INTELLIGENT AGENT FOR E-COMMERCE USING GENETIC ALGORITHM

KOK SUN SUN

(WEK 98267)

Supervisor : Mr. Woo Chaw Seng

Moderator : Associate Prof. Dr. Selvanathan

Bachelor of Computer Science

Faculty of Computer Science &
Information Technology
University of Malaya

Session 2000/2001

INTELLIGENT AGENT FOR E-COMMERCE USING GENETIC ALGORITHM

by

KOK SUN SUN

(WEK 98267)

A thesis presented to the Faculty of Computer Science and Information Technology of University of Malaya in partial fulfillment of the Requirements for the Degree of

BACHELOR OF COMPUTER SCIENCE

UNIVERSITY OF MALAYA

SESSION 2000/2001

ACKNOWLEDGMENTS

This project could not be accomplished without the cooperation and support of one another; therefore I would like to take this precious opportunity to convey my appreciation to my friends, my family and my lecturers.

First of all, I would like to express my utmost gratitude to my teammate, Mr. Lim Chin Wen, who has worked hard with me as a team in achieving the completion of this report and also the system development throughout these two consequence semesters. His valued assistance and support really helped me a lot in completing this project.

Not to forget also, my supervising lecturer, Mr. Woo Chaw Seng, who has provided a lot of invaluable advise, guidance and effort that help me to further this report as well as the system development.

Furthermore, special thanks to members of another group, Ms. Yap Hui Sun and Ms. Law Siew Cheng for their suggestions and comments, which has enhanced my understanding of certain terminologies and concepts.

On top of that, I would also like to convey my special thanks to Alex, who has given me a lot of endorsement and moral support that kept me working hard in achieving the success of this project and also my goal of life.

Last but not least, I would like to thank the ones that I could not mentioned in person here, who have helped me directly or indirectly all the way through the semester.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

1.	ADO	ActiveX Data Object
2.	AI	Artificial Intelligence
3.	ANN	Artificial Neural Network
4.	ASP	Active Server Pages
5.	CGI	Common Gateway Interface
6.	DFD	Data Flow Diagram
7.	DPI	Dot Per Inch
8.	E-Commerce	Electronic Commerce
9.	E-R	Entity-Relationship
10.	FTP	File Transfer Protocol
11.	GUI	Graphical User Interface
12.	HTML	Hyper Text Markup Language
13.	HTTP	Hyper Text Transfer Protocol
14.	IDAPI	Integrated Database Application Programming Interface
15.	IIS	Internet Information Server
16.	ISP	Internet Service Provider
17.	LAN	Local Area Network
18.	NC	Network Computer
19.	ODBC	Open Database Connectivity
20.	OLE	Object Linking and Embedding
21.	PC	Personal Computer
22.	PPM	Paper Per Minute
23.	RAD	Rapid Application Development
24.	RDBMS	Relational Database Management System
25.	SET	Secure Electronic Transaction
26.	SP	Service Pack
27.	SQL	Structured Query Language
28.	TCP/IP	Transmission Control Protocol/Internet Protocol
29.	UI	User Interface
30.	WAN	Wide Area Network

ABSTRACT

This project report outlines the introduction, literature study and methodology, system analysis and design, system implementation, system testing and summary of the whole project. The major objective of this system is to develop an intelligent agent that can gather, analyze and categorize information from the Web on the printers that complying the parameters specified /identified by the web users, and to provide recommendations for online purchases for all types of printers.

The recommended web sites are restricted to web pages that provide online purchase facilities and deliveries in Malaysia. The system is able to update the information from the database provided that the administrator monitors its activities periodically.

This system is achieved using the Genetic Algorithm which is capable of performing information retrieval and learning algorithm. The basic concept of the GA is the natural selection and survival of the fittest, which is identical to the human genetic.

In addition, the system is developed using the developing tools that are essential in web applications, namely, web server, database server, system server and web programming tools such as ASP and Visual Basic. In order to develop an intelligent agent, various programming techniques are used in achieving the property of self-learning, information retrieval and searching algorithm.

The system design is based on the features of a search engine which consist of "Spider" to read through the HTML pages, "Indexer" to decide and index the relevant web pages, and "Search algorithm" to retrieve the data from the results of indexing. Whereas the process of examining through the web pages, retrieving and searching the relevant data in a HTML page, and selecting the best satisfying data are based on the features and operations of the Genetic Algorithms.

CHAPTER I

INTRODUCTION

CHAPTER I : INTRODUCTION

As stated on the title page of this report, this project is to develop an Intelligent Agent for E-Commerce using some of the AI approaches like case-based reasoning and genetic algorithm.

1.1 Project Overview

By implementing this intelligent agent, users do not have to search for printers and scanners by heading nowhere. All they have to do is just specify the features of the printer or scanner that they want, and lists of printers or scanners that match the users' requirements and needs will be displayed. Thus the system helps and assists the users without browsing through several websites to obtain the printers or scanners they want.

1.2 Project Definition

This project can be defined as an Intelligent Agent that is developed for Electronic Commerce; therefore we have to define Intelligent Agent and E-Commerce to further understand the objectives and the significance of the project.

1.2.1 Intelligent Agent

Intelligent agent can be defined as a software entity that acts like a personal assistant with some degree of independence and portrays human intelligence in searching for information. It is similar to any other search engine nowadays. The major significance between a search engine and an intelligent agent is search engine is used to search for general subjects, whereas an intelligent agent is used for a specific subject. Hence, AI techniques are suitable in achieving the purpose of an agent, which can perform tasks independently and be able to learn and portray human intelligence.

1.2.2 E-Commerce (Electronic Commerce)

E-Commerce is the activities of buying and selling of goods, advertising and marketing products, through digital communications such as the Internet. It also

includes electronic funds transfer, smart cards, and digital cash over the digital networks. Therefore, the recommended URLs or web sites must provide the facility of online purchases.

1.3 Objectives of Project

1.3.1 Major Objectives

The project aims at developing a tool that collects information from the web to assist users in decision making for printers. Some AI approaches are included, for example neural network, case-based reasoning, genetic algorithm, and others, in order to gather the information effectively. Recommendations for online purchases of printers shall be provided.

1.3.2 Minor Objectives

This project will be built as a web-based application and will eventually has the following features:

i. This system is meant for members and non-members where previous search information of the members is stored whereas previous search results for non-members are not kept, and this service is given free.
ii. The results can be viewed interactively, at another time or via email based on the option chosen by the users (this feature is only for members).
iii. The search results are stored in a database, and will be updated weekly (maintenance of database).
iv. Detection of changes to existing URLs and new URLs shall be performed by the system.

1.3.3 Significance

After the functions and the objectives of the system have been identified, better understanding of the system's role as an enhancement of the existing system are further elaborated as follow:

i. The hassles of searching a desired printer or scanner are reduced where users do not have to browse through all related websites for printers.
ii. Various AI approaches and techniques are explored in building a web-based Intelligent Agent in E-Commerce.
iii. The system also aims at building a user-friendly agent for all level of users.

1.4 Project Scope

For the system to achieve the above-mentioned objectives, the project scope has been narrowed down to a smaller scale, which only covers the Malaysia-based printer retailers or manufacturers.

In order for the online purchases to take place more effectively, only Malaysia-based retailers are involved due to the fact that deliveries of printers from local retailers or manufacturers are more convenience than the International or Multi-National retailers.

Furthermore, Internet users are often curious over the security of online purchasing using credit cards. Although the banks have ensured the public that they have tightened the security of credit cards usage from time to time over the Internet, the public are still doubtful of purchasing online. To make the public more aware and more confidence on online purchases, local retailers are targeted to start off the E-Commerce Online Purchase trend.

The next section of project schedule of this chapter is an overview schedule of the project, which outlines the project planning for two subsequent semesters of WXES 3181 and WXES 3182.

1.5 Project Schedule

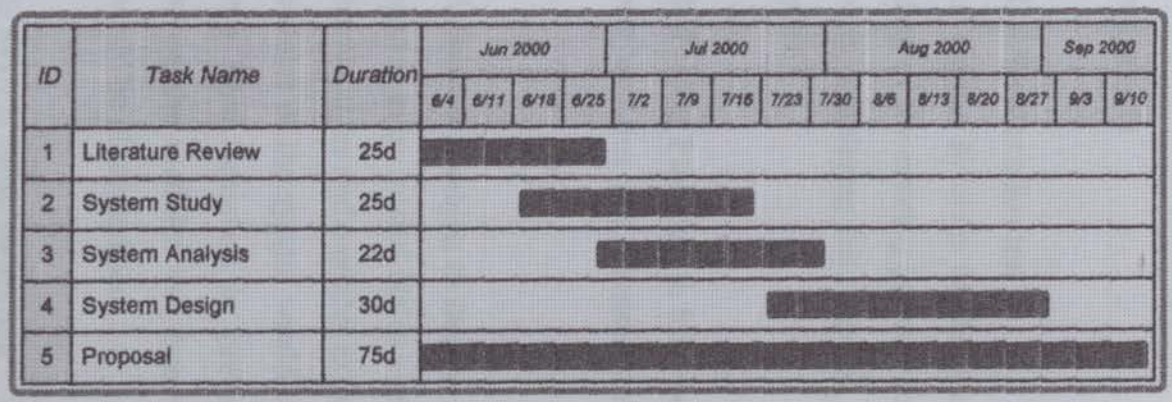


Figure 1-1: Project Schedule for WXES 3181

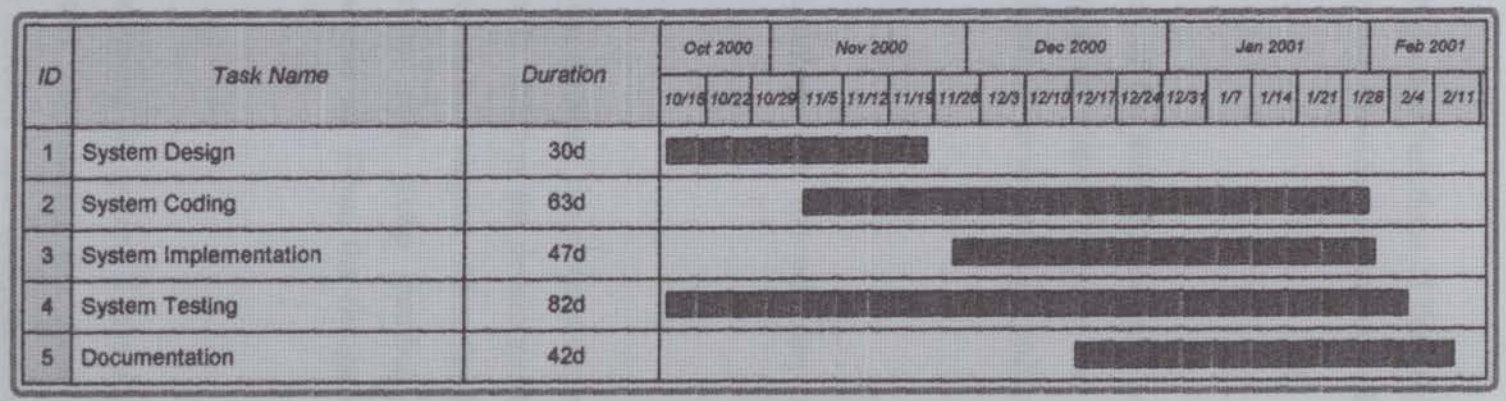


Figure 1-2: Project Schedule for WXES 3182

1.6 Thesis Organization

This section summarizes the organization of this proposal report:

Chapter 1: Introduction

This chapter will introduce briefly on the objectives and functions of the project to let the reader understand the project more deeply and provide fundamental ideas of the proposed project.

Chapter 2: Literature Review

The second chapter involves the literature review of the materials covered and the understanding of the existing projects to furnishing me with the relevant and useful information in completing this proposal, and furthermore to successfully build up the desired system.

Chapter 3: Project Planning & Methodology

This chapter explains and illustrates the methodology and the planning of this project.

Chapter 4: System Analysis & Design

As the project is to be carried out in a group of two students, this chapter explains in detail of the functions and modules in the system. In addition, the development tools and the modules of the system would be covered in this chapter. After the system analysis has been achieved in previous chapter, each group member would concentrate on the modules that one is responsible of. Thus, the design of the modules would be illustrated here.

Chapter 5: System Implementation

This chapter will introduce the process of implementing the system. There may be a few modifications to the earlier proposed system, only during this stage of the project, that the modifications are realized and therefore some components on the system need to be altered. The hardware and software used are explained in details here, and the reasons of each software used is justified here.

Chapter 6: System Testing

This chapter describes the testing process of the system. After the completion of any projects, the systems need to be tested for their integrity, accuracy, limitations, functionality, vulnerability, as well as their interface. All of the procedures of testing are included in this chapter.

Chapter 7: Problems Encountered

All the problems encountered during the completion of the WXES 3182 thesis project are described here. Problems may be solved or may not be solved. Solutions are provided here if problems encountered are successfully solved.

Chapter 8: System Evaluation and Conclusion

This is a summary of the whole thesis project, from the proposal to the end of this project that includes the strengths of the system, the limitations of the system, future enhancements for the system and also the knowledge gained from the accomplishment of this project.

1.7 Chapter Summary

General ideas and the objectives of the project are included in this introductory chapter. Further study and analysis of the project would be covered in the next chapter, which involves the results and analysis of the literature review for this project.

CHAPTER II

LITERATURE REVIEW

CHAPTER II : LITERATURE REVIEW

The overview, the objectives and the scope of the project were introduced in the previous chapter, hence some research and analysis of the project or generally known as literature review need to be carried out. The literature review of a project is very important because the project was placed in the context of other similar projects that helps to understand the features of the existing projects. Another important aspect of literature review is to sufficiently equip the developers with the knowledge of the strengths and the limitations of several development tools before the final decision can be achieved in selecting the most suitable and appropriate development tools.

In this chapter, some further understanding of the project related terms, terminologies and technologies were discussed.

2.1 What is an Intelligent Agent?

People always refer an intelligent agent as an artificial human agent that has the human intelligence characteristics and capabilities to search for the information desired. But, what an intelligent agent really is? The definition and features on an intelligent agent are covered in this section.

2.1.1 Definition

Agent is always be related to sales agent, insurance agent and so on. In AI, an agent has mental properties, such as knowledge, belief, intention, obligation [1].

Thus, in Computer Science, intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires [2].

On the Internet, an intelligent agent (or simply an agent) is a program that gathers information or performs some other service without the immediate presence of a human and on some regular schedule. Typically, an agent program, using parameters that have been provided, searches all or some part of the Internet, gathers information

of interests, and presents them on a daily or other periodic basis. An agent is sometimes called a bot (short for robot).

2.1.2 Features

Other agents have been developed that personalize information on a Web site based on registration information and usage analysis. Other types of agents include specific site watchers that tell users when the site has been updated or look for other events and analyst agents that not only gather but also organize and interpret information for the users. The practice or technology of having information brought to you by an agent is sometimes referred to as push technology, which is discussed in the next section.

2.1.3 Push Technology

Push (or "server-push") is the delivery of information on the Web that is initiated by the information server rather than by the information user or client, as it usually is [3]. An early Web service that specialized in "pushing" information rather than having it "pulled" as the result of requests for Web pages was PointCast, a site that provided up-to-date news and other information tailored to a previously defined user profile. Marimba was a somewhat similar site (and product) that pushed information to the user on a predefined schedule.

In fact, the information pushed from a server to a user actually comes as the result of a programmed request from the client in the user's computer. That is, any information pusher on the Web requires that the user download a client program. This program captures the user's profile and then periodically initiates requests for information on the user's behalf from the server.

A truer form of push is broadcast information. In this case, the information is pushed to everyone that has access to a particular channel or frequency. Broadcast usually (but not always) involves a continuous flow of information.

Another form of "pushed" information is e-mail. Although the e-mail client in your computer has to occasionally go to your local e-mail server to "pick up" the e-mail, the e-mail arrived because someone sent it (pushed) it to you without a one-for-one request having been made [3].

2.1.4 Related Intelligent Agent for E-Commerce in Job Employment Agent

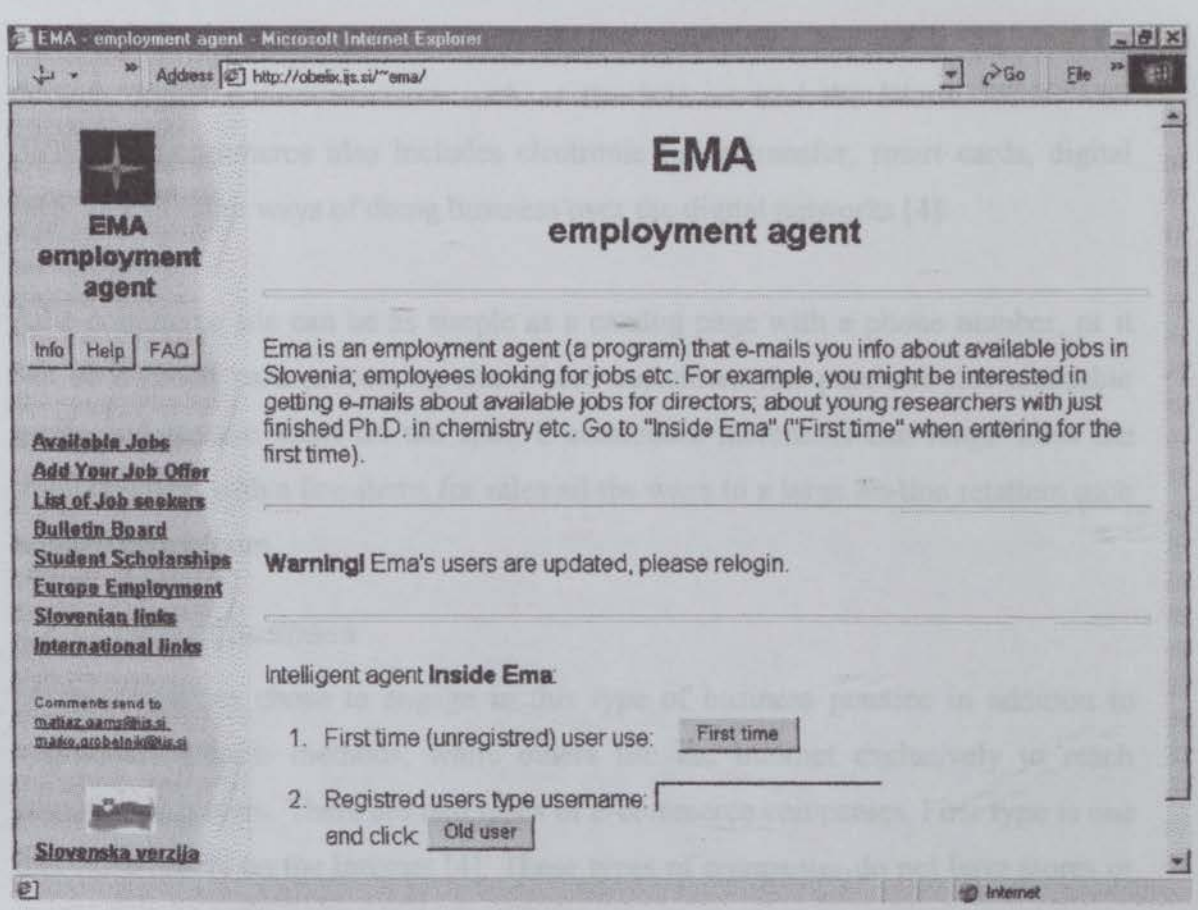


Figure 2-1: An Example of Push Technology in Employment Agent

EMA Employment Agent is an example of web-based system that utilize the "push" technology to inform the customers if there are companies that wish to employ the users through emails. Registered users are also well-informed of the current news and information about the employment issues through newsletters via emails.

2.2 What is Electronic Commerce?

E-commerce is the buying and selling of goods, advertising and marketing products through digital communications such as the Internet and the World Wide Web (WWW). E-commerce also includes electronic funds transfer, smart cards, digital cash, and all other ways of doing business over the digital networks [4].

An e-commerce site can be as simple as a catalog page with a phone number, or it can be a credit card processing site where customers can purchase downloadable goods and receive them on the spot. E-commerce merchants can range from the small business with a few items for sales all the ways to a large on-line retailers such as the Amazon.com

2.2.1 Type of business

Many companies chose to engage in this type of business practice in addition to traditional business methods, while others use the Internet exclusively to reach potential customers. There are two types of e-commerce companies. First type is one that exists solely on the Internet [4]. These types of companies do not have stores or do business in the traditional way. Amazon.com and CdNow.com are the two examples of this type. The second type uses the Internet to compliment their existing business. An example of this type is Midvalley Megamall. In addition to the shops at their shopping mall at Klang Valley, they operates a web site in which consumers can search for goods that they may be interested in buying, and purchase them through their virtual shopping mall at http://www.midvalley.com.my.

2.2.2 Growth of Internet Use Among Consumers

Companies have had a very important reason for moving towards doing business over the Internet. The increased number of Internet users has given businesses a new way to reach customers, and enhance their profits.

According to International Data Corporation (IDC) and Bear, Stearns & Co., the number of people on the web worldwide in 1999 is 212.1 million people, projected to expand to 410.4 million as early as 2003. This is a 94% increase in a single year.

When you divide out the e-commerce spending patterns, each person online will spend US$382 on the Web in 1999. And strategic consultant IDC estimates that e-commerce will rise to over US$288 billion in 2003.

2.2.3 *Benefits of E-Commerce for Businesses*

The benefits that e-commerce has had for businesses in the late 1990's are tremendous.

2.2.3.1 New markets

By doing business online, companies have been able to reach customers all over the world, thereby enabling them to expand their business as well as increase profits. The Internet also gives companies access to potential customers by giving presence 24 hours a day, 7 days a week, for very little outlay.

2.2.3.2 Collect Information from customer

Through the use of cookies, businesses can also collect information about their customers. Cookies are small files that are placed on to a users hard drive when that user logs on to a web site. The cookies help the operator of the web site to collect information about the buying habits of particular groups of people. This information is invaluable to businesses because it allows them to better target their advertising with better information about demographics.

2.2.3.3 Reduce cost

Another benefit that e-commerce offers to businesses is a reduced amount of overhead cost. A company that does business on the Internet will have reduced overhead costs because it is not spending as much money on buildings and customer service as compared to a typical traditional business. This helps companies to increase their profits.

2.2.3.4 Improved after-sales business

The most profitable business is often repeat business - a good after-sales business is essential. Emailing and the online web site can take away some administrative burden that comes with processing queries, maintaining customer records and

keeping customers informed. Customers can easily log into their account and view their order status, for instance.

2.2.4 Benefits of E-Commerce for Consumers

Just as businesses have been willing to embrace e-commerce as a legitimate way to do business, consumers have been eager to take advantage of all the possibilities that e-commerce has to offer.

2.2.4.1 Convenience

The biggest advantage for consumers to do business online is convenience [5]. An Internet shopper can use his PC day or night seven days a week to buy almost anything. A consumer does not have to wait in long lines in stores or even leave the house; all he has to do is click on a product that he wishes to purchase, enter his credit card information, and sit back and wait for the product to arrive by mail.

Some e-commerce companies have made this process even more convenient than this. Many online "stores" store the shopper's credit card information on their server, so the information needs to be entered only one time. Some online businesses do not even ship their products by mail to their customers, especially those that sell computer software. For example, beyond.com allows their customers to download the software that they purchase directly to their computer. More and more products such as video and music will most likely be available this way in the future, as bandwidth improves over time and download time decreases.

2.2.4.2 Lower Cost

Another benefit that e-commerce offers to consumers is reduced cost. Companies that sell stocks online, such as e-trade.com often charge only about $10 a trade, which is much cheaper than buying stock through a traditional stock broker. Many online companies that sell books, videos, and CD's offer low prices, and when shipping and handling is added in, the final cost to the consumer is often comparable to prices found in traditional stores. As the following information shows, consumers can benefit from lower costs by doing business online.

Table 2-1: Comparison of Costs Imposed on Web-based Commerce and Ordinary Commerce

Cost to Process Airline Tickets [5].

$8.00:	Travel agent books, using computer reservation system
$6.00:	Travel agent books direct with airline
$1.00:	Customer books "electronic ticket" direct with airline

Source: Air Transport Association of America, 20/11/97

2.2.5 *The Future of E-commerce*

The combination of commerce and Internet technology has proven to be beneficial for consumers and businesses alike. The benefits brought about by e-commerce will likely continue, and perhaps in the future e-commerce will be even more beneficial to those who choose to take advantage of it. There are a number of obstacles that must be overcome before doing business over the Internet can reach its full potential. E-commerce will not likely replace the local department store anytime soon, but it will allow consumers and businesses to take full advantage of a new medium. Internet businesses will continue to grow and prosper for many years to come.

2.2.6 *Summary*

An e-commerce store is opened for 24 hours a day, 7 days a week, for very little outlay. It can cut down on call-center and customer support costs, using the Web to provide answers and solutions for customers with speed and accuracy. And more than that, it's a great way to do business.

2.3 *What is a Search Engine?*

There are a lot of search engines available in on the Internet, and they are pretty helpful especially when we want to search for specific subjects or topics regarding our field of interests. With the advancement of technology now, people will no longer have to go the library to search for information, where we have to apply for library memberships and pay a large sum of money yearly subscribing magazines while we can get everything on the Internet with just a few mouse clicks and only

pay for our local call charges and the ISPs that we have registered. Although search engines are very popular among us, do we really know how they work?

2.3.1 Definition

Search engines create their listings automatically. Search engines crawl the web, then people search through what they have found. If there are changes on the web pages, these will eventually affect the listings because page titles, body copies and other elements play a role [6].

2.3.2 History of Search Engine

As the Internet grew by leaps and bounds, searching for information went from difficult to virtually impossible. Additionally when older pages were removed from the network or moved to different places, nobody took care of notifying everyone of the changes. As a result of the growth in Internet users, service providers, available information, and retrieval problems, software engineers and information specialists alike attempted to catalog, index, or otherwise provide improved access to useful sites on the Internet, but with generally limited success [6]. The development moved in two different directions, towards directories and search engines. People often cannot differentiate a directory and a search engine, and they always refer both as search engine. The major difference between them is the compilation of the listings.

2.3.2.1 Directories

A directory such as Yahoo depends on humans for its listings, where a short description to the entire site is submitted or written by the editors after they have reviewed the site [6]. When people search for information at Yahoo, the system actually matches the keywords with the descriptions in the listings.

Of course, sites that have not been cataloged cannot be found, and with the rate of World Wide Web growth, there are no teams that can keep up with cataloging new sites as quickly as they appear. Additionally, manual catalogs are fairly limited in cross-reference because it is just not possible to anticipate all of the potentially related topics and to provide pointers to other areas of the catalog to cover all contingencies.

2.3.2.2 Search Engines

Search engines maintain a database with links to Internet resources and are great for quickly identifying specific resource. Search engines crawl the web, then people search through what they have found [6].

2.3.3 How Search Engine Works

Search engines have three major elements:

First is the spider, also called the crawler. The spider visits a web page, reads it, and then follows links to other pages within the site. This is what it means when someone refers to a site being spidered or crawled. The spider returns to the site on a regular basis, such as every month or two, to look for changes.

Second is the index, or sometimes called the catalog, which is like a giant book containing a copy of every web page that the spider has found. If a web page changes, this catalog will have the updated information.

Third element is the search engine software. This is the program that sifts through the millions of pages recorded in the index to find matches to a search and rank them in order of what it believes is relevant [7].

2.3.4 Existing Search Engine

All search engines have the basic parts described above, but there are differences in how these parts are tuned. That is why the same search on different search engines often produces different results.

2.4 Database Server

A database is a collection of data that is organized so that its contents can easily be accessed, managed, and updated. The most prevalent type of database is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. A distributed database is one that can be dispersed or replicated among different points in a network. An object-oriented database is one that is congruent with the data defined in object classes and subclasses [8].

SQL (Structured Query Language) is a standard interactive and programming language for getting information from and updating a database such as IBM's DB2, Microsoft's Access, and database products from Oracle, Sybase, and Computer Associates. Although SQL is both an ANSI and an ISO standard, many database products support SQL with proprietary extensions to the standard language. Queries take the form of a command language that lets you select, insert, update, find out the location of data, and so forth. There is also a programming interface [9].

In this proposal, several database-processing tools, such as Microsoft SQL Server 7.0 and Microsoft Access are considered. The details of each database-processing tool and comparisons are discussed in detail in Chapter IV: System Analysis & Design.

2.5 Web Server

Specific to the Web, a Web server is the computer program (housed in a computer) that serves requested HTML pages or files. A Web client is the requesting program associated with the user. The Web browser in your computer is a client that requests HTML files from Web servers [4].

2.5.1 Client/Server

Client /Server system is a distributed system architecture where client systems are connected to server systems [4]. The client provides an interface to applications and data that are stored on the server. The interface can be provided through a browser such as:

1. Microsoft Internet Explorer
2. Netscape Navigator
3. Sunsoft HotJava

Client activity and processing are said to be on the client-side, while server activity and processing are on the server-side. The network that provides the connection between clients and servers might be a:

- LAN (Local Area Network)
- WAN (Wide Area Network)
- Internet
- Intranet

Client/server network protocols include:

- TCP/IP
- IPX (Internet Packet exchange).

2.5.2 *Client/server Application*

A broad term used to describe an application, which is spread across at least two platforms, including a server that may store application logic and data and server client platform such as PCs or NCs [4]. The Web is a client/server architecture. A variety of RDBMS development tools can be used to develop applications that are able to access ODBC-compliant databases stored on servers. RDBMS development tools useful for developing client/server applications will provide SQL links to interface applications with different databases. For instance it might provide links for Oracle, Sybase, MS SQL Server Informix and InterBase. They may also provide a means of upsizing applications to client/server systems with the minimal amount of coding, or even no coding at all. Many tools aimed at the development of client/server applications have a RAD identity, which hinges largely on the visual programming model.

Web server is essential for building up web-based application; therefore, we have identified several popular web servers for selections, namely, Microsoft IIS 4.0, and Netscape Enterprise Servers. Details of the Web servers are covered in Chapter IV: System Analysis & Design.

2.6 Artificial Intelligent (AI) Methods or Techniques

Information retrieval using probabilistic techniques has attracted significant attention on the part of researchers in information and computer science over the past few decades. In the 1980s knowledge-based techniques also made an impressive contribution to "intelligent" information retrieval and indexing [10]. More recently, information science researchers have turned to other newer artificial intelligence based inductive learning techniques including neural networks, symbolic learning, and genetic algorithms. These newer techniques, which are grounded on diverse paradigms, have provided great opportunities for researchers to enhance the information processing and retrieval capabilities of current information storage and retrieval systems.

2.6.1 Conventional Search Techniques

2.6.1.1 Depth-First Search

Search technique that looks for a solution along each branch of a problem space to its full vertical length, and then proceeds in some defined order, such as left to right.

Advantages of Depth-First Search

- Depth-first search requires less memory since only the nodes on the current path are stored. This contrasts with breadth-first search where all of the tree that has so far been generated must be stored.

- Depth-first search perform an exhaustive exploration of the problem space in an attempt to find a solution, Given that at least one solution exists, a depth-first search technique us guaranteed to find it.

- Depth-first search quickly searches deeply into a problem space. If it is known that the solution path will be long, then depth-first search is a good choice. For example, a game such as chess would require a deep search. Depth-first will not waste time searching for shallow solutions as is the case with breadth-first search.

2.6.1.2 Breadth-first search

Search technique that looks for a solution along all of the nodes on one level of a problem space before considering nodes at the next lower level.

Advantages of Breadth-first Search

- Breadth-first search will not get trapped exploring a blind alley. This contrasts with depth-first searching, which may follow a single, unfruitful path for a very long time, perhaps forever, before the path actually terminates in a state that no successors.

- Like depth-first search, breadth-first will find a solution if one exists. Breadth-first search will not miss shallow or easy solution. If there is a solution, breadth-first search is guaranteed to find it. Furthermore, if there are multiple solutions, then a minimal solution will be found. This is because long paths are never explored until all shorter ones have already been examined.

2.6.1.3 Best-First Search

Search technique that uses knowledge about the problem to guide the search. It guides the search towards the solution nodes of the problem space.

Advantages of Best-first Search

1. The principal advantage of a best-first technique is that it uses knowledge to guide the search. It uses knowledge to know where to best begin and how to best proceed in search a solution.

