A DEVELOPMENT OF A CANCER INFORMATION SYSTEM BY USING INFORMATION RETRIEVAL TECHNIQUES

TAN CHEE MING

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY UNIVERSITY OF MALAYA
KUALA LUMPUR

2006

A DEVELOPMENT OF A CANCER INFORMATION SYSTEM BY USING INFORMATION RETRIEVAL TECHNIQUES

TAN CHEE MING

DISSERTATION SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE MASTER OF SOFTWARE ENGINEERING

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY UNIVERSITY OF MALAYA
KUALA LUMPUR

2006

Abstract

This project has being specifically focuses on the development of a Cancer Information System by using information retrieval (IR) techniques. In this regards, this project tends to produce a prototype system to provide user with two types of cancer information, specifically breast cancer and lung cancer, and a category called “general cancer” that discusses the common characteristics regarding cancer disease.

The system consists of two parts; the Web-based search engine and the documents collection stored in local filesystems. Despite traditional hyperlinks method of Website navigation, the system enables user to search for information by entering natural language query. The query will be processed by the information search engine and then return a maximum of six most relevant documents that possibly contains the information requested by the user. User can then browse any of these documents for detail information.

Acknowledgements

First of all, I would like to thank Pn. Norisma Idris, my project supervisor, for her comments, corrections, advices, and discussions in guiding me throughout the development of this project.

I would also like to extend my gratitude to my family especially my parents for their continuous supports and encouragements since the beginning of the project.

Last but not least, my thanks to all of my friends who have demonstrated unwavering support and encouragement throughout this project.

Table of Contents

List of Figures

List of Tables

Chapter One: Introduction

1.1 Overview

Although there really is no way to know the exact number of cells in a human body, several researchers estimated that average adult human body make up of 50 to 75 trillion cells (eNotes.com, 2006), the smallest structural unit of an organism that is capable of independent functioning. In order to keep the body healthy, cells grow and divide to produce more cells as they are needed. However, if cells keep dividing when new cells are not needed and old cells do not die when they should, a mass of tissue forms. This extra mass of tissues called growth or tumor, and can be either benign or malignant (National Cancer Institute, 2002).

Benign tumors are not cancer. They are non-invasive and often not dangerous to health. However, in some cases, benign tumor may undergo further changes and turn into malignant tumor. Therefore, although benign tumors are harmless, they should be monitored regularly or removed.

Malignant tumors are cancer. Cells in malignant tumors are abnormal, and they keep dividing and forming more cells without control (National Cancer Institute, 2002). Cancer cells are dangerous because they can spread by invading and damaging nearby tissues and organs. Occasionally, a secondary cancerous growth can be formed elsewhere in the body by transmission of cancer cells from primary tumor through the blood vessels or lymphatic system. This is called metastasis (National Cancer Institute, 2002).

1.2 Problem Statement

According to Lim, Yahaya and Lim (2003), in the year of 2002, there were total of 26,089 cancers were diagnosed among all residents in Peninsular Malaysia. Among them, 11,815 cases are male and 14,274 cases are female. Lim at el. (2003) also stated that one out of four Malaysians would have the risk of getting cancer in their lifetime, taking into account cancers which are not registered by the National Cancer Registry (NCR) of Malaysia. NCR of Malaysia only records cancer cases which have presented to doctor practicing Western medicine, but not traditional medicine.

Former Health Minister, Dato' Chua Jui Meng (2002b, 2003) mentioned that with better living conditions and improved life expectancy, the proportion of senior citizens above 60 year-old increase from 4.6% in 1957 to 5.8% in 1990 and possibly 11% by 2020; and the increasing prevalence of unhealthy lifestyles associated with affluence, such as smoking, excessive consumption of alcohol, inappropriate diet, obesity and lack of exercise, and environment pollution, have all contributed to the enormity of the cancer problem in these modern times. The approximate cancer incidence in Malaysia is about 150 per 100,000 populations. It means four persons are afflicted with cancer every hour (Dato' Chua, 2002b). In addition, the incidence of cancer is expected to rise further in the years ahead.

In Malaysia, malignant neoplasm (cancer) is ranked as the third leading cause of mortality in the year of 2002 (World Health Organization, 2005). Cancer has always been a dreaded word. It reminds human of their mortality. Dato' Chua Jui Meng (2001) asserted that "cancer is one of the most formidable and most feared of all human afflictions". Further more, cancer can cause severe problems to its patients in terms of physical, financial and psychological distress. However, despite the truth that cancer afflicts 12 million people and causes six million deaths worldwide annually (Dato' Chua, 2001), the fact remains that awareness is the most powerful weapon in fighting cancer.

Statistics discussed above reveal that cancer is a serious disease that everyone should heed. It is also becoming an increasingly important health agenda for every country, including Malaysia. In reality, Ministry of Health Malaysia has launched many campaigns to increase health awareness of local citizens especially in rural areas, such as a series of Healthy Lifestyle Campaigns and Anti-Smoking Campaign. Adoption of a healthy lifestyle such as appropriate diet and regular exercise will reduce the chances of getting cancer. Smoking has been verified as one of the major factors that will increase the chances of developing lung cancer.

Although the government of Malaysia has taken tremendous effort to increase the cancer awareness of Malaysians, however, the results are not obvious. According to Radiology Malaysia (2006), 30-40% of breast cancer patients see their doctor at a later stage of the disease. This shows that majority of women are still lack of breast cancer awareness.

Nonetheless, cancer is a serious disease, however, Malaysia is lacking of local cancer information system that is devoted to help educate the public and increase public awareness. At the early stage of this project, only one community-centric cancer Website developed by Malaysia Cancer Council (MAKNA) was found. This maybe because of the misunderstanding of Malaysians that having a Malaysia-based cancer Website is not a necessity. Instead, they can browse the Websites built in developed countries since they tend to be more “reliable”. Another reason to explain why Malaysia is lacking of local cancer Website is because Malaysians are truly lacking of cancer awareness. Whatever the reasons are, having a Malaysia-based cancer information system is necessary. This is because although some cancer information may be identical regardless of geography, such as symptoms, detection, treatments and clinical staging, but others are only available locally, such as risk factors (including living habits and environmental factors), causes, statistics and cancer institutions. Dato' Chua Jui Meng (2002) mentioned that there are marked geographical variations in the incidence rate and stage of presentation of cancer. For instance, breast cancer, the incidence rate is the highest in North Europe and North America, intermediate in the Mediterranean countries and South America, and lowest in Asia and Africa. Therefore, a local cancer information system is necessary to help Malaysians to have better understanding of local cancer incidence.

Realizing the necessity of cancers awareness towards Malaysian perspective, this project has being specifically focuses on the development of a cancer information system by using information retrieval (IR) techniques. In this regards, this project tends to produce a prototype system to provide user with two types of cancer information, specifically breast cancer and lung cancer, and a category called "general cancer" that discusses the common characteristics regarding cancer disease. Despite traditional hyperlinks method of Website navigation, the system enables user to search for information by entering natural language query. The query will be processed by the information search engine and then return a list of six best match documents that possibly contains the information requested by the user. User can then browse any of these documents for detail information.

Nowadays, it is common that a single Website consists of a few hundreds of Web pages (for example, the Website reviewed in Section 3.3.3). Conventionally, all Websites that are posted on World Wide Web (WWW) are developed using Hypertext Markup Language (HTML). Although HTML does a good job of formatting Web pages for display, but it introduces little interaction between user and computer, in which user can only has limited access. If user wants to get information, user needs to browse from the very general subject category and then level-by-level narrowing down through the hierarchy of subject categories, and hopefully the target information can be found at the end. This method of information searching requires user to spend a lot of time in finding the target information if user has no idea how information within the Website is categorized. This requires strategy, patience, diligence and luck as well. Therefore, many Websites have included a site map that lists all sections of the Website. Normally, this list is in the form of active hyperlink in which user can click on the link and directly browses to a section or Web page from the site map. However, site map is still not efficient and effective enough because it is impossible to cover all contents of a Websites.

As discussed, cancer is a disease that needs to be treated seriously, Malaysia is lacking of community-centric cancer information system and one of the inadequacies of HTML lies in its ability to support fast information searching. This project tries to solve the problems stated above, that is, to develop a local Web-based cancer information system that will help user, especially Malaysians, to search for cancer information. Besides that, IR techniques will be used to enhance the system with full-text searching capability in order to increase the effectiveness of information searching. Keywords used during the searching are highlighted in the retrieved documents in order to help user to locate them quickly.

1.3 Objectives of the Study

The objectives of this project:

1. To develop a prototype of Malaysia-based cancer information system that covers two types of cancer; specifically breast cancer and lung cancer. The other section called "general cancer" discusses the common characteristics of all types of cancer. The system contains Malaysia specific information such as statistics and local cancer institutes; and
2. To enhance the Cancer Information System by integrating it with a full-text searching capability to help user to minimize the search time on retrieving the required information from the system. This search function is developed based on IR techniques and is capable to interact with user by using natural language.

1.4 Scope of Project

This project tends to build a prototype of Cancer Information System by using IR techniques. These IR techniques include tokenization, stopwords removal, stemming, document indexing, and document matching (will be discussed in Chapter Five). There are more than 200 types of cancer that can be found in the world today. As a matter of fact, it is impossible to build a system that includes all types of cancer due to the time constraint in this project. Therefore, the system covers only two types of cancers only; namely breast cancer and lung cancer. This is because according to Lim at el. (2003), lung cancer is the most common cancer among males of all three major ethnic groups in Peninsular Malaysia and the most frequent cancer for females for all ethnic groups is breast cancer. The scope of "general cancer" in this project provides introductory on common characteristics shared by majority of cancers. For instance, the common treatments for any type of cancer are surgical, radiotherapy, chemotherapy and biotherapy. Therefore, all of this information can be found under the category "*General Cancer → Treatments". However, lung cancer has a specific treatment called Photodynamic Therapy (PDT). Therefore, information regarding PDT can only be found under the category "*Lung Cancer → Treatments*", and as well as other treatments which are applicable to lung cancer.

The system enables user to search information in two different ways. The first one is through typical HTML hyperlinks while the second part concerns on the information search engine, which is developed by using IR techniques. The information search engine is designed to retrieve six most relevant documents in response to user's query. User can select and browse the documents that probably contain the targeted information.

The targeted user for this system is any person, especially Malaysians, who wishes to learn about the cancer disease, or to find out local cancer institutes that provide services in fighting the disease.

The system will be a client-server model where user can access the system through local host and Local Area Network (LAN). Access through Internet is possible if the system is hosted by a Web server.

1.5 Significance of the Study

This project aims to build a cancer information system by using IR techniques. It serves as a pilot study for this type of application. The preliminary model of search function will be tested for its efficiency in retrieving relevant documents. This information serves as a basis for possible future enhancement or development of a new search function that is more advanced and effective. Besides that, this information system also can either be modified into other medical system, or expanded to provide other health information.

The project tries to increase awareness of local community regarding cancers by leveraging on contemporary Information and Communication Technology (ICT) approach. The system has higher degree of accessibility if compared with the traditional way of information access. Actually, most cancers have high possibilities to be cured if patient receives treatments in early stage. Therefore, it is important to increase public awareness by implementing this system for the use of the public.

Furthermore, this project is well-suited to former Minister of Energy, Communications and Multimedia, Datuk Amar Leo Moggie's suggestion. Datuk Amar Leo Moggie (2003) believed that Malaysians are neither shorts in ideas nor capabilities to use ICT to develop homegrown cancer information system that serves Malaysians' interests. He further mentioned that now is the right time for Malaysian to have more usable Websites that are devoted to public education so as to encourage more of community-centric efforts. This in return will benefit a wide community of Malaysia user.

1.6 Outline of Research Report

This research report is organized into following chapters:

Chapter One: Introduction

This chapter provides the requirement specification and introduction to the issues which are related to this project. It contains the project background, problem statement, the objectives of the study, the scope of project, significance of the study and an outline of research report.

Chapter Two: Review of Information Retrieval

This chapter introduces and discusses several topics related to this project, such as Natural Language Processing (NLP) and IR. Some introduction to the cancer disease is also included. Reviews of several existing IR systems are also discussed.

Chapter Three: Methodology

This chapter provides an overall process used to develop the proposed system. The Waterfall Life Cycle model is selected and followed during the development of this project. Reviews of cancer information Websites analyze the pros and cons of current practices that are used to develop information system. Besides that, methodology that was used to conduct questionnaire is also described in the chapter.

Chapter Four: System Analysis

This chapter discusses about the analysis of the results collected from questionnaire. Besides that, the system analysis states the functional and non-functional requirements of the proposed system.

Chapter Five: System Design

This chapter emphasizes all the detail design of the proposed system. It focuses on how the proposed system is designed. Each of the development steps in the proposed system is explained in this chapter.

Chapter Six: System Implementation

This chapter reports the implementation of the proposed system. It aims to discuss the technologies used to implement the system and the algorithms used to program the proposed system.

Chapter Seven: System Testing and Evaluation

This chapter reports different techniques used to test the proposed system. It includes, unit testing, module testing and system testing. Evaluations of the system are also discussed in this chapter.

Chapter Eight: Conclusion

This chapter summarizes the achievements of this project. The system limitations and the recommendation to enhance the system for the future purpose are discussed in this chapter.

1.8 Summary

This chapter is an introductory, which attempts to put the project in perspective. This includes in giving the background, the objectives of developing the project, scope and significance of the project.

Chapter Two: Overview of Information Retrieval and Cancer Disease

2.1 Introduction

This chapter reviews the technologies and concepts related to the project. The main area includes Natural Language Processing (NLP) and information retrieval (IR). A general architecture of IR system will be discussed in details. Then, several existing IR systems are reviewed to identify current "state-of-the-art" technologies that been used in building such system. Besides that, some introductory to the cancer disease is also included.

2.2 Introduction to Natural Language Processing

The last decade has been one of dramatic progress in the field of NLP. As a demand for Human-Computer Interaction (HCI) and information, NLP has found itself at the center of an information revolution ushered in by the Internet era. The major advantage of NLP, as asserted by InfoLab Group (2005), it was believed to be the most natural form of communication and information access for human, rather than constrained user to communicate in a predetermined manner. By implementing NLP, the communication between user and computer will become more interactive. User is free to ask or enquire the system in natural language and the system would response appropriately.

NLP is also known as a language technology, linguistic engineering and computational linguistics, which is one of the application areas of Artificial Intelligence (AI). Other AI application areas include expert systems, machine learning, artificial neural network, automated reasoning and theorem proving, and so on (Luger, 2002). NLP aims to study and develop methods by which natural human languages can be interpreted effectively by computer. These are done by defining languages patterns and describing them to a computer in order to teach a machine to imitate human language and understandings.

NLP has several application areas. While early attempts focused on machine translation, NLP has played a significant role in several other areas such as natural language interfaces, grammatical and stylistic analysis, document processing and information retrieval, and computer aided language learning (Volk, 2003). These applications can be further classified into two major classes: text-based applications and dialogue-based applications (Allen, 1995). Text-based applications are related to the processing of written text and cover reading-based tasks such as (Allen, 1995):

- documents processing,
- language translation,
- information extraction and information retrieval,
- text summarization,
- story understanding and question answering, and
- grammar and style checking.

Dialogue-based applications involve human-machine interaction and this interaction is done either through spoken language or keyboards. Examples of application include (Allen, 1995):

- database front end (natural language interfaces for databases),
- automated customer service over telephone,
- speech recognition system, and
- tutoring system.

A very early and popular natural language system is the ELIZA program that was developed by MIT in the mid-1960s (Allen, 1995). This is a dialogue-based application in which interaction is done through keyboard. Although the program seems behave intelligently in conversation within patient-therapist situation, it posses none natural language understanding. Instead, it responses to user's input based on a set of well-defined pattern matching rules. Each of these rules contain a keyword, a pattern to be match against user's input, one or more associate response options and a priority score to trigger the respective rule. This application reveals the fact that a language recognition system is not necessary able to understand language. Another tangible example is a video cassette recorder that uses speech recognition as commands to control it, but it does not understand the human language specifically.

On the other hand, one of the earliest NLP systems is the SHRDLU program that was developed by Terry Winograd (Luger, 2002). It is a natural language application that could understand simple configuration of blocks of different shapes and colors. This program was successful because it has specialized problem domain and was said to focus on "microworld". Hence, it is worth to note again that natural language system (ELIZA) is different from language understanding system (SHRDLU) in the sense that the former does not understand language.

Language is a complicated and complex phenomenon. To facilitate research, linguists have defined at least seven levels of language analysis which spoken language and text can be extracted. Each of this level contains a number of language processing techniques. However, not all NLP systems use all seven levels (Allen, 1995). These levels are (Allen, 1995):

- *phonetic or phonological level* – study of the speech sounds,
- *morphological level* – study of the structure and form of words from basic meaning units called morpheme,
- *syntactic level* – study of the rules whereby words or other elements of sentence structure are combined to form grammatical sentences,
- *semantic level* – study of the dictionary meaning of words, phases, and sentences, and also for the meaning they derive from the context of the sentence,
- *pragmatic or practical level* – study of the ways that the setting of the sentence in a discourse is used to determine its correct interpretation and its effects on the listener,
- *discourse level* – concerns the effects of preceding sentences to the following sentence, and
- *world level* – concerns knowledge of the physical world, the world of human social interaction, and the role of goals and intentions in communication.

2.2.1 What is Information Retrieval?

Before the introduction of World Wide Web (WWW) in the beginning of 1990s, IR was seen as a narrow area of interest mainly to legal database and searching library catalogues (Baeza-Yates and Ribeiro-Neto, 1999b). It was based on having an intermediary searches on behalf of user. However, in the wake of the explosive growth of online text since the introduction of WWW, a number of researchers in language processing started to develop various IR systems to help user cope with the information explosion. IR is the current practice that answers to the huge amount of online text. Without the help of IR systems (search engines), search information through Internet seems impossible.

What is IR? IR is related to automated manipulation of unstructured natural language text. It concerns the representation, storage, organization and assessing of various information items such as newsletters, journal articles, all types of documents and so on (Salton and McGill, 1983).

The immense increase in the amount of online text available reinforced the view that IR is synonymous with document retrieval, as in the definition (Strzalkowski, Perez-Carballo and Marinescu, 1996):

"A typical (full-text) information retrieval task is to select documents form a database in response to a user's query, and rank these documents according to relevance".

According to Hirschman and Gaizauskas (2001):

"Information retrieval, which, following convention, we take to be the retrieval of relevant documents in response to a user query".

Hence, based on the two definitions above, the most common example of IR system is Web search engines.

However, IR is far broaden than merely just a document retrieval. Allan et al. (2002) mentioned that IR is an open field that encompasses many types of information access or information seeking tasks. These types of information access are:

- retrieval models,
- cross-lingual information retrieval,
- Web search,
- user modeling,
- filtering, topic detection and tracking (TDT), and classification,

- summarization,
- question answering,
- metasearch and distributed retrieval,
- multimedia retrieval, and
- information extraction.

By this assertion, the typical IR definition is just one part of the services provided by Web search engines.

According to Ramakrishnan and Gehrke (2003), IR is a research field separated from databases. Table 2.1 summarizes the comparison between IR and database. Typically, IR deals with plain and unformatted data, and therefore there is no organized structure and the semantics are imprecise. As contrast, database requires every record or object to be specified clearly, or in explicit form. That is, information in database exists in the form of specific data elements stored in tables. Every data entry in a *table* is called *record*, which is in terms divided into several *fields* that contains specific and unique attribute identifying the respective *record.* For instance, every employee exists as a *record* in the *table* called Employee and each *record* must has certain attributes (*fields*) such *first_ name*, *last_name*, *data_of_birth*, *telephone_number* and *employee_ID*. All data in these *fields* must has certain standardized format, for example, *data_of_birth* may has the *ddmmyyyy* (day-month-year) format. Database uses Structured Query Language (SQL) to retrieve data, such as *SELECT (field name) FROM (table) WHERE (criteria)*. Information need in database applications can always be mapped precisely into a query formulation (such as the SQL above) and there is a precise definition of which elements of the database constitute the answer. In IR, precise information requests are quite difficult and both relevant and irrelevant information may be retrieved.

In addition, database is more suitable for applications that the data is more likely to be changed very frequently.

Table 2.1: Comparison between IR and database
(Ramakrishnan and Gehrke, 2003)

Information Retrieval	**Database**
Imprecise semantics	Precise semantics
Keyword search	Structured Query Language (SQL)
Unstructured data format	Structured data format
Frequently read documents. Add documents occasionally.	Frequently change data (create new, update, and/or delete data)
Retrieve top *k* results	Generate full answer

2.2.1.1 General Architecture of Information Retrieval System

Generally, every IR system consists of the following elements:

- A set of information items, such as documents, journals, newsletters and so on;
- An indexing process that creates the representation of information items;
- A set of requests triggered by user's information need; and
- Some mechanism that perform the matching between representation of requests and representation of information items.

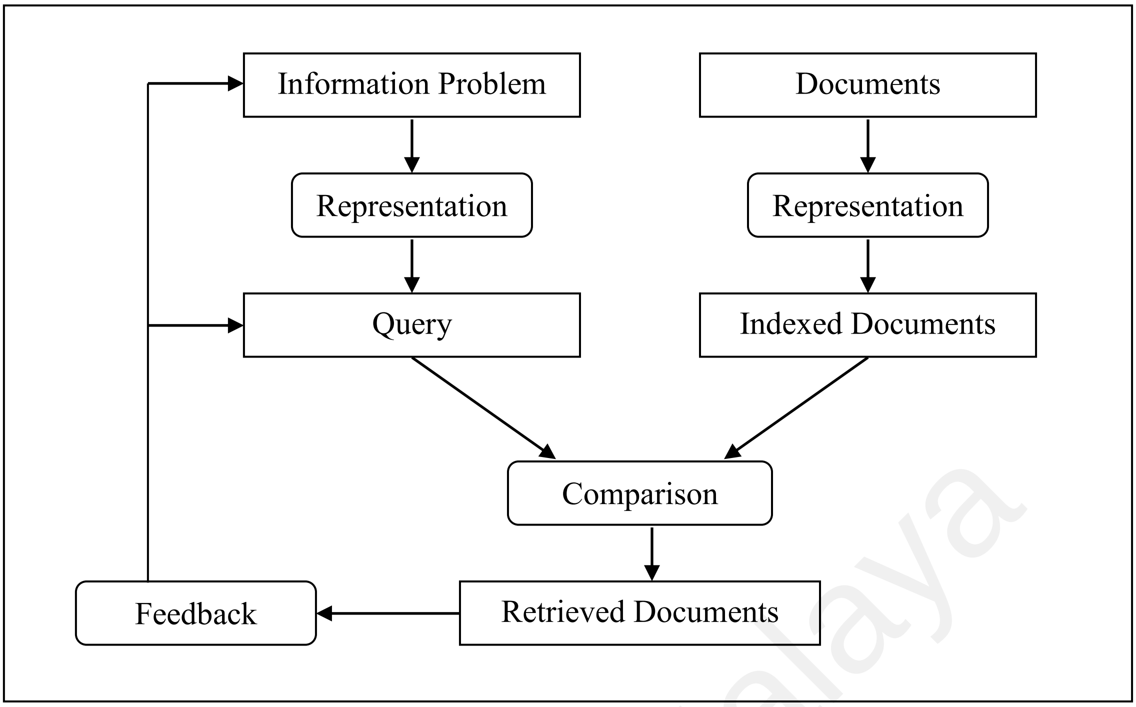


Figure 2.1: General architecture of an IR system (Croft, 1993)

Figure 2.1 shows the general architecture of an IR system. Before an IR system is ready to deploy, it needs to index the documents. The process is labeled as *Representation* (representation of documents) in the Figure 2.1 and it is refers to indexing process. Indexing is a process of assigning appropriate terms and identifiers capable of representing the contents of the documents, normally by selecting the terms from the respective documents. Some indexing process may include the full contents of the documents, but others may involve only title and abstract. Indexing will create an output called index. In the case of IR, index is an alphabetized list of words that gives the page or pages on which each word is located in the documents.

The representation of information problem (also labeled as *Representation* in Figure 2.1) refers to the automated query formulation process and produces system query as result. Generally, IR system has to convert user's query into system query that can be understood by the retrieval system. The *Comparison* between the system query and index is also called the matching process. As result, a ranked list of relevant documents is created and it is basically sorted by descending order of relevance.

Certain IR systems consider feedback from user as one of the factor during document matching process. System that supports feedback allows user to feed back the relevance or non-relevance of a document based on user's judgment. IR system can then make use of the information to adjust the document ranking. For instance, some Web search engines consider the number of clicks on retrieved document as one of the criteria to rank its relevance. A document that is clicked or browsed frequently is more relevance to those documents that are seldom being browsed by user.

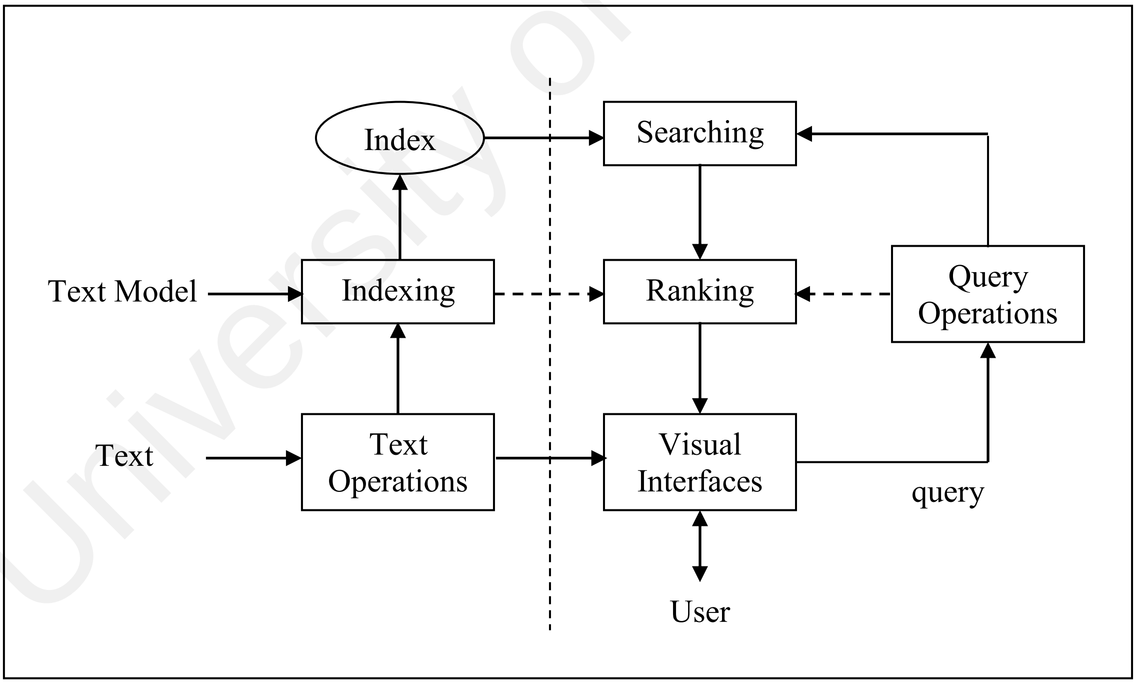


Figure 2.2: The process of retrieving information
(Baeza-Yates and Ribeiro-Neto, 1999)

Figure 2.2 introduces IR system from the perspective of retrieval process. The overall processes are much similar to the one as described during Figure 2.1. An IR system will receive natural language word(s), phrase(s), or sentence(s) from user through the *Visual Interfaces* of the system. Next, the system generates a query from what it received through *Query* Operations. The query is then compared with the index. Finally, the system ranks the retrieved documents by their estimated degree of similarity with the query and presented the results to the user. These documents are called relevant documents. A perfect retrieval system will only retrieve the relevant documents. However, it is impossible to build a perfect retrieval because relevance is a subjective issue of user.

There is an idea worth to be pinpointed. IR system should parsed and transformed user's query by the same text operations applied to the text collection (Baeza-Yates and Ribeiro-Neto, 1999). This ensures constant representations between user's information need and the contents from text collection, and hence searchable. Therefore, both *Text Operations* and *Query Operations* in Figure 2.2 are exactly performing the same series of functions. The same situation is applied to both of the *Representation* of information problem and *Representation* of documents in Figure 2.1.

2.2.1.2 Major Models of Information Retrieval

Generally, IR models can be classified into three main categories: which is the Boolean model, the statistical model or ranked model, and the probabilistic language model (Spoerri, 1995). The Boolean model is often referred to as the "exact match" model and the statistical model as the "best match" models.

There are two metrics that are commonly used to evaluate the effectiveness of a retrieval method, which are *precision rate* and *recall rate*. The former examines the percentage of first *n* retrieved documents that are relevant to user's query, while the latter examines the percentage of relevant documents among first *n* retrieved documents out of all relevant documents in the documents collection (Voorhees, 1999). Precision and recall have inverse relationship. To increase precision, user has to narrow down their query by using only the specific keywords. This ensures that only a few relevant documents are to be retrieved (high precision), but lower the recall rate. To increase recall, user has to broaden their query, for instance, by using synonyms of the keywords. This will cause much more documents to be retrieved and hence increase recall, but compromise precision since not all documents that contain the synonyms are relevant.

(i) Boolean Model

The Boolean model is the simplest model of IR. It uses a precise language for building query expression through three basic logical operators, namely AND (logical product), OR (logical sum) and NOT (logical difference), which are used to link terms in a query (Chowdhury, 1998).

Boolean model represents documents as sets of keywords or terms (Baeza-Yates and Ribeiro-Neto, 1999). These keywords are then matched with the keywords in user's query. Let's the *i*-th term in the documents collection is called t_i and *n* is the number of distinct keywords in the documents collection, a document is represented by:

$$t_1 \wedge t_2 \wedge t_3 \wedge \ldots \wedge t_n$$

If t_2 does not present in a particular document, for instance, the document is represented by:

$$t_1 \wedge \neg t_2 \wedge t_3 \wedge \ldots \wedge t_n$$

Boolean model has the advantages of giving professional user a sense of control over the retrieval system through the logical operators, easy to be implemented and computationally efficient. However, naive or non-professional user may have difficulties on using such system because of the difference between the Boolean AND and OR with the natural language words ‘and’ and ‘or’ (Hiemstra, 2000). Besides that, Boolean model does not rank the retrieved documents and exact matching may lead to too few or too many retrieved documents (Harabagiu et al., 2001).

(ii) Statistical Model

On the other hand, statistical or ranked model comprises of many models such as vector space model, probabilistic model, fuzzy set model, Bayesian network models and so on. Many commercial IR systems were developed based on statistical models because such models provide a ranking of retrieved documents and they support automatic query generation. Automatic query generation enables normal user or non-professional user to enter a real natural language request and the system will generate system query itself.

Among all statistical models, the two most popular IR models in practice are vector space model and probabilistic model (Park, Ramamohanarao and Palaniswami, 2005).

(a) Vector Space Model

The vector space model represents the documents and queries as vectors in a multidimensional space, where the number of dimensional is determined by the number of unique terms in the documents collection:

$$D_i = (w_{i1}, w_{i2}, ..., w_{it})$$

where D_i represents a document text and w_{it} is the weight of term T_k (obtained from user's query) in document D_i (Buckley et al., 1995). The scores for document D_i is in the vector form of weights w_i of all terms T_i. For terms that are absent form a particular document, a weight of zero (0) is used for that particular term in the particular document, and positive weights are assigned for terms that are present. In binary vector space model, weight of one (1) is given to a term if it appears in the document and zero (0) if it does not appear in the document. This is called binary term weights. However, it has two disadvantages: binary term weights do not reflect the frequency of a term and all terms are considered equally importance. Therefore, later vector space models weight terms are by importance. High-frequency terms that occur in many documents are given low weights and terms that are important in particular documents but unimportant in the remainder of the collection are given high weights. A document that contains only few terms can be assigned high ranking if these terms occur infrequently in the collection but frequently in the document.

Vector space model ranks the documents based on the degree of similarity between index representations (that is, documents) and query terms, which is based on the cosine of the angle between them (Maron and Kuhns, 1960). Given a query vector q and document vector d_j that are represented as t-dimensional vector, the similarity between q and d is defined as the inner product of q and d (Baeza-Yates and Ribeiro-Neto, 1999b):

$$\text{sim}(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{\left|\vec{d}_j\right| \times \left|\vec{q}\right|} \qquad \text{(Equation 2.1)}$$

Since longer documents are more likely to be relevant as they have higher term frequencies and contain more terms, therefore vector space model does include document length normalization during computing the similarity score ($\left|\vec{d}_j\right|$ in the

Equation 2.1). By using document length normalization, if two documents have the same score, the shorter document will be ranked more relevant because it is more focused on the information need. Normalization of query vector ($|\vec{q}|$ in the Equation 2.1) does not affect the ranking of documents because the query is same for all documents.

The advantages of vector space model are it ranks documents according to relevance and terms are weighted by importance. The disadvantages are terms of documents and queries are assumed independent, and although weighing is intuitive, but not very formal.

(b) Probabilistic Model

The probabilistic model is based on Probability Ranking Principle that suggests that the best possible effectiveness of retrieval system can be achieved by ranking documents in descending order of relevance probability in respond to user's query (Robertson, 1977). The most common probabilistic model assumes that the query terms are distributed differently in relevant and non-relevant documents, with probability formula is usually derived from Bayes' theorem (Spoerri, 1995).

In probabilistic model, given a query and some documents, one of the following two situations may happen (Sparck Jones, Walker and Robertson, 1998): either a document is relevant to the query, or it is irrelevant. If the probability of belonging to relevant is higher than probability of belonging to non-relevant, then the document will be retrieved.

