Assignment 4 CS 792/PHS 612
Design Document of Team 3

SMART MEDICAL SEARCHING ENGINE

Baoshi Sun
Xiao Meng
Armughan Hafeez

Revision: 2.0
April 10th 2016

Contents

1. Design purpose

In the era of Internet, general-purpose search engines have become a crucial portal for information access, and an efficient outlet for knowledge distribution. As previous research has pointed out, there was a minimum of 6.75 million health-related searches on the web every day in 2003 [1]; we can only imagine how incredible this number would be after more than a decade of expansion in terms of Internet coverage.

Nevertheless, searching for medical or health-related information with a general-purpose search engine, such as Google, could lead to unsatisfactory results. Previous study has reported an inaccuracy rate of 42% in diagnosis using Google as the search engine [2]. More specifically, the irrelevance or unsatisfactory factors come from two ways: first, general-purpose search engine does not take into consideration domain-specific priorities. This is an inherent shortcoming of the PageRank-style ranking schemes [3]. For example, a Wikipedia entry would likely get a higher ranking in the search results from Google than a more professional and informant page from a well-known hospital. Second, the query input from the users, i.e. the search keyword, is likely to be inaccurate or vague that would decrease the relevance and coverage of search results. Therefore, our design purpose is to improve the relevance of search results for health-related queries while keeping the well-received user interface of a general-purpose search engine.

To achieve our design goals, we build a web-based interface and empower the website with Google Scraper [4] to efficiently crawl original results from Google. We leverage expert knowledge in a specific and specialized discipline of medicine, to re-rank and refine the search results returned by Google. We also leverage SNOMED, NLM and MeSH to expand search keywords and return more relevant results to re-rank. To provide users with more diverse functionalities that can be seen in modern search engines, we also implement a medical news feed, a location-based service (LBS) map and a discussion forum for users to connect with each other. Details about the techniques and implementation are further elaborated in the section of Architecture overview.

2. Use cases

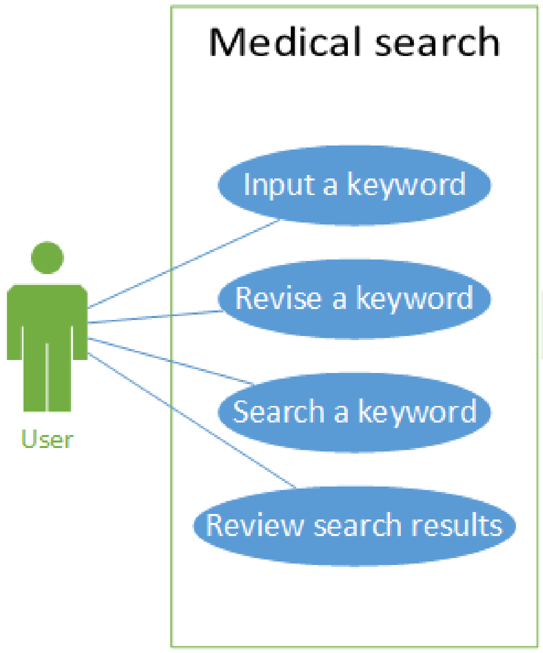
2.1 Target user group

Our system is targeted at users that often use general-purpose search engines for medical or health-related inquiries. More specifically speaking, we envision our targeted users to be: patients, patients' relatives, nursing homes or personals, and community service clients, etc.

2.2 Use cases

Although our system is designed for users with a need of medical information about a specific disease, the functionality goes beyond refined medical search; it also works for use cases, e.g., where users want to join a discussion forum to know other patients facing the same health problems. The only assumption we make about the users is that they are familiar with the usage of a web-based portal of a modern search engine. We present the usage of our system with UML diagrams for a clear illustration. The figures of interfaces and usage can be found in the Appendix.

- Searching using the search box (illustrated in Figure 1). Users can simply input the search keyword, “gall stone” into the dialog box, and press enter to issue the query. The refined search results will be presented.



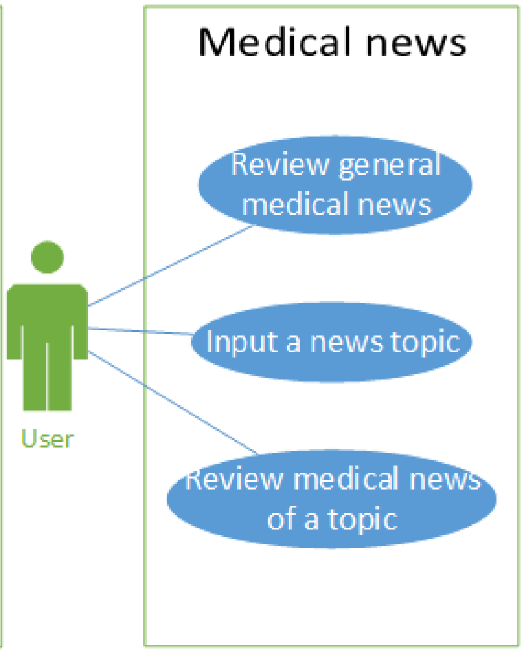


Figure 1 Use case diagram for medical search. *Figure 2 Use case diagram for medical news.*

- Checking breaking medical news (illustrated in Figure 2). Users can browse medical news of the day by clicking the “newspaper” button below the search box. The general medical news is extracted from Med Page Today (http://www.medpagetoday.com/), where medical news is aggregated and refreshed daily. Or, the user can explicitly input a topic in the search box, then click on the “newspaper” button to review medical news about a certain topic.
- Location-based medical service map (illustrated in Figure 3). Users can browse the map with drug stores and health service providers highlighted, simply by clicking the “map” button below the search box. This way, users can efficiently find and locate healthcare products near him/her when in need or in case of an emergency.

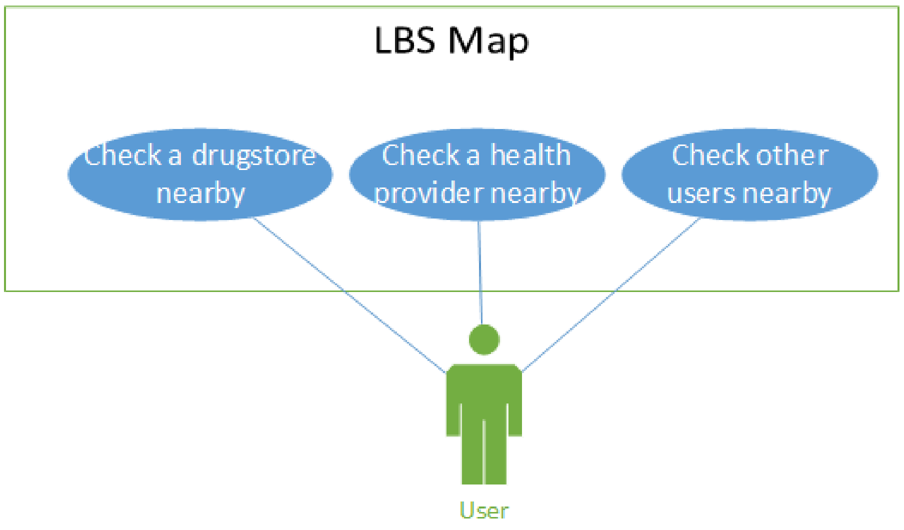


Figure 3 Use case diagram for LBS map.

- Users discussion forum (illustrated in Figure 4). Here we also implement a discussion forum to facilitate communication and social events among users. This way, users with a similar disease can interact with each other, exchange relevant information, and even form off-line groups to help each other dealing with a disease. Users can simply click on the "forum" button below the search bar, and sign up for a free account. After that they can start a discussion by creating a thread, and reply or comment accordingly. They can also pin tags to the discussions so that they can indicate the general topic of a discussion they initiated. Also, searching in the forum is supported so that users can search for relevant discussions to join.

