FACULTY
OF
COMPUTER SCIENCE
AND
INFORMATION TECHNOLOGY
UNIVERSITY MALAYA

E-COMMERCE PACKAGE FOR SMALL RETAILERS (E-Pack)

BY
KENNY TEOH KING HOCK
WET 98001

Under the supervision of
Professor Madya Dr. Selvanathan Narainasamy

Session 2000 / 2001

Dissertation submitted in partial fulfillment for
the Bachelors Degree in Information Technology
University Malaya

ABSTRACT

The E-Commerce Package for Small Retailers (E-Pack) is a Web-based application, which serves as a platform to assist small retailers in developing their businesses using the power of the Internet. At the same time, E-Pack will also serve as an agent in promoting the Internet as an alternative means of expanding the coverage of one's business and making it known to the entire world.

E-Pack is basically divided into two main modules, the shopper's module and the administrator module. The shopper's module deals with the client side of E-Pack, such as the online catalogues, shopping cart, and checkout counter, while the administrator's module deals with customer records and shoppers' orders. E-Pack is developed based on the robust three-tier client/server architecture, which consists of the presentation tier, the business tier, and the database tier.

E-Pack is written using Microsoft Visual InterDev 6.0 and Macromedia Dreamweaver 3.0, which include the usage of Active Server Pages (ASP), with VBScript as the server-side scripting language, and JavaScript as the client-side scripting language. The system will be run on Windows NT 4.0 using the Internet Information Server 4.0 (IIS) as the Web server, and Microsoft SQL Server 7.0 as the database management system.

ACKNOWLEDGEMENT

First of all, I would like to thank my supervisor, Professor Madya Dr. Selvanathan Narainasamy, for giving me the opportunity to take on this challenging, yet exciting project. I would also like to thank him for the many constructive advices and guidance that he has given me throughout this project. His never-ending patience and support is deeply appreciated.

Next, I would also like to thank the project moderator, Mr. Ibrahim Abu Bakar, for his kindness and fairness in evaluating this project.

To Pn. Azlin Mohd. Ali, En. Baharuddin Khalid, and the other staffs of the Faculty of Computer Science and Information Technology (FCSIT), I thank them all for the kind assistance that they have given me in the lab throughout the entire project.

I would also like to convey my heartfelt thanks to my seniors, especially Mr. Lin Koon How, for the generous advice and cheering up that he has given me in keeping this project going. Not forgetting my fellow coursemates and friends, who have been very informative and helpful in times of uncertainty, I thank them all very much.

Last but not least, to my loved ones who have been giving me the encouragement and mental support that I so very much needed for this project, I wish to say thank you from the bottom of my heart!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Project Motivation

The Internet is rapidly gaining its reputation as an essential tool for business-related organizations in developing their businesses online. It has become so popular that many companies have already taken advantage of this opportunity by publishing their businesses on the World Wide Web, performing hundreds of transactions online everyday. Consequently, the action of carrying out businesses and transactions online has brought to the birth of a new concept called **electronic commerce**, or better known as **e-commerce**.

With the benefit of the Internet as a borderless market, limitations no longer exist for organizations to expand their businesses online all over the world. As an outcome, the organizations that have pioneered the usage of e-commerce in running their businesses had already begun reaping the fruits of their efforts from the thousands of profits generated daily from their Web sites.

As the role of e-commerce as a business development tool becomes more and more significant, it is highly likely that the number of organizations carrying out their businesses online will increase in the near future. Hence, e-commerce will unquestionably be a beneficial venture for those who dare to take their businesses to the Internet.

1.2 Project Importance

Web hosting is the activity of providing services to people or organizations in publishing their businesses over the Internet. This activity has become so popular in developed countries like the United States and United Kingdom, such that more and more IT-based organizations are moving their businesses into the Web hosting business, hoping to make money from this profitable venture.

However, the same cannot be said for our country. Web hosting is a relatively new concept in Malaysia. This is not surprising considering the level of awareness among IT-based organizations in Malaysia, which is still low as compared to those in developed countries. As such, this project will serve as a platform to cultivate awareness among Malaysian organizations about the concepts and benefits of Web hosting, as well as to encourage them to take advantage of the Internet to venture into the Web hosting business. It will also act as a catalyst for further studies to improve and encourage future development of similar Web based applications like this in Malaysia.

This project will also strive to promote the usage of the Internet in Malaysia as an alternative means of operating businesses. Consequently, this project highlights the benefits of online retailing, and how it is more advantageous than conducting retailing business using the traditional method.

1.3 Project Definition

Before defining this project and making clear what it is all about, it is only appropriate to discuss the reasons that caused this project to be undertaken at the first place.

1.3.1 The Problems

There are two problem areas that contribute to the basis of this project. They are:

i) Inefficient ways of shopping using the traditional approach

ii) High cost of owning and maintaining a server

The first problem will affect both the shoppers and retailers, while the second problem will only affect the retailers. These two problem areas will be briefly explained in the following sections.

i) Inefficient ways of shopping using the traditional approach

Before E-Commerce and online stores came into the picture, shoppers had to travel to stores or supermarkets in order to obtain their daily supply of goods and services. The typical scenario of the entire shopping process will look something like Figure 1-1:

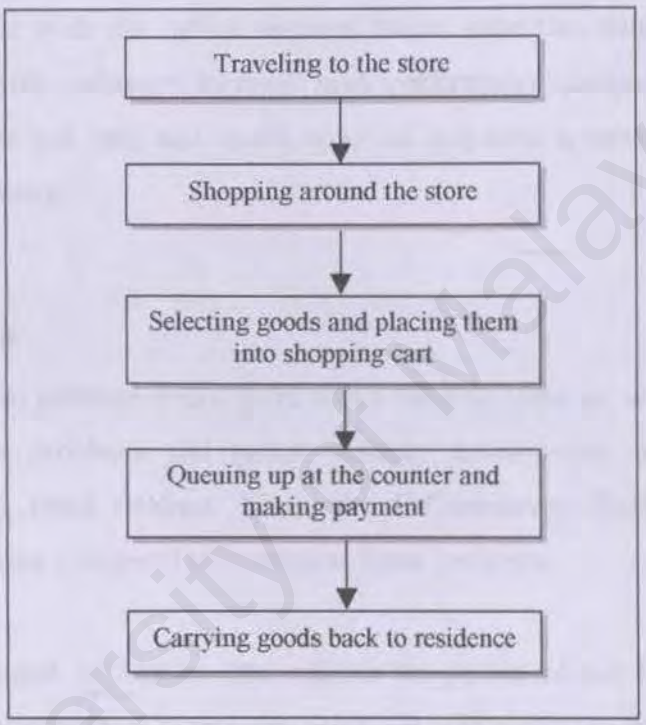

Traveling to the store
Shopping around the store
Selecting goods and placing them into shopping cart
Queuing up at the counter and making payment
Carrying goods back to residence

Figure 1-1: Typical shopping process at a store

Figure 1-1 depicts how shoppers would carry out their shopping in a "traditional" approach. The term "traditional" above literally means in the old fashion way, that is, the way shopping was done prior to the advancement of e-commerce. The traditional approach required shoppers to physically travel to one or more stores in order to perform all of the processes in Figure 1-1. However, this approach has resulted in certain drawbacks, such as:

- Inconvenience of traveling to and shopping around the store
- Unnecessary wastage of resources to travel to the store
- Possibility of erroneous over-the-counter (OTC) transactions
- Excessive usage of paperwork

ii) High cost of owning and maintaining a server

For many small retailers, the businesses that they run usually involve a very small amount of capital. Thus, they cannot afford to purchase a server for the sole purpose of hosting their Web sites. Furthermore, the returns that they might generate from their Web sites may not be sufficient to justify the purchase of a server. Not to mention the high cost of just maintaining the server and the Web site, which requires human expertise, expensive hardware upgrades, costly software licenses, high performance peripherals, and many more. All this will only add to the strain of acquiring a server, especially for the small retailers.

1.3.2 The Solution

Due to the above two problem areas, there was a need to come up with a system that will eliminate these problems and make business development easier and more affordable for these small retailers. Thus, the **E-Commerce Package for Small Retailers** (E-Pack) was conceived to overcome these problems.

E-Pack is a Web-based application that utilizes the power of the Internet to allow small retailers to develop their businesses online. It will make use of the latest Web technologies and tools to provide solutions for small retailers in developing their businesses on the World Wide Web. E-Pack will combine both business-to-business **(B2B)** and business-to-consumer **(B2C)** e-commerce into its architecture

E-Pack will comprise of two main modules, which is the shopper's module and the administrator's module. The shopper's module will cover the client side of E-Pack, which includes the online catalogues, shopping cart, and checkout counter, while the administrator's module will handle customer (in this project, the retailers are the customers) records and shoppers' orders. The basic modules of E-Pack are illustrated in Figure 1-2:

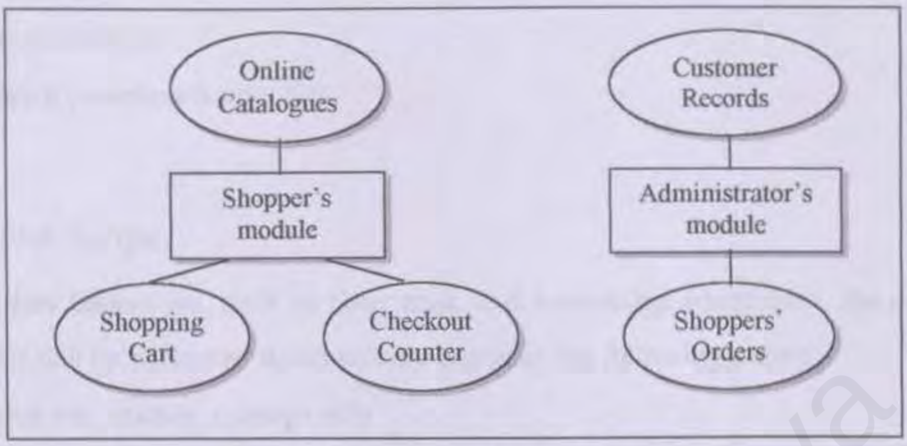


Figure 1-2: Basic modules of E-Pack

E-Pack will act as a Web host that provides simple Web hosting services to small retailers at an affordable price, so that these retailers will have a means of operating their businesses on the Internet. However, these small retailers are expected to subscribe to the services of E-Pack on an annual basis, at a nominal fee before their products and services are published on the Internet. Subscription will be through mail, telephone or facsimile, whichever convenient to these retailers. Members in turn will enjoy the benefit of having their businesses up and running 24 hours a day, everyday on the Internet, generating income as we speak.

1.4 Project Objectives

It is clear up to now that the primary objective of E-Pack is to facilitate small retailers in taking their businesses to the Internet. However, E-Pack also consists of the following, equally important objectives:

- Provide an affordable package for small retailers to run their businesses on the Internet
- Make full use of the latest Web technology to help small retailers in developing their businesses online
- Provide an uncomplicated way of accessing and purchasing products on the Internet through online catalogues, shopping carts, and checkout counters
- Reduce the hassles of over-the-counter transactions

- Take advantage of the borderless marketing concept to increase the coverage of business activities
- Achieve a paperless transaction

1.5 Project Scope

Due to a few limitations, such as time, cost, and knowledge constraints, the scope of this project will be narrowed down to only consider the following points:

- Involves the retailing concept only
- Covers small retailers only (with small inventories for simplicity)
- Subscription of membership will be done through the physical medium (e.g. mail, telephone, facsimile, etc.) instead of the Internet. This is because it is assumed that these small retailers do not have any knowledge of the Internet
- Type of goods sold will depend on the industry that the retailers are involved in
- Covers the Malaysian market only (no foreign transactions)
- All shopping transactions will be made online, except for payments, which will be made through cheques or money orders
- Comprises of only two main modules:
  - Shopper's module – includes online catalogues, shopping cart, and checkout counter.
  - Administrator's module – handles the customer records and shoppers' orders. (The administrator's module will not take into consideration inventory management, as it is assumed that the small retailers will always have the number of items ready in hand, as requested by the shoppers.)

1.6 Project Methodology

The strategy used in the development of E-Pack is the waterfall model. As the name implies, the stages of development cascades from one phase to another. Each stage of a development is required to be completed before proceeding to the next phase. The waterfall model offers the benefit of a structured development, in addition to good visibility and proper documentation for each development stage.

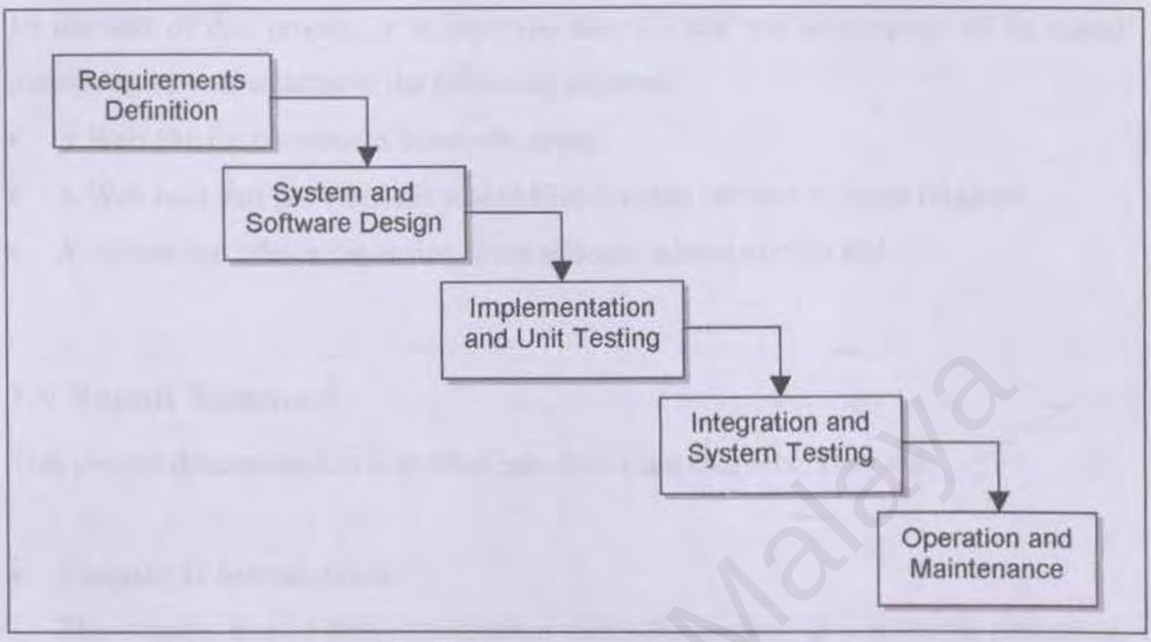

Requirements Definition
System and Software Design
Implementation and Unit Testing
Integration and System Testing
Operation and Maintenance

Figure 1-3: Waterfall model

1.7 Project Schedule

A project schedule is drawn up to estimate the time and resources required to complete each activity in the first stage of this project. Its purpose is also to organize these activities in a coherent sequence, so that works can be carried out smoothly.

Figure 1-3 shows a Gantt chart depicting the planned schedule for this project:

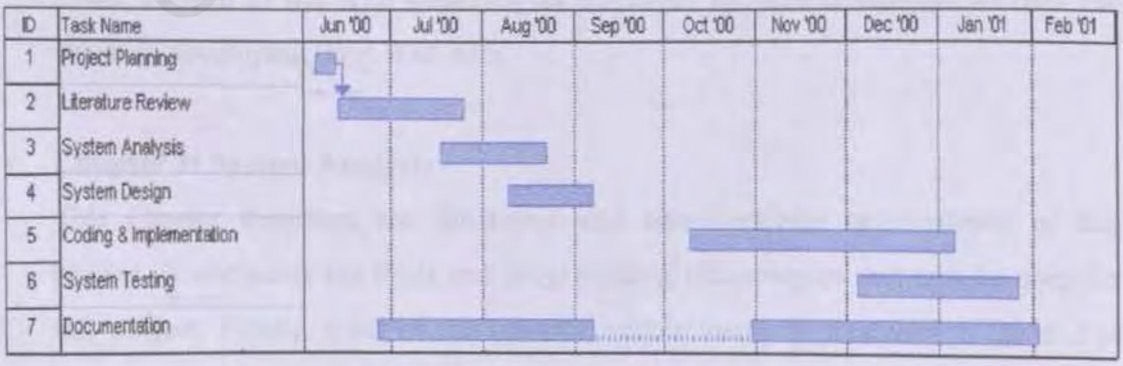


Figure 1-4: Project schedule in Gantt chart form

1.8 Expected Outcome

At the end of this project, it is expected that E-Pack will accomplish all its stated objectives, as well as achieve the following outcome:

- A Web site for convenient home-shopping
- A Web host that provides affordable Web hosting services to small retailers
- A system that allows for easier, more efficient administration tasks

1.9 Report Summary

This project documentation is divided into four main chapters. They are:

- **Chapter 1: Introduction**

  This chapter gives a brief introduction about this project, as well as the objectives, scope, and importance of this project. It also mentions the method used, and the time allocated for this project.

- **Chapter 2: Literature Review**

  This chapter gives a review on the importance of the Internet and e-commerce as a business tool to merchants all over the world. In addition, this chapter will also compare a couple of e-commerce Web sites and dotcom companies on the Internet in order to ascertain the current level of e-commerce today and what each dotcom company had to offer. It will also compare the Web security protocols in use today, as well as the Web technologies employed by Web developers all over the world in developing their Web sites.

- **Chapter 3: System Analysis**

  This chapter describes the functional and non-functional requirements of this project. It compares the tools and programming technologies that may be used for this project. Finally, a list of the run-time requirements is presented at the end of this chapter.

- **Chapter 4: System Design**

  This chapter gives a review on how requirements are transformed into a working system. A set of components and inter-component interfaces are determined before the real implementation.

- **Chapter 5: System Implementation**

  This chapter, deemed to be one of the most important sections of this report, illustrates how the suggested programming technology and development tools were used in the development of this system. It is a close representation of the actual work done on the system.

- **Chapter 6: System Testing**

  This short chapter notes down the tests carried out on the system as well as the level of testing employed during the development of the system.

- **Chapter 7: System Evaluation**

  This chapter lists out all the project findings, such as system strengths, system limitations, future enhancements, problems encountered, and knowledge gained, during the development of this project. Finally, an overall conclusion is presented at the end of this chapter.

CHAPTER 2: LITERATURE REVIEW

2.1 The Internet

2.1.1 History

The Internet represents one of the most successful examples of the benefits of sustained investment and commitment to the research and development of information infrastructure. Beginning with the early research in packet switching technology, the government, industry and academia have since been partners in evolving and deploying this exciting new technology.

Although the Internet has been around for quite a while, many still do not know of its origin. It all began in August 1962, when J.C.R. Licklider of MIT Laboratory for Computer Science discussed his **"Galactic Network"** concept through a series of memos that he wrote himself [**Leiner, 2000**]. He envisioned a globally interconnected set of computers, through where everyone could quickly access data and programs from anywhere in the world. This concept was very much like the Internet of today.

The first ever representation of the Internet that came into existence was the ARPANET. The ARPANET started in late 1969 with only four host computers connected together at different locations in the United States. The Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense (DoD) created the ARPANET as an experiment in packet-switched computer networking.

In 1973, the U.S. Defense Advanced Research Projects Agency (DARPA) initiated a research program to investigate techniques and technologies for interlinking various kinds of packet networks. The objective of this program was to develop communication protocols that would allow networked computers to communicate transparently across multiple, linked packet networks. This program was called the "**Internetting**" project, and the system of networks that emerged from the research was known as the Internet [**Cerf, 2000**].

The system of protocols that was developed over the course of this research effort became known as the **TCP/IP Protocol Suite** [**Cerf, 2000**]. The TCP/IP protocols define how data is subdivided into packets for transmission across a network, as well as how applications can transfer files and send electronic mail. For those applications that did not want the services of the Transmission Control Protocol (TCP), an alternative called the User Datagram Protocol (UDP) was added in order to provide direct access to the basic service of IP (Internet Protocol).

Although the ARPANET was the precursor to the Internet, the two are not the same. The first protocols used on the ARPANET were not TCP/IP; rather the TCP/IP protocols were invented using the ARPANET [**Quaterman, 1994**]. The Internet evolved from the ARPANET, and sometimes people still confuse the ARPANET with the Internet. This confusion results probably because the ARPANET was the first backbone network of the early Internet and remained part of the Internet until it was retired in 1990. Figure 2-1 illustrates the stages in the evolution of the Internet:

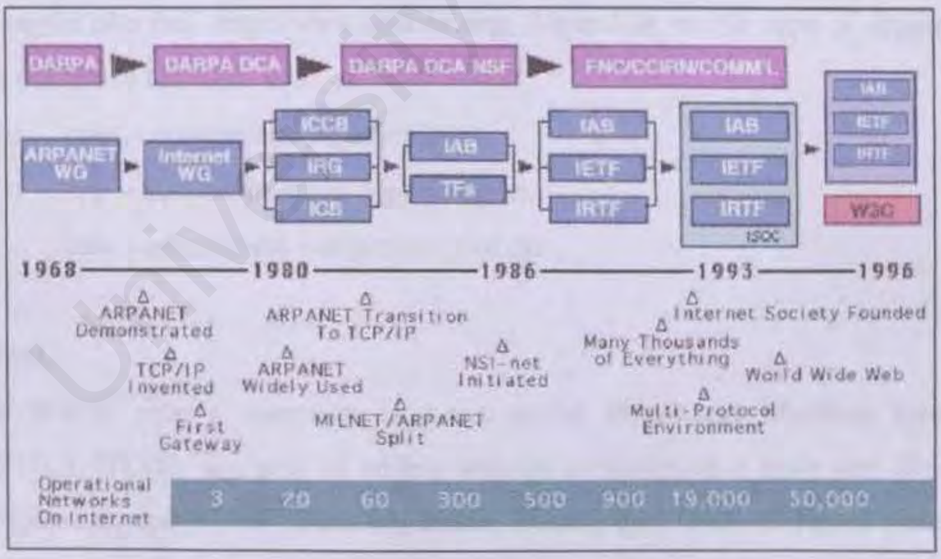


Figure 2-1: Timeline showing the evolution of the Internet

A major initial motivation for both the ARPANET and the Internet was resource sharing. It was considered more economical to connect two computers together to share information rather than duplicating the information for both computers. A key concept of the Internet is that it was not designed for just one application, but as a general infrastructure on which new applications could be conceived, as illustrated by the emergence of the World Wide Web. It is the general-purpose nature of the service provided by TCP and IP that makes all this possible.

