



REPUBLIQUE ALGERIENNE DEMOCRATIQUE
ET POPULAIRE MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE MOULOD MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT INFORMATIQUE

Mémoire de fin d'étude

En vue d'obtention d'un diplôme de Master en informatique

Thème :

**Le Text Mining de questions pour le
tutorat LMD**

Proposé et dirigé par: Mme BOUARAB.F

Réalisé par:

M^{elle} KEBAILI Kahina.

M^{elle} MEDJKOUH Nabila.

Année: 2013/2014

Remerciements



Nous tenons à remercier notre promotrice Mme BOUARAB de nous avoir encadrés et guidés dans la réalisation de notre thème.

Nos vifs remerciements pour les membres de jury, qui nous ont fait l'honneur d'accepter de juger notre travail.

Nos sincères sentiments vont à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce projet.





DEDICACES

Je dédie ce modeste travail à :

La mémoire de mon frère Ferhat.

Mes très chers parents.

Mon frères Nabil et ma sœur Sabrina.

Mon très chère futur époux Madjid et sa famille.

Ma binôme et sa famille.

Tous mes amis (es) ainsi leurs familles.

MEDJKOH Nabila



Dédicaces

Je dédie ce modeste travail à :

Mes chers parents.

Mes frères, leurs femmes et leurs enfants.

Mes sœurs, leurs maris et leurs enfants.

Ma binôme et sa famille

Tous mes amis (es) en particulier "massinissa"

KAHINA Kebaili

Ü Liste des figures et tableaux

Figure I.1 : processus du text mining	4
Figure I.2 : classification des métriques de comparaison des chaines de caractères	10
Figure III.1 : démarche de modélisation de l'application.....	25
Figure III.2 : spécification des taches	27
Figure III.3 : spécification des scénarios	29
Figure III.4 : Diagramme préliminaire des cas d'utilisation.....	31
Figure III.5 : Diagramme des cas d'utilisation : «tuteur »	33
Figure III.6 : Diagramme de cas d'utilisation « Etudiant».....	34
Figure III.7 : Diagramme de séquence du cas d'utilisation « s'authentifier».....	35
Figure III.8 : Diagramme de séquence du cas d'utilisation « Poser question ».....	36
Figure III.9 : Diagramme de séquence du cas d'utilisation « Répondre à une question » ..	36
Figure III.10 : Diagramme d'activité pour le cas d'utilisation «s'authentifier»	38
Figure III.11: Diagramme d'activité pour le cas d'utilisation «Poser une question»	39
Figure III.12 : Diagramme d'activité pour le cas d'utilisation «Répondre aux question» ..	40
Figure III.13 : Diagramme de classe générale de cas d'utilisation «Authentification»	41
Figure III.14: Diagramme de classe général de cas d'utilisation «poser une question»	42
Figure III.15: Diagramme de classe de cas d'utilisation «Répondre aux question»	43
Figure III.16 : Diagramme de classe global	44
Figure III.17 : organigramme globale du système	50
Figure III.18 : description du l'analyse linguistique de la question	53
Figure IV.1 : exemple de fichier HTML	61
Figure IV.2 : exemple du code Style CSS	65
Figure IV.3 : exemple de script jQuery	63
Figure IV.4 : interface "patern MVC"	67
Figure IV.5 : interface "Netbeans 8.0"	69
Figure IV.6 : interface "MySQL Workbench"	70

Figure IV.7 : interface d'accueil.....	71
Figure IV.8 : interface "espace Etudiant"	72
Figure IV.9 : interface "mes question précédente"	73
Figure IV.10 : <i>interface "poser question.exemple 1"</i>	74
Figure IV.11 : <i>interface "poser question.exemple 2"</i>	75
Figure IV.12 : <i>interface "poser question.exemple 3"</i>	76
Figure IV.13 : <i>interface "poser question.exemple 4"</i>	76
Figure IV.14 : <i>interface "poser question.exemple 5"</i>	76
Figure IV.15 : <i>interface "poser question.exemple 5"</i>	77
Figure IV.16 : <i>interface "poser question.exemple 5"</i>	78
Figure IV.17 : <i>interface "poser question.exemple 5"</i>	78
Figure IV.18 : interface "espace Tuteur"	79
Figure IV.19 : interface "Question avec réponse"	80
Figure IV.20 : interface "Question sans réponse"	81

ü Sommaire

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

Chapitre I : Text Mining

Introduction.....	3
I.1. Historique	3
I.2. Définition	4
I.3. Vue d'ensemble d'une opération de Text Mining	5
I.4. Les approches du Text Mining	5
I.4.1. L'approche statistique	5
I.4.2. L'approche sémantique	6
I.5. sous tâches du Text Mining	7
I.5.1. Analyse linguistique.....	7
I.5.2 Recherche d'information.....	8
I.5.2.1 Indexation	8
I.5.2.2 Modèle vectoriel.....	8
I.5.2.3. Notion de similarité.....	9
I.5.2. 4 Les métriques de similarité utilisée.....	9
I.5.2.5 Extraction d'information.....	10
I.5.3 Fouille de données (datamining)	11
I.6. Différence entre text mining et data mining	12
I.7. Domaines d'application et les utilisateurs du text mining	13
Conclusion	14

Chapitre II : Le tutorat LMD et les TICS

Introduction	15
II.1. Le tutorat	15
II.1.1 Qu'est-ce que le tutorat	15

ü Sommaire

II.1.2. L'objectif du tutorat	16
II.1 .3. Le tuteur	16
II.1.4. Le rôle du tuteur	16
II.2. Le tutorat dans le système LMD	17
II.2.1. Aspect informatif et administratif	18
II.2.2. Aspect pédagogique	18
II.2.3. Aspect méthodologique et technique	18
II.2.4. Aspect Psychologique	18
II.3. Les avantage du tutorat dans le système LMD.....	19
II.4. Terminologie	19
II.4.1. les TICE.....	19
II.4.2. Apprenant	19
II.4.3. L'apprentissage	19
II.4.4. Formateur	20
II.5. Outils utilisés	20
II.5.1. Outils techniques	20
II.5.1.1. Outils de communications synchrones	20
II.5.1.2. Outils asynchrones	20
II.5.2. Outils pédagogiques	21
Conclusion	23

Chapitre III : Analyse et conception

Introduction.....	24
III.1. Description générale de la démarche de conception	24
III.2. Analyse et conception de notre système	25
III.2.1. Analyse	25

ü Sommaire

III.2.1.1. Problématique	25
III.2.1.2. Modèle des cas d'utilisation	26
III.2.1.3. Identification des acteurs	26
III.2.1.4. Spécification des tâches	26
III.2.1.5. Spécification des scénarios.....	27
III.2.1.6. Identification et description des cas d'utilisation	29
III.2.1.7. Ajout des relations entre cas d'utilisation	31
III.2.1.8. Finalisation des diagrammes de cas d'utilisation	32
III.2.2. Conception	34
III.2.2.1. La conception d'application web.....	34
III.2.2.2. Elaboration des diagrammes de séquence.....	35
III.2.2.3. Diagramme d'activité	38
III.2.2.4. Diagramme de classe.....	41
III.2.2.5. Conception de la base de données	44
III.2.2.5.a. Le modèle physique des données.....	45
III.2.6. Organigramme, algorithme et procédures du système	48
III.2.6.1. Organigramme global du système	48
III.2.6.2. description des procédures et API utilisés	51
III.2.6.3. Description de l'analyse linguistique de la question	53
III.2.6.4. L'algorithme global	54
Conclusion	60

ü Sommaire

Chapitre IV : Réalisation

Introduction.....	61
IV.1 Outils utilisés	61
IV.1.1 Langages à balises.....	61
IV.1.2 Langages de script.....	62
IV.1.3 Serveurs	63
IV.2 Environnement de développement	64
IV.2.1 langage de programmation java	64
IV.2.2 Architecture MVC.....	65
IV.2.2.1 Avantages de l'architecture MVC	68
IV.2.2.2 Différence avec l'architecture trois tiers	68
IV.3 Les outils de développement.....	69
IV.3.1 IDE NetBeans	69
IV.3.2 Le SGBD (MYSQL 5)	70
IV.3 Présentation de quelques interfaces.....	71
IV.3.1 Page d'accueil.....	71
IV.3.2 Espace étudiant	72
IV.3.3 Espace tuteur.....	79
Conclusion	82
Conclusion générale.....	83

Annexe

Annexe.....	84
-------------	----

Bibliographie

Bibliographie	96
---------------------	----

Introduction générale

L'enseignement à distance fait partie des Technologies de l'Information et de la Communication pour l'Éducation (TICE) et permet de réaliser des activités non présentesielles. Il s'agit le plus souvent de l'utilisation d'ordinateurs ou d'appareils mobiles (smartphones, tablettes, Personal digital assistant, etc.) connectés à internet.

L'un des objectifs de l'enseignement à distance est d'améliorer la qualité de l'enseignement en ligne en introduisant un mode collaboratif de création et de diffusion des savoirs entre enseignants et étudiants et non de se substituer aux modes traditionnels d'enseignement. Les moyens pour atteindre cet objectif sont multiples, complémentaires et indépendants.

Le volume croissant de données textuelles provenant de l'internet, des intranets, des forums de discussion, du nombre très important des apprenants propose une quantité d'informations potentiellement pertinentes mais qui ne sont pas exploitables manuellement. Le traitement de gros volumes de ces données textuelles s'est fait ressentir dans les plateformes par les tuteurs. L'utilisation de ces informations est très importante et l'utilisation du text mining devient incontournable dans les plateformes e-learning.

Le text mining est une technique permettant d'automatiser le traitement de gros volumes de contenus texte pour en extraire les principales tendances et répertorier de manière statistique les différentes données.

Les techniques de text mining sont surtout utilisées pour des données déjà disponibles au format numérique ;

L'objectif de notre travail est d'intégrer un moyen ou une technique de similarité dans un environnement e-learning (forum). Cette technique va permettre d'analyser et de calculer le degré de ressemblance entre les questions saisies par les apprenants et celles de la base de données, ainsi si la réponse de la question existe on l'affiche directement;

Cette technique a beaucoup d'avantages :

- Combinaison du mode synchrone avec le mode asynchrone ;
- Moyen rapide de la diffusion de l'information ;
- Aide pour l'apprenant qui lui permet de retrouver facilement des réponses à ses questions ;
- Réduction des questions dans la base de données ;
- Réduction des tâches répétitives du tuteur ;

Pour comprendre l'intérêt de l'intégration du text mining dans le e-learning, nous avons décomposé en cinq chapitres principaux :

Ø Chapitre 1 : Ce chapitre fait l'objet de l'automatisation du traitement de données textuelles. Cela grâce à la technique text mining. Cette technique repose sur les algorithmes de la théorie linguistique dans les systèmes informatiques d'apprentissage et de statistiques

Ø Chapitre 2 : Dans ce chapitre, nous allons décrire le tutorat dans le contexte des TICE. Nous mettrons plus particulièrement l'accent sur les rôles, fonctions et compétences des tuteurs dans la formation à distance. Par la suite, on passera en revue les différents outils techniques utilisés.

Ø Chapitre 3 : Ce chapitre est consacré à la démarche méthodologique à suivre afin de planifier et concevoir l'application d'une façon rigoureuse tout en mettant en évidence tous les objectifs fixés en amont du projet et veiller à son élaboration.

Ø Chapitre 4 : Dans ce chapitre, nous présentons les différentes étapes de l'implémentation de notre application. nous avons consacré une première partie au choix de langage de développement, l'environnement et les outils d'implémentation de notre application.

Ensuite dans une partie nous avons mis en oeuvre des différentes interfaces de notre site (application), et leurs fonctionnalités essentielles et ainsi que des exemples du code utilisé.



Chapitre I : Text Mining

Introduction :

Depuis quelques années, et grâce aux systèmes de communications modernes, les échanges d'informations entre les individus ont été facilités, et ont permis une large diffusion des connaissances, mais se sont toutefois accompagnés d'une prolifération des sources d'information sur Internet. Des millions de documents électroniques, pour la plupart textuels, sont ainsi facilement accessibles mais ont, cependant, engendré une situation d'abondance et d'hétérogénéité de l'information à travers les réseaux.

Ce contexte de « surinformation » conduit les utilisateurs, aussi bien les entreprises, les organismes, que les individus à passer plus de temps à gérer la « surcharge d'information » que l'information elle-même. Aussi, il devient de plus en plus ardu de dégager les informations pertinentes enfouies dans les textes, sans l'aide de nouveaux moyens qui seraient capables d'organiser, de structurer des volumes importants d'informations, et d'offrir des possibilités d'exploration très fine.

Partant de ce constat, des outils spécifiques permettant d'analyser le contenu des documents, de les organiser et de les représenter automatiquement ont vu le jour. Ces nouveaux outils émergent dans le cadre de dispositifs de veille visant la collecte, le traitement, la diffusion et l'exploitation de l'information utile. Ainsi, le text mining ou « fouille de textes » est venu répondre à ces nouvelles exigences en matière de traitement de l'information.

De ce fait, le text mining utilise des techniques linguistiques et statistiques, destinées à analyser automatiquement et à synthétiser de grandes quantités d'informations textuelles, et ce, en vue d'en extraire l'information pertinente et faciliter la recherche pour la prise de décision.

I.1. Historique :

La fouille de textes, traduit du terme anglais textmining, est apparue dans la deuxième moitié des années 90, en écho à des travaux réalisés depuis les années 80 sur des bases de données. En 1991, Piatetsky-Shapiro introduit comme titre de son ouvrage [PIA 91] le terme de Knowledge Discovery from Databases, abrégé par la suite en KDD et, dont l'équivalent français est Extraction de Connaissances à partir de Bases de Données (ECBD). Ce n'est que vers 1995 que l'usage des termes Knowledge Discovery from Databases et Data Mining se précise. L'extraction de connaissances à partir de bases de données désigne alors le processus global de découverte de connaissances qui permet de passer de données brutes à des connaissances alors que la fouille de données n'est qu'une étape de l'ECBD au cours de laquelle un modèle est construit.

Initialement proposée par [PIA 91], la définition de l'ECBD a été enrichie au cours du temps et celle proposée dans [FAY 96b] est maintenant consensuelle:

I.2. Définition:

Le Text mining, également appelé fouille de textes ou extraction de connaissances, est un Domaine de recherche dont la première définition est donné par (R. Feldman, 1995).

Le Text Mining, qui est une des disciplines du traitement automatique du langage naturel (TALN), est en pleine expansion car il permet de traiter un volume important de données textuelles Provenant d'internet, de mails, d'enquêtes de satisfaction, de contacts clients... qui ne Peuvent être exploitées manuellement.

L'objectif du Text mining est de faire ressortir, dans Une masse très importante de données textuelles, l'information utile afin qu'elle puisse devenir exploitable informatiquement. Il s'agit donc d'extraire de la connaissance de documents sémantiquement proches et de rechercher des relations entre entités textuelles (Termes) ou entre documents et de découvrir des tendances, des concepts...

Le processus du Text mining s'effectue en trois étapes que nous observons également dans le schéma ci-dessous :

- Le prétraitement des données (découpage, nettoyage...),
- L'indexation ou représentation formelle (caractérisation de chaque document par des termes caractéristiques),
- L'analyse des données indexées : classement des documents par thèmes, recherche de relations...

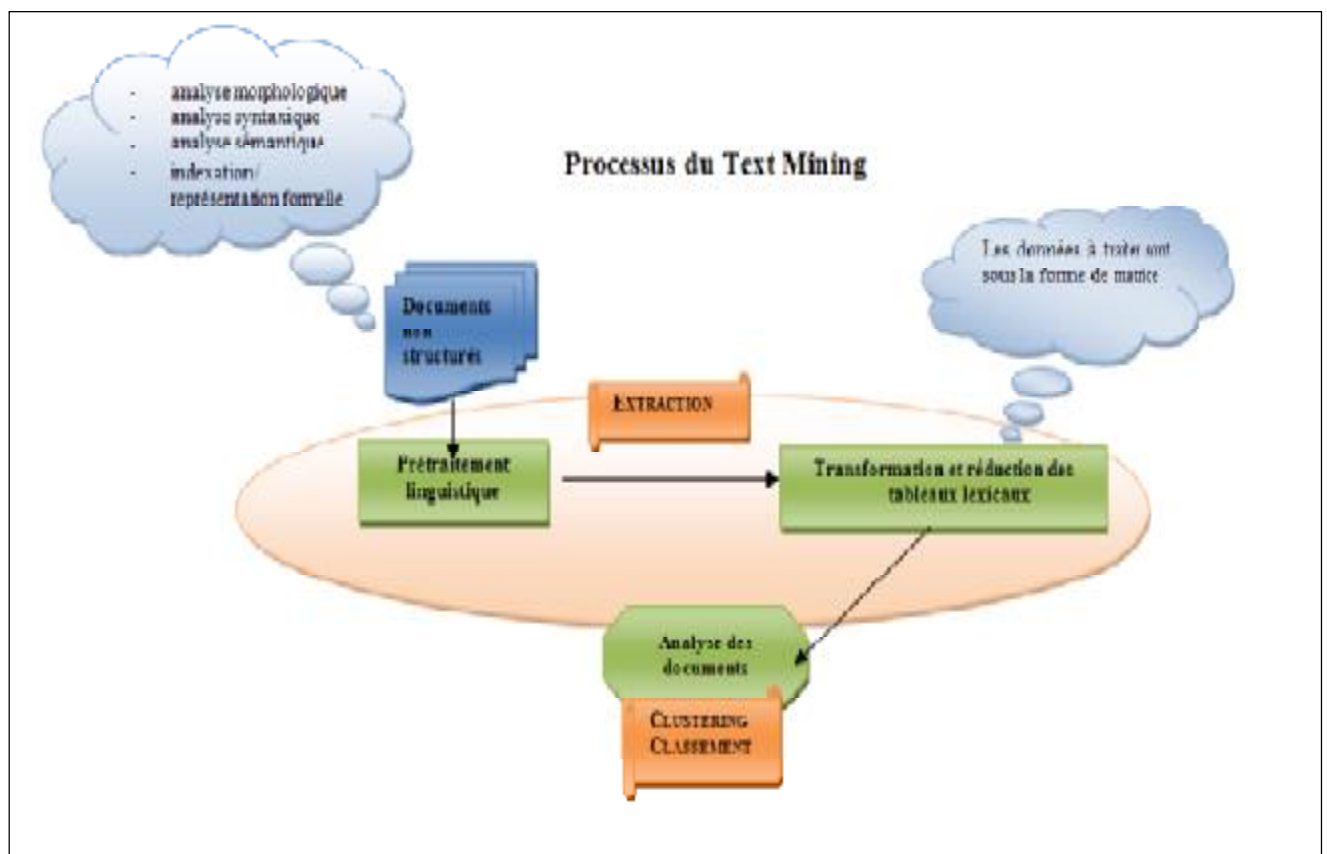


Figure I.1: processus du textmining

I.3. Vue d'ensemble d'une opération de Text Mining :

Le Text Mining repose sur l'analyse des relations qu'entretiennent les mots à l'intérieur des documents d'un réservoir de données textuelles ou d'une base de données. De façon générale, le processus débute par l'acquisition des informations à analyser, en format électronique. On procède alors à l'analyse des données textuelles à l'aide de l'outil de Text Mining choisi. Ce dernier, après avoir extrait les mots des textes, effectue une analyse de la fréquence de cooccurrence des mots afin d'établir les espaces de similarités et de procéder à une clusterisation, c'est-à-dire au regroupement des concepts. Si l'instrument logiciel utilisé pour l'analyse prend en considération les affinités lexicales (la corrélation de certains mots à apparaître ensemble à courte distance), la qualité du distillat sera supérieure (Trybula, 1999). L'étape finale consiste en la présentation des résultats à l'observateur sous forme de résumé, de sur lignage de texte, de taxonomie ou de visualisation.

Pour qui connaît les travaux réalisés dans le domaine de la scientométrie, la proximité est évidente, seules les méthodes et techniques diffèrent.

Le domaine du TextMining est toujours en gestation et diverses solutions sont proposées, alliant technologies et méthodes.

I.4. Les approches du TextMining :

Deux approches peuvent être envisagées :

- Une approche statistique
- Une approche sémantique

I.4.1.L'approche statistique :

Elle consiste à ne voir le document que via le prisme du nombre et des chiffres. Ainsi l'outil statistique de TextMining produit des informations sur le nombre d'occurrence d'un terme, le nombre de cooccurrence de plusieurs termes, la fréquence d'apparition d'un terme dans un document ou un corpus.

Il peut encore produire ce que l'on appelle des « vecteurs de sens », qui sont des « concepts » statistiques de cooccurrence de termes qui permettent de classer et/ou de catégoriser tout un corpus.

- Les avantages de l'approche statistique :

Le principal réside dans son très faible coût d'entretien eut égard au véritable service que cela peut apporter, à condition que le volume du corpus documentaire soit significatif, voire très important.

- Les désavantages de l'approche statistique :

Il n'y a pas de prise en compte des spécificités du corpus documentaire traité : textes médicaux, commerciaux, scientifiques ou autres, seront adressés de manière identique, grâce à la puissance du calcul statistique. la pertinence du traitement qui est difficilement prévisible, et généralement moins élevée que l'approche sémantique.

I.4.2.L'approche sémantique :

Dans le cas de l'approche sémantique, la démarche ne va plus s'appuyer sur la force brute de la puissance de calcul mais sur un élément externe. Cet élément externe appelé référentiel. Il peut être statique ou dynamique. Une fonction comme la reconnaissance d'entité (entity recognition) va ainsi déduire que le groupe de mots « Christian Fauré », écrit sans plus d'information dans un document, aura été cherché mon nom dans un référentiel de personne, et qu'il l'aura trouvé dans la catégorie « personne de nationalité française ».

- Ü Les référentiels statiques peuvent être des mots clés, des listes à plats, des thesaurus, des ontologies. Le moteur de text mining va rajouter aux documents qu'il traite l'ensemble des informations que peut contenir le référentiel.
- Ü Les référentiels dynamiques sont des référentiels qui mettent en œuvre des logiques. Celles-ci peuvent être des logiques formelles, mais aussi des logiques probabilistes (comme les réseaux bayésiens). La différence avec le référentiel statique est que le document qui passe par le moteur de text mining va être enrichi par des informations qui sont déduites du référentiel : le référentiel fait un travail de déduction avant de donner sa réponse au moteur de text-mining qui va enrichir le document.

- Les avantages de l'approche sémantique :

On peut paramétrer le moteur de text mining pour coller à la spécificité du corpus documentaire en exploitant l'ensemble des référentiels du domaine ou de l'organisation. On peut également modéliser des connaissances métiers spécifiques pour effectuer des traitements de text mining qui répondent à un besoin bien identifié. La pertinence des résultats obtenus est beaucoup plus fine et généralement meilleure que dans l'approche statistique (la notion de « meilleur » étant toute fois toujours relative).

- Les désavantages de l'approche sémantique :

Le coût d'exploitation et de maintenance est très fort. Cela demande des ressources matérielles, budgétaires et humaines significatives. De plus si le corpus est important, le temps de traitement requis peut être très long et représenter un frein à la démarche.

En Couplant des approches linguistiques et sémantiques et des approches statistiques, le Text mining Permet de faire émerger du sens à partir de grandes masses de textes qui seraient très difficilement Exploitable « manuellement ». Il permet d'en donner une vue globale ou d'en extraire une information précise, pour adresser cette information aux bons destinataires pour mettre en relation des données textuelles avec d'autres types de données.

I.5. Les sous tâches du Text Mining :

I.5.1. Analyse linguistique :

On distingue cinq niveaux d'analyse :

- 1) Le niveau morphologique : identification des mots d'une phrase, s'intéresse à la composition des mots (préfixes, suffixes, racines...) en dehors de leur rapports dans la phrase.

Ce niveau se décompose en :

- Tokenisation: identification des frontières des mots (simples et composés), et de phrases,
- Tagging: identification de la catégorie - nom, verbe, adjectif - de chaque mot,
- Lemmatisation: identification de la forme canonique (ou lemme) dans le dictionnaire.

- 2) le niveau syntaxique : identification des constituants et des fonctions d'une phrase,

- Analyse surfacique (ou chunking) : identification des frontières majeures de constituants (groupe nominal, verbal, etc.) et/ou des relations majeures entre les mots,
- Etiquetage fonctionnel (ou tagging) : affectation de fonctions grammaticales aux constituants,
- Analyse syntaxique totale (ou parsing) : construction d'un arbre représentant la structure de la phrase complète.

Cette étape, qui a fait et fait encore l'objet de recherches importantes, est maintenant souvent considérée comme finalement moins cruciale, sachant que l'on peut déjà traiter nombre de problèmes aux deux étapes précédentes.

- 3) le niveau sémantique : identification du sens des mots et de la structure logique d'une phrase,

- Sélection de sens (WSD Word sense disambiguation) : choix du sens de chaque mot. Cette fonctionnalité est obligatoire dans un moteur sémantique. On peut dire que c'est le facteur différenciant majeur entre un moteur sémantique et un moteur classique. Pour traiter le niveau WSD, il est indispensable d'avoir déjà réglé le niveau morphologique et au moins le niveau du chunking.
- Structuration logique (ou tagging) : identification des arguments de chaque prédicat et de leur rôle sémantique (agent, but, lieu, etc.).

- 4) le niveau du texte : identification des relations entre les phrases et de la structure d'un texte,

- Résolution des anaphores : antécédents des pronoms, ellipses, références,
- Détermination de la structure rhétorique: commentaires, explications, causalités,
- Détermination de la structure thématique: de quels sujets le texte traite-t-il ?

- 5) le niveau du corpus : identification de la structure d'un ensemble de textes.
- Détermination de la nature des documents : article de presse, article technique, texte réglementaire, brochure commerciale, ...
 - Structure thématique du corpus (ou tagging) : de quels sujets le corpus traite-t-il ?

La plupart de ces tâches correspondent à des sous-disciplines de la linguistique. Du point de vue de l'ingénierie du TALN, elles correspondent à composants spécifiques qui ont chacun besoin d'un type de ressources linguistiques et d'un type d'algorithmique particulier.

I.5.2. Recherche d'information :

I.5.2.1. Indexation :

L'étape d'indexation permet de réaliser le passage d'un document textuel (ou une requête) à une représentation exploitable par un modèle de RI par la construction de mots clés appelé Langage d'indexation.

Document textuel (ou requête) ———→ représentation exploitable par le SRI
Indexation

Cette transformation appelée indexation consiste à extraire du texte un ensemble de mots clés appelés descripteurs. Ces descripteurs vont représenter le document dans le corpus. Chaque descripteur peut être accompagné de connaissances à priori pour mieux appréhender la recherche.

I.5.2.2. Modèle vectoriel :

Le modèle vectoriel est un modèle algébrique où l'on représente les documents et les requêtes par des Vecteurs dans un espace multidimensionnel dont les dimensions sont les termes issus de l'indexation [Salton, 1983], la création de l'index implique le parcours de la collection, la recherche des termes Pertinents, le traitement lexical des termes retenus, les documents et la requête sont représentés comme des vecteurs dans le repère des termes.

La comparaison de la requête au document est effectuée en comparant leurs vecteurs respectifs. Soit R l'espace vectoriel défini par l'ensemble des termes: $\langle t_1, t_2, \dots, t_n \rangle$ Un document d et une requête q peuvent être représentés par des vecteurs de poids comme suit:

$d \langle wd_1, wd_2, \dots, wdn \rangle ;$

$q \langle wq_1, wq_2, \dots, wqn \rangle ;$

W_{di} et w_{qi} correspondent aux poids du terme t_i dans le document d_i et dans la requête q et n correspond au nombre de termes de l'espace. Étant donnés ces deux vecteurs, leur degré de correspondance est déterminé par leur similarité.

I.5.2.3. Notion de similarité :

Évaluer la similarité entre documents textuels est une des problématiques importantes de plusieurs disciplines comme l'analyse de données textuelles, la recherche d'information ou l'extraction de connaissances à partir de données textuelles (Text Mining). Dans chacun de ces domaines, les similarités sont utilisées pour différents traitements.

En Text Mining, les similarités sont utilisées pour produire des représentations synthétiques de vastes collections de documents.

Les techniques mises en œuvre pour calculer les similarités varient bien évidemment selon les disciplines, mais elles s'intègrent cependant le plus souvent dans une même approche générale en deux phases :

1. Les documents textuels sont d'abord associés à des représentations spécifiques qui vont servir de base au calcul des similarités. Bien que la nature précise des représentations utilisées dépende fortement du domaine d'application, il faut noter que, presque dans tous les cas, les documents sont représentés sous la forme d'éléments d'un espace vectoriel de grande dimension.

2. Un modèle mathématique est choisi pour mesurer les similarités.

Les mesures de similarité de textes ont été utilisées dans de nombreux domaines. Par exemple, pour la classification de textes ([Rocchio, 1971]), la désambiguïsation du sens des mots ([Lesk, 1986]), la traduction automatique ([Papineni et al. 2002])...

I.5.2.4. Les métriques de similarité utilisées:

Les termes que nous avons besoins de comparer sont représentés par les chaînes de caractères qui composent d'un groupe de mots.

Comme montré par la **Figure I.2**, les métriques de chaîne peuvent être classées en trois catégories principales :

- ü les méthodes basées sur des caractères.
- ü les méthodes basées sur des tokens.
- ü les méthodes hybrides.

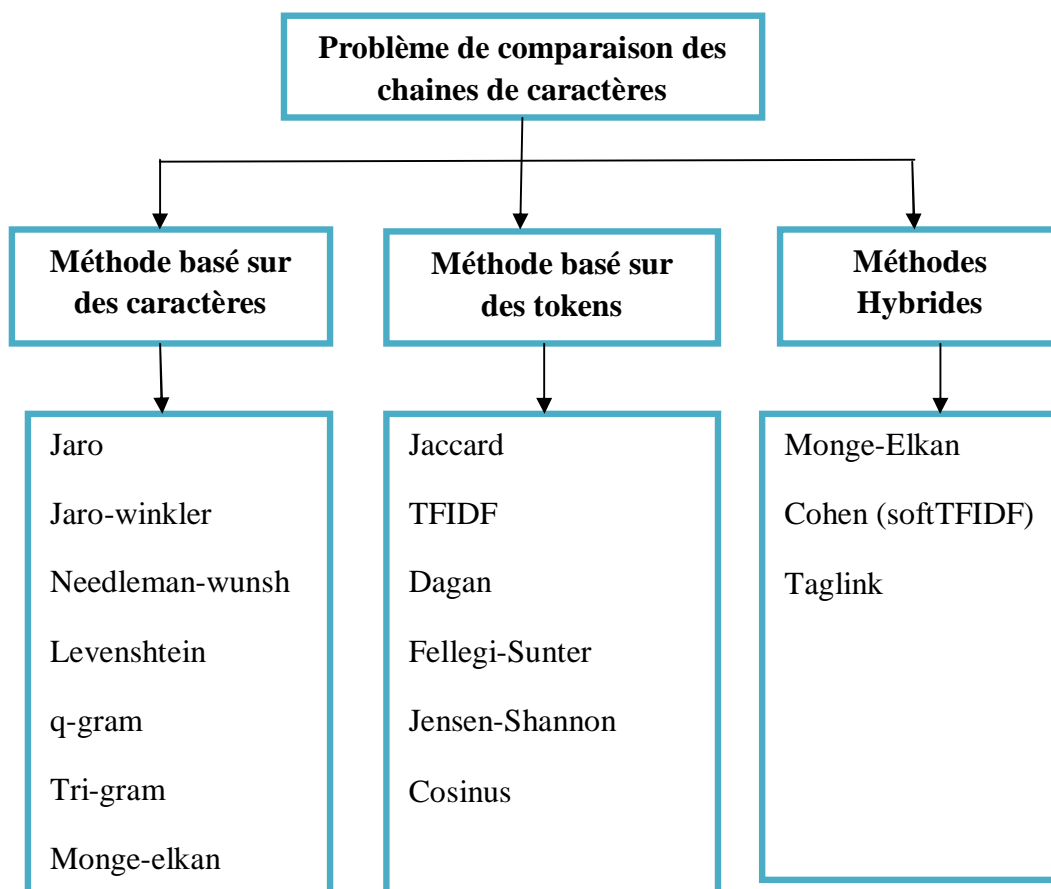


Figure I.2: classification des métriques de comparaison des chaînes de caractères

I.5.2.5 Extraction d'information :

Est la tâche qui permet d'extraire des informations structuré à partir de texte non structuré ou semi structuré. Appliquant l'extraction d'informations sur le texte, est liée au problème du texte afin de créer une vue structurée de l'information présente en texte libre. Créer un texte plus facilement lisible par machine pour traiter les phrases.

