République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Larbi Tébessi –Tébessa-

Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie

Département : Mathématique et Informatique

MEMOIRE DE MASTER

Domaine: Informatique

Filière: Informatique

Option: Système d'information

Thème:

Développement d'un moteur de recherche pour la sélection d'un composant logiciel

Présenté par:

Ghrieb Abdelatif

Saker Mohamed Nabil

Devant le jury:

Président : A.Zeggari	MAA	Université de Larbi Tébessi
Rapporteur : A.Zebdi	MAA	Université de Larbi Tébessi
Examinateur : Y. Menassel	MAA	Université de Larbi Tébessi

Date de soutenance: 29/05/2016

Note :……. Mention :……

Résume 1

CHAPITRE 1 4

1-INTRODUCTION 4

1.1-Génie logiciel[3] 4

1.2-problématique 5

CHAPITRE 2 6

1) ETAS DE LART 6

1.1) Génie logiciel a base de composant 6

1.2) Définitions d 'un composons logiciel 6

1.3) Caractéristiques fonctionnelles et non fonctionnelle 7

2) Travaux réalises 8

2.1) les techniques classique de recherche d'information 10

2.2) les techniques de classification externe 10

2.2.1) représentation par facettes et vocabulaire 10

2.3) Techniques de classification structurelle 11

2.4) RECHERCHE DINFORMATION 11

3) Contributions 13

3.1) MOTIVATION 13

4) Présentations de l'application lucene 13

4.1) Indexation des données 15

4.2) Processus d'indexation 15

4.3) L'analyse 16

5) CARACTERISATION DES COMPOSANTS LOGICIELS 16

6) Classification de composants logiciels (COMPENENTsource) 17

7) RECHERCHE DINFORMATION 18

7.1) Recherche : modèles et requêtes 18

8) Rendu Résultats 19

9) Score lucene[13] 20

CHAPITRE 3 22

Conception de la plateforme

1) Conception Générale 22

1.1) Choix dune méthode de conception 22

1.2) langage de modélisations UML 23

2) Conception détaillée 23

2.1) le diagramme des cas d'utilisation 23

2.1.1) Composition du diagramme de cas utilisateur 24

2.2) Les acteurs de notre système 25

2.3) Diagrammes des séquences 27

2.3.1) Diagramme de séquence de note système 27

2.4) Diagramme de classe 30

2.4.1) Les concepts de base de diagrammes de classes 30

2.4.2) Règles de gestion 30

2.5) Diagramme de déploiement 32

CHAPITRE 4 34

1) Environnement de travail 34

2) Logiciels 34

3) Langage de programmation C# 35

4) Technique utilisé 35

Bibliographie

Résume

Dans la progression à base de composants logiciels ; identifier et sélectionner un composant approprié est une étape primordiale, qui va impacter directement sur le coût et le temps de mise en marché du logiciel.

Nous avons proposé par la suite une approche de sélection tout en tirant profit en même temps de la description externe des composants et leurs propriétés structurelles.

La description externe donnée aux composants est basée sur une nouvelle approche de caractérisation à base de document lucene[1] enrichie par des caractéristiques issues du COMPENENTsource[2] et de score lucene pour la mesure de la pondération des composants(document).

La recherche se fait en se basant sur les différents types de requêtes de la bibliothèque lucene pour la recherche d'information.

ملخص

تقدم مكونات البرنامج القائم على تحديد واختيار عنصر مناسب هو خطوة أساسية،والتي سوف تؤثر بشكل مباشر على التكلفة والوقت لتسويق البرمجيات.

اقترحنا لاحقا نهج للاختيار و ذلك للاستفادة في الوقت نفسه من الوصف الخارجي للمكونات وخصائصهم الهيكلية.

ويستند هذا الوصف الخارجي للمكونات على نهج جديد للوصف يستند علي وثيقة لوسين **lucene** (أوجه),الثرية بالخصائص الناتجة عن COMPENENTsource و نتيجة لوسين **lucene** لقياس وزن المكونات (وثيقة).

واعتمدت الدراسة على أنواع مختلفة من الاستفسارات من مكتبة لوسين **lucene** من أجل البحث عن المعلومات**.**

Abstract

In progression-based software components; identify and select a suitable component is an essential step, which will impact directly on the cost and time of software marketing.

We proposed a subsequent selection approach while leveraging the same time the external description of the components and their structural properties.

The description given to external components is based on a new characterization approach based document lucene (facets) enriched by characteristics resulting from COMPENENTsource and lucene score for measuring the weight of components (document).

The research is based on the different types of queries from the lucene library to research information.

CHAPITRE 1

1-INTRODUCTION

1.1-Génie logiciel

A une certaine époque à ses débuts l'activité d'écriture du logiciel ne reposait que sur l'efficacité personnelle du programmeur laisse pratiquement seul devant l programmation d'un problème.

De nos jours, le programmeur dispose d'outils et de méthodes lui permettant de concevoir et décrire des logiciels.

Le génie logiciel concerne l'ensemble des méthodes et règles relatives a la production rationnelle des logiciels.

L'activité de développements du logiciel vu les couts quelle implique est devenue une activité économique et doit donc être planifiée et soumise a des normes.

C'est pour quoi dans ce thème le mot clef est le mot <génie logiciel

à base de composant> qui tient a la fois de l'activité créatrice

de l'humain et du composant logiciel incluant une activité disciplinée et ordonnée basée pour certaines taches sur des outils (formalises) [3].

Donc: L'objectif principal de l'approche orientée composant est de critères purement économiques :

Critères économiques

. Augmenter la productivité des développeurs

-concurrence international/mondialisation

. Réduire la complexité des logiciel

- diminuer les couts de développements

- diminuer les couts de maintenance

. Réduire le temps de mise sur le marché
.Construire les applications par assemblage de composant logiciels pré-excitants.

1.2-problématique

la problématique centrale de notre travail est:
comment sélectionner des composants logiciels en prenant en compte les techniques de recherche de composants logiciels

et les méthodes de pondérations.

Dans le cadre de notre thème nous apportons une solution à la question de recherche suivante.

- **la première réside dans le choix d'un format de description pour les composants logiciels**
- **le deuxième réside dans le chois d'un mécanisme de recherche et de pondération pour évaluer l'importance des écarts entre propriétés candidates et recherchées.**

Nous apportons deux éléments de réponde a cette question

de recherche.

- **En premier lieu, nous avant proposer une approche de caractériser les composants logiciels bases sur le document lucene et le technique de classification externes (COMPONENTsource).**
- **En second lieu, développer un moteur de recherche capable de sélection des composants logiciels bases sur l'application lucene.**

L'objectif de notre travail est développement un moteur de recherche pour la sélection dune composants logiciel.

CHAPITRE 2

1) ETAS DE LART

1.1) Génie logiciel a base de composant

L'approche à base de composants est depuis une dizaine d'années considérées comme un nouveau paradigme de développement des systèmes d'information [BAR02] les activités de programmation ont été les premiers concernés par ce nouveau paradigme [4].

1.2) Définitions d'un composons logiciel :

La définition qui nous servira de guide dans la suite de ce thème est extraite de (Szyperski, 2002) [5]: « Un composant logiciel est une unité de composition ayant des interfaces

Contractualisées ainsi que des dépendances contextuelles. Ces dépendances explicitent les interfaces exigées ainsi que les plate-forme d'exécutions possibles. Un composant peut être déployé indépendamment, et être sujet à la composition». Un composant est donc une boite noire avec des interfaces bien définies pour les services qu'il fournit ou requiert. Les applications sont ensuite construites

par assemblage (ou composition) de composants en

fonction de leurs interfaces fournies et requises.

1.3) Caractéristiques fonctionnelles et non fonctionnelle

Nous caractérisons un composant logiciel par un ensemble d'interfaces fournies et requises.

Dons les deux cas (interface fournies, interface requise)

d'un composant contient un ensemble de propriétés fonctionnelles et non fonctionnelles qui sont visible depuis son environnement donc.

Les propriétés fonctionnelles des interfaces prennent la forme d'Operations dont on donne la signature et parfois la sémantique.

-les propriétés fonctionnelles décrivent les aspects syntaxiques

et sémantiques des services fournis et requis par un composant.

-les propriétés non fonctionnelles : représentent la manière

de rendre ou d'attendre ce service.

Dans le cadre de notre thème nous considérons de manière

interchangeable les propriétés non fonctionnelles

et la caractéristique d'un composant.

2) Travaux réalises

Les composants sur étagère sont des composants logiciels développés par des tiers et stockés

Dans des référentiels (repos tories) pour être utilisés ultérieurement dans la construction d'applications.

A cet effet, il existe un besoin grandissant pour la sélection

de composants pertinents répondant aux exigences

des développeurs.

Cependant, le succès de la sélection est problématique pour un certain nombre de raisons:

- Nature boite noire des composants : Le fait de ne pas pouvoir accéder au code source rend restrictive la compréhension des composants et par conséquent rend l'évaluation difficile.

- Mises à jour continues des composants : La fréquence élevée des modifications dans les référentiels de composants rend l'évaluation difficile.

Par exemple : Une nouvelle version du composant peut avoir une fonctionnalité qui n'est pas disponible dans le composant en cours d'évaluation.

De nombreux travaux on été réalisés dans ce domaine entre autres les travaux [7], [8].

Apres une étude approfondie sur le travail [8], qui propose une approche de caractérisations a base de facettes pour ce système de recherche.

Vue	Facettes
Vue générale	-nom du composant - rôle du composant - visibilité - licence - version - domaine
Vue technologie De développements	- Langage de programmation - Model

Fig1 : Caractérisation des composants par facettes

Autre travail [9] remarquer que la recherche de composants logiciels a travers les facettes fig1 seulement est insuffisant cela est due

au manque d'information sur la qualité du composant.

Les caractéristiques (qualité) retenues dans le travail [9] sont base sur le standard iso -9126 enrichi par des métriques de mesure (LEVEL, PRESENCE, RADIO).

