Project Report

On

WEB CRAWLER USING JAVA

Submitted for

partial fulfillment of the degree in

Bachelor of Engineering

(Computer science & Engineering)

Fifth Semester

By

Abhishek R.Parmar

Anurag R.Tiwari

Pritam R.Sah

Sambhav S.Chopda

Saurabh R.Nampalliwar

Under the Guidance of

Prof. Aarti M.Karandikar

Department of Computer Science & Engineering

Shri Ramdeobaba College Of Engineering & Management

(An Autonomous College of Rashtrasant Tukadoji Maharaj Nagpur University)

2013-2014

CERTIFICATE

This is to certify that the Project Report on

"Web Crawler Using Java "

is a bonafide work and it is submitted

for

For partial fulfillment of the degree in

Bachelor of Engineering in Computer Science & Engineering

Fifth Semester

By

Abhishek R.Parmar

Anurag R.Tiwari

Pritam R.Sah

Sambhav S.Chopda

Saurabh R.Nampalliwar

during the academic year 2013-2014

under the guidance of

Prof. Aarti M.Karandikar
Guide

Dr. M.B.Chandak
Head, Computer Science & Engg

Dr. V. S. Deshpande
Principal

Department of Computer Science & Engineering

Shri Ramdeobaba College Of Engineering & Management

(An Autonomous College of Rashtrasant Tukadoji Maharaj Nagpur University)

2013-2014

ACKNOWLEDGEMENT

We sincerely express our gratitude to our guide **Prof.Aarti M.Karandikar** for her valuable guidance and kind suggestion. We are indebted her for her untiring devotion and willingness to go with us to any length .She has shown a stimulating interest in this project and has been encouraging us every time.

We sincerely express our gratitude to **Dr.V.S.Deshpande** Principal, SRCOEM, Nagpur and **Dr.M.Chandak**, Head Dept. of CSE, SRCOEM for providing us this opportunity and support.

We are sincerely thankful to our faculty teachers and the Department of Computer Science and Engineering for their valuable suggestion, helping hand and guidance extended to us.

The acknowledgement would be incomplete without giving thanks to our friends for their help throughout the course of this project and all those related people who directly or indirectly helped us for the completion of project successfully.

Name of the Projectees

Abhishek R.Parmar
Anurag R.Tiwari
Pritam R.Sah
Sambhav S.Chopda
Saurabh R.Nampalliwar

ABSTRACT

The main purpose of this project is to present the anatomy of a large scale Hypertext Transfer Protocol (HTTP) based Web search engine by using the system architecture of large search engines such as Google, Yahoo as a prototype. Additionally, a web crawler is developed and implemented in Java jdk1.8.0, which demonstrates the operation of a typical Web crawler.

The project describes in detail the basic tasks a search engine performs. An overview of how the whole system of a search engine works is provided. A WebCrawler application is implemented using Java programming language (Advanced). The GUI of the developed application helps the user to identify various actions that can take place like specifying the start URL, maximum URLs to be crawled, the way crawling has to be done – breadth first or depth first. This paper also lists proposed functionalities as well as features not supported by the web crawler application.

List of Figures

CONTENTS

INTRODUCTION

Engineering a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens of millions of queries every day. Despite the importance of large-scale search engines on the web, very little academic research has been conducted on them. Furthermore, due to rapid advance in technology and web proliferation, creating a web
search engine today is very different from three years ago. There are differences in theways various search engines work, but they all perform three basic tasks:

1. They search the Internet or select pieces of the Internet based on important words.
2. They keep an index of the words they find, and where they find them.
3. They allow users to look for words or combinations of words found in that index.

A search engine finds information for its database by accepting listings sent in by authors who want exposure, or by getting the information from their "web crawlers," "spiders," or "robots," programs that roam the Internet storing links to and information about each page they visit. A web crawler is a program that downloads and stores Web pages, often for a Web search engine. Roughly, a crawler starts off by placing an initial set of URLs,
*S*0, in a queue, where all URLs to be retrieved are kept and prioritized. From this queue, the crawler gets a URL (in some order), downloads the page, extracts any URLs in the downloaded page, and puts the new URLs in the queue. This process is repeated until the crawler decides to stop. Collected pages are later used for other applications, such as a Web search engine or a Web cache.

The most important measure for a search engine is the search performance, quality of the results and ability to crawl, and index the web efficiently. The primary goal is to provide high quality search results over a rapidly growing World Wide Web. Some of the efficient and recommended search engines are Google, Yahoo and Teoma, which share some common features and are standardized to some extent.