2. Best-first search follows the reasoning process of a human expert. Human experts. Humans apply knowledge to aid their reasoning and rarely rely on blind techniques such as seen in the depth-first and breadth-first search methods

2.6.2 Artificial Neural Network (ANN)

In information technology, a neural network is a system of programs and data structures that approximates the operation of the human brain [4]. ANN is a Network of neurons, which function as processing units. ANN is considered parts of the field

of artificial intelligence (AI), and their implementation is an attempt to reconstruct the operation of the human brains, which has some the thousand million neurons [5]. The neuron connections have weights, which determine network behaviors. Given an example, the weights may be learned by the network. Neural network variants include the:

i. Perceptron

ii. Multilayer perceptron

The perceptron neuron was proposed in 1962 by Frank Rosenblatt, a significant figure in the development of synthetic neural networks [4]. A computational neuron has input connections, each of which may have a different weight. The neuron is preprogrammed with a threshold value which, if equaled or exceed by the total weight of inputs, will respond accordingly. Typically the response is to output a specific value, Neurons differ from multiple input logic gates (such as AND, NAND, OR or NOR) in that the inputs may not be one of two logical values. The inputs *I* are assigned the weights *w*, and a positive output is yielded should a predefined threshold *t* be exceeded:

```
If i1w1 = + i2w2 + i3w3 +....+ inwn > t, Then
      output = 1

Else
      Output = 0
```

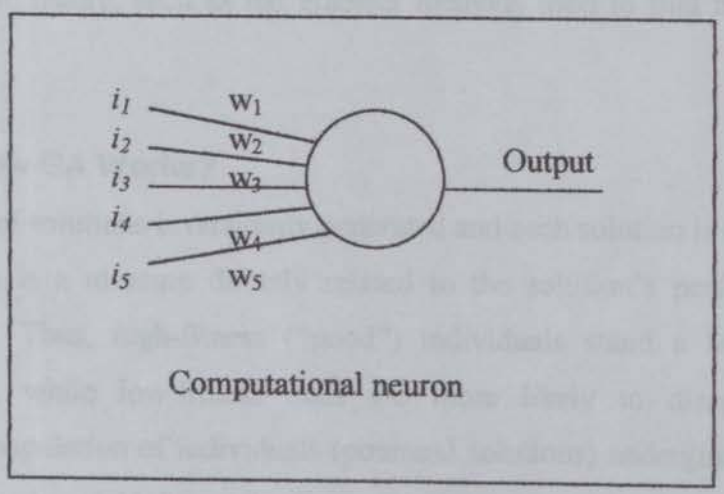


Figure 2-1: Computational Neuron in Artificial Neural Network

Current applications of neural networks include: oil exploration data analysis, weather prediction, the interpretation of nucleotide sequences in biology labs, and the exploration of models of thinking and consciousness [4].

2.6.3 Genetic Algorithm

A Genetic Algorithm is a search method inspired by natural selection [11]. Therefore, genetic algorithms (GA) are computerized search techniques based on the principles of natural genetics. GA exploits the idea of the *survival of the fittest* and *an interbreeding population* to create a novel and innovative search strategy. A population of strings, representing solutions to a specified problem, is maintained by GA. The GA then iteratively creates new populations from the old by ranking the strings and interbreeding the fittest to create new strings, which are (hopefully) closer to the optimum solution to the problem at hand. So in each generation, the GA creates a set of strings from the bits and pieces of the previous strings, occasionally adding random new data to keep the population from stagnating. The end result is a search strategy that is tailored for vast, complex, multimodal search spaces.

GA is a form of randomized search, in that the way in which strings are chosen and combined is a stochastic process. This is a radically different approach to the problem solving methods used by more traditional algorithms, which tend to be more deterministic in nature, such as the gradient methods used to find minima in graph theory.

2.6.3.1 How GA Works?

A population of solutions is randomly generated and each solution is evaluated for its fitness, which is a measure directly related to the solution's performance in the problem task. Thus, high-fitness ("good") individuals stand a better chance of 'reproducing', while low-fitness ones are more likely to disappear. In such algorithms a population of individuals (potential solutions) undergoes a sequence of unary (mutation) and higher order (crossover) transformations. These individuals strive for survival: a selection (reproduction) scheme, biased towards selecting fitter individuals, produces the individuals for the next generation. After some number of

generations the program converges - the best individual represents the optimum solution [10].

The idea of *survival of the fittest* is of great importance to genetic algorithms. GAs use what is termed as a *fitness function* in order to select the fittest string that will be used to create new, and conceivably better, populations of strings. The fitness function takes a string and assigns a relative fitness value to the string. The method by which it does this and the nature of the fitness value does not matter. The only thing that the fitness function must do is to rank the strings in some way by producing the fitness value. These values are then used to select the fittest strings. The concept of a fitness function is, in fact, a particular instance of a more general AI concept, the *objective function.*

Below presents the standard genetic algorithm process flowchart.

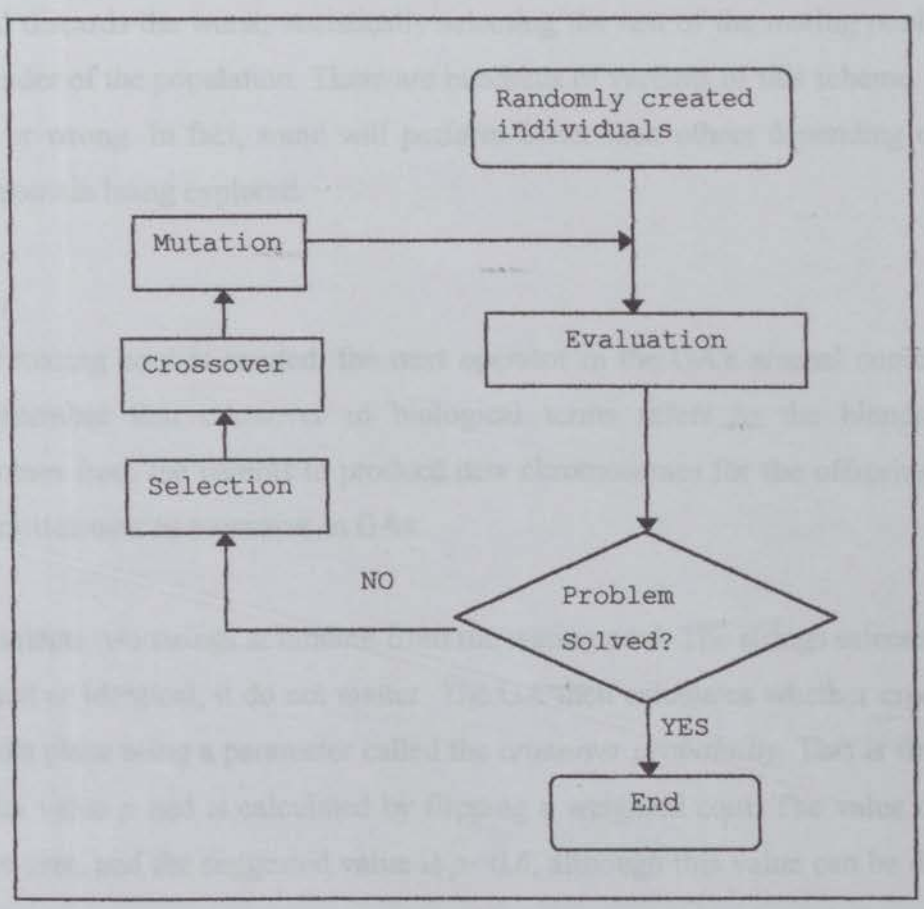


Figure 2-1: Flowchart of Genetic Algorithm Process

2.6.3.2 The Basic Genetic Algorithm Operations

There are three basic operators found in every genetic algorithm [12].

1. Reproduction
2. Crossover
3. Mutation

Reproduction

The reproduction operator allows individual strings to be copied for possible inclusion in the next generation. The chance that a string will be copied is based on the string's *fitness value*, calculated from a *fitness function*. For each generation, the reproduction operator chooses strings that are placed into a *mating pool*, which is used as the basis for creating the next generation.

There are many different types of reproduction operators. One always selects the fittest and discards the worst, statistically selecting the rest of the mating pool from the remainder of the population. There are hundreds of variants of this scheme. None are right or wrong. In fact, some will perform better than others depending on the problem domain being explored.

Crossover

Once the mating pool is created, the next operator in the GA's arsenal comes into play. Remember that crossover in biological terms refers to the blending of chromosomes from the parents to produce new chromosomes for the offspring. The analogy carries over to crossover in GAs.

The GA selects two strings at random from the mating pool. The strings selected may be different or identical, it do not matter. The GA then calculates whether crossover should take place using a parameter called the *crossover probability*. This is simply a probability value p and is calculated by flipping a weighted coin. The value of p is set by the user, and the suggested value is $p=0.6$, although this value can be domain dependant.

If the GA decides not to perform crossover, the two selected strings are simply copied to the new population (they are *not* deleted from the mating pool. They may be used multiple times during crossover). If crossover does take place, then a random splicing point is chosen in a string, the two strings are spliced and the spliced regions are mixed to create two (potentially) new strings. These child strings are then placed in the new population.

Crossover in Action

Figure 2-1: An Example of Crossover Operator

Mutation

Selection and crossover alone can obviously generate a staggering amount of differing strings. However, depending on the initial population chosen, there may not be enough variety of strings to ensure the GA sees the entire problem space. Or the GA may find itself converging on strings that are not quite close to the optimum it seeks due to a bad initial population.

Some of these problems are overcome by introducing a mutation operator into the GA. The GA has a *mutation probability*, *m*, which dictates the frequency at which mutation occurs. Mutation can be performed either during selection or crossover (though crossover is more usual). For each string element in each string in the mating pool, the GA checks to see if it should perform a mutation. If it should, it randomly changes the element value to a new one. In our binary strings, 1s are changed to 0s and 0s to 1s.

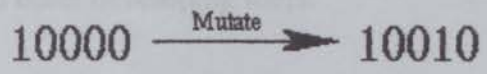


Figure 2-1: An Example of Mutation Operator

2.7 Chapter Summary

This chapter generally explains the literature study of the project, which was carried out since the beginning of the semester. Various topics and terminology were covered and studied in detail especially the AI methods that are going to be applied into the system modules as well as the program flows.

Literature review helps to develop deeper understanding of the system functional and nonfunctional requirements, as well as the system design. Related AI techniques and systems were studied carefully to obtain the underlying ideas and concepts, hence empowering my ability and knowledge to advance to the next stage of the project.

The next chapter regards project planning and methodology, which has helped to identify the functional and nonfunctional requirements of the system. Hardware requirements are listed in details based on the software that has been carefully analyzed and evaluated, such as server consideration, database consideration, web server consideration and other developing tools.

CHAPTER III

PROJECT PLANNING & METHODOLOGY

CHAPTER III : PROJECT PLANNING & METHODOLOGY

The previous chapters have introduced the overview and information gathered of the project. Whereas this chapter would further elaborate the project methodology and various software and hardware considerations on different developing tools.

3.1 Project Planning

Project Planning is the process of organizing a project so that its component parts are completed on time. The compilation of a design, development and production schedule is essential for the efficient use of time or for the providing a completion date [4]. Please refer to the project schedule as in Chapter 1 for details of project schedule in Figure 1.1 for WXES 3182 as well as the whole project schedule from WXES 3181 (proposal stage) to WXES 3182 (System Implementation Stage).

As with costing, estimation may play role, the accuracy of which will increase with greater experience. Due to the fact that the system is developed as a partial fulfillment of the Degree of Computer Science, the cost estimation is unnecessary and can be omitted at the same time.

3.2 Project Development Methodology

During the accomplishment of the project, various methodologies have been used. These include brainstorming, research & analysis, prototyping, inter-group discussions and lecturers advise.

3.2.1 System Development Life Cycle (SDLC)

In System Analysis & Design course, I have learned about the effectiveness of SDLC. It enhances the system development cycle effectively in terms of cost and time. I have utilized the Waterfall Model with Prototyping as a guideline in implementing the system development [13].

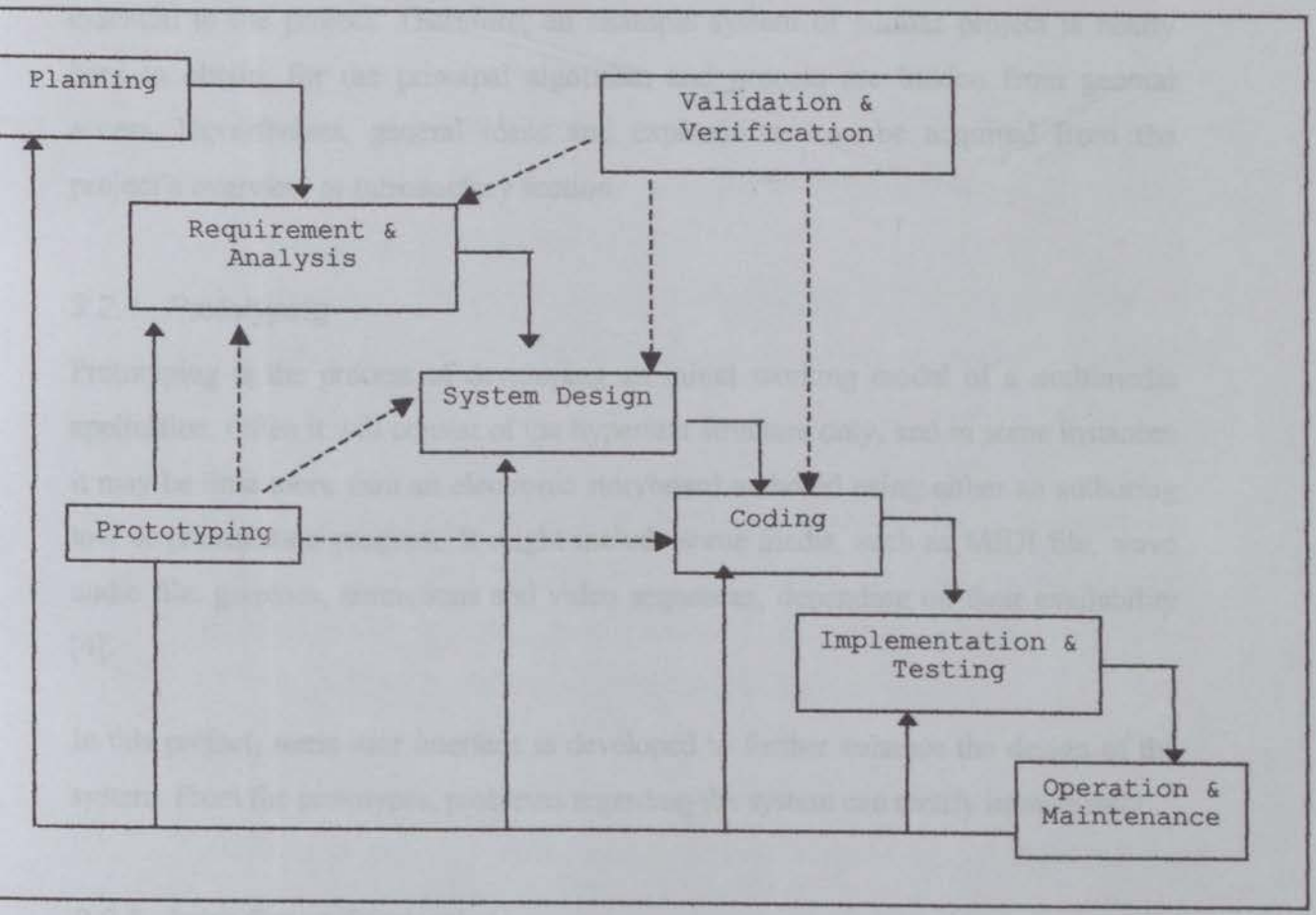


Figure 3-1: Waterfall Model with Prototyping

3.2.2 Brainstorming

Before any actions are taken to accomplish the project, brainstorming of related terms such as intelligent agent, e-commerce, search engine, information searching, information retrieval and database processing are carried out to get a better overview of the underlying ideas of the project title and objectives.

3.2.3 Research & Analysis

Various articles, conference papers, research papers and journals are studied carefully, and analysis was carried out. General algorithm of the GAs and ANN can be obtained from most of the conference papers and research papers. Based on the algorithms and process proposed, the idea of the project can be formulated more effectively. Difficulties often arise when the algorithms of certain related project are essential to the project. Therefore, an example system of similar project is nearly hard to obtain, for the principal algorithm and process are hidden from general access. Nevertheless, general ideas and explanations can be acquired from the project's overview or introductory section.

3.2.4 Prototyping

Prototyping is the process of developing an initial working model of a multimedia application. Often it will consist of the hypertext structure only, and in some instances it may be little more than an electronic storyboard authored using either an authoring tool or presentation program. It might include some media, such as MIDI file, wave audio file, graphics, animations and video sequences, depending on their availability [4].

In this project, some user interface is developed to further enhance the design of the system. From the prototypes, problems regarding the system can rectify immediately.

3.2.5 Inter-Group Discussions

Apart from the methods above, discussions with other groups are very effective in sharing knowledge and different analysis obtained. This will reduce the time consumed in unnecessary materials.

3.2.6 *Lecturers' Advise*

Advise from lecturers are inevitably important, especially the supervising lecturer. Suggestions and comments are often obtained and applied into the project.

3.3 Functional Requirements

Functional requirements are features and limitations of system capability of achieving the goals of the project [14]. Some of the functional requirements have been identified as follow:

i. The system provides a Member Area for registered users to keep track their previous search information, so that they can keep track of their information collected. Unregistered users can still access the agent, but without the facility of storing their history search.

ii. Searching for information can be divided into two major categories:
   a. Database Search – Search information from the database as data in the database is updated every 7 days.
   b. Database Updates – The data in the database will be updated within a period of time, so that the information is up to date. This part is invisible to the users.

iii. Data in the database that is more 7 days old would be updated during the maintenance process configured in the system. If changes have been detected during the online search, data would be updated at the same time. This will reduce the time consumed for maintenance process.

iv. Information would be gathered based on the features of the printer that user has entered earlier. And results of the search would be displayed for the user's reference.

v. The URL of the Website would also be provided that can help the system to keep track of the relevancy of the data provided in the web pages, as well as letting users to purchase online from the website recommended.

3.4 Non-Functional Requirement

Non-functional requirements constraint the behavior of the system in terms of safety, reliability and more for the completeness and correctness of the system [14]. Several requirements have been composed to further enhance the proposed system:

i. The search query page has been designed to let users choose from the pull-down menus of manufacturer, printer type, price range, resolution, and more. Mistyping or misspelling can be avoided.

ii. Date for the results gathered would be displayed due to the fast changing of printer price.

3.5 Web Architecture

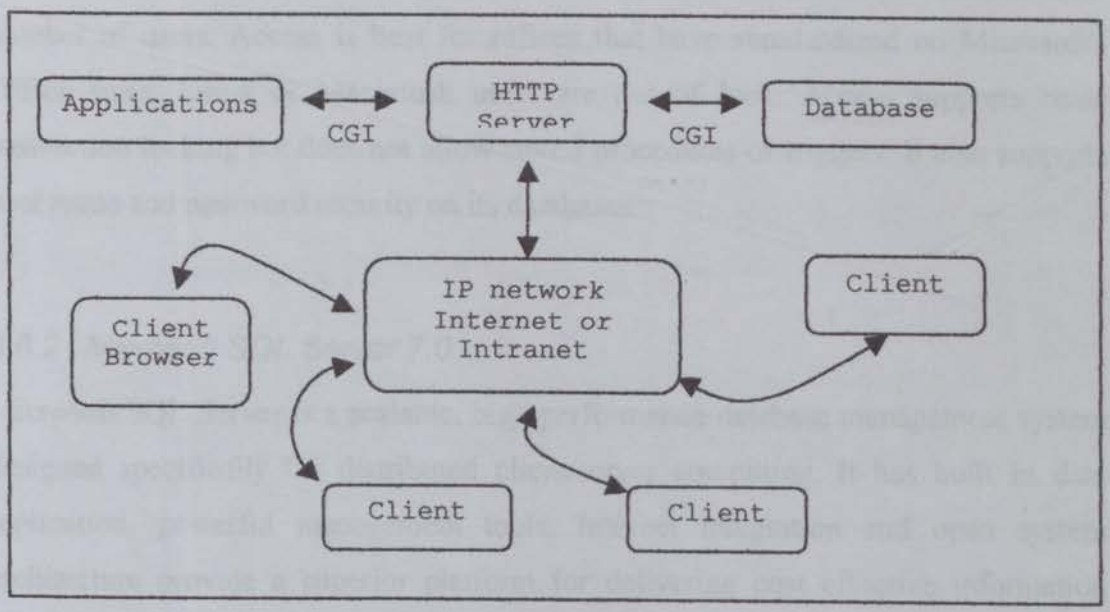


Figure 3-1: Three-tier Web Architecture

Based on the three-tier web architecture, the necessary elements of a web-based system are:

i. Web Browser such as Internet Explorer

ii. Web Server such as Internet Information Server (HTTP Server)

iii. Database Server

iv. Web Protocols between Database Server and Web Server, such as CGI, ASP

v. Programming Tools for Applications

3.6 Database Consideration

Two DBMS were analyzed in this session.

3.6.1 Microsoft Access 97 / 2000

Microsoft Access is a relational database management system created by Microsoft for small office or home user to use for storing data in relational format. Access is inexpensive and very easy to use, but it is not meant as an enterprise-level database. It is still a viable solution for small amounts of data and a small number of concurrent users. Small, of course, is relative, as its specification state that Access can hold up to 2 gigabytes of data per table. However, it only supports 255 concurrent connections, limiting its usefulness in an environment with a large number of users. Access is best for offices that have standardized on Microsoft's Office suite; Linux or Macintosh users are out of luck. Access supports basic transaction locking but does not allow stored procedures or triggers. It also supports user name and password security on its databases.

3.6.2 Microsoft SQL Server 7.0

Microsoft SQL Server is a scalable, high-performance database management system designed specifically for distributed client/server computing. It has built in data replication, powerful management tools, Internet integration and open system architecture provide a superior platform for delivering cost effective information solution.

SQL Server supports 4 terabytes of data per database, transaction locking, stored procedures, and triggers. It also integrates directly with Windows NT and Windows 2000 user accounts and security.

SQL Server is a suitable database engine for powering web site. Combined with Microsoft Internet Information Server and the SQL Server Internet Connector, customers have complete Internet database publishing capabilities. It supports for heterogeneous replication to non-SQL Server databases including Microsoft Access, ORACLE and so on. SQL Server's replication uses ODBC as the connection mechanism.

3.6.3 Summary

Microsoft SQL Server 7.0 outperformed Microsoft Access because Ms. Access is a smaller engine and not truly a client/server element, its use as a server system in a larger implementation is somewhat limited, but it still provides an unmatched development environment for database work. Microsoft Access is quite slow in processing transaction compared to Microsoft SQL server 7.0.

3.7 Server Consideration

3.7.1 Operating System/ Platform

An operating system (sometimes abbreviated as "OS") is the program that, after being initially loaded into the computer by a boot program, manages all the other programs in a computer. The other programs are called applications [14]. The applications make use of the operating system by making requests for services through a defined application program interface. In addition, users can interact directly with the operating system through an interface such as a command language.

There are two servers involved in the development of the system. The first is the platform server, and the latter is the web server. Several platform or servers have been looked into attentively.

3.7.2 Linux

Linux (often pronounced LIH-nuhks with a short "i") is a UNIX-like operating system that was designed to provide personal computer users a free or very low-cost operating system comparable to traditional and usually more expensive UNIX systems [15]. Linux has a reputation as a very efficient and fast-performing system.

Linux is a remarkably complete operating system, including a graphical user interface, X Window System, TCP/IP, the Emacs editor, and other components usually found in a comprehensive UNIX system. Although copyrights are held by various creators of Linux's components, Linux is distributed using the Free Software Foundation copyleft stipulations that mean any copy is in turn freely available to others.

Unlike Windows and other proprietary systems, Linux is publicly open and extendible by contributors. Because it conforms to the Portable Operating System Interface standard user and programming interfaces, developers can write programs that can be ported to other operating systems, including IBM's OS/390 running UNIX. Linux comes in versions for all the major microprocessor platforms including the Intel, Error, invalid term, Sparc, and Alpha platforms. Linux is distributed commercially by a number of companies. A magazine, Linux Journal, is published as well as a number of books and pocket references.

Linux is sometimes suggested as a possible publicly-developed alternative to the desktop predominance of Microsoft Windows. Although Linux is popular among users already familiar with UNIX, it remains far behind Windows in numbers of users.

3.7.3 Windows NT Server 4.0

Windows NT is the Microsoft Windows personal computer operating system designed for users and businesses needing advanced capability. Windows NT (which may originally have stood for "New Technology," although Microsoft doesn't say) is actually two products: Microsoft NT Workstation and Microsoft NT Server. The Workstation is designed for users, especially business users, who need faster performance and a system a little more fail-safe than Windows 95 and Windows 98). The Server is designed for business machines that need to provide services for Error, invalid term-attached computers. The Server is required, together with an Internet server such as Microsoft's Internet Information Server (Internet Information Server), for a Windows system that plans to serve Web pages [16].

3.7.3.1 Windows NT Workstation

32-bit applications will run 20% faster on this system than on Windows 95 (assuming both have 32 megabytes of RAM). Since older 16-bit applications run in a separate address space, one can crash without crashing other applications or the operating system. Security and management features not available on Windows 95 are provided. The Workstation has the same desktop user interface as Windows 95.

3.7.3.2 Windows NT Server

The NT Server is probably the second most installed network server operating system after Novell's NetWare operating system. Microsoft claims that its NT servers are beginning to replace both NetWare and the various UNIX-based systems such as those of Sun Microsystems and Hewlett-Packard. NT Server 5.0 expected to be available in early 2000, is now a product line called Windows 2000.

3.7.4 Platform Server Summary

After various factors and considerations have been carefully analyzed, Windows NT Server 4.0 and Internet Information Server 4.0 have been selected to be included in the system development platform. The detail of the system requirements and the features are discussed in section 3.8.

3.7.5 Web Servers

Based on the previous platform server consideration discussion, we have selected Windows NT Server 4.0, which comes with the Microsoft web server, IIS 4.0.

3.7.5.1 Microsoft Internet Information Server

IIS is a Microsoft Web server, which forms part of the Windows NT 4.0 Server default installation. Including FTP, HTTP and Gopher services, the IIS may be implemented to perform the following [4]:

- Downloading HTML pages to browsers
- Downloading streaming audio, video and multimedia.
- Downloading and uploading files using FTP.

All the services mentioned above or any other IIS server on the network is controlled by its Internet Service Manager (ISM). The ISM is run from a Windows NT Server or from a Windows NT or Windows 95 workstation. A HTML version of ISM can also be run on a browser to achieve the remote administration purpose [17].

The IIS is suitable for both small and large-scale websites, and is equipped with some extra tools that will surely facilitate the development of web applications. IIS includes ASP for building dynamic Web pages, Crystal Reports for custom reporting,

and Microsoft FrontPage 97 for site management. Furthermore, it also includes the Index Server for advanced searching, and NetShow for on-demand multimedia.

3.8 Web Programming

3.8.1 Active Server Pages (ASP)

Active Server Pages, commonly referred to as ASP, is Microsoft's solution to server-side scripting. With simple HTML pages, the client (a web surfer) requests a web page from a server. The server just sends the file to the client, and the page is shown on the client's browser. ASP scripts are similar to other server-side scripting such as Perl, Python, and so on [4].

To use Active Server Pages you must be running a Microsoft web server, specifically Microsoft Internet Information Server (IIS) 3.0 or up, Microsoft Personal Web Server (PWS) or O'Reilly Website Pro. If your web site is run on a UNIX box, you can still use Active Server Pages, but you need to use a third party tool to translate the ASP before it is sent to the client. Such companies like Chilisoft make products of this nature.

If you are running IIS 3 or up, to run an ASP file, all you need to do is create a file on the web server with an .ASP extension. When the browser requests the file, the web server is smart enough to preprocess the file before sending it off to the client.

ASP makes accessing databases very easy. ASP code can be connected to a database (SQL, Access, Oracle, Informix, or any ODBC-compliant database) and dynamically insert the data into your HTML pages. Although developers can use virtually any database library from within their scripts, ASP is really geared towards database access via Microsoft's ActiveX Data Objects (ADO). ADO is fairly lightweight and allows easy access to any ODBC or OLEDB compliant data source including Microsoft Access (Jet), Microsoft SQL Server, and Oracle databases. In addition, ADO has built-in connection pooling.

This leads to some very powerful possibilities including E-Commerce, customizable sites, data entering / retrieving systems run over the Internet, and a slew of other

possibilities. You can view some web sites that use ASP by visiting most any Microsoft sponsored site such as MSNBC.com or The Zone.

3.8.2 Common Gateway Interface (CGI)

The Common Gateway Interface (CGI) is a standard way for a Web server to pass a Web user's request to an application program and to receive data back to forward to the user. When the user requests a Web page (for example, by clicking on a highlighted word or entering a Web site address), the server sends back the requested page [4]. However, when a user fills out a form on a Web page and sends it in, it usually needs to be processed by an application program. The Web server typically passes the form information to a small application program that processes the data and may send back a confirmation message. This method or convention for passing data back and forth between the server and the application is called the common gateway interface (CGI). It is part of the Web's HTTP protocol.

Since the interface is consistent, a programmer can write a CGI application in a number of different languages. The most popular languages for CGI applications are: C, C++, Java, and Perl.

3.8.3 Summary

There are three main reasons to use ASP:

3.8.3.1 Performance

With CGI, each script is run as an executable in its own process. As a result, each time a script is requested the server must: create new process, run the script, kill the (just created) process. This is inefficient and can severely impact the performance of the Web server. The Active Server Pages host is not re-launched with every script access and is therefore much more efficient.

3.8.3.2 Session management

Because HTTP is stateless, keeping track of data between page accesses by a user is a tough problem. ASP provides built-in session management functionality that allows developers to persist data and also COM component instances (like database connections) for the duration of a session.

3.8.3.3 Easy integration of COM components

ASP is designed to rely heavily on COM components for its extensibility. As a result, it is very easy to instantiate and uses any COM component from within an ASP script.

3.9 Programming Language

3.9.1 Hypertext Markup Language (HTML)

HTML, which stands for Hypertext Markup Language, is a simple yet powerful markup language used to generate platform independent hypertext documents that are viewable by a Web browser. The markup tells the Web browser how to display a Web page's words and images for the user. The individual markup codes are referred to as elements [18].

The main thing that had made it so popular was its simple syntax. Therefore, it makes web programming work easy and simple but to certain extent. There is no prior knowledge of any programming languages needed to learn HTML.

HTML is a standard recommended by the World Wide Web Consortium (W3C) and adhered to by the major browsers, Microsoft's Internet Explorer and Netscape's Navigator, which also provide some additional non-standard codes. The current version of HTML is HTML 4.0. However, both Internet Explorer and Netscape implement some features differently and provide non-standard extensions. Web developers using the more advanced features of HTML 4 may have to design pages for both browsers and send out the appropriate version to a user.

3.9.2 Scripting Language

If you want to create scripts for your Web pages, the two primary scripting languages that you use are VBScript and JavaScript.

3.9.2.1 VBScript

VBScript is an interpreted script language from Microsoft that is a subset of its Visual Basic programming language [19].

VBScript is powerful. Various capabilities of VBScript can be used to develop richly interactive Web pages that respond to user input in an intelligent manner. For example, when a user submits a form, a VBScript subroutine can be triggered to verify that the form is properly filled in with valid values. In the case of a server-side CGI application, VBScript can be used to process data submitted by users with the aid of ActiveX controls specially designed for Microsoft Active Server Pages.

VBScript code is lightweight, fast, and optimized to be transmitted via the Internet. Because VBScript code is lightweight, it can be quickly transmitted to users browsing a Website.

Compared to scripting languages such as JavaScript, VBScript is easier to use because it is based on the easy-to-learn BASIC (Beginner's All Purpose Symbolic Instruction Code) language.

In fact, VBScript is a freely available scripting language. Microsoft has made VBScript freely available to software vendors so they can add scripting capabilities to their applications with the aid of VBScript.

VBScript is Microsoft's answer to Netscape's popular JavaScript. Both are designed to work with an interpreter that comes with a Web browser - that is, at the user or client end of the Web client/server session. VBScript is designed for use with Microsoft's Internet Explorer browser together with other programming that can be run at the client, including ActiveX control, automation servers, and Java applet. Although Microsoft does support Netscape's JavaScript (it converts it into its own JScript), Netscape does not support VBScript. For this reason, VBScript is best used for Intranet Web sites that use the Internet Explorer browser only.

3.9.2.2 JavaScript

JavaScript is an interpreted programming or script language from Netscape. It is somewhat similar in capability to Microsoft's Visual Basic, Sun's Tool Command Language, the UNIX-derived Practical Extraction and Reporting Language, and IBM's Restructured Extended Executor. In general, script languages are easier and faster to code in than the more structured and compiler languages such as C and C++. Script languages generally take longer to process than compiled languages, but are very useful for shorter programs.

JavaScript uses some of the same ideas found in Java, the compiled object-oriented programming language derived from C++. JavaScript code can be imbedded in HTML pages and interpreted by the Web browser (or client). JavaScript can also be run at the server as in Microsoft's Active Server Pages (Active Server Page) before the page is sent to the requestor. Both Microsoft and Netscape browsers support JavaScript, but sometimes in slightly different ways [20].

JavaScript is a relatively new scripting language, developed by Netscape Communications and Sun Microsystems, which has rapidly gained popularity among Internet developers. JavaScript preceded the European Computer Manufacturers Association (EMCA) standard and it was the first web scripting language created for dynamic interaction and content.

It is loosely related to Java, which is based on C++, except it is an interpreted language. JavaScript has, however, fewer capabilities than full-fledged object-oriented languages like C++ and Java. It is not a cut-down or simplified language though but rather a more limited language. However, it is not a true object-oriented language and it is limited in performance wise when compared to Java, as it is not compiled.

JavaScript is easy-to-use and designed for creating live online applications. It is analogous to VBScript. A JavaScript-compliant Web browser, such as Netscape Navigator and Microsoft Internet Explorer, is necessary to interpret JavaScript code. Like VBScript, JavaScript is based on a programming language, in this case Java, the Web darling. Unlike VBScript, JavaScript is completely object-based.

Basic online applications and functions can be added to web pages with JavaScript, but the number and complexity of available API functions are less than what is available with Java. JavaScript code can be included in a web page along with the HTML code, is generally considered easier to write than the Java language itself

3.9.2.3 Summary

Both VBScript and JavaScript have strengths and weaknesses, and this makes choosing the right scripting language for Web page or Web development project a challenge. The strengths that some users see in VBScript and JavaScript might also be viewed as weaknesses by other users, because of the primary technologies to which they are linked. Just as VBScript is tied to Microsoft technologies, JavaScript is tied to Netscape. VBScript works best in Microsoft Internet Explorer and JavaScript works best in Netscape Navigator.

