pylinkvalidator
Problem Statement : newAGEtech, Group H

Genevieve Okon (Okong), Abraham Omorogbe(Omorogoa),
Eric Le Forte(Leforte)

October 9, 2015

Contents

1 Project Drivers

1.1 1. THE PURPOSE OF THE PROJECT

1.1.1 1.A) The User Business or Background of the Project

This project will be an attempt to ease the difficult and tedious process of verifying a websites current status and state of availability as well as also being able to assist researchers in scouring the web efficiently for potentially relevant sources of data. Some specific tasks expected to be completed using this software include ensuring all of a websites pages are functioning correctly and resources that should be available are, data mining for certain research topics and for security professionals ensuring that only the webpages that should be visible are.

1.1.2 1.B) Goals of the Project

This product aims to be used as an effective tool dealing with various forms of website analysis. It will be an efficient and customizable resource that allows you to perform various web crawler based activities on websites in order to increase productivity and for businesses to save money.

1.2 2. CLIENTS, CUSTOMERS, STAKEHOLDERS

1.2.1 2.A) The Client

The clients of this product will primarily be analysts and website developers.

1.2.2 2.B) The Customer

The customers of this product will be the companies who employ the aforementioned clients as well as the clients themselves if it is purchased for personal use.

1.2.3 2.C) Other Stakeholders

The other stakeholders include the developers of the software, investors as well as a previously established application store (i.e. Google Play or Apples App Store) for distribution.

1.3 3. USERS OF THE PRODUCT

1.3.1 3.A) The Hands-On Users of the Product

- Software Developers/Testers - Develops, tests and maintains websites. - Strong knowledge of technology, especially technology involved in this products field. - Reasonably educated (most likely a college degree or better). - Anticipated age group: 20+

Researchers - Researches various topics relevant to their business, especially matters pertaining to the stock trade. - No required technological knowledge other than the basics of how to use a computer. - Expected education is a high school diploma or better - Anticipated age group: 20+

Other Members of the General Population - May have any sort of experience, education and skill sets - Anticipated age group: 10+

1.3.2 3.B) Priorities Assigned to Users

Key Users: - Software Developers/Testers - Researchers

Secondary Users: N/A

Unimportant Users: - Other Members of the General Population

1.3.3 3.C) User Participation

Software Developers: N/A

Researchers: - It would be mutually beneficial to obtain a set of desired functionalities that would assist this user in utilizing this product.

Other Members of the General Population: N/A

1.3.4 3.D) Maintenance Users and Service Technicians

N/A

2 Project Constraints

2.1 4. MANDATED CONSTRAINTS

2.1.1 4.A) Solution Constraints

This project will be implemented using the Python programming language. This is due to the languages simplicity and available libraries related to the project.

2.1.2 4.B) Implementation Environment of the Current System

This product will be installed onto a computer system using any platform that accepts the use of Python and will utilize the architecture of the world wide web.

2.1.3 4.C) Partner or Collabrative Applications

This product will be capable of operating on websites that are locally stored.

2.1.4 4.D) Off-The-Shelf Software

Some example of similar existing solutions include search engines (i.e. Google or Bing), and various other generic web crawlers that have already been implemented.

2.1.5 4.E) Anticipated Workplace Environment

This project is expected to be utilized primarily in office environments. These will include a computer that is likely hooked up to an active internet connection.

2.1.6 4.F) Schedule Constraints

This project must be completed by the eleventh of November.

2.1.7 4.G) Budget Constraints

The budget for this project is exactly $0.

2.2 5. NAMING CONVENTIONS & DEFINITIONS

2.2.1 5.A) Definition of All Terms, Including Acronyms, Used in the Project

Web Crawler/Crawler - An internet bot which systematically browses the World Wide Web, typically for the purpose of Web indexing.

Uniform Resource Locator (URL) - A reference (address) to a resource on the internet. Has two main components: a Protocol Identifier (e.g. http) and a Resource Name (e.g. Google.ca)

HyperText Markup Language (HTML) - The standard markup language used to create webpages.

List of standard HTTP status codes and their definitions:

1XX are informational status codes.

Continue (100) - The client should continue with its request.

Switching Protocols (101) - The server has acknowledged and approved a request to switch protocols.

2xx refers to successful requests.

OK (200) - The request has succeeded.

Created (201) - The request succeeded and resulted in a new resource being created.

Accepted (202) - The request has been approved but is still processing.

Non-Authoritative Information (203) - The returned metadata in the entity-header is not the definitive set defined in the origin server, instead it is gathered from a local or third-party copy.