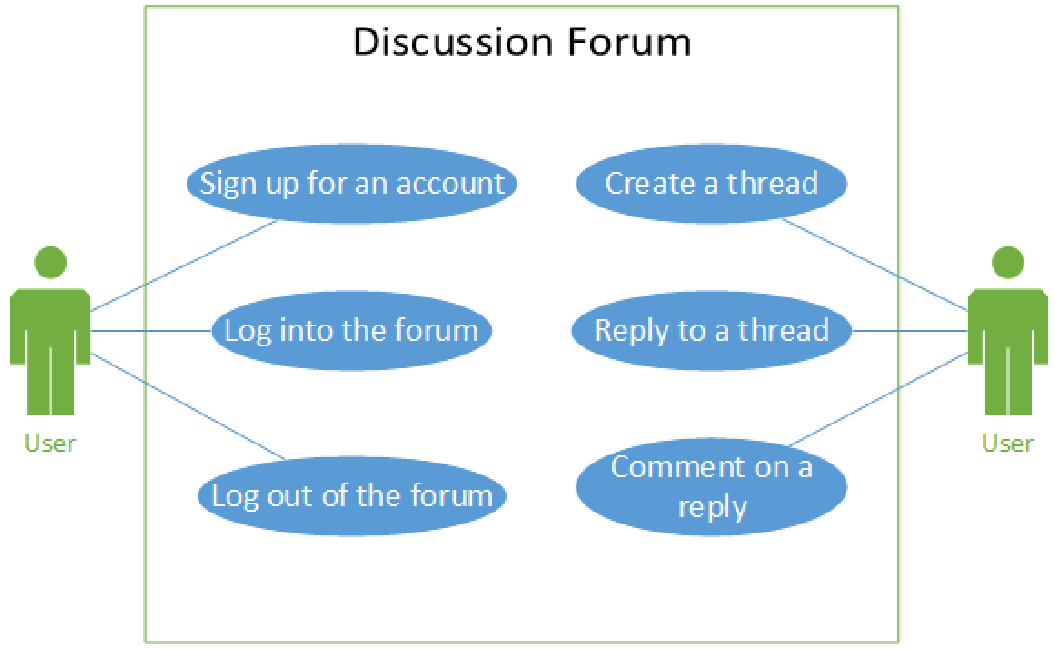


Figure 4 Use case diagram for discussion forum.

3. Data sources and format

Here we present the details of the data sources and format in our system. For a clear view of the data, we categorize them according to their module in the system. We have four modules concerning data formats: Google scraper, expert knowledge base, page ranker, and query generator.

Table 1 Data formats in Google scraper.

Parameter	Data definition	Data type	Note
query	The search keywords input by the user.	String	
engines	The search engines to scrape from, including Google, Bing, Yahoo, Yandex, Baidu and Duckduckgo.	List of string	
pages	The number of webpages for a given query.	Integer	
method	Two methods of search are available. For "basic" method, it fetches results from web search; for "image" method, it fetches results from image search.	String	
gsobj	The Google Scraper object that contains the fetched search results for a given query.	Object	
results	The instance of gsobj that contains the results.	Gsobj	
baserank	The baseline value of ranking for a webpage.	Integer	Initialized as 1000.

Table 2 Expert knowledge collected for our system.

Data source	Data definition	Credibility	Note
sources_armughan	Expert opinions on credible medical webpages put together by Dr. Armughan and his colleagues.	High	Located in expertise.py.
sources_huffington	List of credible medical websites published by the Huffington Post [5].	Medium	N/A
sources_umich	List of credible medical websites published by the Health Services at the University of Michigan [6].	High	N/A
sources_jhu	List of credible medical websites published by the School of Medicine at Johns Hopkins University [7].	High	N/A

Table 3 Data formats in medical page ranker.

Parameter	Data definition	Data type	Note
domain	The domain name of a webpage, e.g., ".org".	String	Located in pageranker.py.

link	The url of the webpage, e.g., https://www.ncbi.nlm.nih.gov/pubmed/.	String	N/A
linktype	The type of a webpage that Google Scraper returned, mostly to indicate whether a webpage is an advertisement page or not. If the value is “ads_main”, then the page will be classified as an ad.	String	N/A
oldrank	The original rank of the webpage returned by Google and collected by Google Scraper.	Integer	N/A
title	The title of the webpage, e.g., “Gallstone – Wikipedia, the free encyclopedia”.	String	N/A
newrank	The optimized rank of the webpage after the re-ranking process.	Integer	N/A

Table 4 Data formats in query generator.

Parameter	**Data definition**	**Data type**	**Note**
QC_MAX_KEYWORDS	The maximum number of keywords generated.	String	Initialized as 10.
query	Original search keywords input by the user, e.g., “gallstone”.	String	
kw_list	The list of words split from the string of query.	List of string	
newkw_list	The expanded list of words after processing each word in kw_list with the query expansion techniques.	List of string	
newquery	The new query assembled from newkw_list.	String	
term_snomed	The terminologies collected from SNOMED API calling.	List of String	
term_nlm	The terminologies collected from NLM API calling.	List of String	
term_mesh	The terminologies collected from MeSH API calling.	List of String	

4. Architectural overview

4.1 System overview

Our system is implemented in Python 3, and deployed on top of an Amazon EC2 instance in the cloud. Overall, it consists of three layers, and mostly four modules to facilitate our design purpose. The overview architecture is shown in Figure 5.

- Front-end: this layer consists of the web-based user interface. To provide a familiar experience to users, we implemented a Google-style search box, with three extra buttons for medical news feed, LBS map and a discussion forum.
- Mid-tier: this layer consists of the two major innovations we made to the original search engine: a Query Generator and a Page Ranker, which enable the refinement of page ranking and the expansion of query scope.
- Back-end: this layer consists of the supporting modules that enable the top two layers, including a data ingest module and an index & storage module.

Major technologies used for our system can be categorized as follows:

- Web server: Nginx, Uwsgi [8], Postfix [9]
- Front-end: Bootstrap [10], AngularJS [11], jQuery [12]
- Back-end: Python/Flask [13]

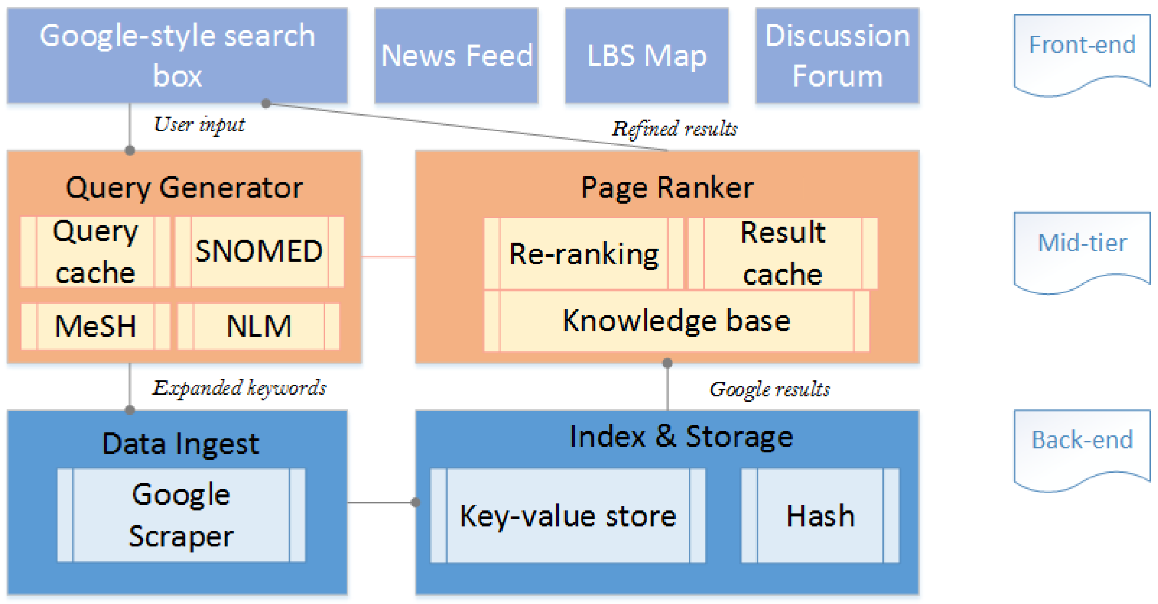


Figure 5 System overview: layers and modules.

Important development tools and frameworks used for different modules in our system are listed based on the modules they work for:

- Data ingest module: Google Scraper [4]
- Query generator module: SNOMED/ICD-10
- Discussion forum module: Flarum [14]
- News feed module: Feedparser [15]
- LBS map module: Google Maps APIs [16]

4.1.1 Query Generator module

This module expands the user-specified search keywords using comprehensive medical terminology products including SNOMED [17], MeSH [18] and NLM [19]. For instance, when user inputs a search keyword "gallstone", our query generator will go through the terminologies in the three vocabularies to properly expand this keyword, and produce a new expanded keyword that unions all the proper expansions.