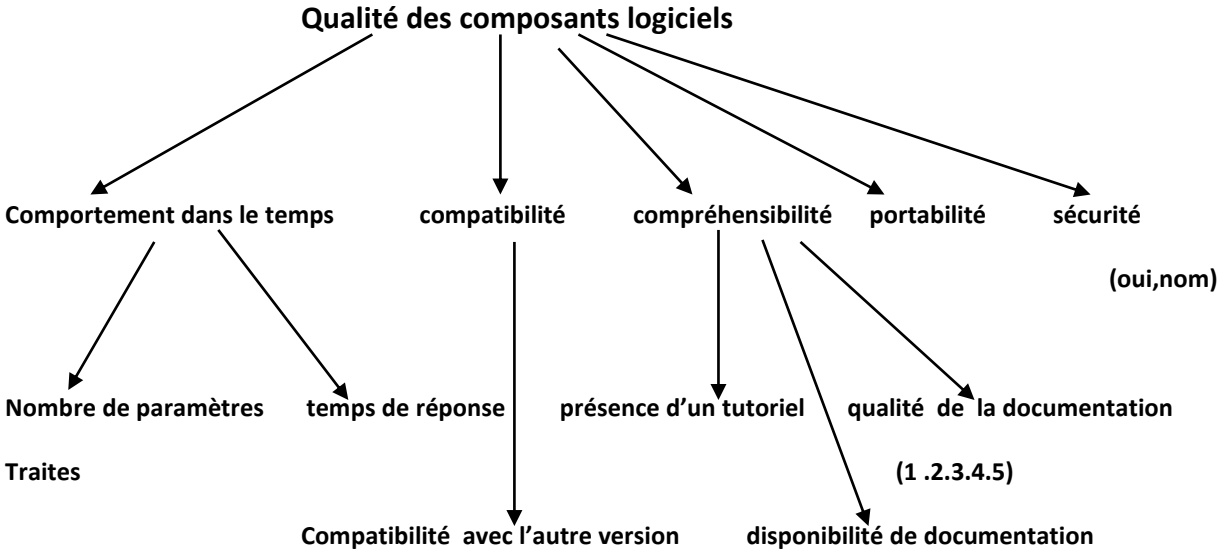


Fig. 2. caractéristiques et sous-caractéristiques « qualité » retenues

2.1) les techniques classique de recherche d'information [6]

il existe plusieurs techniques de recherche d'information dont le but est d'avoir un résulta performant.

La recherche et la sélection des composants sont effectuées selon deux aspects.

aspect non fonctionnel: consiste a rechercher les composants logiciels en utilisant leur descriptions externes a base de facettes baser sur le standard iso -9126 , ainsi que l'ensemble des caractéristiques et des sous caractéristiques du composants logiciels ont été pondérés à l'aide des deux techniques de prise de décision WSM[1] et AHP[2].

Aspect fonctionnel: Définit l'étape consacrée à la recherche en se basant sur les propriétés structurelles (INTERFACES) des composants logiciels [6].

2.2) les techniques de classification externe :

Les techniques de recherche de cette catégorie indexent une représentation externe des composants souvent préparée manuellement.

2.2.1) représentation par facettes et vocabulaire

L'approche de classification par facette enrichie par ISO 12 pour la recherche de composons [PRI 87, ZHA 00]

WSM[1] : Weight Sum Method

AHP[2] :Analyse Hiérarchique des procédés

ISO[3] : International **Organisation for Standardisation**

Consiste caractériser le composant par un ensemble de facettes, chaque facette représente une

Information permettant d'identifier et de sélectionner un composant

Nous distinguons deux formes de facettes

- les facettes utilisant un vocabulaire nom contrôlé et qui ne peuvent pas être cernées par un vocabulaire prédéterminé tels que : le nom du composant, le rôle, l'éditeur.

- les facettes utilisant un vocabulaire contrôlé, ce type de facette possède des descripteurs représentés par un certain nombre

de vocables prédéterminés, tels que : EJB[4], COM[5], NET[6].

2.3) Techniques de classification structurelle

Alors que les techniques précédentes se basent sur les descriptions externes de composants, les techniques de recherche structurelle concernent l'aspect structurel des composants (signatures

des méthodes) [6].

2.4) RECHERCHE DINFORMATION

Le travail [9] effectuer une recherche d'information dans les descriptions des composants ayant des facettes nom contrôler utilise le model vectoriel baser sur le cosinus azar [10]

Et utiliser la méthode intitulée WSM[1] pour pondérer les facettes.

EJB[4] : Entreprise Java Bean

COM[5] : correspond à l'abréviation de « commercial

NET[6] : est le nom donné à un ensemble de produits et de technologies informatiques de l'entreprise Microsoft

Exemple :

Prenons les textes suivants :

Texte 1 : la loutre est dans la rivière
Texte 2 : la loutre est avec les loutres dans la rivière

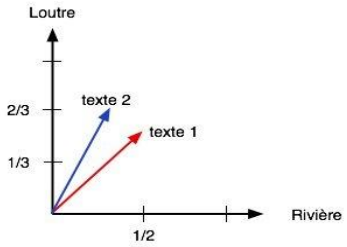
On va lemmatiser tout ça (sans prendre en compte les verbes) :

Texte 1 : loutre, rivière
Texte 2 : loutre, loutre, rivière

On va ensuite représenter chacun de ces textes par un vecteur dans l'espace des fréquences des mots rivière, loutre (dans cet ordre).

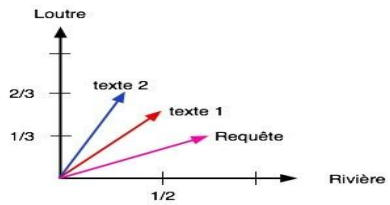
Texte 1 : (1/2, 1/2)
Texte 2 : (1/3, 2/3)

Ce qui donne graphiquement



Si l'on revient sur l'exemple, en prenant la requête « je suis dans

la rivière avec les loutres de la rivière » j'obtiens le vecteur (2/3, 1/3). Sur la graphique j'obtiens :



Et donc le texte le plus proche est le texte 1[11].

3) Contributions

Notre projet de recherche est proposé une application de sélection d'un composantes logiciels base sur le moteur de recherche lucene[12].

3.1) MOTIVATION

Premièrement :

Parmi les nombreuses qualités de cette bibliothèque, nous retiendrons essentiellement sa rapidité d'exécution et sa simplicité d'utilisation, tant pour le programmeur que pour l'utilisateur final.

Deuxièmes :

L'application lucene utilise le concept document comme facettes pour la description d'un composant logiciel.

Troisièmes :

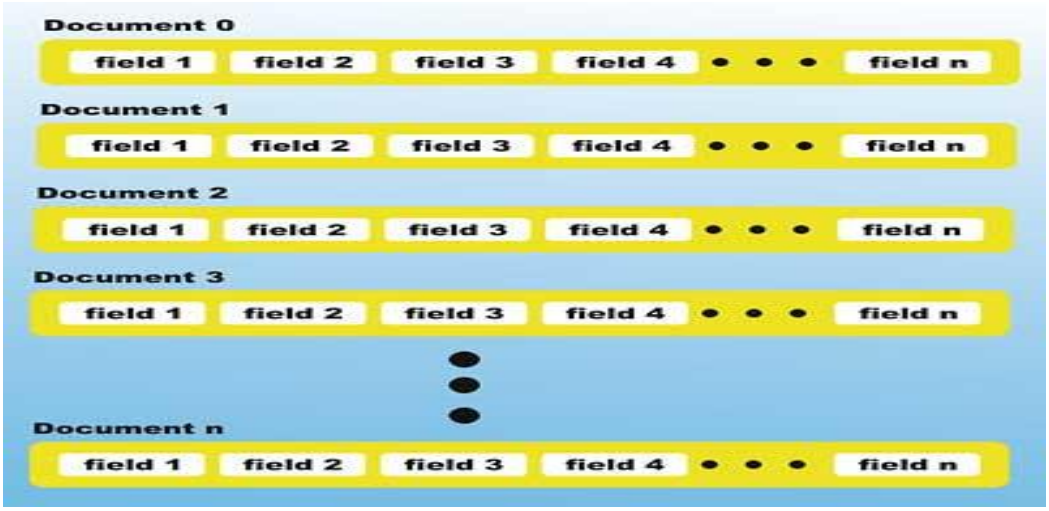
Moteur de recherche

lucene utilise Un moteur de recherche permettant de trouver des documents a partir d'une requête sous forme de mots.

4) Présentations de l'application lucene

Lucene appartient au fameux projet Jakarta de la fondation Apache, bien connu pour ses outils tels que Struts ou Tomcat. Cette API écrite en Java se destine à la création de puissants moteurs de recherche orientés texte. Parmi les nombreuses qualités de cette bibliothèque,

nous retiendrons essentiellement sa rapidité d'exécution et sa simplicité d'utilisation, tant pour le programmeur que pour l'utilisateur final. La découverte de Lucene se déroulera en deux temps. En premier lieu, nous apprendrons à créer un index à partir de nos documents, pour ensuite soumettre des requêtes au moteur, la figure1 montre le forma d'un document lucene[12].



Un document Lucene

D'après l'architecture générale de notre système présenté dans la figure 3, la recherche et la sélection des composants sont effectuée selon l'architecture de l'application lucene suivante.

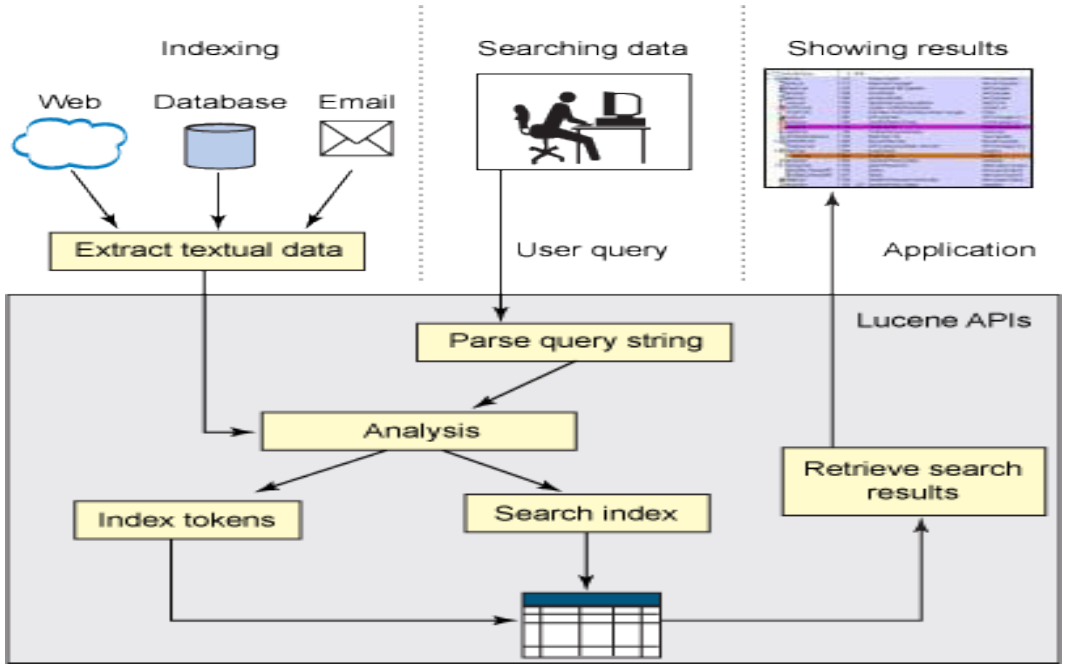


Fig 3 : les étapes de la construction d'application avec lucene

4.1) Indexation des données

Lucene vous permet d'indexer toute donnée disponible dans un format texte. Il peut être utilisé avec presque n'importe quelle source de données tant que de l'information textuelle peut en être extraite. La première étape dans l'indexation des données est de les rendre disponibles sous un format textuel simple. Pour ce faire, on utilise des analyseurs personnalisés et des convertisseurs

4.2) Processus d'indexation

L'indexation est le processus de conversion des données textuelles vers un format qui facilite une recherche rapide. Une analogie simple est celle de l'index que l'on trouve a la fin d'un livre : cet index indique la localisation des différents sujets qui sont traites dans le livre Lucene stocke les données en entrée dans une structure de données appelée index inversé, qui est lui-même stocké dans le système de fichiers ou en mémoire comme un ensemble de fichiers d'index. La plupart des moteurs de recherche Internet utilisent un

index inversé. Cela permet aux utilisateurs d'exécuter des recherches par mot-clé rapides et de trouver les documents répondant à une requête données. Avant que la donnée textuelle ne soit ajoutée à l'index, elle est traitée par un analyseur .

