règles de passage du model objet au model relationnel, après la validation nous avons obtenu le model relationnel suivant:

Document (Code\_doc, Titre\_doc, Nature\_doc, Thème\_doc, M\_clès, Format\_doc, Nbr\_pages, Résumé\_doc, Emplac\_doc, Diff\_doc, # Code\_type, # Id\_ens)

Enseignant (Id\_util, Login\_util, Mot\_pass, Priv\_util, Nom\_util, Prenom\_util, mail\_util, Adr\_util, tél\_util, Grade\_ens, # Id\_group)

Image (Code\_doc, Titre\_doc, Nature\_doc, Thème\_doc, M\_clès, Format\_doc, Nbr\_pages, Résumé\_doc, Emplac\_doc, Diff\_doc, Taille\_img, Résol\_img, #Code\_type, #Id\_ens)

Ouvrage (Code\_doc, Titre\_doc, Nature\_doc, Thème\_doc, M\_clès, Format\_doc, Nbr\_pages, Résumé\_doc, Emplac\_doc, Diff\_doc, Éditeur\_ouv, Dat\_pub\_doc, # Code\_type, # Id\_ens)

Utilisateur (Id\_util, Login\_util, Mot\_pass, Priv\_util, Nom\_util, Prenom\_util, mail\_util, Adr\_util, tél\_util, # Id\_group)

Etudiant (Id\_util, Login\_util, Mot\_pass, Priv\_util, Nom\_util, Prenom\_util, mail\_util, Adr\_util, tél\_util, # Id\_group)

Administrateur (Id\_util, Login\_util, Mot\_pass, Priv\_util, Nom\_util, Prenom\_util, mail\_util, Adr\_util, Tél\_util, # Id\_group, Description)

Type document (Code\_type, Intitu\_type, Desc\_type)

Spécialité (Code\_spec, intitu\_spec)

Module (Id\_module, Intitulé\_mod)

Semestre (Code\_eme)

Groupe (Id\_group, Intitul\_group)

Auteur (Code\_auteur, Nom\_auteur, Prénom\_auteur)

Possède (Code\_doc, Code\_Auteur)

Contenir (Id\_module, Code\_seme)

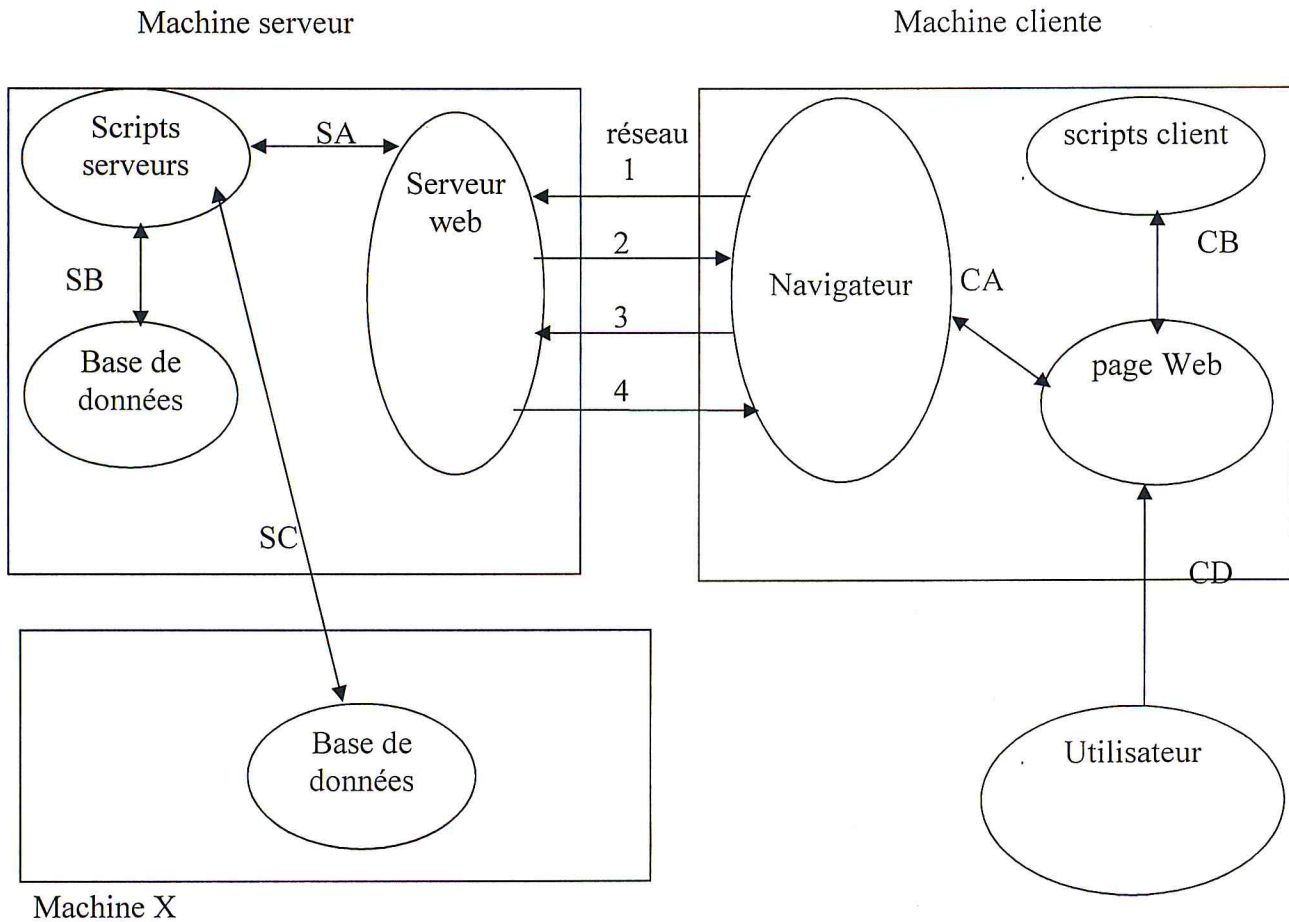
Appartenir (Code\_spec, Id\_module)

Télécharger (Id\_util, Code\_doc)

Concerne (Id\_module, Code\_doc)

# Partie 4: Réalisation

1. Les échanges de données dans une application web [4] :



Machine Serveur

Serveur Web

Scripts exécutés côté serveur. Ils peuvent l'être Par des modules du serveur ou par des Programmes externes au serveur (CGI).