2.1.2 World Wide Web

With the emergence of the Internet came the **World Wide Web** (WWW). The WWW is the most popular and fastest growing part of the Internet, with the others being FTP, Gopher, and Telnet [**Ball & McCulloch, 1996**].

- *URL*

  Every Web site on the WWW has a specific address called a **Uniform Resource Locater** (URL). In order to visit a specific Web site, a Web surfer will have to enter the URL address of that Web site into the location bar on a Web browser. A URL address will most probably look something like this: *http://www.yahoo.com*, or maybe like this: *http://www.apache.org*, depending on the type of organization that published the Web site:
  - *.com* – commercial organizations,
  - *.org* – non-commercial organizations,
  - *.edu* – educational institutions, and etc.

- *HTML*

  The WWW uses a computer language called **Hypertext Markup Language** (HTML). HTML is a way of adding various attributes to a plain text file that is published on the WWW [**Duncan, 1998**]. In fact, the HTML document itself is an ordinary text file, renamed with the ".html" or ".htm" file suffix. In 1991, Tim Berners-Lee created HTML as a way of marking up technical papers so that they could easily be transferred across different platforms [**Reynolds, 2000**]. He created a set of tags that he used to mark up a document so that others could easily render it into a usable format.

When HTML was used exclusively by the scientific community, they were not concerned with how the documents look like, only that it be readable and usable. However, as Web browsers became increasingly popular with people who were not in the scientific community and they began to create their own Web pages, appearances became much more important. To meet the demand, developers of Web browsers began to add their own tags that allowed users to display ever more appealing documents. This led to the inevitable drawback of Web pages being designed for one browser over the other. That is the reason why some Web pages on the WWW today could be viewed on a specific Web browser, but not on the other, and vice versa.

- ***HTTP***

In order for HTML documents to be transferred across the WWW, the HTTP protocol is used. The **Hypertext Transfer Protocol** (HTTP) is an application level TCP/IP protocol that travels on top of TCP [**Reynolds, 2000**]. It defines how HTML documents are to be transferred from server to client.

Each HTTP transaction consists of a request and a response: the client makes a request to the server, and the server provides the response to the client. To carry out HTTP requests, there must be an HTTP or Web server running on the target machine. The server is an application that listens for and responds to HTTP requests on a certain TCP port (by default, port 80). A HTTP transaction is made up of four basic states:

1. The connection
2. The request
3. The response
4. The disconnection

The client software (a Web browser in this case) creates a TCP/IP connection to an HTTP server. Once connected, the client sends a request to the server. An HTTP request is usually for a single item from the Web server. The item may be anything from a Web page to a sound file. Upon receipt of the request, the server attempts to retrieve the data asked for. If the server finds the correct information, it formats and returns the data to the client. If the requested information could not be found, the server will return an error message. After the server has responded to the request, it closes the connection, thus disconnecting the client. Subsequent requests will require the client to re-establish connections with the server.

One of the features of WWW that makes it so popular to millions is its ease of use. Most of the Web sites on the Internet are published on the WWW because it covers a wider assortment of audiences. Furthermore, it possesses excellent graphics, images, and sound capabilities, which makes it ideal for multimedia presentations. It is no wonder that the number of corporations using the WWW to promote and sell their products is increasing day by day.

2.2 E-Commerce

2.2.1 Definition

Till this moment, there is still no single definition for electronic commerce, or better known as e-commerce. Experts have churned out various definitions of e-commerce to explain this phenomenon, yet not one definition accurately describes what e-commerce really is.

E-commerce can be described as a method of buying and selling products and services over the Internet [**Kosiur, 1997**]. Some define e-commerce as the execution or carrying out of a business transaction, supported by advanced information technology to increase the effectiveness of the business relationships between trading partners [**CNEC, 1997**]. Yet another definition of e-commerce is the enablement of a business vision supported by advanced information technology to improve efficiency and effectiveness within the trading process [**ECIC, 1998**].

Whatever definitions given to e-commerce, one thing is for sure. E-commerce is a mix of business vision and enabling technologies. It extends beyond the boundaries of a single enterprise and relies largely upon electronic exchange of data. This refers mainly to the "market space", which is the virtual context in which buyers and sellers discover one another, and transact business. In other words, the market space can be understood as the working environment that arises from the complexity of increasingly rich and mature telecommunications-based services and tools, and the underlying information infrastructure.

2.2.2 Traditional Commerce vs. E-Commerce

Traditional Commerce

Before e-commerce became the talk of the town, business transactions were done manually in a physical environment. For example, in order to purchase some sundry goods, a consumer had to physically travel to the store to get the required items. Once at the store, the consumer would have to shop around the store to look for the desired items, pick up those items, walk around the store with the items in a shopping cart, and finally make payments for them at the counter. A typical scenario of this traditional approach was previously depicted in Figure 1-1.

Traditional commerce has brought to a few drawbacks:

- **Inconvenience of traveling to and shopping around the store**

  Consumers today are living in an increasingly hectic lifestyle. They may be too busy with office work, or even housework. As such, it can be inconvenient for them to travel to the store just to get a few things. Furthermore, after a hectic day at work, a consumer may not have the time and energy to travel to and shop around the store. Instead, their main concern will be to rest at home after a hard day's work.

- **Unnecessary wastage of resources to travel to the store**

  Traveling to the store may pose another problem for the consumer. If the store is strategically located near the consumer's residence, then there would not be much of a problem for the consumer to travel to the store. However, if the store is situated far away, then considerable traveling costs may occur just to travel to the store. Not to mention the wastage of fuel, as well as vehicle wear and tear.

- **Possibility of erroneous over-the-counter (OTC) transactions**
  Even though employees can be trained to handle OTC transactions efficiently, there are still chances for transaction errors to occur. Errors like incorrect calculation, incorrect receipt of cash, and many more can cause considerable damage to the well being of a business.
- **Excessive usage of paperwork**
  Paperwork is a must for traditional commerce in order to record the history of previous transactions executed. This could lead to messy paperwork, as well as inefficient filing and inventory systems.

As such, there was a need for a business model that will offer flexibility, convenience, and efficiency in performing business transactions, minus the hassles of the traditional approach. Thus, the concept of e-commerce was conceived.

E-Commerce

As defined previously, e-commerce is basically the buying or selling of goods and services electronically over the Internet. The term "electronically" here indicates that the process of buying or selling is mostly executed using the advancement of information technology and its components.

With e-commerce, consumers no longer need to travel to stores in order to obtain their daily needs. All they have to do is browse the Internet from the comfort of their homes, at their own convenient time. Furthermore, consumers can now use the time and energy saved from traveling to the store to perform their household chores, as well as to strengthen family ties.

Other advantages of e-commerce are:

- Shorten procurement cycles through use of on-line catalogues, ordering and payment
- Cut costs on both stock and manufactured parts through competitive bidding
- Reduce development cycles and accelerate time-to-market through collaborative engineering and product implementation

- Gain access to world wide markets at a fraction of traditional costs
- Ensure product, marketing information and prices are always up to date.
- Eliminate erroneous over-the-counter transactions.
- Reduce messy paperwork and documentation.

2.2.3 Categories Of E-Commerce

A factor that differentiates the various categories of e-commerce is the agents that carry out the e-commerce transactions. As such, we can summarize e-commerce into five main categories [**CNEC, 1997**]:

i) **Business-to-Business (B2B)**

An example in this category would be a company that uses a network for ordering goods from its suppliers, receiving invoices and making payments. It is basically the process of selling items to another business for the purpose of reselling it to another entity.

ii) **Business-to-Consumer (B2C)**

This category has expanded greatly with the advent of the World Wide Web. It is currently the main form of most e-commerce Web sites today, where organizations interact directly with the consumers.

iii) **Business-to-Administration**

This category covers all transaction between companies and government organizations. For example, in the United States, the details of forthcoming government procurements are publicized over the Internet and companies can respond electronically.

iv) **Consumer-to-Administration**

This category has not yet emerged. However, in the wake of growth of the second and third categories, governments may extend electronic interaction to such areas as welfare payments and self-assessed tax returns.

v) **Bank-to-Consumer/-Business/-Government**

This category seems to gain complexity as well as importance, taking into account that the payment will have to be certified and secured by a third party, which is the key agent to close the deals among trade partners. An example of this category is the online banking services provided by banks.

Conclusively, we may represent the e-commerce environment as in Figure 2-2:

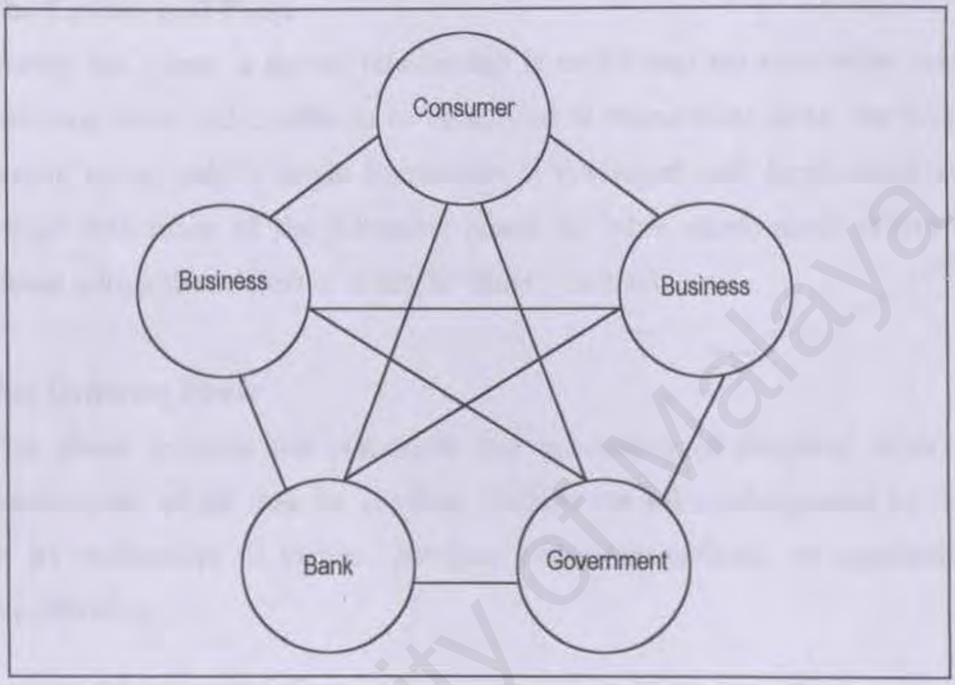


Figure 2-2: The environment of e-commerce

However, most of the e-commerce transactions today involve mainly B2B and B2C transactions. Nevertheless, other categories of e-commerce are expected to come into trend in the near future.

2.2.4 Transactions Of E-Commerce

The purchasing transaction performed in e-commerce is a deliberative purchasing transaction, which in turn is defined as a succession of phases **[CNEC, 1997]**. These phases are as described below:

i) The Pre-Contractual Phase

During this stage, the buyer and the seller are concerned with the gathering of market intelligence. The buyer seeks information about suppliers of goods and services, about the goods and services themselves, and about the prices, availability, terms and conditions applicable to a purchase. The seller in turn seeks information about prospective purchasers of their goods and services.

ii) The Contractual Phase

During this phase, a formal relationship is established between buyer and seller, including terms and conditions to be applied to transactions under the contract. In certain cases, only a single transaction is envisaged, and hence these activities merge with those of the following phase. In other cases, many orders may be placed within the context of a simple "head" contract.

iii) The Ordering Phase

This phase involves the placement and processing of purchase orders. Other transactions, which may be involved, include the acknowledgement by the seller of its preferences to deliver, purchase order amendments, re-negotiations, and cancellations.

iv) The Logistics Phase

This phase deals with the delivery of goods and/or the performance of services. In addition, some post-delivery functions may be involved, in particular inspection, and acceptance or rejection.

v) The Settlement Phase

During this phase, the goods or services are paid for. Relevant transactions include invoicing, payment authorization, payment, and remittance advice transmission. Particularly in the case of contracts, which involve cyclical deliveries or service performance (temporary staff), a succesion of payments may be involved.

vi) The Post-Processing Phase

After the basic transaction has been completed, a number of additional activities may be undertaken. Most commonly, management information is gathered and reported. In some cases, an obligation may exist to store and report trading statistics to an industry association or national statistical authority. In addition, the sale may have resulted in a relationship between the buyer and the seller, relating to the servicing, maintenance, upgrading, and eventual replacement of the goods or asset.

2.2.5 Payment Methods Of E-Commerce

There are dozens, perhaps hundreds of businesses and organizations eager to assist merchants in selling their products online. Basically, the payment methods in e-commerce transactions fall into four categories [**Nightcats, 1999**]:

i) **Credit card transactions**

This method of payment will require the merchant to first have a merchant Internet account, which can be obtained through any regular banking institution. This method of payment is often the preferred way of payment as most consumers are already familiar with credit cards. With credit card transactions, consumers do not have to download and install special plugins. Merchants in turn will have the customers' contact information for follow up sales and marketing purposes.

ii) **Digital cash transactions**

Digital cash or electronic money is an arrangement whereby the customer pays for the merchandise using electronic "money". It is often used by consumers with no credit cards. Examples of this are the well known DigiCash, Cyberbucks, CyberCash, and so on. As consumers become more comfortable providing credit card information over the Internet, these methods are becoming less utilized.

iii) **Electronic Fund Transfers**

In Electronic Fund Transfers (EFT), funds are transferred electronically from the customer's bank account into the merchant's account. The best known method of doing this is through the issuing of electronic cheques. Customers pay for merchandise by writing an electronic cheque that is transmitted by email, facsimile or telephone. The "cheque" is a message that contains all of the information that is found on an ordinary cheque, but it is signed digitally, or indorsed. The digital signature is encoded by encrypting with the customer's secret key. Upon receipt, the merchant or "payee" may further indorse by encoding with a private key. When the "cheque" is processed, the resulting message is encoded with the bank's secret key, thus providing proof of payment.

iv) Telephone billing systems

A very new approach to payment methods over the Internet is the telephone billing system. Telephone transactions allow the customer to purchase an item or service, and the amount is billed to his or her telephone bill. To date, this is being used for the sale of soft items such as downloads, time measured services (e.g. time spent at a Web site) or for making charitable donations online. eCharge Corporation (http://www.echarge.com) is a pioneer in the use of this technology.

2.2.6 Incomplete Views Of E-Commerce

Although e-commerce has been around for quite some time in the market today, incomplete views of it still exist. Some popular misconceptions of e-commerce are:

- **E-commerce & E-business**

  Many people may have heard of the term **e-business**. Most of the time, e-business is often confused with the term e-commerce. Nevertheless, e-business and e-commerce is not the same thing. E-commerce refers mainly to the process of selling goods and services online, where else e-business covers a much more extensive perspective. As far as e-commerce is concerned, it is a subset of e-business, and therefore, e-business involves more than e-commerce. Basically, a business is an e-business when Internet technologies are used in the majority of business operations [**Reynolds, 2000**].

- **E-commerce & EDI**

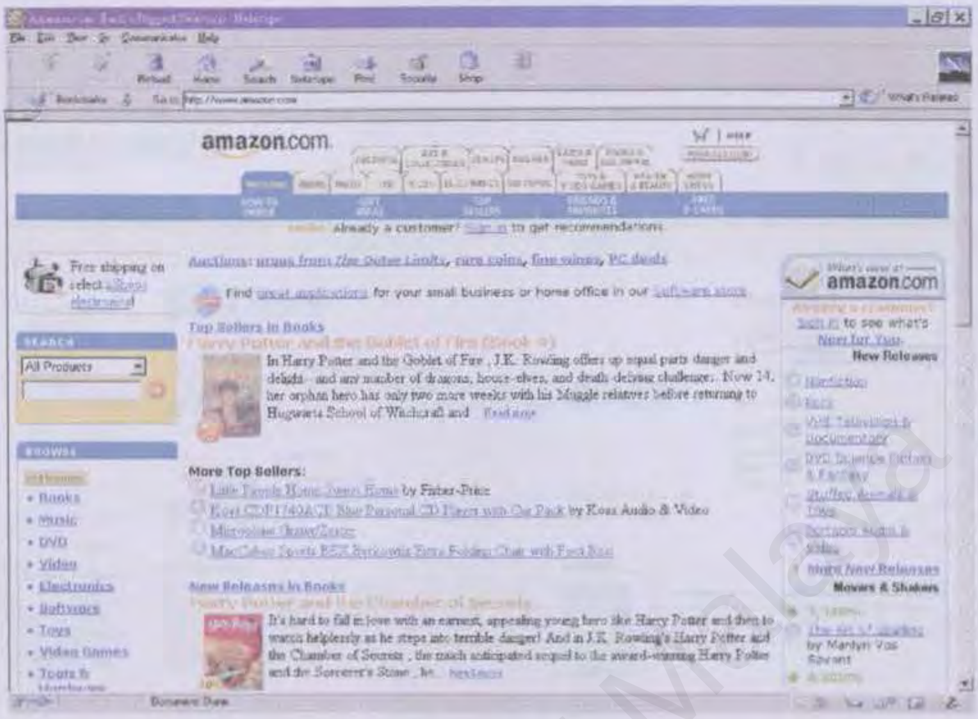
  E-Commerce can often be mistaken as **EDI** (Electronic Data Interchange). EDI is a core business process, which is defined as "the transfer of structured data using agreed message standards by electronic means between computer applications" [**CNEC, 1997**]. This may lead to considerable differences between these two conceptions, as some e-commerce applications may exchange unstructured information, while some others have need for structured information. Therefore EDI is only one of the components of an e-commerce implementation.

2.2.7 E-Commerce & Dotcoms

Over the years, e-commerce has succeeded in drawing together a huge number of entrepreneurs globally. Thousands of companies from all over the world are beginning to participate in the e-commerce gold rush, in hope of securing their fair share of the trade. Those who started early in this lucrative business are already starting to reap the fruits of labor by now.

- ***Amazon***

  One of the earlier pioneers of e-commerce is Amazon (http://www.amazon.com). Although Amazon has yet to make a profit, it is a major online brand. Starting as just a book reseller, it has used that brand to successfully push into music and video, and is now moving into auctions and other retail sectors, such as electronics and kitchen appliances. What made Amazon successful at the first place was its ability to pick up the right merchandise (books), specializing in it, and then market it intensively over the Internet. Amazon employs a systematic and structured Web layout, which makes every visit to its site a pleasant one. This can be clearly noticed by the way it categorizes its products into tabs on the top part of its main page and the search function it provides on every page. Everything is classified according to their categories, and shoppers could easily obtain what they wanted. New releases and top sellers decorate Amazon's main page, giving it a sense of excitement. This, together with its excellent marketing strategy, has contributed to Amazon's present accomplishment.



Already a customer? Sign in to get recommendations.

Free shipping on select eBook devices!

Find great applications for your small business or home office in our Software store.

Top Sellers in Books

Harry Potter and the Goblet of Fire (Book 4)

In Harry Potter and the Goblet of Fire , J.K. Rowling offers up equal parts danger and delight - and any number of dragons, house-elves, and death-defying challenges. Now 14, her orphan hero has only two more weeks with his Muggle relatives before returning to Hogwarts School of Witchcraft and ... Read more

More Top Sellers:

Little People Home Sweet Home by Fisher-Price

Koss CDP1740ACE Blue Personal CD Player with Car Pack by Koss Audio & Video

New Releases in Books

Harry Potter and the Chamber of Secrets

It's hard to fall in love with an earnest, appealing young hero like Harry Potter and then to watch helplessly as he steps into terrible danger! And in J.K. Rowling's Harry Potter and the Chamber of Secrets , the much anticipated sequel to the award-winning Harry Potter and the Sorcerer's Stone , he... Read more

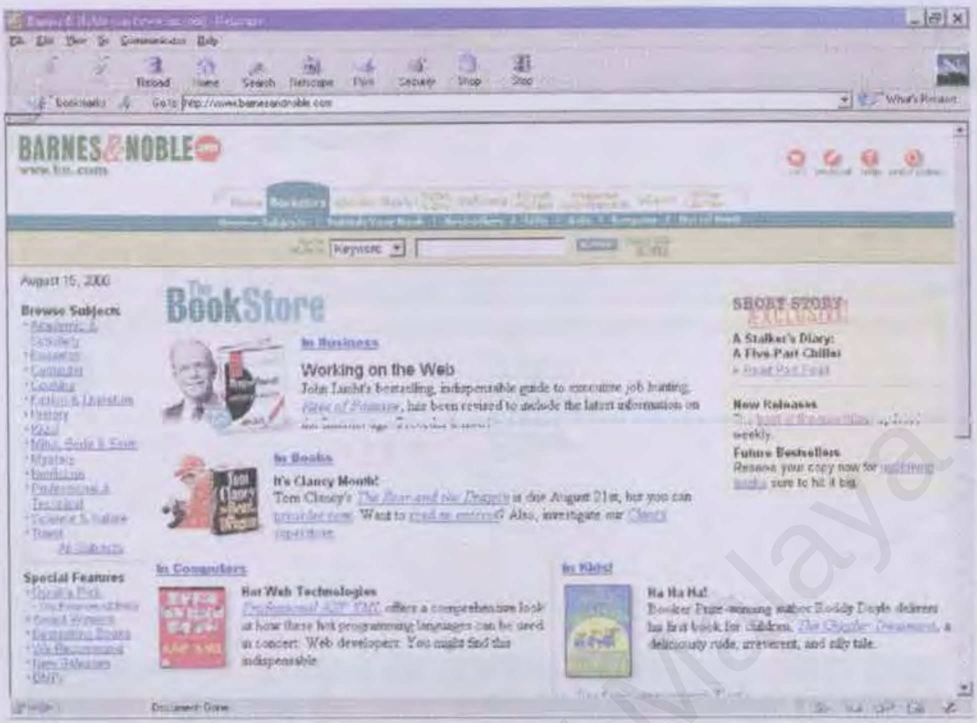
Sign in to see what's New for You.

by Marilyn Vos Savant

Figure 2-3: Amazon's online catalogue

- ***Barnes and Noble***

Another e-commerce Web site that is doing equally well is Barnes and Noble (http://www.barnesandnoble.com). Barnes and Noble is an online retailer of a wide variety of goods, with books being its key merchandise. Its merchandise include video, music, software, magazine subscriptions, and eBooks. Following in the footsteps of Amazon, Barnes and Noble too has been successful in its retailing business due to its excellent marketing strategy. The main page of Barnes and Noble is attractively designed, with the latest arrivals flooding its main page. The goods are carefully categorized into tabs according to their categories at the top part of its main page for easy reference. Just like Amazon, Barnes and Noble provides a search function on every page to facilitate shoppers in looking for the items that they desire in the shortest time possible.



In Business

Working on the Web

John Lucht's bestselling, indispensable guide to executive job hunting, *Rites of Passage*, has been revised to include the latest information on

In Books

It's Clancy Month!

Tom Clancy's *The Bear and the Dragon* is due August 21st, but you can preorder now. Want to read an excerpt? Also, investigate our Clancy superstore.

A Stalker's Diary: A Five-Part Chiller

weekly

Future Bestsellers

Reserve your copy now for upcoming books sure to hit it big.

In Computers

Hot Web Technologies

Professional ASP XML offers a comprehensive look at how these hot programming languages can be used in concert. Web developers: You might find this indispensable.

In Kids!

Ha Ha Ha!

Booker Prize-winning author Roddy Doyle delivers his first book for children, *The Giggler Treatment*, a deliciously rude, irreverent, and silly tale.

Figure 2-4: Barnes and Noble's online catalogue

Transition to dotcom companies

However, not all companies are eager to build their own e-commerce site. Some small and medium-sized companies (and even large-sized companies) believe it is more cost-effective to pay for the expertise offered by companies, called **dotcom** companies that specialize in Web site design and hosting. Thus, e-commerce is fast becoming accessible to companies, big or small, from all over the world.

- ***Iconomy.com***

  An example of a dotcom company offering Web hosting services is Iconomy.com (http://www.iconomy.com). Formerly known as BuySafe.com, Iconomy.com focuses on creating private-label e-commerce storefronts. The concept behind private-label stores is that Iconomy.com provides sites with all of the benefits of online shopping and none of the infrastructure commitments. The typical private label store displays the client's brand name and adopts the look and feel of the client's Web site. Iconomy.com can handle all aspects of Web site development. Their services range from initial site design to shopping carts, credit-card authorizations, and inventory management. Iconomy.com can also manage order fulfillment, handling the actual sale and shipping workload, acting as a middleman between distributors and vendors.

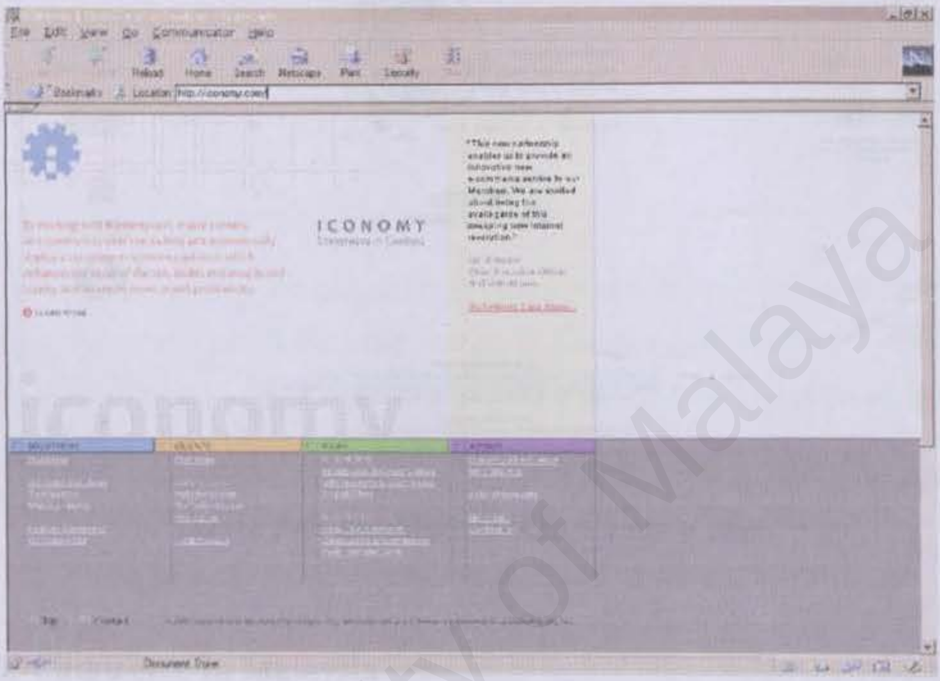


Figure 2-5: Iconomy.com's main page

- ***FreeMerchant.com***

  FreeMerchant.com (http://www.freemerchant.com) is another dotcom company offering Web hosting services. Unlike any other dotcom companies, FreeMerchant.com provides Web hosting services for free. In order to cover the expenses of its services, FreeMerchant.com acquires business partners who pay a fee to put up their advertisements on merchant pages. FreeMerchant.com offers basic site-building capabilities and a well-designed interface, which makes it all too easy to host a Web site. It provides merchants with services like storefront builders, order checking, catalogue maintenance, community building, sales lead generations, and much more.

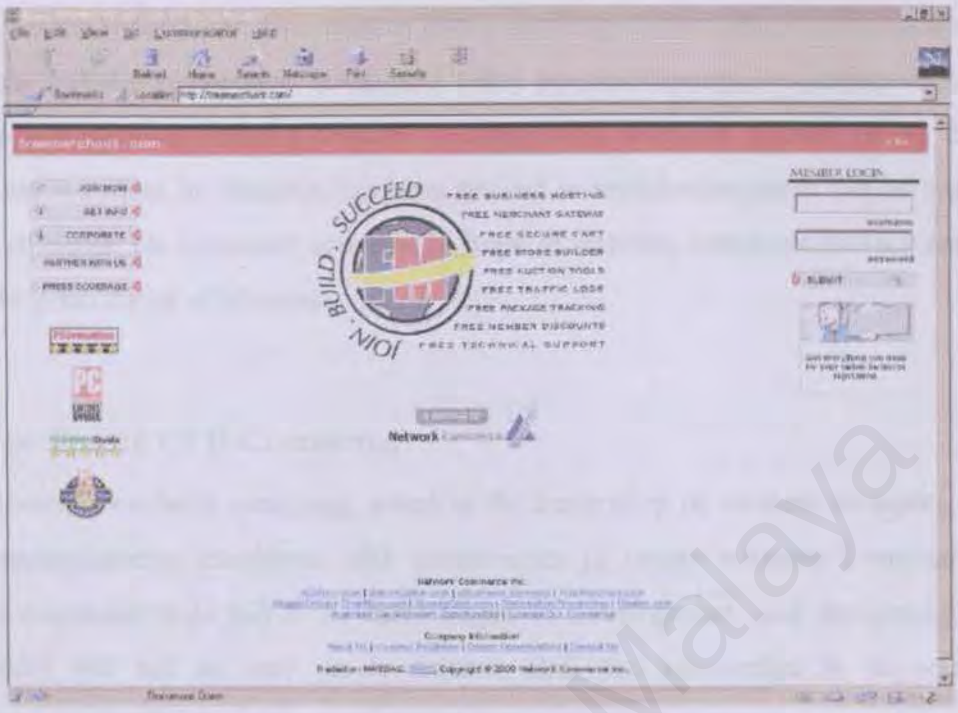


Figure 2-6: FreeMerchant.com's main page

2.2.8 E-Commerce In Malaysia

Although e-commerce has flourished in developed countries such as the United States and the United Kingdom, the same cannot be said for our homeland. E-commerce is still at its teething stage in Malaysia. The number of Malaysian companies involved in e-commerce is still not too encouraging, with barely a handful of Web sites that can really be considered as e-commerce Web sites at all. This clearly shows that e-commerce has not really caught on in Malaysia.

According to the Energy, Communications and Multimedia Minister, Datuk Amar Leo Moggie, global figures showed that by 2003, Web-generated revenue was expected to increase to US$1.23 trillion (RM4.67 trillion), while in Malaysia, the e-commerce market was expected to generate revenues exceeding US$1 billion (RM3.8 billion) by the same year **[Chua, 2000]**.

As such, the Malaysian government has strived to provide all the possible means of encouraging the usage of the Internet as a powerful business tool. The Multimedia

Super Corridor (MSC), the e-commerce and k-economy (knowledge economy) master plans, low Internet connection costs, and cyber laws are among the initiatives taken by the government to further promote e-commerce in Malaysia **[Chua, 2000]**. Even the education system in Malaysia has been revised to include computer related studies into the students' co-curricular activities in hope of instilling computer literacy among the future generations of Malaysia.

2.2.9 The Future Of E-Commerce

A new business model is emerging, which is the integration of wireless networks with data communications, combined with e-commerce to create **wireless e-commerce**. Wireless e-commerce is still in its early stages of development, and developing this new model will not be easy, as it requires skill and knowledge in the various telecommunication fields.

Nevertheless, wireless e-commerce is inevitable, and it will generate significant revenues within the next several years. Wireless communications and e-commerce already are multi-billion dollar global businesses, and the integration of mobile communications with e-commerce has already started. In fact, for years companies in the vertical markets, such as field repair, have been utilizing mobile communications networks to enable their technicians to order parts, check inventories and, in essence, conduct wireless e-commerce **[WIMC, 1999]**. The return on investment for many of these companies has so far been dramatic.

With the advent of wireless e-commerce, comes a new technology, called the **Wireless Application Protocol** (WAP). WAP is an open, global specification that empowers mobile users of wireless devices to easily access live interactive information services and applications from the screens of their mobile phones **[WAPF, 2000]**. It is a new advanced intelligent messaging service for digital mobile phones and other mobile terminals that will allow users to see Internet content in special text format on special WAP-enabled GSM mobile phones. Services and applications available with WAP include email, customer care, call management, unified messaging, weather and traffic alerts, news, sports and information services, electronic commerce transactions and banking services, online address book and directory services, as well as corporate intranet applications.

WAP utilizes HTTP 1.1 Web servers to provide content on the Internet or intranets, thereby leveraging existing application development methodologies and developer skill sets such as CGI, ASP, NSAPI, JAVA and Servlets. WAP defines an XML (eXtensible Markup Language) syntax called WML (Wireless Markup Language). All WML content is accessed over the Internet using standard HTTP 1.1 requests.

It is foreseen in the near future that users will be able to fully access the Internet just by using their mobile phones. In addition to the rapid development of wireless communication technologies, such as WAP, wireless e-commerce is destined to be the next best thing to hit the e-commerce world.

2.3 Web Security

2.3.1 Security Issues

As the Internet and the World Wide Web expand into a massive network of information, users are increasingly tempted to delinquency. Furthermore, lack of proper guidelines in cyber laws has encouraged unscrupulous acts by users all over the Internet. Among the main security concerns of the Internet today are:

- **Hacking**

  A computer hacker is a person who attempts to gain unauthorized entry to a computer system by circumventing the system's access control **[Weber, 1999]**. Hackers can have benign intent. They simply might explore the system they hacked or read files without changing them. Alternatively, hackers can also have malignant intent. They could wreak havoc by deleting critical files, disrupting system operations, or stealing sensitive data and programs. Hackers are often difficult to trace because they can operate from any part of the world, all with the help of the Internet. Organizations can only try to prevent hacking by putting up firewalls and other security measures on their system. Still, this does not guarantee a hack-proof system.

- **Virus attacks**

  A virus is a program that requests the operating system of a computer to append it to other programs **[Weber, 1999]**. In this way, the virus propagates to other programs. Viruses can easily be transmitted via files over the Internet through electronic mails (e-mails). With the recent attack of the "I-Love-You" virus over the Internet, virus protection has become a main concern for many organizations that run their businesses online.

Another Web security concern is the increasing number of frauds over the Internet. E-commerce has inevitably resulted in the higher usage of credit cards in order to complete transactions online. This in turn has lead to the forging of credit card numbers by certain unscrupulous individuals, which resulted in the execution of unauthorized transactions. Although steps have been taken to avoid such immoral actions, including the usage of security protocols and encryption on Web site transactions, as well as the implementation of stricter cyber laws, frauds still continue to lurk within the Internet society.

2.3.2 Security Protocols

There are mainly two security protocols usually implemented on E-Commerce Web sites. They are SSL and SET:

i) ***Secure Socket Layer (SSL)***

Secure Socket Layer (SSL) is a security protocol that uses public-key encryption and digital certificates to set up the interaction and verify that the parties are who they say they are. It uses special session keys to encrypt the data being transmitted. The digital certificates (issued by a certificate authority) are used to verify that the key pairs belong to a particular entity. Session keys perform the cryptographic work for the data exchange.

However, SSL has a few weaknesses. Among them are:

- **Shallow encryption**

  SSL can use only relatively shallow encryption (40-bit internationally, 128-bit in the United States), due to certain restrictions by the Department of States in the United States.
- **Only point-to-point transactions**

  SSL handles only point-to-point interaction. Credit card transactions involve at least three parties: the consumer, the merchant, and the card issuer. This makes SSL not suitable for credit card transactions.
- **High transaction risks**

  With SSL, consumers run the risk that a merchant may expose their credit card numbers on its server, and merchants run the risk that a consumer's card number is fraudulent or that the credit card will not be approved.

ii) Secured Electronic Transaction (SET)

When a customer wishes to make a purchase, the order information is encrypted via the customer's private encryption key and sent to the merchant, while the credit card information is also encrypted and sent to the card issuer, all accompanied by a unique digital signature. The merchant and card issuer decrypt the information using the customer's public key, allowing them to verify its authenticity and complete the transaction.

Like SSL, SET too has its weaknesses, such as:

- **Lack of testing**

  Interoperability among SET implementations is only now being tested.
- **Slow adoption**

  Consumers may be slow to implement electronic wallets.

SSL vs. SET

Between the two security protocols discussed above, SET has more advantages as compared to SSL. SET requires parties, namely the cardholder, the merchant, bank, and anyone else involved, to obtain a digital certificate. In addition, it requires a certificate authority to authenticate all parties in the transaction. SET lets customers keep credit card information in software called an "electronic wallet" on their computers. This gives merchants no access to credit card information, making SET safer than in-person or phone transactions. It also conceals order information from the credit card issuer, maintaining the customer's privacy. SET also uses a higher degree of encryption for transactions, which makes it more difficult for frauds to take place [**CommNett, 2000**].

In Malaysia, a few major banks are already starting to implement the SET technology into their online credit card payment system [**Muhtar, 2000**]. Under this system, dubbed the “MEPS' SET Payment Gateway”, credit card holders will have to obtain an electronic wallet from the issuing bank, as well as a digital certificate from a certified certificate authority (e.g. DigiCert). Once the electronic wallet has been installed on the cardholder's computer, the cardholder can then download a personalized digital certificate for the card and immediately begin transactions online.

However, SET is still at its testing stages, and consumers are yet to adapt to the electronic wallets. As such, SSL remains the preferred security protocol on the Web. Furthermore, SSL is less complex than SET, making it much more suitable for Web sites requiring low to moderate levels of security.

2.4 Web Technologies

2.4.1 Client/Server Architecture

Client/server computing is a cooperative processing of requests submitted by a client to a server, which processes the requests and returns the results to the client [**Gadey, 1996**]. It is really a form of distributed processing where resources are spread across two or more computing systems. Clients and servers are separate logical entities that work together over a network to accomplish a computing task [**Gadey, 1996**].

Client/server computing facilitates the use of GUI that is available on workstations. The visual presentation in turn increases the productivity of the end user, as it is very easy to use. Client/server environment exploits the power of the workstations and due to the fact that client and servers might run on different software and hardware platforms, it encourages the acceptance of open systems.

There are mainly two types of client/server architecture in use today, namely the two-tier client/server architecture and the three-tier client/server architecture.

i) Two-Tier Client/Server Architecture

A two-tier client/server environment consists of the user interface on the first tier (client), and the database on the second tier (server). The application logic is either embedded in the user interface on the client, or within the database on the server. Figure 2-7 illustrates a typical two-tier client/server architecture:

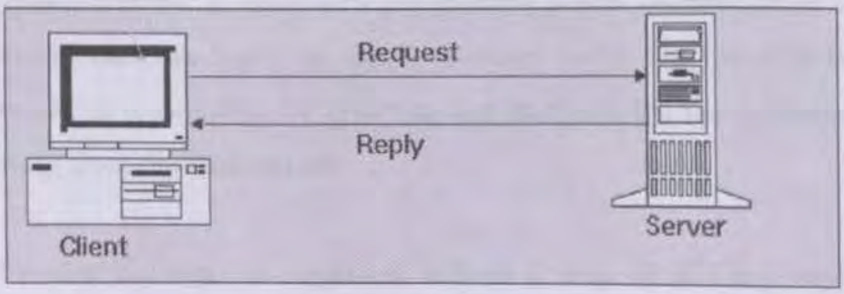


Figure 2-7: Two-tier client/server environment

In a two-tier architecture, the client directly interacts with the database. Tasks are divided between the server and the clients. The server has the responsibility of storing, manipulating and delivering data to the client.

ii) Three-Tier Client/Server Architecture

In a three-tier client/server environment there are three tiers as the name indicates. The first tier consists of user interface on the client, the second tier or the middle tier consists of business or application logic, and the final tier handles the data (database). Figure 2-8 illustrates a typical three-tier client/server architecture:

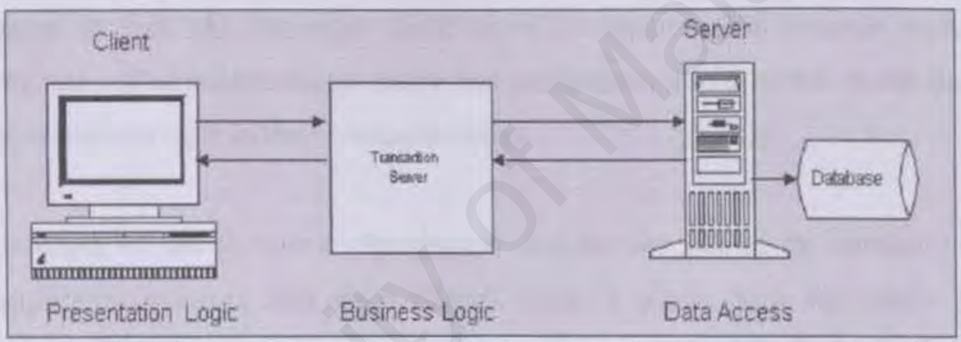


Figure 2-8: Three-tier client/server environment

The presentation tier is responsible for providing the services the application needs to allow the users to view and manipulate the application. In other words, it presents a user interface to the user. The business tier is responsible for supplying a set of procedures and rules to follow while accomplishing any of the system's tasks. It states how the business should operate under a set of rules, called business logic, or simply business rules. On the contrary, the database tier is responsible for querying and manipulating the database under instructions from the business tier.

The three-tier client/server approach defines a way of dividing application services into three distinct roles (concerning presentation, business rules, and data), which are arranged so that each can operate at the maximum efficiency.

As with all three-tier developments, the presentation tier never directly interacts with the database tier. This is deliberately done because of the importance placed on the business tier.

Two-tier vs. three-tier client/server architecture

Presently, an increasing number of companies on the World Wide Web are switching to the robust three-tier client/server architecture in their business model. Although the three-tier client/server architecture is more complex than the two-tier client/server architecture, it is still far more efficient, reliable, and systematic than the usual two-tier model. In the beginning, it may be costly and time-consuming to implement the three-tier architecture, but then again, its long-term benefits greatly outweigh its disadvantages. In fact, the three-tier client/server architecture has become such a necessity for the online businesses of today that companies would be left in the dark just for not implementing it in their business model.

The key strength of the three-tier client/server architecture lies in its capability of dividing application services into three distinct roles. It places high importance on business logic, which is essential in the development of any businesses. As such, the three-tier client/server architecture is extremely suitable for the e-commerce Web sites of today.

2.4.2 CGI vs. ASP

CGI

The **Common Gateway Interface** (CGI) is a standard for interfacing external applications with information servers, such as HTTP or Web servers [**NCSA, 1996**]. A plain HTML document that the Web daemon retrieves is static, which means it exists in a constant state: a text file that does not change. A CGI program, on the other hand, is executed in real-time, so that it can output dynamic information.

Any script can be called a CGI script as long as it's installed on the server end [**Shimar, 2000**]. The majority of CGI scripts are written in Perl, with C/C++ being the next most common. CGI is installed on the server end that makes it able to do things such as submit a form, create a guest book or forum, keep track of and rotate advertisements, and much more.

ASP

With the advancement of CGI and its ability to output dynamic information, comes another similar technology called ASP. ASP or **Active Server Pages** is a server-side framework that lets users create Web applications [**ASC, 2000**]. Developed by Microsoft Corporation, ASP extends standard HTML by adding built-in objects and server-side scripting. It also allows access to databases and other server-side ActiveX components. ASP is interpreted at run time by the Active Scripting engine of Microsoft Internet Information Server (IIS). In reality, ASP files are really just HTML files with scripting embedded within them, such as Visual Basic Scripting Edition (VBScript), JavaScript, and PerlScript. However, the default scripting language for ASP files is VBScript.

Figure 2-9 illustrates how the Web server interprets ASP files:

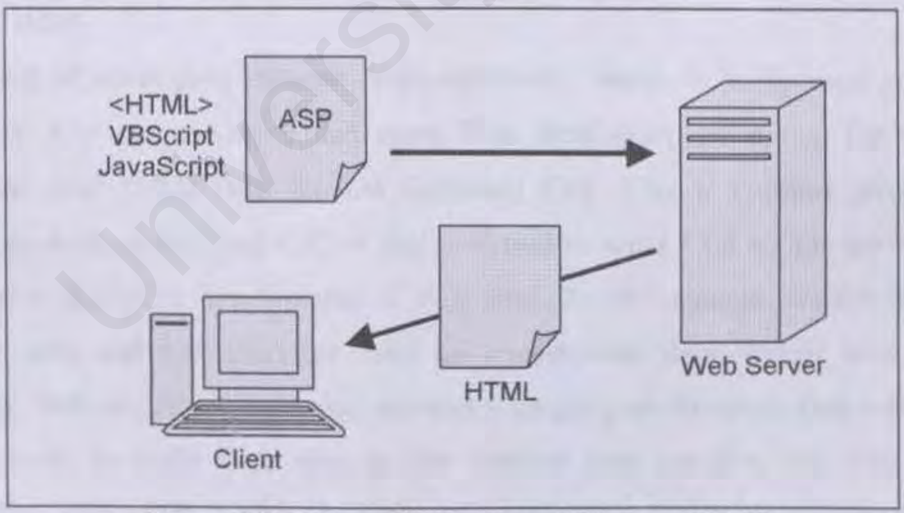


Figure 2-9: How a Web server handles ASP files

When a browser requests an ASP file from the server, it is passed on to the ASP processing DLL (Dynamic Link Libraries) for execution. After processing, the resulting file is then sent on to the requesting browser. The magic of ASP is that it does not matter which browser is used to view the ASP file, because the server returns only pure HTML. Any scripting commands embedded from the original HTML file are executed and then removed from the results, which is excellent for the Web developer because all the scripting code is hidden from the person viewing the Web page. This phenomenon can be explained by the ".asp" extension that is used by all ASP files.

ASP pages have an ".asp" extension instead of the usual ".html" and ".htm" that most HTML files have. It is imperative that ASP files carry the ".asp" extension for two reasons [**Reynolds, 2000**]:

i) In order to let the Web server know to process the scripting in the Web page, it needs to know that there is some in there. By setting the extension of the Web page to ".asp", the server will assume that there are scripts in the page.

ii) Using an ".asp" extension forces interpretation by the ASP processor every time the Web page is requested, thus hiding the ASP scripts on the page and maintaining secrecy of the source codes used in the ASP file.

CGI vs. ASP

The trend of developing dynamic Web sites today seems to be focused around the usage of ASP scripts. More and more Web developers are opting for the more advanced ASP rather than the old fashioned CGI. This is because programming languages such as Perl and C/C++ that are used to write CGI scripts are no longer practical in the quick development of Web sites. These languages require time to be familiar with and will therefore take up considerable development time. On the contrary, Web developers of today demand a scripting environment that will not only enable them to build Web sites in the shortest time possible, but also to build interactive Web pages as well. As such, ASP is the right choice for creating successful e-commerce Web sites.

2.4.3 Objects vs. Components

Object-oriented programming (OOP) is based on the idea that real-world entities or relationships can be represented in code as objects. These code objects have associated with them data and behave in certain ways when asked to. Objects can be linked together to form programs and applications. The strength of objects lies in its ability to be reused over and over again.

However, objects are created from classes, and classes are made up of source codes, which are language-specific. Thus, objects can only be used in one environment. Building on the idea of object-oriented programming, comes component-oriented design, which facilitates even better reuse of objects. Components are precompiled units of binary code, thus they are language-independent. A component may consist of one object or a collection of objects.

COM

The **Component Object Model** (COM) is Microsoft's standard for allowing objects and components to interact irrespective of the language in which the components were first built. It is a binary programming standard [**Varadarajan, 2000**]. By using COM technology, Web developers can call up a component, and providing they know what interfaces it has, they can get it to go away and do things for them. Up to date, COM has become a crucial part of the software infrastructure in the Windows 95 and Windows NT environments.

ActiveX

ActiveX encompasses a set of technologies that each defines the interfaces between software components to implement some type of functionality. It is built based on Microsoft's COM technology [**Chappell & Linthicum, 1997**]. An **ActiveX component**, on the other hand, is an application that stands alone and lets other applications use the classes and objects it contains. ActiveX components are often implemented as **ActiveX DLLs** (Dynamic Link Libraries). DLLs do not have their own interfaces and depend on other applications to call on it. Several programming tools may be used to build ActiveX DLLs, with **Microsoft Visual Basic 6.0** being the preferred choice by developers all around the world.

Active Scripting

ASP pages are able to access ActiveX components through a technology called Active Scripting. Active Scripting is a reusable scripting engine that is capable of supporting many languages by allowing developers to write supporting language plug-ins. It can be found in use in ASP files and **Windows Script Host** (WSH). Active Scripting is shipped with VBScript (a cut down version of Visual Basic) and JScript.

2.4.4 Microsoft Universal Data Access Technology

Universal Data Access is Microsoft's strategy for providing high-performance access to all types of information, including relational and non-relational data [**Sim, 1999**]. Microsoft Data Access Components consists of **Open Database Connectivity** (ODBC), **OLE DB**, and the **ActiveX Data Objects** (ADO).

ODBC, OLE DB & ADO

ODBC is an API (Application Programming Interface), a set of standards defining a library of function calls that can be used in programming languages like C to access databases on remote servers. Originating from Microsoft, ODBC has become the recognized standard for providing database technology consumers. For database development companies such as Microsoft, IBM, Oracle, Sybase, Informix, and others, ODBC has given programmers a way to directly access their database technology through a consistent, common interface.

OLE DB, on the other hand, is an open specification designed to build on the success of ODBC by providing an open standard for accessing all kinds of data. OLE DB is based on Microsoft's own COM technology. However, OLE DB has a more difficult to use interface than ODBC. It is more difficult to learn, as it has more parameters, more options, more structures, more properties, and more settings. In order to solve this problem, Microsoft came up with ADO, which hides the complexities of OLE DB and provides a kinder and gentler way of programming.

ADO is a set of objects that can connect to any ODBC or OLE DB database [**Reynolds, 2000**]. The benefits of ADO include ease of use, high speed processing, low memory overhead, and a small disk footprint.

DAO & RDO

Before ADO, Microsoft had earlier released two other products that achieved the same goal: **Data Access Objects** (DAO) and **Remote Data Objects** (RDO). DAO was the first of the breed of data access technology. Just like ADO, DAO introduced an object model complete with a hierarchy of programmable objects, which included recordsets, fields, and properties. In fact, DAO is still in use today in the Microsoft Office 2000 suite. The difference between DAO and ADO is that DAO was designed specifically to provide direct access to Microsoft Access's underlying database technology, known as the JET database engine [**Holzner, 1998**]. ADO, on the other hand, was not designed with any particular database in mind. It has a more general design, so it complies not only with programming models designed around proprietary databases, but also with those designed around open standards, such as ODBC.

Although ODBC was considered a good interface for supplying data, it left a lot to be desired for when used as a programming interface. To solve this problem, Microsoft invented another technology parallel to the already existing DAO object model: RDO. RDO was necessary because the DAO model was not a clean match to ODBC. However, RDO does have its own equivalents to DAO objects. Rooted in ODBC terminology, RDO uses Resultsets instead of Recordsets (as used by DAO), and Statements instead of QueryDefs.

With the advancement of newer ADO technologies, there is really no more need for DAO or RDO. ADO combines the technology of both DAO and RDO into a single environment, making it appropriate for the development of projects involving complex data manipulations.

CHAPTER 3: SYSTEM ANALYSIS

3.1 Introduction

System analysis is the most important phase in a software life cycle. It is the process of defining a problem, gathering pertinent information, developing alternative solutions and choosing among those solutions.

Among the techniques used for analysis are:

- **Bookstores**

  Several renowned bookstores, such as MPH and Popular, contain relevant references about this project. Suitable reference books have been purchased in order to learn in detail about the technologies used in this project.

- **Document Room (Bilik Dokumen)**

  Previous seniors' theses have been read through in order to gain an overall understanding of the purpose of a thesis. The general structure of each thesis has also been observed to find out the steps taken in carrying out a thesis.

- **Internet surfing**

  The Internet has been the main source of information for this project. Trusted Web sites have been visited and researches of Web authors have been evaluated to obtain reliable information for analysis.

- **Library**

  Books, journals, and magazines from the library have been read through and valuable information has been noted down.

- **Newspapers**

  Articles from daily newspapers about the related field of study for this project have been collected for further references.

- **Software Testing**

  Relevant software and Web development tools have been tested out to evaluate their suitability for this project.

3.2 System Requirements

A requirement is a feature of the system or a description of something the system is capable of doing in order to fulfill the system's purpose [**Pfleeger, 1998**]. Requirements describe not only the flow of information to and from a system and the transformation of data by the system, but also the constraints on the system's performance.

In order to ensure the developers and the customers understand and use the requirements properly, it is important that the requirements are clearly defined. Thus, the requirements have the following characteristic: describe something that is needed by the customer, correct, consistent, complete, realistic, verifiable, and traceable [**Pfleeger, 1998**].

System requirements can be functional or non-functional.

3.2.1 Functional Requirements

A **functional requirement** is a statement of the services or functions that a system should provide, how the system should react to particular inputs, and how the system should behave in particular situations [**Sommerville, 1998**].

Generally, E-Pack should provide Web hosting services to small retailers at a nominal subscription fee. On confirmed membership, Web pages of each retailer will be published. These pages will include the products or services that are offered by each retailer.

E-Pack contains two main modules: the shopper's module and the administrator's module. The functions that are performed by each module of E-Pack are as follows:

i) Shopper's Module

- ***Online Catalogue***

  The online catalogue is used to exhibit the available products that each retailer has for sale. It will include pictures of the product, its category, its description, and its price. The catalogue will save the shoppers' time in choosing their goods, as it provides all the information that the shoppers need, just at a click of the mouse. Since all product information is already available online, the shoppers need not leave their homes simply to do their shopping. All they have to do is browse through the Web site from the comfort of their own homes, thus allowing them time to perform other household chores.

- ***Shopping Cart***

  The shopping cart is a temporary space where the shoppers can place their items after selecting it from the online catalogue. It works exactly like the shopping carts or baskets in any supermarkets, where the shoppers can place their goods in it and bring it around the supermarket before purchasing it. Whenever a shopper feels that an item in the shopping cart is no longer needed, all the shopper needs to do is select that item, then click a button to remove that item from the shopping cart. However, the items in the shopping cart are not finalized for purchase.

- ***Checkout Counter***

  Like supermarkets, E-Pack too has checkout counters where the shoppers can finalize their purchases and make payments for those purchases. The checkout counter will display the total amount to pay for the items purchased. The shoppers will then be asked to enter their personal information and state their method of payment. However, the checkout counters in E-Pack are more efficient than the checkout counters in supermarkets in the sense that, the checkout counter of E-Pack is managed by programs, thus more efficient than the checkout counters in supermarkets, which are manned by human personnel. The use of human personnel leads to human errors in calculations, which in turn leads to erroneous transactions.

ii) Administrator's Module

- *Customer Records*

  In order to keep records about the subscribers of E-Pack, the administrators of E-Pack need an interface, which allows them to gain access to the customer records. The customer records interface will allow them to login to the database to add, edit, or delete the particulars of the subscribers conveniently using a Web browser.

- *Shoppers' Orders*

  There will also be an interface where administrators can manage the orders made online by the shoppers. Just like the customer records interface, the shoppers' orders interface will allow administrators to login to the database to add, edit, or delete shoppers' orders through a Web browser.

3.2.2 Non-functional Requirements

A **non-functional requirement** can be defined as the constraints on the services or functions offered by the system **[Sommerville, 1998]**. The non-functional requirements for E-Pack are:

- **Scalability**

  The system should be capable of adapting itself into environments with large numbers of simultaneous users. It should cope with sudden increases in Web site traffic.
- **Portability**

  The system should be designed to cope with migrations from one platform to another. It should not be much of a hassle to perform migration tasks.
- **Modularity**

  The system should be broken down to modules of functions so that it will be easier for testing and maintenance. Modularity also facilitates future modifications, which leads to further enhancements.

- **Security**

  Security measures should be implemented to prohibit unauthorized access to the administrator's module so that crucial information in the database will not be tempered with.

- **Maintainability**

  The system should be designed in such a way that it can be easily understood and corrected, as well as provides the means for future enhancements. It is of no benefit to develop a system that has no potential for improvements.

- **User-friendliness**

  Consistency in terms of screen design and error displays should always be practiced to minimize ambiguity. The system must be able to accommodate a variety of user levels, as not all users have established technical backgrounds. It should also provide easy and understandable commands and buttons to avoid blurriness. Documentation is a must.

- **Reliability**

  The system should perform all its services and functions accurately, and in a timely manner. It should also be robust in design and reliable so that users are convinced in using the system.

3.3 Consideration Of Operating Systems

3.3.1 Microsoft Windows NT 4.0

Windows NT 4.0 is an operating system that needs no further introduction. With an estimate of between two million to four million computers installed with the Windows NT platform all over the world, it could easily be the most popular operating system in the world [**Hayman, 2000**].

Just like its desktop siblings (Windows 95 and Windows 98), Windows NT is easy to use, fairly quick to learn, and has some great software bundled with it. It also comes with a colorful interface, with plenty of icons to assist users in navigating themselves through the system. There is no need for users to memorize difficult commands, as almost everything or anything on Windows NT is done by the simple clicks of the mouse. This, by far, makes Windows NT (and also Windows 95/98) the most user-friendly operating system available in the world today.

Windows NT is also preferred above many other operating systems because of its ability to inter-operate with a variety of enterprise level software available in the market today. In other words, there are much more software designed to run specifically on the Windows platform as compared to any other operating systems in the world today. NT's improved system architecture also makes it far more superior than its desktop siblings, Windows 95/98, which makes it a better choice for business applications.

3.3.2 Linux

Linux is a complete operating system that is similar, but not identical to UNIX. It was initially created by a young student, Linus Torvalds, at the University of Helsinki in Finland as a hobby. Originally the term Linux referred strictly to the kernel (the core of the operating system), but the phrase itself refers nowadays to a collection of configured software that runs on top of the Linux kernel.

The source code for Linux is freely available to everyone. Freely available in this context means the source code for the kernel and most software cannot be withheld **[Linux, 2000]**. This however, does not mean that Linux and its assorted distributions are free. Companies and developers may charge money for it as long as the source code remains available. Currently, there are various versions of Linux distributed in the world today. Among the companies involved in the distribution of Linux are Red Hat, Caldera, S.U.S.E, and Stampede. These companies do not just compile and configure the software. Caldera, Red Hat and others have added their own proprietary software to their distribution.

Linux is used for a wide variety of purposes including networking, software development, and as an end-user platform. Linux is often considered as an excellent, low-cost alternative to other more expensive operating systems. It is stable in the sense that, crashes of applications in Linux are much less likely to bring down the entire operating system as compared to Windows. Linux servers are also reliable, as they are often up for hundreds of days before requiring any reboots.

Finally, Linux allows true multi-tasking (the ability to run more than one program at the same time), and comes with an excellent window system called X, which is the equivalent of Windows, but much more flexible. This feature makes Linux no harder to configure than its Windows counterpart.

3.3.3 Conclusion Of Operating Systems

It is undeniable that Linux is considerably less expensive to run as compared to Windows NT. Most of the software available for Linux is free, not to mention Linux itself is also free, which makes it suitable for low-cost projects. Linux is much more reliable then Windows NT and hence is the better solution when running mission critical operations.

However, Windows NT still has a lead over Linux when it comes to ease-of-use. Linux is relatively difficult to configure, and learning up its UNIX-based commands may take up a lot of precious time. In other words, Windows NT has a shorter learning curve as compared to Linux. Furthermore, Windows NT has much more enterprise level software available for it as compared to Linux, which may prove to be essential in developments that require the usage of other software. Windows NT also makes administration tasks easier with its easy-to-use wizards and its complete set of help files.

As such, Windows NT 4.0 will be chosen as the platform to develop E-Pack due to familiarity of use, ease-of-use, and its compatibility with other software.

3.4 Consideration Of Web Authoring Tools

3.4.1 Microsoft Visual InterDev 6.0

Visual InterDev is an integrated development environment (IDE) for creating dynamic Web sites [**Lam, 1997**]. It is part of the Microsoft Visual Studio family, which consists of Visual Basic, Visual C++, Visual FoxPro, and Visual J++. Visual InterDev combines a number of tools into a single package to simplify development chores. There is support for project management, syntax highlighting of different types of Web files such as ASP or HTML, automatic synchronization of local files with Web server files, and integrated database access.

With Visual InterDev, a developer can assemble pages that use Microsoft's ActiveX technologies and other similar COM technologies. Data-driven Web applications can also be developed using Microsoft's Universal Data Access, which includes ADO, ODBC, and OLE DB. Visual InterDev also provides a robust development environment with a Scripting Object Model, design-time controls (DTCs), and an extensible toolbox for the purpose of rapid design, testing, and debugging of Web pages. Web teams can now develop pages in isolation and maintain ready access to a master version. In short, Visual InterDev is an excellent all-rounder IDE, both for programmers and non-programmers.

3.4.2 Microsoft FrontPage 2000

FrontPage 2000 is part of the Microsoft Office 2000 family. It runs on the Windows 95/98/NT/2000 operating system, and is one of the simplest and least complex HTML editors available in the market today. Because FrontPage 2000 is integrated into the Office 2000 package, it interacts easily with other components of Office 2000, such as Word, Excel, Access, and PowerPoint.

FrontPage 2000 is aimed at letting non-programmers build no-nonsense pages for their Web sites quickly and easily. It features built-in tools to facilitate the creation of forms, tables, banners, Java applets, and much more. Developers, writers, and designers may also work on the same Web project by using Microsoft FrontPage and Microsoft Visual InterDev in conjunction. In this case, Visual InterDev provides developers with a robust set of tools for developing Web applications, while FrontPage provides a WYSIWYG environment for editing pages that does not require any programming knowledge.

3.4.3 Allaire ColdFusion 4.0

ColdFusion is a complete development and deployment environment ideally suited for building today's Web-based e-commerce, content management, and business-automation initiatives. It is an ideal choice for workgroup, departmental, and extranet applications, in which fast deployment times are critical, often with the work done by less-experienced developers.

ColdFusion applications are essentially collections of pages. As with static Web pages, ColdFusion pages can contain HTML and other client technologies, such as JavaScript or PerlScript. Unlike static pages, ColdFusion application pages are denoted by a specific ".cfm" extension, which is the default ColdFusion file extension. ColdFusion pages also contain an additional language, called the ColdFusion Markup Language (CFML).

The ColdFusion development platform consists of two basic components:

i) **ColdFusion Server**

The ColdFusion Server is a high-performance Web application server for deploying browser-based applications. It offers all the runtime services for delivering e-business applications built on a highly scalable and open architecture. Because CFML is processed on the server, and because Web servers only know how to pass pages to browsers, the ColdFusion Server must be installed on a Web server to provide support for ColdFusion applications.

When the ColdFusion Server is installed on a Web server and a client requests a page with a ".cfm" extension, the following steps take place:

1. The Web server passes files to ColdFusion Server
2. ColdFusion Server scans the page and processes all CFML tags
3. ColdFusion Server returns only HTML and other client-side technologies to the Web server and, in turn, the browser.

ii) ColdFusion Studio

ColdFusion Studio is an integrated development environment (IDE) that includes a variety of tools for building Web applications. It provides a full suite of advanced editing tools including color coding, Web application wizards, and two-way visual programming. ColdFusion Studio also includes visual database tools, an interactive debugger, full integration with source control systems, and support for remote and team development using the ColdFusion Server.

