Interim Planning and Investigation Report

“A Definition First Search Engine”

Julian Smith | 14803252

Contents

Project Scope

Aims & Objectives

The aim and objective of the project is to produce a web based search engine geared towards programmers, that will display search results generated previously from a web indexer. Put simply, a search engine for programmers. What makes this search engine unique, however, is that it will display a definition, if appropriate, of the searched item as well as an example piece of code, if appropriate, before the search results.

Project Stakeholders

- Project Team: Julian Smith
- Project Supervisor: Jane Challenger Gillitt
- Project second reader: Gulden Uchyigit
- Users: Available to everyone but geared towards programmers, regardless of their programming ability
- Development Tools: JetBrains S.R.O's "PyCharm" Python IDE (used to program the back-end of project) and their "WebStorm" Javascript, HTML and CSS IDE (used to program the front-end of the project)

Methods of Communication

The project requires communication with the project supervisor Jane Challenger Gillitt. The progress of the project, and any issues that the project might be facing, will be communicated to ensure that the aims and objectives of the project will be completed. These consultations will occur once a month to start with, with the frequency increasing in the new year and as the project nears the deadline.

Installation Standards and Procedures

The back-end web crawler / indexer will be available as a Python application that will be able to run on any Windows, Linux / UNIX, or Mac OS machine, provided they have Python 3.0+ installed. If this is not the case, Python 3.0+ is available for free to download for a variety of machines and requires little technical ability to install. The back-end database is likely to be written in mySQL and PHP- see Risk Analysis for more details. The front-end website will be available as a local HTML file that can be opened by any web browser. While steps will be taken when coding the website to ensure that as many different web browsers and their relevant version will be able to run the website, an up-to-date modern browser such as Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari is highly recommended.

Project Approach

Due to time constraints, the project is to run as an agile development. This would thus ensure that as many of the prioritised requirements will be completed before the project deadline. Furthermore, the agile approach allows the project to deal with possible changing requirements. This is significant, as the requirements of the project are liable to change not only due to time constrains but also due to feedback the project will receive in the form of a study (beta testing).

Project Requirements List

- The project must have:
  - Back-end: A web crawler / indexer that can visit a website the user inputs and visits every page within that website. It will then save (index) all relevant data from each page into a database readable format.
  - Data gathering: The web crawler / indexer autogenerates “tag” words and a description for each page which will aid in searching if they aren’t provided.
  - Front-end: A website which users can search queries, with the search results (from the database containing the data from the web crawler / indexer) of these search queries displayed to the user. A definition of the searched query as well as an example piece of code before the search results must also be displayed if relevant to the search query.
  - Usability: The front-end website should be easy to use, visually pleasing, and sufficiently intuitive so that the user will require no instructions. The success of this requirement will be measured by the outcomes of the beta testing.
- The project should have:
  - Page Rank: Search query results should be displayed / ranked in an order that prioritises the most relevant search results.
  - Multiple search query result pages.
  - The ability to check for any changes to the indexed webpages to keep content up to date
- The project could have:
  - Website Animations = JavaScript or jQuery animations to improve the usability and overall look and feel of the website.
  - Query recommendations e.g “Did you mean…?”
- The project would like:
  - Ability for third parties to submit their website for indexing.
  - Advertising opportunities (e.g. sponsored first result) for third parties.
  - The ability to search for and index images.
  - Multiple languages.

Project User Testing

As the project nears completion, two project beta tests will take place. This testing will gather information on the usability and design aspects of the website. The participants in the beta testing will be provided with a checklist of tasks to complete, along with a survey which will provide both quantitive and qualitative data. The second project beta testing will require participants to try out the website again, with changes- based on the feedback from the first beta testing- implemented into the system. For more information, please refer to the University of Brighton’s School Ethics Form attached at the end of this paper.

Deliverables

Project Deliverables

- A front-end website (Code Languages: HTML, CSS, JavaScript, PHP)
- A back-end web crawler / indexer (Code Language: Python)
- A back-end database containing indexed results

Schedule of Activities

- 14th December 2016 = Simple web crawler created that can index pages into a database suitable format. Relevant information is indexed.
- 7th January 2017 = Back-end database fully functional.
- 31st January 2017 = Front-end website created and is functional.
- 1st February 2017 = First day of beta-testing (study) for participants to try out the website and give qualitative and quantitive feedback.
- 2nd February 2017 = Bug fixing and any improvements, new features, design changes, in response to the beta-testing, will be implemented into the system during this time.
- 1st March 2017 = Second beta-testing begins
- 2nd March 2017 = More bug fixing and any more improvements needed which were highlighted from the second beta-testing.

Risk Analysis

The nature of the project itself brings about a number of of risks. For example, heavy time constraints, areas in which I have little practical experience, and other commitments outside of the project, create a large volume of risks. As such, this risk analysis will focus on the three most concerning risks that must be tackled in this project.