Currently we're using a web-based wrapper tool of the APIs [20], and keep the collected terminologies for reasonable expansion manually for the two cases: “gall stone” and “HIV”. This process can certainly be automated and integrated into the query generator module to support more queries, more cases and more sophisticated query expansion schemes.

4.1.2 Page Ranker module

Page Ranker refines search results based on medical knowledge (i.e., lists of credible and reliable medical websites) that we've collected.

Inside the Page Ranker module, we implement a knowledge base to store the knowledge, and a result cache to store the original search results by Google. The re-ranking schemes work in three steps:

- Advertisement elimination: if a website is an advertisement, Google will return it with a label indicating that it's an advertisement page. Once we detect that label, we'll significant lower its rank.
- Domain-based re-ranking: websites from education institutes, government agencies and organizations mostly use specific domain names (“.edu”, “.gov” and “.org”). For pages with these three domain names, we promote their rank to appreciate the credibility of the institutes, agencies and organizations.
- Knowledge-based re-ranking: we re-rank the pages comparing against lists of credible medical websites, i.e., collected knowledge. We have four sources of knowledge that we’ve used for this step, and if any page returned by Google Scraper matches a record in our knowledge base, we will significantly promote its rank to make sure it will be included in the refined results:
    1. Expert opinion (aggregated by Dr. Hafeez)
    2. The Huffington Post [5]
    3. University of Michigan [6]
    4. Johns Hopkins University [7]

After this, the results we present to the users will likely be different from what original Google returned, and cover more related terminologies (currently for the two cases we’ve implemented).

4.1.3 Data Ingest module

The data ingest module is essentially the Google Scraper enabled crawler. By using Google Scraper [4], we can support asynchronous mode where we can scrape with thousands of requests per second as long as the search engine does not actively block our IP address. Once the user inputs a search keyword, after the Query Generator processes the query, this module will invoke Google Scraper to fetch the results from Google regarding the expanded query keywords.

4.1.4 Index and Storage module

All the crawled results from the data ingest module will be then stored and indexed in the index & storage module. Right now we're handling it using a SQLite 3 database, but this can definitely be augmented using more sophisticated indexing schemes and a high-performance key-value data store, such as RocksDB [21] brought forward by Facebook Engineering for their internal usage.

The ER diagram is shown in Figure 6 for the two major tables in the storage module in order to capture the data elements in the back-end.

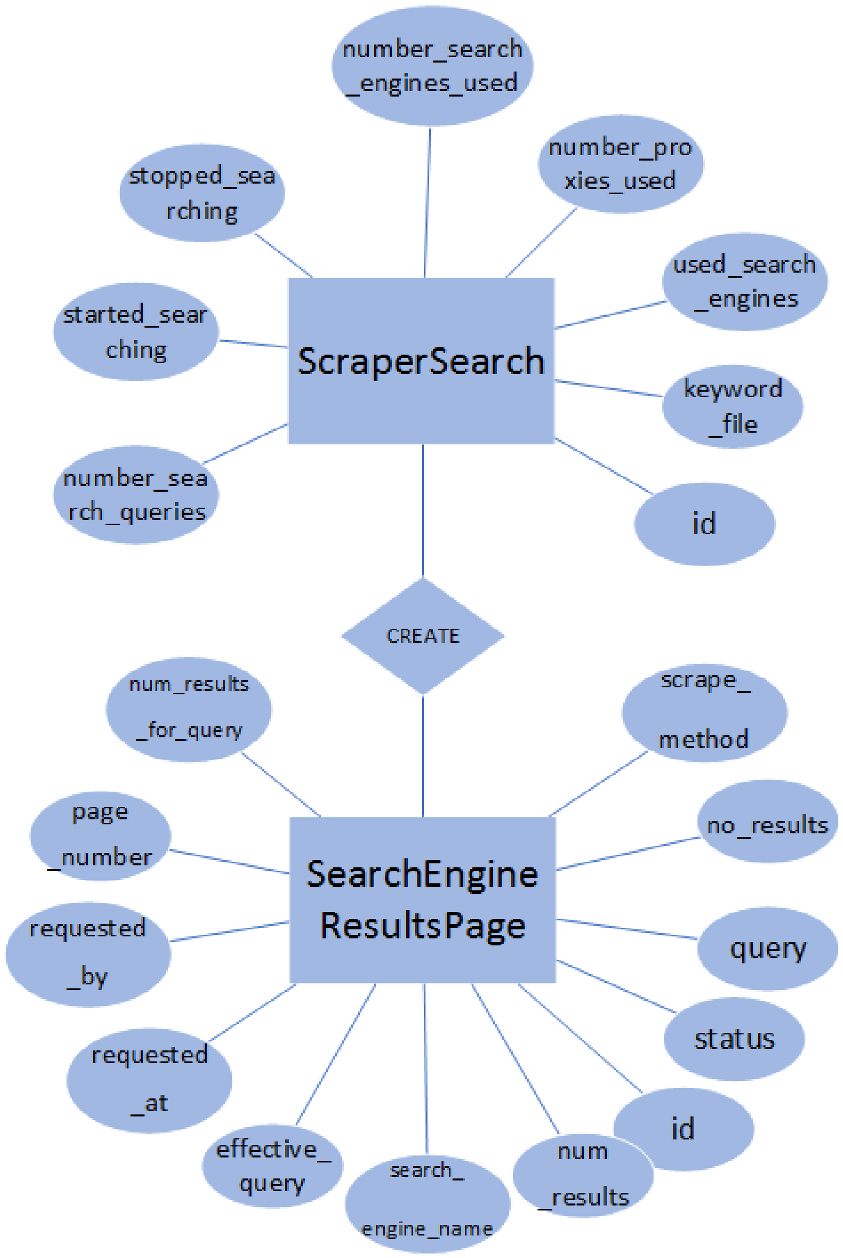


Figure 6 ER diagram for back-end SQLite database.

4.1.5 Front-end modules

The front-end is essentially the web-based user interface. To facilitate the search experience, we implemented a Google-style search box using Bootstrap, AngularJS and JQuery for a modern responsive web design.

*4.1.5.1 **Medical news feed***

To facilitate the medical news feed feature, Feedparser package is used to fetch RSS news sources. In order to acquire high quality medical news, we tried quite a few RSS resources and finally selected three feeds.

- General medical news: http://www.medpagetoday.com/rss/Headlines.xml
- Specific healthcare topics, e.g. woman, man, kids, diet, etc.: http://www.medicinenet.com/rss/general/
- News about diseases, e.g. allergy, Alzheimer's Disease, Arthritis, etc.: http://www.news-medical.net/tag/feed/

*4.1.5.2 **LBS map***

By taking the advantage of Google Maps APIs, we show a roadmap near UW. Some medical facilities within 5km radius, such as hospitals, pharmacies, clinics and dentists, are marked out by calling the Google PlacesService.

*4.1.5.3 **Discussion forum***

With regard to the forum, we adopted and deployed an open-source forum software called Flarum (https://github.com/flarum/flarum), built with PHP and Mithril.js. However, in order to enable the email function, such as user account activation and password retrieval, Postfix was also required.

In the Appendix, we provide a more detailed overview of the components and functionalities that we implemented using screenshots of the demo, to provide a more concrete guide for our system.

4.2 UI design overview

The user interface flow is illustrated as a storyboard diagram in Figure 7 to present a clear view of the transition between different interfaces of our system.

