DEVELOPMENT OF FUZZY BASED WEBSITE QUALITY ASSURANCE SYSTEM

BY

Odunayo Adesola ARIBISALA

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF PHYSICAL SCIENCES

AHMADU BELLO UNIVERSITY,

ZARIA, NIGERIA

November, 2016

.

DEVELOPMENT OF FUZZY BASED WEBSITE QUALITY ASSURANCE SYSTEM

BY

ODUNAYO ADESOLA ARIBISALA **B.Sc. (OAU)**

P15SCMT8007

A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY, IN PARTIAL FULFILLMENT FOR THE AWARD OFMASTER DEGREE IN COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE,

AHMADU BELLO UNIVERSITY,

ZARIA, NIGERIA

NOVEMBER, 2016

DECLARATION

I declare that the work in this dissertation titled "Development Of Fuzzy-Based Website Quality Assurance System" has been carried out in the Department of Computer Science, under the supervision of Prof. A.A. Obiniyi and Prof. O.S. Adewale. The information contained in the literature has been duly acknowledged in the text and a list of references provided. No part of this dissertation was previously presented for another degree or diploma at this or any other Institution.

ARIBISALA, Odunayo Adesola _____

Name of Student Signature / Date

CERTIFICATION

This dissertation entitled "Development of Fuzzy-Based Website Quality Assurance System" by Aribisala Odunayo Adesola meets the regulations governing the award of the degree of Master of Science of the Ahmadu Bello University, and is approved for the contribution to knowledge and literary presentation.

Prof. A.A. Obiniyi _____ Date_____

Chairman, Supervisory Committee

Prof. O.S. Adewale _____ Date _____

Member, Supervisory Committee

Prof. S.B. Junaidu_____ Date _____

Head of Department

Prof. K. Bala _____ Date _____

Dean, School of Postgraduate Studies

DEDICATION

This dissertation is dedicated to my late father, Comrade Chief Adesuyi Joshua Aribisala. Keep resting on Dad.

ACKNOWLEDGEMENT

My sincere and greatest gratitude goes to God Almighty for giving me the grace to come this far.

My special gratitude goes to my supervisors, Prof. A.A. Obiniyi and Prof. O.S. Adewale for their contribution, corrections, and for being very patient with me. God bless you sirs.

My thanks also goes to the Head of Department, of the Department of Computer Science, the PG coordinator, the seminar coordinator and all the staff in department of Computer Science.

I will eternally be grateful to my sweet husband, Mr Ifeoluwapo Aina for his support. I also appreciate my beautiful princesses 'Babara Aina and Momore Aina (God's precious gifts to me during my programme) for being gentle and understanding children. Ours is a formidable union built to last.

My unreserved and golden appreciation goes to my wonderful mother, Mrs Modupeoluwa Aribisala, who filled the space my dad left and stood by me through thick and thin to see that my education does not suffer. I appreciate my lovely siblings Bukola and Adedayo for being there always. I pray that God will continue to keep our bond of love.

My gratitude also goes to my friends / coursemates, Toyin PeteRock, Esther Oludare, Gloria Shidali, John (able class rep) and Onyinye Okpoko for their immense contributions. Dr Emmanuel Ibam, Mr John Shonubi, Abimbola Oluwagbuyi, Adedotun Adenigbo, Julius Iyela, Omolara Aluko, Opeyemi Adekugbe, Abimbola Fajulugbe, Ademola Ayinla, Bidemi Ayeye, Kayode Owoseni, Victory Subair, Esther Babalola, Alaimi, Mariam (just to mention a few), I am grateful for the different roles you all played to ensure I finish my programme successfully, God bless you real good.

Thank you all, may Almighty God reward you.

ABSTRACT

Internet has become ubiquitous and organizations have moved from having their information only in physical files to having them online via their websites. Over the years, different websites' evaluation tools have been developed to assist web administrators with the maintenance of their websites. Before a proper website assessment can be done with these evaluation tools, the web administrator would have to access many of these evaluation tools from different locations. Reasons been that, different tools were developed to measure different parameters that contribute to having a good quality website. Later, Fuzz-web was developed to capture more parameters but with a limitation of the system having to depend on some of these evaluation tools for inputs, meaning unavailability of these tools will lead to the failure of fuzz-web. To overcome these problems, this research establishes a Fuzzy-Based Website Quality Assurance System using fuzzy logic principles and Java programming language to develop the system. The result of the developed system is the detection of broken links, slow loading pages, Hypertext Transfer Protocol related errors in websites, and the overall quality status of the website.The developed system eradicates the dependence on external evaluation tools and with the information provided, allows web administrators to be proactive in amending errors.

TABLE OF CONTENTS

COVER PAGE……..……..…….……i

TITLE PAGE……..……..…….……ii

DECLARATION……..…….……..iii

CERTIFICATION……..……..…… ……..……...iv

DEDICATION……..……..…… ……..……v

ACKNOWLEDGEMENT……..……..… ……..……..vi

ABSTRACT……..……..……vii

TABLE OF CONTENTS……..……..……viii

LIST OF TABLES……..……....xii

LIST OF FIGURES…….....xiii

LIST OF APPENDICES……..…..xiv

LIST OF ABBREVIATION……...xv

CHAPTER ONE: GENERAL INTRODUCTION……..1

1.1 Background to the Study…….....1

1.2 Problem Statements…….….4

1.3 Motivation for the Study……..4

1.4 Aim and Objectives of the Study…….5

1.5 Methodology…….….5

1.6 Scope of the Study…….....6

1.7 Contribution of the Study to Knowledge……....6

CHAPTER TWO: LITERATURE REVIEW……..7

2.1 Introduction……..…7

2.2 Information Retrieval and the Web……...7

2.3 Structure of the Web……...8

2.4 History of Hypertext……..10

2.4.1 HyperText Transfer Protocol(HTTP)…….……...11

2.4.2 HyperText Markup Language (HTML)…….……...12

2.4.3 Some Weaknesses of the WWW…….……..12

2.5 Some Common Problems with Websites……..13

2.5.1 Broken Links……...……..13

2.5.2 Slow Loading Page……...14

2.5.3 Error 404……...15

2.5.4 Error 500……...15

2.6 Website Maintenance……....15

2.7 Website Quality Assurance……...16

2.7.1 Data Collection…….16

2.7.2 Data Analysis……..……..17

2.8 Introduction to Fuzzy Logic…….……..…17

2.8.1 Crisp Set Theory……..….18

2.8.2 Fuzzy Set Theory……..18

2.8.3 Membership Functions……..19

2.8.4 Linguistic Variables……..19

2.8.5 Fuzzy Logic Systems (FLS)…….....20

2.9 Related Work…….…….21

2.9.1 Xenu or Xenu's Link Sleuth…….…….22

2.9.2 Webpage Speed Analyser……..22

2.9.3 Fuzz-web……...23

2.10 Proposed Improvement(s) to the Related Works..……24

CHAPTER THREE: DESIGN OF FUZZY-BASED WEBSITE QUALITY ASSURANCE SYSTEM……..25

3.1 Introduction……25

3.2 Fuzzy-Based Website Quality Assurance System……....25

3.2.1 Defining the Inputs/Output Membership Function for FWQAS……...25

3.2.2 Set-up Fuzzy Rule Base for FWQAS……29

3.2.3 Defuzzify the Output for FWQAS……30

3.3 FWQAS Architecture…….....……..30

3.4 System Requirement……..……31

3.4.1 Functional Requirement for a Website Quality Assurance System…….…32

3.4.2 Non-Functional Requirements for a Website Quality Assurance System……33

3.5 System Behavioural Pattern……...……..33

3.6 User Interface Design…….……35

3.7 Implementation Language…….……35

3.7.1 Reasons for the Choice of Java as Implementation Language……..36

3.8 FWQAS Requirement……36

3.8.1 Hardware Requirement……..36

3.8.2 Software Requirement……...37

3.9 Algorithm……..…..37

3.10 Flowchart……..38

CHAPTER FOUR: IMPLEMENTATION OF FUZZY BASED WEBSITE QUALITY ASSURANCE SYSTEM……..40

4.1 Introduction……40

4.2 System Testing and Results……...40

4.3 Comparison with Related Work……...44

CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATIONS……48

5.1 Introduction……48

5.2 Summary……48

5.3 Conclusions……48

5.4 Future Works……49

References……50

APPENDIX……...55

LIST OF TABLES

Table 4.1:Comparison Table 46

LIST OF FIGURES

Figure 2.1 : Architecture of the Web (Berners-Lee *et al.*, 1992)…….9

Figure 2.2 : The basic Hypertext Model (Berners-Lee *et al.*, 1992)……..10

Figure 2.3 : Fuzzy Logic System (Ahmad *et al.,* 2011)……..…….…. 21

Figure 2.4 : Fuzz-web Flowchart for Assessment of Website (Rekik and Kallel, 2013)…… 23

Figure 3.1 : Membership Function of the input variable “Broken links”…… 26

Figure 3.2 : Membership Function of the input variable “Errors”……27

Figure 3.3 : Membership Function of the input variable “Speed load”…… 28

Figure 3.4 : Membership Function of the output variable “Website Quality”……...….. 29

Figure 3.5 : Architecture of FWQAS traced out from Fuzz-web …… … 30

Figure 3.6 : Architecture of FWQAS ……..…31

Figure 3.7 : Flowchart for the FWQAS ……...…...39

Figure 4.1 : Mainpage of the FWQAS ……..……40

Figure 4.2 : Menu bar……..……41

Figure 4.3 : Check Link page…….……42

Figure 4.4 : Results of individual links checked…….....43

Figure 4.5 : Overall result of links checked displayed ……44

Figure 4.6 : Screenshot of Xenu's result…….....45

Figure 4.7 : Webpage Speed Analyser's result (GTMetrix)…..…….... ..45

Figure 4.8: Comparison chart ……...…….....45

LIST OF APPENDICES

Appendix I: Java Source Codes for FWQAS ……..57

LIST OF ABBREVIATIONS

API: Application Programming Interface

FWQAS: Fuzzy-based Website Quality Assurance System

HTTP: HyperText Transfer Protocol

CAGR: Compound Annual Growth Rate

IT: Information Technology

WWW: World Wide Web

HTML: HyperText Markup Language

SGML: Standard Generalised Markup Language

URL: Uniform Resource Locator

TWT: Tolerable Wait Time

QA: Quality Assurance

MF: Membership Function

CMMI: Capability Maturity Model Integration

RAM: Random Access Memory

CHAPTER ONE

GENERAL INTRODUCTION

1.1 Background to the Study

According to Encarta dictionaries (2009), information is defined as the knowledge acquired or supplied about something or somebody; the collected facts and data about a specific subject; the communication of facts and knowledge. It is often said that information is power. The importance of information to individuals and organizations and the need to manage it well is growing rapidly. Now more than ever, we need to understand the critical role information plays in some aspects of business and life. It drives our communication, our decision making and our reactions to the entire environment. It is no longer news that many organizations are moving information about their products and services online, because they have realized that they have to be online to compete in this fast-paced information age.

According to Brahima (2014), recent statistics on Information Technology show that by the end of 2014, there would be almost 3 billion Internet users, two-thirds of which will be coming from the developing countries, and that the number of mobile-broadband subscriptions would reach 2.3 billion globally. Fifty-five per cent of these subscriptions are expected to be in the developing countries. According to Kende (2014), the global proportion of people using the Internet has risen at a compound annual growth rate (CAGR) of 12% in the period 2008-2012, reaching a level of 37.9% of the global population in 2013. The increase in usage is particularly evident in those regions that had lower levels of Internet usage in 2008, with the comparable growth rates for the period in sub-Saharan Africa and emerging Asia-Pacific exceeding 20%. With these research results, it is obvious that the number of Internet users has increased and still increasing.

People are more informed than ever these days. People put in a lot of time and effort to conduct detailed research before making an online purchase. In fact, 63% of people use the Internet as their first resource when looking for a local service or product, according to a study by Nielsen (2009). According to Discover Small Business Watch Survey (2012), 3,000 people were surveyed and 47% of them responded that they are more likely to purchase services or products from a small business with a website. This indicates that people are more willing to trust businesses that have a website.

Additionally, putting your business online increases its chances of being listed in search sites, like Google, Yahoo and Bing. This will help people find your business more easily online and direct them to your website or store**.** Considering the testimony of Michelle Braun, owner of Final Touch Housekeeping in Aurora, Colo., inVistaprint (2012), she saw an increase in business after creating a website. Braun initially used flyers and other printed materials to promote her business, but customers kept asking for her web address andshe quickly got a successful website up and running in July 2009. She witnessed an increase in the number of deals she closed, within a year she was able to do bids for 500 or 600 people, all contacted through her website.

According to Ghandour (2011), Website is a sales channel (sometimes a business sole's interface) between the business, customers and the world at large. In the contemporary competitive business environment, an innovative, well-designed and managed website can provide the advantage abusiness needs to conduct its activities successfully. Having a website attracts some business benefits; it is a great way to increase business sales and leads. Websites broaden the market reach for business and provide customers 24/7 access to products and/or services. For small-to-medium sized businesses, a website can level the playing field when competing with larger companies. Having a website tells your customers you are committed to your business and to competing with other companies. A website also allows you to make connections with people and one of the most

important aspects of developing your business is networking. Any good salesperson can tell you it's not what you know, it's who you know. A website can allow you to make contacts 24 hours a day and for a very limited cost.

However, it is not sufficient to just have a website; there is need for the website to be of high quality. Talking of quality, web pages should not be loading slowly, errors should be avoided as much as possible, just to mention a few of the standards expected of a website. Studies have shown that users who do a keyword search will spend an average of 12 seconds on any of the returned web pages (Weinreich*et al*., 2008). This implies that businesses have a very short period of time to convince a potential customer to stay on their website**.** Consider the following scenario:

a. A potential buyer of a company's product tries to browse through the company's online catalogue, but the product's gallery section of the website has an error, so the potential buyer simply moves on to the company's competitor website, browses their catalogue and makes his purchase.
b. A potential customer needs a particular service or needs to make a purchase, decides to browse through company's sites offering the desired service / product, only to find out that a particular site is not opening on time while other competitors' sites have opened and he has checked what he needs, he definitely would have lost interest in the site that is not opening even if they have the best offer compared to their competitors.

In the scenarios considered in a and b above, the problem was with the maintenance of the website. Most effort in this regard has always been manual, such that a resource or team, usually from the IT department of an organisation is saddled with the responsibility of ensuring the website is well maintained but this can become overwhelming when web pages and/or content increases. When this happens, it becomes a tedious task for resource or team in charge to cover all the pages of the

website and make sure that they are running without errors. The Information Technology (IT) staff may not be aware of the error(s) till it becomes an issue, which may have resulted in loss of customers or customer's confidence in the company or loss of revenue altogether. Over the years, some developers introduced different tools that can be used by web administrators to maintain their websites. But the limitation with these tools is that each tool addresses a particular problem, for example, a tool might be used to identify broken links on websites while another might be for identifying the speed of webpage load etc. Meaning, for a web administrator, to identify more than one problem on a particular website at a time, he /she will need to have access to different tools. What if any of these tools becomes unavailable, it means the web administrator will not be able to run a broad maintenance check on the website.

1.2 Problem Statements

Can a system be developed that:

a. Assesses a website by putting into consideration more parameters that contribute to having a good quality website?

b. Assesses a website without depending on external evaluation tools?

c. Allows for interoperability between components that contribute to having a good quality website?

