A Project Report

on

HEALTHCARE SEARCH ENGINE

Submitted in partial fulfillment of the
requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under the Supervision of
Dr. Shiv Kumar Verma
Professor

Submitted By

Brijesh Khullar 18SCSE1010017
Shivam Mishra 18SCSE1010217

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA

2022

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the thesis, entitled **"HEALTHCARE SEARCH ENGINE"** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY** submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of August 2021 to December 2021, under the supervision of Dr. Shiv Verma, Professor in Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering, Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Shivam Mishra

Brijesh Khullar

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Shiv Kumar Verma

Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Brijesh Khullar : 18SCSE1010017 and Shivam Mishra : 18SCSE1010217 has been held on 13/05/2022 and his/her work is recommended for the award of Bachelor of Technology in Computer Science and Engineering.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: 13 May, 2022

Place: Greater Noida

ACKNOWLEDGEMENT

It is our privilege and solemn duty to express our deepest sense of gratitude to Dr. Shiv Kumar Verma, under whose guidance we carried out this work. We are indebted to him for his invaluable supervision, heart full cooperation and timely aid and advice till the completion of the report in spite of his pressing engagements. We wish to record our sincere gratitude for his constant support and encouragement in preparation of this report.

We take this opportunity to express our hearty thanks to all those who helped us in the completion of our project work. We are incredibly grateful to the author of various research papers, for helping us become aware of the research currently ongoing in the field. We are very thankful to our parents for their constant support and love.

Last, but not least, we would like to thank our classmates for their valuable comments, suggestions and unconditional support.

TABLE OF CONTENTS

ABSTRACT

In today's world, technology has touched each side of life however it's unhappy to listen to that scores of people die per annum not as a result of lack of medical facilities however as a result of lack of data concerning the offered medical facilities. the correct to physiological condition is of overriding importance. Republic of India has the foremost unjust attention state of affairs possible. On one hand, our country is turning into the hub for medical business wherever people from different countries flock to induce sensible quality medical treatment. On the opposite hand, most of those facilities appear to be untouchable to the natives. the rationale is that the poor status of the people. several government policies and NGOs facilitate such people however knowing concerning them remains a challenge for the people.

One among the ways that to form this search simple is by a probe system. A search system could be a software package that's designed to look for info on World Wide net. The project aims at providing a probe system which might be accustomed build searches associated with medical fields together with careful info concerning the diseases, hospitals and NGOs providing cure for an equivalent. The search system consists of assorted modules sort of a net crawler, Associate in Nursing skilled worker and a ranker. The content of net is increasing chop-chop and search engines square measure accustomed search the online content. together with this, it's conjointly turning into some extent of focus for tutorial analysis. pc programs square measure required so as to conduct any process of sites.

Several net search engines also as more specialised search tools trust net crawlers to accumulate massive assortment of pages for compartmentalization and analysis. Crawlers follow the hyperlinks in sites to mechanically transfer new and updated sites. With the eruption in diseases and increase within the range of individuals littered with it, there was a high got to build the method of treatment quite versatile through the mixture of medication and web.

Keywords- Search Engine for Heath Care system ; SQL Server ; web application ; PHP; database ; medical data ; patients, doctors.

INTRODUCTION

In today's world everything is turning into processed and net based mostly. completely different organizations have already stirred towards processed systems that created lives easier and quicker. one among the foremost necessary sectors of any nation is their health care sector. The organization of individuals, establishments and resources that deliver services associated with health to fulfill the medical wants of the overall public or anyone is observed as Health Care System. The importance of health care is Brobdingnagian during a society and over the past years, this sector has been evolving to supply a a lot of economical and processed system. Asian nations has additionally created a major improvement within the health care system and presently Bangladesh government (ICT division) is additionally undergoing completely different comes associated with health care to create this sector digitalized. the most objective of this paper is to develop an internet based mostly application for the overall public of East Pakistan wherever they can store their own medical knowledge and access it anytime, from anyplace. on-line Health Care (OHC) system is that the application of laptop, mobile phones, web and alternative technologies, which aims to produce services electronically to boost patient's medical wants. In our on-line Health Care system, users will register as patients to store their medical knowledge within the info. The system additionally consists of registered doctors below the noncommissioned hospitals, World Health Organization will provide medical device and bring down necessary medications to the patients once requested for an arrangement. there's additionally Associate in Nursing automaton mobile application related to Health Care system web application.

A search engine is a term often used to refer to web search engine. Web search engine is a tool designed to search for information on the world wide web. Search results are usually presented in a list and are available commonly referred to as hats. Details may contain web pages, images, details and other file types. Other search engines also mine data from for information or open references. Unlike Web Guides, maintained by human editors, search engines they work according to an algorithm or they are a combination of algorithmic as well personal input. Health professionals and researchers need information on reputable online sources for their accomplishment research and patient care work. Unfortunately, the Web has a large number of invalid documents their ork, even those texts that are said to be true "Drug related". The Internet and its randomly distributed database are integrated data access problem. In addition to full details, there is a word problem regarding the receipt of relevant information from programs. Usually, health information searches are by unsure of his direct questions and unfamiliar with it medical terms. A health information search engine is available several different requirements distinguish them traditional web search.

Lack of diversity problem enhanced by the nature of health web pages. Therefore, The search results provided by the current Health Web search engines often contain many semantic

subdivisions, e.g. cannot be easily managed by existing methods of identify next to duplicate texts or results diversity. For useful health information, searches usually exceed the maximum number of Web pages hard pages. (http://www.bettycjung.net/Sesiteb1.htm). The web continues to grow at an astonishing rate. Automatic search engines are becoming increasingly unpopular open useful results for search queries. With the help of Search engine, we can find authoritative, health and medical information.

LITERATURE REVIEW

The authors in [11] have discussed about that the project is developed with the help of several proven theories given by researchers all over the world. Several research papers led to the literature study and comparison of various algorithms of crawling and ranking used in this project, and also helped in understanding the basic working and functionalities of the existing systems.

HEALTH SEARCH ENGINES

MedWorm may be a medical RSS feed provided furthermore as a look Engine engineered on information collected from RSS feeds. RSS stands for extremely easy syndication and it's a technology accustomed merely publish and gather details of the terribly latest info on the web (http://www. medworm.com). GoPubMed may be a knowledge-based programme for medicine texts. The Medical Subject Headings (Me SH) function “Table of contents” so as to structure the over sixteen million articles of the MEDDLINE information base (http://www.gopubmed.com). WebMD may be a nice one-stop medical info web site. WebMD conjointly encompasses a ton of fascinating interactive calculators, quizzes and different fun stuff that helps you perceive medical info a touch easier (http://www. webmd.com). Relemed may be a new programme from the University of Virginia, faculty of drugs that searches PubMed medical literature by distribution connection to ends up in addition to merely searching for keywords (http://www.relemed.com). Daily Master of Education provides high qualitative info concerning marketed medication. This computing machine provides health info suppliers and therefore the public with a regular, comprehensive, up-to-date, look-up and transfer resource of medication content and labeling as found in medication package inserts (http://www.dailymed.nlm. nilh.gov/dailymed/about.cfm). medical man may be a general programme, associate degree innovative and progressive search application that produces it doable for anyone to easily sort in an exceedingly medical term and obtain an in depth, formatted report on it malady, condition or medication inside a number of second (http://www. chennaionline.com/hippocratesweh). MedHunt Suggests alternate spellings in many languages for misspelled terms. Searches human reviewed sites and laptop indexed sites (http://www. hon.ch/MedHunt/). Medical Word search (http://www.mwsearch.com/ mwsframetemplate.htm?http://www.mwsearch.com/). MedInd provides on-line access to full-text

of Indian medicine periodicals to the users in inside and outdoors India http://www.medind.nic.in/). IndMED may be a info covering outstanding peer reviewed Indian medicine journals. It's a info designed to supply medical skilleds/researchers/ students and therefore the medical library professional fast and straightforward access to Indian literature (http://www.indmed. nic.in/).

OpenMed: OpenMED@NIC is associate degree open access archive for Medical and Allied Sciences. Here authors / house owners will self-archive their scientific and technical documents (http://www.openmed.nic.in/). community may be a assortment of gateways which offer access to net resources within the health and life sciences for the medical professionals (http://www.intute. ac.uk/healthandlifesciences/). BioMedNet is a web community for biological and medical researchers; service includes full-text journals and viewing these needs payment of subscription fee (http://www.biomednet.com).

Web Crawler Strategies

- **Breadth First Search Algorithm**

This algorithmic program aims within the uniform search across the neighbor nodes. It starts at the basis node and searches all the neighbor nodes at identical level. If the target is reached, then it's according as success and also the search is terminated. If it's not, it return all the way down to subsequent level sweeping the search across the neighbor nodes at the extent so on till the target is reached. once all the nodes ar searched however the target isn't met, then it's according as failure. Breadth 1st is compatible for things wherever the target is found on the shallower elements in an exceedingly deeper tree. it'll not perform thus well once the branches ar numerous in an exceedingly game tree particularly like board game and additionally once all the trails cause identical objective with identical length of the path [1] [2]. Andy Yoo et al. [3] projected a distributed BFS for various branches Poisson random graphs and achieved high measurability through a collection of clever memory and communication optimizations.

- **Depth First Search Algorithm**

This is a strong technique of consistently traverse through the search by beginning at the basis node and traverse deeper through the kid node. If there's over one kid, then priority is given to the left kid and traverse deep till no a lot of kid is offered. it's backtracked to consecutive unvisited node then continues during a similar manner. This algorithmic program makes positive that every one the sides area unit visited once breadth [5]. it's like minded for search issues, however once the branches area unit giant then this algorithmic program may find yourself in Associate in Nursing infinite loop.

Ranker Strategies

- **HITS Algorithm**

This algorithm's main technique is Web Structure Mining, Web Content Mining. It computes the hubs and authority of the relevant pages. It ranks at top, the relevant as well as important page as the result. The input parameters are the Content, Backward and Forward links. The relevancy of results is more than PageRank algorithm (since this algorithm uses the hyperlinks so according to Henzinger, 2001 it will give good results and also consider the content of the page). Limitations of the algorithm are Topic drift, efficiency problem and quality of result which is less than PageRank.

Examples of Medical Search Engine

- **Bing Health**

Bing Health (previously Live Search Health) could be a health-related search service as a part of Microsoft's Bing program. it's a pursuit engine specifically for health-related info through a spread of trusty and credible sources, together with Medstory, mayonnaise Clinic, and National Institutes of Health's MEDLINE and Search ends up in Live Search Health were given during a three-column layout with health connected articles from the trusty sources within the left, internet search ends up in the center, and sponsored results on the correct. One explicit feature for Live Search Health is that each one health search queries and responses were encrypted to produce a live of privacy and security once managing health problems.