4.3) L'analyse

L'analyse est la conversion des données textuelles en unîtes fondamentales de recherche appelées termes. Pendant l'analyse, le texte subit plusieurs opérations : extraction des mots, suppression des mots les plus courants de la ponctuation, réduction des mots à leurs racines, passage en minuscules. L'analyse du texte est effectuée avant l'indexation et l'analyse des requêtes .Elle convertit les données textuelles en tokens, et ces tokens sont ajoutés en tant que termes à l'index lucene.

Lucene offre une grande variété d'analyseurs, tels que SimpleAnalyzer, StandardAnalyzer, StopAnalyzer, SnowballAnalyzer, sans être exhaustif. Ils diffèrent dans leur manière de découper le texte et d'appliquer des filtres. Comme l'analyse supprime des mots avant l'indexation, elle diminue la taille de l'index, mais peut avoir un effet négatif sur la précision du traitement des requêtes.

5) CARACTERISATION DES COMPOSANTS LOGICIELS

La caractérisation des composants logiciels consiste à déterminer les propriétés définissant un composant logiciel. Sur quelle base devrait-on déterminer et choisir ces propriétés, dans notre travail nous allons caractériser le composant par les champ utiliser dans COMPONENTsource[2].

6) Classification de composants logiciels (COMPENENTsource)

Catégorie

Couche de présentation
Coupe transversale
Couche de données
Couche métier

Composants

Composants .NET
Contrôles ActiveX
DLL
JavaScript / AJAX
Composants Java
Composants VCL
C++ / MFC
Windows Runtime Controls

Environnement / plate-forme

Microsoft
Visual Studio 2008
Visual Studio 2005

Marque d'éditeur

Infragistics

/n software

EldoS

Type de résultat

Compatibilité

Descriptions

Licences

Prix

- **Composant Travail**

Catégorie :……
Editeur :……
Type de résulta :……..
Composants :……
Environnement :…….

7) RECHERCHE DINFORMATION

pour effectuer une RI dans les descriptions des composants ayant comme des document lucene en commençant par le processus de l'application lucene suivantes fig8.

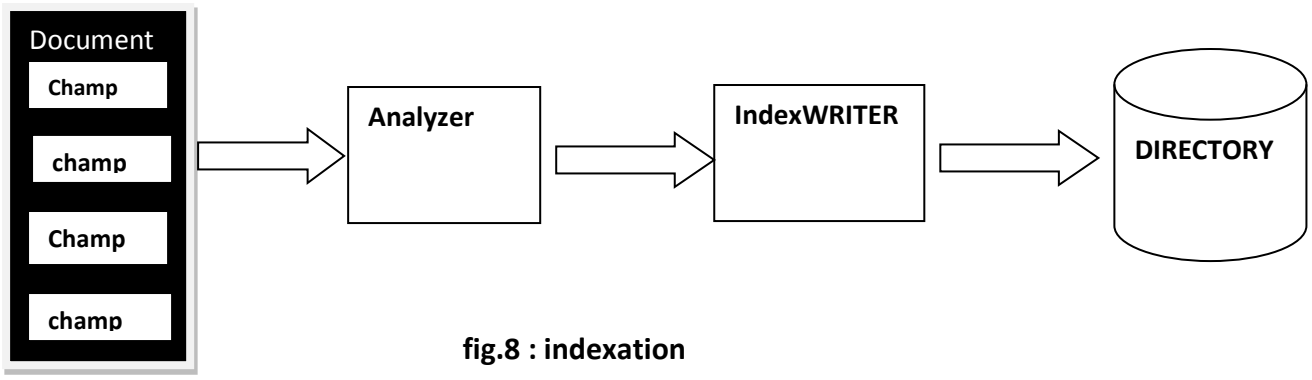


fig.8 : indexation

7.1) Recherche : modèles et requêtes

Lucene permet une multitude de requêtes, qui sont mises en œuvre dans le package org.apache.lucene.search. En voici quelques

exemples :

-Requêtes avec opérateurs booléens :

Les opérateurs booléens doivent toujours être écris en MAJUSCULE sinon ils seront considérés comme des termes. Les opérateurs booléens admis dans Lucene sont : OR, AND, NOT et +. Dans Lucene, l'opérateur par défaut est OR. Cela signifie que le OR s'utilise quand il n'y a pas d'opérateurs booléens entre deux termes.

- Les requêtes par terme :

L'utilisateur doit noter son terme sans guillemets

- Les requêtes par phrase : pour ce type de recherche,

l'utilisateur doit noter sa phrase entre guillemets

- Requêtes par champ :

Ce type de requête permet une recherche beaucoup plus précise. En effet, pour l'index possédant un grand nombre de champs, ces requêtes facilitent la recherche. Un même terme peut faire partie de deux champs différents. Par exemple, si je prends un des documents de ma base, les termes « Marque » et « Maintenon » font partis des champs : titre et contents. Donc, pour affiner et préciser leur recherche, les utilisateurs peuvent préciser le champ.

Exemple : titre : marque contents : Maintenon, cette recherche renverra les documents où le titre est maque et où l'on trouve dans le texte le terme Maintenon.

8) Rendu Résultats :

Une fois résultat est reçu la demande devrait décider de la façon de montrer les résultats à l'utilisateur à l'aide de l'interface utilisateur.

Combien d'informations doit être montré au premier coup d'œil et ainsi de suite.

9) Score lucene

Lucene met en œuvre une variante du modèle de pointage TFIDF
Les facteurs impliqués dans l'algorithme de notation de Lucene sont les suivantes:

1. **tf = fréquence du terme dans le document = mesure de la fréquence d'un terme apparaît dans le document**
2. **idf = document inverse fréquence = mesure de la façon dont souvent le terme apparaît dans l'index**
3. **coord = nombre de termes dans la requête qui ont été trouvés dans le document**
4. **lengthNorm = mesure de l'importance d'un terme en fonction du nombre total de termes dans le domaine**
5. **queryNorm = facteur de normalisation afin que les requêtes peuvent être comparées**

La mise en œuvre, l'implication et les justifications des facteurs 1,2, 3 et 4

1. tf
Mise en œuvre: sqrt (freq)
Implication: le plus souvent un terme se produit dans un document, plus son score
Justification: les documents qui contiennent plus d'un terme sont généralement plus pertinents

2. idf
Mise en œuvre: log (numDocs / (docFreq + 1)) + 1
Implication: plus l'apparition d'un terme dans différents documents, plus son score
Justification: termes communs sont moins importantes que les rares

3. coord
Mise en œuvre: chevauchement / maxOverlap
Implication: des termes de la requête, un document qui contient plus de termes aura un score plus élevé
Justification: auto-explicatif

4. lengthNorm
Mise en œuvre: 1 / sqrt (numTerms)
Implication: un terme adapté dans des domaines avec moins de termes ont un score plus élevé
Justification: un terme dans un champ avec moins de termes est plus important que celui avec plus
queryNorm est pas liée à la pertinence du document, mais tente de faire des scores entre les différentes requêtes comparables. Il est implémenté comme 1 / sqrt (sumOfSquaredWeights) [13] .

CHAPITRE 3

Conception de la plateforme

Dans le cycle de vie de notre projet, la conception représente une phase primordiale déterminante pour produire une application efficace. C'est dans ce stade que nous devons clarifier en premier lieu la conception générale que nous allons suivre dans la partie réalisation de notre projet. Puis, dans un deuxième lieu nous allons détailler notre choix conceptuel a travers plusieurs types de diagrammes [14].

1) Conception Générale

1.1) Choix dune méthode de conception

Il existe plusieurs méthodes de conception, citons comme exemples :UML[7] ,OMT[8],MERISE,CMM[9], autant de méthodes maitrisées par Additeam[4],elles permettent de diriger une transformation continue du système d'information en le simplifiant durablement pour des applications clientes indépendantes, des modules ou composants en forte cohérence, mais en couplage faible.

Dans notre démarche et pour la conception du projet, nous retenons le langage de modélisations UML pour :

-UML[7] est un langage formel et normalise : il permet le gain de précision, encourage l'utilisation d'outils et constitue a cet effet un gage de stabilité.

-UML[7] es t'un support de communication performant :il cadre l'analyse et facilite la compréhension de représentations

abstraites complexes. Sant caractères polyvalent et sa souplesse en font un langage universel.

1.2. Langage de modélisations UML[7]

UML n'est pas une méthode, en fait, et comme son nom l'indique est un langage graphique de modélisations des données et des traitements, fonde sur des concepts orientes objets.

UML2.0 comporte treize types de diagrammes représentant des vues distinctes du système d'information. Ils se repartissent en trois grands groupes.

- **diagrammes structurels ou diagrammes statiques**
- **diagrammes comportementaux ou diagrammes dynamiques**
- **diagrammes d'interaction**

Ces diagrammes, dune utilité variable selon les cas, ne sont pas nécessairement tous produits a chaque modélisations. Pour le cas de notre application, les diagrammes utilises Sant :

- **Diagramme de classe.**
- **Diagramme cas d'utilisation.**
- **Diagramme de séquence.**
- **Diagramme de déploiement.**

2) Conception détaillée

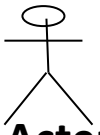
2.1) le diagramme des cas d'utilisation

Les rôles des diagrammes de cas d'utilisation sont de recueillir d'analyser et d'organiser les besoins, ainsi que de recenser les grandes fonctionnalités d'un système .il s'agit donc de la première étape UML pour la conception d'un système.