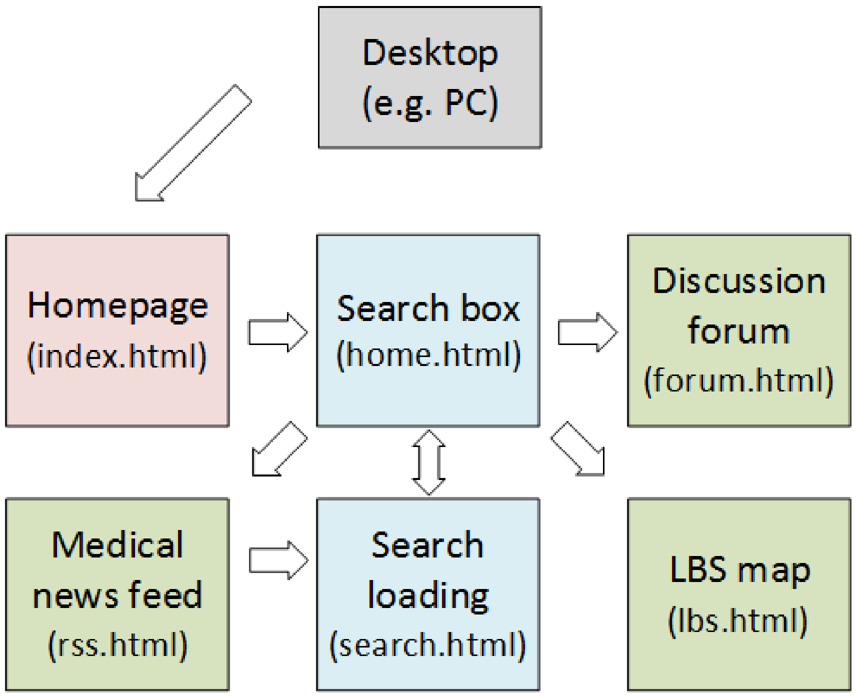


Figure 7 UI flow diagram (storyboard) of our system.

5. Solution evaluation

5.1 Evaluation

5.1.1 Portability evaluation

For portability tests, we aim at testing the portability of our system. Specifically speaking, we want to study the portability of our web-based interface to ensure it is user-friendly and adaptive to different browsing terminals that users could potentially be using in daily life.

- Browse and issue sample queries via our web-based interface on Linux, OS X and Windows computers to test operating system portability for our system;
- Browse and issue sample queries via our web-based interface on browsers based on different kernels to ensure it displays and functions properly. We narrow down our selections to four most common browser kernels, and choose one representative for that kernel: Trident-based Microsoft Internet Explorer, Gecko-based Mozilla Firefox, Webkit-based Apple Safari, and Blink-based Google Chrome.
- Browse and issue sample queries via our web-based interface on mobile devices including Android mobiles, iPhones, Android tablets and iPads, to ensure that our system works as expected on mobile devices as well.

Please note that the final tests could be interleaved. For example, a test run on an iPhone with built-in Safari browser would suffice for both Step 2 and Step 3.

Preliminary evaluation results for portability evaluation have shown that our system works well on all types of devices, mainly because of the web technologies that we adopted already take portability into their consideration in their implementations, which helps to guarantee the portability of our own system.

5.1.2 Accuracy evaluation

The accuracy study focuses on the quality of the response. The detailed evaluation shall proceed as follows:

- To measure the quality of results reported by our search system, a baseline is required to compare with. The widely accepted approach to generate a baseline through a manual process with expertise. In our case, the baseline would be the results manually selected by a medical expert according to his or her professional evaluation of the relevance of the webpage given the search keywords. We set the number of webpages (i.e., results) reported to be the same in both the baseline and our system.
- Then we collect the results presented by our system, and compare them against the baseline collection created in Step 1.
- We measure the precision ratio ρ = (the number of intersected results from two collections / the number of results collected in one collection). Generally speaking, the higher the ρ, the better the accuracy.

Preliminary evaluation results show that in comparison with Google, our search results differ greatly from what Google returns for “gallstone” and “hiv” (top 5 results comparison shown in Table 5 for misspelled keyword “galstone”). Since we’ve already incorporated expert opinions (by Dr. Hafeez) into our implementation, in order to carry out a credible and unbiased accuracy evaluation, we would need other medical experts to help as the baseline expertise.

Table 5 Top 5 search results comparison: our system vs. original Google.

Smart Medical Searching	Google
Bile Duct Stones & Gallstones | Pancreas Diseases (from Digestive Disease Center in Medical Univ. of South Carolina)	Gallstones Picture, Types, Causes, Risks, Symptoms (from WebMD)
What is the gallbladder and the bile duct (from Department of Surgery in USC)	Gallstones: Causes, Risk, & Treatment (from Healthline)
Laparoscopic Gallbladder Surgery for Gallstones (from WebMD)	Gallstones (from Mayo Clinic)
Bile Duct Stones (from Health System in Univ. of Michigan)	Gallstone (from Wikipedia)
Bile duct obstruction (from MedlinePlus Encyclopedia by U.S. National Library of Medicine)	Gallstones: Symptoms, Causes & Treatment (from emedicinehealth)

5.1.3 Stress evaluation
For stress tests, we simulate a large number of client connections to the server, and issue queries to our system concurrently to stress test the that capacity of our system. The purpose is to ensure that it works for massive concurrent queries, and responds with reasonable latencies for all simulated use cases. For “gallstone”, the simulation process works as follows:

- On a client desktop, write a Python script using the threading package to manage N connections to our server, where N equals the predefined number of concurrent connections that we plan to test;
- On the client side, execute the script, and on the server side, use timestamps to collect the response time for the queries issued;
- Use aggregation analysis on the response time collected for each search processing, and report statistical results of the analysis, such as average response time, response ratio (the number of queries responded within a time threshold t / the number of all queries issued), etc.

5.1.4 User acceptance evaluation
For user acceptance evaluation, we plan to do two lines of work to comprehensively understand how users think of our system.

- Online survey: we would add an extra button on the website indicating that a short online survey is available for the users. We could use Google Forms for collecting and analyzing the surveys, or we could seek professional help from SurveyMonkey for more powerful surveys.
- In-house user acceptance testing: this requires professional instructions to ensure a valid and valuable outcome. Crucial steps include: planning and outlining, designing test cases, selecting the testing team, executing test cases and documenting the results, bug fixing, final rating and signing off.

5.2 Future work

1. In our current implementation, Google scraper handles the index and storage using a lightweight SQLite 3 database. In future work, we can extend our system using a more sophisticated storage design (e.g., a high-performance key-value store like RocksDB [21]) to speed up the GET and PUT primitives. Another extension would be to deploy our system in a distributed setting exploiting a MapReduce-style processing, where computation can be delegated to a number of dedicated servers. The distributed setting would be feasible when our system is made public to a considerable number of users after commercialization.

2. As we mentioned before, instead of using the SIM+N tool for query expansion, we can use isolated independent API calls to the terminology vocabularies, and integrate them into our back-end implementation to provide automated query expansion, and it will

open up opportunities for plugging in more fine-grained and optimized query expansion techniques.

3. Another important part of our system for future improvement is the knowledge collection. As Dr. Chen pointed out, expert opinions often disagree or even contradict with each other, therefore it's crucial to manage the expert knowledge wisely. One path to purse is to integrate more advanced reasoning and classification techniques into the underlying logic. With more reasoning about the vocabularies and websites, we can ensure better refinement of the search results.

Acknowledgement

Many thanks to Professor Helen Chen for her insightful advice and supervision through the term, and thanks to the great open-source initiative that brings about so many wonderful software and packages that we can use to help build our system.

Appendix

References

[1] Eysenbach, Gunther, and Kohler. "What is the prevalence of health-related searches on the World Wide Web? Qualitative and quantitative analysis of search engine queries on the internet." AMIA annual symposium proceedings. Vol. 2003. American Medical Informatics Association, 2003.
[2] Tang, Hangwi, and Jennifer Hwee Kwoon Ng. "Googling for a diagnosis—use of Google as a diagnostic aid: internet based study." Bmj 333.7579 (2006): 1143-1145.
[3] Page, Lawrence, et al. "The PageRank citation ranking: bringing order to the web." (1999).
[4] Google Scraper. https://github.com/NikolaiT/GoogleScraper
[5] How to Find the Best Medical Information Online. http://www.huffingtonpost.com/jim-t-miller/online-medical-information_b_3667454.html
[6] Recommended Health Websites. https://www.uhs.umich.edu/websites
[7] Finding Reliable Health Information Online. http://www.hopkinsmedicine.org/johns_hopkins_bayview/patient_visitor_amenities/community_health_library/finding_reliable_health_information_online.html.
[8] Uwsgi with Nginx: a web server. https://wiki.archlinux.org/index.php/Uwsgi
[9] Postfix: an open-source email transfer agent. http://www.postfix.org/
[10] Bootstrap: an open-source collection of tools for web. http://getbootstrap.com/
[11] Angularjs: an open-source JavaScript MVW framework. https://angularjs.org/
[12] JQuery: a cross-platform JavaScript library for HTML. https://jquery.com/
[13] Flask: a python microframework for web. http://flask.pocoo.org/
[14] Flarum: an open-source forum software. http://flarum.org/
[15] Feedparser: a python package for feed parsing. https://github.com/kurtmckee/feedparser
[16] Google Maps APIs. https://developers.google.com/maps/
[17] SNOMED CT. http://www.ihtsdo.org/snomed-ct
[18] MeSH. https://www.nlm.nih.gov/mesh/
[19] NLM Classification 2015. https://www.nlm.nih.gov/class/
[20] The SIM+N Tool. http://snomed.alptown.com/
[21] RocksDB. http://rocksdb.org/