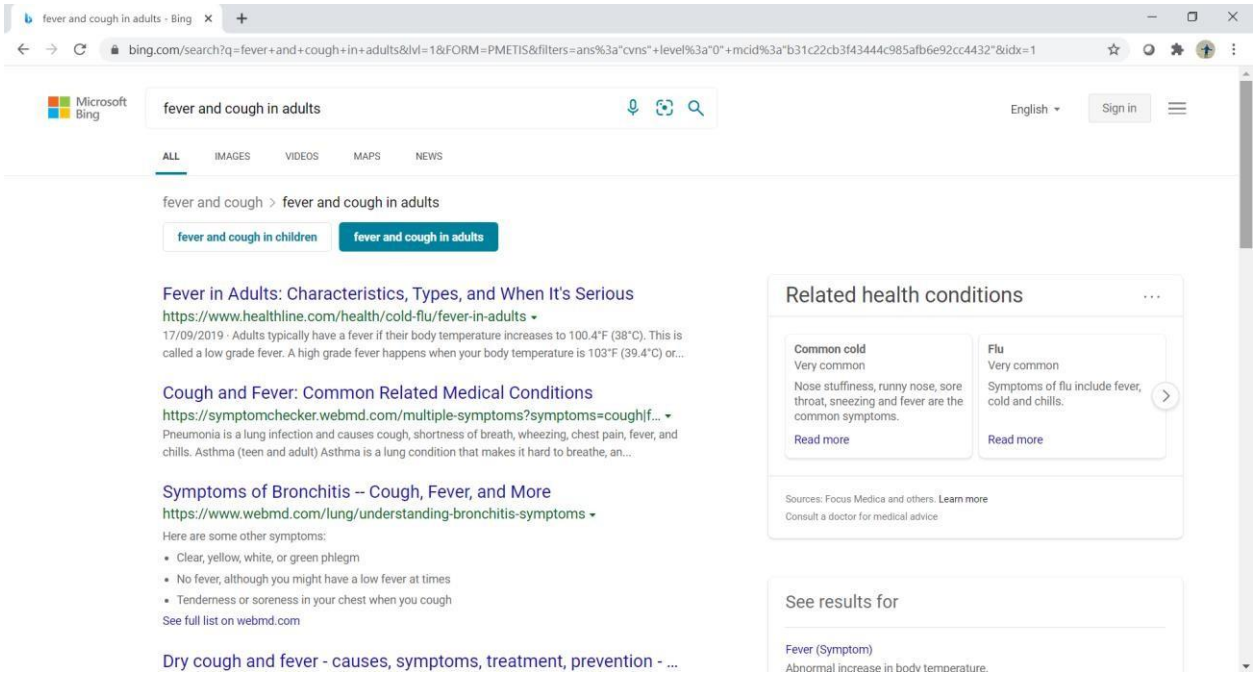


Fig: homepage for bing health search engine

- **MedSearch**

MedSearch could be a specialised medical net program, to handle these challenges. Existing net search engines usually cannot handle medical search well as a result of they are doing not think about its special needs. usually a medical info searcher is unsure regarding his actual queries and unacquainted with medical word. Therefore, he generally prefers to cause long queries, describing his symptoms and scenario in plain English, and receive comprehensive, relevant info from search results. MedSearch uses many key techniques to boost its usability and therefore the quality of search results. First, it accepts queries of extended length and reforms long queries into shorter queries by extracting a set of vital and representative words. This not solely considerably will increase the question process speed however conjointly improves the standard of search results. Second, it provides diversified search results. Lastly, it suggests connected medical phrases to assist the user quickly digest search results and refine the question. we tend to evaluated MedSearch mistreatment medical queries announce on medical discussion forums. The results show that MedSearch will handle numerous medical queries effectively and expeditiously.

MedSearch could be a specialised medical net program, to handle these challenges. Existing net search engines usually cannot handle medical search well as a result of they are doing not think about its special needs. usually a medical info searcher is unsure regarding his actual queries and unacquainted with medical word. Therefore, he generally prefers to cause long queries, describing his symptoms and scenario in plain English, and receive comprehensive, relevant info from search results. MedSearch uses many key techniques to boost its usability and therefore the quality of search results. First, it accepts queries of extended length and reforms long queries into shorter queries by extracting a set of vital and representative words. This not solely considerably will increase the question process speed however conjointly improves the standard of search results. Second, it provides diversified search results. Lastly, it suggests connected medical phrases to assist the user quickly digest search results and refine the question. we tend to evaluated MedSearch mistreatment medical queries announce on medical discussion forums. The results show that MedSearch will handle numerous medical queries effectively and expeditiously.

PROBLEM FORMULATION

Scenarios related to health have always been a major issue of concern in India since years before the independence. A majority of population has been affected since then due to malnutrition or denial to the access of health care facilities. But as the years passed by, researchers developed new technologies which led to significant advancement in science and medicine. In the current scenario, India is becoming a hub of advanced healthcare facilities with prominent medical institutions and hospitals. NGOs have been an active part in the medical field and government policies have also been in the favor of good health. But, even today we encounter a large number of deaths due to critical diseases like Cancer, Hepatitis, AIDS, Rabies, Polio, Tuberculosis, Jaundice and many more. The major reason behind this is that people are not aware of their nearest healthcare facilities or the issues arising regarding the disease they have acquired.

In this project, the main focus is on the development of such a system which provides all round information to the users related to their query.

OBJECTIVE

1. **To design and implement a web crawler:** The crawler crawls the web pages related to the queried disease as well as the hospitals and NGOs working for the same.
2. **To design and implement an indexer:** The indexer indexes all the fetched relevant data (keywords) from the web pages. Inverted Indexing is used to index the keywords.
3. **To design and implement a ranker:** The ranker adds a rank to the indexed pages so that most relevant results appear on the top of the search results.
4. **To design and implement auto complete module:** The auto complete module helps the user to type a word quickly by providing a suggestion of similar words.
5. **To provide the map service to hospitals and NGOs:** This service provides maps to reach the nearest hospitals and NGOs. The users can enter the city and the nearest hospitals and NGOs are provided as a result and they can be pinned on the map.
6. **To design a front-end application for the user to interact with the system:** The front end contains a search bar and a search button in which the query is entered by the user. The results related to disease, hospitals and NGOs are viewed in separate tabs respectively.
7. **To integrate all the above modules:** The final objective is to integrate all the above modules and implement a searching system.

REQUIRED TOOLS

Hardware Requirements

The successful implementation of the project requires:

- A high configuration laptop or personal computer to act as a server for crawling as well as for other modules.
- Internet Connection is required to run the project.

TABLE: Minimum Hardware Requirements

Hardware	Specification
Processor	Intel core 2 duo @ 2GHz
Ram	1 GB
Hard disk	20 GB
Internet Connection	Bandwidth @ 1Mbps

Software Requirements

The software requirements for the project are as follows:

- **Ubuntu:** Ubuntu is an open source software platform that runs everywhere from the smartphone, the tablet and the PC to the server and the cloud. It is a complete desktop Linux operating system, freely available with both community and professional support. The Ubuntu community is built on the ideas enshrined in the Ubuntu Manifesto: that software should be available free of charge, that software tools should be usable by people in their local language and despite any disabilities, and that 10 people should have the freedom to customize and alter their software in whatever way they see fit. "Ubuntu" is an ancient African word, meaning "humanity to others". The Ubuntu distribution brings the spirit of Ubuntu to the software world. Any version of Ubuntu can be used to run this project.

- **Java Runtime Environment:** The Java Runtime Environment (JRE), also known as Java Runtime, is part of the Java Development Kit (JDK), a set of programming tools for developing Java applications. The Java Runtime Environment provides the minimum requirements for executing a Java application; it consists of the Java Virtual Machine (JVM), core classes, and supporting files.

- **Apache Tomcat:** Apache Tomcat is an open-source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements several Java EE specifications including Java Servlet, Java Server Pages (JSP), Java EL, and Web Socket, and provides a "pure Java" HTTP web server environment for Java code to run in. Apache is developed and maintained

by an open community of developers under the auspices of the Apache Software Foundation, released under the Apache License 2.0 license, and is open-source software.

- **MYSQL Database Management System:** MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

TABLE: Minimum Software Requirements

Software	Specification
Linux OS	Any distribution. Preferred - Ubuntu 14.04
Apache tomcat server	Apache tomcat 7
MySQL database	MySQL server 5.5
Java Runtime environment	Java 7 Oracle JRE

Front End

The front end of the project is designed using HTML, CSS, JavaScript, JQuery and Bootstrap. The user can interact with the system through front end. In the front end, three user interfaces are designed namely main page, maps and admin panel. It also includes designing the logo for the project.

1. User

1. Home.html

This is an html file, in which the user finds the link for the Maps service. Once the user clicks on the link, the page is redirected to another html page named maplist.html. The main page contains a search bar, a search button and a link for maps. As soon as the user clicks in the search bar, it shifts at the top of the page and a menu bar appears which contains three tabs namely home, hospitals and NGOs. Below the three tabs is the area in which the search results appear. The user enters the query in the search bar and clicks the search button. As soon as the query in entered in the search bar, the results start appearing in the targeted area. By default, the user gets the results in home tab in which links related to the disease description appears. To get the links of hospitals and NGOs, user has to click on the respective tabs. At a time, ten search results are viewed on the page. To view further results, user has to click the next button. To view the maps, the user has to click on the maps link which is present at the top right corner of the main page.

2. Maplist.html

This html file gives user an interface to select the city and press the go button which enlists the hospitals and NGOs that are stored in the database. The results are displayed in different tabs for hospitals and NGOs respectively. Once the user clicks on any of the hospital or NGO's name, the address is pinned on the map for the user. The maps page is divided in two parts: one part contains two drop-down menus to select the city and treatment and a Go button along with the map canvas and the other part contains the area where nearby hospitals and NGOs are shown. Firstly, the user selects the treatment and city from the drop-down menu and clicks the Go button. Then the list of hospitals and NGOs is viewed in the right half of the page. By default, hospitals are viewed which contains the list of hospitals. To view NGOs, the user has to click on the NGOs tab. When the user clicks on the name of any hospital on NGO from the list, it is pinned on the map present on the map canvas at the left half of the page. The page also contains a link name Back to Searching. When the user clicks on that link, the search page re-appears.

2. Admin

1. AdminHome.html

This is an html file that serves as a homepage for the admin of the maps module. Only admin is authorized to login to this page and feed the addresses of hospitals and NGOs manually. After the admin logs in, this html page is redirected to a Login.java page.

2. AdminPanel.jsp

This is a JSP page in which the admin is welcomed if the login is successful. The admin selects from a drop down menu for which type of location has to be stored, i.e., Hospital's address has to be fed in the database or any NGO's. Then he feeds in the city and complete address and clicks the locate button which calls a function to verify the address by pinning it on the map. If the address is pinned on the map, then a request is sent to the database to store the fed data. The admin panel is a page which is used by the admin of the maps module. The page contains two text boxes: username and password and a Login button. The admin enters the username and password and clicks the login button.

Features

- All the results are viewed in separate tabs.
- The results are provided in well-structured form which makes easy for the user to access the system.
- The user can search the nearby hospitals and NGOs from the maps as well by providing treatment and city as input.
- The hospitals and NGOs are pinned on the map when user clicks it.
- The links of maps and main page are present on the pages which makes it easy for the user to switch between the two pages.
- A large number of results can be viewed by clicking on the next button.

Back End

1. Crawler

At the back end, a main method is called for starting the crawler. It also checks for validation of arguments passed. The arguments passed are the location of seeds.xml file and max depth up to which the crawler would go. It also updates seeds in seeds_table in database and it calls the functions to crawl the pages. Another function checks if the seeds.xml exists at the given location or not. The program will proceed only if the file exists. Seeds.xml file is read and then the functions of InjectSeed.java are called to insert those seeds in database.

