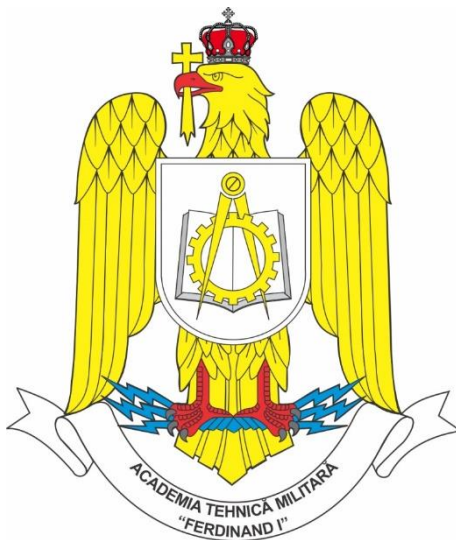


Academia Tehnică Militară



Documentul de specificare a cerințelor software

Web Crawler

Realizatori:

Ghenea Claudiu

Hariga George

Florea Vlad

Neacșu Gabriel

Panțucu Flavius

BUCUREȘTI
2020

Cuprins

1. Scopul documentului.....	2
2. Conținutul documentului.....	2
3. Descrierea generală a produsului	3
3.1. Descriere generală.....	3
3.2. Situația curentă	3
3.3. Misiunea proiectului	4
3.4. Contextul proiectului	4
3.5. Beneficii	4
4. Cerințe funcționale	6
4.1. Actori	6
4.2. Diagrama de sistem	7
4.3. Descrierea cazurilor de utilizare	8
4.4.1 Cazul Crawl Domain	8
4.4.2 Cazul Search.....	8
4.4.3 Cazul Site Map	9
5. Cerințe nefuncționale	9
5.1. Prezentare cerințe.....	9
5.2. Cerințe de interfață.....	10
5.3. Cerințe de performanță.....	10
5.4. Cerințe de fiabilitate.....	11
5.5. Cerințe de securitate.....	11
Referințe	12

1. Scopul documentului

Documentul de specificare a cerințelor software prezent furnizează o descriere completă a aplicației de tip Web Crawler pentru Tema numărul unu din cadrul cursului de Ingineria Programării al Facultății de Calculatoare și sisteme informatice pentru apărare și securitate națională, din Academia Tehnică Militară „FERDINAND I”, București.

În acest document este prezentată comportarea externă a aplicației, sunt descrise cerințele funcționale și nefuncționale, restricțiile de proiectare și alți factori necesari pentru descrierea completă a cerințelor software.

2. Conținutul documentului

Documentul conține cinci capitole, ce descriu toate cerințele de implementare ale aplicației de tip Web Crawler.

Primul capitol este o scurta prezentare a obiectivului pe care acest document și-l propune.

Capitolul al doilea este o scurtă trecere în revistă a informațiilor regăsite în document.

Capitolul al treilea este o descriere generală a produsului, cuprinde situația curentă a necesității unui proiect de tipul Web Crawler, precum și scopul acestui proiect prin îndeplinirea cerințelor de implementare exprimate de beneficiar.

Capitolul numărul patru prezintă cerințele funcționale ale proiectului. În acest capitol se ilustrează și detaliază interacțiunea cu aplicația produsă precum și modul de desfășurare al acestor interacțiuni. De asemenea sunt detaliate și diferitele cazuri de utilizare pentru o mai bună înțelegere a modului de utilizare al aplicației.

Ultimul capitol, numărul cinci enumeră cerințele nefuncționale ale aplicației, aici sunt definite minimal necesitățile proiectului pentru o funcționare optimă, precum și câteva cerințe de producție.

3. Descrierea generală a produsului

3.1. Descriere generală

Un crawler web, un „spider” sau un motor de căutare robot, descarcă și indexează conținut de pe Web. Scopul unui astfel de robot este de a extrage informații de pe fiecare pagină web, astfel încât informațiile să poată fi recuperate atunci când este necesar. Acestea sunt denumite „web crawlers”, deoarece crawling este termenul tehnic pentru accesarea automată a unui site web și obținerea de date printr-un program software.

Acești roboți sunt aproape întotdeauna operați de motoarele de căutare. Prin aplicarea unui algoritm de căutare a datelor colectate de crawler-ele web, motoarele de căutare pot furniza link-uri relevante ca răspuns la interogările de căutare ale utilizatorilor, generând lista paginilor web care apar după ce un utilizator introduce o căutare în Google sau Bing (sau într-un alt motor de căutare) .

Un robot de crawler web este precum o persoană care trece prin toate cărțile dintr-o bibliotecă dezorganizată și întocmește un catalog de rezumate, astfel încât oricine vizitează biblioteca să poată găsi rapid și ușor informațiile de care are nevoie. Pentru a ajuta la clasificarea și sortarea cărților bibliotecii în funcție de subiect, organizatorul va citi titlul, sinteza și o parte din textul intern al fiecărei cărți pentru a afla despre ce este vorba.