2.1.1) Composition du diagramme de cas utilisateur

Le diagramme de cas se compose de trois éléments principaux :

Un Acteur : C'est l'idéalisation d'un rôle joue par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom inscrit dessous.



Acteur

Un cas d'utilisation : C'est une unité cohérente représentant une fonctionnalité visible de l'extérieur.

il Réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui limite.

Il représente par une ellipse contenant le nom du cas (un verbe a l'infinitif)



Les relations :

On deux type de relation :

- **Etendue**
- **inclue**

2.2) Les acteurs de notre système

L'utilisateur : pour effectue une recherche nous pouvons interpréter la requête tapée par l'utilisateur et récupérer les résultats par l'appel de l'analyseur pour déterminer les contenues dans la requête fig. [6].

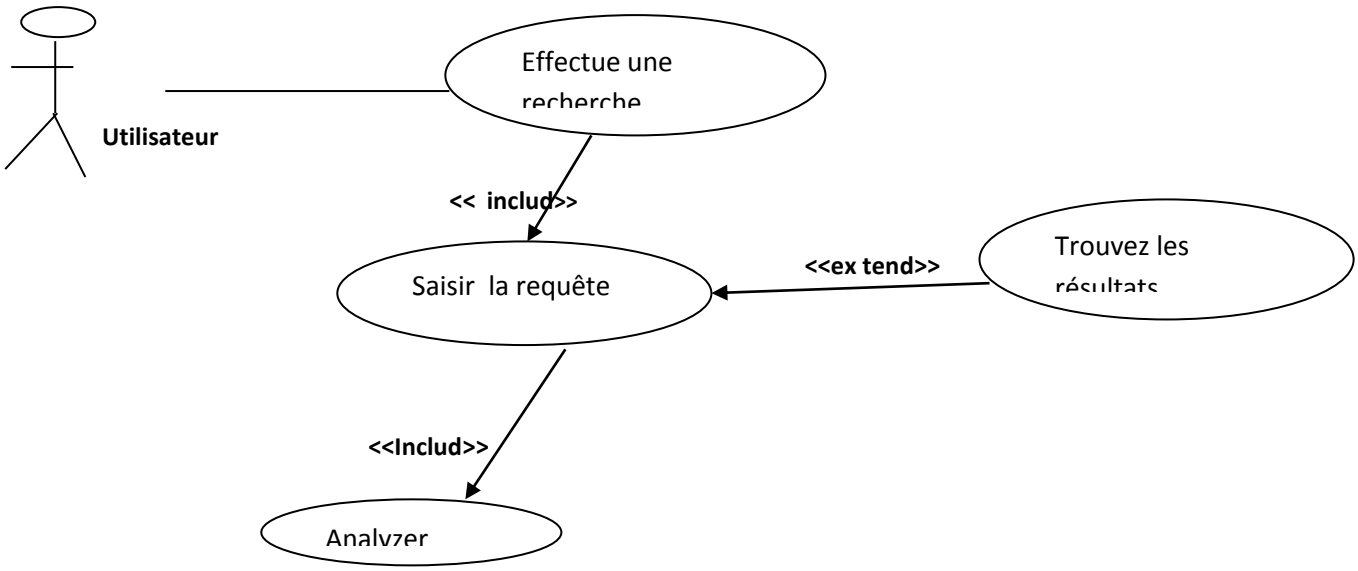


Fig. 6 : diagramme de cas d'utilisation utilisateur

L'administrateur : C'est celui qui assure le dynamisme du

Création des indexes Figure [7].

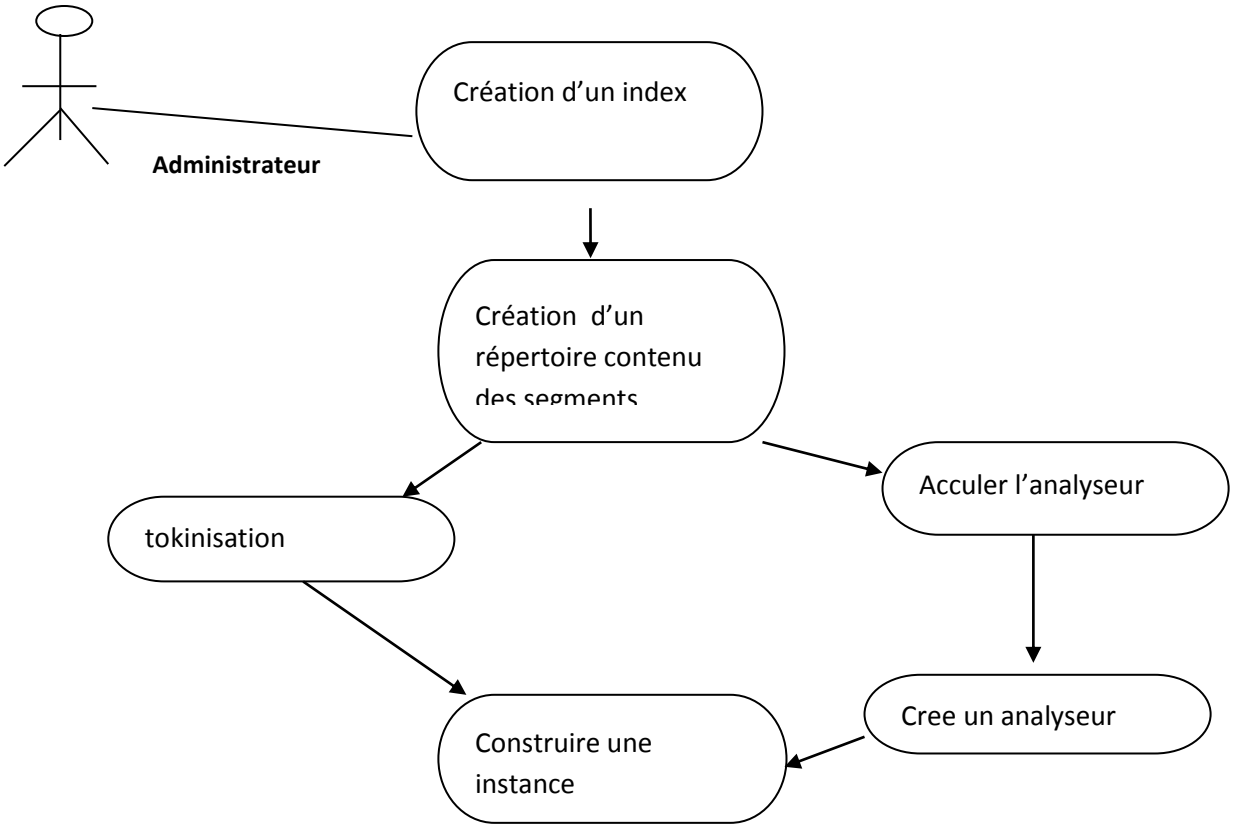


Fig.7 : diagramme de cas d'utilisation d'administrateur

2.3) Diagrammes des séquences

Les diagrammes de séquences permettent de représenter les interactions entre objets selon un point de vue temporel.

L'accent est mis sur la chronologie des envois de messages.

- **L'ordre d'envoi d'un message est détermine par sa position sur l'axe vertical du diagramme, le temps s'écoule '' de haut en bas '' de cet axe.**
- **La disposition des objets sur l'axe horizontal na pas de conséquence pour la sémantique du diagramme.**

Barre d'activation :

Sur un diagramme de séquence, il est aussi possible de représenter de manière explicite les différentes périodes d'activité d'un objet à la moyenne d'une bande rectangulaire superposée a la ligne de vie de l'objet [14].

2.3.1) Diagramme de séquence de note application

Sellent les cas utilisations de notre travail

- diagramme de séquence création d'index fig 8 .

- diagramme de séquence recherche d'un composant fig9.