1.3 Motivation For The Study

Fuzzy logic is a tool that enables the solution of difficult simulated problems with many inputs and output variables that is able to give results in the form of recommendation for a specific interval of output state. No wonder that it is in use in the field of Artificial intelligence to develop various

fuzzy control systems. This powerful tool can likewise be adopted to develop a self-sufficient system that can assist web administrators to assess their websites for quality status rating per time.

1.4Aim and Objectives of the Study

The aim of this research is to develop a system that answers all the questions asked in the problem statements section.The objectives of the research are to:

a. Formulate fuzzy rules to guide the decision making of the system
b. Design a fuzzy based website quality Assurance System
c. Implement the system

1.5 Methodology

The approaches that will be applied in achieving our aim and objectives in this dissertation are to:

a. Review related work on website quality assurance systems.
b. Identify the input parameters, output parameters, their linguistic terms and use MATLAB to generate their membership functions.
c. Formulate the fuzzy rules to guide the decision making of the system
d. Develop and implement the proposed system using Java as the programming language and Netbeans IDE 8.1.
e. Summarise the developed system's result and compare the result with previous workpresented in tabular and graphical forms.

1.6 Scope of the Study

The work is limited to developing a fuzzy based website quality Assurance System for a business / organisation websites, by combining into a system, three parameters (checks for broken links, webpage speed load and errors) that contribute to having a high quality website.

1.7 Contribution of the Study to Knowledge

This research hopes to make a meaningful contribution in the aspect of website maintenance. A fuzzy logic technique is adopted because of its capability in bringing an expert's skill to solve problems of incomplete knowledge or imprecise data. Its contributions are:

a. Production of a system that does not depend on external evaluation tools before identifying broken links, http response status related errors, webpage speed on a website.
b. Provision of a system that uses the principles of fuzzy logic to ascertain the overall quality status of website(s), thereby allowing for pro-activeness in amending errors on time.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter reviews several important concepts that are related to the work presented in this research work, in order to have better understanding of the concept.

2.2 Information Retrieval and the Web

Until the early 1980's information retrieval was viewed as a narrow area of interest by librarians and information experts. People have dreamed of a universal information database since late nineteen forties. In this collection, not only would the data be accessible to people around the world, but it would also easily link to other pieces of information, so that only the most important data would be quickly found by a user (Hughes, 1993). The World Wide Web was introduced in 1990 by the European Laboratory for Particle Physics, as physicists' innovative ways of sharing information (Zeltser, 2015). The concept involved the ability of people working in different locations to learn what each other were working on, through examination of a hypertextual document, accessible on the Internet. Creation of the World Wide Web is credited to Tim Berners-Lee (Harper, 1999). His original vision for the web was to have "a simple scheme to incorporate several different servers of machine-stored information already available at cern". This "scheme" was to use hypertext to provide a single user-interface tomany large classes of stored information such as reports, notes, data-bases, computer documentation and on-line systems help (Relihan, 1995). The official description defines the Web as a "wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents"(Hughes, 1994). It can simply be said to be an Internet-based computer network that allows users on one computer to access information stored on another through a series of interconnected networks. The content of

information present in the web is diverse not only with respect to quality but also in language, range of vocabulary and type (Gromov, 1998).

In 1989, the Web quickly gained great popularity among Internet users. For instance, at 11:22 am of April 12, 1995, the Web server at the SEAS of the University of Pennsylvania responded to 128 requests in one minute. Between 10:00 and 11:00, it responded to 5086 requests in one hour or about 84 per minute(Buchholtz, 1995). At the time, that was a lot! Even years after its creation, the Web continued to expand. According to New Scientist Magazine (1994), in December 1994 the Web was "growing at roughly 1 per cent a day — a doubling period of less than 10 weeks". With the expansion of the Web in an incredible rate, no one knows what awaits this universal information sharing platform as it permeates all aspects of our lives.

2.3Structure of the Web

World Wide Web (WWW) introduced the principle of universal readership, which stated that networked information should be accessible from any type of computer in any country with one easy to use program (Blanchard, 2013). With the principle stated above, if information is available, then any (authorized) person should be able to access it from anywhere in the world. The client-server model was used in the implementation of the web, where a user relies on a program (the client) to connect to a remote machine (the server), where the data is stored. Thearchitecture of the Web is represented in figure 2.1

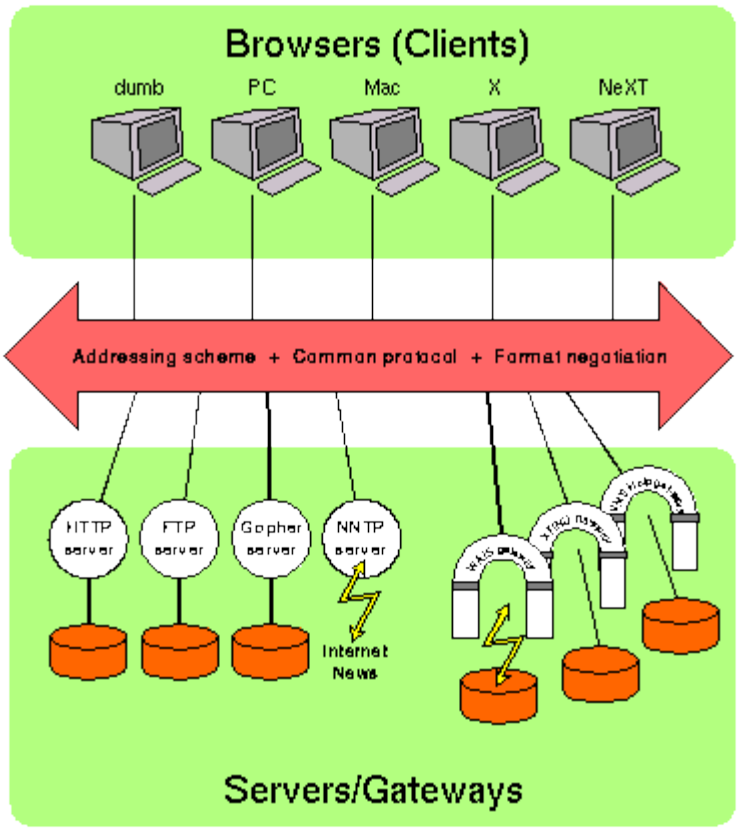


Figure 2.1: Architecture of the Web (Berners-Lee*et al*., 1992)

One of the main features of Web documents is their *hypertext* structure (see Figure 2.2). For instance, a particular reference can be represented by underlined text or an icon. "The user clicks on it with the mouse, and the referenced document appears"(Berners-Lee *et al*., 1992). This method makes copying of information unnecessary: data needs only to be stored once, and all referenced to it can be linked to the original document.

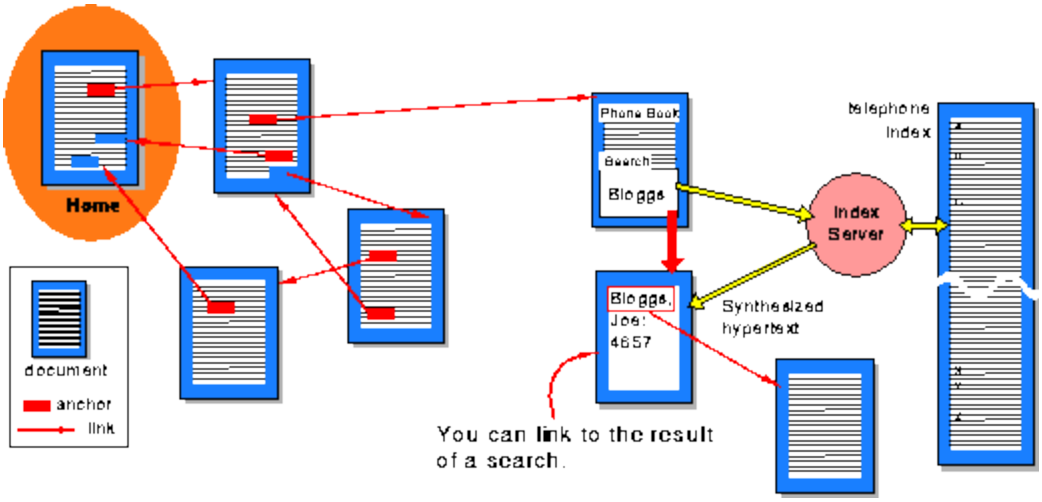


Figure 2.2: The basic Hypertext Model (Berners-Lee*et al.*, 1992)

2.4History of Hypertext

The term *hypertext* was coined by Ted Nelson in his book "Literary Machines," where it was defined as "non-sequential writing," and only later it became considered a medium limited to computers (Hirmes, 1993). By 1945, it was realized that an era of information was approachingand "memex" was written about, a conceptual machine that could store vast amounts of information, in which a user had the ability to create information "trails": links of related text and illustrations. This trail could then be stored and used for future reference (Hirmes, 1993).

Although "memex" was never implemented, in 1960 it inspired Ted Nelson to develop the modern version of hypertext. In his newsletter, Nelson wrote that the future of humanity is at the interactive computer screen, that the new writing and movies will be interactive and interlinked(Nelson, 1994). Learning from Ted Nelson's ideas, Tim Berners-Lee of CERN conceived the idea of the World-Wide Web in 1989 (Relihan, 1995)

2.4.1 HyperText Transfer Protocol (HTTP)

Hypertext transfer protocol is the underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions web servers and browsers should take in response to various commands. For example, when you enter a Uniform Resource Locator (URL) in your browser, this actually sends an HTTP command to the web server directing it to fetch and transmit the requested web page (Jolliffe, 2001).

HTTP is called a *stateless* protocol because each command is executed independently, without any knowledge of the commands that came before it. This is the main reason why it is difficult to implement Websites that react intelligently to user input. This shortcoming of HTTP is being addressed in a number of new technologies, including ActiveX, Java, Javascript and cookies (Tipton, 2014).

The innovations that Berners-Lee added to the Internet to create the World Wide Web had two fundamental dimensions: connectivity and interface. He invented a new protocol for computers to speak as they exchanged hypermedia documents. This Hypertext Transfer Protocol (HTTP) made it very easy for any computer on the Internet to safely offer its collection of documents into the greater whole; using HTTP, a computer that asked for a file from another computer would know when it received the file, if it was a picture, a movie, or a spoken word.

With this feature of HTTP, the Internet began to reflect an important truth – retrieving a file's data is almost useless unless you know what kind of data it is. In a sea of web documents, it's impossible to know in advance what a document is – it could be almost anything – but the web understands "data types" and passes that information along. (Pesce, 1993)

2.4.2 HyperText Markup Language (HTML)

HTML is the authoring software language used on the World Wide Web and it is used for creating webpages(Tomei, 2010).Tim Berners-Lee was the primary author of html, assisted by his colleagues at CERN, an international scientific organization based in Geneva, Switzerland. It is the official language of the World Wide Web and was first conceived in 1990. HTML is a text-based markup language, which means that it contains tags that describe a document's content and appearance. As the basis for creating HTML, Berners-Lee used Standard Generalised Markup Language(SGML) which is a metalanguage used to create document markup languages (Pratt, 2014). However, it was quickly realized by those outside of the discipline of scientific documentation that HTML was relatively easy to learn, was self-contained and lent itself to a number of other applications. With the evolution of the World Wide Web, HTML began to proliferate and quickly spilled over into the mainstream.

2.4.3 Some Weaknesses of the WWW

The World-Wide Web began as a set of simple protocols and formats. As time passed, the web "began to be used as a test bed for various sophisticated hypermedia and information retrieval concepts." (Maurer, 1996) Unfortunately, these concepts were quickly absorbed by the general WWW community. This means that experimental extensions of dubious use are now established as parts of the web.

Another flaw in the current structure of the WWW is the presence of many hypertext links that point to no longer existent documents. This occurs when authors rename or delete their works from the Web (Relihan *et al.,* 1995). Since the system has no way of registering links to one's document, an author cannot notify his readers of the reorganization.

2.5Some Common Problems with Websites

Having discussed the development of the web through the years, discussed in the following sections are some of the problems with web in present days.

2.5.1 Broken Links

Links are the programmatic commands to 'jump-to-another-page' in a web browser. Every webpage is filled with dozens of links, each sending a user to some related web page or picture/file. A link is identified if the mouse pointer changes to a pointing finger when the mouse is placed on its text. Clicking a link or hyperlink is all it takes to activate the jump command. When you click with the pointing finger mouse shape, the hyperlink commands your web browser to load the target web page, ideally within seconds. Indeed, hyperlinks are the very core of how the World Wide Web functions. Millions of hyperlinks are how people jump and discover the massive content of the Web.

According to Boswell (2010), a link is said to be broken when the link or hyperlink no longer points to its original destination. Broken link also known as a dead link is a link on a web page that no longer works due to one or more of the reasons below.

a. An improper Uniform Resource Locator (URL) entered for the link by the website owner.
b. The destination website permanently moved or no longer exists
c. The user has software or is behind a firewall that blocks access to the destination website.
d. The website owner linked to a site that is behind a firewall that does not allow outside access (such as an Intranet site or a restricted access area on a website).

Broken links can be problematic for users of a website as they are unable to access the desired resource and may decide to make use of another site to find the necessary information or service.

A site that hasn't been updated or checked for a long time may suffer from link rot, which is a term used to describe a site with dozens of broken links.

2.5.2 Slow Loading Page

Previous research has shown that user frustration increases when page load times exceed eight to 10 seconds, without feedback (Bouch*et al.,* 2000; King, 2003). Newer evidence shows that broadband users are less tolerant of web page delays than narrowband users. A Jupiter Research survey found that 33% of broadband shoppers are unwilling to wait more than four seconds for a web page to load, whereas 43% of narrowband users will not wait more than six seconds (Akamai, 2006).

In a 2004 study, Fiona Nah found that the Tolerable Wait Time (TWT) on non-working links without feedback peaked at between 5 to 8 seconds (Nah, 2004) and this was improved byadding a means of feedback to the user like a progress bar whichpushed the TWT to an average of 38 seconds. Subsequent attempts at non-working links revealed lower TWTs peaks at 2 to 3 seconds without feedback. Nah concluded that the TWT of web users peaks at about 2 seconds. With regard to behavioral intentions to return to a site, Dennis Galletta and others found that they level out at 4 or more seconds and attitudes flatten at 8 or more seconds (Galletta *et al.*, 2004)

Even small changes in response times can have significant effects. Google found that moving from a 10-result page loading in 0.4 seconds to a 30-result page loading in 0.9 seconds decreased traffic and add revenues by 20% (Linden, 2006). When the home page of Google Maps was reduced from 100KB to 70-80KB, traffic went up by 10% in the first week, and an additional 25% in the following three weeks (Farber, 2006). Tests at Amazon revealed similar results: every 100 ms increase in load time of Amazon.com decreased sales by 1% (Kohavi and Longbotha, 2007). Experiments at Microsoft on Live Search showed that when search results pages were slowed by 1

second: (Kohavi, 2007) , queries per user declined by 1.0% and ad clicks per user declined by 1.5%.After slowing the search results page by 2 seconds: queries per user declined by 2.5% and ad clicks per user declined by 4.4%. Slow web pages lower perceived credibility (Fogg, 2001) and quality (Bouch*et al.*, 2000).

2.5.3 Error 404

A 404 is an http status code often displayed as either "Not Found 404" or "Page Not Found". It is an indication that the web server exists, but the page the web user is attempting to access, does not exist. When this occurs it is an indication that the webpage has moved, been deleted, renamed or is currently being updated. Below is an example of how a generic 404 error message may look on a web page. A 404 error is simply an automated message that shows up on many web sites; it tells the user that the webpage that corresponds to the clicked hyperlink could not be located.