(iii) Statistical Language Model

The statistical language model is relatively new research area compared to both Boolean and statistical models. Researchers started to apply language model in IR system since the late of 1990s. Two of the fundamental language models are the basic retrieval model that defines the system's matching process, and the extension of the basic model, the statistical translation retrieval modal that also include query formulation process in addition to matching process (Hiemstra, 2000).

Statistical language model is probability distribution defined on sequences of words. It treats each document as a language sample and a query as a generation process (Song and Croft, 1999). It ranks documents based on the probability that the document's language model would generate the terms of the query from user.

Table 2.2 summarizes retrieval methods of the three discussed IR models for four different linguistic levels.

Table 2.2: Comparison of methods for three IR models
(Spoerri, 1995)

Linguistic Level	**Boolean Model**	**Statistical Model**	**Language Model**
Lexical	Stopword list	Stopword list	Lexicon
Morphological	Truncation symbol	Stemming	Morphological analysis
Syntactic	proximity operators	Statistical phrases	Grammatical phrases
Semantic	Thesaurus	Clusters of co-occurring words	Network of words/phrases in semantic relationships

According to Allen (1995), Volk (2003) and Liu (2003), IR or document retrieval is an application of NLP. Lewis and Jones (1996) regarded "natural language" as taking indexing terms from the document itself. NLP allows non-professional user to type in a query using their daily speaking language (that is, natural language, avoiding them to memorize Boolean or other predetermined query languages) (Search Tools Consulting, 2002). While the simplest processing just removes stopwords and uses statistical approaches to rank documents, more sophisticated systems may use linguistic analysis.

By far, research shows that simple strategies such as statistical document retrieval approaches perform natural language indexing and searching are effective to an acceptable degree (Lewis and Jones, 1996), and many researchers are currently developing more sophisticated IR systems that implement language model to enhance the results of searching. Nevertheless, Liu (2003) revealed that although IR systems focused on processing natural language text, but only a few of them utilized NLP techniques. In fact, majority systems used statistical approaches for performing NLP tasks. In addition, Voorhees (1999) asserted that "information retrieval can be viewed as a great success story for NLP", and surprisingly, techniques that treat text as little more than a bag-of-words outperformed more sophisticated linguistic processing techniques. Smeaton (1995) suggested that this was because NLP techniques were originally developed for tasks such as machine translation, which are fundamentally different to IR tasks.

Studies above show that statistical approaches are ideal model to perform retrieval tasks. Eventually, vector space model was chosen as the retrieval model in this project because:

- it weights terms according to the importance of the terms. A term that appears in most of the documents in the documents collection is not important because it does not represent anything about which documents the user might be interested. Therefore, the term will be assigned lower weight. On the other hand, a term that appears only in a few documents is given higher weight because it can effectively represent the contents of the documents. Term weighting can substantially increases the retrieval performance of the proposed Cancer Information System;

- the cosine ranking formula ranks the documents according to the degree of similarity between documents and query in descending order. This helps the users of the Cancer Information System locate the relevant documents quickly.

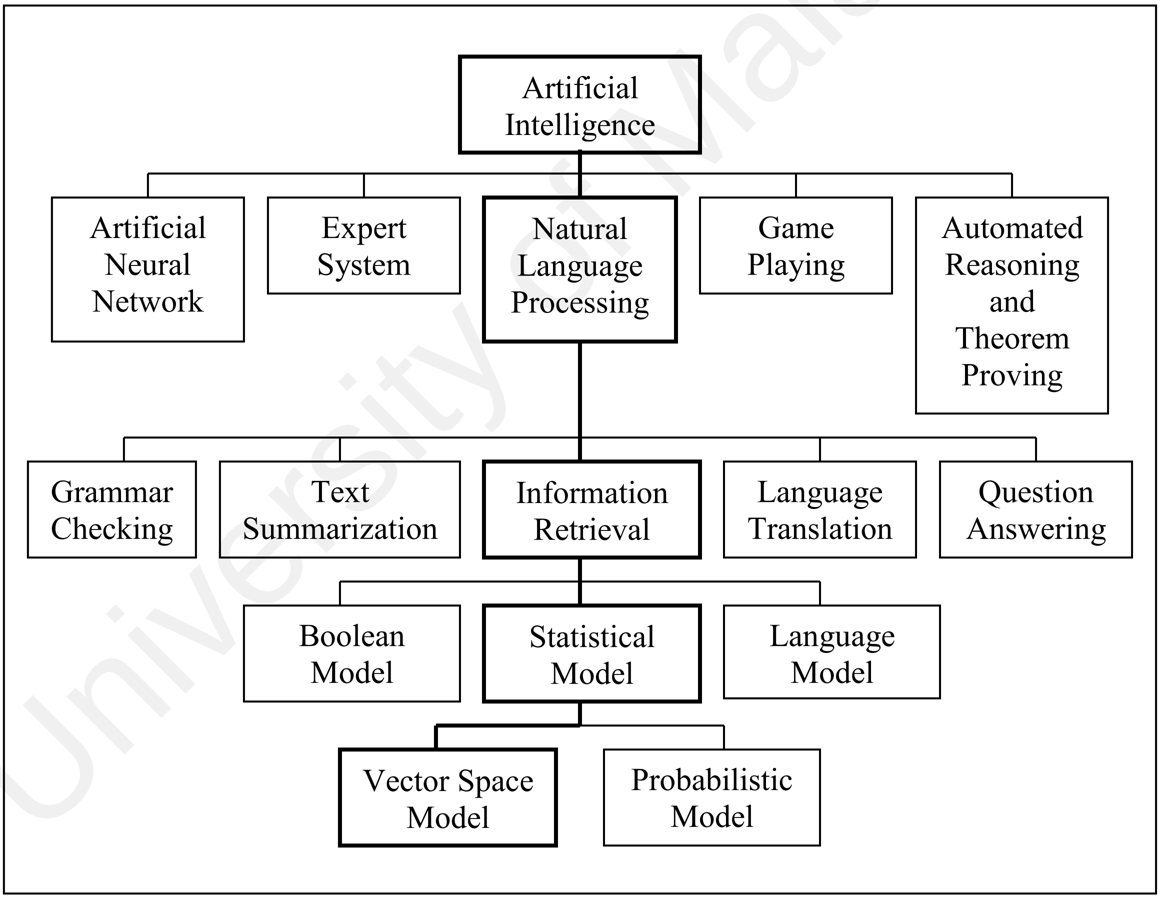


Figure 2.3: Project research field and approach
(Allen, 1995; Volk, 2003; Liu, 2003; Baeza-Yates and Ribeiro-Neto, 1999; Voorhees, 1999)

2.2.2 Review of Information Retrieval Systems

This section reviews an IR engine, specifically SMART and three IR systems, namely Oré, Wordkeys and IR-n. The difference between IR engine and IR system is that IR system is a fully developed functional system with specific domain, while IR engine concerns the development of retrieval models. For instance, many different IR systems that participated in TREC (Text REtrieval Conference) used SMART as their model retrieval engine to perform indexing, query processing and document ranking. TREC is a series of evaluation conferences that aims to encourage research in IR by using large text collection.

(i) SMART Information Retrieval Engine

SMART is one of the earliest researches in the field of IR. It has undergone more than 30 years of development at Cornell University (Buckley et al., 1995). The purpose of the project is to investigate the effectiveness and efficiency of automatic methods of retrieval of large collections of text. The retrieval process starts from an arbitrary piece of natural language text from the user and matching against automatically indexed documents. SMART is an IR engine rather than a ready-to-use IR system. It can be implemented on a number of text databases, ranging from newsletters, textbooks, magazine articles, dictionaries, encyclopedias and so on.

Query Processing and Automatic Indexing

In SMART, user's query and text from documents are processed to derive terms that will be subsequently used in the matching process. Query processing and indexing in SMART comprises of the following steps (Buckley et al., 1995):

1. identify individual text words,;
2. use a stopword list to filter unwanted function words;
3. generate word stems by performing suffix removal;
4. optionally use term grouping methods that based on statistical word adjacency computations to form term phrases; and
5. assign term weights to all remaining word and/or phrase stems to form the term vector.

After finished the steps stated above, term vector manipulations are used to perform the ranking and retrieving.

Scoring Function in SMART

SMART uses a well-known weighting system that is known as *tf.idf* (term frequency times inverse document frequency). This weighting system assigns weight w_{ik} to term T_k in query Q_i based to the frequency of occurrence of the term in Q_i and in inverse proportion to the number of documents which contain the term (Buckley and Salton, 1988). A high weight in *tf.idf* is achieved by a high term frequency in the given document and a low document frequency of the term in the whole collection of documents.

Buckley, Allan and Salton (1994) suggested using the following formulae during the TREC-2:

cosine normalization: $$score(\vec{d},\vec{q}) = \frac{\sum_{k=1}^{m} d_k . q_k}{\sqrt{\sum_{k=1}^{m} (d_k)^2} . \sqrt{\sum_{k=1}^{m} (q_k)^2}}$$

document term weight: $d_k = 1 + \log(tf)$

query term weight: $q_k = (1 + \log(tf)).\log \frac{N+1}{df}$

where tf = the frequency of term t_k in the document (the number of times the term t_k appears in a document divided by the number of total words in the document) .

df = the number of documents a term t_k appears.

N = number of documents in the collection.

m = *f(df)* (frequency of document frequency). For instance, if term t_k appears in total of 5 documents, then *f(df)* = 5.

(ii) Oré Information Retrieval System

Oré is an IR system that participated in the robust track during TREC 2003. Robust retrieval aims to improve the consistency of retrieval technology in the system's least effective topics (Voorhees, 2004). The system participated in five runs as shown in Table 2.3.

Table 2.3: Summary of Oré's Runs in TREC 2003
(Iljin, Brand, Driessen and Klok, 2003)

Run	Ranking Model	Topic's Tags Used	Expansion of Query Terms
1	BM25	Title + Description	Yes
2	BM25	Title + Description	No
3	BM25	Description	No
4	Probabilistic	Title + Description	Yes
5	Probabilistic	Title + Description	No

From Table 2.3, Oré implements two ranking models, specifically BM25 and probabilistic model. BM25 was the ranking model used by Okapi IR engine since TREC-3 in 1994 (Robertson and Walker, 1999), while the probabilistic model was developed by the same team who built the Oré system. Besides that, the system also being tested with other conditions such as which topic's tags were being used to construct the query, and with and without query expansion. The system evaluations showed that the first run in Table 2.3 has the highest precision and BM25 model outperformed probabilistic model. The following sections discuss the query processing and BM25 model.

Query Construction

As shown in Table 2.3, query is constructed from the *title* and *description* automatically by dividing on non-alphanumerical characters. The third run in the table is used as guideline, which only *description* was used. After that, every single character was discarded and all remaining term were converted to lower case. For those runs with query expansion, a dictionary-based stemming that called Knowledge Concept's Content Enabler semantic network was used to get the root form of query terms (Iljin et al., 2003). Then, the root forms were expanded with morphological variants based on the semantic network. In addition, a 'similarity' measure was used to expand the query terms with synonyms. Query's terms that existed in both *topic* and *description* tags were put together without duplicate removal.

Indexing

The index for Oré IR system was built by splitting documents text on non-alphanumerical characters and then single characters were removed from the index. While most IR systems omit stopwords during indexing, index in Oré system was built with stopwords were left in. Iljin et al. (2003) asserted that it was very difficult to construct a universal set of stopwords.

Ranking Model

As stated above, BM25 model outperformed the probabilistic model during TREC evaluations. Therefore, only BM25 model is discussed, as shown below (Iljin et al., 2003):

term frequency: $$tf(q_i, d) = \sum_j tf(q_{i,j}, d)$$

document frequency: $$df(q_i) = \left| \bigcup_j \text{set of documents in which } q_{i,j} \text{ occurs} \right|$$

where q_i = the *i*-th query term in query *q*

$q_{i,j}$ = the *j*-th expansion of q_i

$tf(q_{i,j},d)$ = the term frequency of expansion term $q_{i,j}$

Finally, for a document *d* and query *q*, the score is computed as follow (Iljin et al., 2003):

$$\text{Rel}(d,q) = \sum_{q_i \in q} \frac{\log(N) - \log(df(q_i)).tf(q_i,d).(k_1+1)}{k_1.((1-b)+(b.ndl(d))) + tf(q_i,d)}$$

where N = number of all documents.

$ndl(d)$ = the length of the document *d* divided by the average document length.

k_1 = parameter.

b = parameter.

(iii) Wordkeys Augmentative and Alternative Communication System

Wordkeys is a full-text retrieval system that aimed to provide communication aids for non-speaking people (Langer and Hickey, 1997). The system allowed user to retrieve short text based on a set of per-stored messages and a selection procedure. Besides that, the system also allowed user to type in new messages and the system will automatically indexed the messages and integrated them in the system's database.

Although many retrieval techniques were developed in the past few decades, however, Langer and Hickey (1997) argued that those techniques were designed for larger collection of documents. Therefore, those techniques have to be modified considerably to be applied to the full-text retrieval of short messages, such as the Wordkeys system that used by augmentative and alternative communication (AAC) user that typically contain less than 20 words.

Architecture of Wordkeys System

Figure 2.4 shows that architecture of the Wordkeys system. The following sections discuss the processes of Wordkeys in detail.

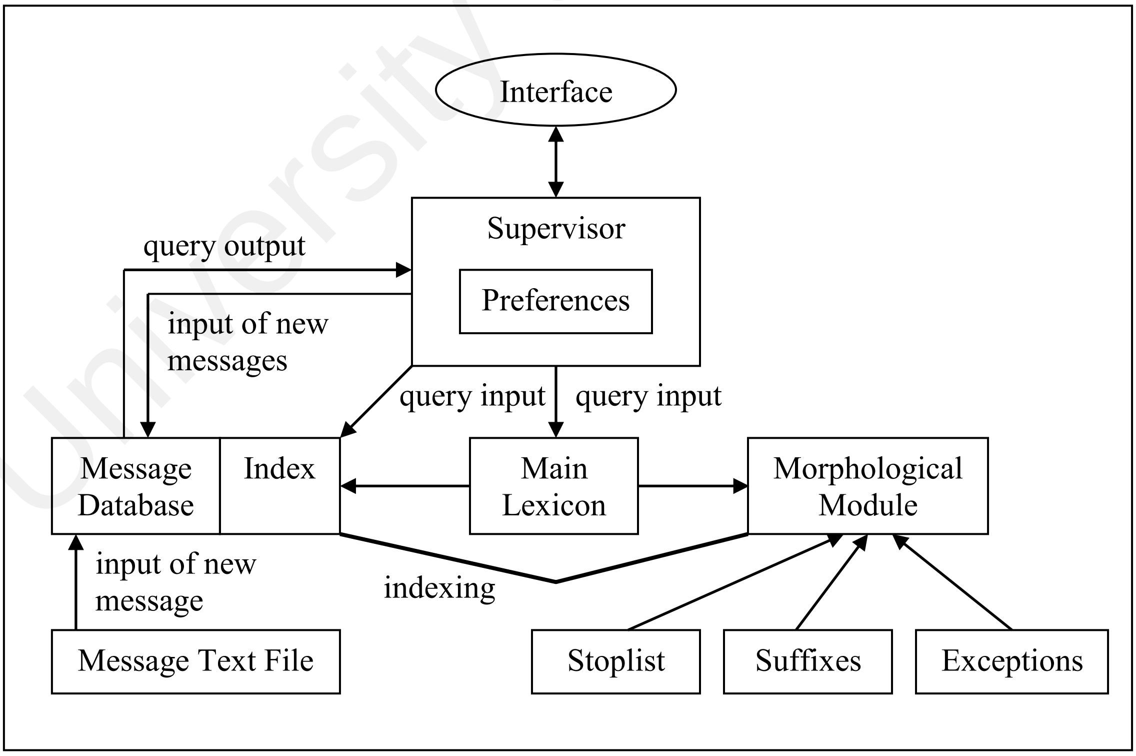


Figure 2.4: Architecture of the Wordkeys system (Langer and Hickey, 1997)

Indexing

Several actions are performed when a message was posed to the Wordkeys system (Langer and Hickey, 1997):

- Tokenization: spitted the text into individual words.
- Morphological analysis: overall, this step identified lemmas and roots of words to determine their syntactic category (will be discussed in more detail in the following section).
- Synonyms of resulting words were looked in the semantic lexicon and added to the list of index words.
- Added the new message and its index to the database.

Morphological Analysis

Morphological analysis is the second step of indexing in Wordkeys. Wordkeys implements a custom programmed morphological module that partially based on the WordNet morphological information. The analysis of a word form comprises of two steps (Langer and Hickey, 1997):

1. Lemmatization. At first, affix is removed from a word and the un-affixed form is checked for syntactic category changed after affix removal. If the form is found in the lexicon, it is accepted as a lemma and added to the message index.
2. After that, stemming algorithms are used to perform inflectional and derivational analysis on the lemmatized word forms. Sometimes, semantic relations between derived words and their root differ considerably. Therefore, information from the morphological analysis is also used to determine the morpho-syntactic categories of word forms and lemmas.

Message Ranking

After the system received keywords from user, the search process is triggered. Searching in Wordkeys comprises of the following processes (Langer and Hickey, 1997):

- Tokenization: keywords are separated into word forms.
- Lemmatization: word forms are analyzed so that could be searched in the lexicon.
- The system searches the message index for the word forms and lemmas. Message numbers are added to the list of search results if a successful match occurred.
- The lemmas are looked up in the semantic lexicon to retrieve related words. If a relation is found, the related words will be applied for all subsequent searches.

After the matching, relevant messages are displayed in the descending order of relevance. As argued by Langer and Hickey (1997), they developed a ranking model that ranked document according to the criterion of semantic distance between keyword and index word. In the beginning, the semantic distance is zero and increased as go down the following list (Langer and Hickey, 1997):

- same word form (semantic distance = 0);
- different word form from the keyword form (cars – car);
- other derivation of the root of the keyword (investigation – investigate);
- synonyms of the keyword (car – automobile);
- other related words that depends on semantic relation.

The above ranking criteria are applied to the case of single keyword. In the case of multiple keywords, a combination of the semantic distances for different keywords is used for ranking and the most relevant message was the message that related to more keywords.

The following example illustrates the message ranking algorithm (Langer and Hickey, 1997). In the example, keyword *swim* is used. The results of retrieval in descending order are as followings:

1. Would you like to go for a swim?
2. Normally I don't like swimming, but this Sunday it was so hot that I spent the whole day on the beach and in the water.
3. I'm not a very good swimmer.
4. Shall we go for a dip?

The first result contains the exact match. The second result contains another word form of the same lemma, which has less semantic distance than derivational analysis. The third message contains a derivation of the keyword and the forth message contains a synonym (*dip*) of the keyword *swim*.

(iv) IR-n Passage Retrieval System

The IR-n passage retrieval system was developed at University of Alicante and participated in the monolingual (Spanish) and bilingual (Spanish-English) tasks at CLEF-2001. The system provides retrieval based on the selection of variable size of passages as basic unit of information (Llopis and Viceso, 2001). Passage retrieval allows the system to retrieve part of document that is more relevant to the query. The system generates passages in overlapping way. For instance, if paragraph of N was selected, the first paragraph will be formed from sentence 1 to N and the second from 2 to N+1. In other words, paragraph or passage is a number of continuous sentences of the text.

Architecture of IR-n

IR-n consists of three modules, namely Indexing module (creates indexes for the documents collection), Question module (processes and expands the query, and translates query if applicable) and Retrieval module (ranks documents according to the similarity of query and documents) (Llopis and Viceso, 2001). Figure 2.5 shows the system architecture of IR-n.

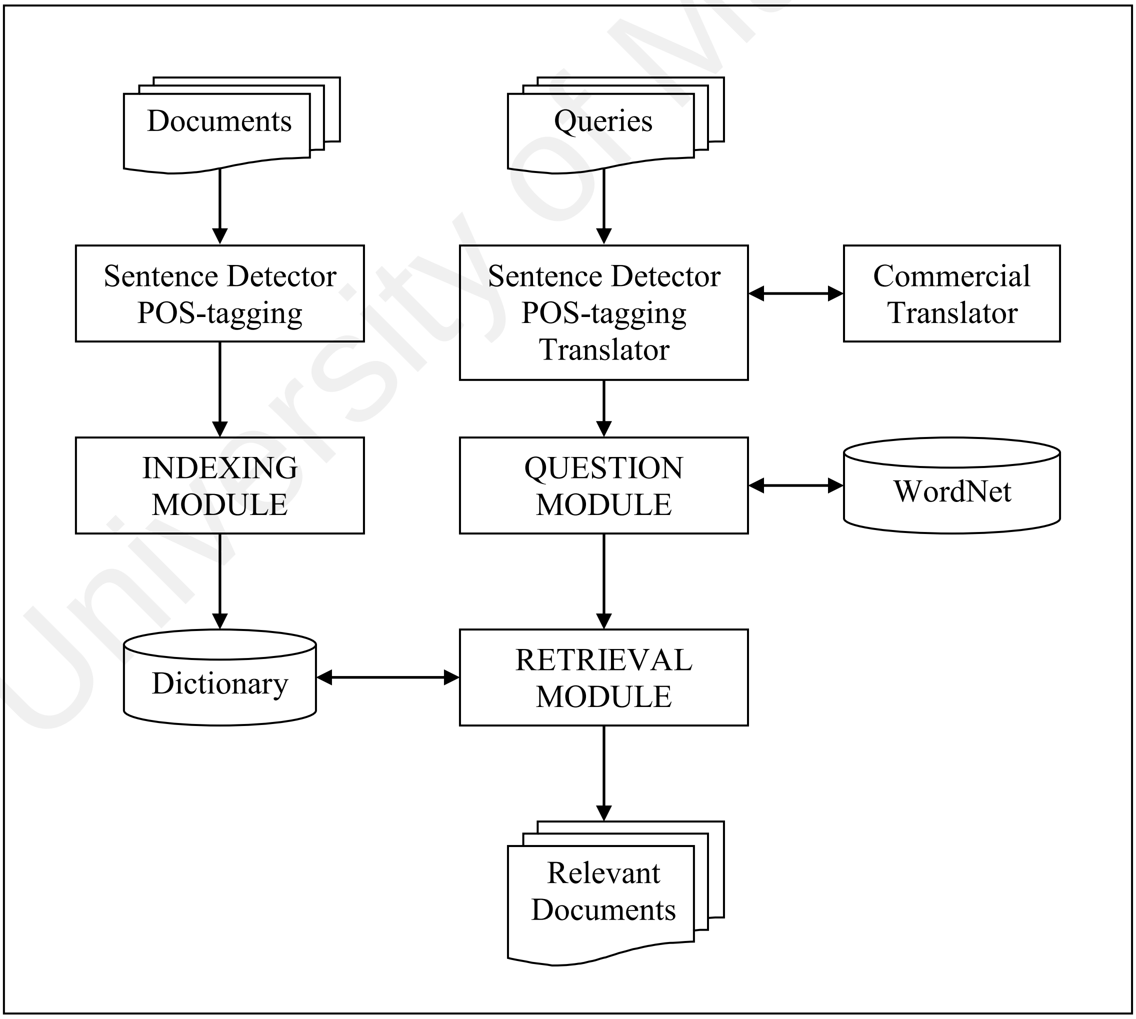


Figure 2.5: System architecture of IR-n (Llopis and Viceso, 2001)

The indexing module aims to create the dictionaries with the necessary information so that retrieval process is possible. The IR-n accomplishes this objective through several processes, such as detection of sentence boundaries and part-of-speech (POS) tagging, and elimination of stopwords. Subsequently, indexes are created. Indexes store each term together with the number of documents where each term appears, and for each one of the documents the number of appearances of each term and its position in the document (Llopis and Viceso, 2001).

In bilingual task, the first process for question module is to translate the text from Spanish to English using commercial translator. Then, the system removes stopwords and detects stem (for English text) or lemma (Spanish text) in both monolingual and bilingual tasks (Llopis and Viceso, 2001). A lexical thesaurus called WordNet is used to draw synonyms for each term in one of the tests during its participation CLEF-2001.

As its name reveal, retrieval module is responsible to retrieval relevant documents based on the measure of similarity of representations of each document and query. Results are obtained through the followings (Llopis and Viceso, 2001):

1. Order the query terms in the order starting from smaller to larger in function of the number of documents collection the terms appear.
2. Get the documents that contains at least one term and calculate the degree of similarity of each document.
3. Order the documents in function of their similarity to the query and generate a list of results for visualization.

The score of each document is obtained by calculating the similarity of query with each passage of the document, and the document is assigned with the highest degree of similarity of the passages within the document. The function is as following (Llopis and Viceso, 2001):

$$\text{Similarity of the passage} = \sum\nolimits_{t \in p \wedge d} W_{p,t} * W_{q,t}$$

where $W_{p,t}$ = $\log(f_{p,t}+1)$, being $f_{p,t}+1$ the number of occurrences of term t in passage p.

$W_{q,t}$ = $\log(f_{q,t}+1) * idf$, being $f_{q,t}+1$ the number of occurrences of term t in query q.

Idf = $\log_e(\frac{N}{f_t+1})$, being N the number of documents in the collection, and f_t is the number of different documents where the terms t appears.

2.3 Introduction to Cancer Disease

As mentioned in Chapter 1, cancer is a group of diseases characterized by uncontrolled growth and spread of abnormal cells. Cancers can be divided into two broad groups, namely cancers of blood forming system such as leukemias, lymphomas and myelomas, and cancers of organs such as lung cancers and breast cancers.

Cancer is not contagious; no one can "catch" cancer from another person who has this disease. Besides that, bruise or wound also does not cause cancer. Researches have shown that cancer is caused by damage to genes that control the growth and death of cells (National Cancer Institute, 2002). Genes can be damaged by both external factors (the use of tobacco, diet, exposure to the sun's ultraviolet radiation, exposure to carcinogens in the workspace or in the environment) and internal factors (inherited mutations, hormones, immune conditions and metabolism mutations). Anyone can develop cancer. Still, having one or more particular risk factors does not always mean that the person will get the disease. In fact, many people who get cancer have none of the known risk factors.

Although there is no guaranteed way to prevent cancer, American Cancer Society, National Cancer Institute of American and American Heart Association have endorsed some guidelines that can help in reducing the risk of developing cancer (Schneider, 2002):

- Avoid tobacco products;
- Minimize alcohol consumption (<1oz daily);
- Avoid sun exposure (especially from 10am to 3pm);
- Reduce fat intake (<30% daily calories);
- Reduce cholesterol (300mg daily);
- Increase carbohydrates (55%-65% daily calories);
- Reduce protein (15% daily calories);
- Increase dietary fiber intake (20-30g daily);
- Eat 5 fruits/vegetables daily;
- Avoid obesity; and
- Minimize salt intake (<6g daily).

Cancer can have a variety of symptoms. However, having these symptoms do not necessary mean that the person is having cancer. These symptoms can be caused by infections, benign tumors or other problems. The key factor in these symptoms is their persistence (Rapaport, 1978). One should see a doctor or oncologist if any condition continues for more than a few days. The possible symptoms of cancer include the following (National Cancer Institute, 2002; Rapaport, 1978):

- new thickening or lump in the body;
- new mole or an obvious change in the appearance of an existing wart or mole;
- a sore that does not heal;
- nagging cough or hoarseness;
- changes in bowel of bladder habits;
- persistent indigestion or difficulty swallowing;
- unexplained changes in weight; and
- unusual bleeding or discharge.

Treatment available for cancer can include surgery, radiation therapy, chemotherapy, hormone therapy and biological therapy. Depending on the conditions of cancer patient, doctor may use one method or a combination of methods (National Cancer Institute, 2002; Rapaport, 1978):

- Surgery is an operation to remove the part of body that affected by cancer and it is the first and oldest method of cancer treatment. Surgery can have certain side effects such as pain, feeling tired or weak for a while after surgery.
- Radiation therapy (or radiotherapy) kills cancer cells by using high-energy rays. The radiation therapy is painless. The side effects of radiotherapy including patients are likely to feel very tired, decrease in the number of white blood cells, temporary hair loss and the skin to become red, dry, tender and itchy.
- Chemotherapy (or drug therapy) treats cancer by using drugs to kill cancer cells throughout the body. Besides cancer cells, healthy cells are also affected because the drug travels throughout the body. Chemotherapy has several possible side effects such as hair loss, temporary fatigue, poor appetite, nausea and vomiting, and mouth and lip sores.

- Hormone therapy (or endocrine therapy) uses hormones to treat certain cancers that depend on hormones for their growth. Cancer cells are being kept from getting or using the hormones they need to grow. This is done by using of drugs to stop the production of certain hormones or changing the way hormones work, or removing organs that make hormones. Tired, fluid retention, weight gain, hot flashes, nausea and vomiting, changes in appetite, blood clots and bone loss in pre-menopausal women are the possible side effects of hormone therapy.
- Biological therapy (or immunotherapy) uses the body's natural ability (immune system) to fight cancer and lo lessen some of the side effects of cancer treatment, directly or indirectly. The side effects of biological therapy include having flu-like symptoms, such as chills, fever, muscle aches, weakness, loss of appetite, nausea, vomiting and diarrhea.

2.3.1 Breast Cancer

Breast cancer begins with cells within a breast undergo changes that cause it to grow and divide uncontrolled. Although it is very rare in men, both women and men can develop breast cancer. Breast cancer can be divided into types: non-invasive carcinoma (carcinoma in situ) and invasive carcinoma (Horwich, 1995). In non-invasive carcinoma, the cancer stays inside the ducts or lobules and has not spread into surrounding fatty tissues in the breast or to other organs in the body. On the other hands, if the cancer penetrates the membrane that surrounds the lobules or ducts, it is called invasive carcinoma and has a greater chance to spread throughout the body.

The risk of developing breast cancer increases with age. The risk of breast cancer is also higher among women who have previous history of benign breast disease, and personal or family history of breast cancer. Inherited genetic mutations (BRCA1 and BRCA 2) will also increase the chances of getting breast cancer. Besides, breast cancer risk is increased in women with the longest known exposures to sex hormones, particularly estrogen. Therefore, breast cancer risk is increased in women who have a history of (American Cancer Society, 2005; Bilimoria and Morrow, 1995; Vogel, 2000):

- early menstrual period (prior to age 14);
- late menopause (after age 55);
- no pregnancy or late pregnancy (after age 30);
- recent use of oral contraceptives; and
- use postmenopausal hormone therapy or estrogen replacement therapy.

The other factors that will increase the risk of develop breast cancer include obesity, radiation exposure and consumption of one or more alcoholic beverages per day (American Cancer Society, 2005; Vogel, 2000).

According to Vogel (2000), there are two effective methods to decrease breast cancer incidence, especially for high-risk women: prophylactic mastectomy and preventive therapy with tamoxifen. For women with low-to-normal risk, the preventive methods are related to lifestyle modifications, such as weight control, no cigarette smoking, decreasing alcohol consumption, exercise and avoidance of non-diagnostic ionizing radiation (Vogel, 2000). Although these lifestyle choices have not been proven to prevent breast cancer, they are generally associated with good health and generally believed to offer some protection against cancer.

The symptoms of breast cancer include (American Cancer Society, 2005):

- breast lump, thickening, swelling, distortion or tenderness;
- skin irritation or dimpling; and
- nipple pain, scaliness, ulceration, retraction or spontaneous discharge.

However, before the symptoms of breast cancer have grown to the point where physical signs exit and can be felt by the women or her health care provider, an early detection tool call mammography can be used to identify the earliest sign of the disease. Annual mammograms are recommended starting at age 40. Women should begin monthly breast self-examination (BSE) at age 20. Besides, women in their 20s and 30s should have clinical breast examination about every 3 years, and every year for women 40 and older. CBE should take place prior to mammography so that if a mass is detected on CBE, radiologist can pay attention on it during mammography (Smith et al., 2002). Women who have family history should have additional tests such as breast ultrasound or magnetic resonance imaging (MRI), or having more frequent exams (American Cancer Society, 2005).

Treatment options for breast cancer are depend on the size and characteristics of tumor and the patient's preferences. Treatment may involve lumpectomy (local removal of the tumor) or mastectomy (surgical removal of the breast), radiation therapy, chemotherapy or hormone therapy. Two or more treatments are often used in combination (American Cancer Society, 2005; Horwich, 1995).

2.3.2 Lung Cancer

There are two groups of lung cancers, specifically non-small cell lung cancer (including squamous cell, adenocarcinoma and large cell) and small cell lung cancer (Horwich, 1995). Small cell lung cancer is unique among lung cancers because of its rapid growth, dissemination at diagnosis and responsiveness to both radiotherapy and chemotherapy (Hinson and Perry, 1993). Based on clinical evidence, metastasis of small cell lung cancers are more likely to happen than non-small cell lung cancers. This is because although the size of cancer cells in small cell lung cancers is small, they can divide quickly and form large tumors, which can spread to other organs.

The most important risk factor for lung cancer is cigarette smoking (Smith et al., 2001; Smith et al., 2002). Cigarette smoking accounts about 90% of lung cancer (Biesalski et al., 1998). Other risk factors include (Biesalski et al, 1998; Hinson and Perry, 1993; American Cancer Society, 2005):

- secondhand smoke;
- genetic predisposition;
- exposure to non-tobacco carcinogens such as radon, asbestos, arsenic, nickel, chromium;
- previous lung disease such as chronic obstructive pulmonary disease;
- previous tobacco-related cancer; and
- low consumption of fruit and vegetables.

