



Deuxième année de licence Informatique

Programmation orientée objet

Rapport de projet

Thème:

Générateur de fichier "sitemap.xml" et moteur de recherche

Réalisé par:

HACHOUD Rassem, HADJADJ Aboubaker



2017 / 2018

SOMMAIRE

✚ Introduction	3
✚ Présentation du projet	3
✚ Répartition des taches.....	3
✚ Présentation des classes et leur fonctionnement.....	4
✚ Présentation de l'interface graphique.....	6
✚ Niveau d'avancement	10
✚ Conclusion.....	10

1 - Introduction

Les projets sont les meilleures occasions aux étudiants d'informatique pour mettre en pratique les compétences et le savoir-faire acquis. Ce rapport est réservé pour vous rendre compte des différentes étapes pour réaliser notre première application informatique, c'est le résultat d'un travail de près d'un mois. En outre, je veux adresser tous mes remerciements à notre professeur de programmation orientée objet Monsieur Marc Lemaire qui nous a encadré tout au long de semestre 3. Je tiens également à remercier mon collègue pour la qualité de son travail et son bonne humeur à toute épreuve.

2 - Présentation de projet

Ce projet du module POO - Java permet de mettre en œuvre les principaux éléments du contenu du module dans le cadre de la conception d'une petite application, qui génère un fichier « sitemap.xml » représentant le plan d'un site web pour faciliter son référencement, et permet à l'utilisateur d'effectuer une recherche par mot-clef d'une page de jeux de données fournit.

