PDF Text Searching System

By

Siti Aezane Bt Ab. Ghani

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information System)

JANUARY 2006

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

PDF Text Searching System

By

Siti Aezane Bt Ab. Ghani

A project dissertation submitted to the
Information System Programme
Universiti Teknologi PETRONAS
In partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION SYSTEM)

Approved by,

_____

(Ms. Amy Foong Oi Mean)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
January 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

SITI AEZANE BT AB. GHANI

ABSTRACT

This project is to develop a text searching system that assist users to develop a simple PDF text-searching system, which is capable of searching and processing the information in text files on user PC and in local networks. The main purpose of developing this project is to assist users in finding PDF documents and files within their local drives, where the appropriate documents can be found by entering the desired search terms (keywords) in the PDF Text Searching System. There are two objectives that have been set for this project. The first objective is to perform a study and have a better understanding on the software that will be used in order to develop PDF text-searching system, and the second objective is to develop a PDF text-searching system, which is capable of searching and processing the information in text files on user PC and in local networks. For the methodology, Rapid Application Development (RAD) approach has been employed. The methodology has been chosen because it is effective and suitable for short duration project. It was designed for developer and user to join together and work intensively toward their goal. By using the RAD methodology, the project is able to be completed within the time allocated. In the results and discussion part, it covers all the outcome that obtains from the project completion, which is based on the surveys conducted and questionnaires. In this chapter, the findings that were gain will determine whether the proposed system is acceptable and meet with the user's needs. In order to provide better services, some suggestion being carried out for future enhancement. This can improve the current system to be more efficient and effective.

ACKNOWLEDGEMENTS

First and foremost, Praise Be upon to Allah S.W.T for His Mercy has given me the guidance and wisdom to come this far in this report presentation.

My gratitude goes to my supervisor, Ms. Amy Foong Oi Mean for the tremendous support and precious assistance in making a success of this Final Year Project: "PDF Text Searching System". Further thanks to all respondents that involved and gave good cooperation in conducting the surveys and questionnaires. I really appreciate the feedback and advice that have been given. Those responds are really valuable for this project.

This appreciation also goes to all lecturers, final year students, and technician for their assistance, ideas, and support throughout completing the project. I also would like to thank to my lovely family for their support for all these times. Thank you for the countless and meaningful advice and *doa.*

Finally, thank you for all individuals that have contributed their ideas, knowledge, support, and assistance in this project. Sincere thanks to the wonderful friends for their cooperation. ^__^

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION AND NOMENCLATURES

ASP	: Active Server Page
CASE	: Computer Aided Software Engineering
CD	: Compact Disk
CPU	: Control Processor Unit
CFML	: ColdFusion Markup Language
DBA	: Database Administration
DFD	: Data flow diagram
DHTML	: Dynamic HyperText Markup Language
GIS	: Geographical Information System
GUI	: User Interface Design
HTML	: HyperText Markup Language
JAD	: Joint Application Design
MB	: Megabyte
OCR	: Optical Character Recognition
PC	: Personal Computer
PDF	: Portable Document Format
PHP	: Plain Hypertext
RAD	: Rapid Application Development
WWW	: World Wide Web
WYSIWYG	: What You See Is What You Get

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Adobe Acrobat Portable Document Format (PDF) is a format specially designed for publishing documents. It is based on postscript, a printer format page description language. [1]

PDF documents are very much like hypertext markup language (HTML) documents, on that they allow the users to insert hyperlinks and various interactive elements. However while HTML documents are meant to change to fit the screen the viewer is using, or to conform to user preference settings, PDF documents guarantee a look and feel with specific pagination, font, spacing and margin settings – just like Word and image files. This format is very portable because it is compatible with a large number of computer platforms. Free readers from a variety of platforms are available on many sites to be downloaded free of charge.

The problem with marking documents available in HTML, word or text formats is that they can be edited and redistributed easily even under a different name. But if the user wants to change a PDF file, they will need to have a full version of acrobat. Most people just have the reader that allows them to read the text, but not alter it. In addition, PDF documents can be encrypted so that only authorized users can access them, or digital signatures can be added for certification. These two features are particularly useful if the users want to email confidential files or send CDs through post.

There are a lot of excellent uses for PDF files. Teachers can make very attractive classroom resources, companies can produce pretty product brochures, and families who are scattered over a large area can build attractive family newsletters. However, the main idea behind PDF files is that they are designed to be printed. If the users are planning to use PDF files to present information online, they may run into various problems.

First of all, although adobe states that more than 500 million copies have been downloaded so far, relatively few surfers have a PDF reader. Readers are easy to find and download, but will visitors to your web site actually take the trouble to do so? They may if they particularly want to see the document, if they have an hour to spare to download the plug in, if they are comfortable downloading and installing new programs.

Webmasters should consider that most surfers are impatient and would rather look for a resource that offers the same information without the hassle.

Secondly, PDF files are formatted to look good in print, but they are not pleasing to the eyes for regular browsing online. If the users try to look at these files when surfing, the text size is fixed, as is the layout. This makes reading PDF files unwieldy because the lines do not automatically fit to the window size. [2]

A recent study investigating the attractiveness of PDF files posted online pointed out that many surfers found the format cumbersome. Comments from users revealed that many avoided PDF resources as much as possible, saying that they were difficult to read, tricky to use and slow to download.

PDF document have a lot going for them and can be useful additions to the website if they are used properly. However they can also make an individual(s) site difficult to use and unattractive to surfers.

1.2 Problem Statement

1.2.1 Problem Identification

"Forcing users to browse PDF documents makes your website's usability about 300% worse relative to HTML pages. This is my rough estimate, based on watching users perform similar tasks on a variety of sites that used either PDF or regular Web pages. Because I have not performed a detailed measurement study of PDF on its own, I can't calculate the precise usability degradation. However, whether the true number is 280% or 320%, one thing is certain: the number is big and reflects significant user suffering in terms of increased task time and more frequent failures." (Jakob Nielsen's Alertbox, June 10, 2001) [3]

"What we've got is a page of a PDF document which is great when printed out, but on the screen it is hard to read. The print is too small. I have so much difficulties in finding the right files and document towards my needs..." [3]

Each of us has been faced with the problem of searching for information more than once. Regardless of the data source we are using (Internet, file system on our hard drive, data base or a global information system of a big company) the problems can be multiple and include the physical volume of the database searched, the information being unstructured, different file types and also the complexity of accurately wording the search query. We have already reached the stage when the amount of data on one single PC is comparable to the amount of text data stored in a proper library. And as to the unstructured data flows, in future they are only going to increase, and at a very rapid tempo. If for an average user this might be just a minor misfortune, for a big company absence of control over information can mean significant problems.

One of the major challenges facing companies at present is the need for quick search of documents in large data volumes. The organization of data access is in direct relation with the technologies and software that are quick and efficient in processing

information. At present there is a great number of technologies performing phrasal search (Google, Hummingbird, Verity and others), but they do not solve the problem of information search in full measure. [4]

PDF was designed to specify printable pages. PDF content is thus optimized for letter-sized sheets of paper, not for display in a browser window. Users often getting lost in PDF because the print-oriented viewer gives them only a small peephole on a big, complicated layout and they can't scroll it in the simple, linear manner they are accustomed to on the Web. Instead, PDF files often use elaborate graphic layouts and split the content into separate units for each sheet of print. Although this is highly appropriate for printed documents, it causes severe usability problems online.

PDF pages lack navigation bars and other apparatus that might help users move within the information space and relate to the rest of the site. Because PDF documents can be very big, the inability to easily navigate them takes a toll on users. PDF documents also typically lack hypertext, again because they are designed with print in mind.

Because PDF is not the standard Web page format, it dumps users into a non-standard user interface. Deviating from the norm hurts usability because, for example, scrolling works differently, as do certain commands, such as the one to make text larger (or smaller). Also, after finishing with a PDF document, users sometimes close the window instead of clicking the Back button, thus losing their navigation history. Although this behavior is not common, it is symptomatic of the problems caused when users are presented with a non-standard Web page that both looks different and follows different rules.

PDF files usually have both text, and graphical representations of the text, with indications of exactly where that text should be displayed. However, there are several cases where this does not work for searching:

- Documents that were scanned directly into PDF may only have the graphic portion: there may be no computer-readable text at all. These documents are not searchable.

- Documents that were scanned and converted from graphic display to digital text using OCR (optical character recognition) may have significant numbers of errors. This is more common if the original document is old or was not perfectly aligned. In this case, many search terms will not be matched although the words were in the original printed or typed text, because they were not correctly interpreted. Some search terms may be falsely matched if the OCR software incorrectly interpreted the original text.

- Documents with multiple columns, which were converted to PDF by some layout programs, will display correctly and contain the correct digital text, but they miss the text flow: the words don't come in the correct sequence. Therefore the search engines will fail to match phrase queries because the phrases were wrapped on the next line of the column in the original, but that relationship was not stored in the PDF.

- Documents generated by some applications will contain partial words due to hyphenation, incorrect coding of ligatures and extended characters (diacriticals and letters beyond the basic 26), and other unusual situations. These mangled words will not match queries, although the words were in the original text.

The usability problems that PDF files cause on websites or intranets are legion: [5]

- Linear exposition. PDF files are typically converted from documents that were intended for print, so the authors wouldn't have followed the guidelines for Web writing. The result? A long text that takes up many screens and is unpleasant and boring to read.

- Jarring user experience. PDF lives in its own environment with different commands and menus. Even simple things like printing or saving documents are difficult because standard browser commands don't work.

- Breaks flow. Users have to wait for the special reader to start before they can see the content. Also, PDF files often take longer time to download because they tend to be stuffed with more fluff than plain Web pages.

- Orphaned location. Because the PDF file is not a Web page, it doesn't show the standard navigation bars. Typically, users cannot even find a simple way to return to their site's homepage.

- Content blob. Most PDF files are immense content chunks with no internal navigation. They also lack a decent search, aside from the extremely primitive ability to jump to a text string's next literal match. If the user's question is answered on page 75, there's close to zero probability that he or she will locate it.

- Text fits the printed page, not a computer screen. PDF layouts are often optimized for a sheet of paper, which rarely matches the size of the user's browser window. Bye-bye smooth scrolling. Hello tiny fonts.

1.2.2 Significant of the Problem

The project is significant in terms of designing and developing software application in distributed environment, storing data within the application of distributed database, maximizing the integrity and consistency of the data, and able in generating graph to ease business users. This project is expected to be finished and deployed within 12 weeks timeframe.