No Content (204) - Request succeeded but no entity-body was required from the server.

Reset Content (205) - Request succeeded, the document view which caused the request to be sent should be updated.

Partial Content (206) - Partial GET request succeeded.

3xx refers to redirection status codes.

Multiple Choices (300) - The requested resource corresponds to any one of a set of representations.

Moved Permanently (301) - The requested resource has been assigned a new permanent URL.

Found (302) - The requested resource currently resides in a temporary location.

See Other (303) - The response to the request can be found under a different URL and should be retrieved using a GET method on that resource.

Not Modified (304) - The client performed a conditional GET request but the document has not been modified.

Use Proxy (305) - The requested resource must be accessed through the proxy given by the Location field.

(306) - No longer in use, the code is now reserved.

Temporary Redirect (307) - The requested resource temporarily resides under a different URL.

4xx refers to Client Error status codes.

Bad Request (400) - The request was not understood due to malformed syntax.

Unauthorized (401) - The request requires user authorization.

Payment Required (402) - Code reserved for future use.

Forbidden (403) - The request was understood however, the server is refusing to fulfill it.

Not Found (404) - The server could not find anything matching the provided URL.

Method Not Allowed (405) - The method specified in the Request-Line is not allowed for the resource identified by the Request-URI.

Not Acceptable (406) - The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.

Proxy Authentication Required (407) - Similar to 401 except the client must first authenticate itself with the proxy.

Request Timeout (408) - The client did not produce a request within the time that the server was prepared to wait.

Conflict (409) - The request could not be completed due to a conflict with the current state of the resource.

Gone (410) - The requested resource is no longer available at the server and no forwarding address is known.

Length Required (411) - The server refuses to accept the request without a defined content length.

Precondition Failed (412) - The precondition given in one or more of the request header fields evaluated to false when it was tested on the server.

Request Entity Too Large (413) - The server is refusing to process a request because the request entity is larger than the server is willing or able to process.

Request URL Too Long (414) - The server is refusing to service the request because the request URI is longer than the server is willing to interpret.

Unsupported Media Type (415) - The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.

Requested Range Not Satisfiable (416) - A request included a range request header field and none of the range-specifier values in this field overlap the current extent of the selected resource and the request did not include and if-range request header field.

Expectation Failed (417) - The expectation given in an expect request header field could not be met by this server

5xx refers to Server Error status codes.

Internal Server Error (500) - The server encountered an unexpected condition which prevented it from fulfilling the request.

Not Implemented (501) - The server does not support the functionality required to fulfill the request.

Bad Gateway (502) - The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.

Service Unavailable (503) - The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.

Gateway Timeout (504) - The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URL it needed to access in attempting to complete the request.

HTTP Version Not Supported (505) - The server does not support, or refuses to support, the HTTP protocol version that was used in the request message.

2.2.2 5.B) Data Dictionary for Any Included Models

This section to be updated as relevant models arise.

2.3 6. RELEVANT FACTS & ASSUMPTIONS

2.3.1 6.A) Facts

The existing solution is 7,000 lines of Python code.

2.3.2 6.B) Assumptions

This product assumes that the operational environment supports the execution of Python code. This product will assume that the HTML code being parsed for links uses correct syntax (otherwise the error messages may be nonsensical).