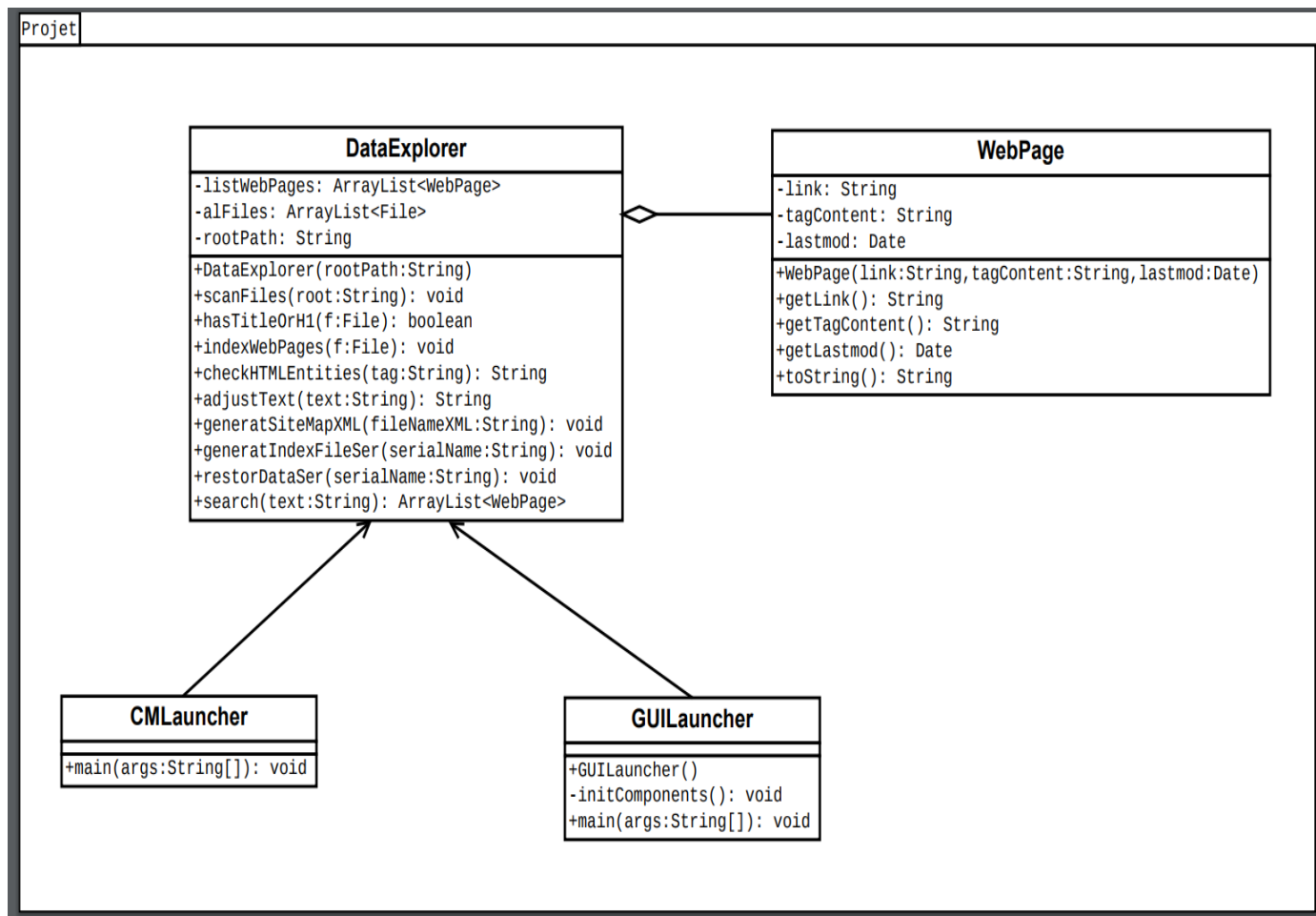
3 - Répartition des tâches

Taches communes	HACHOUD	HADJADJ
# Modélisation UML # Codage de mode console	# Réalisation de l'interface graphique	# Réalisation de rapport

Horaires et lieux de travail:

Avant les vacances de Noël, nous avons travaillé à la bibliothèque universitaire, principalement à la place des CM d'ALGO et POO que nous avons terminés, et durant les vacances, on se rencontre chaque samedi et dimanche à la bibliothèque de George Pompidou.

4 - Présentation des classes et leur fonctionnement:



- La classe << WebPage>>:

Elle permet de représenter les éléments principaux d'un fichier html, qui sont nécessaires pour réaliser l'application : lien de la page, contenu des balises <title> et <h1> et la date de dernière mise à jour de la page.

Elle a que des accesseurs sur les différents attributs et une méthode toString.

C'est des objets de cette classe qui seront sérialisés à la phase de l'indexation.

- La classe << DataExplorer >>:La classe << CMLauncher >> :

C'est la classe de base pour notre application. Elle contient toutes les méthodes nécessaires pour réaliser les trois phases de projet.

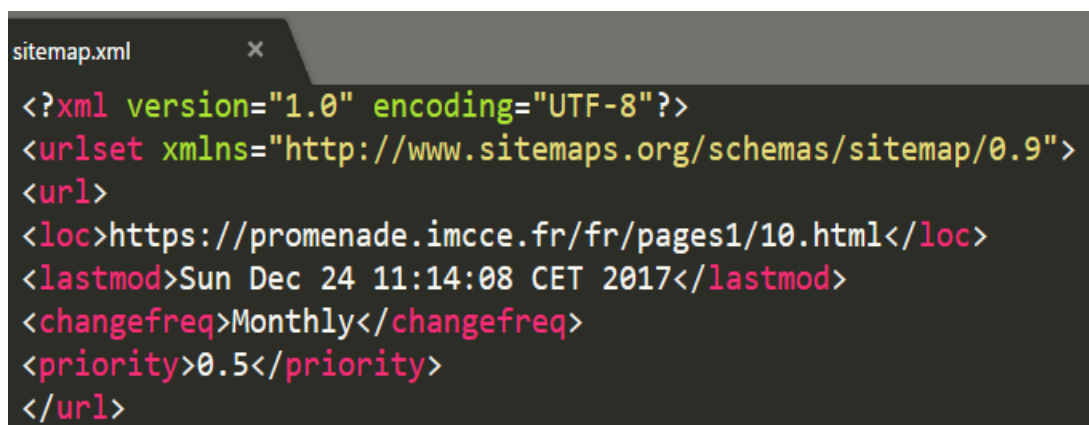
Le principe de fonctionnement est le suivant :

- D'abord on parcourt récursivement l'arborescence des fichiers et on les filtre selon 3 conditions :

- Avoir l'extension .html
- Avoir au moins une balise <title> ou <h1>
- Se situer dans un sous dossier de la racine mais pas dans la racine elle-même

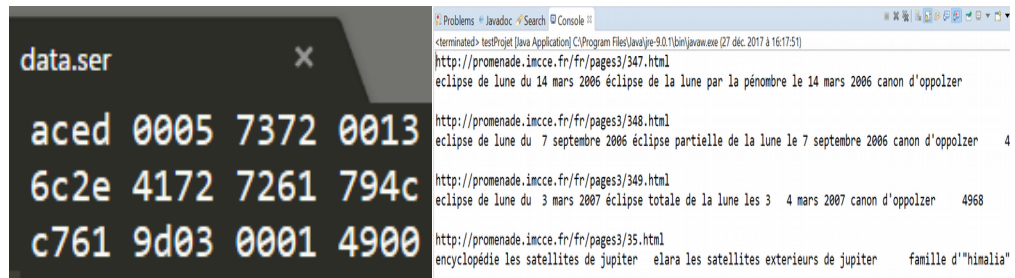
On stocke chaque fichier qui satisfait ces 3 conditions dans l'ArrayList<File> définie comme attribut de la classe.