2.5.4 Error 500

The 500 Internal Server Error is a very general http status code that means something has gone wrong on the web site's server but the server could not be more specific on what the exact problem is (Fisher, 2014). The 500 Internal Server Error is a *server-side* error, meaning the problem probably isn't with the computer being used or Internet connection but instead is a problem with the web site's server. Other server-side errors are 502 meaning Bad Gateway, 503 meaning Service unavailable etc.

2.6Website Maintenance

Your website is your reflexion, which makes it a key part of your business/organisation, often the ‘front of house’ and source of new leads. According to Goel (2014), first impression is the last

impression, first experience of a user with your website makes first and important impression on your business, quality and brand. Because of this, it's vital your audience find your site and have a good experience. In an age where user demands more, they no longer tolerate sites which do not perform. If it takes time to load, a user can't find what the required information on the siteor the site looks poor, the user is off! There are too many other sites offering the same information.

According to Diffily (2006), website maintenance comprises all the activities needed to ensure the operational integrity of your website or intranet. In other words, it is about doing all the things needed to make sure your site runs smoothly and according to plan. Website maintenance comprises of the following activities:

a. Website Publishing: To keep content up-to-date.
b. Website Quality Assurance: To spot errors on a site.
c. Website Feedback Monitoring: To manage communication with website visitors.
d. Website Performance Monitoring: To measure success.
e. Website Infrastructure Monitoring: To supervise hosting.
f. Change Control: To manage technical and other changes in a coordinated way.

These activities are usually carried out by members of a Website Maintenance Team.

For the sake of the thesis, the scope is to cover the website quality assurance aspect.

2. 7 Website Quality Assurance

Quality Assurance is the activity that makes sure a website is operationally sound and in conformance with an organisation's standards (Diffily, 2006). The two main activities of Website Quality Assurance are Data Collection and Data Analysis.

2.7.1 Data Collection

The aim of this task is to collect the data against which a website can be examined for issues of quality. This includes:

a. Checking for broken links
b. Checking for slow loading pages
c. Checking for errors
d. Checking that applications are functioning correctly
e. Checking that pages conform to your organisation's Web Accessibility standard (if any)
f. Checking that the Website Design standard is maintained

2.7.2 Data Analysis

Data Analysis examines all the information that has been collected and from that an Issues Log is compiled. The purpose of this log is to list items that are in violation of a Quality Assurance checkpoint, e.g. broken links, errors etc. These can then be allocated to Developers for adjustment.

2.8 Introduction to Fuzzy Logic

Fuzzy logic is a complex mathematical method that allows solving difficult simulated problems with many inputs and output variables. It is able to give results in the form of recommendation for a specific interval of output state, so it strictly distinguished from the more familiar logics such as Boolean algebra (Vydano, 2012). The concept of fuzzy sets was firstly introduced by Zadeh in 1965. The author's intention wasto create a mathematical representation for handling data

imprecision and information possessing non-statistical uncertainties. Basically, fuzzy sets are a supplement of ordinary crisp sets. To differentiate between the two paradigms andcapabilities of fuzzy sets, it is imperative to take a look at the definition of crisp sets and compare them to fuzzy set definitions.

2.8.1 Crisp Set Theory

Crisp means dichotomous,that is, yes-or-no type rather than more-or-lesstype. In traditional dual logic, for instance, a statement can be true or false—and nothing in between (Wiley, 2010). A classical (crisp) set is normally defined as a collection of elements or objects x € X that can be finite, countable, or over countable. A classical set can be described in different ways: one can either enumerate (list) the elements that belong to the set; describe the set analytically, for instance, by stating conditions for membership (A = {x | x < 5}); or define the member elements by using the characteristic function, in which 1 indicates membership and 0 non-membership.

2.8.2 Fuzzy Set Theory

Fuzzy set is a class of objects with a continuum of grades of membership. It generalizes the concept of a classical set, allowing its objects to have a degree of membership called *membership function* which assigns to each object ranging between zero and one. They relax the concept of *characteristic functions*by allowing assigning continuous values to objects of a particular universe (Zadeh, 1965). Due to this fact, it is possible to express partial affiliations to sets by computing a degree ofmembership rather than dichotomizing elements of a universe (Pedrycz and Gomide, 2007). The support of *U* denotes a set containing every element with a membership degree greater thanzero.

However, to devise a concise theory of Fuzzy logic, fuzzy logiccan be defined as an extension of Boolean logic dealing with the concept of partial truth which denotes the extent to which a proposition is true (Sheo *et al*., 2012).Fuzzy logic provides an inference morphology that enables approximate human reasoning capabilities to be applied to knowledge-based systems. The theory of fuzzy logic provides a mathematical strength to capture the uncertainties associated with human cognitive processes, such as thinking and reasoning. The development of fuzzy logic was motivated in large measure by the need for a conceptual framework which can address the issue of uncertainty and lexical imprecision. Some of the essential characteristics of fuzzy logic relate to the following:

a. In fuzzy logic, everything is a matter of degree.
b. Any logical system can be fuzzified.

2.8.3 Membership Functions

A *membership function* (MF) for a fuzzy set A on the universe of discourse X is defined as a curve that defines how each point (element of X) in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the *universe of discourse*. A membership function shows the extent to which a value from a domain is included in a fuzzy concept ((Ibam, 2011).It can be mathematically denoted as

$$\mu_A : \mathrm{X} \rightarrow [0,1].$$

This valueμ_A, called membership value or degree of membership, quantifies the grade of membership of the element in X to the fuzzy set A. There are various types of MF been used in fuzzy logic they are; triangular MF, trapezoidal MF, Gaussian MF and so on. Membership functions allow us to graphically represent a fuzzy set. In this research work, the trapezoidal MFs will be considered due to their simple formulas and computational efficiency.

2.8.4 Linguistic Variables

Linguistic variables are variables whose values are not numbers but words or sentences in a natural or artificial language. The motivation for the use of words or sentences rather than numbers is that linguistic characterizations are, in general, less specific than numerical ones (Zadeh, 1975). Often, fuzzy sets describe measurements of physical phenomena, e. g. temperature, position, or size. Mathematically, those measurements are formalized using variables. In the sense of fuzzy set theory this concept was supplemented by linguistic variables. They are based on reference variables with defined ranges, for example, $U = [0, 15]$. However, they are further specified using linguistic terms, describing their meaning or purpose. Moreover, they contain multiple linguistic states which correspond to fuzzy sets. Hence, they are an analogy of family of sets for crisp sets.

2.8.5 Fuzzy Logic Systems (FLS)

Most of the systems that are constructed based on fuzzy sets and logic has a common architecture. This FLS based on a specific lifecycle model consisting of four characteristic stages (William, 2011). As shown in Figure 2.3, the four components of a fuzzy logic system are explained below:

a. **Fuzzifier Component**: Here, the crisp input data which is captured from the environment is processed and interpreted using fuzzy sets, i.e. the appropriate linguistic variables states are identified and the degree of membership is determined.

b. **Fuzzy Inference Engine:**Fuzzy Inference Engine is conceptually very simple. It consists of an input, a processing, and an output stage. The input stage maps the inputs to the appropriate membership functions and truth values. The processing stage invokes each appropriate rule and generates a corresponding result. It then combines the results. Finally, the output stage converts the combined result back into a specific output value (Thai, 2002).There are two common inference methods, first is called Mamdani's fuzzy inference

method proposed in 1975 by Ebrahim Mamdani (Mamdani and Assilian, 1975) and the other is Takagi-Sugeno-Kang, or simply Sugeno, introduced in 1985. These two methods are the same in many respects, such as the procedure of fuzzifying the inputs and fuzzy operators. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant but Mamdani's inference expects the output membership functions to be fuzzy sets (Buttazzo*et al*., 2002).

c. **Fuzzy Rules:** these concatenate linguistic variables together with their particular states and memberships. They use the predicates and logical operators. For standard logical operations to hold, the input values can be real numbers between 0 and 1. The other difference is the use of intersection, union, and the complement operators such as "MIN(A,B), MAX(A,B) and 1-A" instead of the "A AND B, A OR B and NOT A". There are other operators other than the three stated above, but these are the most commonly used ones for science and engineering applications.

d. **Defuzzifier Component:** this is responsible for generating proper values depending on particular output variables and generating single values out of combined fuzzy sets.

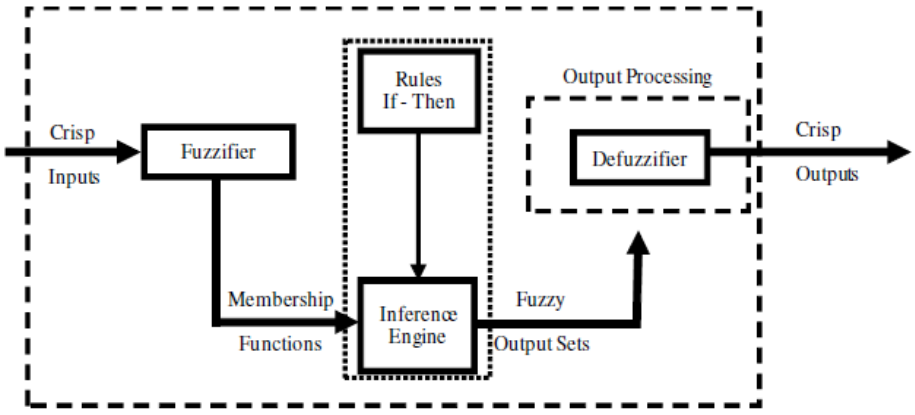


Figure 2.3: Fuzzy Logic System (Ahmad *et al.*, 2011)

2.9 Related Work

To manage a website especially a large website is practically not possible without an assisting tool. Over the years, website evaluation tools have been developed to assist in maintaining a website with each tool focusing on dealing with one parameter that contributes to having a good quality website. For a web administrator to maintain his / her website, many tools will have to be gotten to carry out this task properly. Some of these tools are discussed below:

2.9.1 Xenuor Xenu's Link Sleuth

It is a computer program that checks websites for broken hyperlinks. It is written by TilmanHausherr and it is proprietary software available at no charge. The program is named after Xenu, the Galactic Ruler from Scientology scripture. Link Sleuth runs on Microsoft Windows. The program follows links to other pages, and checks the links on those pages also, so it is possible to check an entire site for broken links in one session. Xenu displays a continuously updated list of URLs which can be sorted according to different criteria. The program utilizes a "simple, no-frills user-interface", and can help users understand how certain Web sites are structured (Steven, 2002).

Limitation of Xenu, or **Xenu's Link Sleuth**

The software only checks websites for broken links; it does not give considerations to other parameters like the speed for webpage upload etc.

2.9.2 Webpage Speed Analyser

Different tools have been developed to measure the speed of webpages download. Some of these tools are: GTmetrix, Dotcom-monitor etc.

Limitation of Webpage Speed Analyser

The focus of these tools is majorly on measuring and analyzing webpage load speed and not on the parameters that contribute to website's good quality.

2.9.3 Fuzz-Web

In 2013, Rim Rekik and Ilhem Kallel developed fuzz-web, a methodology based on fuzzy logic for assessing websites. This methodology follows the process described below.

a. The user selects and evaluates criteria for a website with the evaluation tools.
b. The measured criteria values are as inputs of the fuzzy system to perform the fuzzy computation.
c. Ranking the website.

Figure 2.4 shows clearly the method process.

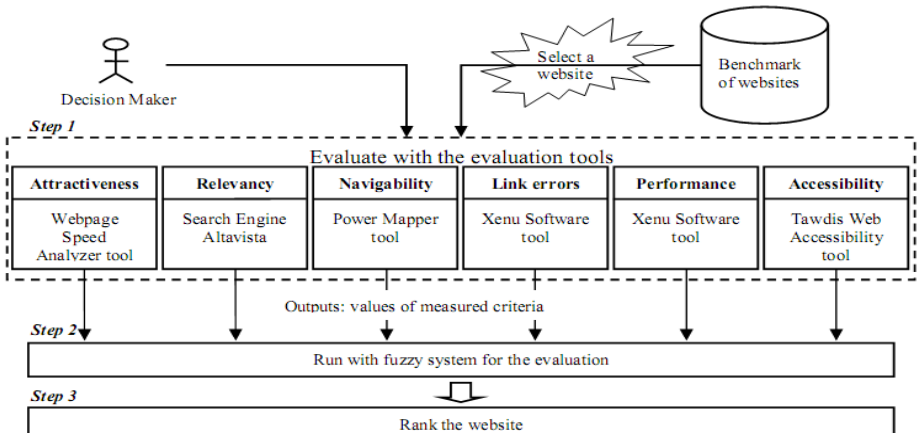


Figure 2.4: Fuzz-web flowchart for assessment of website (Rekik and Kallel, 2013)

Limitation of Fuzz-web

The system depends on evaluation tools for some of the values needed for its overall assessment, making the system not to be self-sufficient. If any of the external evaluation tools is not available

or malfunctions or probably crashes, then fuzz-web will not be able to carry out its task. In bullet points, Fuzz-web's limitations can be listed as:

a. Dependence on external evaluation tools, making the system insufficient for its task.
b. Unavailability or Failure of any of these evaluation tools, is the failure of the Fuzz-web system.

2.10 Proposed Improvement(s) to the Related Works

The proposed system named fuzzy based website quality Assurance System will address the weaknesses / limitations of Fuzz-web stated in section 2.9.3 by eradicating the dependence of the system on external evaluation tools before it can carry out its tasks, thereby allowing for interoperability between the fuzzy system and the parameters that contribute to having good quality website.

CHAPTER THREE

DESIGN OF FUZZY-BASED WEBSITE QUALITY ASSURANCE SYSTEM

3.1 Introduction

This chapter gives a breakdown of the system design process and the requirements needed for the development of a Fuzzy based website quality Assurance System.

3.2 Fuzzy-Based Website Quality Assurance System

The Fuzzy-based website quality Assurance System (FWQAS)will take into consideration more evaluation criteria all infused together in this singular system than just focusing on one aspect and leaving out other important parametersunlike mostonline evaluation tools. Also, the need for dependence on external evaluation tools before the quality assessment system can perform its major task, which is one of the limitations of Fuzz-web, would have be dealt with, making the system to be self-sufficient to perform its overall task even in the case of failure or unavailability of any of the online evaluation tools. For the purpose of this work, three parameters will be considered and infused into the system, they are: broken links, errors and webpage load speed. The overall output the system will produce is the quality of the website.

