MANIAM @ KUMAR A/L MALLANAIDU
WQT000050

ASP AUTHENTICATION USING IP ADDRESS

WXES 3182: LAPORAN PROJEK ILMIAH TAHAP AKHIR

Faculty of Computer Science and Information Technology

2003/2004

ABSTRACT

The project undertaken is ASP Authentication Using IP Addressing; the main objective of this project is to introduce the authentication technique for the websites, which are developed for any commercial organizations. This project just demonstrates the core key concepts of password authentication, data encryption. The project is developed using the Active Server Pages, which has the effective features to demonstrate and implement the authentication techniques. The technical concept of this project lies in providing security. There are many ways to implement the security concepts; one of the concepts we have chosen is the authentication technique using the IP (Internet Protocol) addressing.

The main aim of this project is to implement the secured features for the users entering into a domain website. Here each aspect of the end user is controlled by the website server administrator. The logical reason is to control the user by giving them limited rights to access any entity or organization's information. Networking features behind this project plays an important role in identifying the end user location. Here the end user access to the server is monitored using the Internet Information Server, which supports the server side scripting effectively. The project is fully programmed using the server side scripting techniques, which forms a powerful basis for the web server to manage any number of users entering into the system.

ACKNOWLEDGEMENT.

I would like to take this opportunity to express my heartfelt gratitude to everyone who helped me to complete my thesis report.

Foremost, I wish to forward my deepest thanks and appreciation to my supervisor, Dr. Rosli Salleh for his unique approach in coaching me, giving constructive criticisms and always having enough time from his hectic schedule to meet and discuss about the report.

I would also like to thank En.Mustaffa Kamal for being my moderator. His ideas and comments during my viva were noted and all taken into account so that I could produce a quality report.

My thanks to Mr.D.Jayachandran, an IT consultant for his views and advice throughout my project.

Last but not least, thank you to my dearest wife, my lovely daughter Ramhyaa, my adorable son Tharsann, my beloved parents, my fellow course-mates and friends for being there with me all the time.

Table of contents

Chapter 4. SYSTEM DESIGN

Chapter 5. SYSTEM IMPLEMENTATION

Chapter 6. SYSTEM TESTING

Chapter 7. SYSTEM EVALUATIONS AND CONCLUSION

Table of Figures

Tables of Tables

Chapter 1 – Introduction

1.1 Overview

In a networking environment there must be assurance the sensitive data will remains private so that only authorized users can access it. Not only is it important to secure sensitive information, it is equally important to protect network operations. Every network needs to be kept safe from deliberate or unintentional damage. However, a good network will remember that security requires a balance. A network does not need to be so secure that people have difficulty using it to get their work done. They should not be frustrated trying to get work done.

Even though networks handle the most sensitive and valuable business data, data security is sometimes an after thought. Four major threats to the security of a network are:

- Unauthorized access
- Electronics tampering
- Theft
- Intentional or unintentional damage

Despite the reality of these threats, data security is not always understood or supported properly. It is the administrator's job to ensure that the network remains a reliable, secure business tool free from those threats.

Over the years the Internet technology plays as a dominant role in the world market. The technology trend on information has revolutionized the world of information technology in various aspects. The increased needs of individual organizations have created a demand for the technology to play an important role in the domain of security. The open source of information in World Wide Web has created more facilities to access any volume of information at any time by any user without restriction. Today the organizations are seeking for advance security features in order to restrict the limited access of confidential information related to accounts, transactions and business deals over the net. Over the years the term security is given more highlighted importance but till today there are no concepts of complete security, companies and software organizations are striving hard to develop the hardcore security features that could at least control the wide area network. The security over the networking concepts also depends upon the type of network and the operating system used. The primary concept of security is enhanced by the development and usage of the network protocols like TCP/IP, RC4, PPP and ARP etc.

With the vast array of sensitive transactions occurring on the Web today, security is of the utmost importance. Organizations plan on asking their clients for personal or financial information, protecting that data from outside forces is essential. Furthermore, there are times when you want to prevent a set of users from accessing certain information on your site. Both of these scenarios fall under the Security category. Authentication features introduced in websites makes only the valid user to access certain restricted information. There are many ways to authenticate visitors entering into website by using IIS settings, cookies, or third party components. A TCP/IP based schema presented in this thesis here assumes the IP address for a given machine remains constant. For most networks it is constant. For a few networks IP addressing is dynamic where the address will change for a machine with each start up, and sometimes even during operation. The security schema presented in this thesis works for a fixed range of IP addresses dynamically managed by the internal server within an organization network. Once verified that the user has permission to access the application the user is given a time stamp and a limited access to the information. ASP Authentication using IP plays an important role in this area. This ASP Authentication works better when a website or a web server hosted using the ASP concepts on the windows network local area network/wide area network/ worldwide web (LAN/ WAN/ WWW).

1.2 Objective

The main objective of this project is to introduce the security features using the ASP Authentication concepts with TCP/IP protocol. This project is mainly concentrated on the password authentication and the data encryption features.

1.2.1 Password Authentication.

- The password authentication is mainly developed using the concept of cookies.
- The security method presented here comes close to the simplicity of cookie security without the headaches of losing a cookie - or cookies being turned off by the user.
- The cookie is often tracked by the IP address maintained by the use of TCP/IP protocol.

1.2.2 Data Encryption

- The concepts of data encryption make a transmitted data to a more secure data.
- The data encryption basically decodes the transmitted data to a machine understandable format so that the external user is unable to detect the transmitted data for more safety and security reasons.

1.3 Scope

- The project demonstrates the usage of security features for most of the websites where heavy data transactions happen.
- This project is specially developed for the website that uses the ASP as their server.
- The concept of this project can be effectively used for any financial sectors and other commercial sectors that are involved in business to business, business to customer, customer to customer.

1.3.1 Reason For Fixed IP Inside Organization.

The security schema presented in this thesis works better for a range of fixed IP address managed dynamically within the organization network.

Each time when the user try to access the organization server through their organizations website, password authentication enables the feature to check the authorized entry into the network. The server manages the internal database and give an access to the website when the user name and password matches it.

In case of using direct dynamic addressing within an organization is no a healthy network management because the server may allow any user without detecting or allocating an IP for the log on user. This gives a way to access the organization information outside without any restriction so it not advisable to have dynamic IP addressing for the internal server.

1.4 Project Schedule.

Project schedule is a schedule of the whole activities of project development. It is planned out carefully to make sure a systematic progress and on-time delivery of the project is achieved. Project schedule is very important in order to have a guide of time management to a developer. Besides, developer also can know whether in route of the direction of the project or not. To make certain that this project is completed in time, a project schedule was done. The various stage involved during the project duration are as stated.

Table 1.0 Phase Duration Schedule.

Stages/Phase	**Duration (Weeks)**
Preliminary Study and Planning.	2
Literature Study	5
System Analysis	8
System Design	7
Prototype	8
Development and Coding	11
System Testing	12
Documentation	25
Implementation and maintenance	18

A Gantt chart is shown to describe in detail the project milestone. The chart is shown below.

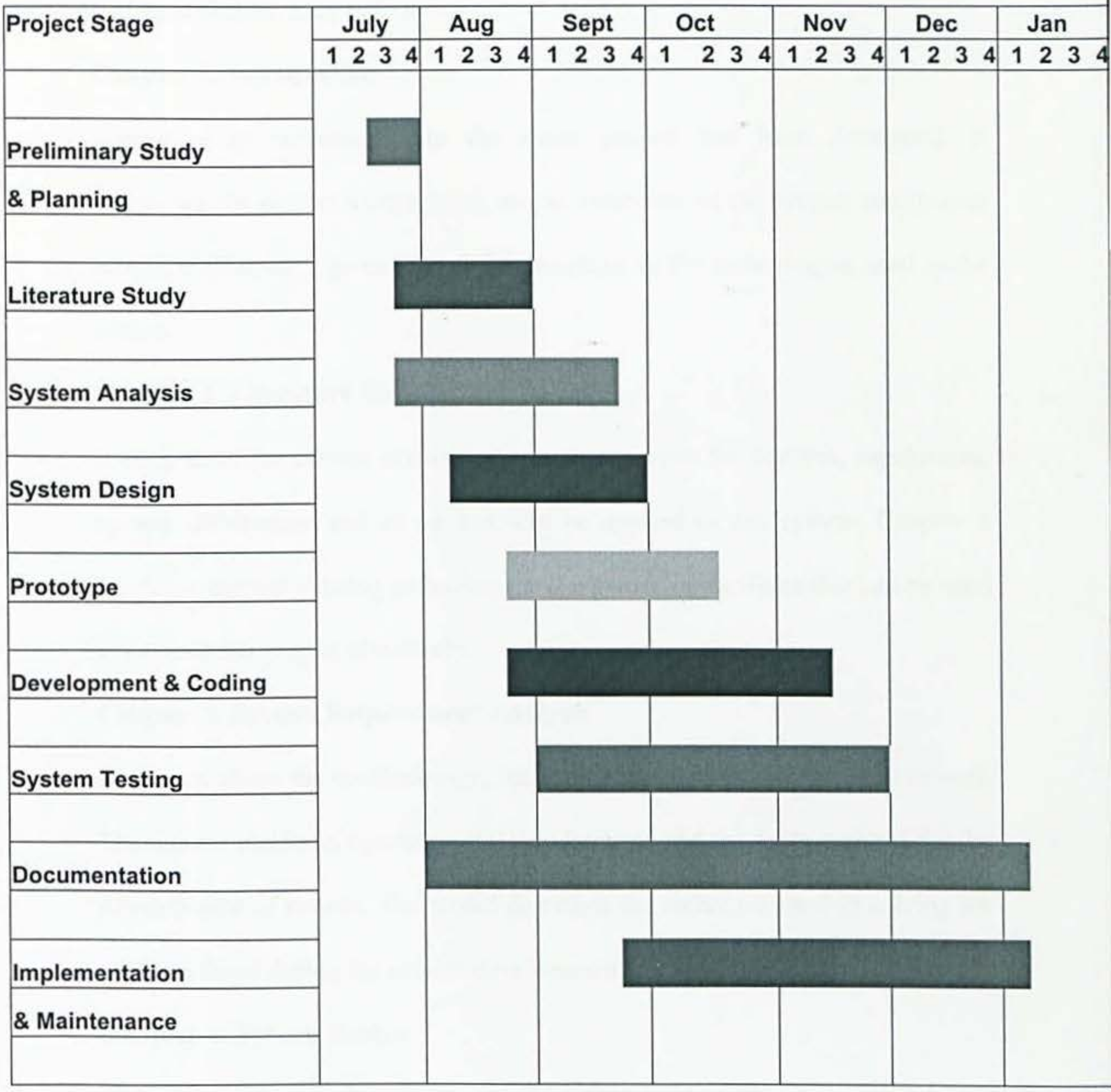


Figure 1.0 Gantt chart of Work Schedule

1.5 Chapter Organization

The purpose of this report is to document all the essential information gathered and used to develop this system. This report is mainly divided into four chapters. A brief synopsis of each chapter is as follow:

Chapter 1. Introduction

Serves as an introduction to the entire project that been developed. It overviews the project's objectives, scope, overview of the system and project schedule. Chapter 1 gives the brief description on the technologies used in the project.

Chapter 2. Literature Review

It talks about the current existing system that reviews the features, capabilities, system architecture and so on that will be applied to this system. Chapter 2 briefs the current existing technology and networking facilities that can be used to develop the project effectively.

Chapter 3. System Requirement Analysis

Discusses about the methodology, functional and non-functional requirement. The chosen platform, operating system, database and the tools requires for the development of system. The model describes the technique used in solving the problem faced during the project development.

Chapter 4. System Design

This chapter discuss about the system architecture, data flow diagram, database design and shows the exact view of the project layout and how the system works and gives a brief descriptions on the database designs and logical layout for the working model.

Chapter 2 – Literature Review

2.1 Overview

In this chapter, various information from many source regarding the development tools, web features and applications, information on database, existing cases and loads of other data relevant to the project was sought and found. Here, all the information gathered is carefully read and processed in order for the analysis to begin.

It is safe to say that Literature Review is the core of this proposal. Without the Information procured during this stage, it's almost impossible to move forward and conduct the analysis and design.

The purpose of this case study is to understand how the password authentication and data encryption effectively works in a particular domain.

This case study I demonstrate the effective use of these concepts within a workgroup or domain. Password authentication in most of the websites traces log in using username and password. The purpose of providing log in is to make user to limit his access within an organizations network.

A literature review of a project is important as it places the project in the context of others, which might have similar characteristics:

- It offers the developer of using the best way to access and analysis information regarding their research topic.

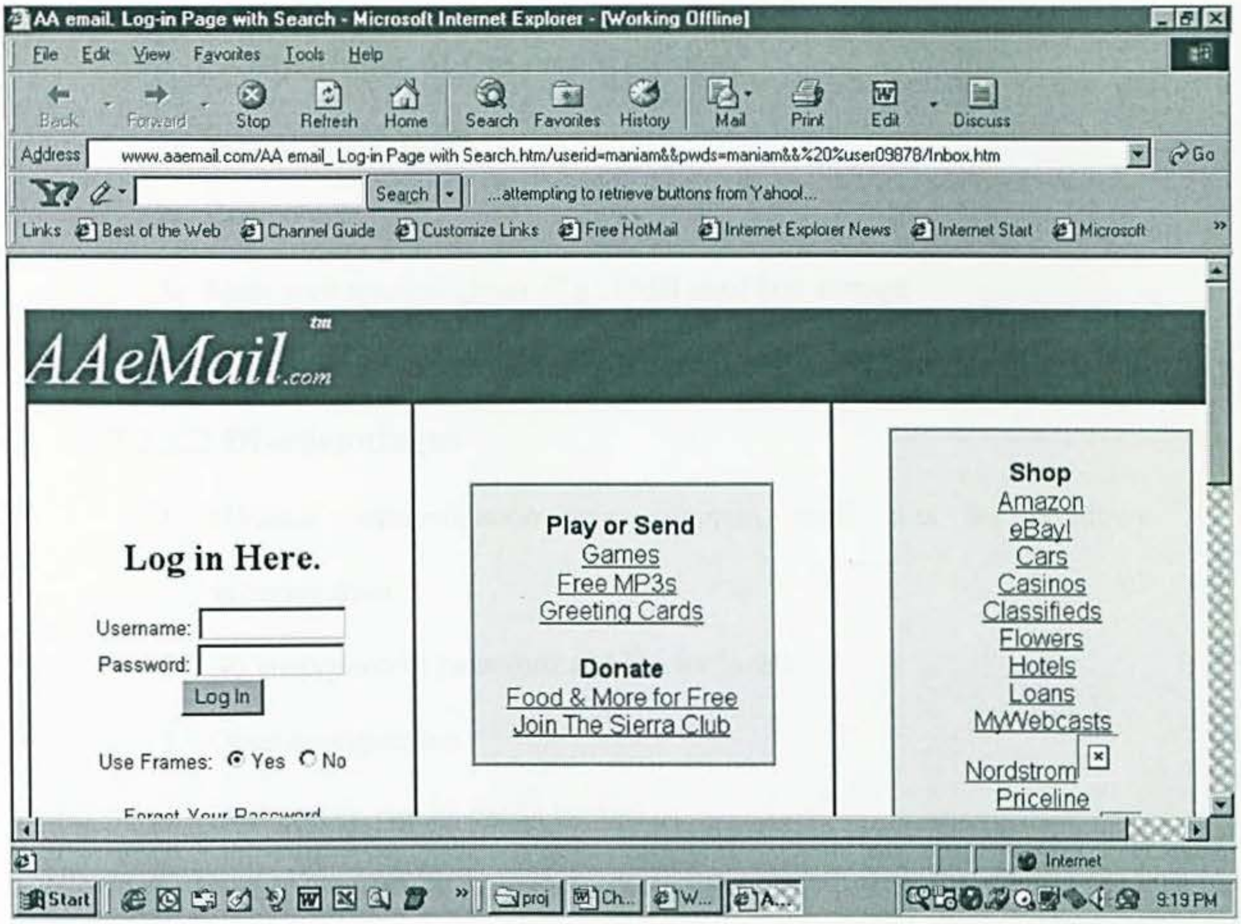
  There is no use of reinventing the wheel that has already been invented. The developer can rather focus on learning the existing system and modify or enhance it into a more powerful feature for the project.

- Another important purpose of literature review is to sufficiently equip the developer with some knowledge of the strengths and limitations of several development tools. This can help the developer to choose the right tools to develop the system.
- It also helps the developer to recognize the relevant information and synthesize and evaluate it according to the guiding concept.

2.2 Analysis Studies

2.2.1 Case Study 1

www.aaemail.com, AaeMail.com is a dotcom organization; they provide the free email facility in order to popularize themselves in the market. Free website login access is shown figure 2.0 below.



AA email. Log-in Page with Search - Microsoft Internet Explorer - [Working Offline]

Address: www.aaemail.com/AA email_ Log-in Page with Search.htm/userid=maniam&&pwds=maniam&&%20%user09878/Inbox.htm

...attempting to retrieve buttons from Yahoo!...

Log in Here.

Username:

Password:

Use Frames: ● Yes ○ No

Forgot Your Password

Figure 2.0 AaeMail Website

The above prototype illustrator shows the login page of AaeMail.com, here in this website free entry of users are given world wide, they open up the user account immediately when they register into the domain in just 3 minutes, but the major drawback for this website is that they lack in authentication. The concept behind the user access is very simple; here even the password is not encrypted. Though windows server provide authentication, the above web page serve the user with minimal authentication level.

2.2.1.1 Advantages of the AaeMail.com

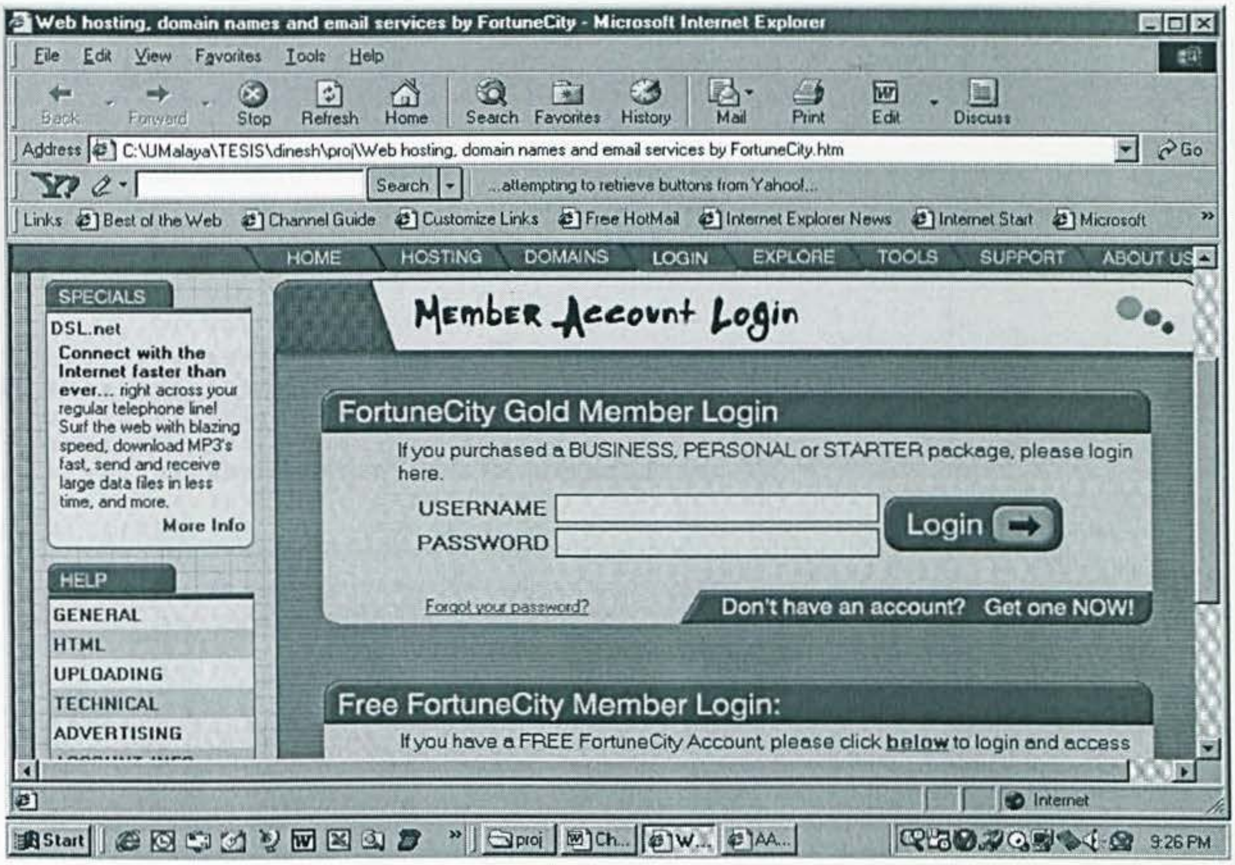
1. Easy access to email.
2. Easy to use
3. More web space is given. E.g.: 6MB mail box storage

2.2.1.2 Disadvantages

1. Minimal Authentication level, domain itself uses the windows authentication.
2. No encryption in password and in user levels.
3. Open configuration
4. Less secure, can be easily hacked.