Sous taches typiques d'IE sont les suivants :

Nommé extraction d'entités qui pourraient inclure:

- Reconnaissance d'entités nommées : la reconnaissance des noms connus de l'entité (pour les personnes et organisations), des noms, des expressions temporelles, et certains types d'expressions numériques.
- Coréférence résolution : consiste à déterminer les si diverses entités nommées se réfèrent au même objet, Par exemple, "International Business Machines" et "IBM" se réfèrent à la même entité du monde réel.
- extraction de la relation : l'identification des relations entre les entités, telles que: Personne travaille pour l'organisation (extrait de la phrase "Bill travaille pour IBM").

I.5.3 Fouille de données (datamining) :

Il existe plusieurs type de fouille de données textuelles permettant d'explorer au mieux le Contenu de corpus textuel. Parmi ces techniques :

L'estimation, La prédiction, La segmentation, La classification Les règles d'association.

✓ la classification :

Consiste à examiner les caractéristiques d'un individu et lui attribuer une classe, la classe est un champ particulier à valeurs discrètes. Par exemple Lorsqu'on cherche à identifier les principales thématiques abordées dans un ensemble de documents, deux approches sont possibles pour regrouper des documents similaires:

La classification supervisée: ou catégorisation, consiste à identifier la classe d'appartenance d'un objet à partir de certains traits descriptifs.

Cette approche permet le classement automatique de documents dans des classes préexistantes (connues à l'avance), comme par exemple les rubriques d'un journal (société, sport, politique).

Les méthodes les plus efficaces sont basées sur un corpus d'apprentissage. Elles permettent de déterminer automatiquement la catégorie d'un document à partir d'échantillons de documents représentatifs pour chaque catégorie choisie. ...)

la classification non supervisée des documents: ou clustering, c'est la découverte de classes de Documents sans a priori, (on ne connaît pas les classes à l'avance).

Comme dans toute approche non supervisée, ces méthodes supposent le choix :

- d'une représentation des objets à classer
- d'une mesure de similarité entre les objets
- d'un algorithme de classification (hiérarchique ou non hiérarchique)

✓ l'estimation :

Elle consiste à estimer la valeur d'un champ à valeurs continues à partir des caractéristiques d'un objet.

L'estimation peut être utilisée dans un but de classification. Il suffit d'attribuer une classe particulière pour un intervalle de valeurs du champ estimé.

Des exemples de tâche d'estimation sont :

-Noter un candidat à un prêt ; cette estimation pourra trouver à une application pour attribuer un prêt

(Classification), par exemple, en fixant un seuil d'attribution,

- Estimer les revenus d'un client.

✓ la prédiction :

Cela consiste à estimer une valeur future. En général, les valeurs connues sont classées chronologiquement. On cherche à prédire la valeur future d'un champ. Cette tâche est proche des précédentes. Les méthodes de classification et d'estimation peuvent être utilisées en prédiction. Des exemples de tâche de prédiction sont :

-Prédire les valeurs futures d'actions,

- Prédire au vu de leurs actions passées les départs de clients.

✓ les règles d'association :

Une règle d'association (RA) est une règle d'implication conditionnelle permettant de trouver les corrélations entre des éléments qui sont liés par une relation R. dans le contexte de

textmining les règles d'association sont interprétées comme une cooccurrence de termes impliquant la présence d'autres termes dans les textes en accord avec la définition usuelle en sémantique lexicale. [Ani 90]

▼ la segmentation :

Il s'agit de créer des groupes homogènes dans la population (l'ensemble des enregistrements). Il appartient ensuite à un expert du domaine de déterminer l'intérêt et la signification des groupes ainsi constitués. Cette tâche est souvent effectuée avant les précédentes pour construire des groupes sur lesquels on applique des tâches de classification ou d'estimation.

I.6. Différence entre text mining et data mining:

C'est-à-dire la distinction entre fouille des données et fouille de données textuelles. Historiquement, le data mining est à la base du textmining au sens où celui-ci est l'extension du même but et du même processus vers des données textuelles. La distinction est donc fondée à son origine principalement sur la nature des données auxquelles s'adressent l'une et l'autre, d'une part des données numériques et factuelles, et d'autre part des données textuelles.

Un autre élément de distinction est l'état de structuration des données. En général le datamining travaille sur des données structurées et stockées dans des bases de données relationnelles. En revanche, le textmining travaille sur des données textuelles non structurées (Feldman et al. 1998a et 1998b; Landau et al. 1998).

Le text mining se distingue du data mining également par les moyens techniques spécifiques qu'il le faut employer pour traiter les données textuelles et non structurées.

Une définition générale du text mining est la suivante : l'extraction d'information à partir des formes ou patrons non manifestes dans des grands corpus de textes. Autrement dit, l'objectif est le traitement de grandes quantités d'information qui sont disponibles sous une forme textuelle et non structurée. (Feldman et al. 1998a ; Landau et al. 1998).

L'intelligence économique est sensée assurer aux acteurs économiques une information exploitable et utile, dans la mesure où cette information est textuelle (notes, lettres, rapports techniques, articles scientifiques, brevets, etc.), l'intérêt que le text mining peut représenter Pour l'intelligence économique vis-à-vis du simple data mining est alors évident.

	Data Mining	Textmining
Objet	numérique & catégorique	textuel
Structure	Structuré	non-structuré
Représentation	Simple	complexe
Dimension	milliers	milliers
Maturité	Implémentation vaste dès 1994	Implémentation vaste dès 2000

I.7. Domaines d'application et les utilisateurs du text mining :

Le text mining trouve ses principales applications dans les dépouillements d'enquêtes ou les projets d'analyse des données pour lesquels certaines des réponses se présentent sous une forme non structurées ou textuelle (par exemple, des messages électroniques, des commentaires, des suggestions dans un questionnaire de satisfaction avec des questions ouvertes, la description de symptômes médicaux par des patients ou des praticiens, des réclamations, etc.) Qu'il est souhaitable d'intégrer dans le cours de l'analyse globale.

Ces techniques sont également fréquemment utilisées pour produire des modèles prédictifs permettant de classer automatiquement du texte, par exemple, pour faire suivre automatiquement les e-mails vers le destinataire le plus approprié, ou pour faire la distinction entre les "spam" et les messages importants. Application typique de text mining :

- Analyse des réponses aux questionnaires ouverts : l'idée est de permettre aux gens d'exprimer leur point de vue ou des avis ou bien de poser des questions sur un sujet d'études, cela peut donner un aperçu de l'opinion des clients.
- Traitement automatique des messages, e-mail : une autre application commune pour l'exploration de texte est d'aider à la classification automatique de texte par exemple filtré automatiquement courrier électronique indésirable, plus tôt d'identifier le courrier électronique indésirable, de cette manière ces messages peuvent être automatiquement rejetés.
- Enquêtes sur les concurrents en rampant de leur site web : très utile pour traiter automatiquement le contenu des pages web dans un domaine particulier par exemple qu'on accède à une page web, on commence à ramper les liens retrouvés pour traiter toutes les pages web qui sont référencés.

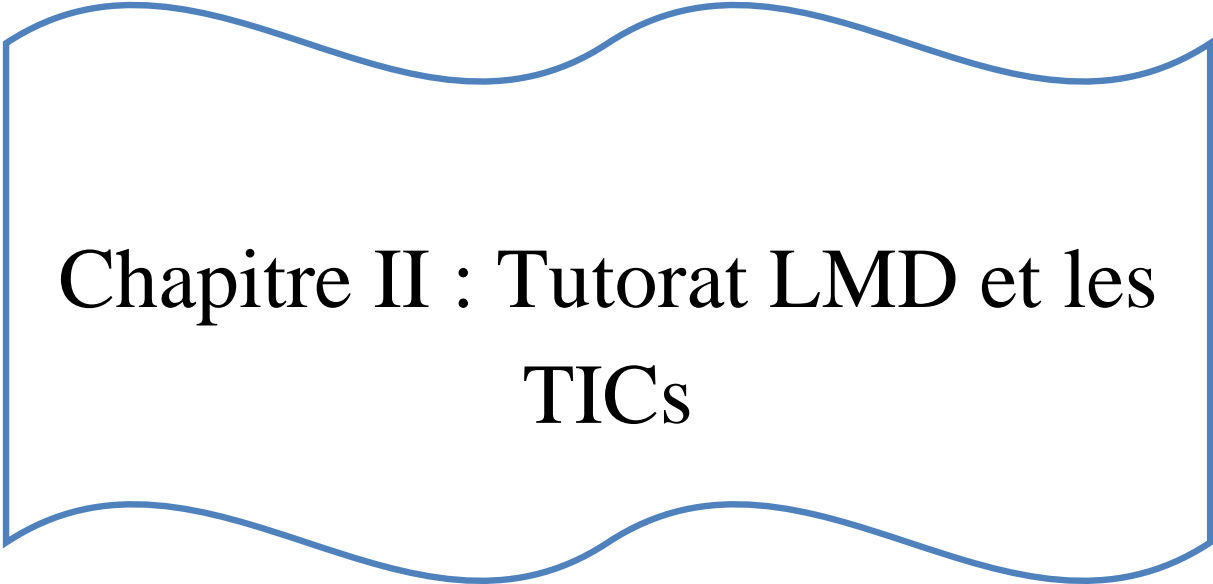
Ø Utilisateurs de texte mining :

- Analystes financiers
- Économistes
- Professionnels du marketing
- Services de satisfaction clientèle
- Recruteurs
- Décideurs

Conclusion

Dans ce chapitre nous avons fait un aperçu sur traitement des données textuelles (fouille de texte) qui est désigné sous le nom text mining, ainsi nous avons présenté brièvement ses principales sous taches.

Le chapitre suivant, fera l'objet de la conception de notre application.



Chapitre II : Tutorat LMD et les TICs

Introduction :

La réforme LMD a apporté des innovations majeures et des concepts nouveaux dans l'enseignement supérieur. Elle vise à améliorer la qualité de la formation, de l'information et de l'orientation de l'étudiant en augmentant la part de sa participation à la construction de son parcours de formation parmi les offres de parcours types qui lui sont proposés, ainsi qu'à la réalisation de ce parcours par l'augmentation du volume de travail personnel à même de lui donner les compétences voulues en méthodologie et en adaptation aux situations. La question qui se pose alors est celle qui permet d'identifier les mécanismes qui aident à la concrétisation de ces principes. Le tutorat est une démarche d'apprentissage ; il est formateur pour l'étudiant car il lui permet de transmettre ses savoirs et ses savoir-faire tout en les approfondissant et en les consolidant. L'étudiant novice reçoit une aide de l'étudiant expert mais les échanges ne sont jamais unidirectionnels ; le tutorat est aussi une aide pour l'enseignant qui ne peut répondre, en temps réel à tous les questionnements, qui a ainsi l'opportunité d'observer les étudiants au travail, d'évaluer leurs compétences et de favoriser une démarche d'auto-évaluation.

Dans ce chapitre introductif, nous allons tout d'abord donner un bref aperçu sur l'histoire du tutorat et le tutorat dans le système LMD, puis nous introduisons les TIC (les Technologies pour l'Information et la communication).

II.1. Le tutorat :

II.1.1 Qu'est-ce que le tutorat :

Le tutorat est une relation entre deux personnes dans une situation formative : un professionnel et une personne en apprentissage d'un métier dans son environnement. Ou encore on peut le définir comme un espace de dialogue entre étudiants ou entre étudiants et enseignants où est apportée une réponse appropriée et personnalisée sur divers sujets tels que :

- Ø L'aide à l'intégration dans un nouvel environnement.
- Ø L'aide à l'organisation du travail personnel.
- Ø L'aide à la maîtrise de méthodes de travail spécifiques.
- Ø Une première approche en matière d'orientation.
- Ø Une prédéfinition d'un projet professionnel.

On définit, en général, deux types de tutorat

- Ø le tutorat **réactif** utilisé lorsque le tuteur explicite une demande ou un problème au tuteur et que ce dernier lui apporte une réponse.
- Ø le tutorat **proactif** utilisé lorsque le tuteur intervient auprès du tuteur sans demande préalable, Permettant au tuteur de surveiller le bon déroulement de l'étude et d'éviter ainsi qu'un apprenant se sente isolé.

II.1.2. L'objectif du tutorat :

Le tutorat a pour objectif de soutenir les efforts d'apprentissage, il vise à rompre l'isolement et à atteindre les résultats des objectifs pédagogiques des apprenants. En effet il peut se sentir perdu et ressentir des difficultés pour "trouver son chemin", il peut manquer de motivation ou d'autonomie face aux apprentissages. Pour toutes ces raisons, la présence du tuteur est essentielle.

Les Objectif du tutorat sont :

- Ü faciliter la coopération intergénérationnelle,
- Ü transmettre et partager les valeurs de l'organisme,
- Ü formaliser les savoir-faire du métier ou d'une activité et les transférer ;
- Ü Capitaliser les bonnes pratiques d'un salarié détenant une expertise changeant d'emploi ou quittant l'organisme,
- Ü organiser des dispositifs de transferts pour éviter la perte de compétences qualifiées de « sensibles».
- Ü Permettre au tuteur de parler de lui-même, de ses difficultés ou de tout autre problème qu'il rencontre.

II.1 .3. Le tuteur :

Le tuteur accompagne le tuteur sur le terrain et de façon régulière (accueillir, aider, informer, guider...) dans le cadre d'une relation individualisée et formalisée, avec un objectif de développement des compétences.

Le tuteur s'engage à :

- accueillir et intégrer le tuteur.
- préparer et organiser la transmission des connaissances et de ses savoir-faire, rendre les situations de travail formatrices.
- élaborer, concevoir des outils ou supports facilitant l'apprentissage du tuteur.
- accompagner le tuteur dans ses nouvelles fonctions ou activités.
- transmettre ses connaissances et vérifier leur bonne mise en pratique.

II.1.4. Le rôle du tuteur :

L'objectif du tuteur est d'amener l'étudiant à progresser et à s'améliorer face aux situations d'apprentissages qu'il rencontre. Pour cela il oriente son tutorat à partir de quatre points clefs:

- Ø Le développement de la **métacognition**, c'est-à-dire à identifier comment l'étudiant apprend à apprendre.
- Ø Le transfert de ses apprentissages, chercher à identifier comment l'étudiant prend conscience de ce qu'il sait déjà, comment il intègre ensuite de nouvelles connaissances à mettre en lien avec celles qu'il possède déjà afin de pouvoir les utiliser dans un autre contexte. Ainsi il va passer d'une phase de contextualisation à celle de la décontextualisation, puis à l'étape de recontextualisation

- Ø La gestion des émotions de l'étudiant demande de l'aider à identifier ses émotions, à les comprendre et à mobiliser ainsi son intelligence émotionnelle
- Ø Le maintien de la motivation de l'étudiant peut se réaliser en l'aidant à construire et à développer son projet personnel et professionnel.

Le tuteur peut prendre différents visages. Il peut être un initiateur, un médiateur, un évaluateur, un facilitateur... Mais peu importe le costume qu'il choisira de revêtir, il semble indispensable que tuteur et étudiant travaillent ensemble et partagent leurs observations. Comme le dit Bachelard : « C'est par l'échange que l'on apprend les uns des autres ».

De même que le tutoré doit développer des compétences professionnelles (savoir, savoir-faire et savoir-être), le tuteur doit mobiliser des compétences pour exercer son activité.

Pour construire le parcours de formation et mettre en place des situations de travail formatrices, le tuteur doit être capable de :

- Ø Définir les différents aspects de son métier (techniques utilisées, ordonnancement) et les présenter à une personne non initiée,
- Ø Définir et présenter l'organisation du travail et de la production en situant le poste dans le processus général,
- Ø Formuler des objectifs de formation permettant de construire, de conduire et d'évaluer le parcours de formation en situation de travail,
- Ø De varier les méthodes pédagogiques (expliquer, montrer, mettre en situation),
- Ø D'analyser une séquence de travail à la fois en termes de résultat attendu et de démarche à suivre.

II.2. Le tutorat dans le système LMD :

Le tutorat est un acte volontaire dont la mission est d'accompagner l'étudiant pour lui faciliter l'intégration dans la vie universitaire et l'accès aux informations du monde du travail. Le tutorat dans le système LMD vise ainsi à conduire l'étudiant à bout de bras dans l'élaboration de son parcours de formation et dans ses parcours d'apprentissages tout au long de son cursus universitaire, on ne peut pas concevoir de cadre efficace d'accompagnement des étudiants dans le LMD sans le tutorat

Dans les formations ouvertes à distance, la tâche du tuteur diffère de celle de l'enseignant. Sans se substituer à l'étudiant, les missions et activités du tuteur revêtent quatre principaux aspects :

- Ø informatif et administratif,
- Ø pédagogique,
- Ø technique et méthodologique,
- Ø Psychologique

II.2.1. Aspect informatif et administratif (accueil, l'orientation et médiation) :

- ✓ **Accueil** : l'étudiant trouve auprès de tuteur une personne qui est toujours à son écoute et qui l'aide à trouver des solutions aux problèmes qu'il rencontre à l'université qu'il découvre pour la première fois. Le tuteur explique à l'étudiant le système LMD :
 - Les unités d'enseignement, les matières.
 - Les crédits et les dettes.
 - La progression.
 - Les cursus de formation.
 - L'assiduité aux enseignements.
- ✓ **Orientation** : L'étudiant qui entre à l'université dont il ignore les structures, le fonctionnement et les méthodes d'enseignements doit être guidé et orienté dans «l'espace et le temps » : amphithéâtres, salles de TD et TP, bibliothèque, Rectorat, Facultés, Sécurité sociale.etc.
- ✓ **Médiation** : L'étudiant peut être conseillé par le tuteur pour les démarches qu'il doit entreprendre auprès des différents services de l'université. Il sera sensibilisé au rôle du délégué.

II.2.2. Aspect pédagogique (Apprentissage de l'organisation du travail personnel)

L'accompagnement consiste à aider l'étudiant dans l'organisation de son travail personnel. On lui apprendra à :

- ü prendre des notes.
- ü relire ses cours.
- ü préparer ses exercices.
- ü réparer ses travaux pratiques.
- ü chercher et consulter des ouvrages.

II.2.3. Aspect méthodologique et technique (Initiation aux méthodes de travail universitaire et utilisation des outils et supports pédagogiques):

Pour guider les étudiants dans leur démarche, le tuteur leur proposera de travailler en petits groupes pour :

- revoir le cours.
- préparer les TD et TP.
- faire des recherches bibliographiques.

II.2.4. Aspect Psychologique (stimulation et motivation à poursuivre son parcours de formation):

Bien que le tuteur n'ait pas une formation de psychologue, il peut :

- écouter l'étudiant et créer une relation de confiance.
- fournir un soutien et des conseils personnalisés.
- valoriser les acquis et l'encourager à les améliorer afin de réussir ses études.

Aujourd'hui, la conception la plus répandue du tutorat dans le système LMD est celle d'un dispositif d'accompagnement méthodologique, pédagogique et psychologique s'adressant aux

nouveaux étudiants qui s'inscrivent en première année de licence (L1) dans une institution universitaire. Il s'exerce sous forme de conseils apportés à l'étudiant pour l'aider à s'orienter et pour organiser son travail, selon la méthode qui lui convient le mieux. Il peut s'étendre dans certains cas au-delà de la L1.

II.3. Les avantages du tutorat dans le système LMD :

- Ø Les apprenants n'ont pas besoin de parcourir des distances loines pour avoir le tutorat.
- Ø C'est une bouée de sauvetage pour les apprenants handicapés qui ne peuvent pas voyager d'avoir le tutorat facilement ;
- Ø Le soutien des apprenants est disponible 24/7 qui permet aux étudiants d'obtenir de l'aide ou l'assistance en tout items qu'ils ont besoin.
- Ø Le tutorat fournit une aide aux étudiants sur certain sujet.
- Ø Le service de tutorat offre avant tout une grande flexibilité. En effet, la recherche d'un tuteur se fait dès lors beaucoup plus rapidement, puisque nous ne faisons face à aucune contrainte au niveau du déplacement et des distances.
- Ø Aussi, pour ceux qui désirent une aide rapide ou ponctuelle (par exemple, une rencontre d'urgence avant un examen), le tutorat en ligne permet d'avoir accès à un tuteur qualifié dans des délais très réduits. Ce service est donc idéal pour ceux qui veulent des réponses rapides sur des sujets précis.

II.4. Terminologie (ensemble de termes propre à un domaine):

II.4.1. les TICE :

Les technologies de l'information et de la communication pour l'enseignement :recouvrent les outils et produits numériques pouvant être utilisés dans le cadre de l'éducation et de l'enseignement (TICE = TIC + Enseignement), elle regroupent un ensemble d'outils conçus et utilisés pour produire, traiter, entreposer, échanger, classer, retrouver et lire des documents numériques à des fins d'enseignement et d'apprentissage.

II.4.2. Apprenant :

Une personne volontairement et explicitement engagée dans un processus de formation par rapport à un domaine de compétences. Dans notre cas il s'agit de compétences dans le domaine de l'informatique, il peut :

- § Consulter le ou les modules de formations mis à sa disposition ;
- § Communiquer et travailler avec les autres apprenants, ou avec son formateur via les outils collaboratifs ;

II.4.3. L'apprentissage :

L'apprentissage est un terme général désignant les façons dont les êtres vivants augmentent leurs connaissances et améliorent leurs aptitudes et mécanismes de raisonnement. L'apprentissage est basé sur les opérations de communication, de transfert et de réception de l'information.

II.4.4. Formateur :

Une personne qui est nécessairement un pédagogue, ayant obtenu une formation spécifique en formation des adultes et, en formation à distance. Il peut activer ou empêcher une stratégie d'apprentissage. Il assure le suivi des apprenants. C'est le responsable du déroulement de la formation, le formateur peut :

Consulter l'activité de l'apprenant : état d'avancement, messages, exercices

§ Effectues et erreurs commises, etc.

§ Communiquer et animer des groupes via la messagerie, les forums et le chat pour le cas de
E-Learning.

II.5. Outils utilisés :

II.5.1. Outils techniques: Une plate forme ou un portail de formation intégrant les outils de communication, une connexion vers un réseau internet.

Les outils utilisés pour assurer la communication entre l'apprenant et les tuteurs sont répertoriés en :

II.5.1.1. Outils de communication synchrones : c'est une communication en temps réel, les sessions de travail synchrone collectives ou d'accompagnement à l'apprenant demande de préparer d'avance et d'animer précisément les séances.

Avantage des outils synchrones :

- § La dynamique de groupe s'installe plus rapidement et les apprenants
- § s'interagissent intensivement à l'écran avec les formateurs ;
- § On peut rapidement créer du contenu prêt à diffuser ;
- § Utilisation de langage oral et visuel.

II.5.1.2. Outils asynchrones : c'est le système le plus utilisé, car il est très souple, il permet des réponses relativement rapide mais les outils sont plus ou (moins) chronophage que les outils synchrones.

Ils permettent aux utilisateurs de gérer leur temps, la communication asynchrone n'imposant pas des horaires particuliers, il n'est pas nécessaire de trouver des plages d'horaire communes. Les apprenants structurent mieux leur savoir en termes d'appropriation (saisie) puisque ils prennent le temps de la réflexion.

Avantage des outils asynchrones :

- § Les apprenants évoluent à leur propre rythme.
- § Ils peuvent adapter l'ordre dans lequel ils appréhendent les éléments du cours ;
- § Ils peuvent revoir et approfondir certains aspects du cours à leur guise ;
- § Un temps de réflexion est donné à chaque acteur (apprenant, formateur) pour donner leurs contributions.

Exemple d'outils de communication :

Outils de communication synchrone	Outils de communication asynchrone
Téléphone Chat Tableau blanc Audio conférence Vision conférence	Fax Courier électronique Liste de diffusion Forum

II.5.2. Outils pédagogiques:

Ces outils pédagogiques devraient pouvoir être mieux valorisés au sein des formations car ils peuvent jouer un rôle fédérateur pour les tuteurs. Avec le scénario pédagogique, le concepteur du cours définit au travers de cet outil les différentes tâches qui sont attendues et la manière dont elles doivent être assurées par le tuteur. parmi ces outils :

- **la charte du tutorat en ligne** : la charte définit comment le tutorat doit se dérouler durant toute la formation, elle a pour objet de formaliser une démarche de tutorat, son objectif est de fournir les grands fondements à respecter par les tuteurs afin de valoriser les interactions dynamiques entre tuteur et apprenants Lors du tutorat d'accueil, le tuteur crée les conditions optimales pour une formation à distance et suit impérativement le planning de la formation en sachant que chaque module n'est en effet ouvert généralement que pour une durée.

✓ Avant la formation, le tuteur doit :

- Ø Prendre connaissance du contenu du module et tester leur fonctionnalité
- Ø Prendre contact par e-mail avec les apprenants pour fixer un rendez-vous du premier Chat tutoré.
- Ø Mettre à disposition des apprenants, le planning général de la formation en expliquant pour chaque module:
- Ø Les objectifs généraux et spécifiques à atteindre.
- Ø Les moyens d'y parvenir (les tâches et activités à accomplir / semaine).
- Ø Les consignes (par activité).
- Ø Les ressources (documents, références ou site Web).
- Ø Les espaces de partage (devoirs individuels ou collaboratifs).
- Ø Les méthodes d'échanges (forum, chat, visioconférences, courriel ou Wiki).
- Ø Le tuteur doit ensuite remplir son rôle organisationnel vis-à-vis des apprenants au début de la formation en:
 - Informant le coordinateur de sa disponibilité pour animer les séances synchrones qu'il compte assurer.
 - Présentant et précisant son rôle en une première réunion commune (soit synchrone soit asynchrone)
 - Définissant les objectifs de la formation, compte tenu des connaissances et des compétences des apprenants.

- Présentant les moyens mis en œuvre, entre autres par des cours médiatisés, des ressources d'accompagnement et des outils de communication en mettant le point sur le travail collaboratif.
- Préparant les apprenants à la bonne utilisation des nouveaux outils (la plateforme).
- Formant les équipes d'apprenant (de 3 ou 4 chacune), tenant compte de leur diversité culturelle et leur recul.
- Choisisant le modérateur et le secrétaire de chaque équipe tout en expliquant leur rôle.
- Demandant aux apprenants de réaliser un plan de travail précis dans les délais requis

✓ Ce qu'il doit faire le tuteur durant la formation :

- 1) Pendant la formation, le tuteur coordonne les apprenants et leur sert de guide.
- 2) Il s'assure que l'organisation du travail soit respectée selon la planification établie en prenant rapidement contact avec les retardataires.
- 3) Il organise leurs travaux individuels et collaboratifs.
- 4) Il s'assure que le cadre de travail soit clair et lisible pour tous.
- 5) Il clarifie les consignes de travail et aide les apprenants à mieux organiser leur travail collaboratif en favorisant les échanges entre eux.
- 6) Il joue le rôle d'expert des contenus (l'apprenant est sous la responsabilité pédagogique du tuteur).
- 7) Le tuteur doit insister sur l'importance de la concertation entre les membres du groupe, celle du travail commun après une réflexion individuelle.
- 8) Il aide les apprenants à comprendre les différentes démarches impliquées dans la réalisation d'un projet.
- 9) Il les incite à consulter les nouvelles régulièrement.
- 10) Il crée un espace commun aux apprenants où il peut intervenir afin d'y partager les idées.
- 11) A tout moment, le tuteur doit assurer une aide technique aux apprenants.
- 12) Il documente et dirige les apprenants vers les meilleures références.
- 13) Il les aide à trouver des solutions.
- 14) Le tuteur doit clarifier les critères d'évaluation tout en reportant régulièrement l'évaluation de chaque activité.
- 15) Il organise une séance de bilan à la fin du module.
- 16) Il valide les modules en fonction des modalités et des dates.

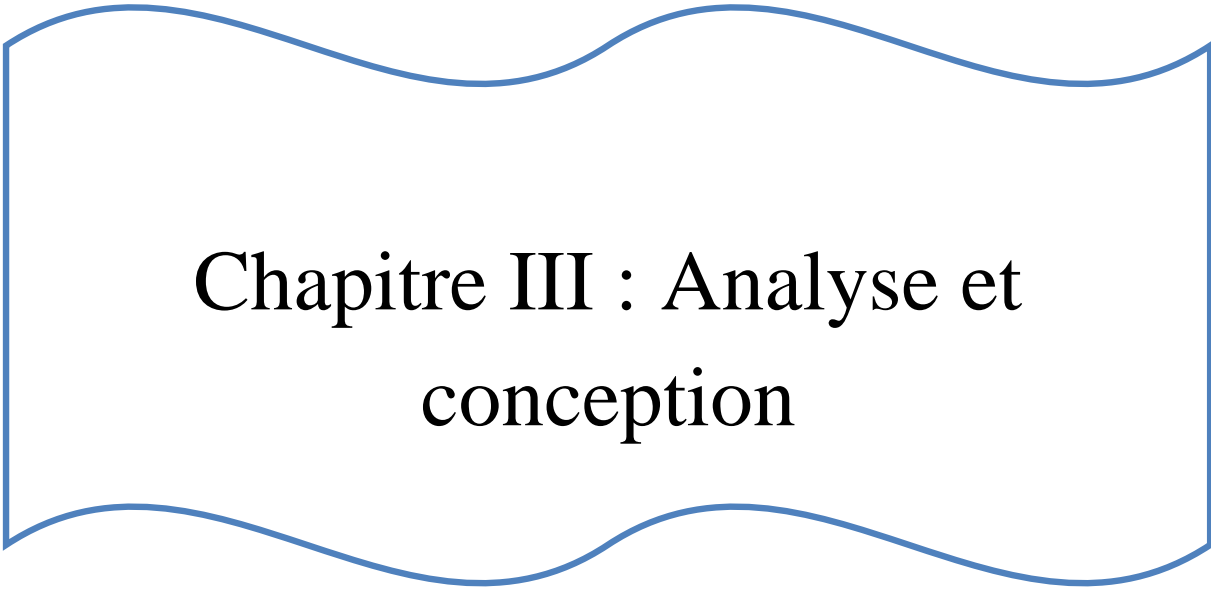
✓ Ce qu'il doit faire durant l'évaluation :

- 1) Le tuteur vérifie que les objectifs d'apprentissage ont été atteints à travers des évaluations formatives et d'autres certificatives.
- 2) Il respecte les modalités de l'évaluation: épreuves, grilles, barèmes; selon les compétences à mesurer en tenant compte des aspects suivants:
 - ü Clairement exprimer le but de l'évaluation
 - ü Prévoir les conditions d'administration
 - ü Prévoir les modalités de correction
- 3) Il valide les épreuves
- 4) Il remet les notes au coordinateur
- 5) Il garantit l'égalité de traitement entre tous les apprenants
- 6) Il prend en charge les problèmes de transparence, des critères d'évaluation et de la répartition des notes.