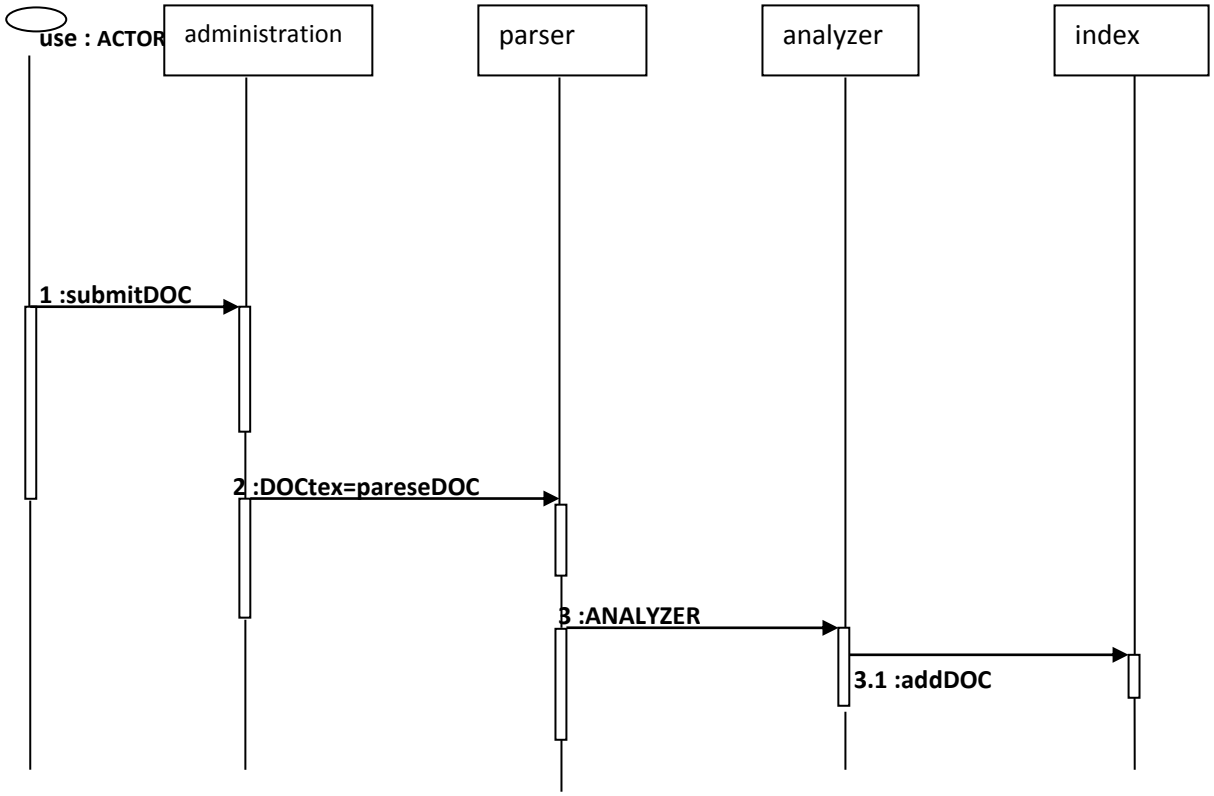


Fig.8 : Diagramme de séquences créations d'un index

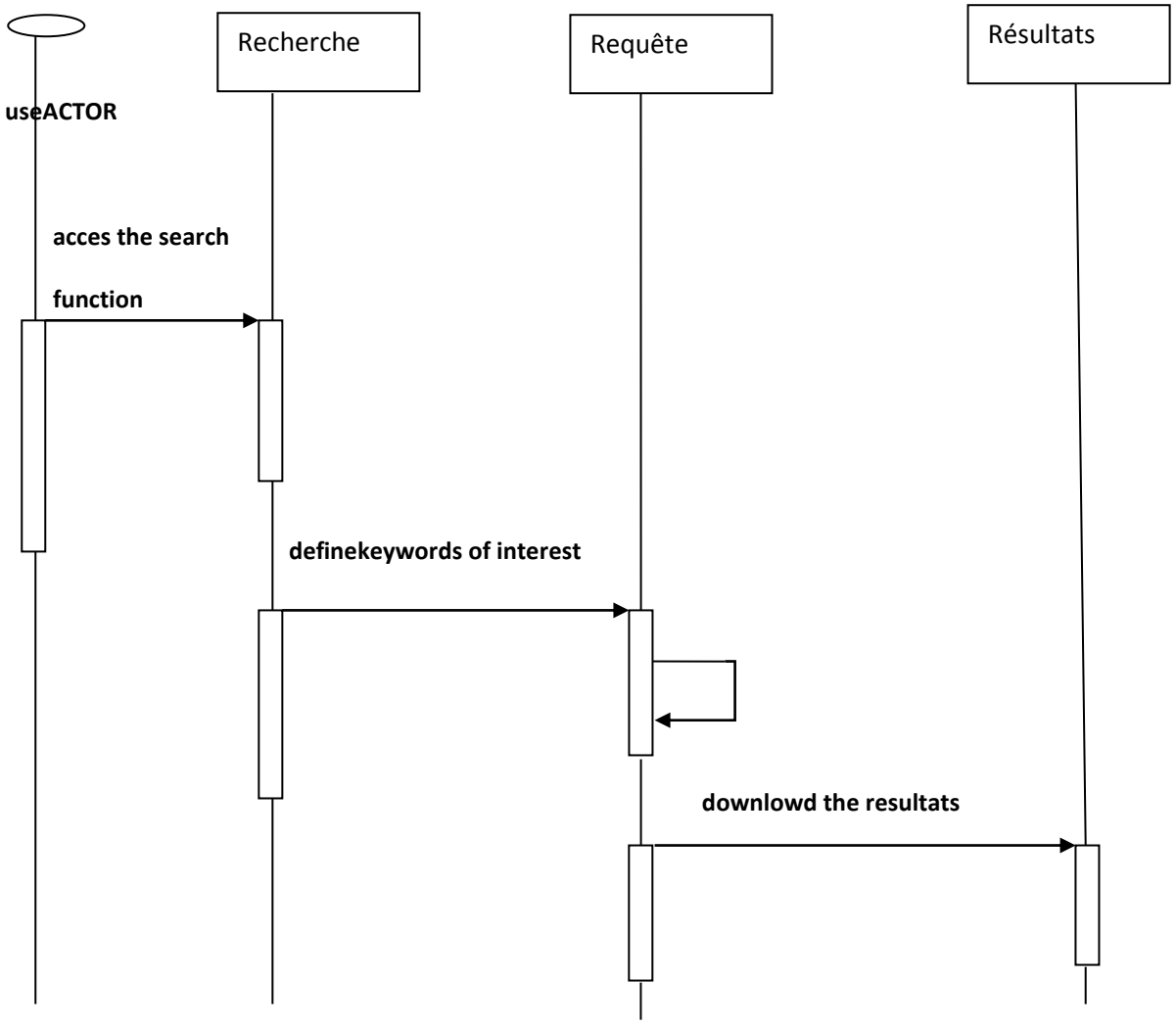


Fig. 9 : Diagramme de séquence recherche d'un composant

2.4) Diagramme de classe

Les diagrammes de classes sont sons doute les diagrammes les plus utilises D'uml .ils décrivent les types des objets qui composent un système et les différents types de relations statiques qui existent entre eux. Les diagrammes de classes font abstraction du comportement du système [15].

2.4.1) Les concepts de base de diagrammes de classes

Associations entre classes :

Une association est représentée par une ligne pleine entre deux classes.

Cardinalité :

Elle indique le nombre d'objets dune classe qui peuvent participer a l'association.

Les attributs :

Les attributs représentent les données encapsulées dans les objets des classes.

2.4.2) Règles de gestion :

1- **La classe Directory C'est une classe abstraite qui représente l'endroit ou les fichiers d'index sont stockes. Elle écrite deux classe(fsdirectory,ramdirectory).**
2- **FS directory : est une implémentation de la classe Directory qui stock les index dans le système de fichier.**
3- **Ram directory : est une implémentation de la classe directory qui stock tous les index en mémoire.**

4- La fonction des analyseurs est de prétraiter les données texte et de les convertir en termes stockes dans l'index.

5- La classe indexWRITER accepte un analyseur utilise pour découper la donnée avant de l'indexer.

Il fournit des méthodes pour ajouter, supprimer ou mètre a jour des documents dans l'index

La classe indexWRITER expose plusieurs champs qui contrôlent la manière dont les index sont mis en mémoire tampon et écrits sur le disque.

6 - la recherche, C'est examiner l'index pour y trouver des mots puis obtenir les documents qui contiennent ces mots.

La classe searcher est une classe abstraite qui possède différentes méthodes de recherche.

7-La classe query est la classe abstraite pour les requêtes

Pour chercher un mots ou une expression se fait en l'encapsulant dans un terme, en les ajoutant a un objet requête et en passant cette requête a la méthode de recherche de l'index searcher.

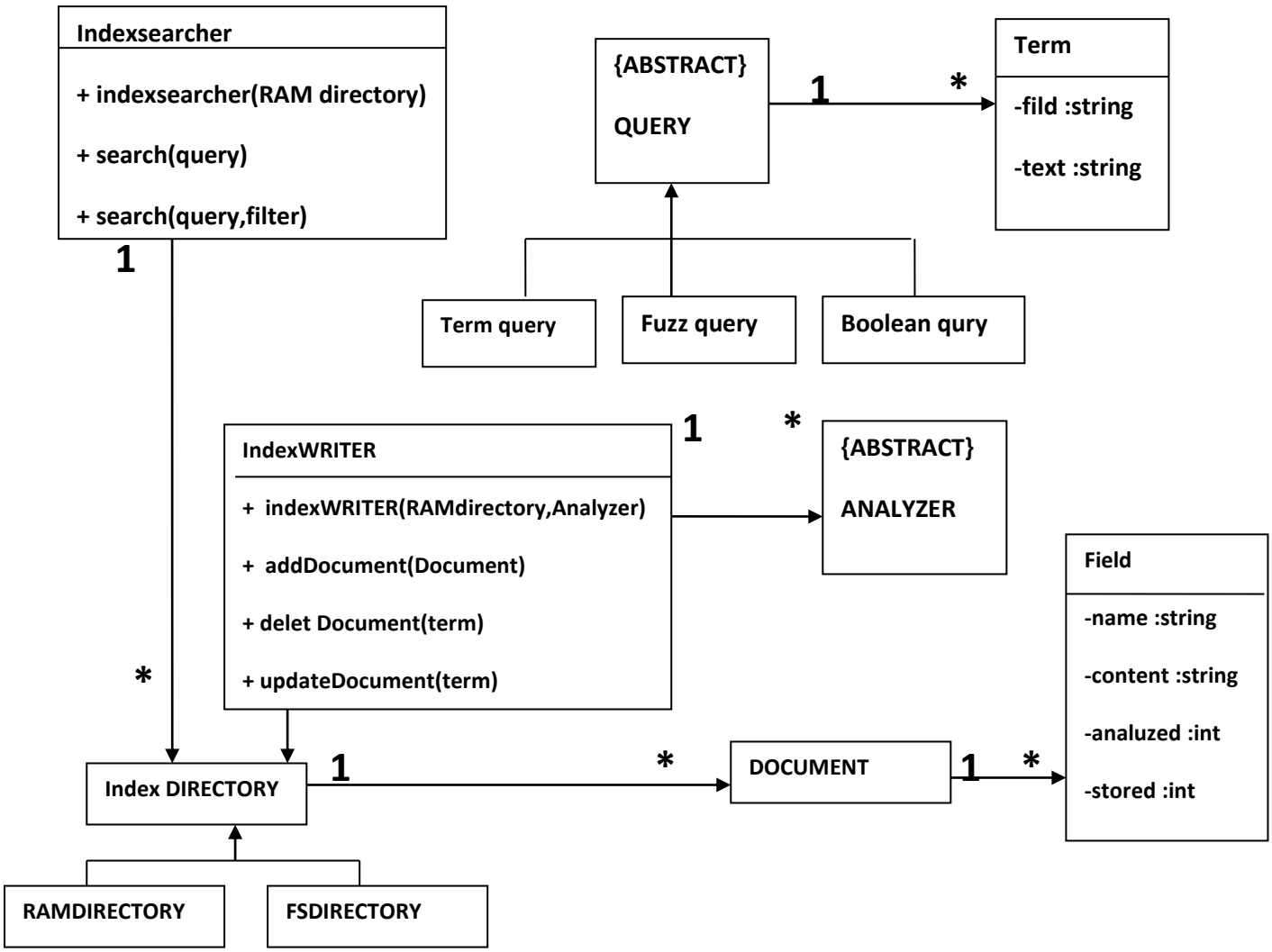


Fig.10 : diagramme de classe

2.5) Diagramme de déploiement :

Le diagramme de déploiement représente l'agencement physique d'un système montrant sur quel composant matériel les différends composantes logiciels s'exécutent.

Les principaux éléments sont des nœuds connectes par des voies de communication.

Un nœud : représentera une unité qui va héberger un logiciel, un équipement matériels ou ordimateur, un environnement d'exécution.

Les nœuds contiennent des artefacts (implémentation d'un composants) tels que les fichiers(.Exe, .dll, scripts).