De données - Celle-ci peut être sur la même machine Que le programme qui l'exploite ou sur une autre Via Internet.

Machine Client

Linux, Windows

Apache (Linux, Windows)

PHP (Apache)

MySQL (Linux, Windows)

Linux, Windows

1- Le navigateur demande une URL pour la 1ère fois (http://machine/url). Aucun paramètre n'est passé.

2- Le serveur Web lui envoie la page Web de cette URL. Elle peut être statique ou bien dynamiquement générée par un script serveur (SA) qui a pu utiliser le contenu de bases de données (SB, SC). Ici, le script détectera que l'URL a été demandée sans passage de paramètres et générera la page WEB initiale.

Le navigateur reçoit la page et l'affiche (CA). Des scripts côté navigateur (CB) ont pu modifier la page initiale envoyée par le serveur. Ensuite par des interactions entre l'utilisateur (CD) et les scripts (CB) la page Web va être modifiée.

Les formulaires vont notamment être remplis.

3- L'utilisateur valide les données du formulaire qui doivent alors être envoyées au serveur web. Le navigateur redemande l'URL initiale ou une autre selon les cas et transmet en même temps au serveur les valeurs du formulaire. Il peut utiliser pour ce faire deux méthodes appelées GET et POST.

A la réception de la demande du client, le serveur déclenche le script (SA) associé à l'URL demandée, script qui va détecter les paramètres et les traiter.

4- Le serveur délivre la page WEB construite par programme (SA, SB, SC). Cette étape est identique à l'étape 2 précédente. Les échanges se font désormais selon les étapes 2 et 3.

### 1.1. Le protocole http [6] :

Nous allons découvrir ici comment un serveur web répond aux demandes de ses clients.

Le service web ou service HTTP est un service TCP-IP qui travaille habituellement sur le port 80 .

Il pourrait travailler sur un autre port.

Dans ce cas le navigateur client serait obligé de préciser ce port dans l'URL qu'il demande

Une URL a la forme générale suivante:

**Protocole://machine [:port]/chemin/infos**

Lorsqu'un utilisateur demande le chargement d'une URL le navigateur:

1- Ouvre une communication TCP-IP avec la machine et le port indiqués dans la partie **Machine [:port]** de l'URL

Ouvrir une communication TCP-IP, c'est créer un "tuyau" de communication entre deux machines. Une fois ce tuyau créé, toutes les informations échangées entre les deux machines vont passer dedans. La création de ce tuyau TCP-IP n'implique pas encore le protocole HTTP du Web.

2- Le tuyau TCP-IP créé, le client va faire sa demande au serveur Web et il va la faire en lui envoyant des lignes de texte (des commandes) au format HTTP. Il va envoyer au serveur la partie chemin/infos de l'URL

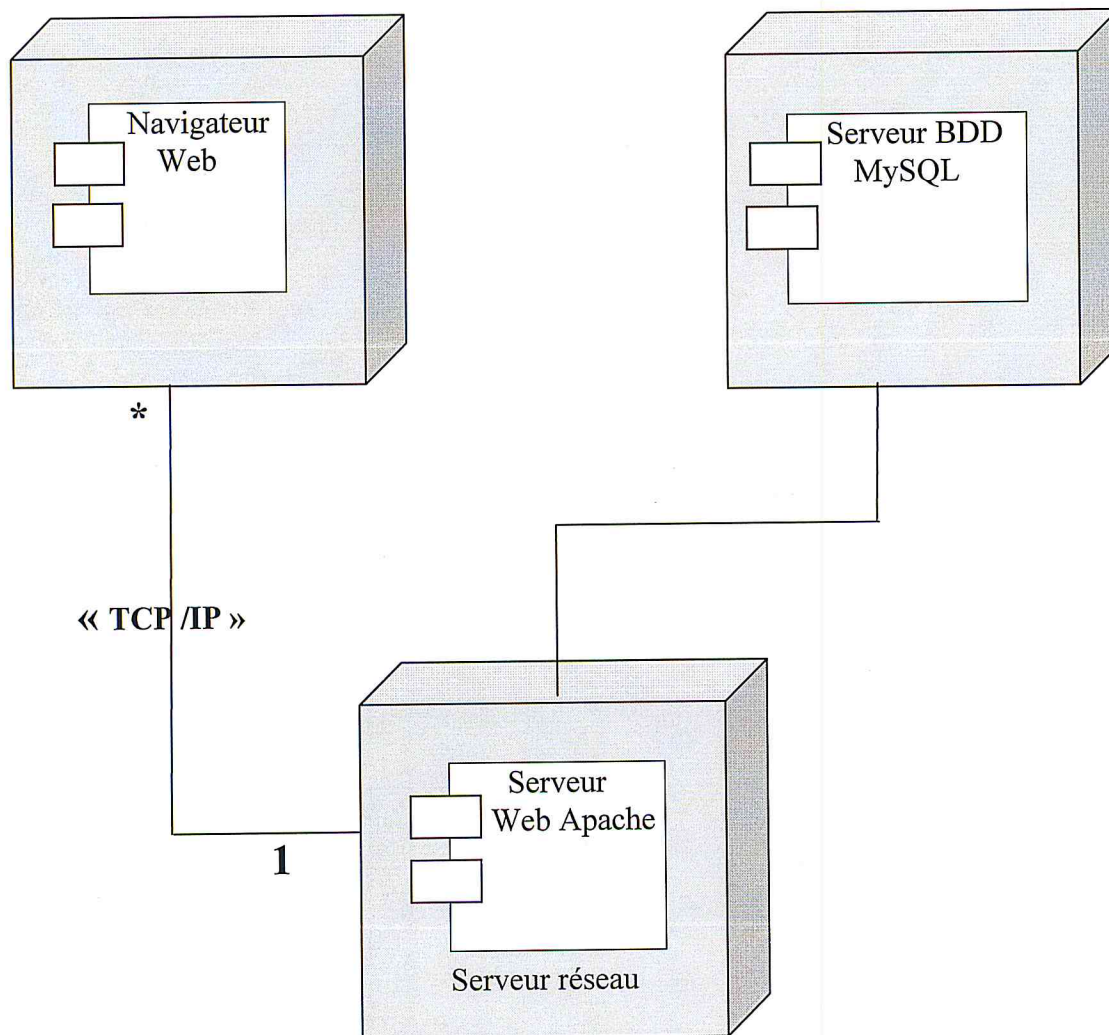
3- le serveur lui répondra de la même façon et dans le même tuyau