3.10 Developing Tools

3.10.1 Microsoft Visual Basic 6.0

The Microsoft Visual Basic development system version 6.0 is the most productive tool for creating high-performance components and applications. Visual Basic 6.0 offers developers the ability to create robust applications that reside on the client or server, or operate in a distributed n-tier environment. Visual Basic 6.0 is the Rapid Application Development (RAD) tool available either as a stand-alone product or as a part of the Visual Studio 6.0 suite of tools [21].

3.10.2 Microsoft FrontPage 2000

Microsoft FrontPage 98 is a quick, effective way to create and manage professional-quality Internet or Intranet sites without programming. It makes it easy for new users and professional Web developers alike to build and maintain great looking, professional-quality Web sites in no time [22].

3.10.3 Matlab

MATLAB is an integrated technical computing environment that combines numeric computation, advanced graphics and visualization, and a high-level programming language [23].

MATLAB includes hundreds of functions for:

- Data analysis and visualization
- Numeric and symbolic computation
- Engineering and scientific graphics
- Modeling, simulation, and prototyping
- Programming, application development, and GUI design

MATLAB is used in a variety of application areas including signal and image processing, control system design, financial engineering, and medical research. The open architecture makes it easy to use MATLAB and companion products to explore data and create custom tools that provide early insights and competitive advantages [24].

3.11 System Requirement

3.11.1 Development Environment

The system requirements needed for the development environment for this project are:

a) System Unit
- At least Pentium II 233 MHz processor
- 32 MB RAM or more
- SVGA color display with 24-bit graphics
- Mouse
- Keyboard
- 2 GB hard disk

b) Software
- Microsoft Internet Explorer 5.0
- Microsoft Visual Basic 6.0
- Microsoft SQL Server 7.0
- Microsoft Word 2000
- Microsoft Front Page 2000
- Matlab ver 5.3 R 11

3.11.2 Server-Side Hardware Requirement

The following computer configuration was found to be the most suitable environment to deploy both Internet Information Server 4 (IIS) and Microsoft SQL server 7. It is wise to run both servers on different machines due to the fact that Microsoft SQL server is always hungry for CPU time and memory resource. This will prevent bottleneck from happening in the web server if both servers were to run on the same machine. Both Ms SQL server 7 and IIS 4 are run on Windows NT server 4 with service pack 5.

3.11.2.1 MICROSOFT SQL SERVER 7.0 MACHINE

- at least with a Pentium 166 MHz processor

- 64 MB RAM (recommended for performance wise)
- 9.4 GB hard disk (recommended for performance wise)
- Other standard computer peripherals

3.11.2.2 INTERNET INFORMATION SERVER 4.0 MACHINE

- at least with a Pentium 166 MHz processor
- 64 MB ram (recommended for performance wise)
- Network Interface card with bandwidth of at least 10Mbps or more
- 4 GB hard disk (recommended for performance wise)
- Other standard computer peripherals

3.11.3 Workstation Hardware Requirement

A different machine for workstation is essential, as this will separate development tasks from being done on the server. Any crash or coding and testing on workstation will not bring down or affect the server's state of running.

The recommended operating for development work is Ms Windows NT workstation 4.0. Although Windows 95 will do also, however it is not as stable as NT platform and Windows 95 can easily crash due to invalid page fault or other minor causes.

3.11.3.1 Windows NT Workstation 4.0 Machine

- at least a Pentium processor (recommended speed 166 MHz)
- 32 MB RAM for Windows 95,48 MB RAM for NT Workstation
- 2 GB Hard Disk
- Network Interface card with bandwidth of at least 10 Mbps or more
- Other standard computer peripherals

3.11.4 Client-Side Hardware And Software Requirement

3.11.4.1 Client Hardware Requirements

- At least with a Pentium 166 MHz processor
- At least 32 MB RAM

- Network connection through existing network configuration or modem (recommended at least 33.6 kbps).

3.11.4.2 Client Software Requirements

- Any operating system such as Windows, UNIX platform etc
- Any web browser, which supports JavaScript. (Recommended Microsoft Internet Explorer 4.0 and above).

3.12 Chapter Summary

As covered in this chapter, various software and hardware have been analyzed that make up the necessary elements of a web-based system. Therefore, platform requirements cum the software required are as follow:

3.12.1 Machine Requirements

- Windows-based PC
- At least Pentium II 233 MHz processor
- 32 MB RAM or more
- SVGA color display with 24-bit graphics
- Mouse
- Keyboard
- 2 GB hard disk

3.12.2 Software Required

- Windows NT Server 4.0 SP 5
- Information Interchange Server 4.0
- Microsoft Internet Explorer 5.0
- Microsoft Visual Basic 6.0
- Microsoft SQL Server 7.0
- Microsoft Word 2000
- Microsoft Front Page 2000
- Matlab ver 5.3 R 11
- Visual Prolog 5.0

CHAPTER IV

SYSTEM ANALYSIS & DESIGN

CHAPTER IV : SYSTEM ANALYSIS & DESIGN

In the previous chapter, the platforms and the software requirements have been discussed in details. This chapter will further introduce the system analysis and design. In system overview, the idea of the design of an agent is enclosed. The next section of the chapter will further elaborate how these ideas will be combined and applied in the system design. System modules and system flow will be available in this chapter, as well as Data Flow Diagrams, E-R Diagrams and tables in the database.

4.1 System Overview

This system has been designed based on the design of a search engine and the idea of GAs. Since the system is an intelligent agent that gathers information from the Internet and present it to the web users or through "Push" technology (sending emails), and as mentioned in Chapter 2, an intelligent agent is similar to a search engine, yet intelligent agent is for searching specific subject, while the search engine helps in searching for various general subjects.

For information retrieval from certain website, the web pages will be read through and retrieve only related and necessary information. That can be achieved in two major parts of information searching and retrieval.

i. Information Searching – searching for web pages that contain the related and relevant information.

ii. Information Retrieval – Retrieve the related information from the web pages and store into database.

4.1.1 Use of GAs in System Design

There are three most important aspects of using GAs [24]:

i. Definition of objective function (fitness function).
ii. Definition & Implementation of genetic representation.
iii. Definition & Implementation of genetic operators (crossover, mutation and reproduction/selection).

4.1.2 Use of Search Engine elements in System Design

For information retrieval, it is based on a search engine basic element:

i. "Spider" – Read through every single words & return the required information to next stage
ii. "Indexer" – Index the information gathered into the database
iii. "Search software" – When users search for specific printers, return the results acquired.

4.2 System Analysis

The system is developed based on the three important elements of the GAs, which are objective functions, genetic representations and genetic operators.

4.2.1 Definition of Objective Function

For this objective function, keyword on price of the printers, printer type, the manufacturer, and so on are compared with the HTML page to obtain the most appropriate information. For information retrieval, GAs can retrieve near-optimal solutions at its best. In many cases, it is not possible to obtain an absolute solution, but at its best effort for optimal solutions. The information obtained may be irrelevant to the search, therefore, GA plays an important in optimizing, the error of searching.

4.2.2 Definition & Implementation of Genetic Operators

The information retrieved from the HTML pages may include dumb values, or may contain characters that are not part of the information. GA can be applied here in

filtering the information retrieved from the HTML pages, either to accept more characters or to omit the unnecessary characters.

4.3 System Modules

The system will be divided in to three major modules, and each are interconnected with each other.

i. Members Area

- Includes registration of members, profile editing, and other utilities such as password retrieval, history search records folder and so on.

ii. Result Management

- This module should be able to search from the database and display the results that the user has requested. This module also includes the emailing of notices of updates in the websites.

iii. Administration Area

This module basically can be divided into three parts in order to gather information from the web and store into the database. When the user requests for specific information, it would then be displayed. The administrators should make sure that the agent gathers the appropriate and correct information.

a. URL Search

- This unit is responsible for searching the valid websites (URL address) that contain the related and necessary information.

b. Information Search

- This unit is responsible for searching and retrieve the information from the web pages provided by the previous module.

c. System Maintenance

- Sending notices or emails to registered users
- Ranking of websites
- Monitoring the system activities

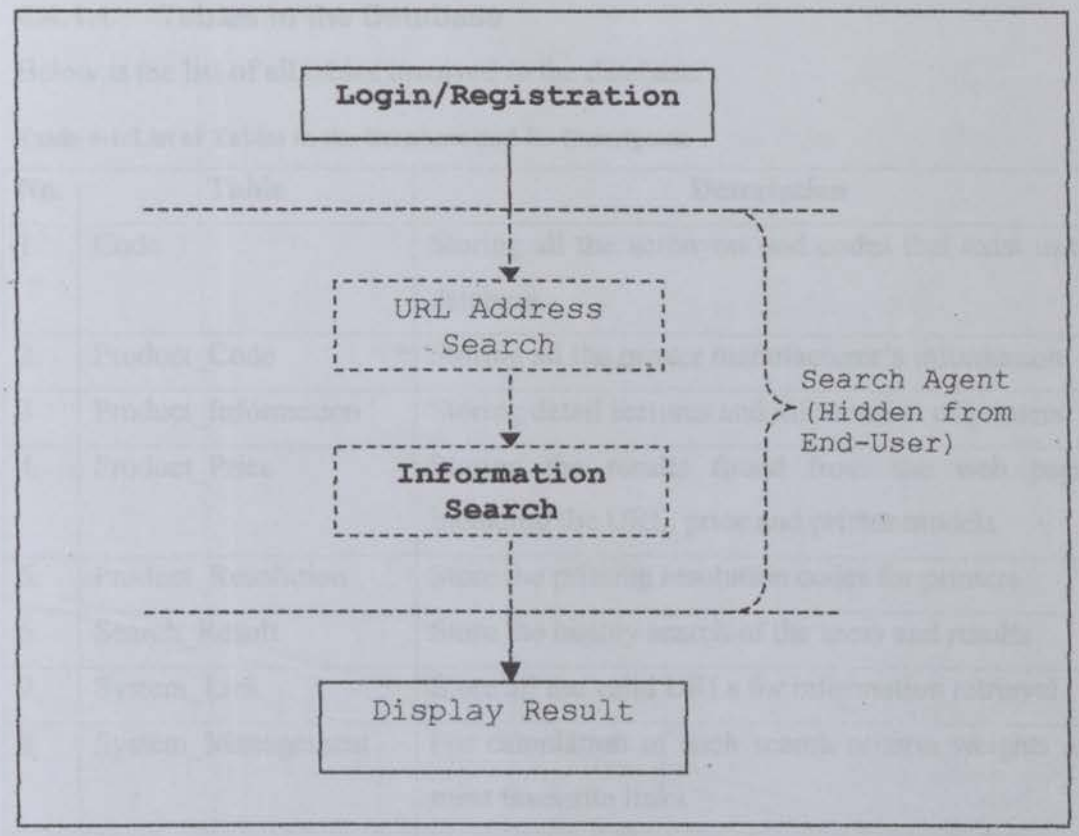


Figure 4-1: System Flow (Overview)

Since the project is accomplished in a group of two persons, we have identified the above three major modules and further divided into units that can be implemented by each individual. From figure 4-1, the modules boxes with dotted borders denote engaged by myself, while the rest of the modules will be accomplished by my teammate.

4.4 System Database

This section would introduce the tables that reside in the database. Since this is a group project, the tables are created based on the agreement between the team members. Some of the tables may be intercept within the modules that caused uploads or downloads of data into or from the database. In addition some of the tables are only applicable and utilizable to certain modules. Therefore in this section, only the tables that are concerned in the responsible modules are illustrated here.

4.4.1.1 Tables in the Database

Below is the list of all tables involved in the database:

Table 4-1:List of Tables in the Database and Its Description

No.	Table	Description
1.	Code	Storing all the acronyms and codes that exist in the database.
2.	Product_Code	Storing all the printer manufacturer's information
3.	Product_Information	Storing detail features and information of printers
4.	Product_Price	Storing the results found from the web pages, including the URL, price and printer models.
5.	Product_Resolution	Store the printing resolution codes for printers
6.	Search_Result	Store the history search of the users and results
7.	System_Link	Store all the valid URLs for information retrieval
8.	System_Management	For calculation of each search criteria weights and most favourite links
9.	System_SeachEngine	Storing all the search URLs
10.	System_Weight	Store all the weights for ranking of the websites.
11.	User_Information	Store all the registered users' personal details.
12.	User_Security	Store the login information of the users.
13.	Temp_Url	Store the search URL from the first part of the Search Module.
14.	XUser	Store the user information of administrators (for authorized personnel)

The previous table is a brief description of each table, and the details, such as the table fields, its data types and size will be included in the following section.

Table 4-2: Details of table CODE

Field Name	Data Type	Size	Description
XCodeType	Text	10	Category of Code
XCode	Text	10	The acronyms
Xdecs	Text	50	The original text
XUpDate	Date	m/dd/yyyy	The date it was updated

Table 4-3: Details of table PRODUCT_CODE

Field Name	Data Type	Size	Description
AutoID	int	4	Assign number for indexing
Manufacturer	Text	20	Name of the printer manufacturer
ManuCode	Text	10	The acronyms for each manufacturer
Website	Text	50	The official website of each manufacturer
Description	Text	50	The description

Table 4-4: Details of table PRODUCT_INFORMATION

Field Name	Data Type	Size	Description
ManuCode	Text	20	Product Manufacturer Code alias ManuCode in PRODUCT_CODE table.
Model	Text	20	The name or model of printer, e.g. BJC2100SP
Type	Text	50	The type of printer, e.g. BubbleJet, LaserJet, InkJet.
Speed	Number	Integer	The printing speed in ppm.
Port	Text	20	The printer port type, e.g. Parallel, USB.
Resolution	Text	10	The printing quality, measured in dpi.
Dimension	Text	20	The size of the printer, measured in mm.
Weight	Number	Decimal	The weight of the printer in kg.
Color	bit	1/0	A Colour printer or a black & white printer.

Table 4-5: Details of table PRODUCT_PRICE

Field Name	Data Type	Size	Description
AutoID	int	4	Assign number for indexing
Model	Text	20	The name or model of printer, e.g. BJC2100SP
URLName	Text	50	The name of the website found
URL	Text	200	The full URL address of the website.
Price	Number	Long Integer	The price of the printer in each website.
XCrDate	Date/Time	m/dd/yyyy	The date the record is inserted.
Xupdate	Date/Time	m/dd/yyyy	The date the record is updated.
Click	Number	Long Integer	The number of users that click on this link
TotalClick	Number	Long Integer	The total clicks of the link
Pos	Number	Long Integer	The overall ranking of the web site which depends on the calculation of weightage, most number of clicks ranking and cheapest price ranking.
Pos_Click	Number	Long Integer	The ranking of web site based on the number of user clicks
Pos_Price	Number	Long Integer	The ranking of web site based on the cheapest price.

Table 4-6: Details of table PRODUCT_RESOLUTION

Field Name	Data Type	Size	Description
AutoID	int	4	Assign number for indexing
XCodeType	Text	10	Category of Code
XCode	Text	10	The code used to represent each resolution
Xdecs	Text	50	The resolution of the printer, e.g. 720x720
XUpDate	Date	m/dd/yyyy	The date that the record was updated

Table 4-7: Details of table SEARCH_RESULT

Field Name	Data Type	Size	Description
AutoID	int	4	Assign number for indexing
LoginName	Text	10	The user login ID
Model	Text	20	The model of the printer
Price	Text	10	The Price of the printer
XCrDate	Date	m/dd/yyyy	The date that the record was created

Table 4-8: Details of table SYSTEM_LINK

Field Name	Data Type	Size	Description
AutoID	int	4	Assign number for indexing
URL	Text	200	The URL address of the website
URLName	Text	100	The name of the website
XcrDate	Date/Time	m/dd/yyyy h:mm:ss	The date and time that this URL was created
XUpDate	Date/Time	m/dd/yyyy	The date and time that the

		h:mm:ss	URL was updated
Found	Int	4	The number of times information is found
Used	Int	4	The number of times the information was used.
Active	Boolean	Yes/No	The URL can be used for next access

Table 4--9: Details of table SYSTEM_MANAGEMENT

Field Name	Data Type	Size	Description
AutoID	int	4	Assign number for indexing
Date	Date/Time	m/dd/yyyy h:mm:ss	The date and time that the weightage was updated
Weight_Price	Number	Double	The weight of the price (0<weight<1)
Weight_Click	Number	Double	The weight of the clicks (0<weight<1)
User	Text	50	The personnel who updated the record.
Stop_Click	Number	Long Integer	Stop learning if the number of clicks achieves the value in this field.
Activate	Boolean	1/0	Weight set activate

Table 4-10: Details of table SYSTEM_SEARCHENGINE

Field Name	Data Type	Size	Description
AutoID	int	4	Assign number for indexing
URL	Text	100	The URL address
Exception	Text	20	Search for URL except the words in this field.

XcrDate	Date	M/dd/yyyy	The date the URL address was inserted.

Table 4-11: Details of table SYSTEM_WEIGHT

Field Name	Data Type	Size	Description
Weight	Number	Double	Value for calculating the weight of number of clicks
Click	Number	Double	The number of clicks
XcrDate	Date	M/dd/yyyy h:mm:ss	The timestamp of this weight set was created
XcrDate	Date	M/dd/yyyy h:mm:ss	The timestamp of this weight set was updated

Table 4-12: Details of table USER_INFORMATION

Field Name	Data Type	Size	Description
LoginName	Text	10	The URL address
FirstName	Text	20	The first name of the registered users
LastName	Text	20	The surname or last name of the registered users
Gender	Text	2	The gender of the users, alias with CODE table, e.g. M or F.
Age	Number	Long Integer	The age of the users
DOB	Date	m/dd/yyyy	The date of birth of the user
Country	Text	20	The country of the users, alias with CODE table, e.g. MY, SG, etc.
Occupation	Text	20	The occupation of the users, alias with CODE table, e.g. ST for student or OT for others.

Email	Text	30	The email address of the users
SendNotice	Boolean	True/False	The option for sending notices to users via email
XcrDate	Date	m/dd/yyyy	The date the record was created.
XupDate	Date	m/dd/yyyy	The date the record was updated.

Table 4-13: Details of table USER_SECURITY

Field Name	Data Type	Size	Description
LoginName	Text	10	The Login ID of users
Password	Text	15	The password for users
SecretQ	Text	50	The hint question
SecretA	Text	20	The hint answer

Table 4-14: Details of table XUSER

Field Name	Data Type	Size	Description
User_Code	Text	10	The Personnel Code
User_Name	Text	50	The Personnel's name
User_Password	Text	10	The personnel password
Active	Boolean	Yes/No	The personnel is activated to access the system
Last_Update	Date/Time	m/dd/yyyy	Last user profile editing timestamp
Last_User_Login	Date/Time	m/dd/yyyy	Last user login timestamp
Email	Text	50	Email address

Table 4-15: Details of table TEMP_URL

Field Name	Data Type	Size	Description
URL	Text	200	The full URL address of the web site
UrlName	Text	200	The Web site's Name
XcrDate	Date/Time	m/dd/yyyy h:mm:ss	The date and time the address was obtained.

4.4.1.2 Data Model of the Database

After identifying all the necessary data to be kept track of, the tables and their properties as well as their attributes have been identified and shown in the above section. In this section, the further description of the relationships and interconnections between these tables are explained using the E-R Diagrams, and the database tool used in this system is the SQL Server which would well illustrate the ER Model of this database.

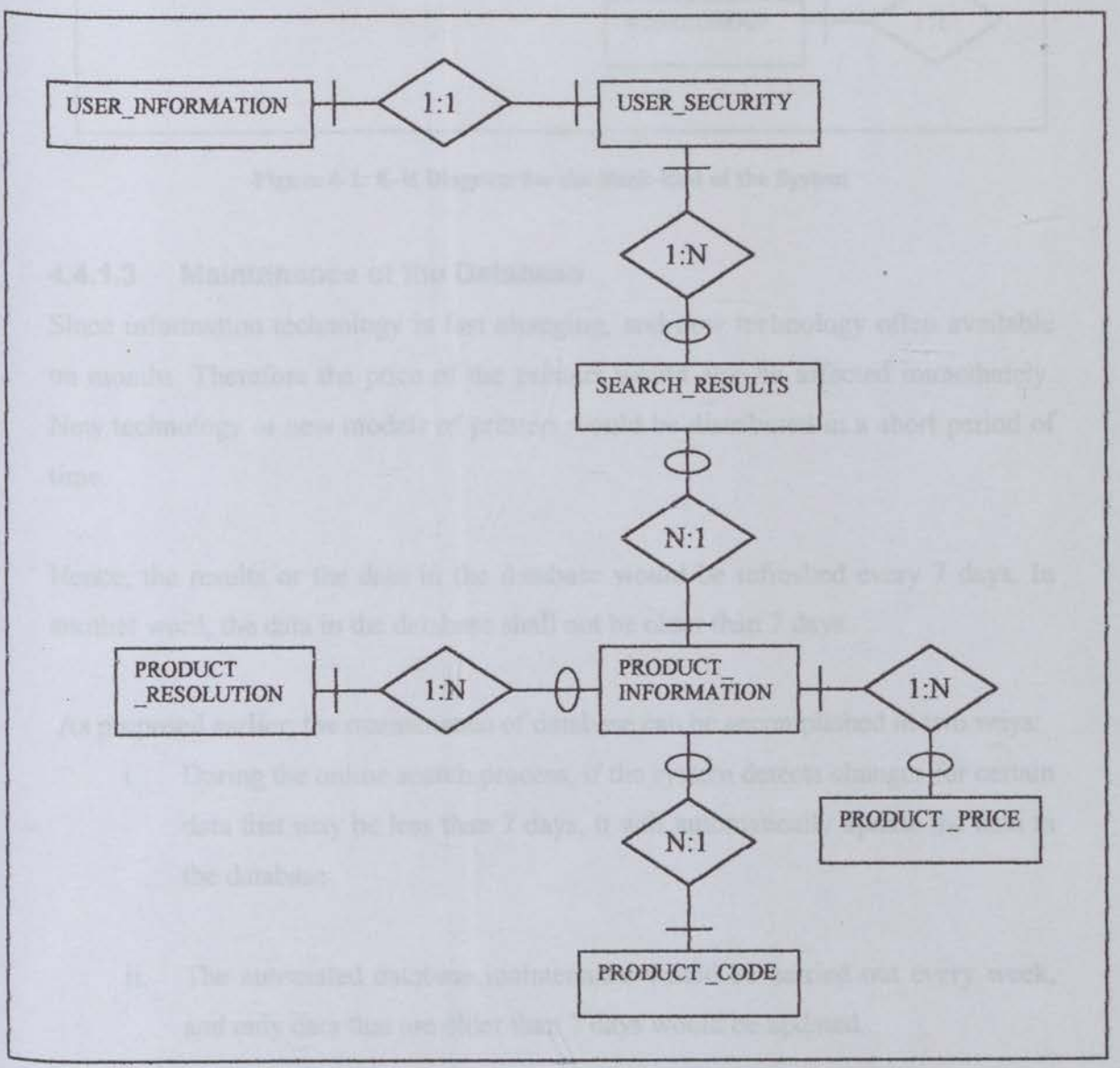


Figure 4-1: E-R Diagram for the Front-End of the System

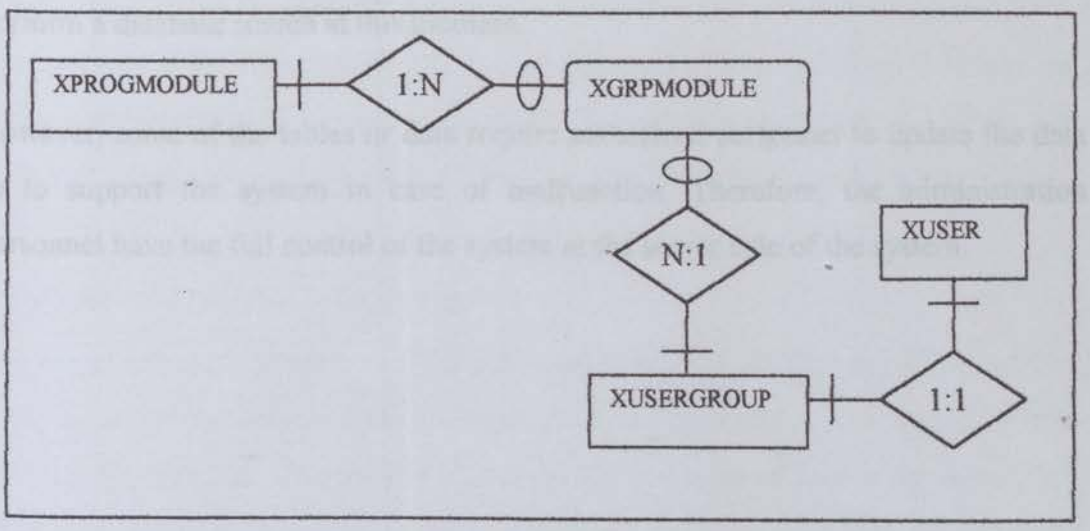


Figure 4-2: E-R Diagram for the Back-End of the System

4.4.1.3 Maintenance of the Database

Since information technology is fast changing, and new technology often available on months. Therefore the price of the printers would also be affected immediately. New technology or new models of printers would be distributed in a short period of time.

Hence, the results or the data in the database would be refreshed every 7 days. In another word, the data in the database shall not be older than 7 days.

As proposed earlier, the maintenance of database can be accomplished in two ways:

i. During the online search process, if the system detects changes for certain data that may be less than 7 days, it will automatically update the data in the database.

ii. The automated database maintenance would be carried out every week, and only data that are older than 7 days would be updated.

After various considerations, the database is updated every 7 days to keep the users well informed of the current situation of the printers. A timestamp would be placed on the results presented to inform the web users that the data found is valid on the

date stated to avoid any conflicts and confusion. Therefore, the users can only perform a database search at this moment.

However, some of the tables or data require authorized personnel to update the data or to support the system in case of malfunction. Therefore, the administration personnel have the full control of the system at the server side of the system.

4.5 System Flow & Design

This section only concentrates on Registration cum Member Area Module and second part of the Administration module – Information searching. It includes the Data Flow Diagrams and the system flowcharts for both of the modules.

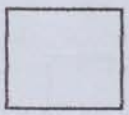
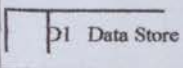
4.5.1 Member's Area Module Design

The Member's Area Module is further divided in to three smaller units that perform different tasks to accomplish the maintenance of the registered users. The three units are User Registration, User Profile Update and User Previous Search Records. These three units are described in the sections below in terms of data flow diagrams and flowcharts.

4.5.1.1 Data Flow Diagram

In this section, a DFD has been provided to understand the flow of the system and the flowchart is to provide better understanding of the flow of the system as well as its logical functionality.

Table 4-1: DFD Components

Component	Description
Entity 	- Depict a source that can send data or receive data from the system.
Flow of Data	- Shows movement of data from one point to another.
Process 	- Show the occurrence of a transformation process. - Process represent work being performed within the system.
Data Store 	- Represents a data store.

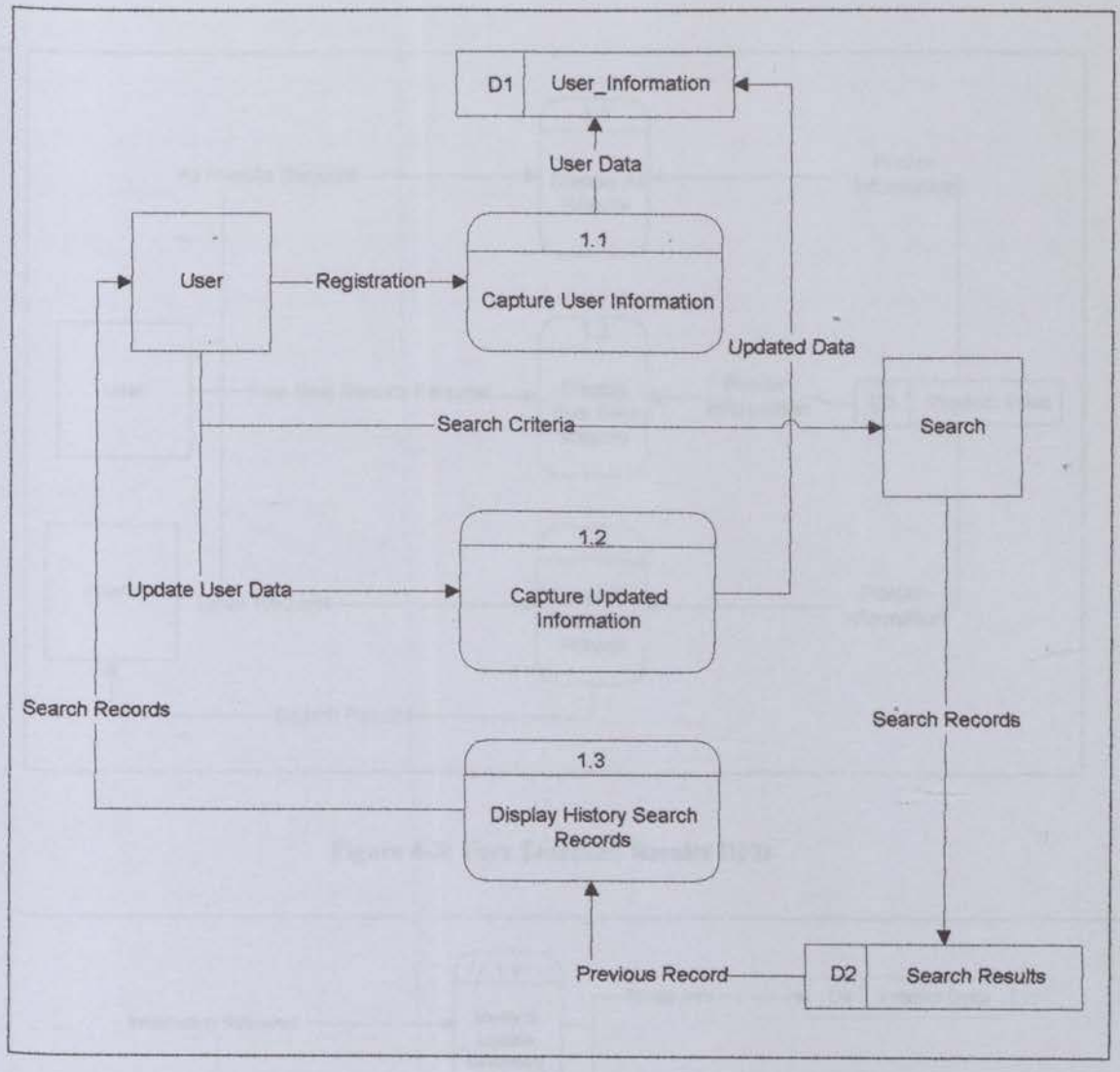


Figure 4-1: User Module DFD

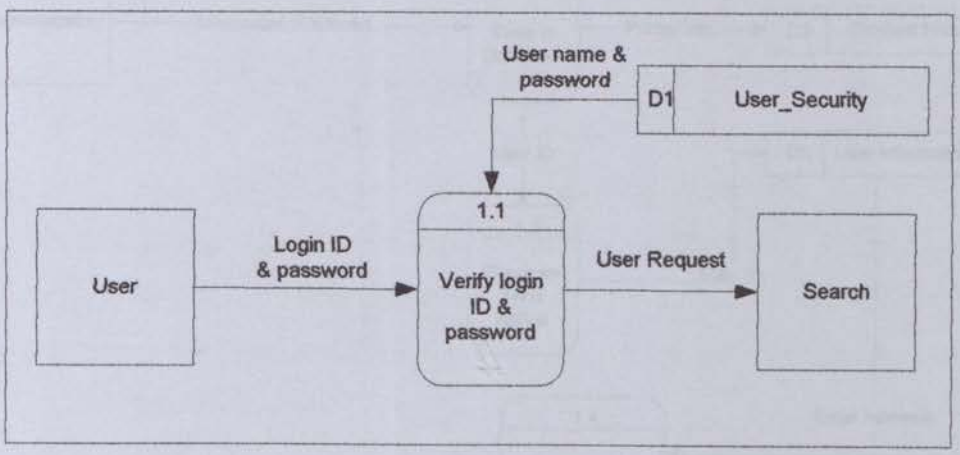


Figure 4-2: User Login DFD

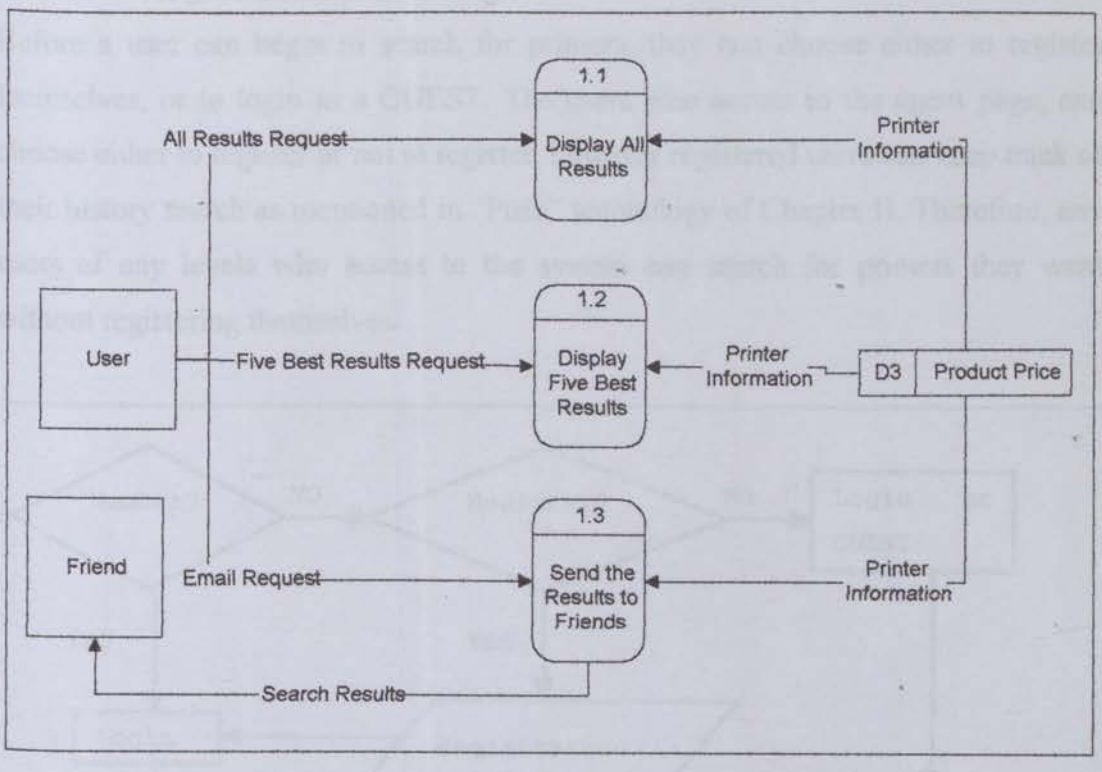


Figure 4-3: User Searched Results DFD

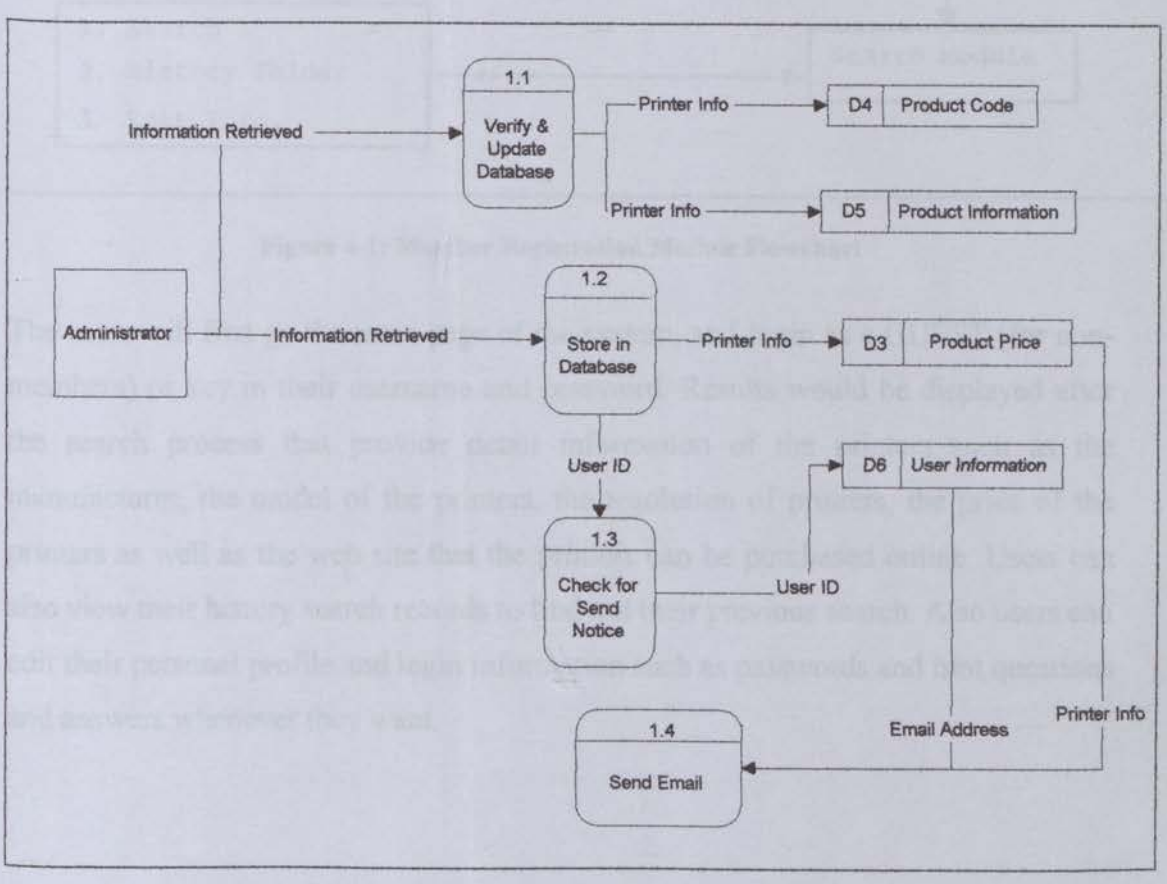


Figure 4-4: Administration Module DFD

4.5.1.2 Registration Unit Design

Before a user can begin to search for printers, they can choose either to register themselves, or to login as a GUEST. The users who access to the agent page, can choose either to register or not to register, however registered users can keep track of their history search as mentioned in "Push" technology of Chapter II. Therefore, any users of any levels who access to the system can search for printers they want without registering themselves.