1.3 Objective and Scope of Study

The project is divided into two phases. In the first phase, the concept of text searching and search engine is to be studied in detailed. In second phase, the design and implementation of the PDF text searching, distributed prototype system and proper user interfaces are to be carried out.

The expected objectives of this project are:

1. To study and have a better understanding on the software that will be used in order to develop PDF text-searching system.

2. To develop a PDF text-searching system, which is capable of searching and processing the information in text files on user PC and in local networks.

In order to ensure the system that will be develop meet the requirement and functional as required, several scope of study have to be define. The scope of study for the project is stated as below:

- To study about text searching system and search engine such as its concept and functionality. For the project, the author will study on what are the major challenges facing companies at the present.

- To focus on portable document format (PDF) extension type, where users are raising the issues on. Based on the study, a prototype of an end product will be developed, which will improve the efficiency of PDF text searching system and also will benefit the user of the system.

CHAPTER 2
LITERATURE REVIEW AND/OR THEORY

2.1 About Text Searching and Search Engine

Text retrieval, full text search (also called free search text) refers to a technique for searching a computer-stored document or database; in a full text search, the search engine examines all of the words in every stored document as it tries to match search words supplied by the user. Full-text searching techniques became common in online bibliographic databases in the 1970s. Most Web sites and application programs (such as word processing software) provide full text search capabilities. Some Web search engines, such as AltaVista employ full text search techniques, while others index only a portion of the Web pages examined by its indexing system. [6]

A search that compares every word in a document, as opposed to searching an abstract or a set of keywords associated with the document. Word processors and text editors contain full-text search functions that let you find a word or phrase anywhere in the document.

The most common approach to full text search is to generate a complete index or concordance for all of the searchable documents. For each word (excepting stop words which are too common to be useful) an entry is made which lists the exact position of every occurrence of it within the database of documents. From such a list it is relatively simple to retrieve all the documents that match a query, without having to scan each document. Although for very small document collections, serial scanning can do full-text searching, indexing is the preferred method for almost all full-text searching. (H.W. Wilson) [7]

As anyone who has performed a free text search will readily recognize, free text searching is likely to retrieve many documents that are not relevant to the search question. Such documents are called false positives. The retrieval of irrelevant documents is often caused by the inherent ambiguity of natural language; for example, in the United States, football refers to what is called American football outside the U.S.; throughout the rest of the world, football refers to what Americans call soccer. A search for football may retrieve documents that are about two completely different sports. [6]

Due to the ambiguities of natural language, a full text search typically produces a retrieval list that has low precision: most of the items retrieved are irrelevant. Controlled-vocabulary searching solves this problem by tagging the documents in such a way that the ambiguities are eliminated. However, a controlled vocabulary search may have low recall: it may fail to retrieve some documents that are actually relevant to the search question. Despite the presence of many irrelevant documents in a free text search's retrieval list, a free text search may be able to locate a document that a controlled vocabulary search failed to retrieve.

Full-text searching is the type performed by most Web search engines on the Web pages that have been retrieved and added to their vast reservoirs. All the words on the pages are searched and then indexed, and the user's search request is satisfied via the indexes. Nevertheless, the user's search winds up being a full-text search on the Web, at least for the part of the Web the search engine has scoured. Although it is very thorough, Web searching often results in too many false drops. (Sergey Melnik, Sriram Raghavan, Beverly Yang, Hector Garcia-Molina, 2001) [8]

Search engines were also known as some of the brightest stars in the Internet investing frenzy that occurred in the late 1990s. Several companies entered the market spectacularly, recording record gains during their initial public offerings. Some have taken down their public search engine, and are marketing enterprise-only editions, such as Northern Light. (Sergey Melnik, Sriram Raghavan, Beverly Yang, Hector Garcia-Molina, 2001) [8]

Before the advent of the Web, there were search engines for other protocols or uses, such as the Archie search engine for anonymous FTP sites and the Veronica search engine for the Gopher protocol. More recently search engines are also coming online, which utilize XML or RSS. This allows the search engine to efficiently index data about websites without requiring a complicated crawler. The websites simply provide an xml feed which the search engine indexes. XML feeds are increasingly provided automatically by weblogs or blogs. Examples of this type of search engine are feedster, with niche examples such as LjFind Search providing search services for Livejournal blogs. [1]

A search engine is a program designed to help find information stored on a computer system such as the World Wide Web, or a personal computer. The search engine allows one to ask for content meeting specific criteria (typically those containing a given word or phrase) and retrieving a list of references that match those criteria. Search engines use regularly updated indexes to operate quickly and efficiently.

Around 2001, the Google search engine rose to prominence. Its success was based in part on the concept of link popularity and PageRank. How many other web sites and web pages link to a given page is taken into consideration with PageRank, on the premise that good or desirable pages are linked to more than others. The PageRank of linking pages and the number of links on these pages contribute to the PageRank of the linked page. This makes it possible for Google to order its results by how many web sites link to each found page. Google's minimalist user interface was very popular with users, and has since spawned a number of imitators. (Ronny Lempel, Shlomo Moran, 2003)

Google and most other web engines utilize not only PageRank but more than 150 criteria to determine relevancy. The algorithm "remembers" where it has been and indexes the number of cross-links and relates these into groupings. PageRank is based on citation analysis that was developed in the 1950s by Eugene Garfield at the University of Pennsylvania. Google's founders cite Garfield's work in their original

paper. In this way virtual communities of webpages are found. Teoma's search technology uses a communities approach in its ranking algorithm. NEC Research Institute has worked on similar technology. Dr. Jon Kleinberg and his team while working on the CLEVER project at IBM's Almaden research lab first developed web link analysis. Google is currently the most popular search engine. [9]

The web is growing much faster than any present-technology search engine can possibly index. Some users found major search-engines became slower to index new web pages. Many web pages are updated frequently, which forces the search engine to revisit them periodically. The queries one can make are currently limited to searching for key words, which may result in many false positives, especially using the default page-wide search. Better results might be achieved by using a proximity-search option with a search-bracket to limit matches within a paragraph or phrase, rather than matching random words scattered across large pages.

Dynamically generated sites may be slow or difficult to index, or may result in excessive results, perhaps generating 500 times more webpages than average. Example: for a dynamic webpage, which changes content based on entries inserted from a database, a search-engine might be requested to index 50,000 static webpages for 50,000 different parameter values passed to that dynamic webpage. Many dynamically generated websites are not indexable by search engines; this phenomenon is known as the invisible web. (Andreas Paepcke, Hector Garcia-Molina, and Gerard Rodriquez, 2000) [10]

In 2006, hundreds of generated websites used tricks to manipulate a search-engine to display them in the higher results for numerous keywords. This can lead to some search results being polluted with linkspam or bait-and-switch pages, which contain little or no information about the matching phrases. The more relevant webpages are pushed further down in the results list, perhaps by 500 entries or more. (Michael J. Cafarella, Oren Etzioni, January 2006)

Web search engines work by storing information about a large number of web pages, which they retrieve from the WWW itself. These pages are retrieved by a web crawler (sometimes also known as a spider) — an automated web browser which follows every link it sees, exclusions can be made by the use of robots.txt. The contents of each page are then analyzed to determine how it should be indexed (for example, words are extracted from the titles, headings, or special fields called meta tags). Data about web pages is stored in an index database for use in later queries. Some search engines, such as Google, store all or part of the source page (referred to as a cache) as well as information about the web pages, whereas some store every word of every page it finds, such as AltaVista. This cached page always holds the actual search text since it is the one that was actually indexed, so it can be very useful when the content of the current page has been updated and the search terms are no longer in it. This problem might be considered to be a mild form of linkrot, and Google's handling of it increases usability by satisfying user expectations that the search terms will be on the returned web page. This satisfies the principle of least astonishment since the user normally expects the search terms to be on the returned pages. Increased search relevance makes these cached pages very useful, even beyond the fact that they may contain data that may no longer be available elsewhere. [11]

When a user comes to the search engine and makes a query, typically by giving key words, the engine looks up the index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text. Most search engines support the use of the boolean terms AND, OR and NOT to further specify the search query. An advanced feature is proximity search, which allows you to define the distance between keywords.

The usefulness of a search engine depends on the relevance of the result set it gives back. While there may be millions of Web pages that include a particular word or phrase, some pages may be more relevant, popular, or authoritative than others. Most search engines employ methods to rank the results to provide the "best" results first. How a search engine decides which pages are the best matches, and what order the

results should be shown in, varies widely from one engine to another. The methods also change over time as Internet usage changes and new techniques evolve. [9]

A recent enhancement to search engine technology is the addition of geocoding and geoparsing to the processing of the ingested documents. Geoparsing attempts to match any found references to locations and places to a geospatial frame of reference, such as a street address, gazetteer locations, or to an area (such as a polygonal boundary for a municipality). Through this geoparsing process, latitudes and longitudes are assigned to the found places, and these latitudes and longitudes are indexed for later spatial query and retrieval. This can enhance the search process tremendously by allowing a user to search for documents within a given map extent, or conversely, plot the location of documents matching a given keyword to analyze incidence and clustering, or any combination of the two. One company that has developed this type of technology is MetaCarta, which makes its search technology also available as an XML Web Service to allow deep integration into existing applications. [12]

MetaCarta also provides an extension for desktop GIS software such as ESRI's ArcGIS, to allow analysts to interactively query the search engine and retrieve documents in an advanced geospatial and analytical context. [12]

A program that searches documents for specified keywords and returns a list of the documents where the keywords were found. Although search engine is really a general class of programs, the term is often used to specifically describe systems like Alta Vista and Excite that enable users to search for documents on the World Wide Web and USENET newsgroups.

Typically, a search engine works by sending out a spider to fetch as many documents as possible. Another program, called an indexer, then reads these documents and creates an index based on the words contained in each document. Each search engine uses a proprietary algorithm to create its indices such that, ideally, only meaningful results are returned for each query.

When people use the term search engine in relation to the Web, they are usually referring to the actual search forms that search through databases of HTML documents, initially gathered by a robot.