4- l'un des deux partenaires prendra la décision de fermer le tuyau. Cela dépend du protocole HTTP utilisé. Avec le protocole HTTP 1.0, le serveur ferme la connexion après chacune de ses réponses. Cela oblige un client qui doit faire plusieurs demandes pour obtenir les différents documents constituant une page web à ouvrir une nouvelle connexion à chaque demande, ce qui a un coût. Avec le protocole HTTP/1.1, le client peut dire au serveur de garder la connexion ouverte jusqu'à ce qu'il lui dise de la fermer. Il peut donc récupérer tous les documents d'une page web avec une seule connexion et fermer lui-même la connexion une fois le dernier document obtenu. Le serveur détectera cette fermeture et fermera lui aussi la connexion.

## 1.2. Diagramme de déploiement :

Les diagrammes de déploiement représentent la configuration des nœuds de processus en phase d'exécution ainsi que les composants qui y résident, ils présentent la vue de déploiement statique d'une architecture et sont liés aux diagrammes de composants dans le sens où un nœud renferme généralement un ou plusieurs composants.

Permet la représentation de la structure d'un système lors de son exécution et la relation entre composants logiciels et matériels



**Fig3 : Diagramme de déploiement avec composants**



### 1.3. HTML statique [10] :

Le type de page web le plus basique est totalement statique, textuel et écrit entièrement en HTML. Lorsque le client lance une requête HTTP pour cette page, le serveur se contente de renvoyer le texte qu'il trouve dans le fichier.

Le code HTML pur, statique présente certains avantages:

- n'importe quel navigateur est capable de l'afficher correctement
- la plupart des périphériques d'affichage conviennent parfaitement
- Chaque requête est traitée rapidement et utilise une quantité minimale de ressources
- Le code HTML est facile à comprendre ou à produire automatiquement
- on peut apporter rapidement de petites modifications aux pages

Les ajouts les plus répandus au HTML pur se trouvent sur le côté client, il s'agit en particulier d'extensions concernant la mise en forme des documents (feuilles de style en cascade, HTML dynamique) d'outils de scripts et des applets Java, toutes ces techniques sont ou ne sont pas selon le cas prises en charge par le navigateur web.

### 1.4. Scripts côté serveur [3] :

L'essentiel du travail d'écriture de scripts côté serveur consiste à connecter des sites web à des serveurs tel que des bases de données, cela autorise une communication à double sens:

- Du serveur vers le client, les pages web peuvent être assemblées à partir des résultats produits par le serveur de données

- Du client vers le serveur, les informations entrées par l'utilisateur peuvent être traitées en amont  
Les scripts côté serveur ne peuvent pas être visualisés côté client, toute l'interprétation du code est réalisée avant que celui-ci ne soit envoyé sur le réseau, une fois sorti du serveur il apparaît de l'autre côté comme s'il s'agissait de HTML classique, Parmi les exemples courants d'interaction client/serveur on peut citer les formulaires en ligne ainsi que certaines listes déroulantes, ces éléments sont pris en charge de façon dynamique sur le serveur

D'un autre côté, la plupart des aspects réellement utiles du web sont parfaitement pris en charge par des langages de script côté serveur tel que PHP:

- Contenus de sites (production et affichage)
- Communautés (forums, listes de diffusions, groupe de discussion...etc.)
- Courrier électronique (webmail, redirection)
- réseaux d'annonceurs
- applications professionnelles délivrées sur le web
- Annuaire et listes de membres
- Surveillance, tests et enquêtes
- remplissage et soumission de formulaires en ligne
- Personnalisation des informations
- Travail en groupe

- Catalogues, brochures et sites a vocation informative
- Par essence tout type d'application qui nécessite la connexion d'un serveur web a un serveur de données (base de données, courrier, LDAP,...etc.)

## 2. Gestion des pages :

On distingue deux types de pages : les pages de contenu et les pages de navigation, appelées aussi pages auxiliaires. Les pages auxiliaires sont utilisées pour faciliter la navigation de l'utilisateur sur le site. Les pages de contenu sont les pages qui présentent l'information recherchée par l'internaute. Cependant une page de contenu pour un internaute peut être une page de navigation pour un autre.

La première étape consiste à indexer les pages du site Web pour faciliter leur manipulation. L'exemple suivant illustre cette étape.

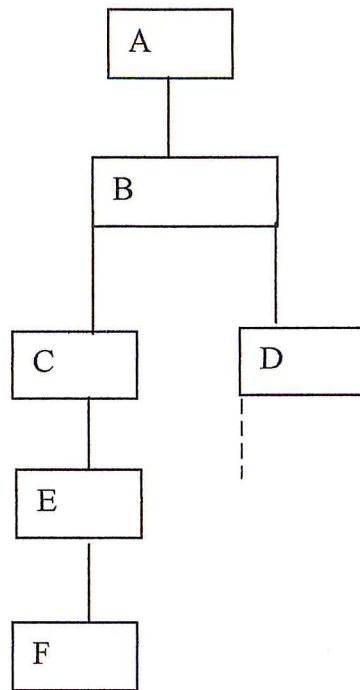
### Exemple de visite :

URL	Refferer
/site/index.htm	/haut.htm
/site/espace_chercheur/cherch_doc.htm	/site/main.htm
/site/servic_ligne/servic_ligne_doc.htm	/site/espace_chercheur/cherch_doc.htm
/site/servic_internet/servic_internet_doc.htm	/site/haut.htm
/site/telech/telech_doc.htm	/site/servic_internet/servic_internet_doc.htm