3.2. Situația curentă

Având în vedere dimensiunea „Internetului” chiar și cele mai mari motoare de căutare nu pot acoperi toată porțiunea publică. Astfel că, un studiu arată că în medie indexarea site-urilor publice, la motoarele de căutare mari, este doar 40-70% completă (Gulli & Signorini, 2005). Această indexare se realizează cu ajutorul a mai multor Web Crawl-eri și ajută la mărirea vitezei de căutare. Spre exemplu, Google rulează milioane de Web Crawl-eri și salvează conținutul a multor site-uri cu mult înainte ca un utilizator să îl fi căutat. Dacă nu ar face asta și ar rula atunci un Web Crawler, ar dura foarte mult.

Echipa noastră își propune să implementeze un Web Crawler în linie de comandă pentru a ajuta la indexarea site-urilor.

3.3. Misiunea proiectului

Obiectivul clasic al unui crawler este crearea unui index. Astfel, crawler-ele sunt baza pentru activitatea motoarelor de căutare. Mai întâi parcurg pe Web conținutul și apoi pun rezultatele la dispoziția utilizatorilor. Crawler-ele focalizate, de exemplu, se axează pe site-urile web actuale, relevante pentru conținut, la indexare.

3.4. Contextul proiectului

Crawler-ele web sunt utilizate și în alte scopuri:

Portalurile de comparație a prețurilor caută informații despre anumite produse de pe web, astfel încât prețurile sau datele să poată fi comparate cu precizie.

În domeniul exploatării datelor, un crawler poate colecta adrese de poștă electronică sau poștale ale companiilor.

Instrumentele de analiză web utilizează crawler-ele pentru a colecta date pentru vizualizări de pagină sau link-uri de intrare sau de ieșire.

3.5. Beneficii

Avantajul evident al unei aplicații de tip Web Crawler îl constituie automatizarea procesului de navigare Web. Site-urile Web, în special cele de actualitate, își concentrează efortul asupra îmbunătățirii experienței utilizatorului.

Nu există un standard general despre cum ar trebui să arate interfața cu utilizatorul a unui web-site. Cu siguranță, majoritatea

site-urilor își crează design-ul pe baza unor modele cunoscute și atent studiate, dar, chiar și atunci, fiecare domeniu are propriul său mod de a organiza informația.

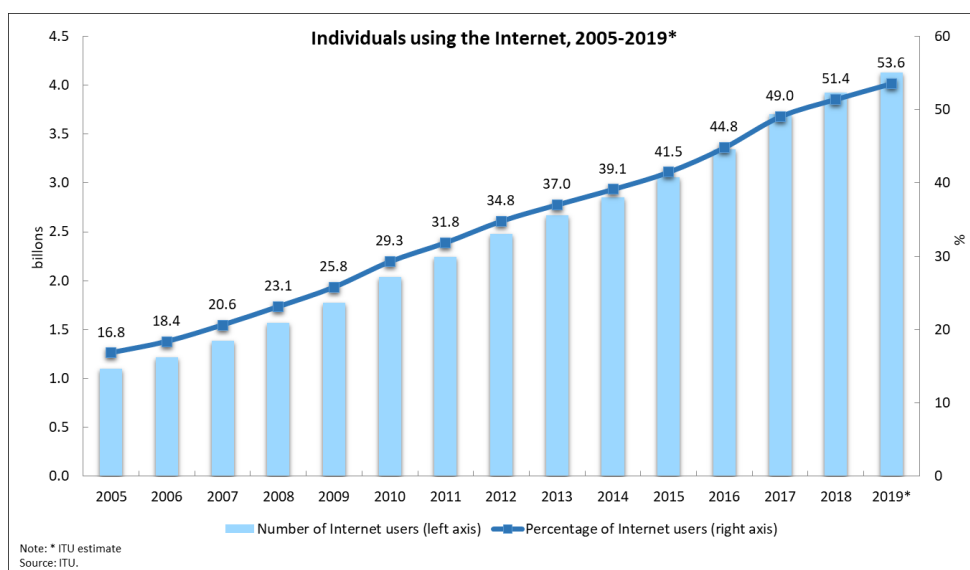
Prin tot acest proces de modelare al design-ului, adesea, îngreunăm accesul la informație. Dacă informația căutată de un utilizator se află distribuită pe mai multe pagini web, utilizatorul adesea trebuie să se confrunte cu diferite design-uri până să obțină toate informațiile necesare.

Un Web Crawler automatizează acest proces de navigare web, având scopul de a extrage doar informațiile de interes și prezentarea acestora într-o formă compactă utilizatorului, optimizând astfel procesul de căutare.