Screenshots

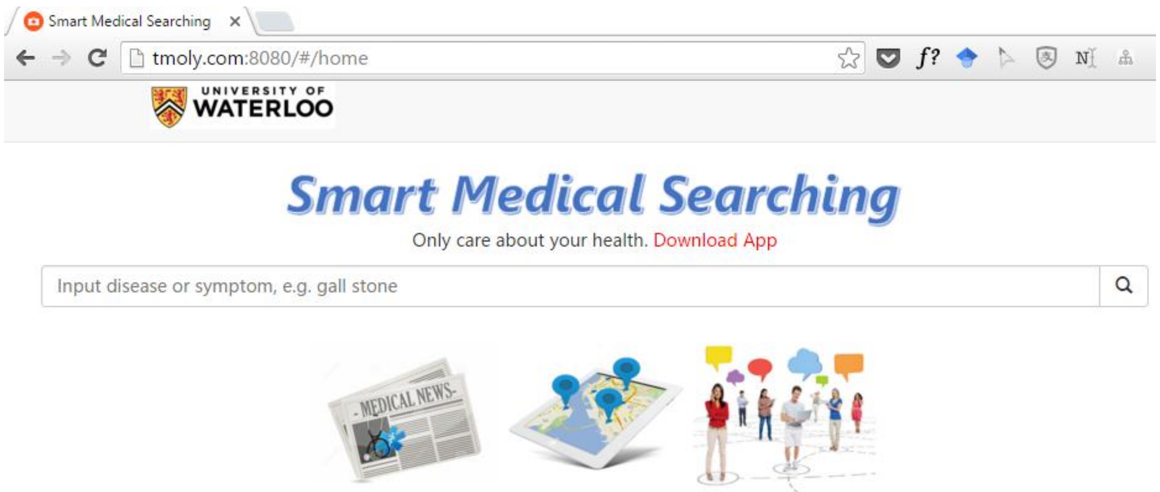
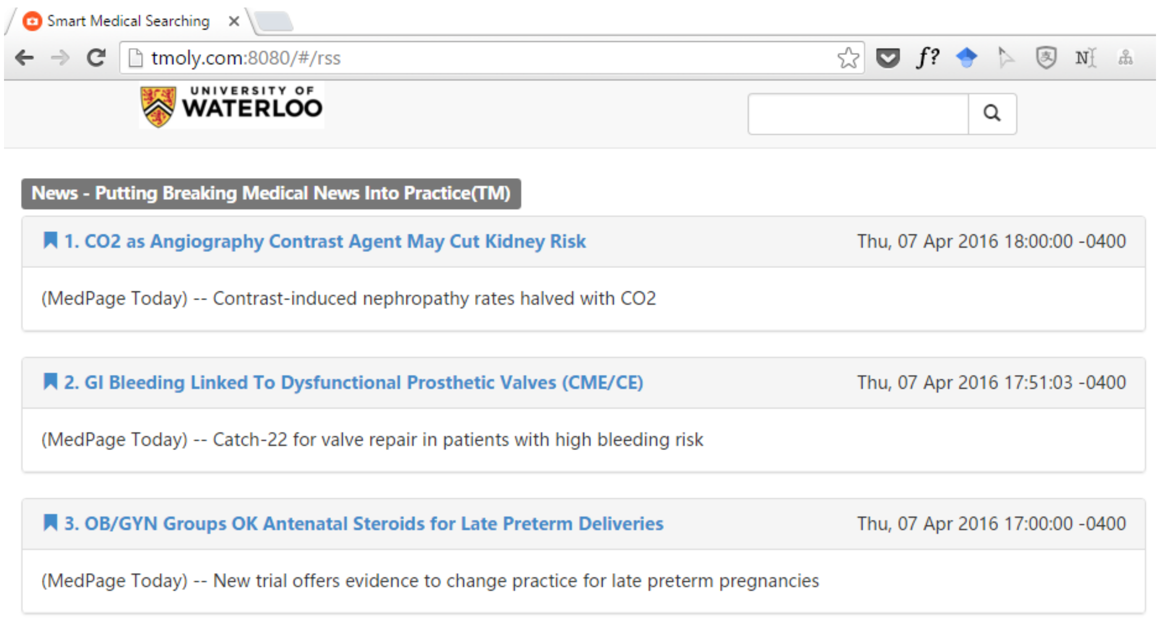


Figure 8 Main interface for searching.



News - Putting Breaking Medical News Into Practice(TM)

1. CO2 as Angiography Contrast Agent May Cut Kidney Risk — Thu, 07 Apr 2016 18:00:00 -0400

(MedPage Today) -- Contrast-induced nephropathy rates halved with CO2

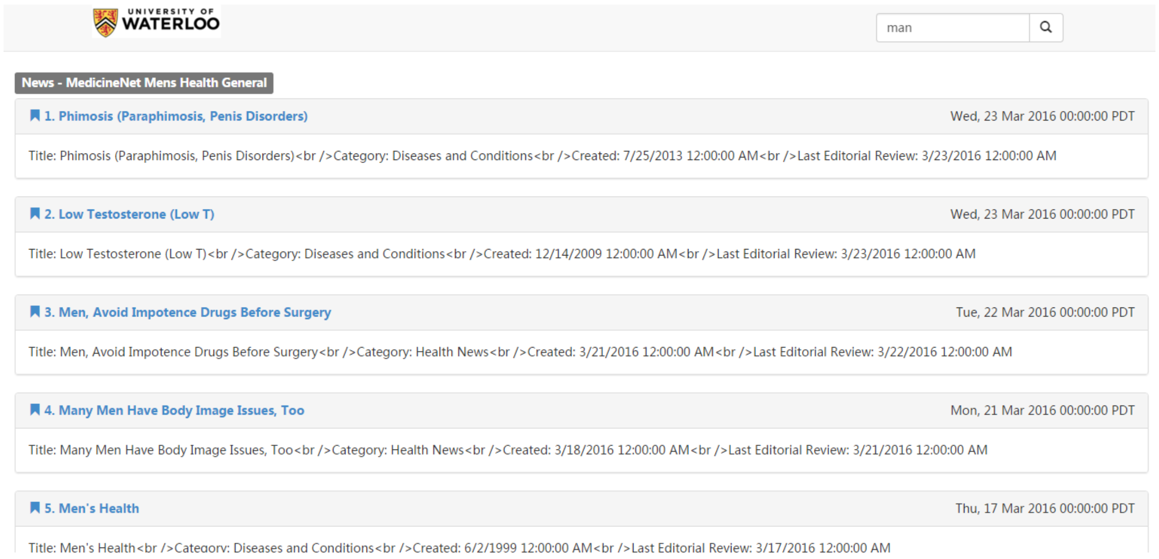
2. GI Bleeding Linked To Dysfunctional Prosthetic Valves (CME/CE) — Thu, 07 Apr 2016 17:51:03 -0400

(MedPage Today) -- Catch-22 for valve repair in patients with high bleeding risk

3. OB/GYN Groups OK Antenatal Steroids for Late Preterm Deliveries — Thu, 07 Apr 2016 17:00:00 -0400

(MedPage Today) -- New trial offers evidence to change practice for late preterm pregnancies

Figure 9 General medical news feed.