There are basically three types of search engines: Those that are powered by robots (called crawlers; ants or spiders) and those that are powered by human submissions; and those that are a hybrid of the two. (Onn Brandman, Hector Garcia-Molina, and Andreas Paepcke, 2000) [13]

Crawler-based search engines are those that use automated software agents (called crawlers) that visit a Web site, read the information on the actual site, read the site's meta tags and also follow the links that the site connects to performing indexing on all linked Web sites as well. The crawler returns all that information back to a central depository, where the data is indexed. The crawler will periodically return to the sites to check for any information that has changed. The administrators of the search engine determine the frequency with which this happens. (Onn Brandman, Hector Garcia-Molina, and Andreas Paepcke, 2000) [13]

Human-powered search engines rely on humans to submit information that is subsequently indexed and catalogued. Only information that is submitted is put into the index. In both cases, when you query a search engine to locate information, you're actually searching through the index that the search engine has created —you are not actually searching the Web. These indices are giant databases of information that is collected and stored and subsequently searched. This explains why sometimes a search on a commercial search engine, such as Yahoo! or Google, will return results that are, in fact, dead links. Since the search results are based on the index, if the index hasn't been updated since a Web page became invalid the search engine treats the page as still an active link even though it no longer is. It will remain that way until the index is updated. (Sergey Brin and Lawrence Page, 1998) [14]

So why will the same search on different search engines produce different results? Part of the answer to that question is because not all indices are going to be exactly the same. It depends on what the spiders find or what the humans submitted. But more important, not every search engine uses the same algorithm to search through the indices. The algorithm is what the search engines use to determine the relevance of the information in the index to what the user is searching for.

One of the elements that a search engine algorithm scans for is the frequency and location of keywords on a Web page. Those with higher frequency are typically considered more relevant. But search engine technology is becoming sophisticated in its attempt to discourage what is known as keyword stuffing, or spamdexing.

Another common element that algorithms analyze is the way that pages link to other pages in the Web. By analyzing how pages link to each other, an engine can both determine what a page is about (if the keywords of the linked pages are similar to the keywords on the original page) and whether that page is considered "important" and deserving of a boost in ranking. Just as the technology is becoming increasingly sophisticated to ignore keyword stuffing, it is also becoming savvier to Web masters who build artificial links into their sites in order to build an artificial ranking. (Sergey Brin and Lawrence Page, 1998) [14]

2.2 About Portable Document Format (PDF)

The Portable Document Format, or PDF, was first introduced by Adobe® Systems Incorporated in June 1993 with the announcement of a suite of products under the name of Adobe Acrobat®. Since then, PDF has become the focus of great attention as a key electronic format within the professional publishing industry and as the way to present printable material on the World Wide Web. (James C. King, August 2004) [15]

In 1990 the personal computers available were quite primitive by today's standards. The common IBM compatible PC shipped with 640K of easily accessible RAM. Color displays were just starting to become commonplace and the resolution even for black and white displays was so low that reading text for any length of time on the typical display was an eye straining endeavor. Networked computers and e-mail were still not widespread in industry although they had become well established among university computer scientists much earlier. There was a clear distinction in price and computing power between "workstations" and "personal computers". It was becoming clear that the widespread use of computers in business would be based upon the personal computer and not workstations, primarily because of cost. (James C. King, August 2004) [15]

All of this provides the technical backdrop in front of which PDF was designed. But there were other significant factors that determined the kind of document that PDF was to represent. Also note historically that in 1990 the World Wide Web (WWW) had not "happened" yet, HTML [5] was not yet invented and interactive, WYSIWYG (what you see is what you get) formatters and layout applications were just beginning to become practically useful, especially on personal computers. (J. Palme, May 1998) [16]

Formatting and layout was generally considered to be relatively complicated and difficult to do in real time. For a WYSIWYG authoring application, waiting for a page to refresh was not pleasing but was accepted. For a browsing or reading application it was not acceptable. So formatting and layout were something that you did once and the

browsing and reading was done primarily from multiple paper copies. PDF introduced the possibility of fast interactive reading and browsing because the formatting and layout were pre-computed. (J. Palme, May 1998) [16]

In June of 1993 the Addison-Wesley Publishing Company in collaboration with Adobe Systems Incorporated published the definitive book called "Portable Document Format Reference Manual" describing in technical detail the PDF version 1.0. Since that time revisions to this technical specification have been available to the public directly from Adobe Systems Incorporated. (Thomas A. Phelps, Robert Wilensky, November 2003) [17]

The current version of PDF is 1.5, which was announced and documented by Adobe in May 2003 along with the availability of Acrobat 6.0. (That document can be found on Adobe's Web site within the technical documentation available for developers at: http://www.adobe.com/supportservice/devrelations/technotes.html. Search for "PDF Spec".) From the start, Adobe has put only simple copyright restrictions on other people or companies ability to read or write PDF files. In fact, the Adobe Solutions Network (ASN) provides extensive software support for developing PDF oriented products for a modest annual membership fee. (Thomas A. Phelps, Robert Wilensky, November 2003) [17]

"Forcing users to browse PDF documents makes your website's usability about 300% worse relative to HTML pages. This is my rough estimate, based on watching users perform similar tasks on a variety of sites that used either PDF or regular Web pages. Because I have not performed a detailed measurement study of PDF on its own, I can't calculate the precise usability degradation. However, whether the true number is 280% or 320%, one thing is certain: the number is big and reflects significant user suffering in terms of increased task time and more frequent failures." (Jakob Nielsen, June 10, 2001) [3]

PDF was designed to specify printable pages. PDF content is thus optimized for letter-sized sheets of paper, not for display in a browser window. Users often getting lost in PDF because the print-oriented viewer gives them only a small peephole on a big, complicated layout and they can't scroll it in the simple, linear manner they are accustomed to on the Web. Instead, PDF files often use elaborate graphic layouts and split the content into separate units for each sheet of print. Although this is highly appropriate for printed documents, it causes severe usability problems online. (John Warnock, Chuck Geschke, December 2001) [18]

PDF pages lack navigation bars and other apparatus that might help users move within the information space and relate to the rest of the site. Because PDF documents can be very big, the inability to easily navigate them takes a toll on users. PDF documents also typically lack hypertext, again because they are designed with print in mind. . (John Warnock, Chuck Geschke, December 2001) [18]

In a recent study of how journalists use the Web, we found that PDF files sometimes crashed the user's computer. This happened most often to journalists working from home on low-end computers (especially old Macs). The more fancy the company's press kit, the less likely it would get quoted. (Dirk Eddelbüttel, William L. Goffe, October 1999) [19]

Because PDF is not the standard Web page format, it dumps users into a non-standard user interface. Deviating from the norm hurts usability because, for example, scrolling works differently, as do certain commands, such as the one to make text larger (or smaller). Also, after finishing with a PDF document, users sometimes close the window instead of clicking the Back button, thus losing their navigation history. Although this behavior is not common, it is symptomatic of the problems caused when you present users with a non-standard Web page that both looks different and follows different rules. (Dirk Eddelbüttel, William L. Goffe October 1999) [19]

CHAPTER 3
METHODOLOGY/PROJECT WORK

Methodology is defined as step-by-step approach that essential in every system development. In other word, methodology can also be refers as system development life cycle. In developing any kind of project, developer has to do detail analysis on what kind of methodology can be used to ensure its suitability with the nature of the project. There are many types of methodology that can be use by developers such as waterfall model, spiral model, rapid application development model and others. However, those methodologies cannot easily been choosing and used for every project.

3.1 Procedure Identification

After researches, studies, and some considerations had been performed, the most suitable methodology for the implementation of PDF Text Searching System is Rapid Application Development (RAD). In general, the methodology is defined as a software development process that allows usable systems to be built in as little as 60-90 days, often with some compromises.

The methodology is an increment software development process model that emphasizes an extremely short development cycle. The RAD model is "high speed" adaptation of the linear sequential model in which rapid development is achieved by using component-based construction. If requirements are well under stood and project scope is constrained, the RAD process enables a development team to create a "fully functional system" within very short time periods, as mentioned above – 60 to 90 days.

RAD is most popular for small to medium-size projects. Principles behind the definition of Rapid Application Design (RAD) are divided into three parts, which are:

1. In certain situations, a usable 80% solution can be produced in 20% of the time that would have been required to produce a total solution.

2. In certain situations, the business requirements for a system can be fully satisfied even if some of its operational requirements are not satisfied.

3. In certain situations, the acceptability of a system can be assessed against the agreed minimum useful set of requirements rather than all requirements.

As referred to the above principles, given the maximum duration of the project development is only 12 weeks; the minimum standard for the project is at least to be at the third principle.

By applying RAD in the implementation of the project, project requirements can be limited and the development time can be saved possibly at the expense of economy or the product quality itself. These highlighted advantages are possible to be achieved as the duration of the project is only 12 weeks, hence the requirements and the time spent to develop the project can not be simply modified or altered as it may affect the overall project.

Some companies offer products that provide some or all of the tools for RAD software development. These products include requirements gathering tools, prototyping tools, computer-aided software engineering tools, language development environments such as those for the Java platform, groupware for communication among development members, and testing tools. RAD usually embraces object-oriented programming methodology, which inherently fosters software re-use. The most popular object-oriented programming languages, C++ and Java, are offered in visual programming packages often described as providing rapid application development.

As mentioned above, Rapid Application Development has two primary advantages; increased speed and increased quality. The speed increases are due to the use of CASE tools, the goal of which is to capture requirements and turn them into usable code as quickly as possible. Quality, as defined by RAD, is defined as both the degree to which a delivered application meets the expected objectives as well as the degree to which a delivered system has low maintenance costs.

3.2 Process Flow

The following diagram depicts the dependency relationships between the stages in the Rapid Application Development Process template.

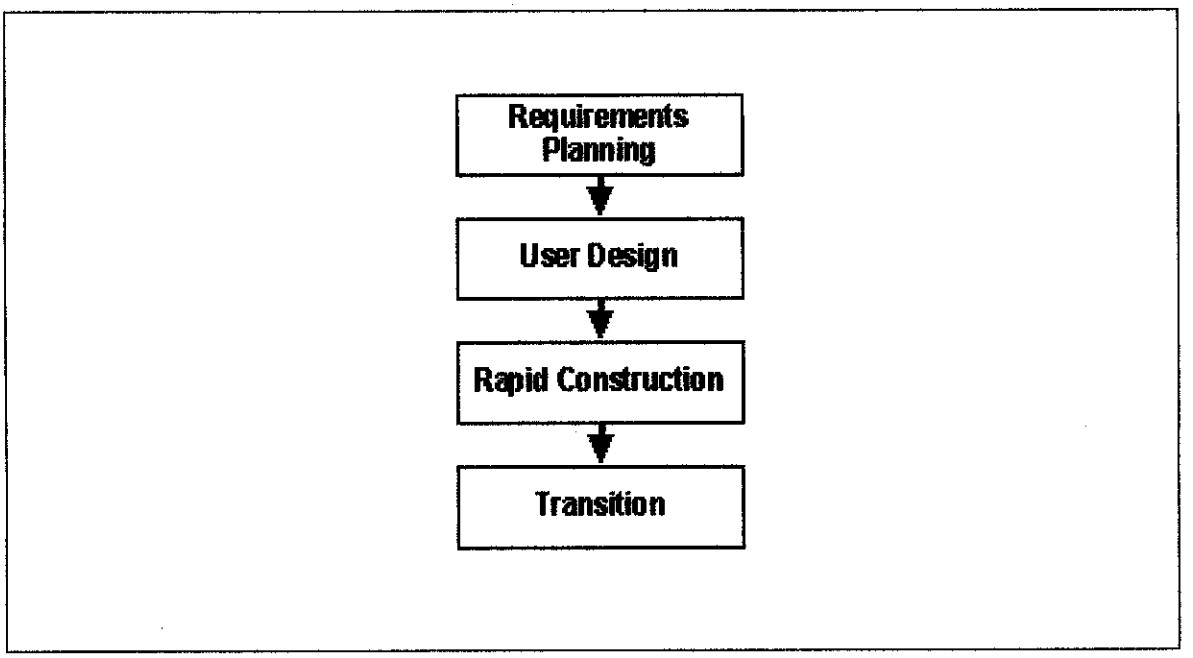

Figure 3.1: Rapid Application Development Process template

3.2.1 Four Phases of RAD Lifecycle

To help ensure that developers build what the user really needs the RAD lifecycle has four phases:

- Requirements planning phase.
- User design phase.
- Construction phase.
- Cutover phase.