### Indexation des pages de visite :

URL	Index
/haut.htm	A
/site/index.htm	B
/site/main.htm	C
/site/haut.htm	D
/site/espace_chercheur/cherch_doc.htm	E
/site/servic_ligne/servic_ligne_doc.htm	F

Une fois les pages visitées par les internautes indexées, un arbre est construit. La racine de l'arbre est le premier referrer non externe dans le ...chier log. Pour chaque requête, si le referrer ou l'URL demandée n'est pas dans l'arbre, un noeud représentant la page est créé. Pour chaque couple de noeuds, si l'un est le referrer de l'autre alors un lien entre les deux noeuds est établi. Il en résulte la construction d'un graphe représentant la structure des hyperliens.



L'arbre représentant la topologie du site Web peut être traduit par une matrice. Cette représentation matricielle présente l'avantage de simplifier la représentation sous forme d'arbre qui devient incompréhensible dans le cas où le nombre de liens entre les pages est élevée. Chaque ligne de la matrice correspond à un noeud de l'arbre et représente une page du site. Il en est de même pour chaque colonne. Ainsi, s'il existe N pages différentes visitées par les internautes, la matrice d'hyperliens sera de dimension (N, N). Chaque entrée (i,j) de la matrice prend la valeur 1 si l'utilisateur a visité la page j à partir de la page i (présence d'un lien direct entre les deux pages) et la valeur 0 sinon. Cette matrice est utilisée pour calculer le nombre d'inlinks (nombre d'hyperliens qui mènent à la page en question à partir des autres pages) et le nombre d'outlinks (nombre d'hyperliens dans la page qui mènent vers d'autres pages). En effet, le nombre d'inlinks est le total sur les lignes alors que le nombre d'outlinks est le total sur les colonnes.

	A	B	C	D	E
A	0	0	0	0	0
B	1	0	0	0	0
C	0	1	0	0	0
D	0	1	0	0	0
E	0	0	1	0	0
F	0	0	0	0	1

On peut aussi construire une d'accès pour identifier la fréquence d'usage des pages et la fréquence d'usage des trajectoires, cette matrice permet de d'identifier les pages les plus visitées et les trajectoires les plus parcourus, Ces deux informations sont utiles pour la réorganisation des sites Web en vue de la personnalisation des services Web.

### 3. Choix d'une base de données:

Plus les sources de données sont importantes plus elles sont évolutives, et plus le web l'emporte sur les autres médias, l'un des avantages offerts par PHP sur des produits comparables est qu'il peut s'associer à un éventail intégrale de bases de données et s'y connecter facilement, il supporte de façon native un grand nombre de bases de données qu'elles soient publiques ou commerciales.

Pratiquement tout système de bases de données qui ouvre son API au public doit pouvoir être inclus, pour les autres il reste la solution du support ODBC générique.

MySQL est un système de gestion de bases de données (SGBD), selon le type de l'application sa licence est libre ou propriétaire il fait partie des logiciels de gestion de bases de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels.

MySQL fait partie du quatuor LAMP : linux, Apache, MySQL, PHP Il appartient également à ses variantes WAMP (Windows) et MAMP (Mac)

MySQL offre le meilleur compromis entre prix, rapidité, simplicité et efficacité, il offre pratiquement tout ce dont on peut avoir besoin et est disponible sous Unix et sous Windows même si la qualité des implémentations n'y est pas équivalente.

De plus la synergie entre PHP et MySQL est totale en particulier sous une plateforme Linux/Apache

#### 3.1. Définition des champs de la base de données:

##### Document

Champ	Type	Description
Code_doc	int(10)	Clé_primaire
Titre_doc	Varchar(50)	
Nature_doc	Varchar(20)	
Thème_doc	Varchar(50)	
M_clès	Varchar(15)	
Format_doc	char(4)	
Nbr_pages	int(4)	
Résumé_doc	Varchar(500)	
Emplac_doc	Varchar(100)	
Diff_doc	char(10)	
Code_type	int(10)	Clé_étrangère
Id_ens	int(10)	Clé_étrangère

**Utilisateur:**

<b>Champ</b>	<b>Type</b>	<b>Description</b>
Id_util	int (10)	Clé_primaire
Login_util	Varchar(15)	
Mot_pass	Varchar(10)	
Priv_util	char(10)	
Nom_util	Varchar(150)	
Prenom_util	Varchar(150)	
mail_util	Varchar(30)	
Adr_util	Varchar(200)	
tél_util	int(15)	
Id_group	int(10)	Clé_étrangère

**Type:**

Champ	Type	Déscription
Code_type	int(10)	Clé_primaire
Intitu_type	Varchar(100)	
Désc_type	Varchar(200)	

**Spécialité:**

Champ	Type	Déscription
Code_spec	int(10)	Clé_primaire
intitu_spec	Varchar(100)	

**Module:**

Champ	Type	Déscription
Id_module	int(10)	Clé_primaire
intitulé_mod	int(100)	

**Semestre:**

Champ	Type	Déscription
Code_seme	Varchat(10)	Clé_primaire

**Groupe**

Champ	Type	Déscription
Id_group	int(10)	Clé_primaire
intitul_group	Varchar(100)	

**Auteur:**

Champ	Type	Déscription
Code auteur	int(10)	Clé_primaire
Nom auteur	Varchar(150)	
Prénom auteur	Varchar(150)	

**Possède:**

Champ	Type	Déscription
Code_doc	int(10)	Clé_primaire
Code_Auteur	int(10)	Clé_primaire

**Contenir:**

Champ	Type	Déscription
Id_module	int(10)	Clé_primaire
Code_seme	int(10)	Clé_primaire

**Appartenir:**

Champ	Type	Description
Code_spec	int(10)	Clé_primaire
Id_module	int(10)	Clé_primaire

**Télécharger:**

Champ	Type	Déscription
Id_util	int(10)	Clé_primaire
Code_doc	int(10)	Clé_primaire

**Concerne:**

Champ	Type	Déscription
Id_module	int(10)	Clé_primaire
Code_doc	int(10)	Clé_primaire

La structure logique fondamentale d'une base de données SQL est simple, une installation SQL donnée peut contenir de multiples bases, chaque base de données contient une ou plusieurs tables, chaque table, à son tour est formée de colonnes soigneusement définies et les lignes de tables sont les enregistrements qui y sont insérés

Quatre instructions de manipulation de données sont supportées par tous les serveurs SQL, elles constituent un pourcentage très élevé de tout ce qu'on peut avoir à faire avec des bases de données relationnelles il s'agit de SELECT, INSERT, UPDATE et DELETE

Ces quatre commandes SQL manipulent uniquement des valeurs et non la structure de la base de données elle-même.