Un website este o entitate dinamică care își modifică conținutul constant prin adăugarea, ștergerea sau editarea informației conținute. Posibilitatea arhivării informației constituie un alt avantaj al unei aplicații de tip Web Crawler. Un exemplu bun îl constituie un site web care se ocupă cu vânzarea de bunuri, astfel, prin intermediul unui Web Crawler putem studia evoluția reală a fluctuației prețului unui produs vândut de acest site.

Un ultim beneficiu adus de o aplicație Web Crawler este că poate alerta utilizatorul când o informație nouă este descoperită.

Conform celor de la “ITU” (ITU, 2015) procentul de utilizatori în internet tinde să crească liniar de la an la an, astfel, este evident, că și informația din internet tinde să crească anual.



Un Crawler Web prezintă avantajul că poate să își desfășoare activitatea pasiv și să alerteze utilizatorul atunci când apare o informație nouă despre un anumit domeniu dorit.

4. Cerințe funcționale

4.1. Actori

Actorii acestei aplicații sunt reprezentați de utilizatorii și entitățile ce interacționează prin linie de comandă.

Aplicația va primi ca input un fișier de configurare și un fișier cu URL-uri pentru funcția de „crawl”, iar output-ul va fi în fișierul țintă din fișierul de configurare. În cazul în care fișierul de configurare nu este dat se va încerca căutarea unui fișier prestabilit, iar dacă nici acesta nu există se va folosi o configurare implicită. Pentru a căuta anumite fișiere dintr-un director în care există un site descărcat, se vor da ca parametrii directorul și filtrele, output-ul fiind fișierele pe care le găsește. Pentru sitemap se va da ca parametru doar fișierul țintă și ca output va afișa structura arborescentă a site-ului țintă.

Pentru fișierul de configurare, actorii vor seta:

- Filtrul pentru tipul fișierului;
- Adâncimea căutării;
- Numărul de thread-uri;
- Dimensiunea maximă;
- Directorul țintă;
- Nivelul de log;
- Delay-ul;
- Robots.txt.

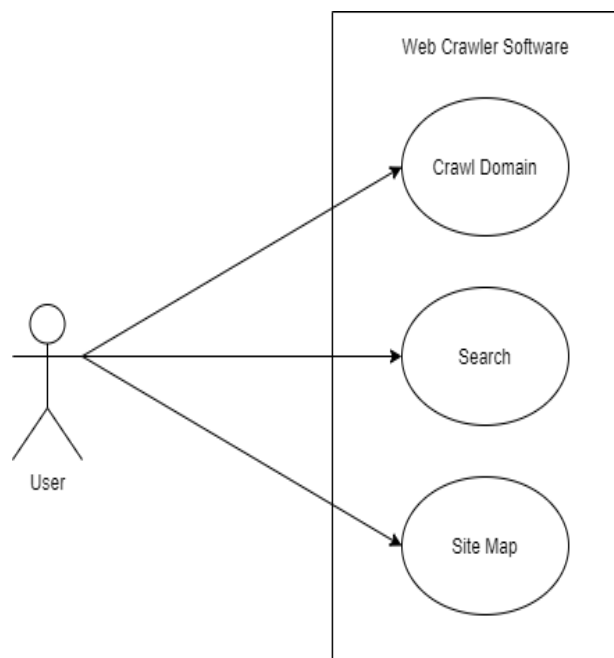
Fiecare request va fi realizat pe un thread separat pentru rapiditate. În directorul țintă, stocarea va fi realizată pe directoare în funcție de extensie.

O altă funcționalitate a utilitarului va fi de a căuta într-un director, unde s-a descărcat anterior un site, și să se poată realiza următoarele operațiuni:

- Filtrare după tip;
- Căutare după cuvinte cheie;
- Limită dimensiune;
- Creare sitemap;

În cazul în care input-ul nu este corect sau apar alte erori pe parcursul rulării se vor afișa mesaje de eroare corespunzătoare.

4.2. Diagrama de sistem



4.3. Descrierea cazurilor de utilizare

4.4.1 Cazul Crawl Domain

Descriere pe scurt

Utilizatorul poate accesa programul cu parametrul „crawl” cu o listă de site-uri și un opțional fișier de configurare pentru a descărca și indexa site-urile în cauză.

Descriere Pas-Cu-Pas

Înainte ca acest caz de utilizare să fie inițializat, utilizatorul trebuie să aibă deja construit un fișier care să conțină pe fiecare linie câte un site web țintă, opțional, utilizatorul poate construi un fișier adițional de configurare asemănător celui standard (defaultConfig.json) pentru o configurare personalizată a aplicației.

1. Utilizatorul completează fișierul cu site-uri țintă.
2. Opțional, utilizatorul completează un fișier de configurare personalizat.
3. Sistemul efectuează operația de crawl, descărcând și indexând recursiv site-urile vizate în directorul standard sau în cel specificat de utilizator.