Requirements Planning Phase

The requirements planning phase requires that high level or knowledgeable end-users determine what the functions of the system should be. It should be a structured discussion of the business problems that need to be solved. It can often be done quickly when the right users and executives are involved.

User Design Phase

The user design phase requires the users to participate strongly in the no technical design of the system, under the guidance of IS professionals. User design is done in a Joint Application Design (JAD) workshop. In the first two phases the users and executives should play a larger part than the IS professionals. Prototyping is used to aid in requirements specification and design. The user does not sign off a paper design, they sign off a CASE representation.

Construction Phase

The design created during the User Design Phase is added to using I-CASE tools. As each transaction is built it may be demonstrated to the end-users for revision. The CASE environment allows for the continuous changes in design. End-users are closely involved in the construction phase. Testing occurs throughout the process. The I-CASE

toolset should generate the code as well as the database descriptions for the final product. Code optimizers may be used to improve the performance of the generated code.

Cutover Phase

When the cutover phase occurs, a variety of actions are needed, comprehensive testing, training of the end-users, organizational changes and operation in parallel with the previous system until the new system settle in.

3.3 Deliverables for Each RAD Phase

In this section, the deliverable of each phases will be discuss. Every phase is necessary to produce output or deliverable that will be use as the input for the next phase. Without deliverable, the process of development cannot be proceeding.

Deliverable Phase 1 – During this phase, the scope and objectives of the project have been defined in order to have a clear picture of the project. The feasibility study also has been carried out in order to ensure the project is worthwhile. All the problems and constraints regarding to the system development has been identified before conducting the system requirements study. This is important to identify the problems in order to make sure the development of the systems is running properly. System requirements also can be regard as guideline to developer in designing the system. In this stage, a system planning also has been carried out to plan for other phases. These are the outputs on the planning stage:

- Determine possible problems.
- Determine possible solutions.
- Determine system requirement specification
- Define the scope and objectives.
- Project schedule timeline.

From the analysis that have been done during this phase, the author have come out with the requirements specification that were stated as below which was divided into 3 sections; development tools requirement, workstations requirements and security requirements.

- **Development Tools Requirements**

    - Macromedia DreamweaverMX

Macromedia Dreamweaver MX is an easier tool used to design a website which makes the ordinary and repetitive tasks of coding easier. The visual editing features in Dreamweaver let us quickly create pages without writing a line of code. However, Dreamweaver also includes many coding-related tools and features. It helps us to build dynamic database-backed web applications using server languages such as ASP, ASP.NET, ColdFusion Markup Language (CFML), Javascript, and PHP.

Macromedia Dreamweaver MX provides a set of visual objects that can be drawn easily onto a window. These controls eliminate the need to develop the code to construct visual interface. The layout of the windows that contain the controls can be changed easily by dragging and dropping the controls to a new location, without require a change in the code. The process for program development and revision becomes much easier and requires much less time and effort.

    - Javascript

JavaScript may be considered a derivative of the programming language Java. But while both are tools for providing interactivity into web pages, they are as different as bananas and papayas.

Java is a complex programming environment where it creates packaged ("compiled") software applications that can be inserted into a web page. The learning curve for Java

is monumental at best (despite claims of the expanding number of software tools). On the other hand, JavaScript offers a simpler set of programming instructions that the author can enter directly among the HTML formatting of the web pages, and code that can be easily accessed and modified.

Before JavaScript, to create interactive forms (web pages with fields, buttons, and menus) it needed to write computer programs ("CGI" scripts) that resided on and ran from a web server. But with JavaScript, the author can perform many form tasks without connecting to a web server. In the jargon, we are processing on the "client-side".

Even better, JavaScript allows creating content that is dynamic, so that the code inside one web page can produce many different types of displays and features depending on the viewer's actions, including the images that change when you move the mouse over a graphic. JavaScript combined with the absolute screen positioning available in web browsers that support HTML 4.0 provide what is known as Dynamic HTML, or DHTML.

- *Advantages of Javascript*

As stated above, JavaScript provides interactivity for the web pages without relying on server-side "CGI") programming, which means the pages can be interactive even when thay are not connected to the Internet. Since the code is typed directly into the HTML files, the author can create Javascript with software as simple as a plain text editor. It can quickly test and modify JavaScript code. JavaScript functionality is built into most newer web browsers since 1996, so there is no extra software for the viewer to download or install.

JavaScript also provides useful commands for testing the viewers' capability to view other types of web multimedia. Although not all web browsers may support JavaScript, there are fairly reliable methods for the author to direct viewers to alternative pages.

- Adobe Acrobat Reader 7.0 Professional

Short for Portable Document Format, PDF is a file format developed by Adobe Systems. PDF captures formatting information from a variety of desktop publishing applications making it possible to send documents and have them appear on the recipient's monitor (or printer) as they were intended to be viewed. A properly prepared PDF will maintain the original fonts, images, graphics as well as the exact layout of the file (think of it as an electronic snapshot). A PDF file can be shared, viewed, and printed by anyone using the free Adobe Reader software regardless of the operating system, original design application or fonts.

- **Workstation Requirements**

The requirements for the workstation in order to run the system effectively, the PC must have the minimum hardware and software. The requirements for the workstation were stated as follows:

- An INTEL based PC with the minimum speed 100 MHz
    - We have to comply with this minimum speed because the CPU processor speed is important to make the systems is running smoothly and the response time to wait the completion of the data process is short.

- 50 MB of memory
    - The systems need a space on the hard drive to perform the software, 50 MB is the minimum space needed to run the system.

- CD ROM drive
    - CD ROM drive is used for the installation of the system but it is optional, as the system can be distributed via network.

- Other requirements:
    - 32 MB RAM drive
    - Microsoft Windows 2000/ XP/ NT 4.0/ UNIX
    - Network card

- **Security Requirements**

To maintain the security level of the systems, the author has specified to use the following access right parameters:

- User Level Access

  User only can view the data without having the access to modify and manipulate current database. This is important to maintain the integrity of the database.

- Database Administrators

  DBA is the owner of the systems who has the access to make a change or modification throughout the times, subject to the future requirements.

Deliverable Phase 2 – This phase cover the activities done in designing the system including detail design of context diagram and data flow diagram (DFD), and Graphical User Interface Design (GUI). This deliverables then is use as the input for the next phase, which is constructing phase. The detail design for the context diagram and data flow diagram, and GUI is important because the shape of the system will be based on this phase. So, the detail design phase was divided into two sections, which are context diagram and DFD.