The possibility of running out of time and failing to complete stages of the project is likely due to heavy time constraints. This is exacerbated by the fact that I am the only person working on this project, in conjunction with other responsibilities to my studies, and so the probability of this risk is high. To mitigate the occurrence this risk, I will focus solely on completing the *must* have requirements of the project as a priority, with any 'secondary' requirements being completed afterwards. Furthermore, the use of agile development in the running of my project will help to quickly and efficiently change the more challenging, time-consuming requirements of this project.

Another related risk is the possibility of certain areas of the project taking far longer to complete than previously estimated. There is a high probability of this risk materialising, as, quite simply, I've never created a search engine before. Thus, there are a number of areas within the project in which I have little experience in successfully exercising, and concepts that are not totally familiar to me. As such, the learning curve may take longer in certain areas than expected, and could have the extraneous effect of slowing down the production speed of the project. To reduce the possibility of this risk materialising, I have done, and will continue to do, extensive research into unfamiliar areas of the project before taking them on, to avoid smaller problems compounding and creating much larger problems in the long run.

The third significant risk associated with this project concerns the database of indexed websites, and how I would go about practically doing this. Resources available in this matter are limited, and thus brings about some uncertainty concerning the success of this project as a whole. At the time of writing this risk analysis, there appears to be only one truly feasible method of creating the database- using mySQL and PHP, with PHP forming the connection between the mySQL server and the front-end website. This long, indirect method of getting the database to function is likely to take much time to get fully functional, and thus proposes a major obstacle in light of intense time constraints. It is certainly challenging to successfully mitigate the chance of this risk from materialising, as it is a central aspect of the project, though the sourcing of as many resources as possible on the topic would ease this somewhat. Though this, as mentioned before, remains a tall order. Perhaps more realistically, the best precaution in light of this risk is to designate more time to the creation of the database than any other aspect of the project, to ensure its completion as a fundamental requirement of the project.

Background Research

Research Summary

Due to the nature of my project, I've conducted extensive research into a number of areas; for example, I've undertaken the study of large volumes of information concerning web crawling and the process of web indexing, looking also to the documentation of programming languages. Without these documentation resources, my project would be significantly more difficult to complete. In addition, I have successfully sourced research papers from the individuals who created Google, which provide excellent resources for developing a search engine.

Annotated Bibliography

Brin, S. and Page, L. (2012) Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, *56*(18), pp.3825-3833.
This paper was written by the two co-founders of Google, the worlds largest search engine, and describes in great detail how they created the Google search engine. For example it explains how they crawl and index webpages and how their PageRank and keywords (Hit List) algorithms work. The information within this paper is extremely useful as it provides valuable information as to how to do elements of my project I need to include, more so that this information comes from the biggest search engine in the world.

Page, L., Brin, S., Motwani, R. and Winograd, T. (1999) The PageRank citation ranking: bringing order to the web.
This paper, written in-part by one of the co-founders of Google, describes how to efficiently rank web page results. This, coupled with the previous cited resource, provides a greater incite to how Google was created, but more significantly, how I myself can create a search engine ranking algorithm. Thus, this resource is very useful in the creation of my search engine.

Oudinet, J. (2006) Search Engine Ranking.
This paper outlines the basic structure of how a web crawler works. Furthermore, it proposes a new technique for the PageRank algorithm as proposed in the study beforehand. This paper is useful for my project as it gives a framework on how to produce an efficient web crawler as well as an improved version of the PageRank algorithm.

Mitchell, R. (2015) Web scraping with python: A comprehensive guide to data collection solutions. United States: O'Reilly Media, Inc, USA.
This resource provides step-by-step instructions on how to web scrape (crawl) the internet using Python- the programming language my project will also be built on. As such, this resource is very useful as it covers a large portion of my project, along with detailed sample code to guide me.

2016 (2001) Overview — python 3.5.2 documentation. Available at: https://docs.python.org/3/ (Accessed: 20 November 2016).
This website is by Python, and provides documentation on all aspects of the python 3.5+ programming language. This is a valuable resource as not only does it provide tutorials on the python programming language itself, but also references to python libraries and how they function.

Richardson, L. (2015) Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation. Available at: https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (Accessed: 20 November 2016).
This website contains documentation for Beautiful Soup 4, a python library which I'll be using heavily in the back-end web crawler to gather all the information I need from a webpage. This means that this website will be a become valuable resource, as it contains instructions and describes in detail functions that come with the Beautiful Soup library.

W3Schools online web tutorials (no date) Available at: http://www.w3schools.com (Accessed: 21 November 2016).
This online resource provides reference information for HTML, CSS and JavaScript languages; the programming languages I'll be using to create the front-end website. More specifically, this resource has a section dedicated to displaying every single type of "Tag" element in HTML and what each tag does. This is incredibly useful for the web crawler / indexer as I can see exactly what "Tags" I need in order to source all the appropriate data.

van Rossum, G., Warsaw, B. and Coghlan, N. (2013) PEP 8 -- style guide for python code. Available at: https://www.python.org/dev/peps/pep-0008/ (Accessed: 21 November 2016).
On this webpage, information on how to properly lint (layout) code for the Python programming language using PEP 8 styling. This resource will allow me to write the back-end web crawler / indexer in not only a clear and concise way, but also following guidelines encouraged by the Python programming language itself and thus other Python programmers.