CRAWL POLICIES

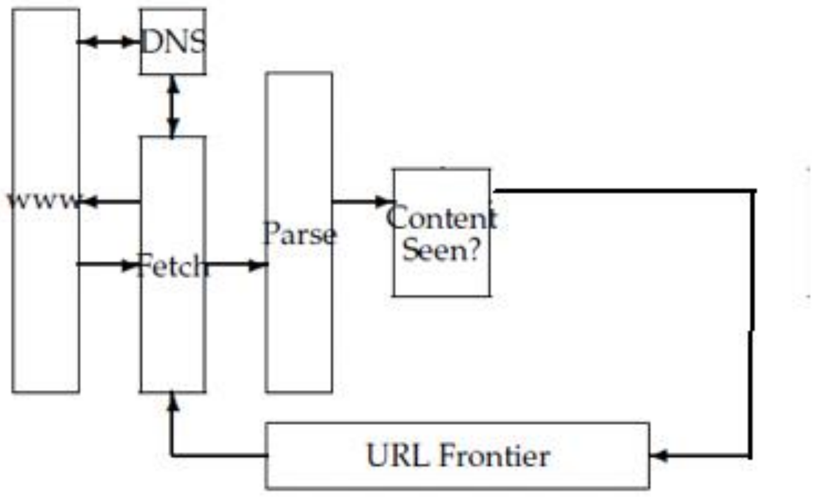
1. **Selection policy**
    a. A *selection policy* that states which pages to download.
    b. Based on Path ascending.

2. **Politeness Policy**

Costs of using web crawlers include:

a. Network resources, crawlers require bandwidth and operate with a high degree of parallelism during a long period of time;
b. Server overload, if the frequency of accesses to a given server is too high;
c. Poorly-written crawlers, can crash servers or routers, download pages they cannot handle; and
d. Personal crawlers, if deployed by too many users, can disrupt networks and web servers.