Database.java contains functions to return the database connections wherever it is needed in the program. MaintainIndex.java contains method to check whether a page is fetched in the previous runs of crawler or not. This function also can clear the indexes of any website if needed.

CrawlPages.java is the main file of the program responsible for crawling the pages and sending the content to PreIndexer.java. The class has functions that crawl the pages, adds the fetched pages to database, maintains ranking factors in the database, check that page has been crawled or not etc. This is the main controlling file of the whole module. It also simultaneously sends the crawled pages for indexing purpose.

PreIndexer.java has functions which fetches the contents from the crawled page according to pre-defined tags. It tokenizes the string and also removes the stop words. Then it sends the remaining content to WordIndexer.java which contains methods that inserts the words with the page number they are present in the database. It is responsible for maintaining all of the word to page mapping

table in the database. The searching takes place in this word to page mapping table only.

2. Auto Complete

Auto complete module helps to bring on suggestions as the user types the queries. This module takes every string of letters that the user types, find the words that start from that string, sorts the words in order of decreasing frequency and picks up the top four results and present them as suggestions to the user. The frequency here is maintained as the number of times a particular word has been searched by the users. When a user types his query and clicks on the “Enter” or “Go” button, that particular query is split by ”space” and the resulting words’ frequencies are increased by one in the database.

3. Maps

In the back end of this module, the location which user inputs is updated on the map by the process of geo-location and pinned on it. Another JSP page is used as an interface between the database and the admin, through which the admin can feed in the data like address of hospitals and NGOs specializing in particular treatments. A mapping function is also used to map the hospitals with their specialized treatments that are offered and remove redundancy.

Features

- The main feature of the crawler is that it is fast when the data is on smaller-scale i.e. it crawls ten pages per minute on an average.
- The crawler is memory efficient and also indexes the data to the database quickly.
- The process of feeding the seeds via the XML file is easy.
- The auto complete module displays fast results out of most frequently searched words.
- The map module pins the searched location of the hospitals and NGOs on the map.
- The map module provides an interface to add treatments that are offered by the hospital and create the database with non-redundant data.
- The map module helps update the auto complete file.
- The map module provides a user interface feature for the user to choose the city and treatment from a list of non-redundant options.

FEASIBILITY ANALYSIS

Economic Feasibility

- **Cost/Benefits Analysis:** The cost/benefits factor is quite low which means that the cost incurred to build the system is low as compared to the benefits this system provides to the user.

Operational Feasibility

- **Reliability:** Any kind of information that is generated in the result is reliable as the admin is responsible to feed in only reliable and trusted websites for crawling.
- **Maintainability:** The system is easy to maintain since only crawled data and the ranked results have to be updated time to time. And any other maintenance of hardware is easy, too.
- **Usability:** The system being user-friendly helps the user search any query on the search engine easily and also provides tabs to display information in a distinguished manner.
- **Affordability:** To use the system, a user only needs an internet connection with a good bandwidth and nothing else which makes it quite affordable.

Technical Feasibility

- **Hardware Feasibility:** The system requires a server, mainly to keep the process of crawling ongoing for several hours as per the need.
- **Software Feasibility:** The software required by the system are a database to index all the crawled data, Eclipse IDE and Apache Tomcat to build the project and few Java tools like JDK, etc. to implement java language required for coding.

Fig: graph representation of immediacy of health system

SYSTEM ANALYSIS

The system analysis details about the existing systems that are both present in theory or market, and their drawbacks which led to the proposed system. It also showcases all the details like technical, operational and economic feasibility, too.

Existing System

There are several online portals and websites helping people these days to find doctors, hospitals and blood banks in their city and nearby localities. Few examples of these are doctorbabu.com, medindia.net, etc. They even help the common people calculate diabetes risk assessment, blood sugar conversion online, help them find medical colleges, resources, specialty services, directories, various articles and drugs for specific diseases and pain.

 MedSearch could be a specialised medical net program, to handle these challenges. Existing net search engines usually cannot handle medical search well as a result of they are doing not think about its special needs. usually a medical info searcher is unsure regarding his actual queries and unacquainted with medical word. Therefore, he generally prefers to cause long queries, describing his symptoms and scenario in plain English, and receive comprehensive, relevant info from search results. MedSearch uses many key techniques to boost its usability and therefore the quality of search results. First, it accepts queries of extended length and reforms long queries into shorter queries by extracting a set of vital and representative words. This not solely considerably will increase the question process speed however conjointly improves the standard of search results. Second, it provides diversified search results. Lastly, it suggests connected medical phrases to assist the user quickly digest search results and refine the question. we tend to evaluated MedSearch mistreatment medical queries announce on medical discussion forums. The results show that MedSearch will handle numerous medical queries effectively and expeditiously.

MedSearch could be a specialised medical net program, to handle these challenges. Existing net search engines usually cannot handle medical search well as a result of they are doing not think about its special needs. usually a medical info searcher is unsure regarding his actual queries and unacquainted with medical word. Therefore, he generally prefers to cause long queries, describing his symptoms and scenario in plain English, and receive comprehensive, relevant info from search results. MedSearch uses many key techniques to boost its usability and therefore the quality of search results. First, it accepts queries of extended length and reforms long queries into shorter queries by extracting a set of vital and representative words. This not solely considerably will

increase the question process speed however conjointly improves the standard of search results. Second, it provides diversified search results. Lastly, it suggests connected medical phrases to assist the user quickly digest search results and refine the question. we tend to evaluated MedSearch mistreatment medical queries announce on medical discussion forums. The results show that MedSearch will handle numerous medical queries effectively and expeditiously.

 Bing Health (previously Live Search Health) could be a health-related search service as a part of Microsoft's Bing program. it's a pursuit engine specifically for health-related info through a spread of trusty and credible sources, together with Medstory, mayonnaise Clinic, and National Institutes of Health's MEDLINE and.

Search ends up in Live Search Health were given during a three-column layout with health connected articles from the trusty sources within the left, internet search ends up in the center, and sponsored results on the correct. One explicit feature for Live Search Health is that each one health search queries and responses were encrypted to produce a live of privacy and security once managing health problems.

Drawbacks

Information available on portals about drugs online might help people in emergency at times but can even worsen the situation if the drugs are taken without any doctor's consent. Since not all are aware of the what they are allergic to or know the proper dosage of the medicine. So, probably only the information about the salts of the drugs online might cause a chaos. Other search engines, like Bing Health only provide articles and information about the disease and the symptoms which can be generally found on any general search engine does not prove to be so effective to medical problems. And another engine, MedSearch do not provide best results when it comes to the doctors and hospitals as they are working commercially and ranks the sites in any order they want.

Proposed System

This project is about a platform that consolidates all the information about the disease, its symptoms, the concerned hospitals and NGOs providing treatment for the same. There is an additional feature of the map, which helps users pin the address of hospitals and NGOs on it to mark the area where they are situated and verify if it actually exists or not.

PROJECT DESCRIPTION

Project description describes all the modules of the system along with the diagrams like DFD, E-R diagram and database design diagram. The input output designs of the system are also included in this.

Overview of the Project

The project is a search system which is dedicated to the medical field. It has a friendly Google-like interface, and helps to solve user's queries related to health problems. The results are separated into three different categories- Informative, Hospitals, and NGO's with easy navigation between each one of them. The project also supports Map service, which helps users to find the nearest Hospitals and NGO's working for the cause.

Module Description

There are 3 main modules in the project- crawler, autocomplete and maps.

Modules

1. Crawler Module

a) StartRun.java

This file contains main method for starting the crawler. Its checks for validation of arguments passed. The arguments passed are the location of seeds.xml file and max depth up to which the crawler would go. It also updates seeds in seeds_table in database and it calls the functions to crawl the pages.

b) LocationCheck.java

This file contains method for checking that seeds.xml exists at the given location or not. The program will proceed only if the file exists.

c) UpdateSeeds.java

This file contains functions to read the seeds.xml file and calls the functions of InjectSeed.java to insert those seeds in database.

d) InjectSeed.java

This file contains methods that perform the function of inserting the seeds from seeds.xml file to database. This also performs the functions to clear the outdated seeds from the database.

e) Database.java

This file contains functions to return the database connections wherever it is needed in the program.

f) MaintainIndex.java

This file contains method to checked weather a page is fetched in the previous runs of crawler or not. This function also can clear the indexes of any website if needed.

g) Crawlpages.java

This is the main file of the program responsible for crawling the pages and sending the content to PreIndexer.java. The class has functions that crawl the pages, adds the fetched pages to database, maintains ranking factors in the database, check that page has been crawled or not etc. This is the main controlling file of the whole module. It also simultaneously sends the crawled pages for indexing purpose.

h) PageInfoBean.java

This is a java bean responsible for holding three variables for a particular page, the page's readable status, number of hyperlinks in a page, and an array of hyperlinks in a page. The object of this class is needed in the internal working of the crawling module.

i) PreIndexer.java

This file has functions which fetches the contents from the crawled page according to pre-defined tags. It tokenizes the string and also removes the stop words. Then it sends the remaining content to WordIndexer.java.

j) ProcessStopWords.java

This file contains a method that is responsible for removing the stop words form the fetched Strings. The stop words are like a, an, the etc.

k) WordIndexer.java

This file contains methods that inserts the words with the page number they are present in the database. It is responsible for maintaining the all of word to page mapping table in the database. The searching takes place in this word to page mapping table only.

2. Auto Complete Module

a) XMLWriter.java

This is a java class that maintains a connection to the database and retrieves the words with higher frequencies from database and writes them in an XML file "transfer.xml".

b) ReadXML.java

This is a java class that reads the XML file "transfer.xml" and returns a Document object "doc".

c) Autocomplete.js

This is a JavaScript file that picks the data from "home.html" page and sends the data to two different servlets- "UpdateFrequency.java" and "Autocomplete.java".

d) Autocomplete.java

This is a servlet which receives each string of data typed by the user through an Ajax request. It further sends the data to a java class "Matcher.java". It is also the main servlet that returns the suggestions to the "home.html" page.

e) UpdateFrequency.java

This is a servlet which receives each word typed by the user, through an Ajax request. It increases the frequency of each received word in the database.

f) Matcher.java

This is a java class that receives the string of data from “Autocomplete.java” and searches the “doc” for words that match the last word of the received string. Then, it sorts the results based on the frequency and returns top 4 frequently used words to “Autocomplete.java”.

Map Module

I. User

a) Home.html

This is an html file, in which the user finds the map link on the homepage of the spider search engine. Once the user clicks the link, the page is redirected to another html page named maplist.html.

b) Maplist.html

This html file gives user an interface to select the city and press the go button which enlists the hospitals and NGOs that are stored in our database. The results are displayed in different tabs for hospitals and NGOs respectively. Once the user clicks on any of the hospital or NGO's name, the address is pinned on the map for the user.

Maplist.html has following functions

i. function init()

A function that initializes the map by a city with the help of the latitude and longitude values (geocoding) predefined by the admin, i.e., Ghaziabad here.

ii. function updateMap()

It is a function that updates the map by pinning the address of any hospital or NGO that the user wants to see on the map. And also calls another function named ShowLocation().

iii. function showLocation()

This function pins the address on the map as well as helps the user zoom in and convert the view type of map from terrain view to maybe a satellite one.

iv. Back to Searching

This is another link on the maplist.html page which directs the user back to the homepage of the spider search engine.