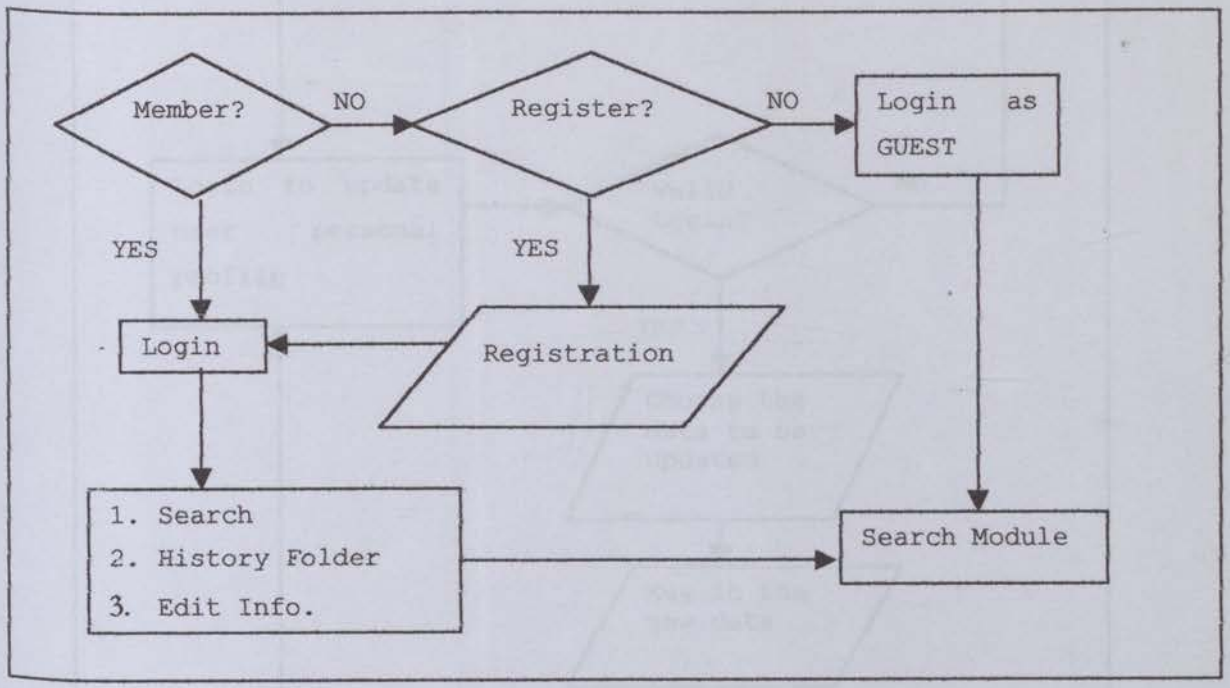


Figure 4-1: Member Registration Module Flowchart

The users will first go the main page of the system, and login as a GUEST (for non-members) or key in their username and password. Results would be displayed after the search process that provide detail information of the printers such as the manufacturer, the model of the printers, the resolution of printers, the price of the printers as well as the web site that the printers can be purchased online. Users can also view their history search records to find out their previous search. Also users can edit their personal profile and login information such as passwords and hint questions and answers whenever they want.

4.5.1.3 User Profile Update Unit

Users can change their profile at any time. The data that can be modified or updated are user's name, country, occupation, email address, option on sending notices, password, hint question and hint answer. Below is the flow of the user profile update module.

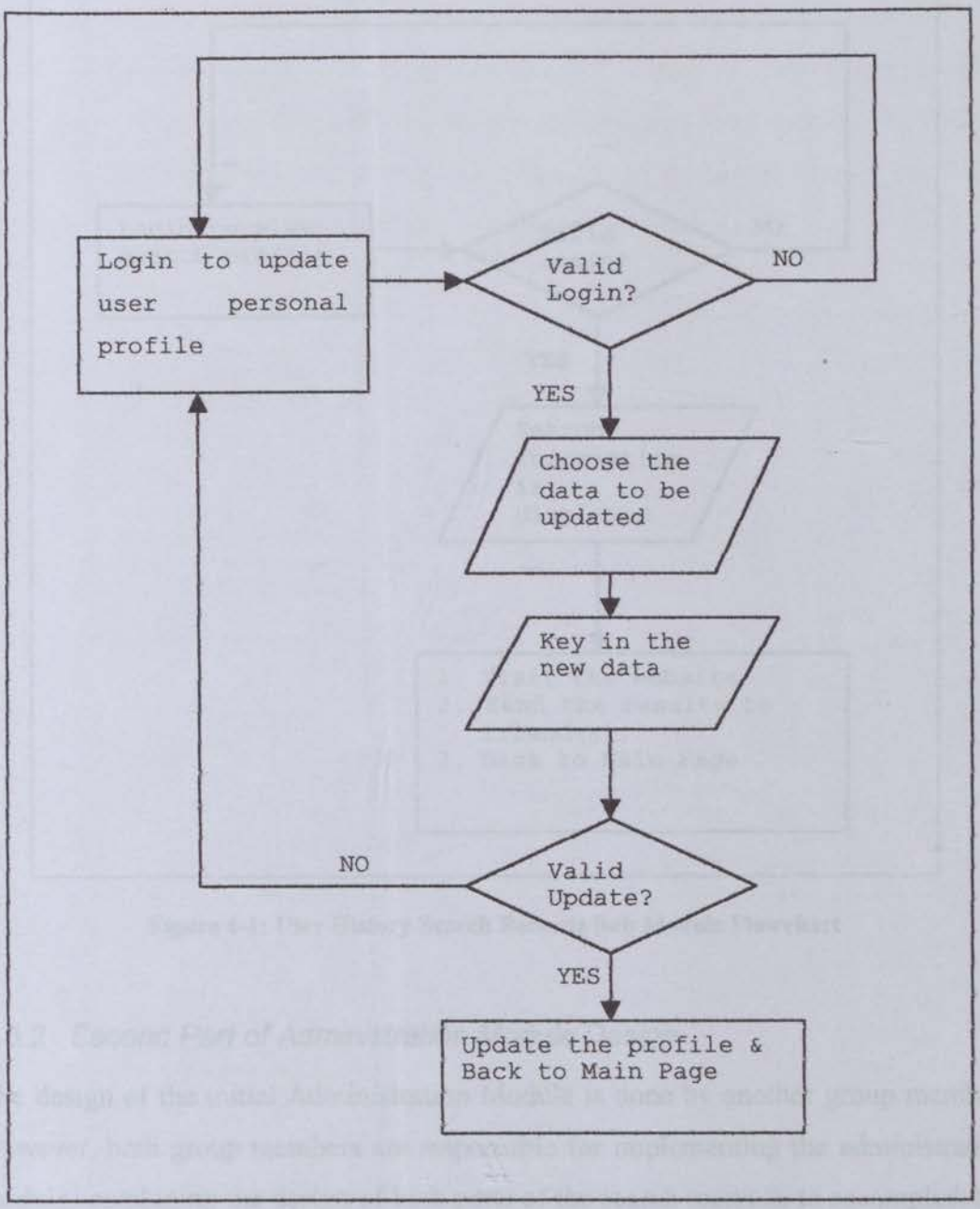


Figure 4-1: User Profile Update Sub Module Flowchart

4.5.1.4 User Previous Search Record Unit

This sub module is able to display what the users have searched before this login. Some users may need to refer back what have they found. The flowchart for this sub module is as below:

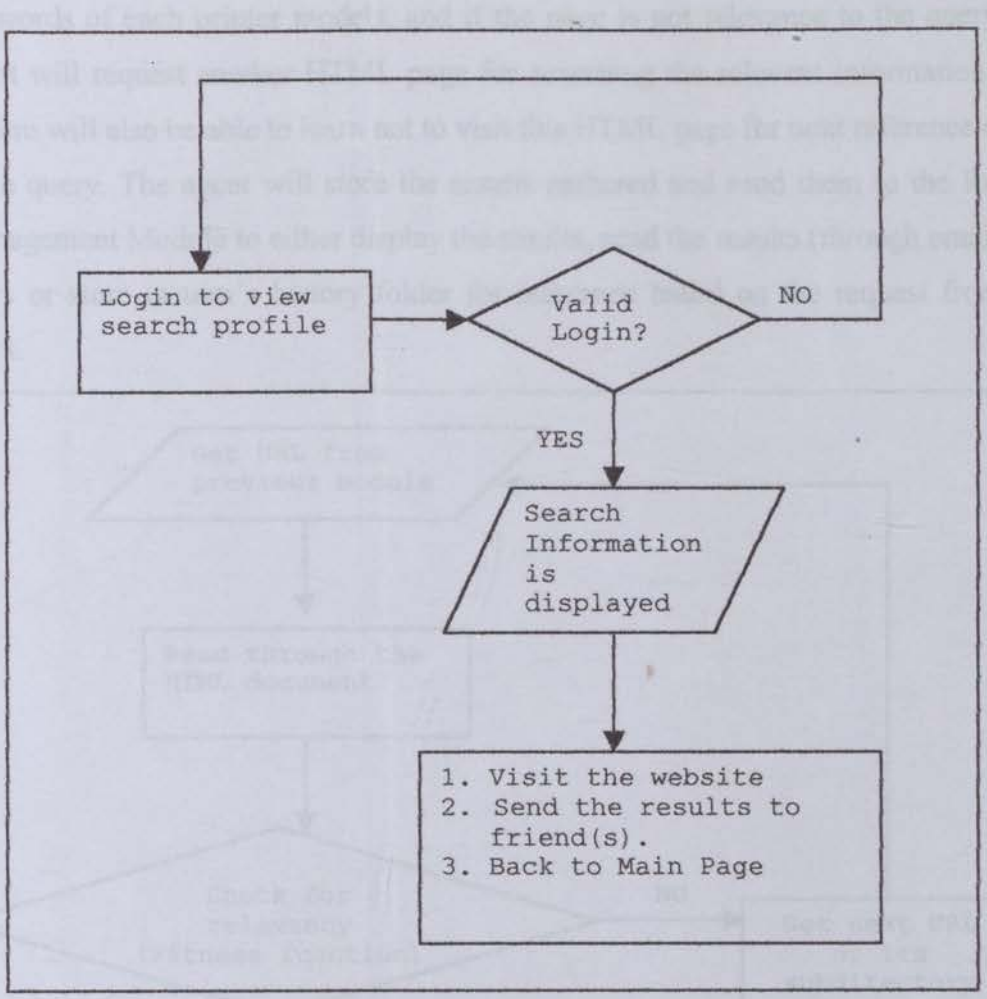


Figure 4-1: User History Search Records Sub Module Flowchart

4.5.2 Second Part of Administration Module Design

The design of the initial Administration Module is done by another group member. However, both group members are responsible for implementing the administration modules combining the design of both parts of the search methods to accomplish the information retrieval and store.

4.5.2.1 Information Search and Retrieval within HTML Page

The initial part of the search agent will send a valid URL to the second part of the agent to read for relevant data. After HTML page has been passed to the agent, it will read through the web page and gather the information based and keep them into the database. Relevancy of the page is checked with the fitness function, which is the keywords of each printer models, and if the page is not relevance to the query, the agent will request another HTML page for searching the relevant information. The system will also be able to learn not to visit this HTML page for next reference of the same query. The agent will store the results gathered and send them to the Results Management Module to either display the results, send the results (through emails) to users or store in user's history folder for reference based on the request from the users.

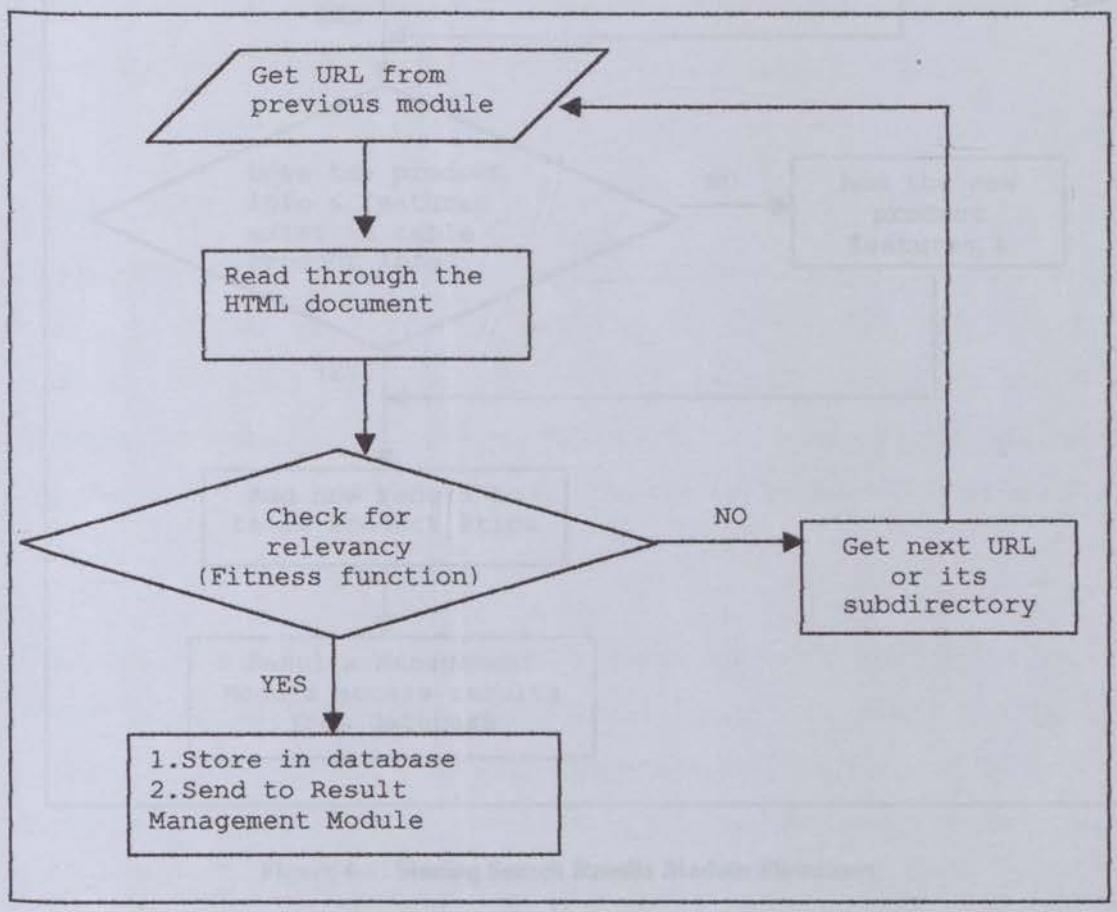


Figure 4-1: Second Part of Administration Module flowchart

4.5.2.2 Storing Search Results Unit

After extracting data from the web pages, it is then stored to the database so that the results can be taken from the database and be displayed or sent to the users for the

next Result Management Module. The flow of the search results storing process is demonstrated in the following flowchart:

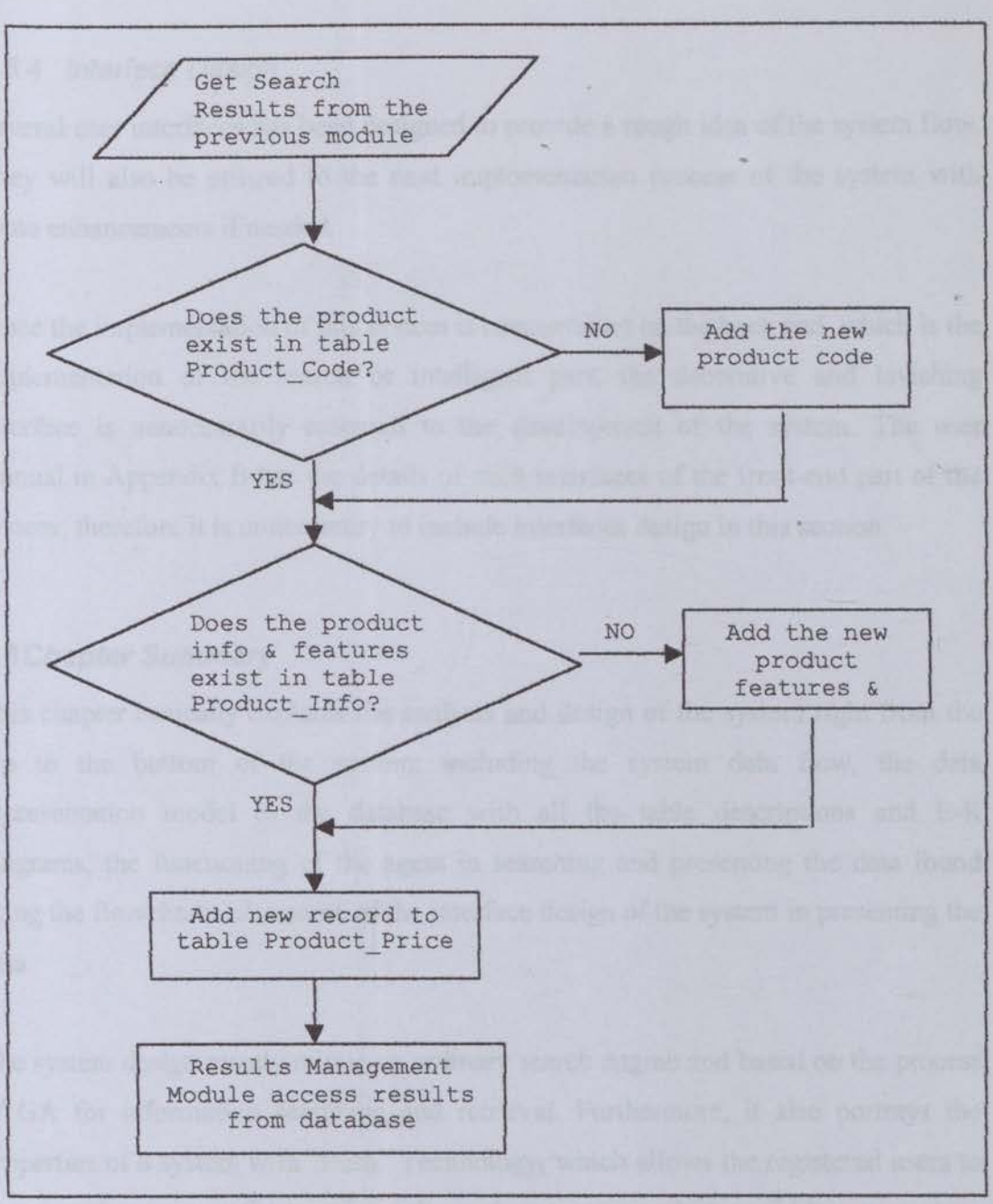


Figure 4-1: Storing Search Results Module Flowchart

4.5.3 System Maintenance

The system is automated in certain parts, such as the module of getting valid URL, and sending emails to users. Administrator will need to monitor the system activities

so that the agent performs the appropriate tasks. This also considered the back-end of the system, where users are unable to view he processes.

4.5.4 *Interface Design*

Several user interfaces has been designed to provide a rough idea of the system flow. They will also be utilized to the next implementation process of the system with some enhancements if needed.

Since the implementation of this system is concentrated on the back end, which is the implementation of the search or intelligent part, the decorative and lavishing interface is unnecessarily essential to the development of the system. The user manual in Appendix B has the details of each interfaces of the front-end part of the system; therefore it is unnecessary to include interfaces design in this section.

4.6 Chapter Summary

This chapter basically explains the analysis and design of the system right from the top to the bottom of the system, including the system data flow, the data representation model in the database with all the table descriptions and E-R diagrams, the functioning of the agent in searching and presenting the data found using the flowcharts, also some of the interface design of the system in presenting the data.

The system design mostly mimic an ordinary search engine and based on the process of GA for information searching and retrieval. Furthermore, it also portrays the properties of a system with "Push" Technology, which allows the registered users to choose either to store the search results in their respective history folders, or receive the results via email address, that they have filled in during the registration.

CHAPTER V

SYSTEM IMPLEMENTATION

CHAPTER V : SYSTEM IMPLEMENTATION

This chapter is the beginning of the second part of the final year thesis project, which is the system implementation. The chapter explains in detail the software and hardware used for implementing the proposed system in the first part of the report. This chapter also explains the detail steps in implementing the system.

Usually, in a software prototyping project, the requirement analysis, system design and implementation phases do not have clear boundaries. Each phase overlap one another. System implementation is a process that converts the system requirements and design into program codes. This phase always involves some modifications to the previous design due to the limitation of the programming languages used.

5.1 System Implementation Environment

Development environment has a certain impact on the development of a system. Procuring suitable hardware and software will help to speed up system development. The hardware and software used for implementing the proposed system after various considerations and studies conducted were introduced in this section.

5.1.1 Hardware Used

As proposed earlier in Chapter III section 3.12.1 regarding the basic hardware requirements, the following hardware have been dedicated, configured and utilized for the implementation of the final year thesis project:

- Windows-based PC with P III 266 MHz processor
- 128 MB RAM
- 4 GB hard disk space
- SVGA color display with 24-bit graphics
- Mouse, Keyboard
- Floppy Disk Drive & CD-ROM Drive

5.1.2 Software Used

The software chosen to implement this system has been studied carefully in Chapter III to support the hardware that has been dedicated to this project. Some of the

software mentioned in section 3.12.2, have been omitted as better tools have been adopted. The followings are the software used:

- Windows NT Server 4.0 SP 6
- Information Interchange Server 4.0 with SMTP and NNTP services
- Personal Web Server
- Microsoft Internet Explorer 5.5
- Microsoft Visual Basic 6.0 SP 4
- Microsoft Visual InterDev 6.0
- Microsoft SQL Server 7.0
- Microsoft Access 2000

In section 3.12.2, Matlab and Visual Prolog have been proposed to implement the system, especially for the back-end of the system. Various journals and papers have been considered, and a few solid reasons that lead to the use of Microsoft Visual Basics. Firstly, there is no GA toolbox provided by the Matlab software, and the GA functions are easier said than done. Although GA toolbox description can be obtained by other sources, the functions are not very practicable and applicable to the implementation of this system. Secondly, using Visual Prolog is not as easy as using the ordinary Prolog, therefore the coding of the system design is very tedious and complex to be accomplished. Thirdly, the Visual Basic software can be used to accomplish the same tasks and responsibilities. The Visual Basic with SP 4 (Service Pack 4) is used as some of the web components and advanced coding can only be achieved with the Visual Basic SP 4 package such as Internet Control, Internet Transfer Control, Rich Textbox Control, and some Windows Common Control are only included in SP 4 of Visual Basic 6.0.

IIS 4.0 with SMTP and NNTP services are used for the system for actual activities. The Personal Web Server (PWS) is used for testing and implementation of the system using home PCs.

The Microsoft Access 2000 is used for system implementation and testing using PCs at home that run on the Windows, while the Microsoft SQL Server is practically in use for the system at the faculty's dedicated Windows-based PC that operates using

Windows NT Server 4.0. The database files are updated to the SQL Server periodically after each task has been completed.

Typically, Microsoft Visual InterDev 6.0 is used for ASP coding, VBScript and as an interface between the database server and the web server. The coding and testing are mainly done using this software.

A small number of software, such as Microsoft Word 2000 has been used for editing test documents, are not included in the list for these were the software that do not affect the implementation of the system. Microsoft FrontPage 2000 is also used for viewing the web pages in HTML codes for easier testing on the validity and integrity of data obtained from the AI part of the system.

The software used and its usage is included in the following table.

Table 5-1: List of Software used and its purpose

Software	Purpose
Windows NT Server 4.0 SP 6	The operating system for the whole development environment.
IIS 4.0 with SMTP & NNTP services	The web server for the project. SMTP for sending notices to registered users. NNTP for publishing news
Internet Explorer 5.5	The browser for testing on web pages and interfaces
Visual Basic 6 SP 4	For implementing the back end of the system.
Visual InterDev 6.0	An interface between web server and database server. Web pages are developed here.
Microsoft SQL Server 7	The database server for the project.
Microsoft FrontPage 2000	The editor for HTML documents.

Microsoft Personal Web Server	For implementing web server at home PC.
Microsoft Access 2000	The database tool for development of the system at home using Windows PC.
Microsoft Word 2000	The editor for documentation (User Manual).

5.2 Development of the System

The system is divided into a few modules as proposed, and each module is carried out by either my teammate or myself respectively as agreed before. I was involved in the user module and read through the web pages, when valid URLs are received.

There are a few more sub modules in the user module, which are user registration module, user update profile module and user search history module. The flowcharts of these sub modules are illustrated in section 4.5 of this report. This part of the system is the front-end of the system that is visible to the end-users.

The second part of the administration module involved receiving valid URL from the previous part of the module, which is developed by my teammate and "read" through the HTML web pages to extract the relevant data and information that is then to be stored in the database. This part of the system is the back-end of the system. It is also considered part of the AI part of the system.

Although ADO object is similar to RDO and DAO in other Microsoft programming languages, it has some rules that have to be followed. For example, when opening a recordset only to find and read a record, adLockReadOnly instead of adLockOptimistic has to be taken as the parameter. Otherwise, it will generate errors when it is executing. Therefore rules for the ADO has to be understand first before using the recordset to manipulate data in the database.

Both functions for retrieving data and inserting data from and into the database have been defined in another file for coding efficiency and to avoid any typing errors in the parameters. The codes for the two mentioned functions are included below:

```
Function RetriveData(dcnDB, strSQL)
        Dim rsData
        ' Create recordset and retrieve values using the open connection
        SET rsData = server.CreateObject ("ADODB.Recordset")
        rsData.CursorLocation = adUseClient
        rsData.Open strSQL, dcnDB, adOpenStatic ,adLockReadOnly
        set RetriveData = rsData
End Function

Function WriteData(dcnDB, strSQL)
        Dim rsData
        ' Create recordset and retrieve values using the open connection
        SET rsData = server.CreateObject ("ADODB.Recordset")
        rsData.CursorLocation = adUseClient rsData.Open strSQL, dcnDB, adOpenKeyset,
                                adLockPessimistic
        set WriteData = rsData
End Function
```

The user-defined functions are used as below:

2.1. To access the data for read only,

```
Set dcnDB = OpenDatabase ()
Set rsData = RetriveData (dcnDB, strSQL)
```

Where rsData represents the recordset, dcnData is the database connection, and strSQL is the SQL statement.

2.2. To save data into the database

```
Set dcnDB = OpenDatabase ()
Set rsData = WriteData (dcnDB, strSQL)
```

Since all the database options have been defined in another function, it is not necessary to include database options in the recordset statements.

5.2.1 User Registration

Based on the table properties that have been identified earlier in Table 4-12 User_Information, the data to be captured has to be identified first. Data to be captured from user are all the attributes exist in the table. Only one registration form is used for capture user's data using normal ASP codes. Some of the data are taken from the database for user to choose from, such as country of origin, occupation that only a few options are provided, the codes for accessing data from database are

```
<option VALUE="<%= rsData.Fields("xcode") %>">
        <%= rsData.Fields("xdesc")%></option>
```

And each attribute of the database is stored into the database using recordset as below:

```
rsData.AddNew
rsData.Fields ("FirstName") = Request.QueryString("firstname")
raData.Update
```

5.2.2 User Profile Update

First, the items that can be changed or updated by users are identified, because user cannot change his or her Login ID for this update would create a lot of misleading links of other tables. Therefore, only user's name, country of origin, occupation, email address, password, option on receiving emails, password and hints can be updated or modified by users. User only has to click on the data to be updated and enter the new data. Passwords are also modified here.

5.2.3 User History Records

The user search records are stored into the database when the users search for printers. Therefore, the users' search records are stored periodically. When users want to view previous search records, the data are retrieved from the Search_Result table and displayed to the users.

5.3 Integration Process

This thesis project involves my teammate and myself; therefore we have to agree on certain milestone on delivering the required components before advancing into new stage in completing our project.

Some components, namely, user registration module, user profile update module and some basic web pages of the system, such as web page to display the terms and conditions of the web site, can be developed simultaneously while waiting for the previous module to be completed, in order to get the valid URL passed to the second search module. Both of the teammates were very rational person, who can accept and appreciate the works done by each other.

Regular meetings were carried out whenever we were free, and since we are classmates and take the same courses, we even have discussions very often so that both of the teammates would not feel left out. Formal meetings were carried out to present the works that have been completed so far in order for us to keep on schedule. Since the computer lab was always available for usage, files or tasks that have been accomplished were updated to the system immediately after completion. This was to let the other member well informed, and if possible provides comments.

5.4 Implementation of Artificial Intelligence into the System

The application of the AI is implementing using Visual Basic SP4 that communicate via the database of the system. The web server would access and update the data in the database, as well as the Visual Basic Application of the system.

This part of the system is to read through the HTML page. First, the form must be included with Microsoft Internet Transfer components to enable HTTP access and transfer of HTML pages.

