Group 33: CSC326 Final Report

Patricia Marukot 1002157499

Jing Yi Li 1002346730

Description of Design:

Table 1: Extra Features

Feature	Description
Improved ranking system	- Compute page-rank scores based on all search keywords - Higher ranking is given to urls that: a. Include keywords in the title b. Contain more occurrences of keywords
Non-Google Login user	- User can create an account with just a user - When signed in, user has access to more features (i.e. history, widgets)
History and most popular	- Feature only shown for signed in users - displays 20 most popular and last 20 most recent searches as implemented in Lab 2
Return results computed in backend	- When making query request to the server, the server returns the complete set of urls matching query - Previously, data structure manipulations (ie creating arrays of 5 for each page) were done in the frontend
Images Tab	- Image tab contains the images from the urls matching the search query - User can click to focus on an image and also to open the full sized image in new tab
Widgets Tab	-]Newsstand, Weather - Obtains top 3 news stories and displays it in news section of widgets tab - Obtains 10 day weather forecast using apis

<u>High Level Documentation of Code:</u>

Table 2: Features and File Locations

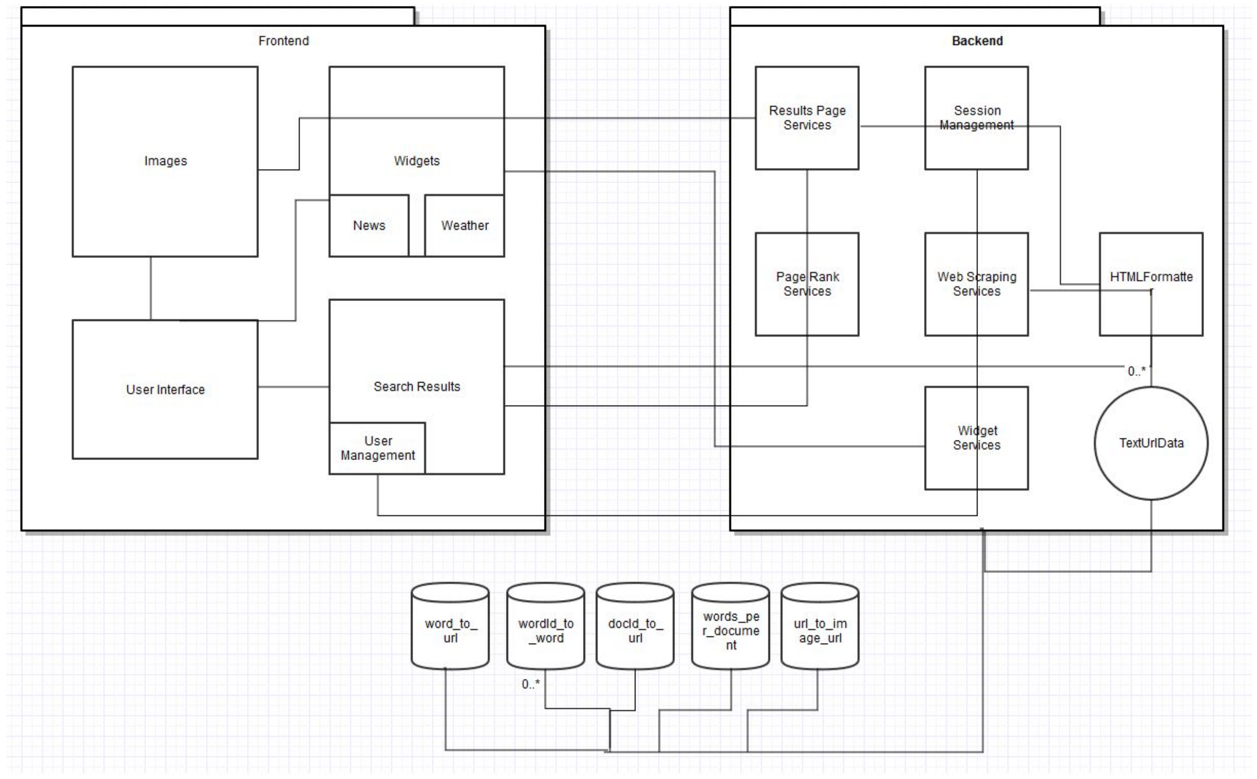
Improved Ranking	/ResultsPageServices - SearchResultsService/Helper.py
Non-Google Login	/SessionManagement
History and Most Popular	/ResultsPageServices and /HTMLFormatter folders
Return results in backend	/ ResultsPageServices/SearchResultsServices.py
Images Tab	Image scraping: /WebScrapingServices/WebScapingHelper.py
Widgets Tab	/WidgetServices