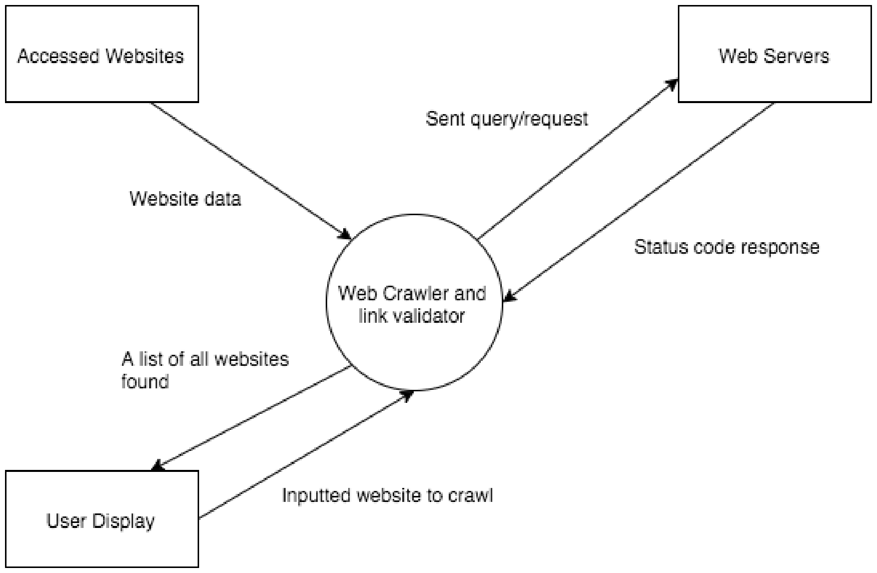
3 Functional Requirements

3.1 7 The Scope of the Work

3.1.1 7.A) The Current Situation

It is difficult often difficult for users of the internet to navigates through the web, it can be difficult to locate pertinent information using standard methods of search such as search engines. A web crawler and link validator will allow users quickly traverse the web and quickly and effectively locate information from different web pages and information about these web pages. A web crawler and link validator will increase users efficiency, it will be able to save money and resources due to users spending less time and effort surfing the web for information.

3.1.2 7.B) The Context of the Work



3.1.3 7.C) Work Partitioning

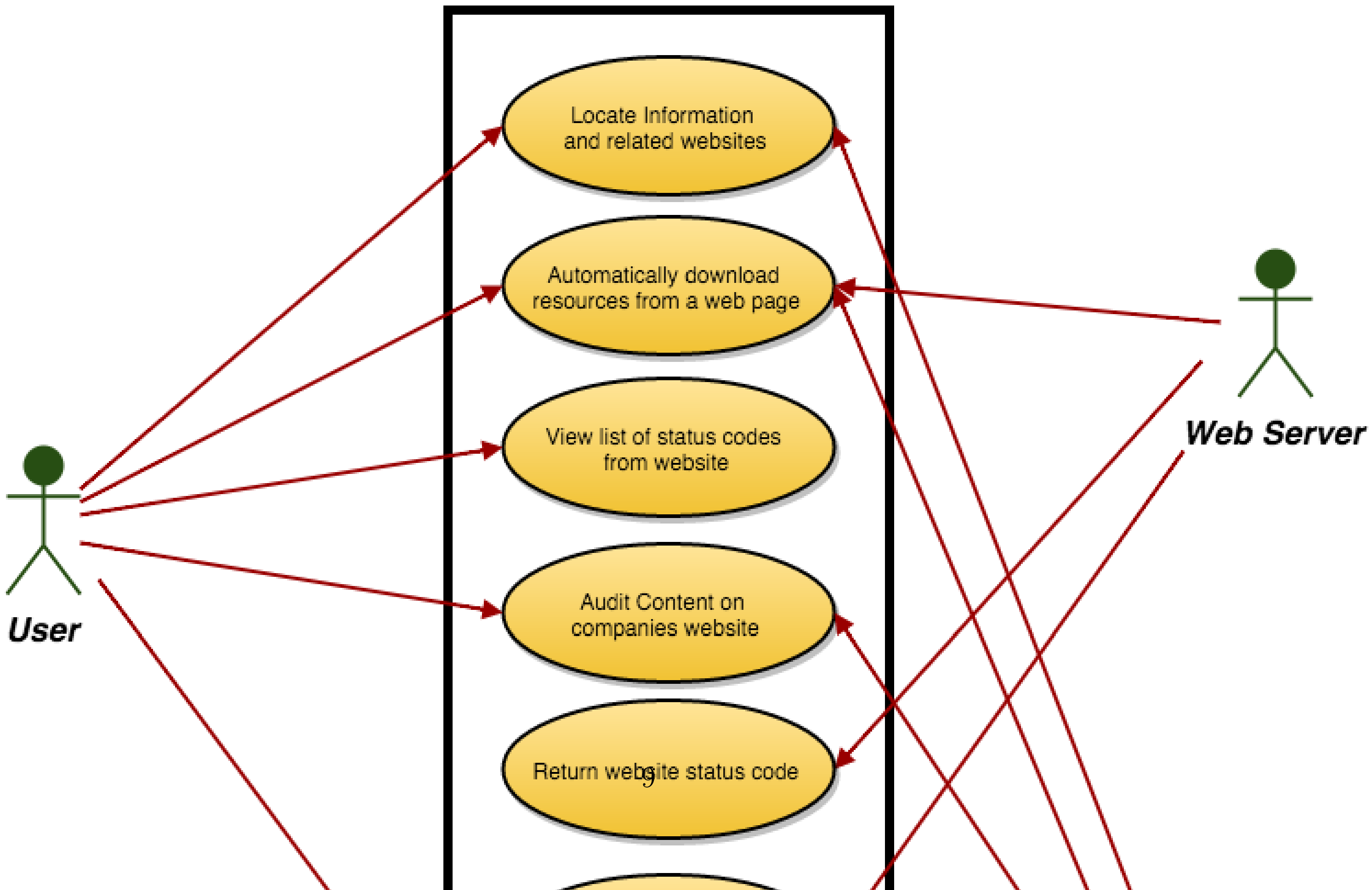
The table 1 is a Business Event List.