Since the major factor that causes lung cancer is cigarette smoking, the best way to prevent lung cancer is to quit smoking, or to never have started in the first place. Besides, avoiding being around people who are smoking and avoiding cigars and marijuana will also help in preventing lung cancer. People who work in environment that exposed to substances known to cause lung cancer needed to use all the proper protective equipments. In addition, according to Biesalski et al. (1998), studies have shown that high consumption of vegetables and fruit have reduced the risk of developing lung cancer. This is because vegetables and fruit contain a lot of antioxidants such as carotenoids, vitamin A, vitamin C and vitamin E.

Symptoms of lung cancer may include persistent cough, sputum streaked with blood, chest pain and recurring pneumonia or bronchitis (American Cancer Society, 2005; Rapaport, 1978).

Effort of lung cancer detection includes chest x-ray and analysis of cells in sputum. However, the need for and usefulness of lung cancer detection has been a persistent debate (Smith et al., 2001). This is because studies of lung cancer detection have not demonstrated persuasively that chest x-ray alone or in combination with sputum cytology saves lives. On the other hands, newer tests such as low-dose spiral computed tomography (CT) scans, fluorescence bronchoscopy and molecular screening have produced promising results in detecting lung cancers at early stage (Mulshine, 2005; Smith et al., 2001). Nonetheless, the International Conference on Prevention and Early Diagnosis of Lung Cancer that was held in 1998 concluded that the current evidence about the efficacy of lung cancer screening was an imperfect basis for public health policy and urged an accelerated program to determine the efficacy and effectiveness of these new technologies for early lung cancer detection (Smith et al., 2001).

The type and stage of lung cancer determine the treatment options. For early stage non-small cell lung cancer, normally the treatment options are chemotherapy following surgery. Surgery is the preferred treatment for non-small cell lung cancer patients that has not spread beyond the chest. However, if the disease has spread, radiation therapy and chemotherapy are often used. For small cell lung cancer, chemotherapy alone or combined with radiation therapy are the best treatment options (Hinson and Perry, 1993).

2.4 Summary

This chapter reviewed the topic of NLP, specifically on IR. The background research or literature review is an important stage to review and gather information regarding this project. Several IR systems were also been reviewed. Based on the findings of literature review, vector space model will be implemented to perform the indexing and searching of the proposed Cancer Information System.

Chapter Three: Methodology

3.1 Introduction

This chapter profiles the methodology undertaken in the various phases of the project. Methodology is defined as a complete technique including step-by-step procedures, deliverables, roles, tools and quality standards for completing the system development (Whitten, Bentley and Barlow, 1994). Methodology steps must be applied from the starting of a project until the end in order to achieve the purposes of the project.

A well-defined methodology is essential to the development of this project since the technique of the methodology will affect the design of the whole system. A number of methodologies have been implemented in this project. These include software development life cycle (Waterfall model with iteration) and methodology for survey, which are discussed in the subsequent sections. Three existing cancer Websites are also reviewed in this chapter.

3.2 Software Development Life Cycle (SDLC)

This project has been developed by using Software development life cycle (SDLC). SDLC is one of the most common and frequently adopted software development methodologies in many contemporary software firms. It is a traditional methodology that features several phases such as requirements, design, implementation and testing. According to Bahrami (1999), software development methodology is a series of process that if followed can lead to the development of a software system that will satisfy the user requirements.

Figure 3.1 shows the general model of SDLC. Each phase produces deliverables required by the next phase in the life cycle. Results of requirements analysis are used to design the software. Then, coding is performed during implementation phase by referred to the design. Finally, testing verifies the deliverable of the implementation phase against requirements.

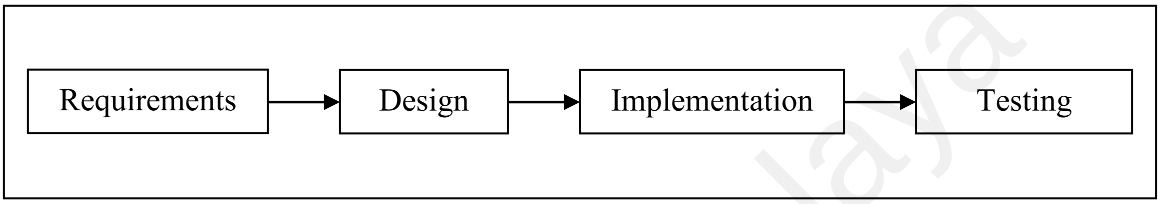


Figure 3.1: The general model of SDLC (Lewallen, 2005)

Based on the phases in SDLC, various life cycle models have been developed with the phases in SDLC being interpreted differently, such as the Waterfall model, V-shaped model, iterative and incremental model, spiral model, prototyping model, rapid application development model and so on. The model implemented in this project is Waterfall model with iteration.

3.2.1 Waterfall Model

Waterfall Life Cycle model is the oldest and most widely used SDLC model (Als and Greenidge, 2003). Waterfall Life Cycle model is also known as Traditional Life Cycle (TLC) model. It is named so because of the difficulty of returning to earlier phase once it is completed. This is very much like the difficulty of swimming up a waterfall. The life cycle in Waterfall model is linear and sequential.

Bennett et al. (2002) asserted that there are many variations of the Waterfall model, differing mainly in the number and names of phases, and the activities inside each phase. This project adopted Waterfall model (with iteration), which is shown in Figure 3.2. The Waterfall model with iteration allows corrections to be made to previous phases if any problem is found at later phase of software development.

The Waterfall Life Cycle model was chosen in this project because of the following advantages (Lewallen, 2005):

- The Waterfall model is relatively simple and easy to use compared to other models. It requires each and every phase needs to be completed one at a time. Therefore, the system will be easier to manage and keep tracked.
- The model works well for small scale projects where requirements are very well understood. The model was implemented in this project because it is a small scale project.

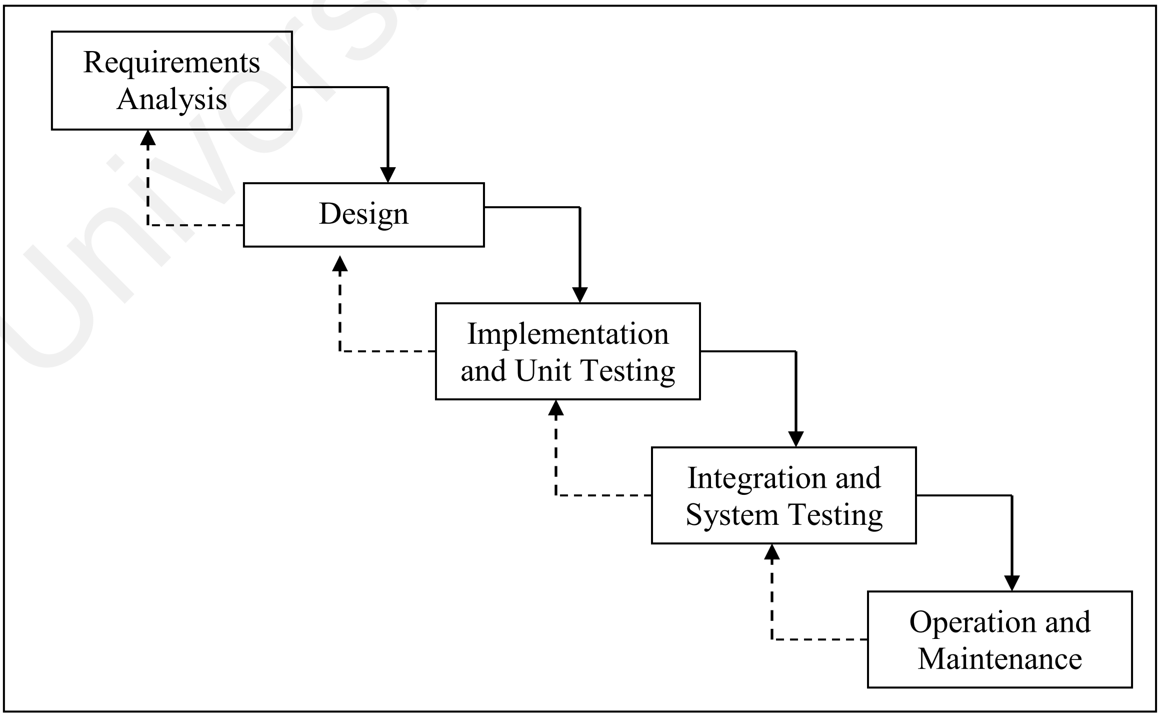

Requirements Analysis
Design
Implementation and Unit Testing
Integration and System Testing
Operation and Maintenance

Figure 3.2: The Waterfall model with iteration (Sommerville, 1998)

3.2.1.1 Requirements Analysis

Requirements analysis was the first phase in developing this project, which is also the most important phase that consisted of several activities such as instigation, scoping and identifying the requirements to design a new system. Requirements analysis is also known as requirements engineering and system engineering. There were two objectives during requirements analysis, which were to identify the problem and to identify the requirements of the proposed system.

In order to accomplish the first objective, background study has been performed to identify the research problem. Background study has been done by reviewing articles released by Ministry of Health Malaysia and National Cancer Registry. These articles reveal that cancer is a serious disease threatening this country. However, there is no local community-centric cancer Website that helps Malaysian to understand the disease. Besides, review on IR articles showed that IR techniques can be implemented in order to help users in accessing information. Therefore, a survey has been performed in order to investigate the feasibility of this project and to help in defining the scope of project. With research problem identified, research objectives and scope of research were determined and discussed in Chapter One.

The second objective aimed to find the need and to define the problem that are necessary to be addressed. The needs are normally based on the user of the software application. Hence, several existing cancer information Websites have been reviewed and a survey has been conducted in order to capture the requirements of the proposed system. Complete, unambiguous and understandable requirements are fundamental that leads to a successful project. Output of requirements analysis phase is the system as a whole and how it performs its tasks. The requirements analysis of this project was discussed in Chapter Four.

3.2.1.2 Design

Design phase was the second phase in this project. The findings of requirements analysis served as the basis for the design phase. Design phase revealed how the proposed system will work by determining how best to build a system that delivers the agreed requirements. Requirements were translated into a representation of the software. First, the architecture of the proposed system that illustrates the major components and their relationships was produced. Next, the design within every system module (Question Analysis Module, Document Indexing Module and Document Retrieval Module) was determined. Besides, the design phase also defined the design of the documents collection and interfaces of the Cancer Information System. The system design of the project was explained in Chapter Five.

3.2.1.3 Implementation and Unit Testing

The objective of implementation phase was to develop code based on the design that will meet the project's requirements. The implementation or coding of the project was performed on one sub-task at a time. This phase is the longest phase of Waterfall model in this project. Java programming language is used as the main programming language in this project and some of the algorithms of programming are addressed in Chapter Six.

After each sub-task has been coded, it was then tested. Unit testing was performed in order to find errors or defects of the program developed and to ensure that the program performs as expected. The details of testing are described in Chapter Seven.

3.2.1.4 Integration and System Testing

After every module has been developed, module testing was performed to ensure that each module is free from errors. After that, all the system modules were combined into a whole functional Cancer Information System. Then, it was tested to indicate errors which originated from interactions between different modules of the system and to ensure that the complete system meets the software requirements.

During system evaluation, the retrieval accuracy of the developed system was tested by using a set of 104 Frequently Asked Questions (FAQs) drawn from various cancer information Websites. Two metrics that were used during system evaluation are *precision* and *recall*. System testing and evaluation are discussed in Chapter Seven.

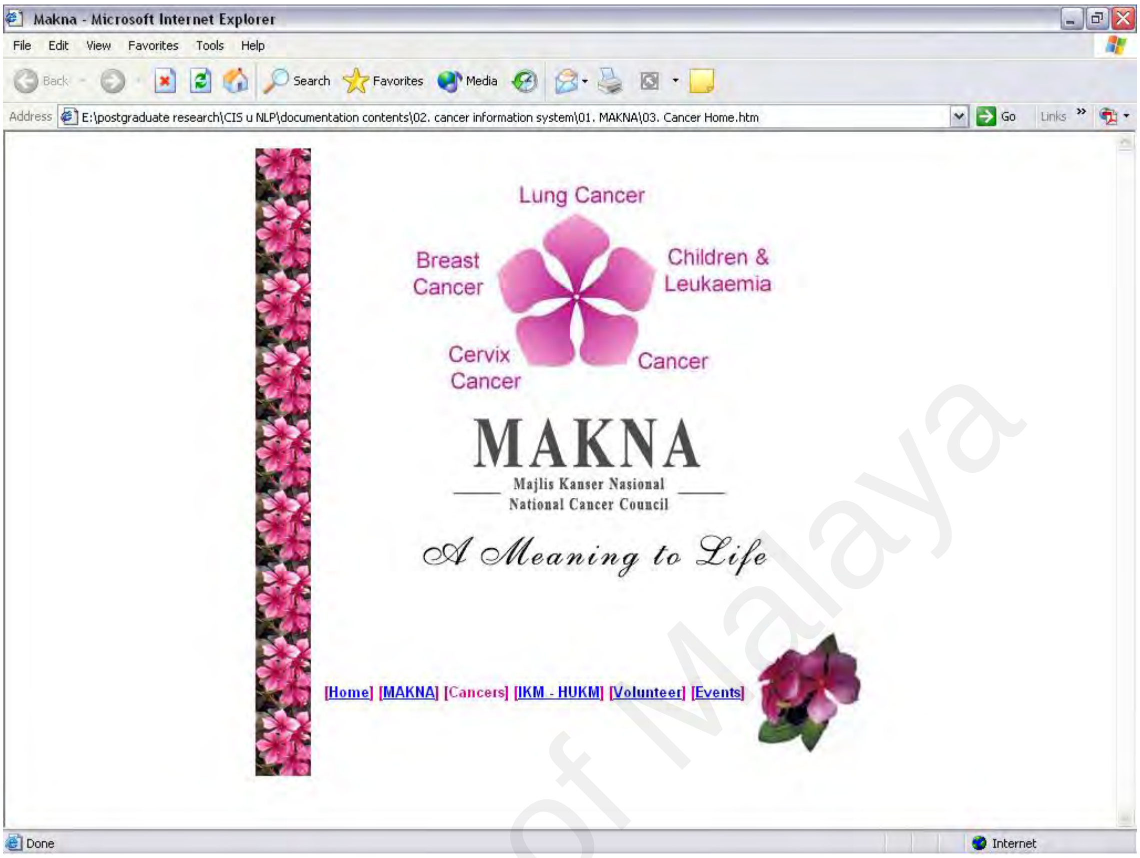
3.2.1.5 Operation and Maintenance

The maintenance is normally the longest phase for any software development project. During the Cancer Information System has been deployed, user may discover errors that have not been identified during system testing phase. These errors need to be corrected. Besides, new requirements may be discovered or existing requirements need to be modified to adapt to the changing environment. Since the Cancer Information System was developed by using Waterfall model with iteration, it is possible to modify and enhance the system in the future.

3.3 Review of Cancer Information Websites

The following sub-sections review three existing cancer information Websites. These Websites are from Malaysia, United States of America (U.S.) and England respectively. The purpose of this review is to study the features provided by these existing Websites.

3.3.1 National Cancer Council (MAKNA), Malaysia



Address: E:\postgraduate research\CIS u NLP\documentation contents\02. cancer information system\01. MAKNA\03. Cancer Home.htm

Figure 3.3: Screenshot of cancer Website developed by MAKNA

The first reviewed cancer information system was developed by National Cancer Council of Malaysia (http://www.makna.org.my/index.aspx) and it was the sole generic purpose cancer information system available from Malaysia. This is a small Website since it only contains fourteen pages. In terms of types of cancer, this Website covers only lung cancer, breast cancer, cervix cancer, children and leukemia, and a general introduction to cancer, as shown in Figure 3.3. General descriptions such as causes, symptoms, diagnosis, treatment and other related information are available for each of this cancer. MAKNA cancer Website is rather a simple cancer information system. It can be considered ineffective in providing information to user because of its limited contents.

3.3.2 National Cancer Institute (NCI), U.S.

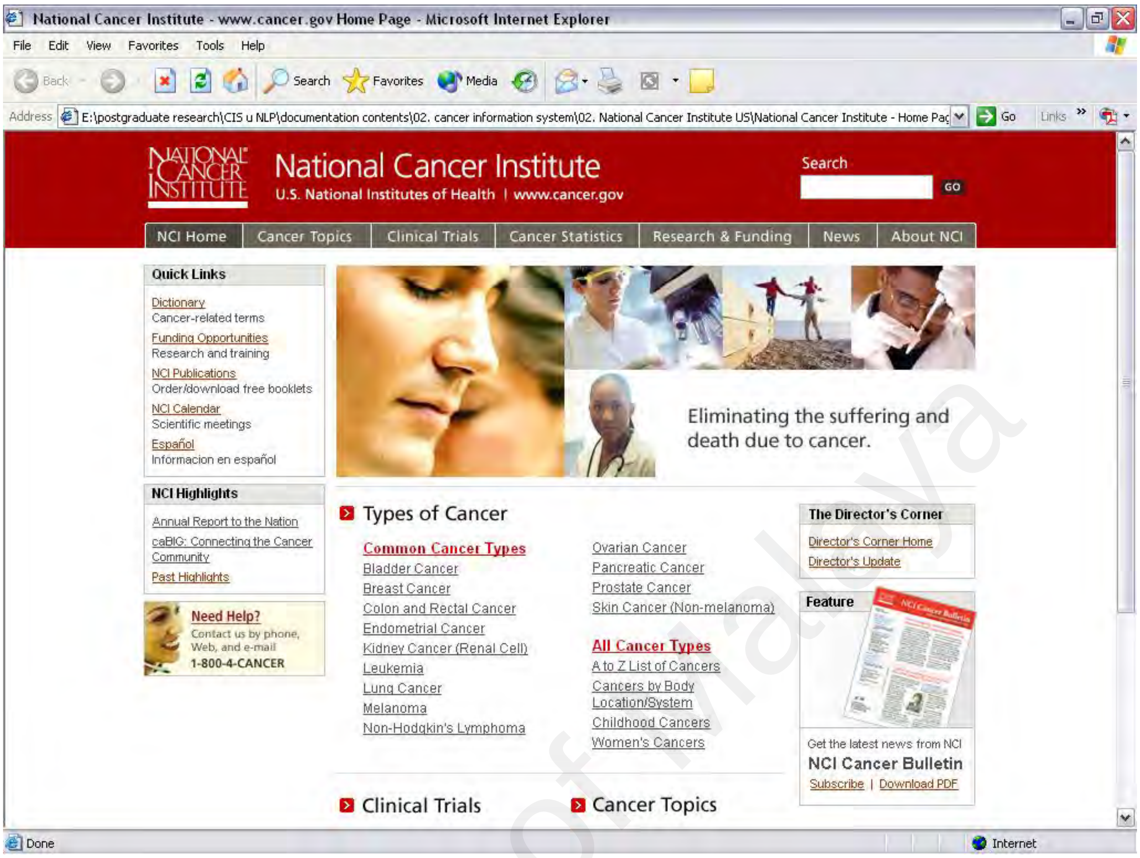


Figure 3.4: Homepage of NCI

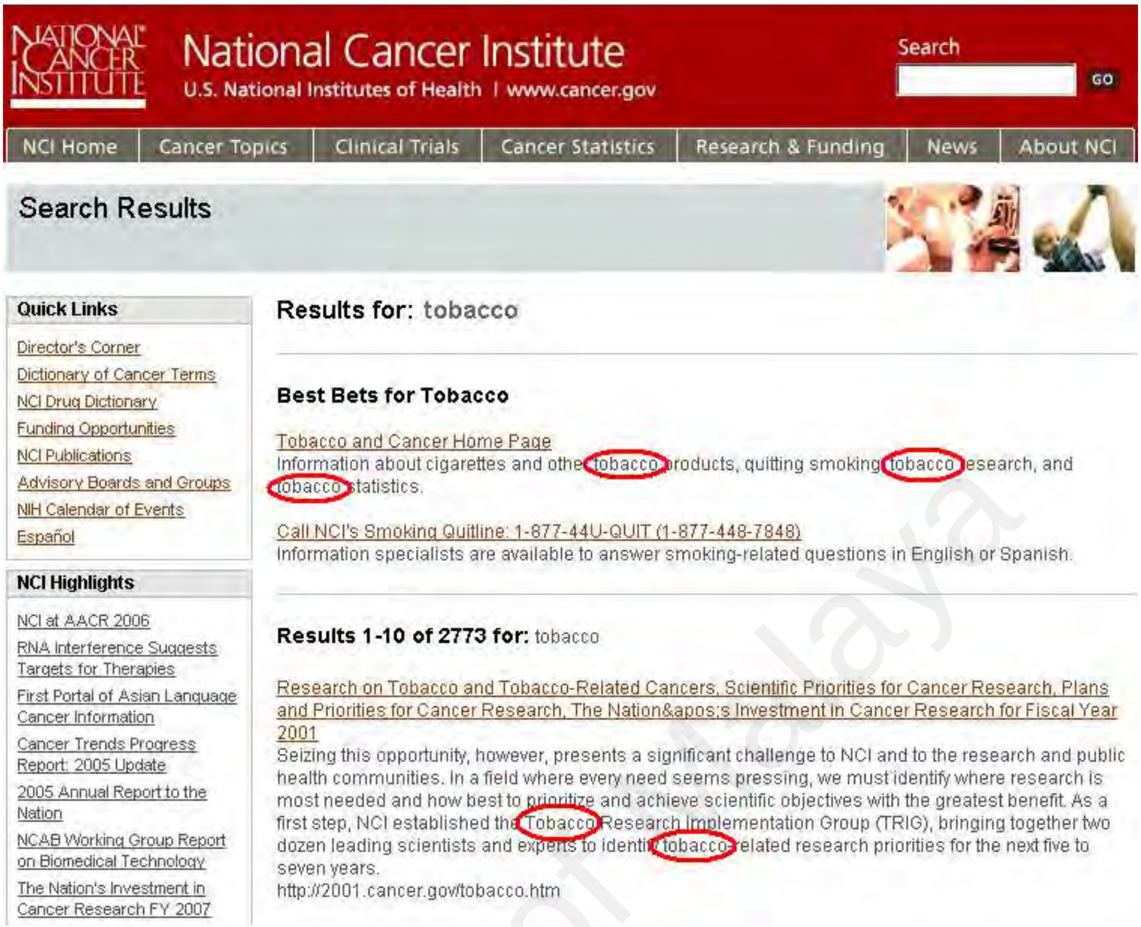
The second cancer information system review was developed by National Cancer Institute, U.S. (http://cancer.gov), which is a Website specialized only on cancer disease. Apparently, the design of this cancer information system is well-structured and too simple. The homepage of this information system provides the high-level categorization of information available in the Website, as shown in Figure 3.4. Different types or categories of information can be reached through well-design tabs:

- *NCI Home* – Links to the homepage of the Website;
- *Cancer Topics* – Provides wide range of information such as all cancer types, treatment, complementary and alternative medicine, prevention, causes, screening and testing, cancer terminology resources and so on;

- *Clinical Trials* – Determines whether new drugs or treatments are both safe and effective, and this service is only available in U.S. The tests are done on a group of voluntary patients;
- *Cancer Statistics* – Provides some statistics related to cancer such as survival rate since diagnosis;
- *Research & Funding* – Provides information regarding NCI's researches and funding policies;
- *News* – Contains all news released by NCI; and
- *About NCI* – Provides some background information on NCI organization.

Additionally, this cancer information system also provides a dictionary to describe terminologies that have been used in the Website, a site map that illustrates the categorization of available information and the system is also available in Spanish version.

Besides that, a full-text search function was provided as an alternative to help user to find specific information more quickly. However, the keyword used during the search was not highlighted both in the search results and in the retrieved documents. Hence, user has to spend some time to locate the keyword manually. For example, Figure 3.5 shows the keyword "tobacco" was being used to search relevant documents, but it was not highlighted in the search results.


National Cancer Institute
U.S. National Institutes of Health | www.cancer.gov
Search
GO
NCI Home
Cancer Topics
Clinical Trials
Cancer Statistics
Research & Funding
News
About NCI
Search Results
Quick Links
Director's Corner
Dictionary of Cancer Terms
NCI Drug Dictionary
Funding Opportunities
NCI Publications
Advisory Boards and Groups
NIH Calendar of Events
Español
NCI Highlights
NCI at AACR 2006
RNA Interference Suggests Targets for Therapies
First Portal of Asian Language Cancer Information
Cancer Trends Progress Report: 2005 Update
2005 Annual Report to the Nation
NCAB Working Group Report on Biomedical Technology
The Nation's Investment in Cancer Research FY 2007

Results for: tobacco

Best Bets for Tobacco

Tobacco and Cancer Home Page
Information about cigarettes and other tobacco products, quitting smoking, tobacco research, and tobacco statistics.

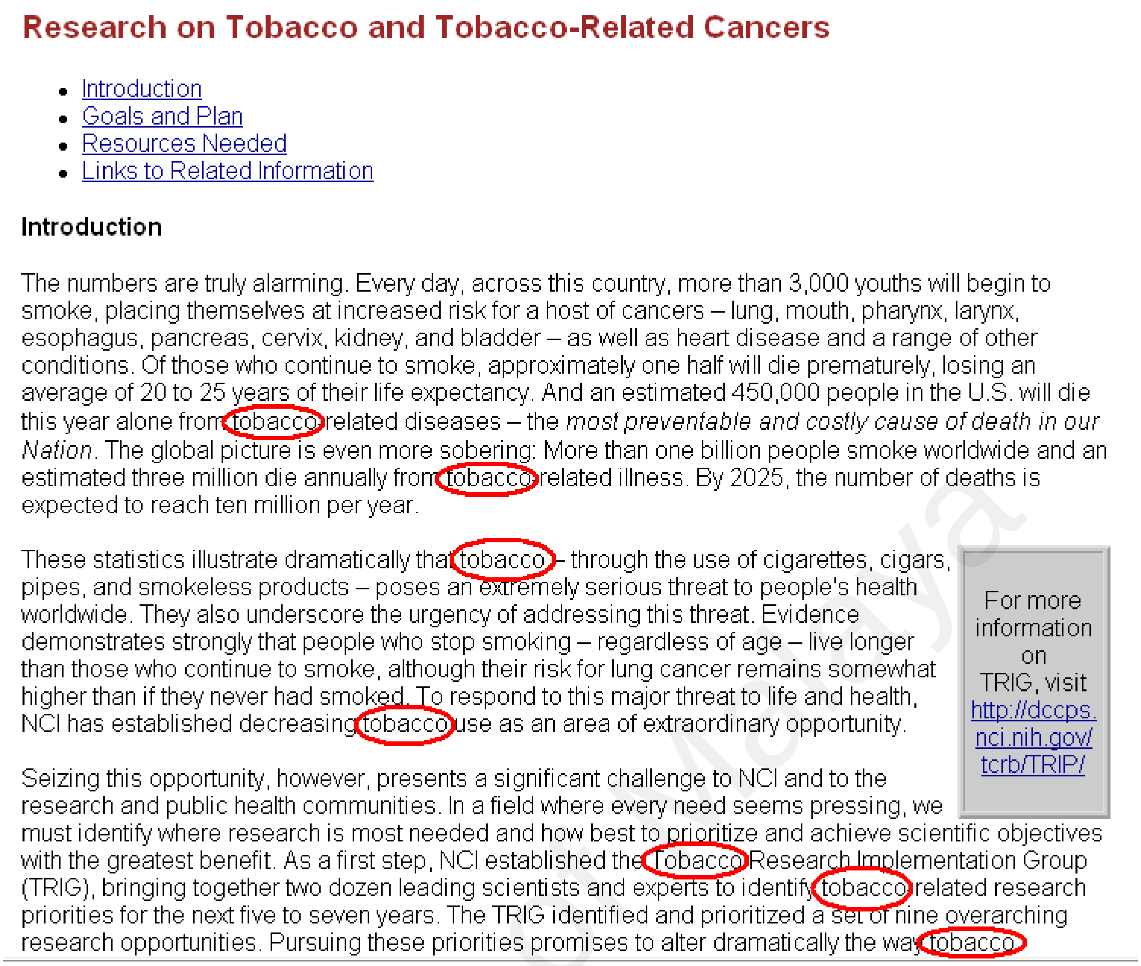
Call NCI's Smoking Quitline: 1-877-44U-QUIT (1-877-448-7848)
Information specialists are available to answer smoking-related questions in English or Spanish.

Results 1-10 of 2773 for: tobacco

Research on Tobacco and Tobacco-Related Cancers, Scientific Priorities for Cancer Research, Plans and Priorities for Cancer Research, The Nation&apos;s Investment in Cancer Research for Fiscal Year 2001
Seizing this opportunity, however, presents a significant challenge to NCI and to the research and public health communities. In a field where every need seems pressing, we must identify where research is most needed and how best to prioritize and achieve scientific objectives with the greatest benefit. As a first step, NCI established the Tobacco Research Implementation Group (TRIG), bringing together two dozen leading scientists and experts to identify tobacco-related research priorities for the next five to seven years.
http://2001.cancer.gov/tobacco.htm

Figure 3.5: Keyword is not highlighted in the search results

If user clicks on any one of the hyperlinks in the search results, user will be redirected to the respective document. Similarly, the word “tobacco” was also not highlighted in the document, as shown in Figure 3.6.



Research on Tobacco and Tobacco-Related Cancers

- Introduction
- Goals and Plan
- Resources Needed
- Links to Related Information

Introduction

The numbers are truly alarming. Every day, across this country, more than 3,000 youths will begin to smoke, placing themselves at increased risk for a host of cancers – lung, mouth, pharynx, larynx, esophagus, pancreas, cervix, kidney, and bladder – as well as heart disease and a range of other conditions. Of those who continue to smoke, approximately one half will die prematurely, losing an average of 20 to 25 years of their life expectancy. And an estimated 450,000 people in the U.S. will die this year alone from tobacco-related diseases – the *most preventable and costly cause of death in our Nation*. The global picture is even more sobering: More than one billion people smoke worldwide and an estimated three million die annually from tobacco-related illness. By 2025, the number of deaths is expected to reach ten million per year.

These statistics illustrate dramatically that tobacco – through the use of cigarettes, cigars, pipes, and smokeless products – poses an extremely serious threat to people's health worldwide. They also underscore the urgency of addressing this threat. Evidence demonstrates strongly that people who stop smoking – regardless of age – live longer than those who continue to smoke, although their risk for lung cancer remains somewhat higher than if they never had smoked. To respond to this major threat to life and health, NCI has established decreasing tobacco use as an area of extraordinary opportunity.

For more information on TRIG, visit http://dccps.nci.nih.gov/tcrb/TRIP/

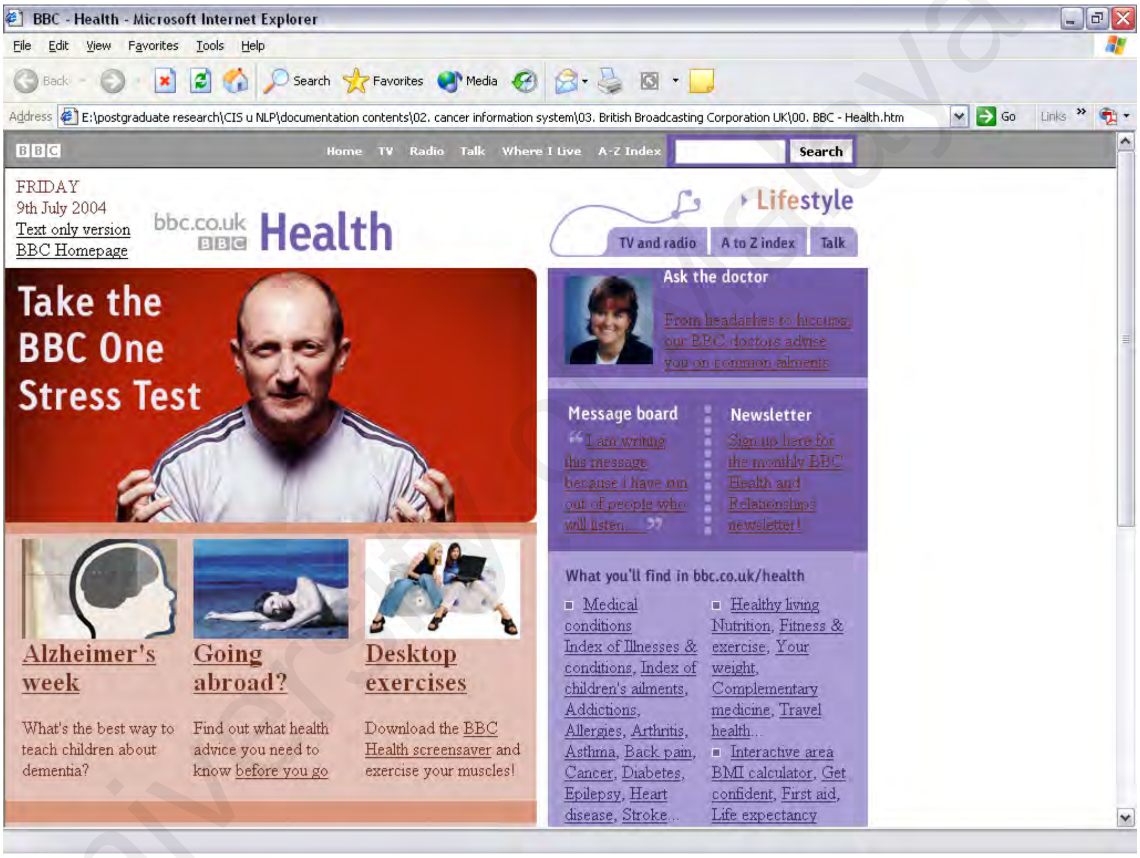
Seizing this opportunity, however, presents a significant challenge to NCI and to the research and public health communities. In a field where every need seems pressing, we must identify where research is most needed and how best to prioritize and achieve scientific objectives with the greatest benefit. As a first step, NCI established the Tobacco Research Implementation Group (TRIG), bringing together two dozen leading scientists and experts to identify tobacco-related research priorities for the next five to seven years. The TRIG identified and prioritized a set of nine overarching research opportunities. Pursuing these priorities promises to alter dramatically the way tobacco

Figure 3.6: Keyword is not highlighted in the retrieved document

As a conclusion, this cancer information system which is developed by NCI provides a lot of useful information related to cancers. Nonetheless, if user wants to browse Web pages through hyperlinks, then knowledge about the categorization is needed in order to get to the targeted information quickly. A full-text search function is also available to help user to find information. However, the only disadvantage is that keyword used in the search is not highlighted. One way to locate the keyword is to start from the beginning of the document and to read through all the text. Apparently, user may miss some of the information when browsing through the search results and retrieved documents especially when the text is too long. Another way is to use the *Find* function provided by Web browser. However, this is not convenient because user has to constantly switch between two Windows, that is, the *Find* Window and the text

Window. These repeat the process of locating the next occurrence of keyword and reading text. The problem becomes worst if user used more than one non-adjacent keyword (keywords that are not exist next to each other) during the search. This is because user has to find every non-adjacent keyword at the beginning of the text.

3.3.3 British Broadcasting Corporation (BBC), England


BBC - Health - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address E:\postgraduate research\CIS u NLP\documentation contents\02. cancer information system\03. British Broadcasting Corporation UK\00. BBC - Health.htm
Go Links
BBC Home TV Radio Talk Where I Live A-Z Index Search
FRIDAY
9th July 2004
Text only version
BBC Homepage
bbc.co.uk BBC Health
Lifestyle
TV and radio A to Z index Talk
Take the BBC One Stress Test
Ask the doctor
Message board
Newsletter
What you'll find in bbc.co.uk/health
Alzheimer's week
Going abroad?
Desktop exercises

From headaches to hiccups, our BBC doctors advise you on common ailments

"I am writing this message because i have run out of people who will listen... "

Sign up here for the monthly BBC Health and Relationships newsletter!

□ Medical conditions
Index of Illnesses & conditions, Index of children's ailments, Addictions, Allergies, Arthritis, Asthma, Back pain, Cancer, Diabetes, Epilepsy, Heart disease, Stroke...

□ Healthy living
Nutrition, Fitness & exercise, Your weight, Complementary medicine, Travel health...

□ Interactive area
BMI calculator, Get confident, First aid, Life expectancy

What's the best way to teach children about dementia?

Find out what health advice you need to know before you go

Download the BBC Health screensaver and exercise your muscles!

Figure 3.7: Homepage of BBC Health topics

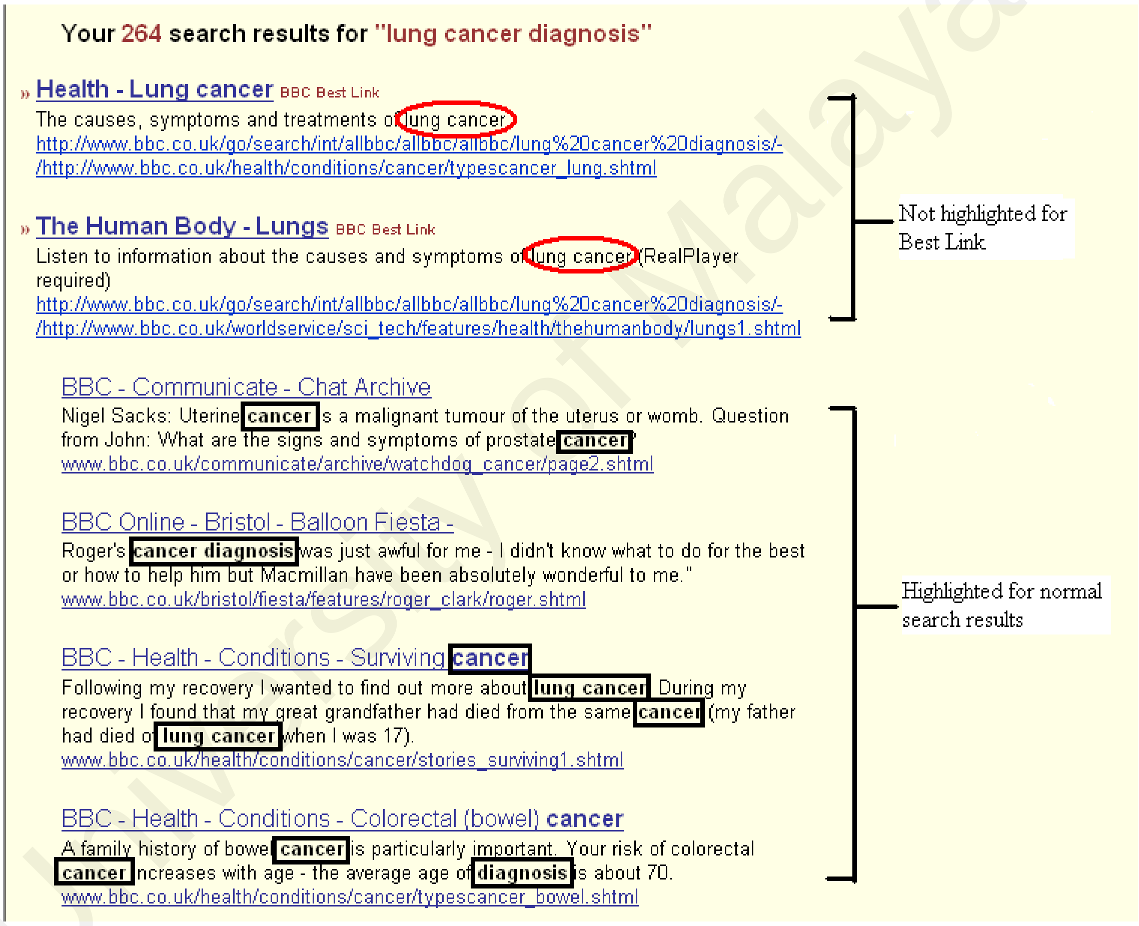
The third reviewed cancer information system was developed by British Broadcasting Corporation (BBC) (http://www.bbc.co.uk/health/cancer/index.shtml). This cancer information system is actually located under the health category section (BBC Health), which is located under the main BBC Website. It is based on England and collaborates with National Health Service (NHS) to offer signposts about what to expect and how to get the services under National Health System. Figure 3.7 shows a screenshot of the homepage of health topics.

This Website clearly exhibits the typical navigation problem that associated with HTML. If user enters this Website through BBC homepage, then user has to search for the *Health* hyperlink that will link directly to the homepage of health topics. Subsequently, user has to search for the *Cancer* hyperlink in order to browse the cancer topics. For huge Websites such as the one developed by BBC, there are many hyperlinks which can be found only in one Web page and therefore finding information in such Websites is not an easy task. In addition, all subcategories or subtopics are not revealed until user navigated to that particular level of navigation tree. Hence, the information categorization has to be designed carefully according to taxonomy relationship. If user is unable to find the information in a short period of time, normally they will get frustrated.

Under the cancer section, the Website provides broad types of cancer such as larynx cancer, childhood lymphomas, colorectal, leukaemia, lung cancer, skin cancer, stomach cancer, male cancers and woman cancers. Within each of these cancers, common topics such as general description of cancer, causes, symptoms, diagnosis and treatment were provided.

Besides, the Website also afford complementary information such as listening and understanding cancer patient, self look after, diet control and so on. Besides the contents mentioned above, BBC also makes a good effort to broadcast real case of cancer storyline through BBC channel to increase cancer awareness.

A full-text search function is also available to facilitate user to search by using specific keywords. Keywords used in the search are highlighted in the normal search results to help user locate them quickly, but not highlighted in the best search results, as shown in Figure 3.8 (keywords used in this example are “lung”, “cancer” and “diagnosis”). In addition, keywords are not highlighted in the retrieved documents, as shown in Figure 3.9.



Your 264 search results for "lung cancer diagnosis"

» Health - Lung cancer BBC Best Link
The causes, symptoms and treatments of lung cancer
http://www.bbc.co.uk/go/search/int/allbbc/allbbc/allbbc/lung%20cancer%20diagnosis/-/http://www.bbc.co.uk/health/conditions/cancer/typescancer_lung.shtml

» The Human Body - Lungs BBC Best Link
Listen to information about the causes and symptoms of lung cancer (RealPlayer required)
http://www.bbc.co.uk/go/search/int/allbbc/allbbc/allbbc/lung%20cancer%20diagnosis/-/http://www.bbc.co.uk/worldservice/sci_tech/features/health/thehumanbody/lungs1.shtml

BBC - Communicate - Chat Archive
Nigel Sacks: Uterine cancer is a malignant tumour of the uterus or womb. Question from John: What are the signs and symptoms of prostate cancer?
www.bbc.co.uk/communicate/archive/watchdog_cancer/page2.shtml

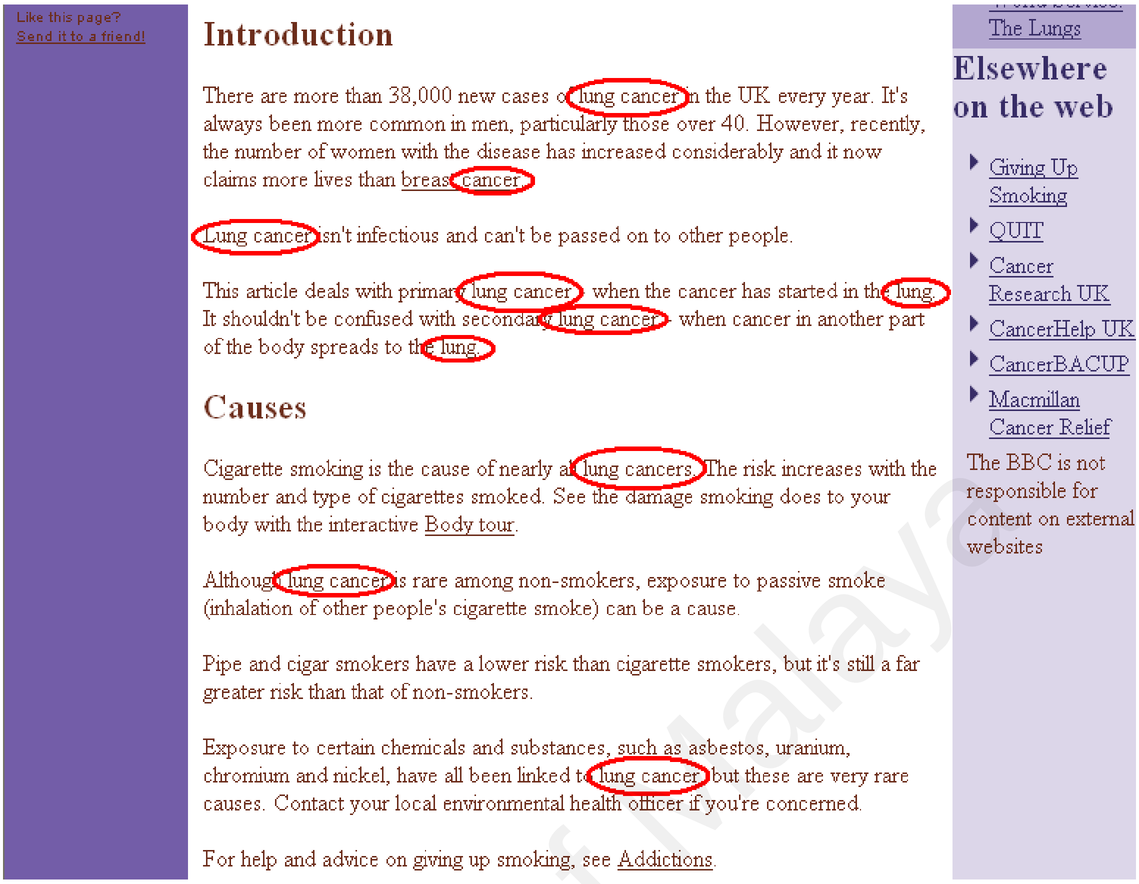
BBC Online - Bristol - Balloon Fiesta -
Roger's cancer diagnosis was just awful for me - I didn't know what to do for the best or how to help him but Macmillan have been absolutely wonderful to me."
www.bbc.co.uk/bristol/fiesta/features/roger_clark/roger.shtml

BBC - Health - Conditions - Surviving cancer
Following my recovery I wanted to find out more about lung cancer. During my recovery I found that my great grandfather had died from the same cancer (my father had died of lung cancer when I was 17).
www.bbc.co.uk/health/conditions/cancer/stories_surviving1.shtml

BBC - Health - Conditions - Colorectal (bowel) cancer
A family history of bowel cancer is particularly important. Your risk of colorectal cancer increases with age - the average age of diagnosis is about 70.
www.bbc.co.uk/health/conditions/cancer/typescancer_bowel.shtml

Not highlighted for Best Link
Highlighted for normal search results

Figure 3.8: Keywords are highlighted in the normal search results but not Best Link



Like this page?
Send it to a friend!

Introduction

There are more than 38,000 new cases of lung cancer in the UK every year. It's always been more common in men, particularly those over 40. However, recently, the number of women with the disease has increased considerably and it now claims more lives than breast cancer.

Lung cancer isn't infectious and can't be passed on to other people.

This article deals with primary lung cancer, when the cancer has started in the lung. It shouldn't be confused with secondary lung cancer, when cancer in another part of the body spreads to the lung.

Causes

Cigarette smoking is the cause of nearly all lung cancers. The risk increases with the number and type of cigarettes smoked. See the damage smoking does to your body with the interactive Body tour.

Although lung cancer is rare among non-smokers, exposure to passive smoke (inhalation of other people's cigarette smoke) can be a cause.

Pipe and cigar smokers have a lower risk than cigarette smokers, but it's still a far greater risk than that of non-smokers.

Exposure to certain chemicals and substances, such as asbestos, uranium, chromium and nickel, have all been linked to lung cancer but these are very rare causes. Contact your local environmental health officer if you're concerned.

For help and advice on giving up smoking, see Addictions.

The Lungs

Elsewhere on the web

- Giving Up Smoking
- QUIT
- Cancer Research UK
- CancerHelp UK
- CancerBACUP
- Macmillan Cancer Relief

The BBC is not responsible for content on external websites

Figure 3.9: Keywords are not highlighted in the retrieved document

Generally, BBC cancer Website provides the most complete contents among three reviewed Web-based cancer information systems. Similar to the second reviewed system, knowledge of the information structure is a must in order to search for the desire information quickly, especially when the Website contains large amount of Web pages. Subsequently, the inconsistency in keywords highlights may confuse particular user.

3.4 Survey

During the requirements analysis phase, a survey has been carried out to collect information from a sample of individuals in order to describe some characteristics related to this project. This survey was a cross-sectional survey, in which information was collected at approximately the same point at a time. The following sub-sections describe the procedures of the survey in detail.

3.4.1 Determining Survey Objectives and Data Collection Mode

The first step in the survey is to identify the objective of investigation (Scheuren, 2004). The survey that has been carried out aims to suffice two major objectives:

- To study the feasibility and necessity of a new Malaysia-based cancer information system by using IR techniques.
- To identify the contents of the proposed Cancer Information System.

Given these objectives, the mode of data collection must be determined. The survey in this project was done through direct administration to a group. The obvious advantage of this approach is the high rate of response, in which in this project the response rate was 97.5% or 39 replies over 40 sets of questionnaire. With comparison to other modes of data collection such as telephone, mail and through Internet, direct administration also has other advantages include relatively low cost and time saving.

3.4.2 Designing Questionnaire

Given the objectives and mode of data collection, the next step is designing the questionnaire. The questionnaire was designed to be short but able to suffice the objectives. This is because excessively long questionnaires are burdensome to the respondents, are incline to induce fatigue among the respondents and hence the results of questionnaire may apt to be inaccurate (Ferber, Sheatsley, Turner and Waksbery, 1980).

The questionnaire can be referred in Appendix A. Table 3.1 summarizes the types of question designed in the questionnaire.

Table 3.1: Summary of questionnaire layout

Type of Question	**Number of Question**
Yes/no question	4
Multiple choice	3
Mix of multiple choice and open-ended question	4

3.4.3 Identification of Target Population and Sample

Usually, it is impractical to collect information from all members of targeted population. Therefore, surveys are needed to gather information from only a portion of the population being studied, which is called sample. The sample reflects the characteristics of the population from which it is drawn. The information collected from a number of respondents or units of sample describe those units; this information is then summarized to describe the characteristics of the population those samples represent.

It is very difficult to achieve random sampling in the real world because of time, cost and ethical constraints (Lunsford and Lunsford, 1995). In a true random sampling situation, each individual of the population has to be identified and each one of them must have equal change to be selected as sample (usually by using a list of the entire population). However, it seems impossible to achieve this through the project. The sampling method used throughout the project was convenience sampling. This method is most commonly used during exploratory study to get a gross estimate of the results without spending too much cost and time to select a random sample (StatPac Inc., 2005). As the name implies, the sample of this non-probability method is selected because of easy availability and accessibility of sample.

The target population has been identified to be all population in Malaysia since this project concerns building a Malaysia-based cancer information system. On the other hand, the sample group has been identified to be the postgraduate students of the Faculty of Computer Science and Information Technology (FCSIT), University of Malaya. The selection of this sample group is because it adequately represents a subset of the target population. The sample size was limited to 40. This is the suitable size based on time and resources constraints.

3.4.4 Pilot Test

Before conducting the survey, pilot test of questionnaire is needed to identify any questionnaire problems or to find out if every question works as expected (Ferber et al., 1980). Generally, a pilot test is important because it is rarely possible to foresee all the possible misunderstandings or biasness effects of different questions in the questionnaire. Pilot test aims to check whether a question is too sensitive, a question invades the respondent's privacy, or it is too difficult to answer (Scheuren, 2004).

In this survey, the pilot test has been done in a small-scale with two participants. The type of pilot test was respondent debriefings. They were asked to complete the questionnaire and then an interview was conducted to draw their interpretations of survey questions. As mentioned above, the primary purpose of pilot test is to make the questions as precise as possible and to stay clear of misunderstanding.

3.4.5 Conducting Survey

Upon finished all the planning procedures, the survey was then conducted. At the beginning of each questionnaire session, a short briefing was given the to survey participants. Participants were informed that their participation was completely voluntary and their identity will not be kept. If they accept to participate, they have to complete the questionnaire at the end of the session.

In order to reduce bias, the survey was conducted at different session. The survey data collected was analyzed and discussed in Chapter Four.

3.5 Summary

In order to complete the proposed system efficiently and to ensure the effective use of resources, well-defined methodologies are important to provide a well-structured, clear and traceable documentation of the SDLC process. The methodology used in this project is the Waterfall Life Cycle (with iteration) model. Even though Waterfall model is the oldest SDLC model, but it is still widely used.

Several cancer information Websites were reviewed to illustrate some relevant applications developed by previous system developers. Analyses on these systems showed the pros and cons of existing systems. This will helps in developing the Cancer Information System.

Besides that, this chapter also described the procedures involved in planning, designing and conducting the survey. This provides some ideas on how the survey is being carried out.

Chapter Four: System Analysis

4.1 Introduction

The aim of system analysis is to determine the requirements of the proposed system. System analysis should set out what the system should perform rather than how the system performs. The requirements of proposed system were derived through observation of existing systems and from the data collected through survey. This is an important phase of software development because inaccurate requirements specification will cause the errors in requirements to be propagated to the system design and implementation, and finally results a system that will not satisfy user. If inaccuracy is being discovered at the later phase, to correct the problems to fulfill the requirements is costly.

Section 4.2 analyzes the data collected from. Detail comments are also accompanied with the tables and figures. Subsequently, based on the analysis of survey data and observation of the existing systems, the system requirements of the proposed system are discussed.

4.2 Analysis of Survey Data

This section discusses the data collected from 39 copies out of 40 distributed copies. The response rate was 97.5%. Analysis was divided into several parts according to the question in the survey form. The bar charts were generated using Microsoft Excel XP.

4.2.1 Research Participants

First section of questionnaire (Question 1 to Question 3) is for the purpose of collecting the background of the survey participants. Initially, the questionnaire collected information regarding the gender and age group of participants. 39 volunteers participated in the survey of this project. There were 15 (38.46%) men and 24 (61.54%) women. The distribution of survey respondents by gender was summarized in Table 4.1.

Table 4.1: Summary of distribution of respondents by gender

Gender	Number of Respondents	Percentage (%)
Male	15	38.46
Female	24	61.54
Total	39	100.00

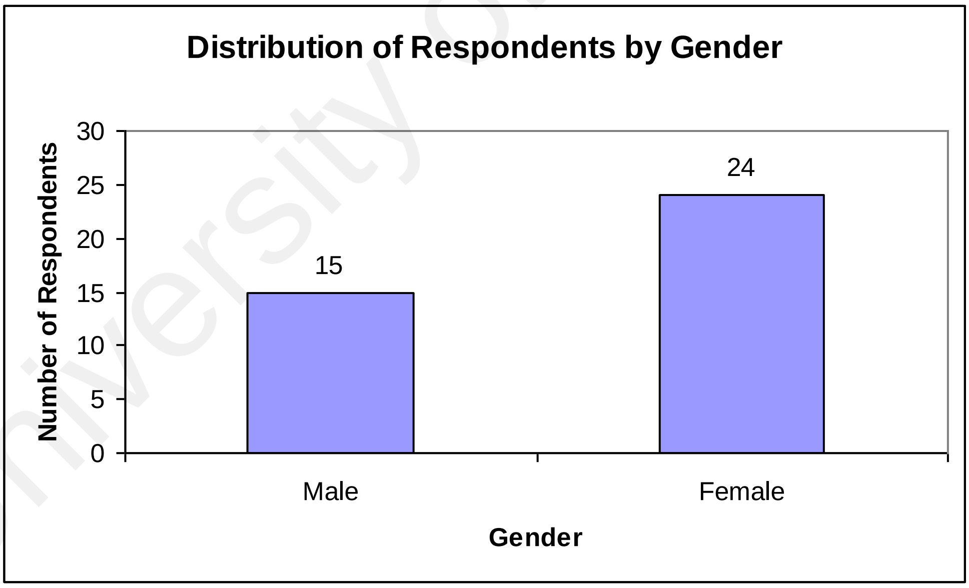


Figure 4.1: Distribution of respondents by gender

Data from questionnaires show that majority of the participants were in the age group of 20-29, which comprised of 34 people, or 87.18% from total respondents. There were 4 (10.26%) respondents in the range of 30-39 and 1 (2.56%) participant in the age group of 40-49. This characteristic suggested that the most of the participants in the survey were young generation. Table 4.2 shows the age group classification of the respondents.

Table 4.2: Summary of distribution of respondents by age group

Age Group	**Number of Respondents**	**Percentage (%)**
below 20	0	0.00
20-29	34	87.18
30-39	4	10.26
40-49	1	2.56
50 and above	0	0.00
Total	39	100.00

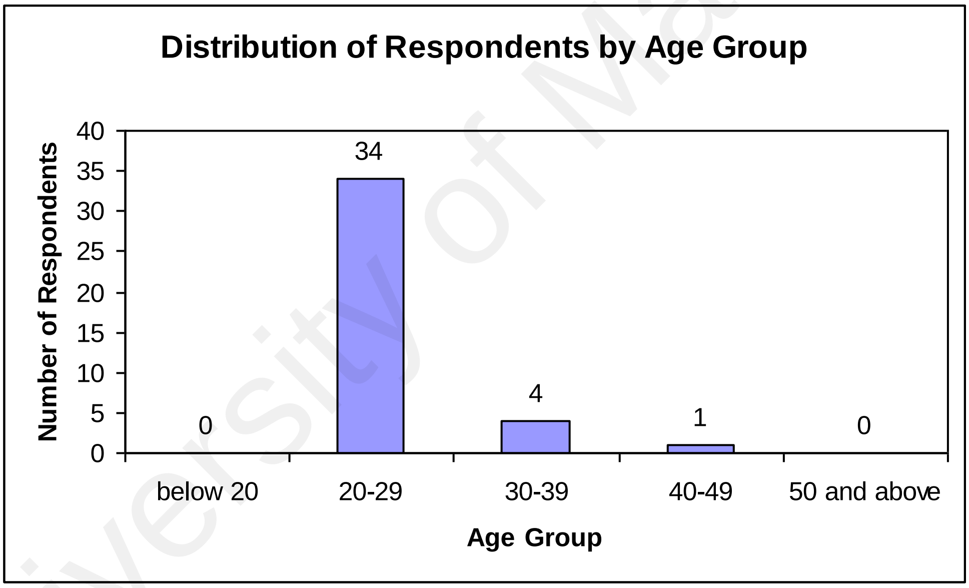


Figure 4.2: Distribution of respondents by age group

In terms of races, 20 (51.28%) out of 39 participants were Malays, 17 (43.59%) Chinese and 2 (5.13%) Indians. Table 4.3 summarizes this information.

Table 4.3: Summary of distribution of respondents by race

Race	**Number of Respondents**	**Percentage (%)**
Malay	20	51.28
Chinese	17	43.59
Indian	2	5.13
Total	39	100.00

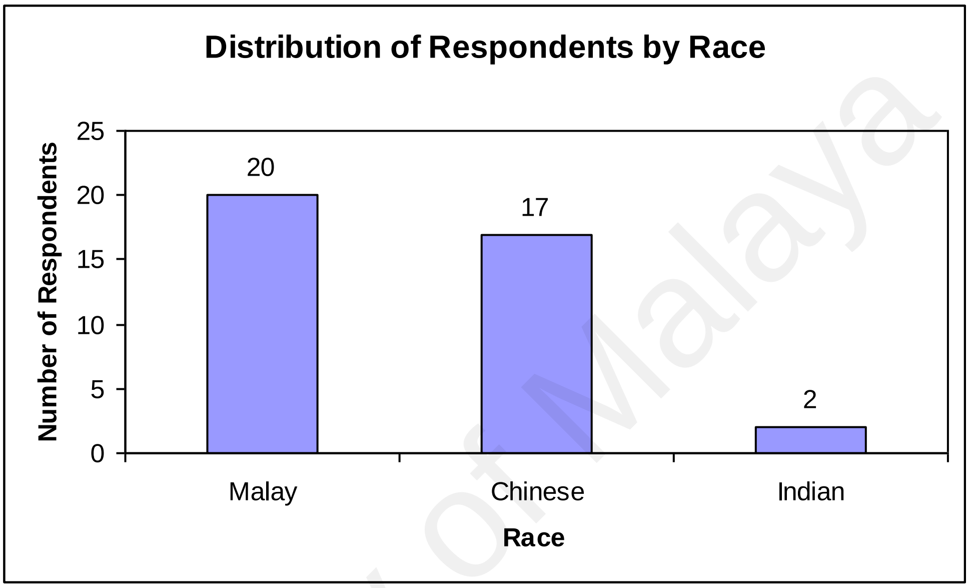


Figure 4.3: Distribution of respondents by race

4.2.2 Time Spend on Accessing Information System

Data collected from questionnaire shows that the majority of respondents spend more than four hours surfing Internet for information daily, which comprises of 15 respondents or 38.46%. 2 respondents (5.13%) spend less than one hour while 4 respondents (10.25%) spend three to four hours daily. Number of respondents that spend one to two hours and two to three hours daily is equal, which is 9 person, or 23.08% each. The results showed that most of the respondents use quite some time in accessing information, which is very positive in building a knowledge-based society.

Table 4.4: Summary of time spend on accessing information system

Hours/Day	**Number of Respondents**	**Percentage (%)**
less than 1 hour	2	5.13
1 to 2 hours	9	23.08
2 to 3 hours	9	23.08
3 to 4 hours	4	10.25
4 hours and above	15	38.46
Total	39	100.00

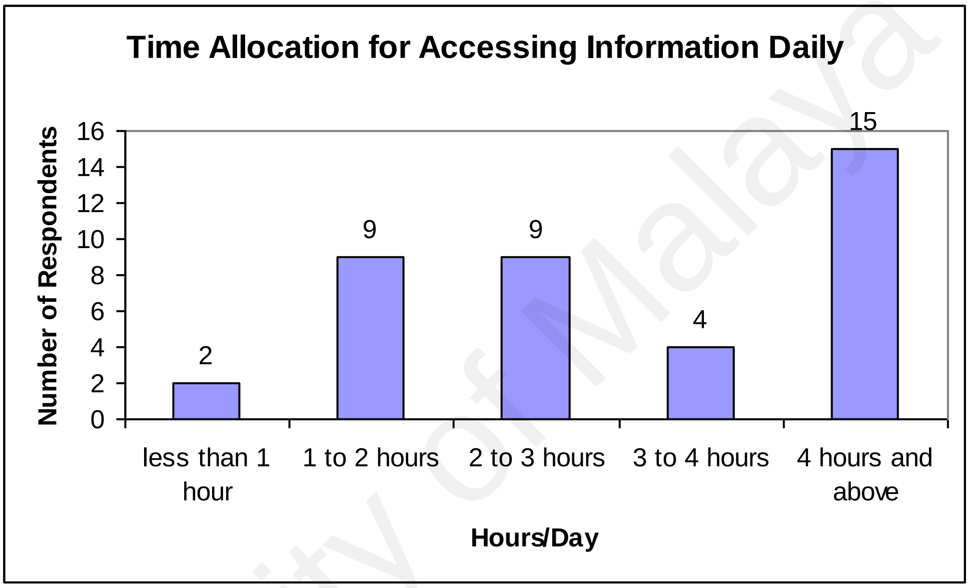


Figure 4.4: Time allocation for accessing information daily

4.2.3 Limitations of Web-based Information System

Question 6 was designed to investigate the limitations of the current Web-based information systems. It is a mix of multiple choice and open-ended question. Four limitations have been suggested. The results were shown in Table 4.4. There was only one limitation that gets majority agreement from respondents (55.85%), that is the current information systems are lacking of Human-Computer Interaction (HCI) approach. This is a common shortfall of Hypertext Markup Language (HTML)-based Web applications. Other limitations such as lack of graphical presentation, difficulty in finding information and lack of contents marked 43.59%, 38.46% and 30.77% votes respectively. None other limitation was suggested by the respondents.

Table 4.5: Limitations of current Web-based information systems

Limitations	**Agree (Person)**	**Percentage (%)**
Lack of human-computer interaction	21	53.85%
Lack of graphical presentation	17	43.59%
Difficulty in finding information	15	38.46%
Lack of contents	12	30.77%

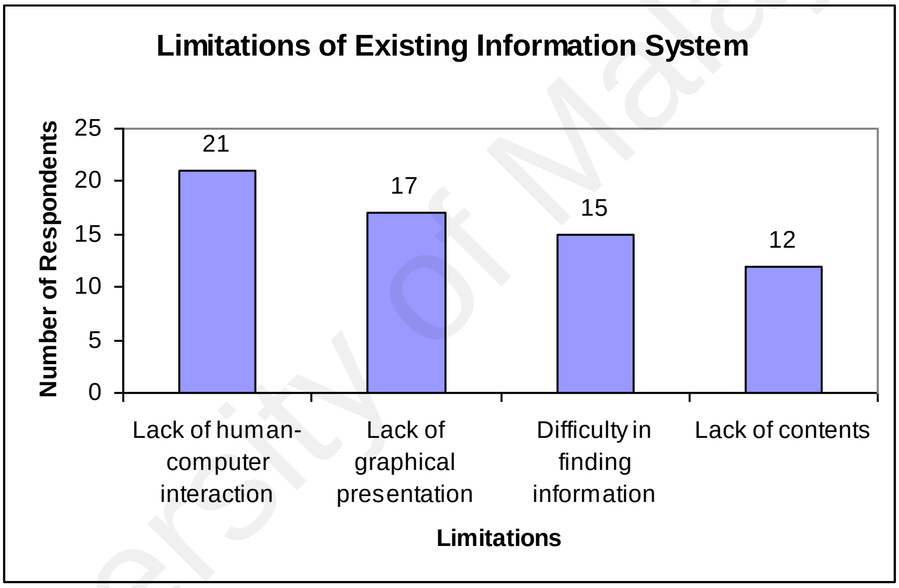


Figure 4.5: Limitations of existing information system

4.2.4 Cancer Awareness among Malaysian

Question 7 aims to provide initial view on the level of cancer awareness of Malaysians. It is a yes/no question. 15 (38.46%) out of 39 respondents have had experience of browsing cancer Websites for cancer information, while the other 24 (61.54%) respondents replied that they do not have the experience. This is very crucial because the percentage of awareness is very low. However, this result may not represent the overall situation. First of all, a larger sample size is needed to provide more accurate result. Secondly, respondents may use other ways to get the information, such as newspaper and magazine.

Table 4.6: Experience of accessing cancer information system

Possess Experience	**Number of Respondents**	**Percentage (%)**
Yes	15	38.46
No	24	61.54
Total	39	100.00

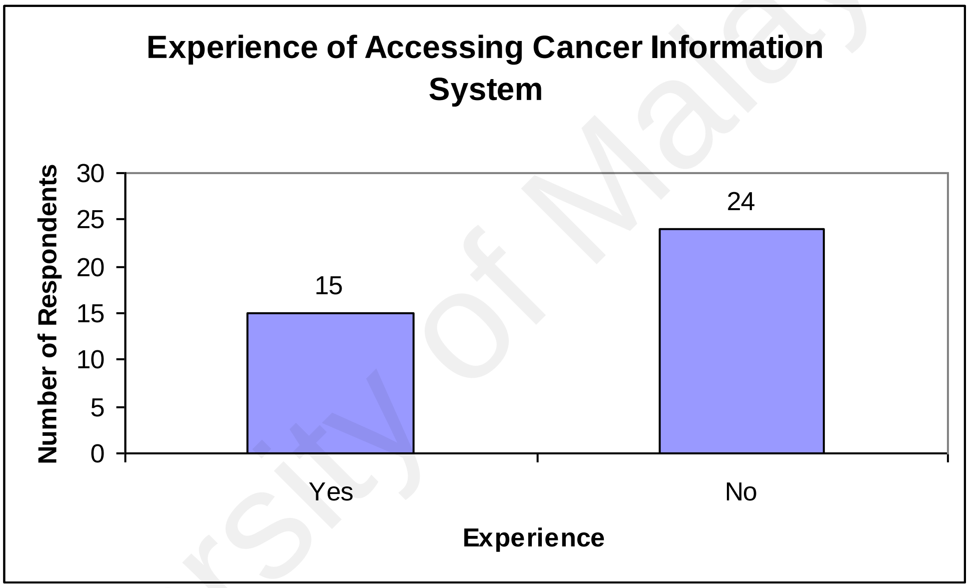


Figure 4.6: Experience of accessing cancer information system

4.2.5 Local or Foreign Cancer Information System

As been expected, among 15 respondents who have surfed online cancer information, 14 (93.33%) of them have never explored any Malaysia-based cancer information system, with 1 (6.67%) respondent had no knowledge of where the cancer information system is being developed. The situation may because of there is no local cancer information system available or local cancer information system is not on the top hit lists of search engines. During the requirements analysis phase of this project, only one local cancer information system was found, which has been developed by National

Cancer Council (MAKNA). Nevertheless, its contents were limited to be considered as information system. So, even though user notices its existence, they will not be satisfied by the limited contents of the Website and eventually will search for other Websites, probably from other country. Table 4.7 summarizes the survey data.

Table 4.7: Origin country of cancer information system

Originate Country	**Number of Respondents**	**Percentage (%)**
Malaysia	0	0.00
Foreign country	14	93.33
No idea	1	6.67
Total	15	100.00

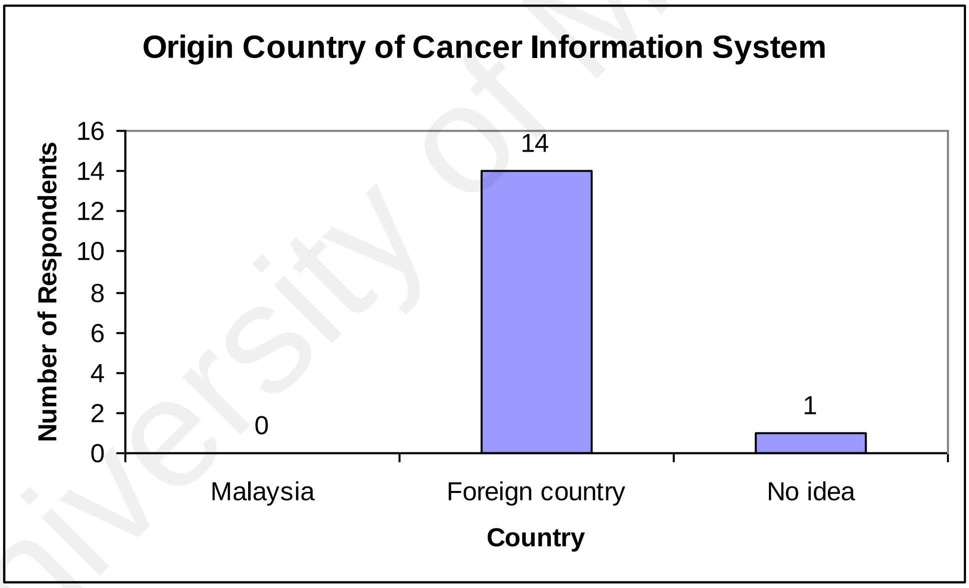


Figure 4.7: Origin country of cancer information system

4.2.6 Feasibility of Local Cancer Information System

Both Question 9 and Question 11 investigated the feasibility of this project. Question 9 studied the necessity to have a cancer information system (can use any technique, not necessary information retrieval techniques) that provides localized cancer information, while Question 11 examines the portion of respondents that willing to use the proposed system that will be built using information retrieval (IR) technique. Both were yes/no questions. Table 4.8 summarizes the survey data. 37 (94.87%) respondents consider that it was necessary to have local cancer information system and 36 (92.31%) of respondents willing to use IR system to help them search for cancer information.

Table 4.8: Feasibility of local cancer information system

Criterion	Agree (Person)	Percentage (%)
Feasibility of local cancer information system	37	94.87
Feasibility of IR technique	36	92.31

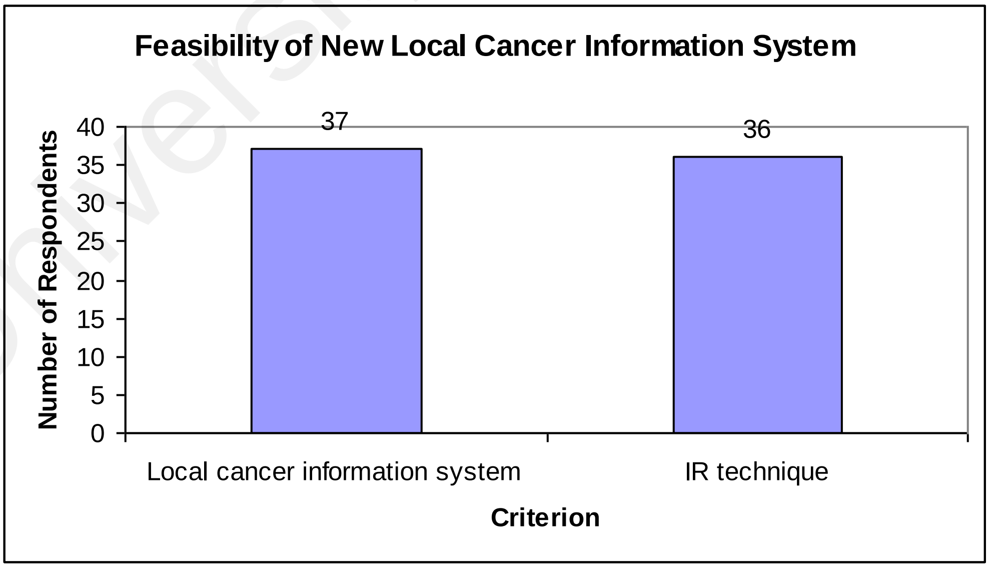


Figure 4.8: Feasibility of new local cancer information system

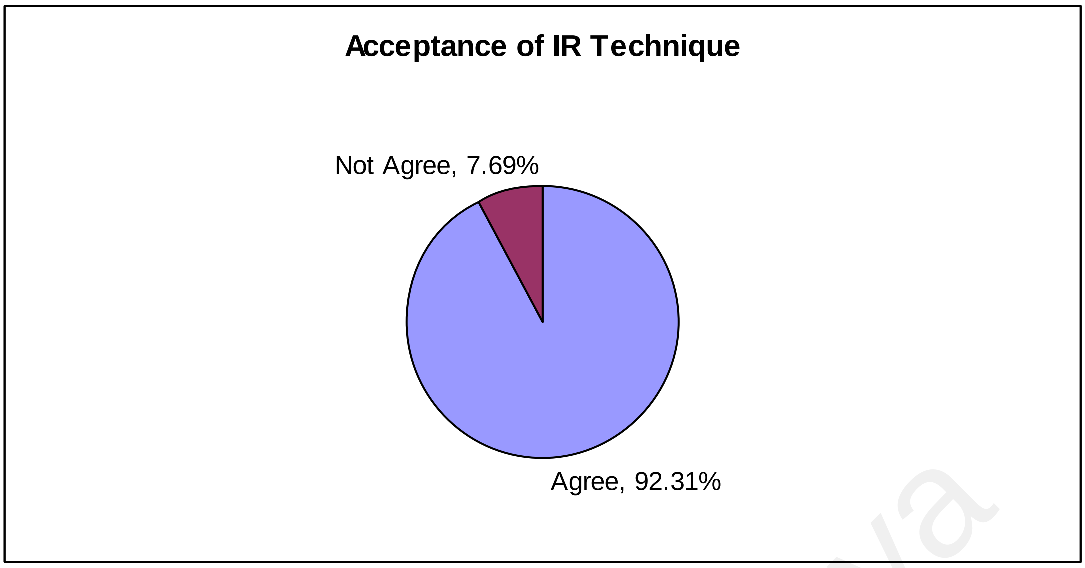


Figure 4.9: Acceptance of information retrieval technique

4.2.7 Contents Coverage of Proposed System

Question 10 in the survey was designed to collect opinions from participants on the types of contents that need to be included in the proposed system. This is a mix of open-ended and multiple selection question. Statistics, introduction of cancer, causes, symptoms, diagnosis, treatment and prevention were accepted by 32 (82.05%), 32 (82.05%), 36 (92.31%), 36 (92.31%), 33 (84.62%), 36 (92.31%) and 37 (94.87%) respondents over size of 39, respectively. Some respondents suggested the following contents are necessary: information of operation center (1 respondent), experience of cancer patient (3 respondents), success story from cancer survival (3 respondents), misunderstanding about cancer (1 respondent), caring for cancer patient (1 respondent) and types of cancer (1 respondent). The results were summarized in Table 4.7.

Table 4.9: Expected contents of proposed system

Contents		Number of Respondents	Percentage (%)
Statistics		32	82.05
Cancer introduction		32	82.05
Causes		36	92.31
Symptoms		36	92.31
Diagnosis		33	84.62
Treatment		36	92.31
Prevention		37	94.87
Others	Operation center	1	-
	Experience sharing	3	-
	Success story	3	-
	Misunderstanding about cancer	1	-
	Caring for patient	1	-
	Types of cancer	1	-

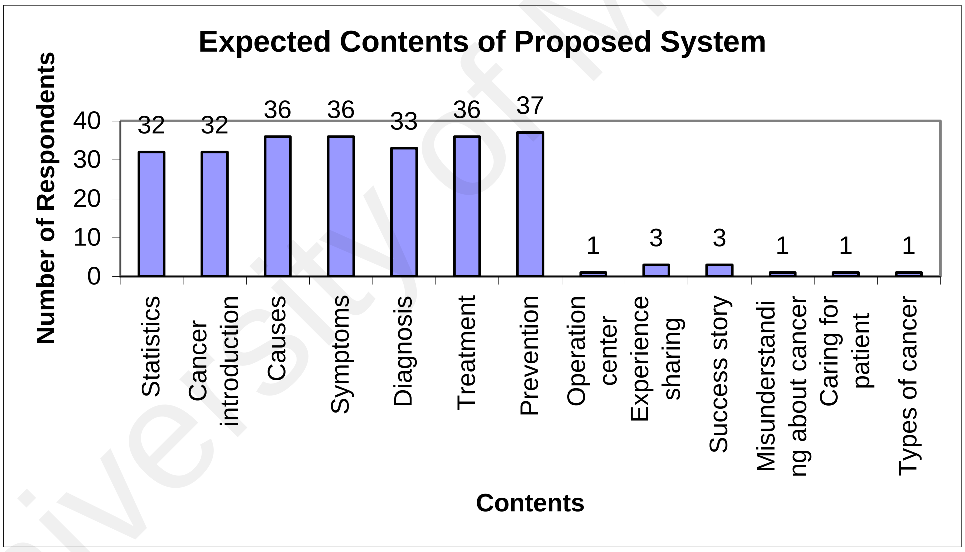


Figure 4.10: Expected contents of proposed system

4.2.8 Survey Discussion

All the major objectives of survey were accomplished. Firstly, responses from survey showed that the development of Malaysia-based cancer information system by using IR techniques is feasible. 37 out of 39 respondents reflected that it is necessary to have Malaysian-based cancer information system. Further more, from the 37 respondents, 36 of them considered that integrating the system with information search engine are crucial to help user to do information searching. The results serve as a huge motivation to the development of this project. Secondly, the survey also collects respondents' opinion on the topics that should be included in the system. Besides the pre-identified type of information, respondents also suggested some information that they think should be available in the system. In addition, observation on information provided by existing cancer information system and availability of information will also influence the final outcome of the results.

4.3 System Requirements

System requirements can be broken up into two categories, namely functional and non-functional requirements. Functional requirements are statements of services that the system should provide and behavior of the system under certain circumstances (Sommerville, 1998). They are often referred as system's functionalities. All services required by the user should be defined under functional requirements. Non-functional requirements concern how well the system provides the functional requirements.

4.3.1 Functional Requirements

Functional requirements describe what a system should do. Based on the results from requirements analysis phase, the following functional requirements have been identified for the proposed Cancer Information System:

- The proposed system shall process query or keywords input by user and retrieve maximum of six best matched documents and displays the results in a list. If no documents were found after the search, message "No result was found" is displayed.

- The proposed system shall provide conventional HTML hyperlinks for all the topics of cancers included in the system. This provides an alternative way of accessing information in addition to the search function that has been developed by using IR techniques.

4.3.2 Non-Functional Requirements

Non-functional requirements define constraints on the service or functions offered by a system (Sommerville, 1998). Non-functional requirements are as important as functional requirements and must be complied with what in order to make sure that the system operates properly. Numbers of non-functional requirements have been set for the proposed system:

- Contents

  The system only covers two types of cancer: breast cancer and lung cancer, and "general cancer". For each of the cancer, the major topics included are introduction, types of cancer, statistics, risk factors, causes, symptoms, prevention, detection, treatment, side effects and staging. Some additional information will be added if applicable.

- Usability

  The system should be convenient and practical to use. Sometimes, it is also called user-friendliness. Usability criteria are easy to learn (learnability) and easy to use.

- Effective

  The system should reflect what it supposed to do.

- Performance

  Performance of the system is measured from two aspects, namely time efficiency and space efficiency. The system should produce an answer in reasonable time and uses a reasonable amount of storage space.

- Portability

  The system should be able to run on computers that using different operating system.

- Reliability

  The system should be reliable in performing its tasks without errors.

4.4 Summary

The data collected from the survey had been analyzed. Results of analysis provided additional information regarding the topics being studied. This information also helped in determining the functional and non-functional requirements. Results also showed an encouraging situation where majority of survey participants welcomed a localized cancer information system that is developed by using IR techniques. Finally, this chapter stated the functional and non-functional requirements of the proposed system.

Chapter Five: System Design

5.1 Introduction

The system design takes place after the system requirements have been determined. System design is a creative process that transforms problems into solutions by building the architecture for software. Pressman (2001) asserted that system design covers several processes such as identifying the software architecture, major components of the system and detailing what they are to accomplish, establishing the interfaces among those components and designing the data for the system to satisfy specified requirements.

Initially, this chapter describes the structural architecture of the proposed Cancer Information System. Next, the functionalities performed by every module are discussed. This chapter also discusses the design of the documents collection and interface design.

5.2 Structural Architecture

Structural architecture design concerns with decomposition of a system into a couple of interacting modules or components. Figure 5.1 shows the structural architecture of proposed Cancer Information System.

As shown in Figure 5.1, the Cancer Information System comprises of three modules: Document Indexing Module, Question Analysis Module and Document Retrieval Module. The Document Indexing Module indexes all the documents in documents collection that are stored in local filesystems and generates a set of index files. Filesystem defines the way files are named, stored, organized and accessed in a disk drive. All the documents in documents collection should be processed by Document Indexing Module before deploying the system.

During the implementation of the system, the system receives query or keywords from user. Input can be in word(s), phrase(s) or sentence(s), and the system treats them as bag-of-word. It is then serves as input into Question Analysis Module. By using bag-of-word approach, a document is treated as an unordered list of words, and there is no relation, such as syntax, exists between each word. For instance, both "*Cow eats grass*" and "*Grass eats cow*" will generate the same query that comprises of three terms; "*cow*", "*eats*" and "*grass*". Therefore, the system will retrieve same set of documents.

The Question Analysis Module is responsible to process the input and then create system query or search index as output. The system query is then becomes the input for Document Retrieval Module. Finally, Document Retrieval Module compares the system query with index built previously and ranks each document. Document Retrieval Module retrieves maximum of six documents with highest score from local filesystems and displays them in a list format so that user can decide which document should be browsed.

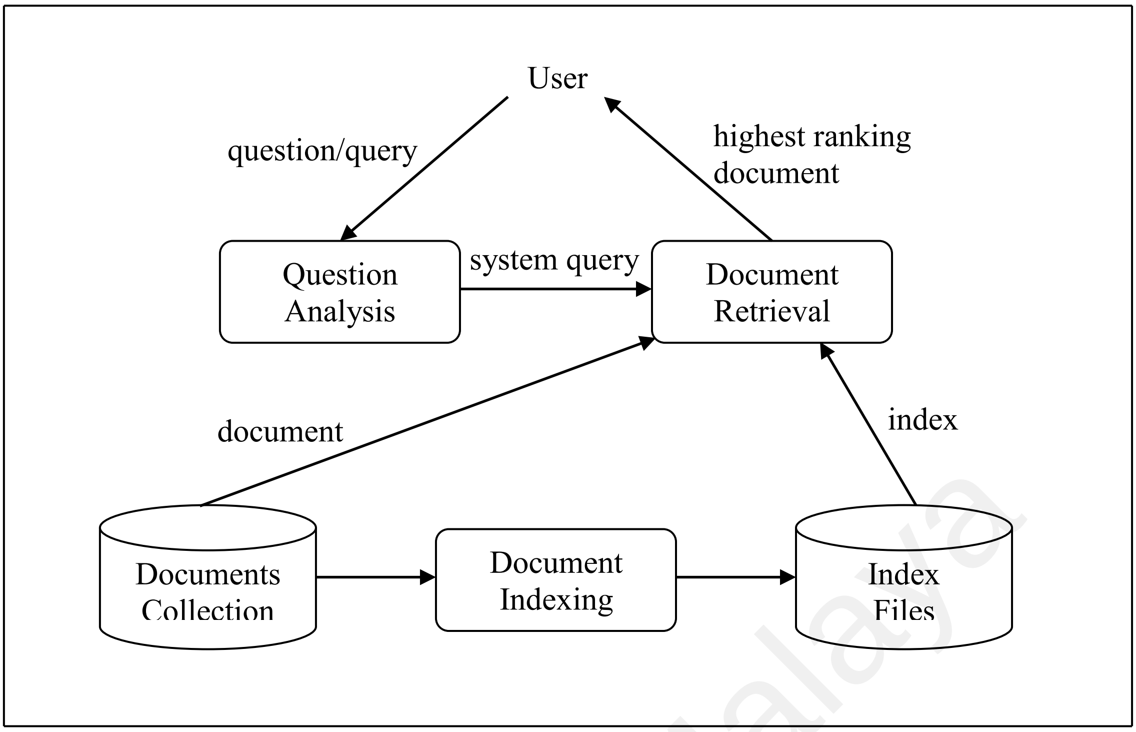


Figure 5.1: Structural architecture of the Cancer Information System

5.3 Functional Architecture

Functional architecture concerns decomposition of a system into a set of interacting functions. Figure 5.2 shows functional diagram of the Cancer Information System.

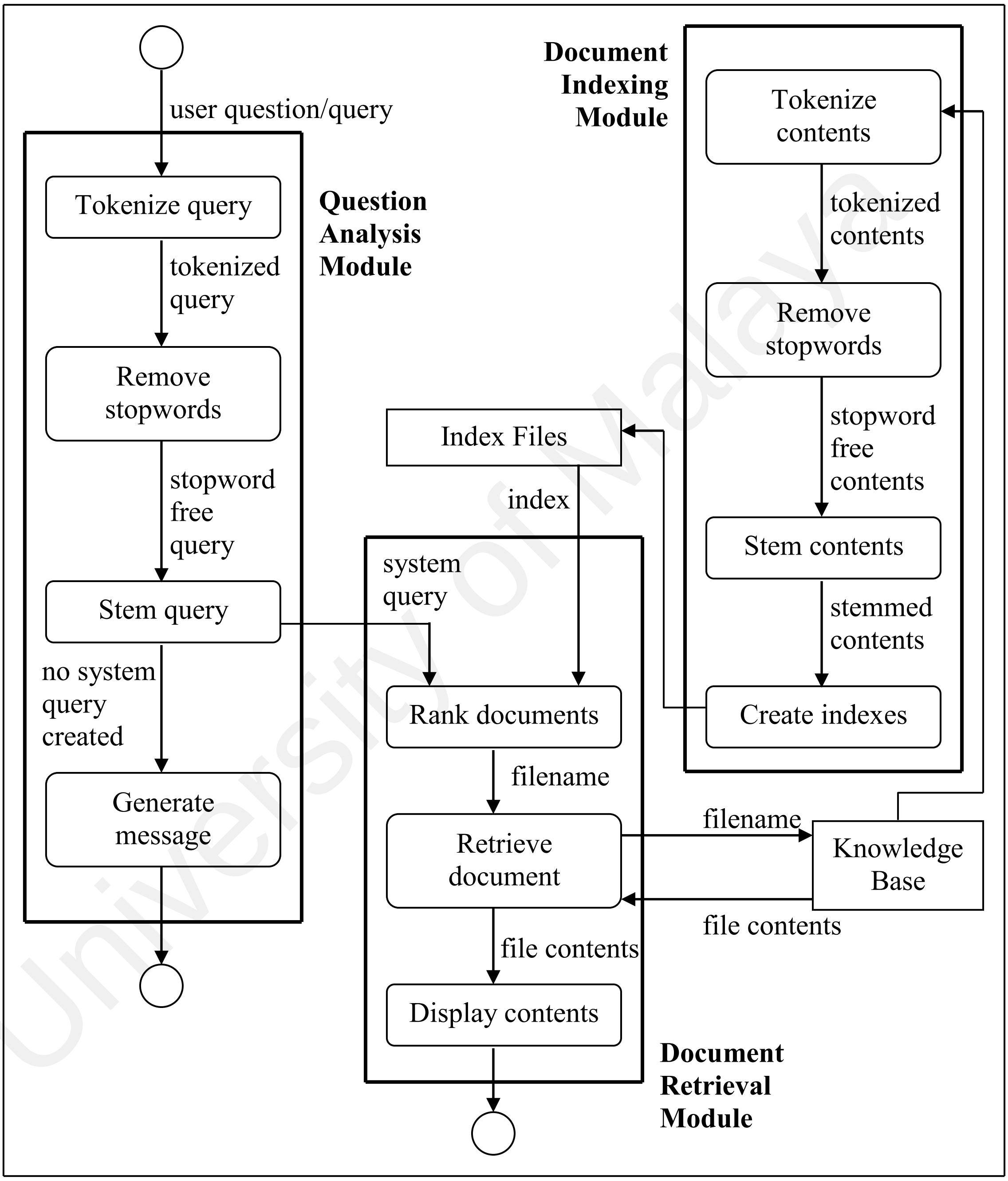


Figure 5.2: Functional diagram of the Cancer Information System

Figure 5.2 shows how data flows through the system and how the output is produced by each method. On the whole, the system receives query or keywords from user and returns maximum of six documents with highest score as search results to user, or produces “No result was found” message if Question Analysis Module is unable to retrieve any document. Note that three first tasks in Question Analysis Module and paragraph indexing module are actually similar. This forces consistent representations of documents contents and system queries, which is crucial during the document ranking process. The following sections describe each module and its tasks in detail.

5.4 Question Analysis Module

The Question Analysis Module of the Cancer Information System aims to analyze the questions posted by user. Several tasks are involved in this phase, including tokenization, stopwords removal (removing poor words) and stemming (reducing the variation in words).

5.4.1 Tokenization

Tokenization is the first task performed in the implementation of the Cancer Information System. During this process, query received by the system is treated as a stream of characters (letters, punctuations, special characters and numerals), and is scanned and divided into a series of consecutive individual units called tokens. According to Covington (2001), a token can be continuous series of letters, a continuous series of digits, or a single special character. Generally, the presence of whitespace or delimiter surrounding a single character or a group of characters defines an explicit token. In the system, the final output of tokenization is a series of words and every word is a single token. It is called system query and is used during subsequent document ranking process.

Subtasks involved during tokenization including conversion of the capitalized words, removing of all punctuations, special characters and Arabic numerals.

First of all, in the processing part, all letters that constitute the query will be converted into lower case. This increases overall performance of the system since all subsequent processes need only to concern lower case text. Then, the system divides text by using delimiter string consisting of a space, a newline, a tab and a carriage return.

For example, all the following instances are treated as one token (Step 1):

side-effect *"staged"* *aren't* *they've* *causes,*

Note that the token *causes,* together with comma trailing it, are treated as one token.

Next, the system checks for the following conditions:

ain't	→	*discarded*
can't	→	*discarded*
won't	→	*discarded*
<root_word>n't	→	*<root_word>*

Hence yielding the following results (Step 2):

side-effect *"stage"* *are* *they've* *causes,*

Note that the first three conditions are checked first. If a token meets one of these conditions, the forth condition will not be executed. Otherwise, the forth condition checking is performed. The abbreviation *n't* stand for *not*, which is also a stopword (will be discussed in Section 5.4.2), therefore the system removes it in advance.

After that, the system checks the occurrence of non-letter text such as punctuations, special characters and numerals, and they will be removed. The output will now become (Step 3):

side effect *stage* *are* *they ve* *causes*

Attention in Step 3 is given to the tokens that have punctuation marks exist in between letters, such as *side effect* and *they've*. The results are *side effect* and *they ve*, respectively. Even though space character exists, they are still treated as a single token during the document ranking process. For example, only documents with the word *side-effect* will be retrieved. Documents that have only the word *side* without *effect* following it or have only the word *effect* without *side* ahead of it will not be retrieved.

5.4.2 Stopwords Removal

Stopwords are commonly used natural language words that have little or no meaning by themselves (Callan, Croft and Broglio, 1995), like articles, prepositions, conjunctions, pronouns and so on. They are used frequently in the documents and can not be used as keywords during searching because they will retrieve many irrelevant records. According to Schauble (1997), indices will be 30% to 50% smaller after removing the stopwords. They are removed to enhance the overall system performance, in terms of time and accuracy. After removing stopwords, all the remaining words are ready for subsequent stemming process.

Normally, stopwords removal is done by listing all words with little meaning that will be discarded during query formulation and indexing. A modified list of stopwords for the proposed system has been created with reference to several existing retrieval engines, such as BioPD (http://www-fog.bio.unipd.it/waishelp/stoplist.html), Onix Text Retrieval Engine (http://www.lextek.com/manuals/onix/stopwords1.html), SMART information retrieval engine (http://www.lextek.com/manuals/onix/stopwords2.html) and THOMAS system that used INQUERY retrieval system (http://thomas.loc.gov/home/stopwords.html). Initially, a list of stopwords was constructed from the references above. Then, with reference to the cancer information documents in the documents collection, those words that are meaningful in the context of cancer disease were removed from the list. The complete list of stopwords is available in Appendix B.

Continue from Step 3 in previous section, the output after stopwords removal (Step 4) is:

side effect *stage* *-discarded-* *-discarded-* *causes*

Both *are* and *they ve* are discarded after Step 4 since they are stopwords. Although *they've* is transformed into a single token *they ve* with a space in between, the system is still able to check *they* and *ve* separately to determine whether they are stopwords or not. This is accomplished by adding *they* and *ve* into list of stopwords and it is identically for the following situations (if *root_word* is not a stopword, else it will be removed):

<root_word>'ve	→	*<root_word>*
<root_word>'ll	→	*<root_word>*
<root_word>'re	→	*<root_word>*
<root_word>'s	→	*<root_word>*
<root_word>'d	→	*<root_word>*

<root_word>'m → *<root_word>*

Once completed Step 4, the remaining tokens are ready for the subsequent Porter stemming processing. These four steps are necessary to transform tokens into acceptable format required by Porter stemmer.

5.4.3 Stemming

IR faces the problem of using free text for query formulation, indexing, document matching and retrieval. This is because various variations in word forms are inevitable (Lennon et al., 1981). However, the IR society has constructed a fairly good remedy to overcome this problem, namely stemming.

Stemming is the process of removing affixes (prefixes and suffixes) from words and reducing them to a common root word or canonical form. IR systems use stemming rather than full morphological analysis. This is because IR systems are required to relate forms, but not to analyze them compositionally. Therefore, the root word may not be the linguistic stem. For instance, in Porter Stemmer, the word *damage*, *damages*, *damaged* and *damaging* are all conflated to *damag*. Notice that the common root after stemmed is *damag* but not the linguistic stem *damage*. This is because the aim of stemming is to reduce inflectional and derivational variant forms of the same word into a single root in order to increase recall.

Stemming is a significant process from IR perspective, because it conflates words forms to prevent mismatches that may undermine recall. It normalizes words from both documents and queries so that documents that contain morphological variants of the query terms can be retrieved. Without stemming, documents that contain the query terms may not be found and retrieved. For example, a query term “symptom” will retrieve documents that contain the exact word “symptom” only, but not “symptoms”.

5.4.3.1 Introduction to Stemmers

Stemmer is a tool that implements a stemming algorithm. There are several stemming algorithms, including S stemmer (also called S-removal stemmer) and Paice-Husk stemmer, and two most widely cited and well-known stemming algorithms are Porter stemmer and Lovins stemmer.

S stemmer perhaps is the most simple among the four stemmers stated above and is a good example for illustrating the concept of stemming. It is a basic algorithm that conflates singular and plural word forms that end with “ies”, “es” and “s” (Harman, 1991). This stemmer is applied only to those words that consist of at least three letters. The algorithm of S stemmer is shown in Figure 5.3. Note that the first applicable rule encountered is the only one used.

```
IF a word ends in “ies”, but not “eies” or “aies”
THEN “ies” → “y”
ELSE IF a word ends in “es”, but not “aes”, “ees”, or “oes”
THEN “es” → “e”
ELSE IF a word ends in “s”, but not “us” or “ss”
THEN “s” → NULL
```

Figure 5.3: The algorithm of S stemmer (Harman, 1991)

Porter stemmer is a stemming algorithm that strips about 60 suffixes without the need for a lexicon (no auxiliary files for the suffixes and their accompanying exception list) (Harman, 1991). It uses multi-step approach in which every step may either transforms the root of word or removes certain inflectional or derivational short suffixes rather than a single removal of the longest-match suffix compared to Lovins stemmer.

In the Lovins stemmer, the longest possible suffix is identified first and the remaining letters (at least two) are checked against an exception list for the given suffix. If the remaining letters pass the check, a cleanup process will be triggered to produce the proper word ending. The algorithm consists of over 260 possible suffixes, a large exception list and the cleanup rules (Harman, 1991).

5.4.3.2 Comparison of Different Stemming Algorithms

Many researches and experiments have been done to test the various stemming algorithms, in terms of their strength, effectiveness and aggressiveness, and their roles in improving the retrieval performance. Experiments carried out by Fuller and Zobel (1998) test the retrieval effectiveness of three stemmers. Two metrics used are number of correct conflations found and the percentage of correctness over total number of attempts. Results showed that S stemmer, Porter stemmer and Lovins stemmer found 61%, 89% and 90% of the correct conflations by term frequency respectively. The accuracy for S stemmer is 99.9%, Porter stemmer is 97% and Lovins stemmer is 86%. Even though S stemmer has accuracy of near 100%, but it has only 61% coverage of potential conflations. Therefore, Fuller and Zobel (1998) concluded that Porter is the best among three of the stemmers.

Paice (1994) used several metrics to access stemming performance of Porter stemmer, Lovins stemmer and Paice-Husk stemmer:

- *desired merge total* (DMT);
- *desired non-merge total* (DNT);
- *global desired merge total* (GDMT);
- *global desired non-merge total* (GDNT);
- *unachieved merge total* (UMT);
- *global unachieved merge total* (GUMT);
- *understemming index*(UI);
- *wrongly-merged total* (WMT);
- *global wrongly-merge total* (GWMT);
- *overstemming index* (OI);
- *stemming weight* (SW); and
- *error rate relative to truncation* (ERRT).

The evaluation is based on counting the actual understemming and overstemming errors committed during stemming of word samples. The results of the research are as following:

UI(Porter) > UI(Lovins) > UI(Paice-Husk)

OI(Paice-Husk) > OI(Lovins) > OI(Porter)

SW(Paice-Husk) > SW(Lovins) > SW(Porter)

ERRT(Lovins) > ERRT >(Porter) > ERRT(Paice-Husk)

The first three results show that Paice-Husk is a heavy stemmer, Porter is a light stemmer and Lovins is in between of the two. This is because a heavier stemmer tends to aggressively removes all sorts of ending and commits more overstemming errors. On the other hand, a light stemmer is too gentle and leaves many understemming errors. Paice also suggested that Lovins stemmer is the less accurate stemmer among three of them in research. In term of accuracy, this conclusion matches results from Fuller and Zobel (1998) that Porter stemmer achieves higher accuracy than Lovins stemmer. In term of weight of stemmer, the results also match the research carried out by Frakes and Fox (2003), asserted that Paice-Husk is the strongest stemmer, Lovins stemmer is in second, Porter stemmer is slightly weaker than Lovins and S stemmer is very weak.

Although the results regarding the performance of different stemming algorithms is very consistent, but there are no consensus whether stemming helps in improving retrieval performance. For instance, Harman (1991) concluded that none of the three stemmers (S stemmer, Porter stemmer and Lovins stemmer) help to improve retrieval performance, but Krovetz (1993) has been shown that stemming could produce reliable retrieval improvement especially when the documents are fairly short. Fuller and Zobel (1998) also suggested that stemming is worthwhile. Nevertheless, stemming was and will continue to become a standard process in most IR system (Harman, 1991).