Context Diagram

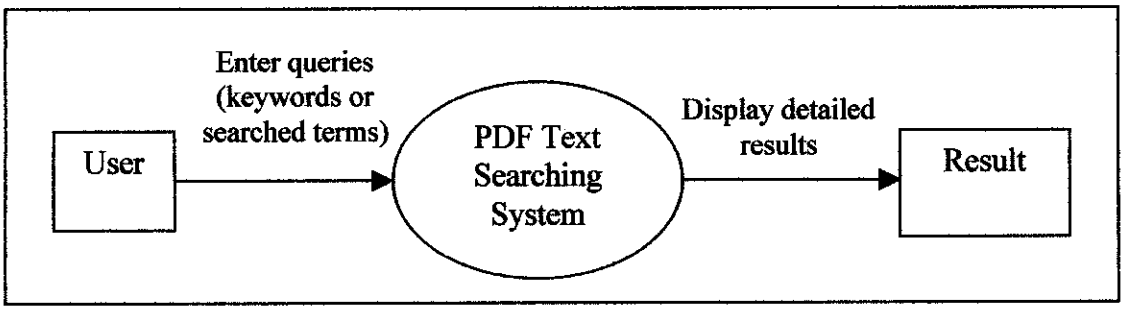

Figure 3.2: Context Diagram of PDF Text Searching System

Data Flow Diagram

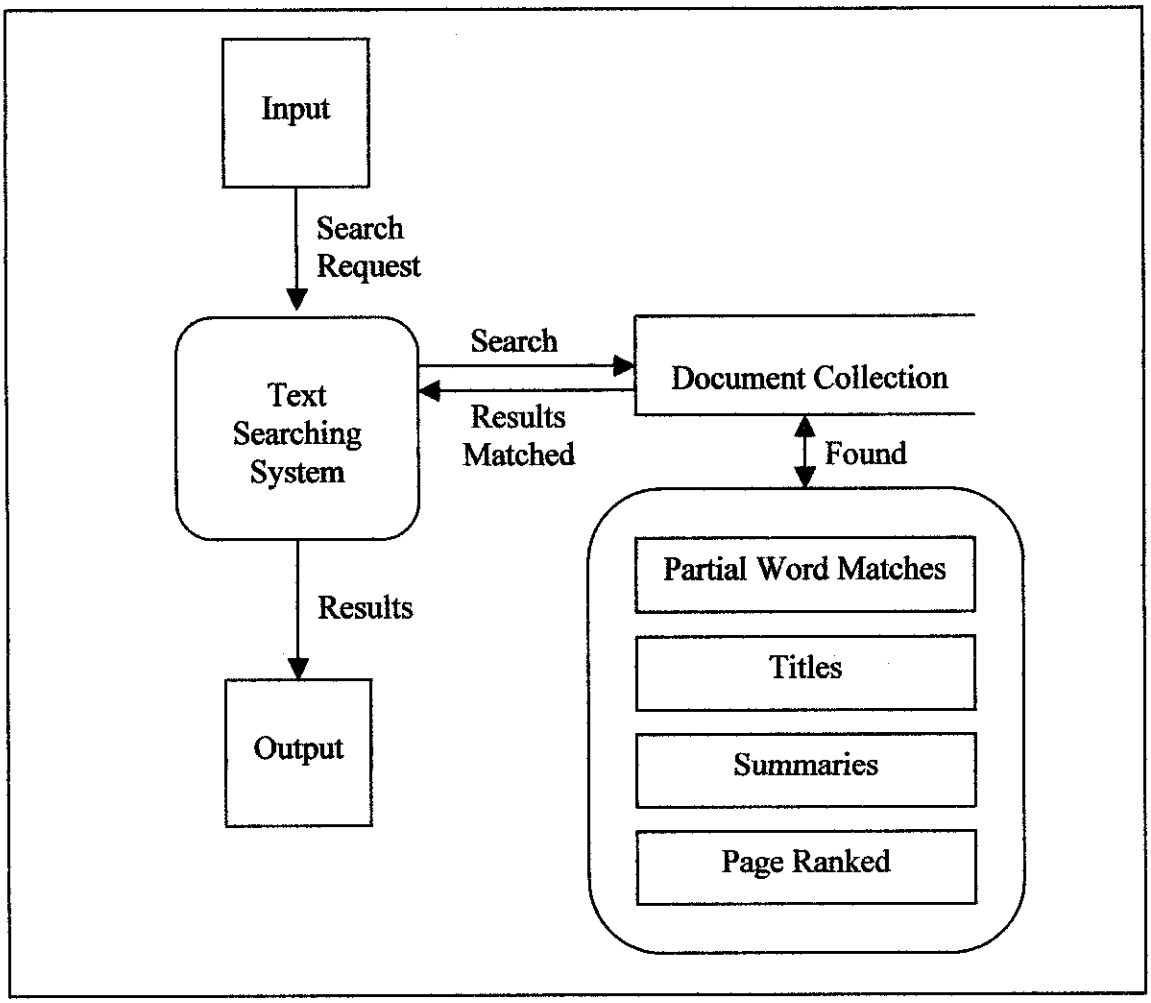

Figure 3.3: Data Flow Diagram of PDF Text Searching System

Deliverable Phase 3 – In this phase, system development has been took place to turn the detailed design into code. Complete system, which is the final product, is the deliverable for constructing phase, which will be the input for the next phase. A complete system is means that user has tested the system and they are agreeing to accept the system. For this stage, the development of system is including develop the template design for interface into system, create data entry, and database (local drive C).

Deliverable Phase 4 – In this phase, system testing is being conducted to detect and fix the bugs and errors. It was divided into two, which are developer testing and user testing. The testing comes with the intent to ensure that the system meets all the requirements stated during the early phase. Complete project documentation is the final deliverable for the final phase of the Rapid Application Methodology (RAD). This not means that the output is not going to be used. The documentation will be keep for references for other project and also for reference to other persons who will enhance this project in future.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Results

1. The results on this report are based on the data from questionnaires conducted by the author, among a sample of 30 people, from the age of 15 to 55 years old. All of these respondents are familiar about the computer usage and have experienced using it. The author have identified that there are three major groups of respondents involved in answering the questionnaires. The first group consisted of school/college/university students, aged from 15 to 25. The second group involved with junior executives, aged from 26 to 32. Last but not least, the third group consisted of the senior citizens (house wives, senior executives), aged from 33 to 55 years old. Based on the survey conducted, most of the respondents have reacted to a positive feedback towards the PDF Text Searching System especially from the respondents of the first and second group.

   27% of the respondents strongly agree with PDF Text Searching System. The second highest percentage denotes by the respondents who agree with the system that is 43%. This concludes that 70% of the respondents agree with the system. Follow by 23% are the respondents perceived the system neither agree nor disagree. 7% of the respondents disagree while the remaining also 0% strongly disagrees with PDF Text Searching System.

Please refer to table 4.1 for the results on the perception of respondents towards PDF Text Searching System. The result of this survey can be evaluated and influence the respondents acceptance towards this research project.

Table 4.1: Perception towards PDF Text Searching System

	PERCEPTION TOWARDS PDF TEXT SEARCHING SYSTEM					
	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	**Total**
Respondents	0	2	7	13	8	**30**

2. With the statistic of 70% from the total respondents (inclusive of agree and strongly agree), the respondents feels that the PDF text searching system would definitely help in assisting the users finding their appropriate documents. Therefore, this research project is proposed together with the enhancement of the system itself.

3. Based on the surveys concerning the level of acceptance of the proposed PDF Text Searching System, 30% of the respondents strongly agree that the system is a good application to be implemented in assisting the users to find their documents. 50% of the respondents agree with the system. This is followed by 17% of respondents neither agree nor disagree and 3% of respondents disagree of the system. The remaining 0% of the participants strongly disagrees. Please refer to Table 4.2 for the results on the level of acceptance of proposed PDF Text Searching System.

Table 4.2: Level of Acceptance of Proposed PDF Text Searching System

	LEVEL OF ACCEPTANCE OF PROPOSED PDF TEXT SEARCHING SYSTEM					
	Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree	**Total**
Respondents	0	1	5	15	9	**30**

4. With the statistic of 80% from the total participants (inclusive of agree and strongly agree), the respondents agree with the implements of PDF Text Searching System in assisting users through finding their appropriate documents.

5. Based on the surveys concerning the level of satisfactory on the searched result of the proposed PDF text searching system, 20% of the total searched result is strongly satisfy. 53% of the total searched result is satisfied with the searched result. The remaining 17% and 10% belong to neither satisfy nor dissatisfy and dissatisfy with the end of result. Please refer to Table 4.3 for the results on the level of satisfactory on the searched result of the proposed system.

Table 4.3: Level of Satisfactory on the Searched result of the Proposed PDF Text Searching System

	LEVEL OF SATISFACTORY ON THE SEARCHED RESULT OF THE PROPOSED PDF TEXT SEARCHING SYSTEM					
	Strongly Dissatisfy	Dissatisfy	Neither Satisfy Nor Dissatisfy	Satisfy	Strongly Satisfy	**Total**
Searched Result	0	3	5	16	6	**30**

6. With the statistic of 73% from the total searched result (inclusive of satisfy and strongly satisfy), the level of satisfactory of the proposed PDF Text Searching System is satisfied.

4.2 Findings

From the survey, majority of the respondents are satisfied and pleased with PDF Text Searching System. This can be proved using figure 4.1 below. Based on the figure 4.1, more than half of the respondents agree with the system. This is directly showed that the system is acceptable among the users.

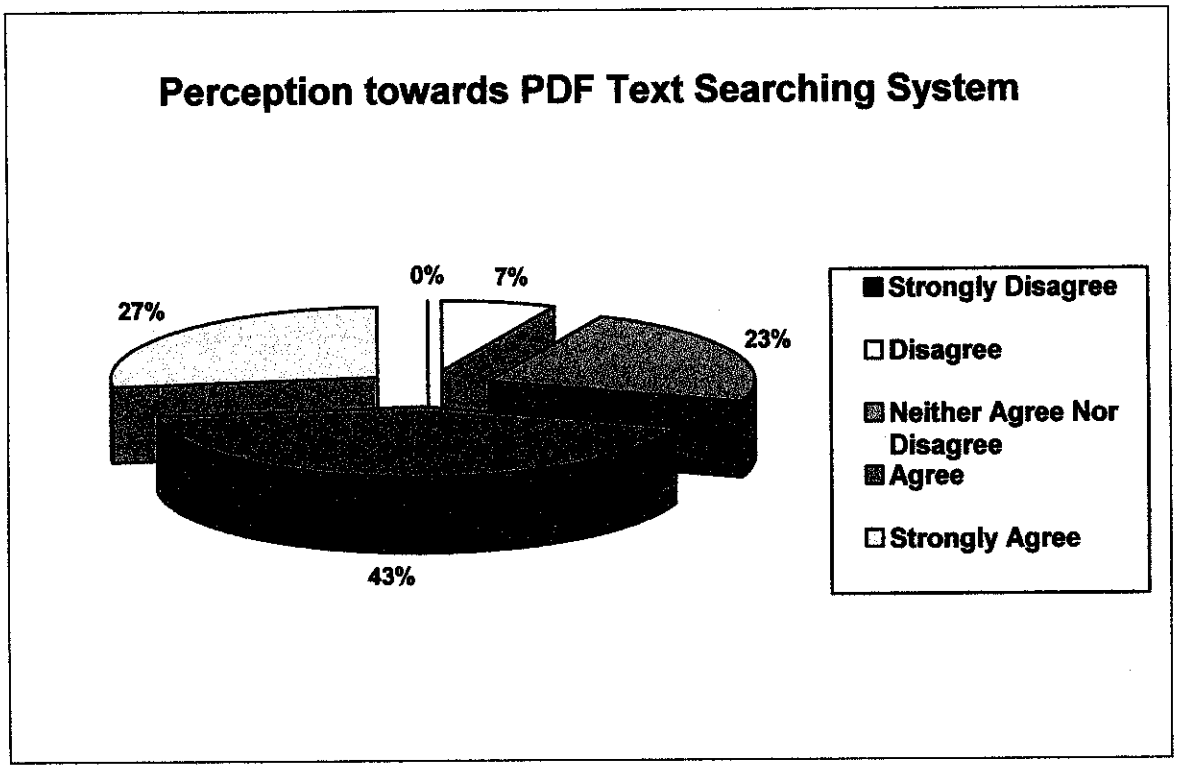

Figure 4.1: Perception towards PDF Text Searching System

Based on the figure 4.2 below, majority of the respondents agree that the proposed PDF Text Searching System would give a good impact in assisting the users in finding their appropriate documents through the system. From the result itself, the users are welcoming and able to accept the new approach and technology that led to high quality and performance in finding text and documents rather than go back to the traditional ways of finding information.

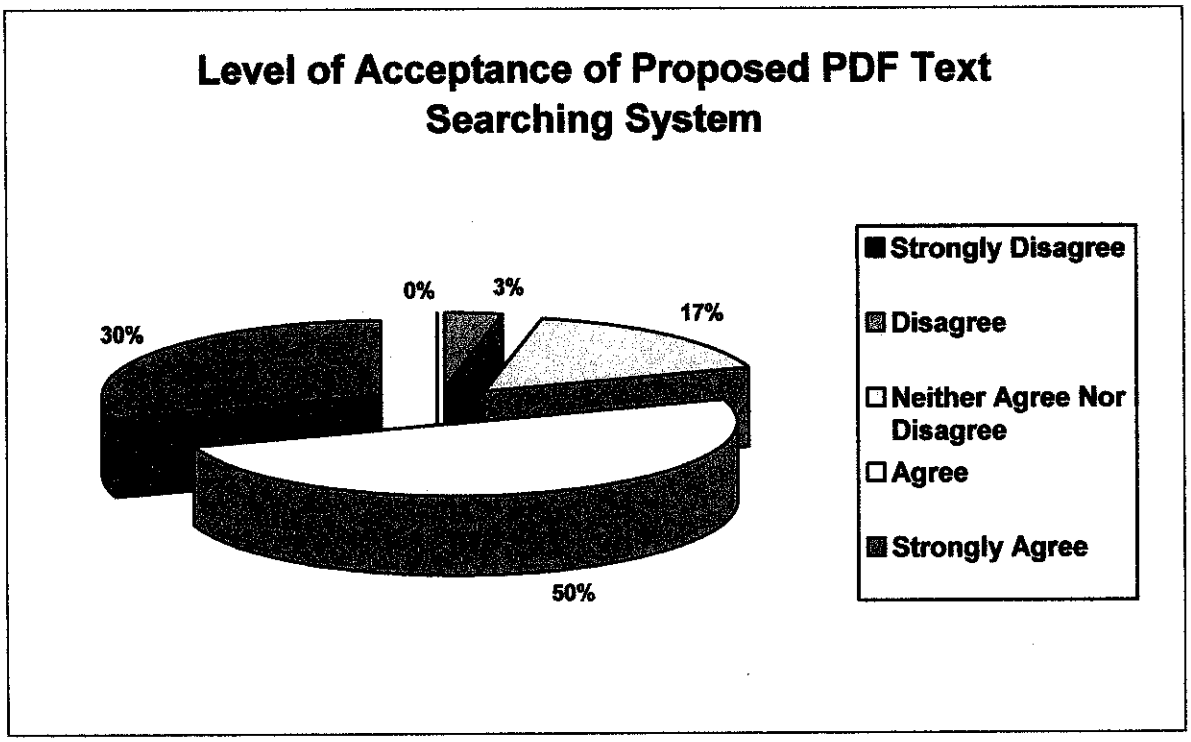

Figure 4.2: Level of Acceptance of Proposed PDF Text Searching System

Based on the figure 4.3 below, the level of satisfactory on the searched result of the proposed PDF text searching system is satisfied by most of the users. This shows that the searched result produced by the system is acceptable and meet with most of the user's requirement.

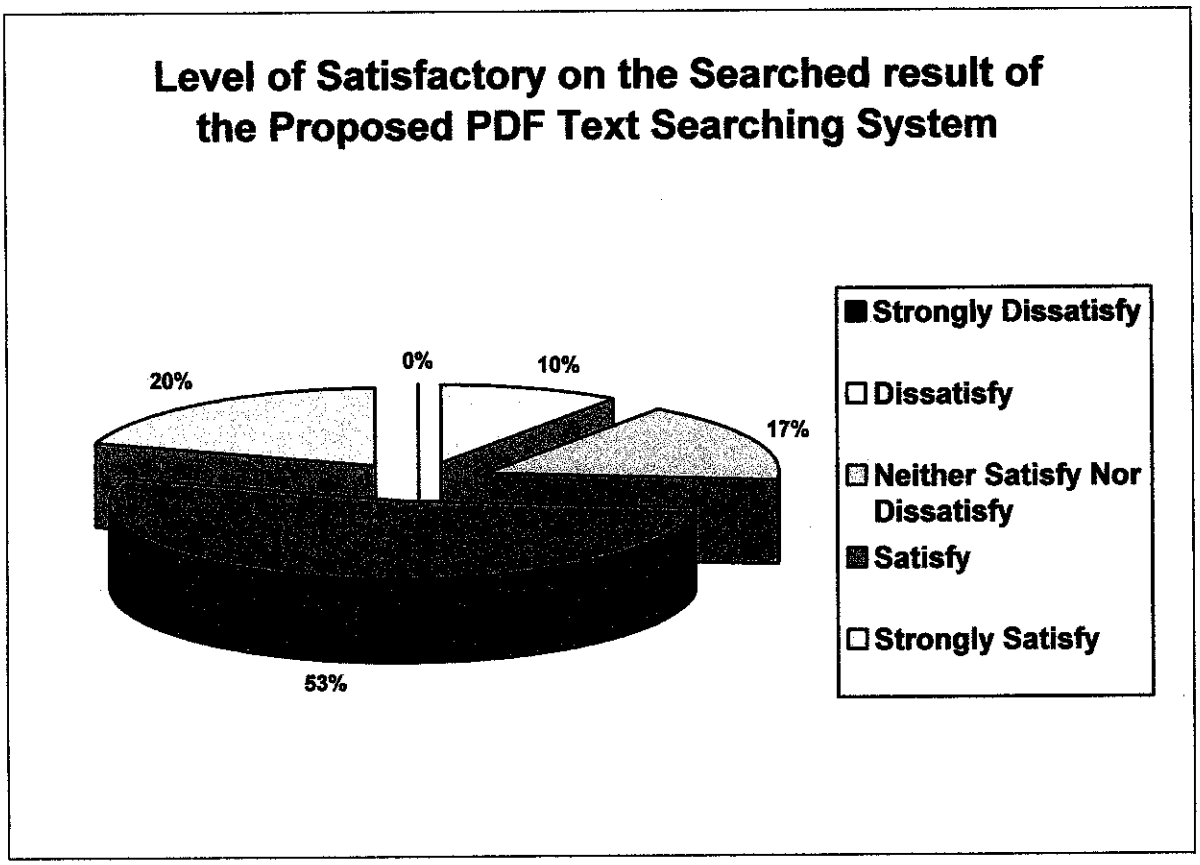

Figure 4.3: Level of Satisfactory on the Searched Result of the Proposed PDF Text Searching System

4.3 User Manual

In this section, users will be shown and explained on how to use the system. In other words, users will be explained about the function of each element in the system.

This is the main page for PDF Text Searching System.

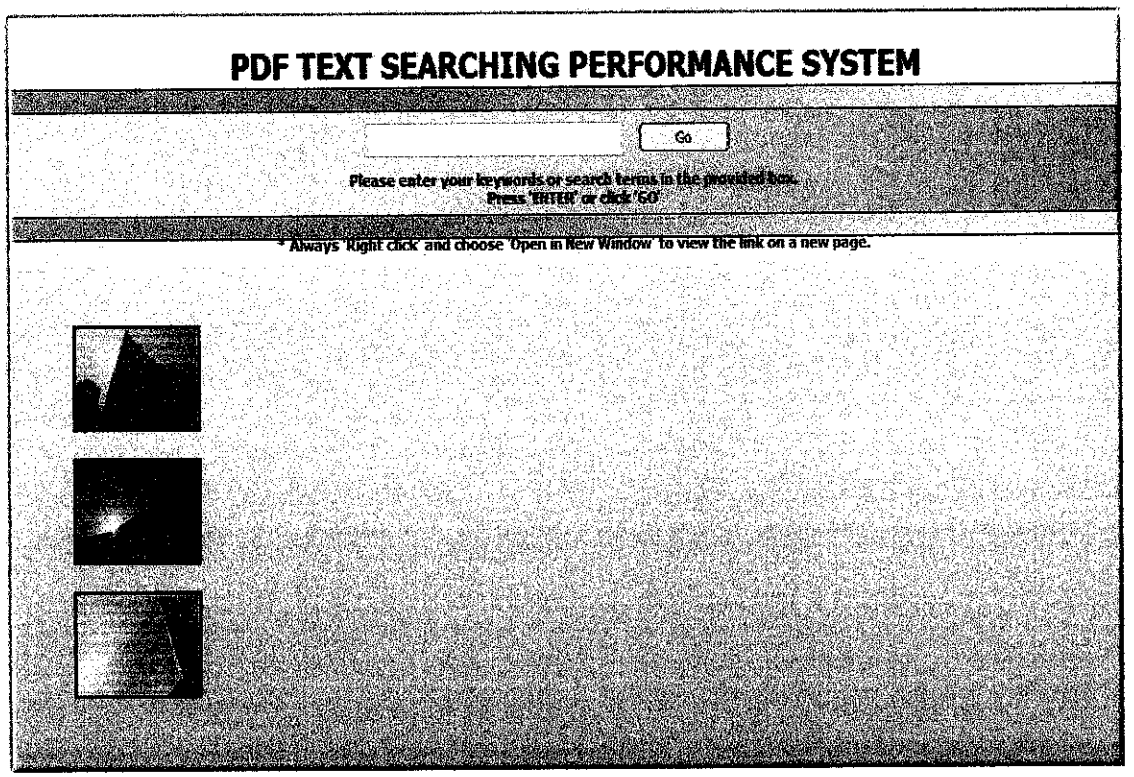

Figure 4.4: Main page of PDF Text Searching System

On the front page of the system, users are provided with a textbox. To begin searching for the text, users are required to enter their keywords or their search terms in the provided box. To view the result, users need to press 'Enter' or click on the button 'Go'. Below is the screen shot of the page.

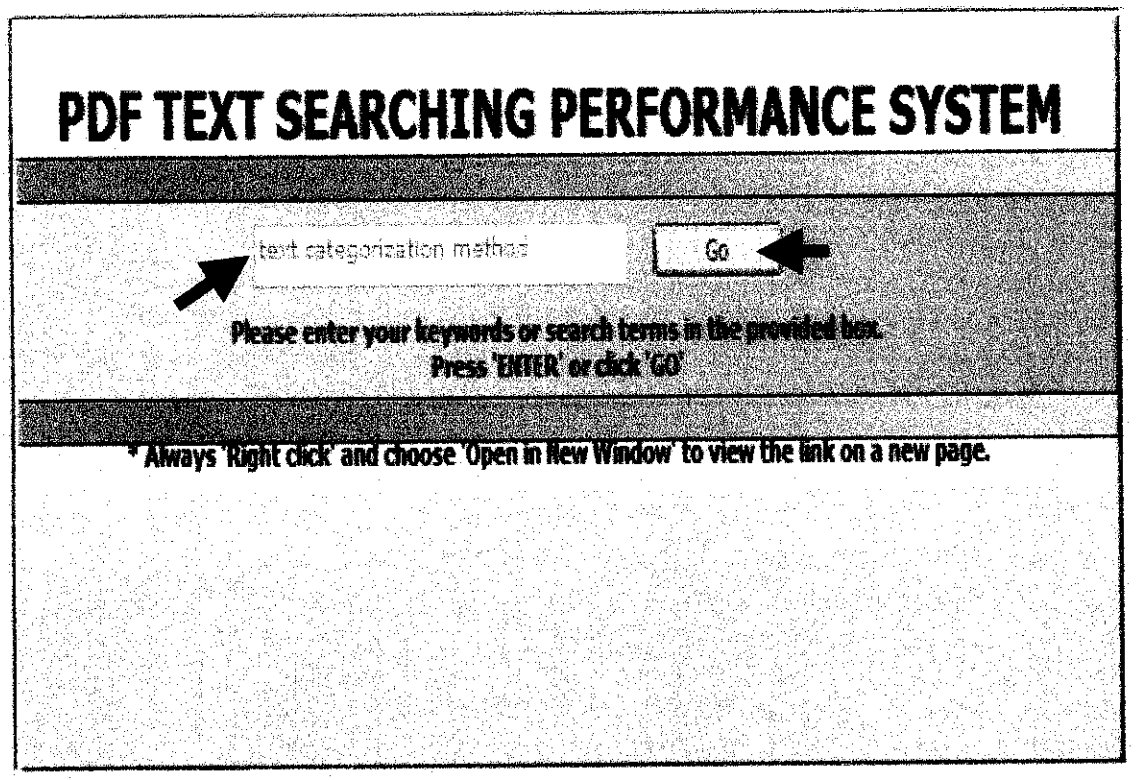

Figure 4.5: Enter keywords or search terms in the provided box.

This is the result page of PDF Text Searching System. The output of the system is the results with titles of the documents, summaries and links for the actual files. They are displayed in google format. Below is the screen shot of the result page.

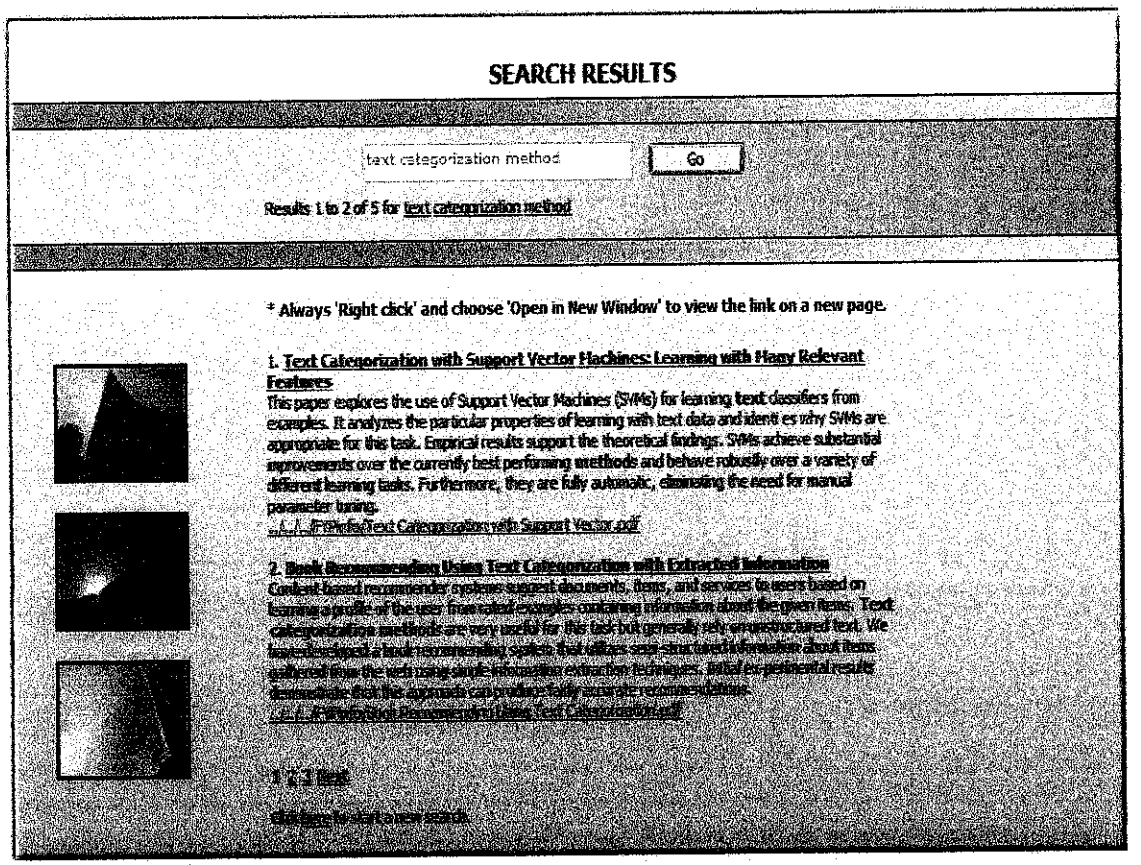

Figure 4.6: The result page

After examining the searched results, users can click on the link (below the summary of the files) to view on the actual files based on the searched results. Below is the screen shot of the page.

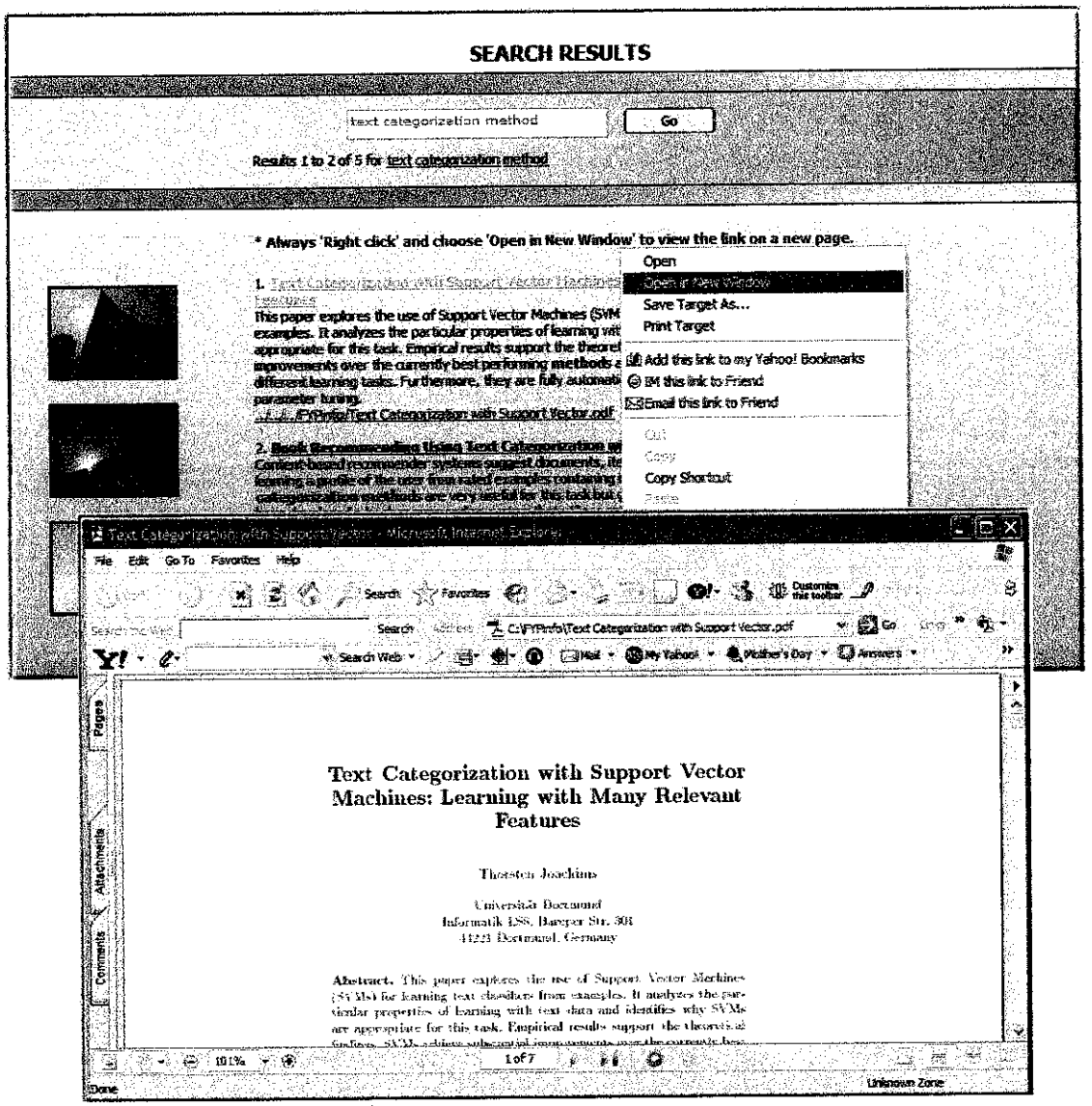

Figure 4.7: View the actual file

If users ever to meet this page, it means that the keywords or the search terms that they have entered did not matched with any documents in the system. Users are advised to insert other keywords or search terms to proceed in using the system. Click on the link 'here' to start with a new search.