3.2.1 Defining the Inputs/Output Membership Function for FWQAS

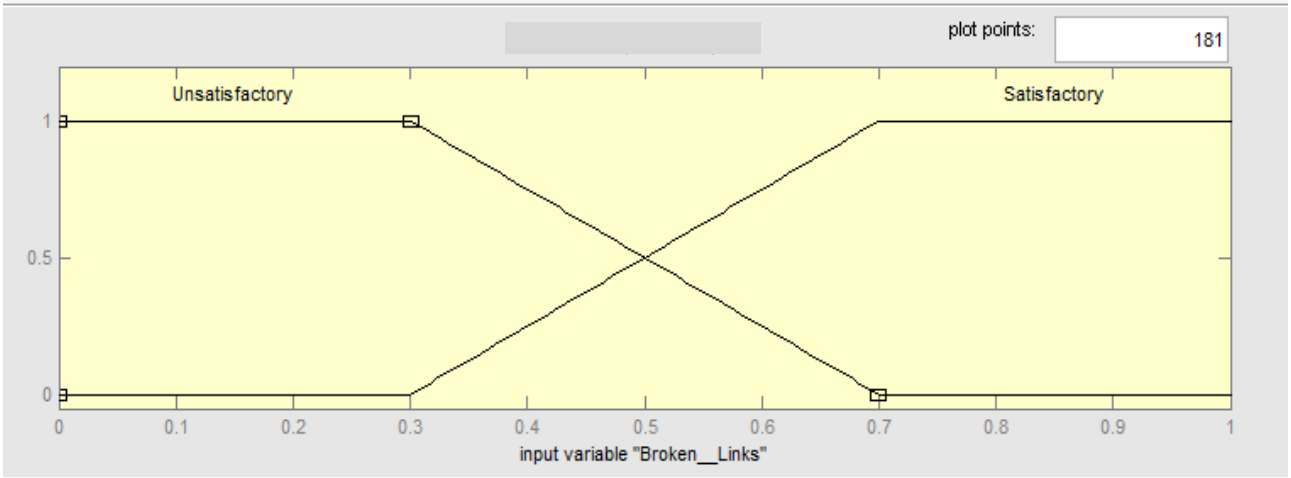
In this section we defined the fuzzy input parameters with their linguistic categories to be used. The input variables for the FWQAS are Broken links, Errors and Webpage Speed load. Two linguistic terms {unsatisfactory, satisfactory} are used to represent the input fuzzy sets while three linguistic terms represent the output {poor, fair, high} The fuzzy input parameters were fuzzified using trapezoidal membership functions. Below are the membership function distributions for the parameters defined:

Broken Link: The scale for broken links indicator is between 0 and 1. If the result tends to 1 then it reveals an acceptable score. Its computation is given as:

$$\text{Broken Link (B)} = \frac{\text{Total number of Un-broken links}}{\text{Total number of Un-broken + broken links}}$$

Figure 3.1 illustrates membership functions of the Broken link input parameter

$$\mu_{uns}(x) = \begin{cases} 1, & 0 \le x < 0.3 \\ \frac{(0.7-x)}{0.4}, & 0.3 \le x \le 0.7 \\ 0, & x > 0.7 \end{cases} \qquad \mu_{s}(x) = \begin{cases} 0, & x < 0.3 \\ \frac{(x-0.3)}{0.4}, & 0.3 \le x < 0.7 \\ 1, & 0.7 \le x < 1 \end{cases}$$



Degree of Mem

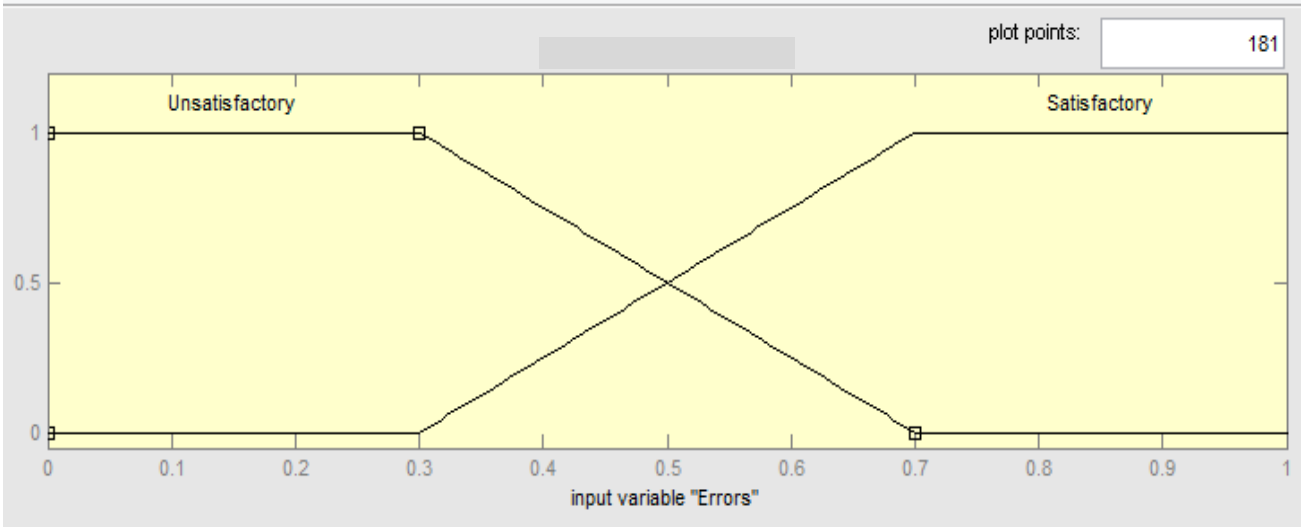
Figure 3.1: Membership Function of the input variable "Broken links"

Errors: The scale for "Errors" indicator is between 0 and 1. If the result tends to 1 then it reveals an acceptable score. Its computation is given as:

$$\text{Errors (E)} = \frac{\text{Total number of Error-free links}}{\text{Total number of Error + Error-free links}}$$

Figure 3.2 illustrates membership functions of the "Errors" input parameter

$$\mu_{uns}(x) = \begin{cases} 1, & 0 \le x < 0.3 \\ \frac{(0.7-x)}{0.4}, & 0.3 \le x \le 0.7 \\ 0, & x > 0.7 \end{cases} \quad \mu_s(x) = \begin{cases} 0, & x < 0.3 \\ \frac{(x-0.3)}{0.4}, & 0.3 \le x < 0.7 \\ 1, & 0.7 \le x < 1 \end{cases}$$



Degree of Membershi

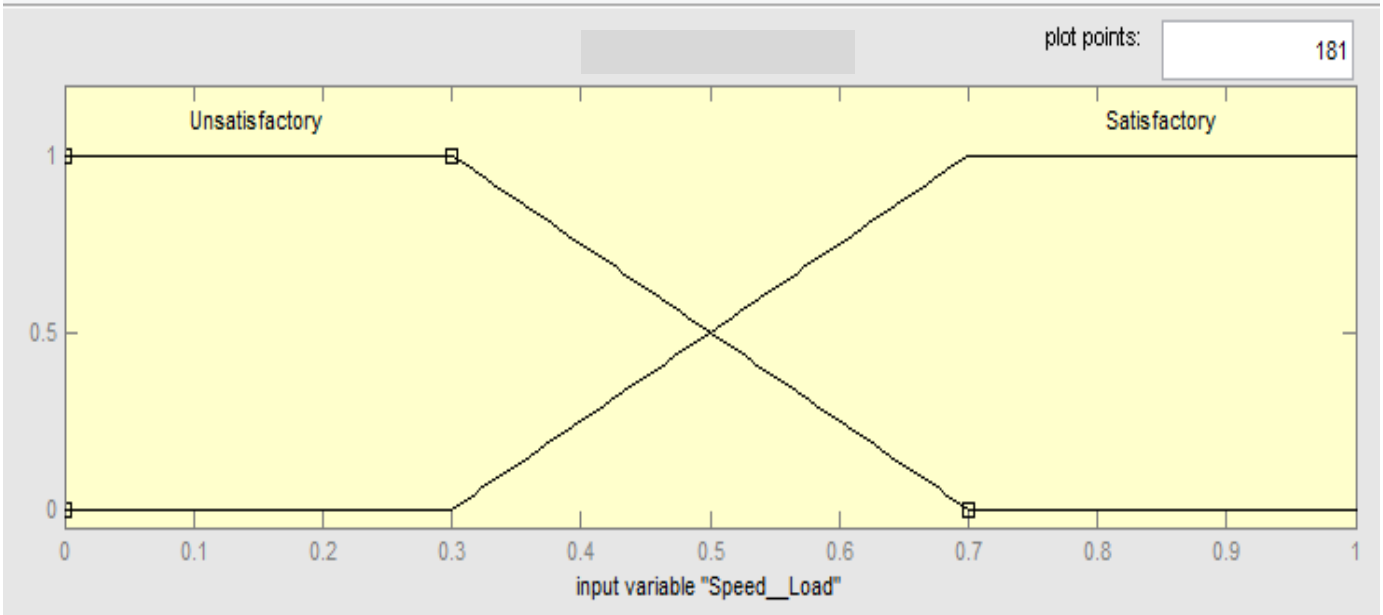
Figure 3.2: Membership Function of the input variable "Errors"

Webpage Speed Load: The scale for "Webpage Speed Load" indicator is between 0 and 1. If the result tends to 1 then it reveals an acceptable score. The benchmark of the speed within limit is 9.5 seconds according to international standard. Its computation is given as:

$$\text{Speed Load (S)} = \frac{\text{Total number of Links within speed limit}}{\text{Total number of links within speed limit + links exceeding speed limit}}$$

Figure 3.3 illustrates membership functions of the "Speed Load" input parameter

$$\mu_{uns}(x) = \begin{cases} 1, & 0 \le x < 0.3 \\ \frac{(0.7-x)}{0.4}, & 0.3 \le x \le 0.7 \\ 0, & x > 0.7 \end{cases} \quad \mu_s(x) = \begin{cases} 0, & x < 0.3 \\ \frac{(x-0.3)}{0.4}, & 0.3 \le x < 0.7 \\ 1, & 0.7 \le x < 1 \end{cases}$$



Degree of Membership

Figure 3.3: Membership Function of the input variable “Speed Load”

Website Quality *(Output)*: three linguistic terms are used to represent the output {poor, fair, high} with range from 0 to 1. Figure3.4 represents the membership function

$$\mu_p(x) = \begin{cases} 1, & 0 \le x < 0.3 \\ \frac{(0.4-x)}{0.1}, & 0.3 \le x \le 0.4 \\ 0, & x > 0.4 \end{cases} \quad \mu_H(x) = \begin{cases} 0, & x < 0.6 \\ \frac{(x-0.6)}{0.1}, & 0.6 \le x < 0.7 \\ 1, & 0.7 \le x < 1 \end{cases}$$

$$\mu_f(x) = \begin{cases} 0, & x < 0.3 \\ \frac{(0.4-x)}{0.1}, & 0.3 \le x < 0.4 \\ 1, & 0.4 \le x < 0.6 \\ \frac{(x-0.6)}{0.1}, & 0.6 \le x \le 0.7 \\ 0, & x > 0.7 \end{cases}$$

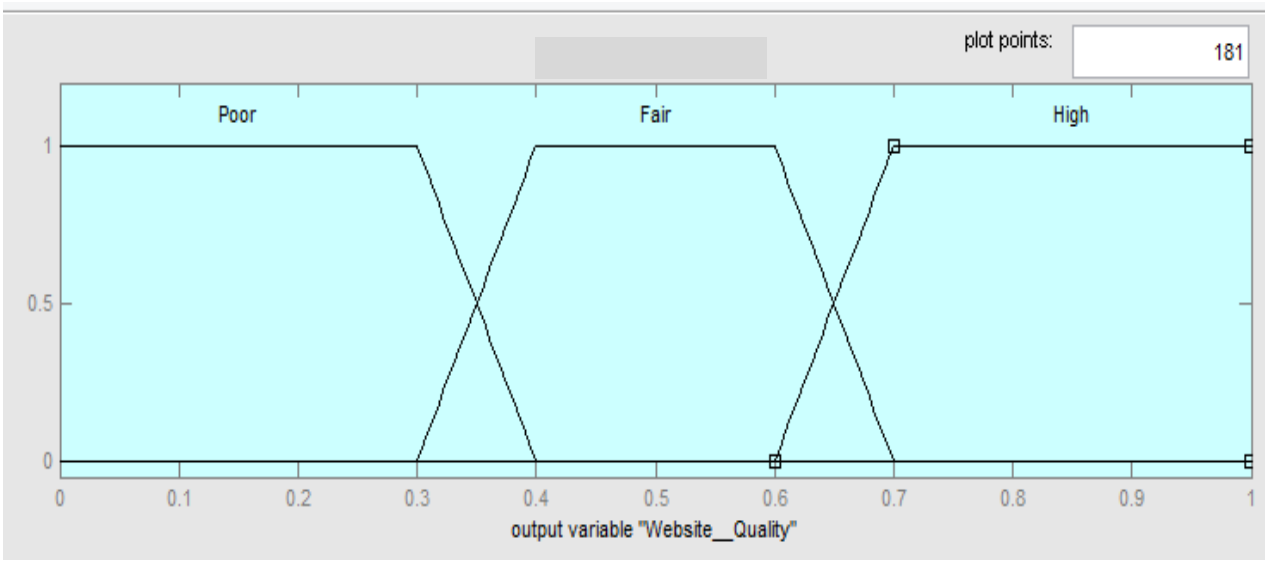


Figure 3.4: Membership Function of the output variable "Website Quality"

3.2.2 Set-up Fuzzy Rule Base for FWQAS

Now that the inputs and output membership functions have been defined in section 3.2.1, the next step is to set-up a fuzzy rule base on the basis of the number of linguistic categories defined for each of the input parameter in section 3.2.1. If *b, e, s* denote the number of linguistic categories defined for broken links, webpage speed load and errors respectively, then the fuzzy rule base will have ($b\times e\ \times s$) rules. Therefore, this research work consists of eight 8 ($2 \times 2\ \times 2$) rules of fuzzy logic. These rules usually take the form of IF – THEN rules and are obtained from human expert. Listed below are the rules where Sat, Unsat, B, E, S and Q represent satisfactory, unsatisfactory, broken links, Errors, Webpage Speedload and Website Quality respectively:

1. If (Sat(B)) and (Sat(E)) and (Sat(S)) then (High(Q))
2. If (Unsat(B)) and (Sat(E)) and (Sat(S)) then (Fair(Q))
3. If (Sat(B)) and (Unsat(E)) and (Sat(S)) then (Fair(Q))
4. If (Sat(B)) and (Sat(E)) and (Unsat(S)) then (High(Q))
5. If (Unsat(B)) and (Unsat(E)) and (Unsat(S)) then (Poor(Q))
6. If (Unsat(B)) and (Unsat(E)) and (Sat(S)) then (Poor(Q))

7. If (Unsat(B)) and (Sat(E)) and (Unsat(S)) then (Poor(Q))
8. If (Sat(B)) and (Unsat(E)) and (Unsat(S)) then (Poor(Q))

3.2.3 Defuzzify the Output for FWQAS

However, since the fuzzy rules have been stated, the next step is to use a standard fuzzy union to aggregate the membership functions of the three parameters based on the fuzzy rules and defuzzify the output using centroid method. This gives a crisp output value which is the website quality rating for the website being assessed.