Event Name	Input and Output	Summary
User requests information about a website	Website start link to crawl and specific info gathering setting (In)	Have a starting point for crawler can gather resources from.
User requests information from a website	Website start link to crawl and specific status checking setting (In)	Have a starting point for crawler, it can continue to verify any links associate with initial website.
Traverse website	Links from initial retrieved website (In)	Has the abilities to reach website that are associated with starting link
List of crawled websites status code	A list of all the site that were visited (Out)	Has the ability to show the user the status codes from crawled websites, can verify if links are down.
List of information from website	All resources from a website (Out)	Has the ability to show the user all the information from website they have elected.

Table 1: Business Event List

3.2 8. The Scope of the Product

3.2.1 8.A) Product Boundary


Locate Information and related websites
Automatically download resources from a web page
View list of status codes from website
Audit Content on companies website
Return website status code
User
Web Server

3.2.2 8.B) Product Use Case List

- Locate Information and related websites
- Automatically download resources from a web page
- View list of status codes from website
- Specify website to crawl
- Audit Content on companies website
- Map website structure
- Return website status code

3.2.3 8.C) Individual Product Use Cases

1. Product Use Case Name: Locate Information and related websites
   Trigger: User requests a website to be processed
   Preconditions: User has to specify website that exists
   Interested Stakeholders:
   Actors: User, Website
   Outcome: If the website is a valid HTML page, the link on the page a verified and information is gathered (text-based), if the website is an invalid, an error is displayed.

2. Product Use Case Name: Automatically download resources from a web page.
   Trigger: User types in command to download data Preconditions: Web server has to have existing resources available, and user needs permission to the website.
   Interested Stakeholders:
   Actors: User, Web Server, Website
   Outcome: User has all the resources from the website (images, attached files etc.).

3. Product Use Case Name: View list of status codes from website Trigger: User types in command to show status code
   Preconditions: User must have Internet connection and must have entered a valid URL.
   Interested Stakeholders:
   Actors: User
   Outcome: User can see a list of all the status codes on websites associated with the initial site.

4. Product Use Case Name: Specify website to crawl
   Trigger: User types in a website to crawl
   Preconditions: User must have Internet connection and a valid website to crawl.
   Interested Stakeholders:
   Actors User, Website

Outcome: The web crawler begins to traverse through website.

5. Product Use Case Name: Audit Content on companies website
   Trigger: User requests a list of all websites associated with the starting point.
   Preconditions: User must enter a website with valid ¡a href=¿ ¡/a¿ tags
   Interested Stakeholders:
   Actors: User, Website
   Outcome: The users can see a list of on the links and resources attached to every webpage associated with the specified domain.

6. Product Use Case Name: Map website structure
   Trigger: User requests a structure of a website
   Preconditions: Must have entered a valid URL/HTML.
   Interested Stakeholders:
   Actors: Web Server, Website
   Outcome: User is shown the website file structure.

7. Product Use Case Name: Return website status code
   Trigger: Web crawler sends request query to web server
   Preconditions: Web server must respond to queries with status codes.
   Interested Stakeholders:
   Actors: Web Server
   Outcome: The web server responds with correct status code.

3.3 9 Functional and Data Requirements

3.3.1 9.A) Functional Requirements

Requirement #: 1 **Requirement Type**: 9 **Event/Use case #**: 7
Description: The product shall return the status code of websites
Rationale: To allow the users verify the status of all the pages on a website. Checks if it is offline, Unauthorized etc.
Originator: Abraham Omorogbe
Fit Criterion: The web crawler displays an error with it reaches an website that is not online (Status Code: 200) and the status code related the error (Status Code: 400,500 etc.).
Customer Satisfaction: 5 **Customer Dissatisfaction**: 5
Priority: High **Conflicts**: None
Supporting Material: None
History: Created October 9, 2015 **Volere**

Requirement #: 2 **Requirement Type**: 9 **Event/Use case #**: 2

Description: The product shall download resources from a website

Rationale: To allow users quickly download all the HTML, images, CSS and other data related to a website

Originator: Abraham Omorogbe

Fit Criterion: The download resources can be retrieved and used.