4.4.2 Cazul Search

Descriere pe scurt

Utilizatorul dorește să caute anumite site-uri deja descărcate și procesate după anumite criterii.

Descriere Pas-Cu-Pas

Înainte ca acest caz să fie inițializat, utilizatorul trebuie să fi descărcat cel puțin o pagină web.

1. Utilizatorul alege să caute după cuvinte cheie, tipuri de fișiere, dimensiuni ale fișierelor sau după o anumită expresie regulată oferită.
2. Sistemul va afișa utilizatorului alegerea făcută.
3. Sistemul va returna utilizatorului fișierele căutate specifice cererii.

4.4.3 Cazul Site Map

Descriere pe scurt

Utilizatorul dorește să afișeze forma arborescentă a unui site deja procesat.

Descriere Pas-Cu-Pas

Înainte ca acest caz să fie inițializat, utilizatorul trebuie să fi descărcat pagina web țintă.

1. Utilizatorul rulează programul cu opțiunea site-map urmată de calea site-ului țintă și, opțional, numele unui fișier pentru afișare.
2. Sistemul va efectua operațiile necesare și va afișa fie în linie de comandă, fie în fișierul dat.

5. Cerințe nefuncționale

5.1. Prezentare cerințe

Aplicația dezvoltată trebuie să respecte următoarele cerințe non-funcționale:

1. Independența față de arhitectura mașinii de calcul și a sistemului de operare prezent, atât timp cât aceasta suportă JVM (Java virtual Machine).
2. Suport multithreading.
3. Nu necesită privilegii elevate.
4. Cost redus de resurse, respectiv, memorie și CPU.
5. Scalabilitate
6. Integrabilitate, aplicația putând fi folosită și de alte servicii software.
7. Independența față de serviciile externe.
8. Respectarea standardului de codare Java.
9. Existența unei documentații asupra modului de utilizare al aplicației.
10. Existența unei documentații asupra arhitecturii aplicației.

5.2. Cerințe de interfață

Proiectul abordat prezintă un utilitar în linie de comandă, nefiind pusă restricție asupra sistemului de operare folosit. Utilitarul primește ca input un fișier ce conține o serie de URL-uri.



Informațiile generate utilizatorului la folosirea utilitarului sunt:


1. Un director ce conține ansamblul fișierelor descărcate;
2. Un fișier cu log-uri;
3. Un fișier ce conține rezumatul activității prezentat sub formă ierarhică.

Fiind un utilitar în linie de comandă, acesta necesită o serie de parametri de configurare pentru folosirea în cadrul scenariului dorit. Astfel, se pot configura următorii parametri:

1. Numărul de thread-uri pe care utilitarul va funcționa;
2. Delay-ul acceptat între două descărcări consecutive pe thread;
3. Path-ul unde se vor descărca documentele;
4. Nivelul de accesare de la URL-ul inițial;
5. Tipul de fișiere descărcate(.txt/.docx/.pdf/.c).

Exemplu de folosire al utilitarului: `java -jar crawler crawl input.txt config.conf`

Mediul de dezvoltare: IntelliJ IDEA 

Limbajul folosit la scrierea utilitarului: JAVA 

5.3. Cerințe de performanță

Obiectivul principal al aplicației este să descarce și să proceseze cât mai multe pagini Web într-un timp cât mai scurt fără a cauza probleme de performanță site-urilor țintă sau host-ului, totodată, profitând la maxim de viteza rețelei și de resursele puse la dispoziție de utilizator (număr de thread-uri, RAM, spațiu hard-disk). Este esențial ca aplicația să se plieze cât mai bine pe hardware-ul sistemul utilizat, iar resursele acestuia să fie folosite cât mai eficient posibil, găsim un raport cât mai bun între resursele alocate și rezultatele aplicației pentru a satisface cerințele utilizatorului.

5.4. Cerințe de fiabilitate

Cerințele de fiabilitate pe care aplicația trebuie să le îndeplinească sunt:

1. Toleranța la erorile de utilizare. Aplicația trebuie să poată determina dacă site-ul ce se dorește a fi accesat este într-adevăr un website.
2. Stabilitatea din punct de vedere al cantității de date procesate.
3. Asigurarea integrității și a originii datelor obținute.

5.5. Cerințe de securitate

Pentru folosirea utilitarului nu sunt necesare modificări la nivelul securității stației de pe care este folosit. Utilitarul filtrează URL-urile accesate spre descărcare, neabordând site-urile ce nu sunt securizate HTTPS (SSL/TLS).

Referințe

Gulli, A., & Signorini, A. (2005). *The Indexable Web is More than 11.5 Billion Pages*.

ITU. (2015). *ITU*. Preluat de pe <https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx>: <https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx>