ON-LINE BUSINESS INFORMATION SYSTEM OF THE COMPUTER SOLUTION COMPANY

by

Mr. Chiraseth Dheeradhada

A Final Report of the Three-Credit Course
CE 6998 Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer and Engineering Management
Assumption University

November, 2000

ON-LINE BUSINESS INFORMATION SYSTEM OF THE COMPUTER SOLUTION COMPANY

by
Mr. Chiraseth Dheeradhada

A Final Report of the Three-Credit Course
CE 6998 Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer and Engineering Management
Assumption University

November 2000

Project Tile	On-Line Business Information System of the Computer Solution Company
Name	Mr. Chiraseth Dheeradhada
Project Advisor	Mr. Smith Tungkasmit
Academic Year	November 2000

The Graduate School of Assumption University has approved this final report of the three-credit course, CE 6998 PROJECT, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer and Engineering Management.

Approval Committee:

(Mr. Smith Tung t)
Advisor

(Prof. Dr. Srisakdi Charmonman)
Chairman

Chamnong
(Dr. Chamnong Lin irapanich)
Dean and Co-advisor

(Asst.Prof.Dr. Boonmark Sirinaovakul)
Member

(Dr. Prapon Phasukyud)
Member

(Assoc. Prof. Somchai Thayarnyong)
MUA Representative

November 2000

ABSTRACT

This on-line business information system project is developed to discover the way to reduce operation of web server for on-line business. The project emphasized on creating a web site for on-line business information system. The project studies and collects data on on-line business information system deriving from theories, practice, and praxis of previous and current researches and companies' operations. The ordering system's requirements are determined. The database of the system is developed and designed based on a Relational database approach. The system is designed according to requirements. Microsoft Access is used to develop the database because it helps reduce cost of developing the database for a new company. The web site is designed and developed by using ASP as the core language.

Finally, the outcome of the project is the way to reduce operation of web server for on-line business by using ASP as the core language.

ACKNOWLEDGEMENTS

This project is completed by the encouragement of Mr. Smith Tungkasmit for his individual expertise, which is very helpful. His patient assistance, guidance, and constant encouragement have led me to complete this project.

I would like to express appreciation to all my teachers who gave education, which enables me to complete this project.

I also appreciate all my friends for their encouragement.

Special appreciation is due to my family who encourage me to study at the Graduate School of Assumption University, and who always give valuable support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

I. INTRODUCTION

1.1 Background of the Project

Nowadays, emerging information and computer technology are growing and improving at an incredible pace. The information age has unwittingly changed not only the perception of our world but also the environment in which we live. No one is immune to the effect of the emerging information economy in all aspects of our lives--commercial, communication, education and entertainment. The basic goal of a user-centered technology is to act as a tool to help us accurately and precisely manage complex and demanding tasks faster and easier in a safer environment.

At present, the Internet has already integrated into the fabric of our everyday life. Through the use of an Internet intelligent agent, we are able to seek faster broader, and deeper, information from around the globe and beyond. The Internet is widely used effectively and efficiently in business, government, and personal use. For business, the use of the Internet can reduce cost, improve communication, and enhance business operations. In a marketing context, electronic commerce provides a convenient way to promote and/or sell products and services.

Active Server Pages is a technology of Microsoft that provides the capability for the web server to process application logic and then deliver standard HTML to client browser. ASP enables web server to deliver HTML, client-side script, web controls, and server-side processing and connectivity features. ASP is executed on the web server, so high performance server is required to support the use of ASP. This project is developed to solve this problem for on-line business.

Hence, this report is to try to develop a web program to sell products on the Internet using ASP as the core program and to find out another computer language, which will help to reduce the operation of the server for on-line business. In order to find out another computer language, it is necessary to develop a fictitious company. The company ComputerSolution.com included in this report is a fictitious company that provides computer products and services to end-users. The products sold include computer hardware components and operating systems. The supporting services are mainly telephone technical support related activities and these maintenance that cannot be done locally to ensure product quality and customer satisfaction. To provide a framework for emerging e-business in Thailand, a link between the company database and customers is needed. The focus of this report is on the online database operations, which involve suppliers, technical support, service center, product warehouse, computer solution database, online link, and customers. There are many suppliers who provide products to product warehouse. The responsibility of the product warehouse is to provide products according to customers' order. Customers can place their orders by using company web site or using company's agencies. Company web site targets are mainly on the Internet-users according to the report of NEC ILC source. At present, there are approximately one million people who are using the Internet. Forty-six percent of the Internet-users are students, and the remainders are ordinary-users. Most of the target group are people who have their own computers and Internet. People who do not have their own computers and access to the Internet can use the company's agency to place their orders. Note that the company's agencies are a supplement not a replacement to the business, the company's agencies enable the company to cover the target group that have no access to the Internet. In other words, the company's agencies can never replace the core operation of getting computer products to customers through the Internet.

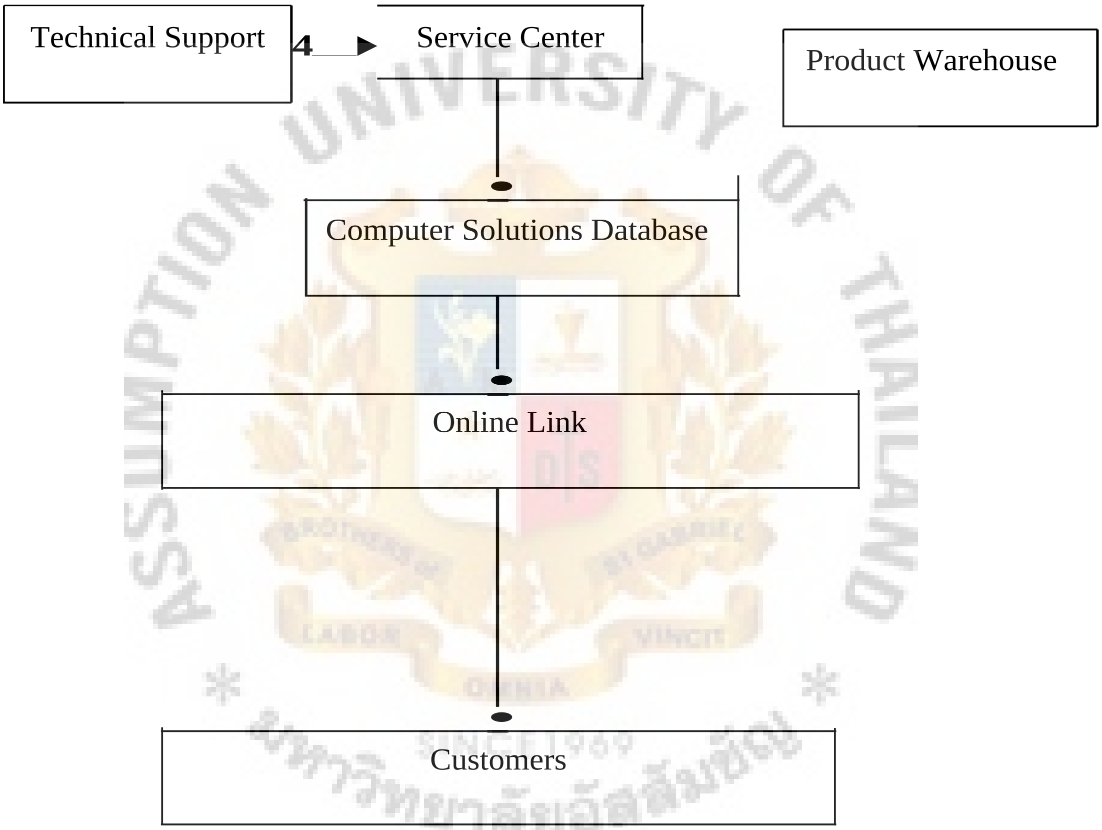


Figure 1.1. Customer Related Operations.

1.2 Objectives of the Project

(1) To design, develop, test, and produce a prototype of an on-line business information system of the Computer Solution Company.

(2) To design the system, which is expandable for future needs.

(3) To analyze hardware and software specifications appropriate for the ordering system of the Computer Solution Company.

(4) To develop the system which reduces operation of the web server.

1.3 Scope of the Project

(1) The scope of this project is to develop an on-line business information system for Computer Solution.

(2) The web program covers the on-line business information system of the company.

(3) The web program is designed for a Windows NT platform with programming codes based on ASP language and Java script.

(4) The web site is developed for Internet Explorer and Netscape Navigator only.

1.4 Research Methodology

(1) To study and collect data on on-line business information system deriving from theories, practice, and praxis of previous and current researches and companies' operations.

(2) To determine the ordering system requirements.

(3) To design ordering system: Database design, Input design, Output design, and security and control design.

(4) To develop a web site.

(5) To test the web site.

(6) To develop a plan for actual implementation of the project.

(7) To conclude and recommend this project.

1.5 Deliverables

(1) A database which is suitable for on-line computer business.

(2) Online business web site, which enables anybody to place the order.

(3) This web site can be applied to other organizations.

(4) The web site reduces the operation of the web server.

II. LITERATURE REVIEW

2.1 Active Server Pages

Active Server Pages is a new technology from Microsoft that provides the capability for the web server to process application logic and then deliver standard HTML to the client browser. Although fairly straightforward, this simple concept has great ramifications in shifting traditional applications to the Web.

The first direct result of using Active Server Pages is that only HTML is sent to the client's browser. ASPs do not automatically send ActiveX controls to the browser. By default, ASPs send only ASCII text to the browser This lets any browser type run on any operating systems to access the applications and workflow logic embedded in the ASP scripts.

Active Server Pages give you the ability to deliver more than just HTML. ASP enables you to deliver HTML, client-side scriping, Web controls, and server-side processing and connectivity features. ASP scripts can deliver client-side scripts, such as VBScript and JavaScript, to be executed on the client's browser. In addition, ASP can also deliver a wide range of Web functionality by acting as a transfer vehicle for ActiveX controls, Java applets, and other Web components. In addition, ASPs not only produce dynamic HTML depending on the client request, but also provide the capability to tap into existing systems, such as databases, document retrieval services, mail servers, groupware servers, and other COM-based information servers. ASPs now act as an HTML interpreter that was once only accessible through native interfaces, such a Microsoft Exchange client, Lotus Notes client, or a customized Visual Basic application. With the wide range of functionality, Active Server Pages and the Internet Information Server act as a medium for porting existing applications to-and building new applications for-the Web (Johnson 1997).

2.2 Static versus Dynamic Content

The World Wide Web has progressed from Web pages that couldn't support images to a global network capable of supporting full-scale applications. This transition is made possible through changing static information into dynamic, database-driven applications using logic executed on the Web server.

2.2.1 Static Content

The World Wide Web was initially created to share textual document around the world. These textual documents were static documents that served primarily as online reference materials. The term static is used to refer to these documents because the requesting user had no ability to interact with the content delivered from the Web server. These documents were converted from their original source type into Hypertext Markup Language (HTML). Once in HTML format, the document could be delivered from a Web server to any Web browser using the Hypertext Transfer Protocol **(HTTP).** This static approach to creating and reading Web pages and Web content is still used in many sites today to display personal information, corporate profiles, or online reference material that does not frequently change.

2.2.2 Dynamic HTML Content

The billboard approach, or the posting of static information on the Web, suited the needs for general information dissemination. However the Web user community started to require greater function from this vast information medium called the Internet. This need for advanced function using dynamic content eventually grew into the need to develop and deploy entire applications that used the World Wide Web as a global network. The term dynamic describes the process of creating HTML content depending on the information that is sent or submitted to the Web server. The Web server would process the information and convert the output into HTML. The resultant HTML, tailored for the specific user input and application settings, is then delivered to the browser.

2.3 Designing and Deploying Web-Enabled Applications: A Changing Architecture for Changing Needs

Developing Web-based applications relies on many network and application components working together to deliver information to the requesting client. In the once simple world of HTTP, Web browsers extracted information from Web servers, as shown in Figure 2.1. This Web- based architecture relied on existing network infrastructure and the TCP/IP protocol stack to broadcast, carry, route, and assemble in-transit data TCP/IP is not exclusively used with the World Wide Web; it is also used in other services such as Gopher, File Transfer Protocol (FTP), and Telnet.

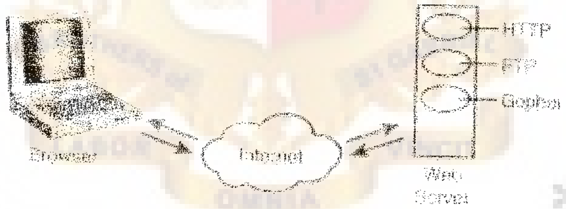


Figure 2.1. Web Browser Extracted Information from Web Server.

However after some standardization of HTML languages, image specifications, and protocol stacks, the world slowly began to see the Intranet for what it is a global network of computers. This band of network computers is owned by a combination of public and private localized networks. These networks consist of private and public organizations and government organizations that pay and maintain these backbones.

Although there is not a centralized organization unit for controlling the Internet itself, there are several organizations that propose standards and guidelines on how the Internet should be used. For example, the Internet Activities Board (IAB) is responsible for reviewing Internet networking issues, such as architecture and TCP/IP and domain name issues. The World Wide Web Consortium (W3C) suggests guidelines and standardization for the World Wide Web.

With so many forces influencing how the Internet behaves and operates, utilizing this global network in the same way you use your own private network presents many challenges. To transform the global network into a reliable application platform, the original architecture of the Web must be enhanced to meet many needs that we take for granted when developing traditional applications. The enhanced architecture to develop and deploy your Web-enabled applications can be seen in Figure 2.2.

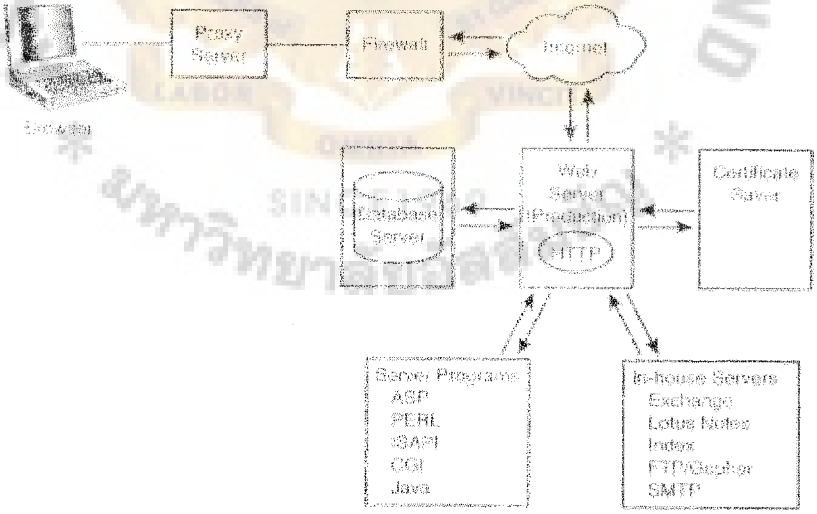


Figure 2.2. Architecture of Web-Enabled Application.

As you can see, not only is there a dramatic shift in the number of components now needed to create, develop, and deploy applications, but there is also a fundamental shift in the method for developing applications. For the past several years, the shift to client/server architecture has crept into IT shops around the world. The client/server world relies on the processing power found on the client machine to off load server requests and processing loads.

However, the benefits of distributing the processing onto the client in traditional client/server applications are quickly overshadowed by the amount of manpower needed to deploy and update applications. As can be seen in Figure 2.2, the Web-based architecture represents a shift back to the server-centric deployment model. This model enables a centralized deployment and distribution mechanism without relying on individual client configurations. Furthermore, the server-centric model also enables a central connection point to external resources, such as in-house data stores like Microsoft Exchange, Lotus Notes, or your own custom-built proprietary data systems. Let's take a closer look at the different components needed in the new Web architecture paradigm.

2.3.1 The Web Browser

The Web browser provides a graphical, text-based terminal interface to the Web server. This terminal approach provides an interface between the user and the Web server. The Web browser is responsible for translating HTML sent by the Web server into a graphical user interface within the browser.

The choice of Web browser is often directly related to the needs of the targeted user. For example, if one design goal of your Web application is to reach the largest user group, your application should rely only on HTML standards, which can be used to deliver information to all browsers. However, if your targeted user group is somewhat more defined, you can design your solutions to take advantage of the special characteristics of the planned user community. For example, if you need to build an Intranet application in which all desktops are running Windows 95 and NT operating systems, you can use ActiveX controls that run in Internet Explorer. If your application assumes the use of Internet Explore 3.0 or Netscape Navigator 3,0 and up, an HTML-only or HTML and Java applet application can be deployed. In most situations, you will find that defining your targeted user community will dictate the look and feel of your applications and Web browser

The Web browser can also use client-side scripting to perform tasks and operations within the browser. Scripting languages such as JavaScript, JScript, and VBScript are used to reduce server processing and network bandwidth by performing actions such as field validation or calculations within the requesting browser. For example, you can create a mini-online help system by displaying a help message when the user moves the mouse over a hyperlink.

2.3.2 The Web Server

The Web server has several responsibilities that all center around delivering HTML to the requesting client browser. The traditional billboard or information-sharing approach, which initially created the need for the World Wide Web, quickly demanded more functions from the user community. The Web servers were soon able to process executable scripts that gave needed functions and connections to other systems. The capability to process and establish these connections to other server-side component is made possible through program execution on the server from various executable sources. Some of these sources include the Common Gateway interface (CGI). ISAPI, PERL, REXX, Java, VBScript, and JScript. All of these processes are aimed at developing application logic that is processed on the server. The server-side scripts, either acting as stand-alone applications or embedded in Active Server Pages, can be used to transform the Web server into a gateway that exposes information stored in other servers. In particular, the Internet Information Server can access information from database servers, mail and news servers, or any other COM-based server, such as Lotus Notes or Microsoft Exchange.

2.3.3 HTTP TCP/IP Connection

One of the most amazing features of the Internet is that all the Internet technologies are based on sending and receiving simple ASCII text. The ASCII text is transferred via TCP/IP. TCP/IP is responsible for wrapping or packaging data into smaller packages and sending the information across the network. These smaller packages each have a destination address that is used to route and re-route the packages. Therefore, packets can take a variety of approaches to reach the destination depending on network loads. When the data packets are received at the destination, the data is re-assembled back into a complete document. TCP/IP also performs error checking to ensure that no data packets were lost in transit TCP itself is a connection-based protocol, which is responsible for establishing a communication session between a client and server.

The HTTP service runs as an application on the Web server, and is responsible for managing the Hypertext Transfer Protocol. HTTP is a client/server protocol that accepts requests and delivers requests via HTML. However, unlike most traditional client/server applications such as FTP and Gopher, once HTTP delivers the information to the requesting client, the connection or persistence is released. This feature was built into the original design of HTTP. HTTP was designed to accept and handle large numbers of client requests. To maximize the number of potential clients, HTTP typically releases the connection after the information is delivered by the server to minimize the number of busy or concurrent connections.

Services such as FTP and Gopher, however, keep the connection between the client and server open until the session is terminated. If a user has an open connection to the server but is not transmitting any data, the connection is still kept open, possibly preventing other users from connecting to the server. Fortunately, the inherent drawbacks of the connectionless HTTP protocol can be avoided (depending on the client configuration) by using the HTTP KeepAlive request.

The HTTP KeepAlive request maintains a TCP connection between the client and server (if both the server and client's browser are configured to use this feature). The importance of the KeepAlive function arises when you need to ensure that information delivered from the server is completely transmitted to the requesting browser. When the client receives the complete data stream, a trigger indicating a completed transmission is returned to the server. The HTTP KeepAlive also enables multiple requests over one connection. Because every item on a Web page generates individual server requests, the KeepAlive session will increase performance by using one concurrent connection to the HTTP server instead of having to spend resources making separate connections.

By itself, the HTTP service is incapable of directly interfacing with the network interface to exchange data. The Web server relies on Windows Sockets and the Transportation Layer and Internet Protocol (IP) to provide the communication layer between the HTTP service and the network interface, as shown in Figure 2.3.

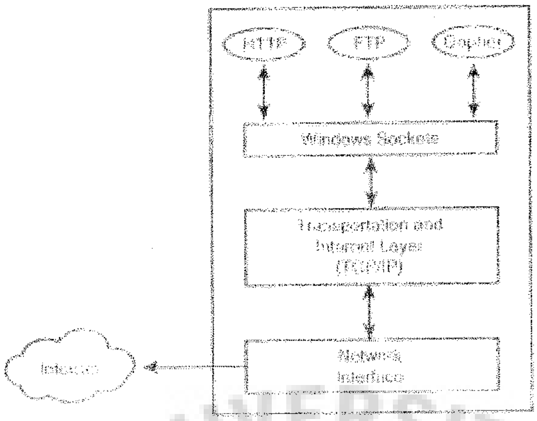


Figure 2.3. Communication Layer.

2.3.4 Database Server

The database server plays a vital role in Internet application development. The database server can be used to store, search, and retrieve information that is stored in a database. This same database that distributes information to Web users can also be accessed and maintained from within your corporate walls. For example, a database can be used for an online class registration system where users can search the database for classes they are interested in and register for classes via the Web. The administration department can then access this same information to process the new class registration from either a traditional application built with a tool such as Visual Basic, C++, Delphi, or PowerBuilder, or it can access the information via a browser interface.

When using the HTTP server as a connection utility to database servers, there is a slight shift in architecture that differs from traditional application development. For example, suppose you are creating an online classified section for your local newspaper and you want to create a search mechanism that searches for all vehicles of a particular type. From an architecture perspective, this application will have three components: the requesting browser the Web server, and the database server (see Figure 2.4.). The browser is responsible for submitting. query requests and displaying the results from the database. The Web server is responsible, accepting the query from the browser, creating a connection to the database, querying the database, formatting the results into HTML, and delivering the HTML to the requesting browser. The database server is responsible for accepting requests from the Web server and delivering results back to the Web server.

Notice that a database connection is not made directly from the browser to the database server. Instead, the database connection is established only between the Web server and the database server. As a result, the Web server-not the Web browser-acts as the client to the database server. This point is particularly important when you create and administer permissions to your database server. The Internet Information Server automatically creates an NT user account named IUSR_Computername to accept anonymous requests from browsers. In order for you to create a connection from the Web server to the database server, the IUSR_Computername user account must have permission on the database server.

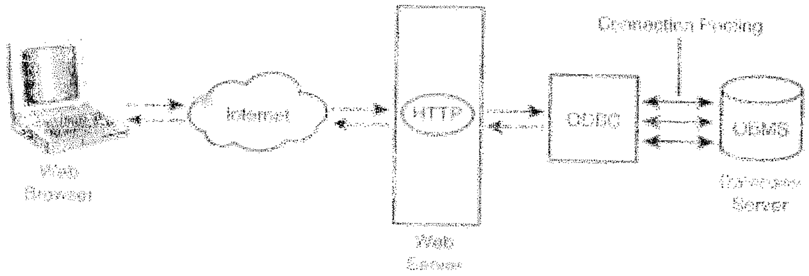


Figure 2.4. Requesting Browser the Web Server and Database.

2.3.5 Proxy Servers

Proxy servers are designed to mimic the functions of the browser cache, but instead of being applied at the browser level, they are applied at the network level. This temporary cache stores Web page information when a user first downloads it. This information can then be accessed by additional users from the proxy server, instead of users retrieving information from the site itself.

Although proxy servers increase the apparent speed of the Internet connection, they can cause problems when developing applications for the Web. For example, because the proxy server is designed to act as a replacement for the Web server, how do you ensure that the user is receiving the results from the Web server and not the proxy server cache? Furthermore, not all proxy servers support the different security authentication methods, such as NT Challenge/ Response (NTLM) password authentication.

2.3 . 6 Firewalls

A firewall is a combination of hardware and/or software that is designed to keep private networks secure. Although firewalls are intended to keep internal networks free from hackers, often these same firewalls prevent you from delivering applications to your users. For example, depending on the security permissions assigned by the firewall administrators, firewalls can prevent cookies, ActiveX controls, and Java applets from reaching the requesting browser. If users of your Web-based applications are reporting irregular behavior, remember to check possible firewall and proxy server limitations.

2.3.7 Certificate Servers

As the popularity of the Internet explodes, people are finding new ways to exploit this global network of computers. Companies are using the Internet as an extension of their own private networks by offering access to internal information for their employees or customers through the public Internet. This capability to access private Intranet information via an Internet connection is often referred to as an extranet, which requires explicit security configuration and management.

However, protecting the private network is only half the security concern. Protective measures must also be in place to ensure user security and privacy rights. For example, what assurance does a user have that the online catalog site he is about to place a credit card order with is really the company the site portrays it to be? Unfortunately, this is one of the largest disadvantages of having an unregulated global network with no central administration unit. Anybody can place information on the Web, with no preventative measures to ensure that this information is accurate or true.

The certificate server was developed to help ensure data integrity for your Web sites or applications. Data integrity is composed of two parts. The first part, client and server authentications, is often overlooked. On the public Internet, third-party vendors offer authentication services to verify the identity and validity of the Web server and Web browser. However, if you are building an application for your Intranet or extranet, you do not have to rely on these external sources for authentication. The certificate server helps to ease this burden of proof for client and server authentication by enabling you to issue digital certificates. These digital certificates can then be administered and managed by the certificate server to regulate entrance to a particular site or application.

After user and server identification has been properly established, the second part of maintaining data integrity over the Internet is to ensure data protection while information is being transferred from the Web server to the client browser. The certificate server uses the digital certificates with client and server encryption schemes, such as Secure Socket Layers (SSL) and Private Communication Technology (PCT) protocols, to ensure data integrity.

2.3.8 External Servers

One of the most exciting aspects of moving to a server-centric application deployment model is the capability to use the HTTP server as a gateway to other application servers. These other application servers can consist of pre-built application-based software, such as Microsoft Exchange, the Index Server, or the News Server. However, you are not limited to these canned applications. You can build your own application server to meet your specific application needs. This flexible method for extending the reach of your Web server into other applications presents exciting opportunities for Web-based application development. Now these servers, which were once accessible only via a proprietary client interface, can be accessed by any browser anywhere in the world (Johnson 1997).

2.4 The Evolution of Active Server Pages

The concept of processing application logic on the Web server and connecting to external servers from the Web server is not a new idea specific to Active Server Pages. The capability to create dynamic HTML by sending information to an executable file existed before the emergence of Active Server Pages through the Common Gateway Interface.

2.4.1 Gateway Interface

The Common Gateway Interface (CGI) was one of the first methods used to create dynamic HTML. CGI enables direct communications between the HTTP server and executable scripts. Programming in CGI provided a standard communication and processing mechanism between the requesting client browser, the gateway program, and the HTTP server The CGI programs help create a standard interface with the HTTP server to eliminate having to learn the specifics of Hypertext Transfer Protocol.

CGI programs are usually written in a scripting language such as the Practical Extraction and Report Language (PERL). The early PERL scripts were created to run in UNIX because the early HTTP servers only existed on the UNIX platform. However, with the emergence of HTTP servers for NT the Internet Information Server supports PERL 5.0 scripts. To initiate a CGI executable, simply reference the name of the executable script and pass any required parameters as shown below:

<A Href = "/myCGIscripts/CalcInterest.exe?Principal=300000"> Calculate Interest using CGI</A>

In this example, the script named CalcInterest.exe is created and passed an argument named Principal with the value of 300000.

When a PERL script is called on the Web server, the Web server treats the PERL script as a separate executable. This executable program is not limited to just one script; it can consist of multiple scripts running on separate machines. However because the hosting server treats CGI applications as separate executables, a new process is created for each instance in which the CGI application is called. Creating new processes on the server is a very expensive resource task and can cause significant resource drain and performance issues, especially when considering scalability issues as your Web sites and applications grow. Furthermore, CGI applications suffer from the inability to share information across applications. This is because each new CGI process is created within its own memory space and cannot dynamically share information with other memory spaces of other instantiated CGI programs.

2.4.2 Internet Server Application Programming Interface

The Internet Server Application Programming Interface (ISAPI) builds on the lessons learned from the shortcomings found in CGI applications. ISAPI shares the same functional aspects of CGI programming but differs from traditional CGI programming in the way the script is executed. ISAPI relies on loading scripts into the HTTP server's memory space to reduce the resource drain required to create a new process, as shown in Figure 2.5. The significance of using a shared memory space is that all the resources made available to the HTTP service are now made available to the ISAPI applications.

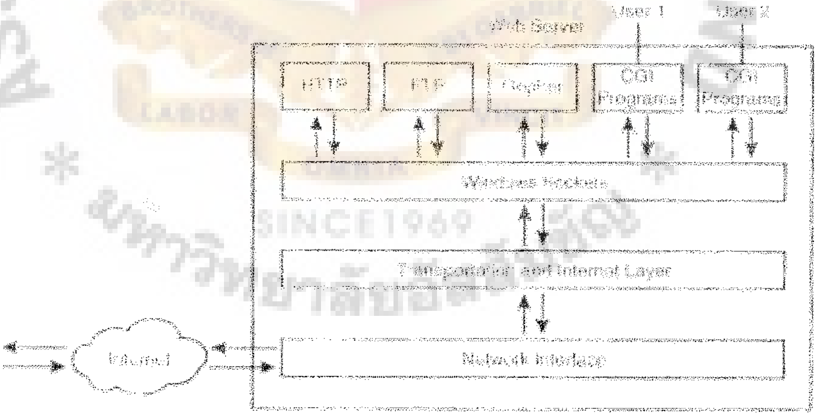


Figure 2.5. The Process of HTTP Server's Memory to Reduce Resource.

However, as in most situations, often the greatest strength of a product is also its greatest weakness. Careful and thorough coding and testing must be used when creating an ISAPI application. Because the ISAPI application shares the same memory space as the HTTP server, if the ISAPI application crashes, you can also bring down the HTTP server. If the ISAPI application terminates without cleaning up the corrupted memory space, runaway processes or memory leaks could result.

ISAPI is implemented in one of two ways: either as an ISAPI application or as an ISAPI filter. ISAPI applications are used to encapsulate the function found in traditional desktop applications, but now load that logic into the Web-based arena. To reference an ISAPI application, use the same methodology used to reference the CGI executable by calling the ISPAPI DLL and passing the appropriate filters, as shown in the following code:

<A Href = "imyCGIscripts/CalcInterest.DLL?Principal=300000"> Calculate Interest ISAPI filters are used to process information being sent to the HTTP server and post-process the HTTP server responses. The filter acts as an analytical tool to monitor information being sent to and being generated from the HTTP server and acts as another layer protecting the HTIP server from the network layer. See Figure 2.6.

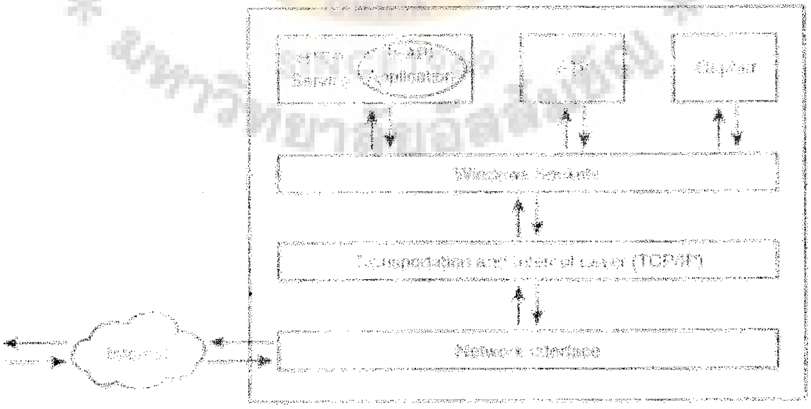


Figure 2.6. The Filter Acts as Analytical Tool to Monitor Information.

This inline filtering capability can provide extensive functions to your Web server. For example, you can use ISAPI filters to provide public or customizable encryption, authentication, and compression schemes. Furthermore, you can use ISAPI filters to initiate authentication methods, create advance-logging features, or scan incoming requests to thwart possible security attacks. With the Internet Information Server 4.0, ISAPI applications and filters can be applied not only globally across the entire server, but also to specific applications residing on the server.

2.4.3 The HTN/11_, Object Model

Up to this point, I have focused largely on the various servers and server components needed to create dynamic content. However, to fully consider the entire Internet architecture, you also have to consider how the browser translates and organizes the HTML sent to it from the Web server.

Just as programming languages adhere to a defined specification identifying the structure of the language, each Web browser follows its own internal model or blueprint that defines how the browser operates. The object model usually represents a hierarchical organization chart that identifies the different relationships between objects and collections for the particular browser system. For example, Figure 2.7 illustrates the Internet Explorer object model.

In most situations when you design Web-enabled applications, you only have to consider the specific browser's object model when you develop client-side scripts. These scripts, interpreted within the browser, provide advanced functions not found in native HTML Client-side scripting languages, such as JavaScript, VBScript, and JScript, are used to manipulate the content delivered by the server. The scripts manage the content by manipulating the object model of the browser. Typically, this involves controlling. The object model as a response to a particular user event. This client-side programming function can be as complex as you need it to be. For example, the HTML object model is used when the user navigates to a different frame page, as shown in the following example:

<A Href = "/frames/readme.html TARGET = "frmain">Terms and Conditions </A>

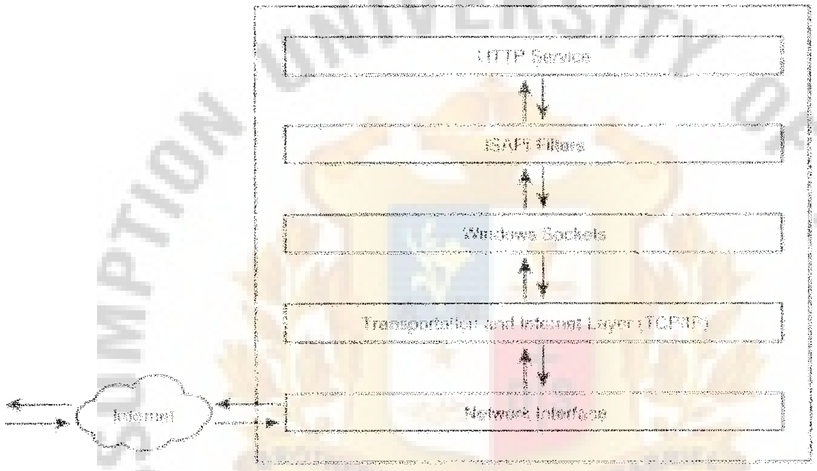


Figure 2.7. The Object Model of Hierarchical Organization Chart.

When user clicks the Terms and Conditions link, the HTML model is responsible for displaying the readme.html in the frame window named frmain. The client-side scripts work the object model to manipulate the information. The HTML object model can be used to read and extract information from or about a page. For example, the HTML object model exposes the hierarchical navigation to determine how many frames are located on a page, as shown in the following code:

Window.Document.Frames.Count

Furthermore, client-side scripting can be used to manipulate the HTML objects on a page. For example, the following code illustrates a JavaScript function, named set value, which is used to set the value of a check box named chkReturnEmail:

```
<SCRIPT LANGUAGE = "JavaScript'>
    Function SetValue(){ If (Document.FrmDa2a.ChkReturnEmail.Checked){
                Document.FrmData.ChkReturnEmail.Value = "True"


                Else{
                      Document.FrmData.ChkReturnEmail.Value = "False"
                }
    }
</SCRIPT>
```

This script uses the HTML object model to determine whether a check box named chkReturnEmail is selected. The HTML object model is used to set the hierarchical relationship between objects on the page. For example, an HTML checkbox named chkReturnEmail is located on a HTML form named frmData. The JavaScript function uses the HTML hierarchy to check if check box is selected. If the check box is selected, the script function sets the value of the check box to True. This value would then be sent to the Web server when the page is submitted.

Keep in mind that the object models for different browsers differ slightly. As a result, your Web-based applications should be thoroughly tested to ensure compatibility across browsers and client-side scripting engines.

2.4.4 Dynamic HTML Object Model

As the evolution of HTML continues, the Dynamic HTML object model expands the function found in the standard HTML object model. The Dynamic HTML object model now enables almost complete control over all aspects of Web pages by expanding properties of the document object, as shown in Figure 2.8.

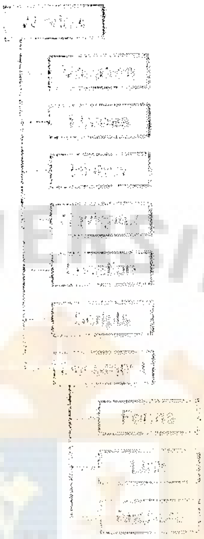


Figure 2.8. All Aspect of Web Pages.

This new and improved object model embraces using stylesheets, content positioning, and downloadable fonts to control the look and feel of your sites and applications. Stylesheets enable Web page authors to control the visual characteristics of their sites from a single template. The visual style components for an entire site, such as alignment and font attributes, can be managed from a single template file. New content-positioning features eliminate the limitations of positioning items that existed in standard HTML. In standard HTML, you could not layer items, such as text or images, on top of each other Dynamic HTML gives you the capability to layer items on a Web page. In addition, Dynamic HTML also enables you to download fonts that are required for the design or display effect for your site. Downioadabie fonts eliminate the dependence of having to rely on fonts that have been previously installed on the client's machine.

Dynamic HTML is intended to give greater flexibility in programming functions and controlling and regulating how the site is displayed, In order to use these new features of Dynamic HTML, a Dynamic HTML-compatible browser is required. Internet Explorer 4.0 and Netscape Communicator 4.0 browsers support the Dynamic HTML standard. However, regardless of what client interprets it, the HTML is still delivered from functions to process application logic on the Web server.

2.4.5 Active Server Pages and the Active Server Model: A COM-Based Evolution

Everything released from Microsoft has the tag "Active" attached to it: Active Server Pages, Active servers, ActiveX controls, the ActiveX framework, ActiveX Control Pad, and Active scripting. The goal for this section is to define the implications that the "Active" tag carries and what it means for applications that you want to port to the Web.

The Active revolution can trace its origins back to the development of the Common Object Model (COM). COM was a standard developed by Microsoft to create a standard communication mechanism between components. This binary standard would enable non-vendor specific components to interface with each other. When components had a communication standard, cross-vendor components should share their property methods and events with other components. This process of activating another component's properties, methods, and events is known as OLE automation.

An example of OLE automation is the capability to embed an Excel spreadsheet in a Word document. However, the OLE automation offers great flexibility in application development. You can create your own OLE servers in COM-based tools that expose their methods to calling programs. These OLE servers can be interfaced from native Win32 clients, such as Visual Basic, Delphi, and PowerBuilder applications, or from middle-tier components, such as the Web server. OLE automation servers can be created as stand-alone executables (.exe) or they can exist as process Dynamic Link Libraries. As stand-alone executables, the OLE servers perform in a similar manner to CGI scripts, where the application is created as a new process in its own memory space. You can also compile your OLE server as a DLL, to create ISAPI application and filters.

The reach of COM automation originally enabled OLE controls, or OCX controls, to decrease development time by adding extra function to the native application development environment, such as Visual Basic or Delphi. These controls represent pre-canned function that would otherwise have to be coded by hand. For example, a grid OCX control could be added to a Visual Basic form to retrieve, filter, and display information from a database into a grid layout. This drag-and-drop database function would have taken a considerable amount of time if that functionality had to be coded by hand. These OLE controls have been recompiled and optimized as ActiveX controls to add that same function to Web pages.

Distributed COM (DCOM) is a Microsoft standard that enables COM objects to communicate across machine boundaries regardless of network configurations. This distributed component communication mechanism is the foundation that provides the Internet Information Server and Active Server Pages the flexibility to communicate with other COM-based systems. For example, this function enables you to expose database information from any ODBC-compliant data source, access mainframe data via the SNA server, and extract information from COM-based servers, such as Lotus Notes, Exchange, or your own customized server.

Active Server Pages represent the capability to process application logic on the Web server. The ASP scripting engine can process any script-compliant language, such as Vl3Script and JScript. These script-compliant languages are referred to as ActiveX scripts because they can be processed on the Web server. The ActiveX scripting engine is actually a COM component itself.

Furthermore, the ActiveX scripting engine can also process requests made by non-ActiveX scripting languages, such as PERL, LISP and REXX scripts. The process that makes this possible is using COM to provide an OLE wrapper around these scripts. This wrapper acts as an interpreter between the non-ActiveX scripts and the ActiveX scripting language.

Furthermore, ActiveX documents refer to the same process of converting OLE controls to ActiveX controls, but instead of being applied at the individual component level, COM technology is applied at the application level. ActiveX documents refer to the capability to load entire applications as a COM object within the Internet Explorer.

As you can see, COM provides ASPs with great functions to access various information stores on the server side and deliver that information-as HTML-to the client. However, the flexibility is not necessarily guaranteed when delivering solutions via client-side ActiveX controls or ActiveX documents. Keep in mind that in order to load the ActiveX controls or ActiveX documents, the requesting browser must be COM compatible, meaning the ActiveX controls and documents must be compatible with both the browser and the browser's operating system. Using client-side ActiveX controls may not be the best solution to reach the maximum of users on the Internet, but they do provide great value for intranet/extranet applications. Intranet/ extranet applications are unique because, in most situations, the user group's development platform is standardized and well defined. This standardization enables you to tailor your solutions to the user group's profile instead of trying to deliver a solution to the lowest common denominator (Johnson 1997).

2.5 Logical Database Design

The logical database design should define:

(1) Tables and their names (also called entities).

(2) Column names (also called attributes) for each table.

(3) Column characteristics, such as requiring unique values and allowing nulls, and the type of data that the column will store.

(4) The primary key for each table. This is the column (or set of columns) that stores values that uniquely identify each row in a table. Although other unique columns may exist in the table, only one is identified as the unique access key for retrieving rows—the primary key. Each table can have only one primary key. Although a primary key is not strictly required, you should always define one.

(5) Relationships between tables. Rows in some tables are dependent on one or more rows in another table. These intertable dependencies are called relationships. To define a relationship, a column (or set of columns) in one table, called a foreign key, references the primary key of another table.

For example, each row in an Orders table is dependent on a row in a Customers table because each order must be placed by a customer. This is a relationship between the Orders table and the Customers table. The Orders table must have a column that holds values that reference an individual row in the Customers table. Rows in the Orders table must be guaranteed to reference only one customer so the relationship should be based on the primary key of the Customers table. The Orders table column that references the primary key of the Customers table is called a foreign key.

2.5.1 Types of Relationship

Three types of relationships are possible between tables:

(1) One-to-One

Each row in the primary is related to only one row in the foreign table. A one-to-one relationship is implemented by defining the foreign key to be unique (that is, not allowing duplicates)

(2) One-to-Many

Each row in the primary table is related to one or more rows in the foreign table. For example, one customer can place many orders, but one order cannot be placed by many customers.

(3) Many-to-Many

Many rows in one table are related to many rows in another table. For example, an author can write many books and a book can be written by more than one author. A many-to-many relationship between two tables is implemented by creating a third table and creating a one-to-many relationship to this junction table from each of the initial tables.

2.5.2 Normalization

When designing the tables and relationships for a database certain logical inconsistencies commonly occur. A process called normalization helps you to ensure that these inconsistencies do not occur.

Normalization is the process of refining tables, keys, columns, and relationships to create a consistent database design. Normalization is achieved by applying a number of tests to tables. Three levels of normalization (first, second, and third normal form) are commonly applied, although others are defined

The benefits of normalization include:

Sorting and index creation, because tables are narrower

More clustered indexes, since there are more tables

Compact and narrow indexes

Fewer indexes per table, improving UPDATE performance

Fewer NULLS and less redundant data, increasing database compactness

Reduced concurrency impact of table locks, because they will affect less data

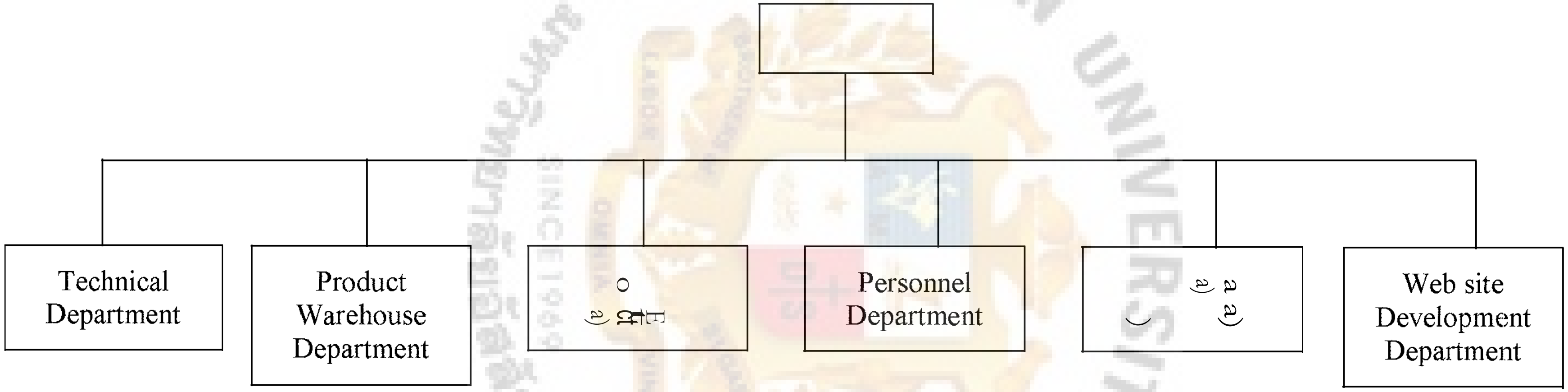
III. SYSTEM DESIGN

The purpose of this report is to develop a web program to sell products and services on the Internet. The company ComputerSolution.com included in this report is a fictionalized company providing computer products and services to end-users. The products sold include computer hardware components and operating systems. The supported services are mainly telephone technical support, related activities, and maintenance that cannot be done locally.

In order to design ordering system of company's web site, business system should be planed first.

3.1 Business System

To optimize the resources and personnel of the new company, Computer Solution, a break down of the company's structure is needed. The company will have a flat structure with wide span of control. This kind of structure has fewer levels of management. This has a direct impact on the bottom line by reducing the overhead cost and making the operations more efficient by reducing decision time (Schermerhorn 1997).



b0 cr; Organization Chart.

According to the above figure, the company divides its departments into 6 departments:

(a) Technical Department

(b) Product Warehouse Department

(c) Accounting Department

(d) Personal Department

(e) Marketing Department

(f) Web site Development Department

There are two parts of the designing process: job design and job enrichment. Job design specifies what people must do to carry out the tasks. Job enrichment is used to make the jobs more interesting and rewarding to the system users. For example, providing a good interface ensures that people will not be frustrated with the complexies and demands of the system.

3.1.1 Technical Department

The technical department is responsible for answering technical questions and problems encountered by Computer Solution's customers from the company's web board or customers' email.

Responsibilities

(a) To check customers' question entry.

(b) To reply customers' question entry.

(c) To repair customers' products.

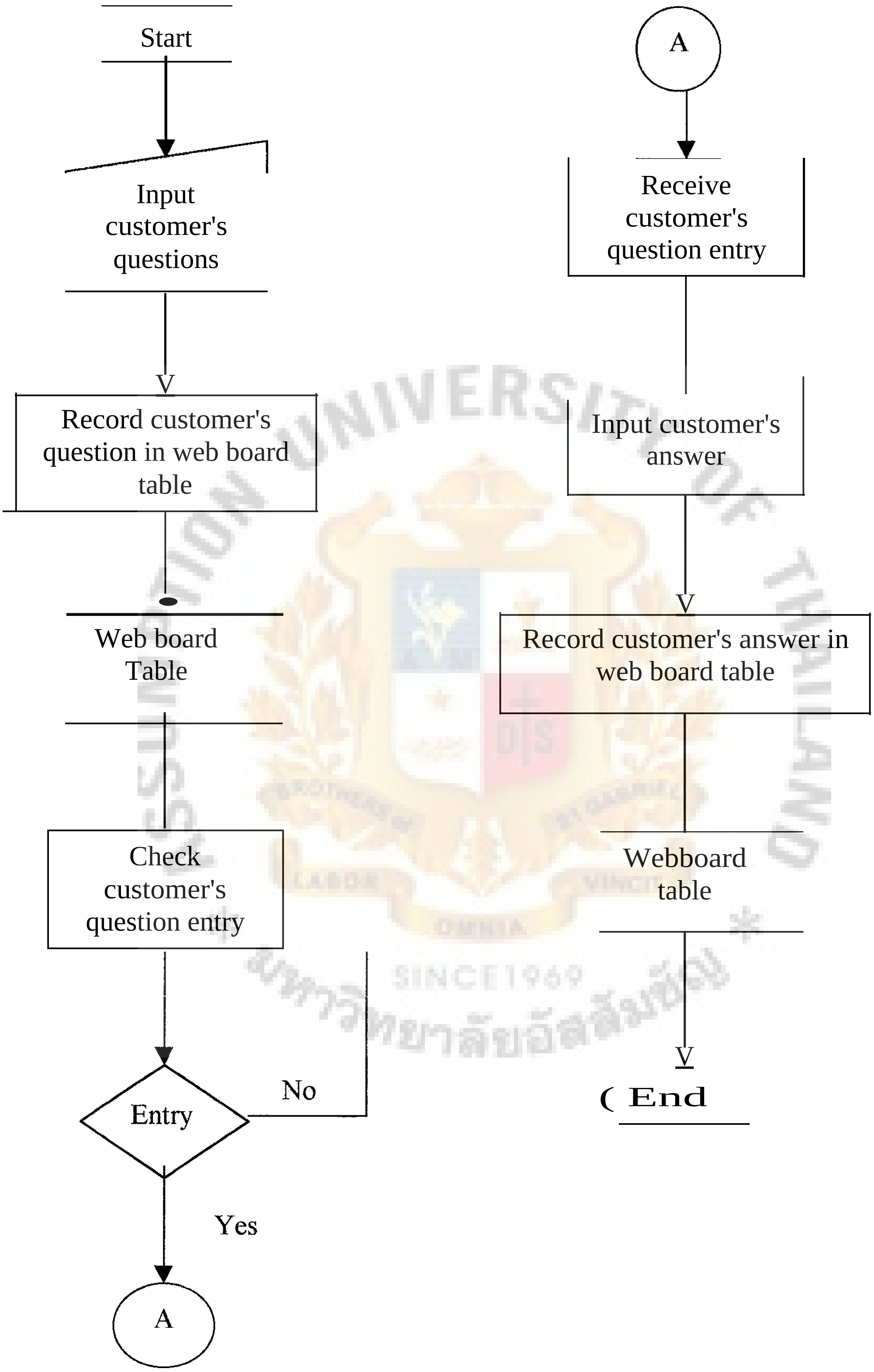

Start
Input customer's questions
Record customer's question in web board table
Web board Table
Check customer's question entry
Entry
No
Yes
A
A
Receive customer's question entry
Input customer's answer
Record customer's answer in web board table
Webboard table
End

Figure 3.2. Flow Chart of Technical Department.

3.1.2 Product Warehouse Department

Product warehouse department is responsible to check customer's order entry, delivery products according to customer's order and manage stock of computer products.

Responsibilities

(a) To check and receive customer's order entry.

(b) To search products in stock according to customer's order.

(c) To deliver products and invoices.

(d) To manage and control the stock of computer products which have reached a low level.

(e) To add, delete, or update product in the product table of database.

(f) To purchase products from suppliers.

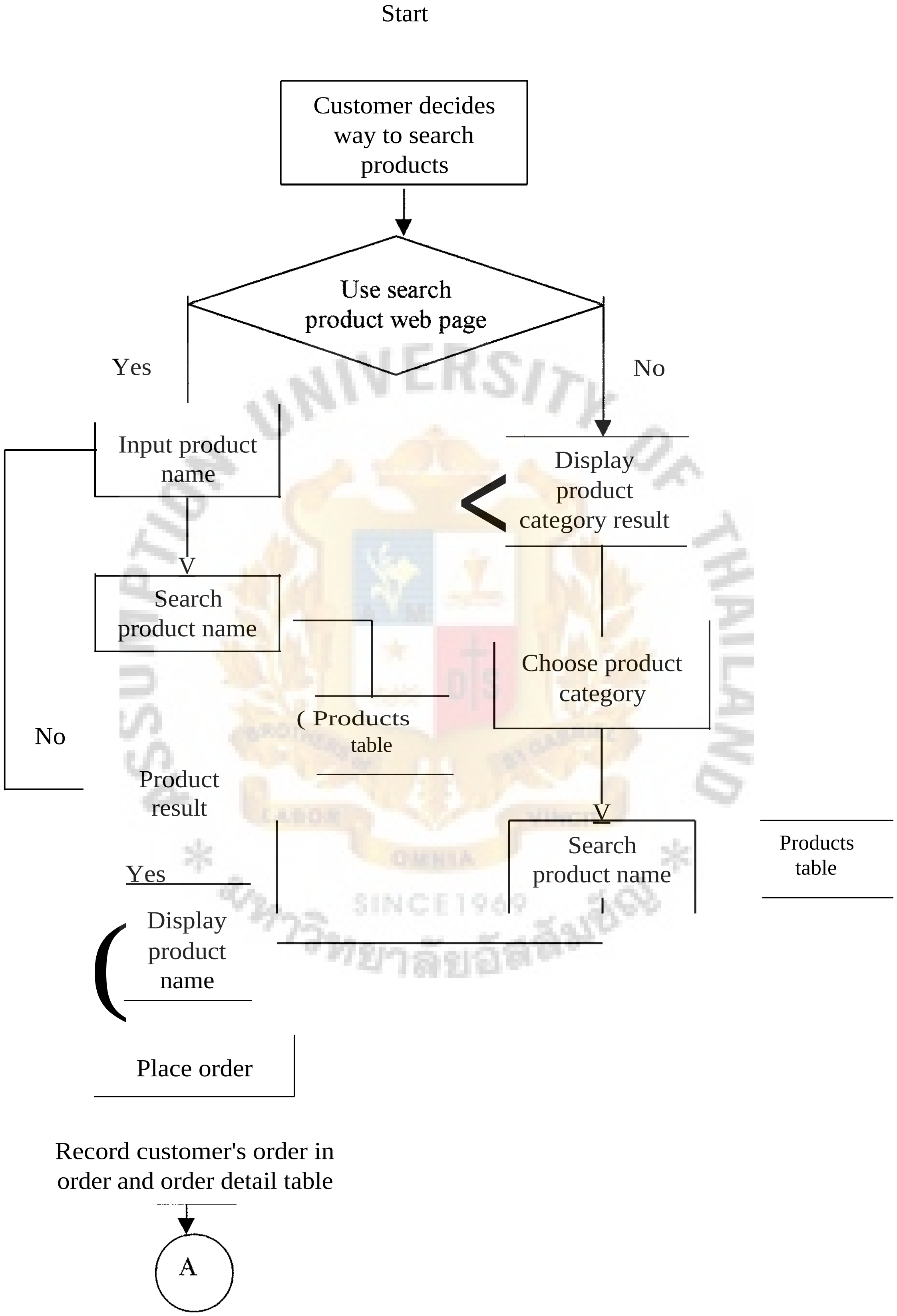


Figure 3.3. Flow Chart of Product Warehouse Department.

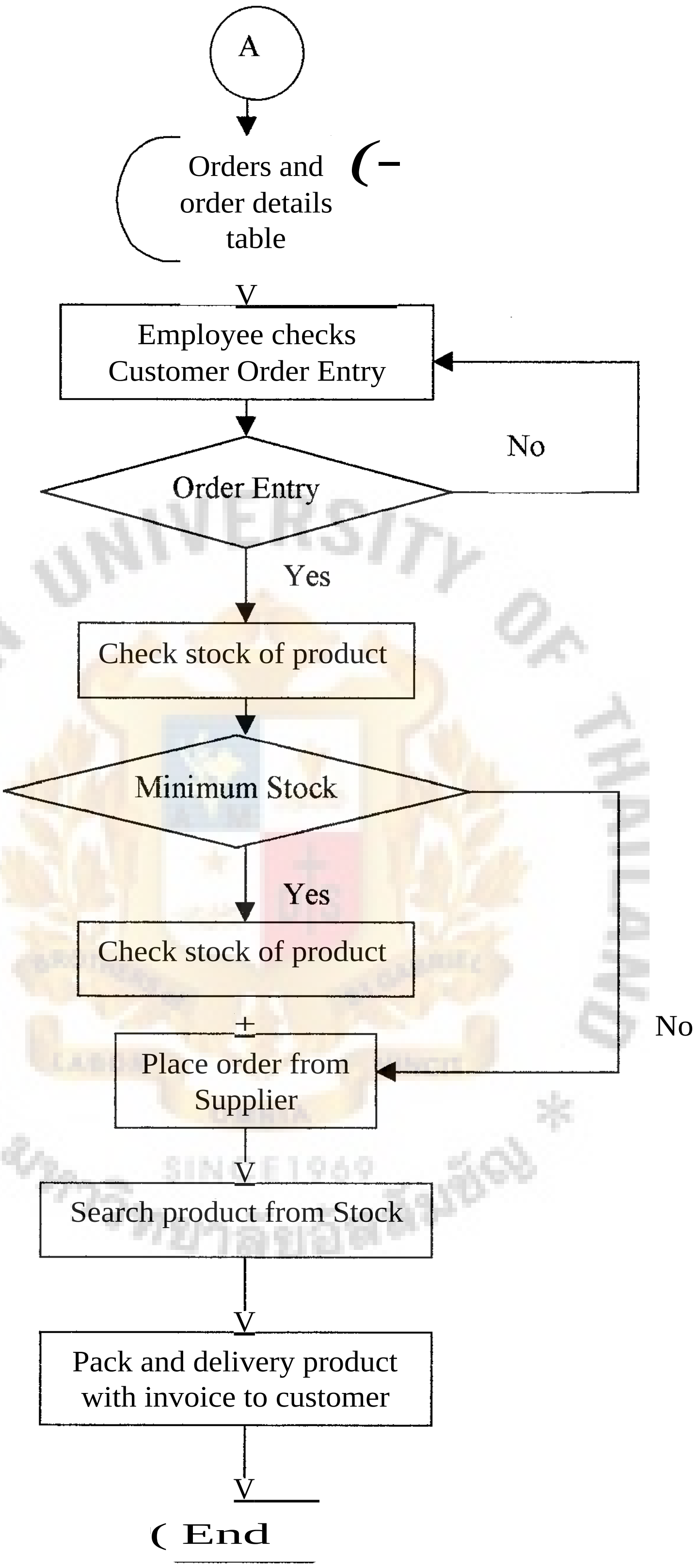

A
Orders and order details table
Employee checks Customer Order Entry
Order Entry
No
Yes
Check stock of product
Minimum Stock
Yes
Check stock of product
No
Place order from Supplier
Search product from Stock
Pack and delivery product with invoice to customer
End

Figure 3.3. Flow Chart of Product Warehouse Department. (Continued)

3.2 User Requirement

3.2.1 Business Requirements

(1) The database of web site system should be up-to-date and accurate information.

(2) The system should have web board for customer to post any questions, especially technical question.

(3) The system should have search program for users to retrieve product data easily.

(4) The system should have inventory management program for adding, updating and deleting product data.

(5) The system should have security to allow only authorized person to add, update and delete data.

(6) The system should have ordering system that automatically cuts stock when customer buys products.

(7) The system should be user-friendly for customers, because people who do not have experience in using web site can use it easily the first time.

3.2.2 Input Requirements

(1) Customer information contains all details of customer, which are Customer id, Gender id, last name, first name, address, amphur, province id, postal code, phone, mobile phone, fax, email, and identification card number.

(2) Customer's order information contains all details of customer's order, which are order id, customer id, retailer id, order date, credit id, expire date, institute id, credit name, product id, unit price, quantity, and discount.

(3) Product information contains all details of product, which are product id, product name, supplier id, category id, quantity per unit, unit price, units in stock, reorder level, discontinued, and description.

3.2.3 Output Requirements

(1) Invoice is designed to help product warehouse department check customer for inventory control.

(2) Search program should display only products, which exist in stock.

IV. DESIGN AND DEVELOP WEB SITE

4.1 Web Site Design

In designing the on-line business information system, the company's database is the heart and soul of the business operations. The good design database ensures that there are no problems occurring after implementation. The good database will cover the requirements and reduce problems. Figure 4.1 shows database model of the company.

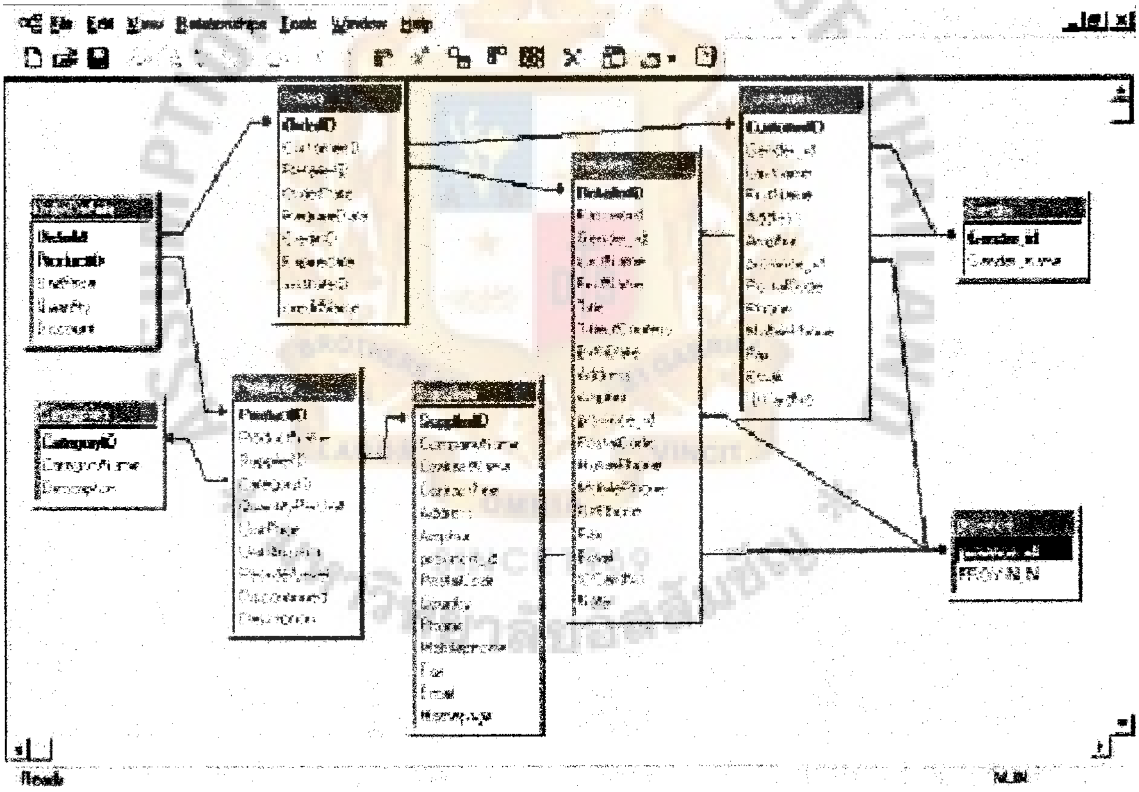


Figure 4.1. Database Relationships.

4.1.1 Hardware and Software Requirements

(1) Hardware Requirements

For this project study, hardware requirements are as follows:

(a) CPU Pentium II 300 MH$_z$

(b) RAM 128 MB

(c) Hard disk 4 GB

(d) CD-ROM - 12X

(e) Net work Adapter

(0 4 MB Video Adapter

(g) Super VGA 17" Color

(h) Mouse

(i) Soundcard

(j) Keyboard

(2) Software Requirements

For this project study, software requirements are as follows:

(a) Window NT 4.0 Server

(b) Window NT Server 4.0 Server Pack3

(c) Internet Explorer 4.01 SP1

(d) Window NT 4 Option Pack

(e) Visual InterDev 6.0

(0 Front Page Server Extensions

(g) Window NT Service Pack4 and Y2K Update

(h) Site Server Service Pack2

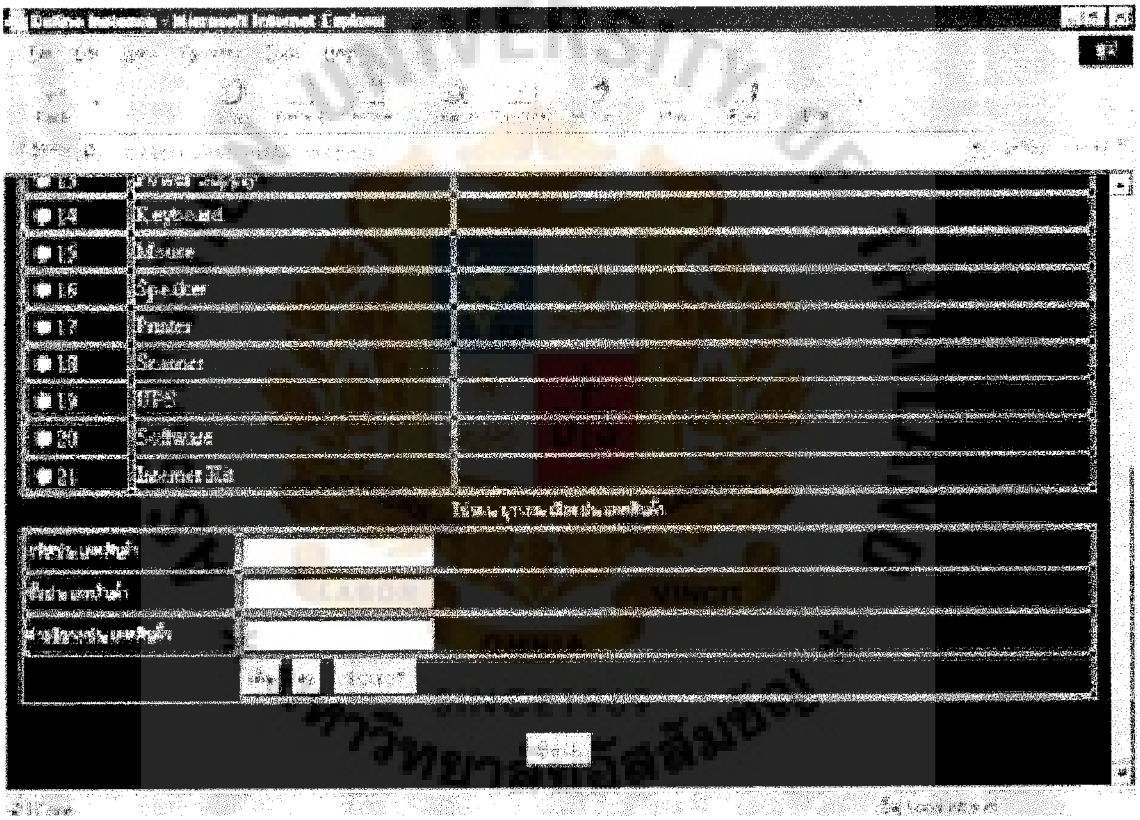
(i) Window NT Server 4.0 Service Pack5

4.1.2 Input Design

A customer inputs his order to interface with Microsoft Access's database. The input data can be divided into 3 parts as follows:

(1) Defining constant values for the system

Defining constant values are the first step to start developing the web site such as defining the category of the product.



● 13	Power supply	
● 14	Keyboard	
● 15	Mouse	
● 16	Speaker	
● 17	Printer	
● 18	Scanner	
● 19	UPS	
● 20	Software	
● 21	Internet Kit	

Figure 4.2. Example of Designing Constant Values Web Page.

(2) Management of data

(a) Adding data

Adding data means to record product details to the company's database based on customer's order.

(b) Updating data

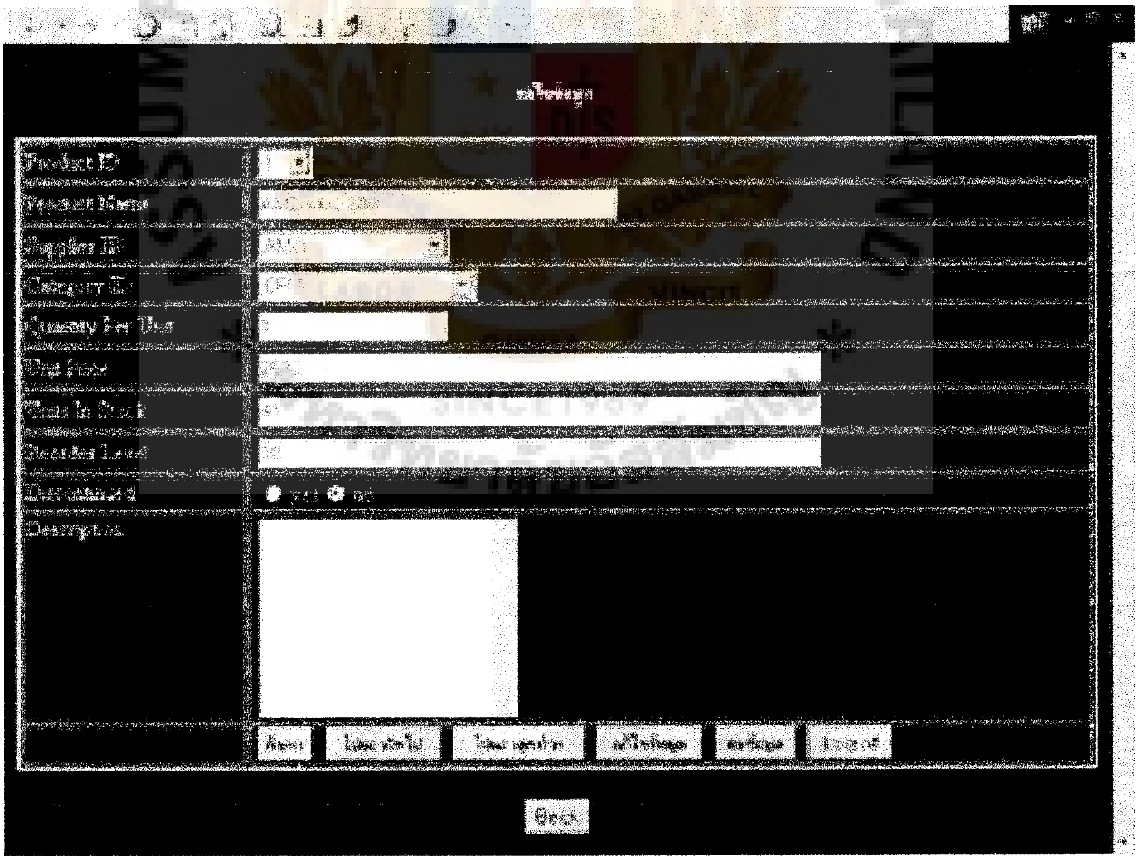
Updating data means to put new values into the database.

(c) Deleting data

Deleting data means to permanently get rid of the obsolete data.

(d) Displaying data

Displaying data means to show the data on screen for verification purposes.

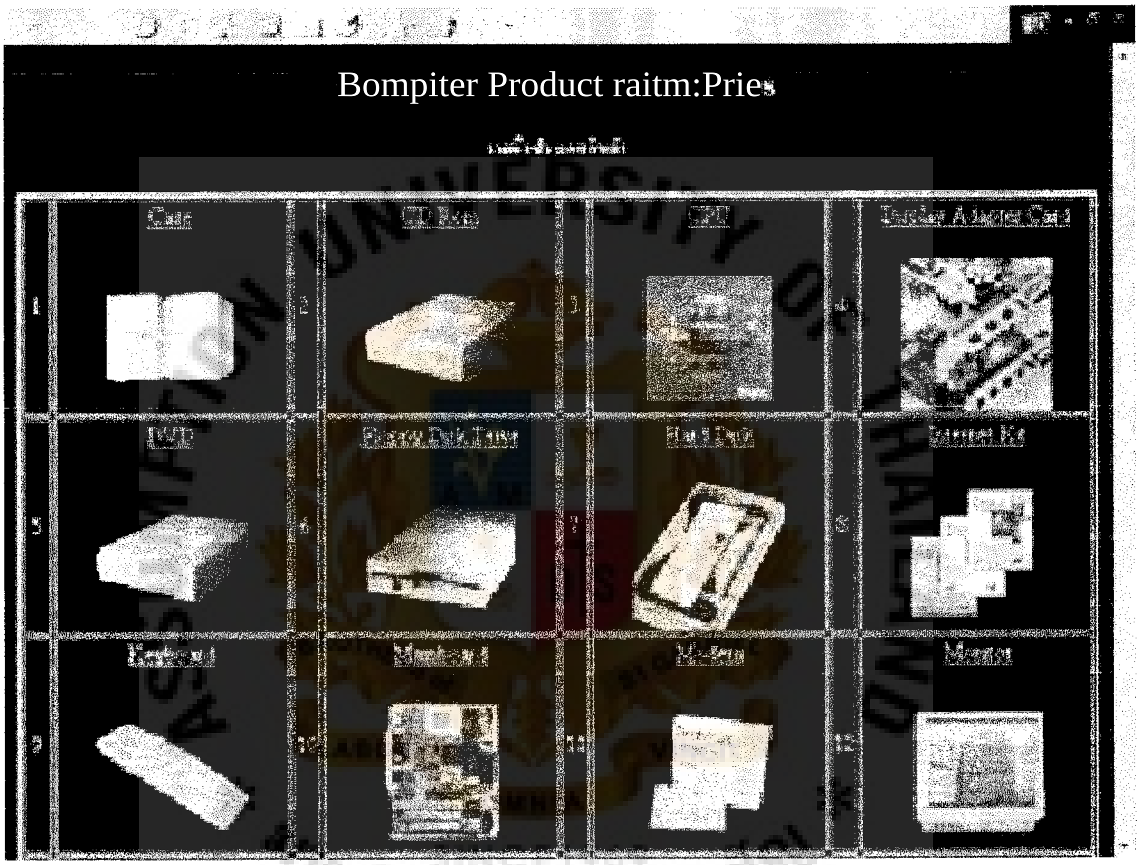


Product ID	[illegible]
Product Name	[illegible]
Supplier ID	[illegible]
Category ID	[illegible]
Quantity Per Unit	[illegible]
Unit Price	[illegible]
Units In Stock	[illegible]
Reorder Level	[illegible]
Discontinued	○ yes ● no
Description	

Figure 4.3. Example of Management of Data Web Page.

(3) Ordering data

Ordering data means to record ordered products, credit card, and personal data from customers.

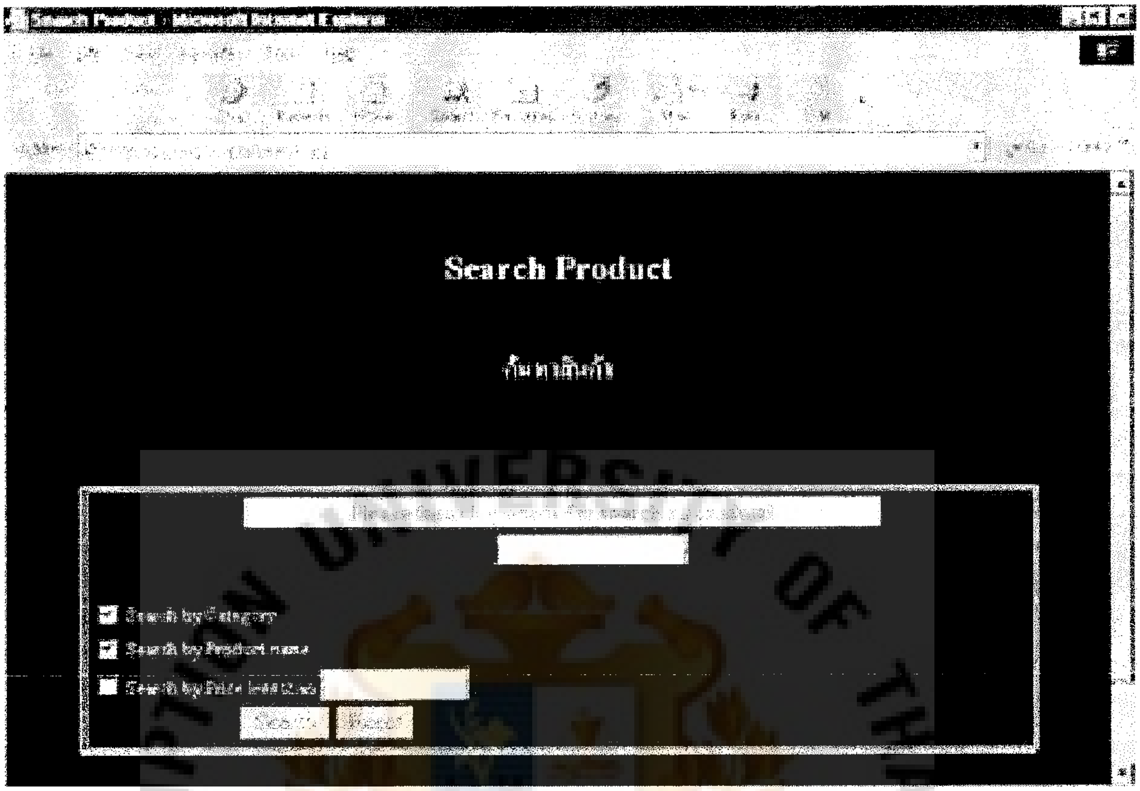


Bompiter Product raitm:Pries

Figure 4.4. Example of Ordering Data Web Page.

4.1.3 Output Design

Output design means to present a detail of searched products to customers on screen.



Search Product

ค้นหาสินค้า

☑ Search by Category

☑ Search by Product name

□ Search by Price limit (Less

Figure 4.5. Example of Output Design Web Page.

4.1.4 Security Control Design

(1) Database security

Database security is designed to control every access by specifying who has privileges to read, write, delete and modify information in the database.

(2) Program security

The Microsoft Access software can set up rules that help maintain a degree of accuracy between tables. Rules to enforce the integrity of information between tables are known as referential integrity. These rules keep relationships between tables, and prohibit changing data in ways that invalidate the links between tables. For example, deleting a customer record is illegal modifications. If you did delete a customer record without first deleting the customer's order, you would have a system that has an order without a customer.

All data files are backed up in secondary storage such as diskettes to ensure the correctness of data operation.

(3) Terminal access

Terminal access is used to control the security and integrity of the database by verifying acceptable usage. Customers are allowed only lower level privileges. Only certain warehouse personnel are allowed to make direct modifications to the company's database.

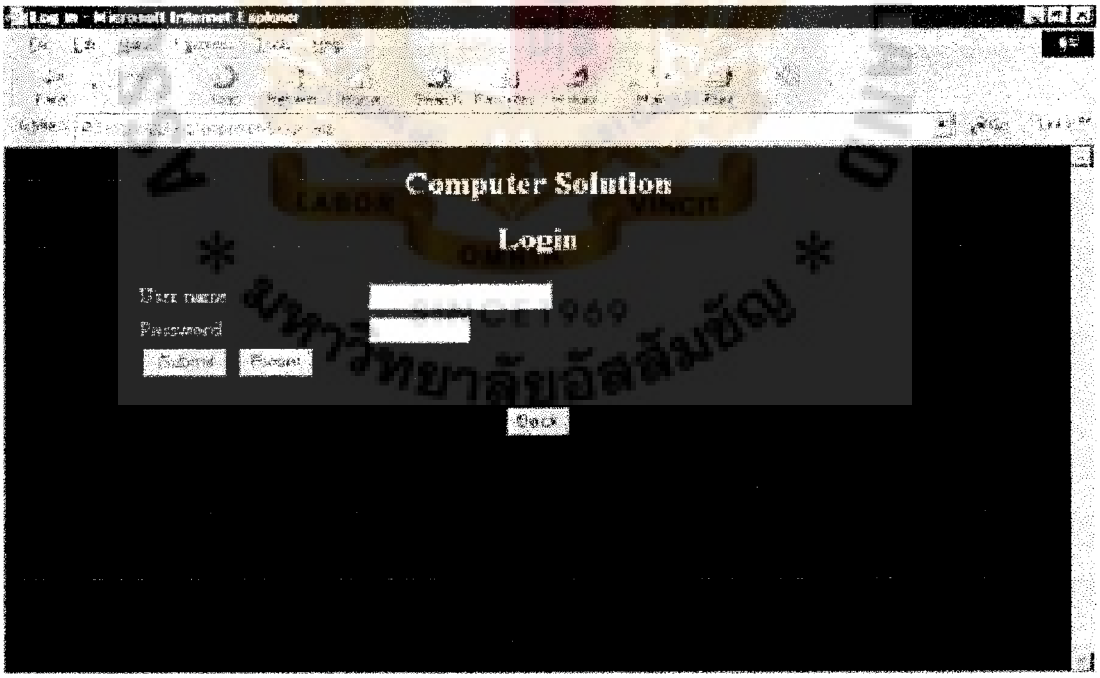


Figure 4.6. Example of Security Control Web Page.

Hence, online access of information can be met by verifying user name and password. Users have to input the right username and password to log into the database. The web program checks user authentication before allowing any user into the system.

4.2 Develop Web Site

Developing the database is an important component of the information system development. Input design, output design, and security design is used to develop the database for the system to have the following capabilities: controlling redundancy, sharing data, restricting unauthorized access, providing multiple interface, enforcing integrity constraints, and availability of up-to-date information.

Access program and FrontPage 2000 program are used as the core programs to develop the company's web site. The Access program is used to develop database of the company. The FrontPage 2000 program is used to develop the user interface and web program by using the ASP and the Java Script language.

4.2.1 Web Site Development's Steps

(1) Creating tables and relationships

(2) Creating site diagram

(3) Creating user interface of web site

(4) Coding web program

(5) Testing and prototyping the web site

Creating Tables and Relationships

According to tables and relationships that is designed on Figure 4.7 and Figure 4.8, the database of the web site can be developed by using the Access program.

(a) Creating table

(b) Defining field name by using attribute name

(c) Defining data type and field size

(d) Defining field property

(e) Do 1.1-1.4 until finishing all tables

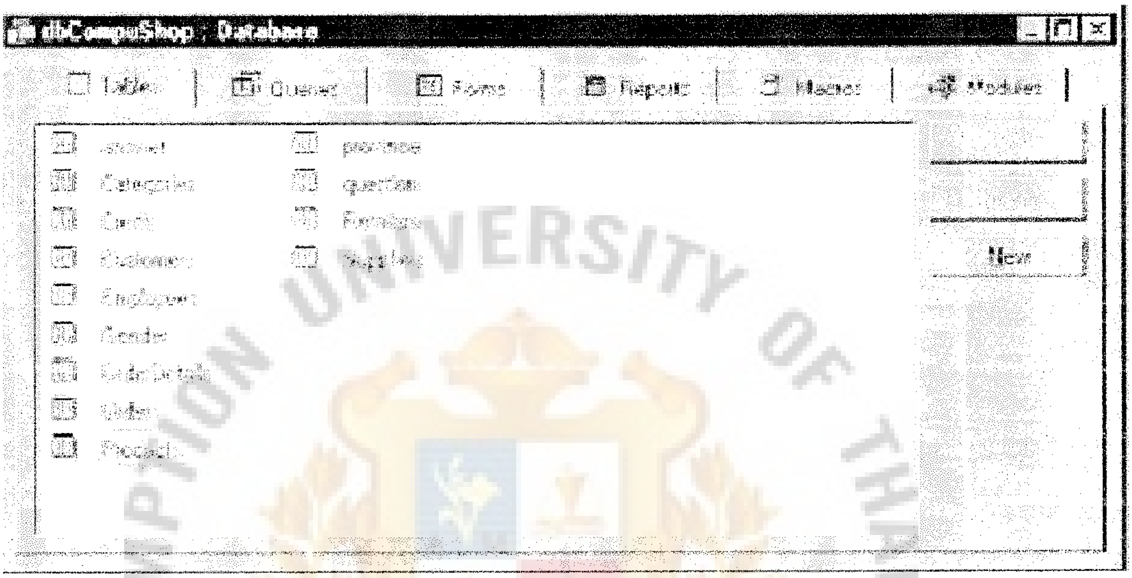


Figure 4.7. Example of Creating Table.

After all the tables are created, creating relationships of each table are the next step.

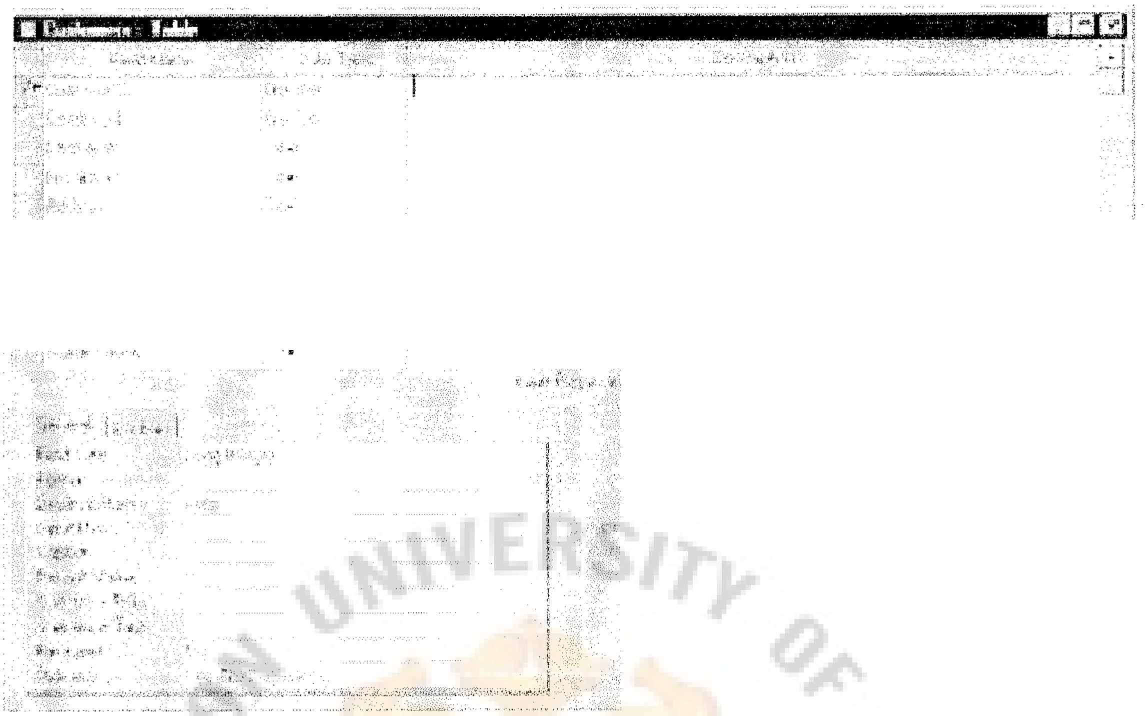


Figure 4.8. Example of Defining Field Name, Data Type, and Field Size.

Creating Site Diagram

This step is the process to create a map of the web site, which shows the programmer how to plan his or her web site.

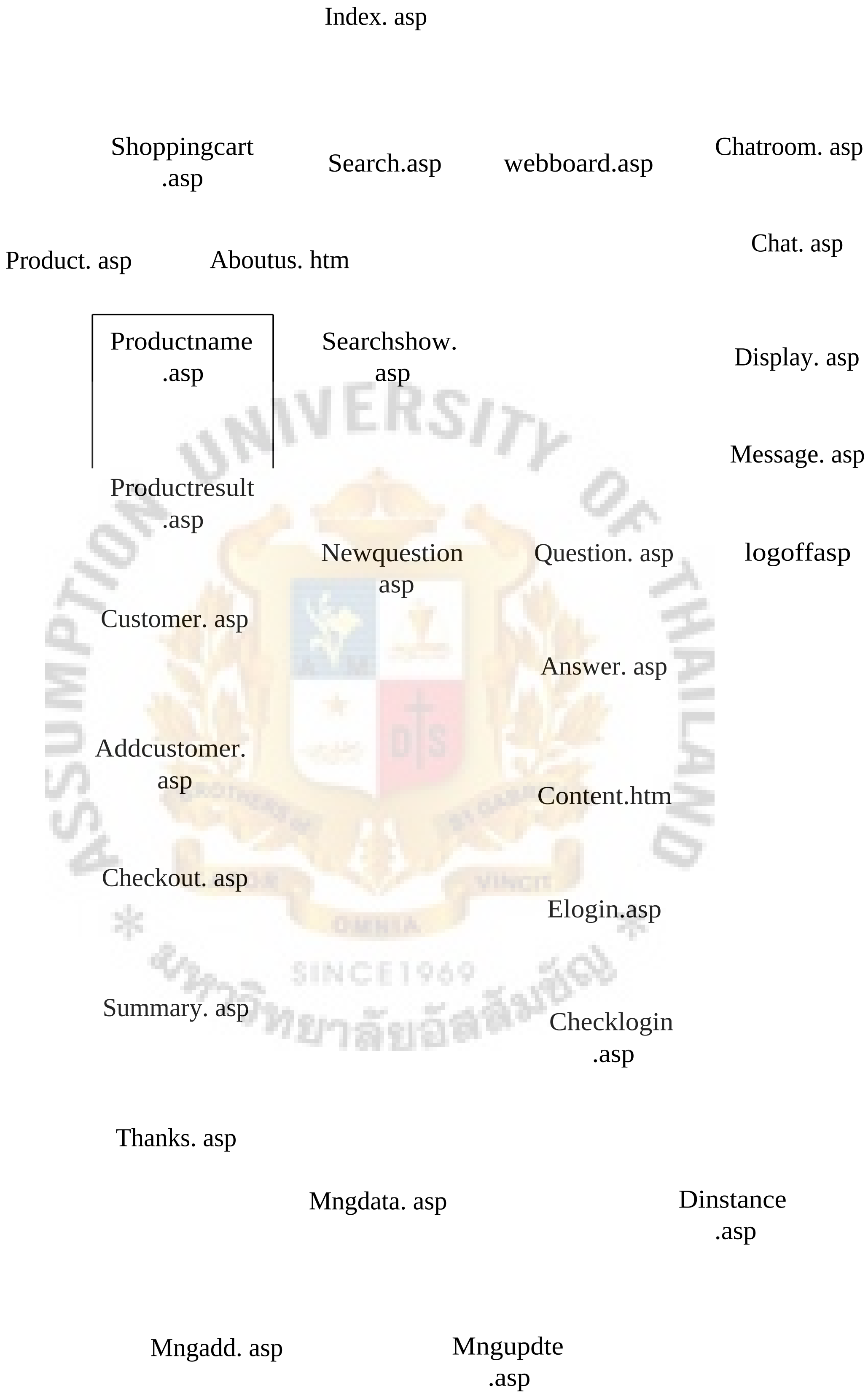


Figure 4.9. Site Diagram.

Creating User Interface of Web Site

This step is to create a graphic user interface of the web page. FrontPage, Adobe PhotoShop, and Adobe PhotoShop Image Ready are used to create the user interface.

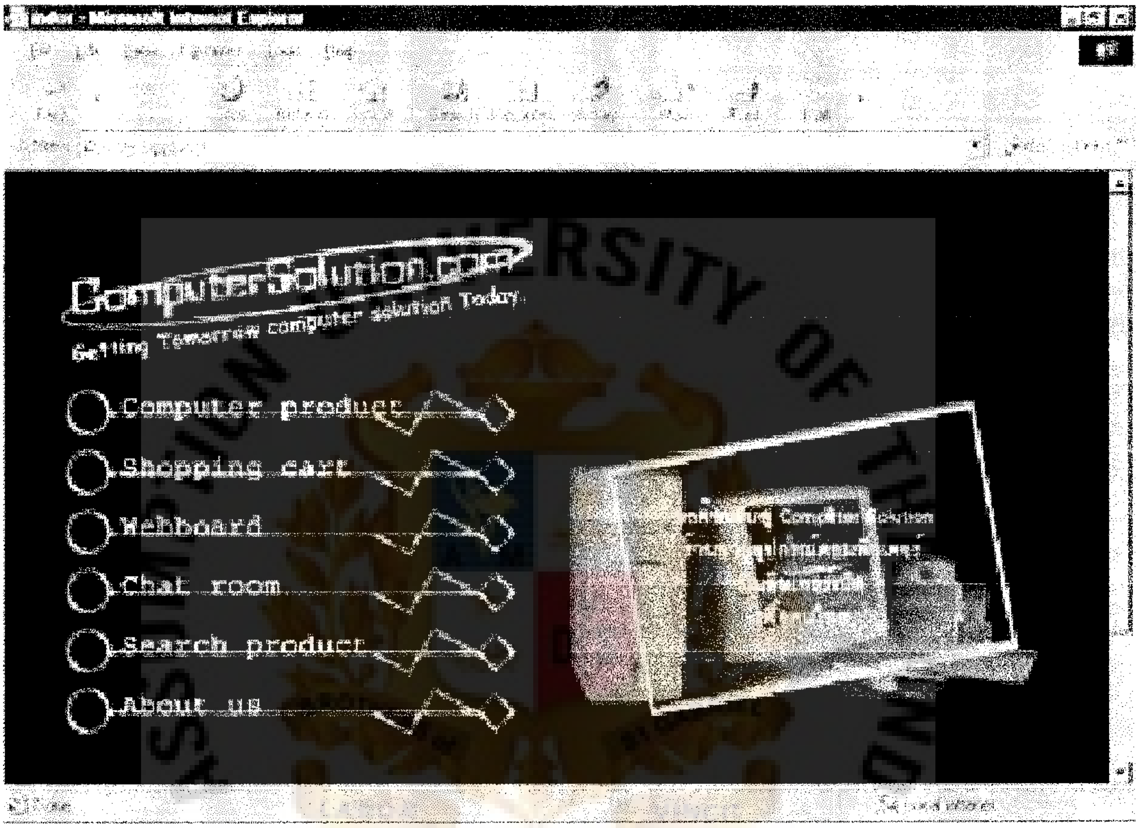


Figure 4.10. Example of Creating User Interface.

Coding the Program

This step is to code the program for each action event according to the button object on the form by using ASP and Java Script language.

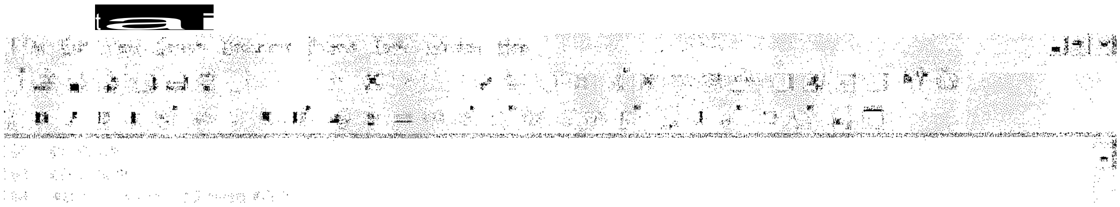


Figure 4.11. Example of Coding the Program.

Testing and Prototyping the Web Site

This step is used to test the program and developing a prototype system.

V. WEB SITE TESTING AND IMPLEMENTATION DEVELOPMENT PLAN

5.1 Web Site Testing

After developing web site, the next step is to test the web site so that it reaches the objectives and the requirements of the project. Code testing is to execute every instruction path in a program and determine how the program operates under specific conditions.

In testing, real data will be used to assure that web program works well according to the company's objectives and requirements. For example, inputting ordering data is to test customer's ordering. The results of this example will be used to determine time to deploy this web site.

After the programmer tests the program by using the test data, the result of the test should be at a satisfactory level.

5.2 Implementation Development Plan

After the web site system is designed and developed, which should be assured that the system is operating correctly and allowing user to use and evaluate. Because this is a fictionalized company, the need of conversion from the old system to the new system is not necessary. Employees can adopt this operation immediately, however training, and post implementation review should be included.

Training employees to understand and interact with system is an important part of implementation development plan. Although the system is excellently developed, without good training, the system cannot operate well without human. Employees will be trained in different methods depending on their jobs, personality, experience, and background. Therefore, a combination of methods to make employees understand the system will be used such as lectures, demonstration, and practice.

Post implementation review is to evaluate web system by using operation evaluation technique, which is to evaluate by regarding operation of the system function such as difficulties, response time and integrity. And development performance is one of many techniques, which is considered to evaluate with regarding time, performance and cost of the web site and whether the results of the project are profitable or not.

Gantt Chart of Project Plan.

[illegible]	Time					
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1 .Previous Ordering system Investigation						
2.Determination of Ordering system requirements						
3.Web site Analysis						
4.SWeb site Design						
5.web site Testing, and Implementation						
6. Written the report						

VI. CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

The objective of this project is to design and develop online business system, which is expandable. In developing database, Microsoft Access is designed to keep data, which has an important feature. Relational database management system is an important feature of Access. Access has the ability to connect with other programs such as Visual basic, ASP, etc. Access is flexibility for future needs. Tools of Access is helpful to design database with limited time for web programmers.

In developing web site, Adobe Photo Shop is one of the programs that are used to develop home page, because the first page of web site should create the first impression for customers. For other web pages, the importance of graphic will be reduced and the importance of time to browse the web pages will be increased. Most customers usually want to browse a web page within 30 seconds. Most people do not want to waste their time to wait for pretty graphic web sites.

Another objective is to design and develop an on-line business information system, which reduces the operation of the web server. Active Server Pages and Java script language is considered to develop the web program, which connects to database. The advantage of ASP is customers can not see the code of the program, but it will use resource of the server. Java script is program language that helps ASP to reduce works of the server because Java script runs at client side. In order to develop web site effectively and efficiently, the combination of these two languages is necessary. Somehow there are some problems, because ASP and Java script are not compatible. Microsoft develops ASP. Netscape Corp. and Sun Micro System Corp develop Java script. The researcher found that there is one interception between Java script and ASP. Both are developed for developing web site, so both of them support HTML. From this point of view, HTML is used as a medium for both languages. Tag input is the key for solving this in sending the variables of Java script to ASP or ASP to Java script.

After testing, a web program is done the program operates correctly according to all of the objectives.

6.2 Recommendations

The web site is designed under the limitation of time. This project does not provide any thing more than those requirements.

However, some feature should be added in. These are:

(a) The system should provide password according to level of authorization.

(b) The agency of the company should be added when the company grows.

(c) Search program of users should have more flexibility than the existing system. The user should search for product by providing more constraints. For example, the user can input keywords more than one keywords.

APPENDIX A

DATA FLOW DIAGRAM

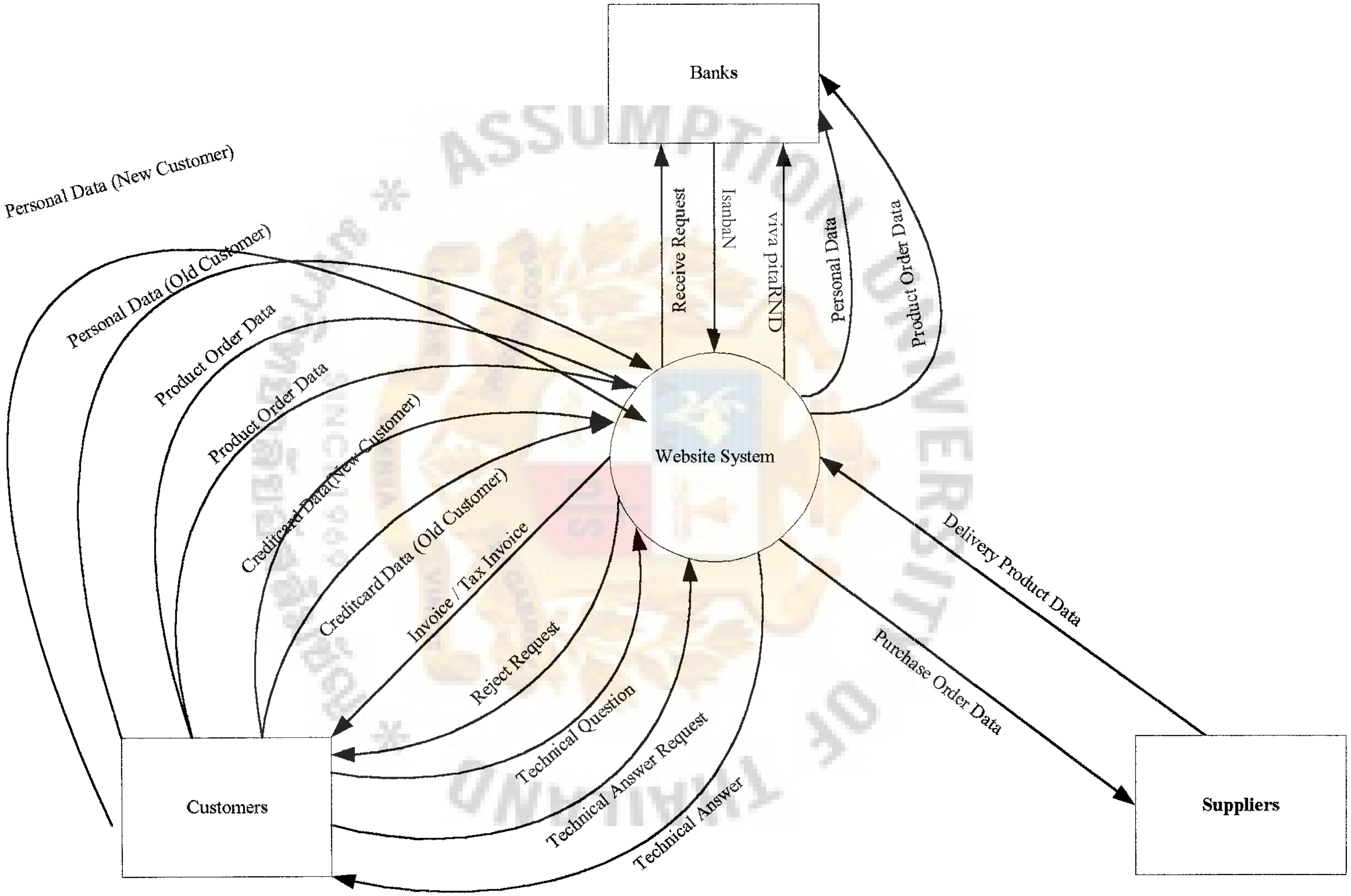
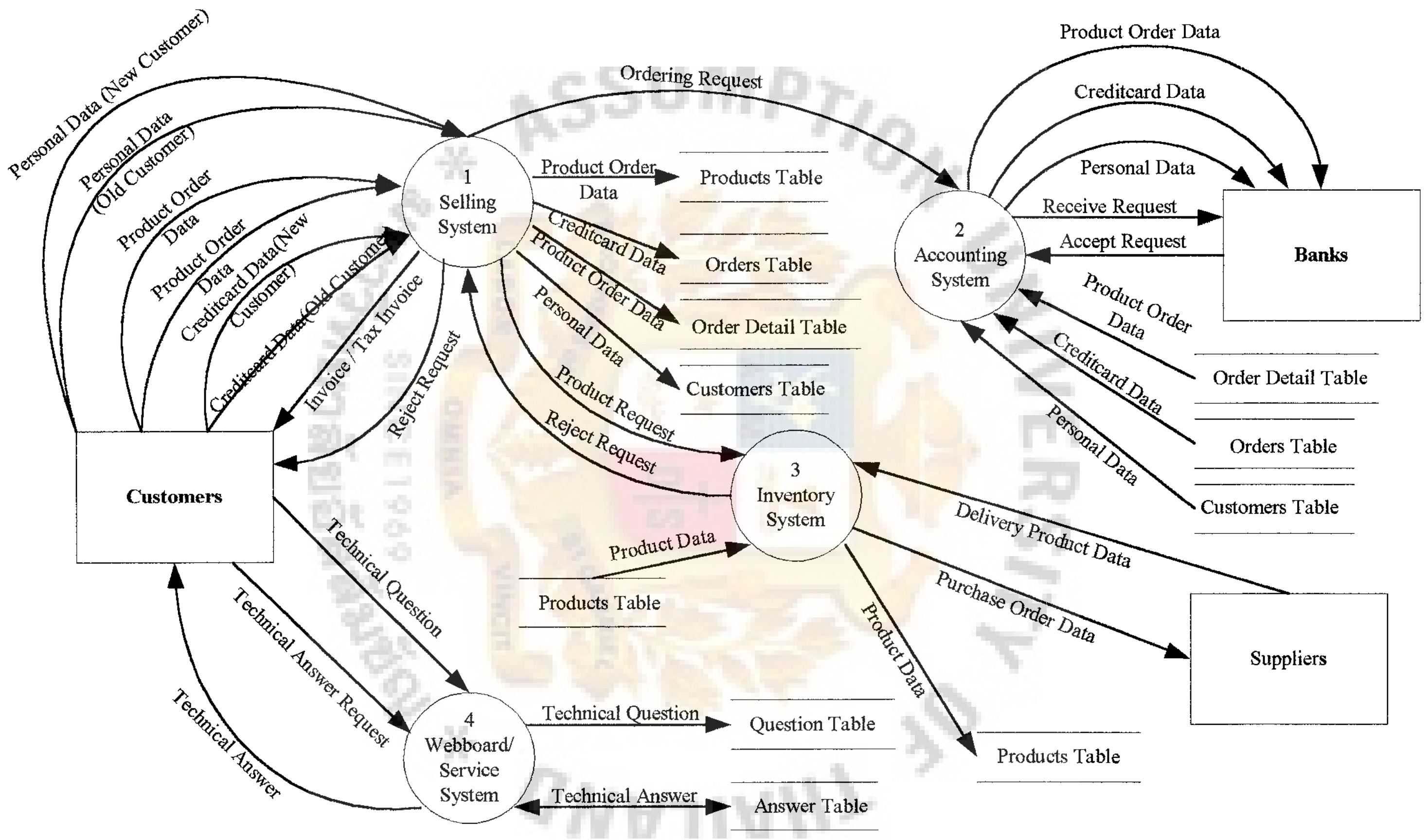
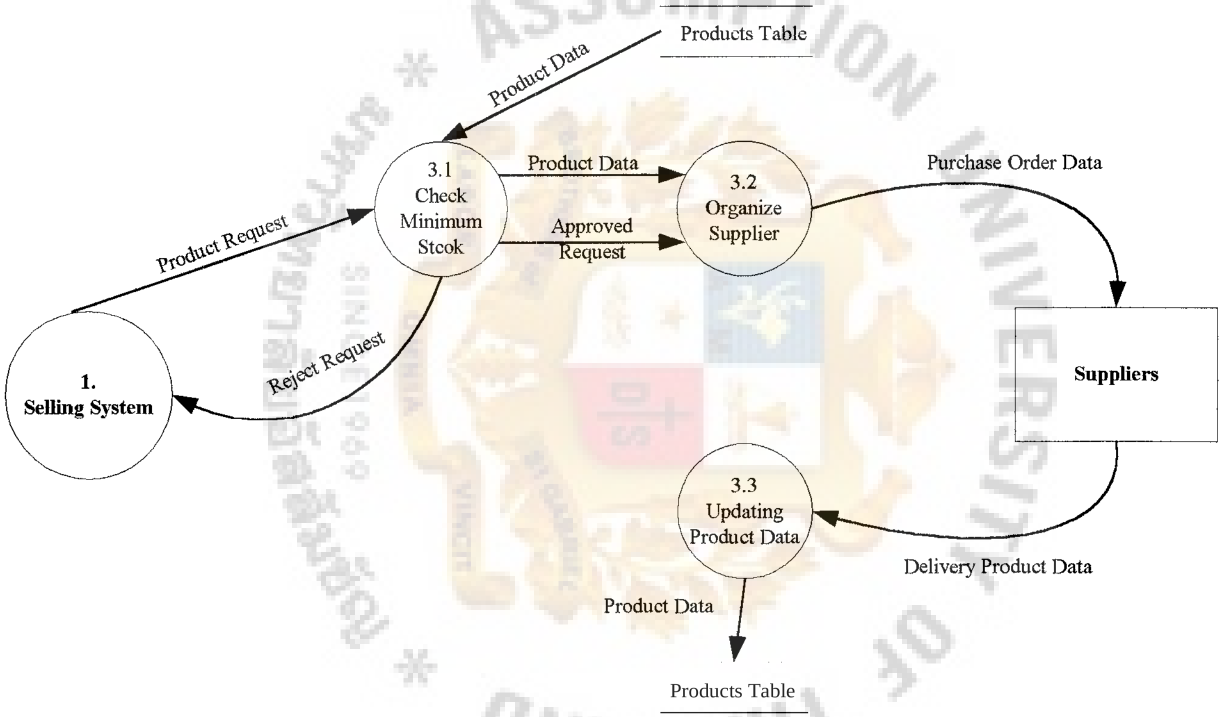


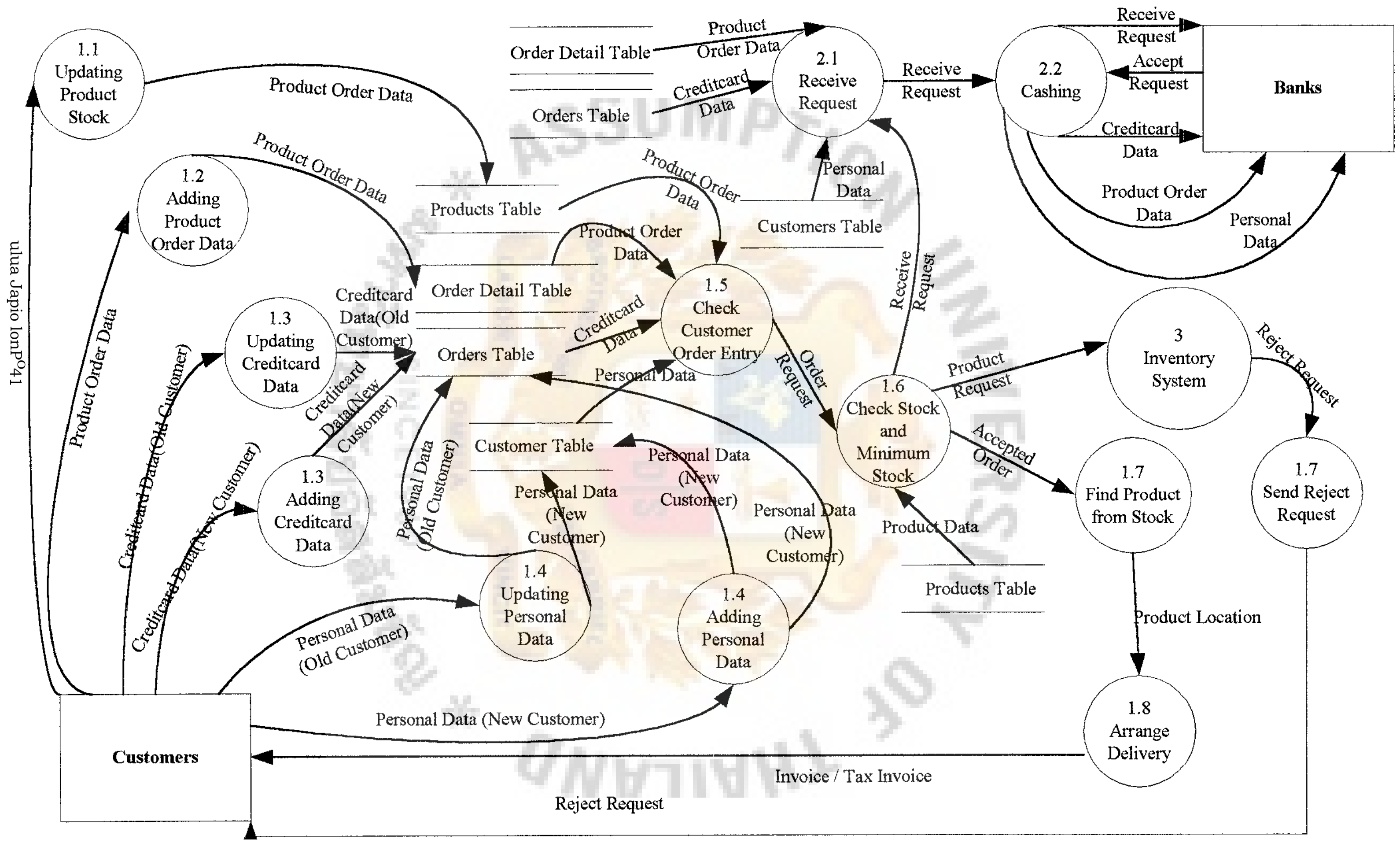
Figure A.1. Context Diagram.



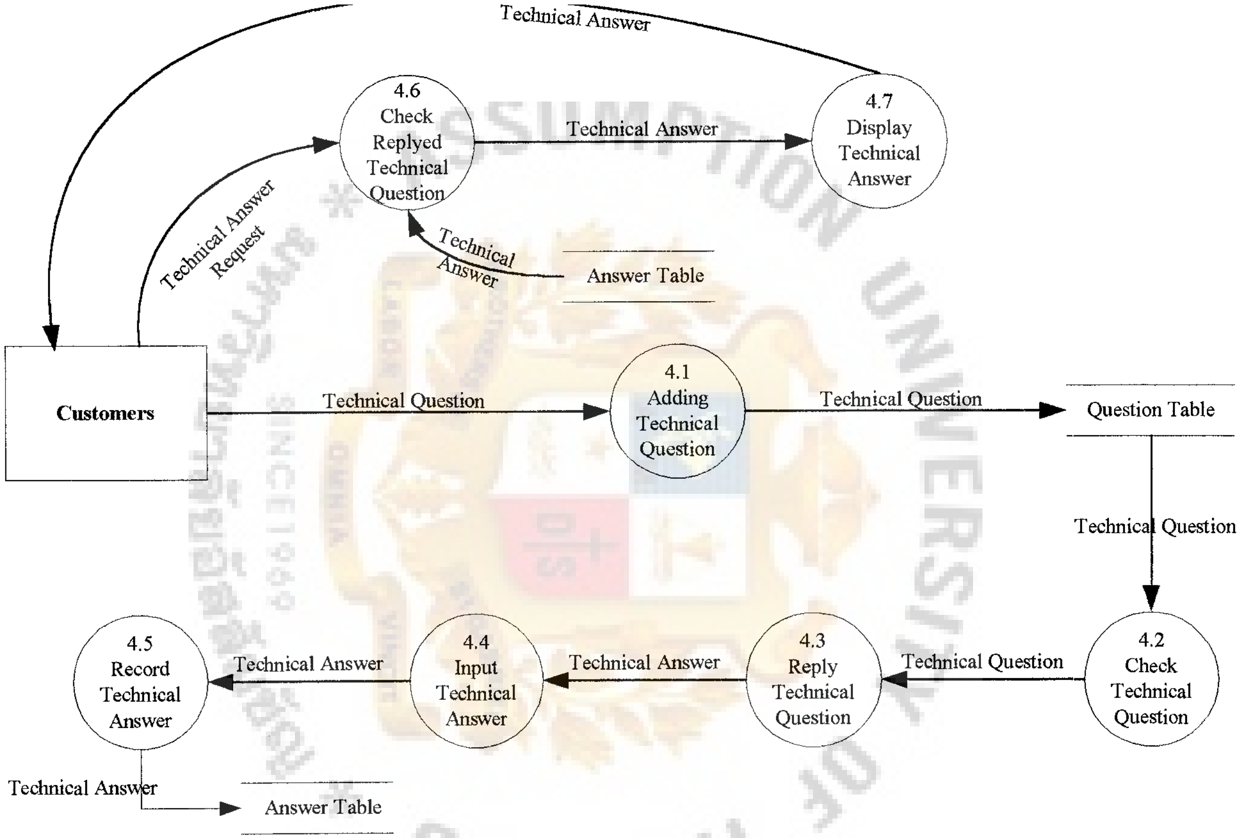
Data Flow Diagram Level 1.



12 e Lev ram iag D low F ta Da m te s Sy ry to e Inv cn



Selling System and Account System Data Flow Diagram Level 2.



an ci) Webboard System Data Flow Diagram Level 2.

APPENDIX B

ENTITY RELATIONSHIP DIAGRAM

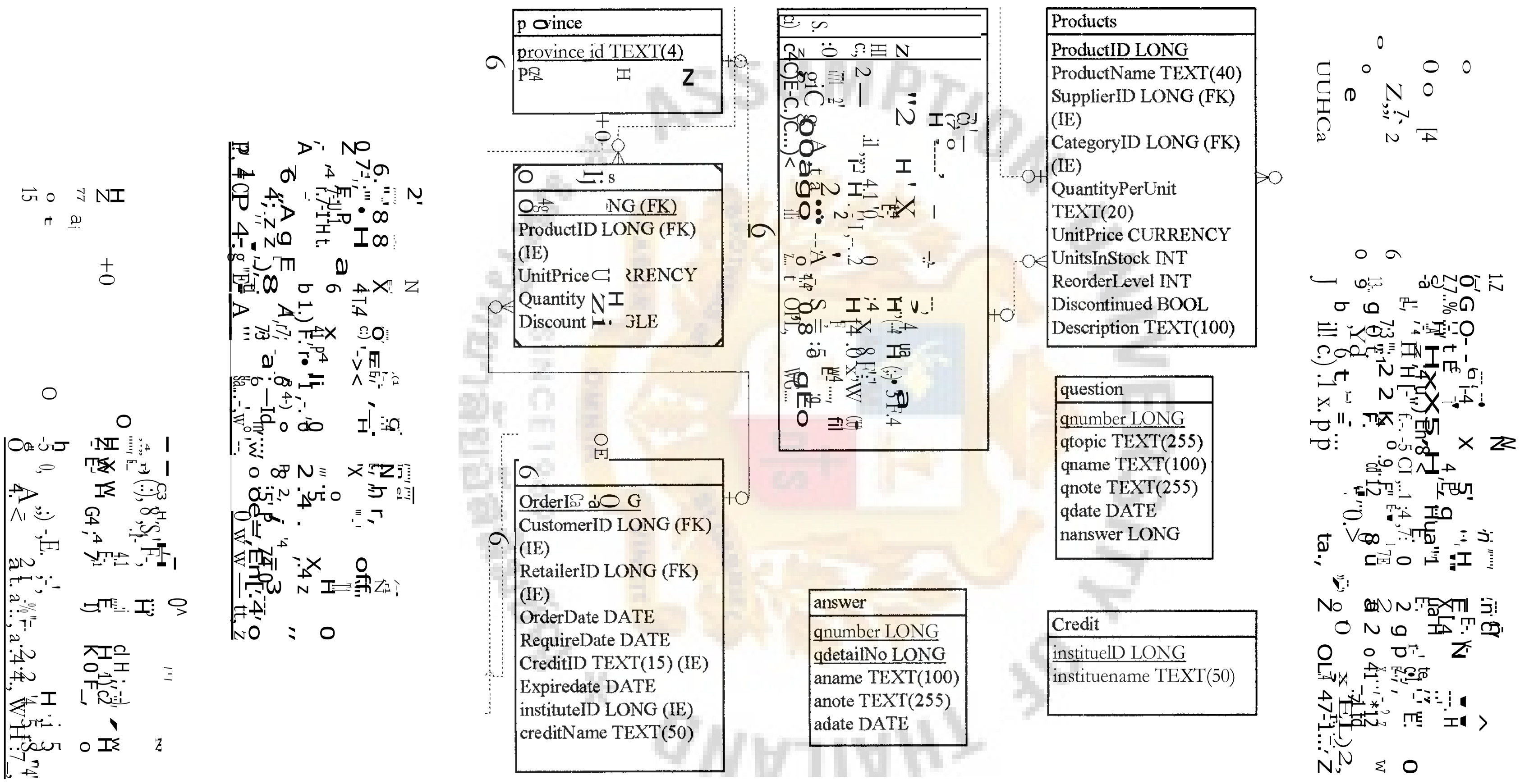


Fig re B.1. Entity Relationship.

APPENDIX C

WEB PAGE SCREEN

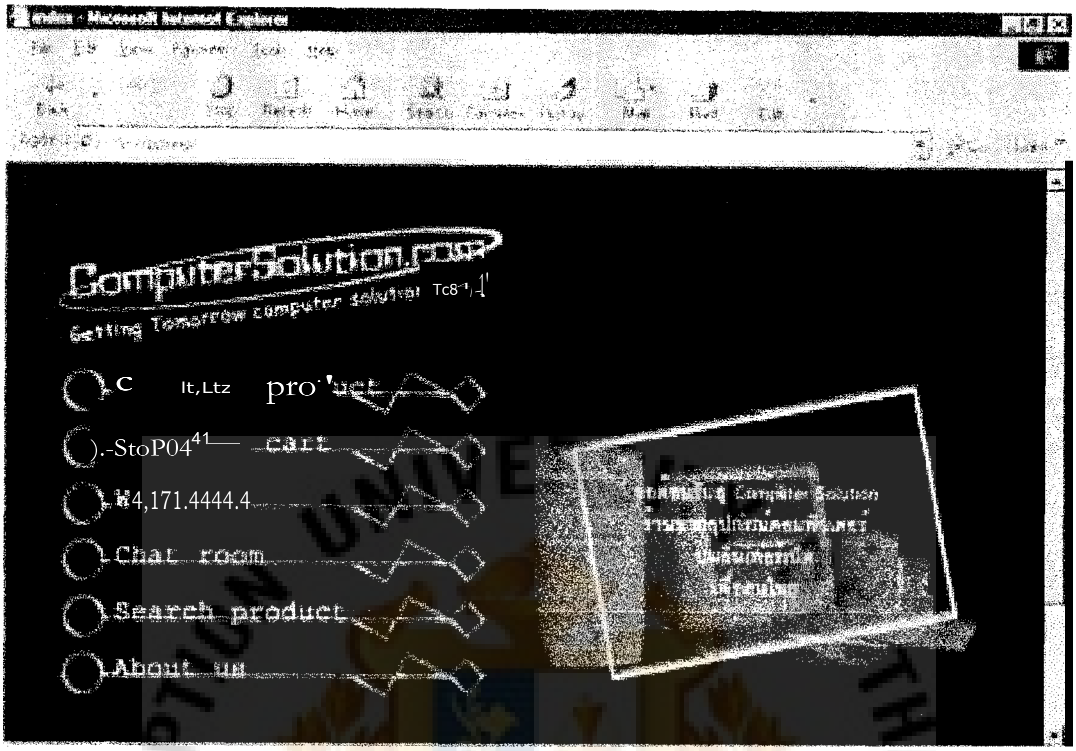
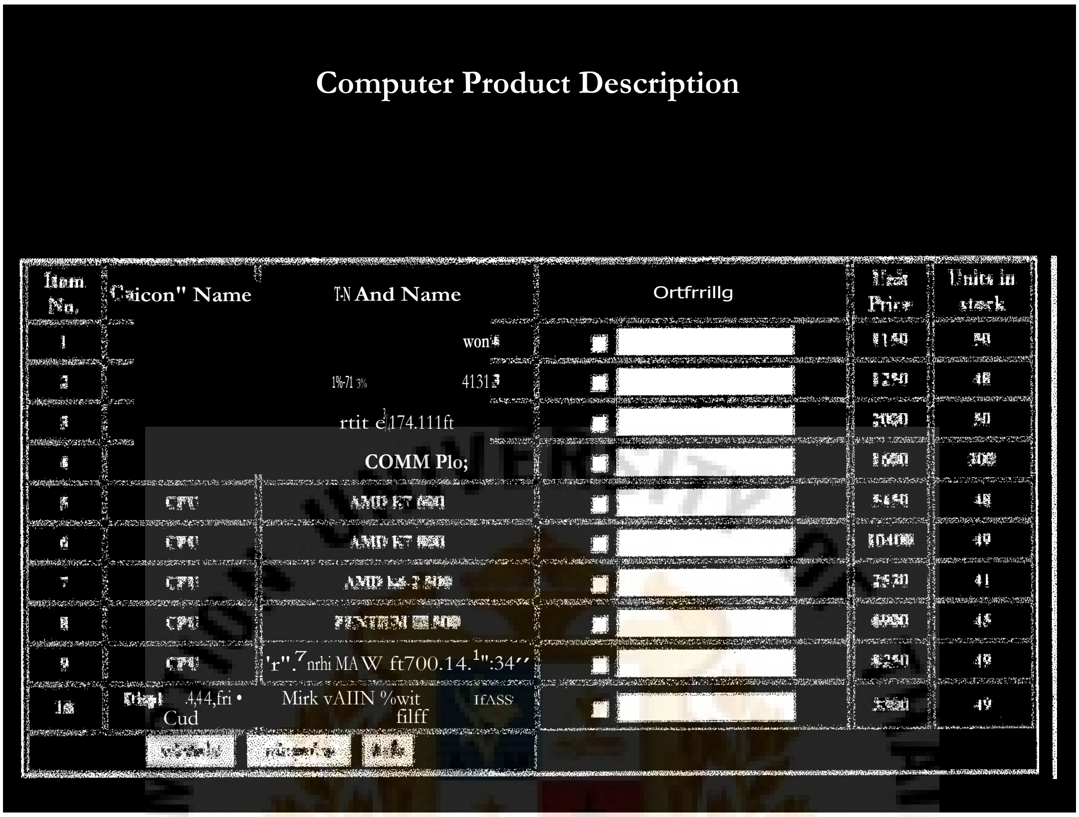


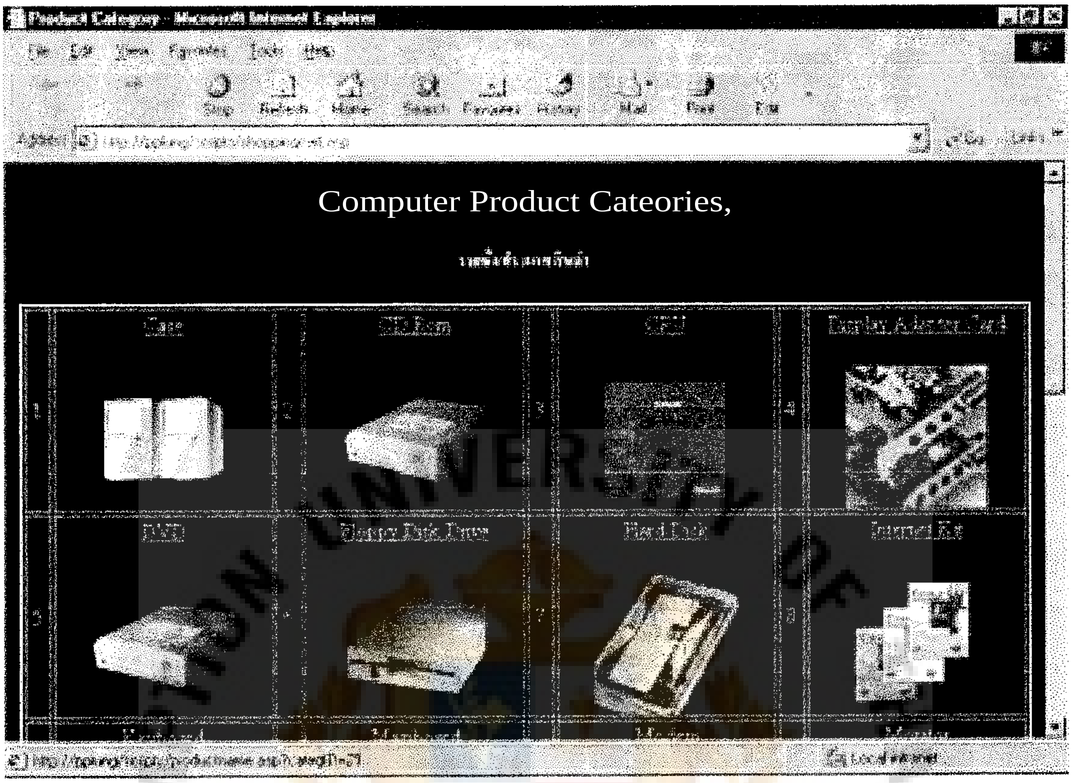
Figure C.1. Web Page Screen of Home Page.



Computer Product Description

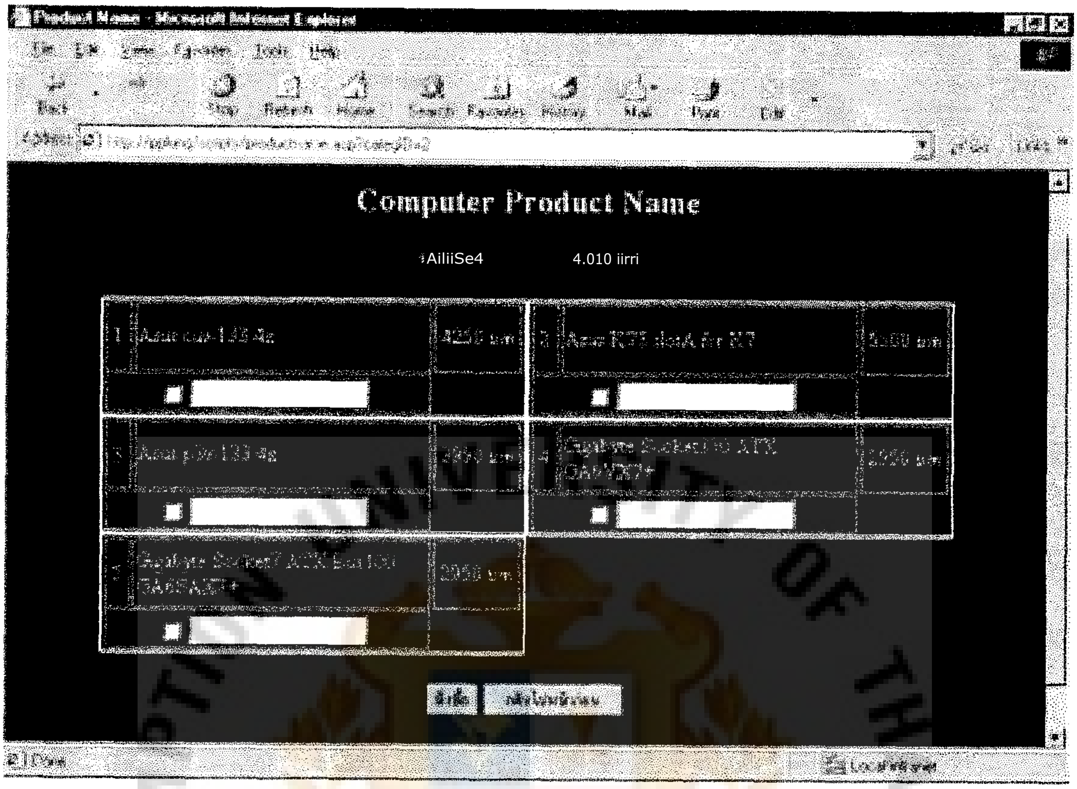
Item No.	Caicon" Name	T-N And Name	Ortfrrillg	Unit Price	Units in stock
1		won's	■	1140	50
2		1%-71 3% 41313	■	1250	48
3		rtit e 174.111ft	■	2020	50
4		COMM Plo;	■	1600	300
5	CPU	AMD K7 [illegible]	■	5450	48
6	CPU	AMD K7 [illegible]	■	10400	49
7	CPU	AMD K6-2 500	■	2530	41
8	CPU	PENTIUM [illegible] 500	■	4900	45
9	CPU	'r".7nrhi MAW ft700.14.1":34''	■	8250	49
10	[illegible] 4,44,fri • Cud	Mirk vAIIN %wit IfASS filff	■	5500	49

Figure C.2. Web Page Screen of Product.asp.



Computer Product Cateories,

Figure C.3. Web Page Screen of Shoppingcart.asp.

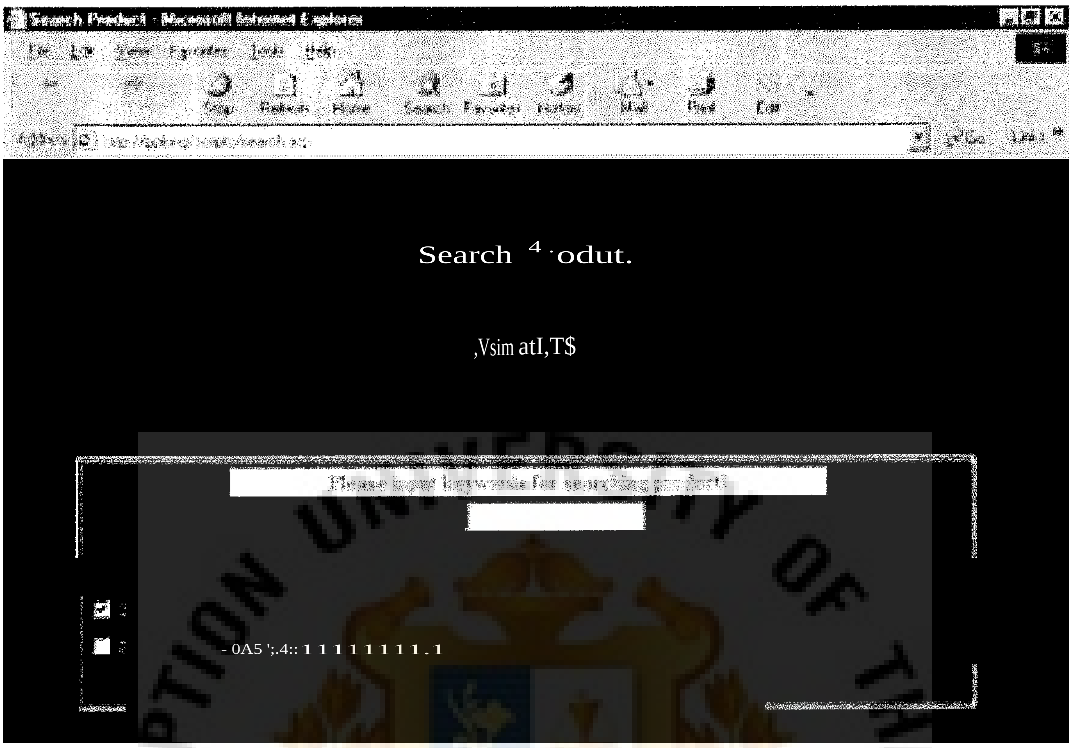


Computer Product Name

AiliiSe4 4.010 iirri

1	Asus cuv-133 4x	4250 [illegible]	2	Asus [illegible]	[illegible]
□			□		
3	Asus p3v 133 4x	[illegible]	4	Gigabyte [illegible] ATX GA6VX7+	[illegible]
□			□		
5	Gigabyte Socket7 ATX [illegible] GA65AX7+	2050 [illegible]			
□					

Figure C.4. Web Page Screen of Productname.asp.



Search Product - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Stop Refresh Home Search Favorites History Mail Print Edit
Address
Go Links

Search 4 ·odut.

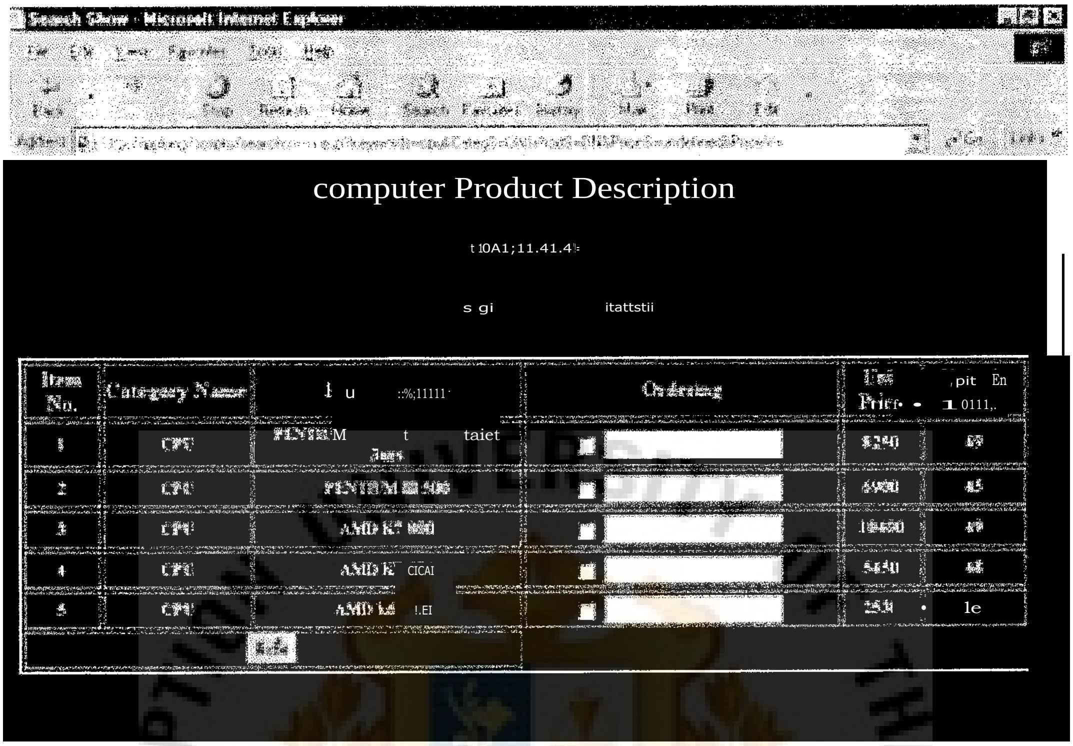
,Vsim atI,T$

Please input keywords for searching product

☑

☐ - 0A5 ';.4:: 1 1 1 1 1 1 1 1 .1

Figure C.5. Web Page Screen of Search. asp.



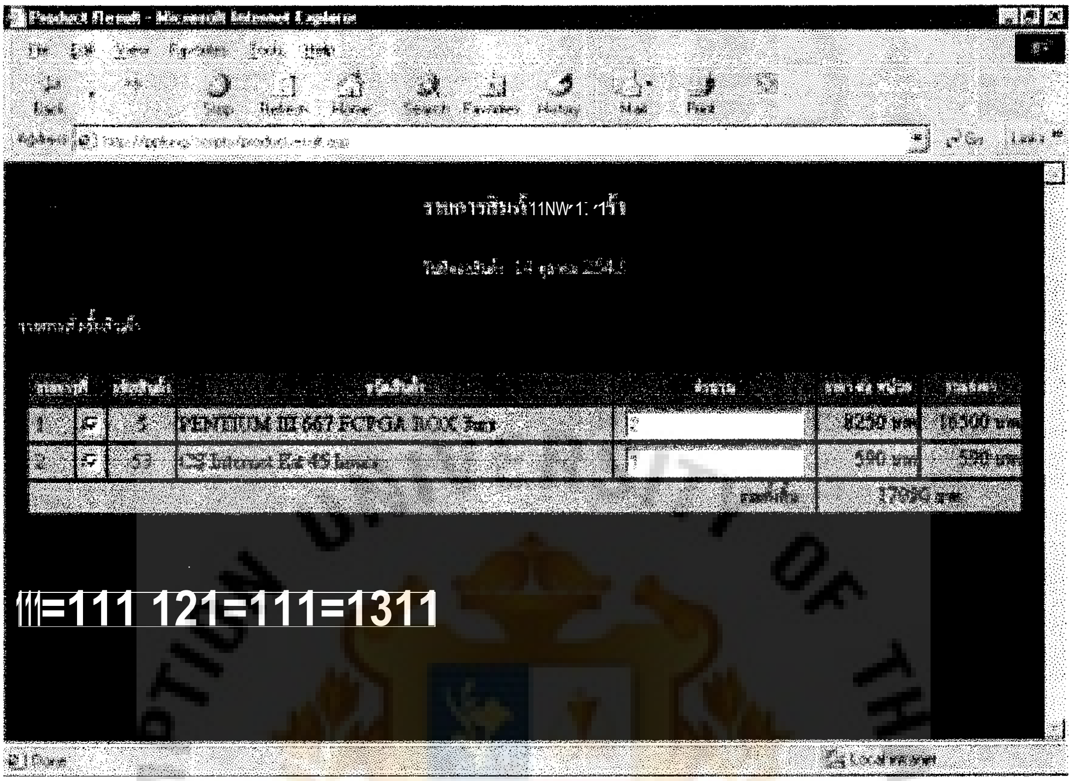
computer Product Description

t10A1;11.41.4

s gi itattstii

Item No.	Category Name	I u ::%;11111	Ordering	Unit Price •	,pit En 1 0111,.
1	CPU	PENTIUM t taiet [illegible]	☐	8290	69
2	CPU	PENTIUM III 500	☐	6900	45
3	CPU	AMD K7 800	☐	10450	49
4	CPU	AMD K CICAI	☐	5450	48
5	CPU	AMD K6 !.EI	☐	2530 •	1e

Figure C.6. Web Page Screen of searchshow.asp.



Product Result - Microsoft Internet Explorer

[illegible]11NW1: [illegible]

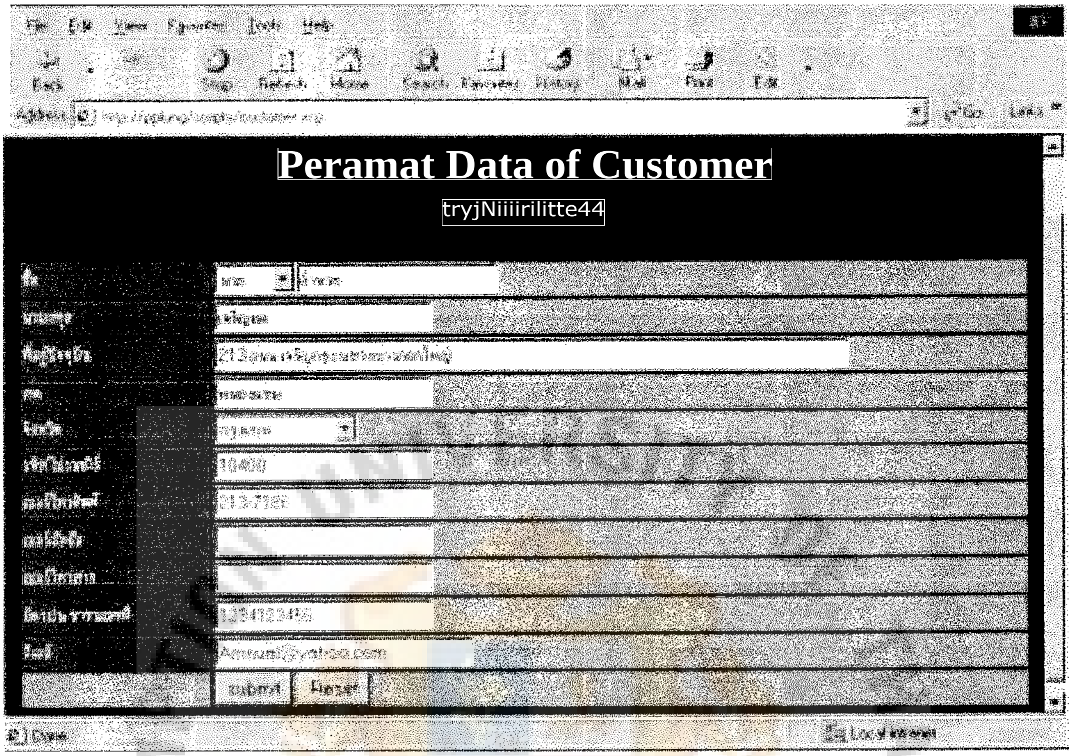
[illegible] 14 [illegible] 2545

[illegible]

[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	☑	5	PENTIUM III 667 FCPGA BOX [illegible]	2	8250 บาท	16500 บาท
2	☑	53	CSI Internet Kit 45 hours	1	590 บาท	590 บาท
				รวมทั้งสิ้น	17090 บาท	

111=111 121=111=1311

Figure C.7. Web Page Screen of Productresult.asp.



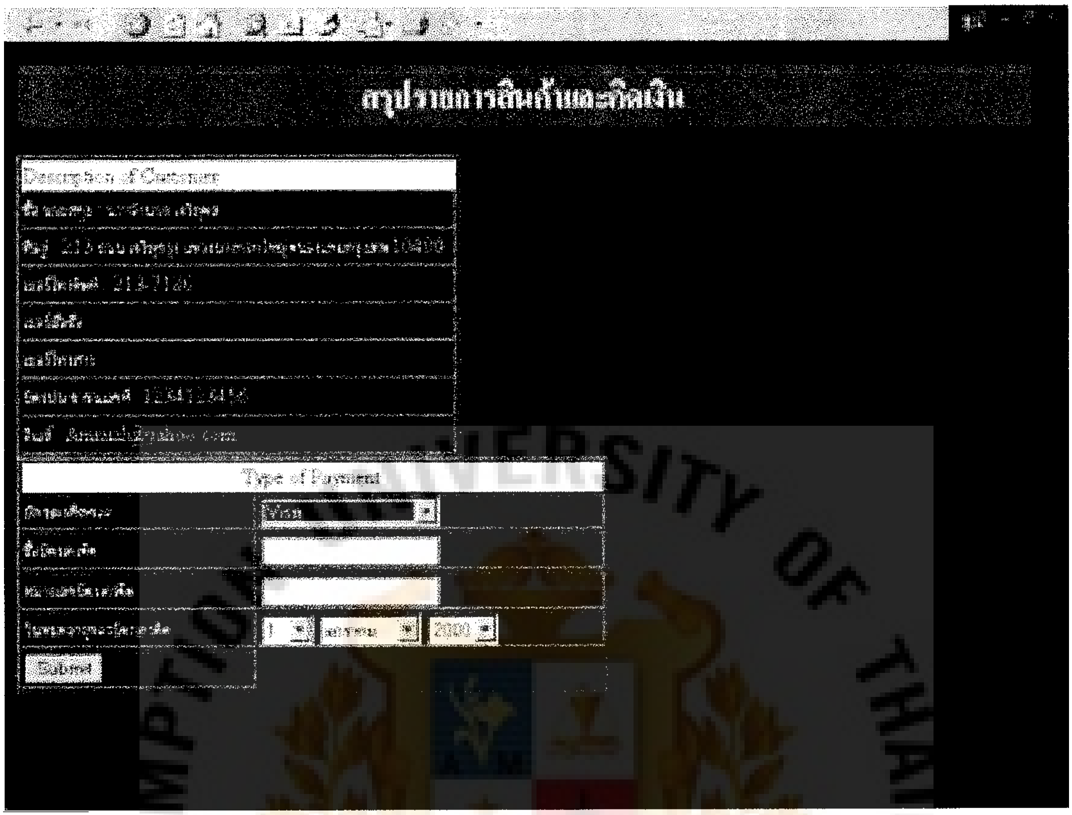
Peramat Data of Customer

tryjNiiiirilitte44

[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	213 [illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	10400
[illegible]	213-7188
[illegible]	
[illegible]	
[illegible]	1234123456
[illegible]	[illegible]@yahoo.com
	submit Reset

File Edit View Favorites Tools Help
Back Stop Refresh Home Search Favorites History Mail Print Edit
Address
Go Links
Done
Local intranet

Figure C.8. Web Page Screen of Customer.asp.