- 7) Il garde la communication continue entre lui et les différents membres du groupe dans une ou plusieurs.
- 8) Il rédige un compte rendu pour faire une évaluation finale du déroulement du module.

Conclusion :

Dans ce chapitre on a présenté qu'est-ce qu'un tutorat LMD, son intérêt dans les formations à distance ainsi que ses objectifs, les rôles et les fonctions des acteurs participants dans les tutorats, les TICS et les différents outils utilisés.



Chapitre III : Analyse et conception

Introduction :

Dans le but d'une meilleure organisation et une bonne maîtrise du travail, tout processus de développement d'applications ou systèmes informatiques doit suivre une méthode ou démarche bien définie.

Pour notre projet, nous allons suivre un processus de développement d'application web, en utilisant le langage de modélisation très répandu UML (Unified Modeling Language) et son extension pour le Web, qui permet de bien représenter l'aspect statique et dynamique d'une application par une série de diagrammes qu'il offre.

Les diagrammes d'UML peuvent être utilisés librement à tous les niveaux d'un processus d'élaboration d'un produit logiciel. Cependant, la pratique montre bien que certains diagrammes s'apprêtent bien mieux à certaines phases qu'à d'autres.

III.1. Description générale de la démarche de conception :

Notre démarche s'articule autour de deux étapes : l'analyse et la conception.

§ la phase d'analyse :

Cette partie comprend l'identification des besoins fonctionnels du système, des acteurs et leurs interactions avec le système ainsi que les cas d'utilisation.

§ la phase de conception :

Le processus de conception de notre projet se caractérise par deux niveaux : le niveau applicatif et le niveau donné.

Le niveau applicatif : s'appuie essentiellement sur quelques diagrammes de l'extension du langage de modélisation UML pour les applications web. A cet effet nous avons adopté la démarche suivante :

Ø Après l'identification des différents acteurs ainsi que les cas d'utilisation qui sont mis en œuvre par ces acteurs, le diagramme de cas d'utilisation est élaboré dont une description textuelle sera présentée.

Ø Chaque cas d'utilisation se traduit par un ou plusieurs scénarios. Chaque scénario fait l'objet d'une description sous forme graphique à l'aide d'un diagramme de séquence et un diagramme d'activité.

Ø Une identification des classes est fournie par la synthèse des diagrammes de séquence, Ainsi le diagramme de classe sera élaboré.

Le niveau donné : concerne l'organisation conceptuelle, logique et physique des données manipulées. Durant la partie analyse nous avons pu identifier les données nécessaires et indispensables au bon fonctionnement de l'application et à travers la conception du niveau applicatif nous allons dégager les classes significatives, dès lors on peut élaborer la conception de la base de données.

La figure suivante donne la représentation graphique de la démarche de modélisation :

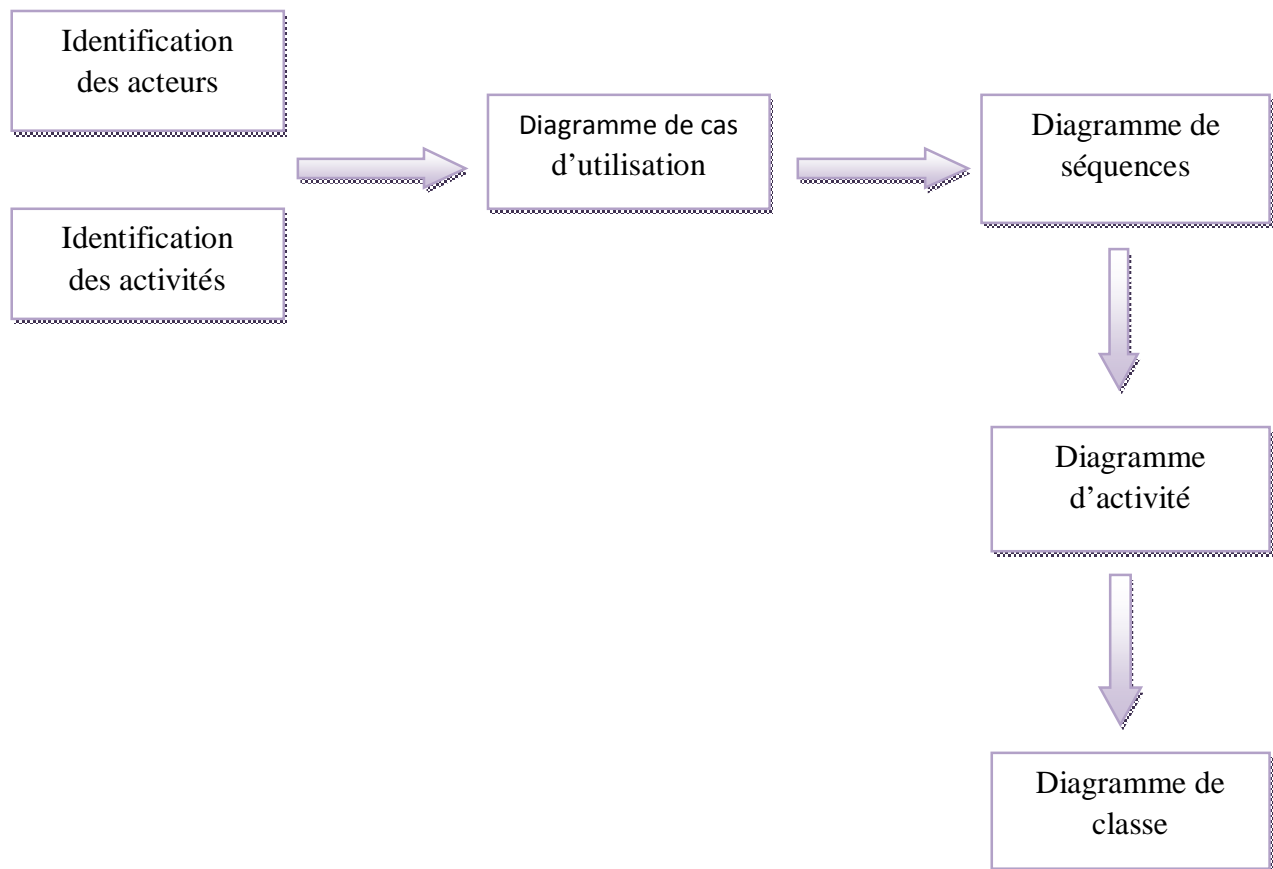


Figure III.1 : démarche de modélisation de l'application.

III.2. Analyse et conception de notre système :

III .2.1. Analyse :

Cette activité commence par la mise en évidence des différents acteurs intervenants dans le système cible ainsi que leurs besoins. Ensuite, la phase conception donnera la modélisation des objectifs à atteindre en s'appuyant sur les résultats de la phase analyse.

III.2.1.1. Problématique :

Nous avons vu dans les chapitres précédents La réussite du tutorat dans le cadre du LMD est liée, à la fois, au contenu de la formation et à la manière dont l'étudiant est mis en contact avec ce contenu en effet les tutorats jouent un rôle très important dans ce système, ils portent une aide et une solution aux problèmes (questions) posées par les étudiants dans les forums, on constate que la plus part de ces questions se répètent (fréquentes).

Vu le nombre important d'étudiants le nombre actuel d'enseignants est insuffisant pour tutorer l'ensemble, pour cela l'enseignant doit assurer une orientation en dehors des séances

d'enseignement, a fin de répondre a ces exigences il faut des techniques nouvelles d'enseignement/apprentissage pour favoriser la réussite du plus grand nombre d'étudiants.

Pour mener à bien cette tâche on va introduire un outil de text mining dans un forum pour aider les étudiants durant leur formations en apportant des réponses appropriées et personnalisées sur divers sujets, il est aussi une aide pour l'enseignant qui ne peut répondre a tous ces questionnements.

Notre approche permet de retrouver des réponses aux questions fréquentes contenue dans un forum de discussions ce qui permet d'améliorer les fonctionnalités de la recherche d'information. Ce projet vise à concevoir et à développer un tutorat de questions pour le système LMD.

III.2.1.2. Modèle des cas d'utilisation :

Détaillons les différentes étapes de la démarche que nous allons adopter afin d'aboutir au modèle des cas d'utilisation.

- Identification des acteurs.
- Spécification des tâches et scénarios.
- Identification des cas d'utilisation.
- Ajout des relations entre cas d'utilisation.
- Finalisation des diagrammes de cas d'utilisation.

III.2.1.3. Identification des acteurs :

Les acteurs sont des entités qui représentent des rôles à travers une certaine utilisation (cas). Notre application fait intervenir quatre acteurs :

- Ø **Etudiant**: Est toute personne concernée par le système LMD au sein du site, après avoir finalisé son inscription administrative, l'étudiant ouvre droit a un compte ainsi il devient membre.
- Ø **Tuteur** : son rôle est de répondre aux questions posées par les apprenants (étudiants).
- Ø **Visiteur** : Est toutes personne qui peut avoir accès à une partie restreinte du site (Consultation de questions déjà posées, contacter le gestionnaire, ...).

III.2.1.4 Spécification des tâches :

Les acteurs définis précédemment effectuent un certain nombre de tâches, ces tâches sont résumées dans le tableau ci-dessous :

Acteurs	Taches
Etudiant	T1 : se connecter au site ; T2 :s’authentifier ; T3 : poser une question ; T4 : lire le(s) réponses à sa ou ses questions T5 : rechercher et consulter les questions par catégories, sous catégories. T6 : se déconnecter
Tuteur	T7 : se connecter au site ; T8 :s’authentifier ; T9 : répondre aux questions. T10 : consulter la liste des questions ; T11 : ajouter une nouvelle catégorie de questions; T12 : rechercher dans les catégories de questions ; T13 : se déconnecter ;
Visiteur	T18 : se connecter au site ; T20 : consulter la liste des questions ; T21 : Rechercher dans les catégories de questions ;

Figure III.2 : *spécification des taches*

III.2.1.5. Spécification des scénarios:

▼ Définition des scénarios:

Chaque résultat possible de l’exécution d’un cas d’utilisation est appelé scénario, c’est un chemin logique traversant le cas d’utilisation. Un scénario est une instance de cas d’utilisation. Un ensemble de scénarios pour un cas d’utilisation identifie tout ce qu’il peut arriver lorsque ce cas d’utilisation est en œuvre.

Acteurs	Taches	Scénarios
Etudiant	T1 : Se connecter au site	S1 : Saisir URL dans le navigateur ;
	T2 :s’authentifier	S2 : choisir le statut (étudiant) à partir de la liste statut ; S3 : saisir le pseudo et le mot

		de passe ; S4 : cliquer sur le bouton « valider » ;
	T3 : saisir une question	S5 : saisir la question S6 : cliquer sur le bouton « rechercher » ; S7 : consulter la liste de questions proposées et leurs réponses;
	T4 : consulter la liste des Questions	S8 : cliquer sur le lien « questions » ;
	T5 : rechercher dans les catégories de questions.	S9 : accéder au menu des catégories S10 : choisir la catégorie dans la liste de catégories ; choisir la sous-catégorie ;
	T6 : se déconnecter	S 11: cliquer sur le lien « déconnexion » ;
Tuteur	T7 : Se connecter au site	S 12 : Saisir URL dans le navigateur ;
	T8 :s'authentifier	S 13 : choisir le statut « tuteur » à partir de la liste statut ; S12 :S3, S4 ;
	T9 : répondre aux Questions	S 14 : choisir la question puis saisir la réponse. S15: cliquer sur le bouton « valider » ;
	T10 : consulter la liste des questions ;	S16 : choisir question dans le menu ;
	T11 : ajouter une nouvelle catégorie de questions;	S 17 : cliquer sur le lien ajouter une nouvelle catégorie ; S 18 : Saisir la catégorie, la sous-catégorie et les mots clés puis valider;
	T12 : se déconnecter	S 19 : cliquer sur le lien « déconnexion »

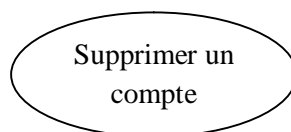
Visiteur	T13 : Se connecter au site	S20 : Saisir URL dans le navigateur
	T14 : consulter la liste des Questions	S21 : cliquer sur le lien « questions »
	T15 : rechercher dans les catégories de questions.	S22 : accéder au menu des catégories S23 : choisir la catégorie dans la liste de catégories ; choisir la sous-catégorie.

Figure III.3 : spécification des scénarios

III.2.1.6. Identification et description des cas d'utilisation :

Les cas d'utilisation ou *use cases*. Un cas d'utilisation modélise un dialogue entre un acteur et le système. C'est la représentation d'une fonctionnalité offerte par le système. L'ensemble des cas d'utilisation forme toutes les façons possibles d'utilisation du système.

Dans UML, le cas d'utilisation est représenté par un ovale comme suit:



Les cas d'utilisation principaux identifiés dans notre système ont été bien mis en évidence par les tâches dans l'étape précédente à savoir : l'ajout d'un acteur, la consultation des catégories de questions,....etc.

On note que chaque cas d'utilisation doit avoir un objectif en soi, et pouvoir être réalisé indépendamment des autres.

Pour chaque acteur identifié précédemment, nous allons rechercher les différentes intentions « métier » selon lesquelles il utilise le système.

L'ensemble des cas d'utilisation est le suivant :

Pour l'**étudiant** :

- Ü s'authentifier.
- Ü poser une question.
- Ü Consulter la liste des questions.
- Ü Consulter les réponses des questions
- Ü Rechercher dans les catégories de questions.
- Ü Se déconnecter.

Pour le **tuteur** :

- Ü s'authentifier.
- Ü Consulter la liste des questions posées.
- Ü Répondre aux questions posées.
- Ü Rechercher dans les catégories de questions.
- Ü Ajouter une nouvelle catégorie de questions.
- Ü Ajouter une nouvelle sous-catégorie de questions.
- Ü Ajouter des mots clés associés aux sous categories.
- Ü Se déconnecter.

Pour le **visiteur** :

- Ü Consulter la liste des questions.
- Ü Consulter liste des réponses.
- Ü Rechercher dans les catégories de questions.

On obtient un diagramme préliminaire (voir **Figure III.4**), en représentant sur un schéma les cas d'utilisation (ovales) reliés par des associations (lignes) à leurs acteurs. Un cas d'utilisation doit être relié au moins à un acteur.

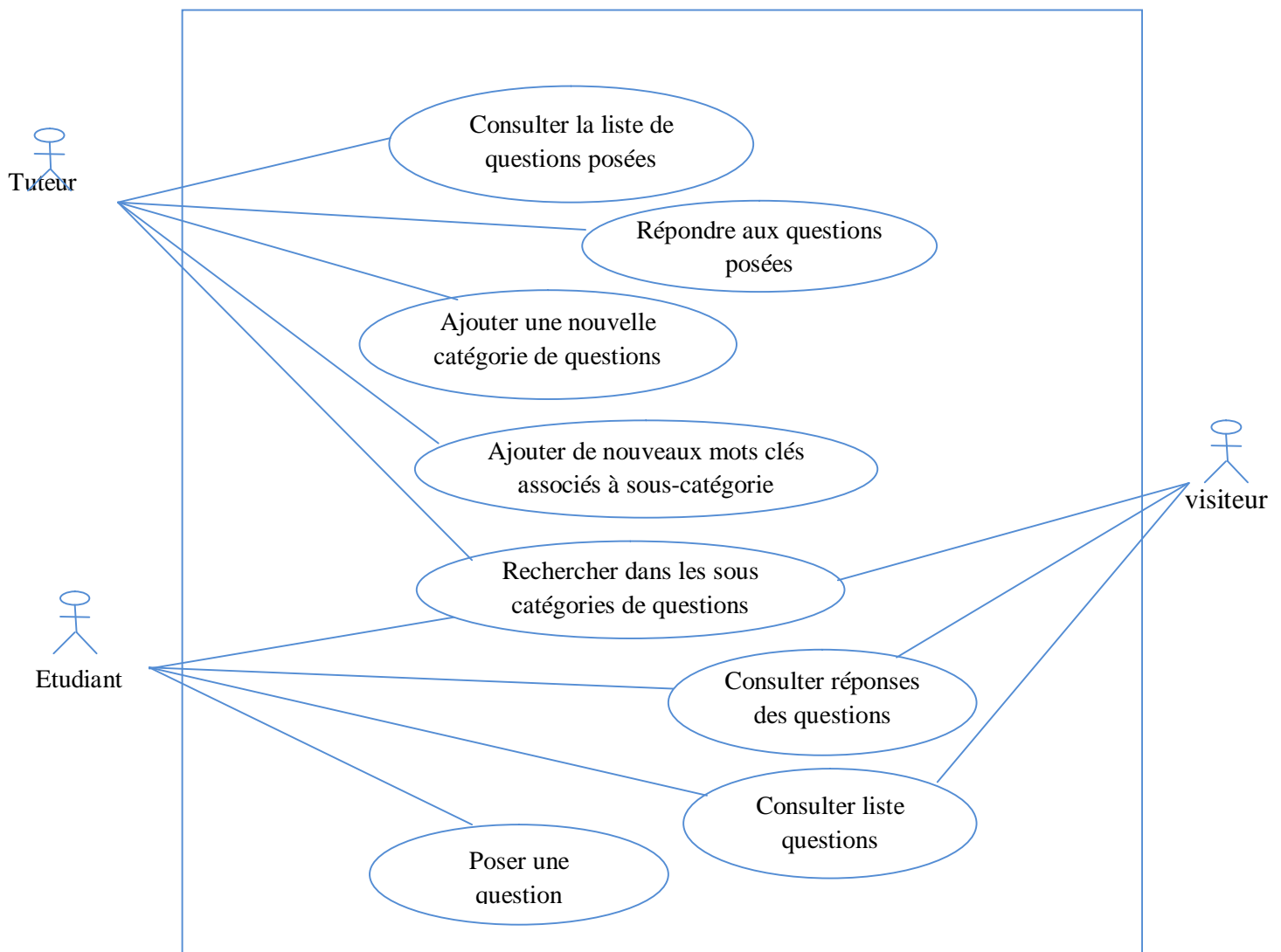


Figure III.4 : Diagramme préliminaire des cas d'utilisation.

III.2.1.7. Ajout des relations entre cas d'utilisation :

Affinement du modèle de cas d'utilisation :

Pour affiner le diagramme de cas d'utilisation, nous allons utiliser trois types de relations standardisées entre cas d'utilisation qu'UML : une relation d'inclusion, une relation d'extension, une relation de généralisation/spécification.

On utilise la relation de généralisation/spécialisation afin de modéliser :

- ∅ Les différentes possibilités de recherche qui sont : la recherche par catégorie, par sous-catégorie et la recherche de questions et réponses.

On utilise la relation de d'extension afin de modéliser :

- Ø L'accès d'un étudiant à son espace peut donner lieu à la saisie d'une question, la consultation de la liste de questions, recherche dans les catégories de questions.
- Ø L'accès d'un tuteur à son espace peut donner lieu à la saisie des réponses pour les questions posées.

Enfin, on pourrait relier les cas d'utilisations par des relations d'inclusion.

- Ø A l'exception des cas d'utilisations du visiteur (se connecter au site, consulter la liste des questions, rechercher dans les catégories de questions) tous les autres cas nécessitent une authentification des acteurs (étudiant, tuteur, gestionnaire) Donc nous allons créer un cas d'utilisation que nous appelons « Authentification », qui doit être réalisé afin de permettre à ces acteurs d'exécuter leur propres cas d'utilisation.

III.2.1.8.Finalisation des diagrammes de cas d'utilisation :

Les cas d'utilisation constituent une technique qui permet de déterminer les besoins des utilisateurs et de capturer les exigences fonctionnelles d'un système. En d'autres termes ils décrivent le comportement d'un système de point de vue de ses utilisateurs. Ils décrivent les interactions entre les utilisateurs d'un système et le système lui-même.

Les cas d'utilisation peuvent être structurés. En plus de la relation de communication, qui consiste au déclenchement d'un cas d'utilisation par un acteur, les deux types de liens ou relations qui sont les plus utilisés : le lien d'utilisation et le lien d'extension.

- Ø **Le lien d'utilisation :** Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. Cette relation permet de décomposer des comportements et de définir les comportements partageables entre plusieurs cas d'utilisations.
- Ø **Le lien d'extension :** Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à des conditions.

A) Diagramme des cas d'utilisation pour le tuteur :

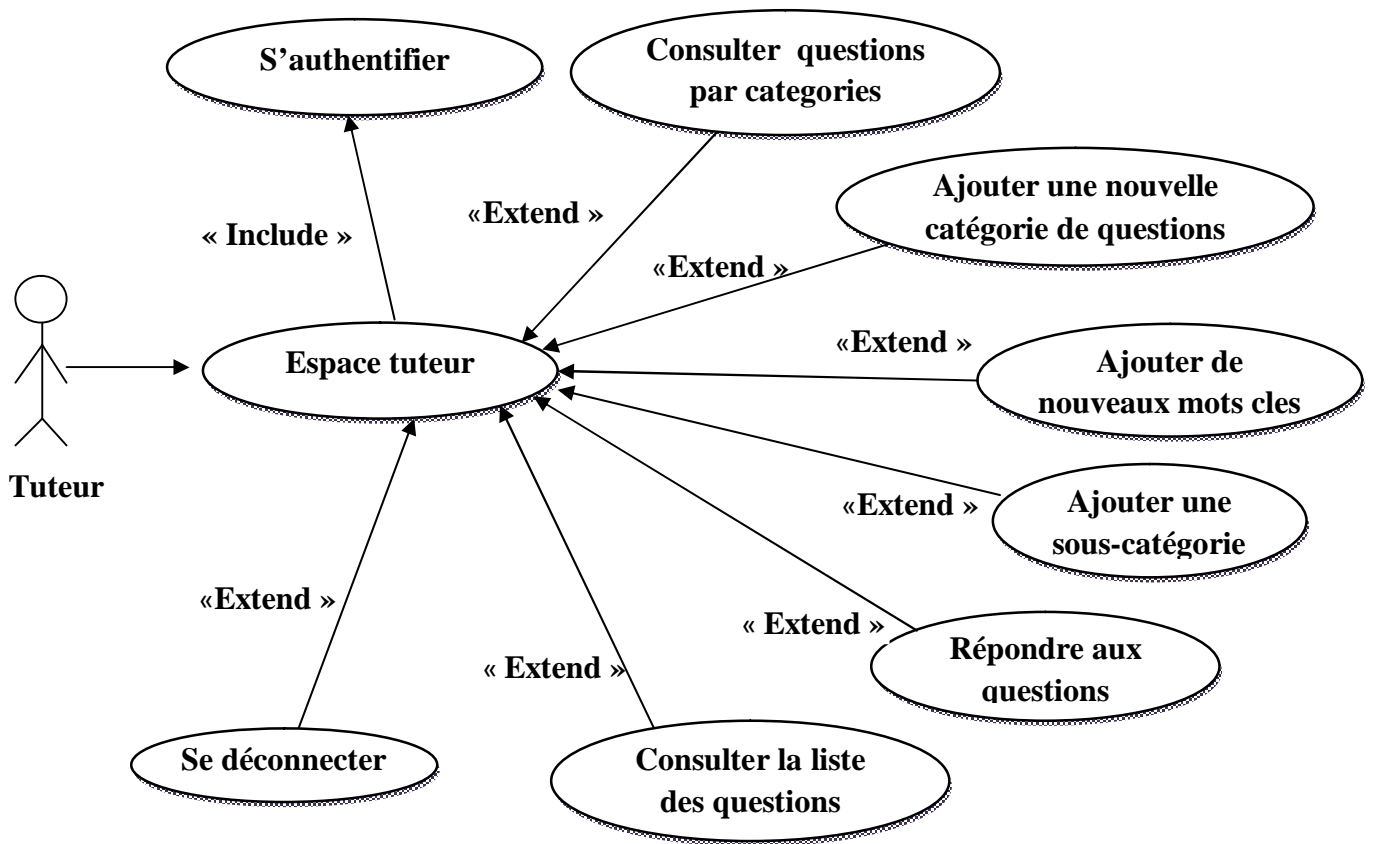
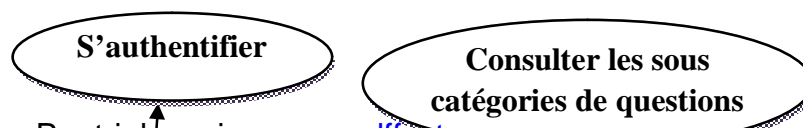


Figure III.5: Diagramme des cas d'utilisation : «tuteur »

B) Diagramme de cas d'utilisation pour l'étudiant :



III.2.2. Conception :

La conception est la phase la plus complexe du projet, surtout avec la fréquence des évolutions chronologiques. Dans cette phase une nouvelle vue du modèle fait son apparition. Cette vue exprime les modules et les exécutable physiques sans aller à la réalisation concrète du système.

III.2.2.1. La conception d'application web :

La conception d'application Web se distingue d'autres systèmes par deux activités majeurs, à savoir la répartition des objets sur le client et serveur et la définition de l'interface utilisateur sous forme de pages Web. Une application web implique l'existence d'au moins trois composants d'architecture significatifs :

- § Le navigateur client ;
- § Le serveur web ;
- § Le serveur d'application ;

III.2.2.2. Elaboration des diagrammes de séquence :

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur : saisir une donnée, consulter une donnée, lancer un traitement ; il met en évidence les objets manipulés ainsi que les opérations qui font passer d'un objet à l'autre. Dans notre cas on s'intéresse seulement à effectuer la représentation du diagramme de séquence pour quelques cas d'utilisation.

Ø Diagramme de séquence de réalisation de cas d'utilisation «s'authentifier»

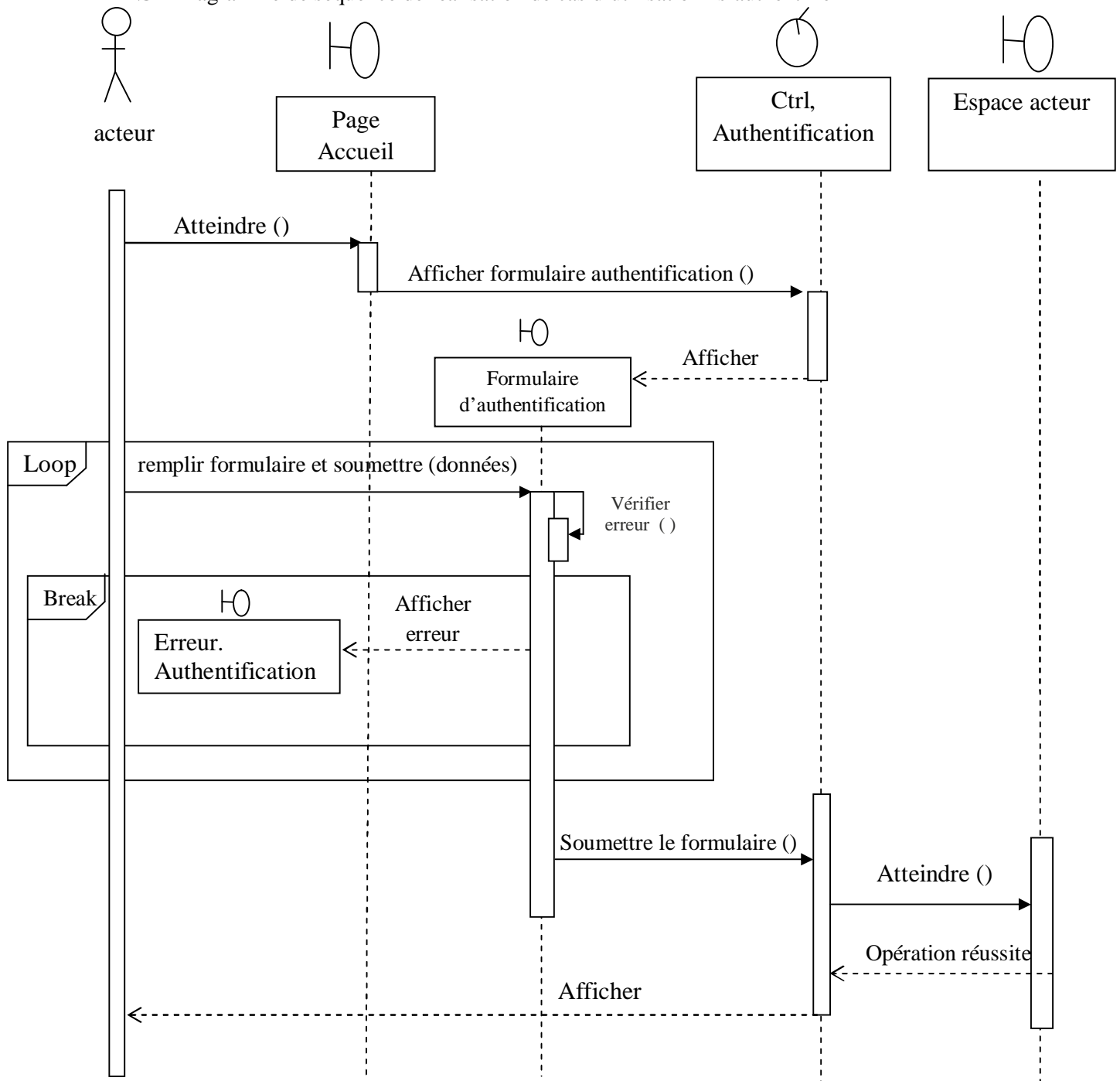


Figure III.7 : Diagramme de séquence du cas d'utilisation « s'authentifier ».