- Générer le sitemap : il suffit de parcourir cette ArrayList<File>, et avec un BufferedWriter on écrit les informations relatives à chaque page web dans un fichier texte.



```
sitemap.xml
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
<loc>https://promenade.imcce.fr/fr/pages1/10.html</loc>
<lastmod>Sun Dec 24 11:14:08 CET 2017</lastmod>
<changefreq>Monthly</changefreq>
<priority>0.5</priority>
</url>
```

- Générer le fichier d'indexation : il suffit aussi de parcourir cette ArrayList<File>, et avec un BufferedReader on lit chaque fichier et on lui extrait le contenu de ces balises <title> et <h1>, puis on lui crée un objet de la classe WebPage qui va contenir les informations de la page. Cet objet sera stocké dans l'ArrayList<WebPage> définie comme attribut de la classe. A la fin de la lecture des fichiers, on sérialise cet ArrayList avec un ObjectOutputStream.



- Rechercher une page : On déséréalise l'ArrayList<WebPage> sauvegardée précédemment et on la parcourt. A chaque fois qu'on trouve un objet dont la variable d'instance tagContent contient la chaine de caractères saisie par l'utilisateur, on le stocke dans une liste qu'on va retourner à la fin de parcours.

- La classe << CMLauncher >> :

Elle permet de lancer l'application en mode console, par un passage d'arguments à main () par lignes de commandes:

Génération de sitemap : `java -jar exemple.jar -d ./promenade_fr.d/ -o sitemap.xml`

Génération de fichier index : `java -jar exemple.jar -d ./promenade_fr.d/ -c data.ser`

Recherche par mot-clef : `java -jar exemple.jar -s soleil`

- La classe << GUILauncher >> :

Elle permet de lancer l'application en mode graphique.

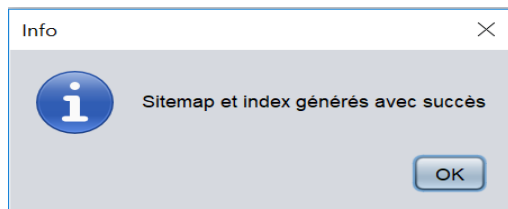
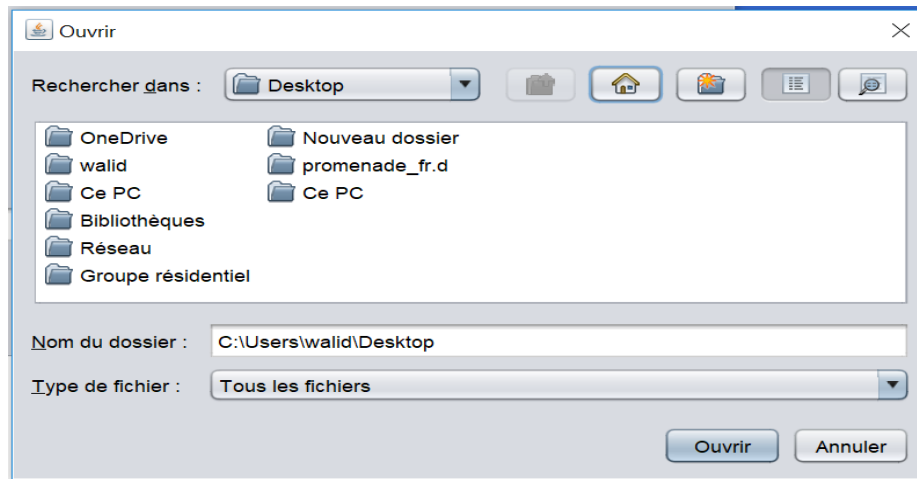
Le fonctionnement de l'interface est décrit dans le paragraphe suivant.