2.2.2 Case Study 2

www.fortunecity.com the fortunecity is the dotcom company registered as FortuneCity.com, Inc. **Ampira** Corporation. The corporation deals with free domain hosting services. The prototype of their view is shown below.



Web hosting, domain names and email services by FortuneCity - Microsoft Internet Explorer

Address C:\UMalaya\TESIS\dinesh\proj\Web hosting, domain names and email services by FortuneCity.htm

...attempting to retrieve buttons from Yahoo!...

DSL.net

Connect with the Internet faster than ever... right across your regular telephone line! Surf the web with blazing speed, download MP3's fast, send and receive large data files in less time, and more.

FortuneCity Gold Member Login

If you purchased a BUSINESS, PERSONAL or STARTER package, please login here.

USERNAME

PASSWORD

Free FortuneCity Member Login:

If you have a FREE FortuneCity Account, please click **below** to login and access

Figure 2.1 Fortunecity Website

The above website is shows the user login page for the member who has the account, but the critical part of it is that any user can register and play with the website, the website administrator has given no access limits of restriction to the above website, but the website is well designed to attract the clients. The website is only meant to display information's and subscribed user gets the news letter from the fortunecity.com web service.

2.2.2.1 Advantages

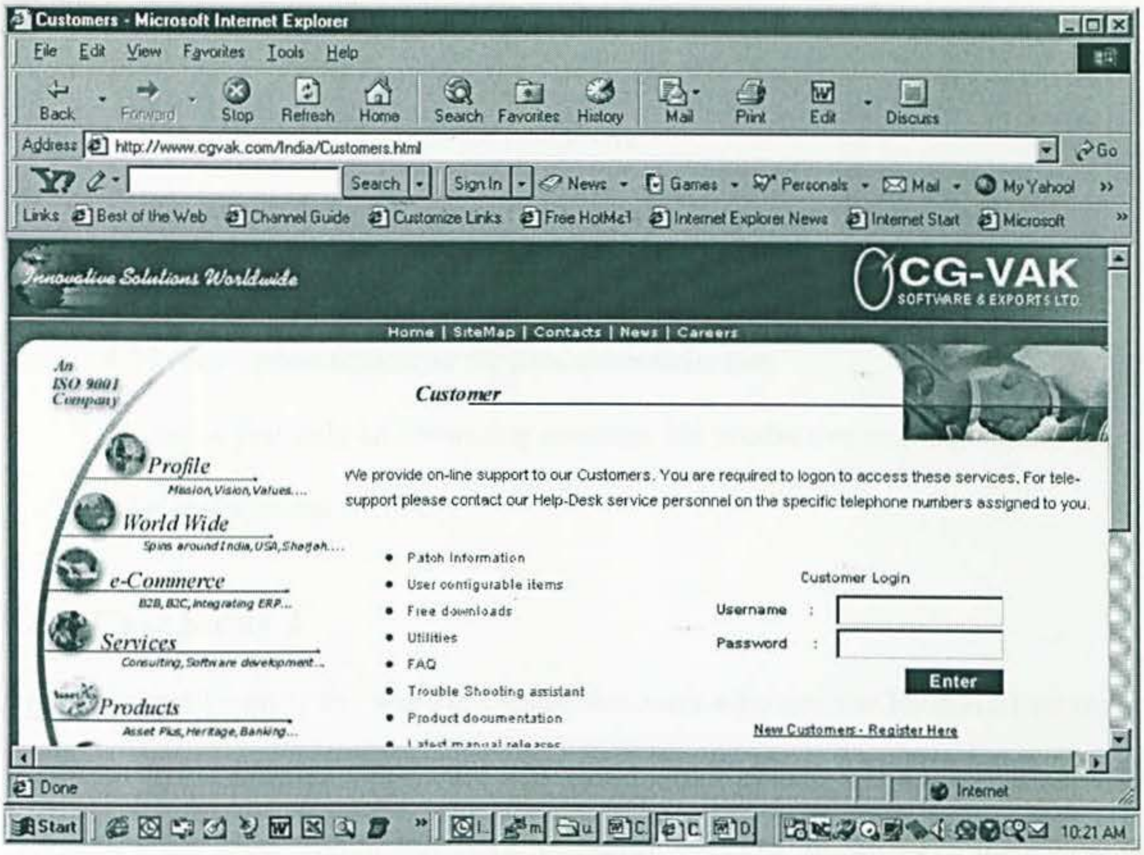
1. Easy Access to member login
2. Download any information regarding the service.
3. Unlimited storage capacity
4. Free email service.

2.2.2.2 Disadvantages

1. Minimal Authentication
2. No secure password
3. Any member can enter into website
4. No administration levels for user access
5. No encryption of data because information is wide open for access.
6. No communication between clients.

2.2.3 Case Study 3

www.cgvak.com is the website for CGVAK software exports company registered as a software solution provider. The company deals with software products and services. The prototype of their view is shown below.


Customers - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://www.cgvak.com/India/Customers.html
Go
Search Sign In News Games Personals Mail My Yahoo!
Links Best of the Web Channel Guide Customize Links Free HotMail Internet Explorer News Internet Start Microsoft
Innovative Solutions Worldwide
CG-VAK
SOFTWARE & EXPORTS LTD
Home | SiteMap | Contacts | News | Careers
An ISO 9001 Company
Profile
Mission,Vision,Values....
World Wide
Spins around India,USA,Sharjah....
e-Commerce
B2B, B2C, integrating ERP...
Services
Consulting, Software development...
Products
Asset Plus, Heritage, Banking...
Done
Internet
Start
10:21 AM

Customer

We provide on-line support to our Customers. You are required to logon to access these services. For tele-support please contact our Help-Desk service personnel on the specific telephone numbers assigned to you.

- Patch Information
- User configurable items
- Free downloads
- Utilities
- FAQ
- Trouble Shooting assistant
- Product documentation
- Latest manual releases

Customer Login

Username :

Password :

Enter

New Customers - Register Here

Figure 2.2:CGVak Website

The above website shows the customer login page, basically this login is created for the user or clients who ever buys the product with them. Each individual client can login and check the status and communicate with the company personnel when ever they require any service.

2.2.3.1 Advantages

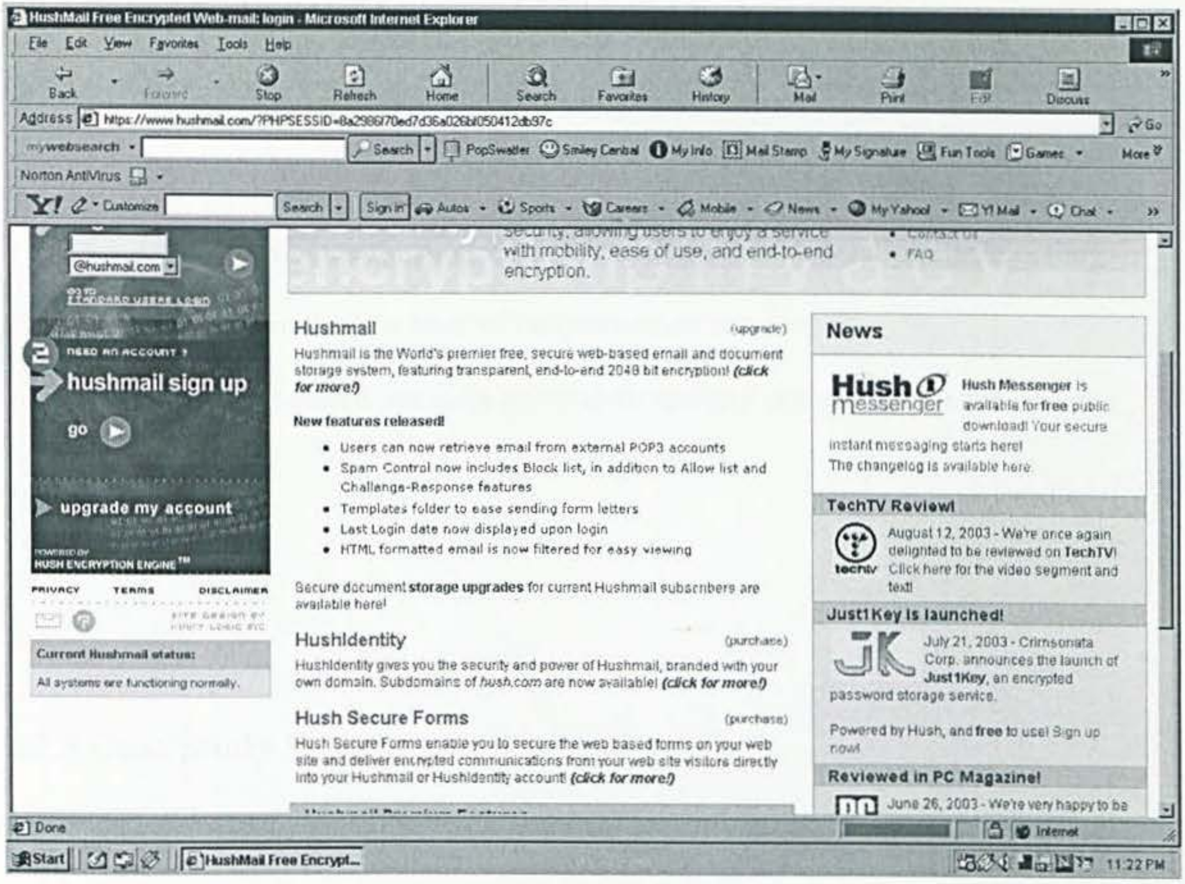
1. Easy access for clients from any part of the world.
2. Wide Customer care consulting.
3. Better service with instant reply through email services.

2.2.3.2 Disadvantages

1. No secure login features for customers.
2. Any person from the company can view the customer information.
3. No authentication services providers are available.
4. No encryption technique for data communication.
5. Used as just only an answering machine. No productive technique involved for maintaining clients.

2.2.4 Case Study 4

www.hushmail.com is the website created for users who use the HushMail services. Hushmail is the premier free secure web-based email system and document storage, featuring transparent, end-to-end 2048 bit encryption. The prototype of their view is shown below.


HushMail Free Encrypted Web-mail: login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address https://www.hushmail.com/?PHPSESSID=8a2986f70ed7d36a026bf050412db97c
Go
mywebsearch Search PopSwatter Smiley Central My Info Mail Stamp My Signature Fun Tools Games More
Norton AntiVirus
Y! Customize Search Sign In Autos Sports Careers Mobile News My Yahoo! Y! Mail Chat
@hushmail.com
go to standard users login
need an account ?
hushmail sign up
go
upgrade my account
powered by HUSH ENCRYPTION ENGINE™
PRIVACY TERMS DISCLAIMER
Current Hushmail status:
All systems are functioning normally.
Contact us
FAQ
Done
Internet
Start
HushMail Free Encrypt...
11:22 PM

security, allowing users to enjoy a service with mobility, ease of use, and end-to-end encryption.

Hushmail (upgrade)

Hushmail is the World's premier free, secure web-based email and document storage system, featuring transparent, end-to-end 2048 bit encryption! *(click for more!)*

New features released!

- Users can now retrieve email from external POP3 accounts
- Spam Control now includes Block list, in addition to Allow list and Challenge-Response features
- Templates folder to ease sending form letters
- Last Login date now displayed upon login
- HTML formatted email is now filtered for easy viewing

Secure document **storage upgrades** for current Hushmail subscribers are available here!

HushIdentity (purchase)

HushIdentity gives you the security and power of Hushmail, branded with your own domain. Subdomains of *hush.com* are now available! *(click for more!)*

Hush Secure Forms (purchase)

Hush Secure Forms enable you to secure the web based forms on your web site and deliver encrypted communications from your web site visitors directly into your Hushmail or HushIdentity account! *(click for more!)*

News

Hush Messenger is available for **free** public download! Your secure instant messaging starts here!
The changelog is available here.

TechTV Review!

August 12, 2003 - We're once again delighted to be reviewed on **TechTV**! Click here for the video segment and text!

Just1Key is launched!

July 21, 2003 - Crimsonata Corp. announces the launch of **Just1Key**, an encrypted password storage service.

Powered by Hush, and **free** to use! Sign up now!

Reviewed in PC Magazine!

June 26, 2003 - We're very happy to be

Figure 2.3:Hushmail Website

The above website is create for the users to have free email based services and documents personnel to any user can be updated and maintained in the website. The above web page shows the login page and the features of the login services.

2.2.4.1 Advantages

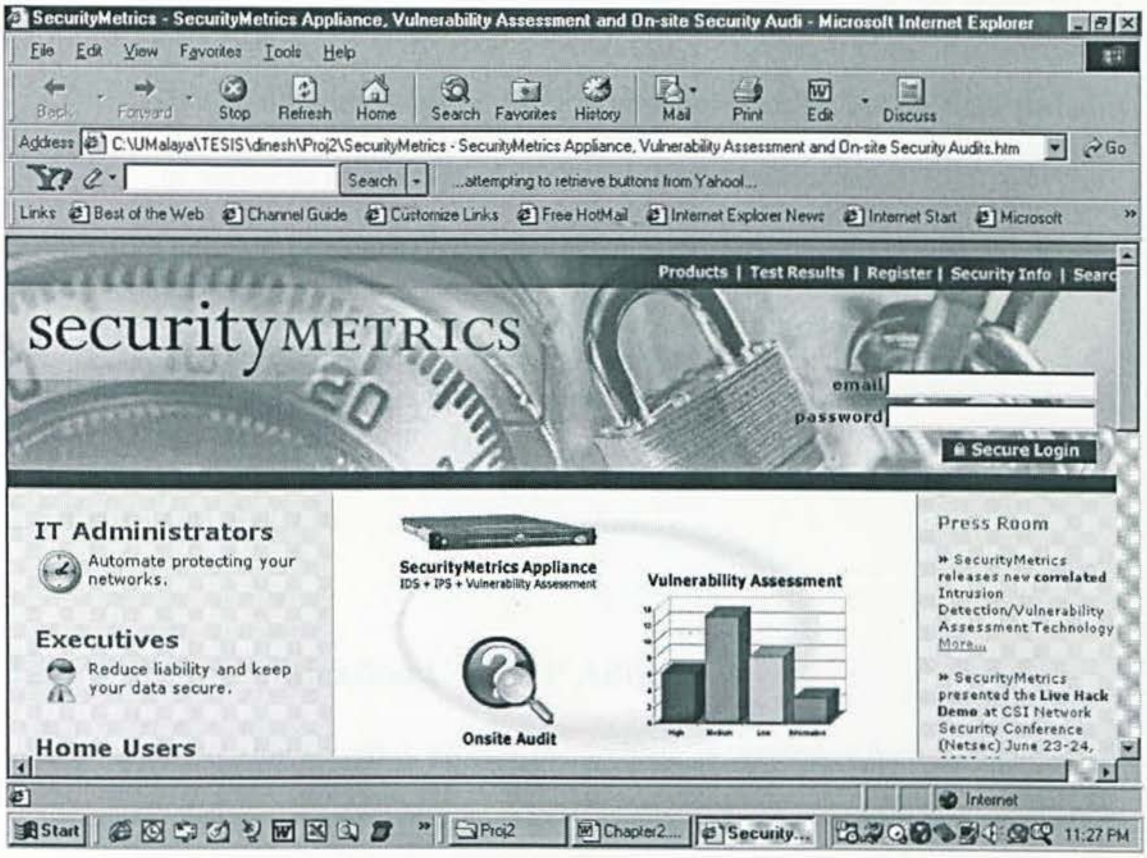
1. Free and easy access for any user sign into the login service.
2. Free encryption feature.
3. Secure and reliable when the user accepts the terms and polices or norms of operating the website email.
4. POP3 and SMTP features are strongly implemented.

2.2.4.2 Disadvantages

1. No high level encryption, code can be easily decoded.
2. Security features are not implemented through software media.
3. Though the authentication services are taken from Operating System, they still perform the low level of authentication services.
4. Documents stored are more transparent and any one can access the updated documents.

2.2.5 Case Study 5

www.securitymetrics.com is the website for the security metrics corporation, the company deals with all types of security measures that can be handled through any software and hardware related networks. The prototype of the website is shown below.



SecurityMetrics - SecurityMetrics Appliance, Vulnerability Assessment and On-site Security Audi - Microsoft Internet Explorer

Address C:\UMalaya\TESIS\dinesh\Proj2\SecurityMetrics - SecurityMetrics Appliance, Vulnerability Assessment and On-site Security Audits.htm

...attempting to retrieve buttons from Yahoo!...

IT Administrators

Automate protecting your networks.

Executives

Reduce liability and keep your data secure.

Home Users

Press Room

» SecurityMetrics releases new **correlated** Intrusion Detection/Vulnerability Assessment Technology More...

» SecurityMetrics presented the **Live Hack Demo** at CSI Network Security Conference (Netsec) June 23-24,

File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Go
Search
Links Best of the Web Channel Guide Customize Links Free HotMail Internet Explorer News Internet Start Microsoft
Products | Test Results | Register | Security Info | Searc
securityMETRICS
email
password
Secure Login
SecurityMetrics Appliance
IDS + IPS + Vulnerability Assessment
Onsite Audit
Vulnerability Assessment
High
Medium
Low
Information
Internet
Start
Proj2
Chapter2....
Security...
11:27 PM

Figure 2.4 Security Metrics Website

The above website shows the login page which basically used for the members and customer access into the website. The login above shown has both authentication and encryption features like the proposed system. The only basic difference is the security features.

2.2.5.1 Advantages

1. Hard to break the password features.
2. Strong authentication techniques implemented.
3. Better Access features and easy to use.
4. Background technical features are not transparent to any user except administrator.

2.2.5.2 Disadvantages

1. Though the authentication features are implemented, but the major security for networks solely depend upon their hardware system
2. Though they have encryption and other features IP tracking facility is not inborn, so a hard hacker can easily intrude their system into the areas where they have less hardware penetration.

2.3 Proposed System

2.3.1 ASP Authentication Using IP Address.

Using the server side scripting creates the website is created, which implements the features of password authentication and data encryption techniques. The major features are listed below.

1. Website login is protected by password authentication technique programmed using server side scripting.
2. Administrator has been implemented to improve the user access levels. The user access levels are also associated with the Internet Information Server.
3. Data encryption techniques are implemented at communication levels.
4. User is checked before the information is downloaded, information before down loading normally stored in network using data encryption technique.
5. Fully secure integrated system is developed using the benefits available in the Internet Information Server.

6. When ever user login into the system the user IP address is identified and tracked so that the administrator obtains complete information of user.
7. Administrator is able track the usage of the end user like login time and logout time, monthly usage, daily usage, domain access etc.

2.4 Authentication System

2.4.1 History of Authentication

Early password systems, meant for mainframe users, stored plain-text passwords in files, confident that no mere mortal could uncover them. As computer systems became more accessible, storing password in plain became unfeasible. Instead, people began storing password-equivalents, which are created by hashing a plain-text password and storing the hashed version. In this model, password you enter at login is hashed, and the hash is compared to the stored password-equivalent. If the computed hash matches the stored hash, the system assumes you know the password and permits you to log in.

Using hashes and storing password-equivalents worked well enough until networking became popular. In the early days of networking programmers simply extended the techniques used for local authentication to remote authentication. Protocols such as Telnet and FTP simply send the user's name and password across the network to the remote server, which then hashing algorithm to try to match the result to stored password-equivalent.

The problem with this approach is that Telnet, FTP and others protocols such as POP and HTTP send passwords in the clear, making them easy to capture. By 1994, hackers had written sniffers specifically designed to capture passwords sent by these protocols, along with software that could hide the sniffers themselves.

Hackers installed these sniffers primarily on ISP networks, which at the time were poorly secured but carried a high volume of traffic. New methods of network authentication were thus developed. The most common is challenge-response. Still used by Microsoft file sharing, this approach does not send the password itself across the network. Instead, the server transmits a challenge, which the client operates upon by using some permutation of user's password. The client then transmits a response to the server.

2.4.2 Strength Of Authentication Security

- Reasonable security, which meets the requirements of both user and the developer of data, is important. This is somewhat subjective and depends on what is being protected, how easily is it hacked and what are the chance or consequences of the breach either on a single or systematic basic.
- Possible data gathered by vendor can be protected.
- IP filtering, an IP address (or range of addresses) is used to filter access to a database or service so that only users with a PC within a proper network domain may gain access.

2.4.3 Features and Benefits of Authentication

- Widely used
- Well understood
- Works for local and remote users
- Can be used from anywhere

2.4.4 Review On Authentication.

Authentication is any process by which a system verifies the identity of a user who wishes to access it. Since access controls are normally based on the identity of the user who requests access to a resource, Authentication is essential to effective security.

Authentication may be implemented using credentials, each of which is composed of a user id and password. Alternately, Authentication may be implemented with Smart Cards, an Authentication Server or even a Public Key Infrastructure

Users are frequently assigned (with or without their knowledge) Ticket, which are used to track their Authentication state. This helps various systems manage Access Control without frequently asking for new Authentication information.

Authentication means the act of providing that you say you are. The way of doing this for most computer systems is to log in using the username and password that you have been allocated by the people who manage the computer that you are using.
Since the username and password combination uniquely identifies you, it is essential you keep your password secure and confidential to you. If you suspect that someone has found out your password, you should change it as soon as possible.