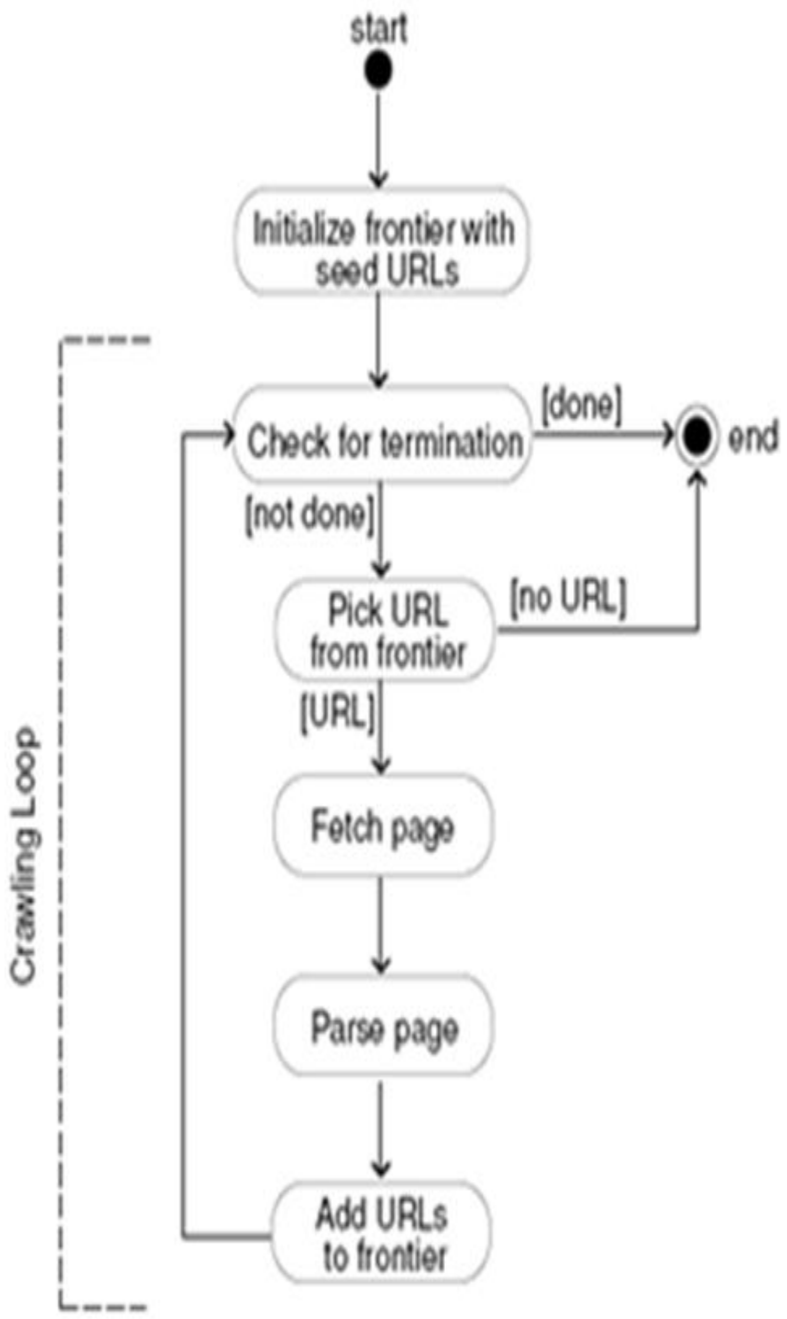
ARCHITECTURE



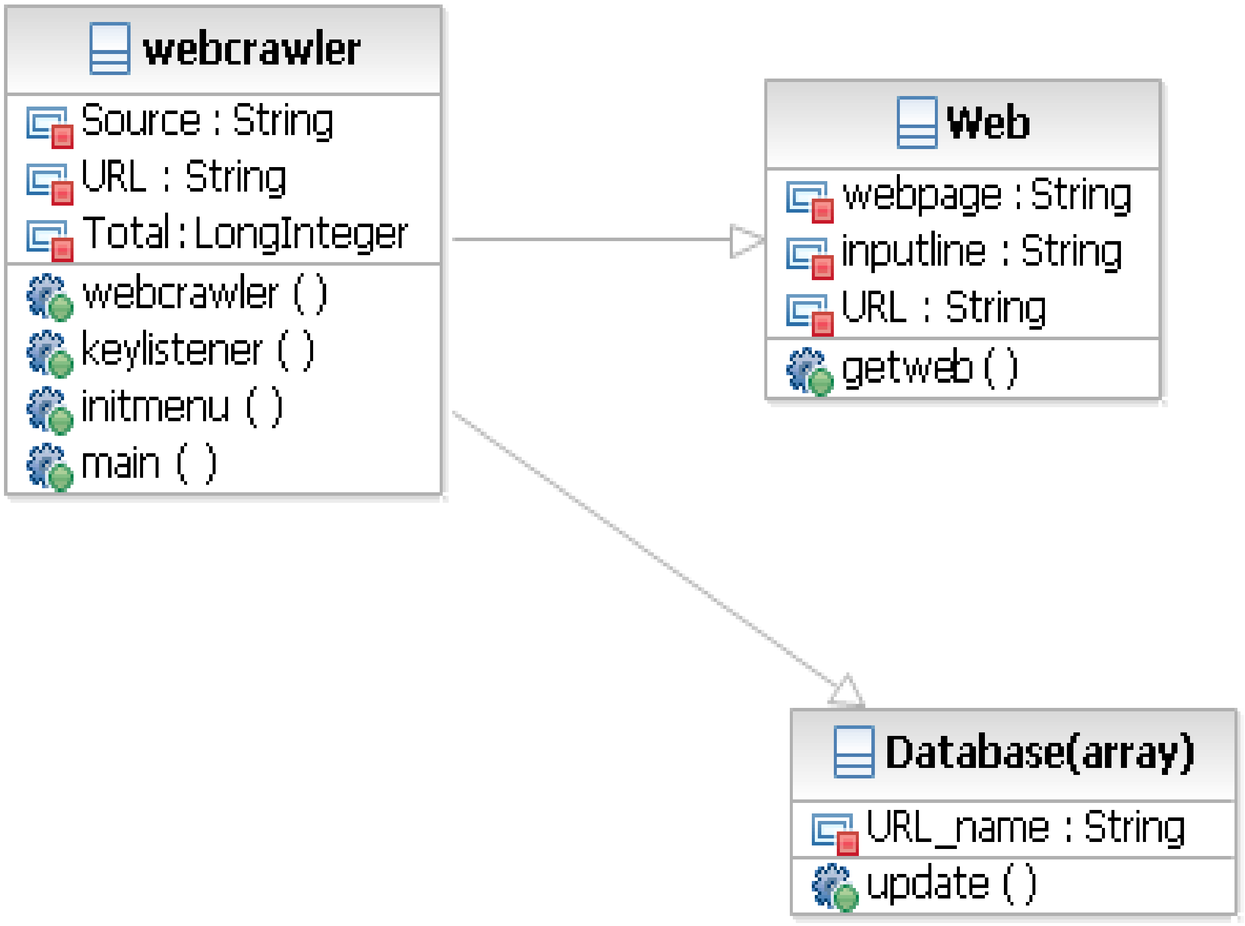
► **Figure** The basic crawler architecture.