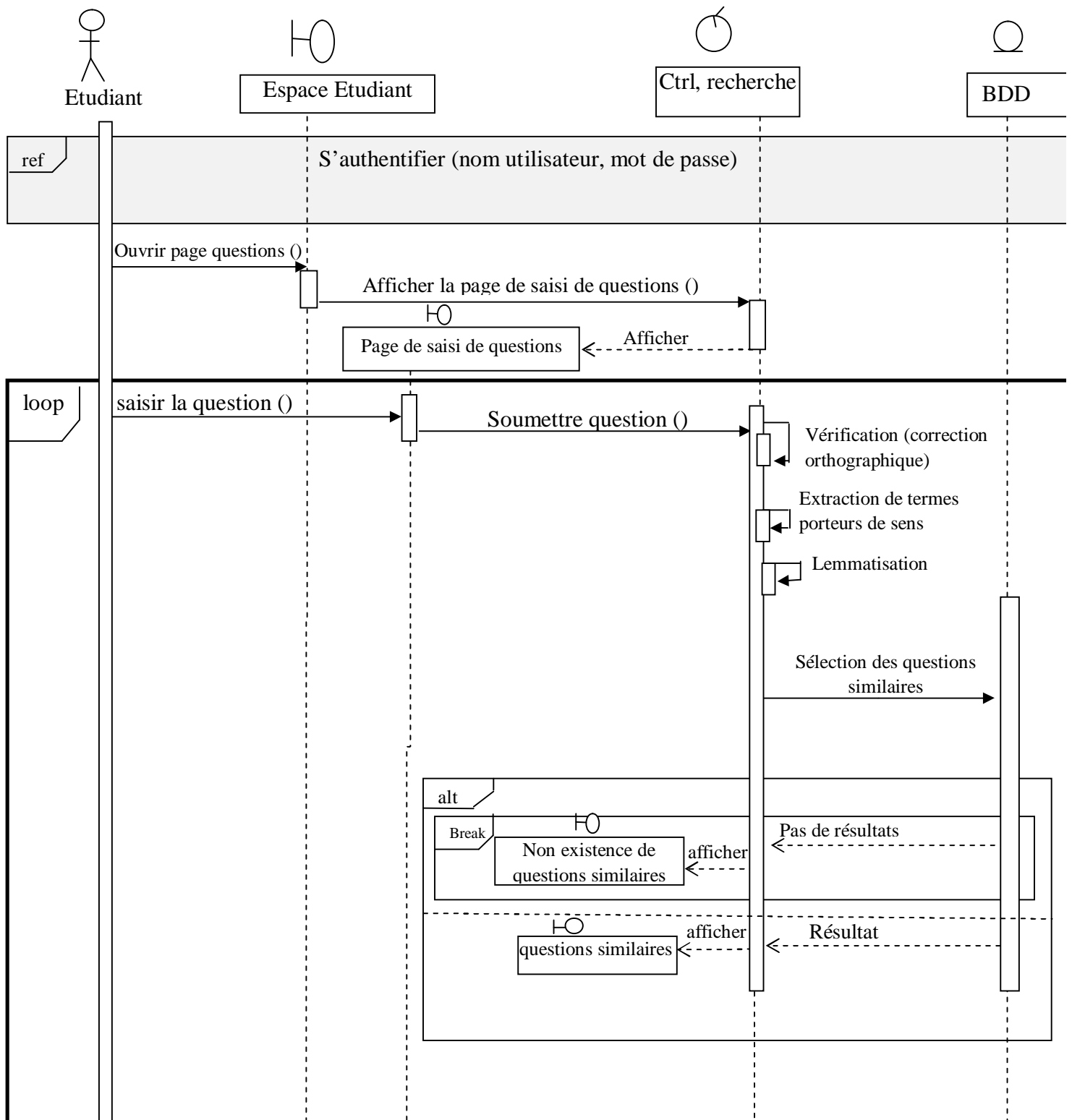


Figure III.8 : Diagramme de séquence du cas d'utilisation « poser question ».

Ø Diagramme de séquence de cas d'utilisation «Répondre à une Question »:

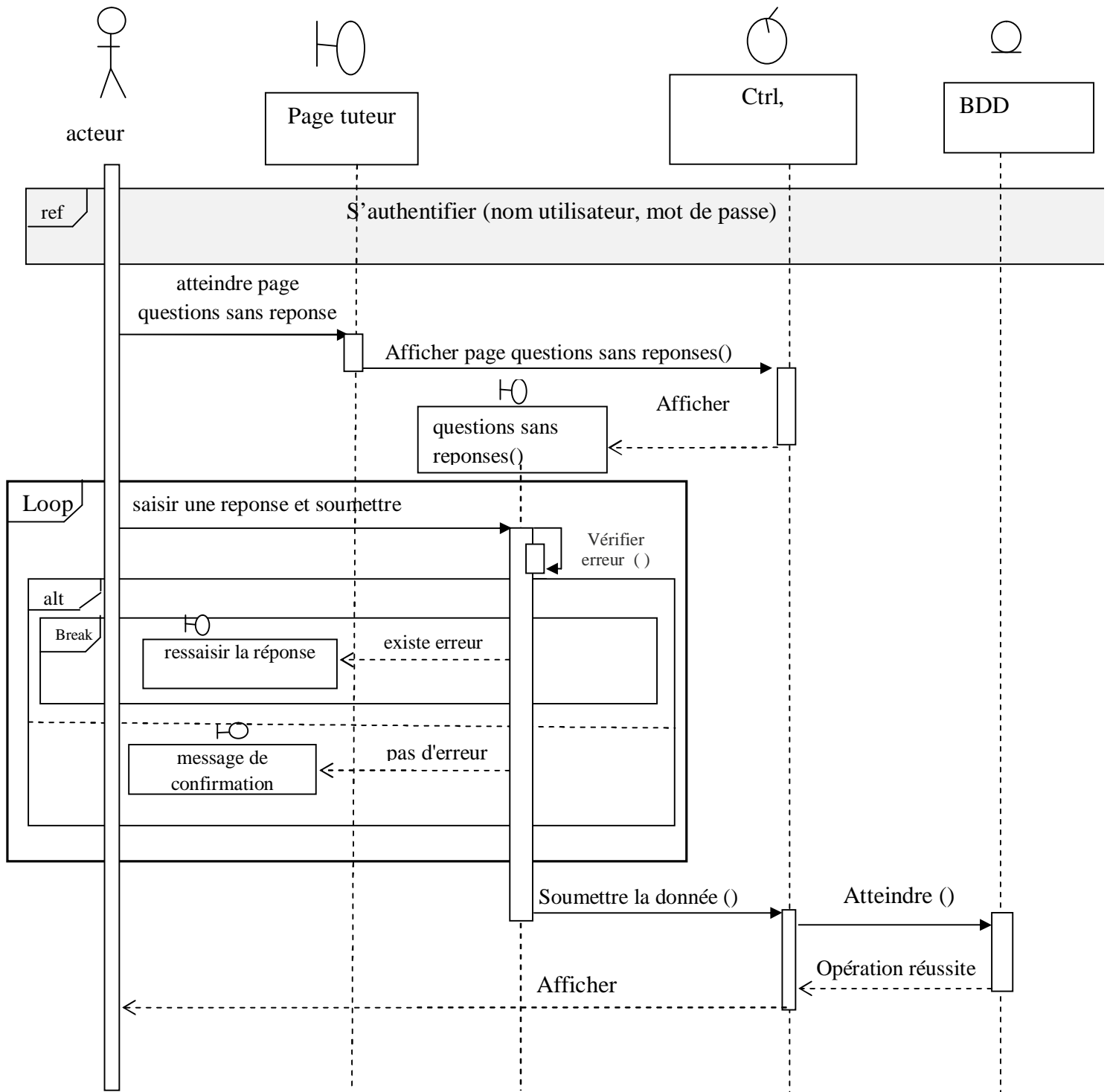


Figure III.9 : Diagramme de séquence du cas d'utilisation «Répondre à une Question ».

III.2.2.3. Diagrammes d'activité :

Ø Diagramme d'activité pour le cas d'utilisation : «s'authentifier».

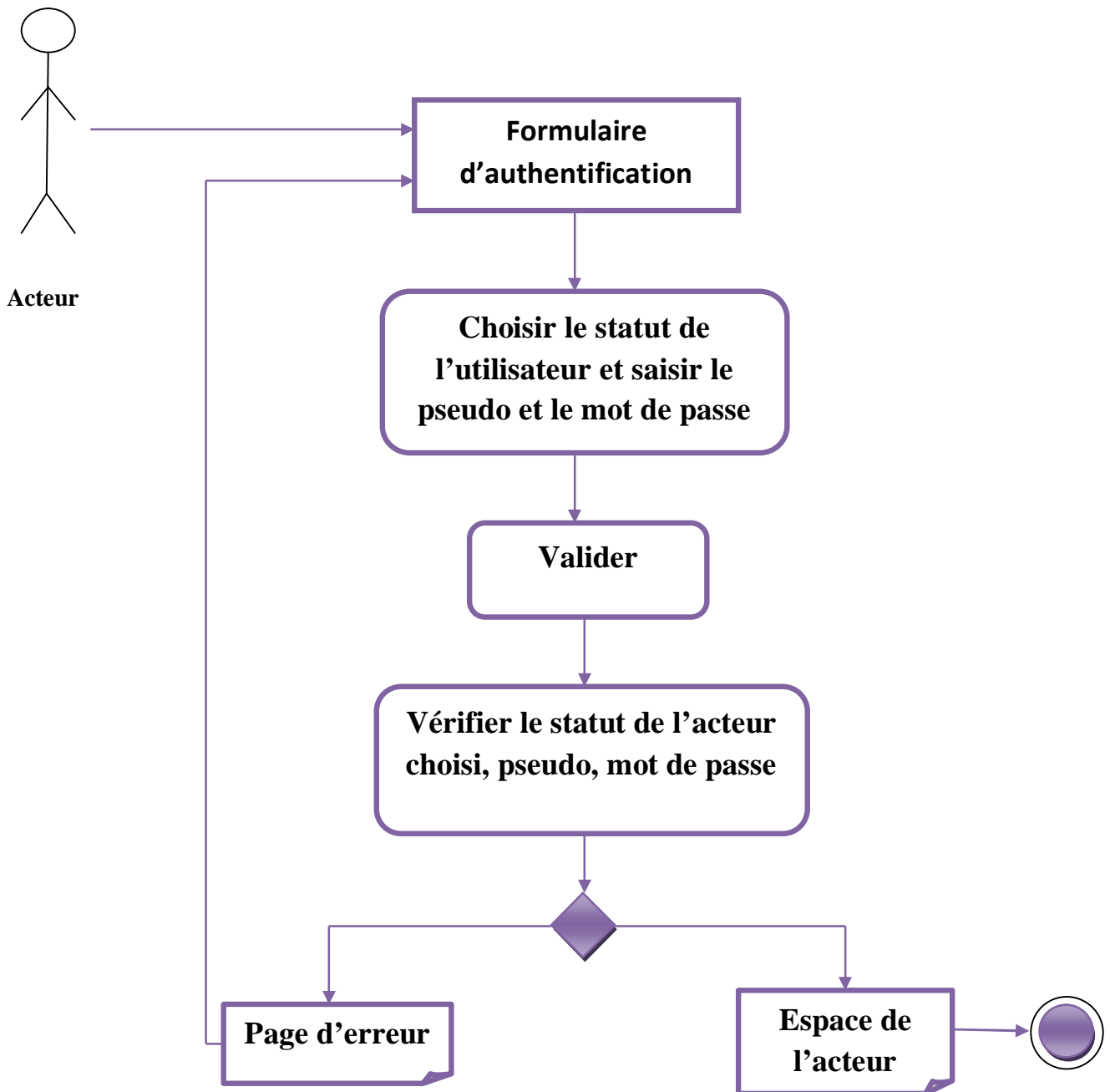


Figure III .10 : Diagramme d'activité pour le cas d'utilisation «s'authentifier».

Ø Diagramme d'activité pour le cas d'utilisation «poser une question».

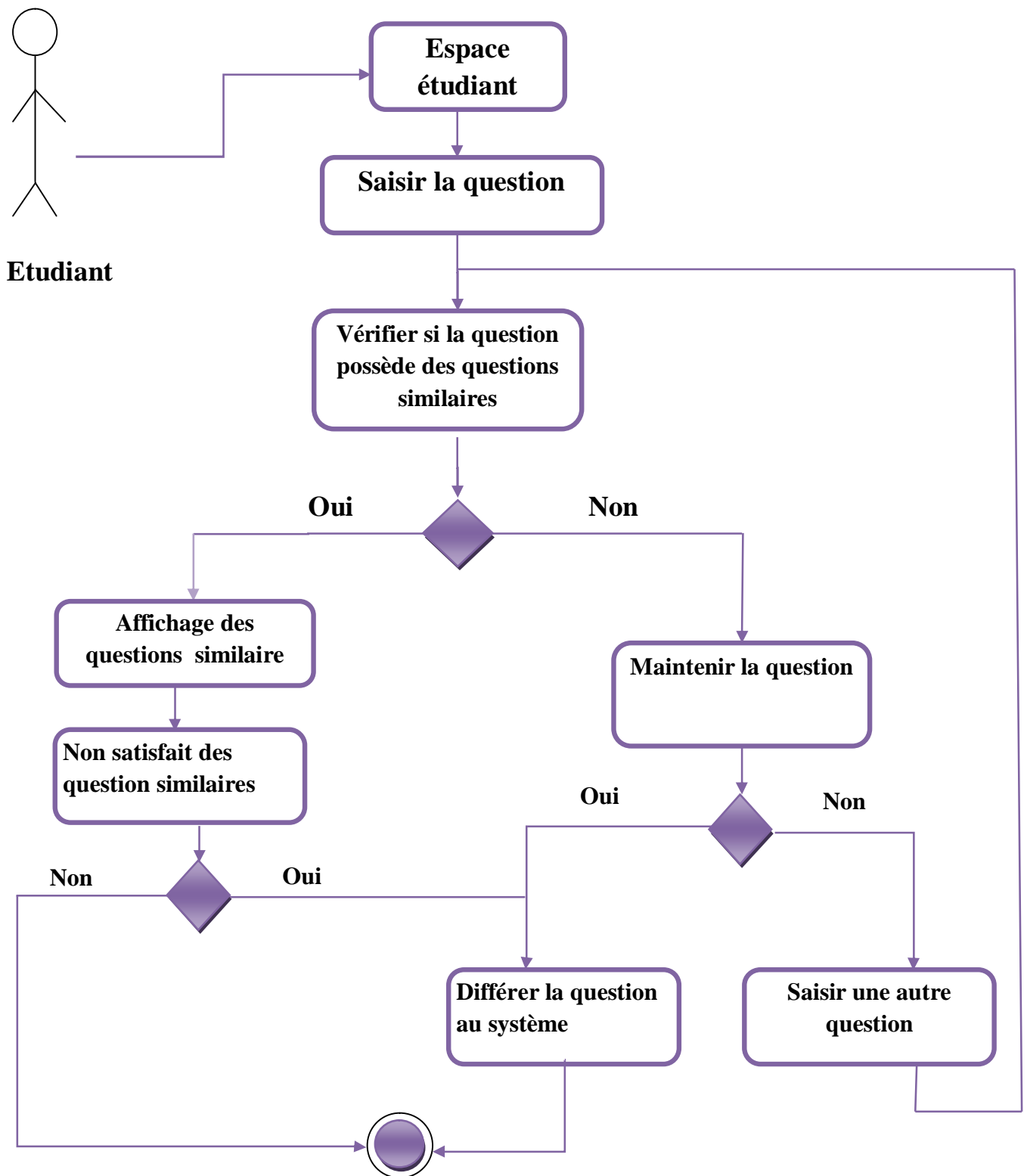


Figure III.11 : Diagramme d'activité pour le cas d'utilisation «poser une question».

Ø Diagramme d'activité pour le cas d'utilisation «Répondre à une question».

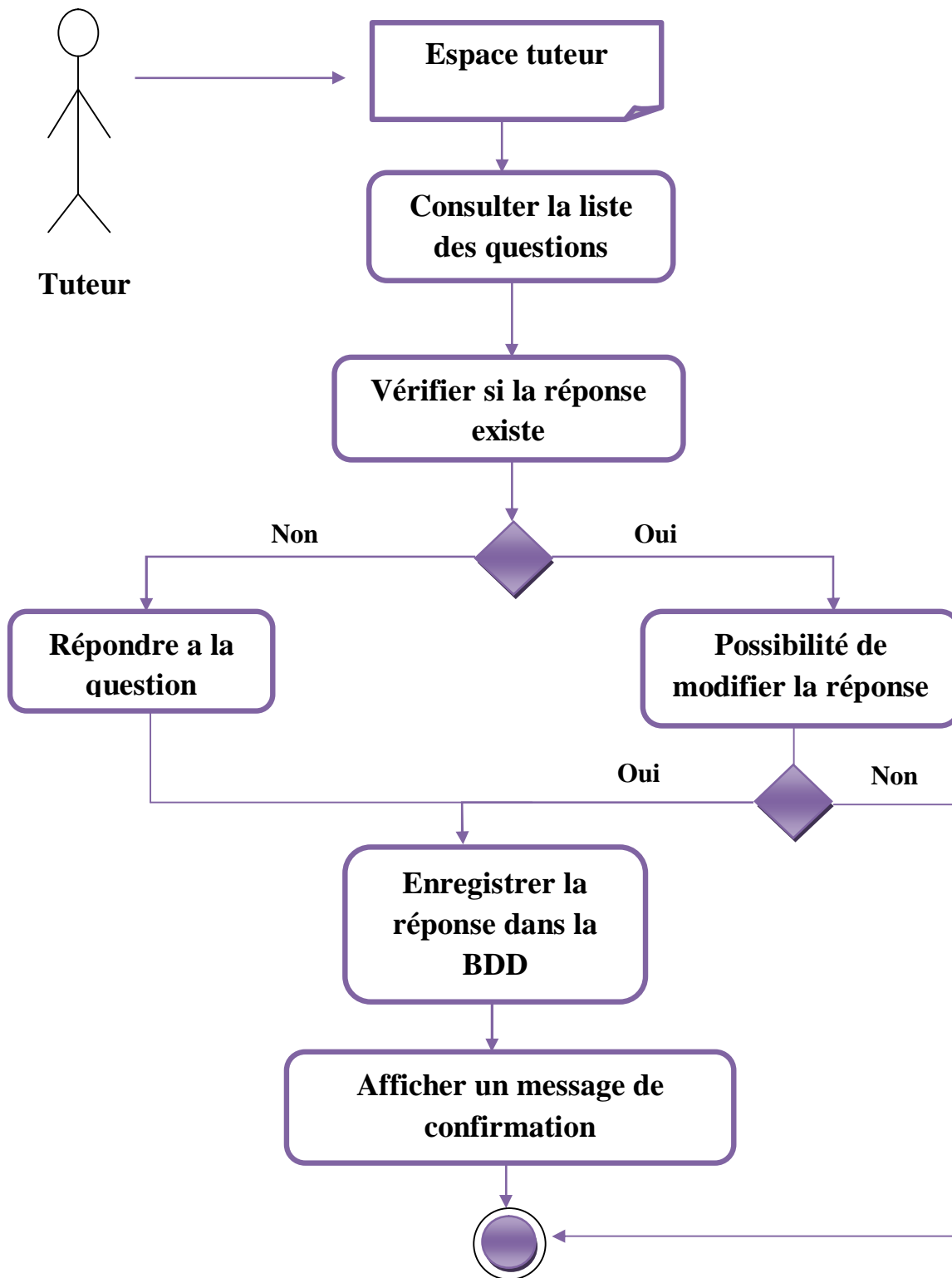


Figure III.12 : Diagramme d'activité pour le cas d'utilisation «Répondre à une question».

III .2.2.4. Diagramme de classe :

Les diagrammes de classes représentent la structure statique d'un système.

Il contient principalement des classes ainsi que leurs associations, mais on peut aussi y trouver des objets.

L'intérêt majeur du diagramme de classe est de modéliser les entités du système.

Dans ce qui suit, nous allons présenter des diagrammes de classes de quelques cas d'utilisation

∅ Diagramme de classe général du cas d'utilisation «Authentification» :

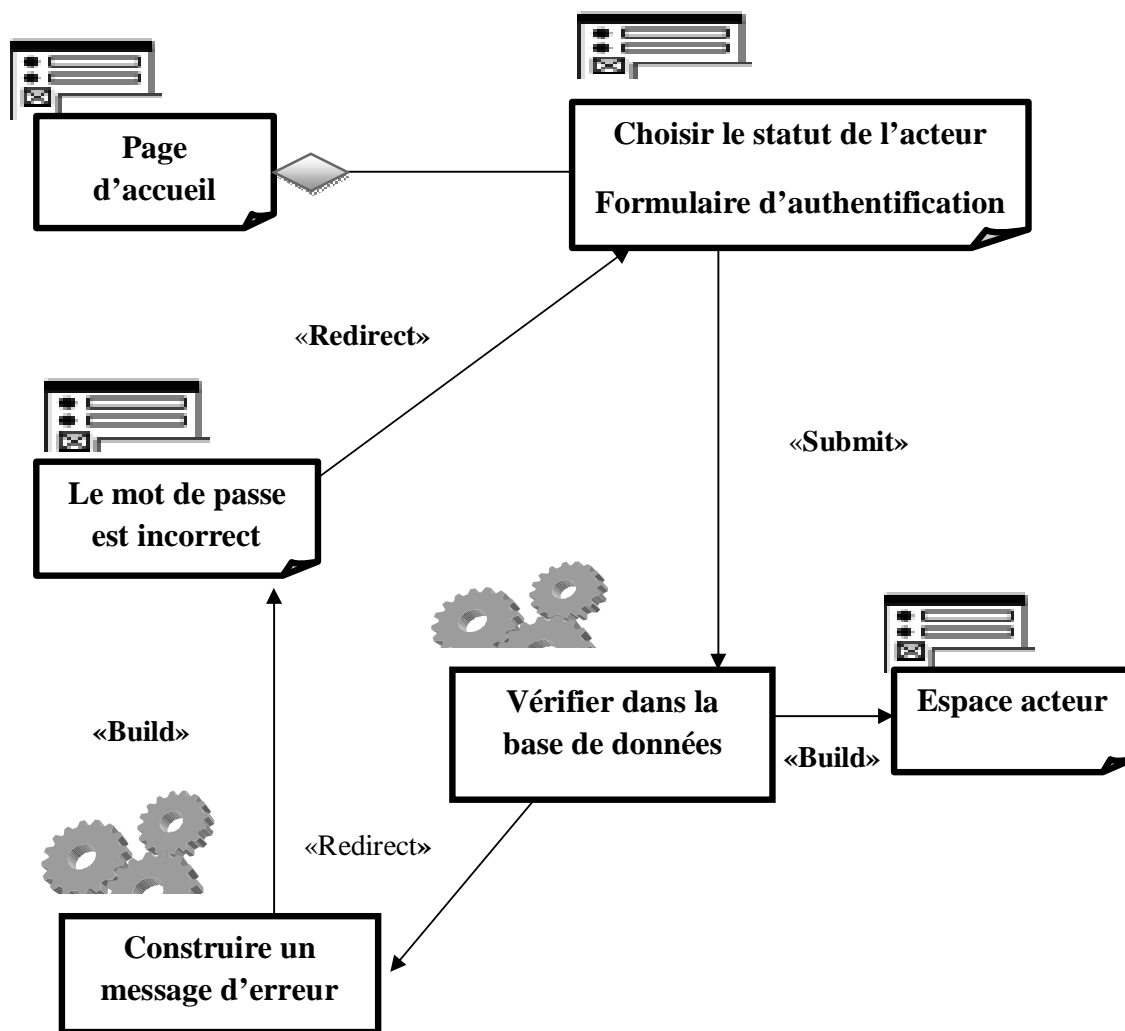


Figure III.13 : Diagramme de classe générale de cas d'utilisation «Authentification».

Ø Diagramme de classe général de cas d'utilisation «poser une question» :

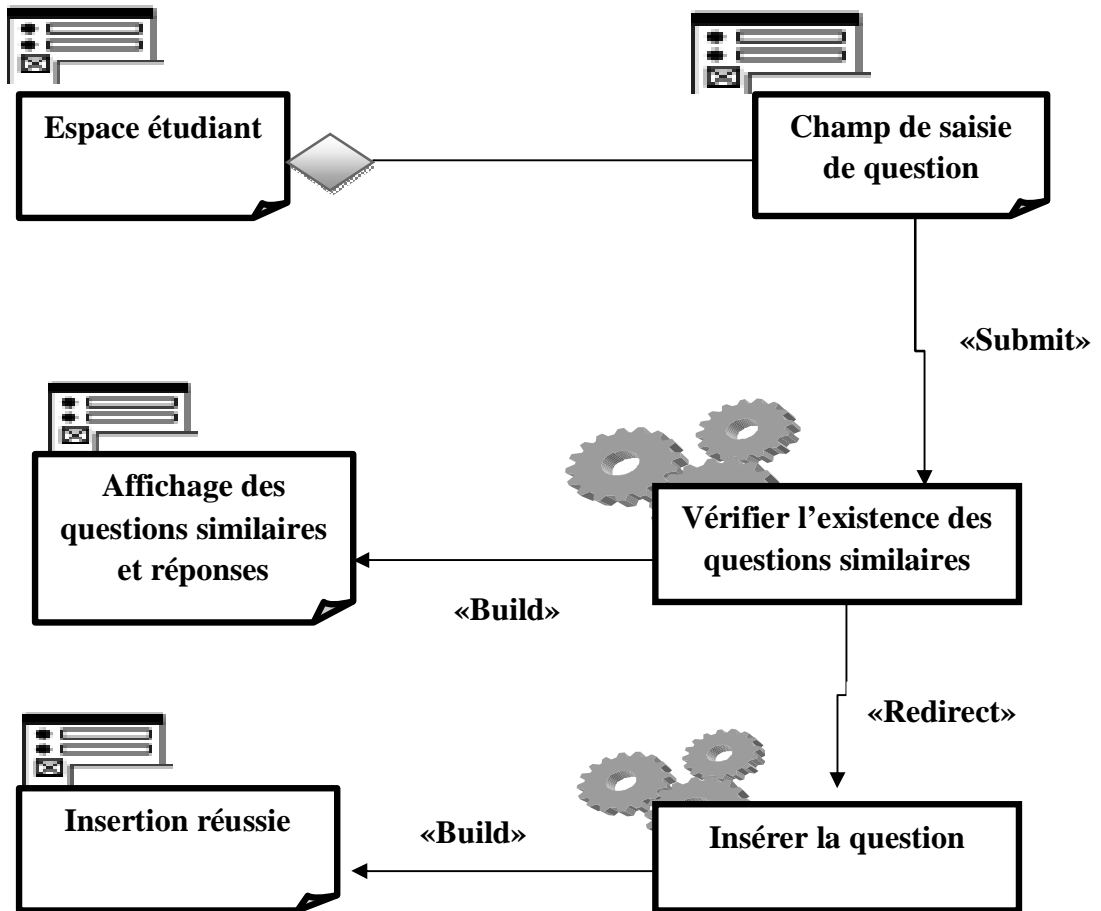


Figure III.14 : Diagramme de classe général de cas d'utilisation «poser une question».

Ø Diagramme de classe général de cas d'utilisation «Répondre à une question» :

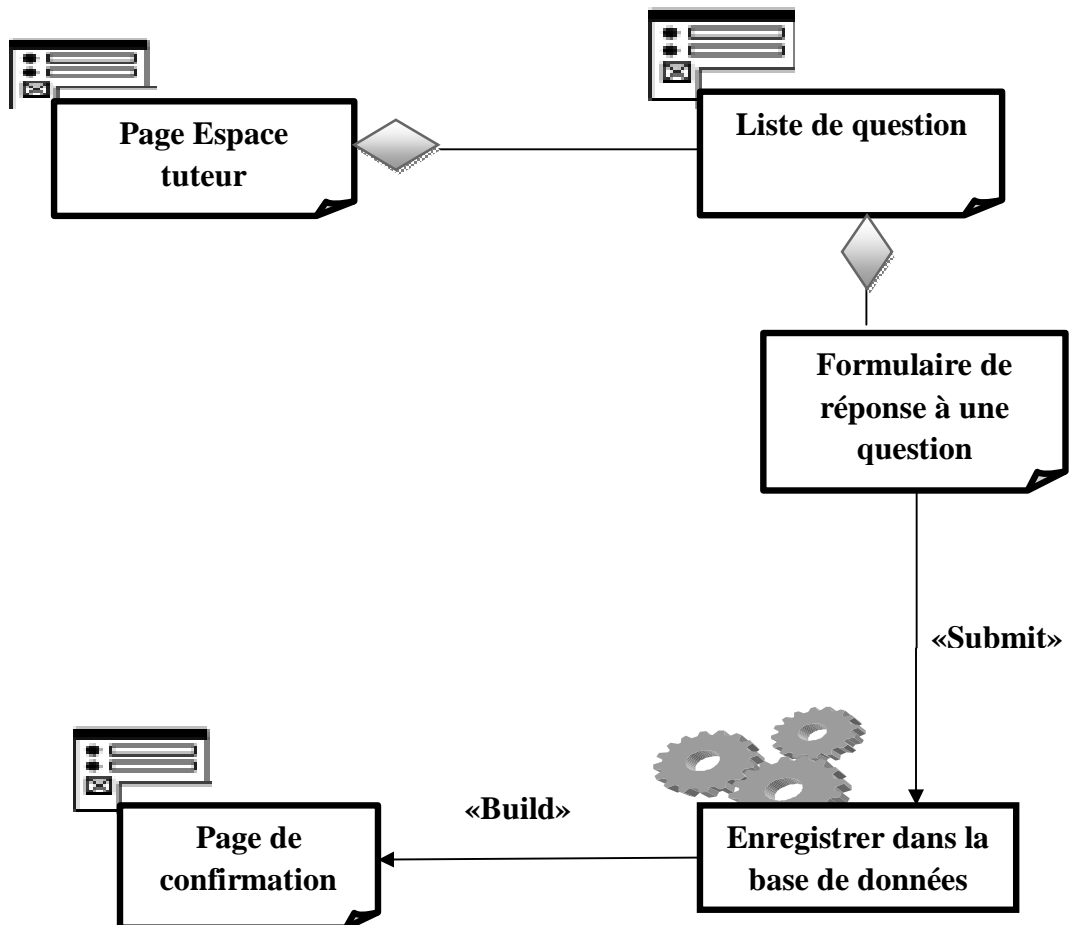


Figure III.15 : Diagramme de classe général de cas d'utilisation «Répondre à une question».