II. Admin

a) AdminHome.html

It is an html file that serves as a homepage for the admin of the maps module. Only admin is authorized to login to this page and feed the addresses of hospitals and NGOs manually. After the admin logs in, this html page is redirected to a Login.java page.

b) Login.java

It is a servlet in which the login of the user is checked by validate.java class which welcomes the user if login is successful otherwise returns to the AdminHome.html page.

c) AdminPanel.jsp

A JSP page in which the admin is welcomed if the login is successful. The admin selects from a drop down menu for which type of location has to be stored, i.e., Hospital's address has to be fed in the database or any Ngo's. Then he feeds in the city and complete address and clicks the locate button which calls a function to verify the address by pinning it on the map. If the address is pinned on the map, then a request is sent to the database to store the fed data. AdminPanel.jsp has following buttons:

a. Update Auto Complete

This button calls a JavaScript function updateAutocomplete() which makes an Ajax call to XMLWriter.java. This servlet updates the transfer.xml file.

b. Get Frequency

This button calls a JavaScript function getFrequency() which makes an Ajax call to GetFerquency.java. This servlet returns the sum of frequencies of the words and also tells weather the updation of auto complete xml file is recommended or not.

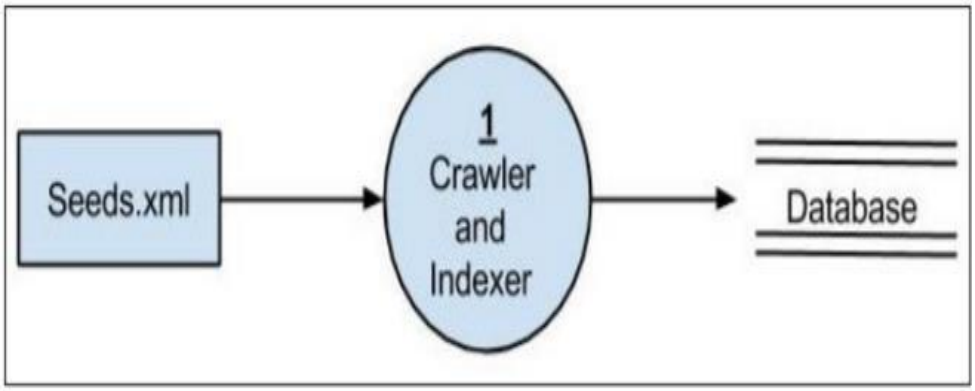
c. Clear Frequency

This button calls a JavaScript function clearFrequency() which makes an Ajax call to ClearFrequency.java. This servlet sets the frequencies of all the words in the database to one. This is done so that the frequencies do not go to very high integer values.