News - MedicineNet Mens Health General

1. Phimosis (Paraphimosis, Penis Disorders) Wed, 23 Mar 2016 00:00:00 PDT

Title: Phimosis (Paraphimosis, Penis Disorders)<br />Category: Diseases and Conditions<br />Created: 7/25/2013 12:00:00 AM<br />Last Editorial Review: 3/23/2016 12:00:00 AM

2. Low Testosterone (Low T) Wed, 23 Mar 2016 00:00:00 PDT

Title: Low Testosterone (Low T)<br />Category: Diseases and Conditions<br />Created: 12/14/2009 12:00:00 AM<br />Last Editorial Review: 3/23/2016 12:00:00 AM

3. Men, Avoid Impotence Drugs Before Surgery Tue, 22 Mar 2016 00:00:00 PDT

Title: Men, Avoid Impotence Drugs Before Surgery<br />Category: Health News<br />Created: 3/21/2016 12:00:00 AM<br />Last Editorial Review: 3/22/2016 12:00:00 AM

4. Many Men Have Body Image Issues, Too Mon, 21 Mar 2016 00:00:00 PDT

Title: Many Men Have Body Image Issues, Too<br />Category: Health News<br />Created: 3/18/2016 12:00:00 AM<br />Last Editorial Review: 3/21/2016 12:00:00 AM

5. Men's Health Thu, 17 Mar 2016 00:00:00 PDT

Title: Men's Health<br />Category: Diseases and Conditions<br />Created: 6/2/1999 12:00:00 AM<br />Last Editorial Review: 3/17/2016 12:00:00 AM

Figure 10 Medical news feed with topic "man".

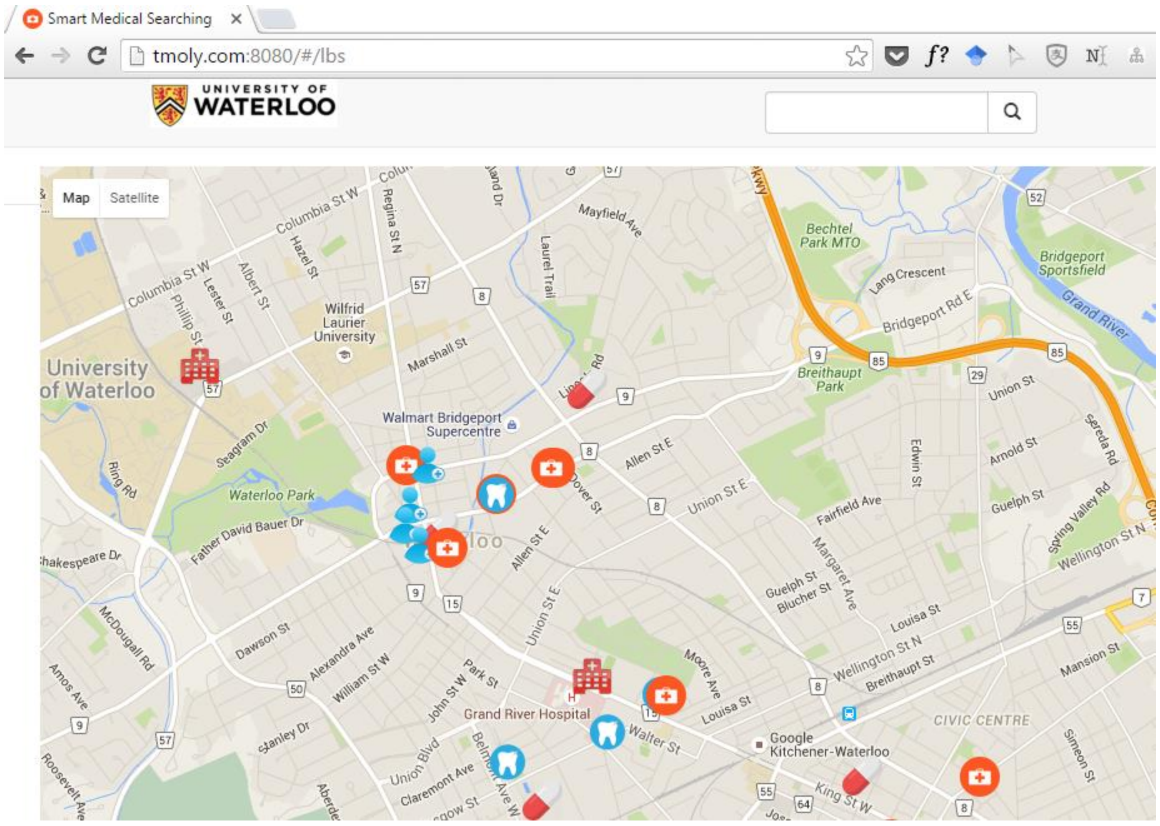


Figure 11 LBS map.

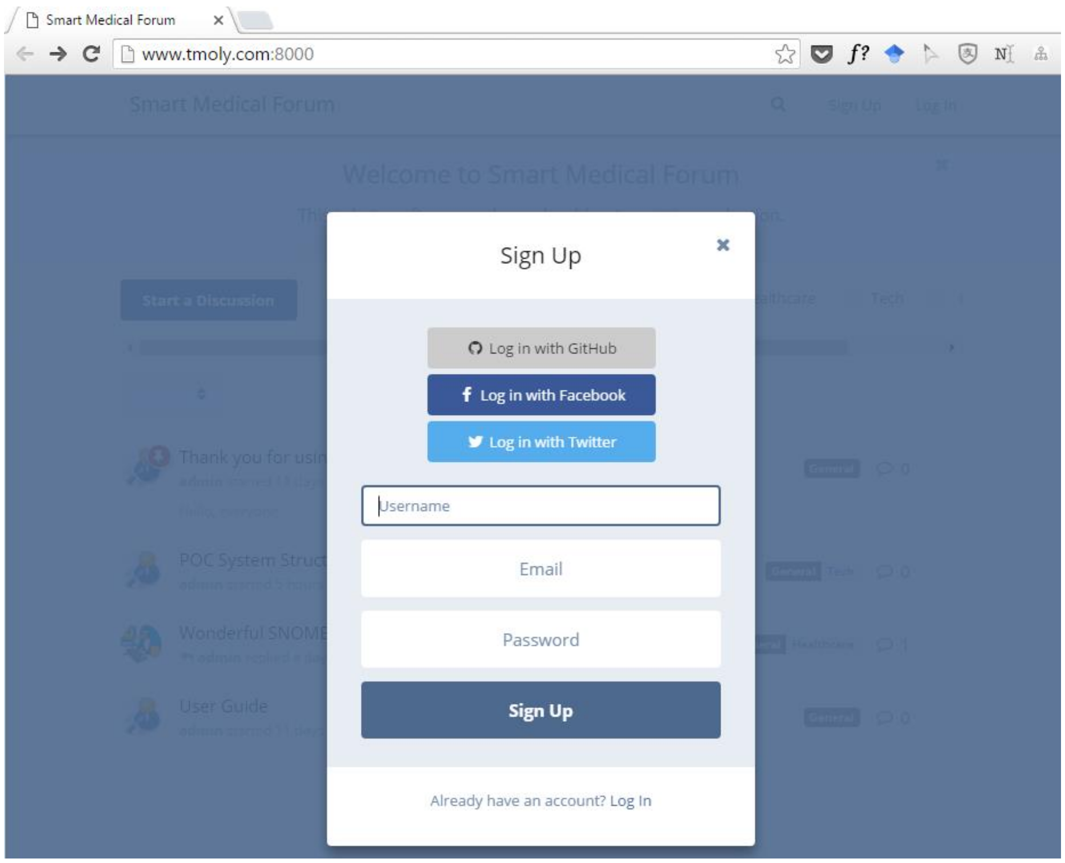


Figure 12 Sign up for a free account at the forum.