3.3 FWQAS Architecture

The Figure 3.5 and 3.6 depict the architecture of the Fuzzy based website quality Assurance System. The architecture is divided into four layers, which are:

a. Layer 1: User enters Website's URL
b. Layer 2: The highlighted components are checked for on the website (Broken links, errors, page load speed)
c. Layer 3: Fuzzy System absorbs the values and processes them
d. Layer 4: Rates the website as either, poor, fair or high quality

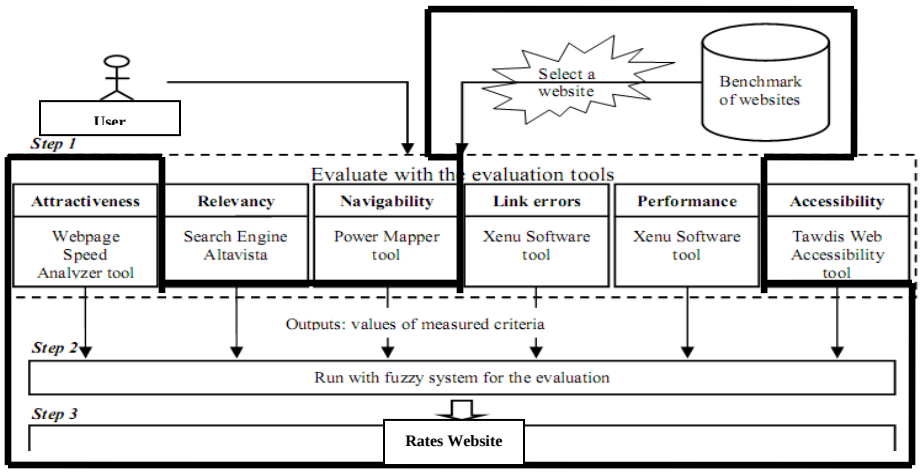


Figure 3.5: Architecture of FWQAS traced out from Fuzz-web

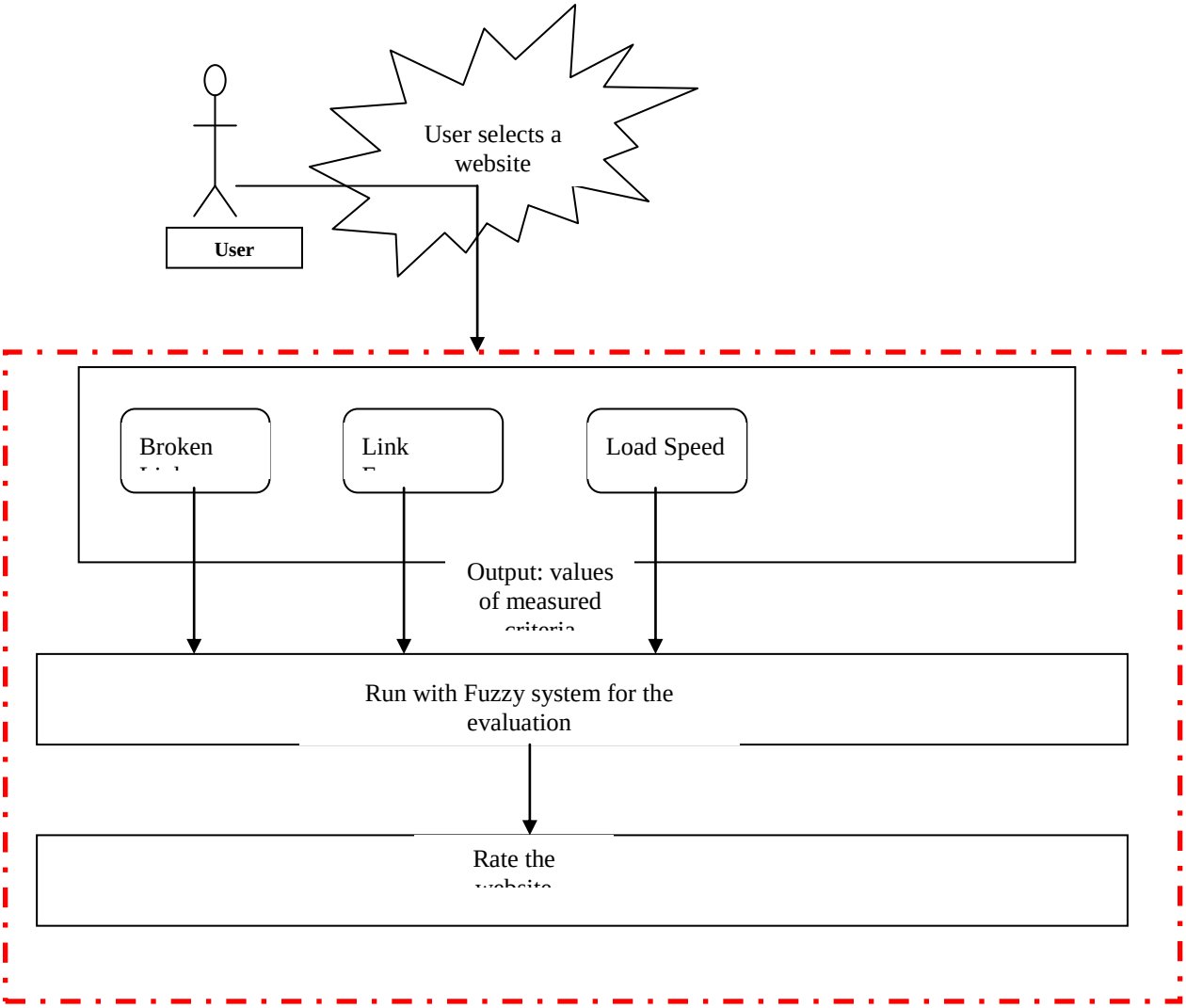


Figure 3.6: Architecture of FWQAS

3.4 SystemRequirement

A requirementis simply a statement of what the system must do or what characteristics it needs to have**.** Requirements' gathering is one of the first steps in successful software development. With respect to research work such as this, requirements gathering entails a study of the present subject area, and expectations of the new tool being developed, stated in high-level, informal terms.

Requirements are categorized into functional requirements (which state the barebones – must have functionalities for any software product), and non-functional requirements are those which dictate how the software must do its work (these include but are not limited to aesthetics, ease of use, speed, error recovery – which include ability to limit/prevent human errors, and provision meaningful and descriptive error messages if errors happen).

3.4.1 Functional Requirements for a Website Quality Assurance System

A website quality assurance systemmust be able to do/possess the following at the very least:

a. Be able to connect to a website and read its contents – this means that the system must provide an interface where a user can type in the website URL
b. Be able to read and understand a website's HTML contents – this means that the system must be able to communicate with the webserver via the hypertext transfer protocol (HTTP), and understand some basic hypertext markup language (HTML) to understand which code points to another resource (webpage or file or web media).
c. Be able to navigate from resource to resource on the web server, provided it has the proper authorization – this means the application must be able to even access secure HTTP sites, and in the case of content that can only be accessed after information is provided in HTML forms, the software must be able to fill HTML forms, and access the contents provided after the form is submitted.
d. The system must be able to produce results of its checks in a meaningful form, showing the number of inaccessible web resources, and general statistics of how long it took to access the accessible resources.

3.4.2 Non-Functional Requirements for a Website Quality Assurance System

While a Website quality assurance systemwould be considered functional if it can do the functions listed above, the points below further measure how well the software performs its tasks.

a. The system must be able to return results considerably fast
b. The system must not consume too much of system resources to drag a PC to a halt.
c. Thesystem must be able to validate a user's entry to confirm if the entry looks like a valid web address or not
d. The system must be easy to use

3.5System Behavioural Pattern

This section illustrates how the system would react when under different circumstances. It shows the states and transitions which the system can be in. Modeling the real world, analyzing a website entails the following processes:

a. Navigate to the index page of a web domain
b. Open the web page, and go through the HTML content
c. If HTML content is a link to a web resource, navigate to the link
d. If HTML content is a web form, fill the contents of the web form and submit the form, which redirects to either the same page or another web page
e. Iterate the processes from c to d till all web pages have been analyzed.

The system terminates under either of 3 conditions:

a. All the links from the home page have been analysed

b. The system loses network connection to the web server

c. The system user manually terminates the link checking process

A website quality assurance systemis basically an application, whose major goal is to connect to a web server, and analyze the HTML source of its web pages.For the solution implementation, a simple application client has been developed, which makes use of the hypertext transfer protocol to connect over the network to any web server.

To develop a client that connects to and analyses web pages, any of these 3 options could be used:

a. A command-line client

b. A web application client

c. A desktop application client

A command-line client is one whose interface for issuing commands and viewing results is the Disk Operating System (DOS) prompt. This approach, which is the lightest, fastest to develop, most secure and consumes least of system resources is the least user-friendly of the options. Non-geek users would be least attracted to this user interface.

A web application client is the most modern and user friendly of these options, which entails delivering the solution via a web browser. However, this is the least rugged, and least secure of the options. The application would have to depend on the ruggedness of the browser – which is the platform for delivering the solution. Most browsers time out after certain periods of time of waiting on a network resource. If this happens in our case, our application degrades, and would suffer time outs before it can finish analyzing all the links on a web server.

A desktop application client runs directly on the computing platform's operating system, presenting the solution in form of windows user interfaces. This solution delivery platform is more

rugged than that of the web browser, and communicates with web servers via operating system calls. With a solution delivery platform like this, we can make better use of advanced operating system optimizations such as multi-threading, parallel programming etc. Though this would require some user training to use the system, the application would be more stable and more secure.

3.6User Interface Design

A system must have inputs and outputs.The only input identified for this application is:

a. An input box to specify the website address to be analyzed

The system should display the following outputs:

a. Webpage URL
b. Link title / URL
c. HTTP response code
d. Time taken to load
e. Speed
f. Broken links

To achieve a non-cluttered interface, the output will be displayed in the form of a tableto present all the output in a compact form.

3.7 Implementation Language

The choice of programming language for the implementation of this system is limited to those languages that can produce desktop application, based on the architectural design of the system.

The major development platforms for creating desktop application clients are the Java programming platform (Java standard edition), and the .NET framework.

Java programming platform was chosen for the implementation of this system.

3.7.1 Reasons for the Choice of Java as Implementation Language

a. Java applications can be run on any platform (operating system, system architecture). Java was one of the first programming languages to implement the write-once, run-anywhere computing paradigm.

b. Java has a rich toolset of computing constructs that allows for parallel, multi-threaded programming and simultaneous process execution (which is implemented in thread).

c. Java has an active user community that can provide support in cases where suggestions and help are needed in extending the application.

d. Java, and its associated APIs are free to use, and there are no complicated licensing requirements or legal implications, which perfectly suits research activity like this

3.8FWQAS Requirement

The system requirement are the minimum hardware and software requirement for the development and implementation of the software been designed. Below are the minimum hardware and software for the development.

3.8.1Hardware Requirement

Below are hardware devices that will be needed for the system.

a. A Personal Computer with Dual Core processor (preferably).

b. 1G RAM (Random Access Memory)

c. A stable internet connection

3.8.2Software Requirement

a. Minimum of window XP operating system, ideal for remote and standalone computer.

b. Java runtime environment

3.9Algorithm

The algorithm for the system being develop is generated as follows;

a. Input into the system a valid website's URL
b. System connects to the website's index page if it is a valid URL that was entered, else the system terminates and gives an error message
c. Checks for links on the index page.
d. Searches for the last child link of the first link on the index page
e. Checks for the following parameters: load speed, broken links, http status codes error
f. Displays the result of the above listed parameters on the homepage of the application..
g. System goes to the parent link if it exists and repeats steps e and f
h. If no parent link exists, system opens the next link on the index page and searches for the last child link
i. If there is no next link on the index page, go to step k else
j. Repeat steps e to h
k. Calculate the fuzzy model Q using the values of the parameters listed in step e
l. System displays overall quality rate using the value of Q (above) as high, fair or poor.

3.10 Flowchart

The flowchart presented in Figure 3.7 is a guide used in accomplishing the goals and objectives of this dissertation:

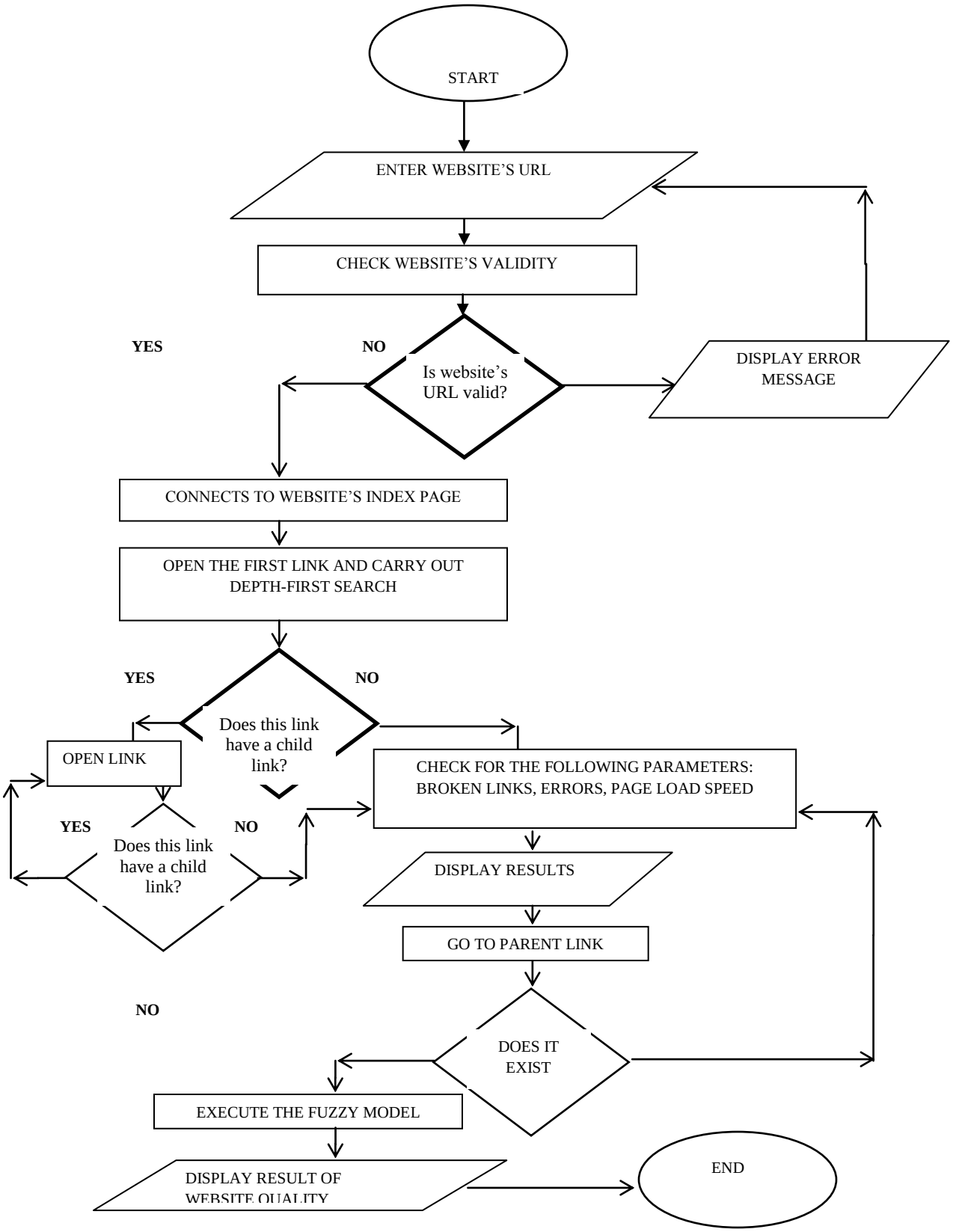


Figure 3.7: Flowchart for the FWQAS

CHAPTER FOUR

IMPLEMENTATION OF FUZZY BASED WEBSITE QUALITY ASSURANCE SYSTEM

4.1Introduction

The fuzzy-based website quality assurance system makes it possible to check and know the quality of your website per time by checking the status of the following parameters on the website: number of broken links, page load speed, error, file support and giving you the overall quality rating of the website using fuzzy logic.

4.2 System Testing and Results

Below are the screenshots of the developed system illustrating how the fuzzy-based website quality assurance system works:

a. Mainpage Overview

The figure 4.1 below shows the welcome page of the website quality assurance system with a major menu “Home” on the menu bar.