#### 4. Définir des permissions [2] :

Définir correctement les autorisations est un moyen pour se protéger, les pirates informatiques sont réputés pour savoir tirer profit des négligences, de l'ignorance, ou encore des événements de la vie de la société.

Les autorisations d'accès à une base de données se répartissent typiquement ainsi:

- Visiteur du site: uniquement SELECT
- Équipe chargée des nouveaux contenus : SELECT, INSERT et peut être UPDATE
- Équipe de maintenance : INSERT, UPDATE, peut être DELETE, peut être GRANT
- Équipe d'administration : SELECT, INSERT, UPDATE, DELETE, GRANT, DROP

DROP est la bombe atomique de SQL, car elle permet de vider totalement une base de données à partir d'une seule instruction, quelqu'un doit en porter la lourde responsabilité, et il ne peut s'agir que de l'administrateur principal du site.

Dans la plupart des bases de données, y compris MySQL, les mots de passe sont cryptés par d'autres algorithmes que ceux liés au système, ils sont évidemment enregistrés à un endroit totalement différent, même si l'un est découvert cela ne veut pas dire que l'autre est devenu vulnérable alors il faut prendre le temps de définir correctement les autorisations, de choisir de bons mots de passe

Les bases de données peuvent enregistrer directement des noms d'utilisateur et des mots de passe, ce qui peut permettre de les contrôler à partir d'un script php, cela peut concerner le nom d'utilisateur et le mot de passe pour l'accès au système, plus les permissions accordées aux utilisateurs sur la base de données

#### 5. Recherche documentaire [7] :

Une utilisation courante de l'informatique est le stockage de données présentant des caractéristiques communes, et la recherche, parmi ces données, d'éléments satisfaisant certains critères (valeur d'une clé). En général, le critère de recherche porte sur la valeur d'une clé

Cette clé peut aussi être simplement la valeur de l'objet (recherche d'un entier dans une liste d'entier). Les algorithmes de recherche peuvent aussi varier selon qu'il existe ou non plusieurs éléments ayant la même clé.

Il faut alors décider si la recherche doit rendre toutes les solutions ou une solution quelconque. Ici, nous ne retiendrons qu'une solution quelconque. Pour avoir toutes les solutions, il suffit de relancer l'algorithme sur le reste de l'ensemble des données. Cela ne pose en général pas de problèmes.



On cherche des algorithmes de recherche rapides. Mais il faut aussi que les adjonctions et suppressions d'éléments soient rapides. Il faut donc discuter également de la structure de donnée.

Les algorithmes de recherche que nous allons présenter retournent un élément quelconque ayant une clé de la valeur cherchée. Il est parfois plus utile de retourner la place de cet élément dans l'ensemble des données pour par exemple effectuer ensuite une suppression. Il n'est pas difficile de modifier les algorithmes en ce sens.

Sur des structures de données usuelles comme les listes, les tables ou les arbres, il existe des algorithmes bien connus que l'on peut facilement mettre en œuvre. Ces algorithmes exploitent les propriétés de la structure de données et du domaine.

Un exemple classique est la recherche dichotomique où l'on divise en deux l'espace de recherche à chaque tentative ce qui donne une complexité logarithmique donc très avantageuse.

Pour des problèmes complexes, la recherche de solutions relève de l'intelligence Artificielle

On dit qu'un algorithme est de recherche par force brute lorsque toutes les entrées sont vérifiées une à une. Ce type de recherche peut s'avérer efficace si l'espace des solutions est d'une taille raisonnable vis à vis de la puissance de la machine utilisée pour le parcourir. Il s'agit donc d'une méthode à tenter en dernier recours s'il n'y a pas d'autres possibilités.

On parle de recherche heuristique lorsque des connaissances ou des propriétés supplémentaires permettent de rendre la recherche plus efficace. Ainsi, l'exploitation des symétries géométriques dans la résolution d'un puzzle permet de réduire fortement l'espace de recherche. De la même façon, une recherche de chemin peut être facilitée par la connaissance même approximative de la direction dans laquelle se trouve l'objectif ou de sa distance.

Le problème de la recherche

Un tableau d'éléments de type  $E$ ,  $x \in E$

But déterminer si  $x \in t$

- Le résultat est de nature booléenne
- Vrai si  $x \in t$  : recherche avec succès
- Faux si  $x \notin t$  : recherche avec échec

Il faut aussi être accompagné d'un indice  $i \in [a..b]$  tel que  $t[i]=x$  s'il en existe.