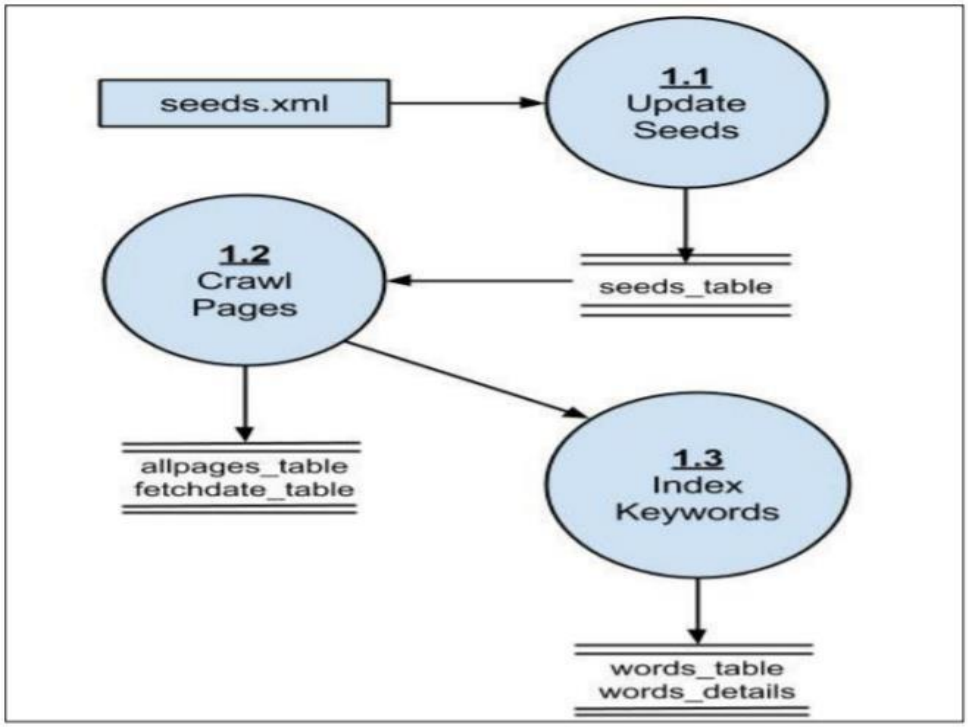
DIAGRAMS

Data Flow Diagram

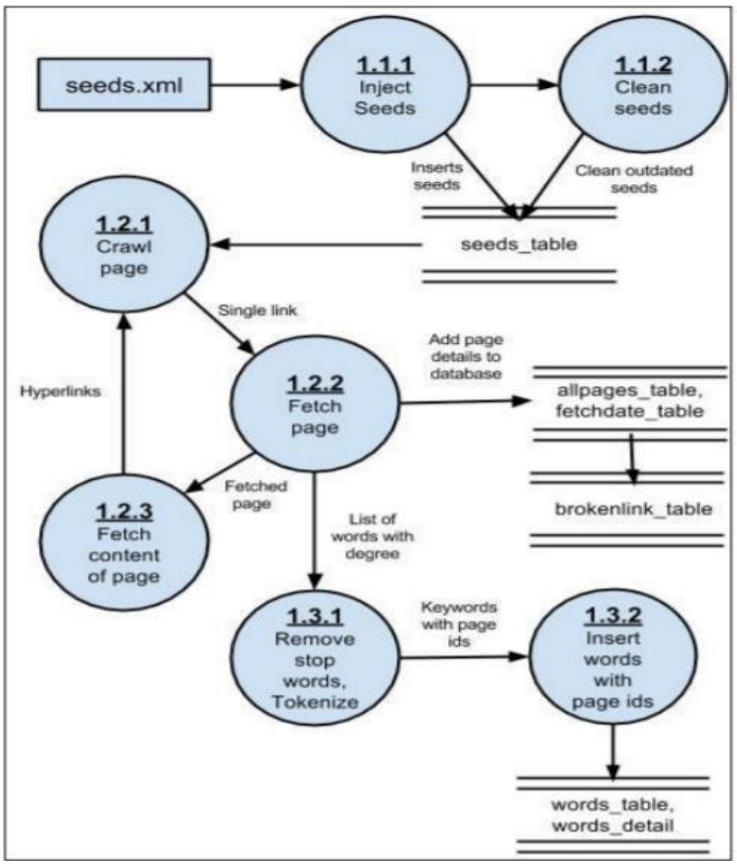
1. **Crawler and Indexer**



0 Level DFD for Crawler and Indexer



1 Level DFD for crawler and Indexer



2 Level DFD for Crawler and Indexer

1. Searcher and Ranker

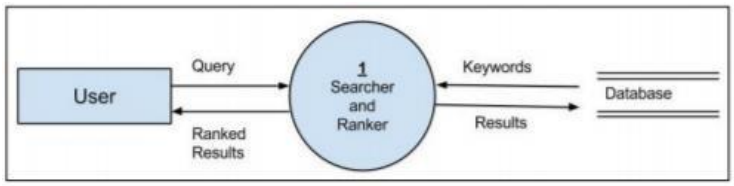


Fig: 0 Level DFD for Searcher and Ranker

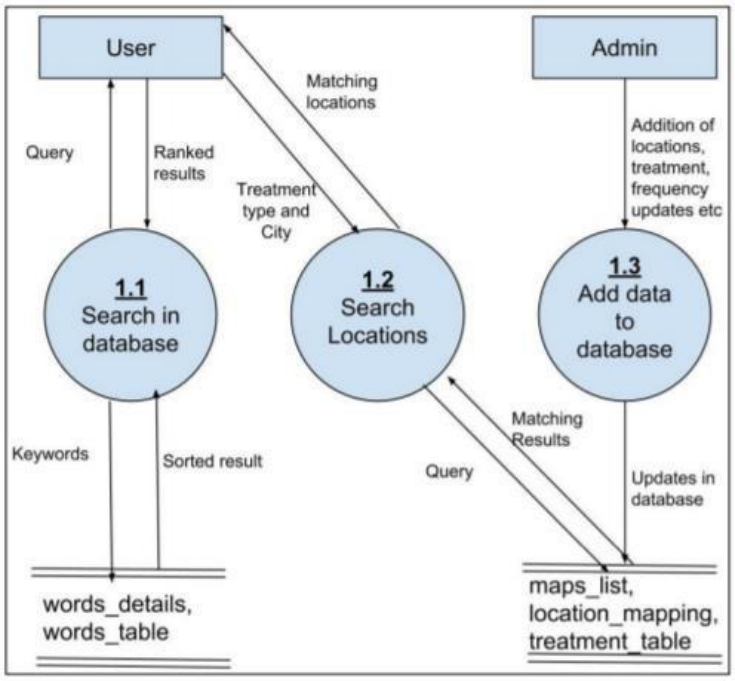


Fig: 1 Level DFD for Searcher and Ranker

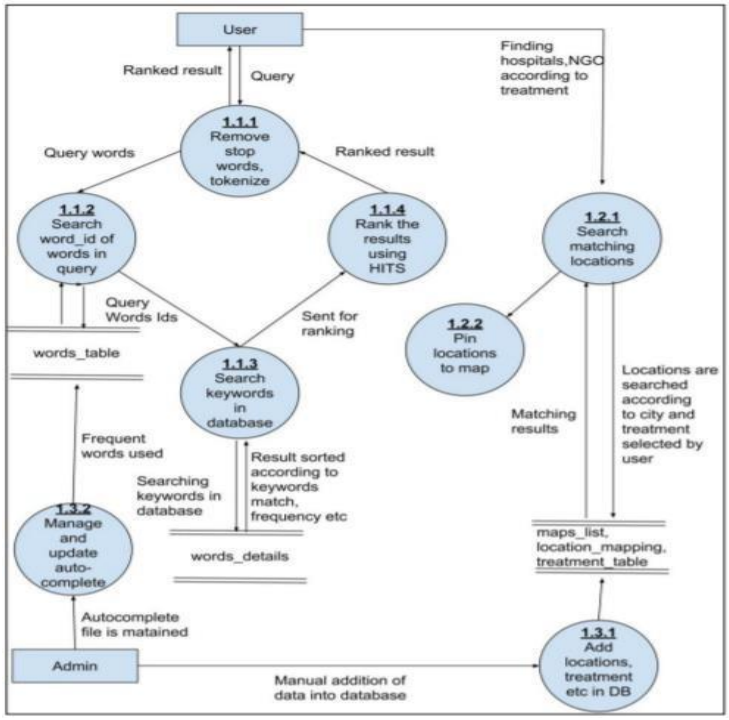
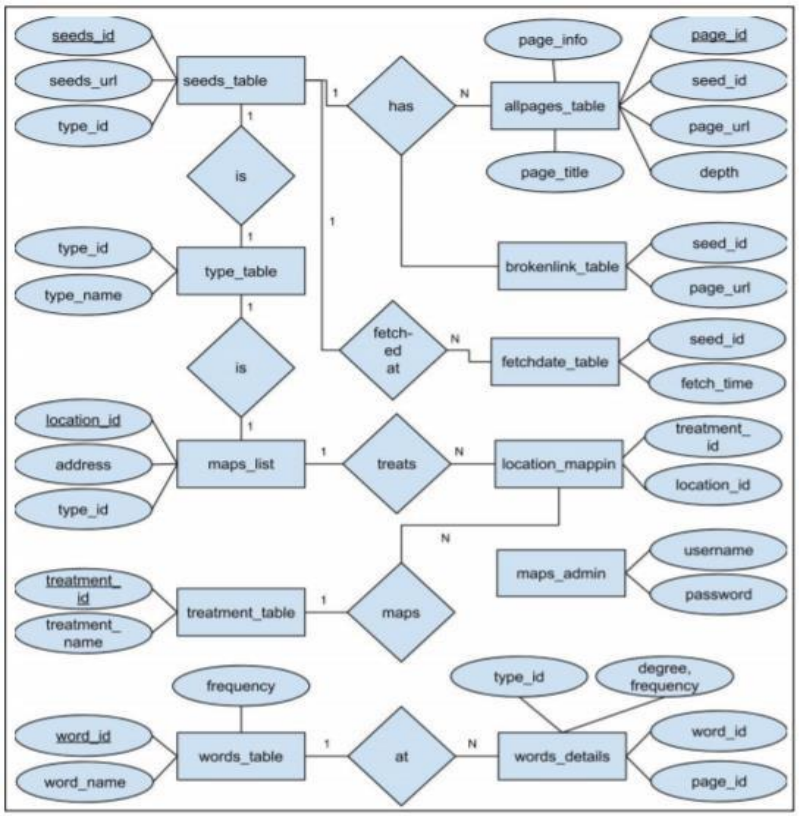
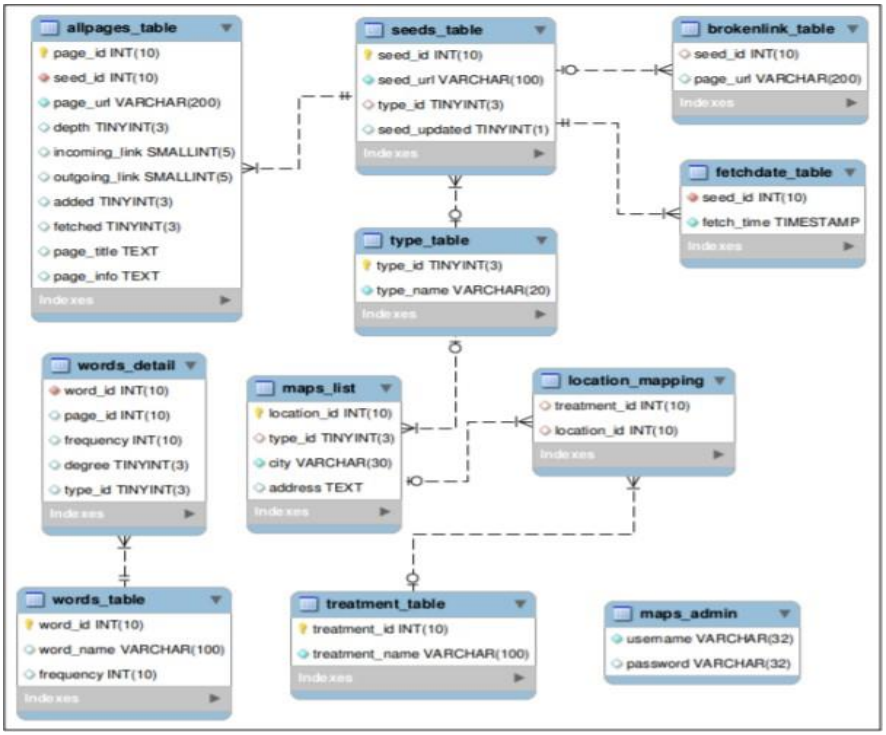


Fig: 2 Level DFD for Searcher and Ranker

E-R Diagram



Database Design



SYSTEM TESTING

Different types of testing like unit testing, acceptance testings are performed with the help of suitable test cases in this module.

Unit Testing

In this phase, each module and its sub-modules has been tested independently.

 The Crawler module is able to successfully crawl the websites from the web and index it accordingly.

 The AutoComplete module can successfully read the indexed words from the database and sort the words based on frequency.

 The Map module successfully pins the requested landmarks on the map based on the location selected.

Acceptance Testing

The entire project has been tested by some random users from their terminal.

 The user-interface has been thoroughly checked for its ease of use and navigation.

 The users have validated how the project gathers and presents results based on the search query.

 The presentation of the results in the three categories has also been verified.

 The results shown to the user have been validated for their decreasing order of relevance.

 The suggestions provided by the Auto complete module have also been validated.

 The Map module has been validated for its accuracy to pin-point correct landmarks, and also their relation to the disease queried.

 The behaviour of the system in case of no-input has also been tested.

 The behaviour of the system in the case of non-relevant query has been tested.

SYSTEM IMPLEMENTATION

System Implementation comprises of the algorithm of the functions that different module perform in the system. The algorithm is a step by step procedure of how the modules work.

1. Crawler

1. Read seeds from seeds.xml and insert them to database.
2. Read seeds from the database one by one and pass them to crawler function.
3. The crawler function will fetch the page if it hasn't been fetch.
4. It sends the page to indexer as document object.
5. It also extracts the hyperlink from the page and sends them to the crawler function in step 2.

2. Indexer

1. The indexer receives the page.
2. Fetches textual content from it.
3. Remove the stop words.
4. Tokenizes the content.
5. Insert those words with their page ids and tag information.

3. Auto Complete

1. Write the indexed words from database into an XML file and set the initial frequency of each word to 1.
2. Read the XML file into a Document doc.
3. As soon as the user types some string in the input box,

```
 if (string has no space)
 {
```

Pick it through AJAX,

Search Doc for words that have this string as prefix.

From the list of matched words, display the top 4 results which have highest frequency.

}

else

{

Pick that string using AJAX,

Spilt the string based on “spaces”,

Pick the last word from the string,

Search for words that start with the last word in the Document doc,

Now, among the list of all the words that start with that string, sort the list in decreasing order based on frequency,

Concatenate top 4 results with the results of previous words,

Display them to the user.

}

4. When the user types some string and presses “Enter” or clicks on “Go”,

- Split the string using space,
- Remove stop words,
- Pick each word,
- Find that word in the Doc, and

  Increase its frequency in the XML file and Database by 1.

5. For each string entered thereafter, repeat steps 3-4.

4. Map Module

1. Take user input.

2. User selects name of city and treatment.

3. The treatment and city data are fed to the SQL query being processed to display the results.

4. The address of the hospitals and NGOs are displayed in the respective tags.

5. When the user clicks the address of any hospital or NGO, it gets pinned on the

map displayed on the page.

6. The user can go back to searching by clicking the link on the top right corner of the page.

5. Admin Panel

1. The admin panel is authenticated by two fields, the username and password.
2. If the username and password matches the data previously stored in the database, the authentication process is complete and the login is successful.
3. The admin can feed the information by selecting a hospital or NGO.
4. Then feeds the address and city of the hospital or NGO, respectively. And click the submit button to insert new data in database.
5. Admin feeds the treatments and then maps the treatment with their corresponding hospitals and NGOs. The mapping function helps avoid redundancy in the records of the database.
6. Get Frequency and Clear Frequency functions counts the number of words searched and clears the xml file if the words searched increases a limit of 100 words, respectively.
7. Update AutoComplete function updates the frequency of words in an xml file after reading the frequencies from the database.

SOURCE CODES AND OUTPUT

Crawler & Indexer

1. The seeds in seeds.xml file is feed into process Controller() function of crawl pages CrawlPages.java. This is a recursive function which sends each link it gets into the crawling function.

```
//This method will insert the seed and will recursively find all its pages at different depth.

Public static void processController(int seed_id, String page_url ,int depth) {
if(depth<=maxdepth) {
if(page_url.length()<200)       {
boolean    isUnique    =    false;
boolean    isfetched    =    false;
isUnique = isUnique(page_url);
```

```
if(!isUnique)
{
isfetched = fetched(returnPageId(page_url));
}
if(!isfetched)
{
PageInfoBean page_object = pageInfo(seed_id, page_url,
depth,isUnique);
if(page_object.isReadStatus())
{
for(String url: page_object.getLinks()) {
if(!isBroken(url))processController(seed_id,url,
depth+1);
}
}
page_object = null;
}
elseincreaseLink(returnPageId(page_url), depth);
}
}
}
        }
```

2. The fetched page is passed to writeToFile() function of PreIndexer.java class in the form of document object. This class contains functions for fetching the specific content from the page.

```
static String[] tofetch= {"h1","h2","h3","h4","h5","h6","em","strong","b"};
Elements elements = null;
//Replace all symbols, stopwords and fetches the data between the above tags.

Public void writeToFile(int page_id, String page_url, int type_id, Document doc) {
ArrayList<String> array = new ArrayList<String>();
for(Element meta : doc.select("meta"))
{
```

```
String text = meta.attr("content").replaceAll("\\[.*?\\]", "");
array = new ProcessStopWords().removeStopWords(text);
for(String word : array)
{
WordIndexer.wordIndexer(word, page_id, 1, type_id);
}
}
for(String tag: tofetch)
{
elements = doc.select(tag);
for(Element e: elements)
{
array = new ProcessStopWords().removeStopWords(e.text());
for(String word : array)
{
WordIndexer.wordIndexer(word, page_id,degreeReturn(tag),type_id);
}
}
}
array = null;
}
```

3. The method below adds an entry of word to page mapping for the existing words in the database.

```
//This method adds a page entry for a particular word.

Public static boolean addPageEntry(int word_id, int page_id, int degree, int type_id)
{
boolean add = false;
try
{
PreparedStatement stmt = conn.prepareStatement("INSERT INTO words_detail
(word_id, page_id, frequency, degree, type_id) VALUES (?,?,?,?,?)");
```

```
        stmt.setInt(1, word_id);
stmt.setInt(2,          page_id);
stmt.setInt(3, 1);
stmt.setInt(4,  degree);
stmt.setInt(5, type_id);
int status = stmt.executeUpdate();
if(status != 0 ) add = true;
stmt.close();
}
catch (SQLException e)
{
System.out.println("Error: SQLException occurred while adding an pageentry to
database.");
}
return add;
        }
```

Searcher& Ranker

Now after the indexes are made in the database in the “words_details” table, the system is ready for searching. The searching query is entered from the Home page of the project, and the words are matched with the words in the database, the best page links are fetched in return which are ranked high according to ranking algorithms.