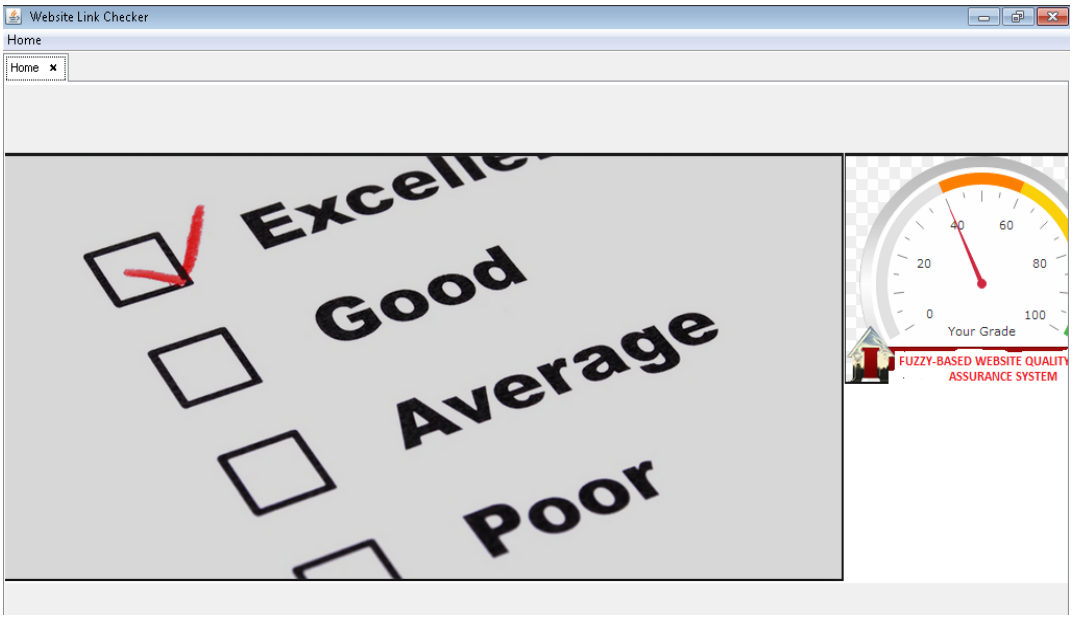


Figure 4.1: Mainpage of the FWQ

b. Menu Bar

In the Figure 4.2 below, the only menu on the menu bar is “Home” which has two submenus “check a link” and “exit”.

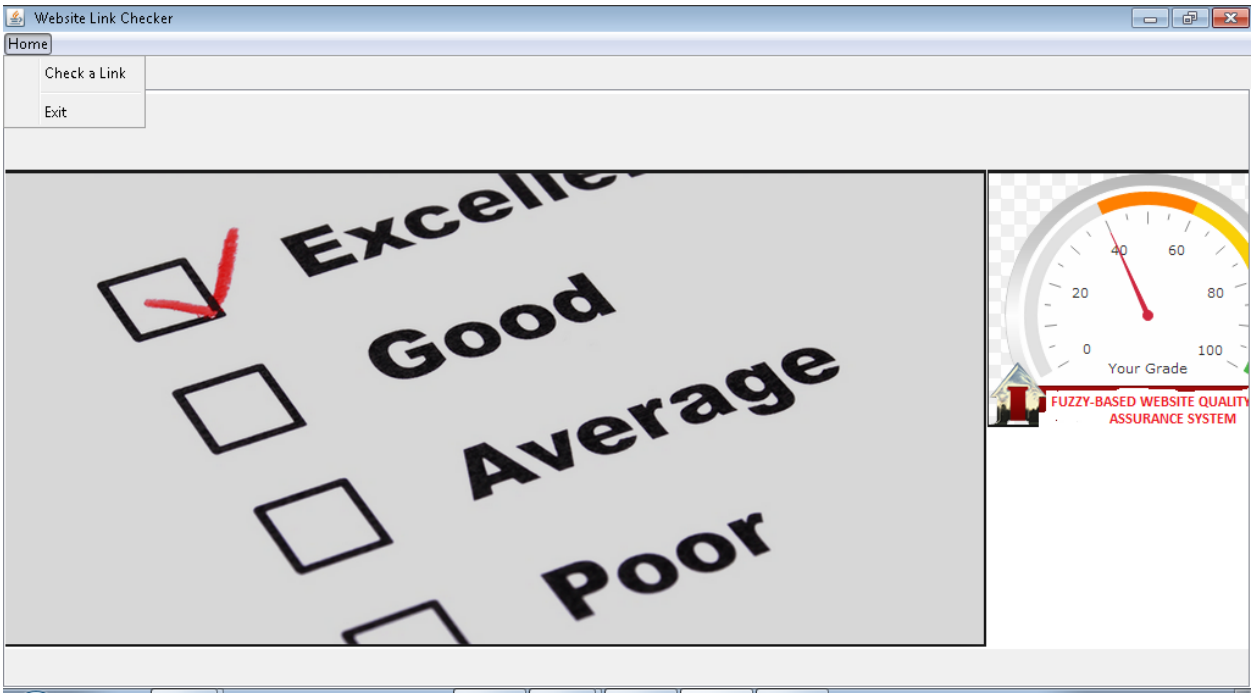


Figure 4.2: Menu bar

When the “check a link” submenu is clicked, it takes the user to another page that will allow for a website address to be entered and analysed while the “exit” submenu terminates the program when clicked on.

c. Check Link Page

Below is Figure 4.3, where the result of each link checked is being displayed.

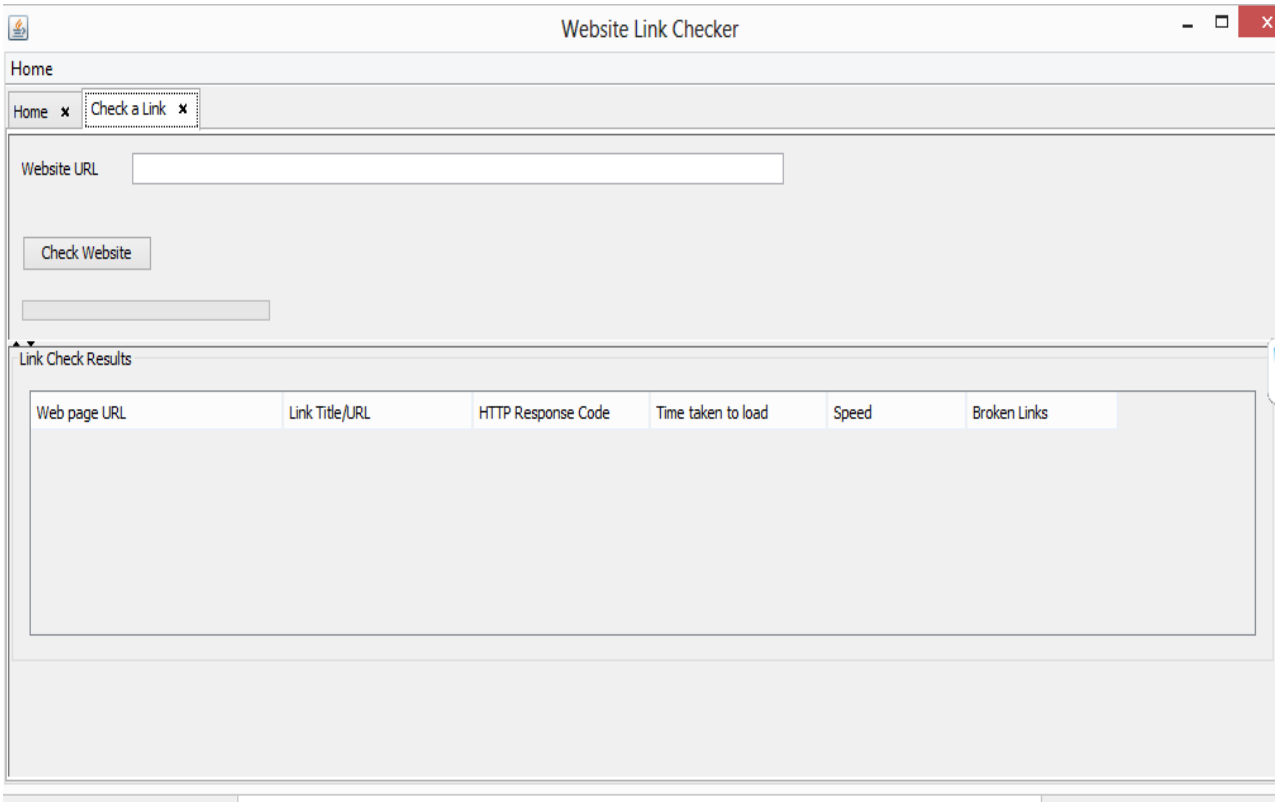


Figure 4.3: Check Link page

The "Check Link page**"** consists of a textbox for entering the website's URL by the user, a "check website" command button; that when clicked on, if the URL is valid, the system initiates a connect to the index page and carries out a depth-first search for the first link reached to link to the last child link. Also, on this page, is a table for displaying the results of the links checked. The following parameters are checked Web page URL, Link title/URL, HTTP response code, Time taken to load, speed and Broken links for each link accessed. The Web page URL column displays the website address / URL of each webpage within the website that is accessed. The HTTP response code column displays the response code currently on the webpage accessed .The Link title/URL column displays the title / name given to the webpage accessed. Java measures time in milliseconds, so the "Time taken to load" column displays in milliseconds the time taken for the webpage to load, please note, a very stable internet connection is needed to avoid

unnecessary delay in the time taken for webpages upload. The Speed column displays "Within limit" if the time taken for webpage upload has not exceeded 2 seconds. The Broken links column displays "Active link", if the link being accessed can be opened and "Broken link" if the link cannot be opened. Every link currently checked is displayed on the status bar located at the bottom, right-hand side of the "Check Link Page" and the results displayed in a tabular form on the same page (see figure 4.4 for visibility). When all the links in the website have been checked, the system takes as its inputs the results from all the individual links checked and applies fuzzy logic to compute the overall rating of the website. An example is shown in Figure 4.5.

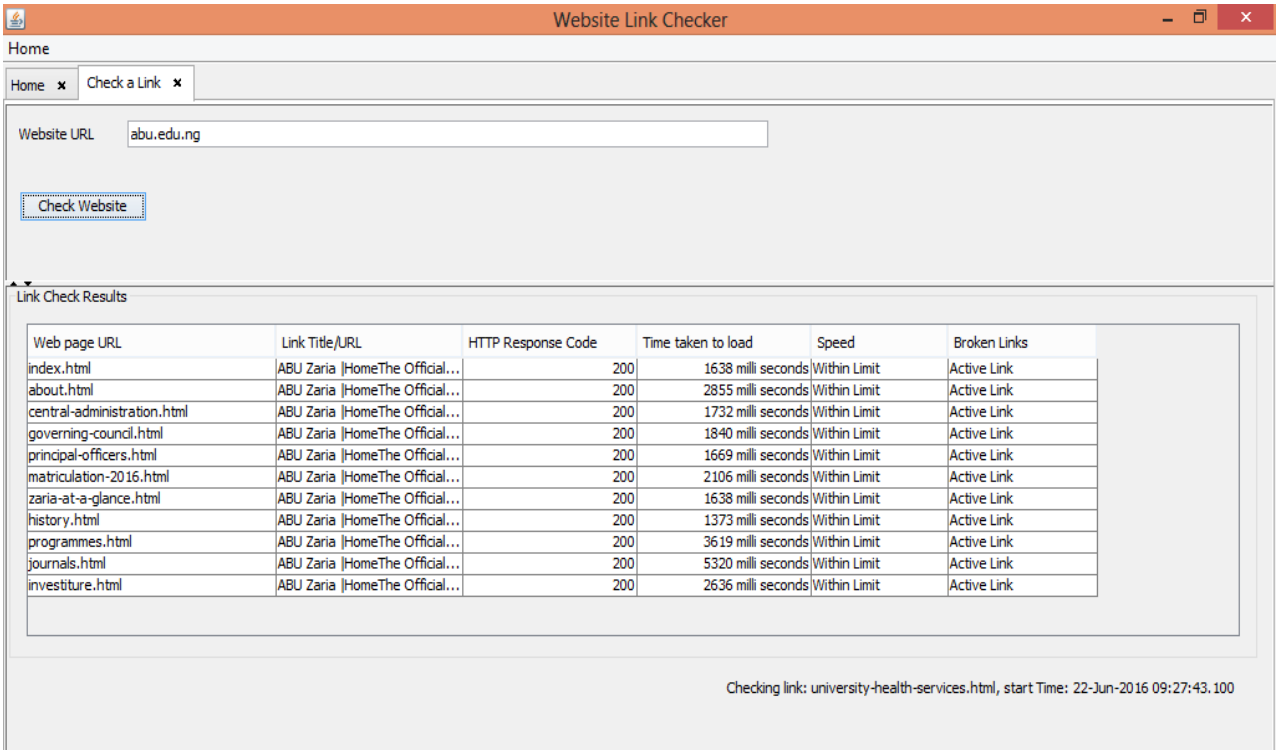


Figure 4.4: Results of individual links checked

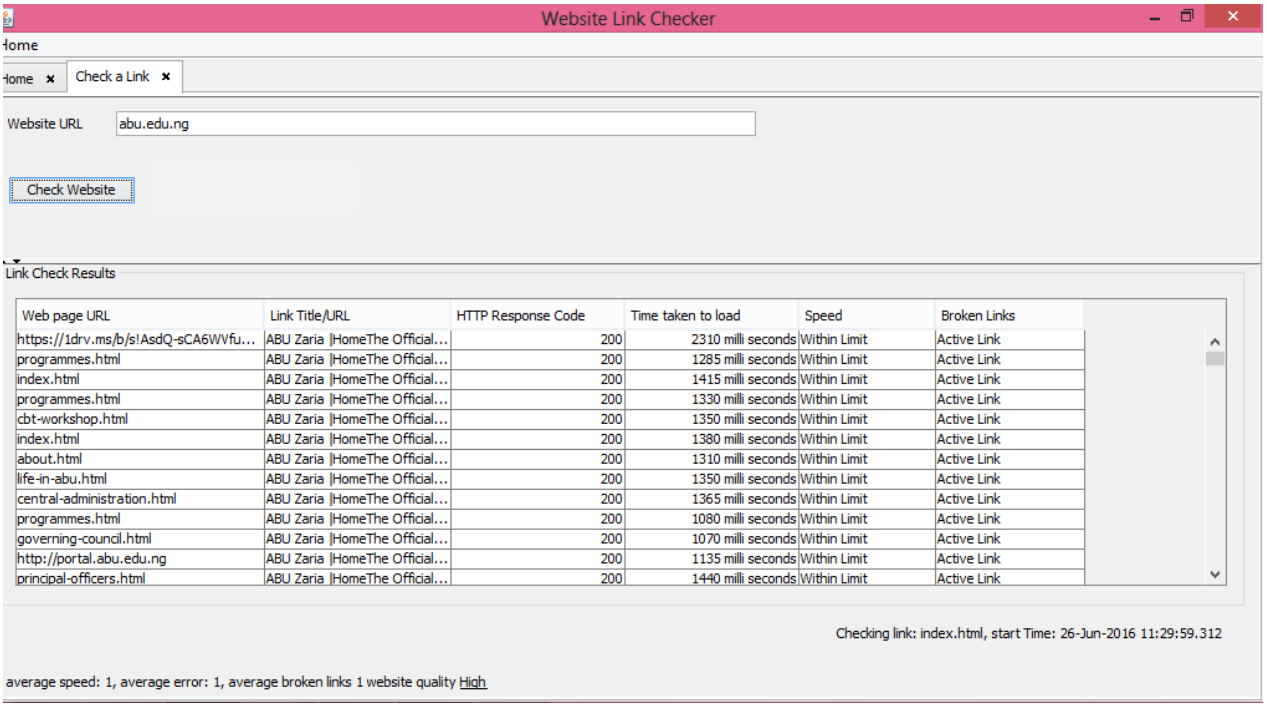


Figure 4.5: Overall result of links checked displayed

4.3 Comparison with Related Work

Comparing the FWQAS with some related works discussed in Chapter two, section 2.9; from the results shown from figure 4.1 to figure 4.5, it can be seen that FWQAS is more self-sufficient and handles more parameters that contribute to having a good quality website. With FWQAS, a web administrator does not need to have different evaluation tools handling different parameters. In comparison with Fuzz-web, FWQAS does not depend on any external evaluation tool, thereby making it not to be at the mercy of any tool before it can perform its task efficiently.