Let us assume you want to restrict access to your website. For example, you might have valuable information, such as real-time stock quotes (like Reuters or DataStream), or you want to charge a monthly fee in order to access your database of superbowl picks or you may be such a nice person that you don't want to charge your visitors a dime, just want to track your visitors.

In these cases, you want to let people in, but only after checking that the visitors have an authorized username and password. Additionally, you might want to limit certain visitors to only certain areas of your website.

So, have you figured it out yet? Verifying that a visitor is authorized to visit a particular part of your website (usually via a username and password) is called Authentication. Authentication allows you to control the access to your entire website.

2.4.5 Why Have We Introduced Authentication.

- To ensure that only members of the organization are able to use our facilities.

  There have been a number of occasions when people who are not members of the organization have been found using the facilities.

- To enable us to identify people misusing the facilities.

  Occasionally members of the organization are found to be misusing the computers network. Requiring the users to authenticate using a username and password will allow us to identify and take action against this type of anti-social behavior. It is important, therefore to log off when finished using the system, otherwise you may find that you are held responsible for someone else's misuse.

- To ease administration

  Users are required to authenticate to access other network resources such as file servers, electronics mail and Unix machines. The ultimate aim is to simplify the process of computer usage by requiring users to log on only once.

- To allow us to provide additional network services

  There are several network services that cannot provide without authentication, such as central file store and central printing. This service allows us to provide additional functionality for users of authenticating PC's.

2.5 Encryption.

Encryption is one of the oldest science that human has undertaken. It essentially concerns the basic human need to protect certain sensitive information from prying eyes; one trait that may be considered an absolute necessity to survive. This is done through encryption, rendering the information unusable in the desired context by unauthorized reader.

Data transmitted across the Internet is open to viewing by anyone unless it is encrypted. Passwords and secret documents alike are there for all eyes to see. Encryption provides privacy, reliability and authenticity. It is the only way to be sure that no one is tampering with your messages.

Computer security is about keeping data secure. One type of data is kept in files on the computer; information about the company such as budgets, newly developed product information, employee information (phone numbers and salaries) etc. Another type of data is transaction data. Transaction data is transmitted between the local computer and some external computer via a network. This type of data is sensitive as it usually involves passwords and other secret information. When we talk about encryption and the Internet we are usually talking about securing Transaction data.

A networked computer that is unsecured is easy prey to even the moderately competent attacker. There are people in this world that still depend on the common decency, good will, and the force of law. These are the people who do not use encryption. Encryption is any procedure that utilizes cryptography to convert plain text into cipher text in order to prevent any but the intended recipient from reading that data. Encryption was chosen as this web site's topic due to the lack of information and materials about encryption and cryptography readily and easily available to the public. Intended as both a guide and a reference about encryption, this is a public resource about private data.

The Science of Encryption is basically the application of the math of cryptography in order to get data from one person to another person without anyone in between being able to find out what it is. Simply, to keep information private, this is an incredibly difficult task. Ever since there was a need to send information that could not be guaranteed delivery to the right person, there was a need for encryption. The beginnings were simple, from code word messages to decoder rings, and at present, complex mathematical algorithms flipping through thousands of calculations.

A message before being changed in any way is called plaintext. Plaintext messages are converted to ciphertext via some encryption method. A particular such method is called a **cryptosystem**. The study of breaking encryption is termed **cryptanalysis**.

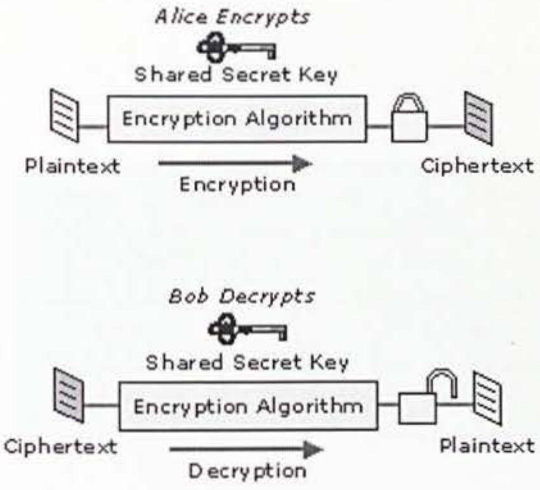


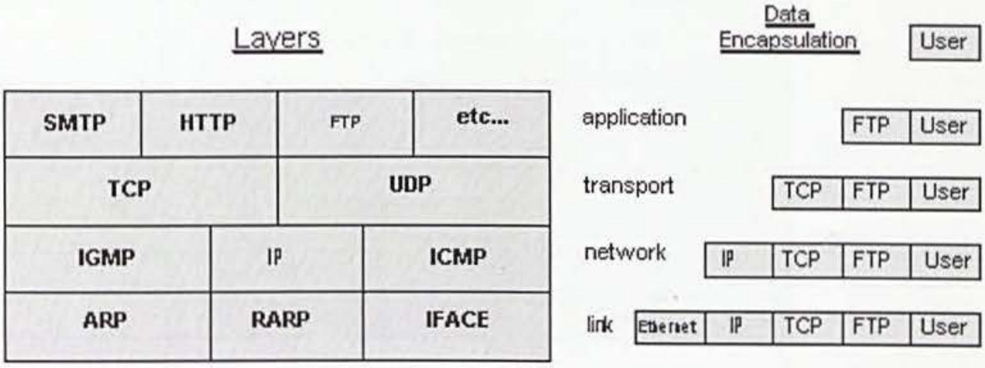
Figure 2.5Encryption and Decryption

2.6 TCP/IP and IP Security

The TCP/IP suite of protocols was developed as part of the research done by the Defense Advance Research Project Agency (DARPA). Later, TCP/IP was included with the Berkeley Software Distribution of UNIX. The Internet protocol can be used to communicate across any set of interconnected networks. They are equally well suited for both Local Area Network (LAN) and Wide Area Network (WAN) communication. The Internet protocol suite includes not only Layer 3 and Layer 4 specifications, but also specifications for such common applications as e-mail, remote login, terminal emulation and file transfer.

2.6.1 TCP/IP Overview

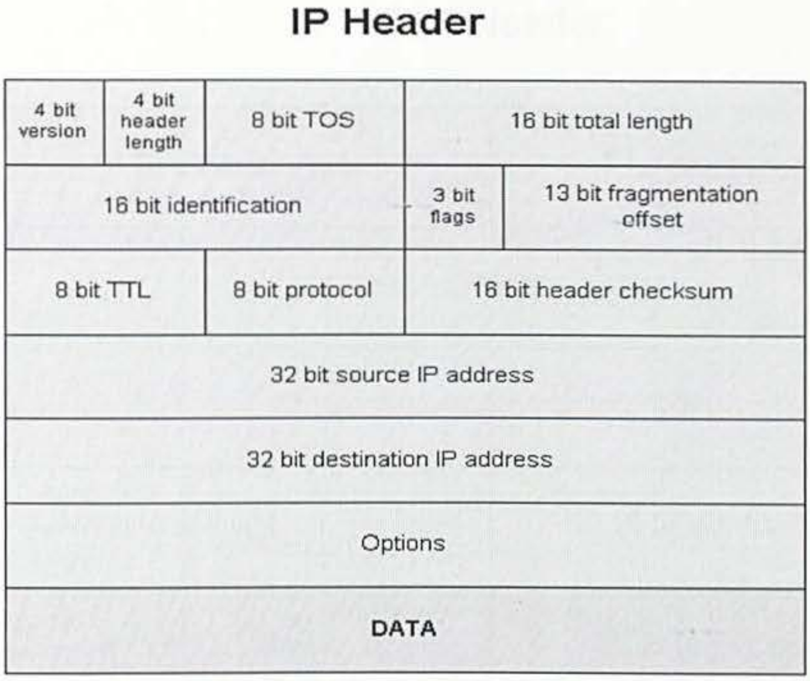
The TCP/IP protocol consists of several protocols operating at different layers. TCP/IP network model can be represented by the 4-layer model shown below (which is a simplified version of the OSI 7 layer model).



TCP/IP 4 Layer Model

Figure 2.6: TCP/IP 4 layer Model

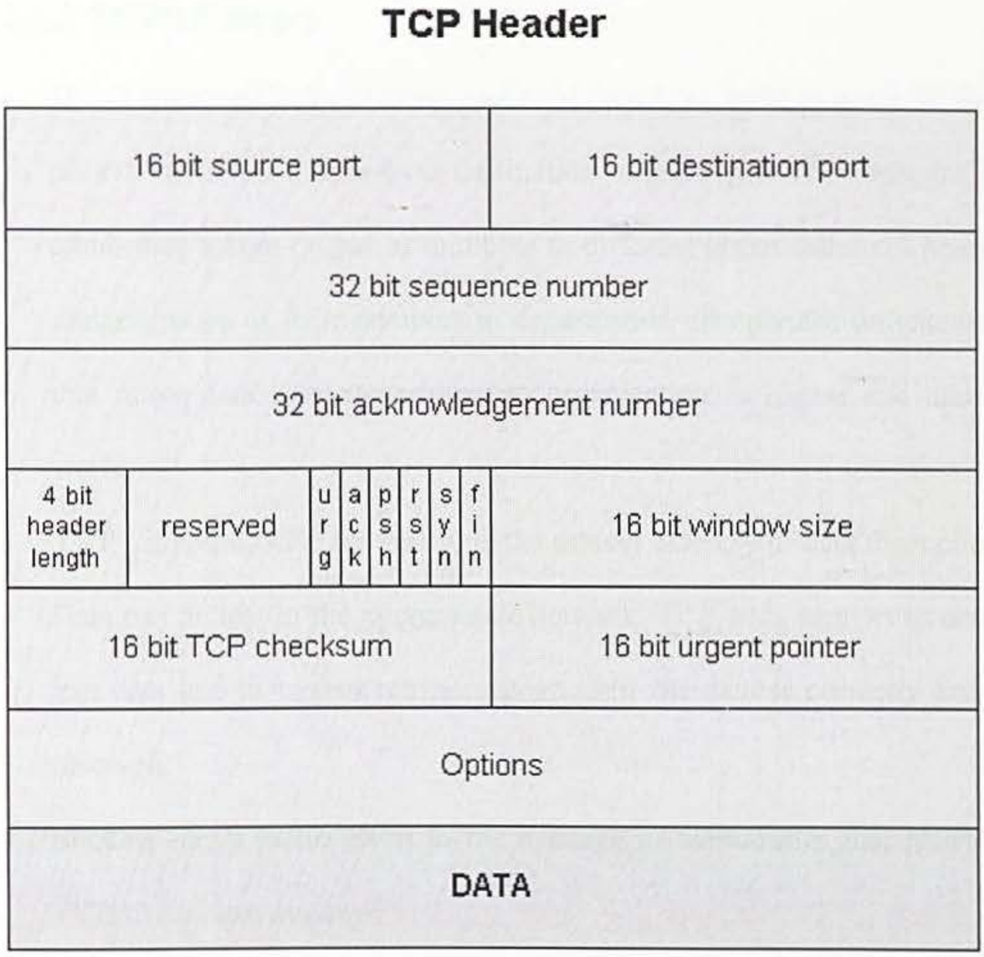
As a user sends data, it passes through each layer that encapsulates it, each adding a protocol header to the packet. When the packet arrives at the destination, it passes up through the stack, each layer stripping off the encapsulation previously applied.