5.4.1 Retrieving URL from Database

First, the valid URL that has been identified and verified is retrieved from the database. Each URL has been verified of its accessibility and in relation to printer information. Each URL will be updated after it has been retrieved from the database and used for searching information. The URL is updated in terms of its usability for information searching next time, and whether data are found from the URL HTML codes.

Below are the partial codes for getting URL from the database:

```
StrSql = "SELECT url "
StrSql = StrSql & "FROM System_Link "
StrSql = StrSql & "WHERE active = yes ;"
rsOpen.Open StrSql, adconn, adOpenStatic, adLockReadOnly
If Not rsOpen.EOF Then
    rsOpen.MoveFirst
     For j = 0 To i
            strtemp = rsOpen!URL
            rsOpen.MoveNext
            If rsOpen.EOF Then
             j = i
            End If
    Next
i = i + 1
End If
txtURL.Text = strtemp
```

5.4.2 *Extract the HTML Codes of the URL*

The URL obtained from the database has to be converted into HTML codes in order to read through every single word to retrieve the related information. The retrieve algorithm is a public function in the Visual Basic Project, because module of getting valid URL addresses also need to convert the results obtained from the search engine into HTML codes, before the links are retrieved and stored into database.

```
mstrHTML = Inet1.OpenURL(inputUrl)
Inet1.Protocol = icHTTP
Inet1.Execute CStr(inputUrl), "GET / "
While Inet1.StillExecuting
    DoEvents
Wend
RetriveHTML = mstrHTML
```

The function above is supposed to return the HTML codes into a string. The string is then displayed first in a rich text box. The HTML codes are then passed to next function to obtain the related information.

```
TxtHTML = RetriveData (strTemp)
```

5.4.3 Obtain the Printer Information from HTML codes

This unit involves counting of characters and how each website organize their printer information.

Some website put their printer information in tables, therefore, keyword <table> can be used. And some websites use <frameset> and in this frame, there are more than one HTML files that help to build up the web page. Therefore, the HTML page is read from the first character till the last character. Each character is compared with the keywords of the printer models or printer type. Then the price of the printer is searched right after the position of the first printer model or information found. The information is then stored into the database.

5.5 Chapter Summary

This chapter basically outlines the process of the development of the system and the distribution of works between the teammates. The hardware and software that were used in completing the system are included here, as well as the synchronization of works to be carried out. The implementation of related modules and its sub units are also included in this chapter.

The next chapter will outline the various steps that have been taken in order to test on the modules and the system.

CHAPTER VI

SYSTEM TESTING

CHAPTER VI : SYSTEM TESTING

This chapter explains the testing of the system in various stages and using all possible values and responds by users. The testing phase also involves the error checking reliability of the system.

6.1 Testing Strategy

Testing is by definition a process of examining the project and its components to ensure that its performances are similar to the agreed specifications. This includes the look and feel, the interactivity and, of course the robustness of the codes. In today's computing world, there exists an infinite of system configurations. Hence, it is a good practice to test as early as possible, as often as possible and on as many respondents as possible.

Below are the testing strategies that have been conducted on the system that has been developed through the whole development process.

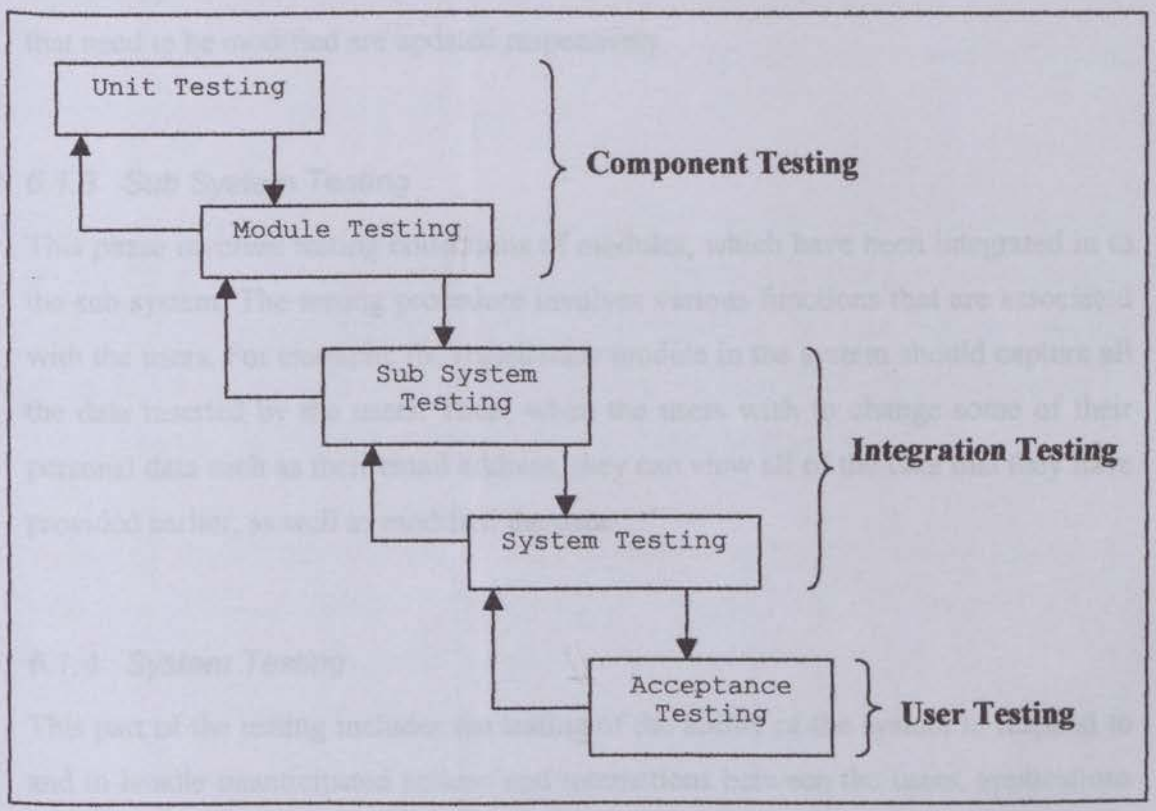


Figure 6-1: The Stages of Testing Strategy

6.1.1 Unit Testing

Unit testing involves the smallest system unit whereby individual components are tested to ensure they are operating properly and correctly. Each component is tested independently, without other system component.

6.1.2 Module Testing

A module encapsulates related components and can be tested without the need for the other system module. An example of a module that can be tested upon completion is he system registration module.

Systems are normally tested at every stage to avoid any mistakes and to facilitate the debugging process. The web pages developed for the users to register, to update profile and to view history folder have been tested concurrently whenever a web page was completed. This was to avoid any difficulties in trying to rectify any errors that occurred. These three modules can be tested independently to ensure that data inserted by the users are captured correctly and completely into the database. Data that need to be modified are updated respectively.

6.1.3 Sub System Testing

This phase involves testing collections of modules, which have been integrated in to the sub system. The testing procedure involves various functions that are associated with the users. For example, the registration module in the system should capture all the data inserted by the users. Then, when the users wish to change some of their personal data such as their email address, they can view all of the data that they have provided earlier, as well as modified the data.

6.1.4 System Testing

This part of the testing includes the testing of the ability of the system to respond to and to handle unanticipated actions and interactions between the users, applications and its environment. This testing also concerned with finding errors, which normally result from unanticipated interactions between sub systems and components. It is

also concerned with validating that the system meets its functional and non-functional requirements.

6.1.5 Acceptance or Interface Testing

Basic arrangement of the interface was carried out at the first trial of the testing. The interface was updated using the method of trial and error, because, the location of the links to be placed or the design of the interface for users are very prejudice according to ones' preferences.

The interfaces were sent to other users to try on, in order to receive valuable responds and comments from them. The respondents were usually the family members, friends and classmates, because they were the potential users, who can understand the thoughts and preferences of the public. The responds and comments were then reviewed and the interfaces were altered if necessary.

6.2 System's Intelligence Testing

This section is the most important testing of the system because it proves that system is intelligent and is able to act like a human.

Various HTML pages and websites are tested on the module. During the unit testing, all other module has been disabled to fasten the process of running and debugging of modules. The unit may not be applicable to some websites because they are in frames or use images to represents their data. Therefore, the choices of which websites to be tested are very important in order to collect the right information.

Before the real website is tested, the pre-downloaded HTML files are used to test the unit. The Common Dialog box is used to open a file, and this Common Dialog component can only be found in Visual Basic Service Pack 4.

6.3 Chapter Summary

This chapter basically reveals how the various tests were accomplished and the purpose of the tests. The testing of the system were carried out to further ensure that the system can accept and debug the errors arise.

CHAPTER VII

PROBLEMS ENCOUNTERED

CHAPTER VII : PROBLEMS ENCOUNTERED

Any given tasks would encounter problems, and regardless of the severe level of the problems, they would prevent us from advancing into the next stage of the system development. Basically, there were no hardware problems as we were concerned until the end of the system development.

7.1 Software Problems

All the required software that was needed for the implementation of the system was available at the faculty, and we can easily obtained them from the System Analyst of the faculty. We rarely encountered any major software problem, except the following that caused us to alter our earlier proposed plan.

7.1.1 Genetic Algorithm Toolbox was not provided in Matlab

The GA toolbox was not provided in the Matlab software, unlike the Neural Network Toolbox. Therefore the work of implementing the GA into the system using AI programming tools is very difficult to achieve. There were GA functions used for Matlab written by lecturers from University of Sheffield, U.K. [25] found over the Internet, unfortunately, the functions were not very practicable and applicable to the system that we were developing because the functions are more to the biological genetics implementation of the system. Therefore, we have decided to use the Visual Basic 6.0 to perform the characteristics of the Genetic Algorithm.

7.1.2 Microsoft Internet Transfer Control Component

Some of the Internet Controls components are not included in the normal version of Visual Basic. In order to use these components to extract HTML files and to get the valid links, these components have to be installed. These components are only provided by the VB Service Pack 4. The Common Dialog box is used for opening a html file before the real web site is tested, which is also included in the VB Service Pack 4.

7.2 Programming Problems

This section of problem normally concerned the problems when trying to perform coding on some algorithms. Problems often arise due to lack of programming skills of certain programming languages that were never in practice on our normal working drives. Therefore, experience and practices play very important role in developing programming skills in order to fasten the process of coding and debugging of errors detected.

7.2.1 Implementation of Search Agent

The most difficult part of this system is the implementation of the search agent. Since the search part of the agent was divided into two parts for efficient distribution of tasks between the group members, each of us has to figure out the way to implement the algorithm mentioned earlier. The first part of search agent has been developed and can accomplish the tasks appropriately, while the second part of the search agent, which was implemented by me, has encountered problems while trying to implement the algorithm using Visual Basics.

At the initial stage, the home PC was not properly installed with the Internet Tools that was needed for the coding and testing of the Visual Basics codes that required its built-in Inet class.

7.2.2 Difficulties in Extracting Printer Information

The information organization of each website is different with each other. Therefore it is very difficult to guess the trend of each website. Some printer models are defined in different manner from the standard form, for example: Canon BJC-1000SP is the standard model name, but there are websites that refer to this model as Canon BJC 1000SP. And there are websites that sell printers that are not in the Canon printer list, for example Canon BJC-1000SP, are different versions, like BJC-1100SP. The integrity of this information remained unknown for its correctness.

7.3 Chapter Summary

This chapter describes the major problems encountered during the process of implementing the system. These problems mentioned above were the bottlenecks that have disable the progression of the project. Nevertheless, various solutions have been taken to resolve the problems and to make the system possible.

CHAPTER VIII

SYSTEM EVALUATION & CONCLUSION

CHAPTER VIII : SYSTEM EVALUATION & CONCLUSION

This chapter explains the system strengths, system limitations and its possible enhancements for future use after a detailed evaluation carried out on the system.

8.1 System Strengths

The system strengths normally refer to the system's advantages in accomplishing certain tasks that outstand other similar systems. This surely applied to the system in this project.

The system strengths are identified as the followings:

1. The system sends the requests of printer related websites to search engine, and the results returned are related to what the users normally wants. This helps to facilitate the job of getting valid and appropriate websites when updating the database.
2. The administrator can monitor the system when updating the information in the database. The websites and their HTML contents are displayed so that the administrator can verify the accuracy of the results found. This avoids the high possibilities of getting unrelated information.
3. The administrator of the system can configure the agent so that it performs as intended.

8.2 System Limitations

There are a few limitations that the system will not be able to perform on certain tasks. Various actions and solutions have been taken to omit these limitations, however wise man can make mistakes, so as a system that is developed by mortals like us. Below are the limitations that have been identified:

1. This system is not able to perform an online search because various websites have various organization of information, and it is very difficult to identify the related and necessary information. Users can only search through what the system has found.

2. Some of the functionality of the system, such as the searching of web URL is still has to be searched manually by administrator. It means that the administrator has to click on the menu to start searching since the automated process has not been successfully implemented.
3. The interface of the front-end part of the system is not attractive, neither it is creative and innovative. Since the system is more concentrated on the back end or intelligent part of the system.
4. The Extracting Related Information page still has to be manually updated to monitor the accuracy of information found before they are inserted to the database.
5. The users have to login each time they want to search for printers, edit their profile and to view history search records, even if they have already logged in previously. This may cause the users to be impatient with system.

8.3 Future Enhancements

Time constraints and resource constraints always result in unsatisfactory works done. Therefore, the system can still be enhanced in the future for better performance. Based on the system's limitations as stated above, a few enhancements can be carried out, along with some enhancements that are due to technology change. The following are the possible enhancements:

1. The interface of the system can be further enhanced. Because the system's concentration is on the backend of the system, and the interface is to facilitate the users in carried certain tasks as long as the tasks are done correctly.
2. The system can only carried out database search, and users search through what we have found on the Internet websites. The online search of the system can be done for future enhancement. Due to lack of time for implementing such online search.

8.4 Knowledge Gained

The main purpose of this final year project is to equip the undergraduates with the necessary knowledge and experience to advancing into the next stage of life.

Through the execution of the projects, many have realized that life is a series of hard works, however the rewards of one's effort are invaluable and priceless.

During the two consequence semesters of carrying out this project, I have attained a lot from friends, family members and lecturers. Most importantly is the personalities and self-discipline that are required in any area, any industry, and anywhere in the world.

Furthermore, I have realized the importance of teamwork and time management. Team members have to tolerate with each other, especially when we have a lot projects, assignments and exams that happened at the same time. One has to accept the other's limitations.

Lastly, I learnt the various techniques in Visual Basic programming and also the ASP. Since the project is carried out in teams, exchange of knowledge and experience are necessary, and this can also broaden our viewpoints.

8.5 Chapter Summary

In general, the system is not as desirable as mentioned in the system proposal. Some of the elements such as online search have been omitted due to time constraints. The system can carried out the tasks of searching printers and users management as smoothly as stated in the earlier proposal.

8.6 Conclusion

The system is finally completed successfully although some of the components have been omitted. Future enhancements can be carried out on the system to achieve better performance.

The system is not very effective in finding information for users; only limited websites can be visited for further recommendation of printers. The previous three minor objectives have been achieved, except the detection for changes in URL addresses. The information can be updated automatically by configuring the

timestamp. Besides that users are able to send emails to friends. All the recommendation websites found are Malaysia-based e-commerce website, because queries for printer's links are obtained from local search engines, such as Catcha.com. The information retrieval part is not as effective as it has to be. Manual activities still have to be carried out in order for the system to run smoothly.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Gini, Maria (1999). Agents and Other 'Intelligent Software' for E-Commerce [Online]. Available HTTP: http://www.cs.umn.edu/Research/airvl/ [2000, July 16].

[2] TechTarget.com, Inc. (2000). Intelligent Agent [Online]. Available HTTP: http://www.whatis.com/ia.html [2000, July 17].

[3] TechTarget.com, Inc. (2000). Push (also known as "server-push") [Online]. Available HTTP: http://www.whatis.com/push.htm [2000, July 17].

[4] Bolto, Francis. Dictionary of Multimedia & Internet Applications: A Guide for Developers & Users.

[5] The Emerging Digital Economy. Digital Delivery Of Goods And Services [Online]. Available HTTP: http://www.ecommerce.gov/danc4.htm [2000, July 15].

[6] Seltzer, Richard, Eric J. Ray & Deborah S. Ray. The AltaVista Search Revolution. McGraw-Hill: Osborne, 1997.

[7] Sullivan, Danny (1998). Search Engine Watch: How Search Engine Works [Online]. Available HTTP: http://searchenginewatch.com/. [2000, July 13].

[8] TechTarget.com, Inc. (1999). Database [Online]. Available HTTP: http://www.whatis.com/database.htm [2000, July 17].

[9] TechTarget.com, Inc. (2000). SQL (Structured Query Language) [Online]. Available HTTP: http://www.whatis.com/sql.htm [2000, July 17].

[10] Chen, Hsinchun. (2000). Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms [Online]. Available HTTP: http://ai.bpa.arizona.edu/papers/mlir93/mlir93.html , [2000 July].

[11] Goldberg, D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley: MA, 1989.

[12] Takagi, Hideyuki. Introduction to Fuzzy Systems, Neural Networks, and Genetic Algorithms. IN Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms, edited by Da Ruan. Belgium: Kluwer, 1997, pp. 3-33.

[13] Pfleeger, Shari L. Software Engineering: Theory & Practice. Prentice-Hall: USA, 1997.

[14] TechTarget.com, Inc. (2000). Operating System [Online]. Available HTTP: http://www.whatis.com/WhatIs_Definition_Page/0,4152,212714,00.html, [2000, August].

[15] TechTarget.com, Inc. (2000). Linux [Online]. - Available HTTP: http://www.whatis.com/WhatIs_Definition_Page/0,4152,212482,00.html, [2000, August].

[16] TechTarget.com, Inc. (1999). Windows NT [Online]. Available HTTP: http://www.whatis.com/WhatIs_Definition_Page/0,4152,213368,00.html., [2000, August].

[17] Ziff-Davis Inc. (1997). Microsoft Internet Information Server [Online]. Available HTTP: http://www8.zdnet.com/pcmag/features/webserver/iwsr5.htm [2000, July].

[18] TechTarget.com, Inc. (1999). HTML [Online]. Available HTTP: http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,212286,00.html [2000, August].

[19] TechTarget.com, Inc. (1999). VBScript [Online]. Available HTTP: http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,213279,00.html [2000 August].

[20] TechTarget.com, Inc. (1999). JavaScript. [Online]. Available HTTP: http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,212418,00.html [2000, August].

[21] Microsoft. (2000). Visual Basic General Q & A. [Online]. Available HTTP: http://msdn.microsoft.com/vbasic/prodinfo/qa.asp [2000, August].

[22] Microsoft. (2000). Microsoft FrontPage 98 Online Brochure. [Online]. Available HTTP: http://www.microsoft.com/frontpage/98/98IntPg1.htm [2000, August].

[23] Prolog Development Center. (2000). Introduction to the Prolog Language. [Online]. Available HTTP: http://www.visual-prolog.com/viptechinfo/introduction.htm [2000, August].

[24] The MathWorks. (2000). MATLAB Introduction. [Online]. Available HTTP: http://www.mathworks.com/products/matlab/ [2000, August].

[25] MIT (2000). Introduction to Genetic Algorithms. [Online]. Available HTTP: http://lancet.mit.edu/~mbwall/presentations/IntroToGas/index.html, [2000, August].

[26] University of Sheffield (1996). Evolutionary Computation. [Online]. Available HTTP: http://www.shef.ac.uk/uni/projects/gaipp/gatbx.html, [2000, December].

APPENDICES

APPENDICES

SOURCE CODE

APPENDIX A –

SOURCE CODE

Appendix A – Source Code

Coding using Microsoft InterDev for Web Interface

Util.asp

```
<%
Option Explicit
Const DBLocation = "D:\THESIS\DATABASE\GEN2K.mdb"
%>
<!--#include file="adovbs.inc" -->
<%
''''''''''''''''''''''''''''''''''''''''''''''''''
'
' Application Constants
'
''''''''''''''''''''''''''''''''''''''''''''''''''
Const ACTION_VALIDATE = "v"
Const MODE_ADD = "a"
Const MODE_MODIFY = "m"
Const MODE_POST = "p"
Const MODE_REPLY = "r"
''''''''''''''''''''''''''''''''''''''''''''''''''
'
' Function OpenDB
'
' This function opens the database and returns a valid
' ADO Connection object.
'
''''''''''''''''''''''''''''''''''''''''''''''''''
Function OpenDatabase()
        Dim dcnDB ' As ADODB.Connection
        Set dcnDB = Server.CreateObject("ADODB.Connection")
'***** Access connection code *****
        dcnDB.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" _
                & "Persist Security Info=False;Data Source=" _
                & DBLOCATION & ";"

'***** SQL Server connection code *****
'       dcnDB.ConnectionString = "Provider=SQLOLEDB.1;"_
'               & "Persist Security Info=False;" _
'               & "User ID=sa;Initial catalog=gen;data source=GAEC;"
        dcnDB.Open

    Set OpenDatabase = dcnDB

End Function

Function RetriveData(dcnDB, strSQL)
        Dim rsData
        ' Create recordset and retrieve values using the open connection
        SET rsData = server.CreateObject ("ADODB.Recordset")
        rsData.CursorLocation = adUseClient
        rsData.Open strSQL, dcnDB, adOpenStatic ,adLockReadOnly
        set RetriveData = rsData
End Function

Function WriteData(dcnDB, strSQL)
        Dim rsData
        ' Create recordset and retrieve values using the open connection
        SET rsData = server.CreateObject ("ADODB.Recordset")
        rsData.CursorLocation = adUseClient
        rsData.Open strSQL, dcnDB, adOpenKeyset, adLockPessimistic
        set WriteData = rsData
End Function

''''''''''''''''''''''''''''''''''''''''''''''''''
'
' Sub CloseDB
```

```
'
' This routine closes the open connection.
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Sub CloseDB(dcnDB)

   dcnDB.Close
   set dcnDB = nothing

End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
' Function CreateValidateURL
'
' This routine creates the URL that will cause
' a page to reload in validation mode.
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Function CreateValidateURL()

   Dim strTemp    ' As String
   strTemp = Request.ServerVariables("SCRIPT_NAME") _
      & "?action=" & ACTION_VALIDATE
   CreateValidateURL = LCase(strTemp)

End Function
%>
```

Register.asp

```
<!--#include file="..\util.asp" -->
<html>
<head>
<title>Registration Form </title>
</head>
<body bgcolor="#438db1">
<font face="Verdana, Helvetica, Geneva">

<p align="center"><font face="Brush Script MT" size="+4">Registration Form</font>
<br><br><font size="-1">Items in <font color="red">red </font>are required to process
your registration </font></p>
<%
   Dim dcnDB, rsData
   Dim strSQL

   ' Create and establish data connection
   set dcnDB = OpenDatabase()

%>


<form action="SAVEDATA.asp" method="get" id="form" name="form">
<table align="center" border="0" cellspacing="2">
<tbody>
<tr><td colspan="4" align="center" bgcolor="99FF99"><b>Personal
Information</b></td></tr>
<tr><td align="left"><font color="red"><b>First Name </b></font> </td><td>
  <input type="text" name="fullName" size="20"></td></tr>
<tr><td align="left"><font color="red"><b>Last Name </b></font> </td><td>
  <input type="text" name="lName" size="20"></td></tr>
<tr><td align="left"><font color="red"><b>Age </b></font> </td><td>
  <input type=number name="age" size="3"></td><td><font
color="red"><b>Gender</b></font>
  <input type="radio" value="F" id="rGender" name="rGender"><b>F</b></td><td>
  <input type="radio" value="M" id="rGender" name="rGender"><b>M</b></td>
  </tr>
<tr><td align="left"><font color="red"><b>Date of Birth</b></font></td><td><b>Day</b>
  <select name="day">
   <option value="1">1</option>
   <option value="2">2</option>
   <option value="3">3</option>
   <option value="4">4</option>
   <option value="5">5</option>
```

```
    <option value="6">6</option>
    <option value="7">7</option>
    <option value="8">8</option>
    <option value="9">9</option>
    <option value="10">10</option>
    <option value="11">11</option>
    <option value="12">12</option>
    <option value="13">13</option>
    <option value="14">14</option>
    <option value="15">15</option>
    <option value="16">16</option>
    <option value="17">17</option>
    <option value="18">18</option>
    <option value="19">19</option>
    <option value="20">20</option>
    <option value="21">21</option>
    <option value="22">22</option>
    <option value="23">23</option>
    <option value="24">24</option>
    <option value="25">25</option>
    <option value="26">26</option>
    <option value="27">27</option>
    <option value="28">28</option>
    <option value="29">29</option>
    <option value="30">30</option>
    <option value="31">31</option>
    </select>
   <b>Month</b>
   <select name="month">
    <option value=1>January</option>
    <option value=2>February</option>
    <option value=3>March</option>
    <option value=4>April</option>
    <option value=5>May</option>
    <option value=6>June</option>
    <option value=7>July</option>
    <option value=8>August</option>
    <option value=9>September</option>
    <option value=10>October</option>
    <option value=11>November</option>
    <option value=12>December</option>
    </select><b>Year </b>
   <input type=number name="year" size="4"></td></tr>
<tr><td align="left"><b>Country </b> </td><td>
   <%
   strsql = "select xcode,xdesc from code where XcodeType ='Country';"
   set rsData = RetriveData(dcnDB, StrSQL)
   If Not rsData.EOF Then
        rsData.MoveFirst
   %>
<select NAME="country">
   <%
Do While Not rsData.EOF
   %>
     <option VALUE="<%= rsData.Fields("xcode") %>">
   <%= rsData.Fields("xdesc")%></option>
   <%
        rsData.MoveNext
        Loop
   %>
</select>
   <%
   End If
   rsData.Close
   %>
</td></tr>

<tr><td align="left"><b>Occupation </b> </td><td>
   <%
   strsql = "select xcode,xdesc from code where XcodeType ='Occupation';"
   set rsData = RetriveData(dcnDB, StrSQL)
   If Not rsData.EOF Then
        rsData.MoveFirst
   %>
<select NAME="occupation">
   <%
Do While Not rsData.EOF
```

```
  %>
    <option VALUE="<%= rsData.Fields("xcode") %>">
  <%= rsData.Fields("xdesc")%></option>
  <%
       rsData.MoveNext
       Loop
  %>
</select>
  <%
  End If
  rsData.Close
  %>
</td></tr>

<tr><td align="left"><font color="red"><b>Login Name </b></font> </td><td>
  <input type="text" name="login" size="20"></td></tr>
<tr><td align="left"><font color="red"><b>Password </b></font> </td><td>
  <input type="PASSWORD" name="password" size="15"></td></tr>
<tr><td align="left"><font color="red"><b>Please type again
</b></font> </td><td>
  <input type="PASSWORD" name="password_verify" size="15"></td></tr>

<tr><td align="left"><font color=red><b>E-mail </b></font> </td><td>
  <input type="text" name="email" size="30"></td></tr><br>

<tr><td colspan="4" align="left"><b>Please fill in a hint question and its answer
</b>
    </td></tr>
<tr><td align="left"><b>Question </b> </td><td>
  <input type="text" name="secretq" size="50"></td></tr>
<tr><td align="left"><b>Answer </b> </td><td>
  <input type="text" name="secreta" size="20"></td></tr>
<tr><td colspan="2"><b>Do you want to receive notices on data found ?</b></td></tr>
  <tr><td><input type="radio" value=TRUE id="rNotice" name="rNotice"><b>Yes</b></td>
  <td><input type="radio" value=FALSE id="rNotice" name="rNotice"><b>No</b></td>
</tbody></table>

  <p align="center">
  <input type="submit" value="Register" name="B1" style="LEFT: 100px; TOP:
15px">  </a>    
  <input type="reset" value="Clear" name="B2"></p></form>

<p align="center"><font size="2">
Read the <a href="./Terms.htm"> Terms of Service</a> of this Website 
</font></p>
</font>
</body>
</html>
```

SaveData.asp

```
<!--#include file="..\util.asp" -->

 <%
 Dim dcnDB1, dcnDB2, rsData1, rsData2
   Dim SQLText1
   Dim SQLText2
   set dcnDB1 = OpenDatabase()
   Dim varLogin
   varLogin = Request.QueryString("Login")

   If Request.QueryString("password") <> Request.QueryString("password_verify") Then
      Response.Redirect "PassErr.htm"
   End If


   ' Store data into the User_Information Table

      SQLText1 ="Select * from User_Information where LoginName = '" & varLogin & "'"

   set rsData1 = WriteData (dcnDB1, SQLText1)

   If rsData1.EOF  then
      rsData1.AddNew
      rsData1.Fields ("LoginName") = varLogin & ""
```

```
      rsData1.Fields ("FirstName") = Request.QueryString("fullname")& ""
      rsData1.fields ("LastName") = Request.QueryString("lname")& ""
      rsData1.fields ("Gender") = Request.QueryString("rgender")& ""
      rsData1.fields ("Age") = Request.QueryString ("age")& ""
      rsData1.fields ("DOB") = Request.QueryString("month") & "/" &
Request.QueryString("day") & "/" & Request.QueryString("year")
      rsData1.fields ("Country") = Request.QueryString("country")& ""
      rsData1.fields ("Occupation") = Request.QueryString("occupation")& ""
      rsData1.fields ("Email") = Request.QueryString("email")& ""
      rsData1.fields ("SendNotice") = Request.QueryString("rNotice")& ""
      rsData1.fields ("XCrDate") = Date
      rsData1.fields ("XUpDate") = Date
        rsData1.Update

   Else
      Response.Redirect "RegErr.htm"

end if
CloseDB (rsData1)
CloseDB (dcnDB1)

%>

<%
   set dcnDB2 = OpenDatabase()

   ' Store data into the User_Information Table

      SQLText2 ="Select * from User_Security"

   set rsData2 = WriteData (dcnDB2, SQLText2)

   if not rsData2.EOF  then
         rsData2.MoveFirst

      rsData2.AddNew
      rsData2.Fields ("LoginName") = Request.QueryString("login")
      rsData2.Fields ("Password") = Request.QueryString("password")
      rsData2.fields ("SecretQ") = Request.QueryString("secretq")
      rsData2.fields ("SecretA") = Request.QueryString("secreta")
      rsData2.Update
  end if
CloseDB (rsData2)
CloseDB (dcnDB2)

%>

<%
   ' if the above statements are successful
   ' Moveto saving datainto next table

 Response.Redirect "MsgSaveData.htm"
%>
```

CheckLogin.asp

```
<!--#include file="util.asp" -->

<%
'this is the response page to login & password

Dim dcnDB, rsData
set dcnDB = OpenDatabase()

dim varPassword
dim varLoginName
dim sqlText
varLoginName = Request.Form("login")
varPassword = Request.Form("password")

sqlText="Select * from User_Security "
sqlText=sqlText & "WHERE LoginName = '" & varLoginName & "'"
sqlText=sqlText & "AND Password = '" & varPassword & "';"
set rsData = RetriveData(dcnDB, sqlText)

if  varLoginName = "Guest" and varPassword = "Guest" then
```

```
        Response.Redirect "SearchPrn.asp"
'               Response.Write "You have not successfully logged into the system!
Please check your Login ID!"
'else if varLoginName <> "Guest"

Elseif not rsData.EOF  Then
    Response.Redirect "../login/MsgLogin.htm"


Else
        Response.Write "<TABLE width='100%' border=0><TR><TD width='18'></TD><TD
width='82%'><FONT color=#ff0000 size=3><b>You have failed to login. Please try
again!!<b></font></td></tr></table>"
%>
<!-- #include file="default.htm" -->
<%
'if trim(Request.Form("login"))="" or trim(Request.Form("password"))="" Then
'       Response.Redirect "LoginErr.asp"
'Else
End if

rsData.close
set rsData= nothing

%>
```

