[Date]

Maksim Livingstone

SQA AH COURSEWORK

URL path finder

Contents

[Description of the Problem 2](#_Toc21428588)

[Outline of problem. 2](#_Toc21428589)

[Scope 2](#_Toc21428590)

[Boundaries 3](#_Toc21428591)

[Constraints 3](#_Toc21428592)

[UML 4](#_Toc21428593)

[Requirements 5](#_Toc21428594)

[End user requirements: 5](#_Toc21428595)

[Survey 5](#_Toc21428596)

[Functional requirements: 5](#_Toc21428597)

[Project plan 6](#_Toc21428598)

[Identified tasks 6](#_Toc21428599)

[Resources required 6](#_Toc21428600)

[Estimate of timings 6](#_Toc21428601)

# Analysis

## Description of the Problem

### Outline of problem.

I intend to design a program to find paths between webpages. The program will include the following features: a help screen, text-based UI to help user find their way, a database of webpages crawled that will be created using SQL, with a linked database of pages they have linked to and a pathfinding system.

The end users of my program will be people willing to find orphan links in webpages if a search system is not used, therefore will probably be tech literate.

My project meets the advanced higher computing requirements as it will have a UI suitable for tech literate users with validation for if the pages have valid URLs by using a try catch with a get() procedure and checking the code sent from the sever is not 404 and that the domain exists. My project will interface with an SQL database, creating a database and writing and reading URL’s from it. My project will also have a sorting algorithm for a verbose mode to print the node maps.

### Scope

* A full design including pseudocode and data dictionary and query design for creating databases.
* A working data scraper and path finder, both being called from one system
* A help file that can be displayed in command prompt
* A test plan including test persona, test cases and outputs when different data is inputted and the results of the testing
* End user survey
* An evaluation of the solution.
* Receive start page, end page and jump limit from user and mode.
* Scrape start page for links and then follow the links and repeat for number of jumps specified by user.
* Save URL and URL’s that the pages lead to by using SQL into a database.
* Load data from database into a class.

~~Scope the clearly defined outline of what the solution will deliver in terms of functionality~~

### Boundaries

* Will not follow links onto a different domain.
* Max number of jumps will be 300 as too not slow down the computer.

UI will be text based only and not a graphical interface.

* There will be no graphical representation of the possible paths.

### Constraints

Some technical, legal and time constraints apply to this project.

* This project will need to be completed between X and Y due to SQA requirements.
* This project can only use open source modules for data scraping.
* I will be using primarily windows 10 and Linux Fedora for development.
* The project will be primarily written in python 3.7 as it is the language I am most proficient in, with the exception of the SQL that is being used to store and fetch the URL’s from the database and any external modules that are programmed in another language.

~~Constraints the restrictions that apply to the development.~~

## UML

<https://www.lucidchart.com/invitations/accept/fcc0b091-b583-4fe0-82f1-93118e9e6ccf>

### Requirements

#### End user requirements:

* User must be able to use text-based UI to input a starting website, and an end website to find a path to.
* User must be able to input the number of moves they wish it to be done in.
* The user must be able to view the pages that the spider finds without the pathfinder if they wish.
* The user must be able to view the requested path, or receive an error message that there is no path

#### Functional requirements:

* The program must display a text-based UI that can take in a starting page, end page(optional), number of moves and mode.
* The program will be able to crawl a URL and find all links on the URL, follow them, and repeat the process until the maximum jumps is achieved.
* The webpage’s URL is to be stored then all the links leading off also need to be stored in a database using SQL.
* These are to be stored in a database using SQL queries.
* Depending on the mode that the user requested, the program will either display the contents of the database (aka the results of the spider) or it will:
  + The program will then call the separate path finding algorithm.
  + SQL queries then need to be written into a 2d array.
  + (possible: sort 2d array so that easier to read.)
  + Then a node map object is created using the data in the 2d array.
  + Then using a pathfinding algorithm, find A path (not always shortest) from first link to second link in the maximum number of jumps give

##### Survey



## Project plan

### Identified tasks

### Resources required

The resources that will be required at each stage of development are listed here:

|  |  |
| --- | --- |
| Analysis | * Gantt project 2.8.10 (windows) * Google chrome (windows) * Firefox (Debian) * Microsoft word (windows) * Survey monkey. |
| Design | * Microsoft word (windows) |
| Implementation | * Visual studio code 1.38.1 * Python 3.7 with scrapy and MySQL installed * Anaconda environment * Google chrome * Git 2.20.1.windows.1 for backup and version control * GitHub account * Access to official scrapy documentation and tutorial * Access to MySQL documentation. |
| Testing |  |
| Evaluation |  |

### Estimate of timings