Based on the various comparisons above, Porter stemming algorithm had been implemented in the Cancer Information System because of its effectiveness, accuracy, average in aggressiveness and publicly available.

5.4.3.3 Porter Stemming Algorithm

Porter stemmer is a lexicon-free rule-based stemming algorithm. As stated above, it removes about 60 suffixes in multi-step approach. This section illustrates the Porter stemming algorithm.

A couple of notations and definitions are needed to be addressed before explaining the algorithm. According to Porter (1980), consonant (marked as *c*) is a letter other than A, E, I, O or U, and other than Y preceded by a consonant. Any letter that is not consonant is a vowel (marked as *v*). Assume that *C* represents a sequence of consonants of length greater than 0 (zero) and *V* represents a sequence of vowels of length greater than 0 (zero), then any word or part of word can be written in the form of $[C](VC)^m[V]$. The presence of contents of square brackets is not compulsory. $(VC)^m$ denotes *VC* repeated *m* times and *m* is called *measure* of the word or part of word that it is attached to. Table 5.1 lists some examples of measure of word. For instance, PRIVATE:

PR = *C*

I = *V*

V = *C*

A = *V*

T = *C*

E = *V*

When combining the sequence, *CVCVCV* is obtained and can be simplified to $C(VC)^2V$. Hence, $m = 2$.

Table 5.1: Examples of measure of word (Porter, 1980)

Measure	Examples
$m = 0$	TR, EE, TREE, Y, BY
$m = 1$	TROUBLE, OATS, TREES, IVY
$m = 2$	TROUBLES, PRIVATE, OATEN, ORRERY

Other conditions include the following:

- *S indicates that the word end with the letter S (applicable to other letters).
- *v* indicates that the word contains a vowel.
- *d indicates that the word ends with a double consonant.
- *o indicates that the word ends with consonant-vowel-consonant, where the final consonant can not be W, X or Y.

The rules of transformation are in the format of *(condition) S1 → S2*, which mean that if a word ends with suffix S1 and the stem before S1 (excluded S1) fulfills the stated condition, S1 is replaced by S2. Besides that, only one with the longest matching S1 for the given word is applied for each group of rules.

The first step of the algorithm is divided into three constituent sub-steps. It focuses on removing several inflectional endings such as plurals, past tenses and '-ing' endings.

Table 5.2: Step 1a of Porter Stemming Algorithm (Porter, 1980)

Condition	**Suffix**	**Substitution**	**Example**
–	SSES	SS	caresses → caress
–	IES	I	ponies → poni ties → ti
–	SS	SS	caress → caress
–	S	–	cats → cat

Table 5.3: Step 1b of Porter Stemming Algorithm (Porter, 1980)

Condition	**Suffix**	**Substitution**	**Example**
(m>0)	EED	EE	feed → fee agreed → agree
(*v*)	ED	–	plastered → plaster bled → bled
(*v*)	ING	–	motoring → motor sing → sing

If the second or third rule in Step 1b is successful, Step 1b1 will be performed. Otherwise, the algorithm continues with Step 1c. Letters in bracket in examples in Step1b1 is the suffix that has been removed in Step 1b.

Table 5.4: Step 1b1 of Porter Stemming Algorithm (Porter, 1980)

Condition	**Suffix**	**Substitution**	**Example**
−	AT	ATE	conflat (ed) → conflate
−	BL	BLE	troubl (ed) → trouble
−	IZ	IZE	siz (ed) → size
(*d and not (*L or *S or *Z))	−	single letter	hopp (ing) → hop tann (ed) → tan fall (ing) → fall hiss (ing) → hiss fizz (ed) → fizz
(m=1 and *o)	−	E	fail (ing) → fail fil (ing) → file

Table 5.5: Step 1c of Porter Stemming Algorithm (Porter, 1980)

Condition	**Suffix**	**Substitution**	**Example**
(*v*)	Y	I	happy → happi sky → sky

Step 2 until Step 5 is straightforward process that involves suffix stripping based on a pre-defined list of recognized suffixes.

Table 5.6: Step 2 of Porter Stemming Algorithm (Porter, 1980)

Condition	Suffix	Substitution	Example
(m>0)	ATIONAL	ATE	relational → relate
(m>0)	TIONAL	TION	conditional → condition rational → rational
(m>0)	ENCI	ENCE	valenci → valence
(m>0)	ANCI	ANCE	hesitanci → hesitance
(m>0)	IZER	IZE	digitizer → digitize
(m>0)	ABLI	ABLE	conformabli → conformable
(m>0)	ALLI	AL	radicalli → radical
(m>0)	ENTLI	ENT	differentli → different
(m>0)	ELI	E	vileli → vile
(m>0)	OUSLI	OUS	analogousli → analogous
(m>0)	IZATION	IZE	vietnamization → vietnamize
(m>0)	ATION	ATE	predication → predicate
(m>0)	ATOR	ATE	operator → operate
(m>0)	ALISM	AL	feudalism → feudal
(m>0)	IVENESS	IVE	decisiveness → decisive
(m>0)	FULNESS	FUL	hopefulness → hopeful
(m>0)	OUSNESS	OUS	callousness → callous
(m>0)	ALITI	AL	formaliti → formal
(m>0)	IVITI	IVE	sensitiviti → sensitive
(m>0)	BILITI	BLE	sensibiliti → sensible

Table 5.7: Step 3 of Porter Stemming Algorithm (Porter, 1980)

Condition	Suffix	Substitution	Example
(m>0)	ICATE	IC	triplicate → triplic
(m>0)	ATIVE	–	formative → form
(m>0)	ALIZE	AL	formalize → formal
(m>0)	ICITI	IC	electriciti → electric
(m>0)	ICAL	IC	electrical → electric
(m>0)	FUL	–	hopeful → hope
(m>0)	NESS	–	goodness → good

Table 5.8: Step 4 of Porter Stemming Algorithm (Porter, 1980)

Condition	Suffix	Substitution	Example
(m>1)	AL	–	revival → reviv
(m>1)	ANCE	–	allowance → allow
(m>1)	ENCE	–	inference → infer
(m>1)	ER	–	airliner → airlin
(m>1)	IC	–	gyroscopic → gyroscop
(m>1)	ABLE	–	adjustable → adjust
(m>1)	IBLE	–	defensible → defens
(m>1)	ANT	–	irritant → irrit
(m>1)	EMENT	–	replacement → replac
(m>1)	MENT	–	adjustment → adjust
(m>1)	ENT	–	dependent → depend
(m>1 and (*S or *T))	ION	–	adoption → adopt
(m>1)	OU	–	homologou → homolog
(m>1)	ISM	–	communism → commun
(m>1)	ATE	–	activate → activ
(m>1)	ITI	–	angulariti → angular
(m>1)	OUS	–	homologous → homolog
(m>1)	IVE	–	effective → effect
(m>1)	IZE	–	bowdlerize → bowdler

Step 5 is the last step in Porter stemming algorithm and is divided into two sub-steps as shown in Table 5.9 and Table 5.10.

Table 5.9: Step 5a of Porter Stemming Algorithm (Porter, 1980)

Condition	Suffix	Substitution	Example
(m>1)	E	–	probate → probat rate → rate
(m>1 and not *o)	E	–	cease → ceas

Table 5.10: Step 5b of Porter Stemming Algorithm (Porter, 1980)

Condition	Suffix	Substitution	Example
(m>1 and *d and *L)	–	single letter	controll → control roll → roll

5.5 Document Indexing Module

Indexing is the process of creating an index, while index is a systematic arrangement of entries designed to help user to locate information in a document quickly (Wyman and Harrison, 2005). Indexing helps user to search through a large collection of files for specific information that otherwise seems impossible, or burdensome.

This section discusses the design of Document Indexing Module in the project. Note that the first three tasks of Document Indexing Module were designed identically to Question Analysis Module. This is to ensure that the same word from user and document from documents collection will has the same representation in the index and hence it is searchable and retrievable.

5.5.1 Introduction to Lucene Indexing

The project utilized Lucene classes (introduction to Lucene can be found in Section 6.2.3) to perform document indexing. According to The Apache Software Foundation (2005), there are four key concepts in Lucene, specifically index, document, field and term:

- An index is a sequence of documents
- A document is a sequence of fields.
- A field is a named sequence of terms.
- A term is a string.

Documents are referred to any file that needed to be indexed. A document must be scanned first in order to create a list of postings. Posting is a procedure for adding data or information to an existing record. Each document in an index contains at one or more named fields, depend on the developer when creating the index. Each field holds sequence of terms that is either queried against or retrieved from the index during search. The same term in different fields is considered different term. Therefore, developers have to specify both the name of field and the terms during indexing and searching. Finally, every term is a string (series of letters treated as single unit).

In keyword-based search engine such as Lucene, the major consideration is creating and maintaining an inverted index (Goetz, 2000). It is called inverted index because it lists the documents that contain a term, as opposed to the natural relationship where documents list terms (The Apache Software Foundation, 2005). Lucene maintains indexes that composed of multiple independent sub-indexes, or segments. Once Lucene indexed a new document, it creates a new index segment and merges it with larger segments periodically. Each Lucene index segment records the following information (The Apache Software Foundation, 2005):

- Field names – Set of field names created during indexing.
- Stored field values – Stores information about the indexed document and it will be used during searching.
- Term dictionary – A dictionary containing all terms used in all of the indexed fields of all of the documents. It also saves the number of documents which contain the term, pointers to the term's frequency and proximity data.
- Term frequency data – Records all the documents that contain that term and the term frequency in that document.
- Term proximity data – Positions of the term locates in each document.
- Normalization factors – A value that is used during scoring.
- Term vectors – A term vector consists of term text and term frequency.

- Deleted documents – An optional file specifying documents that have been deleted.

5.5.2 Design of Document Indexing

As Figure 5.2 shown, the contents of every document went through several processes before they were saved in the indexes, specifically tokenization, stopwords removal and stemming as described previously. After that, the remaining terms was saved with the field name "contents". Besides that, the respective filename was also saved into the index. The filename will be used to retrieve the file during searching.

5.6 Document Retrieval Module

Document Retrieval Module concerns the process of looking up terms in the indexes to find documents where they appear, calculate the score for each document that containing the term, get the contents of the document with highest score and display it. Again, the searching process in this module used classes provided by Lucene.

5.6.1 Scoring in Lucene

Every system that performs IR needs to implement retrieval model that predicts and explains what a user will find relevant given the user query. Generally, models are divided into either Boolean or ranked model. Lucene uses vector space model scoring function (Gupta, 2004), which is classified under ranked model. Ranked models usually consider the number of occurrences of terms in the documents or in the index to rank documents.

The term weighting (ranking) formula used in Lucene is as following (Hatcher and Gospodnetic, 2004b; Aylett, 2004):

$$score\_d = sum\_t(\frac{tf\_q * idf\_t}{norm\_q} * \frac{tf\_d * idf\_t}{norm\_d\_t} * boost\_t) * coord\_q\_d$$

where score_d = score for document d

sum_t = sum for all terms t

tf_q = the square root of the frequency of t in the query q

tf_d = the square root of the frequency of t in d

idf_t = $\log(\frac{numDocs}{docFreq\_t + 1}) + 1$

numDocs = number of documents in index

docFreq_t = number of documents containing t

norm_q = $sqrt(sum\_t((tf\_q * idf\_t)^2))$

norm_d_t = square root of number of tokens in d in the same field as t

boost_t = the user-specified boost for term t (default value of 1.0 was used)

coord_q_d = number of terms in both query and document / number of terms in query

5.7 Documents Collection Design

All documents for the Cancer Information System were stored in local filesystems in a single directory. Each document contained one specific topic about one of the three cancers, specifically “general cancer”, breast cancer and lung cancer. Documents were in HTML file format (.htm extension). For example, Figure 5.4 shows the contents of “Lung Cancer – Introduction” (cancer type = lung, topic = introduction). Every document in documents collection has the same format as show in the figure. From the figure, <b>, <u>, </b>, </u> and <br> are all HTML tags that used to format text when viewing the file in Web browser such as Microsoft Internet Explorer, as show in Figure 5.5.

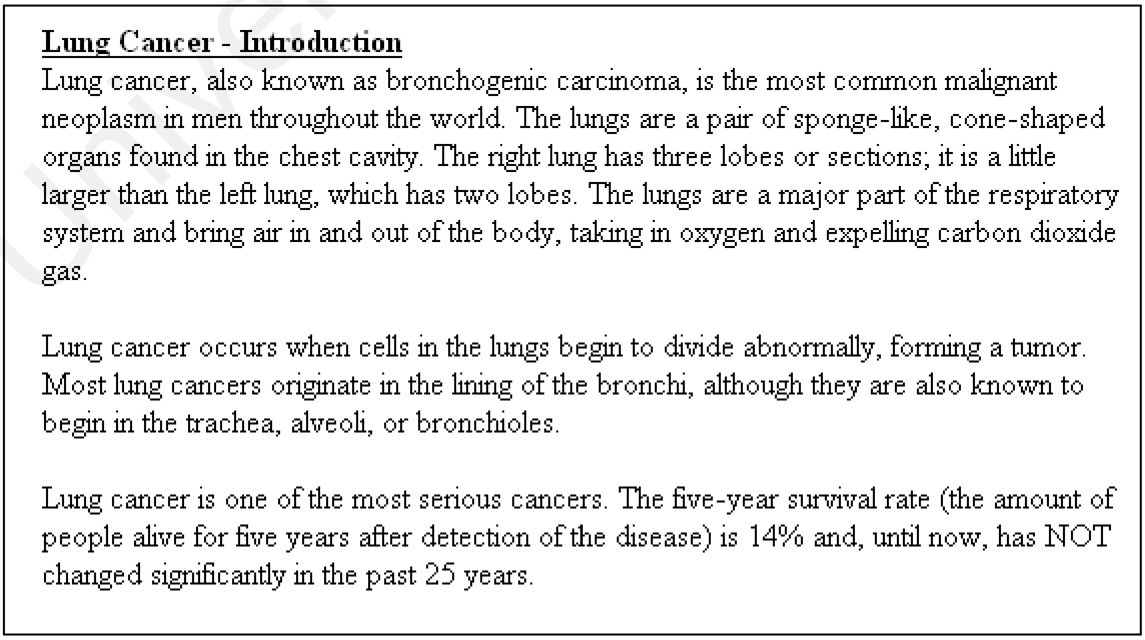
```
<head><title>Lung Cancer - Introduction</title></head>
<b><u>Lung Cancer - Introduction</b></u><br>

Lung cancer, also known as bronchogenic carcinoma, is the most common
malignant neoplasm in men throughout the world. The lungs are a pair of sponge-
like, cone-shaped organs found in the chest cavity. The right lung has three lobes or
sections; it is a little larger than the left lung, which has two lobes. The lungs are a
major part of the respiratory system and bring air in and out of the body, taking in
oxygen and expelling carbon dioxide gas.<br><br>

Lung cancer occurs when cells in the lungs begin to divide abnormally, forming a
tumor. Most lung cancers originate in the lining of the bronchi, although they are
also known to begin in the trachea, alveoli, or bronchioles.<br><br>

Lung cancer is one of the most serious cancers. The five-year survival rate (the
amount of people alive for five years after detection of the disease) is 14% and, until
now, has NOT changed significantly in the past 25 years.
```

Figure 5.4: Contents of “Lung Cancer – Introduction”



Lung Cancer - Introduction

Lung cancer, also known as bronchogenic carcinoma, is the most common malignant neoplasm in men throughout the world. The lungs are a pair of sponge-like, cone-shaped organs found in the chest cavity. The right lung has three lobes or sections; it is a little larger than the left lung, which has two lobes. The lungs are a major part of the respiratory system and bring air in and out of the body, taking in oxygen and expelling carbon dioxide gas.

Lung cancer occurs when cells in the lungs begin to divide abnormally, forming a tumor. Most lung cancers originate in the lining of the bronchi, although they are also known to begin in the trachea, alveoli, or bronchioles.

Lung cancer is one of the most serious cancers. The five-year survival rate (the amount of people alive for five years after detection of the disease) is 14% and, until now, has NOT changed significantly in the past 25 years.

Figure 5.5: Example of file contents layout in Web browser

The documents collection contained a total of thirty four searchable documents (documents that have been indexed and can be searched through querying). The followings summarize the cancer topics provided by the system:

- General cancer

  Introduction, types, statistics, risk factors, causes, symptoms, prevention, detection, treatment, side effect, staging and clinical trial.

- Breast cancer

  Introduction, types, statistics, risk factors, causes, symptoms, prevention, detection, breast self-exam, mammogram, treatment and staging.

- Lung cancer

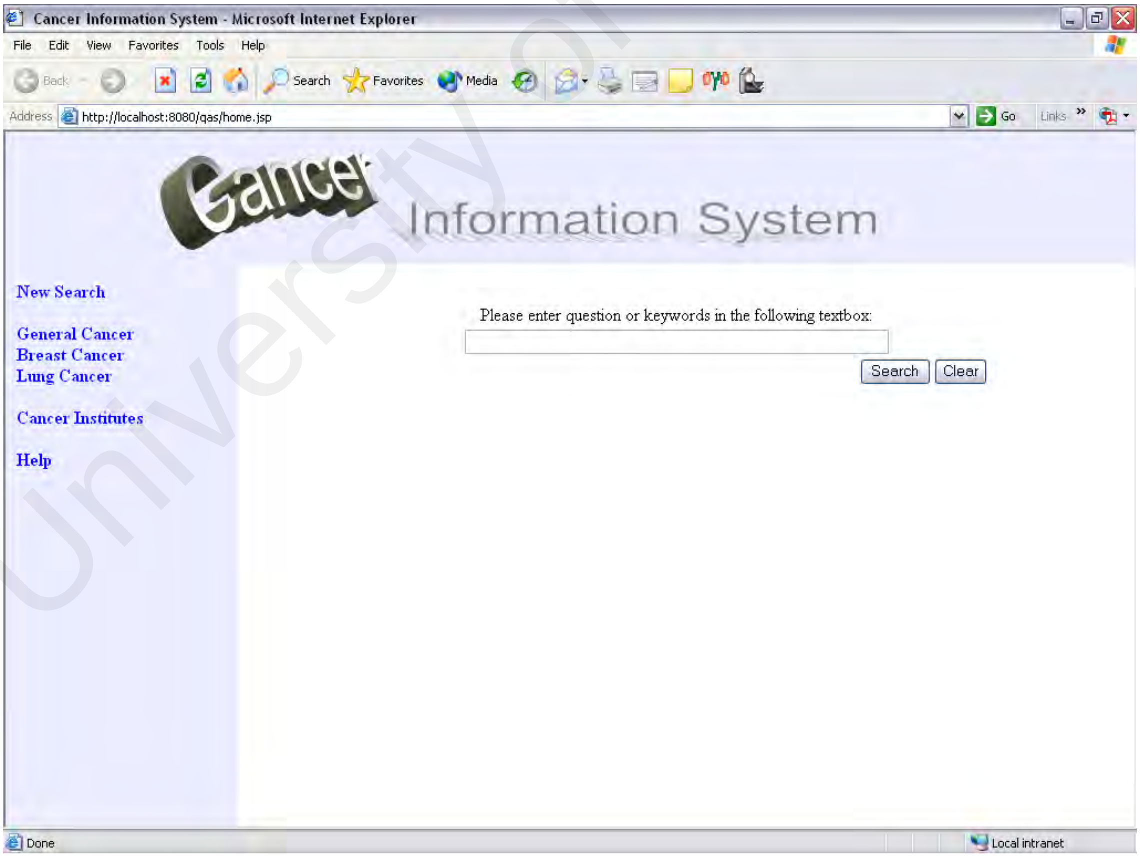
  Introduction, types, statistics, risk factors, causes, symptoms, prevention, detection, treatment and staging.

The contents of documents for the Cancer Information System were hard coded from various existing cancer information Websites. In addition, the information used by the Cancer Information System has been verified and validated by using published articles. This ensures the accuracy of cancer information provided by the system.

In addition, the system also provides user with the names and contacts of local cancer institutes that participated in the Second Report of the National Cancer Registry Malaysia. This information was not indexed and hence can not be searched through querying. However, user can access to this information through conventional Web page browsing.

5.8 Interface Design

This section describes the interfaces design of the Cancer Information System. Interfaces were designed to be as simple as possible so that user can easily learn how to use the system. Figure 5.6 shows the homepage of the system. On the whole, user is able to search for information either through hyperlinks or the search function developed by using IR techniques. The hyperlinks are available through the links on the side bar. Click on the *New Search* link will return to the homepage of the system, *Cancer Institutes* provides lists of local cancer institutes that participated in the Second Report of the National Cancer Registry - Cancer Incidence in Malaysia 2003, while the *Help* provides user some introduction regarding the system.



Cancer Information System - Microsoft Internet Explorer

Address http://localhost:8080/qas/home.jsp

Please enter question or keywords in the following textbox:

Figure 5.6: Homepage of the Cancer Information System

When user moves the mouse cursor onto *General Cancer*, *Breast Cancer*, or *Lung Cancer* on the side bar, a respective drop-down menu will be displayed, as shown in Figure 5.7.

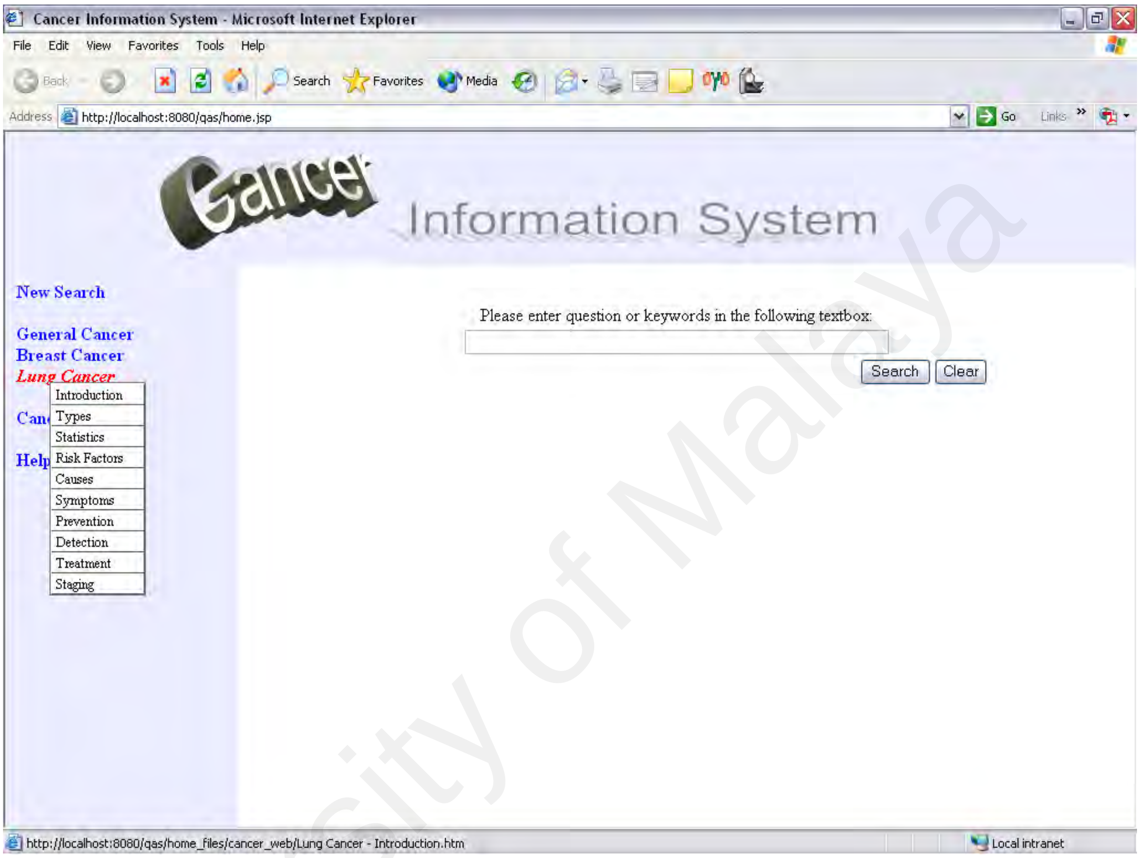
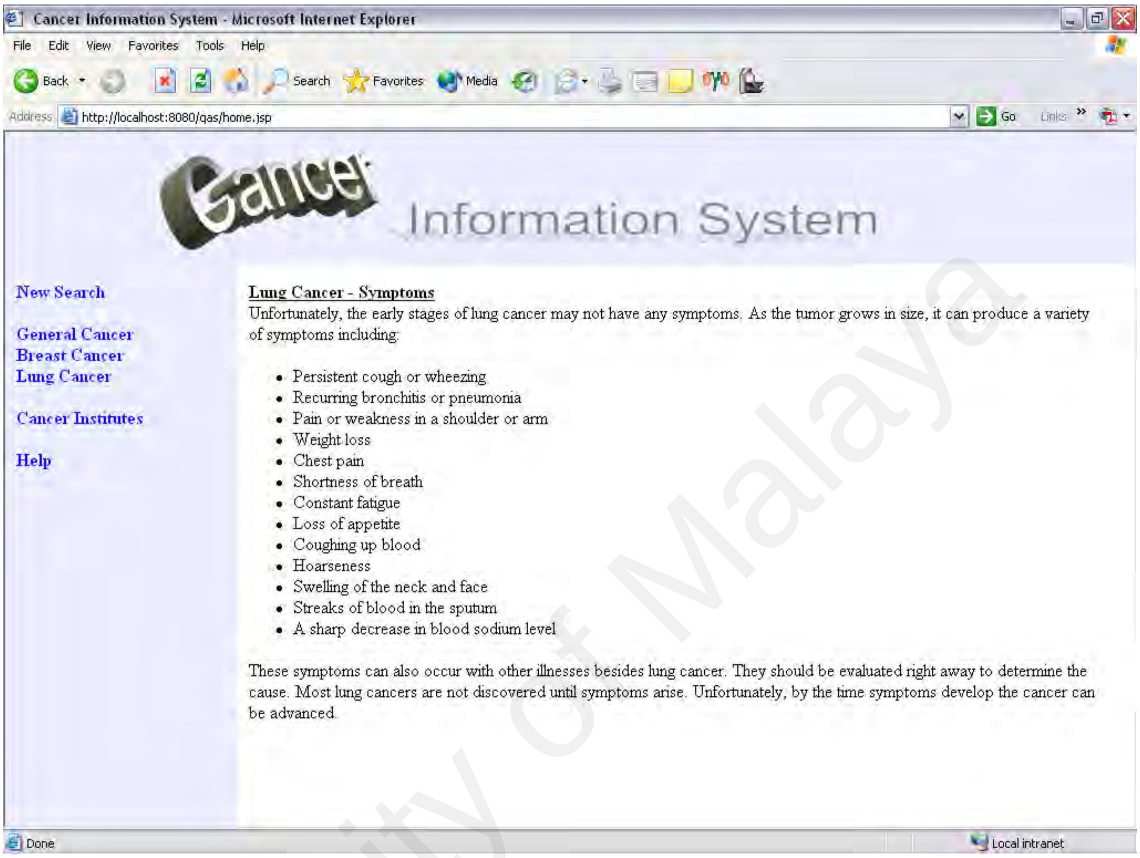

Cancer Information System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/qas/home.jsp
Go Links
Cancer Information System
New Search
General Cancer
Breast Cancer
Lung Cancer
Can
Help
Introduction
Types
Statistics
Risk Factors
Causes
Symptoms
Prevention
Detection
Treatment
Staging
Please enter question or keywords in the following textbox:
Search
Clear
http://localhost:8080/qas/home_files/cancer_web/Lung Cancer - Introduction.htm
Local intranet

Figure 5.7: Drop-down menu for lung cancer

If user clicks on one of the topic in the drop-down menu, the contents of the topic will be displayed in the main window, as shown in Figure 5.8 (symptoms of lung cancer).


Cancer Information System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/qas/home.jsp
Go Links
Cancer Information System
New Search
General Cancer
Breast Cancer
Lung Cancer
Cancer Institutes
Help
Done
Local intranet

Lung Cancer - Symptoms

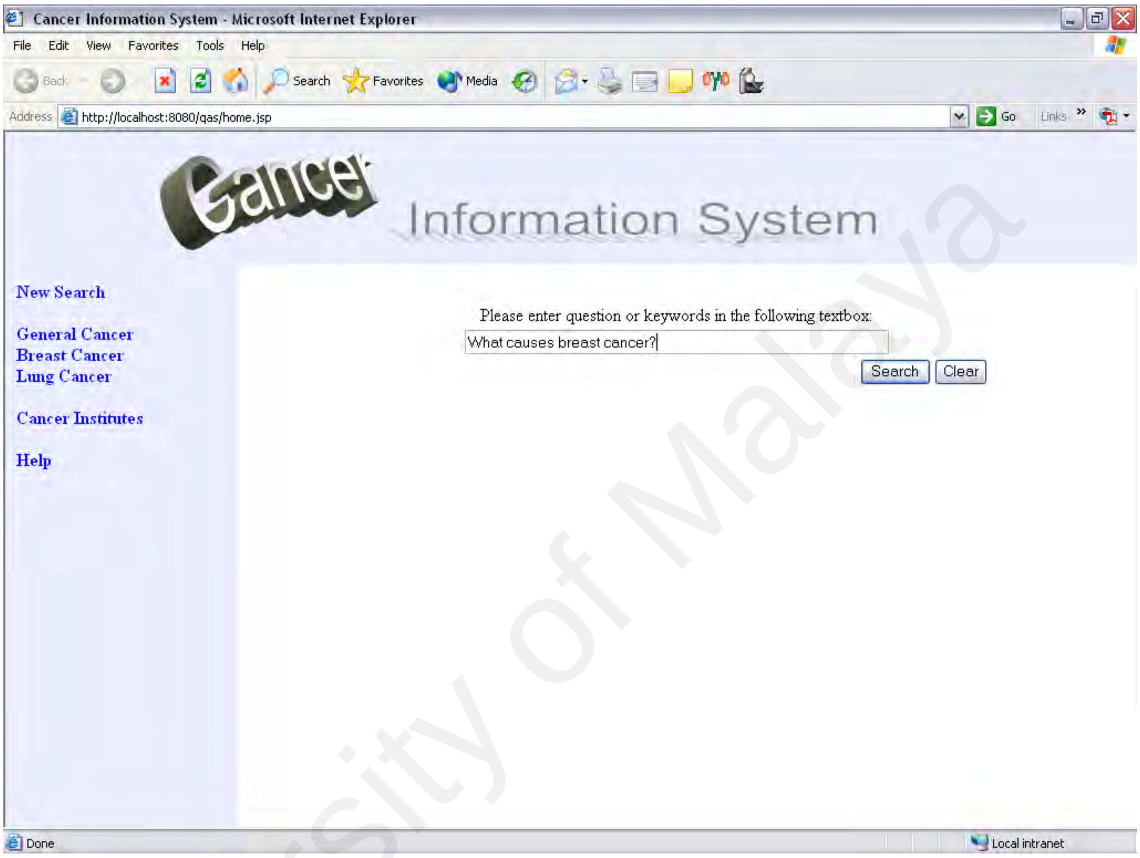
Unfortunately, the early stages of lung cancer may not have any symptoms. As the tumor grows in size, it can produce a variety of symptoms including:

- Persistent cough or wheezing
- Recurring bronchitis or pneumonia
- Pain or weakness in a shoulder or arm
- Weight loss
- Chest pain
- Shortness of breath
- Constant fatigue
- Loss of appetite
- Coughing up blood
- Hoarseness
- Swelling of the neck and face
- Streaks of blood in the sputum
- A sharp decrease in blood sodium level

These symptoms can also occur with other illnesses besides lung cancer. They should be evaluated right away to determine the cause. Most lung cancers are not discovered until symptoms arise. Unfortunately, by the time symptoms develop the cancer can be advanced.

Figure 5.8: Example layout of the file contents

Figure 5.9 shows the screenshot with a query to the system, namely "What causes breast cancer?" Clicking on *Search* button will trigger the system to start looking for the most relevant document for the query.

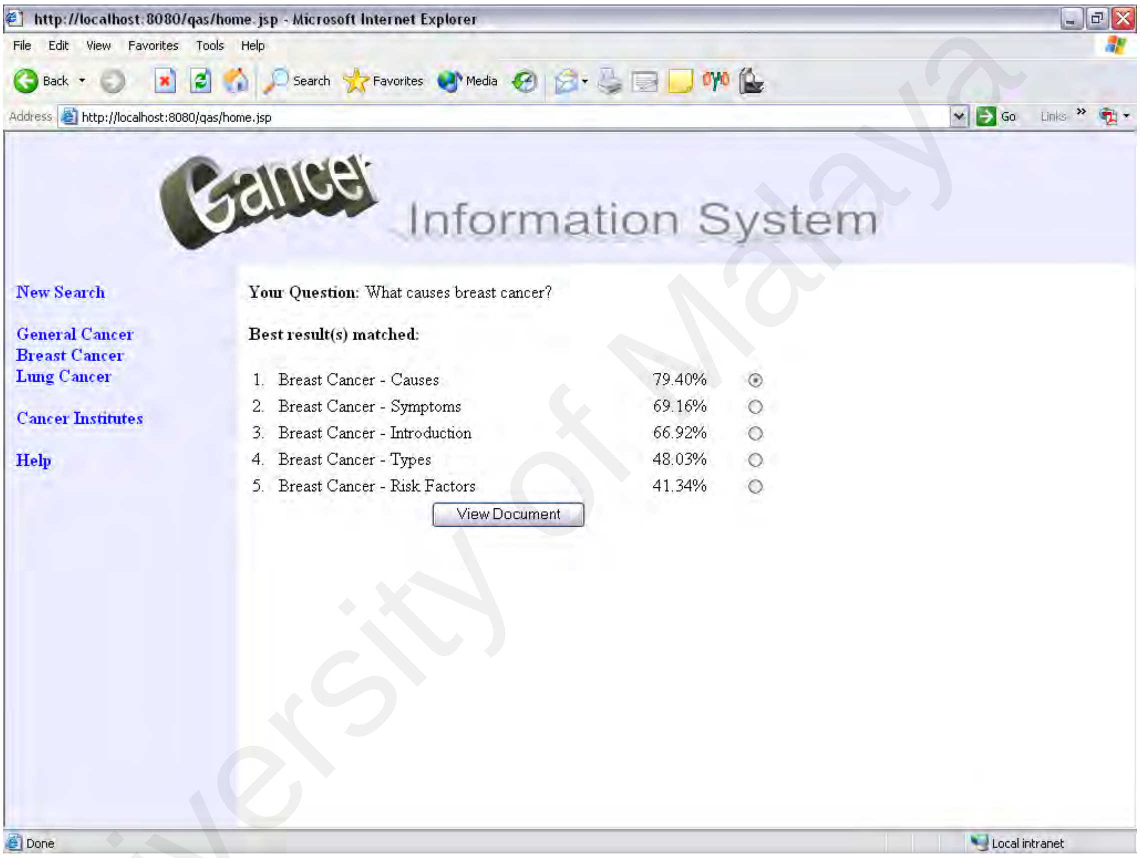

Cancer Information System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/qas/home.jsp
Go Links
Cancer
Information System
New Search
General Cancer
Breast Cancer
Lung Cancer
Cancer Institutes
Help
Search
Clear
Done
Local intranet

Please enter question or keywords in the following textbox:

What causes breast cancer?

Figure 5.9: Querying the system by searching

Subsequently, the system will return maximum of six search results in a list as shown in Figure 5.10. The results are displayed in the descendent order of similarity. The system displays each result with the title of document and its associated degree of similarity. User can then browse any of these documents which they think possibly contain the information of interest.


http://localhost:8080/qas/home.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/qas/home.jsp
Go Links
Cancer Information System
New Search
General Cancer
Breast Cancer
Lung Cancer
Cancer Institutes
Help
View Document
Done
Local intranet

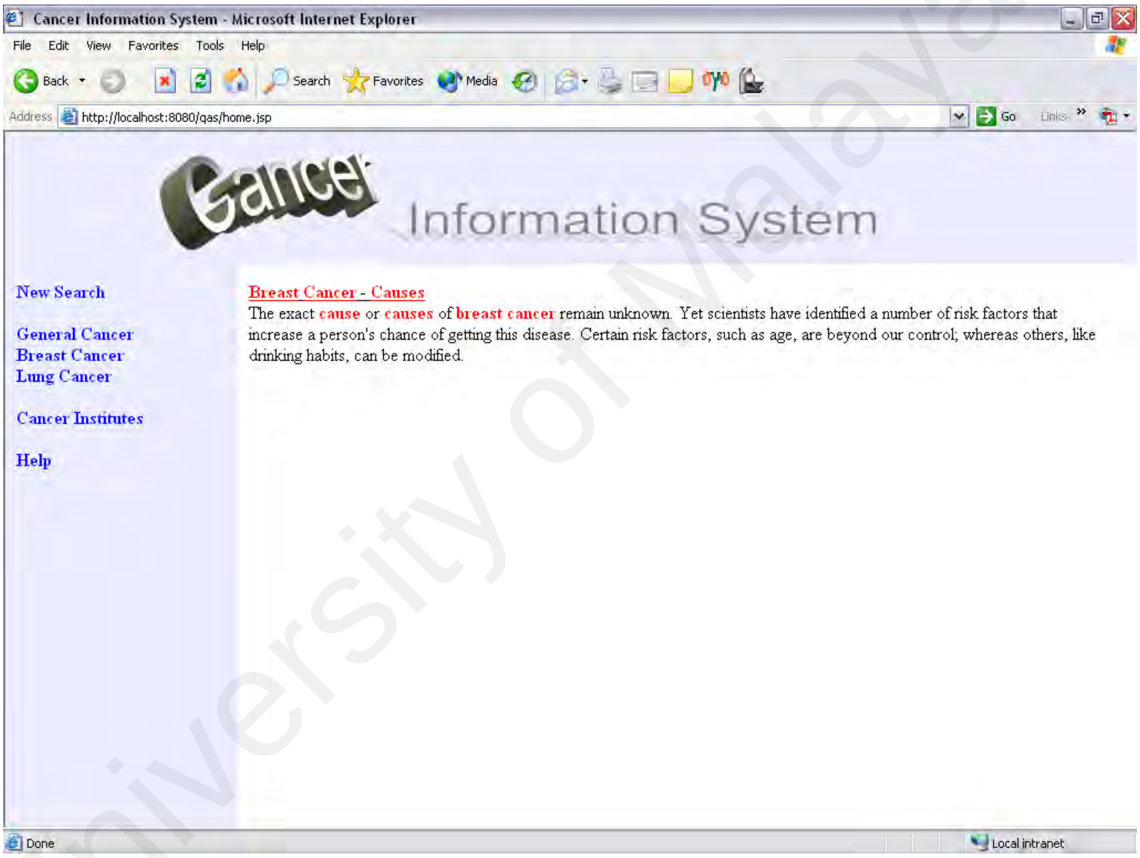
Your Question: What causes breast cancer?

Best result(s) matched:

1.	Breast Cancer - Causes	79.40%	●
2.	Breast Cancer - Symptoms	69.16%	○
3.	Breast Cancer - Introduction	66.92%	○
4.	Breast Cancer - Types	48.03%	○
5.	Breast Cancer - Risk Factors	41.34%	○

Figure 5.10: List of search results for query in Figure 5.9

Continuation from Figure 5.10, if user selected the first document, which is *Breast Cancer – Causes*, the system will then retrieve the contents of the document. From the query given by user, the system generates system queries through a series of processes defined within Question Analysis Module. System queries are used during the matching process and all system queries will be highlighted in the retrieved document to help user locates them quickly as shown in Figure 5.11.

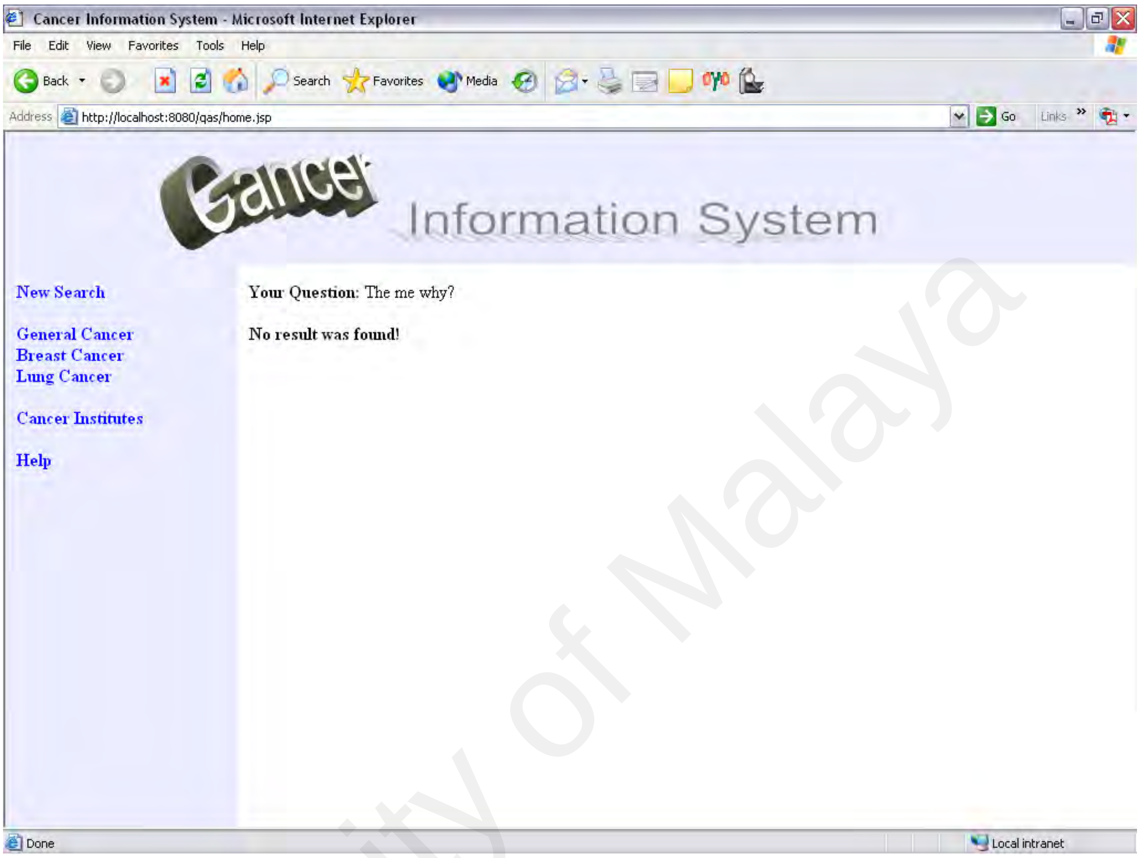


Breast Cancer - Causes

The exact cause or causes of breast cancer remain unknown. Yet scientists have identified a number of risk factors that increase a person's chance of getting this disease. Certain risk factors, such as age, are beyond our control; whereas others, like drinking habits, can be modified.

Figure 5.11: The retrieved document with highlighted keywords

Figure 5.12 shows the "No result was found!" message if the system was unable to retrieve any document based on the query given by user.

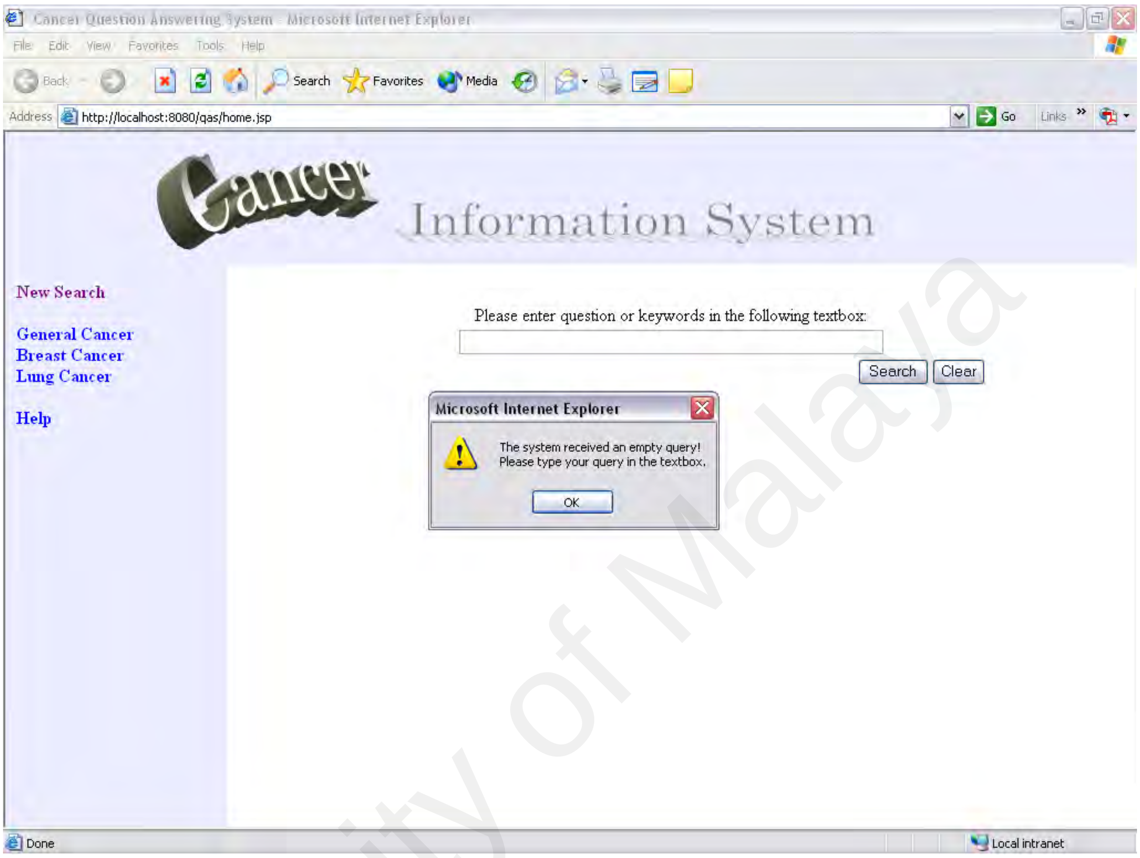


Your Question: The me why?

No result was found!

Figure 5.12: No result was found message

However, if user clicks on *Search* button without typing anything in the textbox, an error message will pop up, as shown in Figure 5.13.


Cancer Question Answering System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://localhost:8080/qas/home.jsp
Go
Links
Cancer
Information System
New Search
General Cancer
Breast Cancer
Lung Cancer
Help
Search
Clear
Microsoft Internet Explorer
OK
Done
Local intranet

Please enter question or keywords in the following textbox:

The system received an empty query!
Please type your query in the textbox.

Figure 5.13: Error handling during searching

5.9 Summary

This chapter elucidated the design of the Cancer Information System. The structural architecture of the system discussed the interaction between system components. Functionalities provided by each system module were also explained in detail. This chapter ended up with discussion on documents collection and interface design.

Chapter Six: System Implementation

6.1 Introduction

System implementation deals with the implementation of the system classes – how the system design is implemented in computer logic. This chapter outlines the implementation issues involved in the development of the Cancer Information System. First of all, the development tools used are described. Then, pseudocode of some system programs is presented to illustrate the logic of the programming.

6.2 Development Tools Used

Several tools were implemented to build the Cancer Information System. These tools include Java technology, Apache Tomcat 5.5.9, Lucece, 1.4.3, JavaScript and HTML. The following sub-sections describe each of these development tools. Anyway, the main tools used were the Java programming language, Apache Tomcat Web server and Lucene. JavaScript and HTML have been the major scripting languages.

6.2.1 Java Technology

Java technology is used in the development of proposed system. It is comprised of both programming language and a selection of specialized platforms (Sun Microsystems, 2004). It has several characteristics such as secure, portable, reliable and scalable.

Java programming language is the programming language used in the coding of the proposed system. Java programs are run on, and interpreted by a program called Java Virtual Machine (JVM). Java is well-known of its portability or platform independence because of JVM. First, Java code is compiled to bytecode, which are simplified machine instructions specific to the Java platform. Then, JVM translates the bytecode into usable machine specific native code.

Several platforms are defined due to the continuous development of Java technology, including Java 2 Platform, Micro Edition (J2ME), Java 2 Platform, Standard Edition (J2SE) and Java 2 Platform, Enterprise Edition (J2EE) (Sun Microsystems, 2004). Each of these platforms consist a set of specific Application Programming Interfaces (APIs). This project uses Java 2 Platform Standard Edition Development Kit 5.0 (JDK 5.0). It provides APIs that are needed in the project and serves as the basis for Java Servlet and JavaServer Pages (JSP), which will be discussed in Section 6.2.2.

Java is a programming language that implements object-oriented method that utilizes the advantages of classes. The class concept and inheritance promotes the code reusability, which have ease the system development process in term of reducing the time for development process, enhancing the quality of the system where the classes can be maintained and modified easily with less impact to others.

6.2.2 Apache Tomcat 5.5.9

Tomcat is a small standalone, fully-featured and free Web server developed under Jakarta Apache Project. The version of Tomcat used is 5.5.9, which is the official reference implementation of the Java Servlets 2.4 and JavaServer Pages (JSP) 2.0 (The Apache Software Foundation, 2005). The Java Servlets and JSP specifications are developed by Sun Microsystems. Apache Tomcat 5.5.9 is used for the purposes of development and testing Java Servlets and JSP pages, and finally as a Web server for the deployment of proposed system upon its completion.

Java Servlet is a Java program that resides and executes on the server side to provide functionality to the server or processing of data on the server. Servlet provides extensive APIs that able to add dynamic content to a Web page. It is the Java counterpart to dynamic Web content technologies such as Common Gateway Interface (CGI) or Active Server Page (ASP) from Microsoft.

JSP is a Java-based server-side scripting language. It is a technology that allows developers to mix regular, static HTML with dynamically content generated from Servlets. It enables Java code to be dynamically embedded within normal HTML Web page. JSP runs on Web server to modify the Web page before it is sent to the client that requested the Web page.

6.2.3 Lucene 1.4.3

Lucene is a high performance, scalable and most popular information retrieval (IR) library, and is one of the projects developed under Apache Jakarta (Hatcher and Gospodnetic, 2004a). Similarly to Apache Tomcat, Lucene is a free and open-source project implemented in Java programming language. Lucene was implemented in the project as the methods for document indexing and searching.

One important point to insist here is that Lucene is not a complete search application that one can use it immediately after installed it. Lucene, however, is a collection of powerful Java classes, or APIs that concern itself with text indexing and searching. It is up to developers to design all other issues such as filesystems, databases, interfaces and presentations designs, and so on. When come to indexing and searching, developers can call the related classes from Lucene. Lucene is a technology suitable for nearly any application that requires full-text searching (Harr, 1999). Basically, Lucene takes a given set of files and parse the contents, adding the words into a searchable index for later use.

6.2.4 JavaScript

Although named with the term *Java*, however, JavaScript is a technology totally different from Java programming language. While Java is compiled to byte-code before being interpreted, Java is a purely interpreted language. JavaScript was developed by Netscape Communications Corporation (Johnson, 1995). JavaScript was used in the system to perform form validation when receiving query from user.

6.2.5 Hypertext Markup Language

Hypertext Markup Language (HTML) is the important language of the Internet's World Wide Web (WWW). It is a scripting language but not programming language, which means that it does not need a compiler to compile the codes, but requires interpreter in order to convert the HTML scripts into Web page's user interface. Interpreter is also referred to Web browser, such as Microsoft Internet Explorer. It can be used in any type of platform. In this project, HTML is used to generate hyperlinks, HTML forms that pass the data to Servlets and is used to create static Web pages.

6.3 Implementation of the System

The Cancer Information System comprises three modules, each module consisted several tasks that were written in one or more class files. Implementation of classes of the system is described in the following sub-sections.

6.3.1 Implementation of Class Indexer.java

Class Indexer.java performs document indexing and creates index. It was supported by Lucene APIs. This task needs to be accomplished before deploying the system. Therefore, it was a separate module (document indexing) to the whole system because it will not be called during system operation. The pseudocode is shown in Figure 6.1.

```
t1 is set to the target directory where index will be saved
t2 is set to the target directory where documents to be scanned located

if t2 does not exist or t2 is not a directory
    display error message
else{
    create an instance of IndexWriter class
    use class MyAnalyzer to analyze the files
    create an array to store all filenames in t2

    while array still has filename
        if filename refers to a directory
            recursive call to current method
        else if filename has .html extension{
            create an instance of Document class
            add a field named "contents" and take value of file contents
            add the respective filename
            save the index in t2
        }
    optimize index
    close the IndexWriter stream
}
```

Figure 6.1: Pseudocode for Indexer.java

At first, two arguments were defined. The first argument was the target directory where index will be saved and the second one was the target directory where documents to be indexed were located. Then, the program checked whether the directory where documents to be indexed exists or not. Trying to open an invalid directory will cause a runtime error. In this case, if the directory is indeed not exist, an error message will be displayed and the program will terminate normally.

IndexWriter was used to create and maintain an index (The Apache Software Foundation, 2004). *IndexWriter* was constructed to take three arguments: path to save index in local filesystems, analyzers that used to analyze text (namely MyAnalyzer, will be described in Section 6.3.4), and the final argument told the *IndexWriter* to create a new, empty index, replacing the existing index, if any.

Document is the unit of indexing and search. *Document* is a set of <*field, value*>. In this project, two fields were created in *Document*. The first field named *contents* and the value was the analyzed text, and the other was named *filename*, which stored the filename of respective file. Both fields and their associate values will be used during searching process. The *contents* field was used to match the query from user while the *filename* field provided a way to access the file through hyperlink.

At the end of indexing, index was optimized for search by merged all segments together into a single segment. Then, *IndexWriter* stream was closed explicitly to reduce resource usage in the program and hence increased performance.

6.3.2 Implementation of Class MainProgram.java

MainProgram.java is a Servlet class in the system. Servlet is a Java class that runs on server. MainProgram.java has the common structure of a Servlet, such as:

- extends *HttpServlet* class,
- defines *doGet* and *doPost* methods,
- sets the *content type* that will be returned to the browser (in this project, the *content type* is set so that the servlet returns an HTML document), and
- uses the *println* method of the *PrintWriter* object to return the results to the user's browser.

There are two HTTP methods that a browser can use to sent HTTP request; *get* method and *post* methods. On the other hand, *doGet* and *doPost* are the two methods that used by any Servlet class to receive the HTTP request. *doGet* method provides the code that is executed when a browser uses the *get* method to request a Servlet while *doPost* will be triggered if the *post* method is used to request a Servlet. Therefore, the MainProgram.java was programmed to use *doPost* method to handle HTTP request and *doGet* was programmed as a method that will call and pass HTTP request to *doPost* method if it has been triggered. In this way, an HTTP request that uses the *get* method will actually execute the *doPost* method of the Servlet. This is a good programming practice since is allows a Servlet to use the same code to handle both the *get* and *post* methods of an HTTP request.

The responsibility of MainProgram.java in the system is much alike intermediate program that communicate with other class files, in which it receives query sent by user through HTTP request, passes the query to other java class file (MainSearcher.java) for processing and then gets the results, and displays the search results in a list in user's browser.

6.3.3 Implementation of MainSearcher.java

Class MainSearcher.java is the heavyweight class in the Document Retrieval Module. The main task of the class is to perform document matching and to return maximum of six relevant documents as the search results. Similarly to indexing, this was performed by using Lucene APIs. It interacts with a couple of other classes such as MainProgram.java and FirstStringToken.java (will be described in Section 6.3.4) that provide services to each other. Figure 6.2 shows the pseudocode of MainSearcher.java.

```
a1 is set to an array to the size of six

receive query from MainProgram.java

create an instance of FisrtStringToken class
pass query to FirstStringToken
get result from FirstStringToken

if result is not null{
        create an instance of File class that has value of directory path of index
        use IndexSearcher and FSDirectory classes to open index for searching

        parse user query into Lucene's query
        search index and create references to underlying documents

        while has more documents{
                save filename and score in Hits object
                print filename and score of document

                if result is smaller than six entries
                        assign the next filename to array a1
        }
}
```

Figure 6.2: Pseudocode for MainSearcher.java

Initially, this class file receives the original raw query entered by user from MainProgram.java. Then, the program pass the query to FirstStringToken.java for initial text processing (will be discussed in Section 6.2.4) and get the result. If the result was not null, the following process will be triggered. Else, the program will terminate and result not found message will be displayed.

Next, class *IndexSearcher* was used to search the index in the given directory. The directory was set by using *FSDirectory* class's method *getDirectory*. Method *getDirectory* cached directory so that the same *FSDirectory* instance will always be returned for a given canonical path (The Apache Software Foundation, 2004).

Search methods in Lucene required user query to be transformed into *Query* object and this job was done by a method in class *QueryParser*, namely *parse*. Again, MyAnalyzer was used to analyze query from user. Then, the search started.

Search results were accessed through *Hits* object. *Hits* object provided efficient access to search results because it returned a set of matching documents sorting in decreasing relevance order. Finally, the program assigns six documents with highest score to the array *a1* that has been defined in the very early of the program. If matched documents were smaller than six, the remaining array is assigned with null.

6.3.4 Implementation of FirstStringToken.java and MyAnalyzer.java

Both of these classes are used to perform several analyses on user's query and file contents, namely tokenization, stopword removals and stemming. Although Lucene provides extensive classes to analyze text, but it still lack of some processing that required in this project. For instance, Class FirstStringToken.java has been built to filters several types of token such as *<root_word>n't*, *ain't*, *can't* and *won't* (described in Section 5.4.1). In addition, it is also convenient to modify the Lucene APIs to suit specifications of different applications. In this project, the list of stopwords provided by Lucene has been modified. Then, MyAnalyzer.java that comprised of several analyses has been created and used to filter text from user and documents (described in Section 5.4).

6.3.5 Implementation of FileRetriever.java

After MainSearcher.java finished the ranking of documents, MainProgram.java will get the search results and display in user's browser. Next, user can select any document in the search results which they think contains the target information. FileRetriever.java is called to open the document requested by user, read and highlight the keywords being used during the search, and display the contents to user.

Figure 6.3 shows the pseudocode for FileRetriever.java. FileRetriever.java is a Servlet class that implements the same Servlet features as the MainProgram.java.

In the program, *FileReader* class was called to read streams of characters from character files. Then, an instance of BufferedReader object was instantiated. Finally, they were combined to read contents of character files and then append the contents in variable c*1*. By default, streams are not buffered. Wrapping BufferedReader around FileReader will buffer the input from the specific file. This will make the reading process more efficient.

After reading the contents of document requested by user, the program then checks each token or word to determine whether it is a keyword or not. If yes, highlight the token and append it to the end of *c2*. Else, just append it to *c2*. Finally, the program display the final output in user’s browser with all keywords being highlighted.

```
receive filename and path from MainProgram.java
receive keywords used during the search from MainProgram.java

create an instance of FileReader class
create an instance of BufferedReader class
combine FileReader instance to BufferedReader instance
assign filename and path to the BufferedReader instance

while file has more lines
    read line and append to variable c1
close the stream

while c1 has more tokens
    while has more keywords
        if current token equals keyword
            highlight current token and append to variable c2
        else
            append current token to variable c2

display c2
```

Figure 6.3: Pseudocode for FileRetriever.java

6.4 Summary

System implementation involved translating the system design into system coding. However, it does not completely concern on coding. As illustrated in Waterfall Life Cycle model, the third phase of system development consists of both implementation and unit testing. Therefore, testing and debugging were carried out to test the codes from time to time.

This chapter discussed the technologies used in the project, mainly focused on Java programming language, Apache Tomcat Web server and Lucene. Subsequently, the design of Java classes was discussed. Pseudocodes for indexing, searching and file retrieval also have been explained. Next chapter will address the testing and evaluation of the system.

Chapter Seven: System Testing and Evaluation

7.1 Introduction

Testing is an important phase in any software development project. It ensures that the software developed performs its tasks in a predictable manner. Testing also ensures that the requirements of the project have been met.

Several types of testing were carried out in this project, such as unit testing that test individual components independently, module testing that test each module independently and system testing that test the system as a whole. The system is tested by incremental testing. In incremental testing, small units are developed and tested before they are integrated to form larger unit. This allows defects or errors to be discovered earlier and make debugging easier since smaller units are tested before proceeding to larger one. For instance, the unit testing was performed in conjunction with system implementation or programming, and module testing was carried out after a module has been developed, and finally system testing took place.

System evaluations have also been performed to test the performance of system in terms of retrieval accuracy. This chapter discusses the results of the evaluation.

7.2 Testing Process

The testing process is important to ensure that the system will perform appropriately without any errors upon its deployment. The testing process adopted in this project comprised of three phases, which include unit testing, module testing and system testing.

7.2.1 Unit Testing

Unit testing is a very basic level of testing that is used to verify the smallest logical units of system code, which normally are individual subroutines, functions, or methods. In unit testing, every unit is treated as an independent unit without other system components (Sommerville, 1998). Unit testing is a part of white box or structural testing technique. It requires the knowledge of code and program internal structure to derive test data. The tests written based on the white box testing strategy incorporate coverage of the program code, branches, paths, statements and internal logic of the program (Parekh, 2005b). Errors resulted from unit testing can be logic, overload or overflow, timing and memory leakage detection errors (Dustin, Rashka and Paul, 1999). This is an iterative process and starts as the implementation begins since it is easier to locate and correct errors when the size of coding is still small.

7.2.1.1 Results of Unit Testing

Unit testing was done parallel to system programming. Every piece of code needs some sort of testing. Therefore, it is impossible to discuss all testing that has been conducted. Below are the examples of the test results for several coding and algorithms related to Question Analysis Module. Since user is free to type anything as query, therefore some test data used were not linguistic stems. Besides that, test data is also not necessary cancer related.

The results of unit testing that was measured against the following algorithms are summarized in Table 7.1:

```
if (token equals “can’t” or “ain’t” or “won’t”)
        discard token;
else
        remove the “n’t” ending;
```

Table 7.1: Unit test results for removing “*n’t*” ending

Test Case	Input	Output	Test Result
1	don’t	do	correct
2	couldn’t	could	correct
3	can’t	–	correct
4	abc’t	abc’t	correct
5	ain’t	–	correct
6	xyzn’t	xyz	correct
7	won’t	–	correct
8	didn’t	did	correct

Unit test in Table 7.1 shows that the program performed as intended. In addition, results for *can’t* (test case 3), *ain’t* (test case 5) and *won’t* (test case 7) show that if an input meets either one of the first three conditions, then the last condition will not be applied. White box testing techniques was used in unit testing. It requires every line of code to be ran or test at least one time, and this requirement was accomplished by the test case shown in Table 7.1.

Table 7.2 shows the test results related to stopword removal. If a particular token exists in the stopwords list (refer to Appendix B), then the system will remove it before proceed to subsequent process. For example, the word *keep* in test case 3 is an instance in the stopwords list; therefore, it is discarded. In test case 5, the word *author* does not exist in the stopwords list. Hence, it is passed to subsequent process (stemming) for further processing.

Table 7.2: Unit test results for stopwords removal

Test Case	**Input**	**Output**	**Test Result**
1	go	–	correct
2	gos	gos	correct
3	keep	–	correct
4	keeps	–	correct
5	author	author	correct
6	s	–	correct
7	familiar	familiar	correct
8	complex	complex	correct

7.2.2 Module Testing

Module testing tests every system module against any defects or errors. It is performed after completion of each system module. Module testing is needed to ensure that the module demonstrates and works according to the specification and requirements of the system. Black box or functional testing technique is used to perform the module testing, and the results are validated with reference to the correlation between the inputs and outputs of each module. Black box testing strategy focuses one the testing for functionality of the program (Parekh, 2005c). Every module is tested independently.

7.2.2.1 Results of Module Testing

During module testing of Question Analysis Module, several questions were used to test the system and the outputs were observed and analyzed. Table 7.3 shows some examples input and output of Question Analysis Module.

Table 7.3: Test data for Question Analysis Module

No.	Input (user query)	Output (system query)
1	What are the side effects of surgery?	side effect surgeri
2	What are the types of cancer?	type cancer
3	What are the causes of breast cancer?	caus breast cancer
4	What are the complications of treatment for lung cancer?	complic treatment lung cancer
5	How to treat lung cancer?	treat lung cancer

Generally, all test data above contain *what*, *are*, *the* and *of*. They are all stopwords and are removed during stopword removal. The question mark (?) in each of the user query (test data) is also discarded. This is one of the requirements for using Porter stemmer. The Porter stemmer requires input that is only comprises of a sequence of letters.

In first test data, the letter *s* at the end of *effects* are removed based in the Step 1a during stemming (please refer Section 5.4.3.3 for any Porter stemming algorithm described in this section). The word surgery is transformed into surgeri by Rule 1c, or Step 1c.

In second test data, *types* is transform into *type* using Rule 1a and *cancer* remain unchanged. In third test data, *causes* is replaced by *cause* using Rule 1a and then changed to *caus* in Step 5a. Both *breast* and *cancer* remain unchanged.

In forth test data, *treatment*, *lung* and *cancer* remain unchanged while *complications* is replaced by *complication* using Rule 1a, then transformed into complicate in Step 2, and finally *complic* in Step 3.

In fifth test data, both *how* and *to* are stopwords. Therefore, they are removed. No processing performed on *treat*, *lung* and *cancer*. If compare the forth and fifth test data, notice that Porter stemmer does not stem *treatment* into *treat* although both word share the same semantic. Even though there is a rule in Step 4, specifically (m>1) MENT → NULL, however, *treat* is denoted by *CVC* => *C*(VC)^1, where m is only 1. Therefore, this transformation will not be executed. This flaw will definitely compromise the search results.

7.2.3 System Testing

System testing starts after all modules were integrated into a complete system. System testing aims to verify that the complete system successfully performs all the system functions that were discussed in the system requirements deliverable. It also ensure that the system comply with the non-functional requirements specified. Besides that, it also tests against any possible errors that occur from inconsistent communications or interfaces between system modules.

7.3 System Evaluation

After system testing, the developed system has been tested against the accuracy of retrieving relevant document with response to user’s query. The system evaluation was done in two different ways. Firstly, the system was tested with a set of 104 questions that were drawn from Frequently Asked Questions (FAQs) on several existing cancer Websites. Those questions and the results of evaluation are shown in Appendix C. Table 7.4 summarizes the results. Secondly, the system was implemented and tested by using questions from user. Table 7.5 summarizes the results for this test.