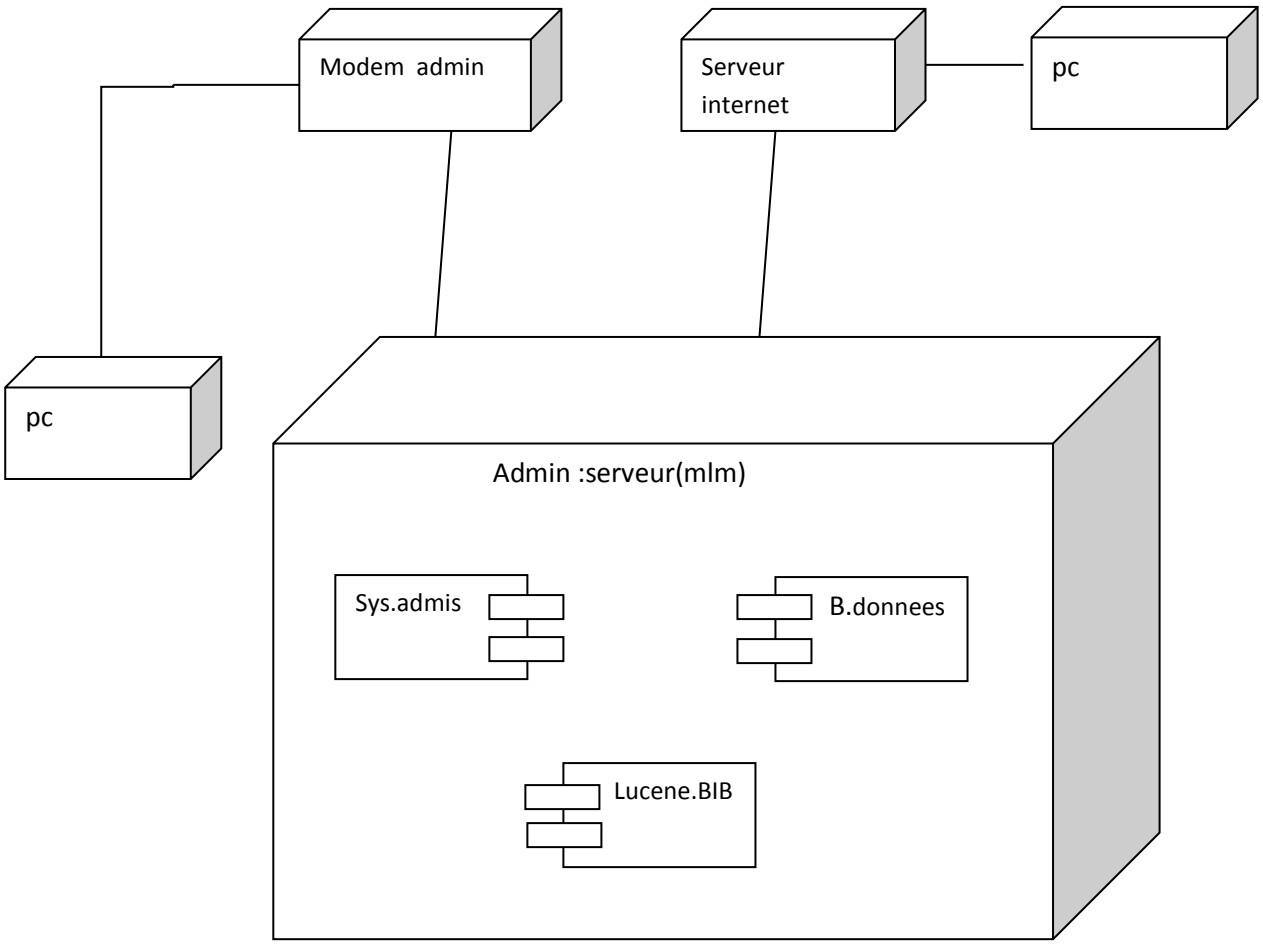


Fig. 11 : Diagramme de déploiement

CHAPITRE 4

Ce chapitre couvre la création et la mise en œuvre des différents programmes, interfaces et basse de données, qui servent à la constitution de notre application et de ses fonctionnalités. Nous présenterons quelques interfaces résultantes.

1) Environnement de travail

Au niveau de cette dernière partie, nous allons énumérer les outils soft et hard que nous avons utilises pour réaliser notre moteur de recherche ainsi que ses principales interfaces.

- **Hôte :**
- **Microprocesseur : Intel /4**
- **RAM :4GO**

2) Logiciels

Microsoft Visual Studio :

Microsoft Visual Studio est une suite de logiciels de développement pour Windows conçue par Microsoft. La dernière version s'appelle Visual Studio 2015.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages[16] .

3) Langage de programmation C# :

Le C# (*C sharp* [si:.ʃɑ:p] en anglais) est un langage de programmation orienté objet, commercialisé par Microsoft depuis 2002 et destiné à développer sur la plateforme Microsoft .NET.

Il est dérivé du C++ et très proche du Java dont il reprend la syntaxe générale ainsi que les concepts, y ajoutant des notions telles que la surcharge des opérateurs, les indexeurs et les délégués. Il est utilisé notamment pour développer des applications web sur la plateforme ASP.NET[16].

4) Technique utilisé

- **ASP.NET**

ASP.NET est une technique pour générer à la demande des pages web, lancée par Microsoft en juillet 2000[2], et utilisée pour mettre en œuvre des applications web[3]. Il s'agit d'une évolution majeure d'Active Server Pages (abr. ASP), par laquelle cette technique a été incorporée dans la plateforme Microsoft .NET[4].

Le moteur d'ASP.NET est un filtre branché sur le service web Internet Information Services (abr. IIS). Il est distribué avec le Framework .NET. ASP.NET peut être utilisé avec n'importe quel langage de programmation pour la plateforme .NET - Visual Basic .NET, C#, JScript[4][16].

- **Microsoft .NET**

Microsoft .NET (prononcé « dot net ») est le nom donné à un ensemble de produits et de technologies informatiques de l'entreprise Microsoft pour rendre des applications facilement portables sur Internet. Le but est de fournir un serveur web local permettant de gérer des services et évitant d'externaliser des données privées sur un service web de stockage ou un hébergement tiers.

La plate-forme .NET se base sur plusieurs technologies :

-les systèmes d'exploitation propriétaires Microsoft Windows
-des protocoles de communication basés sur le Framework .NET et non plus sur les modèles COM ou OLE