Using the ColdFusion development platform, developers can build Web applications that connect to a wide range of existing business systems, such as RDBMS, messaging servers, file repositories, directory servers, and distributed object middleware. All the above makes ColdFusion an excellent tool for heavy-duty Web development.

3.4.4 Macromedia Dreamweaver 3.0

Macromedia Dreamweaver is a professional visual HTML editor for creating and managing Web sites and pages. It gives developers the productivity of a visual Web page layout tool, the control of an HTML text editor, and support for new Web technologies, all in one software package.

Developers can use Dreamweaver to create Web sites visually, with confidence that the HTML being generated is concise and always editable. Dreamweaver includes advanced features that take advantage of the latest innovations on the Web, such as

Dynamic HTML and Cascading Style Sheets, while still ensuring that Web pages work well in a variety of Web browsers. All of the code generated by Dreamweaver is carefully created to work on as many platforms and browsers as possible.

Other features of Dreamweaver include the easy integration of ActiveX components, Java applets, and plug-ins for improved Web page interactivity. Dreamweaver also integrates seamlessly with the other components of Macromedia, such as Flash Movies, Shockwave, and Fireworks, which are essential for the development of interactive Web pages.

3.4.5 Conclusion Of Web Authoring Tools

Visual InterDev 6.0 provides an ideal environment for developers to visually construct sophisticated HTML and ASP pages. InterDev is also easier to use than Allaire ColdFusion 4.0 in the sense that most Web developers are already familiar with Visual InterDev and ASP. Furthermore, there are more resources on Visual InterDev and ASP in books and on the Internet in contrast with ColdFusion.

ColdFusion 4.0, on the other hand, uses a special, relatively new scripting language called CFML (ColdFusion Markup Language). CFML files require a special server, called the ColdFusion Server, to be installed on the server first before they can be processed. The ColdFusion Server in turn needs to be configured, and this leads to wastage of precious time. As such, ColdFusion 4.0 will not be feasible for the development of small projects like E-Pack.

Microsoft FrontPage 2000 is a simple to use HTML editor that churns out simple Web pages in a short period of time. However, the development of E-Pack will also require more dynamic Web pages, with livelier animations. Thus, Dreamweaver 3.0, with its superb animation capabilities, should be used in line with FrontPage 2000 to produce more impressive Web pages.

Since E-Pack will be written using ASP, therefore Microsoft Visual InterDev 6.0, together with Microsoft FrontPage 2000 and Macromedia Dreamweaver 3.0, will be chosen as the authoring tool to be used for the development of E-Pack.

3.5 Consideration Of Scripting Languages

3.5.1 VBScript

VBScript, or by its full name, the Microsoft Visual Basic Scripting Edition language, is a simplified version of the Visual Basic and Visual Basic for Applications family of programming languages [**Wrox, 1997**]. It is closely related to the BASIC programming language. While it does not offer the functionality of Visual Basic, it does provide a powerful, easy to learn tool that can be used to add interaction to Web pages.

VBScript is a scripting language, or more precisely a scripting environment, which can enhance HTML Web pages by making them active, as compared to a simple static display. It is the default language of Active Server Pages (ASP) and is event-driven. VBScript provides a small but sufficient set of error handling capabilities. Handling multi-dimensional arrays in VBScript is also a breeze. VBScript arrays can easily be dimensioned and re-dimensioned, as required by the developer.

Specifically, VBScript was created by Microsoft to use either as a client-side scripting language for the Microsoft Internet Explorer (versions 3.0 and later), or as a server-side scripting language with the Microsoft Internet Information Server (versions 3.0 and later). However, VBScript is more often used as a server-side scripting language because of its potent processing capabilities on the server-side.

3.5.2 JavaScript

JavaScript is a platform-independent, event-driven, interpreted programming language developed by Netscape Communications Corporation and Sun Microsystems. Originally called LiveScript (and still called LiveWire by Netscape), JavaScript is affiliated with Sun's object-oriented programming language Java primarily as a marketing convenience [**IDM, 2000**]. They interoperate well but are technically, functionally and behaviorally very different. A popular misconception about JavaScript is that many people believe that JavaScript is Java because of their similar names. This is definitely not true as JavaScript is not Java.

JavaScript is useful for adding interactivity to the World Wide Web because scripts can be embedded in HTML files. All modern browsers are able to interpret JavaScript, which makes it a suitable client-side scripting language. In practice, JavaScript is a fairly universal extension to HTML that can enhance the user experience through event handling and client-side execution, while extending a Web developer's control over the client's browser.

3.5.3 Perl/PerlScript

Perl is a "Practical Extraction and Report Language" freely available for a wide variety of operating systems. It is the most used language for programming CGI scripts. Perl has powerful text-manipulation functions and it combines features and purposes of many command languages. Perl has enjoyed popularity for programming World Wide Web electronic forms and generally as an intermediary between systems, databases, and users.

PerlScript is the key tool for using Perl within the ASP environment [**Powers, 2000**]. It is an ActiveX scripting engine that allows the developer to use Perl with any ActiveX scripting host. Perl can be used for ASP scripting through PerlScript, but developers also have full access to the objects necessary to work within the ASP environment, namely the ASP built-in objects such as the Response and Request objects. Furthermore, PerlScript is fully functional within the ASP scripting environment as VBScript. With PerlScript, developers can develop within an ASP environment and still have access to Perl features, such as the Perl built-in functions, and the vast library of free or low-cost Perl modules to use in the codes.

3.5.4 PHP

PHP (officially "PHP: Hypertext Preprocessor") is a server-side HTML-embedded scripting language. PHP codes are executed on the server. This means, the client would receive the results of running that script, with no way of determining what the underlying code may be. The Web server can even be configured to process all HTML files with PHP, and there is no way that users can tell that the HTML file was written in PHP.

PHP was first conceived sometime in the fall of 1994 by Rasmus Lerdorf. Early non-released versions were used on his home page to keep track of who was looking at his online resume. The first version used by others was available sometime in early 1995 and was known as the Personal Home Page (PHP) Tools [**Bakken et al, 2000**]. It consisted of a very simplistic parser engine that only understood a few special macros and a number of utilities that were in common use on home pages back then. The parser was rewritten in mid-1995 and named PHP/FI Version 2. The FI (Form Interpreter) came from another package Rasmus had written which interpreted html form data. He combined the Personal Home Page tools scripts with the FI, and added mSQL support and PHP/FI was born. PHP/FI grew at an amazing pace and people started contributing code to it. Today, PHP ships with a number of commercial products, with Red Hat Linux being one of them.

PHP is different from a CGI script written in other languages like Perl or C. Instead of writing a program with many commands to output HTML, the programmer writes an HTML script with some embedded code to output text instead. However, PHP can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. Perhaps the strongest and most significant feature in PHP is its support for a wide range of databases. Writing a database-enabled Web page is, therefore, incredibly simple with PHP.

Developers have literally flocked to PHP, with its modest learning curve, free and open development, native database connectivity, stability, and availability for a variety of platforms. PHP's primary strength is in its rapid development of dynamic

Web pages. Developers without heavy programming experience can leverage PHP to complete tasks otherwise cryptic or obtuse in alternative languages. The architecture of the PHP language is simple but powerful, and includes an extremely wide variety of functions suited for many tasks, both traditional data processing tasks and more Web-oriented functions as well.

3.5.5 CFML

CFML or ColdFusion Markup Language is a set of special tags that are typically placed inside HTML pages and interpreted by the ColdFusion Server [**Future Systems, 1999**]. The tags perform such tasks as database connectivity and conditional logic, but they also handle other needs of the Web developer.

This server-side scripting language contains more than 70 tags for database connectivity, conditional logic, input and output, and integration with other Internet and file services [**Lipschutz, 1999**]. Over 200 additional functions handle items such as date and time, mathematical functions, and string manipulations. For database connectivity, SQL statements are embedded into Web pages by using CFML shortcuts, such as CFFORM and CFTABLE, which add functionality to HTML forms and simplify HTML tables, respectively. Additionally, CFML is extensible and it seamlessly integrates with major distributed objects standards such as COM and CORBA.

CFML files must be processed on the ColdFusion Server each time an application page is requested. As such, Web servers will need to be installed with the ColdFusion Server first before any CFML files could be processed. Despite the limitations, CFML still provides a dynamic application environment that is powerful and easy to use.

3.5.6 Conclusion Of Scripting Languages

Since the development of E-Pack will require the use of ASP, it is vital that VBScript be utilized as the server-side scripting language, and JavaScript as the client-side scripting language. Furthermore, there are a lot of resources about VBScript and JavaScript in books and on the Internet, which facilitates the development of this project.

PerlScripts, on the other hand, uses the old fashion Perl programming language. This is not in accordance with the objective of this project, which is to make use of the latest Web advancements to set up an e-commerce Web site. Hence, PerlScript will not be used for this project.

The usage of PHP and CFML will each require the setting up of the PHP Hypertext Preprocessor and the ColdFusion Server on the server. This will only lead to additional configuration works, which may take up a lot of precious time. Not to mention the time taken up just to learn up these two relatively new scripting languages. As such, PHP and CFML are considered to be not feasible for this project.

3.6 Consideration Of Databases

3.6.1 Microsoft Access 2000

Microsoft Access 2000 is a Windows-based database management system. It is a member of the Microsoft Office 2000 family and it runs under the Windows 95/98/NT/2000 operating systems. Due to the fact that Access is part of the Office 2000 suite, it interoperates well with the other components of the Office 2000 family.

Access is easily the world's most popular relational database management software (RDBMS) **[Sellapan, 1999]**. It is powerful and yet easy to use. With Access, the database administrator can design and use databases very quickly, as it has a very user-friendly interface. Furthermore, tables, forms, queries, and reports can be generated just at a snap of a finger, just by using the set of wizards that come with this software. All this makes Access an excellent all-in-one database tool for creating standalone database applications.

3.6.2 Microsoft SQL Server 7.0

Microsoft SQL Server 7.0 is a modern, full-featured SQL database designed for small or midsize organizations. Its complete set of tools, high-end engine features, and robust analysis capabilities provide most of what other corporations could offer only in their Enterprise Edition databases, all at a reasonable price. In addition, SQL Server 7.0 is amazingly easy to use, yet still powerful enough to crank through hundreds of complex transactions per second without choking [**Dyck, 1999**].

Many engine settings in SQL Server 7.0 are self-tuning. Developers need not assign memory to the data cache or store procedure cache separately. SQL Server dynamically balances memory between the two. SQL Server expands or contracts the amount of memory it is using as a whole. It can automatically make room in memory for other applications when they are running, and expand again to fill extra memory when the applications are closed down.

Automatic memory tuning is not especially important on a dedicated database server machine. But on a server that has to run other applications like a mail server and Web server at the same time as a database, dynamic memory sizing makes a huge difference to system usability. SQL Server's unmatched auto-tuning features also means that it is extremely suitable for organizations that do not have database administrators on staff. With SQL Server 7.0, a part-time or beginner administrator is all it takes to manage the database effectively.

Despite all its great features, SQL Server 7.0 runs only on the Windows platform. Nevertheless, if a developer is running generic packaged Windows applications to access the contents of a database, or custom applications that use the ODBC standard to get information from the database, then SQL Server 7.0 is the best choice.

3.6.3 Conclusion Of Databases

Access 2000 and SQL Server 7.0 are excellent database management software from Microsoft. Both of them are easy to use and are efficient at handling databases. As such, both of them are worthy of consideration for this project.

Access 2000 is suitable for use with small to medium-sized databases. However, the size of a database may grow rather huge at times, and this is when Access 2000 starts to reveal its deficiencies. Access 2000 has the tendency of slowing down when the tables in a database gets too huge. This may not be practical for e-commerce Web sites running huge databases, as the opportunity costs arising from the slowing down of the database may be significant. As such, Access 2000 is unsuitable for the management of huge databases.

SQL Server 7.0, on the other hand, works well with databases of any size. It contains all the user-friendly features of Access 2000, yet it works so many times more efficiently than Access 2000. It has the ability of handling hundreds of transactions simultaneously without affecting performance.

SQL Server 7.0 will therefore be chosen to act as the database management software for the development of E-Pack.

3.7 Consideration Of Web Servers

3.7.1 Internet Information Server 4.0

Internet Information Server (IIS) 4.0 is a built-in Web server that comes with the Microsoft Windows NT 4.0 operating system. It provides the easiest way to share information, build and deploy business applications, as well as host and manage sites on the Windows platform.

IIS 4.0 comes with three default services: WWW, FTP, and Gopher [**Alwang, 1997**]. Its Internet Service Manager (ISM) application controls these services on this or any other IIS server on the network. ISM is run from the Windows NT Server or from a Windows NT or Windows 95/98 workstation. For remote administration, ISM can be run from any browsers using an HTML version.

IIS 4.0 teams up well with ASP to provide an extensive server-side platform supporting compile-free, language-independent scripts and ActiveX components. This, coupled with the fact that IIS returns all ASP requests as standard HTML, lets Web developers create truly dynamic Web sites and online applications accessible by any browser. Database access has also been extended in IIS 4.0. ActiveX Data Objects (ADO), an ASP component, allow developers to access and control data in any ODBC- or OLE DB-compliant database using any ActiveX scripting language.

Building on Windows NT's security prowess, IIS also provides additional levels of security. With IIS, Web developers can restrict access to a directory or URL by user, group, or IP address, or by using Windows NT's Challenge/Response authentication or SSL 3.0. This ensures that the security is always optimal on any Web sites managed on IIS 4.0.

3.7.2 Apache Server

The Apache server is a powerful, flexible, HTTP 1.1 compliant Web server. It is highly configurable and extensible with third-party modules. It provides full source code and comes with an unrestrictive license. It runs on Windows NT/9x, OS/2, and most versions of Unix, as well as several other operating systems. Apache is actively being developed and encourages user feedback through new ideas, bug reports and patches.

The Apache server allows administrators to easily set up password-protected pages with enormous numbers of authorized users, without slowing down the server. It also permits administrators to set up customized files, or even CGI scripts, which are returned by the server in response to errors and problems. This allows the administrator to perform on-the-fly diagnostics for both users and administrator.

Apache is also flexible enough to perform multiple DirectoryIndex directives, where administrators can instruct the server to either send back index.html or run index.cgi when a directory URL is requested, whichever it finds in the directory. Those running Apache servers will also find that it has unlimited flexible URL rewriting and aliasing. Apache has no fixed limit on the numbers of Aliases and Redirects that may be declared in its configuration files. In addition, a powerful rewriting engine can be used to solve most URL manipulation problems.

Today, the Apache server is the most widely implemented Web server on the Internet [**Herel, 1997**]. It offers a powerful and customizable approach for any Unix-based server. It has been shown to be substantially faster, more stable, and more feature-full than many other Web servers, including IIS. Apache is run on sites that get millions of hits per day, and they have yet experienced any performance difficulties.

3.7.3 Conclusion Of Web Servers

Although Apache is proven to be the more powerful Web server here, it is still not as user friendly as IIS 4.0. Furthermore, Apache servers are known for their difficulty in installation and configuration [**Herel, 1997**]. As such, it will take considerable time and effort to configure and manage the Apache server, thus making it unsuitable for the development of this project.

Since Windows NT will be used as the platform for the development of E-Pack, IIS 4.0 would seem to be a better choice here. Furthermore, IIS is included in the Windows NT 4.0 Option Pack and is easily configurable on Windows NT 4.0.

3.8 Conclusion

As a conclusion, the tools and programming language to be used for the development of E-Pack are:

Software/Tools

Software	Function
Microsoft Windows NT 4.0	Operating system
Microsoft Internet Information Server 4.0	Web server
Microsoft SQL Server 7.0	Database Management System (DBMS)
Microsoft Visual InterDev 6.0	Integrated Development Environment (IDE) for ASP files
Macromedia Dreamweaver 3.0	Visual HTML editor
Adobe Photoshop 5.0, Microsoft Paint	Graphics designer
Animagic GIF Animator 1.21	Graphics animator
Microsoft Internet Explorer 5.x/ Netscape Navigator 4.x	Web browser

Table 3-1: Software Requirements for E-Pack

Programming/Scripting Languages

- Active Server Pages (ASP)
- VBScript
- JavaScript
- Hypertext Markup Language (HTML)

Database Manipulation

- ActiveX Data Objects (ADO)
- Structured Query Language (SQL)

CHAPTER 4: SYSTEM DESIGN

4.1 Introduction

Software design is the process of devising and documenting the overall architecture for a software system. It includes identifying the major components of the system, specifying what they are to accomplish, and establishing the interfaces among the components.

Design is the first step in the process of transforming the requirements into a close representation of the eventual function software. It also includes lower level work such as detailed specification of data structures and algorithms within the identified components.

4.2 Design Issues

There are many issues involved in the creation of a design. For example:

- What is best for the application?
- What is most comfortable for the designer/developer?
- What makes sense for the overall architecture?

Thus, no one style is best for every situation. We must understand and compare the new styles and techniques with the current methods. Below are the two main issues that must be addressed when creating the design for this project:

i) **Database Design**

This project involves a large storage of information. In order to handle it, there are two recommended ways: file-processing system or database processing system. File-processing programs directly access files of stored data. In contrast, database-processing programs call the DBMS to access the stored data.

The latter is chosen for this project because it was developed largely to overcome the limitations of the file-processing systems. Database processing has a few advantages over file-processing systems **[Kroenke, 1998]**:

- Data is integrated
- Reduced data duplication
- Application programs and data are independent
- Easier representation of the users' perspectives.

ii) User Interface

The user interface can sometimes be tricky to design because different users have different styles of perceiving, understanding, and working. For example, some users may prefer to use the keyboard to handle their tasks, whereas others may rely on the mouse.

4.3 System Functionality Design

4.3.1 Three-Tier Client/Server Architecture

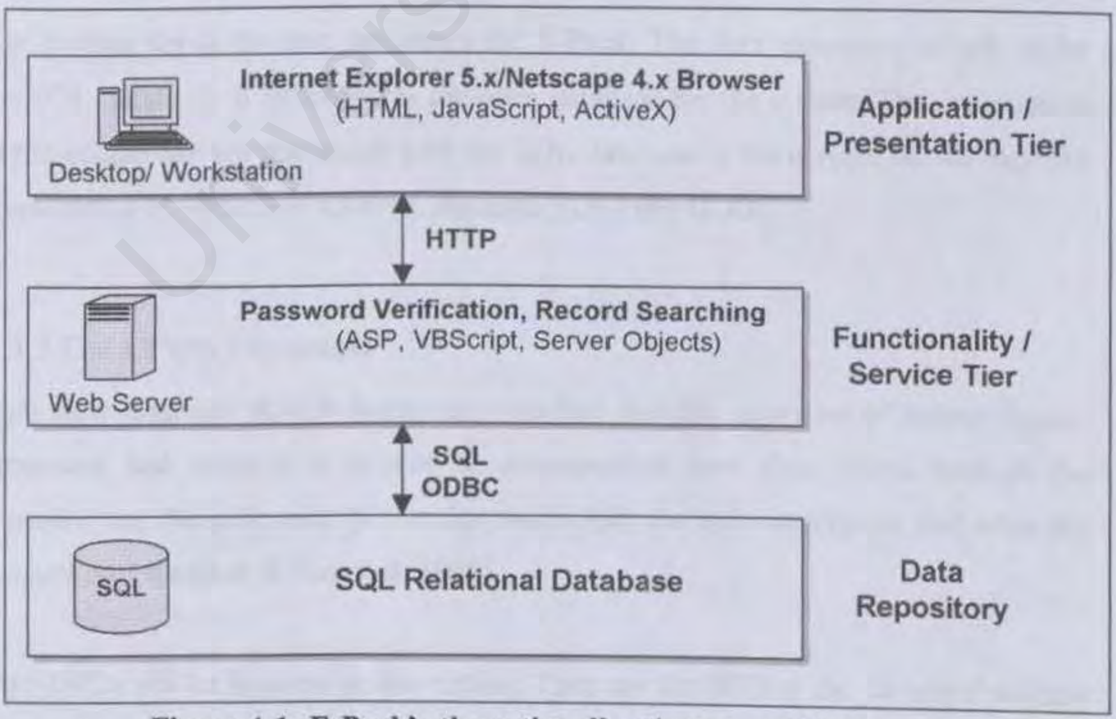

Internet Explorer 5.x/Netscape 4.x Browser
(HTML, JavaScript, ActiveX)
Desktop/ Workstation
Application / Presentation Tier
HTTP
Password Verification, Record Searching
(ASP, VBScript, Server Objects)
Web Server
Functionality / Service Tier
SQL
ODBC
SQL
SQL Relational Database
Data Repository

Figure 4-1: E-Pack's three-tier client/server architecture

In E-Pack, the application/presentation tier consists of all the necessary applications (Figure 4-1). In this layer, the main application component that appears to the user is the Web browser, which will provide the user interface. HTML, JavaScript and ActiveX controls are used to activate the application layer. All of them provide the most flexible and dynamic interface for the users. The application always resides within the Web server, which is the Internet Information Server (IIS) 4.0.