EditData.asp

```
<!--#include file="..\util.asp" -->

<html>
<head>
<title>Personal Profile Update </title>
</head>
<body bgcolor="#438db1">
<font face="Verdana, Helvetica, Geneva">

<%
   Dim dcnDB, rsData1, rsData2
   Dim strSQL1, strSQL2
   Dim varLogin
   Dim varPassword

   ' Create and establish data connection
   set dcnDB = OpenDatabase()
   varLogin = Request.QueryString("login")
   varPassword = Request.QueryString("password")

   strSQL1="Select * from User_Security WHERE LoginName = '" & varLogin & "' AND
Password = '" & varPassword & "'"
   set rsData1 = RetriveData (dcnDB, strSQL1)

' Verify LoginName and Password
If  not rsData1.EOF Then

'   strSQL2="Select * from User_Information WHERE LoginName = 'varLogin'"
   strSQL2="Select * from User_Information WHERE LoginName = '" & varLogin & "'"
   set rsData2 = RetriveData (dcnDB, strSQL2)

   Response.Write "<P align=center><font face=Brush Script MT size=+4>Personal
Profile Update</font><br><br>"
   Response.Write "<font size=-1>Please Click on the <font
color=blue><u>Field</u></font> to update your profile</font></P>"
   %>
<table align="center" border="0" cellspacing="2">
<tbody>
<tr><td colspan="4" align="middle" bgcolor="#99ff99"><b>Personal
Information</b></td></tr><tr></tr>

<tr><td align="left" width="30%"><b>First Name </b></font> </td>
    <td width="60%">
    <%
                    Response.Write "<a href=""" _
                        & "SaveEdit.asp?" _
                        & "column=1&login=" & varLogin & """
onMouseOver=""window.status='Change " _
```

```
                                & rsdata2.Fields("FirstName") & " ';return true""" _
                                & "onMouseOut=""window.status='';return true""" _
                                & "target=""_blank"" >" _
                                & rsData2.Fields("FirstName") & "</a>"

        %></td></tr><tr></tr>

<tr><td align="left"><b>Last Name </b></font> </td>
    <td><%
            Response.Write "<a href=""" _
                                & "SaveEdit.asp?" _
                                & "column=2&login=" & varLogin & """
onMouseOver=""window.status='Change " _
                                & rsdata2.Fields("LastName") & " ';return true""" _
                                & "onMouseOut=""window.status='';return true""" _
                                & "target=""_blank"" >" _
                                & rsData2.Fields("LastName") & "</a>"

        %></td></tr><tr></tr>

<tr><td align="left"><b>Country </b></font> </td>
    <td>
        <%
          strSQL1 = "Select * from Code WHERE XCode = '" & rsData2.Fields ("country")
& "'"
          set rsData1= RetriveData (dcnDB, strSQL1)
        %>
    <%
            Response.Write "<a href=""" _
                                & "SaveEdit.asp?" _
                                & "column=3&login=" & varLogin & """
onMouseOver=""window.status='Change " _
                                & rsdata1.Fields("XDesc") & " ';return true""" _
                                & "onMouseOut=""window.status='';return true""" _
                                & "target=""_blank"" >" _
                                & rsData1.Fields("XDesc") & "</a>"

        %></td></tr><tr></tr>

<tr><td align="left"><b>Occupation </b></font> </td>
    <td>
        <%
          strSQL1 = "Select * from Code WHERE XCode = '" & rsData2.Fields
("occupation") & "'"
          set rsData1= RetriveData (dcnDB, strSQL1)
        %>
     <%
            Response.Write "<a href=""" _
                                & "SaveEdit.asp?" _
                                & "column=4&login=" & varLogin & """
onMouseOver=""window.status='Change " _
                                & rsdata1.Fields("XDesc") & " ';return true""" _
                                & "onMouseOut=""window.status='';return true""" _
                                & "target=""_blank"" >" _
                                & rsData1.Fields("XDesc") & "</a>"

        %></td></tr><tr></tr>

<tr><td align="left"><b>Email </b></font> </td>
    <td><%
            Response.Write "<a href=""" _
                                & "SaveEdit.asp?" _
                                & "column=5&login=" & varLogin & """
onMouseOver=""window.status='Change " _
                                & rsdata2.Fields("email") & " ';return true""" _
                                & "onMouseOut=""window.status='';return true""" _
                                & "target=""_blank"" >" _
                                & rsData2.Fields("email") & "</a>"

        %></td></tr><tr></tr>

<tr><td align="left"><b>Send Notice ?</b></font> </td>
    <td><%
            Response.Write "<a href=""" _
                                & "SaveEdit.asp?" _
                                & "column=6&login=" & varLogin & """
onMouseOver=""window.status='Change " _
```

```
                                & rsdata2.Fields("SendNotice") & " ';return true""" _
                                & "onMouseOut=""window.status='';return true""" _
                                & "target=""_blank"" >" _
                                & rsData2.Fields("SendNotice") & "</a>"

        %></td></tr><tr></tr>

<tr><td align="left"><b>Change Password?</b></font> </td>
    <td><%
            Response.Write "<a href=""" _
                                & "SaveEdit.asp?" _
                                & "column=7&login=" & varLogin & """
onMouseOver=""window.status='Change " _
                                & "Password';return true""" _
                                & "onMouseOut=""window.status='';return true""" _
                                & "target=""_blank"" >" _
                                & "Password "& "</a>"
        %></td></tr><tr></tr>

<tr><td align="left" width="30%"><b>Hints </b></font> </td>
    <td width="60%">
    <%
                        Response.Write "<a href=""" _
                                & "SaveEdit.asp?" _
                                & "column=8&login=" & varLogin & """
onMouseOver=""window.status='Change " _
                                & "Hint Question and Hint Answer';return true""" _
                                & "onMouseOut=""window.status='';return true""" _
                                & "target=""_blank"" >" _
                                & "Hint Question and Hint Answer " & "</a>"

       %></td></tr><tr></tr></tbody></table>

<p align="center"><font size="2">
Read the <a href="../login/Terms.htm"> Terms of Service</a> of this Website 
</font></p>
<p align="center"><font size="2"><a href="../default.htm"><img
src="../images/BUT1_IDL.GIF" WIDTH="68" HEIGHT="67"></a>
<br>HOME</font></p>
<%
 CloseDB (rsData1)
 CloseDB (rsData2)
 CloseDB (dcnDB)

Else
Response.Write "<TABLE width='100%' border=0><TR><TD width='18'></TD><TD
width='82%'><FONT color=#ff0000 size=3><b>You have failed to login. Please try
again!!<b></font></td></tr></table>"
 %>
 <!-- #include file="ChkLoginID.htm" -->
 <%


End if

%>

</font>

</body>
</html>
```

SaveEdit.asp

```
<!--#include file="..\util.asp" -->
<HTML>
<HEAD>
<TITLE>Saving your data into database</TITLE>
</HEAD>
<BODY bgcolor="#438db1">

<P>
<br><br><br>

<%
 Dim dcnDB, rsData, rsData1
```

```
Dim StrSQL
set dcnDB = OpenDatabase()
Dim strtemp
Dim varLogin
varLogin = Request("login")
Dim varEdit
varEdit = Request("column")

StrSQL = "Select * from User_Information WHERE LoginName =  '" & varLogin & "'"
set rsData = RetriveData (dcnDB, StrSQL)

select case varEdit
            case 1:
                        Response.Write "<h3>First Name</h3>"
                        Response.Write rsData.Fields ("FirstName")
                        Response.Write "<h3>New First Name</h3>"

      %>
      <form action="Update.asp" method="get" id="form" name="form">
      <input type=text size=20 name=fname>
      <input type=hidden value="<%= rsData.Fields("LoginName") %>" name=login>
      <input type=hidden name=column value="<%= Request ("column") %>">
      <input type="submit" value="Save" name="B1">
      </form>
            <%
      CloseDB (rsData)
    case 2:
                        Response.Write "<h3>Last Name</h3>"
                        Response.Write rsData.Fields ("LastName")
                        Response.Write "<h3>New Last Name</h3>"

      %>
      <form action="Update.asp" method="get" id="form" name="form">
      <input type=text size=20 name=lname>
      <input type=hidden value="<%= rsData.Fields("LoginName") %>" name=login>
      <input type=hidden name=column value="<%= Request ("column") %>">
      <input type="submit" value="Save" name="B1">
      </form>
      <%
      CloseDB (rsData)
    case 3:
                        Response.Write "<h3>Country</h3>"
                        strSQL = "select * from Code WHERE XCodeType = 'Country' "
                        set rsData1 = RetriveData (dcnDB, strSQL)
                        Response.Write rsData1.Fields("XDesc")
                        Response.Write "<h3>New Country</h3>"
            If Not rsData1.EOF Then
                  rsData1.MoveFirst

            %>
            <form action="Update.asp" method="get" id="form" name="form">
            <select NAME="country">
            <%
            Do While Not rsData1.EOF
            %>
              <option VALUE="<%= rsData1.Fields("xcode") %>">
              <%= rsData1.Fields("xdesc")%></option>
              <%
                        rsData1.MoveNext
               Loop
            %>
        </select>
      <%
        End IF

      %>

      <input type=hidden value="<%= rsData.Fields("LoginName") %>" name=login>
      <input type=hidden name=column value="<%= Request ("column") %>">
      <input type="submit" value="Save" name="B1">
      </form>

      <%
      CloseDB (rsData1)
      CloseDB (rsData)

    case 4:
```

```
                        Response.Write "<h3>Occupation</h3>"
                        strSQL = "select * from Code WHERE XCodeType = 'Occupation'
"
                        set rsData1 = RetriveData (dcnDB, strSQL)
                        Response.Write rsData1.Fields("XDesc")
                        Response.Write "<h3>New Occupation</h3>"
                If Not rsData1.EOF Then
                     rsData1.MoveFirst

                %>
                <form action="Update.asp" method="get" id="form" name="form">
                <select NAME="Occupation">
                <%
                Do While Not rsData1.EOF
                %>
                  <option VALUE="<%= rsData1.Fields("xcode") %>">
                  <%= rsData1.Fields("xdesc")%></option>
                  <%
                              rsData1.MoveNext
                    Loop
                %>
              </select>
            <%
              End IF
            %>

            <input type=hidden value="<%= rsData.Fields("LoginName") %>" name=login>
            <input type=hidden name=column value="<%= Request ("column") %>">
            <input type="submit" value="Save" name="B1">
            </form>

            <%
            CloseDB (rsData1)
            CloseDB (rsData)

                case 5:
                        Response.Write "<h3>Email Address</h3>"
                        Response.Write rsData.Fields("Email")
                        Response.Write "<h3>New Email Address</h3>"
            %>
            <form action="Update.asp" method="get" id="form" name="form">
            <input type=text size=20 name=email>
            <input type=hidden value="<%= rsData.Fields("LoginName") %>" name=login>
            <input type=hidden name=column value="<%= Request ("column") %>">
            <input type="submit" value="Save" name="B1">
            </form>
            <%
            CloseDB (rsData)

        case 6:
                        Response.Write "<h3>Do you want to receive notice via Your
Email?</h3><br>"
            %>
            <form action="Update.asp" method="get" id="form" name="form">
            <input type="radio" value=TRUE id="rNotice" name="rNotice"><b>Yes</b>
            <input type="radio" value=FALSE id="rNotice" name="rNotice"><b>No</b>
            <input type=hidden value="<%= rsData.Fields("LoginName") %>" name=login>
            <input type=hidden name=column value="<%= Request ("column") %>">
            <input type="submit" value="Save" name="B1">
            </form>

            <%
            CloseDB (rsData)

                case 7:

            %>
            <form action="Update.asp" method="get" id="form" name="form">
            <table align="center" border="0" cellspacing="2">
               <tbody>
            <tr><td align="left"><b>New Password </b> </td><td>
                <input type="PASSWORD" name="newpassword" size="15"></td></tr>
            <tr><td align="left"><b>Please Confirm Again </b> </td><td>
                <input type="PASSWORD" name="password_verify" size="15"></td></tr>
            </tbody></table>
            <input type=hidden value="<%= rsData.Fields("LoginName") %>" name=login>
            <input type=hidden name=column value="<%= Request ("column") %>">
```

```
        <p align="center"><input type="submit" value="Save" name="B1"></P>
        </form>

        <%
        CloseDB (rsData)

      case 8:
            StrSQL = "Select * from User_Security WHERE LoginName =  '" & varLogin
& "'"
            set rsData = RetriveData (dcnDB, StrSQL)

                  Response.Write "<h3>Old Hint Question</h3>"
                  Response.Write rsData.Fields ("secretq")
                  Response.Write "<h3>Old Hint Answer</h3>"
                  Response.Write rsData.Fields ("secreta")
               Response.Write "<h3>New Hint Question</h3>"

        %>


        <form action="Update.asp" method="get" id="form" name="form">
        <input type="text" name="secretq" size="50"></td></tr>
        <%
        Response.Write "<h3>New Hint Answer</h3>"
        %>

        <input type="text" name="secreta" size="20"></td></tr>
        <input type=hidden value="<%= rsData.Fields("LoginName") %>" name=login>
        <input type=hidden name=column value="<%= Request ("column") %>">
        <input type="submit" value="Save" name="B1">
        </form>

        <%
        CloseDB (rsData)

  end select
   %>

 <%
   CloseDB (dcnDB)
  %>
</BODY>
</HTML>
```

Update.asp

```
<!--#include file="..\util.asp" -->

<%

Dim dcnDB, rsData
Dim strSQL
set dcnDB = OpenDatabase ()
Dim varLogin
varLogin = Request.QueryString("login")
Dim varEdit
varEdit = Request.QueryString("column")


select case varEdit
            case 1:
                  strSQL = "Update User_Information "_
                  & "SET FirstName = '" & Request.QueryString("fname") & "' "_
                  & "WHERE LoginName ='" & varLogin & "'"
               dcnDB.Execute(strSQL)

      case 2:
                  strSQL = "Update User_Information "_
                  & "SET LastName = '" & Request.QueryString("lname") & "' "_
                  & "WHERE LoginName ='" & varLogin & "'"
               dcnDB.Execute(strSQL)

      case 3:
                  strSQL = "Update User_Information "_
                  & "SET Country = '" & Request.QueryString("country") & "' "_
                  & "WHERE LoginName ='" & varLogin & "'"
```

```
                dcnDB.Execute(strSQL)

        case 4:
                    strSQL = "Update User_Information "_
                    & "SET Occupation = '" & Request.QueryString("occupation") & "'
"_
                    & "WHERE LoginName ='" & varLogin & "'"
                dcnDB.Execute(strSQL)

        case 5:
                    strSQL = "Update User_Information "_
                    & "SET Email = '" & Request.QueryString("email") & "' "_
                    & "WHERE LoginName ='" & varLogin & "'"
                dcnDB.Execute(strSQL)

        case 6:
                    strSQL = "Select * from User_Information WHERE LoginName ='" &
varLogin & "'"
                    Set rsData = WriteData (dcnDB, strSQL)
                    if not rsData.EOF Then
                        rsData.Fields ("SendNotice") =
Request.QueryString("rNotice")
                        rsData.Update
                    End IF
                    CloseDB (rsData)

        case 7:
                    strSQL = "Select * from User_Security WHERE LoginName ='" &
varLogin & "'"
                    Set rsData = WriteData (dcnDB, strSQL)
                    if not rsData.EOF Then
                       If not Request.QueryString("newpassword") <>
Request.QueryString("password_verify") Then
                        rsData.Fields ("Password") =
Request.QueryString("newpassword")
                        rsData.Update
                      Else
                        Response.Write "<TABLE width='100%' border=0><TR><TD
width='18'></TD><TD width='82%'><FONT color=yellow size=4><b>The new password does
not match the verified password<br>Please re-login</b></font></td></tr></table>"
                        %>
                        <!-- #include file="ChkLoginID.htm" -->
                        <%
                      End IF
                    End If
            CloseDB (rsData)

             case 8:
                    strSQL = "Update User_Security "_
                    & "SET SecretQ = '" & Request.QueryString("secretq") & "', "_
                    & "SecretA = '" & Request.QueryString("secreta") & "' "_
                    & "WHERE LoginName ='" & varLogin & "'"
                dcnDB.Execute(strSQL)


end select

CloseDB (dcnDB)
%>

<%
 Response.Redirect "MsgUpdate.asp"
%>
```

MsgUpdate.asp

```
<HTML>
<title>Your Profile Has Been Updated</title>
<HEAD>
</HEAD>
<BODY bgcolor="#438db1">
<%
        Response.Write  "<center><TABLE width='100%' border=0><TR><TD
width='18'></TD><TD width='82%'><FONT color=yellow size=4><b>Your Profile has been
updated...</b></font></td></tr></table>"
```

```
                Response.write "<form><p>"
                Response.write "<input onclick='window.close()' type='button' "
                Response.write "value=""Close Window""></p></form></center>"

%>
<P> </P>

</BODY>
</HTML>
```

Coding using Microsoft Visual Basic SP4 for Intelligence Applications

```
Public Function RemoveSpace(inputStr As String) As String
Dim p As Long
Dim str As String
Dim i As Integer
   p = 1
   p = InStr(p + 1, inputStr, " ")
   Do While p <> 0
      If Mid(inputStr, p, 1) = " " And Mid(inputStr, p + 1, 1) = " " Then
         inputStr = Mid(inputStr, 1, p - 1) + Mid(inputStr, p + 1)
         p = p - 1
      End If
      p = InStr(p + 1, inputStr, " ")
   Loop

   RemoveSpace = inputStr
End Function

Public Function RetriveHTML(ByVal inputUrl As String) As String

On Error GoTo error_handler:
Dim mstrHTML As String

   mstrHTML = Inet1.OpenURL(inputUrl)
   Inet1.Protocol = icHTTP
   Inet1.Execute CStr(inputUrl), "GET / "

   While Inet1.StillExecuting
      DoEvents
   Wend

   RetriveHTML = mstrHTML

   Exit Function

error_handler:

   MsgBox Err.Number & vbCrLf & Err.Description, vbExclamation, "Error Handle"
   Resume Next

End Function

frmScan.frm

Option Explicit
Dim i As Integer


Private Sub cmdClose_Click()
   Unload Me
End Sub

Private Sub cmdOpenAgain_Click()
   Screen.MousePointer = vbHourglass
   StatusBar1.Panels.Item(1).Text = "Retrieving Information...."

      txtInfo.Text = FindData(txtHTML.Text)
```

APPENDIX B –

USER MANUAL

Content

List of Figures

System Requirements for User's PC

This user manual is for users who access to the Main Page of the Web Agent to search for printers. In order to access to the Website, the user should have the following web browsers installed in PC:

1. Internet Explorer 4.01 and above

Please note that neither Netscape Navigator nor the Netscape Communicator can access this web site due to the lack of some display features.

If the user does not have the required web browser, there are several websites that provide freeware for the convenience of the users, such as http://www.download.com and http://www.freeware.com.

We surely hope that this user manual would be helpful to all the end-users of the system through the Internet. Don't hesitate to contact the Webmaster of the Agent to learn more on the system. We hope that you enjoy browsing through our website.

Step 1: Access to the Main Page of the Web Site

1. To access to the Main Page of the Agent, make sure that you are connected to the Internet via a modem or connected to the LAN.
2. At the Address Bar, type http://202.185.108.36, and you will arrive to the Main Page of the Agent.

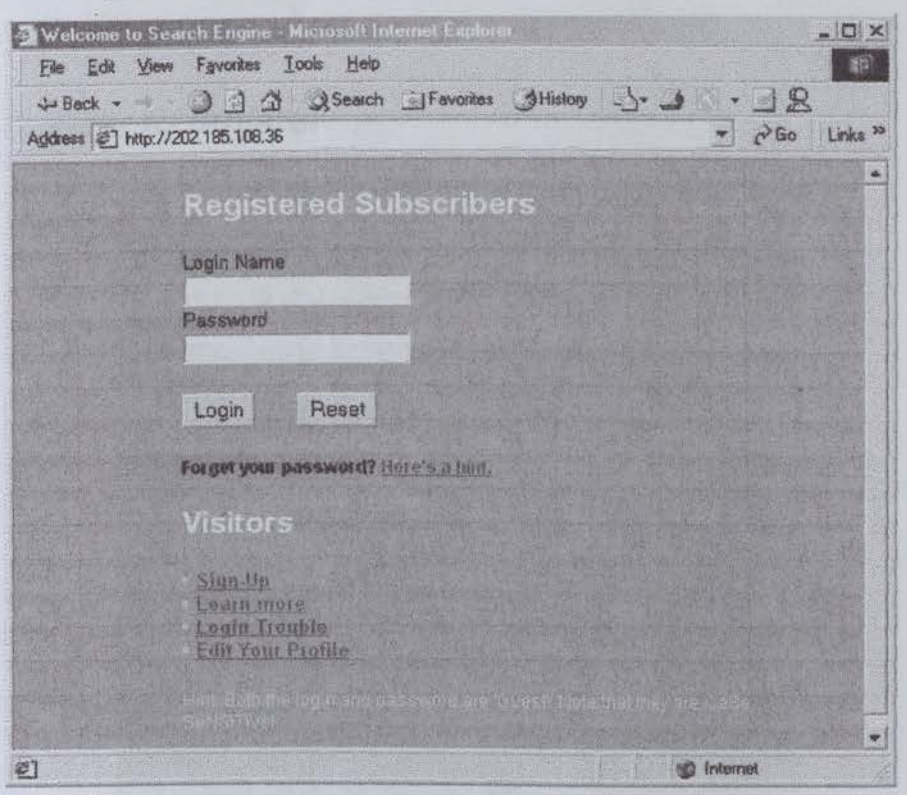


Figure 1-1: Access the Main Page of the website

Step 2: User Registration

When you arrive at the initial page of the agent, you can choose to register yourself or to login as Guest. The difference between a registered user and a guest, is the users can keep track of his history search records, and send recommendation to friends via email provided by the agent. A guest can only search through information that the agent has gathered on printers.

2.1 How to register

1. Click on the Sign-Up link at the main page of the agent

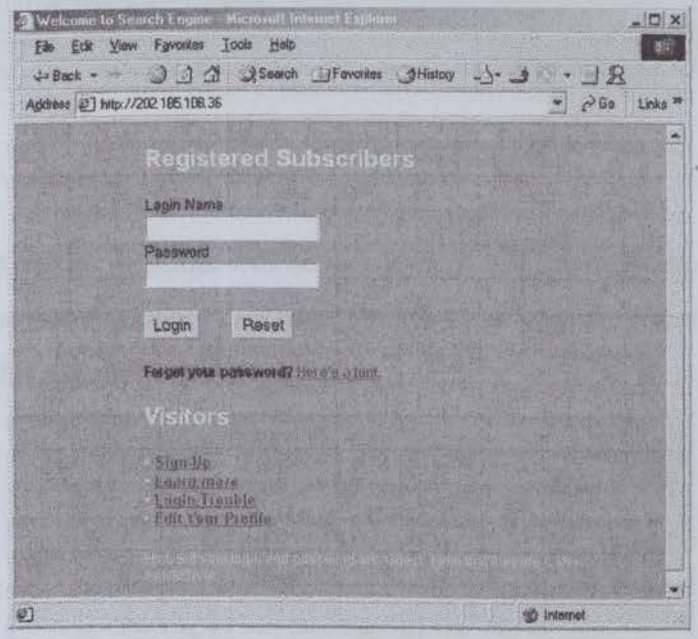


Figure 2-1: The main page of the agent

2. Then you will see a registration form, just fill in the registration form appropriately. Remember that the items in red are required for your registration.

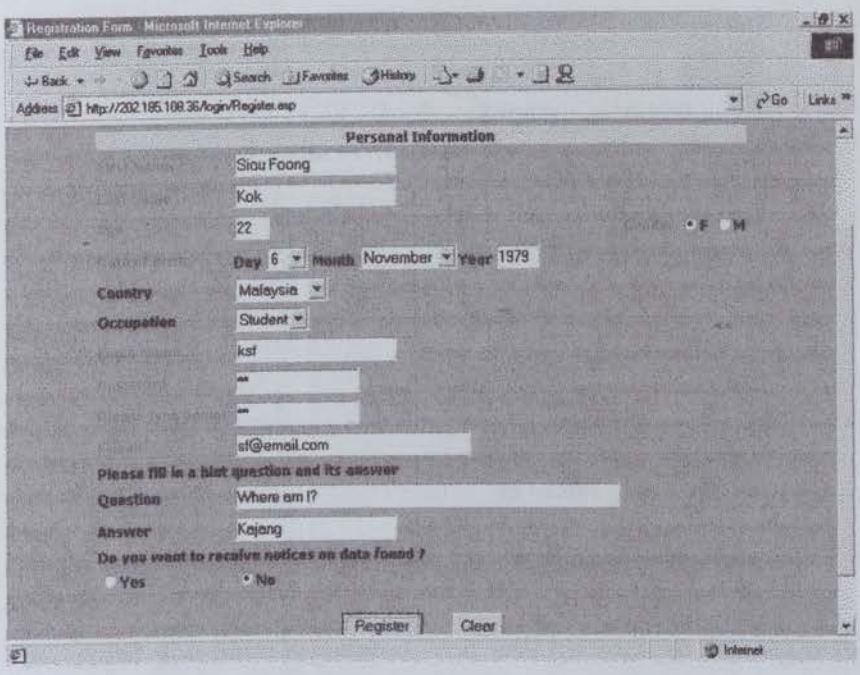


Figure 2-2: User Registration Form

3. An error message will be displayed if the Login Name has been taken by another user. You have click on the Register button to go back to the registration form again.

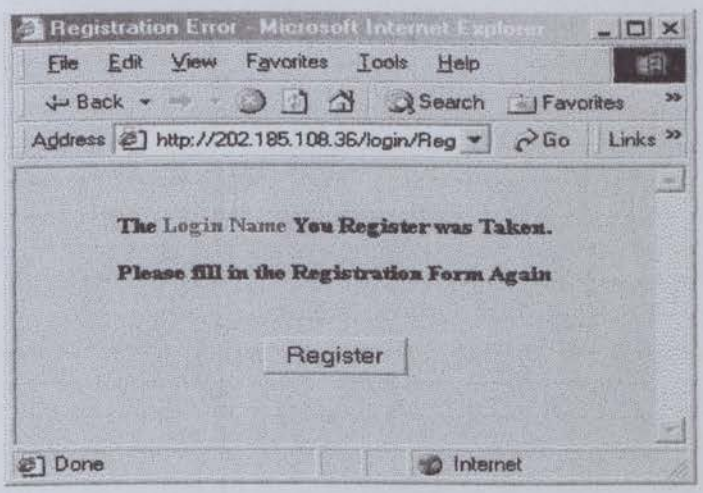


Figure 2-3: Error Message on Invalid Login Name Registered

4. An error message will be displayed if the initial Password inserted is different with the confirmed password typed. You have click on the Register button to go back to the registration form again.

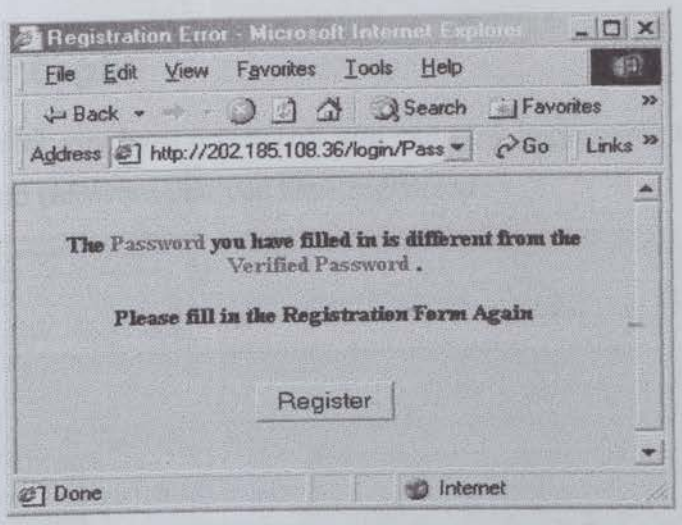


Figure 2-4: Error Message on Invalid Passwords inserted

5. After you have registered, you will go back to the Main Page of the Agent again. Now you can Login, edit your profile to search for printers.

2.2 User Login

1. After you have registered, you can now login using the Login Name and password that you have registered.

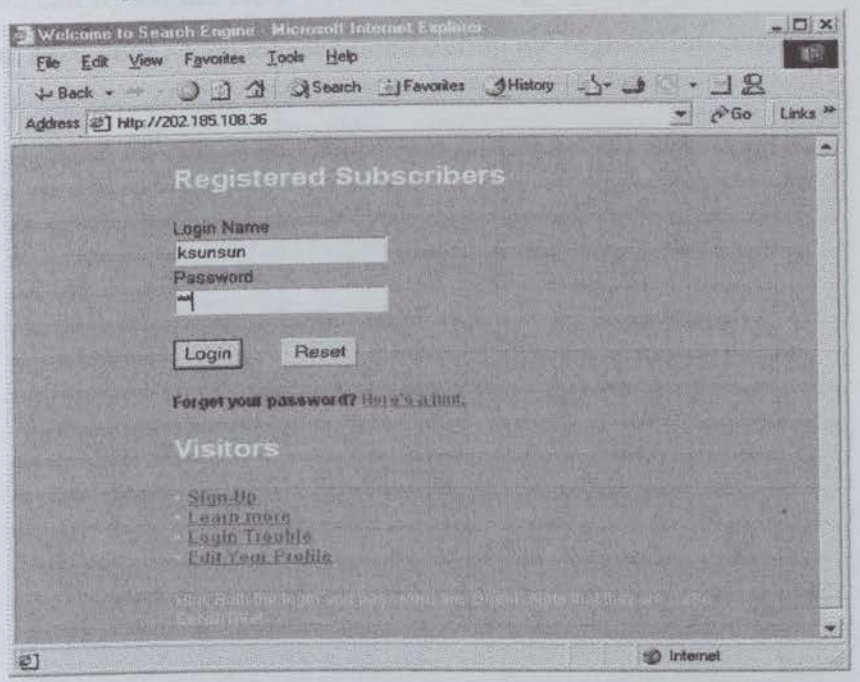


Figure 2-5: User Login

2. Then, a menu will be pop up. You can choose any of the options.

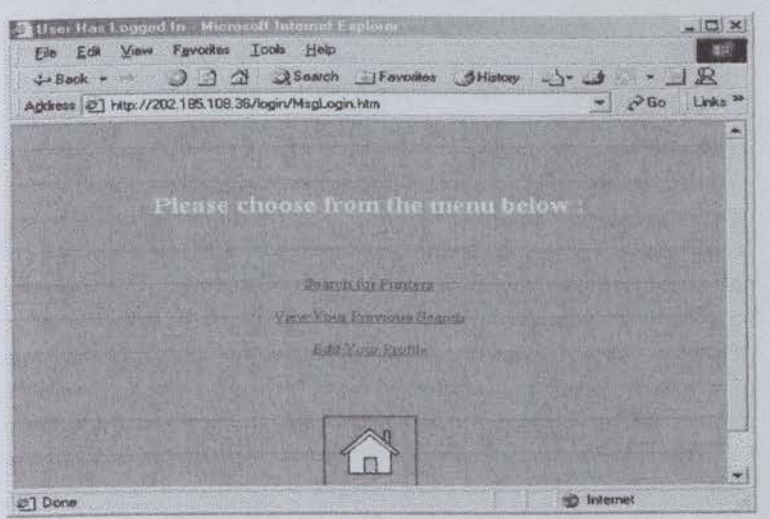


Figure 1-6: Menu after logging in

3. Please refer to the following to proceed:
   a. Refer to Step 3 if you choose to Search For Printers.
   b. Refer to Section 2.2 of Step 2 to Update Your Profile

c. Refer to Section 2.3 of Step 2 to View your Previous History Search Records.

Trouble logging in

1. You can try on the Here's A Hint link to get your password. Make sure that you have your secret answer filled during the registration process.

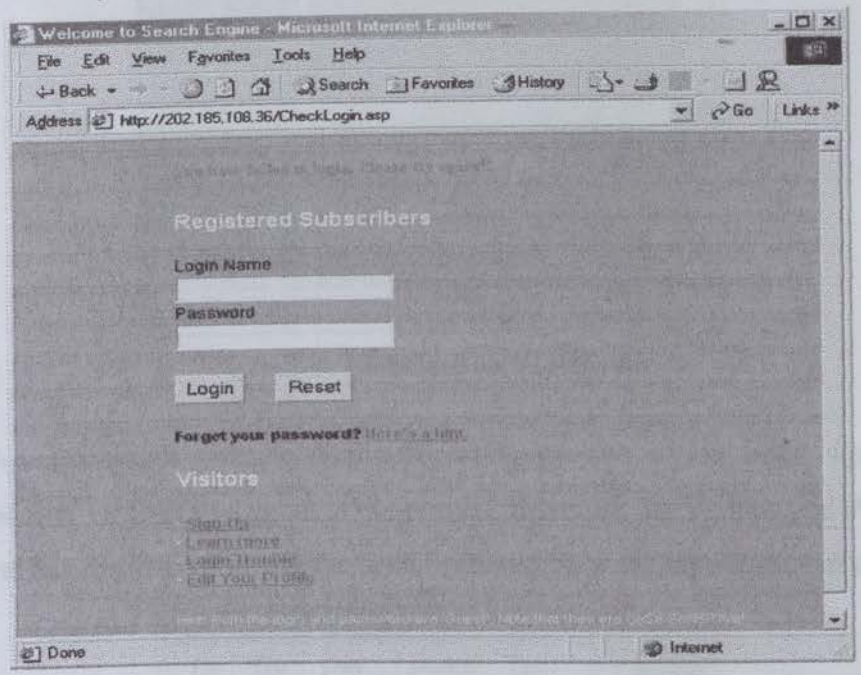


Figure 2-7: Trouble logging in

2. Please fill in your login name to retrieve your secret question

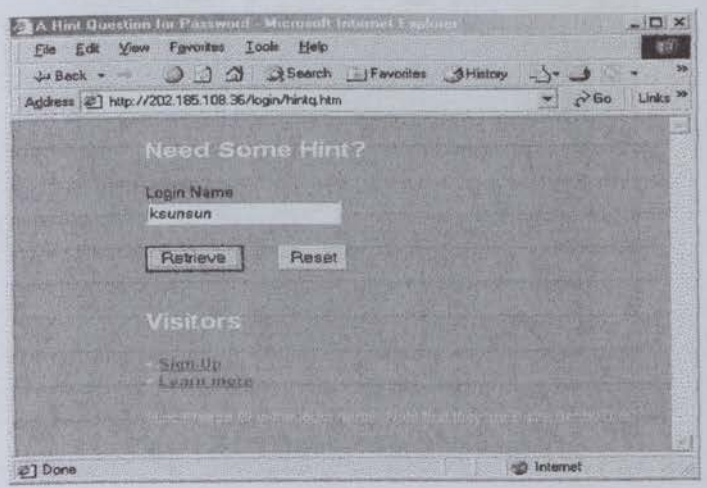


Figure 2-8: Login to get secret question

3. Please answer your hint question correctly.

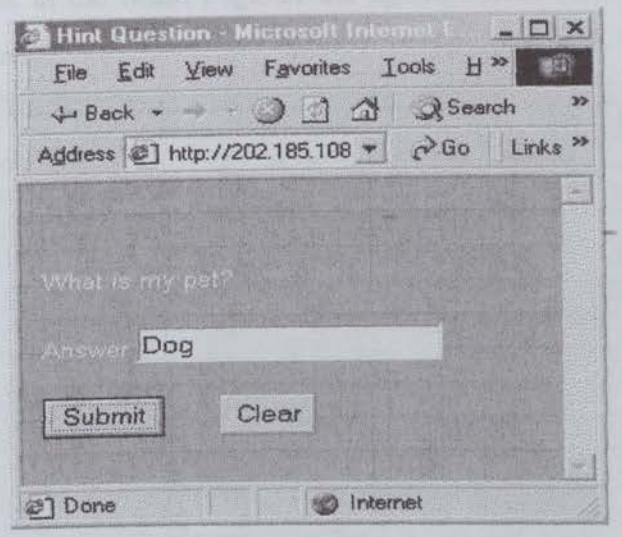


Figure 2-9: Hint Question and Answer

4. Your password will be displayed, and you can try to login again. (The security issue in this agent is not crucial, therefore the Webmaster of this web site has decided to reveal the users' password for the convenient of users).