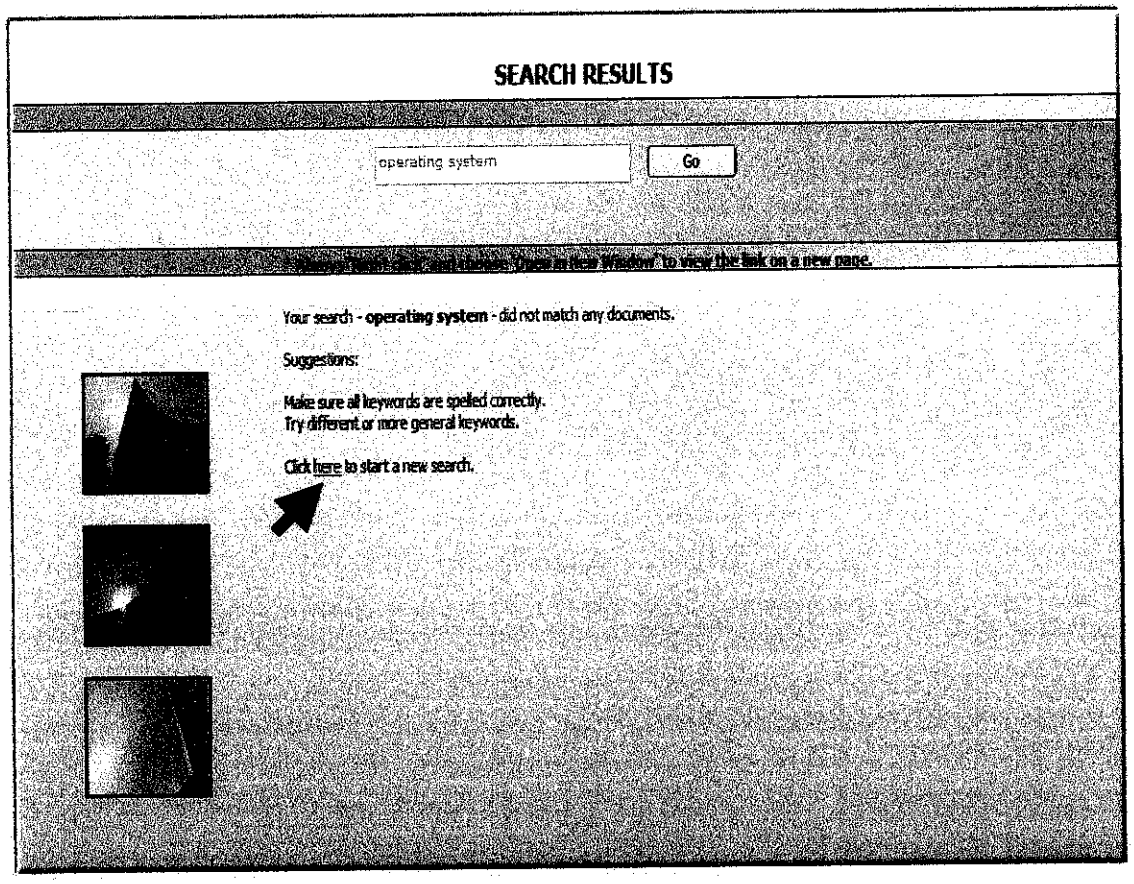

Figure 4.8: Keywords did not match with any document.

CHAPTER 5
CONCLUSION & RECOMMENDATION

5.1 Conclusion

In this chapter, the author will conclude the overall project that has been progressed out. For this project, there are two objectives that have been set up. The first objective is to study and have a better understanding on the software that will be used in order to develop PDF text-searching system. In order to satisfy this objective, the author has gather information from variety of sources such as Internet, books and asking some advice and idea from experts. The purpose to perform the study is to find out the information about text searching system and search engine system such as its concept and functionality.

The first objective of the project is important and prerequisite for the second objective to be executed. The second objective is to develop a simple PDF text-searching system, which is capable of searching and processing the information in text files on user PC and in local networks. The knowledge regarding text searching and search engine have been gathered to fulfill the first objective is then used for system development purpose. This is the key factor that determines the achievement of the project. The successful or failure of the project is depend on the ability to accomplish all of the objectives that have been defined during the beginning of the project.

In conclusion, many important things should be realized and take into consideration in developing any kind of project. The most important thing is to clearly understand the scopes and objectives of the project. This is to ensure the end product of the project meet all the requirements and expectations. The second thing is to determine the time

to be taken to complete each task and follow the schedule that has been set up during the planning phase. This is to ensure the progression of the project run smoothly and able to be completed as scheduled. Next, is to ensure all information and data regarding the project is properly documented into meaningful information and stored for future reference. The selection of the suitable methodology is also important. Developer cannot easily employ any methodology that they like without performing proper analysis and investigation. This is because every methodology has difference purpose and can resulting different impact.

5.2 Recommendations

There are several recommendations and suggestion that can be done in the future:

- Full text search capabilities

In text retrieval, full text search refers to a technique for searching a computer stored document or database; in a full text search, the search engine examines all of the words in every stored document as it tries to match search words supplied by the user. Currently, this research project has the ability to search on selected page of the document or file. Due to the time constraint and lack of knowledge about text searching technology, the research project can only focus and concentrate in limited page of the document or file.

- Stop words

Stop words are those words, which are so common that they are useless to index or use in search engines or other search indexes. Usually articles, adverbials or ad positions are stop words. In English some obvious stop words would be "a", "of", "the", "I", "it", "you", and "and". Presently, the system does not recognize any word as being a stop word with respect to indexing. That is, all words are effectively searchable.

There is no definitive list of stop words, as they can depend on the purpose of the search. Full phrase searches would not want words removed.

- Stemming

A stemmer is a computer program or algorithm which determines a stem form of a given inflected (or, sometimes, derived) word form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root.

A stemmer for English, for example, should identify the string, "stemmer", "stemming", "stemmed" as based on "stem".

English stemmers are fairly trivial (with only occasional problems, such as "dries" being the third-person singular present form of the verb "dry", "axes" being the plural of "axe" as well as "axis"); but stemmers become harder to design as the morphology, orthography, and character encoding of the target language becomes more complex.

- Boolean queries.

Searches that use Boolean operators (for example, "encyclopedia" AND "online" NOT "Encarta") can dramatically increase the precision of a free text search. The AND operator says, in effect, "Do not retrieve any document unless it contains both of these terms." The NOT operator says, in effect, "Do not retrieve any document that contains this word." If the retrieval list retrieves too few documents, the OR operator can be used to increase recall; consider, for example, "encyclopedia" AND "online" OR "Internet" NOT "Encarta". This search will retrieve documents about online encyclopedias that use the term "Internet" instead of "online."

REFERENCES

[1] "What is PDF? A word Definition From the Webopedia Computer Dictionary" URL: http://www.webopedia.com/TERM/P/PDF.html, (Accessed in March 2006)

[2] Maria O'Daniel, "Pros and Cons Using PDF Documents", *News Strait Times*, September 15, 2003

[3] Jakob Nielson, "Avoid PDF for On-Screen Reading", June 10,2001 URL: http://www.useit.com/alertbox/20010610.html (Accessed in March 2006)

[4] "Information Search Problem", *SearchInform: SoftInform Technology,* 2004. URL: www.searchinform.com/site/en/main/search-inform-technology.htm (Accessed in May 2006)

[5] Jakob Nielsen, "PDF: Unfit for Human Consumption", July 14, 2003. URL: http://www.useit.com/alertbox/20030714.html (Accessed in March 2006)

[6] URL: http://en.wikipedia.org/wiki/Full_text_search (Accessed in February 2006)

[7] H.W. Wilson, "Stopwords in WilsonWeb", URL: http://www.hwwilson.com/default.cfm (Accessed in May 2006)

[8] Sergey Melnik, Sriram Raghavan, Beverly Yang, Hector Garcia-Molina, "Building a Distributed Full-Text Index for the Web", *Stanford University,* 2001. URL: http://www-db.stanford.edu/~melnik/pub/melnik_TOIS01.pdf (Accessed in March 2006)

[9] Sepandar Kamvar, Taher Haveliwala, Chris Manning, and Gene Golub. "Extrapolation Methods for Accelerating PageRank Computations", *Stanford University*, 2003. (Accessed in March 2006)

[10] Andreas Paepcke, Hector Garcia-Molina, and Gerard Rodriquez, "Collaborative value filtering on the Web", *Stanford University*, 2000
URL: http://dbpubs.stanford.edu/pub/showDoc.Fulltext?lang=en&doc=2000-47&format=pdf&compression= (Accessed in March 2006)

[11] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, "The Google File System" 2003,
URL: www.cs.rochester.edu/sosp2003/papers/p125-ghemawat.pdf (Accessed in March 2006)

[12] "Geospatial Ranking of Search Engine Results"
URL: http://www.cs.dal.ca/~watters/Watters2.doc (Accessed In April 2006)

[13] Onn Brandman, Hector Garcia-Molina, and Andreas Paepcke, "Crawler-Friendly Web Servers", *Stanford University*, 2000.
URL: http://dbpubs.stanford.edu/pub/showDoc.Fulltext?lang=en&doc=2000-25&format=pdf&compression=

[14] Sergey Brin and Lawrence Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", *Stanford University,* 1998.
URL: http://dbpubs.stanford.edu/pub/showDoc.Fulltext?lang=en&doc=1998-8&format=pdf&compression=

[15] James C. King, "A format design case study: PDF", *Proceedings of the fifteenth ACM conference on Hypertext and hypermedia,* ACM Press, August 2004
URL: http://portal.acm.org/citation.cfm?doid=1012810

[16] J. Palme, "RFC2346: Making Postscript and PDF International", May 1998
URL: www.rfc-archive.org/getrfc.php?rfc=2346

[17] Thomas A. Phelps, Robert Wilensky, "Optimizing document format: Two diet plans for fat PDF", *Proceedings of the 2003 ACM symposium on Document engineering,* November 2003.
URL: http://portal.acm.org/citation.cfm?id=958253

[18] John Warnock, Chuck Geschke, "PDF Reference: Adobe Portable Document Format Version 1.4 with Cdrom", *Addison-Wesley Longman Publishing Co., Inc.,* December 2001.
URL: http://www.powells.com/cgi-bin/partner?partner_id=719&cgi=product&isbn=0201758393

[19] Dirk Eddelbüttel, William L. Goffe, "Display and Interactive Languages for the Internet: HTML, PDF, and Java", *Computational Economics,* Volume 14 Issue 1-2, Kluwer Academic Publishers, October 1999.
URL: http://heldref.metapress.com/index/NW51132173054552.pdf

APPENDICES

Appendix 1: About PDF Text Searching System

Strongly Disagree (Worst)	Disagree	Neither Agree Nor Disagree (Average)	Agree	Strongly Agree (Good)
1	2	3	4	5

With reference to the Likert Scale above, circle the extent to which you agree with the following statement

The system is easy to use	**1**	**2**	**3**	**4**	**5**
The system flow is easy to follow	**1**	**2**	**3**	**4**	**5**
The content of the results is understandable	**1**	**2**	**3**	**4**	**5**
The result made by the system is useful	**1**	**2**	**3**	**4**	**5**
I am very satisfy with the search results	**1**	**2**	**3**	**4**	**5**
I am very confident of my searching abilities to find the information	**1**	**2**	**3**	**4**	**5**
I will use this system for searching text and documents on PDF	**1**	**2**	**3**	**4**	**5**

Please give your suggestions to improve the system/website:

_____

_____

_____

_____

THANK YOU!

APPENDIX 2: Project Timeline for One (1) Semester Final Year Project

No.	Details/ Activities	Week													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Research Title Initial Proposal														
2	Preliminary Research Work - Introduction - Objective and Scope of Study - Methodology - Literature Review - Project Planning														
3	Submission of Preliminary Report														
4	Project Work - Reference/ Literature - Tools and Software - User Design Process - Contruction Phase														
5	Submission of Progress Report														
6	Project Work Continue														
7	Submission of Dissertation Final Draft														
9	Preparation on Project Dissertation														

Work Plan Submission Date