The middle tier is known as the functionality or service tier. The communications between this tier and the application layer depends on the Hypertext Transfer Protocol (HTTP) for the Web pages transfer. The functionality tier consists of the components that are created to support E-Pack such as password verification, searching for records, and other configuration. All these components require Active Server Pages, VBScript, and ActiveX objects to perform the functions in the Web servers. The IIS in this tier will processes the request from the client and produces the result in Web pages format. The IIS will also process any data request of the user by linking to the database server, which is contained at the bottom tier of the three-tier client/server architecture. The IIS will do other extra additional activity during the data processing

The bottom tier is the data repository for E-Pack. The data repository is built up by the SQL database. It functions as the main database for the system. The components in the middle tier are connected with the SQL database in the bottom tier through the combination of Structured Query Language (SQL) and ODBC.

4.3.2 Data Flow Diagrams

Data flow diagrams (DFD) depict the broadest possible overview of system inputs, processes, and outputs. It is able to conceptualize how data moves through the organization, the processes or transformation that the data undergoes, and what the outputs are [**Kendall & Kendall, 1999**].

Two DFDs will be featured in this section. They are the DFD of the shoppers' module (Figure 4-2) and the DFD of the administrator's module (Figure 4-3).

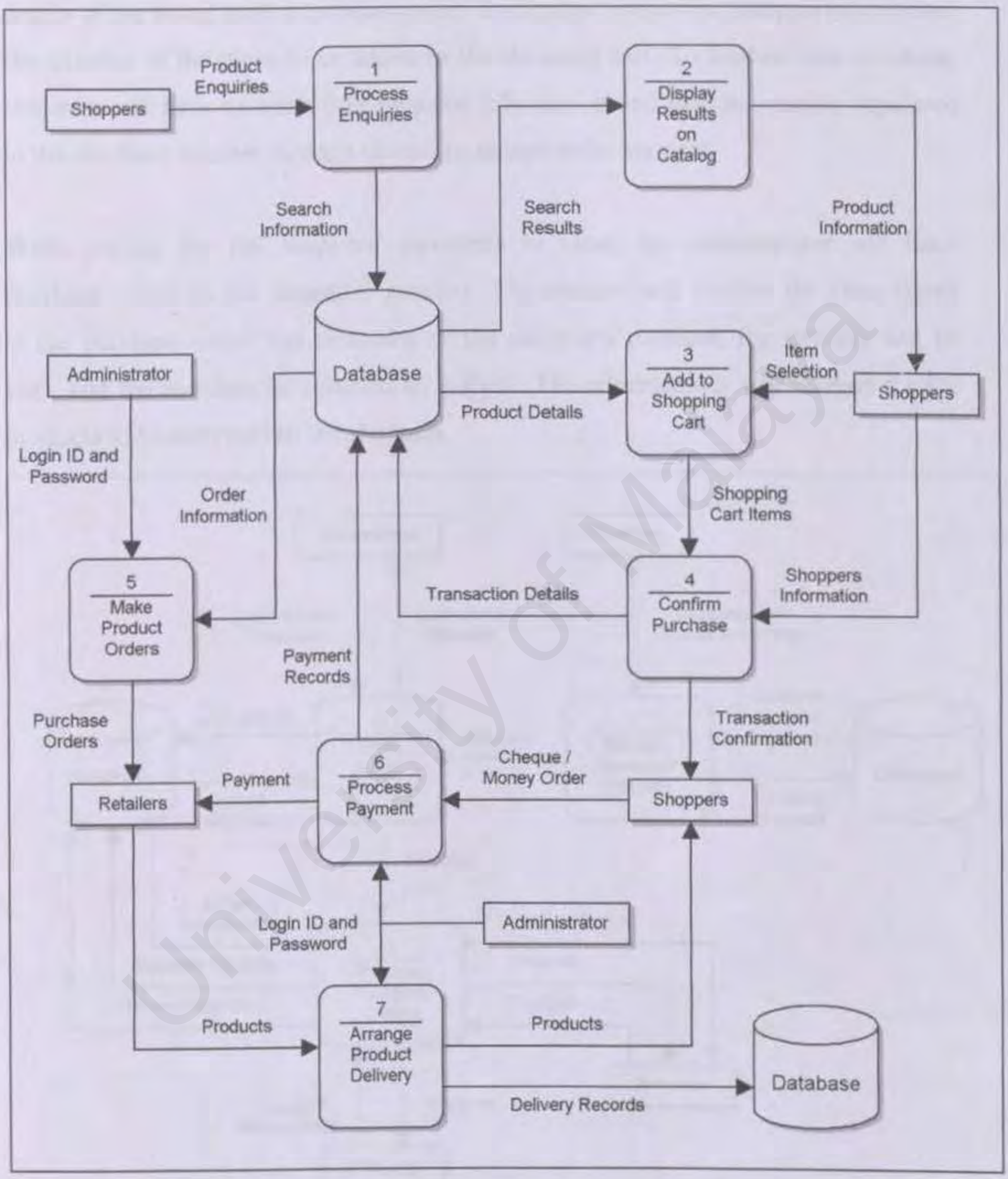


Figure 4-2: DFD of the shoppers' module of E-Pack

The DFD of the shoppers' module shows that the shoppers will have to initiate the request for the products that they are looking for, either through the links provided on the menu or the search box provided on the Web page. The search results will then be displayed on a Web page, also known as the catalogue, where shoppers can view the details of the items, such as product price, description, and so on. Shoppers then select the quantity of the items to be added to the shopping cart. To finalize their purchase, shoppers will have to enter their personal information and pay the amount stipulated in the checkout counter through cheque or money order via post.

While waiting for the shoppers' payments to clear, the administrator will issue purchase orders to the respective retailers. The retailers will standby the items stated in the purchase order. On clearance of the shopper's payment, the retailers will be paid, and the merchandise released to E-Pack. The administrator will arrange for the products to be delivered to the shoppers.

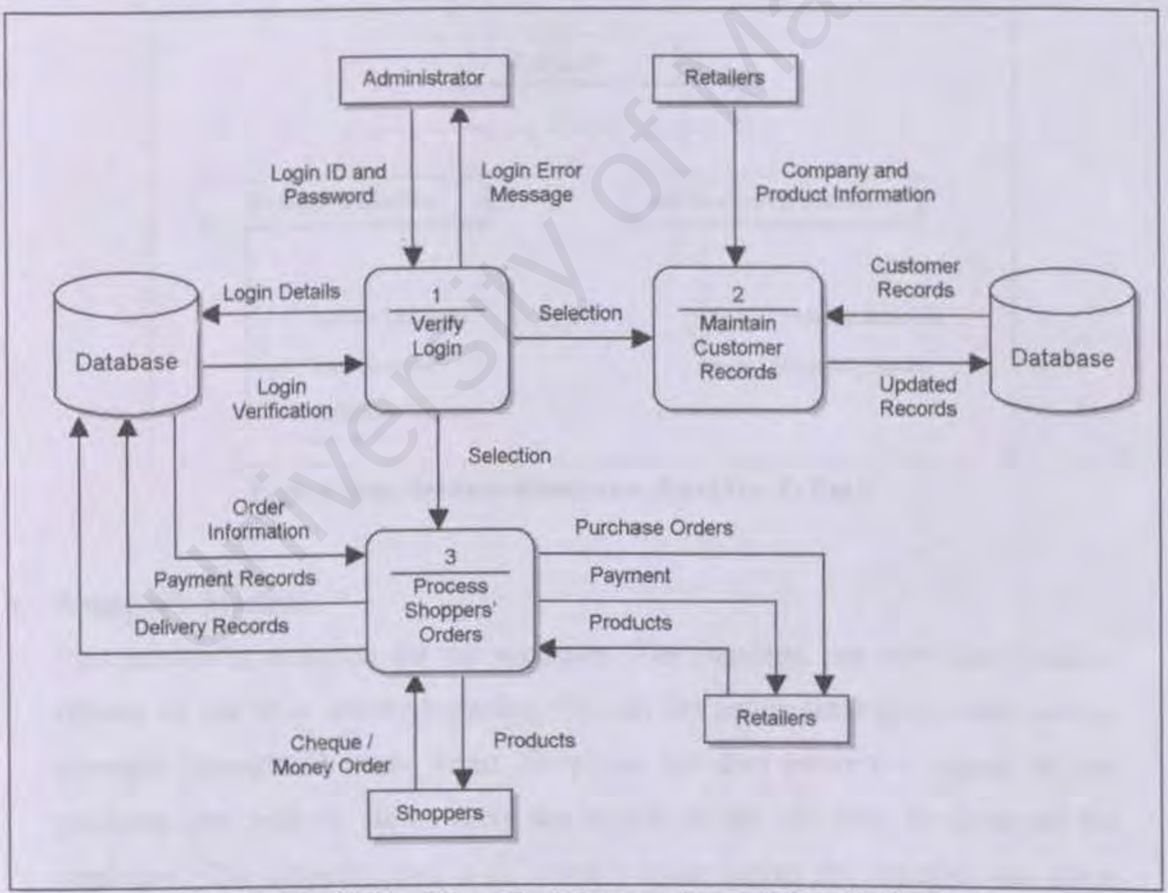


Figure 4-3: DFD of the administrator's module of E-Pack

From the DFD of the administrator's module, we can see that the process of verifying a user login to the system will require the administrator's login ID and password. The system will then check whether the user is authorized to administer the database. Otherwise, the system will return a login error message to the user. Administrators will then select whether to perform maintenance works on the customer records, or process the shoppers' orders.

4.3.3 System Structure Chart

E-Pack is divided into two main modules, which are the shopper's module and the administrator's module. Each of these modules will perform different system functions. The modules are further divided into sub-modules, as shown in the following chart:

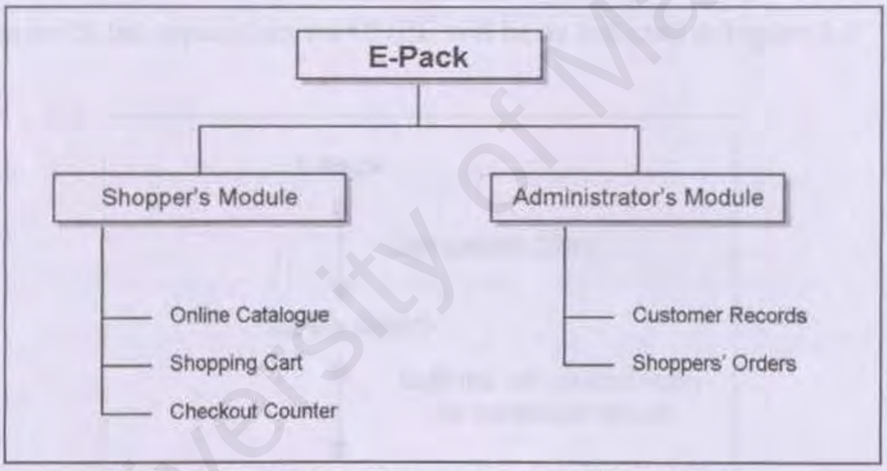


Figure 4-4: System structure chart for E-Pack

- **Shopper's Module**

  This module is designed for the shoppers. The shoppers can view the products offered on the Web site by browsing through the online catalogue, which can be accessed through the main menu. Shoppers can also perform a search on the products they wish to view, where the search results will later be displayed the catalogue. The shopping cart is a "virtual" space where the shoppers can place their selected items before finalizing their purchase. The checkout counter on the other hand will finalize the transaction by requesting for the shoppers' information, method of payment, and mode of delivery.

- **Administrator's Module**

  This module is designed for administrators, where they can perform maintenance work for customer records, as well as the processing of shoppers' orders.

4.4 Database Design

4.4.1 Database Structure

Database technology has been used to store data in a flexible format. However, the data cannot simply be stored. Thus, the data models that were developed for the system users during the definition phase must be translated into data structure. Since E-Pack is using Microsoft SQL Server 7.0 as the database platform, the mapping of the SQL Server to the application via ODBC will be as depicted in Figure 4-5:

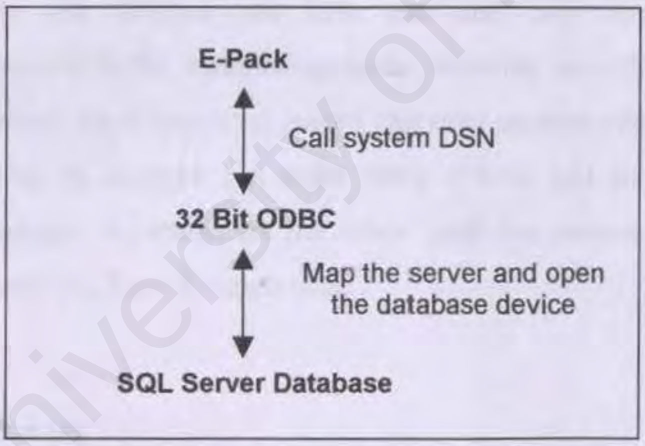


Figure 4-5: The mapping of the database to the application

4.4.2 Entity-Relationship (E-R) Model

Entity-relationship model is the constructs and conventions used to create a model of the users' data. The things in the users' world are represented by entities, and the associations among those things are represented by relationships. The results are usually documented in an entity-relationship (E-R) diagram [**Kroenke, 1998**].

Figure 4-6 illustrates the database E-R diagram of E-Pack:

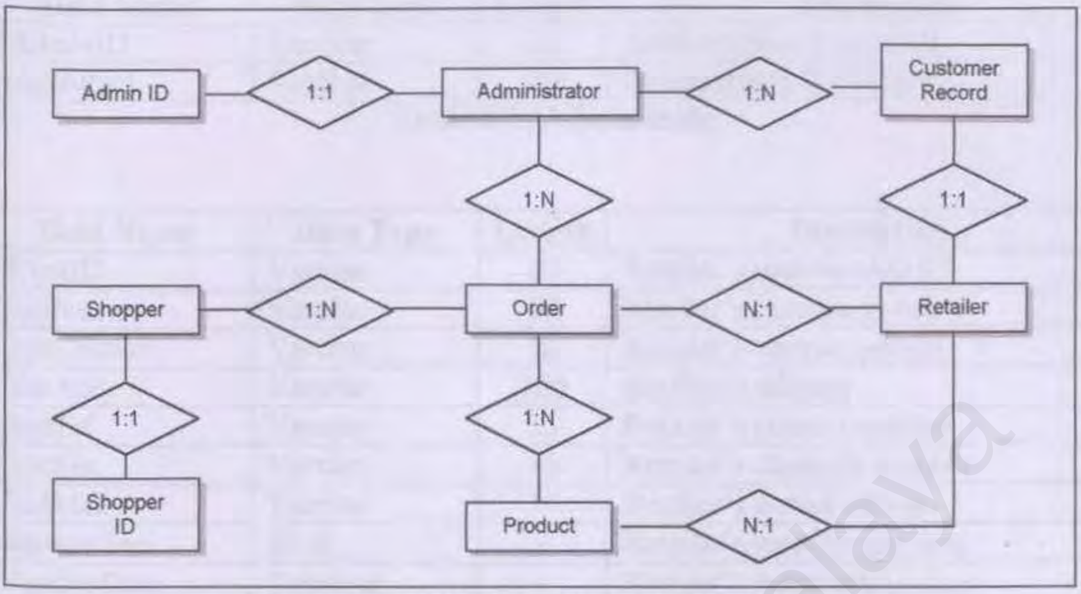


Figure 4-6: E-Pack's database E-R diagram

Each administrator and shopper can have one and only one login ID. Each administrator is responsible for maintaining many customer records, and may process many shoppers' orders. Each customer record can only contain information about one and only one retailer. A shopper can make many orders, and an order can contain many items or products. A retailer on the other hand can receive many orders, and they may supply many products for each order.

4.4.3 Data Dictionary

The data dictionary is a specialized application of the kinds of dictionaries used as references in everyday life. It is a reference work of data about data (metadata), one that is compiled by system analysts to guide them through analysis and design **[Kendall & Kendall, 1999]**.

The following are the data dictionary that explains the items and fields of the database that will be used in E-Pack, where asterisks (*) in tables represent primary keys in their respective tables:

Field Name	Data Type	Length	Description
*AdminID	Varchar	20	Administrator's login ID
AdminPwd	Varchar	20	Administrator's password

Table 4-1: Admin table

Field Name	Data Type	Length	Description
*CustID	Varchar	20	Retailer's membership ID
CustName	Varchar	50	Retailer's / company name
CustContact	Varchar	30	Retailer's contact person
CustAdd	Varchar	100	Retailer's address
CustTel	Varchar	15	Retailer's contact number
CustFax	Varchar	15	Retailer's facsimile number
CustMail	Varchar	30	Retailer's e-mail address
MemberFee	Float	-	Retailer's membership fees
MemberDate	Datetime	-	Retailer's membership date
LastPaidFee	Datetime	-	Date of last paid membership fee
CustRemarks	Varchar	200	Remarks

Table 4-2: Customer table

Field Name	Data Type	Length	Description
*ShopperID	Varchar	20	Shopper's login ID
ShopperPwd	Varchar	20	Shopper's password
ShopperName	Varchar	30	Shopper's name
ShopperAdd	Varchar	100	Shopper's address
ShopperTel	Varchar	15	Shopper's contact number
ShopperMail	Varchar	30	Shopper's e-mail address

Table 4-3: Shopper table

Field Name	Data Type	Length	Description
*ProdID	Varchar	20	Product ID
ProdName	Varchar	30	Product name
ProdDetail	Varchar	200	Product details
ProdPrice	Float	-	Product price
ImageURL	Varchar	30	Product image URL
CustID	Varchar	20	Retailer related to this product
DeptName	Varchar	30	Product department
ProdDesc	Varchar	2000	Product description
FeaturedProd	Datetime	-	Featured products / latest arrivals

Table 4-4: Product table

Field Name	Data Type	Length	Description
*DeptID	Integer	-	Department ID
DeptName	Varchar	30	Product category (e.g. Books, Food)

Table 4-5: Department table

Field Name	Data Type	Length	Description
*BasketID	Integer	-	Basket ID
BasketExpires	Datetime	-	For deletion purposes

Table 4-6: Basket table

Field Name	Data Type	Length	Description
BasketID	Integer	-	Basket ID
ProdID	Varchar	20	Product related to this basket
UnitPrice	Float	-	Unit price
Quantity	Integer	-	Product quantity
Total	Float	-	Total price

Table 4-7: BasketItem table

Field Name	Data Type	Length	Description
*OrderNo	Integer	-	Order number
OrderDate	Datetime	-	Order date and time
ShopperID	Varchar	20	Shopper's login ID
PayMethod	Varchar	20	Payment method
Subtotal	Float	-	Total payment before delivery
DeliveryMethod	Varchar	20	Delivery mode
DeliveryCharges	Float	-	Delivery charges
ServiceFee	Float	-	Fees for services provided by E-Pack
Total	Float	-	Total payment (delivery + service fee)
ProcessStatus	Integer	-	Order status (processed or not)
PayStatus	Integer	-	Payment status (shopper paid or not)
DeliveryStatus	Integer	-	Delivery status (delivered or not)
OrderRemarks	Varchar	200	Remarks

Table 4-8: Orders table

Field Name	Data Type	Length	Description
OrderNo	Integer	-	Order number
ProdID	Varchar	20	Product related to this order
UnitPrice	Float	-	Unit Price
Quantity	Integer	-	Product quantity
ItemTotal	Float	-	Total price

Table 4-9: OrderItem table

Field Name	Data Type	Length	Description
*DeliveryID	Integer	-	Delivery mode ID
DeliveryMethod	Varchar	20	Mode of delivery (land/air)
DeliveryCharges	Float	-	Delivery charges

Table 4-10: DeliveryCharges table

Field Name	Data Type	Length	Description
ServiceFee	Float	-	Fees for services provided by E-Pack

Table 4-11: ServiceFee table

The Admin table is used to store the login ID and password of the administrators authorized to access the database. The Customer table, on the other hand, is used to store the details of the retailers who are currently subscribing to E-Pack. The Shopper table, as its name suggests, stores the details of the online shoppers in E-Pack. Shopper information may be useful in reconciling future sales.

The products displayed on the Web site require a table to store all its information. Hence, the Product table is used to store these products. In order to systematically catalog these products into easy-to-recognize categories, the Department table is employed, where each product is given a department name.

When a shopper selects items to be placed in the shopping cart (basket), a temporary table is needed to represent this "virtual" basket. Therefore, the Basket table is used to store the items of the basket, where else the BasketItem table houses the contents of each item in the basket. As the records of the Basket and BasketItem tables are only temporary, records exceeding two days will be deleted each time the administrator runs the administration function.

After the shopper confirms the purchase of the items in the shopping basket, the order will be placed in the Order table, while the selected items will be transferred into the OrderItem table. The DeliveryCharge table, on the other hand, stores the delivery charges for the mode of delivery selected by the shopper. Both the Order and OrderItem tables are vital in processing and keeping track of the shoppers' orders.

Last but not least, the ServiceFee table stores the percentage of service fee to be charged for the services provided by E-Pack. Although this table is just a simple table containing only one record (there can only be one service fee rate at any one time), this table plays a crucial role in determining the percentage of income to be gained from each order processed on E-Pack.

4.5 User Interface

The user interface is the system for most users. There are several kinds of user interface, such as natural-language interfaces, question-and-answer interfaces, menus, form-fill interfaces, command-language interfaces, and graphical user interface (GUI) [**Kendall & Kendall, 1999**].

User interfaces are designed to improve the efficiency and effectiveness of the user when using the entire system. Thus, the user interface for E-Pack is easy to understand and easy to use. Users need not remember any complex commands as everything is done at the click of the mouse. As such, the interfaces in E-Pack will be created as friendly as possible.