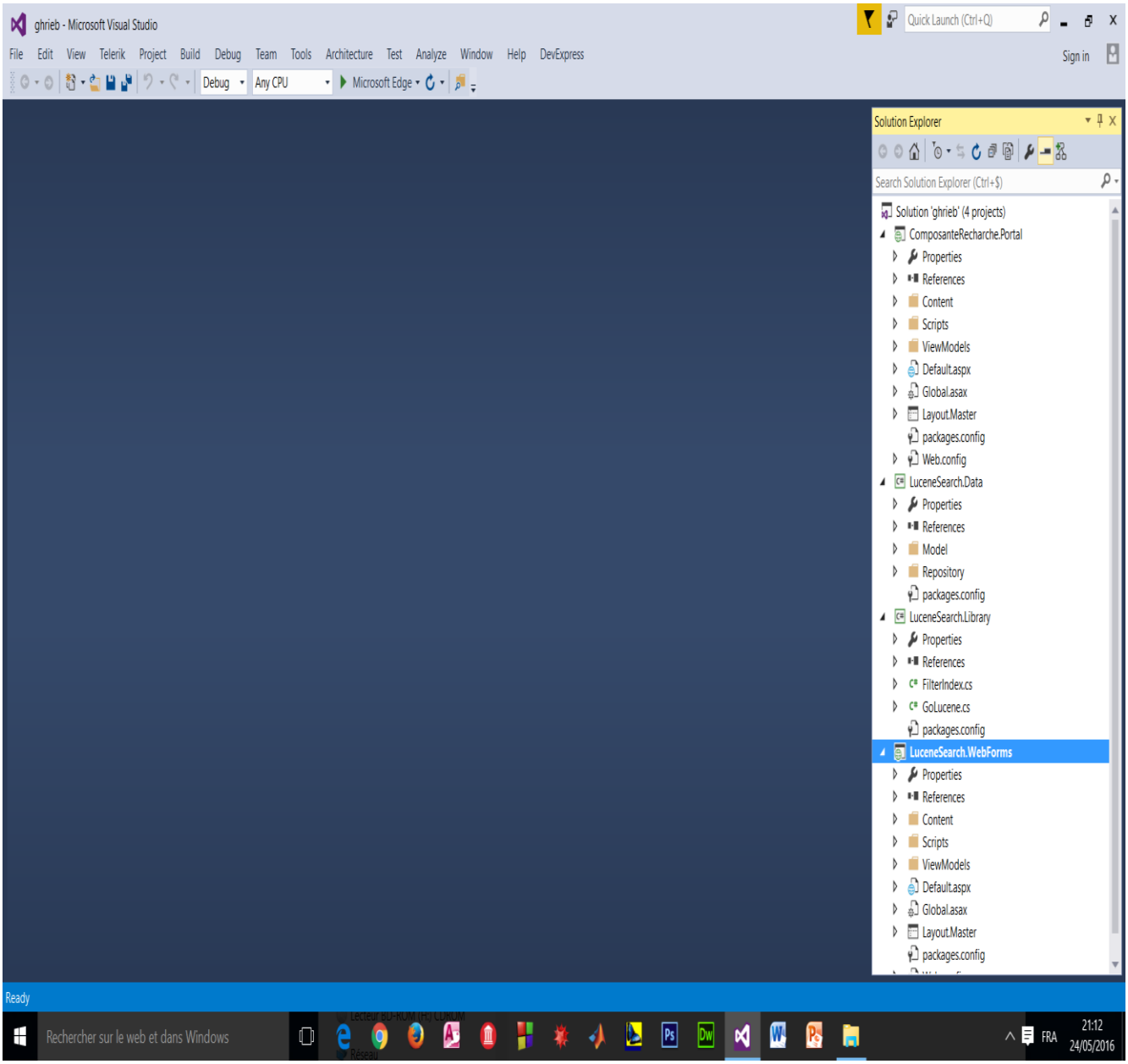


Fig. 12 : Aperçu de la page solution explorer

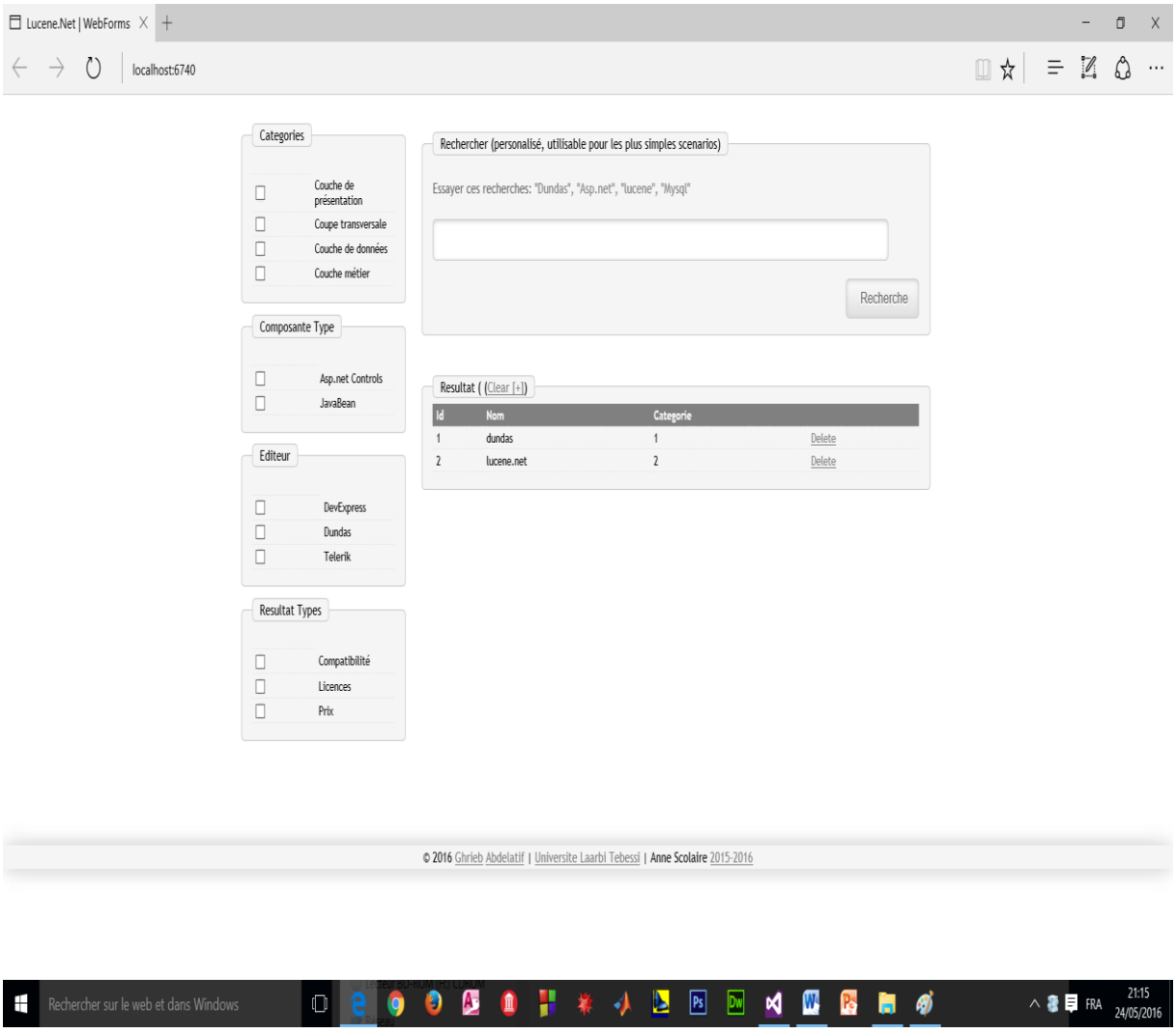


Fig. 13 : Aperçu de la page recherche utilisateur

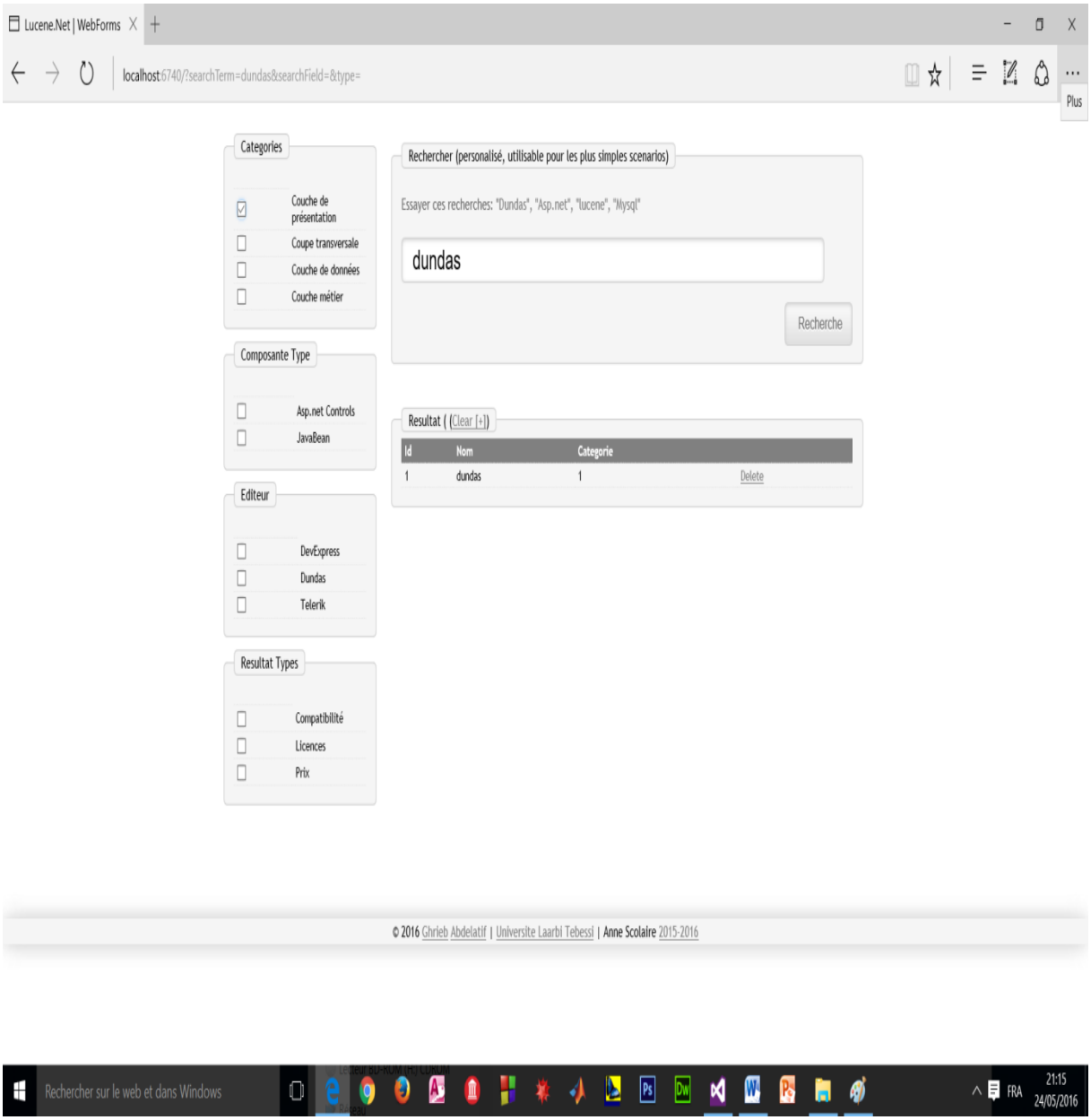


Fig. 14 : Aperçu de la page recherche utilisateur par catégorie

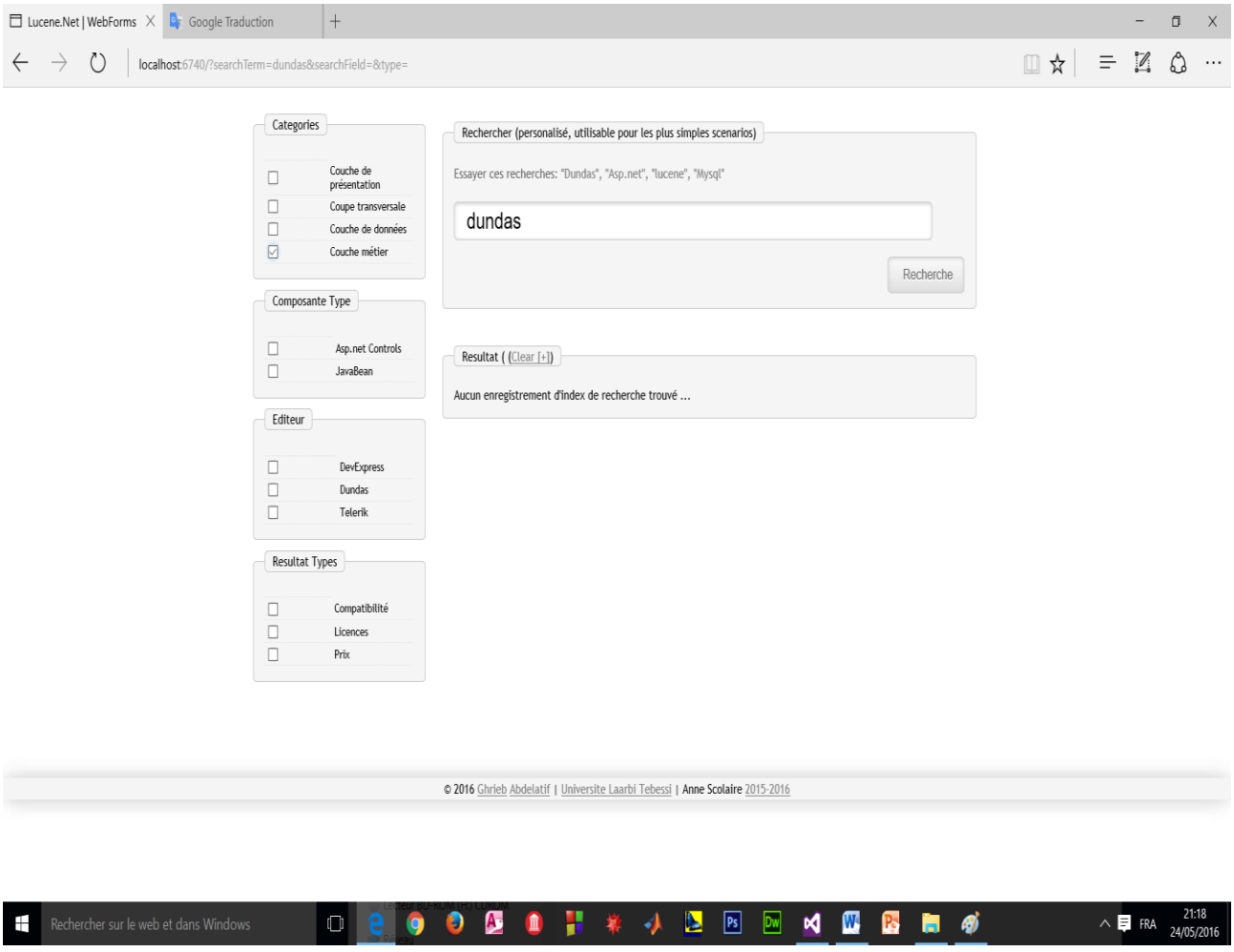


Fig. 15 : Aperçu de la page recherche utilisateur par catégorie couche métier (aucune résulta).