Table 7.4: Results of system evaluation

Position of highest score document in search results	**Total**	**Percentage (%)**	**Cumulative Percentage (%)**
Top	56	53.85	53.85
Second	19	18.27	72.12
Third	13	12.50	84.62
Forth	5	4.81	89.43
Fifth	1	0.96	90.39
Sixth	2	1.92	92.31
Seventh to tenth	4	3.85	96.16
Eleventh to fifteenth	2	1.92	98.08
Sixteenth to twenty second	2	1.92	100.00
Total	104	100.00	–

From the table, 56 out of 104 (53.85%) questions were ranked first in the search results and the probability for target document to exist top six in the results list was 92.31% (96 over 104 test questions). Initially, it seems that the percentage was very encouraging. However, the questions used in the evaluation were guaranteed that their corresponding answers can be found in the document in documents collection. In reality, user is free to ask any kind of questions or type in anything. In this scenario, the system will be able to return any document as result as long as the system able to generate system query and if any keyword in the system query exist in the documents, even though user may ask for something other than cancer information, and hence compromising the accuracy of searching (this situation can be observed in the second evaluation).

From Table 7.4, it is clear that as moving down the search results, the increase in the percentage which relevant document can be found is insignificant. For instance, from seventh to tenth, there are only four relevant documents in four positions; from eleventh to fifteenth, there are only two relevant documents in five positions; and two relevant documents in the following seven positions. Since the percentage of finding relevant documents beyond sixth position drops drastically, the system was set to retrieve a maximum of six best matched documents in the list of search results.

In the second test, the system was implemented so that it can be tested by user. There were total of ten users participated in the testing. The test aimed to reveal the accuracy of the system by retrieving target document within sixth position in the search results for each user. The conditions of testing are as following:

- User was free to query the system with topics related to "general cancer", breast cancer and lung cancer.
- Each user has to ask at least five questions. However, there was no upper limit to the number of question. The more questions asked, the more representative the result of testing.
- Questions that ask for same answer but phrased with different word were considered as separate questions. For instance, "Who is at risk of developing cancer?" and "What are the risk factors for developing cancer?" were considered as two different questions.
- Since the accuracy was counted differently for each user, two different users might ask a same question twice.

Table 7.5: Results of system evaluation by real users

User	Number of questions	Number of documents within top sixth in search results	Percentage (%)
1	8	5	62.50
2	5	3	60.00
3	9	4	44.44
4	12	7	58.33
5	5	2	40.00
6	6	4	66.67
7	9	5	55.55
8	11	7	63.64
9	7	5	71.43
10	14	9	64.29
Overall	86	51	59.30

Results of second evaluation show that the overall percentage of target within sixth position in the search results was dropped to 59.30% compared to the first evaluation, which was 92.31%, an effective of 35.76% ($\frac{92.31-59.30}{92.31}$*100%) decrease in the accuracy. As explained in the first evaluation, this was because user has no idea about the contents of the documents collection. Although a particular query is related to one of those cancer information covered by the system, its respective answer may not exist in the documents collection. Expanding the documents collection with more information can overcomes this problem. Another reason for the decrease in accuracy is that questions retrieved from FAQ are normally short, clear and focus so that they are understandable. However, in real world, user tends to ask tricky questions, especially during system testing in order to test the system to its limit. Hence, this compromises the accuracy of search results.

Anyway, results of the system evaluation showed that the system performed well in retrieving relevant document with condition that required information can be found in at least one document in documents collection.

7.3.1 The Precision and Recall Metrics

As mentioned the Chapter Two, precision and recall are the two most popular metrics to evaluate the effectiveness of IR system. Majority of the 104 FAQs has only one relevant document in the documents collection, except for the questions that are shown in Table 7.6.

Table 7.6: FAQs with more than one relevant documents

Question Number	**Ranking for Subsequent Document (s)**
21	2nd
27	3rd and 4th
29	2nd and 3rd
30	2nd and 3rd
32	2nd and 3rd
34	3rd
36	3rd

Table 7.7 summarizes the average precision and recall of the Cancer Information System by using the 104 FAQs. Details of the precision-recall evaluation can be found in Appendix D.

Table 7.7: Average Precision-Recall of the Cancer Information System

Recall (%)	Precision (%)
0	68.86%
10	68.86%
20	68.86%
30	68.86%
40	68.54%
50	68.54%
60	67.90%
70	67.98%
80	67.98%
90	67.98%
100	67.98%

Figure 7.1 shows the average precision at 11 standard recall levels of the Cancer Information System. The 11 standard recall levels are 0%, 10%, 20%, ..., 100%.

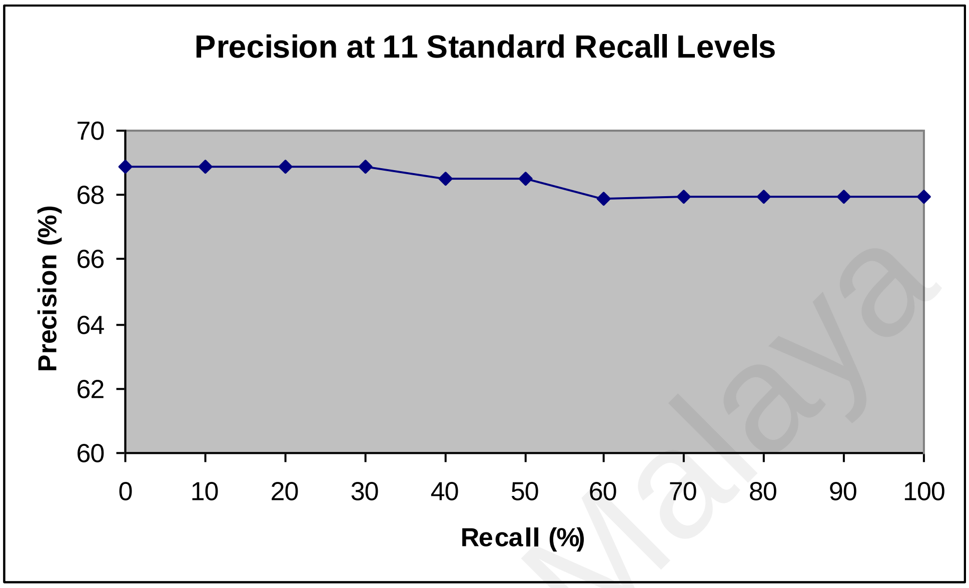


Figure 7.1: Precision at 11 standard recall levels

Table 7.7 and Figure 7.1 show that precision and recall have the inverse correlation. As the recall rate increases, the precision rate will decrease, and vice versa, except for recall level 70%, where the precision has bounced back from 67.90% at recall level 60% to 67.98% at recall level 70%. This is because majority (93.27%) of the test questions has only one relevant document in the documents collection (which means that individual precision-recall for these test questions has the form of flat line graph), and exists a special condition (test question #27) that is able to increase precision at higher recall levels. This special case is discussed in the following paragraph in part (c).

The followings are the examples of individual precision-recall for several test questions (Full list of precision-recall for all 104 test questions can be referred in Appendix D):

(a) Test question #1 has its relevant document ranked in third position. Therefore, at position #3, the recall is 100%, and the precision is 33.33% (one document out of three is relevant). According to interpolation rule (Baeza-Yates and Ribeiro-Neto, 1999), the interpolated precision at the *i*-th standard recall level is the maximum known precision at any recall level between the *i*-th and the (*i*+1)-th recall level. Therefore, the precision for all 11 standard recall levels (0% until 100%) is 33.33% and hence, yield a flat precision-recall line graph.

(b) Test question #29 has three relevant documents that are ranked first, second and third in the search results. At position #1, the recall is 33.33% and the precision is 100%. At position #2, the recall is 66.67% and the precision is 100%. At position #3, the recall is 100% and the precision is 100%. Therefore, the precision for all 11 standard recall levels (0% until 100%) is 100% and hence, yield a flat precision-recall line graph.

(c) Test question #27 has three relevant documents that are ranked first, third and forth in the search results. At position #1, the retrieved document is relevant. This document corresponds to 33.33% of all the relevant documents in the documents collection. According to the interpolation rule, at recall levels 0%, 10%, 20% and 30%, the interpolated precision is 100% (which is the known precision at the recall level 33.33%). At position #3, the retrieved document corresponds to 66.67% of all the relevant documents in the documents collection. Therefore, the interpolated precision is 66.67% (two out of three retrieved documents are relevant) at recall levels 40%, 50% and 60% (which is the known precision at the recall level 66.67%). At position #4, the recall is 100% (all relevant documents have been retrieved) and the precision is 75% (three out of four documents are relevant). Hence, the interpolated precision at recall levels 70%, 80%, 90% and 100% are 75%. This is the special case where precision is increase from 66.67% at recall level 60% to 75% at recall level 70%.

(d) Test question #20 has its relevant document ranked #15 in the list of search results. However, the system is designed to retrieve six most relevant documents. Hence, the relevant document for question #20 is not in this range. Therefore, the precision for all 11 standard recall levels is 0%.

The precision and recall are the two most common metrics to evaluate the performance of IR system. Typically, there is a number of design issues associated with an IR system, such as the term weighting formulae, stopwords, stemming algorithm and so on. Modification or future enhancement on any these design parameters will affect the performance of the system. Therefore, these metrics serve as a basis to compare the performance of the system under different parameters setting. Subsequently, a better system can be built.

7.4 Summary

This chapter discussed the testing and evaluation of the Cancer Information System. It may seem that everything will fall into place without any preparation and a bug-free product will be delivered. However, in the real world, test plan is required for locating and removing bugs (Bahrami, 1999).

Testing was done in a sequence starting from unit testing, module testing and system testing as more system codes were developed. During each of the phase of testing, steps were taken to overcome the problems encountered due to overlook in certain areas of the system codes.

Evaluation of results accuracy showed that most of the relevant documents (92.31%) were positioned within top six in the results list.

Chapter Eight: Conclusion

8.1 Introduction

This chapter outlines the overview of the whole project in regards of developing the Cancer Information System and pinpoints a couple of limitations and future enhancements that can further improve the performance of the system.

8.2 Achievements

The objectives of this project were successfully accomplished. A localized cancer information system has been developed by using information retrieval (IR) techniques that facilitates information searching. The system requirements were achieved successfully.

A study on other existing cancer Websites and IR systems has been carried out throughout the requirement specifications stage. The study of cancer Websites helped to determine some popular topics of cancer that were needed in developing this project. On the other hand, the study on the IR systems helped to identify current "state-of-the-art" technologies that been used in building such system.

Waterfall Life Cycle Model was adopted in this project. The five phases in the model have been studied and followed strictly during developing this project. Questionnaires were carried out in order to collect opinion on feasibility of the project and to reveal the contents of the system.

The system was designed and developed based on findings from literature review and questionnaire. Several technologies were used to develop the system, including Java technology, Apache Tomcat, Lucene, JavaScript and HTML. The Graphical User Interface (GUI) was design to be as simple as possible so that user can use the system with ease.

Several testing processes were performed to verify that the system has fulfilled the requirement specifications and free from bugs. A set of 104 questions has been used to evaluate the efficiency of the system in matching user's query with documents.

8.3 System Limitations

Since this Cancer Information System was only a prototype system. It was only limited to "general cancer", lung cancer and breast cancers. The system does not provide information related to other types of cancers.

Currently, the system treats both the query and documents as model of bag-of-words. This further explain that a document is treated as an unordered list of words, and there is no relation, such as syntax, exists between each word, and hence, it loses semantic relation. Therefore, *lung cancer* and *cancer lung* will retrieve the same set of documents. Additional language processing capabilities are required to handle this problem.

The Cancer Information System is built based on IR techniques. This means that the system will retrieve most relevant documents in response to user's query. Therefore, the system will not answer user's query directly with exact and precise answer. User has to judge which document is most likely contains the required information, and then click on the link in order to browse the document and get the specific information. In addition, it is also possible that the most relevant documents retrieved by the system may not contain the information requested by user due to the fact that perfect matching model does not and will not exist and relevancy of document is a subjective issue.

Besides that, the system was only available in English language. So, it is impossible for user who does not know English language to use the system.

8.4 Future Enhancements

The Cancer Information System has been successfully developed. However, more improvements can be added to the system to enhance the functions provided by the system in future. Normally, the possible enhancements are highly coupled with limitations of the system. The additional enhancements are discussed in the following sub-sections.

8.4.1 Content Development

First of all, the Cancer Information System only provides two types of cancer information details, namely breast cancer and lung cancer, and common characteristics of cancer disease in “general cancer”. This number may suits in the context of prototype to recover the techniques required, but it is definitely not enough if we want to fully utilize the system through Internet. In the future, the system may be developed to include all types of cancer.

Besides, the survey done is more concern to justify the needs and feasibility of a Malaysia-based cancer information system. Although Question 10 aims to collect opinions from participants on the types of information that need to be included in the

Cancer Information System, but those participants are not professional or domain expert. Therefore, in the future, interview can be conducted with a few personnel from the Ministry of Health in order to determine what information should be made known to the users about cancer disease.

8.4.2 Additional Functionalities

Since the Cancer Information System was developed by using IR techniques, it is designed to retrieve most relevant documents in response to user's query. Instead of retrieving full size of documents, more advance features can be integrated into the current system. This includes enhancing the system to be able to answer user's query directly as the role performed by question answering system. In addition, rule-based expert system also can be implemented to predict the possibility of being develop cancer.

8.4.3 Stemming Algorithms

There are still spaces for improvement in Porter stemming algorithms since the algorithm does not stem certain words with similar meaning into same canonical form, as the one reported in Chapter 7. However, the modification needs to be done systematically and probably requires large amount of words for testing purpose. Krovetz (1993) has tried to revised Porter stemmer and found that changing any rule can caused errors to occur at other points in the algorithm. The changed rule also caused the algorithm not able to work with other suffixes. Finally, Krovetz ended up with developing an entirely new algorithm. Tremendous efforts are required to refine Porter stemming algorithm.

8.4.4 Syntax Processing

In the case of bag-of-word approach, stopwords were identified to filter out words that generally have little or no meaning. This method is not efficient enough to remove all words that have little or no meaning. In additional, sometime a word may have different meanings depended on the structure of the sentence. A more precise method is to introduce syntax processing capability, which is able to identify the part-of-speech (POS) of a word in a sentence. Basically, nouns are words that carry most salient meaning with verbs in the second.

8.4.5 Language

This system can also be made for other languages to cater users with different language preferences. At the moment, the system uses English as the medium. However, in the future, the system can be built to include Malay text as the medium.

8.5 Summary

This chapter concluded the development processes of the project. The system caters for all Malaysians that wish to find localized cancer information. Besides that, the system introduces new method, namely natural language search function for accessing information as opposed to traditional hyperlinks method. Natural language searching proved to be a better solution compared to conventional hyperlinks technique.

Even though the system has been successfully developed, it still has a number of limitations. Hence, the system can still be improvised for better performance.

Reference

Allan, J. (1995). *Natural language understanding* (2nd ed.). Redwood City: The Benjamin/Cummings Publishing Company, Inc.

Allan, J., Aslam, J., Belkin, N., Buckley, C., Callan, J., Croft, B., Dumais, S., Fuhr, N., Harman, D., Harper, D. J., Hiemstra, D., Hofmann, T., Hovy, E., Kraaij, W., Lafferty, J., Vavrenko, V., Lewis, D., Liddy, L., Manmatha, R., McCallum, A., Ponte, J., Prager, J., Radev, D., Resnik, P., Robertson, S., Rosenfeld, R., Roukos, S., Sanderson, M., Schwartz, R., Singhal, A., Smeaton, A., Turtle, H., Voorhees, E., Weischedel, R., Xu, J., & Zhai, C. X. (2002). Challenges in information retrieval and language modeling. *SIGIR Forum*, 37(1), 2003.

Als, A., & Greenidge, C. (2003). *The waterfall model*. Retrieved June 13, 2005, from http://scitec.uwichill.edu.bb/cmp/online/cs22l/waterfall_model.htm.

American Cancer Society (2005). *Cancer facts and figures 2005*. Atlanta: American Cancer Society. Retrieved July 17, 2006, from http://www.cancer.org/downloads/STT/CAFF2005f4PWSecured.pdf.

Aylett, J. (2004). *Lucene ranking formula*. Retrieved October 3, 2005, from http://lists.tartarus.org/pipermail/xapian-discuss/2004-November/000571.html.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Retrieved January 6, 2006, from http://www.sims.berkeley.edu/~hearst/irbook/1/node5.html.

Baeza-Yates, R., & Ribeiro-Neta, B. (1999b). *Modern information retrieval*. United States of America: Addison Wesley.

Bahrami, A. (1999). *Object-oriented system development* (International ed.). Singapore: Irwin McGraw-Hill.

Bennett, S., McRobb, S., & Farmer, R. (2002). *Object-oriented systems analysis and design using UML* (2nd ed.). Berkshire: McGraw-Hill Education.

Biesalski, H. K., Bueno, M. B., Chesson, A., Chytil, F., Grimble, R., Hermus, R. J.J., Kohrle, J., Lotan, R., Norpoth, K., Pastorino, U., & Thurnham, D. (1998). European consensus statement on lung cancer: Risk factors and prevention. *A Cancer Journal for Clinicians*, *48(3)*, 167-176.

Bilimoria, M. M., & Morrow, M. (1995). The woman at increased risk for breast cancer: Evaluation and management strategies. *A Cancer Journal for Clinicians*, *45(5)*, 263-278.

Buckley, C., & Salton, G. (1988). Improving retrieval performance by relevance feedback. *Information Processing and Management, 24(5)*, 512-523.

Buckley, C., Allan, J., & Salton, G. (1994). Automatic routing and ad-hoc retrieval using SMART. In *Proceedings of the second Text Retrieval Conference TREC-2*, 45-55.

Buckley, C., Singhal, A., Mitra, M., & Salton, G. (1995). New retrieval approaches using SMART: TREC 4. *NIST Special Publication 500-236*.

Callan, J. P., Croft, W. B., & Broglio, J. (1995). TREC and TIPSTER Experiments in INQUIRY. *Information Processing and Management*, 327-343, 1995.

Chinchor, N. (1997). MUC-7 named entity task definition. In *Proceedings of MUC-7*.

Chowdhury, G. G. (1998). *Introduction to modern information retrieval*. John Wiley & Sons.

Covington, M. A. (2001). *Tokenization using DCG Rules*. Available from http://www.ai.uga.edu/mc/projpaper.pdf.

Croft, W. B. (1993). Knowledge-based and statistical approaches to text retrieval. *IEEE Expert 8 (2)*, 8–12.

Dustin, E., Rashka, J., & Paul, J. (1999). *Automated software testing*. Reading, MA: Addison-Wesley.

eNotes.com (2006). *How many cells are in the human body?* Retrieved January 6, 2006, from http://science.enotes.com/science-fact-finder/how-many-cells-human-body.

Ferber, R., Sheatsley, P., Turner, A., & Waksbery, J. (1980). *What is a survey?* Washington: American Statistical Association.

Frakes, W. B., & Fox, C. J. (2003). Strength and similarity of affix removal stemming algorithms. *ACM SIGIR Forum*, *4(37)*, 26-30.

Fuller, M., & Zobel, J. (1998). Conflation-based comparison of stemming algorithms. In *Proceedings of the Third Australian Document Computing Symposium, Sydney, Australia*.

Goetz, B. (2000). *The Lucene search engine: Powerful, flexible, and free*. Retrieved May 17, 2005, from http://www.javaworld.com/javaworld/jw-09-2000/jw-0915-lucene_p.html.

Gupta, V. (2004). *A question about scoring function in Lucene*. Retrieved June 23, 2005, from http://java2.5341.com/msg/90379.html.

Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Girju, R., Rus, V., & Morarescu, P. (2001). Falcon: Boosting knowledge for answer engines. In *Proceedings 9th Text Retrieval Conference (TREC-9).* NIST Special Publication 500-249.

Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science, 42(1)*, 7-15, 1991.

Harr, C. (1999) *Find it with Lucene*. Retrieved May 15, 2005, from http://cameron.harr.org/cs462/cs462-finalreport.html.

Hatcher, E., & Gospodnetic, O. (2004a). *Chapter 1: Meet Lucene* (Ebook ed.). Available from http://www.manning-source.com/books/hatcher2/hatcher2_chp1.pdf.

Hatcher, E., & Gospodnetic, O. (2004b). *Chapter 3: Adding search to your application* (Ebook ed.). Available from http://www.manning-source.com/books/hatcher2/hatcher2_chp3.pdf

Hersh, W. R., Bhuptiraju, R. T., Ross, L., Jonhson, P., Cohen, A. M., & Kraemer, D. F. (2004). TREC 2004 genomics track overview. *NIST Special Publication 500-261*.

Hiemstra, D. (2000). Using language models for information retrieval. *CTIT Ph.D. Thesis Series No. 01-32*.

Hinson, J. A., & Perry, M. C. (1993). Small cell lung cancer. *A Cancer Journal for Clinicians*, *43(4)*, 216-225.

Hirschman, L., & Gaizauskas, R. (2001). Natural language question answering: the view from here. *Natural Language Engineering 7(4)*, 275-300.

Horwich, A. (Ed.). (1995). *Oncology: A multidisciplinary textbook*. London: Chapman & Hall Medical.

Iljin, P., Brand, R., Driessen, S., & Klok, J. (2003). Oré at TREC 2003. *NIST Special Publication 500-255*.

InfoLab Group (2005). *START natural language question answering system*. Retrieved January 14, 2005, from http://www.ai.mit.edu/projects/infolab/about.html.

Inquiry Stopword List for Thomas. (1998). Retrieved April 18, 2005 from http://thomas.loc.gov/home/stopwords.html.

Johnson, M. (1995). *Java Script: Manual of style*. Retrieved June 13, from http://pbs.mcp.com/ebooks/1562764233/contents.htm.

Krovetz, R. (1993). Viewing morphology as an inference process. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 191-203, 1993.

Langer, S., & Hickey, M. (1997). Automatic message indexing and full-text retrieval for a communication aid. In *Proceedings of the ACL/EACL workshop on NLP for Communication Aids*.

Lewallen, R. (2005). *Software development life cycle models*. Retrieved August 4, 2005, from http://codebetter.com/blogs/raymond.lewallen/archive/2005/07/13/129114.aspx

Lim, G.C.C., Yahaya, H., & Lim, T.O. (2003). *The first report of the National Cancer Registry: Cancer incidence in Malaysia, 2002*. National Caner Registry, Malaysia.

Lim, G.C.C., & Yahaya, H. (2004). *Second report of the National Cancer Registry: Cancer incidence in Malaysia, 2003*. National Cancer Registry: Malaysia.

Lennon, M., Paice, D. S., Tarry, B. D., & Willett, P. (1981). An evaluation of some conflation algorithms for information retrieval. *Journal of Information Science, 3*, 177-183.

Lewis, D. D., & Jones, K.S. (1996). Natural language processing for information retrieval. *Communications of the ACM*, *39(1)*, 1996.

Liu, X. (2003). *Natural language processing*. Retrieved January 6, 2006, from www.cnlp.org/publications/03NLP.LIS.Encyclopedia.pdf.

Llopis, F., & Vicedo, J. L. (2001). IR-n, a passage retrieval system from University of Alicante, at Clef2001. *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, 244-252.

Luger, G. F. (2002). *Artificial intelligence – Structures and strategies for complex problem solving* (4th ed.). Addison Wesley.

Lunsford, T. R., and Lunsford, B. R. (1995). Research Forum – The research sample, part I: sampling. *Journal of Prosthetics and Orthotics, 7(3)*, 105-112.

Maron, M. E., & Kuhns, J. L. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the Association for Computing Machinery 7*, 216-244.

Mulshine, J. L. (2005). Clinical issues in the management of early lung cancer. *Clin Cancer Res*, *11(13)*, 4993-4998.

Onix Text Retrieval Toolkit: Stopword List 1. Retrieved April 18, 2005 from http://www.lextek.com/manuals/onix/stopwords1.html.

Onix Text Retrieval Toolkit: Stopword List 2. Retrieved April 18, 2005 from http://www.lextek.com/manuals/onix/stopwords2.html.

Paice, C, D. (1994). An evaluation method for stemming algorithms. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, 42-50.

Parekh, N. (2005a). *The waterfall model explained*. Retrieved August 8, 2005 from http://www.buzzle.com/editorials/1-5-2005-63768.asp.

Parekh, N. (2005b). *Software testing – White box testing strategy*. Retrieved August 17, 2005, from http://www.buzzle.com/editorials/4-10-2005-68350.asp.

Parekh, N. (2005c). *Software testing – Black box testing strategy*. Retrieved August 17, 2005, from http://www.buzzle.com/editorials/4-10-2005-68349.asp.

Park, L. A. F., Ramamohanarao, K., & Palaniswami, M. (2005). A novel document retrieval method using the discrete wavelet transform. *ACM Transactions on Information Systems, 23(3)*, 267-298, July 2005.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14(3)*, 130-137.

Pressman, R. S. (2001). *Software engineering: A practitioner's approach* (5th ed.). Singapore: McGraw-Hill.

Ramakrishnan, R., & Gehrke, J. (2003). *Database management systems*. McGraw Hill, 2003.

Rapaport, S. A. (1978). *Strike back at cancer*. New Jersey: Prentice-Hall, Inc.

Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation 33(4)*, 294-304.

Robertson, S. E., & Walker, S. (1999). Okapi/Keenbow at TREC-8. *NIST Special Publication 500-246*.

Salton, G., & McGrill M. J. (1983). *Introduction to modern information retrieval*. McGraw-Hill Book Company.

Saggin, R. (2003). *BioPD Stopword List*. Retrieved April 18, 2005, from http://www-fog.bio.unipd.it/waishelp/stoplist.html.

Satoh, K., Okumura, A., & Yamabana, K. (1994). Information Retrieval System for TREC-3. *NIST Special Publication 500-225*.

Schauble, P. (1997). *Multimedia Information Retrieval: Content-based Information retrieval from large text and audio database*. Kluwer Academic Publishers.

Scheuren, F. (2004). *What is a survey*. Washington: American Statistical Association.

Schneider, K. (2002). *Counseling about cancer: Strategies for genetic counseling* (2nd ed.). New York: Wiley-Liss, Inc.

Search Tools Consultation (2002). *Natural Language Processing in Information Retrieval Research*. Retrieved June 13, 2005, from http://www.searchtools.com/info/ir-nlp.html.

Singhal, A. (2001). *The TREC ad-hoc and Web tracks*. Retrieved 27 June, 2005, from http://www10.org/cdrom/papers/317/node2.html.

Smeaton, A. F. (1995). *Natural language processing & information retrieval*. Retrieved January 6, 2006, from www.umiacs.umd.edu/~jimmylin/LBSC796-INFM718R-2005-Spring/papers/Smeaton95tutorial.pdf.

Smith, R. A., Cokkinides, V., Eschenbach, A. C., Levin, B., Cohen, Carmel, Runowicz, C. D., Sener, S., Saslow, D., & Eyre, H. J (2002). American Cancer Society guidelines for the early detection of cancer. *A Cancer Journal for Clinicians*, *52(1)*, 8-22.

Smith, R. A., Eschenbach, A. C., Wender, R., Levin, B., Byers, T., Rothenberger, D., Brooks, D., Creasman, W., Cohen, C., Runowicz, C., Saslow, D., Cokkinides, V., & Eyre, H. (2001). American Cancer Society guidelines for the early detection of cancer: Update of early detection guidelines for prostate, colorectal, and endometrial cancers. *A Cancer Journal for Clinicians*, *51(1)*, 38-75.

Sommerville, I. (1998). *Software Engineering* (5th ed.). Harlow: Addison-Wesley.

Song, F., & Croft, W. B. (1999). A general language model for information retrieval. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*.

Sparck Jones, K., Walker, S., & Robertson, S. E. (1998). A probabilistic model of information retrieval: Development and status. In *Information Processing and Manegement*.

Spoerri, A. (1995). *InfoCrystal: A visual tool for information retrieval*. Retrieved January 15, 2006, from http://www.scils.rutgers.edu/~aspoerri/InfoCrystal/.

StatPac Inc. (2005). *Sampling methods*. Retrieved July 15, 2005, from http://www.statpac.com/surveys/sampling.htm.

Strzalkowski, T., Perez-Carballo, J., & Marinescu, M. (1996). Natural Language Information Retrieval in Digital Library. In *Proceedings of the First ACM International Conference on Digital Libraries,* Bethesda, Maryland, 117-125.

Sun Microsystems. (2004). *Java technology overview*. Retrieved April 25, 2005, from www.java.sun.com.

The Apache Software Foundation (2004). *Lucene 1.4.3 API*. Retrieved June 15, 2005, from http://lucene.apache.org/java/docs/api/index.html.

The Apache Software Foundation (2005). *Apache Jakarta Tomcat*. Retrieved April 26, 2005, from http://jakarta.apache.org/tomcat.

Vogel, V. G. (2000). Breast cancer preventive: A review of current evidence. *A Cancer Journal for Clinicians*, *50(3)*, 156-170.

Volk, M. (2003). *Frequently asked questions about Computational Linguistics*. Retrieved August 17, 2004, from http://www.ifi.unizh.ch/CL/CL_FAQ.html.

Voorhees, E. (1999). Natural language processing and information retrieval. In M. T. Pazienza, (Ed.), *Information Extraction: Towards Scalable, Adaptable Systems*, 32-48. Germany. Springer.

Voorhees, E. (2004). Overview of the TREC 2004 robust retrieval track. *NIST Special Publication 500-261*.

Whitten, Bentley, & Barlow (1994). *Systems analysis and design methods* (3rd ed.). Purdue University: Irwin.

World Health Organization (2005). *Malaysia – Environmental health country profile*. Retrieved March 4, 2006, from www.wpro.who.int/NR/rdonlyres/ C78DA990-ABC1-437A-BCBC-63D9CC73CA23/0/Malaysia.pdf.

Wyman, L. P., & Harrison, L. (2005). *Frequently asked questions about indexing*. Retrieved May 4, 2005, from http://www.asindexing.org/site/indfaq.shtml.

Minister Speech

Dato' Chua, J. M. (2001). *Speech by Y.B. Dato' Chua Jui Meng, Minister of Health Malaysia, in officiating the Regional Technical/Training Workshop on Prevention of Cervical Cancer for the Western Pacific Region, at Hotel Park Plaza, Kuala Lumpur, on 29 March at 9.00 am*. (2001). Retrieved August 16, 2004, from http://www.moh.gov.my/Speech/menteri/290301.htm.

Dato' Chua, J. M. (2002). *Speech by YB Dato' Chua Jui Meng, Minister of Health Malaysia, at the launch of Avon's kiss goodbye to breast cancer – fund raising and awareness campaign 2002, at the lobby, Sentral Station, Kuala Lumpur, on 26 August 2002 at 10.30 am*. (2002). Retrieved August 16, 2004, from http://www.moh.gov.my/Speech/MENTERI/260802.htm.

Dato' Chua, J. M. (2002b). *Speech by Dato' Chua Jui Meng, Minister of Health Malaysia, at the presentation of ISO 9001/2000 certification to Nilai Cancer Institute, Nilai, Negeri Sembilan, on 30 September 2002 at 10.00 am.* (2002). Retrieved August 16, 2004, from http://www.moh.gov.my/Speech/MENTERI/300902.htm.

Dato' Chua, J. M. (2003). *Speech by YB Dato' Chua Jui Meng, Minister of Health Malaysia, at Avon's "Kiss Goodbye to Breast Cancer" Campaign – presentation of Mammatome Machine, at Hospital Putrajaya, on 25 April 2003 at 11.30 am*. (2003). Retrieved August 16, 2004, from http://www.moh.gov.my/speech/menteri/250403.htm.

Datuk Amar, L. M. (2003). *Speech by YB Datuk Amar Leo Moggie, Minister of Energy, Communications & Multimedia at the prize presentation ceremony for the Radiology Malaysia Champ Quiz 2003 & Creative Design Contest 2003I.* Retrieved August 16, 2004, from http://www.radiologymalaysia.org/Media/Press%20Release/2003/Champ%20Quiz%20Prize%20Giving/Oct28LMSpeech.pdf.

Homepage of Cancer Websites

Alt.support.cancer newsqroup (2003). Retrieved September 5, 2005, from http://www.cancersupporters.com/.

BBC Health (2004). Retrieved July 9, 2004, from http://www.bbc.co.uk/health/conditions/cancer/.

Breast Cancer Source (2005). Retrieved September 9, 2005, from http://www.breastcancersource.com.

Best Doctor (2000). Retrieved September 5, 2005, from http://www.bestdoctors.com/en/default.htm.

iVillage (2005). Retrieved July 9, 2004, from http://health.ivillage.com/.

Majlis Kanser Nasional (2003). Retrieved July 9, 2004, from http://www.makna.org.my/index.aspx.

Medindia (2004). Retrieved September 9, 2005, from http://www.medindia.net/index.htm.

National Cancer Institute (2002). Retrieved July 9, 2004, from http://cancer.gov.

National Research Center for Women & Families (2005). Retrieved September 6, 2005, from http://www.center4research.org/womenhlth1.html.

National Women’s Health Resource Center Inc. (2005). Retrieved September 6, 2005, from http://www.healthywomen.org.

Principle Health News (2003). Retrieved Sepember 5, 2005, from http://www.principalhealthnews.com/topic/womens.

Providence Health System (2005). Retrieved September 6, 2005, from http://www.providence.org/Oregon/Health_Resource_Centers/Lung_Cancer/FAQ1.htm.

Radiology Malaysia (2006). Retrieved March 4, 2006, from http://radiologymalaysia.org.

Singapore Cancer Society (2000). Retrieved September 5, 2005, from http://www.singaporecancersociety.org.sg/.

The National Women’s Health Information Center (2003). Retrieved September 9, 2005, from http://www.4woman.gov.

The Ohio State University Medical Center (2003). *Breast Cancer FAQ*. Retrieved September 5, 2005, from http://www.jamesline.com/.

Yahoo! Health (2005). Retrieved September 9, 2005, from http://health.yahoo.com/health/centers/cancer/1.