สรุปรายการสินค้าและคิดเงิน

Description of Customer

ชื่อ นามสกุล : [illegible]

ที่อยู่ 213 [illegible] 10400

เบอร์โทรศัพท์ 213-7126

เบอร์มือถือ

เบอร์โทรสาร

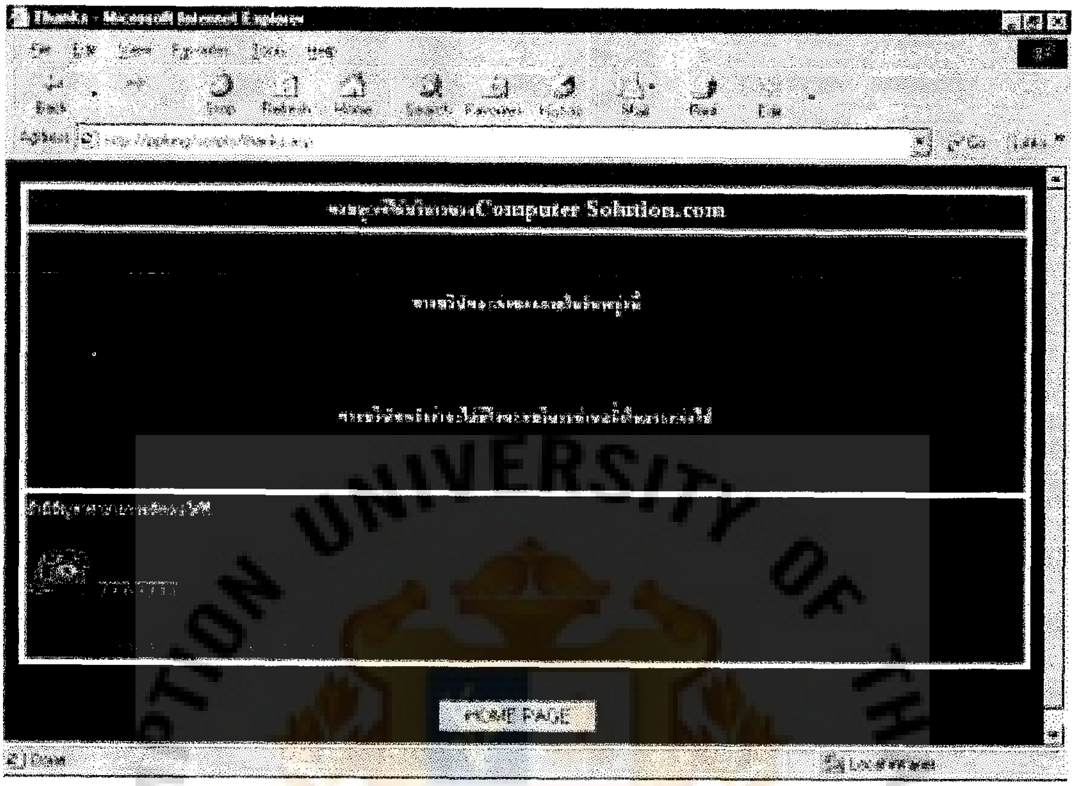
รหัสประจำตัวประชาชน 1234[illegible]

อีเมล์ [illegible]@yahoo.com

Type of Payment

ประเภทบัตรเครดิต	Visa
ชื่อบัตรเครดิต	
หมายเลขบัตรเครดิต	
วันหมดอายุของบัตรเครดิต	1 [illegible] 2000

Figure C.9. Web Page Screen of Checkout.asp.



[illegible]Computer Solution.com

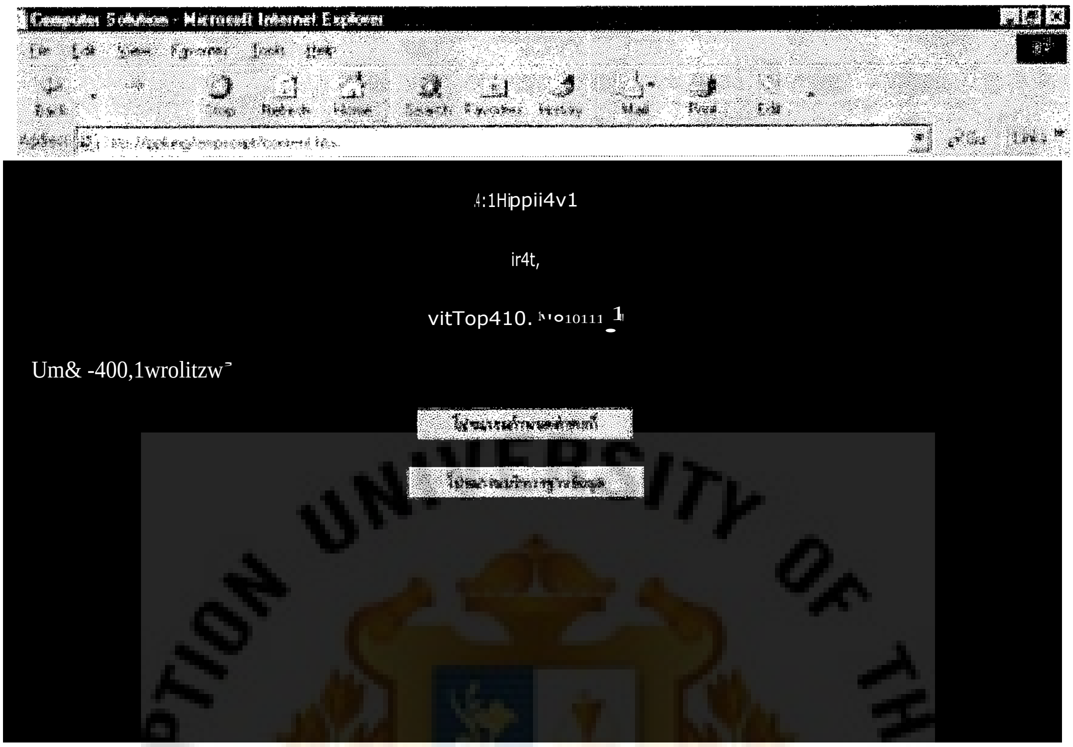
[illegible]

[illegible]

[illegible]

HOME PAGE

Figure C.10. Web Page Screen of Thanks.asp.


Computer Solutions - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Stop Refresh Home Search Favorites History Mail Print Edit
Address
Go Links

4:1Hippii4v1

ir4t,

vitTop410. ৸৹10111 1

Um& -400,1wrolitzw¯

Figure C.11. Web Page Screen of Content.htm.

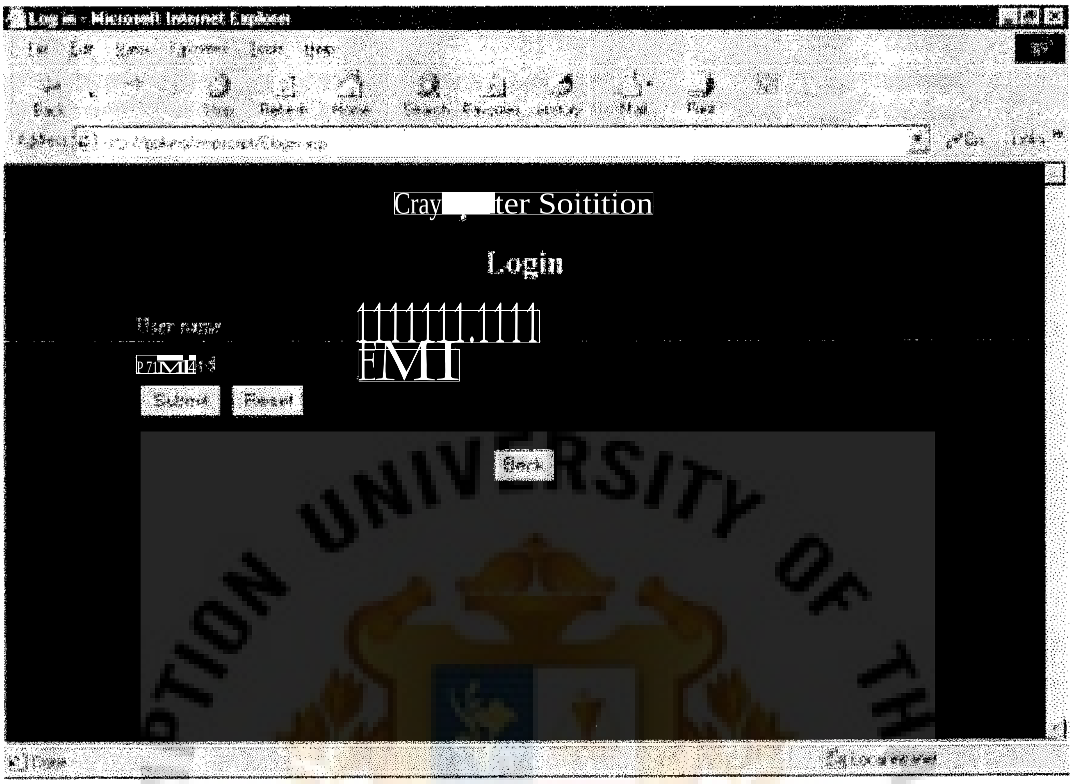
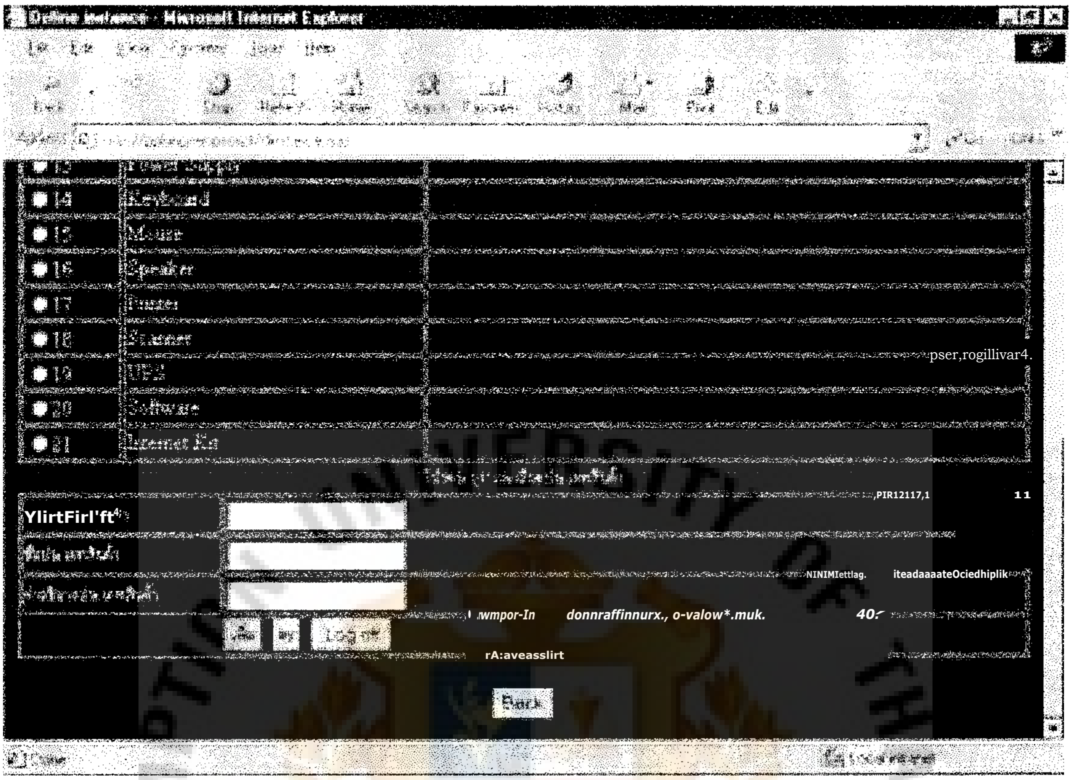


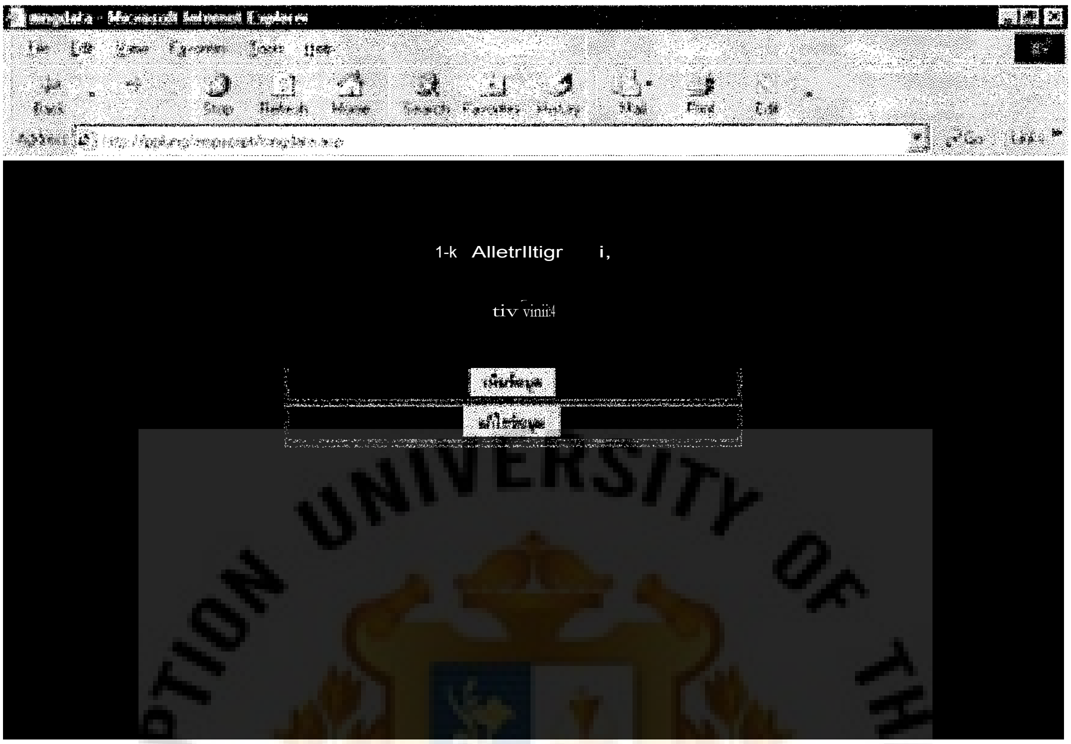
Figure C.12. Web Page Screen of Elogin.asp.



Define Instance - Microsoft Internet Explorer

● 13	Power Supply
● 14	Keyboard
● 15	Mouse
● 16	Speaker
● 17	Printer
● 18	Scanner
● 19	UPS
● 20	Software
● 21	Internet Kit

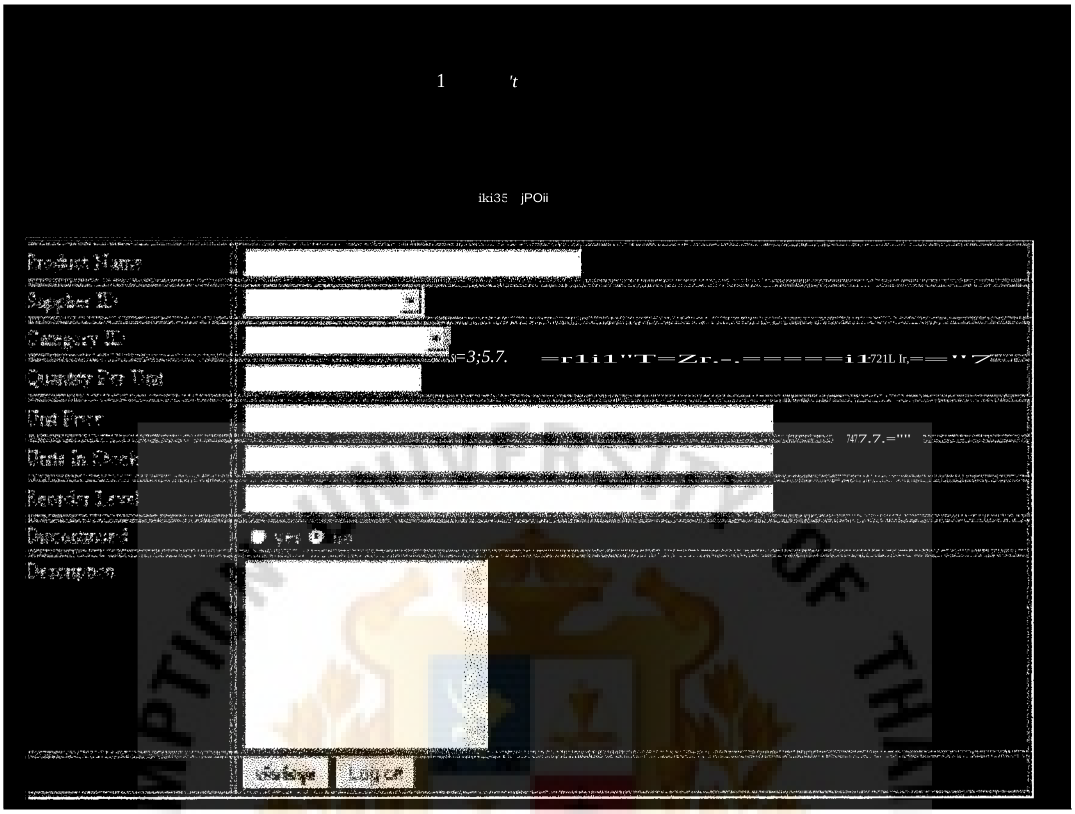
Figure C.13. Web Page Screen of Dinstance.asp.



1-k Alletrlltigr i,

tiv vinii4

Figure C.14. Web Page Screen of Mngdata.asp.



Product Name

Supplier ID

Category ID

Quantity Per Unit

Unit Price

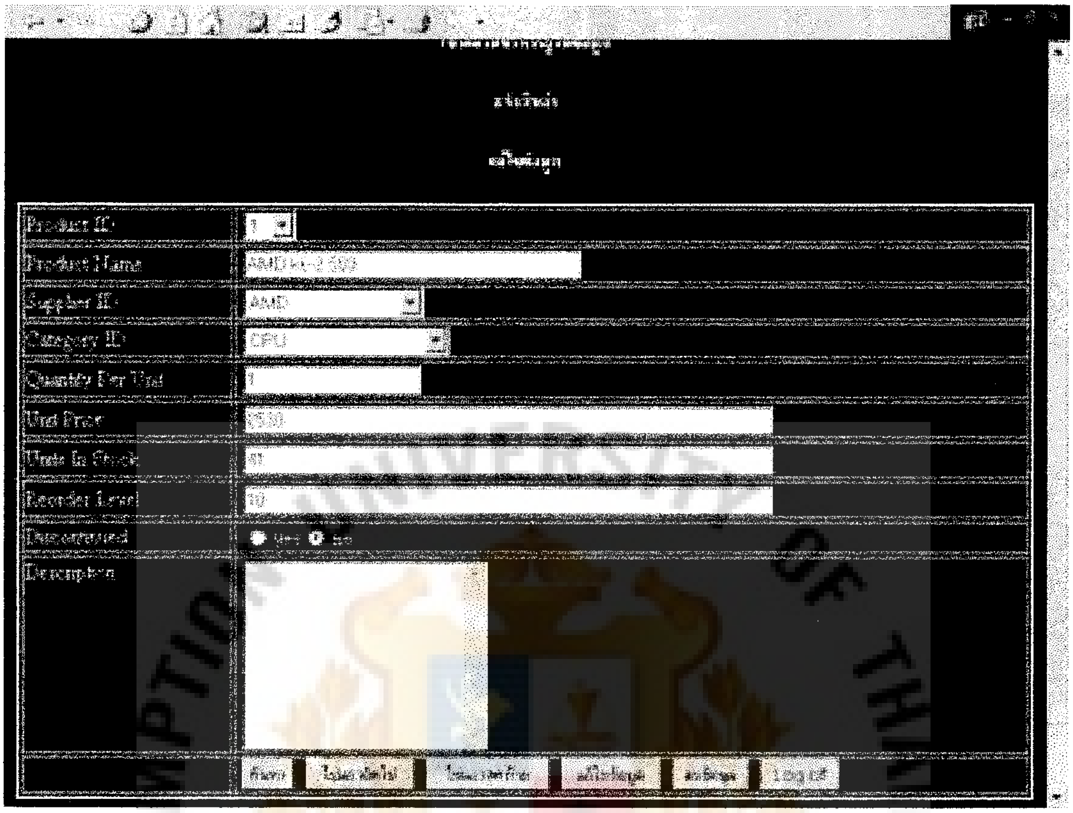
Units In Stock

Reorder Level

Discontinued ○ yes ● no

Description

Figure C.15. Web Page Screen of Mngadd.asp.



Product ID	1
Product Name	AMD K6-2 500
Supplier ID	AMD
Category ID	CPU
Quantity Per Unit	1
Unit Price	2500
Units In Stock	41
Reorder Level	10
Discontinued	○ yes ● no
Description	

Figure C.16. Web Page Screen of Mngupdte.asp.

BIBLIOGRAPHY

1. Eliason, Alan L. Systems Development: Analysis, Design, and Implementation. Chicago: Harper Collins Publishers, 1990.

2. Gelinas, Ulric J. Accounting Information System, 3rd Edition. Ohio: South Western College Publishing, 1996.

3. Iverson Computer Training Center. Microsoft Access 97: Introduction to Database, 1998.

4. Johnson, Scot. Using Active Server Pages. Indiana: Que Corporation, 1997.

Kendall, E. K. and E. J. Kendall. System Analysis and Design, 3rd Edition. New Jersey: Prentice-Hall Inc., 1995.

Schermerhorn, John. Management, 6th Edition. New York: John Wiley & Son, 1997.