IP Header

<table>
<tr><td>4 bit version</td><td>4 bit header length</td><td>8 bit TOS</td><td colspan="2">16 bit total length</td></tr>
<tr><td colspan="3">16 bit identification</td><td>3 bit flags</td><td>13 bit fragmentation offset</td></tr>
<tr><td colspan="2">8 bit TTL</td><td>8 bit protocol</td><td colspan="2">16 bit header checksum</td></tr>
<tr><td colspan="5">32 bit source IP address</td></tr>
<tr><td colspan="5">32 bit destination IP address</td></tr>
<tr><td colspan="5">Options</td></tr>
<tr><td colspan="5">DATA</td></tr>
</table>

Figure 2.7: IP Header



TCP Header

<table>
<tr><td colspan="8">16 bit source port</td><td>16 bit destination port</td></tr>
<tr><td colspan="9">32 bit sequence number</td></tr>
<tr><td colspan="9">32 bit acknowledgement number</td></tr>
<tr><td>4 bit header length</td><td>reserved</td><td>urg</td><td>ack</td><td>psh</td><td>rst</td><td>syn</td><td>fin</td><td>16 bit window size</td></tr>
<tr><td colspan="8">16 bit TCP checksum</td><td>16 bit urgent pointer</td></tr>
<tr><td colspan="9">Options</td></tr>
<tr><td colspan="9">DATA</td></tr>
</table>

Figure 2.8: TCP Header

2.6.2 TCP/IP layers

- **IP** - is responsible for moving packet of data from node to node. IP forwards each packet based on a four-byte destination address (the IP number). The Internet authorities assign ranges of numbers to different organizations. The organizations assign groups of their numbers to departments. IP operates on gateway machines that move data from department to organization to region and then around the world.
- **TCP** - is responsible for verifying the correct delivery of data from client to server. Data can be lost in the intermediate network. TCP adds support to detect errors or lost data and to trigger retransmission until the data is correctly and completely received.
- **Sockets** - is a name given to the package of subroutines that provide access to TCP/IP on most systems.

2.6.3 Review On IP Security

IP is the underlying technology of networks used by the government academia and the corporate world, as well as the public Internet to send myriad types of packet all round the world. Yet the strength of IP as a WAN protocol security has never been strong suit.

The layer 3 protocol overseas packet forwarding through different types of networks. But, IP's wide acceptance as the basis for both public and private networks has led to great concern over the issue of security.

Several types of attacks have been known to take place over IP networks. One is called IP spoofing, in which an intruder tries to gain access by changing a packet's IP address to make it appear that the packet came from somewhere else. Other attacks include eaves dropping on an IP transmission. This can be done by using a protocol analyzer to record network traffic. Another types of IP attack involve taking over a session as one of the parties involved in the communication.

The IP security problem will probably only get worse as companies rely on the protocol more and more for remote communications. Also, virtual private network (VPN), which allow companies to create a private connection over the public Internet, require strong safeguards.

Although VPN to exist to a certain extent today, most industry watches will tell you that for this concept to really fly, authentication, encryption and other security measures need to be in place. The hope is that an IETF standard called IP security (IPsec), an extension to IP will be catalyst for private and secure communication over the Internet.

IP security is actually a suite of protocol being developed by IETF. The suite includes the Authentication Header (AH), which addresses authentication for IP traffic and the encapsulating security payload (ESP), which defines encryption for IP data.

The authentication header ensures that the packet has not been altered or tampered with during transmission. It can simply be used to verify the authentication of regular IP packet.

2.7 Web Programming Languages.

2.7.1Visual Basic.

Visual Basic is a programming language that evolved from a long line of BASIC programming languages but differs in certain features such as graphic user interface (GUI), object-oriented, event handling, timing and others.

Visual Basic is based on three fundamental aspects; controls (like list boxes), properties (size and colour) and methods (actions that a control can do).

- A stand-alone language designed to easily be written to work under Microsoft Windows.
- It is rich, powerful and uses built-in objects.
- It is a compiled language and is distributive using runtime support for applications. (Under the Integrated Development Environment, IDE)
- Run only on windows platform and this makes the application more limited.

2.7.2 Java

An excellent tool for working in this environment is the Microsoft Visual Studio development system. With Visual Studio, I can both program in Java and create a client and server-side Web site. Java is used to create the business objects and the complex interfaces.

Java is not only a programming language, but also an environment in itself. Every Java-based program starts as a source code file with the extension. Java. The source code file, say Sample. Java is then compiled into a byte code file called Sample. Class.

The Java Virtual Machine (VM) is like a processor, except that it can only decipher Java byte codes. Byte code is the native language of the Java VM. Java byte code and Intel object code is pretty much the same thing, except for the niggling fact that they are not compatible with each other. The byte code must be run by a Java VM, which converts the byte code instructions into something that the hosting processor can understand. This is how the same Java code can be executed on various kinds of processors without alteration.

Because a Java VM can be built for any processor, there are certain ironclad rules. One is the size of data: an integer must be 32 bits, a long must be 64 bits, and so on. Another rule is that the behavior of the VM itself must be the same across processors. A special feature of one processor must not conflict with the behavior of the Java program.

Because Java byte code is a form of assembly code, any language can theoretically be compiled to produce Java byte code. There are caveats, though. The Java VM is a safe environment, so it does not support the concept of pointers. While this makes code safer and more crash resistant, it makes it difficult to fully support C++. There are other safety features in Java, such as code checking and verification, but these aspects are generally compile-time implementations that do not influence the design of a language. They do, however, make it possible to execute programs that do not overwrite the system, a feature known as sand boxing. In a typical Java project setup, the top-level directory is named Java. Below that are three other directories called classes, lib, and trustlib.

2.7.3 HTML (Hyper Text Markup Language)

When one mentions web pages, HTML is the first word that comes to mind. It is synonymous with web features and documents. HTML gives developers means to:

- Publish on-line documents with headings, text, tables, lists, photos, etc.
- Retrieve on-line information via hypertext links at a click of a button.
- Designs form for conducting transaction with remote services, for use in searching data, making reservations and ordering products.
- Include spreadsheets, video clips and other applications directly into their documents.

HTML works well on different platforms and browsers; achieving interoperability lowers costs to content providers since they develop only one version of a document. HTML has been developed with the vision that all manner of devices should be able to use information on the Web; PC's with graphical displays of varying resolution and colour depths, cellular phones, hand held devices (PDA), devices for speech for output and input, computers on high and low bandwidth and so on.

2.7.4 Dynamic HTML (DHTML).

Microsoft and Netscape introduced DHTML in the 4.0 versions of their web browsers. DHTML is a suite of technologies that gives a web designer the ability to add any functions to the web pages as quickly and easy as HTML. DHTML does not rely on plug-ins the visitor might or might not have or does not have complicated programming languages, except maybe a little JavaScript.

For most part, DHTML is created the same way as HTML and requires no special software to produce. Like HTML, DHTML should work with all browsers and on all platforms. DHTML also should be able to enhance the interactivity and visual appeal of the web page. The reason as to why a web page using DHTML is so dynamic is because:

1. It allows the developer to control the HTML displays the web pages content.
2. It also allows the document to react and change with the actions of their web site visitor.
3. It has the ability to exactly position any element in the window and change that position after the document has been loaded.
4. It can hide and show content as needed.

2.8 Scripting Languages.

2.8.1 Visual Basic Script (VBScript)

Based on analysis from previous section, VBScript is a powerful Web development scripting language that can be used on On-line Cancer Info. It used directly within a web page using the <SCRIPT> tag-no<APPLET> or plug –ins required. VBScript is based on Visual Basic, which is different from other language like Java and Java Script where they based on C and C++. VBScript is similar to Visual Basic stand-alone application programming environment, it shares common elements and construct, making any successful Visual Basic programmer as a successful Web scripter.

Most of the pages that use VBScript use its various controls to enable the user to enter information or do number of things on the web. One of the biggest attributes of any scripting language is the capability to execute commands without requesting information from the server. In a plain HTML files, user can insert buttons and controls for the user to navigates through user site. However, when the user clicks on links or other items the client machine must request information from the server. With this language, some actions can be executed within the client computer rather than requesting information from the server. This significantly reduces network traffic as well as speeds up things for the user.

2.8.2 JavaScript

JavaScript is not Java, per say, but more like a relation to it. Developed by Netscape Corp., it is the rival of Microsoft's VBScript. Among its features are:

- Extends the capability of the standard web page.
- More flexible and dynamic than HTML; enables programmers to implement animations and form generation in HTML pages
- An interpretive code that is written directly into the web page and the codes are visible. Its interpreter is integrated within browsers.

JavaScript is relatively easier to learn compared to Java; no code compilation and easier commands.

2.8.3 ActiveX

ActiveX is not exactly a scripting language but rather a technology used by developers to write software components that interoperate, regardless of the language to create them. For example, ActiveX applications can be written using C++, Java, Visual Basic and Delphi. The real power of ActiveX comes from its consistent, comprehensive implementation. This means that with the same component based approach, developer can:

- Script objects inside a HTML page.
- Assemble interfaces for Windows Applications.
- Communicate between client and server components.
- Script business rules or web server applications.
- Coordinate transactions across multiple servers.

2.8.4 Active Server Pages (ASP).

ASP provides 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 a developer create truly dynamic Web sites and online applications accessible by any browser.

ActiveX Data Objects (ADO), an ASP component, lets developer's access and control data in any ODBC- or OLE DB-compliant database using any ActiveX scripting language. Developers can put a Web front end on almost any legacy database without arcane CGI programming.

To tap into the power of ASP and server-side scripting, IIS includes native scripting engines for VBScript and JScript. Server plug-ins are available for other scripting languages such as Perl.

An ASP page is a document that resides on the web server and that contains a mixture of HTML code and server-side scripts. Such scripts process requests coming from client browsers and can build a response page for that particular client. This is an advantage using ASP as it lets the developer create dynamic HTML pages that be downloaded by any browser that support plain HTML. For this reason, ASP can play an important role in Internet applications, whereas DHTML should be used only in more controlled environments, such company Intranet. ASP technology does not deliver pages with animation and transitions effect, rather with it a developer can create pages on the fly that are customized for each client.