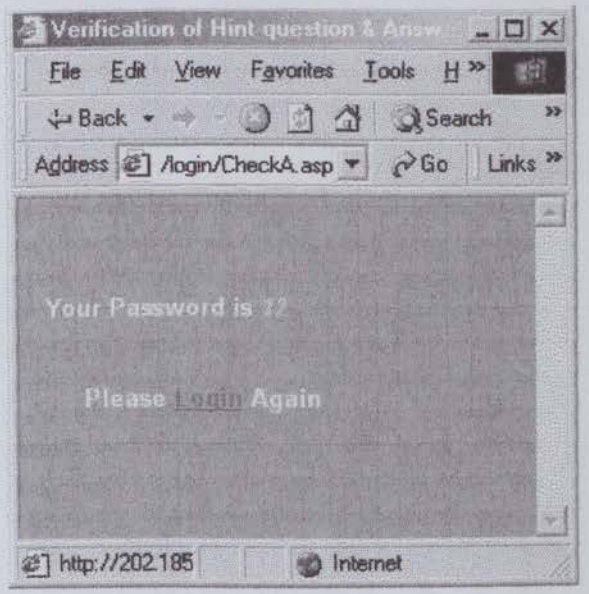


Figure 2-10: Password will be displayed

5. If you do not fill this option during registration, you can try on the Login Trouble link at the Main Page of the agent. Please follow the instructions provided here.

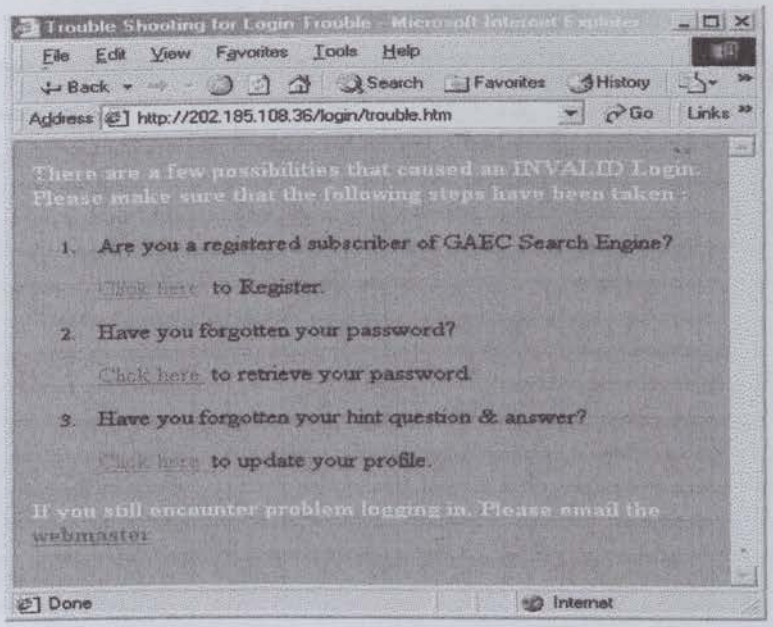


Figure 2-11: Login Trouble Instructions

2.3 Update User Profile

1. Just click on Edit Your Profile link at the Main Page of the Agent, please login again even if you have logged in previously.

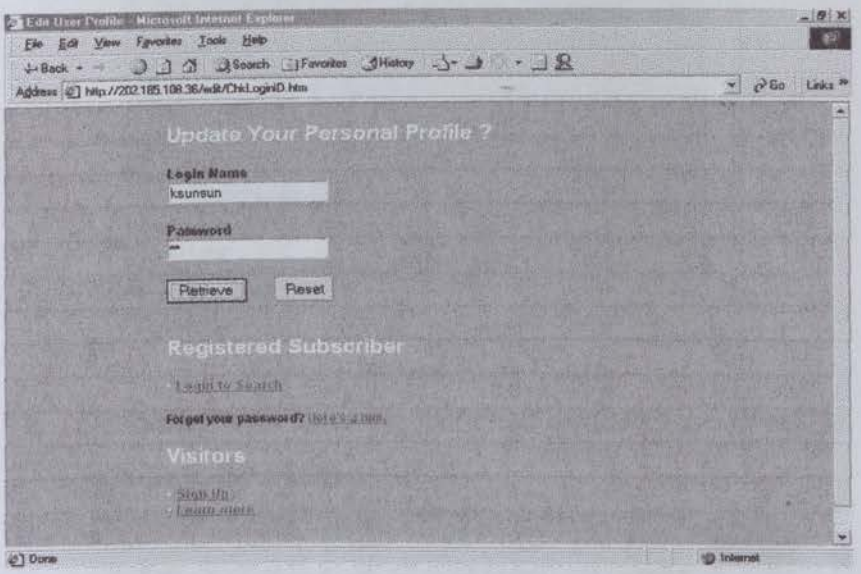


Figure 2-12: User Profile Update Main Page

2. Just click on the data tat you would like to update.

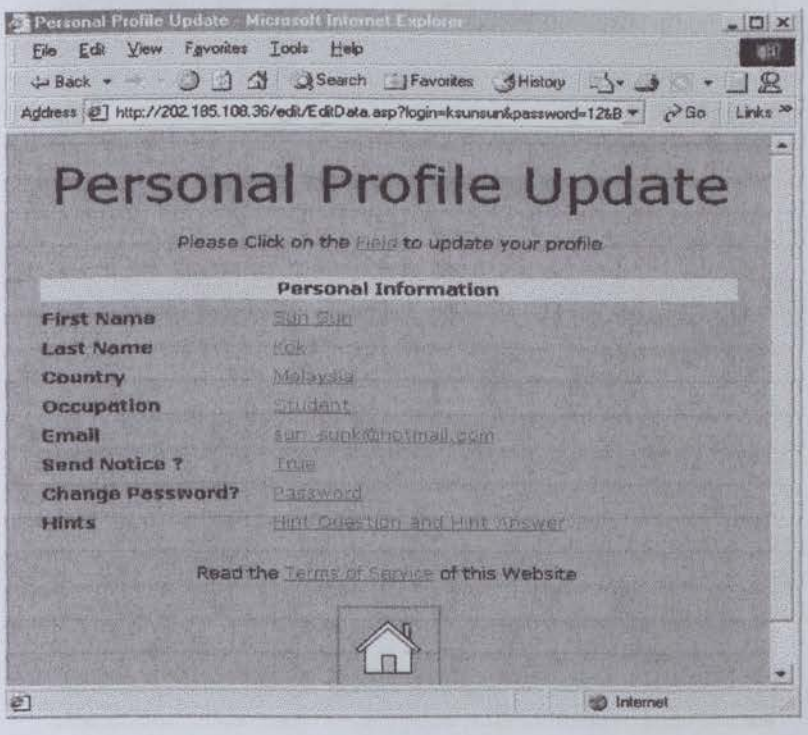


Figure 2-13: Personal Profile Update Page

3. For example, if you click on your name, please fill in the new name.

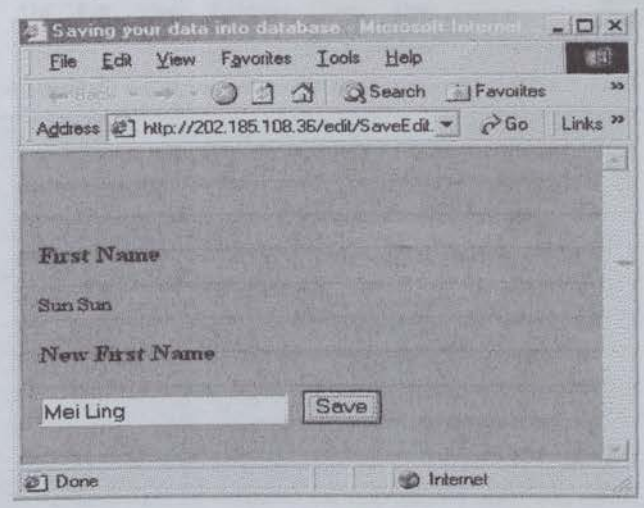


Figure 2-14: User Name Update

4. A message will be displayed, click on the Close Windows button to go back to the initial page.

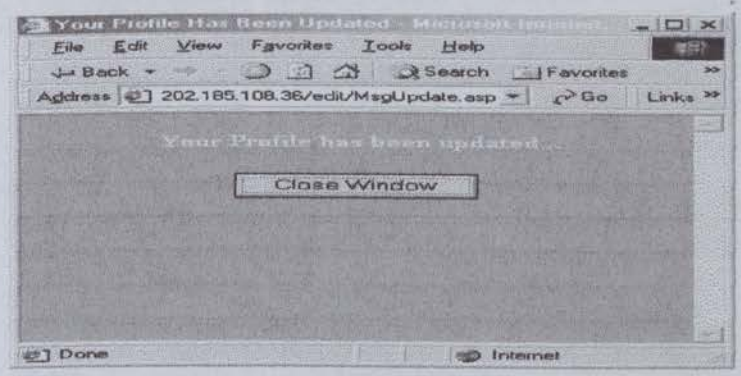


Figure 2-15: Profile Updated Message

5. Here are some of the update samples

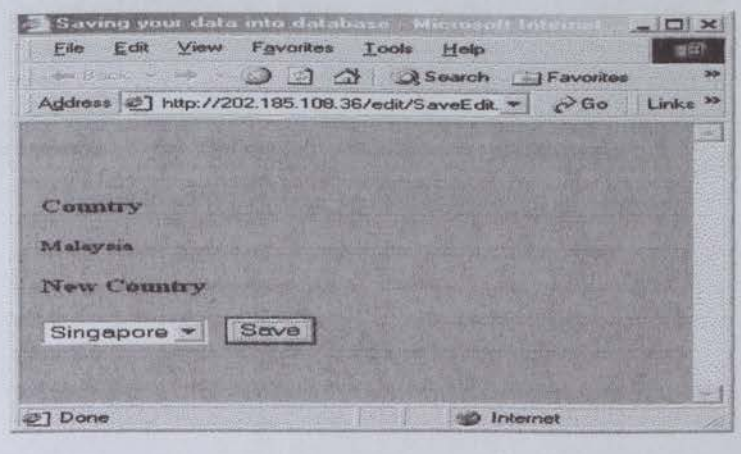


Figure 2-16: Update your country of origin

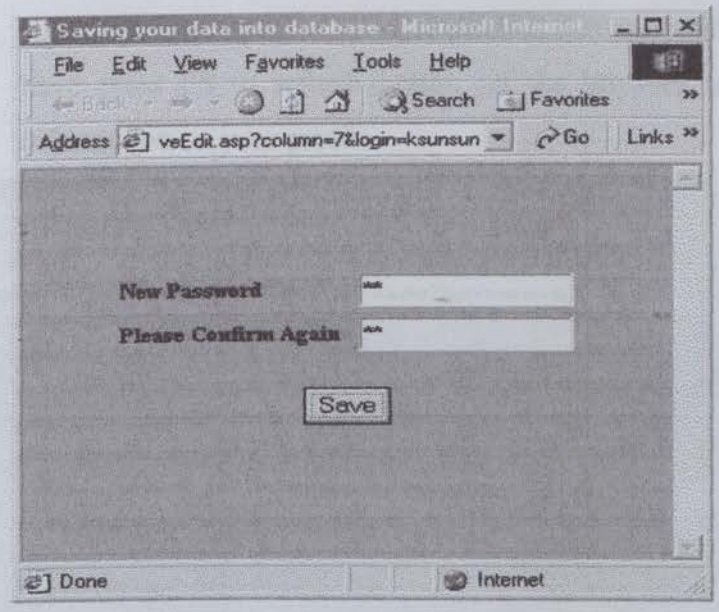


Figure 2-17: To Change your password

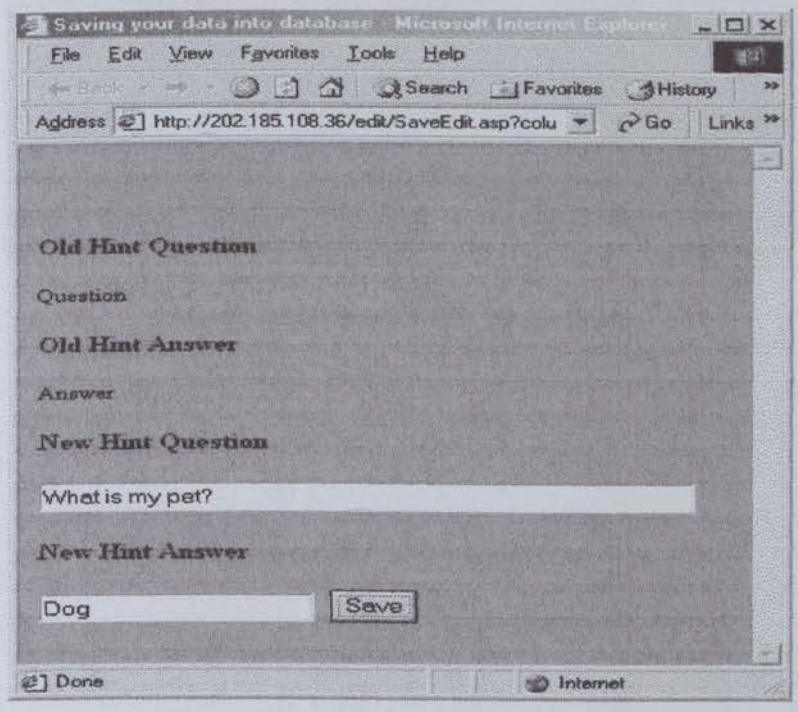


Figure 2-18: To change your hint question and answer

6. After you have updated all your profiles that you wish to update, you have press F5 to refresh button on the browser to get the latest information on your profile.

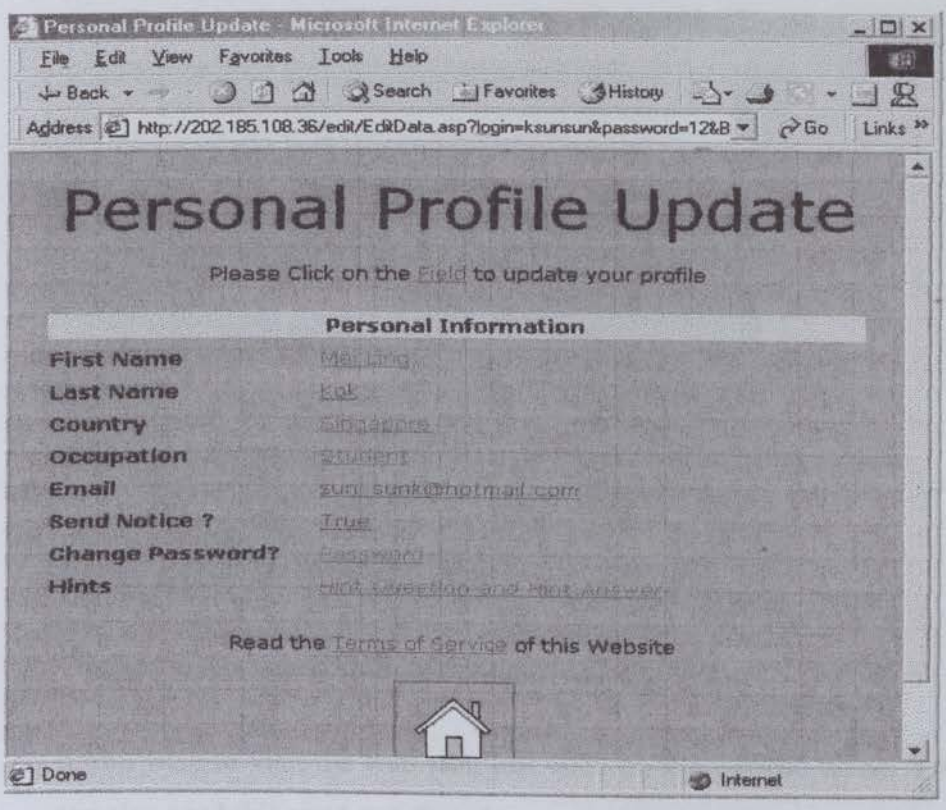


Figure 2-19: Updated User's Information

2.4 View User History Search Records

1. If you wish to see your previous search results, click on the View Your Previous Search link after you have logged in. Login again.

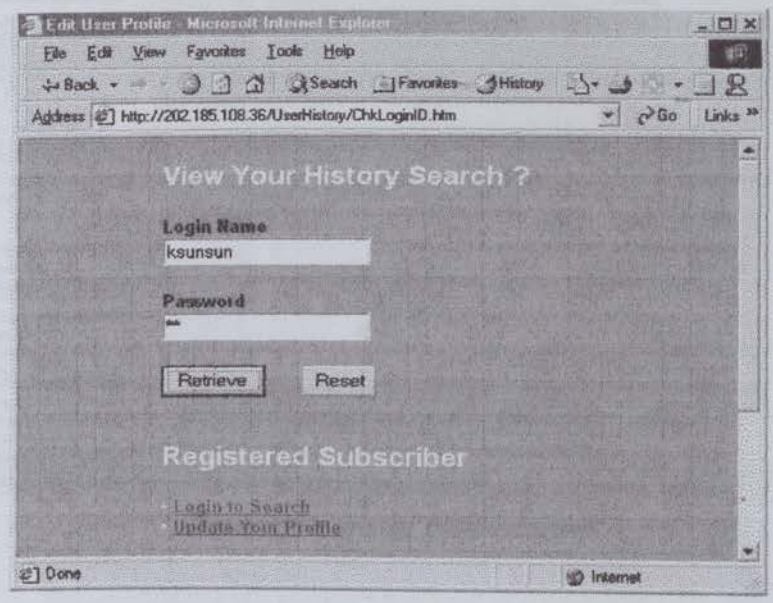


Figure 2-20: Login to view your previous search records.

2. If you wish to visit the web site, just click on the website name. You can also send this page to a friend

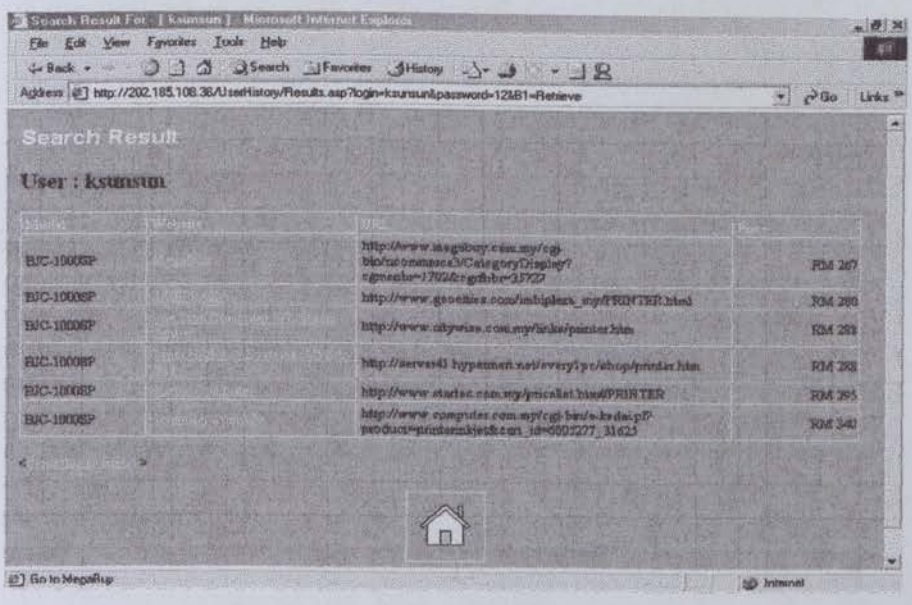


Figure 2-11: The previous search listing

3. A new window will be opened for the website you have chosen.

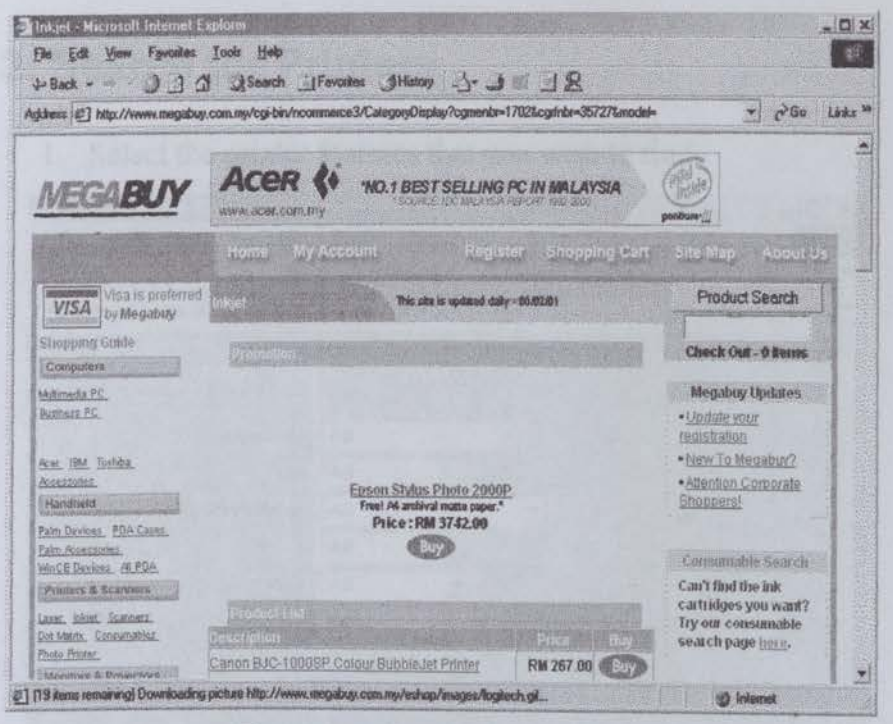


Figure 2-22: Search Result website

Step 3: Search for Printers

1. Select the printer features that you wish to find

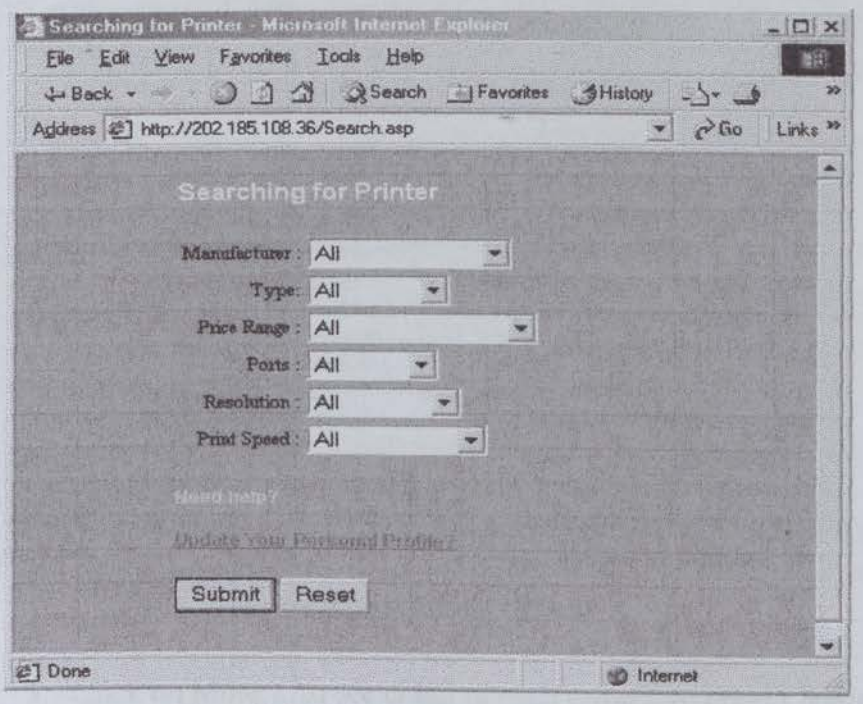


Figure 3-1: Main Page for searching printer

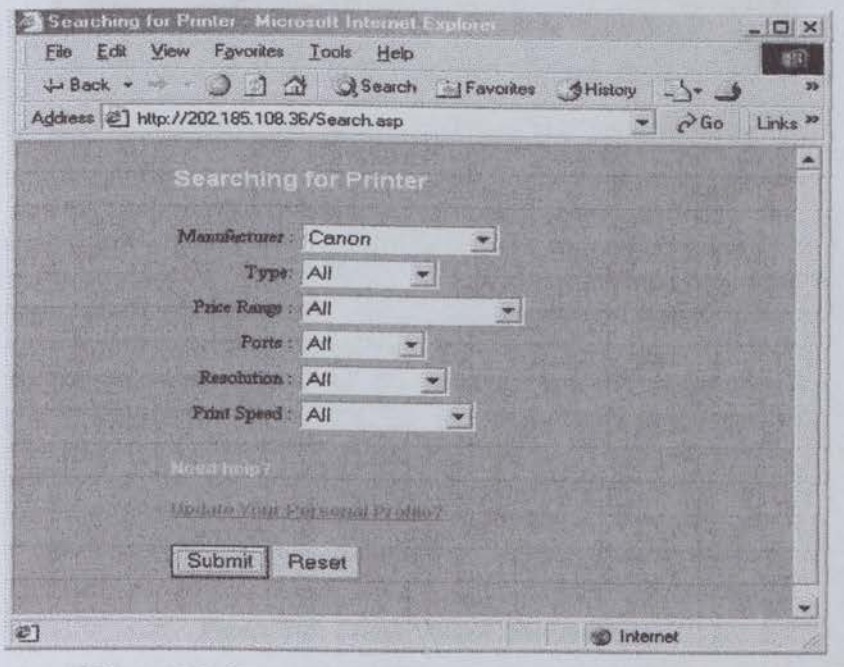


Figure 3-2: Choose the features from the pull-down menu

2. The maximum listing on each page is 10, please click on Next Page to see more printer listing.

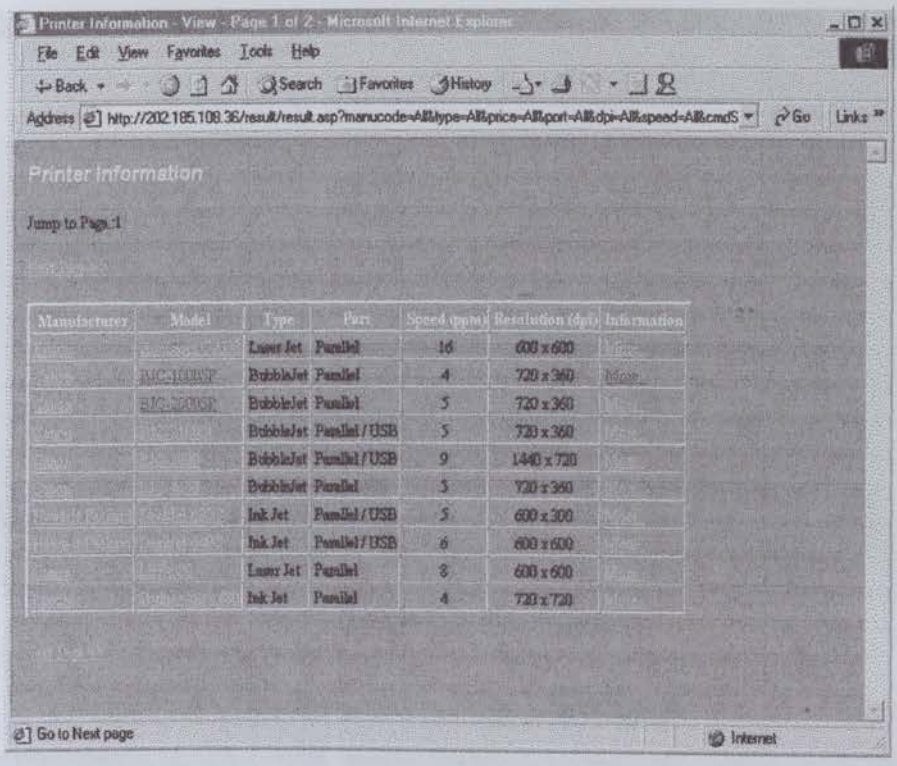


Figure 3-3: Printer Listing

3. Click on More… to see detail of each respective printer features

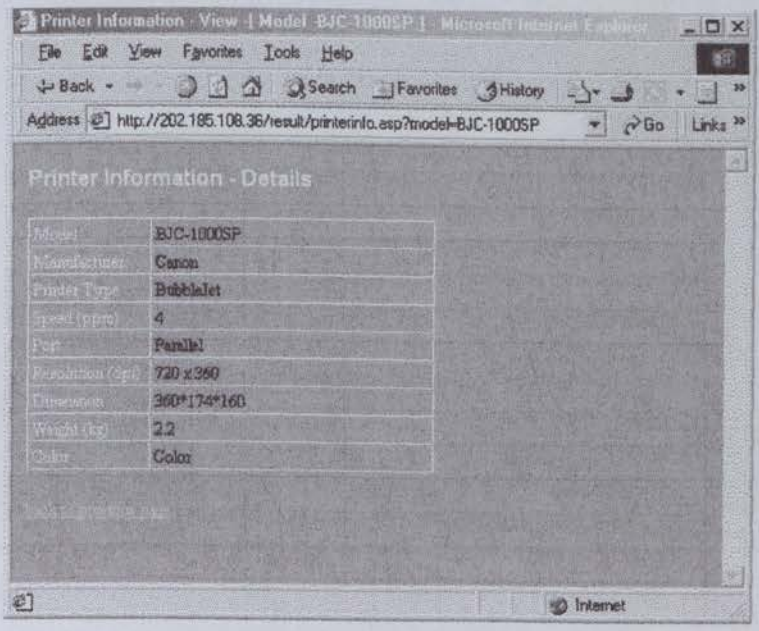


Figure 3-4: Printer Detail Information

4. Click on the printer model to see the price listing

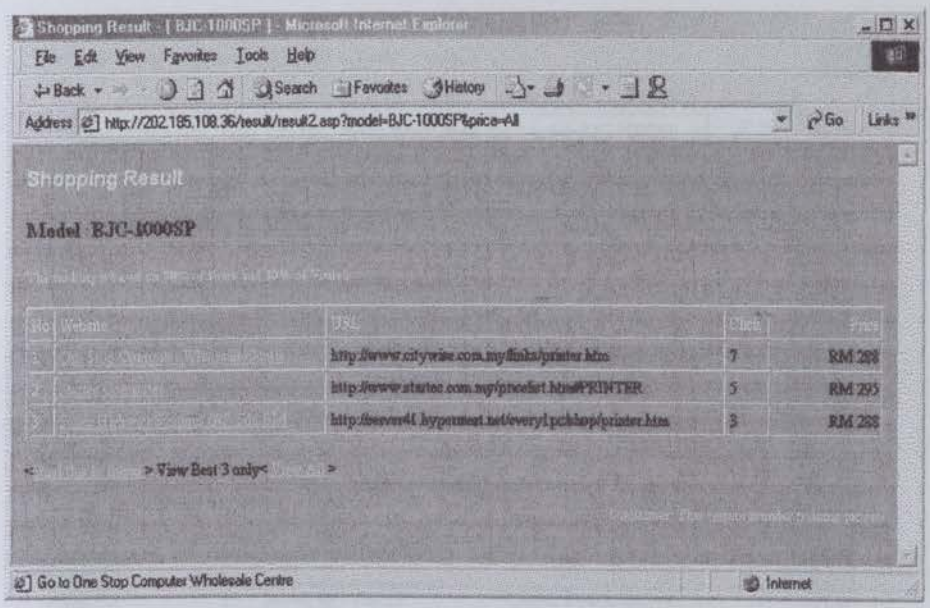


Figure 3-5:Three best Results

5. Click on View All to see the full price listing

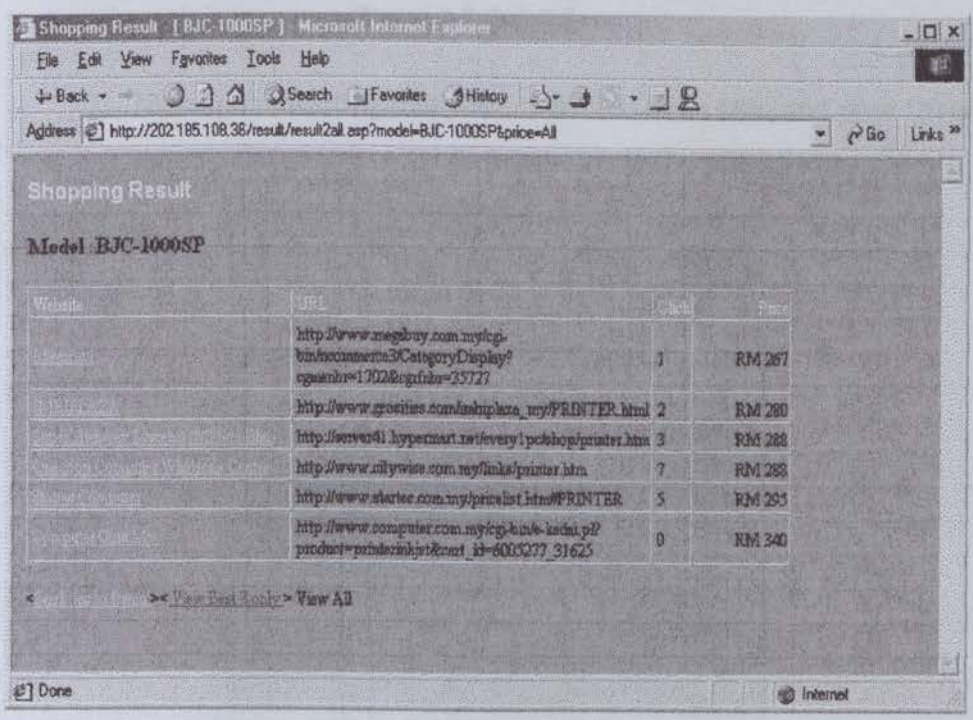


Figure 3-6: Full price listing

6. If you wish to recommend this price listing to your friends, just click on Send to Friends link.

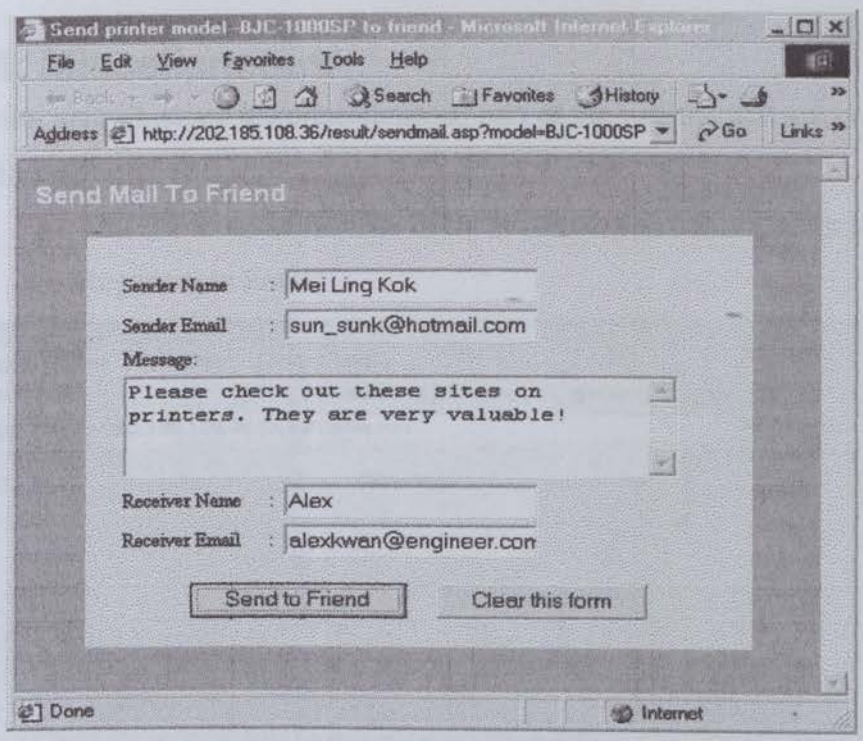


Figure 3-7:Email Form

7. Message indicating that your email has been sent, and your friend will receive the page exactly like you saw in the price listing.