5 - Présentation de l'interface graphique :

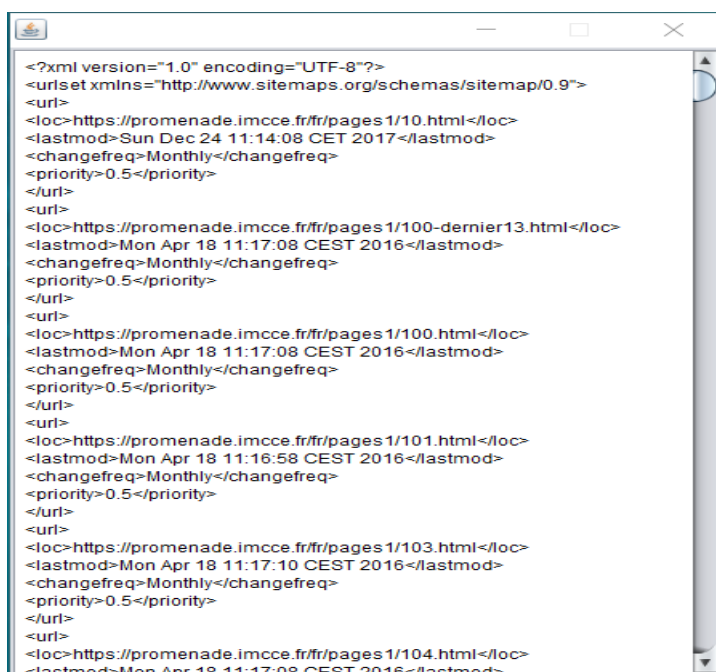
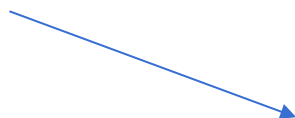
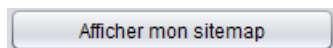
- Ouvrir l'explorateur de fichiers pour sélectionner le dossier de départ:

Cette sélection provoquera la génération du fichier sitemap.xml ainsi que l'indexation de l'ensemble des pages.