2.9 Operating Systems.

2.9.1 UNIX

UNIX is a computer operating system. An operating system is the program that controls all the other parts of a computer system, both the hardware and the software. It allocates the computer's resources and schedules tasks. It allows user to make use of the facilities provided by the system. Every computer requires an operating system. UNIX is a multi-user and multi-tasking operating system. Multiple users may have multiple tasks running simultaneously. This is very different than PC operating systems.

UNIX is a machine independent operating system. It is not specific to just one type of computer hardware. It is designed be independent of the computer hardware. UNIX is a software development environment whereas born in and designed to function within this type of environment.

Following are the benefit of UNIX:

1) Hardware independence:

~ Operating system code is written in C language rather than a specific assembly language

~ Operating system software can be easily moved from one hardware system to another

~ UNIX applications can be easily moved to other UNIX machines. Porting is usually as simple as transfer of the source and a recompile

2) Productive environment for software development:

~ Rich set of tools versatile command language

3) UNIX is available at virtually all HPC centers, allowing researchers relative ease in utilizing the facilities at each center. Distributed processing and multi-tasking

2.9.2 Windows NT

Microsoft Windows NT Server 4.0 is now a better choice than ever. With the new features introduced with the Windows NT 4.0 Option Pack, Windows NT Server is the most complete platform available for building and hosting Web-based applications, and the easiest server operating system available. User will be up and running less than an hour after user takes it out of the box. Its so flexible and compatible user will realize significantly reduced hardware and software costs. User will experience far less downtime thanks to its reliability and easy management.

Following are the Windows NT benefits

- Windows NT Server 4.0 was designed to help developers build and deploy business applications faster than ever before. The Option Pack integrates new Web, transaction, scripting, component, and message queuing services directly into Windows NT Server 4.0.
- New management tools in Windows NT Server 4.0 and the Option Pack help you set up Web sites, manage content, and analyze usage patterns to improve your site as it evolves.
- Multiple Web sites on a single machine, innovative Web publishing features, customizable tools, and new wizard technologies make Windows NT Server 4.0 the best platform to publish and share information securely over corporate intranets and the Internet.

2.9.3 Windows 98

The Microsoft Windows 98 or operating system is the upgrade to Windows that makes user computer work better and plays better. It works better by making it simple to access the Internet and by providing better system performance along with easier system diagnostics and maintenance. With Windows 98, user system plays better as well with support for the latest graphics, sound, and multimedia technologies, the ability to easily add and remove peripheral devices with support for Universal Serial Bus (USB), and it also enables you to watch TV on your PC. Windows 98 Second

Edition is an update to Windows 98 that enhances the leading consumer operating system with the latest Internet, home networking and hardware technologies.

Windows 98 Second Edition offers consumers a variety of new and enhanced capabilities related to the Internet, and hardware compatibility, Internet Explorer 5. Microsoft's popular browsing technologies provide breakthroughs in Web performance, usability and flexibility. Windows NetMeeting 3. The latest version of NetMeeting brings Internet conferencing capability to consumers by offering enhanced usability, performance, Security and support for Internet standards.

Internet Connection Sharing (ICS). ICS is a set of advanced home networking technologies that enable users to share a single connection to the Internet for simultaneous Internet access. Enhanced hardware support. Windows 98 Second Edition offers improved native support for technologies such as Universal Serial Bus (USB), IEEE 1394 and ACPI as well as broadband network connections, enabling consumers to connect to and more easily use a wider range of devices. Windows 98 Service Pack addresses top issues in existing features, such as year 2000 updates. Windows 98 Second Edition continues to maintain the best support for older Windows-based applications and technologies.

Following are the Windows 98/2000 benefits:

- Improved Ease of Use & Internet Access--Dynamic Web-based help and 15 Wizards help make your PC easier to use. Windows 98's Web-aware user interface lets you find information more easily with the same view of content on your PC, network, or the Web. Windows 98 Second Edition now provides Internet Connection Sharing, allowing users to share a single Internet connection over multiple networked PC's.

- Improved Performance & Reliability--Reduce the time it takes to launch applications, get help cleaning your hard disk and improve its efficiency. This is all possible with improvements that make Windows 98 a more robust and reliable operating system.
- Enables a New Generation of Hardware & Entertainment--Take advantage of the latest hardware advances, like the USB, DVD, and IEEE 1394, and expand the use of your PC with multiple monitor and digital imaging support and Microsoft WebTV for Windows.

2.9.4 Linux

Linux is a free, from scratch operating system based heavily on the POSIX and UNIX API's. It supports both 32 and 64 bit hardware and provides a stable multi-user Internet ready operating system.

Linux itself is not Unix, although many people call it that and user would be very hard pushed to tell the difference. This is because the Unix trademark is specific to systems that meet a complex set of X/Open standards and has a cost. Some Linux vendors however are working on "Unix" branding.

Some of the many applications for Linux are X11 Desktop, File server, Computing Backend, Web Server, Usenet News, Terminal Server, FTP Archive, and Firewall. Linux uses Internet and industry standard components and protocols giving a system with complete network integration. The operating system can act as a server for most major file serving protocols, and provide all the major Internet applications. The X window system provides a networked and platform independent graphical interface that (unlike proprietary user interfaces) allows one desktop to access applications running on multiple machines across local and wide area networks.

Linux is normally obtained as a *"distribution"*. This is a combination of the Linux operating system kernel and other tools, utilities and applications. Some of these are available for free over the Internet, and others on CD-ROM. Because Linux itself is free software that can be freely copied, many distributions are available both over the Internet and sold on CD-ROM with added convenience and support.

2.10 Web Server.

A web server is a program, located on a computer with Internet access, that responds to browser's request for a URL. That is a Web Server meets the demands of users by supplying or serving them the web pages requested. Ideally, the server should have an uninterrupted Internet connection, so that the pages it handles are always available. A web server is accessible to many of us through work or school and many ISPs include space on a web server as part of the basic set of services covered in their monthly fee. The system administrator responsible for the server can usually fill us in about the site-specific details for publishing our web pages.

2.10.1 Apache

Apache is a general web server, which is designed to be correct first, and fast second. Even so, its performance is quite satisfactory. Most sites have less than 10Mbits of outgoing bandwidth, which Apache can fill using only a low end Pentium-based web server. In practice sites with more bandwidth require more than one machine to fill the bandwidth due to other constraints (such as CGI or database transaction overhead). For these reasons the development focus has been mostly on correctness and ability to configure.

There is a bare minimum performance that is acceptable, beyond that extra speed only caters to a much smaller segment of the market. But in order to avoid this hurdle to the acceptance of Apache in some markets, effort was put into Apache 1.3 to bring performance up to a point where the difference with other high-end web servers is minimal.

2.10.2 Netscape Enterprise Server

Netscape Enterprise makes Web development easy. Its excellent content management features can put managing and publishing power in the hands of your users, an unrivaled and forward-thinking feature. Netscape Enterprise combines innovative features and performance meets most users' needs. The most significant features with Netscape's are in workgroup-based content management, Lightweight Directory Access Protocol (LDAP) integration, hardware-based SSL support, and Java servlets. Netscape Enterprise runs on many platforms, including NetWare, five flavors of Unix, and Windows NT. We tested under Windows NT but also did performance testing under Solaris. (We review Novell Netscape Enterprise Server Pro for NetWare separately.)

Netscape Enterprise provides an excellent foundation, offering more programming flexibility than any other server. For application creation, Netscape Enterprise provides several routes for the developer in addition to standard CGI. For higher performance than CGI offers, the Netscape Server API (NSAPI) lets developers write in-process server applications. For Java developers, Netscape Enterprise supports the Sun Java Servlet API for server-based applications.

Netscape Enterprise also comes with the Livewire runtime environment, which lets applications written using server-side JavaScript tools (such as Netscape's Visual JavaScript) run on the server. Livewire is a scripting environment--similar to Microsoft's Active Server Pages--that lets developers create simple Web applications using JavaScript, which is easier to learn than Java or C++. Livewire combines JavaScript scripting with a number of prebuilt server objects that automate tasks such as session maintenance, parsing CGI environment variables, and application initialization. Livewire also provides native connectivity to many databases, including Informix, Oracle, and Sybase, along with ODBC access to other sources.

Most who have large user lists will bypass Server Manager and use the much more useful LDAP capabilities in Netscape Directory Server. Using this bundled product, administrators create LDAP Data Interchange Format (LDIF) files to add large numbers of users and groups. For the Webmaster, Netscape Enterprise's content management features can put a lot of power in users' hands, making the product a good choice for intranet environments.

2.10.3 Personal Web Server (PWS)

Microsoft Personal Web Server (PWS) for Windows 95 turns any Windows 95 computer into a Web server and enables easy publication of personal Web pages. Easy to install and administer, PWS simplifies sharing information on their corporate Intranets or Internet for all users. PWS is ideal for developing, testing and staging Web applications, as well as peer-to-peer publishing with its support for sharing files over HTTP and FTP protocols. Just like Microsoft Internet Information Server (IIS), PWS supports all ISAPI extensions and CGI scripts. PWS has been optimized for interactive workstation use, and does not have the system requirements of a full Web server such as IIS.

It is designed for small-scale peer-to-peer or small Web server usage. As our Web Server needs continue to grow, Microsoft offers a full range of Internet/Intranet Web server products that run on Windows NT Workstation to the powerful enterprise-based solution, Windows NT Server.

The software is fully integrated into the Windows 95 Task Bar and Control Panel, making it easy for users to start and stop HTTP and FTP services whenever they wish. PWS provides all the features you'll need to create and host sophisticated Web pages and applications. However, Microsoft optimized PWS for performance on workstation class machines instead of on servers (requiring fewer system resources) and intended its use as a production, or low-traffic volume, Web server. PWS requires only Windows 95 and less than 1 MB of disk space.

2.11 Database.

2.11.1 Universal Data Access Overview

Universal Data Access is Microsoft's strategy for providing access to information across the enterprise. Today, companies building database solutions face a number of challenges as they seek to gain maximum business advantage from the data and information distributed throughout their corporations. Universal Data Access provides high-performance access to a variety of information sources, including relational and non-relational, and an easy to use programming interface that is tool and language independent. These technologies enable corporations to integrate diverse data sources, create easy-to-maintain solutions, and use their choice of best of breed tools, applications, and platform services.

Microsoft Data Access Components (MDAC) provides easy-to-use, high-performance access to all types of data throughout the enterprise. Developers creating client/server and Internet/intranet-based data driven solutions use these components to easily integrate information from a variety of sources, both relational and non-relational. Microsoft Data Access Components consists of new versions of ActiveX Data Objects

(ADO), OLE DB, and Open Database connectivity (ODBC) which is released, documented, and supported together.