Customer Satisfaction: 5 **Customer Dissatisfaction**: 5

Priority: High **Conflicts**: None

Supporting Material: None

History: Created October 9, 2015 **Volere**

Requirement #: 3 **Requirement Type**: 9 **Event/Use case #**: 1,5

Description: The product shall have adjustable search max-depth

Rationale: To allow the user specific how many layers of the website, the user want to analyse. Helps with auditing and finding related sites.

Originator: Abraham Omorogbe

Fit Criterion: The web crawler only show top-level links when max-depth is at 0, and shows more levels of the website when depth ¿ 0

Customer Satisfaction: 5 **Customer Dissatisfaction**: 5

Priority: High **Conflicts**: None

Supporting Material: None

History: Created October 9, 2015 **Volere**

Requirement #: 4 **Requirement Type**: 9 **Event/Use case #**: 4

Description: The product shall always users enter a starting URL or local HTML page

Rationale: To allow the user crawl any website that is on or offline

Originator: Abraham Omorogbe

Fit Criterion: The crawler starts from the inputted website.

Customer Satisfaction: 5 **Customer Dissatisfaction**: 5

Priority: High **Conflicts**: None

Supporting Material: None

History: Created October 9, 2015 **Volere**

Requirement #: 5 **Requirement Type**: 9 **Event/Use case #**: 3,5

Description: The product shall display all a report of all found results and corresponding websites

Rationale: This is one of the main functionality of a web crawl, the application should be able to start at one sites and find related links. List will be used to gather info and check status codes

Originator: Abraham Omorogbe

Fit Criterion: The reports are accurate. All link that are reported as status code 404 are offline and all details of resource match source website.

Customer Satisfaction: 5 **Customer Dissatisfaction**: 5
Priority: High **Conflicts**: None
Supporting Material: None
History: Created October 9, 2015 **Volere**

Requirement #: 6 **Requirement Type**: 9 **Event/Use case #**: 5,6
Description: The product shall able to display websites structure
Rationale: Give to users a visually representation of the structure for entered website.
Originator: Abraham Omorogbe
Fit Criterion: The web structure matches the structure of inputted website.
Customer Satisfaction: 5 **Customer Dissatisfaction**: 5
Priority: High **Conflicts**: None
Supporting Material: None
History: Created October 9, 2015 **Volere**

Requirement #: 7 **Requirement Type**: 9 **Event/Use case #**: 1
Description: The product shall find specific information from specified websites and related sites
Rationale: To allow user enter details they are looking for, and the application can return data based on starting URL.
Originator: Abraham Omorogbe
Fit Criterion: Returned websites most be related with inputted website and information return must match what the user specified.
Customer Satisfaction: 5 **Customer Dissatisfaction**: 5
Priority: High **Conflicts**: None
Supporting Material: None
History: Created October 9, 2015 **Volere**

3.3.2 9.B) Data Requirements

- Valid HTML pages to be parsed and crawl through.
- The websites resources that the crawler can download and check for.

4 Non-Functional Requirements

4.1 10. Look and Feel Requirements

4.1.1 Appearance requirements:

The command line interface has to be intuitive for all users of the web crawler.

4.1.2 Style requirements:

It would be recommended for users to use our program with a command line interface.

4.2 11. Usability and Humanity Requirements

4.2.1 Ease of Use Requirements:

Interface of the program should be intuitive and easy to understand. People with no training of the command line interface can use the product.

4.2.2 Personalization and international requirements:

This program can be used by anybody who has access to a computer and has adequate knowledge of a command line.

4.2.3 Learning requirements:

This program shall be easy to learn by an individual with adequate knowledge of the command line. If the user is using the webcrawler, he/she shall spend no more than a few seconds waiting for the results of the webcrawler depending on the number of links on a website.

4.2.4 Understandability and Politeness Requirement:

The program shall hide the details of its construction from the user.

4.2.5 Accessibility requirements:

A person using the program needs all fingers needed for typing by the user and eyesight. The product shall conform to the Americans with Disabilities Act.

4.3 12. Performance Requirements

4.3.1 Speed and Latency Requirements:

Speed of the program depends on the speed of the operating system. The program shall respond to the user?s requests within seconds, dependent on the number of links on the webpage.

4.3.2 Safety-Critical Requirements:

The program should not infringe the user?s privacy. This will be verified prior to completion of the program.