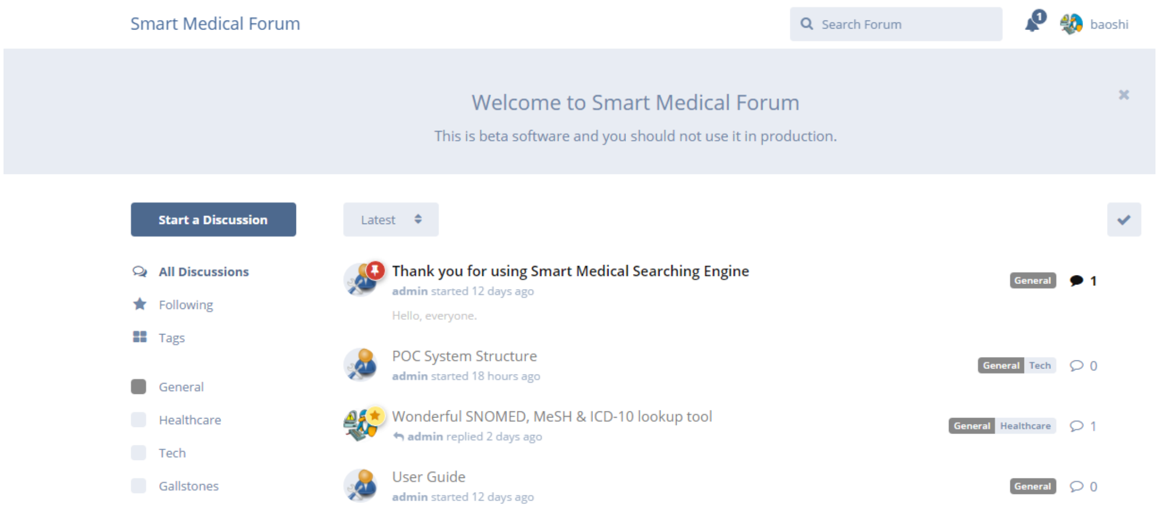
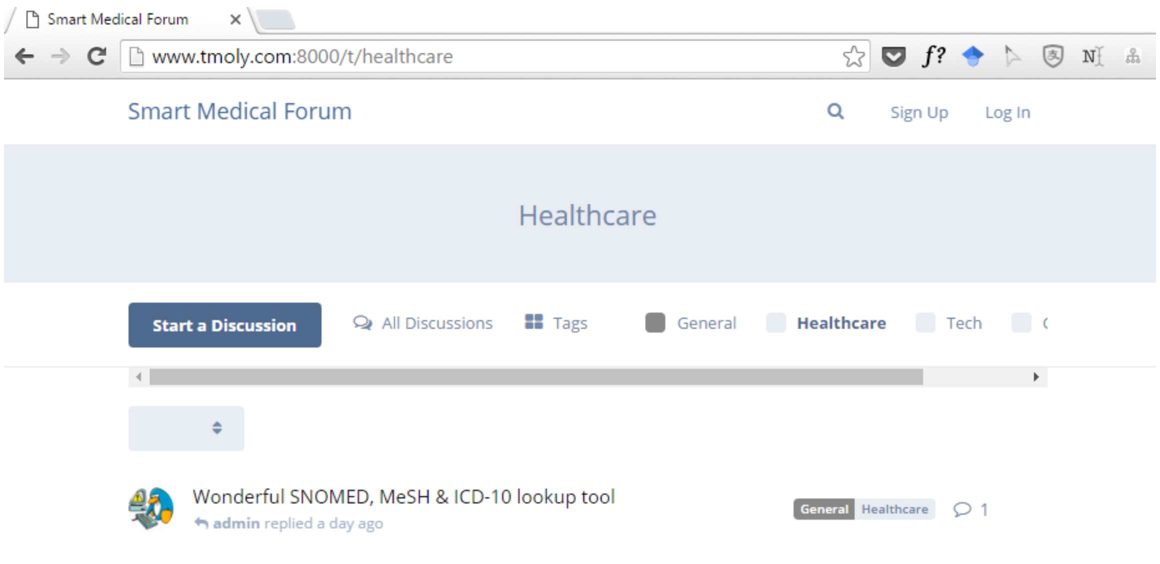

Smart Medical Forum
Search Forum
baoshi
Welcome to Smart Medical Forum
This is beta software and you should not use it in production.
Start a Discussion
Latest
All Discussions
Following
Tags
General
Healthcare
Tech
Gallstones
Thank you for using Smart Medical Searching Engine
admin started 12 days ago
Hello, everyone.
POC System Structure
admin started 18 hours ago
Wonderful SNOMED, MeSH & ICD-10 lookup tool
admin replied 2 days ago
User Guide
admin started 12 days ago

Figure 13 Main interface for discussion forum.



Smart Medical Forum

www.tmoly.com:8000/t/healthcare

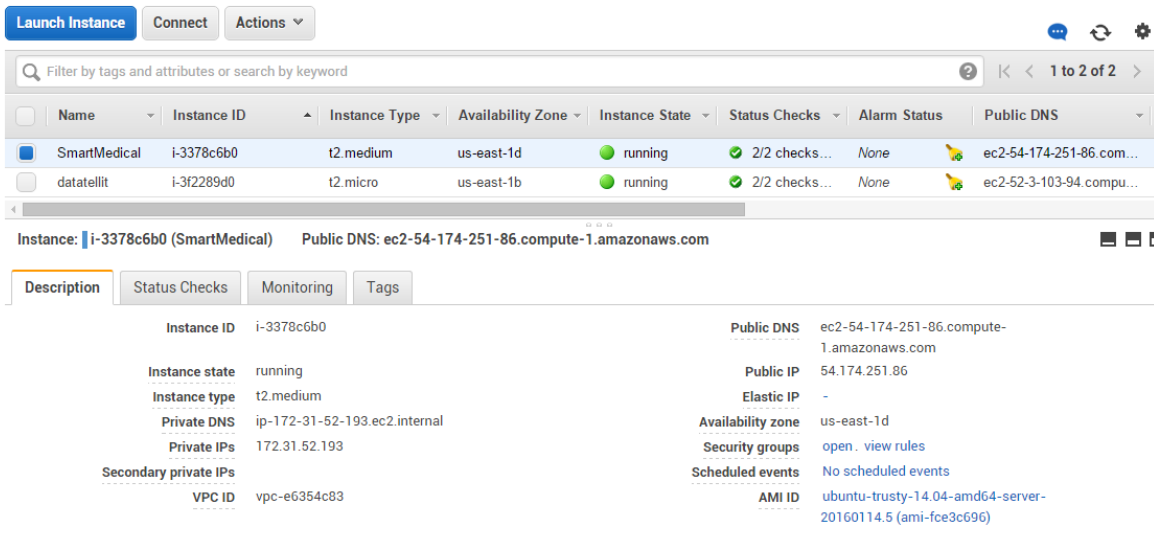
Smart Medical Forum Sign Up Log In

Healthcare

Start a Discussion All Discussions Tags General Healthcare Tech

Wonderful SNOMED, MeSH & ICD-10 lookup tool
admin replied a day ago
General Healthcare 1

Figure 14 Discussion forum showing all threads with the tag "general".



Launch Instance Connect Actions

Filter by tags and attributes or search by keyword 1 to 2 of 2

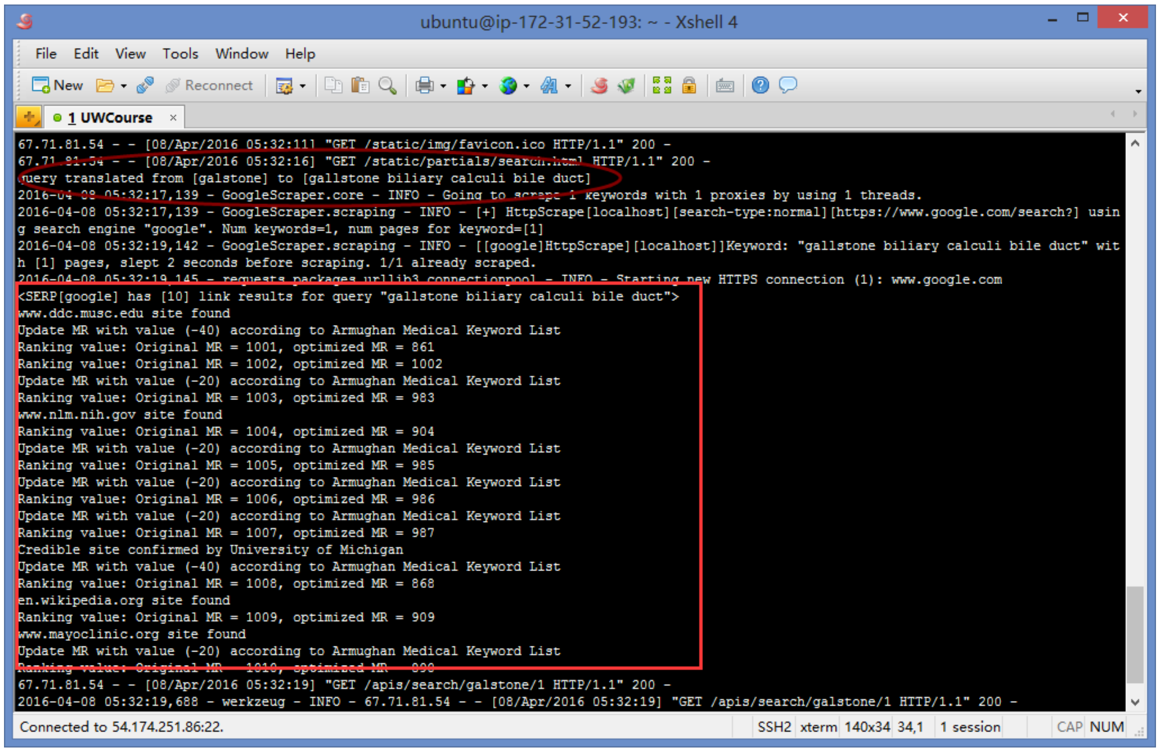
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
■	SmartMedical	i-3378c6b0	t2.medium	us-east-1d	● running	✓ 2/2 checks…	None	ec2-54-174-251-86.com…
□	datatellit	i-3f2289d0	t2.micro	us-east-1b	● running	✓ 2/2 checks…	None	ec2-52-3-103-94.compu…