Dans la recherche séquentielle il faut parcourir les éléments de  $t$  un à un à la recherche de  $x$

Par ordre croissant ou décroissant des indices jusqu'au premier indice  $k$  tel que  $t[k]=x$

Ou bien jusqu'à avoir parcouru l'intervalle  $[a, b]$  en entier sans trouver  $x$

Hypothèse : Soit  $k \in [a, b]$  et supposons que  $x \in t[a..K-1]$

Si  $t[k] = x$ , alors recherche terminée avec succès

Si  $t[k] \neq x$ , alors passer à l'élément suivant

S'il y en a un

Sinon recherche terminée avec échec

```

k := a
{x ∈ t [a..k - 1] }
Tant que k ≤ b et t[k] ≠ x faire
    {x ∈ t [a..K]}
    inc(k)
    {x ∈ t [a..k - 1] }
fin tant que
{(k > b et x ∈ t),
Ou bien (k ≤ b et t[k] = x)}
si k > b alors

```

Recherche avec échec

**Sinon**

Recherche avec succès  
ET k est le plus petit indice tq t[k] = x

Nombre de comparaisons (t[k] = x) effectuées dans la recherche Séquentielle dans un tableau de n éléments

- pour une recherche qui échoue :  $c(n) = n$
- pour une recherche qui réussit, cela dépend

$C(n) = 2$  si  $t[1] = x$  : meilleur des cas

$C(n) = n$  si  $t[n] = x$  : pire des cas

$1 \leq c(n) \leq n$  dans tous les cas

Il y'a aussi des algorithmes de recherche dichotomiques qui permettent de rechercher dans une moitié du tableau.

Les éléments du tableau sont deux a deux distincts et qui sont rangés dans l'ordre croissant

1 addition, 1 division, entre 1 et 3 tests et un appel récursif et aussi division par deux de la taille du tableau.

Alors l'ordre de complexité est de :  $O(\log_2(n))$

On a vu que les algorithmes précédents traitent les tableaux a une dimension et vu que notre travaille va se situer autour d'une base de données nous allons les appliquer et les adaptés pour qu'ils puissent être opérationnels

La recherche va s'effectuer essentiellement sur les documents, les utilisateurs et aussi sur les autres tables.

Vu que les documents en questions soient stockés dans des répertoires la recherche va se faire en utilisant les métas donnés et les mots clés qui sont mémorisés dans la base de données

## 6. Implémentation

### 6.1. Scripte d'ajouter un document

<?php

```

/*****connexion à la base de données*****/
        mysql_connect("localhost","root","") or die ('Erreur de connexion');
        mysql_select_db("sharedoc") or die ('Erruer lors de selection de la base de donnée');
/*****/

$Code_doc = $_POST['Code_doc']; $Titre_doc = $_POST['Titre_doc'];
$Nature_doc = $_POST['nat_doc']; $image = $_POST['Livres'];
$resol = $_POST['date_pub']; $editeur = $_POST['editeur'];
$date_pub = $_POST['date_pub']; $Thème_doc = $_POST['Thème_doc'];
$Mots_doc = $_POST['Mots_doc']; $Format_doc = $_POST['Format_doc'];
$Nbr_doc = $_POST['Nbr_doc']; $Résumé_doc = $_POST['textarea'];
$Emplac_doc = $_POST['semestre']; $semestre = $_POST['semestre'];
$Diffusion_doc = $_POST['choix']; $date_p = $_POST['date_pub'];
$groupe = $_POST['group']; $name = $_FILES ['doc']['name'];
$taille_maxi = 100000000;
$taille = filesize($_FILES['doc']['tmp_name']);
$extensions = array('.png', '.gif', '.jpg', '.jpeg', '.pdf', '.doc', '.ppt', '.txt', '.rar', '.zip');
$extension = strrchr($_FILES['doc']['name'], '.');
//Début des vérifications de sécurité...
if(!in_array($extension, $extensions)) //Si l'extension n'est pas dans le tableau
{
    $erreur = 'Vous devez uploader un fichier de type png, gif, jpg, jpeg, txt, rar, zip, ou doc...';
    echo $erreur;
}
if($taille>$taille_maxi)
{
    $erreur = 'Le fichier est trop gros...';
    echo $erreur;
    echo      '<a      href="javascript:history.back()"><font      color="#99FF00"
size="5"><strong>Retour</strong></font></a>';
}
if(!isset($erreur)) //S'il n'y a pas d'erreur, on upload
{

if(isset($_FILES['doc']))
{
    if ($Diffusion_doc=="Privé"){
    $dossier = "Base des documents/Privé/$groupe/$semestre/";
    $fichier = basename($_FILES['doc']['name']);
    $result1=mysql_query("INSERT INTO document VALUES
('$Code_doc','$Titre_doc','$Nature_doc','$Thème_doc','$Mots_doc','$Format_doc','$name','$Nbr_doc','$Résumé_doc','$Emplac_doc','$groupe','$date_p','$Code_doc')") or die ("votre identificateur exist déjà choisi un
autre S.V.P      ?????      <a      href='dipotdocument.php'><font      color=#99FF00"
size='5'><strong>Retour</strong></font></a>");

```

```

echo '<br>';

if(move_uploaded_file($_FILES['doc']['tmp_name'], $dossier . $fichier))

{
    echo '<br><br><center><b><font color=verdana>transfert effectue avec succes! </font></b> ';
        echo'<ahref="javascript:history.back()"><font
color="#99FF00"size="5"><strong>Retour</strong></font></a></center>';
    }
    else //Sinon (la fonction renvoie FALSE).
    {
        echo '<center><b><font color=verdana>Echec du transfert !</font></b> <a
href="javascript:history.back()"><font color="#99FF00" size="5"><strong>Retour</strong></font></a>
</center>';
            // <a href="dipotdocument.php"><font color="#99FF00"
size='5'><strong>Retour</strong></font></a></center>';
        }
    }
    else
        if ($Diffusion_doc=="Public"){
            $dossier = "Base des documents/Public/";
            $fichier = basename($_FILES['doc']['name']);
            $result1=mysql_query("INSERT INTO document VALUES
('$Code_doc','$Titre_doc','$Nature_doc','$Thème_doc','$Mots_doc','$Format_doc','$name','$Nbr_doc','$Résumé_doc','$Emplac_doc','$Diffusion_doc','$date_p','$Code_doc')") or die ("votre identificateur existe déjà
choisi un autre S.V.P ????? ");
            echo '<br>';
            if(move_uploaded_file($_FILES['doc']['tmp_name'], $dossier . $fichier))
            {
                echo '<br><br><center><b><font color=verdana>transfert effectue avec succes!
</font></b></center>';
            }
            else //Sinon (la fonction renvoie FALSE).
            {
                echo '<center><b><font color=verdana>Echec du transfert !</font></b></center>';
            }
        }
    }
}
}
?>