Below in figure 4.6 is the screenshot of the result displayed by Xenu evaluation tool

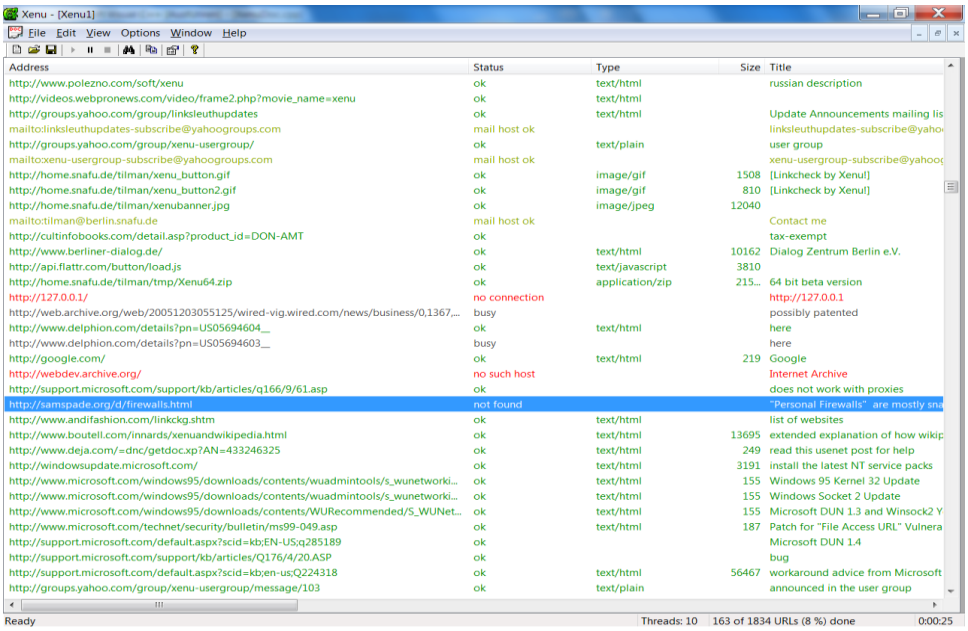


Figure 4.6: Screenshot of Xenu's result (Steven, 2002)



Figure 4.7: Webpage Speed Analyser's result –GTMetrix (www.gtmetrixpro.com)

	Broken Links Check`	Speed Check	Link Errors Check	Independent of External Evaluation tools
Xenu	✓	X	X	✓
Webspeed analyser	X	✓	X	✓
Fuzz-Web	✓	✓	✓	X
FWQAS	✓	✓	✓	✓

Table 4.1:Comparison table

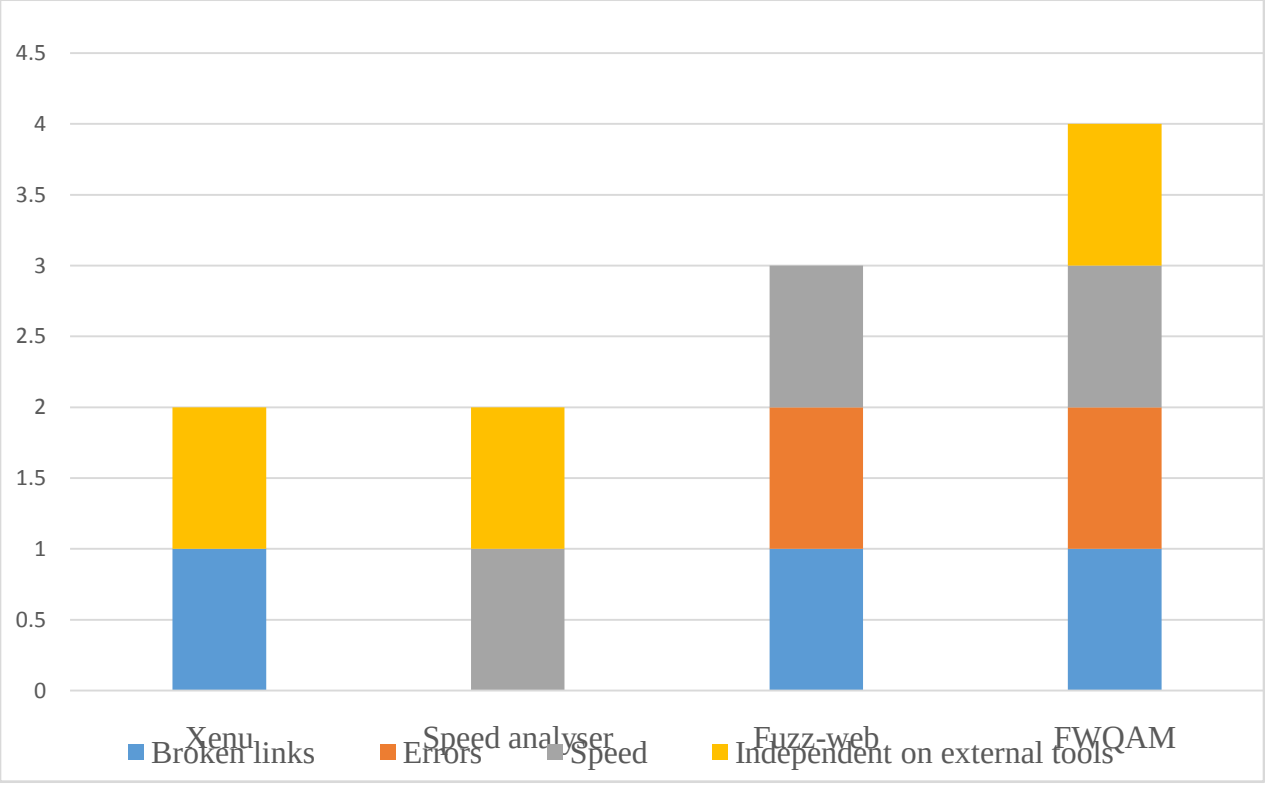


Figure 4.8: Comparison chart

Depicted in the table 4.1and figure 4.8 are the strengths and weaknesses or limitations of the tools discussed in this research. Xenu as an evaluation tool is majorly used to identify broken links on websites and to perform its task, it does not depend on another external tool but it was not developed to measure the speed of webpages download. Likewise webpage speed analysers, there major task is measuring the speed of webpages download without identifying broken links on website. The limitations of these tools (Xenu and webpage speed analyser) make them individually insufficient to be used to carry out maintenance activity on websites. Although, Fuzz-web can carry out broader assessment on websites but it is dependent on these tools (Xenu, Webpage speed analyser) for it to be able to carry out its task, making it liable to fail if any of these tools it depends on fails or becomes unavailable. FWQAS has been able to combine the strengths of these three tools (Xenu, webpage speed analyser and Fuzz-web) into a single system.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

The chapter summarizes the results of the research and analysis of fuzzy-based website quality assurance system. This chapter also gives recommendations and draw conclusion on the proposed system.

5.2 Summary

The major contributions of the fuzzy-based website quality assurance systemis to provide an easy-to-use system that can assist website owners to know the state of their websites per time and to be proactive in making necessary amendments to their website to guarantee a good and wonderful experience for visitors checking their websites, thereby harnessing the benefits of having an online presence in this time and age. Also, FWQAS has been able to show that an assessment system can be self-sufficient not having need to depend on external evaluation tools before performing its task.

5.3 Conclusions

The research has answered the questions it set out in the problems statements section in chapter one and in line with the objectives of the research, the following have being achieved;

a. An easy-to-use website quality assurance system that successfully carries out the following:
    I. Reports broken links
    II. Reports webpages load time
    III. Reports HTTP response codes to ascertain if there are errors on the website

IV. Rates the quality of the website.

5.4Future Works

The following areas can be considered for future work as far as this research is concerned:

a. The system design and implementation can be improved in order to support more parameters used in assessing websites
b. The system can be improved to give deeper analysis of the assessment results and also suggest solutions to the problems identified
c. The system can be improved on to allow users enter more than one website URL at once thereby allowing for the assessment of more than one website at a time.

References

Ahmad, S., Rodziah, B. and Nasir, B. (2011). *Data Quality Enhancement Technology to Improve Decision Support.* Croatia: Intech. pp 34-41

Akamai, A. (2006). *Retail Web Site Performance: Consumer Reaction to a Poor Online Shopping* . Boston: Jupiter Kagan Inc. pp 16-18

Berners-Lee, T., Cailliau, R., and Groff, J. (1992). The world wide web. *Joint European Networking Conference* 10(1), 454-459.

Blanchard, M. (2013). *History of the Mass Media in the United States.* Retrieved on April 4, 2014 from https://books.google.com.ng/books?isbn=1135917426

Bouch, A., Kuchinsky, A., and Bhatti, N. (2000). Quality is in the Eye of the Beholder: Meeting Users' requirement for Internet Quality of service. *An International Journal*, 2(3), 297-304.

Brahima, S. (2014). *The World in 2014: ICT Facts and Figures.* Switzerland: International Telecommunications Union. pp 7

Buchholtz, C. (1995). *Eniac's WWW Server.* Retrieved on June 11, 2014 from http://www.upenn.seas.eniac

Buttazzo, G., Lipari, G., Caccamo, M. and Abeni, L. . (2002). Elastic scheduling for flexible workload management. *Institute of Electrical and Electronics Engineers Transactions.* 51(3), 289-302.

Diffily, S. (2006). *The Website Manager's Handbook.* Chicago: Adventure works. pp 3

Encarta, M.(2009). Microsoft Corporation. Retrived on February 7, 2014, from http://search.msn.com/encarta/results.aspx

Farber, D. (2006). *Google's Marissa Mayer: Speed Win.* Retrieved on June 11, 2014 from http://blogs.zdnet.com/BTL

Fisher, T. (2014). *500 Internal server error: How to fix a 500 internal server error*. Retrieved on March11, 2015 from http://pcsupport.about.com/od/findbyerrormessage/a/500servererror

Fogg, B. (2001). What Makes Web Sites Credible? A Report on a Large Quantitative Study. *Journal of computing*, 9(2), 61-68.

Galletta, D., Henry, R., McCoy, S., and Polak, P. (2004). Web Site Delays: How Tolerant are users? . *Journal of the Association for Information Systems*, 5(3), 110-120.

Ghandour, A., Benwell, G. L., and Deans, K. R. 2011. (2011). Measuring the Performance of eCommerce website. *Pacific Asia Journal of the Association for Information Systems (PAJAIS)* , 10(1),263-274.

Goel, R. (2014). *Top 10 characteristics and qualities of a Good Website*. Retrieved on February 10, 2016 from http://www.webinfozone.com/articles

Gromov, R., (1998). *History of Internet and WWW: The roads and crossroads of Internet history*. Retrieved on June 30, 2016 from http:www.webfoundation.org

Harper, C. (1999). *Weaving the Web*. Retrieved on June 30, 2016 from http://www.webfoundation.org/about/vision/history-of-the-web

Hirmes, D. (1993). *Frequently Asked Questions (FAQ) File for alt.hypertext.* Retrieved on May 18, 2014 from http://www.kevcom.com/words/ht93/hypertext.faq.txt

Hughes, K. (1993, february 19). *Entering the World-wide web: A guide to cyberspace.* Retrieved on May17, 2014 from http://epics.aps.anl.gov/demo/guide-www.guide.html

Ibam, E. (2011). A fuzzy system model of an online career counseling system . *International Journal for Teaching and Education*,6(3), 101-108.

Jolliffe, A., Ritter, J., Stevens, D. (2001). *The Online Learning Handbook: Developing and Using Web-based Learning.* London: Kogan Page. pp5-6 .

Kende, M. (2014). *Global Internet Report 2014.* Switzerland: Internet Society. pp 2

Kohavi, R., and Longbotha, R. (2007). "Online Experiments: Lessons Learn. *International Journal*, 3(1),103-105.

Linden, G. (2006). *Marissa Mayer at Web 2.0.* Retrieved on August12, 2013 from http://glinden.blogspot.com/2006/11/marissa

Mamdani, E. and Assilian, S. . (1965). An experiment in linguistic synthesis with a fuzzy logic controller 7(1). *International Journal of Man-Machine Studies*, 73(3),1-13.

Maurer, H. (1996). *Hyperwave- The Next Generation Web Solution.* Boston: Addison Wesley Longman. pp55.

Nah, F. (2004). A study on tolerable waiting time: how long are Web users willing to wait? *Journal of Computing*, 21(1),123-134.

Nelson, T. (1994). *The Story so far.* Retrieved on March 8, 2015 from http://www.aus.xanadu.com/itimes

Nielsen, W. (2009). *Webvisible.* Retrieved on February 5, 2014 from http://www.marketwired.com

Pedrycz, W., and Gomide, F. (2007). *Fuzzy Systems Engineering: Toward Human-centric computing.* Wiley: Institute of Electrical and Electronics Engineers. pp 34-35.

Pesce, M. (1993). *VRML Browsing and Building Cyberspace.* Retrieved on April 24, 2014 from http://www.newriders.com/vrml

Pratt, P. and Last, M. (2014). *Concepts of Database Management 7th Edition.* Michigan: Joe Sabatino. pp 51.

Relihan, L. , Cahill, T., and Hinchey, G. (1995). *Untangling the World-Wide Web.* Retrieved on March 5, 2014 from http://dl.acm.org/citation.cfm?id=192531

Rekik, R., and Kallel, I. . (2013). Fuzz-Web: A Methodology Based on Fuzzy Logic for assessing websites . *International Journal of Computer Information Systems and Industrial Management Applications*, 6(4),126-136.

Sheo D., Payal G. and Kawaljeet K. (2012). A Fuzzy Approach Scheduling on More Than One Processor System in Real Time Environment. *International Journal of Scientific Research Engineering & Technology (IJSRET)*, 10(1),289-293

Tipton, H. (2014). *Information Security Management Handbook, 4th Edition.* USA: Adventure Works Press. pp 12.

Tomei, L. (2010). *Lexicon of Online and Distance Learning.* New York: Rowman and Littlefield. pp 23-24.

Vistaprint. (2012, March 8). *The Guide: Benefits of a Company Website*. Retrieved on February 15, 2014 from http://www.vistaprint.com

Vydano, D. (2012). *Basic principles of fuzzy logic.* Prague: Albatros media. pp 45-48.

Weinreich, H., Obendorf, H., Herder, E., and Mayer, M. (2008). Not quite the average: An empirical study of the web use. *ACM Trans.* 2(1), 31-35.