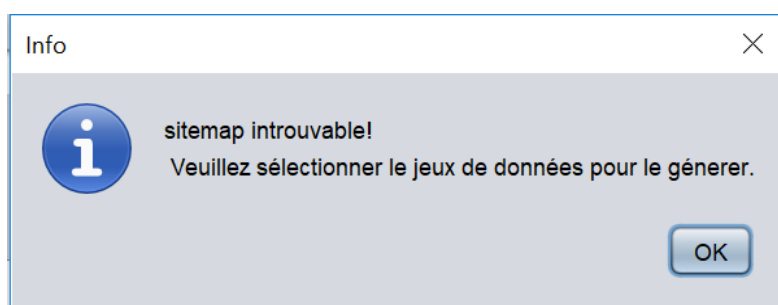
Sélectionner le jeux de données



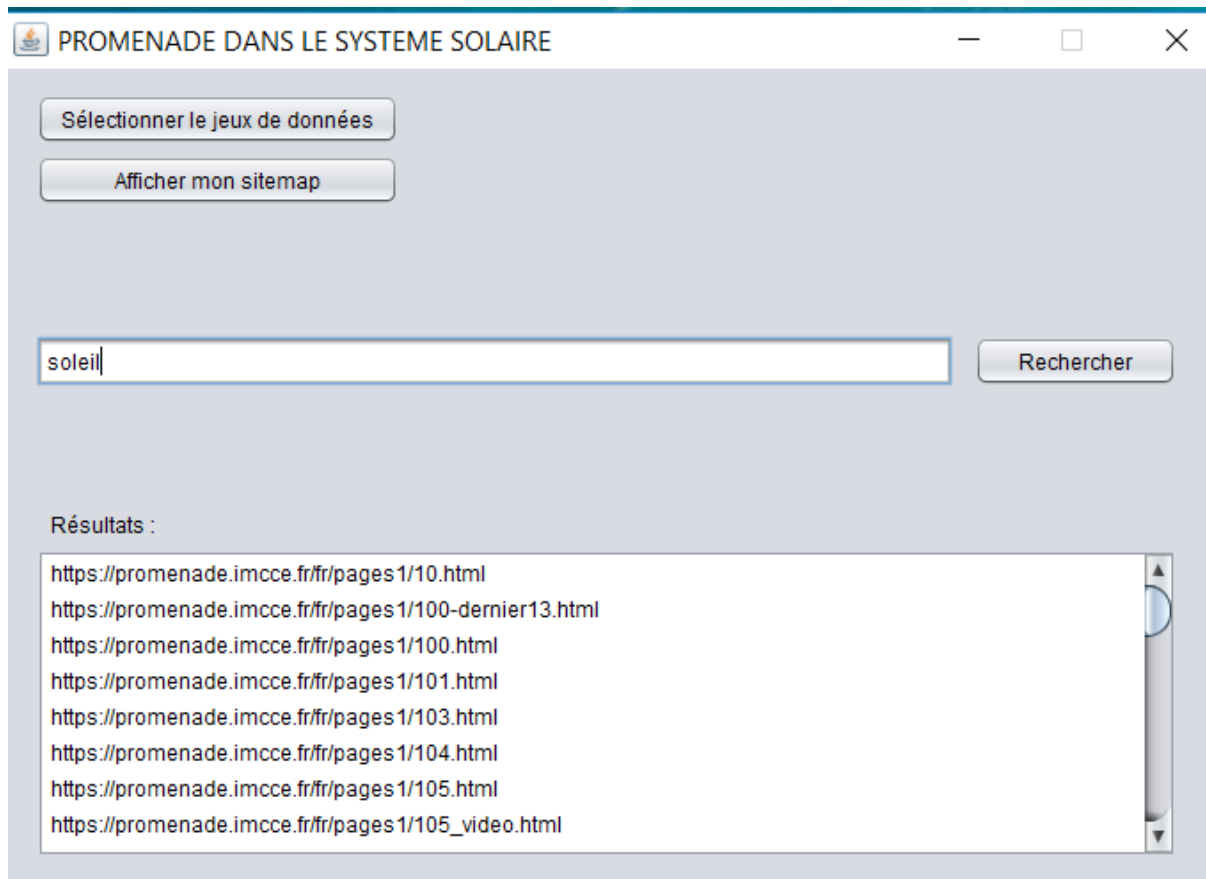
- Afficher le sitemap :



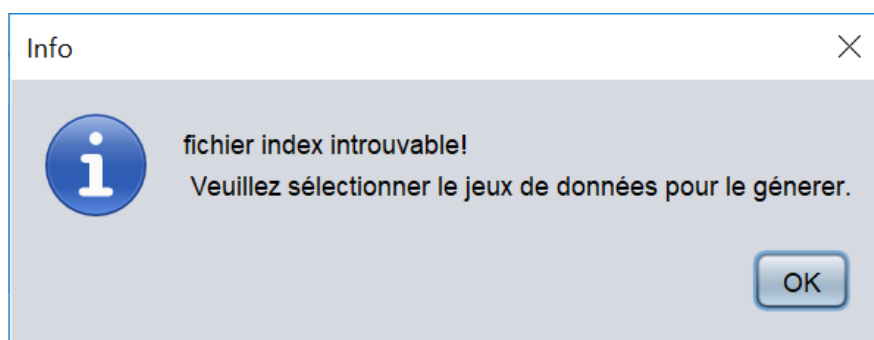
- ❖ Si le fichier sitemap n'est pas trouvé :