*Note: Project frontend uses AngularJS framework to render the views and retrieve data from backend - all frontend code (i.e. HTML, Angular controllers/directives etc.) can be found in /static/js/GoogaoAngularApp

External Dependencies:

BeautifulSoup4	Bottle	Beaker	json
Boto	Weather API	NYTimes APIs	

UML Diagram:



<u>Proposed Design vs. Completed Design:</u>

<u>Proposed Design Features:</u>

- These features were required by us for Labs 1-3

#	Feature	Description
1	Words and word count	Display each word in search query in a table with the number of their occurrences in query
2	History and most popular	Display the number of times each word was searched overall in 2 tables (20 most popular and 10 most recent)
3	Display links using pagerank algorithm	Display the url results matching the search query in a readable format for the user The urls chosen were based on the pagerank algorithm on the urls that matched the first word in the query

<u>Completed Design Features:</u>

- Complete list of features found in Table 1: Extra features

Changes from Proposed: Omitted feature 1 and made alterations to algorithm in Feature 3

Altered Pagerank algorithm:

- Made alterations because we wanted to customize the pagerank score of the urls for each query instead of having the same set of pagerank scores for every query
- We kept the original scores compute from Lab 3 using the algorithms but increased a url's score based on 2 factors:
    a) If a keyword appeared in the url's title
    I.e. wikipedia.com has original score 0.158 and title: "hello i'm wikipedia" ➔ enter query: hello world ➔ update scare is 0.158 + 1 = 1.158
    b) Increment the url's score for each keyword that appears in the url's webpage
- We further improved the results displayed to the user by matching the urls to each of the keywords entered by the user instead of just the first word in the query as implemented in the proposed design
- These alterations provided the user with more accurate and relevant results for the search queries

<u>Testing:</u>

- Used the PyUnittest Framework to perform unit testing on each of the individual features (i.e. search results, most popular/history)
- Features are grouped together by functionality and each functionality as a unit testing file in /UnitTests folder
- We generate our own files for testing the corner cases so that we can easily obtain the correct results and compare it with our original functions

Lessons Learned:

- For large projects such as this one, it is important to modularize the code instead of having a single file with all the functionalities because it makes it easier to find bugs and problems and it makes it more organized for other teammates
- Unit testing is an excellent way to test individual functionalities of a project
- It is important to test your code as you go instead of writing all of the functionality at once and testing it in the end. This will allow for fewer mistakes and is easier to debug in the long run

Describe what you would do differently if you had to do it again.

- We would not save all of the lab work for the weekend before the lab is due.

What would you do if you had more time.

- If we had more time, we would improve existing functionalities (i.e. improve performance and query time for a smoother user experience)
- Include other functionalities to the search engine

Did any parts take longer than you thought, and Why?

- Setting up the AWS Instance and the Google Login took longer than expected
- For example, the majority of time spent on Lab 2 was spent attempting to set up the Instance and the Google Apis rather than the actual functionalities of the search engine due to unclear instructions
- More tutorial and lecture time spent explaining how to set these up would be very helpful

How the material from the course helped you with the project.

- It taught us python syntax and basic libraries and functions.

How much time it takes for you to complete each lab outside the lab sections?

- Approximately two (2) days

Which part of the project you think is useful and you believe the labs should spend more time on it

- Launching our AWS instances
- Exploring technologies that were not taught to us in lecture such as Bottle, Beaker, Unittest, Boto, Google Apis, BeautifulSoup etc.
    - Should spend more time learning these technologies not only in tutorial but also in lecture
    - Maybe include more modern technologies (i.e. Flask, Django)

Which part of the project you think is useless and you think it should be removed from the labs when this course is being offered in the future.

- We don’t think anything should be removed as the project was a great learning experience for full stack web development in the real world.