4.3.3 Precision or Accuracy Requirements:

The program should specify the type of error accurately. If a link is not working the program should identify that it?s not working with the correct error code.

4.3.4 Reliability and Availability Requirements:

This program should work whenever the user wants to use it - 24 hours a day, 7 days a week.

4.3.5 Robustness or Fault-Tolerance Requirements:

The system should tolerate corrupted HTML data, unexpected server configurations and behaviors and any other strange issues. Our system detects these situations and makes decisions or provides error messages for the system administration function. If there is an error found while a webpage is being crawled it should prompt a message stating the error next to the link and continue checking for other links.

4.4 Capacity Requirements:

Each user who?s using the webcrawler is independent of another. Therefore there?s no limit as to the number of people who can use this program at once.

4.5 Scalability or Extensibility Requirements:

Scalability, in terms of the increasing number of users over time, is not a concern.

4.6 Longevity Requirements:

This program performs a webcrawl on links. So it is reliable until there?s a major change in the website being crawled.

4.7 13. Operational and Environmental Requirements

4.8 Expected Physical Environment:

This program can be used anywhere using a device with suitable software requirements.

4.9 Requirements for Interfacing with Adjacent Systems:

pylinkvalidator(the webcrawler) does not require external libraries if executed with python 2.x. It requires beautifulsoup4 if executed with python 3.x. It has been tested on python 2.6, python 2.7, and python 3.3.

4.10 Production Requirements:

The product shall be distributed as pip. pylinkvalidator(the webcrawler) file can be downloaded from Github. It can be used on any device, which has an operating system that can run a python program. The device also needs to have a suitable interface for user interaction (e.g. command line interface).

4.11 Release Requirements:

A new release of the program shall not intervene with the data of the previous program. A webpage can still be crawled with the older version of the program, given the there exists a newer version.

4.12 14. Maintainability and Support Requirements

4.13 Maintenance Requirements:

The maintenance of the app has to be done by developers, and end user has no role in this regard.

4.14 Supportability Requirements:

This webcrawler program is self-supporting as its interface is very intuitive.

4.15 Adaptability Requirements:

The product is expected to run on python under Windows XP and Linux, mac os x.

4.16 15. Security Requirements

4.17 Access Requirements:

Only the user has access to their own personal information, like the last webpage crawled. The program will not keep track of this information, and developers have no access to it either.

4.18 Integrity Requirements:

This is not applicable since this program does not rely on stored data any result produced comes from the webpage which is crawled.

4.19 Privacy Requirements:

The product shall notify customers of changes to its information policy.

4.20 Audit Requirements:

Not applicable.

4.21 Immunity Requirements:

Not applicable.

5 16. Cultural Requirements

5.1 Cultural Requirements:

The webcrawler can be used by anyone regardless of his or her cultural background.

6 17. Legal Requirements

6.1 Compliance Requirements:

Not applicable.

6.2 Standards Requirements:

The webcrawler complies with all software development standards.

7 Project Issues

7.1 18. Open Issues

- Parts of user interface not known.
- Testing of user interface not known.
- Updating user interface.

7.2 19. Off-the-Shelf Solutions

Ready-Made Products: Not applicable.

7.3 Reusable Components:

Not applicable.

7.4 Products That Can Be Copied:

There are other WebCrawlers that are available, which we can use as a foundation for our program. Its specification can cut our analysis effort by a lot.

7.5 20. New Problems

Effects on the Current Environment: Not applicable.

7.6 Effects on the Installed Systems:

Not applicable.

7.7 Potential User Problems:

Not applicable.

7.8 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Not applicable.

7.9 Follow-Up Problems:

Making changes to the webcrawler as more changes occur

7.10 21. Tasks

- Work on prototype and get feedback from supervisor.
- Discuss missing requirements and add them.
- After getting feedback on prototype, work on revisions.

7.11 22. Migration to the New Product

Not applicable.

7.12 23. Risks

7.13 Loss of Personnel:

The loss of one member would be bad since we are such a small group. Each group member has been assigned tasks of equal value. We are having weekly meetings to share detailed progress reports as a precautionary measure.

7.14 Learning of the software tools:

Some of the software tools are new to the team members, so in order to successfully complete implementation of the program and meet project deadlines it is imperative that the members of the group learn to use these tools in a timely manner.

7.15 24. Costs

This program is not for profit.

7.16 25. User Documentation and Training

Completed course in fundamentals of software development which documentation was taught.

7.17 26. Waiting Room

Not applicable

7.18 27. Ideas for Solutions

Not applicable