Crawling is performed by anywhere from one to potentially hundreds of threads, each of which loops through the logical cycle. We begin by assuming that the URL frontier is in place and non-empty and defer our description of the implementation of the URL frontier to Section. We follow the progress of a single URL through the cycle of being fetched, passing through various checks and filters, then finally (for continuous crawling) being returned to the URL frontier. A crawler thread begins by taking a URL from the frontier and fetching the web page at that URL, generally using the http protocol. The fetched page is then written into a temporary store, where a number of operations are performed on it. Next, the page is parsed and the text as well as the links in it are extracted. The text (with any tag information – e.g., terms in boldface) is passed on to the indexer. Link information including anchor text is also passed on to the indexer for use in ranking in ways that are described. In addition, each extracted link goes through a series of tests to determine whether the link should be added to the URL frontier. First, the thread tests whether a web page with the same content has already been seen at another URL. Next, a *URL filter* is used to determine whether the extracted URL should be excluded from the frontier based on one of several tests. For instance, the crawl may seek to exclude certain domains (say, all .com URLs) – in this casethe test would simply filter out the URL if it were from the .com domain.

WORKING


start
Initialize frontier with seed URLs
Check for termination
[done]
end
[not done]
Pick URL from frontier
[no URL]
[URL]
Fetch page
Parse page
Add URLs to frontier
Crawling Loop

DATA STRUCTURE



SCREEN SHOTS

Front Screen:

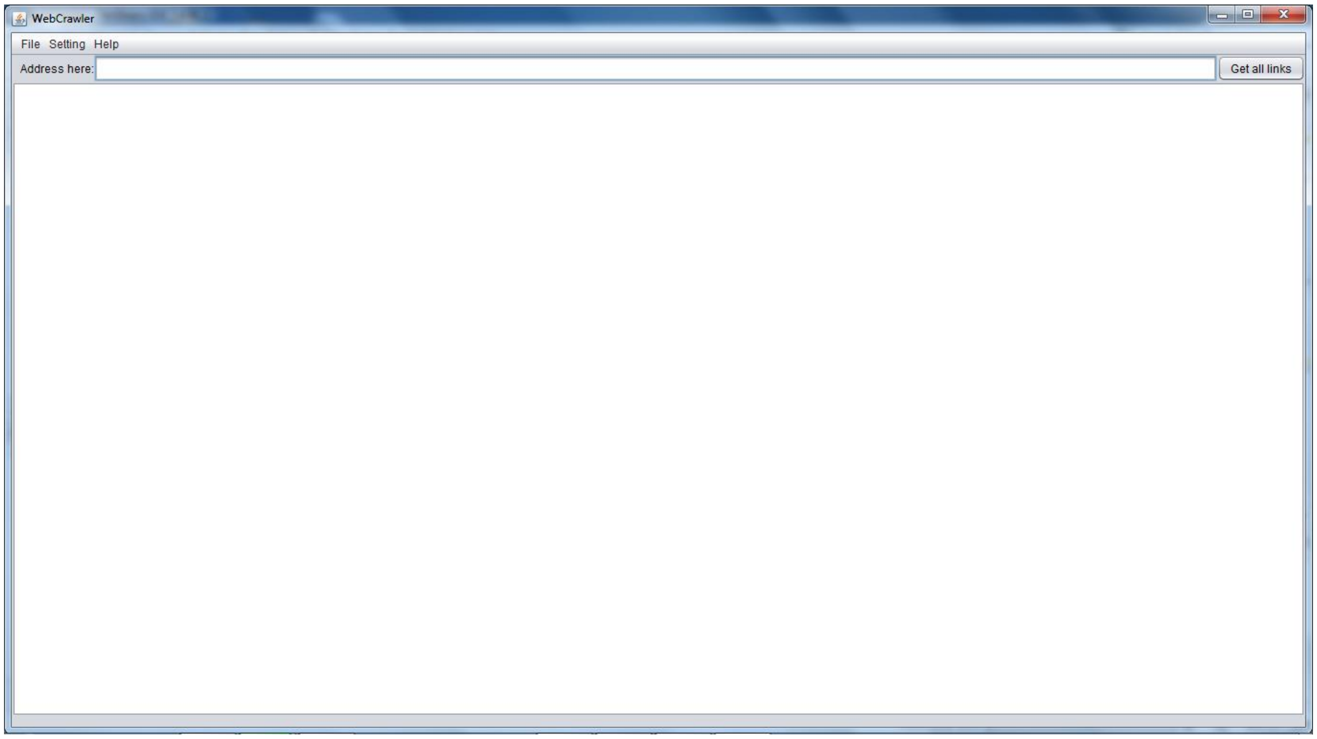
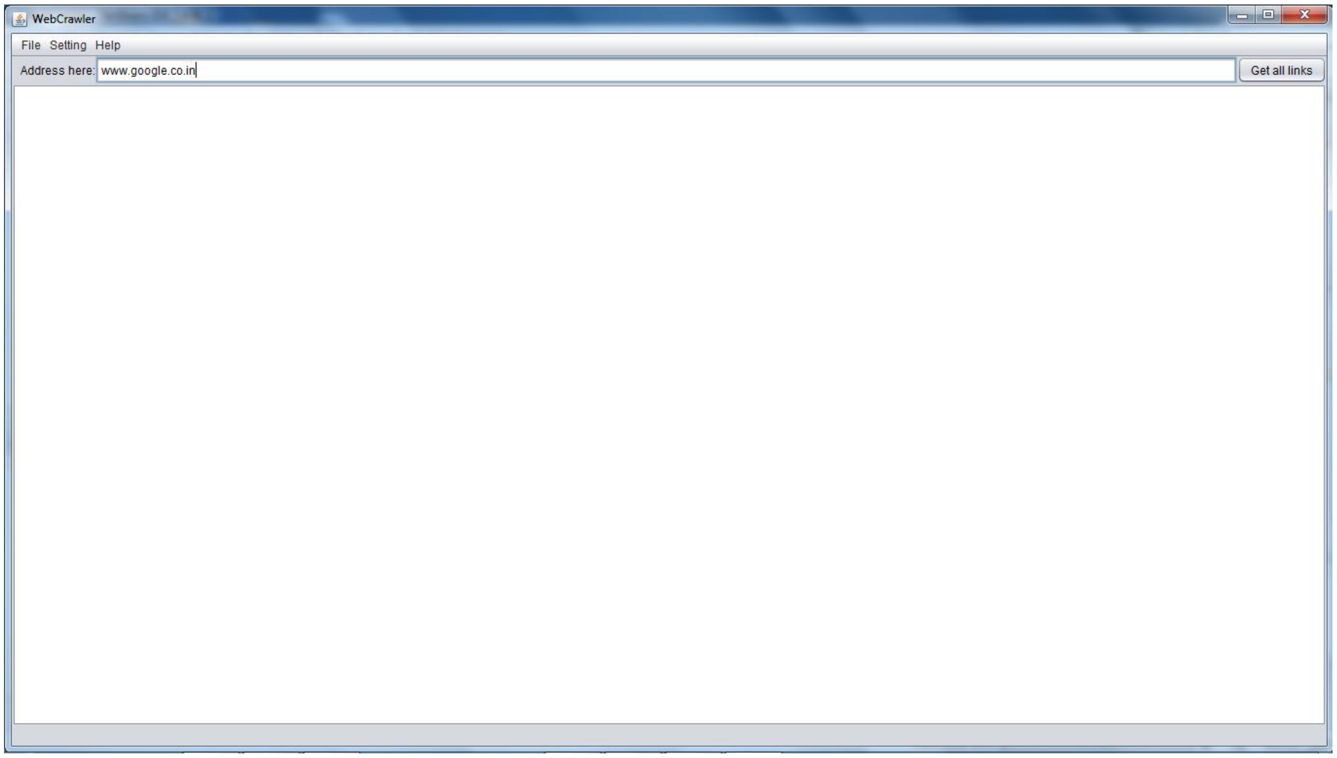