Figure 3-8: Email Sent Message

Step 4: Miscellaneous

This section only illustrate the miscellaneous part of the website that are only for the user's information.

1. This the terms of Service of the web site, you can access to it from the Registration Form.

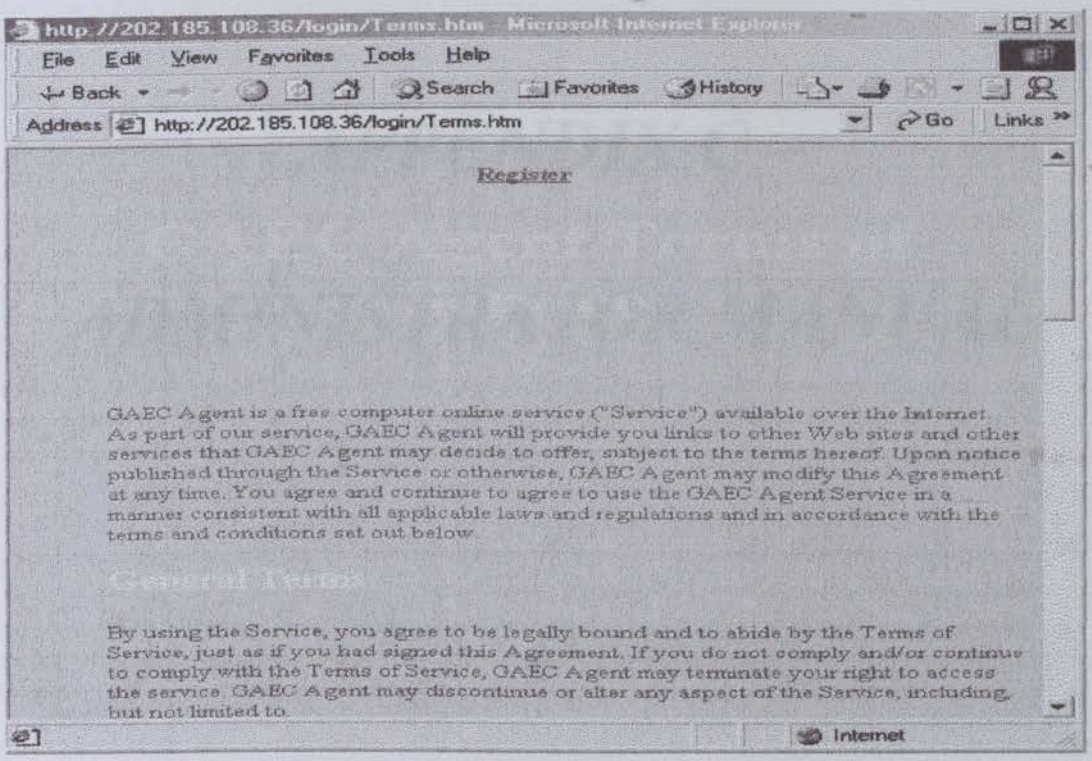


Figure 4-1: The terms of Service of the Website

2. This is the Learn More, which explains in brief of the Agent

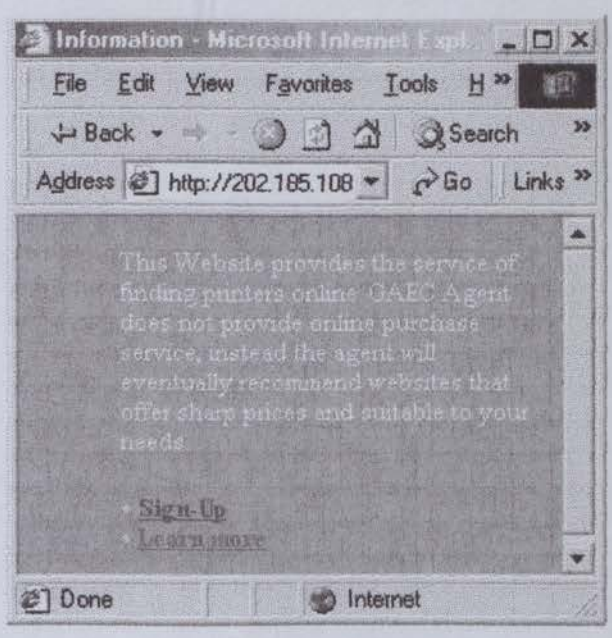


Figure 4-2: Learn More Page

APPENDIX C –

ADMINISTRATOR MANUAL

Contents

List of Figures

Chapter 1 : About The Manual

This manual is designed for the Administrator of the system, and the end-user manual is located at the Appendix B of the report. This admin manual covers the following items:

i. Hardware and software requirements.
ii. How to access to the system.
iii. How to configure the system
iv. How to use the system

Chapter 2 : Hardware and Software Requirements

2.1 Hardware Requirements

- At least Pentium II 233 MHz
- At least 2 GB hardisk
- At least 64 MB RAM
- Other standard peripherals such as monitor, mouse and keyboard.

2.2 Software Requirements

- Operating System : Windows NT Server 4.0 SP6
- Database : Ms SQL Server 7.0
- Web Server : Internet Information Server 4.0
- Tools : Ms Visual Basic 6 SP 4
- Browser : Internet Explorer 4.01 and above

Chapter 3 : Getting Started

As this system is a client/server application, it needs an established network connection to put the system into functions. Before you are able run the application, you must meet all those hardware and software requirements

3.1 Starting Intelligent Agent for E-Commerce System.

To start using the system, you need to

1) **double click** on the icon (in figure A-1)on desktop named "**Intelligent Agent for E-Commerce**", or
2) **click** on the icon on desktop named "**Intelligent Agent for E-Commerce**" and then press **Enter** key, or
3) Click at the start menu on Windows taskbar, and then click at programs, then chooese **Intelligent Agent for E-Commerce,** and click at the **Intelligent Agent for E-Commerce.**

Figure 3-1 : Icon on desktop

3.2 Logon

Login form is the first thing you will see before you are to go into the system. You must a authorized username and valid password. Username and password are not case sensitive when you log into MS SQL Server. After key in the username and password, user needs to click the 'OK' button at the bottom. Username and password validation will be done on server.

.

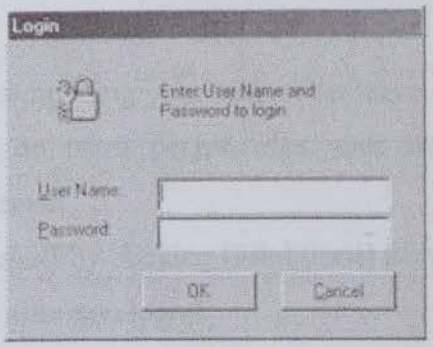


Figure 3-1 : Default Login Page

If the username and password match with the database, a splash screen will show as figure 4-2. And then will direct to the main screen as in figure 4-3

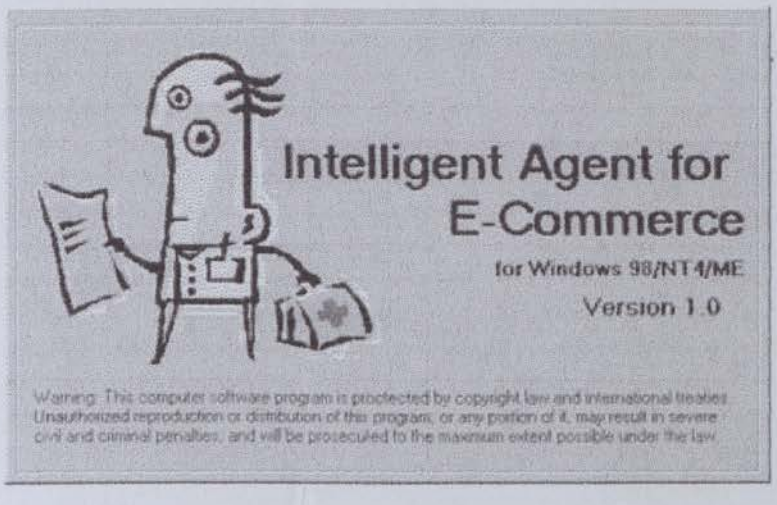


Figure 3-2 : Splash Screen

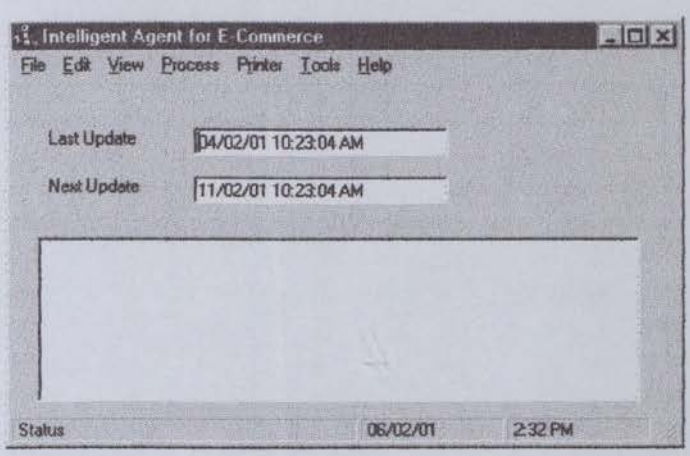


Figure 3-3 : Main Screen

3.3 Log Off

You should logoff after completing your task or to leave the program. This is for security purpose so that no other person uses your authority to do their task, especially unauthorized person.

In the program, you can logoff by clicking [File]|[Exit] from the menu bar. The program will be unloaded after you logoff

Chapter 4 : User Section

4.1 User's Rules and Regulations

1) User must has a valid username and password in order to enter the system.
2) User cannot give his/her sign in information to others, every user will have their own sign in name and password and it's confidential.
3) User can only browse the system and get records information.

4.2 How to configure the system

This system is an automatic system, which will run after every seven (7) days. User can change some options in the Option screen.

To change the option, first you have to chooese [Tools] | [Options...] Figure 5.1 will appear on the center of the screen.

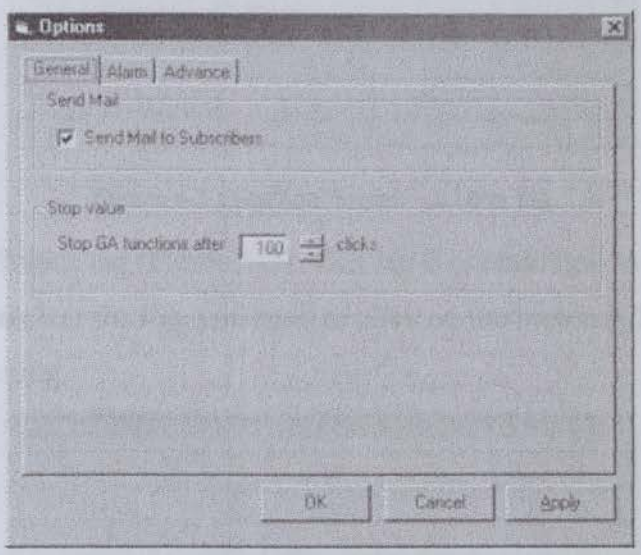


Figure 4-1 : Options screen – General Tag

There are three (3) different group of option. First tab (figure 5-1) is regarding about the general information, which include Sending email to notice the subscribers and the stop value.

To change the send email notification, just click on the check box. The value will change after a mouse click.

To change the stop value, you can use the mouse click on the up or down scroll bar. The value will increase or decrease when you click it. Or you can highlight the value and just type the value.

Second tab is about the alarm interval(figure 5-2). To disable the alarm, you need to uncheck by clicking the check box. To change the alarm interval, you can use mouse th click on the up or down scrool bar to change the day value. The maximum value is 10 and the minimum value is 1.

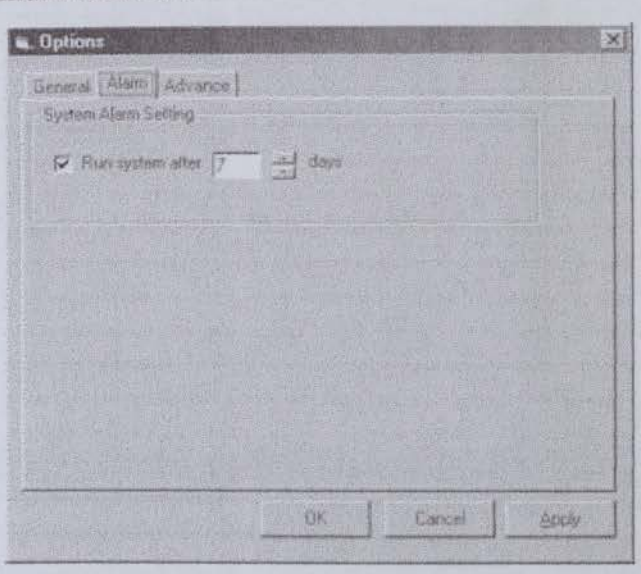


Figure 4-2 : Options Screen – Alarm tag

Third tag is the advance tag (Figure 5-3). This tag is contain option to clear the log in the main menu. To clear the log, you need to click on the button [Clear]. The log will clear after the clicking.

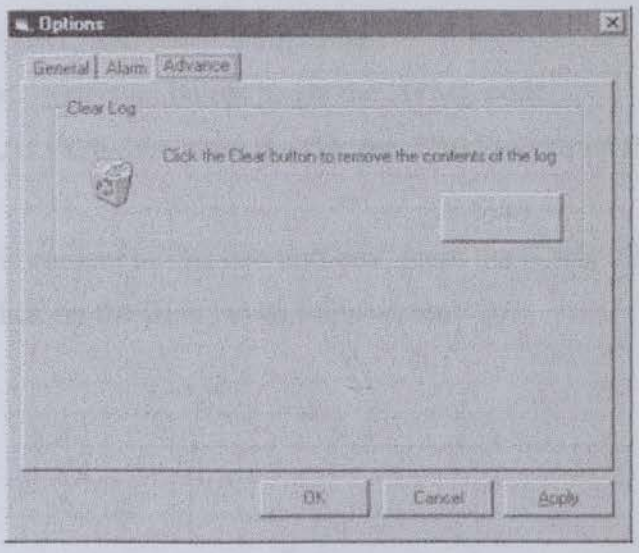


Figure 4-3 : Options Screen - Advance Tag

4.3 How to find URL link manually

To find the url about the printer manually, you need to click on [Process] | [Get Link] from main menu. A form will display as shown in figure 5-4

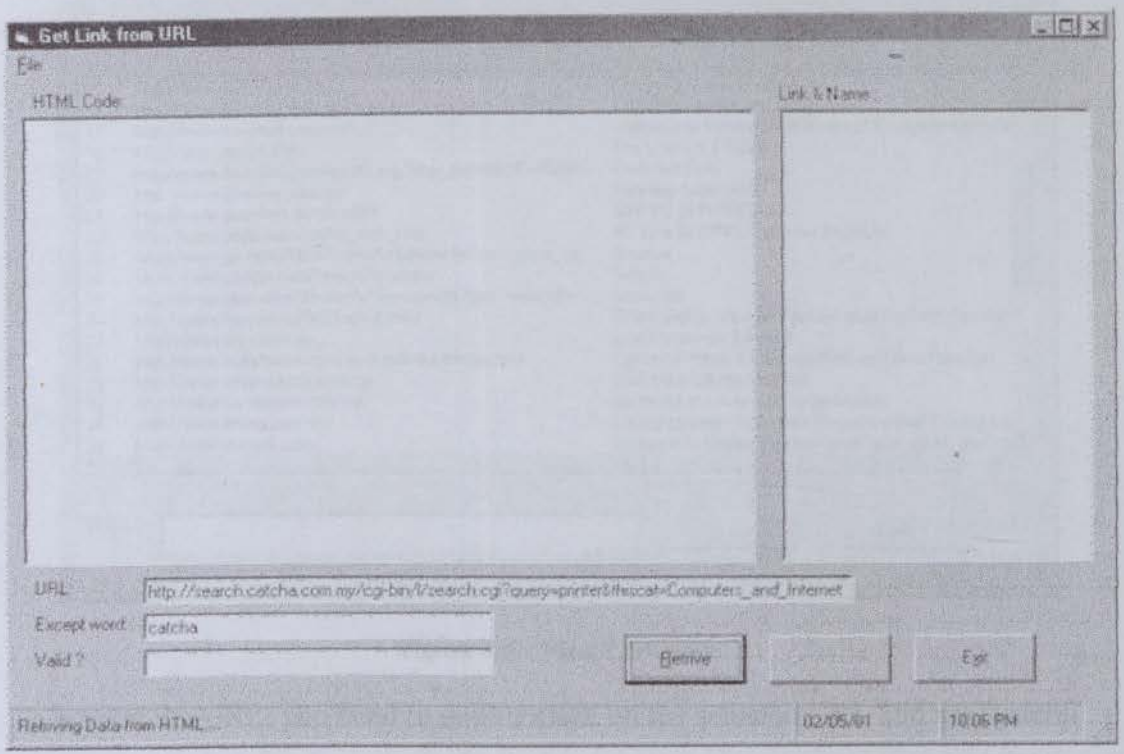


Figure 4-4 : Get Link page

1) You need to enter a URL in the URL text box.
2) An exception word is used to eliminate the url which have the keyword from the current URL, and this is an optional.
3) Press on the [Retrieve] button to get the HTML code.
4) If the URL is valid then press the [Extract Link] to extract the link and the name.
5) To clear the content of this two text box, press the [Clear all] button.
6) You can click on the [Exit] to exit this screen.

4.4 How to find an exact URL

To find an exact URL about the printer's price, user need to filter the URL manually. First click on [Process] | [View Link] , and a form will show as figure 5-3

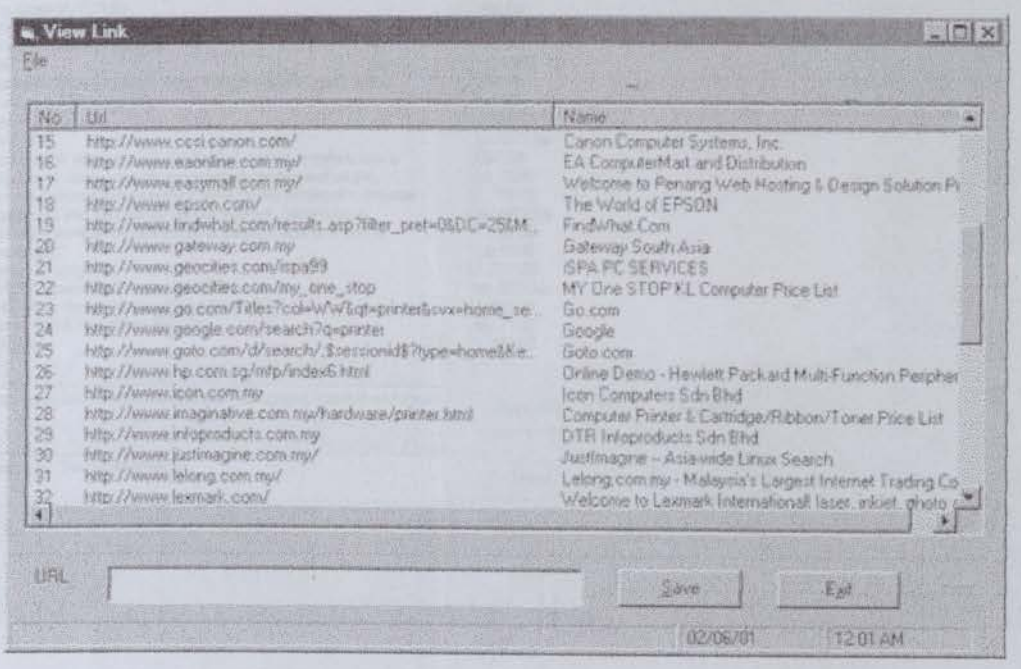


Figure 4-5 : View Link page

To browse the URL, you need to double click on the selected URL and the system will redirect to the web site. Press the [Save] button if you want to save the URL.

4.5 How to find relavent information in HTML

To get the information of printer model and price in a HTML page, you need to click on the [Process] | [Scan Page]. A screen will display as figure 5-6.

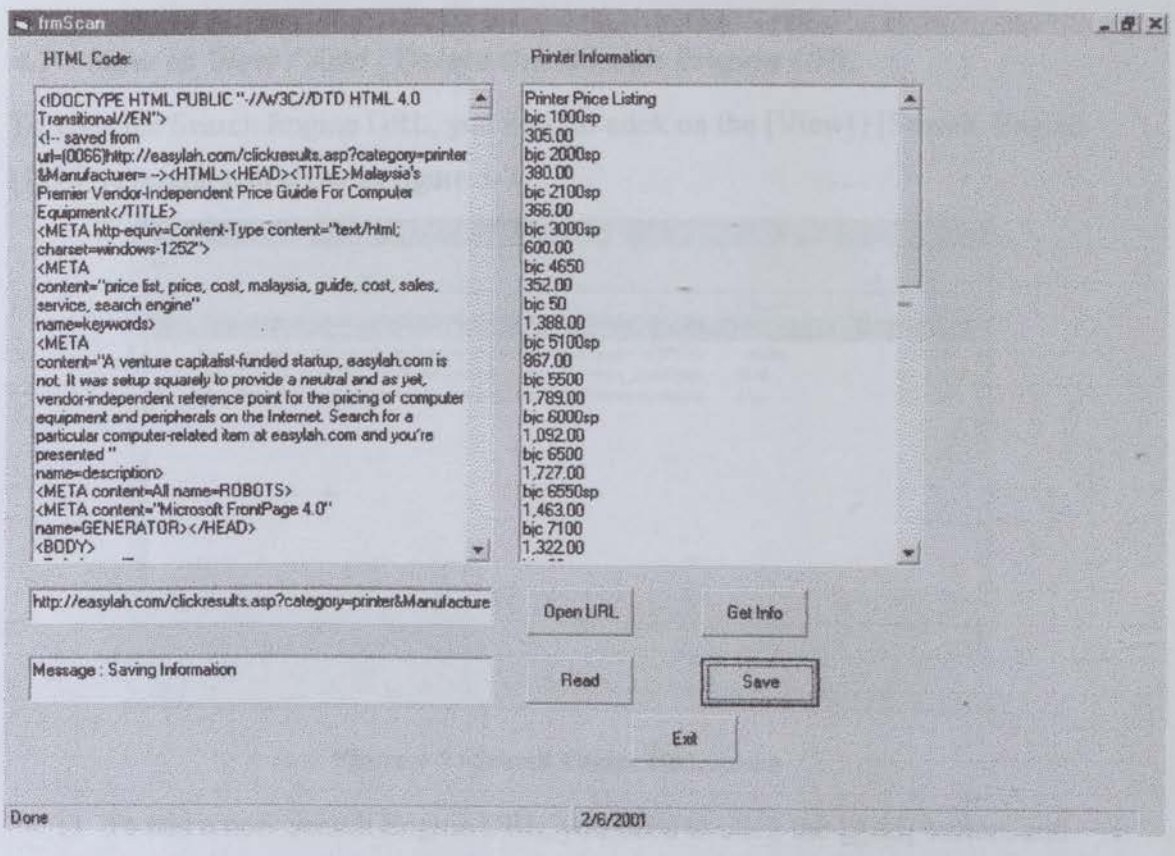


Figure 4-6 : Scan file page

1) Press the [Open URL] to get a URL from the database.
2) To get the HTML code, you need to press the [Read]. The HTML code will show on the upper left box.
3) [Get Info] is used to retrieve the printer model and price inside the HTML code.
4) Press [Save] button if you want to save the information in the database
5) To end this process, click on the [Exit] button.

4.6 How to send email to subscriber

By default the system will send email to user after updated the database. To run this option manually, you need to click [Process] | [Send Email]. Remember: DO NOT uncheck the send email option in [Tools] | [Options...]

4.7 How to View / Add / Delete the Search Engine URL

To view the Search Engine URL, you need to click on the [View] | [Search Engine URL]. A screen will show as figure 5-7

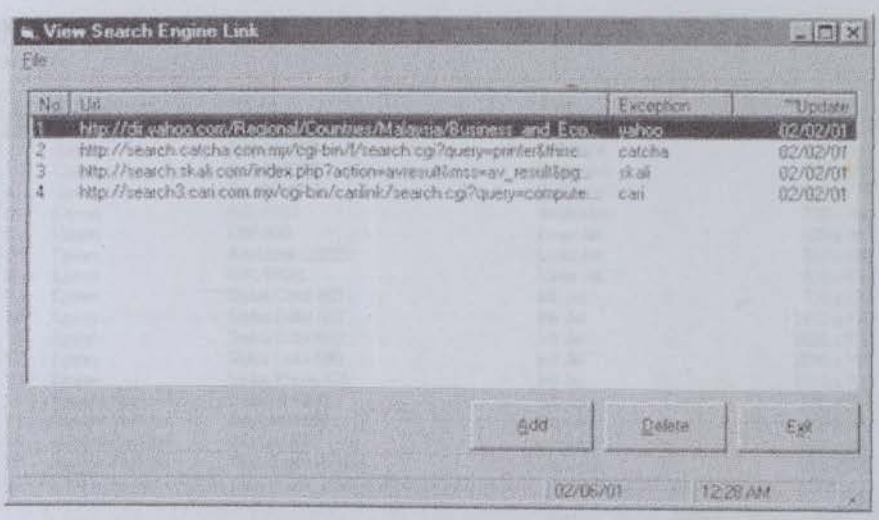


Figure 4-7 : Search Engine URL screen

1) To add a new Search Engine URL, you need to click the [Add] button, and key in the data. You need to fill up two text box before pressing the [Save] button.
2) To delete the URL, you need to press the [Delete] button. System will conmfirm with you before deleting the URL.

4.8 How to View / Add / Delete Printer

To view , add or delete a particular printer, you need to open a page by click at [Printer] | [View Printer Information]. A page will dispay as figure 5-8.

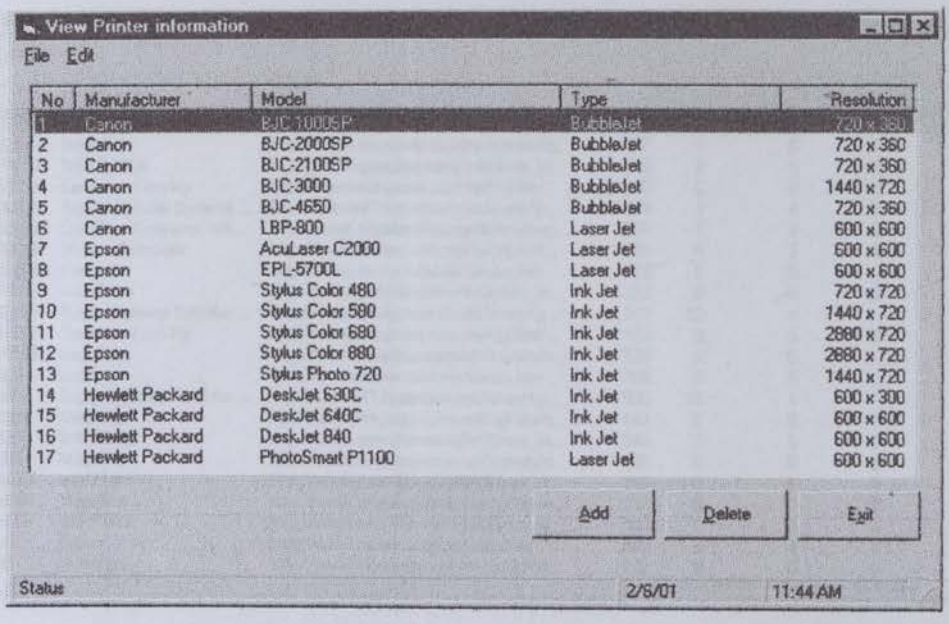


Figure 4-8 : View Printer Information

1) This screen is only display a few information for all printer. To view more details about the selected printer, you need to double click on the printer model.
2) To add a new printer model, you need to click on [Add] button. Fill up all the information and press the [Save] button. Remember, you have to enter all the data and can not enter alphabet in price field
3) To delete a printer model, you have to select a printer model and then press [Delete] button. You CAN NOT delete if you do not selected a printer model.

4.9 How to Add / View an URL

To view a URL you need to click on [Printer] | [Printer's URL List] to open a page as show in figure 5-9

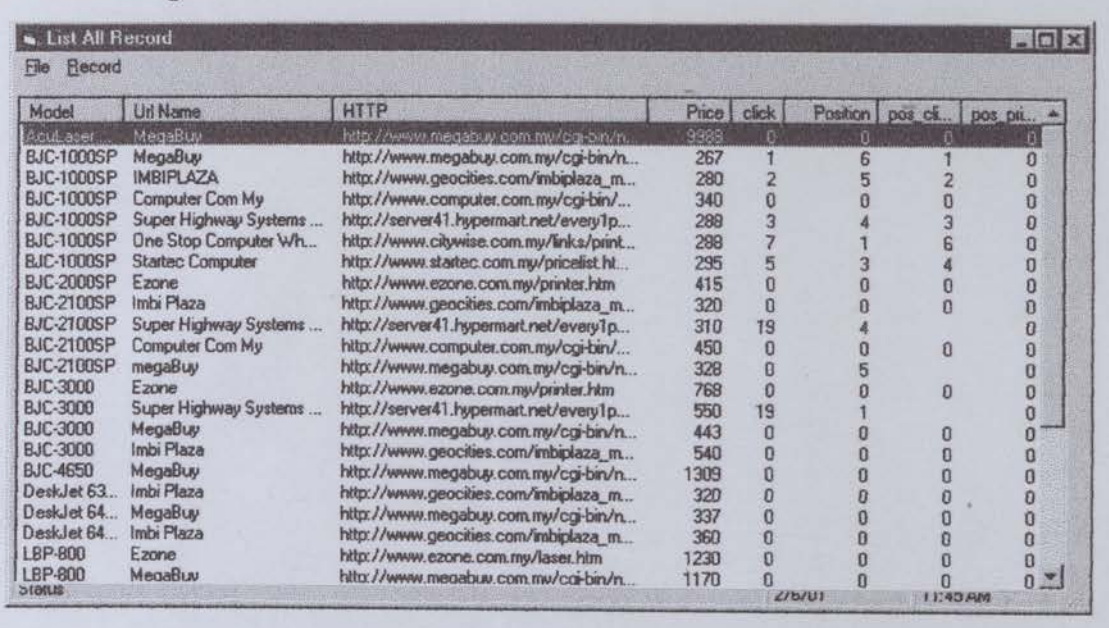


Figure 4-9 : Printer's URL List screen

You can view the URL and printer's price in here. To add a new URL, go to [Record] | [Add], and you need to fill up the information about printer model, URL Name, URL and price. Press the [Save] button when you want to store the information. Do remember, you need to fill up all the field and you must enter a number in price field.