Wiley, J. (2010). Fuzzy Sets and Fuzzy Logic: Theory and Applications. *International Journal of Computer Applications*, 2(3),1-4.

William, G. (2011). *An Optimization Approach to Employee Scheduling Using Fuzzy Logic.* California : Ardvent works. Pp 14-30.

Zadeh, A. (1965). Fuzzy Sets. *Journal of Information and Control*, 57(6),338-353.

Zeltser, L. (2015). *Early History of the World-Wide Web: Origins and Beyond.* Retrieved onAugust 2, 2015 from https://zeltser.com/web-history

APPENDIX

Appendix I: Java source code for FWQAS

```
package website_link_checker;

import com.itextpdf.text.*;

import com.itextpdf.text.pdf.*;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.text.DecimalFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.swing.table.DefaultTableModel;

import net.sourceforge.jFuzzyLogic.FIS;

import net.sourceforge.jFuzzyLogic.FunctionBlock;

import net.sourceforge.jFuzzyLogic.rule.Variable;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.xslf.usermodel.XMLSlideShow;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import org.apache.poi.xwpf.usermodel.XWPFDocument;

import org.jsoup.Connection.Response;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Element;

import org.jsoup.select.Elements;
```

```
import javax.net.ssl.*;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
/**
 * @author o-aribisala-aina
 */
public class CheckLink extends javax.swing.JPanel {
    double avgSpeed, avgError, avgBrokenLinks;
    public CheckLink() {
        initComponents();
        extraInit();
    }
    public void extraInit(){
        ((DefaultTableModel) jTableCheckResults.getModel()).setRowCount(0);
        jButtonDisplayGraphs.setVisible(false);
    }
    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
```

```
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {
    jSplitPane1 = new javax.swing.JSplitPane();
    jPanel2 = new javax.swing.JPanel();
    jButtonCheckLink = new javax.swing.JButton();
    jProgressBar1 = new javax.swing.JProgressBar();
    jLabel8 = new javax.swing.JLabel();
    jTextFieldURL = new javax.swing.JTextField();
    jButtonDisplayGraphs = new javax.swing.JButton();
    jPanel3 = new javax.swing.JPanel();
    jLabelSiteStats = new javax.swing.JLabel();
    jPanelResult = new javax.swing.JPanel();
    jScrollPane2 = new javax.swing.JScrollPane();
    jTableCheckResults = new javax.swing.JTable();
    jLabelStatus = new javax.swing.JLabel();
    jSplitPane1.setDividerLocation(130);
    jSplitPane1.setOrientation(javax.swing.JSplitPane.VERTICAL_SPLIT);
    jSplitPane1.setOneTouchExpandable(true);
    jButtonCheckLink.setText("Check Website");
    jButtonCheckLink.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonCheckLinkActionPerformed(evt);
        }
    });
    jLabel8.setText("Website URL");
```

```
        jButtonDisplayGraphs.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonDisplayGraphsActionPerformed(evt);
            }
        });
        jLabelSiteStats.setText("<html></html>");
        jLabelSiteStats.setHorizontalTextPosition(javax.swing.SwingConstants.LEFT);
        jPanelResult.setBorder(javax.swing.BorderFactory.createTitledBorder("Link Check Results"));
        jTableCheckResults.setAutoCreateRowSorter(true);
        jTableCheckResults.setModel(new javax.swing.table.DefaultTableModel(
            },
            new String [] {
                "Web page URL", "Link Title/URL", "HTTP Response Code", "Time taken to load", "Speed",
"Broken Links"
            }
        ) {
            Class[] types = new Class [] {
                java.lang.String.class, java.lang.String.class, java.lang.Integer.class, java.lang.Integer.class,
java.lang.String.class, java.lang.String.class
            };
             jTableCheckResults.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_OFF);
        jScrollPane2.setViewportView(jTableCheckResults);
        if (jTableCheckResults.getColumnModel().getColumnCount() > 0) {
            jTableCheckResults.getColumnModel().getColumn(0).setMinWidth(200);
            jTableCheckResults.getColumnModel().getColumn(1).setMinWidth(150);
            jTableCheckResults.getColumnModel().getColumn(2).setMinWidth(140);
```

```
        jTableCheckResults.getColumnModel().getColumn(3).setMinWidth(140);
        jTableCheckResults.getColumnModel().getColumn(4).setMinWidth(110);
        jTableCheckResults.getColumnModel().getColumn(5).setMinWidth(120);
    }
    private void jButtonCheckLinkActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jProgressBar1.setVisible(true);
    jProgressBar1.setIndeterminate(true);
    ((DefaultTableModel) jTableCheckResults.getModel()).setRowCount(0);
        Thread tt = new Thread(){
      @Override
      public void run(){
        checkLink();
      }
    };
        tt.start();
        jProgressBar1.setVisible(false);
    jProgressBar1.setIndeterminate(false);
  }
  private void jButtonDisplayGraphsActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
String fileName =
"C:\\Users\\TAYE\\Desktop\\Website_Link_Checker\\src\\website_link_checker\\websiteChecker.fcl";
 FIS fis =
FIS.load("C:\\Users\\TAYE\\Desktop\\Website_Link_Checker\\src\\website_link_checker\\websiteChecke
r.fcl", true);
    FunctionBlock fb = fis.getFunctionBlock(null);
```

```
// Set inputs
fb.setVariable("Broken_Links", avgBrokenLinks);
fb.setVariable("Errors", avgError);
fb.setVariable("Speed_Load", avgSpeed);
// Evaluate
fb.evaluate();
// Get output
//jLabelWQuality.setText(fb.getVariable("Website_Quality").getValue() + "");
// Error while loading?
      if( fis == null ) {
         System.err.println("Can't load file: '" + fileName + "'");
         return;
      }
      // Show
      FunctionBlock functionBlock = fis.getFunctionBlock(null);
      System.out.println("FIS is " + fis);
      System.out.println("Function block is " + functionBlock);
      JFuzzyChart.get().chart(functionBlock);
      // Evaluate
      fis.evaluate();
      // Print ruleSet
      System.out.println(fis);
   }
   private void checkLink() {
      String home = "";
```

```
if (! jTextFieldURL.getText().startsWith("http://") && ! jTextFieldURL.getText().startsWith("https://")) {
   home = "http://" + jTextFieldURL.getText();
} else {
   home = jTextFieldURL.getText();
}
System.out.println("Home: " + home);

//home = jTextFieldURL.getText().startsWith("http://") ? jTextFieldURL.getText() : "http://" +
jTextFieldURL.getText();
//home = jTextFieldURL.getText().startsWith("https://") ? jTextFieldURL.getText() : "https://" +
jTextFieldURL.getText();
      Response  = null;
long startTime;
      org.jsoup.nodes.Document doc = null;
try {
   enableSSLSocket();
   //response = Jsoup.connect(home).data("query", "Java").userAgent("Mozilla").cookie("auth",
"token").timeout(3000).execute();
   doc = Jsoup.connect(home).data("query", "Java").userAgent("Mozilla").cookie("auth",
"token").timeout(3000).post();
} catch (java.net.UnknownHostException _ex) {
   System.out.println(_ex.toString());
   javax.swing.JOptionPane.showMessageDialog(this, "The website you have entered does not exist,\n
website quality cannot be determined for a non-existent site",
         "Website does not exist", javax.swing.JOptionPane.ERROR_MESSAGE);
} catch (Exception _ex) {
   System.out.println(_ex);
```

```
        }
        Elements links = doc.select("a[href]");
        String thisLink = home;
        Element headElement = (doc.select("head")).first();
        String title = headElement.select("title").first().text();
        title = "";//headElement.select("title").first().text();

        int nonErrorResponse = 0, numLinks = links.size();
        int totalDocLinks = 0, totalOpeningDocs = 0;
        double totalLinkSpeed = 0d;

        int numActiveLinks = 0;
        for (Element link : links) {
           numActiveLinks++;
           //thisLink = link.attr("href").startsWith(home) ? link.attr("href") : home + link.attr("href");
           thisLink = link.attr("href");
           jLabelStatus.setText("Checking link: " + thisLink + ", start Time: " + new SimpleDateFormat("dd-
MMM-yyyy hh:mm:ss.SSS").format(new Date()));
           if (thisLink.endsWith(".pdf")) {
              totalDocLinks++;
              if (checkPDFDoc(thisLink)) {
                 totalOpeningDocs++;
              }
           } else if (thisLink.endsWith(".doc") || thisLink.endsWith(".xls") || thisLink.endsWith(".ppt") ||
thisLink.endsWith(".docx") || thisLink.endsWith(".xlsx") || thisLink.endsWith(".pptx")) {
              totalDocLinks++;
```

```
            if (checkMSDocument(thisLink)) {
                totalOpeningDocs++;
            }
        }

        startTime = new Date().getTime();

        boolean isBroken = false;
        try {
            response = Jsoup.connect(home).data("query", "Java").userAgent("Mozilla").cookie("auth",
"token").timeout(3000).execute();
            doc = response.parse();
        } catch (Exception _ex) {
            System.out.println(_ex);
            isBroken = true;
            numActiveLinks--;
        }
        //Web page URL, Link Title/URL, HTTP Response Code, Time taken to load
        title = (doc.select("head")).first().select("title").first().text();
        long timeTaken = (new Date().getTime()) - startTime;
        if ((response.statusCode() == 200 || response.statusCode() == 404)) nonErrorResponse++;
        ((DefaultTableModel) jTableCheckResults.getModel()).addRow(
              new Object[] {thisLink, title, response.statusCode(), timeTaken + " milli seconds",
                 timeTaken < 2000 ? "Within Limit" : "Slow loading ",
                 (response.statusCode() != 200 ? "Broken Link" : "Active Link")
        });
```

```
                totalLinkSpeed += (timeTaken < 2000 ? 1 : 0);
        }
            ((DefaultTableModel) jTableCheckResults.getModel()).addRow(
            new Object[] {}
        );
            ((DefaultTableModel) jTableCheckResults.getModel()).addRow(
            new Object[] {thisLink, title + "/" + home, numLinks, numActiveLinks}
        );
            jLabelSiteStats.setText(title);
        String siteStats = calculateSiteStat(totalLinkSpeed, nonErrorResponse, numActiveLinks,
totalOpeningDocs, totalDocLinks, links.size());
            jLabelSiteStats.setText("<html> " + siteStats + "</html>");
    }
        private void enableSSLSocket() throws KeyManagementException, NoSuchAlgorithmException {
        HttpsURLConnection.setDefaultHostnameVerifier(new HostnameVerifier() {
            public boolean verify(String hostname, SSLSession session) {
                return true;
            }
        });
        SSLContext context = SSLContext.getInstance("TLS");
        context.init(null, new X509TrustManager[]{new X509TrustManager() {
            public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
            }
            public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
            }
```

```
        public X509Certificate[] getAcceptedIssuers() {
            return new X509Certificate[0];
        }
    }}, new SecureRandom());
    HttpsURLConnection.setDefaultSSLSocketFactory(context.getSocketFactory());
  }


  private String calculateSiteStat(double totalSpeed, int nonErrorResponse, int numActiveLinks, int
totalOpeningDocs, int totalDocLinks, int totalLinks) {
    DecimalFormat myFormatter2 = new DecimalFormat("#.##");


    //avgSpeed, avgError, avgBrokenLinks
    double avgSpeed = totalSpeed / (double) totalLinks;
    double avgError = nonErrorResponse / totalLinks;
    double avgBrokenLinks = ((double) (numActiveLinks) / (double) totalLinks);
     String stats = myFormatter2.format(statsD);
    String quality = "";
    }
    return (" average speed: " + myFormatter2.format(avgSpeed) + ", average error: " +
myFormatter2.format(avgError)
        + ", average broken links " + myFormatter2.format(avgBrokenLinks)
        + " website quality " + "<u>" + quality + "</u>");
  }
    if (link.endsWith("xlsx")) {
      FileInputStream file = null;
      try {
```

```
        file = new FileInputStream(new File(link));
    } catch (Exception _ex) {}

    try {
        XSSFWorkbook workbook = new XSSFWorkbook(file);
    } catch (Exception _ex) {
        documentOpened = false;
    }

}

if (link.endsWith("ppt")) {
    FileInputStream file = null;
    try {
        file = new FileInputStream(new File(link));
    } catch (Exception _ex) {}

    try {
        XMLSlideShow ppt = new XMLSlideShow(file);
    } catch (Exception _ex) {
        documentOpened = false;
    }

}
```

```
        return documentOpened;
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButtonCheckLink;
    private javax.swing.JButton jButtonDisplayGraphs;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabelSiteStats;
    private javax.swing.JLabel jLabelStatus;
    private javax.swing.JLabel jLabelWQuality;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanelResult;
    private javax.swing.JProgressBar jProgressBar1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JSplitPane jSplitPane1;
    private javax.swing.JTable jTableCheckResults;
    private javax.swing.JTextField jTextFieldURL;
    // End of variables declaration
}
package website_link_checker;
//import javax.swing.JFrame;
import java.sql.*;
import javax.swing.JPanel;
//import javax.swing.UnsupportedLookAndFeelException;
/*
```

```
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author o-aribisala-aina
 */
public class DashBoard extends javax.swing.JFrame {

    /** Creates new form Dashboard */
    public DashBoard() {
        initComponents();
        setTitle("Website Link Checker");

        BackgroundPanel home = new BackgroundPanel();
        home.setName("Home");
        initTab(home, "Home");

    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
```

```
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jTabbedPane1 = new javax.swing.JTabbedPane();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenuFile = new javax.swing.JMenu();
    jMenuItemCheckLink = new javax.swing.JMenuItem();
    jSeparator8 = new javax.swing.JPopupMenu.Separator();
    jMenuItemExit = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("REPORTS");
    getContentPane().setLayout(new java.awt.GridLayout(0, 1));
    getContentPane().add(jTabbedPane1);

    jMenuFile.setText("Home");

    jMenuItemCheckLink.setText("Check a Link");
    jMenuItemCheckLink.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItemCheckLinkActionPerformed(evt);
        }
    });
```

```
    jMenuFile.add(jMenuItemCheckLink);
    jMenuFile.add(jSeparator8);

    jMenuItemExit.setText("Exit");
    jMenuItemExit.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItemExitActionPerformed(evt);
        }
    });
    jMenuFile.add(jMenuItemExit);

    jMenuBar1.add(jMenuFile);

    setJMenuBar(jMenuBar1);

    pack();
}// </editor-fold>

private void jMenuItemCheckLinkActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    CheckLink checkLink = new CheckLink();
    initTab(checkLink,"Check a Link");
}

private void jMenuItemExitActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        // TODO add your handling code here:
        System.exit(0);
    }

    private void initTabComponent(String title,int i) {
        jTabbedPane1.setTabComponentAt(i, new ButtonTabComponent(title, jTabbedPane1));
    }

    private void initTab( JPanel pp,String ss){
        jTabbedPane1.addTab(ss, pp);
        int i = jTabbedPane1.getTabCount() - 1;
        initTabComponent(ss,i);
        jTabbedPane1.setSelectedIndex(i);
    }

    // Variables declaration - do not modify
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenu jMenuFile;
    private javax.swing.JMenuItem jMenuItemCheckLink;
    private javax.swing.JMenuItem jMenuItemExit;
    private javax.swing.JPopupMenu.Separator jSeparator8;
    private javax.swing.JTabbedPane jTabbedPane1;
    // End of variables declaration

}
```