Fig 1. Front Page

This is the front page of our project. It contains the text box near Address Bar label in which the user need to enter the address of which he/she wants the links then the fetched link will be shown in the text box as shown in Fig3.

Screen with Entered Address:

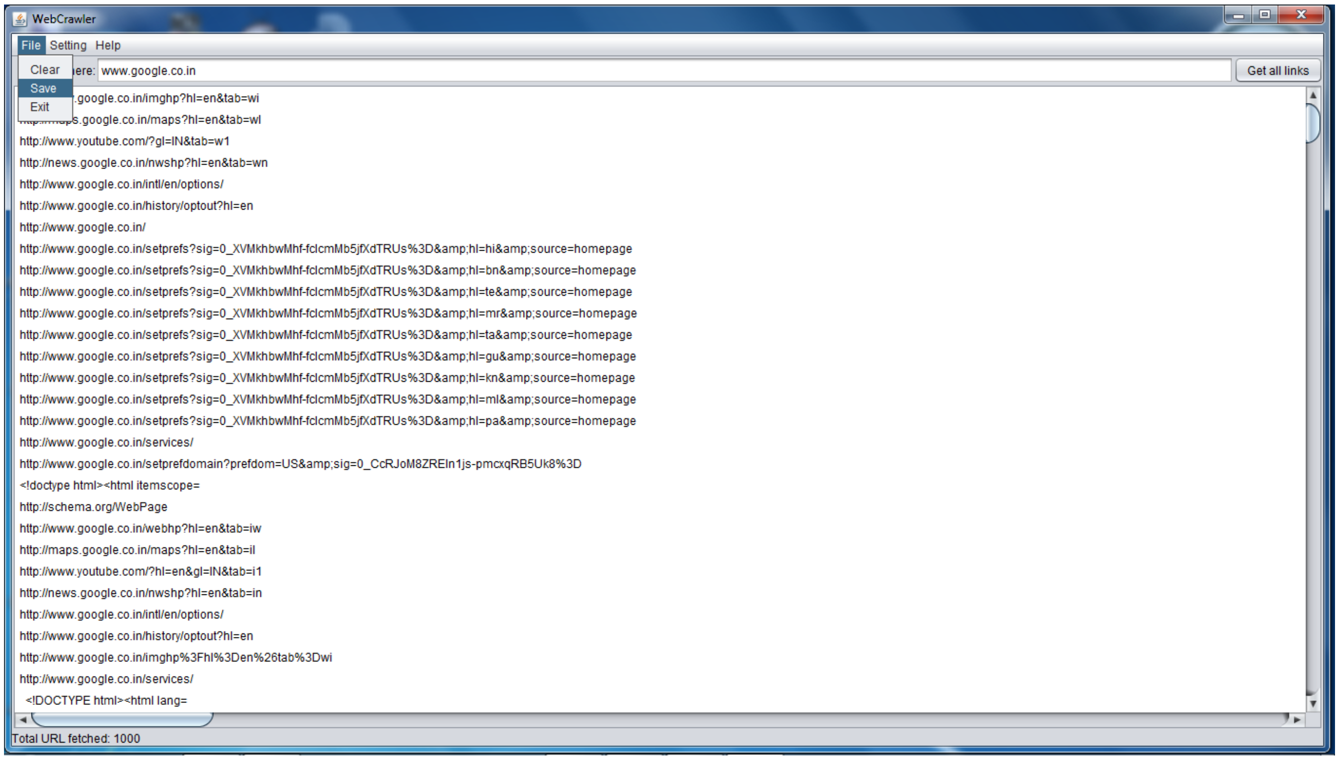


www.google.co.in

Fig2. Entering Address

The user enter the web address in text box "www.google.com". Then click on label "Get all links".

Screen with Fetched Links:

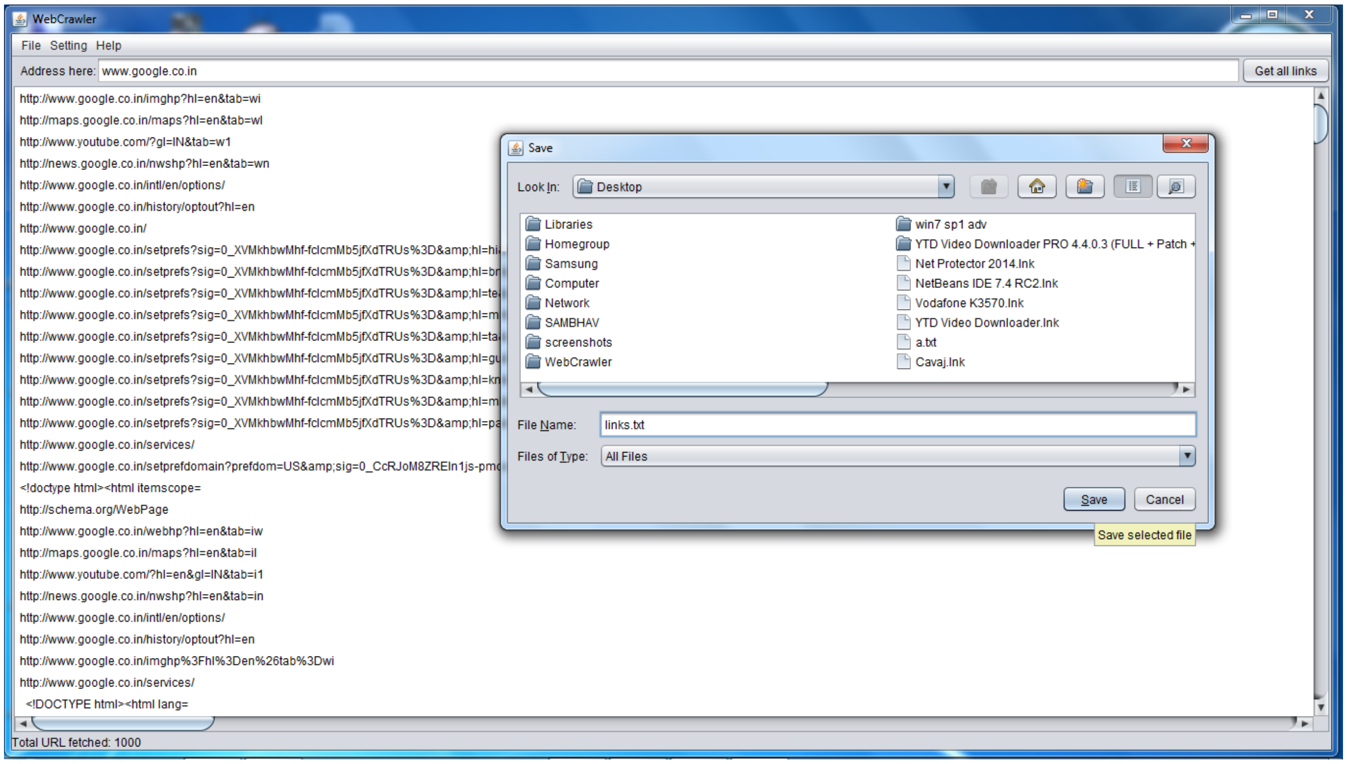


```
ere: www.google.co.in
.google.co.in/imghp?hl=en&tab=wi
s.google.co.in/maps?hl=en&tab=wl
http://www.youtube.com/?gl=IN&tab=w1
http://news.google.co.in/nwshp?hl=en&tab=wn
http://www.google.co.in/intl/en/options/
http://www.google.co.in/history/optout?hl=en
http://www.google.co.in/
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=hi&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=bn&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=te&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=mr&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=ta&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=gu&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=kn&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=ml&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=pa&amp;source=homepage
http://www.google.co.in/services/
http://www.google.co.in/setprefdomain?prefdom=US&amp;sig=0_CcRJoM8ZREIn1js-pmcxqRB5Uk8%3D
<!doctype html><html itemscope=
http://schema.org/WebPage
http://www.google.co.in/webhp?hl=en&tab=iw
http://maps.google.co.in/maps?hl=en&tab=il
http://www.youtube.com/?hl=en&gl=IN&tab=i1
http://news.google.co.in/nwshp?hl=en&tab=in
http://www.google.co.in/intl/en/options/
http://www.google.co.in/history/optout?hl=en
http://www.google.co.in/imghp%3Fhl%3Den%26tab%3Dwi
http://www.google.co.in/services/
  <!DOCTYPE html><html lang=
Total URL fetched: 1000
```