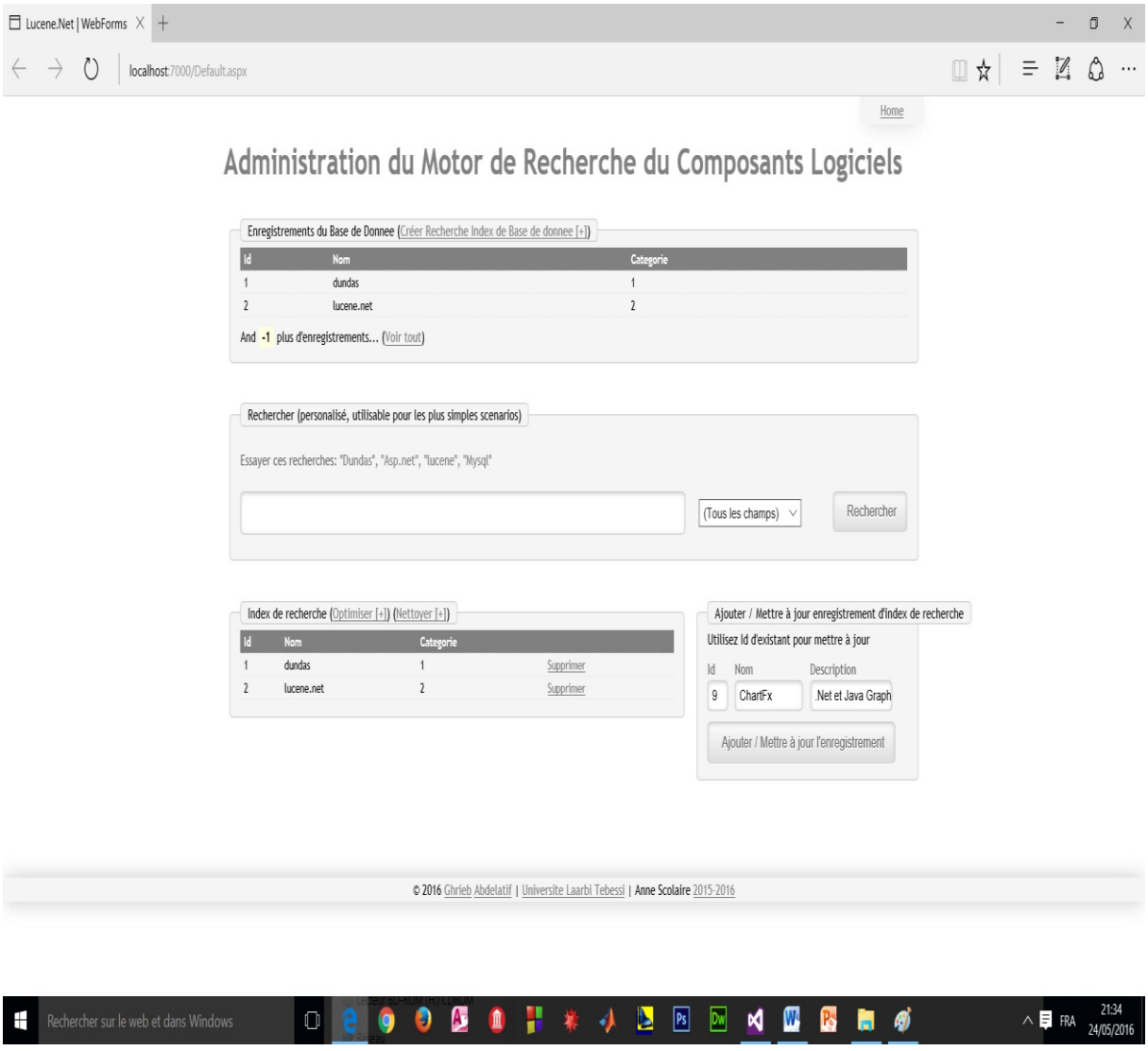


Fig. 16 : Aperçu de la page administrateur

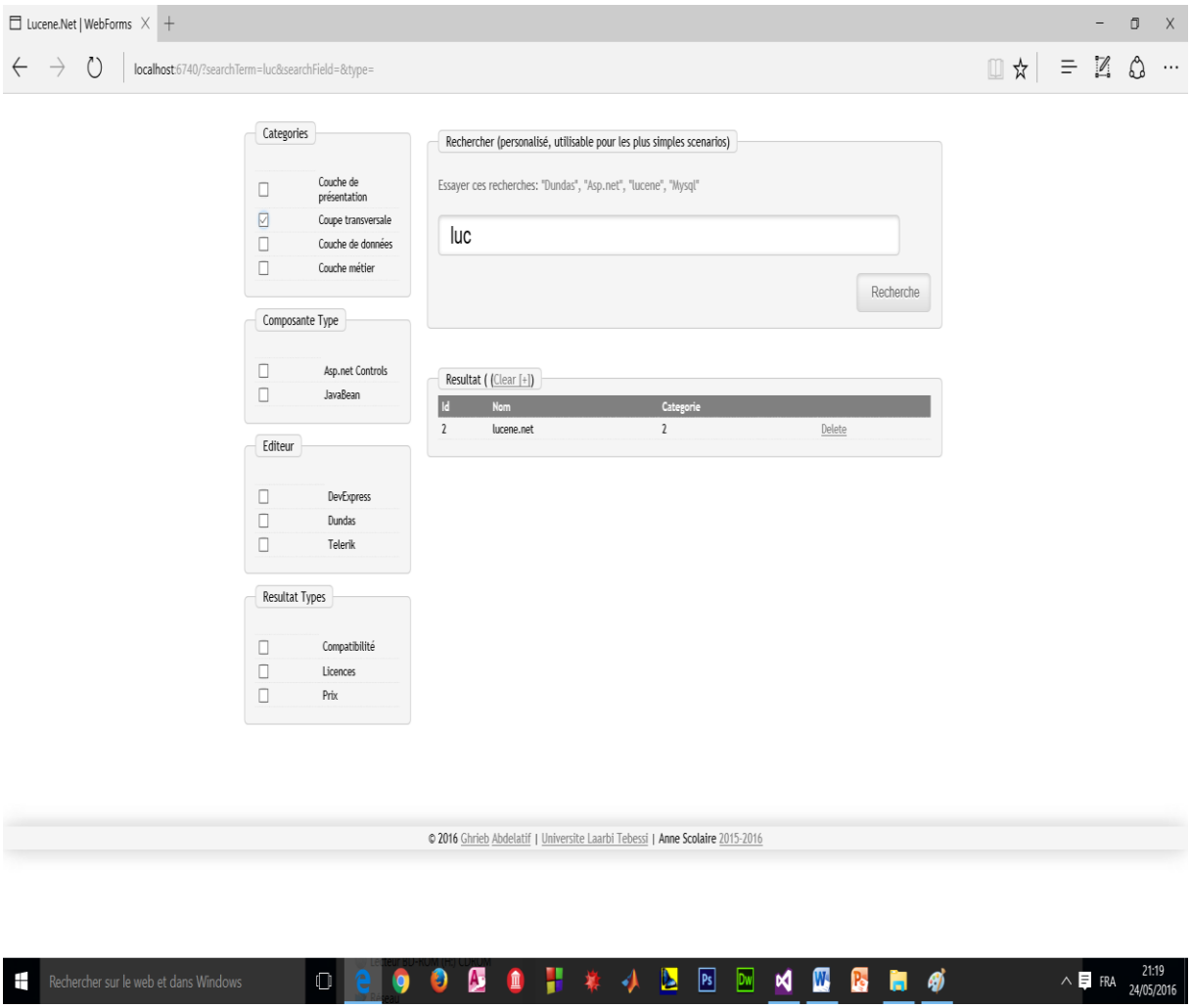


Fig. 17 : Aperçu de la page recherche utilisateur par terme (token).

Bibliographie

Référencé par un livre, these, Sites web :

[1] : https://lucene.apache.org/.../4.../lucene/document/Document.html

[2] : https: // www . Component Source .co.jp

[3] :http://WWW.berti-editions.com

[4] : ff.tu-sofia.bg/~bogi/France/SoftEng/11Developpement_par_composants.ppt

[5] : books.google.com> Informatique> Programmation> Orienté Objet

[6] :YES6274/THESE

[7] : J.Dong, P.S.C. Allencar, D.D. Cowan; "A component spécification Template for COTs based software développent"; 1st Workshop on Ensuring Successful COTS Development,USA, 1999

[8] : W. Mezeghrane ;"La réutilisation des composants logiciels : Approche de caractérisation", Mémoire de Magister, Faculté Electronique et Informatique, USTHB, Alger, Mars 2010.

[9] : Sofiane BATATA Ecole Nationale Supérieure d'Informatique (Ex. INI)

[10] : http://www.quentinfily.fr/cosinus-de-salton/#sautarticle

[11] : https://fr.wikipedia.org/wiki/Modèle_vectorie

[12] : www.torrefacteurjava.fr/.../utiliser-apache-lucene-pour-effectuer-des-recherches-textu

[13] : http://www.lucenetutorial.com/advanced-topics/scoring.html

[14] :mémoire licences département math informatique université Tébessa

[15] :livre pratique des systèmes d'information(UML)

[16] : https://www.microsoft.com/france/visual-studio/evenements/visual-studio2015.aspx

Listes des figures

Figure N	Titre	Page
Fig.1	Diagramme de déploiement	9
Fig.2	Caractérisation des composants par facettes	9
Fig.3	Caractéristiques et sous-caractéristiques « qualité » retenues	15
fig.4	Diagramme d'indexation	18
Fig.6	Diagramme de cas d'utilisation utilisateur	25
Fig.7	Diagramme de cas d'utilisation d'administrateur	26
Fig.8	Diagramme de séquence recherche d'un composant	28
Fig.9	Diagramme de séquences créations d'un index	29
Fig.10	Diagramme de classe	32
Fig.11	Diagramme de déploiement	33
Fig. 12	Aperçu de la page solution explorer	37
Fig. 13	Aperçu de la page recherche utilisateur	38
Fig. 14	Aperçu de la page recherche utilisateur par catégorie	39
Fig. 15	Aperçu de la page recherche utilisateur par catégorie couche métier (aucune résulta).	40
Fig. 16	Aperçu de la page administrateur	41
Fig. 17	Aperçu de la page recherche utilisateur par terme (token).	42