Instance: i-3378c6b0 (SmartMedical) Public DNS: ec2-54-174-251-86.compute-1.amazonaws.com

Description Status Checks Monitoring Tags

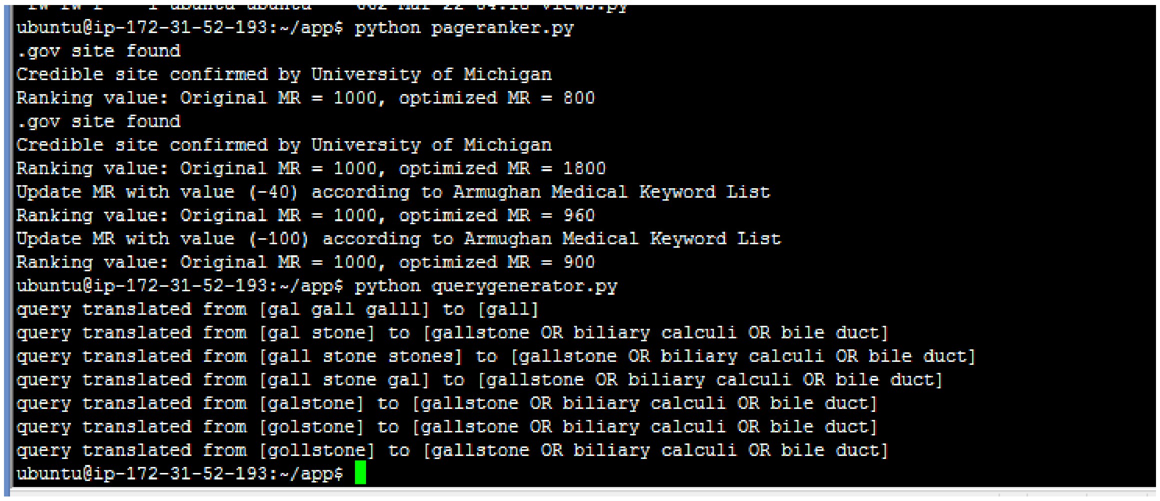
Instance ID	i-3378c6b0	Public DNS	ec2-54-174-251-86.compute-1.amazonaws.com
Instance state	running	Public IP	54.174.251.86
Instance type	t2.medium	Elastic IP	-
Private DNS	ip-172-31-52-193.ec2.internal	Availability zone	us-east-1d
Private IPs	172.31.52.193	Security groups	open . view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-e6354c83	AMI ID	ubuntu-trusty-14.04-amd64-server-20160114.5 (ami-fce3c696)

Figure 15 Backend - Amazon EC2 instance management.


ubuntu@ip-172-31-52-193: ~ - Xshell 4
File Edit View Tools Window Help
New Reconnect
1 UWCourse

```
67.71.81.54 - - [08/Apr/2016 05:32:11] "GET /static/img/favicon.ico HTTP/1.1" 200 -
67.71.81.54 - - [08/Apr/2016 05:32:16] "GET /static/partials/search.html HTTP/1.1" 200 -
query translated from [galstone] to [gallstone biliary calculi bile duct]
2016-04-08 05:32:17,139 - GoogleScraper.core - INFO - Going to scrape 1 keywords with 1 proxies by using 1 threads.
2016-04-08 05:32:17,139 - GoogleScraper.scraping - INFO - [+] HttpScrape[localhost][search-type:normal][https://www.google.com/search?] usin
g search engine "google". Num keywords=1, num pages for keyword=[1]
2016-04-08 05:32:19,142 - GoogleScraper.scraping - INFO - [[google]HttpScrape][localhost]]Keyword: "gallstone biliary calculi bile duct" wit
h [1] pages, slept 2 seconds before scraping. 1/1 already scraped.
2016-04-08 05:32:19,145 - requests.packages.urllib3.connectionpool - INFO - Starting new HTTPS connection (1): www.google.com
<SERP[google] has [10] link results for query "gallstone biliary calculi bile duct">
www.ddc.musc.edu site found
Update MR with value (-40) according to Armughan Medical Keyword List
Ranking value: Original MR = 1001, optimized MR = 861
Ranking value: Original MR = 1002, optimized MR = 1002
Update MR with value (-20) according to Armughan Medical Keyword List
Ranking value: Original MR = 1003, optimized MR = 983
www.nlm.nih.gov site found
Ranking value: Original MR = 1004, optimized MR = 904
Update MR with value (-20) according to Armughan Medical Keyword List
Ranking value: Original MR = 1005, optimized MR = 985
Update MR with value (-20) according to Armughan Medical Keyword List
Ranking value: Original MR = 1006, optimized MR = 986
Update MR with value (-20) according to Armughan Medical Keyword List
Ranking value: Original MR = 1007, optimized MR = 987
Credible site confirmed by University of Michigan
Update MR with value (-40) according to Armughan Medical Keyword List
Ranking value: Original MR = 1008, optimized MR = 868
en.wikipedia.org site found
Ranking value: Original MR = 1009, optimized MR = 909
www.mayoclinic.org site found
Update MR with value (-20) according to Armughan Medical Keyword List
Ranking value: Original MR = 1010, optimized MR = 990
67.71.81.54 - - [08/Apr/2016 05:32:19] "GET /apis/search/galstone/1 HTTP/1.1" 200 -
2016-04-08 05:32:19,688 - werkzeug - INFO - 67.71.81.54 - - [08/Apr/2016 05:32:19] "GET /apis/search/galstone/1 HTTP/1.1" 200 -
```

Connected to 54.174.251.86:22.
SSH2 xterm 140x34 34,1 1 session CAP NUM



```
-rw-rw-r--  1 ubuntu ubuntu   602 Mar 22 04:18 views.py
ubuntu@ip-172-31-52-193:~/app$ python pageranker.py
.gov site found
Credible site confirmed by University of Michigan
Ranking value: Original MR = 1000, optimized MR = 800
.gov site found
Credible site confirmed by University of Michigan
Ranking value: Original MR = 1000, optimized MR = 1800
Update MR with value (-40) according to Armughan Medical Keyword List
Ranking value: Original MR = 1000, optimized MR = 960
Update MR with value (-100) according to Armughan Medical Keyword List
Ranking value: Original MR = 1000, optimized MR = 900
ubuntu@ip-172-31-52-193:~/app$ python querygenerator.py
query translated from [gal gall galll] to [gall]
query translated from [gal stone] to [gallstone OR biliary calculi OR bile duct]
query translated from [gall stone stones] to [gallstone OR biliary calculi OR bile duct]
query translated from [gall stone gal] to [gallstone OR biliary calculi OR bile duct]
query translated from [galstone] to [gallstone OR biliary calculi OR bile duct]
query translated from [golstone] to [gallstone OR biliary calculi OR bile duct]
query translated from [gollstone] to [gallstone OR biliary calculi OR bile duct]
ubuntu@ip-172-31-52-193:~/app$
```

Figure 16 Backend - Query Generator and Page Ranker.