Fig3. Fetched Links

All the links of www.google.co.in will be ferched and will be shown in text box as shown in figure above.

Screen with ‘Save’ Dialog box:



Address here: www.google.co.in

```
http://www.google.co.in/imghp?hl=en&tab=wi
http://maps.google.co.in/maps?hl=en&tab=wl
http://www.youtube.com/?gl=IN&tab=w1
http://news.google.co.in/nwshp?hl=en&tab=wn
http://www.google.co.in/intl/en/options/
http://www.google.co.in/history/optout?hl=en
http://www.google.co.in/
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=hi
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=br
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=te
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=m
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=ta
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=gu
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=kn
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=m
http://www.google.co.in/setprefs?sig=0_XVMkhbwMhf-fclcmMb5jfXdTRUs%3D&amp;hl=pa
http://www.google.co.in/services/
http://www.google.co.in/setprefdomain?prefdom=US&amp;sig=0_CcRJoM8ZREln1js-pmc
<!doctype html><html itemscope=
http://schema.org/WebPage
http://www.google.co.in/webhp?hl=en&tab=iw
http://maps.google.co.in/maps?hl=en&tab=il
http://www.youtube.com/?hl=en&gl=IN&tab=i1
http://news.google.co.in/nwshp?hl=en&tab=in
http://www.google.co.in/intl/en/options/
http://www.google.co.in/history/optout?hl=en
http://www.google.co.in/imghp%3Fhl%3Den%26tab%3Dwi
http://www.google.co.in/services/
  <!DOCTYPE html><html lang=
```

Total URL fetched: 1000

Look In: Desktop

Libraries	win7 sp1 adv
Homegroup	YTD Video Downloader PRO 4.4.0.3 (FULL + Patch +
Samsung	Net Protector 2014.lnk
Computer	NetBeans IDE 7.4 RC2.lnk
Network	Vodafone K3570.lnk
SAMBHAV	YTD Video Downloader.lnk
screenshots	a.txt
WebCrawler	Cavaj.lnk

File Name: links.txt

Files of Type: All Files

Save Cancel

Save selected file

Fig4. Save the fetched links

The dialog box will open that will save all the fetched links. You can save the fetched links in text files.