1. This function has the search query used in searching and ranking. It makes the query ready by adding the word_id string, limits of the result and type of the result in the query.

```
//The SQL query used for searching and ranking.

protected String SQLQuery(String array, String type, int page)
{
int lowerlimit = (page*10)-10;
String query = "SELECT B.page_id FROM allpages_table AS A INNER JOIN \n" +"(SELECT
page_id, COUNT(page_id) AS matchCount, AVG(degree) AS avgDeg, SUM(frequency) as
sumFreq \n"
+ "FROM words_detail WHERE type_id="+type+" and word_id IN ### \n" + "GROUP BY
```

```
page_id
ORDER BY matchCount DESC, avgDeg, sumFreq DESC limit "+lowerlimit+",10) AS B \n" +
"ON
A. page_id=B.page_id ORDER BY (A.incoming_link + A.outgoing_link)/2.0 DESC;";
query = query.replaceAll("###", array);
return query;
        }
```

Auto Complete

The codes below are related to Auto Completion system that is used in Searching interface.

The Auto Completion system fetches the most searched words which match with the word the user is typing. It uses the functions below.

The incomplete words that are in process of being typed are sent to the AutoComplete.java which receives the search string with incomplete word and match the incomplete word with the most searched words in the transfer.xml file. The matching is done with the help of match() function in Matcher.java class.

1. The searching for AutoComplete system is done in the transfer.xml file and the top four results with highest frequencies are returned back to the home page and put in the drop down for user convenience. The matching and sorting according to frequencies are done with the help of code below.

```
//Receives the string and returns the matching results.

public String match(String string, File xmlfile) throws ParserConfigurationException,
SAXException,
IOException
{
String result="";
String[] words = string.split("\\s+");
int arraysize = words.length;
string = words[arraysize-1];
Document doc = newReadXML().read(xmlfile);
NodeList first = doc.getDocumentElement().getChildNodes();
HashMap<String, Integer> hmap = new HashMap<String, Integer>();
String freq = null;
for(int i=0;i<first.getLength();i++)
```

```
{
if(first.item(i).getTextContent().startsWith(string))
{
freq=first.item(i).getAttributes().getNamedItem("freq").getNo   eValue();
int                                          intfreq=Integer.parseInt(freq);
hmap.put(first.item(i).getTextContent(), intfreq);
}
}
String prevwords="";
for(int i=0;i<(arraysize-1);i++)
{
        prevwords = prevwords.concat(words[i])
        prevwords = prevwords.concat(" ");
}
Map<String, Integer> map = sortByValues(hmap);
Set<Entry<String, Integer>> set = map.entrySet();
Iterator<Entry<String, Integer>> iterator2 = set.iterator();
int count=1;
while(iterator2.hasNext() && count<=4)
{
Entry<String, Integer> me = iterator2.next();
result=result.concat("<option>"+prevwords+""+me.getKey()+"</option>");
count++;
}
return result;
        }
```

Maps

The Search Engine is equipped with Maps feature which help the user to search for hospitals and NGOs for particular treatment and city. These Hospitals and treatments are manually fed by the admin by an admin panel.

1. The Maps page of user takes the input of treatment and city from the user. The both parameters are sent to Location.java servlet. The following function receives it, fetches the matching results and sends it back to the user.

```
//doGet method of Location.java. Resposible for searching matching Hospitals and NGOs
according to users requirement.

Protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
response.setContentType("text/html");
PrintWriter out= response.getWriter();
String city = request.getParameter("city");
String treatment = request.getParameter("treatment");
String result="";
String type = request.getParameter("type");
if(treatment.equals("0")) treatment = "treatment_id";
if(city.equals("0")) city = "B.city";
else {
city = "'".concat(city);
city = city.concat("'");
}
try{
        Connection con= newDatabase().returnConnection();
Statement stmt = con.createStatement();
ResultSet rs= stmt.executeQuery("SELECT DISTINCT
B.address,B.type_id,B.location_id FROM maps_list AS B INNER JOIN (SELECT location_id
FROM
location_mapping where treatment_id="+treatment+") AS A ON A.location_id=B.location_id
WHERE
B.city="+city+" and B.type_id="+type+"");
while(rs.next()) {
result=result.concat("<div style='cursor:pointer;'
id="'+rs.getInt("B.location_id")+"'
onclick='UpdateMap(this)'>"+rs.getString("B.address")+"</div><br>"); }
if(result.equals("")) result = result.concat("NO RESULTS FOUND !!!");
out.println(result);
}catch(Exception e){
System.out.println(e); }
}
```

Test Cases

Test Case 1:

Input: URLs in seeds.xml.

Expected Output: Crawling starts, with successfully crawled sites indicating Success, else fail.

Actual Output: Crawling starts, and successfully crawled sites indicate Success, else fail.

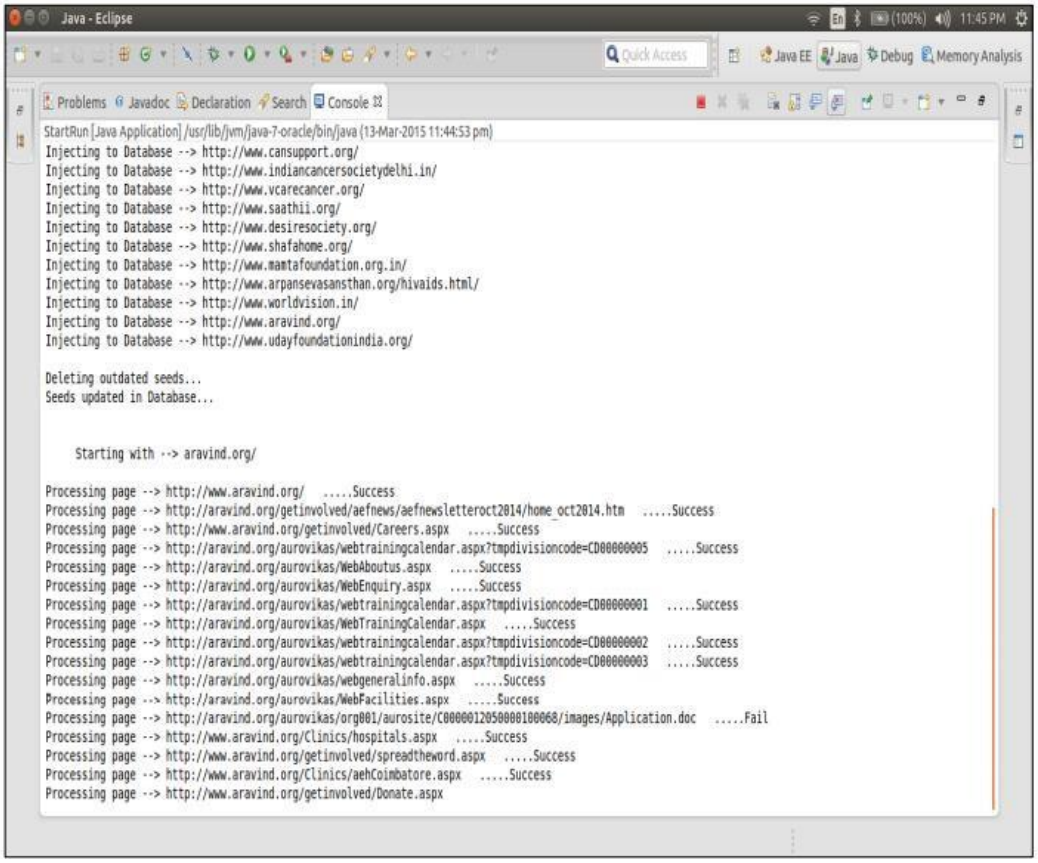


Fig: crawling of web pages

Test Case 2:

Input: Breast c

Expected Output: Words with “Breast c” as the prefix in sorted order.

Actual Output: Words with “Breast c” as the prefix in sorted order.

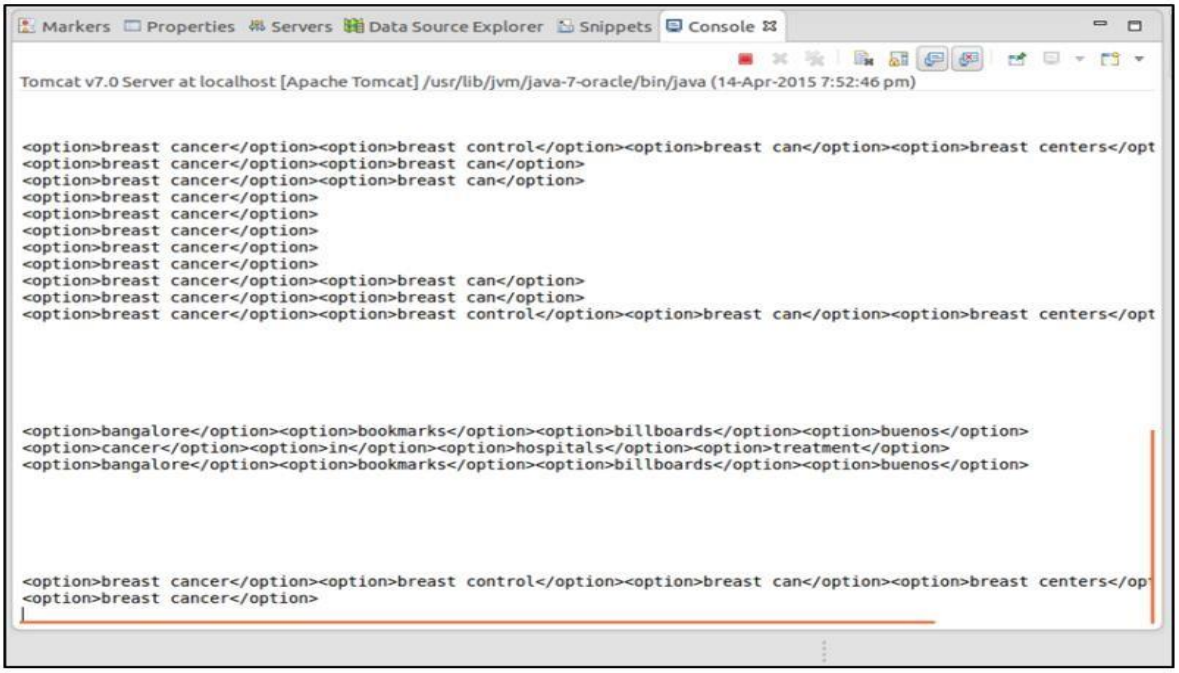


Fig: console of search engine

Test Case 3:

Input: City, treatment

Expected Output: List of Hospitals and NGOs, and on clicking on a result, it shows its location on the map.

Actual Output: List of Hospitals and NGOs, and on clicking on a result, it shows its location on map.