```

## 6.2. Scripte d'ajouter un utilisateur

&lt;?php

```

    $id=$_POST['id'];
    $nom=$_POST['nom'];
    $pre=$_POST['prenom'];
    $tel=$_POST['tél'];
    $adr=$_POST['adr'];
    $email=$_POST['email'];
    $log=$_POST['login'];
    $mdp=$_POST['password'];
    $choix=$_POST['choix'];
    $date = $_POST['date'];
        /*****connexion à la base de données*****/
            mysql_connect("localhost","root","") or die ("Erreur de connexion");
            mysql_select_db("sharedoc") or die ("Erreur lors de selection");
        /*****/

$result1=mysql_query("insert into utilisateur values
('$id','$log','$mdp','$choix','$nom','$pre','$date','$email','$adr','$tel','$id')") or die ("votre identificateur exist
déjà choisi un autre S.V.P ????? ");

//$result2= mysql_query("select Id_group from group where Intitul_group= $choix") or die ("requete2");

//$result2=mysql_query("insert into groupe values ('1','$choix')") or die ("erreur de requete2");
if ($result1==true ){

    echo"<center>Merci vous information sont enregistrer avec succès!!! <br>
    <center><a href='index.php'>Reteur</a></center>
</center>";}
else {
    echo"<center>Erreur d'enregistrement <br>
    <center><a href='javascript:history.back()'>Reteur</a></center>
</center>"; }
?>

```

## 6.3. Scripte de recherche

&lt;?php

```

echo"<fieldset><legend><font color='#FF33CC' size='5'><strong><em>Résultat de la
recherche</em></strong></font></legend>";
//echo"<center><table border='2'>";
$refcode=$_POST['ref'];
    /*****connexion à la base de données*****/
        mysql_connect("localhost","root","") or die ("Erreur de connexion");
mysql_select_db("sharedoc") or die ("Erreur lors de selection de la base de donnée");
    /*****/
    $res = mysql_query("SELECT * FROM document where Nom_doc like'$refcode%'");
    if ($res==true){
        // echo"<tr bgcolor='#CCCCCC'><th>Année de publication<th>Nombre de page<th>Type
Document<th>Resume</tr>";
        $nbr=1;
        while($row = mysql_fetch_assoc($res)){
            $nam="$row[Nom_doc]";
                //echo"<tr>";
                //echo"<td>$tuple[Code_D]</td>";
                echo"<a href='Base des documents/Public/$nam' title='Télécharger le
document'>$row[Nom_doc]</a> <br>";
                echo" <font color='#6699FF' size='3'><strong><em>Date de publication :</em></strong></font>
$row[date_pub] <br/>";
                echo" <font color='#6699FF' size='3'><strong><em>Nombre de page :</em></strong></font>
$row[Nbr_pages] <br/>";
                echo"<font color='#6699FF' size='3'><strong><em>Résumé
:</em></strong></font>$row[Résumé_doc]<br/>";
                /* echo "<td bgcolor='#CCFFFF'>$row[date_pub]</td>";
                print "<td bgcolor='#CCFFFF'>$row[Nbr_pages]</td>";
                print "<td bgcolor='#CCFFFF'>$row[Nature_doc]</td>";
                print"<td bgcolor='#CCFFFF'>$row[Résumé_doc]</td>";
                print"<td bgcolor='#CCFFFF'><a href='Base des documents/Public/$nam' title='Télécharger le
document'>$row[Nom_doc]</a></td>";*/
            }
            //echo"</tr>";
        }
        else {
            //echo"<fieldset><legend><font color='#FF33CC' size='5'><strong><em>resultat de la
recherche</em></strong></font></legend>";
            //echo"<table border='2' ><tr bgcolor='#CCCCCC'><td><div align='center'><font color='#33FF99'
size='5'><strong><em>Pas de resultat
//</em></strong></font> </div></td></tr></table>";
            echo"Pas de resultat " ;
            //echo"</tr>";
        }
        echo"</fieldset>";

?>

```

## 6.4. Interface de l'application

### Page d'accueil : index.php

**Share**  
**Doc** Bienvenue dans Share Doc

Share Doc est un service Web ayant pour objectif de rendre les échanges plus intuitifs, efficaces et utiles. Et qui sait...

Connectez-vous au service Share Doc  
à l'aide de votre **Compte**

Nom d'utilisateur :

Mot de passe :

[Oblier le mot de passe](#)

Nouveau chez Share Doc? C'est  
gratuit et facile

## Page d'inscription : inscription.php (ajouter un utilisateur)

Share  
Doc

Bienvenue dans Share Doc

[Retour](#)

\*\*\*\*\*Bienvenue dans l'espace d'inscription\*\*\*\*\*

## Bonjour !

Nous allons vous aider à créer un compte :  
il vous suffira de répondre à quelques questions très simples  
et de spécifier l'identifiant et le mot de passe de votre choix

### 1. Présenter vous

Votre identifiant

Nom et Prénom :

Date de naissance :

Adresse

:

Téléphone

:

### 2. À quel groupe vous appartient

Choisi votre groupe : 

### 3. Login et Mot de passe

Login :

Mot de passe :

Confirmer mot de passe :