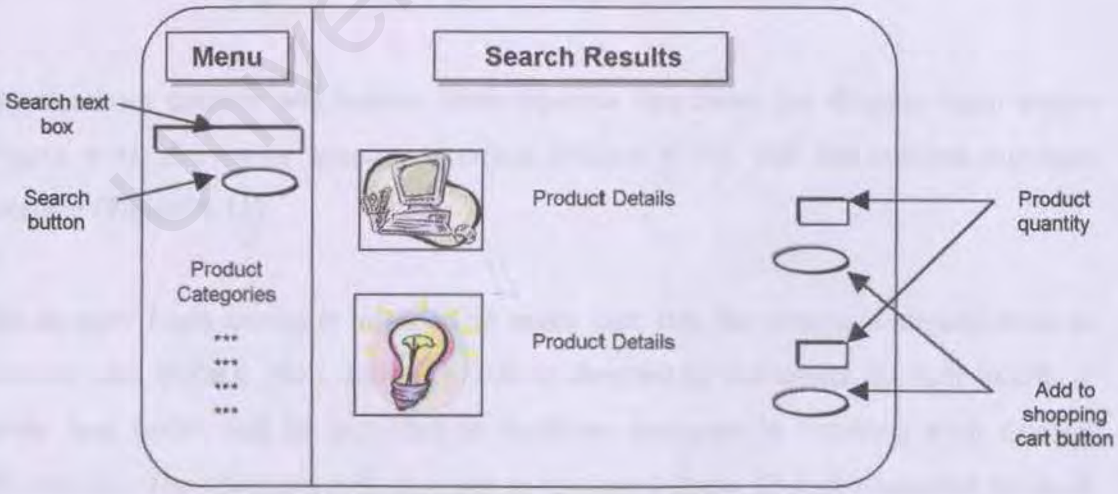


Figure 4-7: E-Pack's online catalogue interface

The online catalogue will be designed as attractive as possible in order to convince the shoppers to purchase the items on the catalogue. It will also feature clear and easy to understand buttons to facilitate the shoppers in browsing the catalogue, as well as in choosing the items they want on the catalogue.

The interface of the shopping cart will display the current items chosen by the shoppers, as well as show the quantity and price of each items selected by the shoppers. A subtotal is provided to aid the shoppers in totaling the item prices. This prevents the shoppers from having to calculate the amounts themselves, which may lead to calculation errors.

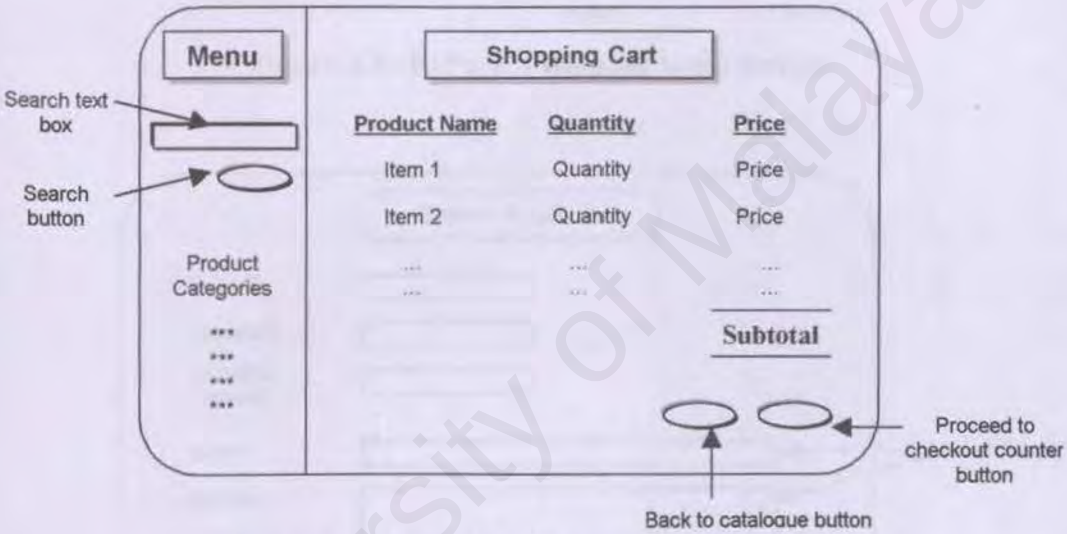


Figure 4-8: E-Pack's shopping cart interface

The checkout counter will feature three separate interfaces: the shopper login screen (Figure 4-9), the create account interface (Figure 4-10), and the confirm purchase interface (Figure 4-11).

The shopper login screen is intended to make sure that the shoppers already have an account with E-Pack. New shoppers will be directed to the create account interface, where text boxes will be provided to facilitate shoppers in inputting their contact information. The shoppers will also get to choose a login ID and password for their account on this page.

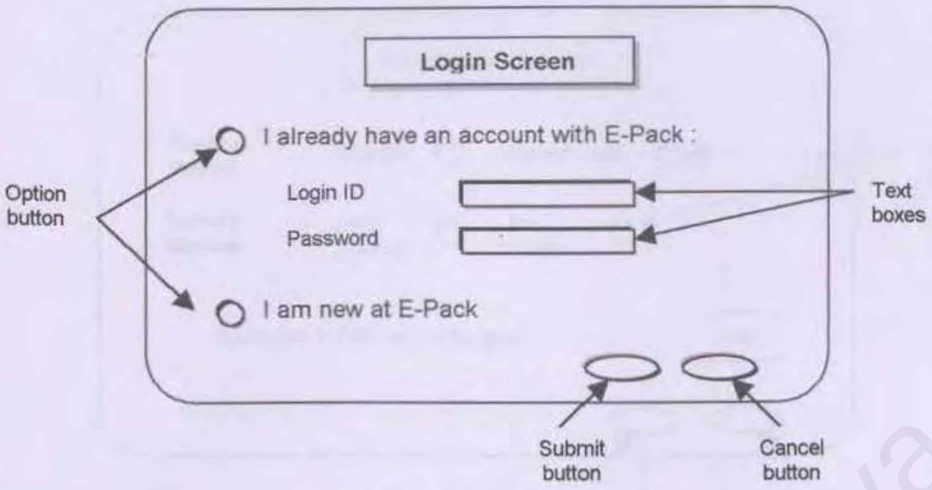


Figure 4-9: E-Pack's shopper login screen

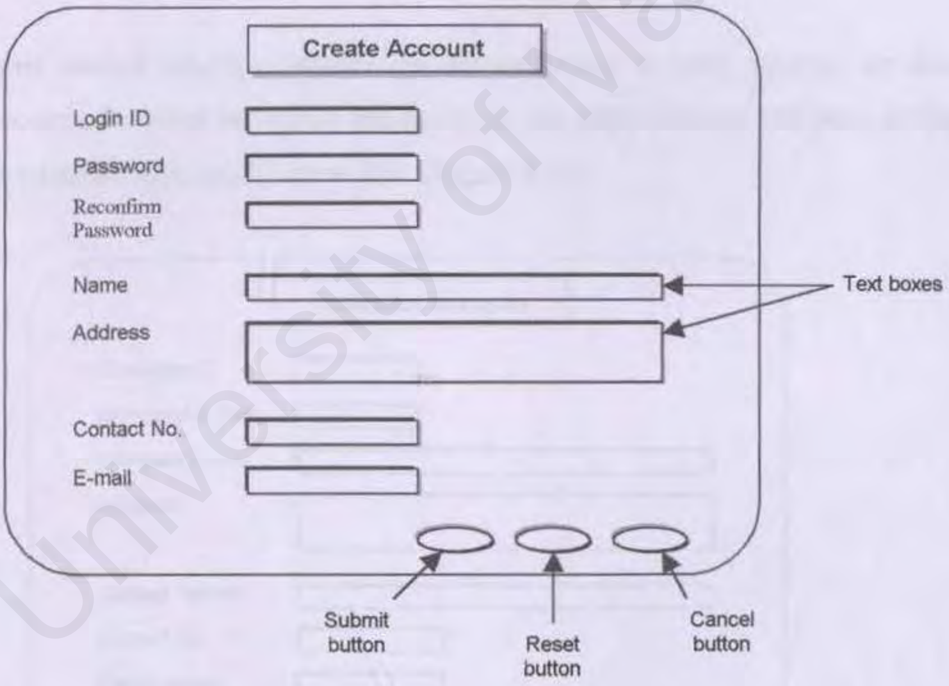


Figure 4-10: E-Pack's create account interface

Returning shoppers, on the other hand, will straightaway be directed to the confirm purchase page from the shopper login screen. This is the page where the shoppers are requested to state their method of payment and delivery medium through the usage of option buttons. The total amount to be paid is then stated at the bottom of the page.

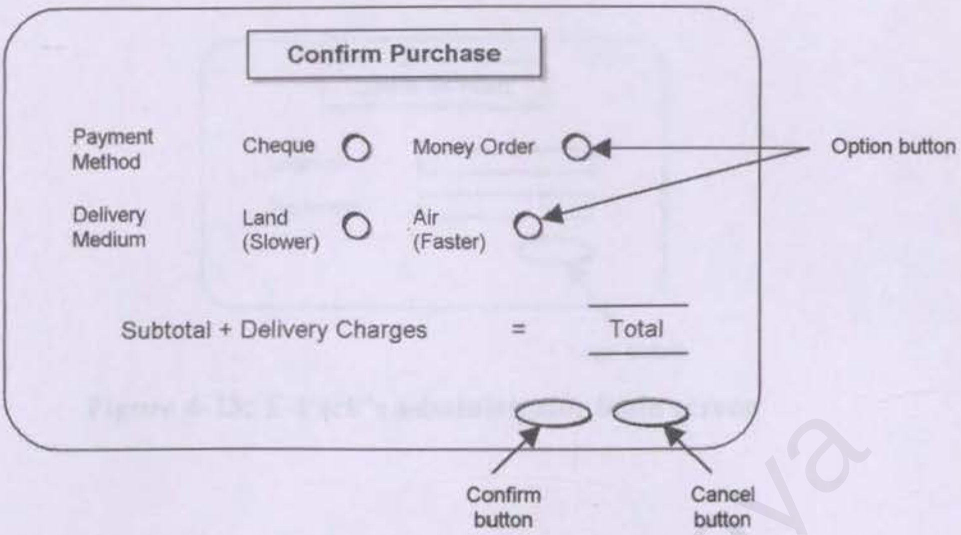


Figure 4-11: E-Pack's confirm purchase interface

The customer record interface allows the administrator to add, update, or delete customer records. In order to access this function, the administrator will have to login through the administrator login screen first (Figure 4-13).

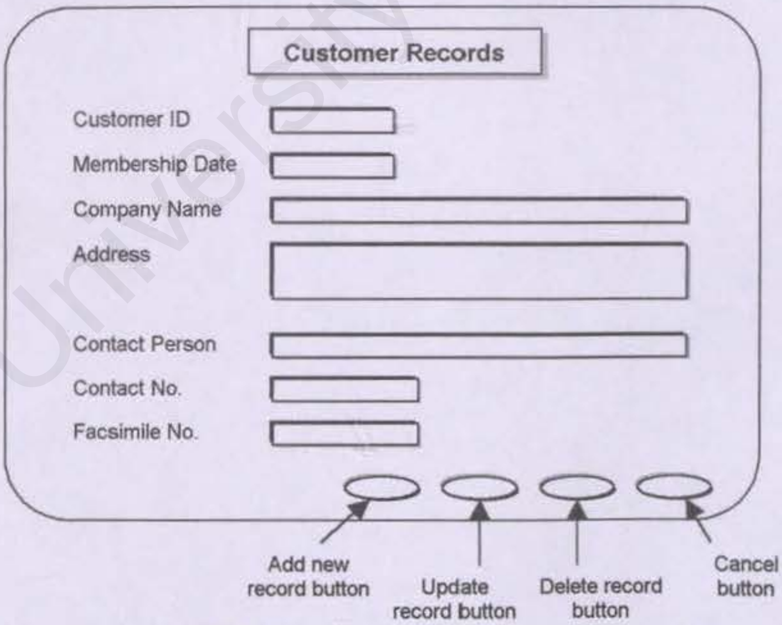


Figure 4-12: E-Pack's customer record interface

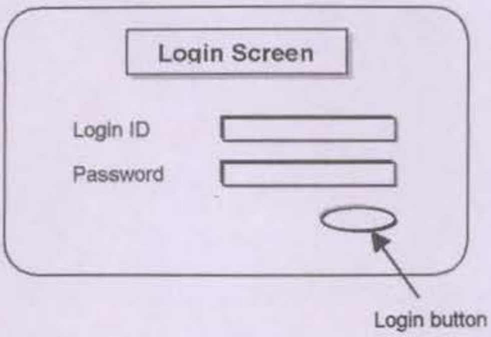


Figure 4-13: E-Pack's administrator login screen

CHAPTER 5: SYSTEM IMPLEMENTATION

5.1 Introduction

In the system implementation stage of this project, the design model of E-Pack was transformed into a feasible, workable product. This chapter briefly summarizes the work that has been done on the system, as well as how the suggested programming technology and development tools were used in the development of this system. The standards of programming used to develop the system will also be shortly explained in this chapter.

This chapter will be divided into two main sections. They are the platform development and the modules implementation.

5.2 Platform Development

The server used for the development of E-Pack is the Hewlett Packard Netserver with 500 MHz Pentium III processor and 128 MB RAM running on Microsoft Windows NT 4.0, situated in the Projects Laboratory of the Faculty of Computer Science and Information Technology. Since the HP server already has Windows NT 4.0 installed in it, together with its various option packs and service packs, there was no need to configure the server. Therefore, the platform development will only include the setting up of the IIS server and the configuration of the SQL server.

5.2.1 Setting Up IIS

The Microsoft Internet Information Server 4.0 (IIS) can be found in the Windows NT Option Pack 4. After installing the IIS, a virtual directory called "EPack" is created within the root directory of IIS (by default is **C:\InetPub\wwwroot**) so that the user can access the system. In addition to that, the IIS was also configured to allow anonymous access to the system, so that users can access the system from anywhere in the world. The system can be accessed at http://10.100.1.205/EPack/

5.2.2 Configuring SQL Server

The Microsoft SQL Server 7.0 was also installed in the same server as the IIS. After SQL Server has been installed successfully, a database with the name "**EPack**" was created. Then, the tables were created according to the **data dictionary** that was presented in chapter four of this report **(refer Section 4.4.3)**. A user account with the login ID "**EPackAdmin**" was then created to allow access to the "EPack" database.

5.3 Modules Development

5.3.1 Programming With Visual InterDev

Once Microsoft Visual InterDev 6.0 has been installed, a new project by the name of "**EPack**" was created in InterDev to represent the system. New ASP files were then added one-by-one to the project in order to build the Web pages of the system. The ASP files were given the ".asp" file extension to denote that they contain scripting (VBScript) in them. This way, IIS could interpret the scripts on the ASP pages, while hiding these scripts from the viewers, and maintaining the secrecy of the source codes used in the ASP files.

- **The <% %> tag**

  All ASP codes are enclosed in the <% %> tag. For example:

```
<HTML>
<BODY>
The year is
<%
   Response.Write Year(Now())
%>
</BODY>
</HTML>
```

  The above codes will display:

```
The year is 2001
```

  on the Web browser.

- **Program variables**

  In order to use variables in the ASP pages, they have to be declared first. For example:

```
<%
   Dim counter
   Dim a
%>
```

This is to ensure that only variables relevant to the program are used in the ASP codes. To prevent the use of undeclared variables in the ASP pages, an extra line of code is entered into the top of each ASP page:

```
<% Option Explicit %>
```

Meaningful names were chosen for the variables in the development of E-Pack. These names reflect their use or meaning, and do not confuse readers into believing it as referring to something else. For example, writing

```
<% counter = counter + 1 %>
```

makes more sense to the reader than

```
<% a = a + 1 %>
```

- **Program Comments**

  In InterDev, a program comment can begin with either the apostrophe sign (') or the "Rem" abbreviation (abbreviation for "remark"). All comments are single-lined comments that terminate at the end of the current line. For example:

```
<%
   ' This is a comment beginning with the apostrophe sign
   This is not a comment
   Rem This is another comment beginning with the Rem abbrev.
%>
```

An abundance of program comments were inserted into the ASP pages of E-Pack to make understanding the program codes easier.

- **Formatting To Enhance Understanding**

The format of the program codes can help a reader understand the objective of the codes and how the objective is met. The indentation and spacing of program codes and statements can reflect good control structuring. Notice how unindented codes like this:

```
<%
If ( Counter >= 3 ) Then
Response.Write "Login Failed!"
Else
Response.Write "Try again!"
End if
%>
```

can be made better by using indentation and spacing, like this:

```
<%
    If ( Counter >= 3 ) Then
        Response.Write "Login Failed!"
    Else
        Response.Write "Try again!"
    End if
%>
```

Hence, all program codes and statements in E-Pack are appropriately indented and spaced to make reading these codes better.

- **Include Files**

Since the Web pages in E-Pack will be standardized to have the same design layout (e.g. same header, menu, and footer designs), there was therefore a need to create **templates** in order to minimize repetitious codes on the ASP pages. These templates are built using sets of include **files**, which can consist of any ASP, HTML, or text files. One include file can be used in multiple ASP pages, and maintenance is made easier as changes are only made in one place, that is the include file itself. Include files are inserted into ASP pages as follows:

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

- **Form Validation**

  Since this system will involve lots of data entry, form validation controls were therefore necessary to guarantee the correctness of the data and data-type entered into the system by the user. This was mostly done by using JavaScript, and in some parts, using VBScript. Users will be prompted to enter all required fields on the specific page before they are allowed to continue to the next process. This is to ensure that all essential information is entered.

- **Error Handling**

  Error handling is used to handle unexpected events in the program. When an error has occurred, the program will usually prompt the programmer about that specific error. The programmer will then be able to debug the program. Debugging is the process of modifying part of the program that does not work properly.

  Unfortunately, InterDev offers only limited error handling capabilities. All errors on InterDev are handled with the simple "`On Error Resume Next`" statement placed at the top of each ASP page, which in turn, does not give the programmer much error handling capabilities.

  In the development of E-Pack, much consideration has been given in error handling, as it is not appropriate to display critical error messages to the user. As such, in the occurrence of errors, the user will be directed to an error page (error.htm) to prevent them from viewing the error message generated by IIS.

5.3.2 Database Manipulation

The system needs to interact with the "EPack" database stored in the SQL Server. As such, a suitable database manipulation language was needed to accomplish this task. E-Pack is using ADO or ActiveX Data Objects, together with SQL (Structured Query Language) to communicate with the database. Below are some sample codes for connecting to the database and opening a recordset:

```
<%
    ' create the connection
    Dim conn
    Set conn = Server.CreateObject("ADODB.Connection")
    conn.Open "DRIVER=SQL Server;DATABASE=EPack;" & _
              "UID=sa;PWD=sa;SERVER=localhost"

    ‘ create the recordset
    Dim rs
    Set rs = Server.CreateObject("ADODB.Recordset")

    ‘ opens the recordset
    Dim sql
    sql = "SELECT * FROM Products″
    rs.Open sql, conn, adOpenKeyset, adLockOptimistic
%>
```

The below codes will display all the records in the table (with *n* records) together with their corresponding fields.

```
<%
    rs.MoveFirst
    Do While Not rs.EOF
        Response.Write rs(fieldname_1)
        Response.Write rs(fieldname_2)
        Response.Write rs(fieldname_3)
            .
            .
            .
        Response.Write rs(fieldname_n-1)
        Response.Write rs(fieldname_n)
        rs.MoveNext
    Loop
%>
```

In the above code, the program will loop until the **EOF** or end-of-file criterion is met. This means the recordset has reached the end of the table and there are no more records left in the table to be displayed.

5.3.3 Graphics Design

Several graphic tools were used to design the images on the Web site of E-Pack. They are Adobe Photoshop 5.0, Microsoft Paint, and Animagic GIF Animator 1.21. Photoshop and Paint were used to create GIF images, where else Animagic was used to animate the GIFs to give them a livelier appearance on the Web site.

Besides that, the navigation buttons at the top of each Web page in the shoppers' module were given **MouseOver** events in order to give the extra spice to the Web site. MouseOver events are actions that take place when the user places the mouse over a specific image or button. Macromedia Dreamweaver 3.0 was used to write these MouseOver events.

5.3.4 Cascading Style Sheets

To make the Web site appear more interesting, **Cascading Style Sheets (CSS)** were used so that the Web pages can have a standardized appearance. A CSS is a separate file containing style rules, to which Web pages can link. CSS either alters the default actions of existing HTML tags, or contains user-defined CSS **classes**. If the fonts on the Web site are to be changed, there is actually no need to search through each page for modifications, if CSS is used. All that needs to be done is to alter the style sheet, and changes will take place globally.

To use CSS on the Web pages, the following tag was inserted into the <HEAD> tag of each ASP page:

```
<HTML>
<HEAD>
   <LINK rel="stylesheet" type="text/css" href="style.css">
</HEAD>
</HTML>
```

5.3.5 Program Documentation

Program documentation is a set of written description of a program that explains to a programmer or reader what a specific program does, as well as how it was done. It is equally important to include internal documentation (program comments, etc.), as it is to include external documentation (user manuals, help files, etc.).