<u>**Destination of saved file:**</u>

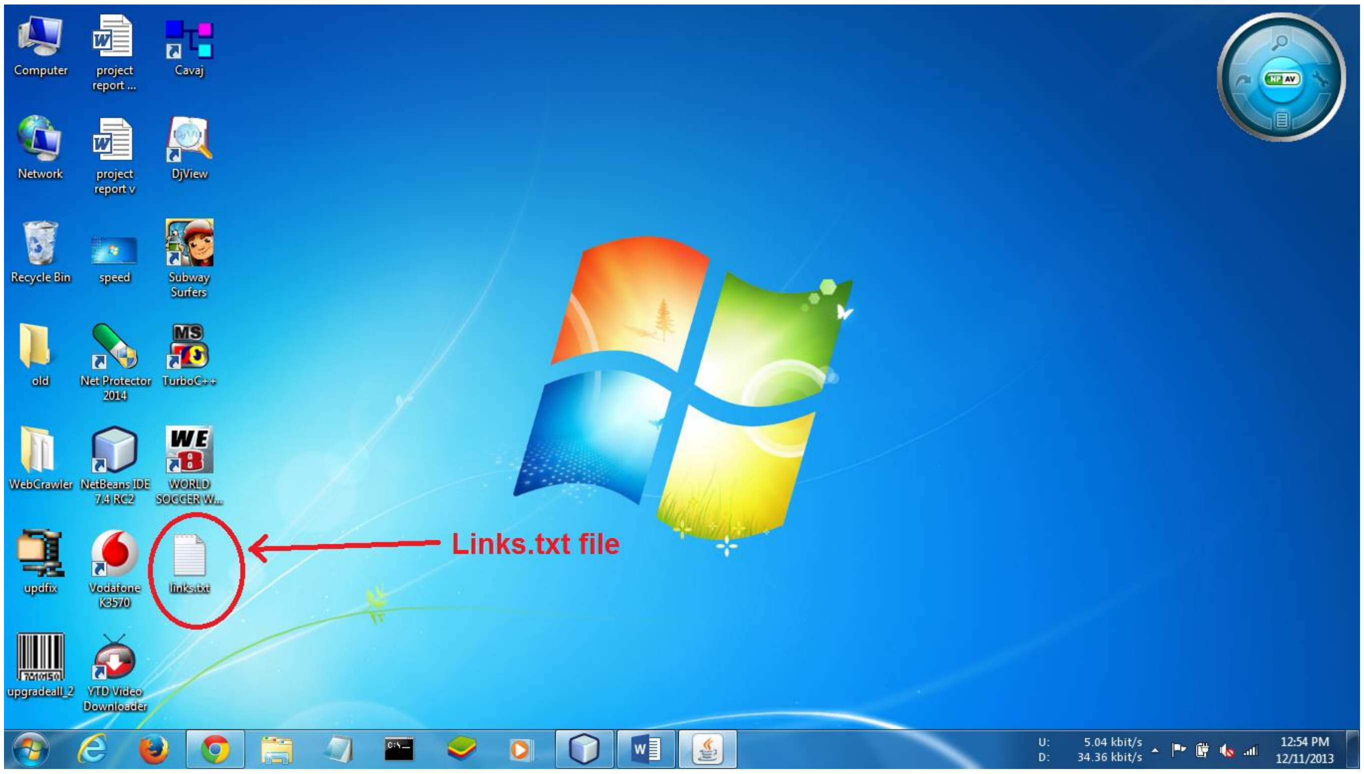


Fig5. Saved File

In above steps we have save the links in text file namely "links.txt". We have saved the file in desktop as highlighted in the figure.

TEST CASES

Case 1: No Internet Connection

If user type correct input in address bar, but if the internet connection is not available in that system then message will display telling that No Internet Connection is available into your system.

Case 2: Invalid URL

If the user type invalid URL into the address box then it will display message telling that the URL entered is not a valid URL. So user need to write URL again into the address bar.

Case 3: URL not exist

If the user type URL and if it does not exist or may be that particular site has been deactivated then it will display message that URL is not valid or does not exist.

Case 4: Security concern

Suppose the user type the URL but firstly user need to log in to access the particular site then it will prompt that please log-in.
E.g. In our college we need to login Cyberoam to use Internet Connection.

ADVANTAGES

- To create copy of websites for use in search engine.
- Monitoring the Websites.
- For checking and searching engines.
- Automation (easy to find a reader in internet).

APPLICATIONS

CONCLUSION

Web crawlers are an important aspect of the search engines. Web crawling processes deemed high performance are the basic components of various Web services. It is not a trivial matter to set up such systems:

1. Data manipulated by these crawlers cover a wide area.

2. It is crucial to preserve a good balance between random access memory and disk accesses.

FUTURE SCOPE

In this project, various challenges in the area of web data extraction have been discussed. Although this system extracts, collects the data from various websites successfully, this work could be extended in near future. In this work, a web crawler has been created which was tested. This work could be extended for other domains by integrating this work with the unified search interface and search engines.

REFERENCES

Book:

[1] Herbert Schildt, Java 2 The complete reference, Fifth Edition. New York: McGraw-Hill/Osborne,2002.

Websites:

[1] https://en.wikipedia.org/wiki/Web_crawler

[2] http://s3.amazonaws.com