III.2.2.5. Conception de la base de données :

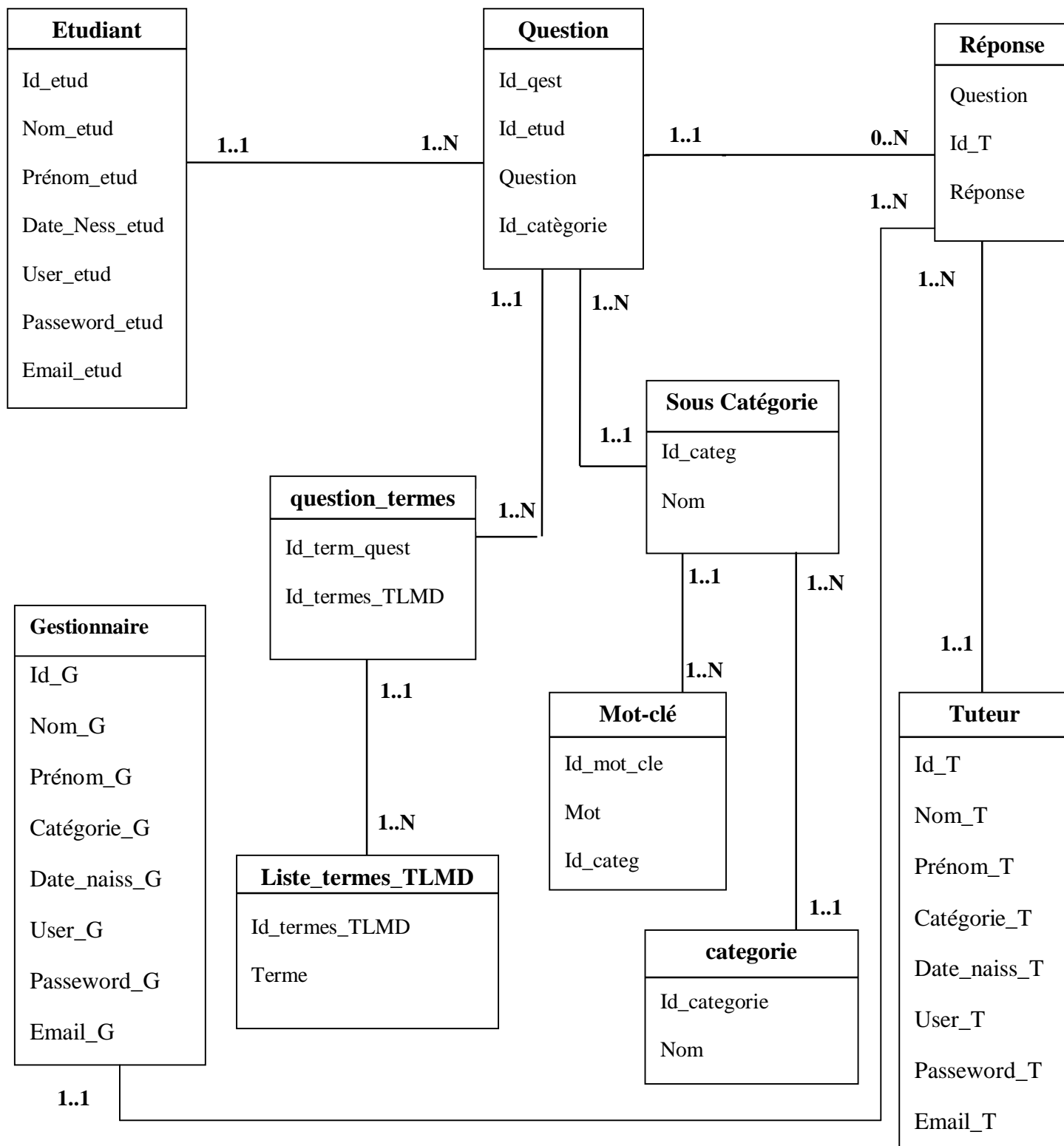


Figure III.16 : Diagramme de classe global

III.2.2.5.a. Le modèle physique des données :

Ce modèle nous donne la représentation physique de l'ensemble des tables de la base des données du système étudié.

✓ définition :

- **clé primaire** : identifie d'une façon unique une ligne dans une table ;
- **clé étrangère** : dépend d'une clé primaire située dans une autre table et migrée à partir de cette table ;
- **unique** : index dans lequel deux lignes ne peuvent pas avoir la même valeur.

✓ Structure des tables :

∅ Table étudiant :

Nom du champ	Type	description	Clé	Unique
Id-etud	Int(11)	identifiant étudiant	Clé primaire	OUI
Nom	Varchar(15)	Nom étudiant		NON
Prénom	Varchar(15)	Prénom étudiant		NON
Dat_naiss_etud	Varchar(20)	Date de naissance étudiant		NON
User_etud	Varchar(15)	Pseudo étudiant		NON
Password_etud	Varchar(15)	Mot de passe étudiant		NON
Email_etud	Varchar(45)	L'adresse électronique de l'étudiant		OUI

∅ Table tuteur :

Nom du champ	Type	description	Clé	Unique
Id-T	Int(11)	Identifiant tuteur	Clé primaire	OUI
Nom_T	Varchar(15)	Nom tuteur		NON
Prénom_T	Varchar(15)	Prénom tuteur		NON
Date de naissance	Varchar(20)	Date de naissance tuteur		NON
User_T	Varchar(15)	Pseudo tuteur		NON

Password_T	Varchar(15)	Mot de passe tuteur		NON
Grade	Varchar(45)	Grade du tuteur		NON
Email_T	Varchar(45)	L'adresse électronique de l'étudiant		OUI

∅ Table Gestionnaire :

Nom du champ	Type	description	Clé	Unique
Id_G	Varchar(11)	Identifiant gestionnaire	Clé primaire	OUI
Nom_G	Varchar(15)	Nom gestionnaire		NON
Prénom_G	Varchar(15)	Prénom gestionnaire		NON
User_G	Varchar(15)	Pseudo gestionnaire		NON
Password_G	Varchar(15)	Mot de passe gestionnaire		NON
Email_G	Varchar(45)	L'adresse électronique de l'étudiant		OUI

∅ Table question :

Nom du champ	Type	Description	Clé	Unique
Id-quest	Int(11)	Identifiant question	Clé primaire	OUI
Id-etud	Int(11)	Identifiant étudiant	Clé étrangère	OUI
question	Varchar(80)	Enonce de la question		OUI
Id_S_catégorie	Int(11)	La sous catégorie de la question	Clé étrangère	OUI

∅ Table Réponse :

Nom du champ	Type	Description	Clé	Unique
Id-quest	Int(11)	Identifiant de la question	Clé primaire	OUI
Id-T	Int(11)	Identifiant du tuteur	Clé étrangère	OUI
Réponse	Varchar(100)	La réponse de la question		NON

∅ Table catégorie :

Nom du champ	Type	description	Clé	Unique
Id_catégorie	Int(11)	Identifiant de la catégorie	Clé primaire	OUI
Nom	Varchar(40)	Nom de la catégorie		NON

∅ Table sous_catégories :

Nom du champ	Type	description	Clé	Unique
Id_S_catégorie	Int(11)	Identifiant de la sous catégorie	Clé primaire	OUI
Nom	Varchar(40)	Nom de la sous catégorie		NON
Id_catégorie	Int(11)	Code de la catégorie	Clé étrangère	OUI

∅ Table liste_termes_TLMD :

Nom du champ	Type	description	Clé	Unique
Id-term	Int(11)	Identifiant du terme	Clé primaire	OUI
Terme	Varchar(25)	Le terme		OUI

∅ Table liste_terme_Question :

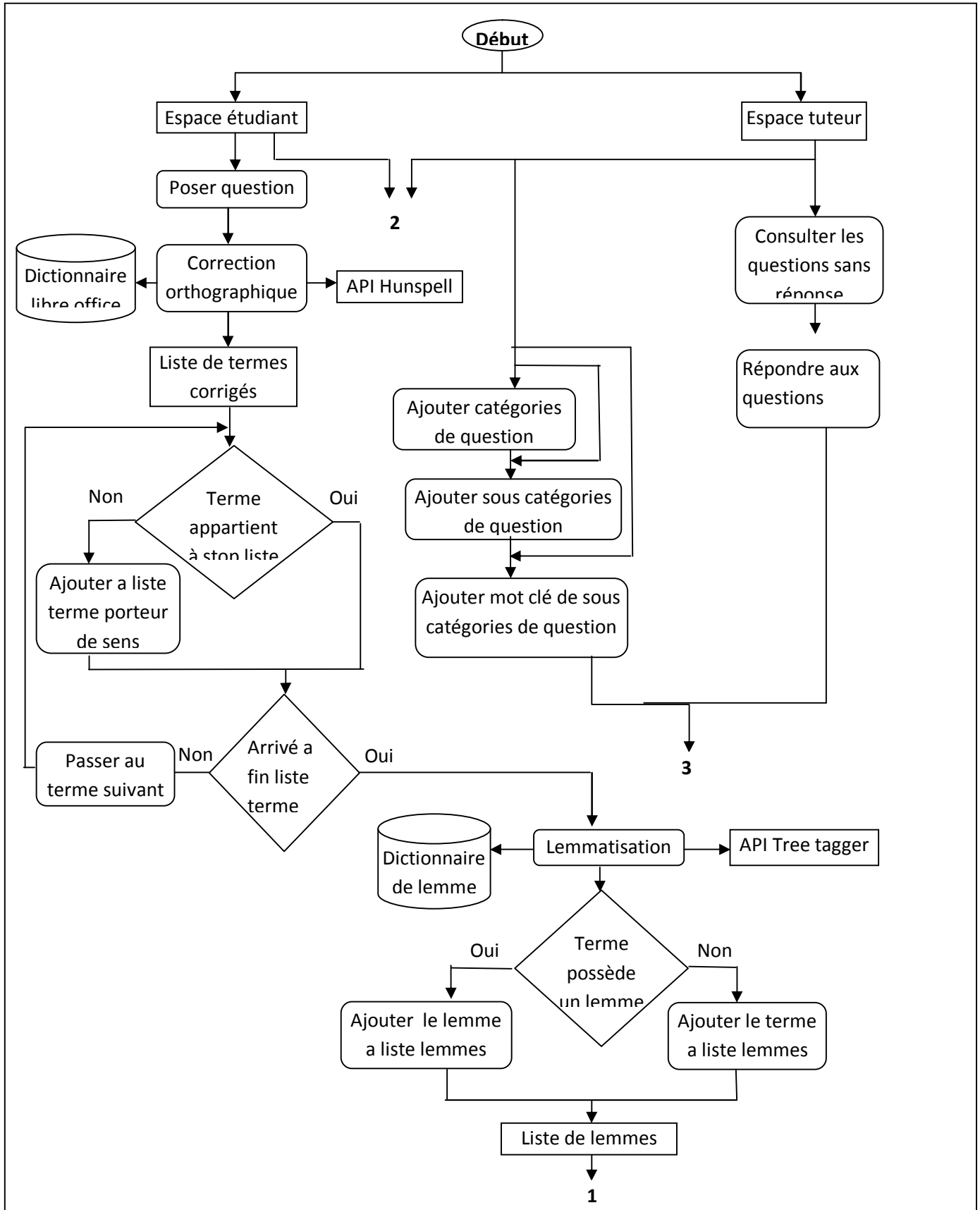
Nom du champ	Type	description	Clé	Unique
Id-term	Int(11)	Identifiant du terme de la question	Clé primaire	OUI
Id-quest	Int(11)	Identifiant de la question	Clé étrangère	OUI

∅ Table Mot_clé :

Nom du champ	Type	description	Clé	Unique
Id_mot_cle	Int(11)	Identifiant du mot clé	Clé primaire	OUI
Mot	Varchar(25)			NON
Id-categ	Int(11)	Code de la catégorie	Clé étrangère	OUI

III.2.6. Etude détaillée de l'architecture du système:

III.2.6.1. Organigramme global du système:



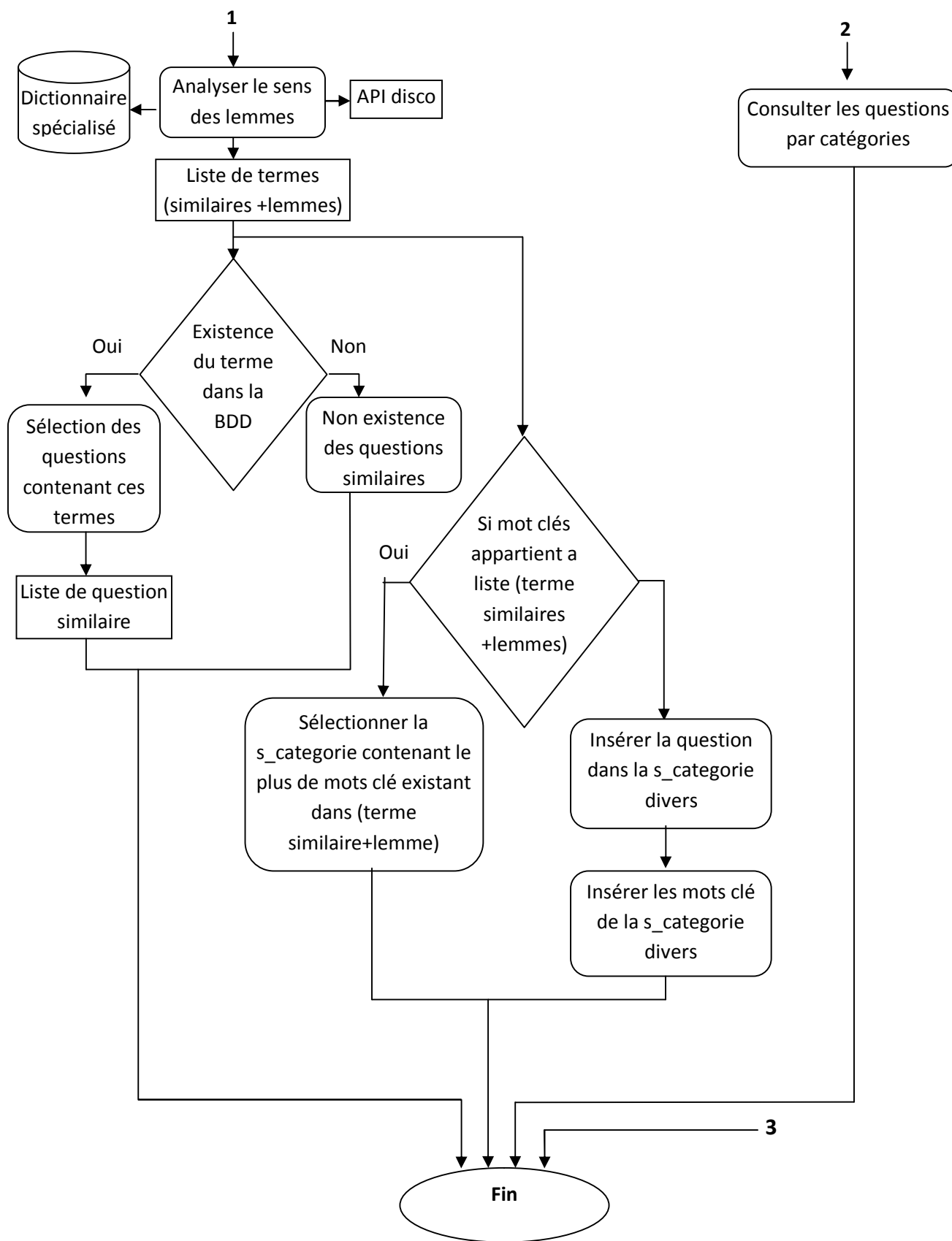


Figure III.17 : organigramme globale du système

III.2.6.2. description des procédures et API utilisés :

Ø Correction orthographique :

La correction orthographique possède plusieurs sens ou on peut la définir comme une action de vérifier et retoucher un texte ou une phrase d'une manière orthographique. ou encore c'est un programme intégré à des logiciels informatiques permettant de corriger les fautes d'orthographe faites par l'utilisateur lors de la saisie d'un texte ou d'une phrase.

La correction orthographique peut se faire de manière manuelle ou de manière automatique. Afin de faire cette correction automatique, cela nécessite un moteur et un ou plusieurs dictionnaires. Les moteurs possibles sont spell, ispell, aspell, myspell et hunspell. Ce dernier étant le plus récent et complet, de plus en plus des logiciels l'utilisent. - **Hunspell** est le correcteur orthographique Open Source de référence à l'heure actuelle. Il s'agit du correcteur utilisé par des logiciels comme Firefox, Chrome, Opera, OpenOffice, LibreOffice, Thunderbird, etc. C'est un logiciel qui a pris la suite de Ispell, lui-même ayant succédé à Aspell, lui-même à Myspell. L'amélioration apportée par Hunspell est l'encodage des dictionnaires en UTF-8 à la place de l'encodage en ASCII utilisé par Myspell. Hunspell a l'avantage de s'appuyer sur une large communauté qui met à jour différents dictionnaires dans différentes langues.

D'un point de vue informatique, pour la plate-forme Windows, Hunspell se présente sous la forme d'une « dll », résultat de la compilation d'un programme écrit en langage C. Par le biais d'un module JNA (Java Native Access), cet exécutable peut être intégré dans un programme Java.

- **LibreOffice** est une suite bureautique libre et gratuite, contenant un dictionnaire pour les corrections orthographique, ce dernier peut être interrogé par des API de correction orthographique telle que **Hunspell**.

Ø Extraction de termes porteurs de sens :

C'est l'exclusion des mots vides qui n'apportent pas de sens lors de l'analyse lexicale d'un texte et lors de la recherche. il s'agit d'éliminer par exemple :

- Les pronoms pronominaux : il, elle, je, lui, ceux....
- Les auxiliaires : avoir, être et toutes leurs conjugaisons possibles
- Les opérateurs de conjonction : avec, et, ou, aux...
- L', c' ...

Un mot vide désigne tout élément d'un texte n'ayant pas de référence concrète ou notionnelle dans la réalité et dont le rôle est d'assurer la liaison ou la cohérence entre les principales parties d'un texte. Il s'agit des articles, des conjonctions, des adjectifs non qualitatifs, des prépositions et de la ponctuation.

Les mots vides sont gérés dans des **bases de données** à l'aide d'objets appelés **listes de mots vides** (stop List). Une liste de mots vides est une liste qui est associée à un index de recherche en texte intégral, s'applique aux requêtes de texte intégral sur cet index.

Ø Lemmatisation :

Il s'agit d'une procédure qui permet de ramener un mot portant des marques de flexion (par exemple, la forme conjuguée d'un verbe) à sa forme de référence (dite lemme). Autrement dit, il s'agit, dans le cadre d'une analyse de texte par ordinateur ou d'un traitement automatique de la langue de manière générale, de reconnaître un mot par rapport à sa forme de base, et cela, quelle que soit la forme sous laquelle il apparaît dans un texte.

Exemple : lemmatisation (sèches) = sec.

L'outil utilisé pour la lemmatisation est TreeTagger [Schmid, 1994], est un outil qui permet d'analyser une phrase ou un texte syntaxiquement afin de remplacer les verbes par leurs formes infinitives et les noms par leurs formes singulières. Il attribue donc à chaque mot sa catégorie grammaticale.

L'algorithme développé est efficace pour les langues anglaise, française, allemande et italienne. Dans cet algorithme, l'analyse grammaticale est effectuée en utilisant des arbres de décisions puis des fichiers de paramètres spécifiques à chaque langue.

Dans notre proposition nous utilisons TreeTagger pour détecter les racines des termes.

Il existe d'autres outils comme KEA de Jones et Paynter, TERMS de J. Justeson et S. Katz [Justeson & Katz, 1995] etc.

Ø Synonymie :

La synonymie est un rapport de similarité sémantique entre des mots ou des expressions d'une même langue. La similarité sémantique indique qu'ils ont des significations très semblables. Des termes liés par synonymie sont des synonymes. Des mots sont dits synonymes lorsqu'ils peuvent être remplacés l'un par l'autre dans un contexte, sans modification majeure du sens.

Un postulat explique qu'il ne peut pas exister de synonymes parfaits; sinon, la langue ferait disparaître l'un des termes alors inutile. Dans les faits, il y a toujours une différence, bien que minime, entre des synonymes, qu'elle soit liée au signifié lui-même, aux connotations véhiculées, au registre de langue ou encore au contexte d'emploi des mots. Strictement parlant, il s'agit donc de para synonymes. En règle générale, le lien de similarité réside dans la seule dénotation des termes liés.

Le choix des mots est rarement indifférent. Les sens des mots, même synonymes dans le même contexte, ne se recouvrent jamais exactement l'un l'autre.

DISCO: Est une classe Java qui permet de récupérer la similarité sémantique entre les mots. Les similitudes sont basées sur l'analyse statistique de très grandes collections de textes. L'outil fonctionne sur tous les systèmes d'exploitation, y compris Windows, Linux, Solaris et MacOs. L'API Java fournit les méthodes suivantes: récupérer la plupart des mots sémantiquement semblables pour un mot d'entrée, récupérer la valeur de la similarité sémantique entre deux mots d'entrée, récupérer collocations pour un mot d'entrée

DISCO peut être intégré dans vos propres applications en utilisant l'API Java. L'API Java fournit plusieurs méthodes pour chercher des mots sémantiquement similaires, la similarité sémantique entre les mots, les collocations, les fréquences corpus, etc.

III.2.6.3. Description de l'analyse linguistique de la question:

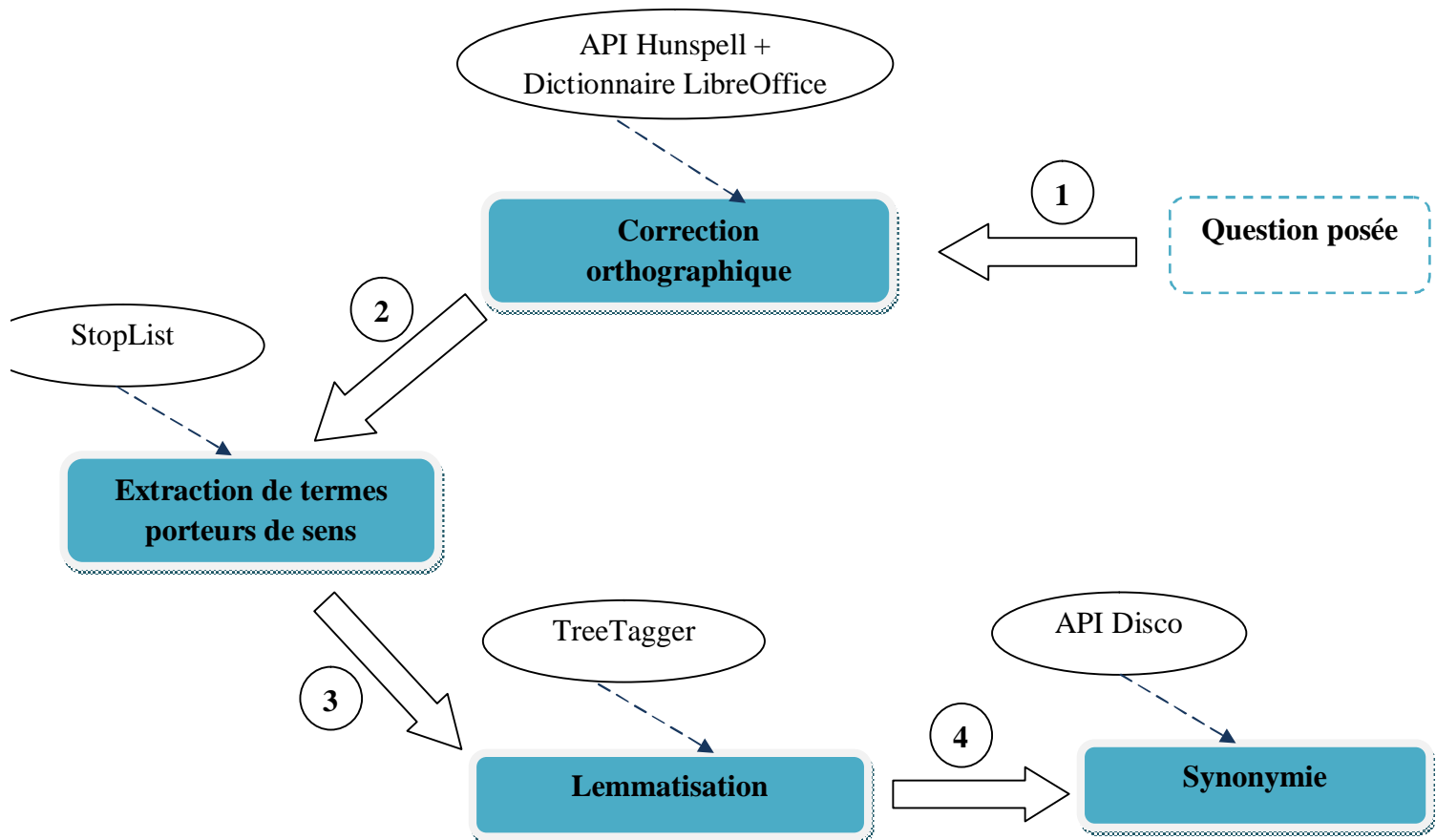


Figure III.18 : description de l'analyse linguistique de la question

- ① : Les erreurs d'orthographe dans la question posée seront corrigées à l'aide de l'API Hunspell et le dictionnaire libre Office et on aura comme résultat une question corrigée.
- ② : L'extraction des termes porteurs de sens a partir de la question corrigée et ceci par l'exclusion des mots vides, ces derniers ce sont les mots appartenant a la stop List.
- ③ : Les termes extraits a partir de la question corrigée seront lemmatisés à l'aide de l'API TreeTagger, une liste de mots appelés "lemmes" sera le résultat.
- ④ : L'analyse du sens des lemmes et l'extraction des synonymes a l'aide de l'API Disco et un dictionnaire spécialisé. Le résultat est une la liste des lemmes et leurs synonymes, ils seront comparés avec la liste des termes dans la base de données a fin d'extraire les questions contenant les termes en commun et calculer ensuite les seuils de similarité

III.2.6.4. l'algorithme globale:

Correction orthographique :

Fonction correction (chaîne : questionQ) :chaîne

Début

Charger le dictionnaire (libre office) à l'aide de l'API Hunspell;

Initialiser terme de dictionnaire: T_i ;

Initialiser chaîne: question_corrigeé;

Parcourir question et extraire T termes de questionQ ;

Pour (chaque terme T de questionQ) faire

Si terme $T \notin$ dictionnaire et $T \notin$ liste abréviations

Sélectionner le terme T_i le plus proche de T ;

Remplacer T par T_i dans question ;

Finsi

Fait

Retourner question_corrigeé;

Fin

Extraction des termes porteurs de sens :

Fonction Extraction_termes(chaine: questionQ) : liste

```

Début
  Initialiser liste: liste_terme ;
  Charger la liste des mots vides (stoplist) ;
  Pour (chaque terme T dans questionQ) faire
    Si T ∉ stoplist
      Insérer T dans le vecteur liste_terme ;
    Finsi
  Fait
  Retourner liste_terme ;
Fin

```

Lemmatisation des termes:

fonction lemmatisation (liste : liste_termes) : liste

```

Début
  Initialiser liste: liste_lemmatisée ;
  Charger le dictionnaire des lemmes avec l'API Treetager ;
  Initialiser terme Ti collection des lemmes ;
  Pour (chaque terme T dans liste_termes) faire
    Si T possède un lemme Ti alors
      liste_lemmatisée  $\mathbf{\beta}$  liste_lemmatisé + Ti
    sinon
      liste_lemmatisée  $\mathbf{\beta}$  liste_lemmatisée + T ;
    Finsi
  Fait
  Retourner liste_lemmatisée ;
Fin

```

Synonymes d'un mot :**Fonction synonymes (chaîne : terme T) : liste**

Début

Initialiser liste_synonymes ;

Charger les termes similaires de T à partir du dictionnaire en utilisant l'API disco

Insérer les termes similaires dans liste_synonymes ;

Retourner liste_synonymes ;

Fin

calcul de seuil de similarité :**Fonction similarite (chaîne : q1, chaîne: q2) : reel**

Début

long_q1 **↔** longueur (q1) ;long_q2 **↔** longueur (q2) ;

calculer nbre_termes_commun le nombre de termes en commun entre q1 et q2 ;

seuil= (nbre_termes_commun / long_q1 + nbre_termes_commun/ long_q2)/2

Retourner seuil;

Fin

Programme principale

Début

Initialiser chaine: Question_Initial ;

Initialiser chaine: Question_Corrigée ;

Initialiser liste: L_termes_porteur_sens ;

Initialiser liste: L_lemmatisée ;

Initialiser liste: L_synonymes ;

Question_Corrigée $\mathbf{\beta}$ correction (Question_Initial) ;

L_termes_porteur_sens $\mathbf{\beta}$ Extraction_termes (Question_Corrigée) ;

L_lemmatisée $\mathbf{\beta}$ lemmatisation (L_termes_porteur_sens) ;

Pour (chaque terme T1 de L_lemmatisée) faire

 L_synonymes $\mathbf{\beta}$ synonymes(T1) + T1 ;

Fait

Sélectionner la table termes_Tutorat à partir de la base de données

Pour (chaque terme T2 de L_synonymes) faire

 Si (T2 \in termes_Tutorat) alors

 Récupérer toutes les question Q qui contient le terme T2 ;

 Liste_questions $\mathbf{\beta}$ Liste_questions + Q ;

 Pour (chaque question Q de Liste_questions) faire

 calculer le seuil de similarité de Q par rapport a la question posée (Question_Initial)

 seuil_similarite = similarité (Q, Question_Initial)

 fait

 Trie décroissant des seuils de similarité des questions, la question contenant le plus grand seuil c'est la plus similaire

 Finsi

Fait

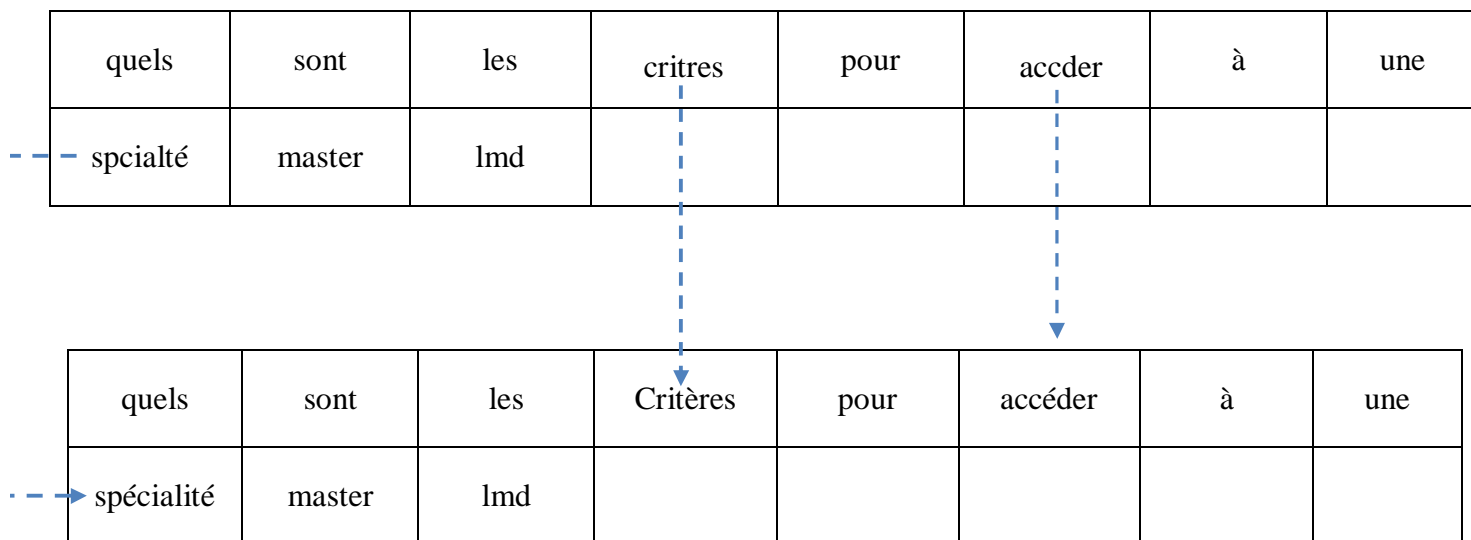
Fin.

Exemple d'analyse d'une question:

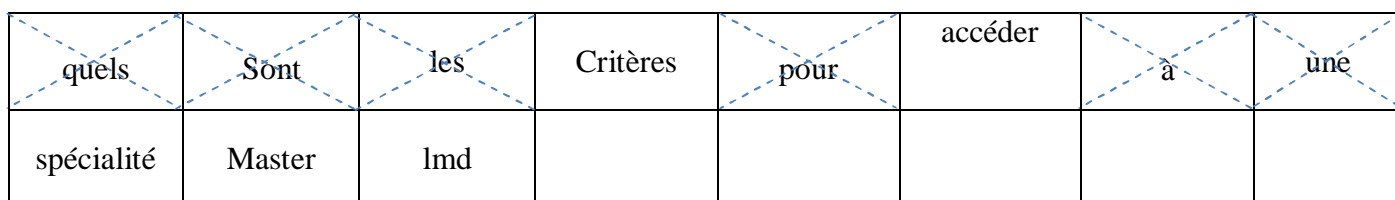
La recherche de questions similaires pour la question : "quels sont les critères pour accéder a une spécialité master "

Application de l'algorithme expliqué auparavant

- correction orthographique de la question:

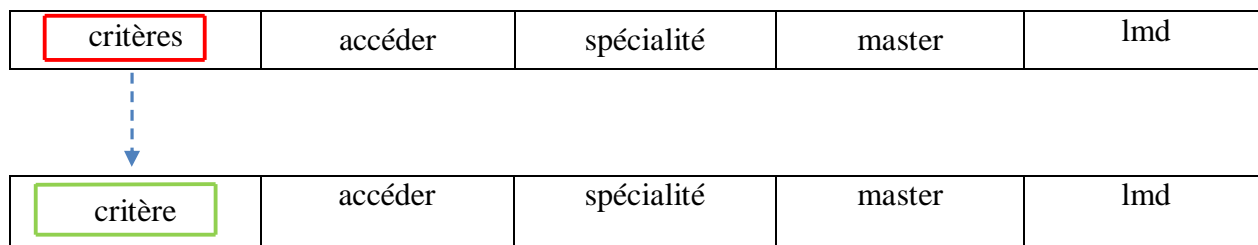


- extraction des termes porteurs de sens de la question donne comme résultat la liste de termes suivant:



"quels", "sont", "les", "pour", "à", "une" sont exclus de l'analyse car ils sont des mots vides, ils appartiennent a la stopList.

- la lemmatisation de liste des mots extraits



- l'application de la fonction de synonymie sur les lemmes donne la liste suivante:

critère	règle	exigence	condition	notion	passer	accès	déplacer
spécialité	option	filière	Master	diplôme	doctorat	lmd	

- comparaison des lemmes et leurs synonymes avec les termes porteurs de sens des questions existantes dans la base de donnée, sélectionner les questions contenant ces termes en communs.
- application de la fonction de mesure de similarité:

Q1 c'est la question posée, sa longueur est le nombre de termes porteurs de sens extraits qui est égale à 5.

les questions sélectionnées de la BDD:

Q2: " condition accès a une option master lmd"

Q3: " enseignement de master lmd"

longueur: le nombre de termes porteurs de sens de la question posée

nbreTermes_commun(question1, question2): c'est le nombre de termes en commun ou synonyme entre la question1 et question2

$$\text{seuil} = (\text{nbreTermes_commun}(Q1, Qi) / \text{longueur}Q1 + \text{nbreTermes_commun} / \text{longueur}Qi) / 2$$

Qi	longueur	nbreTermes_commun(Q1, Qi)	Seuil(Q1, Qi)
Q2	5	5	$(5/5+5/5)/2=1$
Q3	3	2	$(2/3+2/5)/2=0.53$

Figure III.22: exemples applicatifs de l'algorithme de mesure de similarité

Donc:

La question Q2 est similaire à Q1 à 100%

La question Q3 est similaire à Q1 à 53%

La figure IV.10 illustre le résultat.

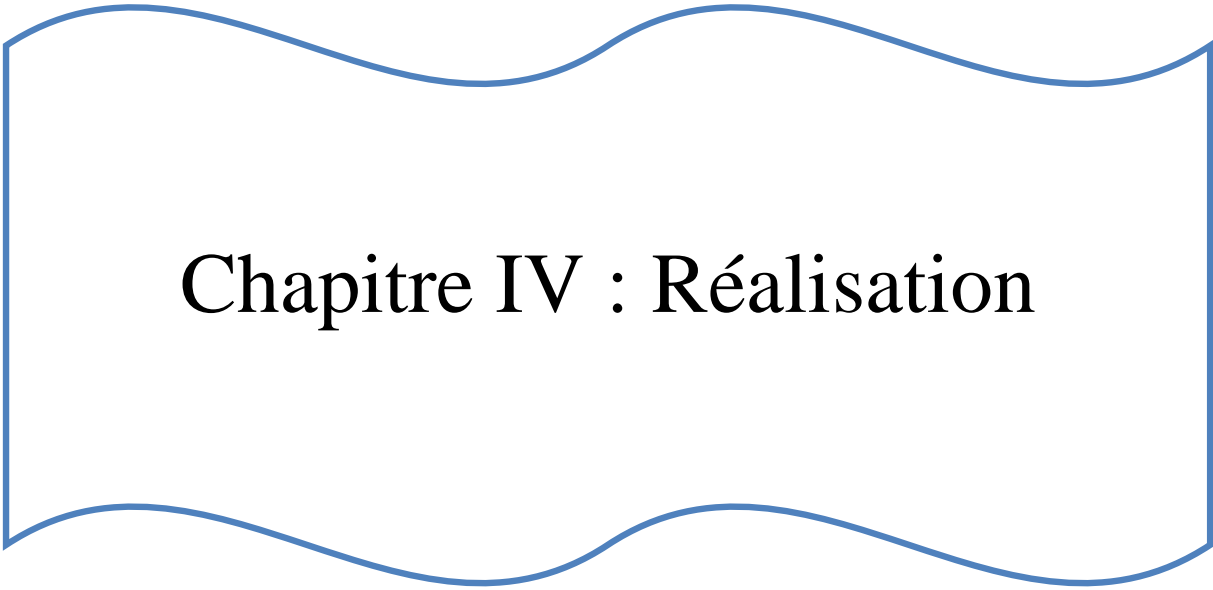
Remarque:

- lorsque la question posée contient exactement les même termes que la question trouvée dans la base de données, cette dernière est similaire a la question posée a 100%.
- Lorsque les termes en commun entre la question posée et celle de la base de donnée sont des synonymes, le seuil diminue car il n'existe pas de synonymes parfaits.

Conclusion :

Dans ce chapitre, nous nous sommes intéressés à la conception de l'application. Nous avons d'abord présenté l'approche globale de conception que nous avons suivi pour l'élaboration de notre application. Par la suite nous avons cité les fonctionnalités qu'elle offrait aux différents acteurs. Nous nous sommes également appuyé sur les diagrammes du langage UML afin de nous approfondir dans l'analyse et la conception et ce en modélisant graphiquement certains cas d'utilisation de l'application, et pour conclure ce chapitre une représentation des tables de la base de données utilisée par notre application est nécessaire.

Dans Le chapitre suivant, nous allons définir certains outils nécessaires au développement de l'application.



Chapitre IV : Réalisation

toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent.

```

body
{
padding: 0px;
margin: 0px;
background-color: #eeeeee;
width: 100%;
}
div#Entete
{
border-width: thin;
border-color: #000000;
width: 1000px;
height: 100px;
padding: 0px;
margin: 0px auto 0px auto;
background-image: url('../Image/serch-bg.jpg');
border-right-style: groove;
border-left-style: groove;
}
div#connexion
{
width: 40%;
padding: 2px;
margin: 0px;
float: right;
}
div#Main
{
border: thin groove #000000;
width: 1000px;
height: 1200px;
padding: 0px;
margin: 0px auto 0px auto;
background-color:White;
min-height:100%;
}
}

```

Figure IV.2 : Exemple de code style.css.

IV.1.2 Langages de script:

Ø jQuery :

jQuery est une extension de JavaScript, il permet aux développeurs web d'ajouter des fonctionnalités supplémentaires à leurs sites Web. Il est open source et disponible gratuitement sous la licence MIT. Au cours des dernières années, jQuery est devenue la bibliothèque JavaScript la plus populaire utilisée dans le développement web .

Pour mettre en œuvre jQuery, un développeur web doit simplement faire référence au fichier JavaScript jQuery dans le code HTML d'une page Web . Certains sites Web hébergent leur propre copie locale de jQuery, tandis que d'autres simplement référence à la Bibliothèque hébergé par Google ou le serveur jQuery. Par exemple, une page Web peut charger la bibliothèque jQuery en utilisant la ligne suivante dans la section <head> du HTML:

```

<Script type= " text \ javascript "
src = "// ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"> </ script>

```

Une fois la bibliothèque jQuery est chargée, une page Web peut appeler n'importe quel jQuery fonction soutenue par la bibliothèque. Les exemples communs incluent la modification du texte, le traitement des données de formulaire, des éléments sur une page en mouvement, et l'exécution des animations. jQuery peut également fonctionner avec Ajax code

et langages de script tels que ASP et PHP pour accéder aux données d'une base de données. Depuis jQuery fonctionne sur le côté client (plutôt que le serveur web), il peut mettre à jour les informations sur une page Web en temps réel, sans avoir à recharger la page. Un exemple courant est "autocomplete", dans lequel au fur et à mesure ou on saisie des données dans un formulaire, grâce à jQuery, il est possible de suggérer la suite des mots tapés présent dans une base de données.

Outre sa licence libre, l'autre raison principale jQuery a acquis une telle popularité est son cross-browser compatible. Étant donné que chaque navigateur affiche HTML, CSS et JavaScript différemment, il peut être difficile pour un développeur web pour faire apparaître un site web la même dans tous les navigateurs. Au lieu d'avoir à écrire des fonctions personnalisées pour chaque navigateur, un développeur Web peut utiliser une fonction de jQuery unique qui fonctionnera dans Chrome, Safari, Firefox et Internet Explorer. Ce support multi-navigateur a conduit de nombreux développeurs à passer du niveau javascript jQuery, car il simplifie considérablement le processus de codage.

```

<link href="jquery-ui.css" rel="stylesheet" type="text/css"/>
<script type="text/javascript" src="jquery.min.js"></script>
<script type="text/javascript" src="jquery-ui.min.js"></script>
<script type="text/javascript">
$(document).ready(function () {
    SearchText();
});
function SearchText() {
    $(".autosuggest").autocomplete({
        source: function (request, response) {
            $.ajax({
                type: "POST",
                contentType: "application/json; charset=utf-8",
                url: "Recherche.aspx/GetAutoCompleteData",
                data: '{"username":"' + document.getElementById('txtSearch').value + '"}',
                dataType: "json",
                success: function (data) {
                    response(data.d);
                },
                error: function (result) {
                    alert("Error");
                }
            });
        }
    });
}
});

```

Figure IV.3 : Exemple de script jQuery.

IV.1.3 Serveurs :

Ø Serveur Web :

Pour exécuter des applications Web, il faut disposer d'un serveur. Un serveur Web est un logiciel permettant à des clients d'accéder à des pages web, à partir d'un navigateur (aussi appelé browser). Un serveur Web est donc un simple logiciel capable d'interpréter les requêtes http arrivants sur le port associé au protocole http et de fournir une réponse avec ce même protocole.

Exemple de serveur web :

- Microsoft IIS (Internet Information Serveur)
- Apache
- Microsoft PWS (Personal Web Server)

- Notre application est gérée par le serveur Microsoft Apache qui est le serveur Web de Microsoft, il contrôle les pages Web (celles évidemment qui sont hébergées sur sa machine) et qui envoie le HTML de ces pages Web aux navigateurs des clients qui en font la demande.

Ø Serveur de base de données :

Présentation de Microsoft SQL Server :

Microsoft SQL Server est un système de gestion de base de données (abrégé en SGBD ou SGBDR pour « Système de gestion de base de données relationnelles ») développé et commercialisé par la société Microsoft.

Bien qu'il ait été initialement Codéveloppé par Sybase et Microsoft, Ashton-Tate a également été associé à sa première version, sortie en 1989. Cette version est sortie sur les plates-formes Unix et OS/2. Depuis, Microsoft a porté ce système de base de données sous Windows et il est désormais uniquement pris en charge par ce système.

En 1994, le partenariat entre les deux sociétés ayant été rompu, Microsoft a sorti la version 6.0 puis 6.5 seul, sur la plate-forme Windows NT.

- Microsoft a continué de commercialiser le moteur de base de données sous le nom de SQL Server
- Tandis que Sybase, pour éviter toute confusion, a renommé Sybase SQL Server en Sybase Adaptive Server Enterprise.

Microsoft SQL Server fait désormais partie de la stratégie technique de Microsoft en matière de base de données. Le moteur MSDE, qui est la base de SQL Server, doit à terme remplacer le moteur Jet (celui qui gère les bases Access) dans les applications telles qu'Exchange et Active Directory.

La version 2005 de SQL Server est sortie le 3 novembre 2005 en même temps que Visual Studio 2005. La prise en charge de Windows Vista de Windows Server 2008 n'a été ajoutée qu'à partir du Service Pack 2 (SP2).

La version 2008 de SQL Server (nom de code Katmaï) est disponible depuis août 2008. Elle est actuellement au niveau de service pack 3. Elle est disponible en 9 langues, dont le français.

IV.2 Environnement de développement

IV.2.1 langage de programmation java:

Java est un langage de programmation récent (les premières versions datent de 1995) développé par Sun Microsystems. Il est fortement inspiré des langages C et C++. Comme C++, Java fait partie de la « grande famille » des langages orientés objets. Il répond donc aux trois principes fondamentaux de l'approche orientée objet (POO) : l'encapsulation, le polymorphisme et l'héritage.

Java a rapidement intéressé les développeurs pour quatre raisons principales :

- Û C'est un langage orienté objet dérivé du C, mais plus simple à utiliser et plus « pur » que le C++. On entend par « pur » le fait qu'en Java, on ne peut faire que de la programmation orienté objet contrairement au C++ qui reste un langage hybride, c'est-à-dire autorisant plusieurs styles de programmation. C++ est hybride pour assurer une compatibilité avec le C ;
- Û Il est doté, en standard, de bibliothèques de classes très riches comprenant la gestion des interfaces graphiques (fenêtres, boîtes de dialogue, contrôles, menus, graphisme), la programmation multi-threads (multitâches), la gestion des exceptions, les accès aux fichiers et au réseau ... L'utilisation de ces bibliothèques facilitent grandement la tâche du programmeur lors de la construction d'applications complexes ;
- Û Il est doté, en standard, d'un mécanisme de gestion des erreurs (les exceptions) très utile et très performant. Ce mécanisme, inexistant en C, existe en C++ sous la forme d'une extension au langage beaucoup moins simple à utiliser qu'en Java ;
- Û Il est multi plates-formes : les programmes tournent sans modification sur tous les environnements où Java existe (Windows, Unix et Mac) ;

Ce dernier point a contribué à utiliser d'abord Java pour développer des applets qui sont des applications pouvant être téléchargées via l'Internet et exécutées dans un navigateur sur n'importe quelle plate-forme. Ainsi, une page statique HTML peut s'enrichir de programmes qui lui donneront un comportement dynamique.

Il possède deux particularités que je tiens à souligner :

- **Multi plates-formes**

Java est un langage qui doit être compilé et interprété. Dans une première phase on compile un programme (un ou plusieurs fichiers source .java) en fichiers .class. Le compilateur génère un fichier .class pour chacune des classes définies dans le(s) fichier(s) .java. L'ensemble des fichiers .class est ensuite interprété par la Machine Virtuelle Java (Java Virtual Machine) pour exécuter le programme (seule la JVM dépend du système). Ce principe donne à Java une portabilité maximale (Windows, Unix, MacIntosh ...).

- **Applets et applications**

Il est possible de développer des applications isolées (standalone applications), fonctionnant avec l'interpréteur comme un programme habituel dans un langage classique mais aussi des "applets". Les applets sont des programmes qui peuvent être téléchargés sur l'Internet puis exécutés automatiquement quand ils sont intégrés à dans des pages HTML. Dans ce dernier cas, l'ensemble des fichiers .class est utilisé avec un fichier HTML qui fait appel à une des classes.

IV.2.2 Architecture MVC:

L'organisation d'une interface graphique est délicate. L'architecture MVC ne prétend pas éliminer tous les problèmes, mais fournit une première approche pour ce faire. Offrant un cadre normalisé pour structurer une application, elle facilite aussi le dialogue entre les concepteurs.

L'idée est de bien séparer les données, la présentation et les traitements. Il en résulte les trois parties énumérées plus haut : le modèle, la vue et le contrôleur.

Ø Modèle

Le modèle représente le cœur (algorithmique) de l'application : traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. La base de données sera l'un de ses composants. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle ne s'occupent pas de la présentation. Le modèle ne contient aucun lien direct vers le contrôleur ou la vue. Sa communication avec la vue s'effectue au travers du patron Observateur.

Le modèle peut autoriser plusieurs vues partielles des données. Si par exemple le programme manipule une base de données pour les emplois du temps, le modèle peut avoir des méthodes pour avoir tous les cours d'une salle, tous les cours d'une personne ou tous les cours d'un groupe de TD.

Ø Vue

Ce avec quoi l'utilisateur interagit se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur (hover, clic de souris, sélection d'un bouton radio, cochage d'une case, entrée de texte, de mouvements, de voix, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

Plusieurs vues peuvent afficher des informations partielles ou non d'un même modèle. Par exemple si une application de conversion de base a un entier comme unique donnée, ce même entier peut être affiché de multiples façons (en texte dans différentes bases, bit par bit avec des boutons à cocher, avec des curseurs). La vue peut aussi offrir à l'utilisateur la possibilité de changer de vue.

Ø Contrôleur

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de la vue et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle afin que les données affichées se mettent à jours. D'après le patron de conception observateur/observable, la vue est un « observateur » du modèle qui est lui « observable ». Certains événements de l'utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée. Il analyse la

requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

Par exemple, dans le cas d'une base de données gérant les emplois du temps des professeurs d'une école, une action de l'utilisateur peut être l'entrée (saisie) d'un nouveau cours. Le contrôleur ajoute ce cours au modèle et demande sa prise en compte par la vue. Une action de l'utilisateur peut aussi être de sélectionner une nouvelle personne pour visualiser tous ses cours. Ceci ne modifie pas la base des cours mais nécessite simplement que la vue s'adapte et offre à l'utilisateur une vision des cours de cette personne.

Quand un même objet contrôleur reçoit les événements de tous les composants, il lui faut déterminer quelle est l'origine de chaque événement. Ce tri des événements peut s'avérer fastidieux et peut conduire à un code peu élégant (un énorme switch). C'est pourquoi le contrôleur est souvent scindé en plusieurs parties dont chacune reçoit les événements d'une partie des composants.

Ø Flux de traitement

En résumé, lorsqu'un client envoie une requête à l'application :

- la requête envoyée depuis la vue est analysée par le contrôleur (par exemple un clic de souris pour lancer un traitement de données) ;
- le contrôleur demande au modèle approprié d'effectuer les traitements et notifie à la vue que la requête est traitée ("via" par exemple un handler ou callback) ;
- la vue notifiée fait une requête au modèle pour se mettre à jour (par exemple affiche le résultat du traitement "via" le modèle).

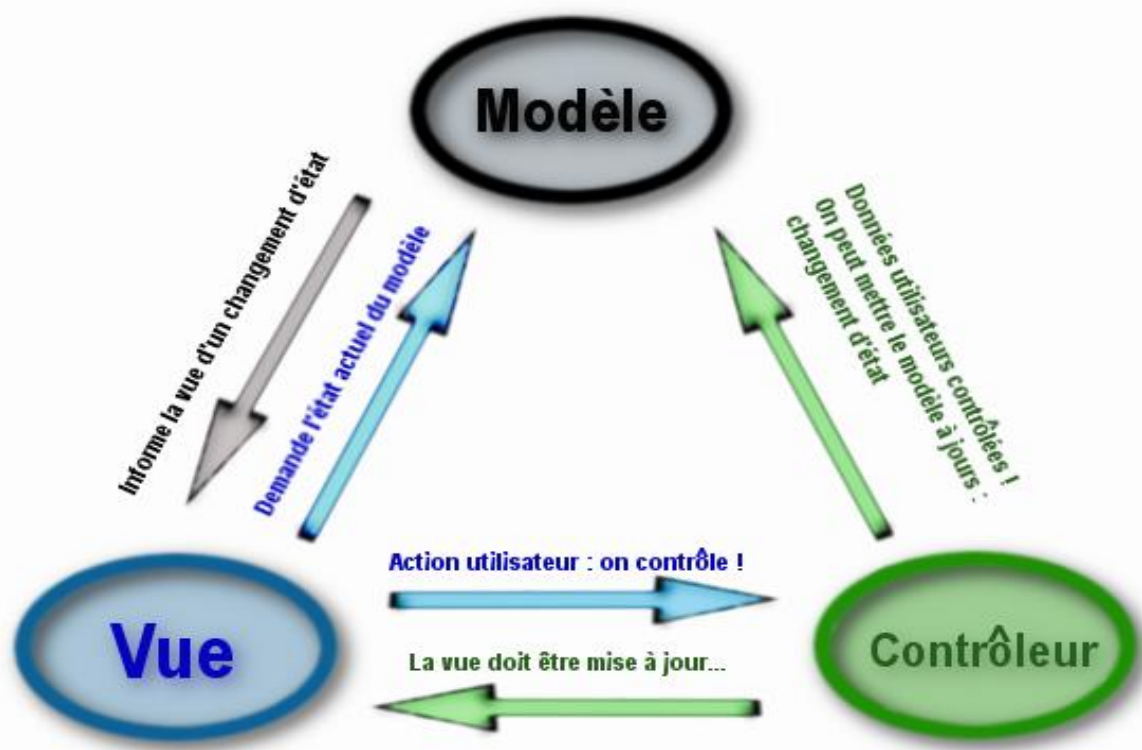


Figure IV.4 : interface " pattern MVC "

Ø Le Framework spring MVC:

Le cœur de l'environnement Spring est un «conteneur léger». Un conteneur léger sert à contenir un ensemble d'objets instanciés et initialisés, formant un contexte initial (ou une hiérarchie de contextes) pour une application. Ce contexte initial est souvent construit à partir d'une description externe (xml) décrivant les objets à créer, les valeurs initiales et les dépendances entre objets. Les dépendances (liens) entre objets sont automatiquement créées à partir de la description (on parle d'injection de dépendances) et non par les objets eux-mêmes par programmation.

SpringMVC est un framework de presentation, pour applications web, suivant le modèle MVC, et fondu sur le conteneur léger de SPRING

IV.2.2 Avantages de l'architecture MVC :

Un avantage apporté par ce modèle est la clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. En effet, la modification des traitements ne change en rien la vue. Par exemple on peut passer d'une base de données de type SQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne s'en trouvent pas affectées.

Le MVC montre ses limites dans le cadre des applications utilisant les technologies du web, bâties à partir de serveurs d'applications. Des couches supplémentaires sont alors introduites ainsi que les mécanismes d'inversion de contrôle et d'injection de dépendance..

IV.2.3 Différence avec l'architecture trois tiers :

L'architecture trois tiers est un modèle en couches, c'est-à-dire, que chaque couche communique seulement avec ses couches adjacentes (supérieures et inférieures) et le flux de contrôle traverse le système de haut en bas; les couches supérieures contrôlent les couches inférieures, c'est-à-dire, que les couches supérieures sont toujours sources d'interaction (clients) alors que les couches inférieures ne font que répondre à des requêtes (serveurs).

Dans le modèle MVC, il est généralement admis que la vue puisse consulter directement le modèle (lecture) sans passer par le contrôleur. Par contre, elle doit nécessairement passer par le contrôleur pour effectuer une modification (écriture). Ici, le flux de contrôle est inversé par rapport au modèle en couches, le contrôleur peut alors envoyer des requêtes à toutes les vues de manière à ce qu'elles se mettent à jour.

Dans l'architecture trois tiers, si une vue modifie les données, toutes les vues concernées par la modification doivent être mises à jour, d'où l'utilité de l'utilisation du MVC au niveau de la couche de présentation. La couche de présentation permet donc d'établir des règles du type « mettre à jour les vues concernant X si Y ou Z sont modifiés ». Mais ces règles deviennent rapidement trop nombreuses et ingérables si les relations logiques sont trop élevées. Dans ce cas, un simple rafraîchissement des vues à intervalle régulier permet de surmonter aisément ce

problème. Il s'agit d'ailleurs de la solution la plus répandue en architecture trois tiers, l'utilisation du MVC étant moderne et encore marginale.

IV.3 Les outils de développement:

IV.3.1 IDE NetBeans:

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme.

L'IDE Netbeans s'enrichit à l'aide de plugins.

Il comprend toutes les caractéristiques d'un IDE moderne (coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web, etc).

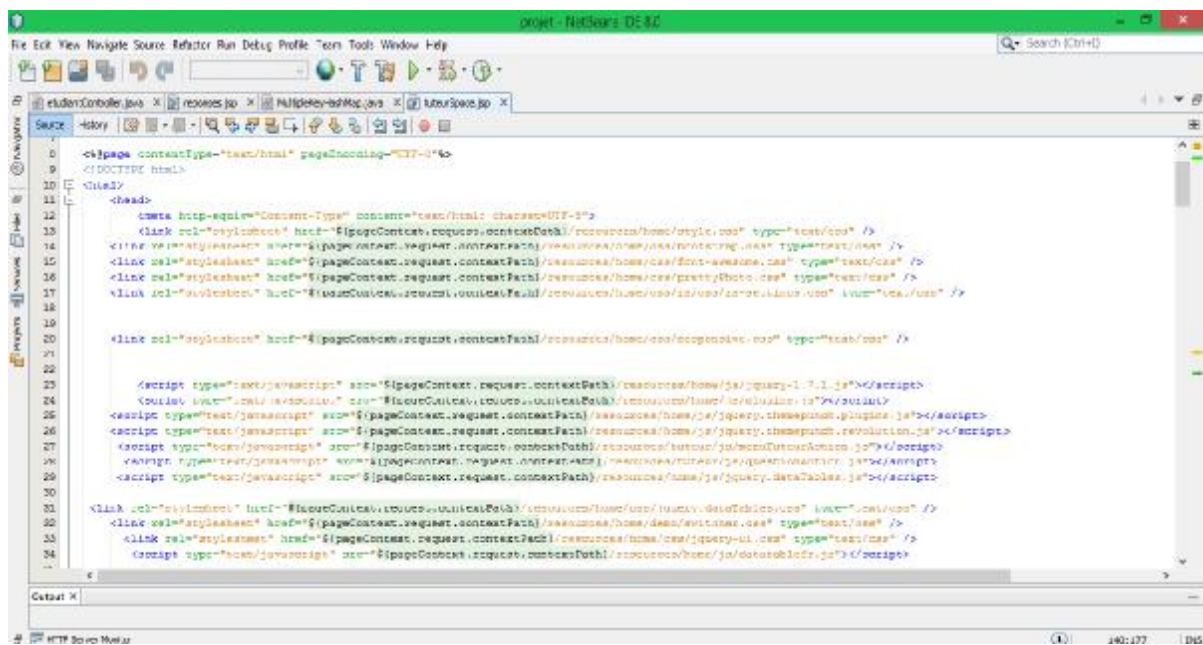


Figure IV.5 : interface " NetBeans 8.0 "

IV.3.2 Le SGBD (MYSQL 5) :

MySQL est un véritable serveur (ou gestionnaire) de base de données SQL multi-utilisateur et multi-threaded. SQL est le plus populaire langage de base de données dans le monde. *MySQL* est une configuration client/serveur ce qui consiste en un serveur démon mysqld, différents programmes clients et des bibliothèques. SQL est un langage standardisé qui rend facile le stockage, la mise à jour et l'accès à l'information. Par exemple, vous pouvez utiliser le SQL pour récupérer des informations sur un produit ou stocker des informations client sur un site web. *MySQL* est suffisamment rapide et flexible pour gérer des historiques et des images.

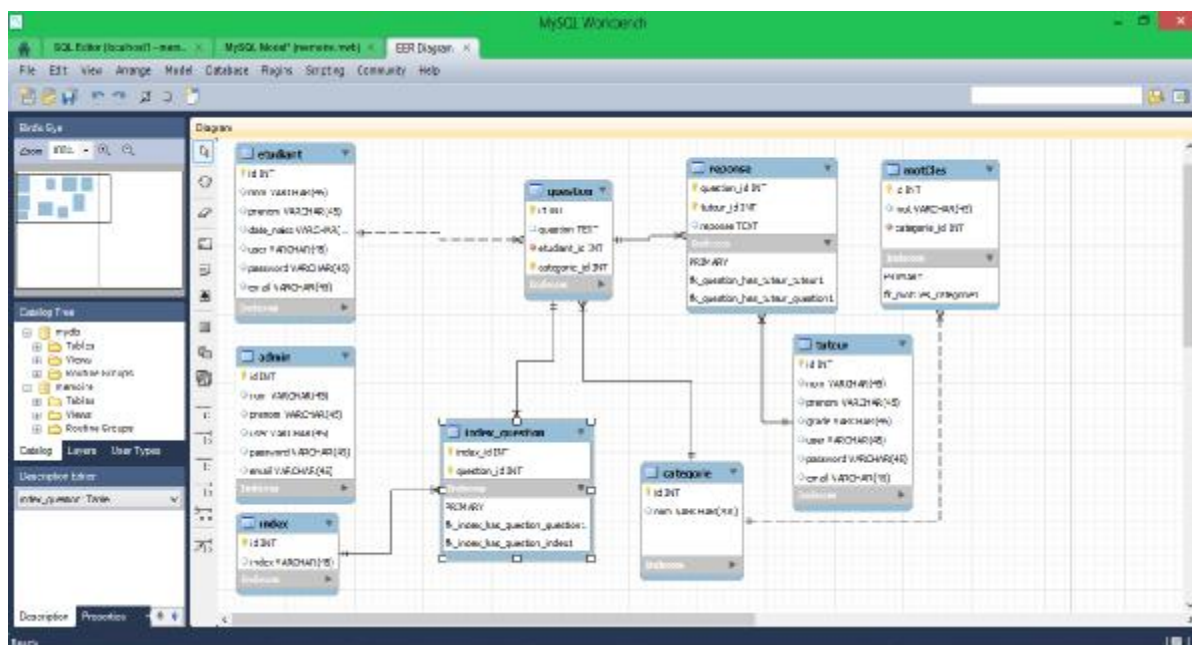


Figure IV.6 : interface "MySQL workbench 5.2 "

IV.3 Présentation de quelques interfaces :

IV.3.1 Page d'accueil :

c'est la première page qui apparaîtra à tout utilisateurs de l'application.

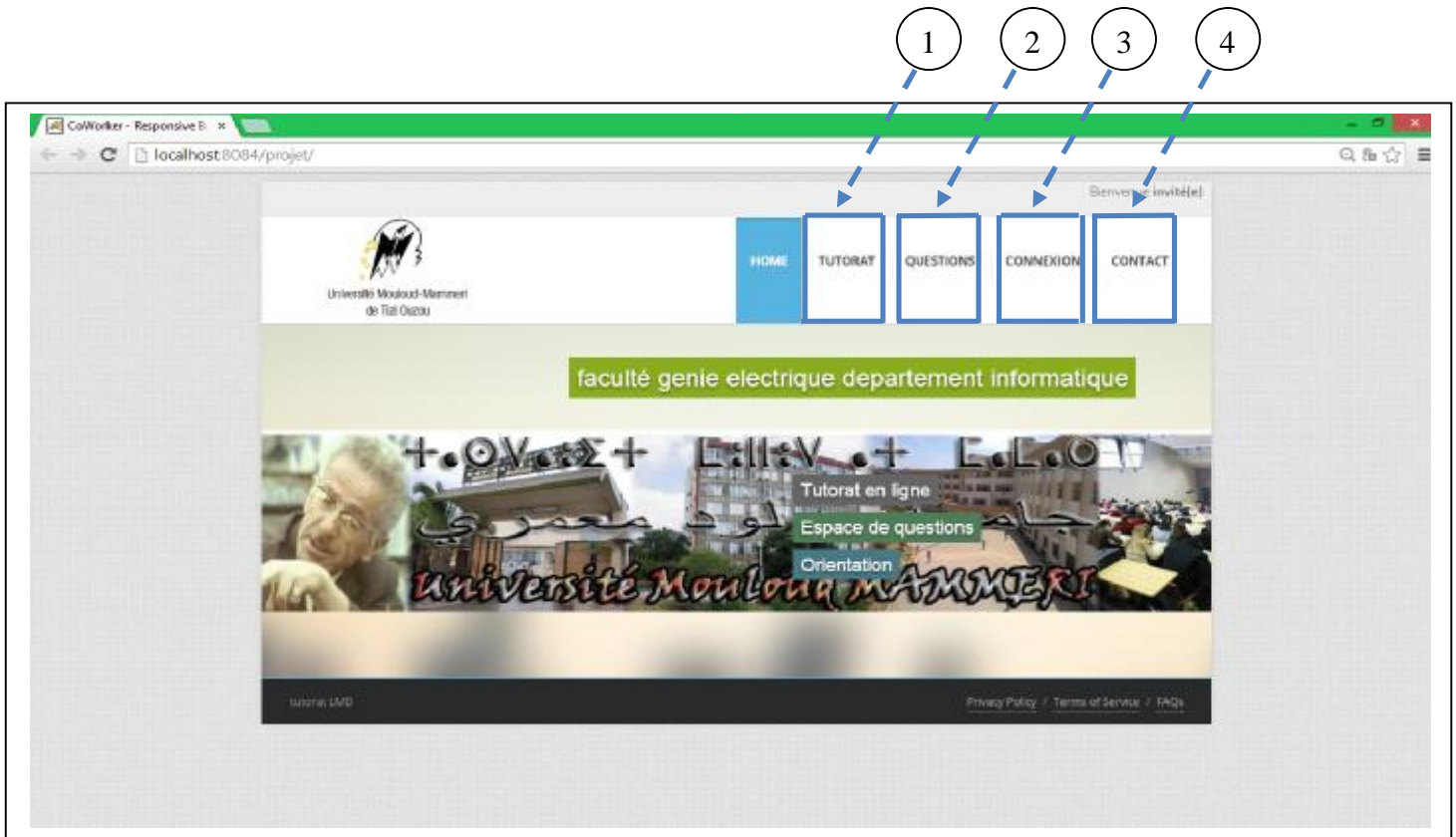


Figure IV.7 : interface d'accueil

- ① : donne accès a une page d'informations et guide d'utilisation du tutorat
- ② : permet a l'utilisateur de voir toute les questions et réponses ainsi de consulter les questions par catégorie, sous catégorie.
- ③ : donne accès au formulaire d'authentification.
- ④ : pour contacter le gestionnaire du tutorat.

IV.3.2 Espace étudiant :

Dans son espace l'étudiant peut poser des questions, suivre ses questions.

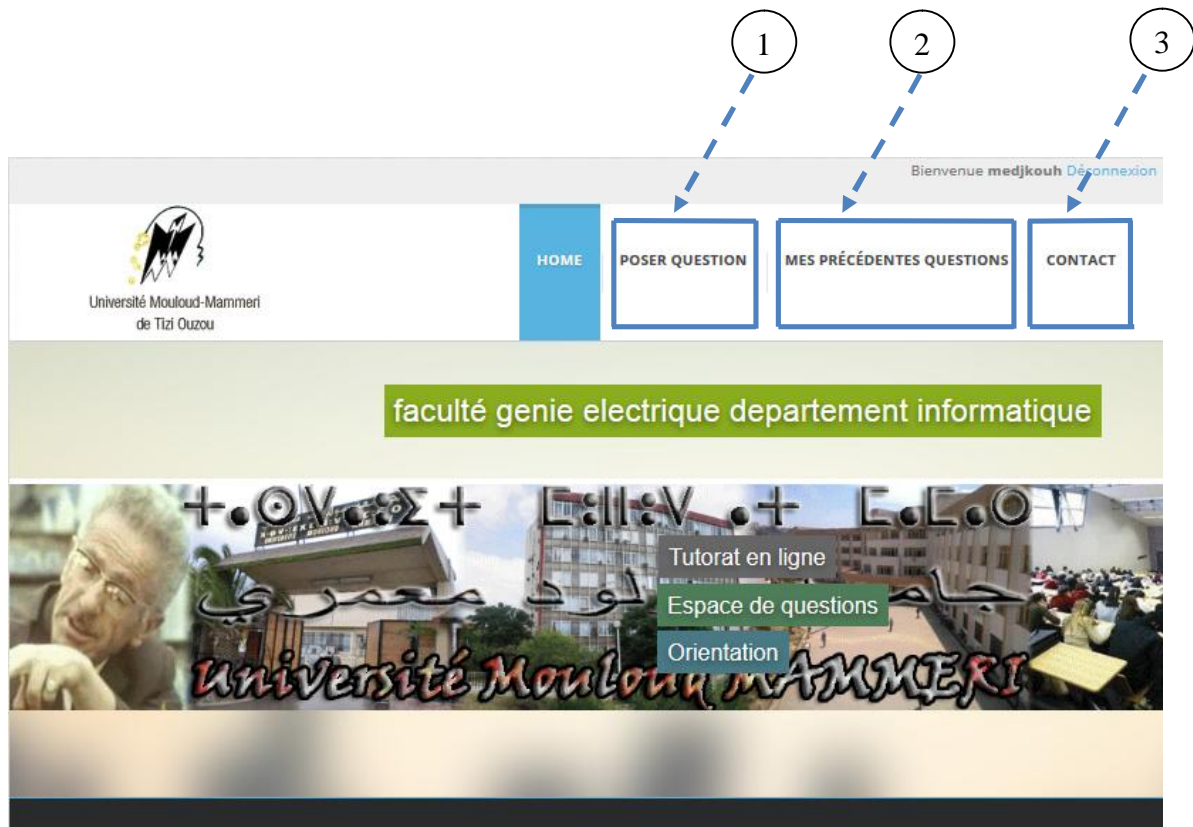



Figure IV.8 : espace Etudiant

- ① : gérer les questions (poser une question, consulter les questions similaires, différer la question).
- ② : suivre mes questions(consulter mes questions, consulter les réponses du tuteur).
- ③ : contacter le gestionnaire du tutorat.

Bienvenue **keballi** Déconnexion



Université Mouloud-Mammeri
de Tizi Ouzou

[HOME](#)
[POSER QUESTION](#)

MES PRÉCÉDENTES QUESTIONS

[CONTACT](#)

Toutes mes questions

Afficher entrees

Toutes Mes Questions

Questions Avec Réponse

Questions Sans Réponse [Rechercher](#)

Mes questions

question condition accès au master lmd

reponse le master est accessible après validation des six semestres de licence, la licence est validé avec 180 ECTS

le tuteur qui a répondu:
tuteur

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question master lmd

reponse le Master est le 2ème cycle de la formation LMD de durée de 2 ans (M1 M2), 4 semestres, le master est validé avec 120 crédits

le tuteur qui a répondu:
tuteur

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question options du master lmd

reponse aucune reponse

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question condition accès a une option master lmd

reponse aaaaaaaaaa

le tuteur qui a répondu:
tuteur

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question etudier en licence

reponse aaaaaaaaaa

le tuteur qui a répondu:
tuteur

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

Figure IV.9 : interface "mes questions précédentes"

quels sont les critre de passage a une spcialté master lmd RECHERCHER

si vous n'etes pas satisfait des questions proposées veuillez poser la question au tuteur en cliquant sur differer la question DIFFERER AU TUTEUR

Afficher 25 entrees Rechercher

questions similaires



<p>pourcentage de similarité</p>  <p>95 %</p>	<p>question condition accès a une option master lmd</p> <p>reponse aaaaaaaaaa</p>	<p>l'etudiant qui a posé la question: kebailli</p>	<p>le tuteur qui a répondu: tuteur</p>	<p>la categorie de la question: Aspect informatif et administratif</p>	<p>la sous categorie de la question: Orientation</p>
<p>pourcentage de similarité</p>  <p>53 %</p>	<p>question enseignement en master lmd</p> <p>reponse aucune reponse</p>	<p>l'etudiant qui a posé la question: kebailli</p>		<p>la categorie de la question: Aspect informatif et administratif</p>	<p>la sous categorie de la question: Orientation</p>

Figure IV.10 : interface "poser question.exemple 1"

apprendre a programmer en langage java RECHERCHER

si vous n'etes pas satisfait des questions proposées veuillez poser la question au tuteur en cliquant sur DIFFERER AU TUTEUR

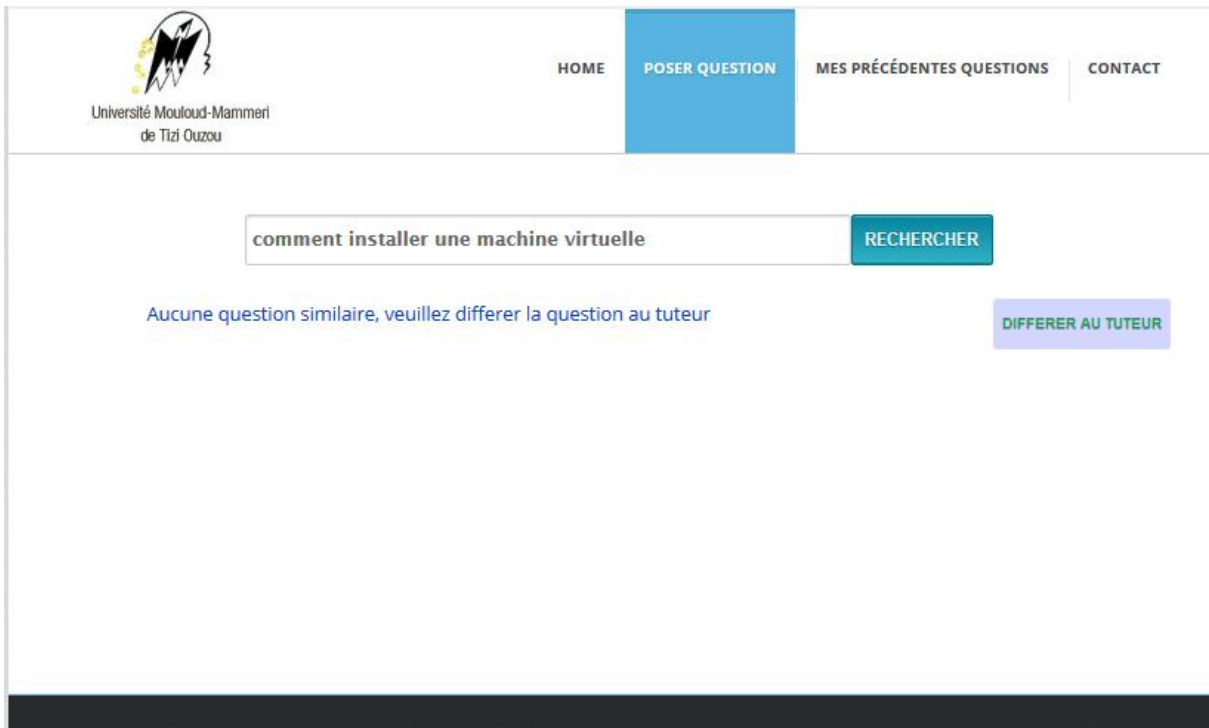
Afficher 25 entrées Rechercher

questions similaires

pourcentage de similarité	question	apprendre a programmer en langage java		
100 %	reponse	aucune reponse		
	l'etudiant qui a posé la question:	la categorie de la question:	la sous categorie de la question:	
	medjkouh	Suivi de la formation	programmation	
pourcentage de similarité	question	outils de programmation en langage java		
75 %	reponse	aucune reponse		
	l'etudiant qui a posé la question:	la categorie de la question:	la sous categorie de la question:	
	medjkouh	Suivi de la formation	programmation	

Afficher 1 a 2 de 2 entrees 🔍

Figure IV.11 : interface "poser question.exemple 2"



Université Mouloud-Mammeri
de Tizi Ouzou

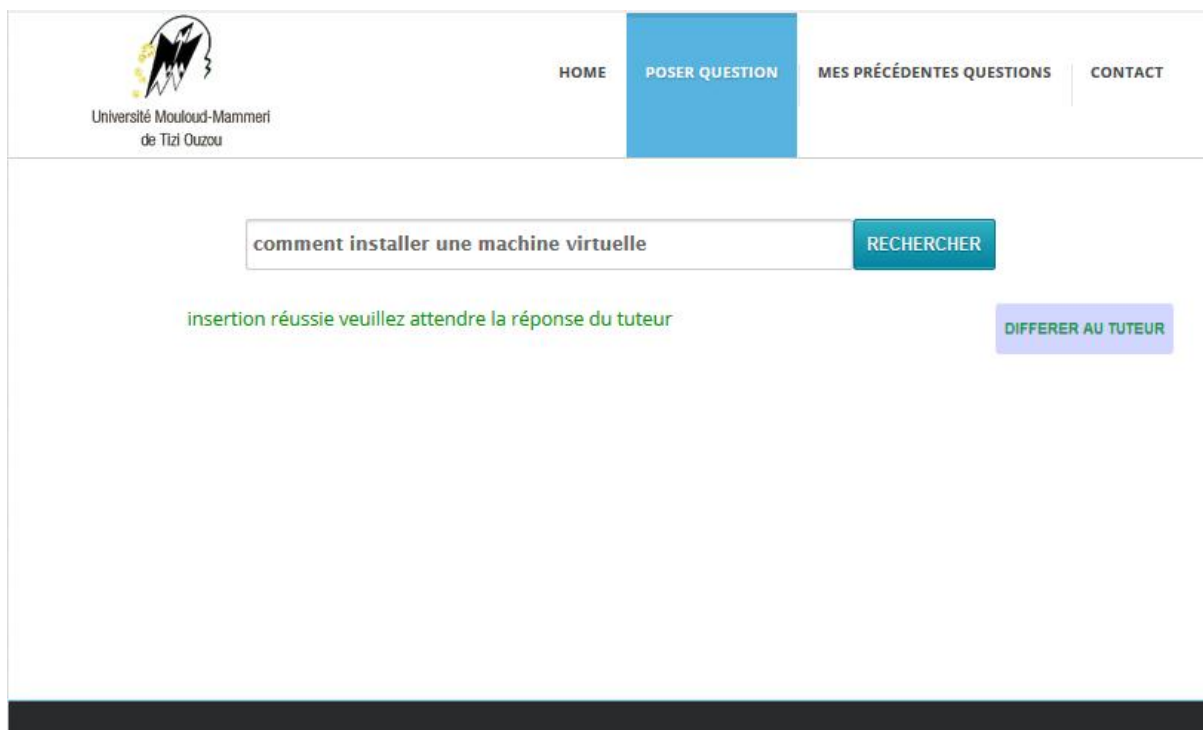
HOME POSER QUESTION MES PRÉCÉDENTES QUESTIONS CONTACT

comment installer une machine virtuelle RECHERCHER

Aucune question similaire, veuillez différer la question au tuteur DIFFERER AU TUTEUR

Figure IV.12 : interface "poser question.exemple 3"

Lors du clique sur "différer au tuteur", l'interface suivante s'affiche



Université Mouloud-Mammeri
de Tizi Ouzou

HOME POSER QUESTION MES PRÉCÉDENTES QUESTIONS CONTACT

comment installer une machine virtuelle RECHERCHER

insertion réussie veuillez attendre la réponse du tuteur DIFFERER AU TUTEUR

Figure IV.13 : interface "poser question.exemple 4"

Recherche de question similaire

The screenshot shows a search interface with the following elements:

- Search bar: "configuration de machine virtuelle ubuntu sous windows 7" with a "RECHERCHER" button.
- Text: "si vous n'etes pas satisfait des questions proposées veuillez poser la question au tuteur en cliquant sur differer la question" and a "DIFFERER AU TUTEUR" button.
- Display settings: "Afficher 25 entrees" and a "Rechercher" input field.
- Section header: "questions similaires".
- Result card:
 - pourcentage de similarité: 50 % (indicated by a circular progress bar).
 - question: comment installer une machine virtuelle
 - reponse: aucune reponse
 - l'etudiant qui a posé la question: medjkouh
 - la categorie de la question: Suivi de la formation
 - la sous categorie de la question: Systeme d'exploitation
- Footer: "Afficher 1 a 1 de 1 entrees" and a refresh icon.

Figure IV.14 : interface "poser question.exemple 5"

L'interface suivante apparait lors du clique sur "différer au tuteur"

The screenshot shows the interface after clicking "différer au tuteur":

- Navigation menu: "HOME", "POSER QUESTION" (highlighted), "MES PRÉCÉDENTES QUESTIONS", "CONTACT".
- Logo: Université Mouloud-Mammeri de Tizi Ouzou.
- Search bar: "configuration de machine virtuelle ubuntu sous windows 7" with a "RECHERCHER" button.
- Text: "insertion réussie veuillez attendre la réponse du tuteur" (in green).
- Button: "DIFFERER AU TUTEUR" (in light blue).

Figure IV.15 : interface "poser question.exemple 5"

lors du clique sur "différer au tuteur" l'interface suivante s'affiche

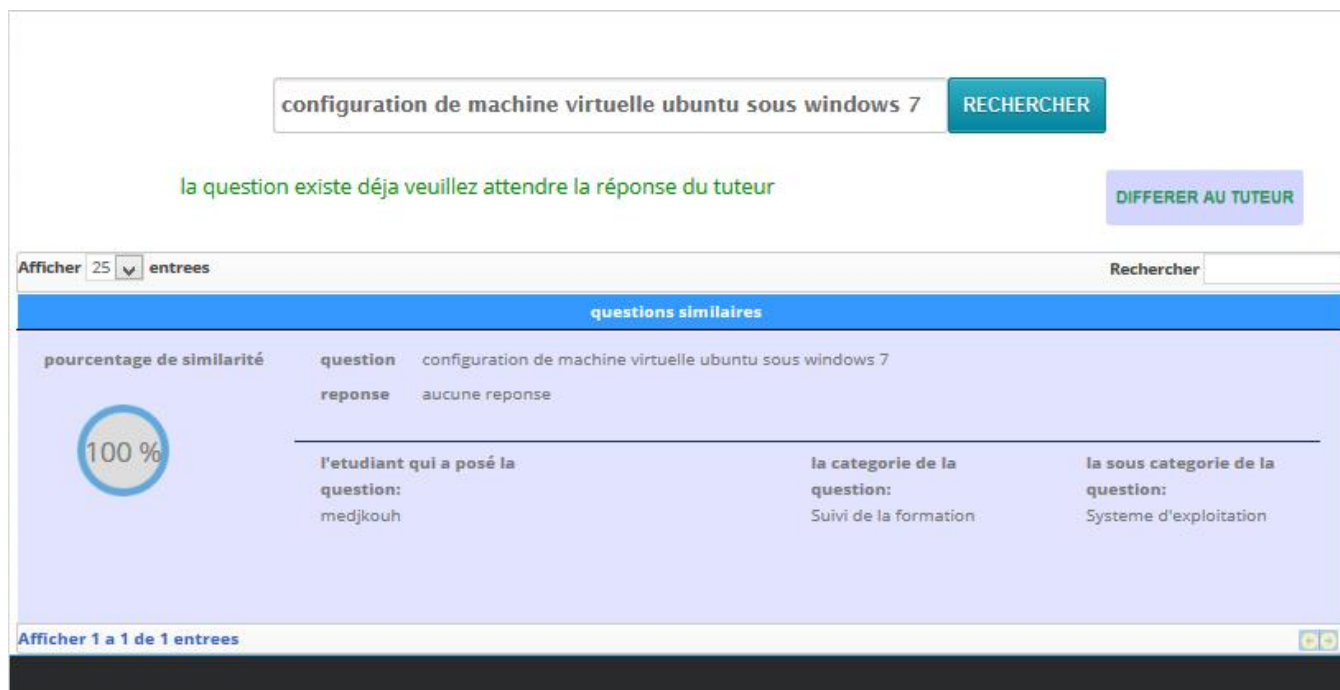


Figure IV.16 : interface "poser question.exemple 5"

Lors du clique sur "rechercher" l'interface suivante s'affiche

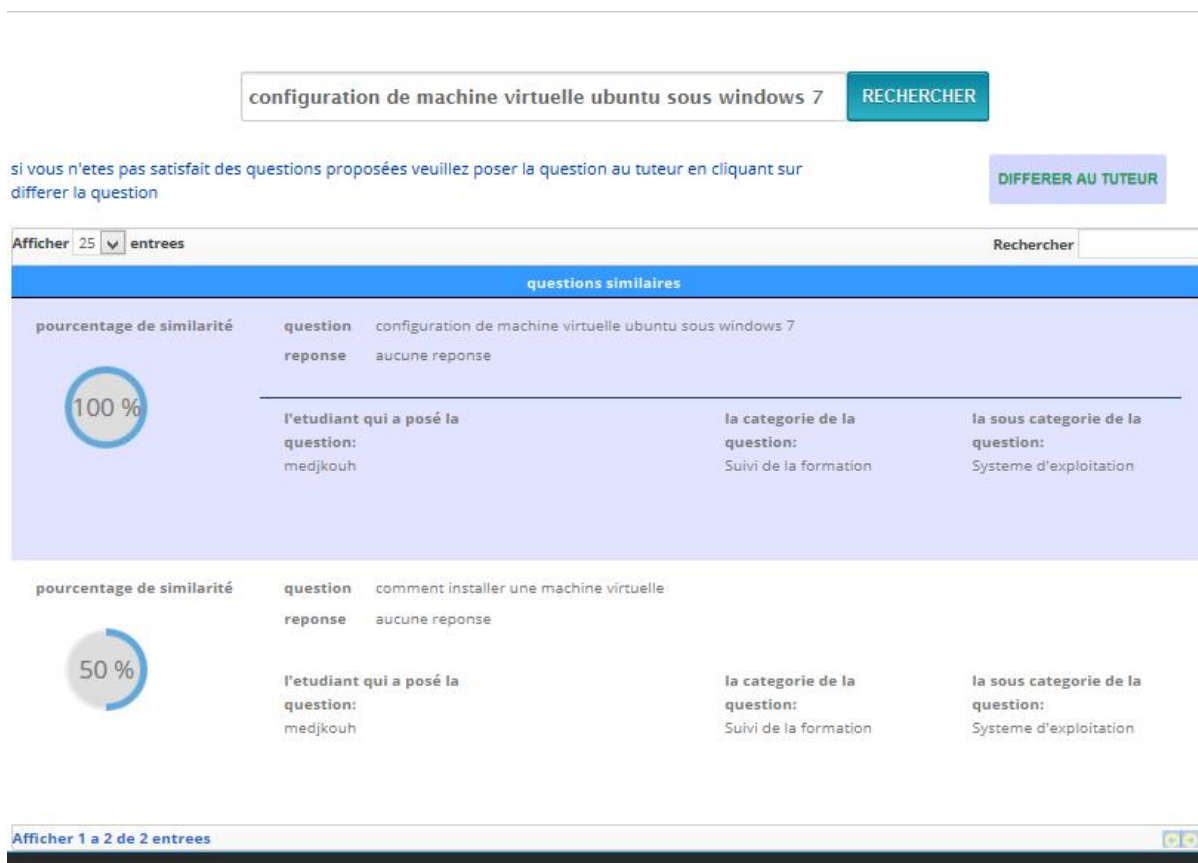


Figure IV.17 : interface "poser question.exemple 5"

IV.3.3 Espace tuteur :

Dans son espace le tuteur peut:

- Ü gérer les questions des étudiants (répondre au questions)
- Ü gérer les catégories de questions (l'ajout de nouvelles catégories et sous catégories)

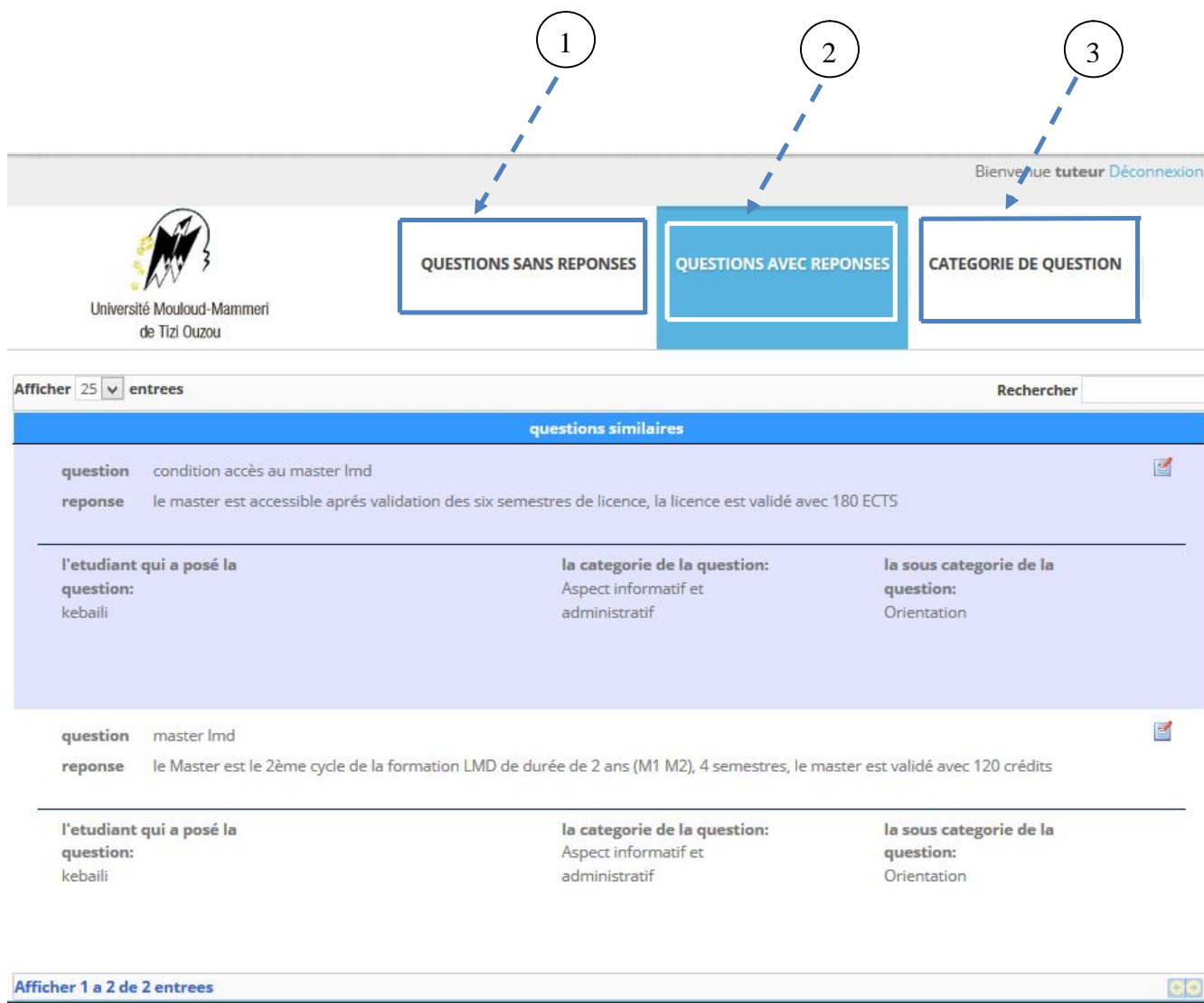


Figure IV.18 : interface "espace tuteur"

- 1 : consulter et répondre aux questions sans réponse
- 2 : consulter et modifier les questions avec réponse
- 3 : gérer les catégories, les sous catégories et les mots clés associés aux questions



HOME

QUESTIONS SANS REPONSES

QUESTIONS AVEC RÉPONSE

CATEGORIE QUESTIONS

Afficher 25 entrees

Rechercher

questions similaires

question condition accès au master lmd
reponse le master est accessible après validation des six semestres de licence, la licence est validé avec 180 ECTS

l'etudiant qui a posé la question:
kebaili

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question master lmd
reponse le Master est le 2ème cycle de la formation LMD de durée de 2 ans (M1 M2), 4 semestres, le master est validé avec 120 crédits

l'etudiant qui a posé la question:
kebaili

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question quels sont les différent types ue
reponse toute offre de formation comporte en général quatre grandes catégories d'UE : Les UE fondamentales correspondant aux enseignements que tous les étudiants doivent suivre ; Les UE complémentaires réservées à certains étudiants (approfondissement ou professionnalisation) ; Les UE transversales (enseignements destinés à donner des outils à des étudiants de diverses provenances : langue, informatique, mathématiques, droits de l'homme, animation culturelle et sportive, etc.). Les UE libres que l'étudiant choisit librement selon ses goûts et ses besoins.

l'etudiant qui a posé la question:
medjkouh

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question quel est le lien entre crédit et note
reponse Les crédits et les notes ne doivent pas être confondus. Les crédits expriment le volume de travail qui incombe à l'étudiant, tandis que les notes traduisent la qualité des résultats obtenus par l'étudiant. L'étudiant obtient les crédits d'une UE uniquement lorsqu'il satisfait aux modalités d'évaluation de cette UE, c'est-à-dire s'il obtient au moins 10/20 dans cette UE.

l'etudiant qui a posé la question:
medjkouh

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question quelle est la différence entre master professionnel et recherche
reponse Un Master professionnel est destiné vers une entrée dans le monde professionnel, le master recherche vers la poursuite en étude doctorale.

l'etudiant qui a posé la question:

la categorie de la question:
Aspect informatif et

la sous categorie de la question:

Figure IV.19 : interface "Question avec reponses"



Université Mouloud-Mammeri
de Tizi Ouzou

HOME

QUESTIONS SANS REPONSES

QUESTIONS AVEC RÉPONSE

CATEGORIE QUESTIONS

Afficher 25 entrees

Rechercher

questions similaires

question options du master lmd
reponse aucune reponse

repondre

l'etudiant qui a posé la question:
kebailli

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question que veut dire ue
reponse aucune reponse

repondre

l'etudiant qui a posé la question:
medjkouh

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question comment obtenir un semestre
reponse aucune reponse

repondre

l'etudiant qui a posé la question:
medjkouh

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question enseignement en master lmd
reponse aucune reponse

repondre

l'etudiant qui a posé la question:
kebailli

la categorie de la question:
Aspect informatif et administratif

la sous categorie de la question:
Orientation

question Insertion de ligne automatique dans un tableau Excel
reponse aucune reponse

repondre

l'etudiant qui a posé la question:
kebailli

la categorie de la question:
Suivi de la formation

la sous categorie de la question:
Bureautique

Figure IV.20: interface "questions sans reponse"

Conclusion :

Au niveau de ce chapitre nous avons présenté l'environnement de développement et l'implémentation de notre application. Nous avons aussi décrit quelques interfaces d'utilisation.

Conclusion générale

Ce mémoire traite l'intégration d'un moyen ou d'une technique text minig (de similarité) dans un environnement e-learning. L'ensemble du document s'articule autour du développement d'un module automatisé de tutorat dans le cadre du système LMD.

L'accent est en particulier mis sur la mise en n œuvre des différentes interfaces de cette application et leurs fonctionnalités essentielles.

Dans un premier temps, un état de l'art sur l'évolution des systèmes technologies de l'information et de la communication pour l'éducation (TICE).

Le tutorat qui consiste à suivre l'apprenant et à l'assister dans son parcours de formation.

Avec l'enseignement à distance se développe une nouvelle notion de tutorat. Le formateur devient un guide dans les processus d'enseignement et d'apprentissage personnalisés.

L'accompagnement implique des moments (avant, pendant, après la formation), des acteurs (Formateurs, enseignants, experts, intervenants, techniciens) et des outils (téléphone, courrier, Forum de discussion). Il peut être d'ordre technique, pédagogique ou méthodologique.

Il peut aussi inclure des phases de travail en commun entre les apprenants (apprentissage de type collaboratif).

Pour le traitement automatique des données textuelles, une technique de text mining a été utilisée. Dans le chapitre analyse et conception, démarche méthodologique à été présentée.

Des étapes de l'implémentation de notre application en utilisant le JAVA ont été mise en œuvre.

Dans ce chapitre, nous présentons les différentes étapes de l'implémentation de notre application.

En optant dans un premier temps sur le choix de langage de développement, l'environnement et les outils d'implémentation de notre site. Par la suite, du script JAVA a été réalisé et synthétisé. Nous avons pu mettre en pratique ce site grâce aux connaissances acquises tout au long de ce projet.

Bibliographie:

[Nicaud, Vivet, 88] : Nicaud J.F, Vivet M., les Tuteurs Intelligents : Réalisation et tendances de recherches. TSI, numéro spécial : Application De L'informatique A La Formation, Dunod-Afcet, 1988.

[Bruillard, Vivet, 94] : Bruillard E., Vivet M., "Concevoir des EIAO pour des situations scolaires : Approche méthodologique", Recherche en Didactiques des mathématiques, n°1.2, vol.14, pp.275-304, édition La Pensée Sauvage, 1994.

[Ait adda, Oukhelifa, 06] : Ait adda S., Oukhelifa S., "Mise à distance de jeux d'entreprise multimédias coopératifs". Mémoire d'ingénieur en informatique, université Mouloud Mammeri, Tizi-Ouzou, 2006.

[Madiou, Mesloub, 08] : Madiou S., Mesloub F., "Evaluation d'apprenants en e-learning des bases de données relationnelles". Mémoire d'ingénieur en informatique, université Mouloud Mammeri, Tizi-Ouzou, 2008.

[Hammache, 06] : Hammache A., "Système d'inférence pour une indexation de documents basée sur une ontologie de domaine ".Thèse de magister, université Mouloud Mammeri, Tizi-Ouzou, 2006.

[Bernatchez (2001)]: vers une nouvelle typologie des activités d'encadrement et du rôle du tuteur, revue du conseil québécois de la formation à distance ;

[GLIKMAN, V., (2002)] : *Des cours par correspondance au e-learning*, Paris : PUF. TICE et développement, Numéro 0b, 9 octobre 2006. Récupéré le 7 avril 2009 de <http://www.revuetice.info/document.php?id=686>.

[Guillaume, N. (2009)] : *Un modèle d'animation : vision synthétique des fonctions tutorales*, Tutorales, la revue de t@d, n°2, Février 2009.

[Denis, B. (2003)] *Quels rôles et quelle formation pour les tuteurs intervenant dans des dispositifs de formation à distance ?* Distance et savoir. Volume 1, no. 1, pages 19 a 46.

Récupéré le 15 mai 2009 de <http://www.cairn.info/revue-distances-et-savoirs-2003-1-page-19.htm>;

[christian-faure] : introduction au text mining ; 2007

[Salton ., 1983] : Salton, G. and McGill, M.J. Introduction to Modern Information Retrieval. 1983. "Paul Jaccard." *Wikipedia, l'encyclopédie libre*

[Ani 90] :P.anick et J.pustejovsky :an application of lexical semantics to knowledge acquisition from corpora . dans conference on computational linguistics COLING 90 ,volume 3 ,Helsinki ,1990.

[R. Feldman] : (1998a) Trend Graphs: Visualizing the Evolution of Concept Relationships in Large Document Collections, in Zytkow et Quafafou (1998) p. 38-46.

[R. Feldman]: (1998b) Text Mining at the Term Level, in Zytkow et Quafafou (1998) p. 65 73.

[D. Landau] : (1998) An Integrated Visual Environment for Text Mining, in Zytkow et Quafafou (1998) p. 56-64.

[Fay 96]:U.fayyad ,G.Piatetsky-Shapiro et P.Smith:from data mining to knowledge discovery . dans AI Magazine ,1996

[Isabelle Valembois ,Louis Millecam] : « Java autoformation », Ellipse2001

Webliographie :

- [1]: <http://www.learn-on-line.be/le-tutorat-en-ligne>
- [2]: <http://www.umc.edu.dz/vf/index.php/services/formations/e-learning/48-plate-forme-elearning-wikipedia->
- [3]: <http://fr.wikipedia.org/wiki/Similarité-sémantique>
- [4]: <http://www.seo-camp.org/wp-content/uploads/2008/12/le-role-des-mesures-de-similarite.ppt>
- [5]: <http://lexicometrica.univ-paris3.fr/jadt/jadt2014/01-ACTES/51-JADT2014.pdf>
- [6]: <http://www.theses.fr/2012BOR14589>
- [9]: http://www.memoireonline.com/12/09/2917/m_Algorithmes-dapprentissage-pour-la-classification-de-documents1.html
- [10]: <http://alias-i.com/lingpipe/web/download.html>
- [11]: <http://garr.dl.sourceforge.net/project/hunspell/Hunspell/Documentation/hunspell4.pdf>
- [12]: <http://www.mysql.fr/products/workbench/>
- [13]: <http://www.developpez.net/forums/d1257813/logiciels/logiciels-libres-open-source/bureautique/openoffice-libreoffice/grammalecte-0-2-dictionnaire-4-6-a/>
- [14]: <http://www.drouizig.org/index.php/fr/binviou-fr/correcteur-hunspell/240-correcteur-hunspell-pour-mozilla-firefox-libreoffice-openoffice-org>
- [15]: <http://www.atelier.net/trends/articles/combinaison-affine-exclusion-mots-non-porteurs-de-sens>
- [16]: <http://taln09.blogspot.com/2009/02/etiquetage-morpho-syntaxique-et.html>
- [17]: <http://www.developez.fr/collatinus/>
- [18]: <http://fr.wiktionary.org/wiki/lemmatisation>
- [20]: <http://blog.onyme.com/lemmatisation-et-racinement-en-francais-flexion-lemme-et-racine-dun-mot/>
- [21]: <http://www.ranks.nl/stopwords/french>
- [21]: <http://www.atelier.net/trends/articles/combinaison-affine-exclusion-mots-non-porteurs-de-sens>
- [22]: <http://msdn.microsoft.com/fr-fr/library/ms142551.aspx>
- [23]: <http://www.developpez.com/developpons-java-spring-mvc>
- [24]: <http://api.jquery.com/jquery.ajax/>
- [25]: <http://stackoverflow.com/questions/4244521/javadoc-like-tool-for-spring-mvc-rest-api-documentation>
- [36]: <http://fr.wiktionary.org/wiki/lemmatisation>
- [26]: http://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm
- [27]: <http://docs.spring.io/docs/Spring-MVC-step-by-step/>
- [28]: http://www.linguatools.de/disco/disco_en.html#was
- [29]: <https://disco.readthedocs.org/en/0.4.5/lib/index.html>
- [30]: <https://discoapi.com/>
- [31]: <https://www.openthesaurus.de/synonyme/Disco>
- [33]: <http://www.naunaute.com/liste-stop-words-francais-393>
- [34]: <http://www.java2s.fr>

Introduction :

UML (Unified Modeling Language) est un langage standard conçu pour l'écriture de plans d'élaboration de logiciels. Il peut être utilisé pour visualiser, spécifier, construire et documenter les artefacts d'un système à forte composante logicielle.

UML est adapté à la modélisation de systèmes, depuis les systèmes informatiques d'entreprises jusqu'aux applications distribuées basées sur le Web, en passant par les systèmes temps réel embarqués. C'est un langage très expressif qui couvre toutes les perspectives nécessaires au développement puis au déploiement de tels systèmes.

UML n'est qu'un langage et ne constitue qu'une partie d'une méthode de développement logiciel. UML est indépendant du processus, c'est un formalisme, un outil de modélisation.

1. Historique :

Regardons tout d'abord ce qui s'est passé au début des années 90. Par rapport à la cinquantaine de méthodes d'analyse et de conception objet qui existaient au début des années 90, seulement trois d'entre elles se sont détachées nettement au bout de quelques années. En effet, la volonté de converger vers une méthode unifiée était déjà bien réelle et c'est pour cette raison que les méthodes OMT, BOOCH et OOSE se sont démarquées des autres.

OMT (*Object Modeling Technique*) de James Rumbaugh et BOOCH de Grady Booch ont été les deux méthodes les plus diffusées en France durant les années 90. Par ailleurs, OOSE de Ivar Jacobson s'est aussi imposée dans le monde objet pour la partie formalisation des besoins. Pour aller plus loin dans le rapprochement, James Rumbaugh et Grady Booch se sont retrouvés au sein de la société Rational Software et ont été ensuite rejoints par Ivar Jacobson en se donnant comme objectif de fusionner leur méthode et créer **UML** (*Unified Methode Language*).

Il est important de noter que contrairement à ce qui avait été envisagé au départ, le processus de développement a été sorti du champ couvert par le projet de norme. UML est donc une norme du langage de modélisation objet qui a été publiée, dans sa première version, en novembre 1997 par l'OMG (*Object Management Group*), instance de normalisation internationale du domaine de l'objet.

En quelques années, UML s'est imposée comme standard à utiliser en tant que langage de modélisation objet.

Aujourd'hui, en cette fin de la première décennie des années 2000, nous avons déjà une dizaine d'années de recul sur l'enseignement et la pratique d'UML en entreprise.

Les grandes étapes de la diffusion d'UML peuvent se résumer comme suit :

1994-1996 : rapprochement des méthodes OMT, BOOCH et OOSE et naissance de la première version d'UML.

23 novembre 1997 : version 1.1 d'UML adoptée par l'OMG.

1998-1999 : sortie des versions 1.2 à 1.3 d'UML.

2000-2001 : sortie des dernières versions suivantes 1.x.

2002-2003 : préparation de la V2.

10 octobre 2004 : sortie de la V2.1.

5 février 2007 : sortie de la V2.1.1 (version de référence du présent ouvrage).

2. Définition UML :

UML (Unified Modeling Language traduit en « langage de modélisation objet unifié ») est un langage de description orienté objet qui permet de modéliser une application selon une vision objet. Un objet est décrit par les attributs qui le compose et les traitements, appelé méthodes, qui peuvent lui être appliqués. Par exemple, l'objet client possède un numéro et un nom. Les méthodes applicables à l'objet client peuvent être : Consulter le client, créer, modifier ou supprimer un objet client.

3. Modèle conceptuel d'UML :

Le modèle conceptuel d'UML comprend les notions de bases génériques du langage. Il définit trois sortes de briques de base :

- Les éléments, qui sont les abstractions essentielles à un modèle.
- Les relations, qui constituent des liens entre ces éléments.
- Les diagrammes, qui regroupent des éléments et des liens au sein de divers ensembles.

4. Quelques définitions :

4.1 Les interactions :

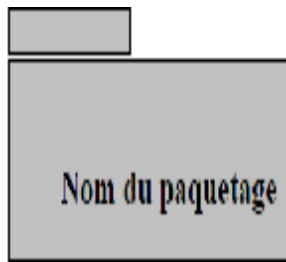
Une interaction est un comportement qui comprend un ensemble de messages échangés au sein d'un groupe d'éléments, dans un contexte particulier, pour atteindre un but bien défini. Le comportement d'un ensemble d'objets ou celui d'une opération individuelle peut être précisé par une interaction. Comme le montre la **figure suivante** un message est représenté par une ligne fléchée, qui indique le nom de son opération.

Atteint


4.2 Les éléments de regroupement :

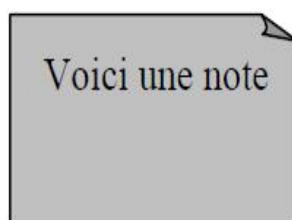
Les éléments de regroupements représentent les parties organisationnelles des modèles UML. Ce sont des boîtes dans lesquelles un modèle peut être décomposé. Il existe un seul type fondamental d'éléments de regroupement : le « paquetage ».

Un **paquetage** est un ensemble d'éléments de modélisation : des classes, des associations, des objets, des composants....



4.3 Les éléments d'annotation :

Les éléments d'annotation représentent les parties explicatives des modèles UML. Ce sont les commentaires qui peuvent accompagner tout élément dans un modèle, à des fins de description, d'exploitation et de remarque. Il existe un type fondamental d'éléments d'annotation appelé « note » qui est simplement un symbole utilisé pour représenter les contraintes et les commentaires rattachés à un élément ou un ensemble d'éléments. Comme le montre la **figure suivante** une note est représentée par un rectangle écorné qui contient un commentaire textuel ou graphique.



Dans l'étude conceptuelle de notre site, nous n'avons utilisé que trois diagrammes, à savoir le diagramme de cas d'utilisation, le diagramme de séquence et le diagramme de classe.

Nous allons décrire ces diagrammes ainsi que les éléments et les relations les constituant dans ce qui suit.

5. Les Diagrammes d'UML :

UML dans sa version 2 propose treize diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles.

• **Les diagrammes structurels** : Ces diagrammes, au nombre de six, ont vocation à représenter l'aspect statique d'un système (classes, objets, composants...).

- *Diagramme de classe.*
- *Diagramme d'objet.*
- *Diagramme de composant (modifié dans UML 2).*
- *Diagramme de déploiement (modifié dans UML 2).*
- *Diagramme de paquetage (nouveau dans UML 2).*
- *Diagramme de structure composite (nouveau dans UML 2).*

• **Les diagrammes de comportement** : Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Sept diagrammes sont proposés par

UML :

- *Diagramme des cas d'utilisation.*
- *Diagramme d'état-transition (machine d'état).*
- *Diagramme d'activités (modifié dans UML 2).*
- *Diagramme de séquence (modifié dans UML 2).*
- *Diagramme de communication (anciennement appelé collaboration).*
- *Diagramme global d'interaction (nouveau dans UML 2).*
- *Diagramme de temps (nouveau dans UML 2).*

5.1 Description des cas d'utilisation :

Les cas d'utilisations sont des outils formels qui permettent de consigner et d'exprimer des interactions entre les utilisateurs et le système. On peut noter que les cas d'utilisations sont utilisés durant tout le processus car ils servent à la création de l'IHM, à la spécification des tests.

a. L'acteur :

Un acteur est un type stéréotypé représentant une abstraction qui réside juste en dehors du système à modéliser. Un acteur représente un rôle joué par une personne ou une chose qui interagit avec le système.

Un acteur n'est pas nécessairement une personne physique : il peut être un service, une société, un système informatique ...

Il existe 4 catégories d'acteurs :

- **les acteurs principaux** : les personnes qui utilisent les fonctions principales du système
- **les acteurs secondaires** : les personnes qui effectuent des tâches administratives ou de maintenance.
- **le matériel externe** : les dispositifs matériels incontournables qui font partie du domaine de l'application et qui doivent être utilisés.
- **les autres systèmes** : les systèmes avec lesquels le système doit interagir.

b. Le cas d'utilisation :

Le cas d'utilisation (ou use case) correspond à un objectif du système, motivé par un besoin d'un ou plusieurs acteurs.

L'ensemble des use cases décrit les objectifs (le but) du système.

c. La relation :

Elle exprime l'interaction existant entre un acteur et un cas d'utilisation.

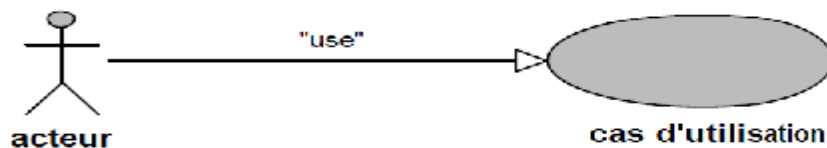
Il existe 3 types de relations entre cas d'utilisation :

- la relation d'utilisation
- la relation d'extension
- la relation d'inclusion

c.1. La relation généralisation :

Dans une relation de généralisation entre 2 cas d'utilisation, le cas d'utilisation enfant est une spécialisation du cas d'utilisation parent.

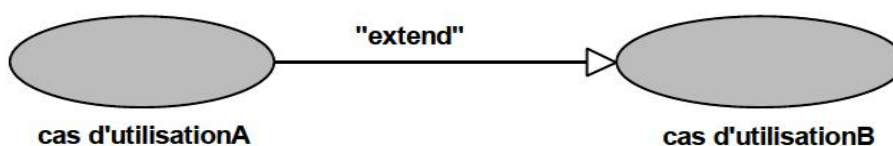
Formalisme :



c.2. La relation d'extension :

Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à condition. Le comportement ajouté est inséré au niveau d'un point d'extension défini dans le cas d'utilisation destination. Cette relation permet de modéliser les variantes de comportement d'un cas d'utilisation (selon les interactions des acteurs et l'environnement du système).

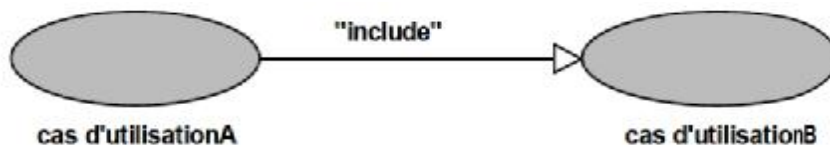
Formalisme :



c.3. La relation d'inclusion :

Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. L'inclusion a un caractère obligatoire, la source spécifiant à quel endroit le cas d'utilisation cible doit être inclus. Cette relation permet ainsi de décomposer des comportements et de définir des comportements partageables entre plusieurs cas d'utilisation.

Formalisme :



5.2 Le diagramme de classes:

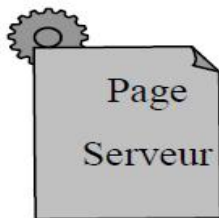
Le diagramme de classes est un diagramme structurel qui exprime d'une manière générale la structure statique d'un système en termes de classes, d'interfaces et de collaborations, ainsi que leurs relations.

5.2.1 Eléments de Base :

a. Classe :

- Page serveur « Server page » :

Icône :

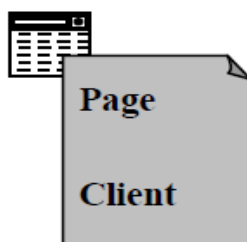


Description : Représente une page Web possédant des scripts qui interagissent avec des ressources serveur telles que les bases de donnée, ces scripts sont exécuté par le serveur.

Contraintes : Les pages serveur ne peuvent avoir de relation qu'avec des objets sur le serveur.

- Page client « client page » :

Icône :

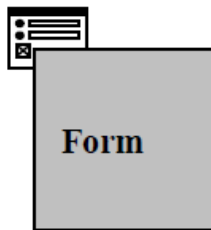


Description : Une instance d'une page client est une page Web formatée en HTML. Les pages clients peuvent contenir des scripts interprétés par les navigateurs lorsque celles-ci sont restituées par ces derniers. Les fonctions des pages clients correspondent aux fonctions des scripts de la page web.

Contrainte : Aucune.

- **Formulaire :**

Icône :



Description : Une classe stéréotypée « form » est un ensemble de champs de saisie faisant partie d'une page client. A une classe formulaire correspond une balise HTML <form>, les attributs de cette classe correspondent aux éléments de saisie d'un formulaire HTML (zone de saisie, zone de texte, boutons d'option.).

Un formulaire n'a pas d'opérations, puisqu'il peut les encapsuler. Toute opération qui interagit avec le formulaire appartient à la page qui la contient.

Contraintes : Aucune.

b. Association :

- **Lien « link » :**

Icône : Aucune

Description : Un lien est un pointeur d'une page client vers une autre page. Dans un diagramme de classe, un lien est une association entre une page client et une autre page client ou une page serveur.

Contraintes : Aucune.

- **Soumet « submit » :**

Icône : Aucune.

Description : submit est une association qui se trouve toujours entre un formulaire et une page serveur. Les formulaires soumettent les valeurs de leurs champs au serveur, par l'intermédiaire de pages serveur, pour qu'il les traite. Le serveur Web traite la page serveur, qui accepte et utilise les informations du formulaire.

Contraintes : Aucune.

- **Construit « build » :**

Icône : Aucune.

Description : La relation « build » est une relation particulière qui fait le pont entre les pages client et les pages serveur. L'association « build » identifie quelle page serveur est responsable de la création d'une page client. C'est une relation orientée, puisque la page client ne connaît pas la page qui est à l'origine de son existence.

Une page serveur peut construire plusieurs pages client, en revanche, une page client ne peut être construite que par une seule page serveur.

Contraintes : Aucune.

- **Redirige « redirect » :**
Icône : Aucune

Description : Une relation « redirect », est une association unidirectionnelle avec une page web, peut être dirigée à partir d'une page client ou serveur vers une autre page client ou serveur.

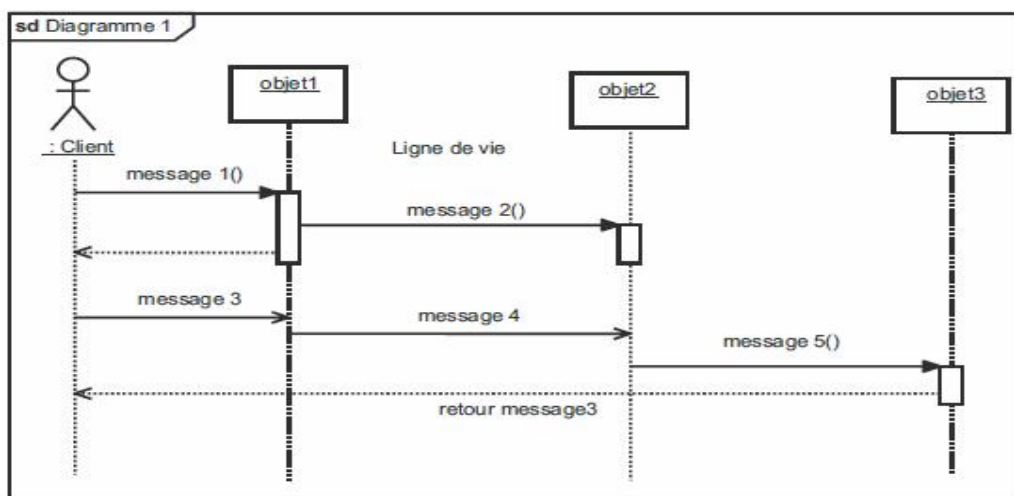
Contraintes : Aucune.

5.3. Diagramme de séquence :

Le diagramme de séquences montre les interactions entre objets selon un point de vue temporel. La représentation du contexte des objets se concentre sur l'expression des interactions.

Un objet est matérialisé par une barre verticale appelée ligne de vie des objets. Les objets, communiquent en échangeant des messages représentés au moyen de flèches orientées, de l'émetteur du message vers le destinataire. L'ordre des messages est donné par leur position sur l'axe vertical

Formalisme :



Les objets d'analyse utilisés sont repartis en trois catégories :

- Les objets d'interfaces :

Un objet d'interface représente l'interface entre l'acteur et le système tels des pages web ou les écrans de saisie.

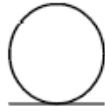
L'icône :



- Les objets entité :

Ils sont des objets décrits dans un cas d'utilisation et qui se trouvent dans d'autres cas d'utilisation tels les fournisseurs, article, commande ...

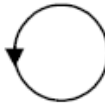
L'icône :



- Les objets contrôle :

Il représente les activités des processus du système, ils dirigent les activités des entités et interfaces. Ces objets sont obtenus en extrayant les verbes des cas d'utilisation.

L'icône :



5.3.1 Fragment d'interaction :

a. Types de fragments d'interaction :

Dans un diagramme de séquence, il est possible de distinguer des sous-ensembles d'interactions qui constituent des fragments.

Un **fragment d'interaction** se représente globalement comme un diagramme de séquence dans un rectangle avec indication dans le coin à gauche du nom du fragment.

Un port d'entrée et un port de sortie peuvent être indiqués pour connaître la manière dont ce fragment peut être relié au reste du diagramme. Dans le cas où aucun port n'est indiqué c'est l'ensemble du fragment qui est appelé pour exécution.

Un **fragment d'interaction** dit combiner correspond à un ensemble d'interaction auquel on applique un **opérateur**. Un fragment combiné se représente globalement comme un diagramme de séquence avec indication dans le coin à gauche du nom de l'opérateur.

Treize opérateurs ont été définis dans UML : alt, opt, loop, par, strict/weak, break, ignore/consider, critical, negative, assertion et ref.

b. Les opérateurs :

Opérateur « alt »

L'opérateur « **alt** » correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il est aussi permis d'utiliser les clauses de type sinon.

Formalisme

L'opérateur « alt » se représente dans un fragment possédant au moins deux parties séparées par des pointillés.

Opérateur « opt »

L'opérateur « **opt** » (optional) correspond à une instruction de test sans alternative (sinon).

Formalisme

L'opérateur « opt » se représente dans un fragment possédant une seule partie

Opérateur « loop »

L'opérateur « **loop** » correspond à une instruction de boucle qui permet d'exécuter une séquence d'interaction tant qu'une condition est satisfaite.

Il est possible aussi d'utiliser une condition portant sur un nombre minimum et maximum d'exécution de la boucle en écrivant : «loop min, max ». Dans ce cas, la boucle s'exécutera au minimum min fois et au maximum max fois. Il est aussi possible de combiner l'option min/max avec la condition associée à la boucle.

Formalisme

L'opérateur « loop » se représente dans un fragment possédant une seule partie et englobant toutes les interactions faisant partie de la boucle.

Opérateur « par »

L'opérateur « **par** » (parallel) permet de représenter deux séries d'interactions qui se déroulent en parallèle.

Formalisme

L'opérateur « par » se représente dans un fragment possédant deux parties séparées par une ligne en pointillé. C'est un opérateur qui est à notre avis plutôt utilisé dans l'informatique temps réel.

Opérateurs « strict » et « weak sequencing »

Les opérateurs « **strict** » et « **weak** » permettent de représenter une série d'interactions dont certaines s'opèrent sur des objets indépendants :

- L'opérateur « strict » est utilisé quand l'ordre d'exécution des opérations doit être strictement respecté.
- L'opérateur « weak » est utilisé quand l'ordre d'exécution des opérations n'a pas d'importance.

Opérateur « break »

L'opérateur « **break** » permet de représenter une situation exceptionnelle correspondant à un scénario de rupture par rapport au scénario général. Le scénario de rupture s'exécute si la condition de garde est satisfaite.

Opérateurs « ignore » et « consider »

Les opérateurs « **ignore** » et « **consider** » sont utilisés pour des fragments d'interactions dans lesquels on veut montrer que certains messages peuvent être soit absents sans avoir d'incidence sur le déroulement des interactions (ignore), soit obligatoirement présents (consider).

Opérateur « critical »

L'opérateur « **critical** » permet d'indiquer qu'une séquence d'interactions ne peut être interrompue compte tenu du caractère critique des opérations traitées. On considère que le traitement des interactions comprises dans la séquence critique est atomique.

Opérateur « negative »

L'opérateur « **neg** » (negative) permet d'indiquer qu'une séquence d'interactions est invalide.

Opérateur « *assertion* »

L'opérateur « **assert** » (assertion) permet d'indiquer qu'une séquence d'interactions est l'unique séquence possible en considérant les messages échangés dans le fragment.

Toute autre configuration de message est invalide.

Opérateur « *ref* »

L'opérateur « **ref** » permet d'appeler une séquence d'interactions décrite par ailleurs constituant ainsi une sorte de sous-diagramme de séquence.

5.3.2 Message synchrone et asynchrone :

Dans un diagramme de séquence, deux types de messages peuvent être distingués :

- **Message synchrone** : Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message. Le message retour peut ne pas être représenté car il est inclus dans la fin d'exécution de l'opération de l'objet destinataire du message.
- **Message asynchrone** : Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. C'est une flèche avec une extrémité non pleine qui symbolise ce type de message.

5.4. Les autres diagrammes UML :

- **Diagramme d'objet** : Le diagramme d'objet permet la représentation d'instances des classes et des liens entre instances.
- **Diagramme de composant (modifié dans UML 2)** : Ce diagramme représente les différents constituants du logiciel au niveau de l'implémentation d'un système.
- **Diagramme de déploiement (modifié dans UML 2)** : Ce diagramme décrit l'architecture technique d'un système avec une vue centrée sur la répartition des composants dans la configuration d'exploitation.
- **Diagramme de paquetage (nouveau dans UML 2)** : Ce diagramme donne une vue d'ensemble du système structuré en paquetage. Chaque paquetage représente un ensemble homogène d'éléments du système (classes, composants...).
- **Diagramme de structure composite (nouveau dans UML 2)** : Ce diagramme permet de décrire la structure interne d'un ensemble complexe composé par exemple de classes ou d'objets et de composants techniques. Ce diagramme met aussi l'accent sur les liens entre les sous-ensembles qui collaborent.
- **Diagramme d'état-transition (machine d'état)** : Ce diagramme montre les différents états des objets en réaction aux événements.
- **Diagramme d'activités (modifié dans UML 2)** : Ce diagramme donne une vision des enchaînements des activités propres à une opération ou à un cas d'utilisation. Il permet aussi de représenter les flots de contrôle et les flots de données.
- **Diagramme de communication (anciennement appelé collaboration)** : Ce diagramme est une autre représentation des scénarios des cas d'utilisation qui met plus l'accent sur les objets et les messages échangés.

- *Diagramme global d'interaction (nouveau dans UML 2)* : Ce diagramme fournit une vue générale des interactions décrites dans le diagramme de séquence et des flots de contrôle décrits dans le diagramme d'activités.
- *Diagramme de temps (nouveau dans UML 2)* : Ce diagramme permet de représenter les états et les interactions d'objets dans un contexte où le temps a une forte influence sur le comportement du système à gérer.

6. Les points forts et points faible d'UML

6.1 Les points forts :

- a. UML est un langage formel et normalisé, gain de précision, gage de stabilité encourage l'utilisation d'outils
- b. UML est un support de communication performant, il cadre l'analyse, il facilite la compréhension de représentations abstraites complexes, son caractère polyvalent et sa souplesse font un langage universel.

6.2. Les points faibles :

La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation. La nécessité de s'accorder sur des modes d'expression communs est vitale en informatique. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue

Les auteurs d'UML sont tout à fait conscients de l'importance du processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse objet performant et standard.

Conclusion

Cette annexe a proposé une présentation brève des principaux concepts de modélisation d'UML et son extension pour le web ainsi que leur notation.