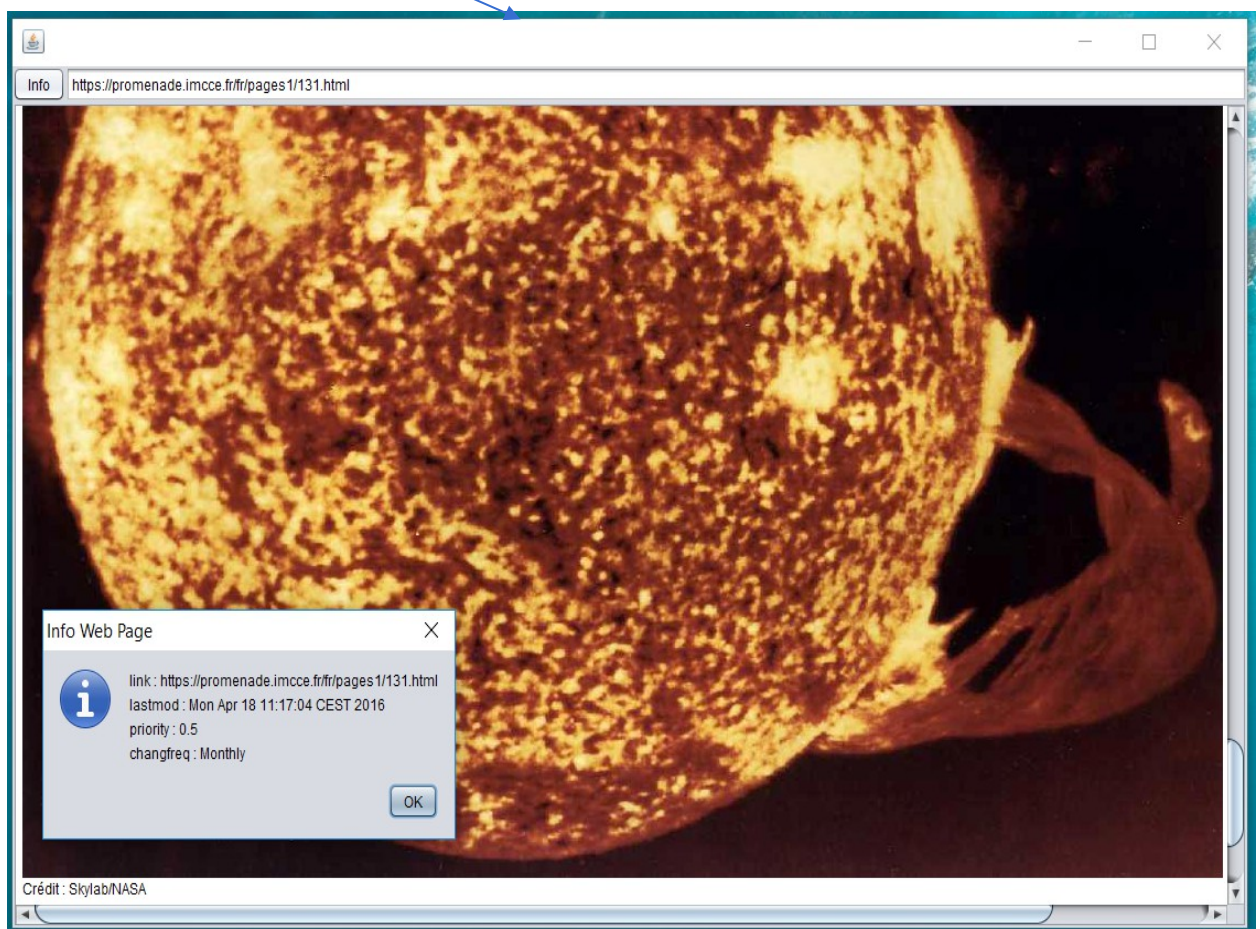
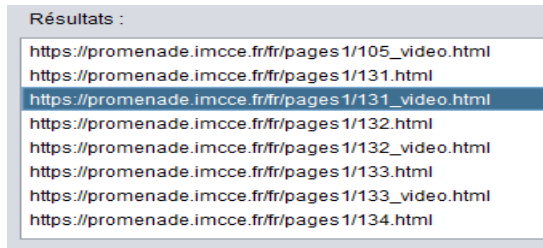
- Rechercher une page : on saisit une chaîne de caractères sur la zone de texte et on clique sur le bouton rechercher



- ❖ Si le fichier index n'est pas trouvé :



- Pour visualiser une page trouvée, il suffit de cliquer sur le lien correspondant dans la liste des résultats. On peut également afficher les principales caractéristiques de la page en cliquant sur le bouton « info »



5 - Niveau d'aboutissement et regard critique :

Nous avons réussi à atteindre la totalité des résultats attendus de ce projet, donc notre travail termine bien dans ce sens.

Points Forts	Points Faibles
<ul style="list-style-type: none">• complétude des fonctionnalités• précision des résultats• simplicité et facilité d'utilisation	<ul style="list-style-type: none">• indexation : temps d'exécution relativement long.• Visualisation des pages : le sommaire s'affiche dans toutes les pages.

6 - Conclusion :

Ce projet nous a été très intéressant puisqu'il nous a d'abord permis de mettre en œuvre les connaissances que nous avons acquises en programmation orientée objet. De plus, il nous a appris beaucoup de choses sur le fonctionnement des moteurs de recherche, puis il nous a permis d'avoir une idée sur l'HTML et XML. En outre, c'était un projet qui a demandé beaucoup d'investissement, il nous a donc demandé une bonne organisation, notamment au niveau de la répartition des tâches, et fut une expérience très enrichissante pour chacun d'entre nous. Grâce à ce projet nous avons acquis l'esprit d'un concepteur en informatique, et nous avons eu une idée plus claire sur le monde professionnel qui nous attend.