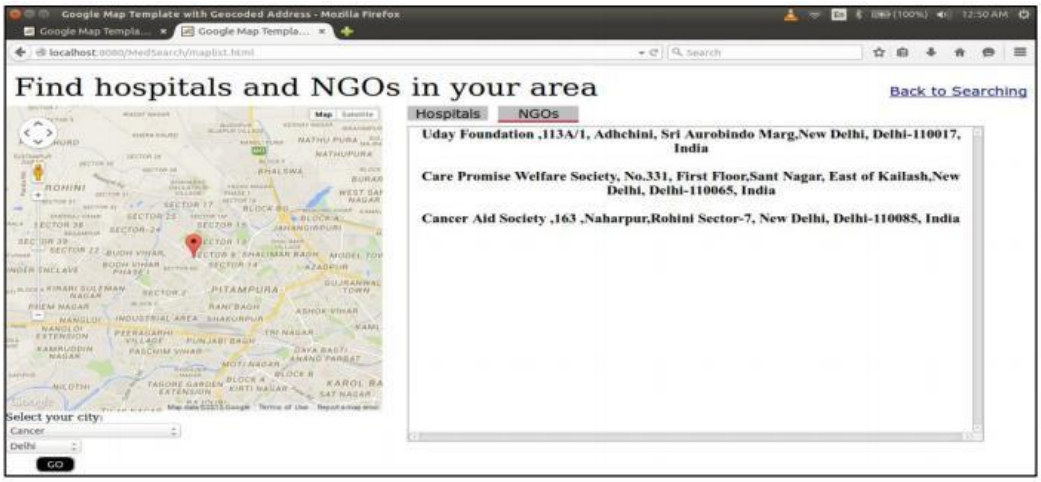


Fig: test case for finding hospitals on search engine

Test Case 4:

Input: Correct Username and Password

Expected Output: Google Map template with Geocoded address page.

Actual Output: Google Map template with Geocoded address page.

Fig: login page

Test Case 5:

Input: Location

Expected Output: Successful pin on the map.

Actual Output: Successful pin on the map.

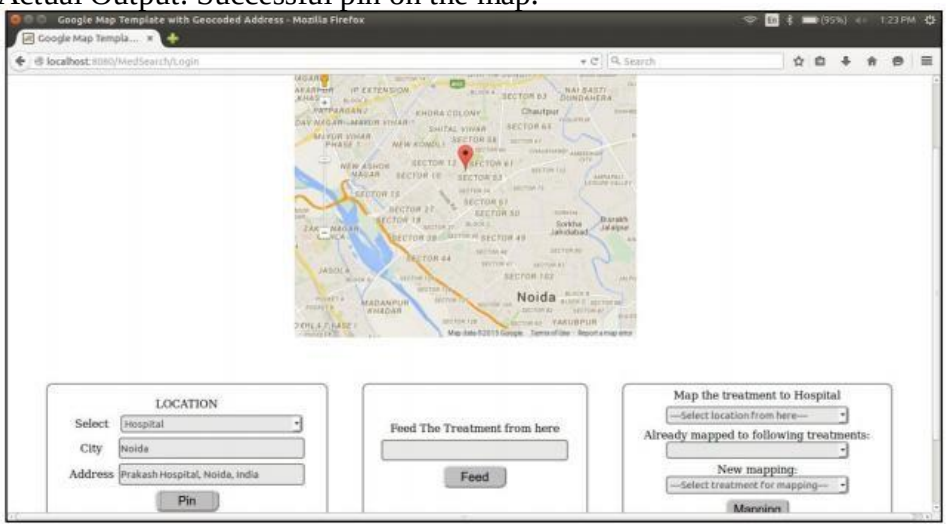


Fig: mapping of health related services

Test Case 6:

Input: Any treatment name

Expected Output: Treatment name inserted.

Actual Output: Treatment name inserted.

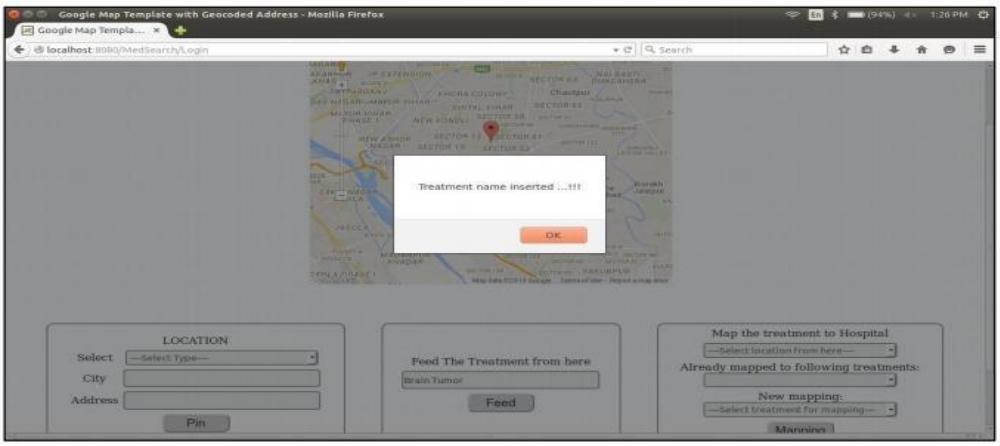


Fig: test case 6 success

Test Case 7:

Input: Hospital, Treatment in “Map the treatment to Hospital”

Expected Output: Mapping Successful.Actual Output: Mapping Successful.

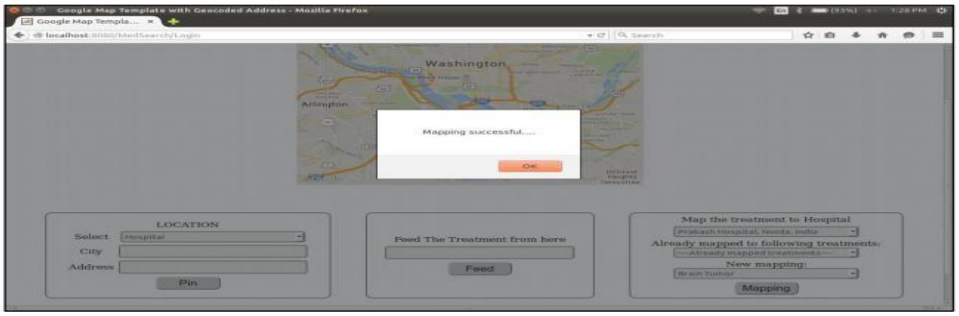


Fig: test case 7 success

Test Case 8:

Input: Breast Cancer

Expected Output: Series of results including Title, Information and Link.

Actual Output: Series of results including Title, Information and Link.

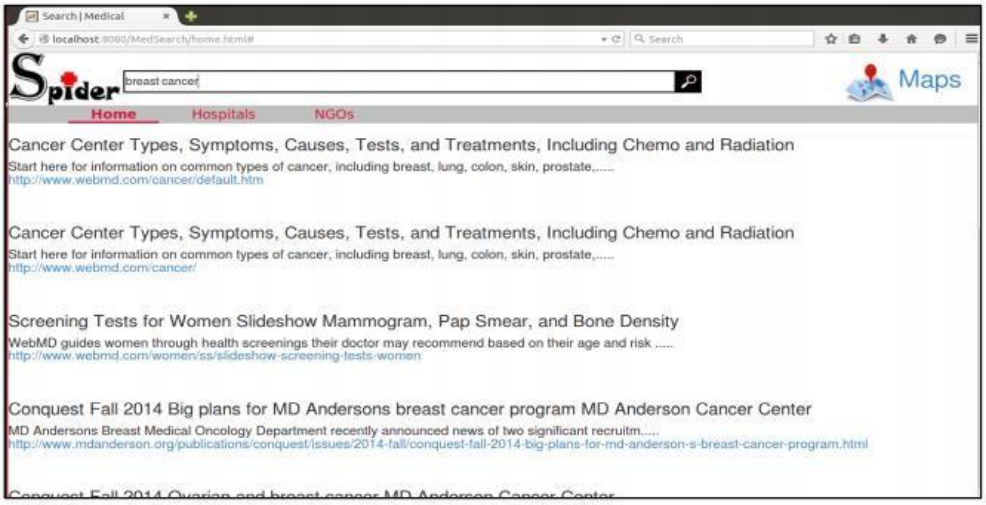


Fig: search results for breast cancer

Test Case 9:

Input: Cancer, Home tab

Expected Output: List of informative results.

Actual Output: List of informative results.

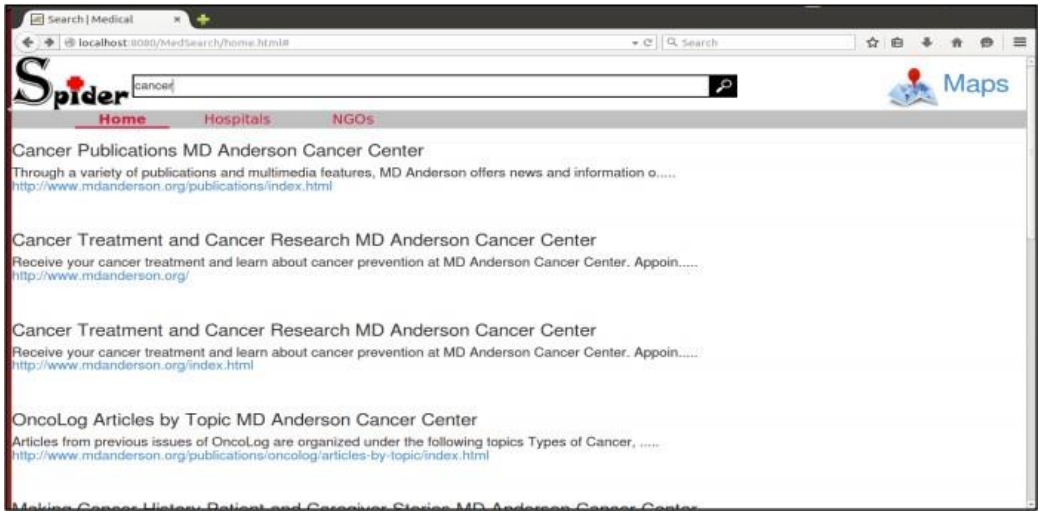


Fig: test case 9 success

Test Case 10:

Input: Cancer, Hospitals tab

Expected Output: List of sites which provide hospitals for the disease.

Actual Output: List of sites which provide hospitals for the disease.

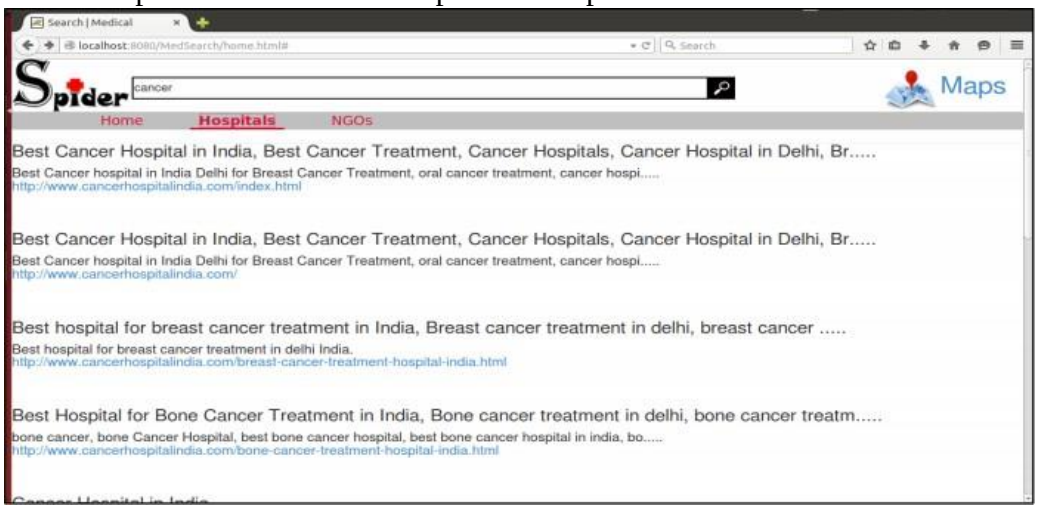


Fig: test case 10 success

Test Case 11:

Input: None

Expected Output: List of all Hospitals and NGOs.