Additional comments in the program codes will help the programmer to understand how the program was implemented. Other than providing a line-by-line explanation of what the program is doing, the comments can also break the code into phases that represent major activities. In the development of E-Pack, suitable comments were inserted into the program codes for easier reference of the system in the future.

Manuals and guides were also written to facilitate the usage of the system. These external documentations were a must to guide the administrator in setting up and managing the system. Without these documentations, the user might take a little bit longer to get familiar with the system. As such, setup guides and user manuals were prepared to make it easier for these users in using the system.

CHAPTER 6: SYSTEM TESTING

6.1 Introduction

In the testing stage, tests were done on the system to ensure that the final system functions as it is supposed to. Many types of testing have been performed to make sure that the system is developed according to its specification, and every function implemented in the system works correctly. As such, three types of testing techniques were used to test the effectiveness of E-Pack. They are unit testing, module testing, and system testing.

6.2 Unit Testing

Unit testing is the basic technique used for the entire testing process. It involves performing independent tests on units (or functions), and making sure each unit functions as it is supposed to. The unit testing technique was used to ensure that any bugs arising from units within a module are fixed without causing any side effects to the module, and to the system at whole. After one new module was developed, tests were carried out independently in order to assure their accuracy, and to find any faults arising from the modules.

The strategy employed for unit testing is the **code reviewing strategy**. Using this strategy, the ASP codes were examined line by line in order to reveal any bugs that might have occurred during the implementation phase of the system. In reviewing the codes, the correctness of each line of code was identified by comparing it to the original design of the program flow. When the logical flow of the program was identified, the codes were commented so that they could be traced in the future.

The code was also examined and debugged in order to identify any fault coding. However, debugging the ASP codes was difficult, as there were no proper ASP debugger and tester available in Visual InterDev. As such, the "Response.Write" command was inserted into the ASP code to examine the value of each variable. For example:

```
<%
   Dim userID
   userID = Request.Form ("loginID")

   If userID = "admin" Then
      Session ("AdminLogin") = True
   End if
%>
```

For the above codes, there is no way to "watch" the value of the userID and Session ("AdminLogin") variable using InterDev. Therefore, these codes were debugged like this:

```
<%
   Dim userID
   userID = Request.Form ("loginID")
   Response.Write userID        ' print out the value of userID

   If userID = "admin" Then
      Session ("AdminLogin") = True

      ' print out the value of Session ("AdminLogin")
      Response.Write Session ("AdminLogin")

   End if
%>
```

By using the "Response.Write" command, the value of the variables in the ASP codes could be printed on the browser. Thus, this method was used to 'watch' the value of these variables. However, if the program code is only a simple code, and was written with confidence, there is actually no need to test the codes using this method, as it is just redundant work.

After reviewing the codes, relevant comments were written on top, or by the side of the corrected codes for future references and enhancements.

6.3 Module Testing

After the unit testing, the module testing was carried out to check the interactivity between each unit within a specific module. This test was essential to make sure that all units within a module could work together without any conflicts. If any errors are present from a particular unit, the part of the module that went wrong could be identified, and the unit testing technique would be used again to identify these errors.

For example, the "Add New Product" function was tested together with the "Edit Product" function within the administration module to ensure that both functions could interact properly once they are integrated into the system. To do this, fabricated product records (or dummy data) were added into the database by using the "Add New Product" function. Later, the system was tested on whether the records of the just-entered products could be modified through the "Edit Product" function or not.

The module testing was carried out on all the units/functions within the shoppers' and administrator's module of E-Pack.

6.4 System Testing

After all the modules were believed to have satisfied the requirements, the system was finally tested as a whole to ensure that it runs smoothly without any interruptions. The system was tested in a testing environment (local intranet), where the number of PCs connected to the server is limited to less than ten units. The program flow and the testing needs for each of the modules were identified before the system was further reviewed and tested.

After that, the entire system was tested with some **test cases**. Using this method, several data entries and order transactions were performed on the system to test the reaction of the system towards the user's input. For example, to test the authenticate login function, several combinations of login IDs and passwords were entered into the system to test the login authentication process. To test the order making abilities of E-Pack, various products were chosen and placed into the shopping basket, and later were purchased at the checkout counter. Later, these orders were retrieved using the administrator's module of the system to see whether these orders were saved into the database or not. With this, the reaction of the system towards the user's input could be tested. This could identify the system's faults, which would probably arise in normal conditions.

The performance of the entire system over the intranet was also tested to ascertain the system's true reliability and efficiency. To check the accessibility of the system over the intranet, other PCs in the lab were used to access the server where the system was hosted. In addition to that, several PCs were used to access the system **simultaneously** in order to test the scalability of the system.

Finally, the system was published on the intranet to allow other users test it. This is probably one of the most effective ways of testing the system, as errors and bugs usually appear through regular, daily operations. In addition, independent comments on the system could be obtained from the users through this method of testing.

CHAPTER 7: SYSTEM EVALUATION

7.1 Introduction

System evaluation is the post-implementation review to determine the strengths and limitations of the system. This assessment is essential in providing the necessary information for future project enhancements. It also highlights the knowledge obtained from this project, and identifies the problems encountered in the development of this system, as well as the steps taken in solving these problems.

7.2 System Strength

7.2.1 Easy Deployment

E-Pack was developed on the Windows platform, the most popular and widely used operating system in the world. Therefore, incompatibility issues are least likely to arise, as it is easily deployed into any machine running on the Windows platform.

7.2.2 Practical Functionalities

E-Pack is packed with functions that are practical and easily understood. They are kept as simple as possible in order to reduce user reaction, and at the same time minimizing errors. Functions that are not supposed to be in the system are left out. In short, only important functions are included in the system. This was necessary to keep the system simple and straightforward, as well as not to confuse the user with complicated functions. Thus, the system remains easy-to-maintain.

7.2.3 User-Friendly Interfaces

Each function in E-Pack is arranged systematically at the top of each Web page, as well as on the menu at the side of each page, to give the Web site a structured appearance. This makes it easy for users to get along and be familiar with E-Pack. The Web site is also designed to be pleasant to the eyes. The colors used on each page are standardized and carefully coordinated, so as not to confuse the users with too many colors. Graphics and images are used where possible in order to liven up the Web site. However, animated graphics are minimized so as to give the Web site a clean, professional look, as well as to reduce the time needed to load the Web pages.

7.2.4 Simple Searching Capabilities

One of the plus points of E-Pack is its ability to search the database for text entered by the shopper in the search box. This reduces the time needed by the shopper to look for the items they need from the Web site.

7.2.5 Mobile Administration

Users can access the administration page from anywhere in the world, just as long they are connected to the Internet. Once in the administration page, they can add, edit, or delete the data stored in the database. However, only authorized users are allowed to access the administration page of the system. As such, users will be prompted to enter their login ID and password before given access to the administration page.

7.3 System Limitations

Despite its strengths, there are some limitations to the system. These limitations were mainly caused by the time, cost, and knowledge constraints of the project. The limitations of E-Pack are:

7.3.1 Lack Of Animation

Due to the fact that the Web pages in E-Pack were designed to load in the fastest time possible, it cannot contain too many images, especially heavily animated graphics. As such, this system still lacks eye-catching animated graphics, which makes the system not very graphically appealing to the users. As a result, some users might find the Web pages a little bit dull.

7.3.2 Inadequate Searching Capabilities

The system only possesses simple searching capabilities and is not capable of searching based on complex combinations of keywords. As such, it can only search the database for text that exactly matches the entered text in the search box.

7.3.3 No Mailing Capabilities

E-Pack does not have any automatic mailing capabilities for the time being. As such, important notices from the system had to be e-mailed one-by-one to the shoppers by the system administrator, and this may be a tedious task.

7.3.4 Manual Purchase Orders

Once the shoppers' orders have been processed, Purchase Order forms need to be sent to the corresponding retailers to attain the requested products. These Purchase Orders will have to be created manually, and as such, may take up a lot of precious time.

7.3.5 Payment Through Physical Medium

Currently all shoppers' payments are made through the physical medium (e.g. cheque, money order). This leads to payment **floats**, as there is a time lapse between the moment when the cheque is posted, and the moment when the bank clears the cheque. This will cause payments to be received at a date later than the date the orders were made.

7.3.6 Unsecured Transactions

All transactions on E-Pack are performed over an unsecured line. Data sent through an unsecured line is vulnerable to surfers from all over the world and as such, encourages eavesdropping and hacking. As a result, data could easily be tapped and modified by the perpetrator, which leads to unauthorized intrusions that may jeopardize the entire system.

7.4 Future Enhancements

Because E-Pack has certain weaknesses, future enhancements are therefore necessary:

7.4.1 Graphics Animation

It is recommended that more graphics animation be included in the future development of E-Pack in order to create a more lively appearance. However, it should be taken into consideration that heavily animated graphics will inevitably slow down the loading time of each Web page. Therefore it is advisable to keep graphics at its minimum.

7.4.2 Search Engine

In order to enhance its functionalities, a more potent search engine is required for the system. This search engine should be able to search the database for text based not only on one keyword, but based on two or more combinations of keywords. This will provide the shoppers with a more flexible search engine capable of multiple keyword searches.

7.4.3 Mailing Module

A mailing module should be incorporated into the system to enable messages and notices from the system to be automatically generated and sent to the shoppers through e-mail. This will certainly decrease the time needed to convey important messages to the shoppers, as well as reduce the effort needed from the administrators to e-mail these messages to the shoppers.

7.4.4 Purchase Order Module

The system should also have a module that can generate all the Purchase Order forms automatically. This will eliminate the need to prepare a Purchase Order manually as well as decrease the possibility of making mistakes on the Purchase Order forms. Other than Purchase Orders, this module should also be able to generate other supporting documents, such as Goods Received Note, Delivery Order, Sales Order, Invoices, Receipts, and so on.

7.4.5 Credit Card Payment

The future development of E-Pack should also include a credit card payment system to facilitate future transactions. A credit card payment system will definitely speed up the order-processing phase by eliminating payment floats. Additionally, payment by credit card is a good way to ascertain whether the shopper is a genuine buyer or not.

7.4.6 Security Protocols and Cryptographic Controls

The system also deserves better security controls in order to improve its reliability. As such, future developments of E-Pack should consist of the usage of security protocols, such as the **Secure Socket Layer (SSL)** in order to secure the line used to perform the transactions on E-Pack. Nevertheless, the implementation of security protocols should not be taken for granted as 100% secured, because there is still a chance for unauthorized intrusions somewhere in the world. Therefore, there should be efforts to implement **cryptographic controls** to protect the privacy of data and to prevent unauthorized modifications of data in the system.

7.5 Problems and Solutions

During the development of this project, several problems were encountered. Nevertheless, solutions were quickly brewed up to counteract these problems:

7.5.1 Doubt In Choosing Development Tools

Choosing the right tools to develop the system was a difficult task because of inexperience and unfamiliarity with the new programming tools.

Solution:

Discussion with supervisor and software testing has equipped me with some guidelines on choosing the suitable tools for the system.

7.5.2 Difficulty In Figuring Out Business Logic

Once the development tools were chosen, came the issue of the system design. It was a challenging task trying to figure out the correct business logic for the system, as the concept of e-commerce was still rather new to me.

Solution:

Discussion with supervisor and fellow course mates, as well as visiting famous Web sites of the similar nature, such as Amazon, and Barnes and Nobles, has improved my understanding on the system.

7.5.3 Insufficient Programming Knowledge

The biggest problems faced during the development of this project was the lack of knowledge and experience in using programs like Microsoft Visual InterDev, and programming languages like ASP, ADO, VBScript, and JavaScript. Several times I was caught in a situation where I was at a complete loss.

Solution:

Discussion with supervisor and fellow course mates, as well as conducting research on reference books and the Internet, has helped me in continuing the development of the system.

7.5.4 Lack Of Graphic Design Skills

A lot of time was taken up to design the graphics needed on the Web site of the system, as I had very little graphic designing knowledge. Furthermore, the mouse device is not an appropriate tool for graphic designs.

Solution:

More efforts were spent practicing how to design these graphics using graphic designing programs such as Adobe Photoshop and Microsoft Paint.

7.5.5 Problem With PCs In The Lab

Most of the personal computers (PC) in the lab were not fully operational. Some of the PCs were infested with virus, and some were just not operational (corrupted software). Some PCs did not have the required software at all. Incidentally, some of the PCs had hardware problems and could not be rebooted. Furthermore, the servers in the lab were initially not operational and therefore, no connections could be established to these servers.

Solution:

The assistance of the System Administrators was requested to rectify these problems.

7.6 Knowledge and Experience Gained

Throughout the development of this system, valuable knowledge and experiences have been acquired. They are:

- A golden opportunity to get acquainted with new software, such as Windows NT, Visual InterDev, SQL Server, and Macromedia Dreamweaver
- A chance to learn new programming languages, such as ASP, ADO, VBScript, and JavaScript
- An opportunity to fully utilize all the theoretical and practical skills learnt from our IT degree course
- Obtained a hands-on experience on how to plan and develop a system
- Discovered the importance of having good time management
- Learnt how to be independent and resourceful
- Sharpened up skills on problem solving
- Learnt how to handle work pressure
- Brushed up skills on documentation
- Discovered the importance of setting priorities in life

7.7 Conclusion

E-Pack is a Web host that provides an affordable, innovative, and effective way for small retailers to market their products and services on the Internet. It is also a Web site for convenient home shopping, as shoppers no longer need to leave the comfort of their homes in order to shop. Administration tasks are also made easy with its simple and practical administration page.

Due to certain limitations, such as time, cost, and knowledge constraints, there is only so much that can be done on the system. Therefore, it is highly recommended that future enhancements be performed in order to further improve the functionality of this system. Future enhancements will guarantee that E-Pack will continue to provide more exciting features, as well as improve its reliability and security for many years to come.

In my opinion, undergraduates should appreciate this subject (WXET 3181 & 3182) as a learning process that gives them the opportunity to develop and improve their skills in designing, implementing, and managing a project under the guidance and supervision of a supervisor. Moreover, this subject also serves as a channel for students to apply both their theoretical and practical skills into a strictly time-budgeted project.

Judging from the overall results, it can be concluded that E-Pack has achieved and fulfilled all its stated objectives successfully!

REFERENCES

[Alwang, 1999]

Alwang, Greg. *Microsoft Internet Information Server*. PC Magazine, 1997.

http://www8.zdnet.com/pcmag/features/webserver/iwsr5.htm

[Apache, 1999]

Apache Server Frequently Asked Questions. Apache Group, 1999.

http://www.apache.org/docs/misc/FAQ.html

[ASC, 2000]

Active Server Pages: General FAQ. Active Server Corner, 2000.

http://www.kamath.com/faqs/aspgen_faq.asp

[Bakken et al, 2000]

Bakken, Stig Saether et al. *PHP Manual*. PHP Documentation Group, 2000.

http://www.php.net

[Ball & McCulloch, 1996]

Ball & McCulloch. *International Business: The Challenge of Global Competition*, 6th Edition. Irwin, 1996.

[Cerf, 2000]

Cerf, Vinton G. *A Brief History of the Internet and Related Networks*. The Internet Society, 2000.

http://www.isoc.org/internet-history/cerf.html

[Chappell & Linthicum, 1997]

Chappell, David & Linthicum, David S. *ActiveX Demystified*. CMP Media Inc., September 1997.

http://www.byte.com/art/9709/sec5/art1.htm

[Chua, 2000]

Chua, Clarence. *Big earnings in e-commerce*. The Star, 7th July 2000.

[CNEC, 1997]

A Brief Insight in Electronic Commerce. Competence Network Electronic Commerce (CNEC), 1997.

http://www.cnec.org/ecinsight/insight.htm

[CommNett, 2000]

SSL vs. SET. CommNett Caribbean Ltd., 2000.

http://www.commnett.com/features/ssl_set.htm

[Duncan, 1998]

Duncan, R. *Formatting World Wide Web pages with HTML tags*. PC Magazine, October 1998.

[Dyck, 1999]

Dyck, Timothy. *Microsoft SQL Server 7.0*. PC Magazine, July 1999.

http://www.zdnet.com/pcmag/stories/reviews/0,6755,408776,00.html

[ECIC, 1998]

An Introduction to Electronic Commerce. Electronic Commerce Innovation Center, 1998.

http://www.cf.ac.uk/masts/ecic/eleccomm.html

[Future Systems, 1999]

Getting Started with ColdFusion 4.0. Future Systems, 1999.

http://futuresystems.com/cfdocs/Getting_Started_with_ColdFusion/1_ColdFusion_Overview

[Gadey, 1996]

Gadey, Venkatesham. *Client/Server FAQ*. CS527, Department of Computer Science, Illinois Institute of Technology, 1996.

http://danada.rice.iit.edu/~cs527/clinserv.html

[Hayman, 2000]

Hayman, Michael. *Windows NT 4.0: Will NT become the world's most popular operating system?* Internet.com Corporation, 2000.

http://www.winplanet.com/winplanet/reviews/568/1/

[Herel, 1997]

Herel, Heath H. *Apache*. PC Magazine Online, 1997.

http://www8.zdnet.com/pcmag/features/webserver/iwsr1.htm

[Holzner, 1998]

Holzner, Steven. *Visual Basic 6 Black Book*. Scottsdale, Arizona: Coriolis Technology Press, 1998.

[IDM, 2000]

JavaScript FAQ: What is JavaScript? Internet.com Corporation, 2000.

http://idm.internet.com/faqs/jsfaq/gen1.html

[Kendall & Kendall, 1999]

Kendall, Kenneth E. & Kendall, Julie E. *Systems Analysis and Design*. 4th Edition. Upper Saddle River, New Jersey: Prentice-Hall Inc., 1999.

[Kosiur, 1997]

Kosiur, David. *Understanding Electronic Commerce*. Redmond, Washington: Microsoft Press, 1997.

[Kroenke, 1998]

Kroenke, David M. *Database Processing: Fundamentals, Design, and Implementation*. 6th Edition. Upper Saddle River, New Jersey: Prentice-Hall Inc., 1998.

[Lam, 1997]

Lam, John. *Visual InterDev and ASP*. PC Magazine, December 1997.

http://www.zdnet.com/pcmag/pctech/content/16/22/pp1622.001.html

[Leiner, 2000]

Leiner, Barry M. *A Brief History of the Internet*. The Internet Society, April 2000.

http://www.isoc.org/internet-history/brief.html

[Linux, 2000]

Linux Online - About The Linux Operating System. Linux Online Inc., 2000.

http://www.linux.org/info/index.html

[Lipschutz, 1999]

Lipschutz, Robert P. *ColdFusion 4.0*. PC Magazine, January 1999.

[Microsoft, 2000]

An ASP you can grasp: The ABC's of Active Server Pages. Microsoft Corporation, 2000.

http://msdn.microsoft.com/workshop/server/asp/ASPover.asp

[Muhtar, 2000]

Muhtar, Fauziah. *Online Security with MEPS' SET payment*. Computimes, New Straits Times, 27th July 2000.

[NCSA, 1996]

Common Gateway Interface. National Center for Supercomputing Applications, University of Illinois, 1996

http://hoohoo.ncsa.uiuc.edu/cgi/intro.html

[Nightcats, 1999]:

Beginners Guide to E-Commerce. Nightcats.com, 1999.

http://www.nightcats.com/sales/free.html

[Pfleeger, 1998]

Pfleeger, Shari Lawrence. *Software Engineering: Theory and Practice*. Upper Saddle River, NJ: Prentice-Hall Inc., 1998.

[Powers, 2000]

Powers, Shelley. *PerlScript: A Hot Ingredient for ASP*. CMP Media Inc., 2000

http://www.webtechniques.com/archives/2000/05/powers/

[Quaterman, 1994]

Quaterman, John S. *The Internet Connection: System Connectivity and Configuration*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1994.

[Reynolds, 2000]

Reynolds, Matthew. *Beginning E-Commerce with Visual Basic, ASP, SQL Server 7.0 and MTS*. Birmingham, UK: Wrox Press Ltd., 2000.

[Sellapan, 1999]

Sellapan, P. *Access 2000 Through Examples*. Shah Alam: Times, Federal Publications Sdn.Bhd., 1999.

[Shimar, 2000]

Shimar, Ada. *Ada's Introduction To CGI*. 2000.

http://adashimar.hypermart.net/

[Sim, 1999]

Sim, Phee Lip. *Offline E-Business Solution*. Kuala Lumpur: Faculty of Computer Science and Information Technology, University of Malaya, 1999.

[Sommerville, 1998]

Sommerville, Ian. *Software Engineering*. 5th Edition. Harlow, England: Addison-Wesley Publishing Company Inc., 1998.

[Varadarajan, 2000]

Varadarajan, Premkumar. *What is COM?* Active Server Corner, 2000.
http://www.kamath.com/columns/comsutra/cs001_whatiscom.asp

[WAPF, 2000]

WAP – Frequently Asked Questions. Wireless Application Forum Ltd., 2000.
http://www.wapforum.org/faqs/index.htm

[Weber, 1999]

Weber, Ron. *Information Systems Control and Audit*. Upper Saddle River, New Jersey: Prentice-Hall Inc., 1999.

[WIMC, 1999]

Wireless E-Commerce: A New Business Model. Wireless Internet & Mobile Computing, 1999.
http://home.earthlink.net/~aareiter/wireless2.htm

[Wrox, 1997]

A Tutorial in VBScript. Wrox Press Ltd., 1997.
http://idm.internet.com/corner/wrox/progref/vbt/ch21_15.shtml