### Adresse email

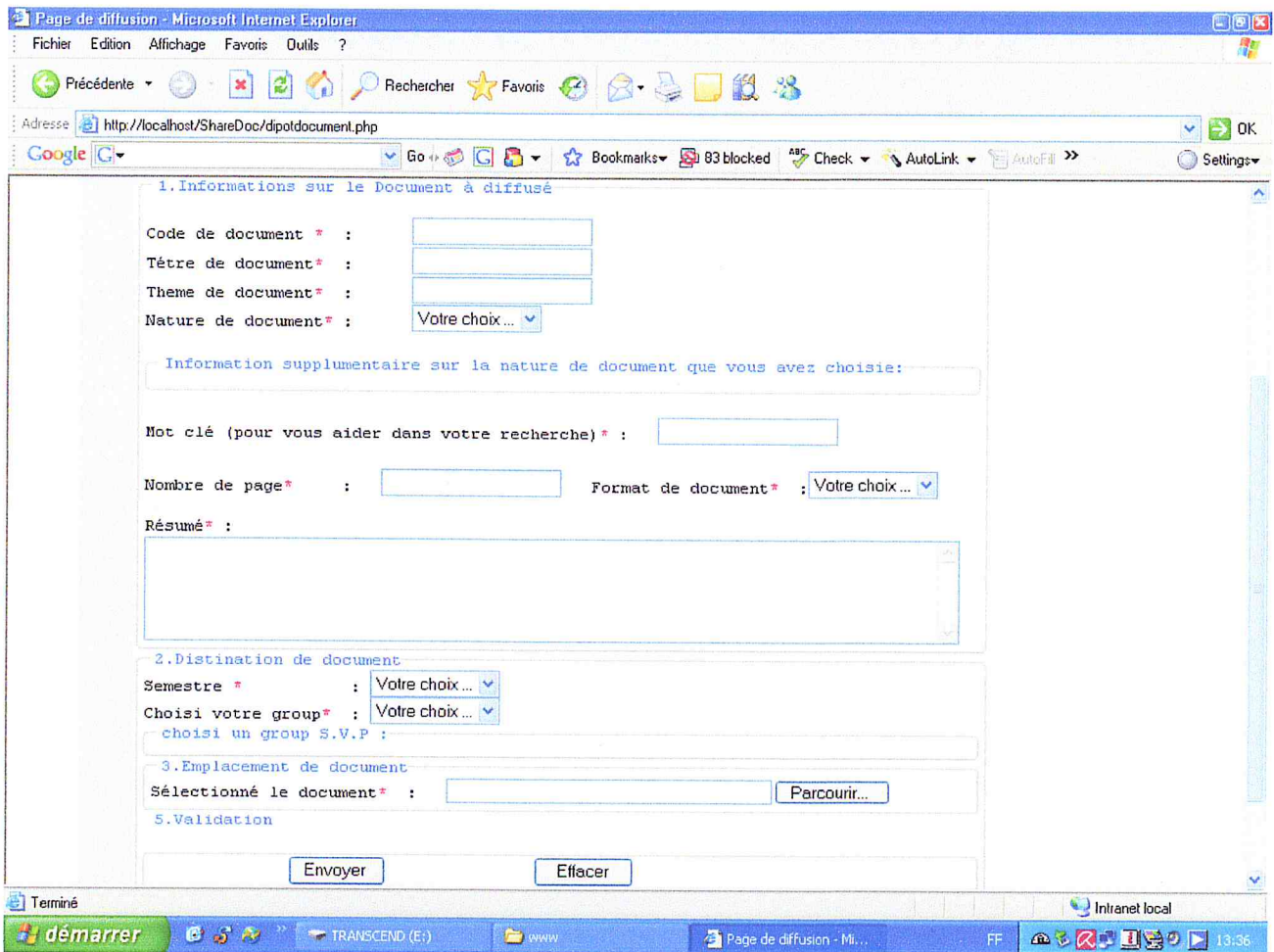
4. Votre adresse email :

**Remarque** : respecter la syntaxe S.V.P

### 5. Validation



Page de diffusion : dipotdocument.php



## **Conclusion générale :**

Notre travail permet d'offrir une meilleure lisibilité et compréhension des pratiques dans le domaine de la gestion documentaire

La conservation des documents numériques sur le long terme est un problème ouvert. Le Manque de recul est certain. Les formats dont les spécifications ne sont pas connues posent Problème néanmoins les questions de formalisme ne vont pas résoudre tout les problèmes. Il Fallait rappeler ici l'importance pour la conservation des documents numériques des aspects Matériel et logiciel.

Aujourd'hui il ne faut pas trop penser en termes de format pérenne. Les formats ne sont pas Pérennes, les logiciels non plus et le matériel encore moins. Mais il faut penser en terme de Migration de fichiers, migration du matériel et en termes de conversion de formats. Les Difficultés de conservation sont en fait intrinsèque à cette même technique qui permet tant de Prodige pour la création ou la diffusion.

Quel est l'avenir ? En fait il est très difficile de deviner les évolutions, les miracles possibles. Certains aspects qui nous semble aujourd'hui inquiétants ne le seront plus demain, non pas Que les problèmes auront été résolu mais simplement ne seront plus d'actualités ou bien Auront été contournés.

## Bibliographie :

### Livres

- [1] Jim Conallen, «*Concevoir des applications web avec UML*», Eyrolles, Octobre2000, 300 p
- [2] Nicolas Rivereau, Antoine Pichot, «*MySQL4*», Micro Application, 2003, 984 p
- [3] Tim Converse, Joyce Park, «*PHP4 Totale initiation*», OEM, 2000, 686 p
- [4] Guillaume Rossolini, «*Cours de PHP5* », <http://www.developpez.com/>.
- [5] Hugo Etievant, «*Cours complet de MySQL*», <http://www.developpez.com/>.
- [6] Jakob Nielson, «*Conception de sites web : L'Art de la simplicité*», CampusPress,2000,385 P
- [7] Jean Bézivin, «*Les bases de l'informatique et de la programmation*»,  
<http://www.polymorphe.com/>.

### Publications

- [8] Jean Bézivin, «*UML2 : Ingénierie des modèles logiciels MDA et MOF*»,
- [9] Frédéric Vivien, «*Algorithmique*», <http://www.polymorphe.com/>.
- [10] Josselin Willette, «*les-bases-du-html[1]*», <http://www.developpez.com/>.
- [11] Serge P, «*Introduction au javascript*», <http://www.developpez.com/>
- [12] Hendrik Busch, Christian frank, Michael Gardoni «*Partage de Documents au sein d'une équipe a partir d'une caractérisation de leurs contenus selon des points de vues multiples*», Autrans, janvier 2004, 13 p.

### Site internet

- [13] <http://fr.wikipedia.org/>
- [14] <http://www.evido-ged.com/>
- [15] <http://www.maarch.com/>
- [16] <http://cyberzoide.developpez.com/securite/privileges-base-de-donnees/>
- [17] <http://www.polymorphe.com/>.