Actual Output: List of all Hospitals and NGOs.

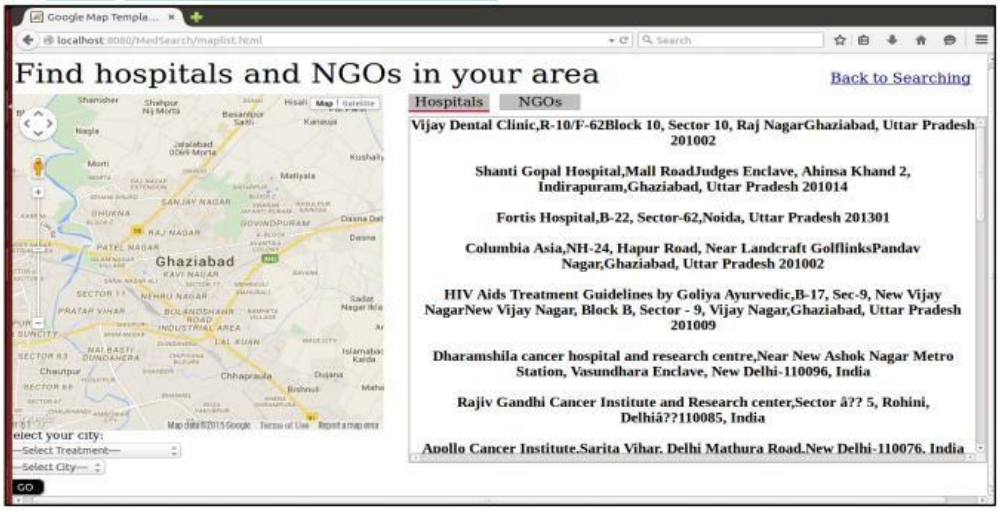


Fig: test case 11

Test Case 12:

Input: Eraser

Expected Output: End of Result.

Actual Output: End of Result.

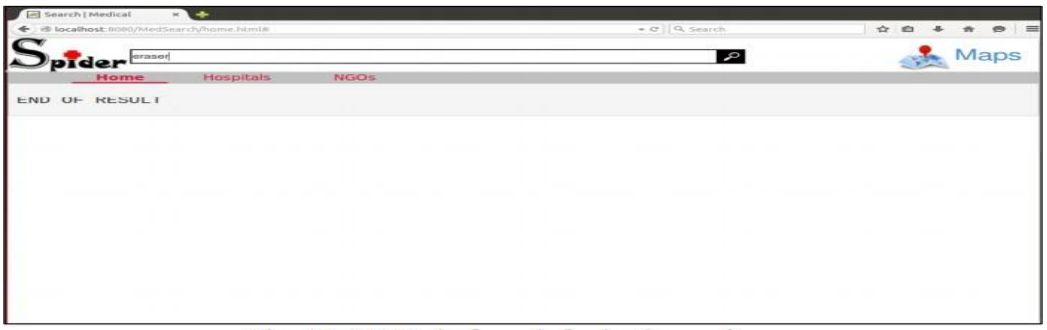


Fig: test case 12

RESULTS AND CONCLUSION

Results, Discussions, Conclusion and Future Scope of Study

The results describe the outcome of all the proposed objectives described previously and their successful completion. Discussions are made to enlighten all the outcomes we derive from this project.

Conclusion concludes all the work that was done in the project with their suitable outcomes, and future scope tells us about different fields where the system can be used in further and what all developments are possible with this developed project.

Results

- The web crawler is fully implemented and crawls almost ten pages per minute.
- The indexer is completely designed and implemented and works hand in hand with the crawler.
- The ranker is fully implemented with the help of HITS algorithm, which ranks the pages crawled for better ranked results.
- The auto-complete module and the map module are completely implemented.
- The front end is designed and implemented along with the integration of all the other modules of the project.

Discussions

In response to the massive market want, Health line, a well-liked internet program for medical info, came into existence in Gregorian calendar month 2005. Shortly thenceforth, Google announced its own medical internet program, Google Health, in might 2006. There are also many alternative medical internet search engines. whereas these systems have their own merits, they principally treat medical search in a lot of identical manner as ancient internet search. Medical search has many distinctive needs that distinguish itself from ancient Web search. a typical state of affairs within which an individual performs medical search is that he feels uncomfortable however is unsure concerning his actual medical issues. during this case, the searcher typically prefers to find out all types of information that's associated with his state of affairs. However, existing medical internet search engines square measure optimized for exactitude and concentrate their search results on many topics. This lack of diversity downside is aggravated by the character of medical web content. once discussing a medical topic, many medical internet sites use similar, however not identical, descriptions by paraphrasing contents in medical textbooks and analysis papers. Hence, search results provided by existing medical internet search engines usually contain a lot of

linguistics redundancy, that can't be easily handled by existing strategies for characteristic near-duplicate documents or result relevancy. to seek out helpful medical info, the searcher usually should undergo a large number of web content laboriously. this technique is employed to look the queries connected to medical field and supply relevant results like info concerning the unwellness, hospitals and NGOs providing treatment for identical. a further feature of the system is that the maps service through that user will pin the hospitals and NGOs of a selected town on the map and realize places wherever the treatment is obtainable.

Conclusion

People square measure thirsty for medical data. Existing internet search engines usually cannot handle medical search well as a result of they are doing not think about its special needs. usually a medical data searcher is unsure regarding his actual queries and unfamiliar language. Therefore, he generally prefers to create long queries, describing his symptoms and scenario in plain English, and receive comprehensive, orthogonal data from search results. The project presents a specialised internet computer program for medical data retrieval to beat these challenges. It will facilitate standard web users throughout the complete method of medical treatment. the look of this method takes into thought the distinctive needs of medical search. It supports queries written in plain English, provides relevant search results, and suggests connected Hospitals, NGOs on the maps, with correct ranking and annotation. These options square measure enticing to standard web users United Nations agency have very little medical information and square measure unfamiliar medical language. Consumer-centric medical search is that the main aim of this project. this method is employed to look the queries associated with medical field and supply relevant leads to 3 tabs specifically, Home, Hospitals and NGOs. It conjointly provides maps service through that user will opt for his treatment and town reminiscent of that the results of hospital's address and Ngo's address square measure displayed during a tab. The address are often stapled on the map to verify its existence. The system provides a more robust read of all the consolidated information that a user needs for medical treatment.

Future Study

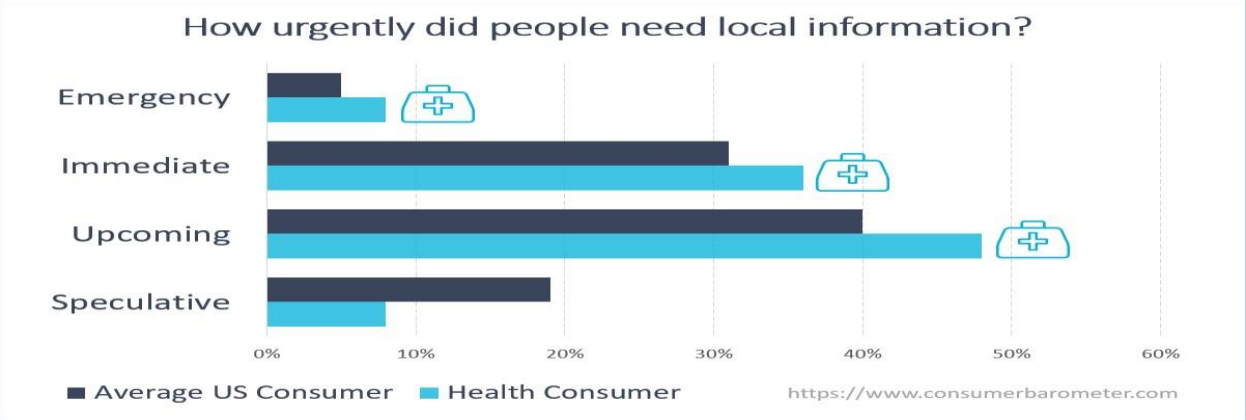


Fig: urgency of accessing health information

- The current project is entirely based on medical field, be it the queries, maps, or results. In the future, the scope of this search engine can be extended to other fields like:

- Tourism
- Education
- Transport, etc.

- In the current project, keyword-based searching is implemented, i.e., searching solely depends on the keywords that the user types in the text box. The results presented as an output may not concern the same interpretation. Thus, in the future, Artificial Intelligence can be used to give relevant results for context-based searching in the system.

- Word association can be implemented in this project in future.

- The present system crawls only websites and does not include files. In future, the crawler can crawl files in PDF, PPT, DOCX and many other formats. Also it can include images in search results.

- Auto correction is a feature which enables the user to retrieve the correct search results even for incorrectly spelled query.

- The present map module pins the hospitals and NGOs on the map but does not provide the direction to each the destination. In future, map direction module can also be implemented.

- IP Address based searching can be implemented in future.

- Voice based searching is searching the results via voice input instead of providing the result through the keyboard. The benefits of voice searching are that it makes input easier and faster

REFERENCES

[1] Skiena, S. S. (2008). Introduction to Algorithm Design. In The Algorithm Design Manual (pp. 3-30). Springer London.

[2] Coppin, B. (2004). Artificial intelligence illuminated. Jones & Bartlett Learning.

[3] Yoo, A., Chow, E., Henderson, K., McLendon, W., Hendrickson, B., & Catalyurek, U. (2005, November). A scalable distributed parallel breadth-first search algorithm on BlueGene/L. In Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference (pp. 25-25). IEEE.

[4] Shen, A. (2011). Algorithms and programming: problems and solutions. Springer Science & Business Media.

[5] Deo, N. (2004). Graph theory with applications to engineering and computer science. PHI Learning Pvt. Ltd..

[6] Anderson, S. S. (1970). Graph theory and finite combinatorics. Markham Pub. Co...

[7] Bodino, G. A. (1962). Economic applications of the theory of graphs (Vol. 1). Routledge.

[8] Behzad, M., & Chartrand, G. (1972). Introduction to the Theory of Graphs. Allyn and Bacon.

[9] Chen, W. K. (2012). Applied graph theory (Vol. 13). Elsevier.

[10] Nilsson, N. J. (1998). Artificial intelligence: a new synthesis. Morgan Kaufmann.

[11] Hersh, WilliamR,.(2002).Information Retrieval. Springer.

[12] Jackson, P. C. (1985). Introduction to artificial intelligence. Courier Corporation.

[13] Luger, G. F. (2005). Artificial intelligence: structures and strategies for complex problem solving. Pearson education.

[14] Minsky, M. (1995). Computation & intelligence: Collected readings. Luger, GF.

[15] Minton, S. (1988). Learning search control knowledge: An explanation-based approach (Vol. 61). Springer Science & Business Media.

[16] Luo, G., Tang, C., Yang, H., & Wei, X. (2008, October). MedSearch: a specialized search engine for medical information retrieval. In Proceedings of the 17th ACM conference on Information and knowledge management (pp. 143-152). ACM.