2.11.2 OLE DB

OLE DB, a set of interfaces for data access, is Microsoft's component database architecture that provides universal data integration over an enterprise's network— from mainframe to desktop— regardless of the data type. Microsoft's Open Database Connectivity (ODBC) industry-standard data access interface continues to provide a unified way to access relational data as part of the OLE DB specification. Over time, OLE DB is expected to lead new database products that are assembled from best-in-class components rather than from the monolithic products available today.

OLE DB provides a flexible and efficient database architecture that offers applications, compilers, and other database components efficient access to Microsoft and third-party data stores. OLE DB is the fundamental Component Object Model (COM) building block for storing and retrieving records and unifies Microsoft's strategy for database connectivity. It will be used throughout Microsoft's line of applications and data stores.

OLE DB defines interfaces for accessing and manipulating all types of data. These interfaces will be used not just by data-consuming applications but also by database providers. By splitting databases apart, the resulting components can be used in an efficient manner. For example, components called *service providers* can be invoked to expose more sophisticated data manipulation and navigation interfaces on behalf of simple *data providers*.

2.11.3 ADO

The ActiveX Data Objects (ADO) programming model represents the best of the existing Microsoft data access programming models. If you are familiar with Data Access Objects (DAO) or Remote Data Objects (RDO), you will recognize the interfaces and will be able to work with them very quickly. You will also notice considerable improvements in the model, and tasks that were awkward in previous models have either been fixed or eliminated from the ADO model.

The ADO objects provide you with the fastest, easiest and most productive means for accessing all kinds of data sources. The ADO model strives to expose everything that the underlying data provider can do, while still adding value by giving you shortcuts for common operations.

ADO is Microsoft's strategic, high-level interface to all kinds of data. ADO provides consistent, high-performance access to data, whether you're creating a front-end database client or middle-tier business object using an application, tool, language, or even an Internet browser. ADO is the single data interface you need to know for 1- to n-tier client/server and Web-based data-driven solution development.

ADO is designed as an easy-to-use application level interface to Microsoft's newest and most powerful data access paradigm, OLE DB. OLE DB provides high-performance access to any data source, including relational and non-relational databases, email and file systems, text and graphics, custom business objects, and more. ADO is implemented with a small footprint, minimal network traffic in key Internet scenarios, and a minimal number of layers between the front-end and data source-all to provide a lightweight, high-performance interface. ADO is easy to use because it is called using a familiar metaphor - the OLE Automation interface, available from just about any tool and language on the market today. And since ADO was designed to combine the best features of, and eventually replace RDO and DAO, it uses similar conventions with simplified semantics to make it easy to learn for today's developers.

2.12 Database Server.

2.12.1 Microsoft Access

Microsoft Access is a powerful relational database application with which a desktop user can efficiently create and manipulate database systems. Access targets the desktop category and works best for individuals and workgroups managing megabytes of data. For multi-user access to the same database, Access uses file-server architecture, rather than client-server architecture. Access is included in the Professional and Developer Editions of Microsoft Office.

As a leader in the desktop database category, Microsoft Access makes it easy for users to find and manage their data to make better business decisions. With strong integration with Microsoft Office, Access offers a similar appearance and functionality to that found in the popular Microsoft Word and Excel applications. For general business users, Access provides easy-to-use wizards throughout, such as the Database Wizard for getting up and running quickly, and the Simple Query Wizard for easily finding information from the data. More advanced users appreciate the power behind the Microsoft Visual Basic for Applications (VBA) programming language, programmable toolbars, and the freely distributable run-time version of Access available with the Office Developer Edition. The combination of ease of use and power in Access makes it the top choice among developers who frequently use Access as a front end to SQL Server in a client-server scenario.

Access has two major components. The first contains an application development environment for Visual Basic for Applications programmers that include forms technology, reports, and database administration. In addition, as mentioned earlier, there is also the user interface (UI) common to both Access and the other Office applications. Before Access 2000, users and developers were using the Jet data engine, whether they knew it or not. In the Access 2000 version, users and developers will be given a choice of data engine, another component. They can continue with an improved version of the default Access data engine, or MSDE, a new data engine option in Access 2000.

2.12.2 Oracle Database

The Microsoft Visual Database Tools have been designed to work with Oracle databases transparently we can design database diagrams, queries, and views in the same way we would for any database. In addition, the tools are Oracle-aware, allowing us to incorporate Oracle data objects such as synonyms, produce Oracle-specific SQL commands, and so on.

For Oracle databases, we will see some differences in the Project Explorer (File View in Microsoft Visual C++) and Data View windows. For example, when we add a connection to an Oracle database, we will see these folders under the connection: Database Diagrams, Tables, Views, Synonyms, Stored Procedures, and Functions.

- The Tables folder contains the base tables in your database. The Views folder contains any SELECT statements saved as views.
- The Synonyms folder contains synonyms in user database, which are names assigned to tables or views that, may thereafter be used to refer to them. In oracle we can also create new synonyms.
- The Functions folder contains the functions in user database.
- The Stored Procedures folder contains stored procedures.
- When user add a new trigger to an Oracle database from Data View, a template for the new trigger is displayed in the editor This template adheres to correct Oracle syntax for triggers (just as new triggers added for SQL Server databases do).

2.12.3 SQL Server

Microsoft SQL Server is the complete database and analysis offering for rapidly delivering the next generation of scalable e-commerce, line-of-business and data warehousing solutions.

SQL server benefits as following:

- **Fully Web-Enabled**

  Query, analyze and manipulate data over the Web. Use Extensible Markup Language (XML) in SQL Server 2000 to exchange data between loosely coupled systems. Access data easily and securely from a browser, through firewalls, and perform fast full-text searches of formatted documents. Analyze and link online analytical processing (OLAP) cubes, even over the Web. Perform click stream analysis to learn about your Web customers.

Highly Scalable and Reliable

Grow without limits with enhanced scalability and reliability features. Partition your database workload to achieve scale-out of applications. Take full advantage of Symmetric Multiprocessing (SMP) hardware, and, with the Microsoft Windows 2000 Data Center. Server operating system, support up to 32 CPU's and 64 GB of RAM.

Fastest Time-to-Market... Rapidly build, deploy, and manage e-commerce, line-of-business, and data warehousing solutions. Perform sophisticated data mining on customer and financial data. Reduce development time with the integrated T-SQL debugger, and develop your own functions that can be reused in different applications. SQL Server 2000 provides the fastest route to Web application development.

Chapter 3 - System Requirement Analysis

3.1 System Methodology

The software engineering process consists of a set of steps that encompass methods, tools and procedures. These steps are often referred to as software engineering paradigms or software life cycle models. A paradigm for software engineering is chosen based on the nature of the project and applications, the methods and tools to be used and the controls and deliverables that are required. Every system development process model (see Figure 3.0) includes system requirements (user, needs, resource) as input and a finished product as output.

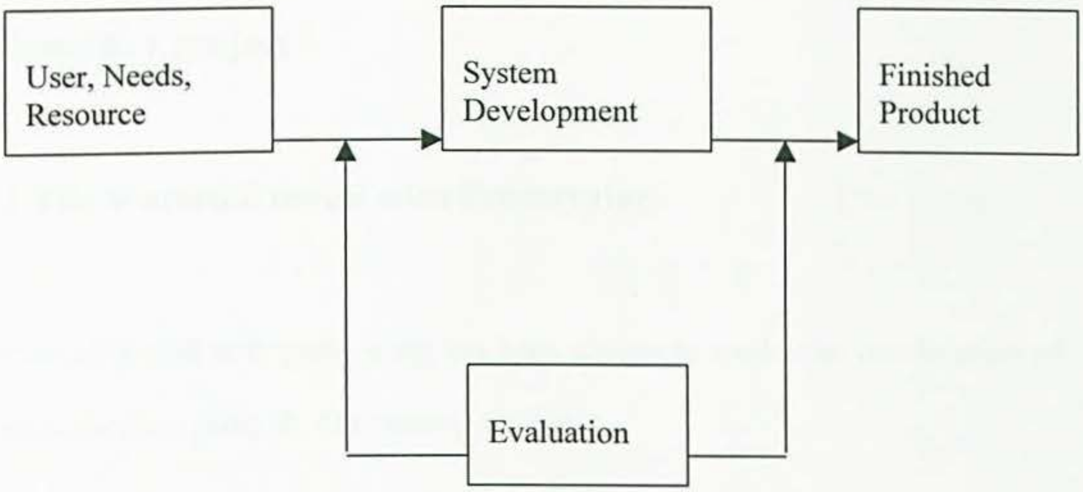


Figure 3.0 System Development Process Model

There are several process models in system development:

- Waterfall model
- V Model
- Waterfall Model with prototyping
- Spiral Model
- Prototyping Model
- Operational Specific Model
- Transformational Model
- System Development Life Cycle (SDLC)
- Authentication Method
- Encryption Model
- Butterfly Model

3.2 Model For Project

3.2.1 The Waterfall model with Prototyping

The waterfall model with prototyping has been chosen to model the development of ASP authentication using IP. The reasons are that: -

1. It is best suited for the actual environment as the development progress from a stage to another stager under the supervision of the lecturer.
2. It is useful to prescribe software development activities.

3. It is easy to use and understand.
4. Prototyping is included as a sub-process that enhances understanding.
5. It mainly helps user to identify key requirements of a system and demonstrate the feasibility of a design approach.

The extended the waterfall model (Pfleeger, 1991) by placing a larger emphasis on the testing component. In this model all steps are interconnected to allow a prototyping approach to be used in conjunction with the model. Every development stage should be completed before another stage begins. The Waterfall model with prototyping consists of 8 stages. The development stages are:

- Requirement Analysis

  Understand and determine the user's need by having brainstorming, eliciting and analyzing user requirements by having interview, survey or questionnaire session, collecting and specifying all the user requirements and validating requirements.

- System Design

  Outlining system functional by having feasibility studies or case studies on current system, determining and specifying hardware or software architecture and verifying system design.

- Program Design

  Determine and specifying program design and database design and verifying program design.

- Coding

  Involve programming personal planning, tool acquisition, database development, component level documentation and programming management.

- Unit and Integration Testing

  Test units separately and integrate the tested units. Then, testing on the integrated units.

- System Testing

  Combining all integrated user into a system. Specifying, reviewing and updating of the system and validating of system.

- Acceptance Testing

  Testing on system completed. The system is delivered.

- Operation and Maintenance

  Control and maintain the system. Revalidating of system.

The validation during system testing is to ensure that the system has implemented all the requirements, so that each system function can be tracked back to a particular requirement in the specification. As for the verification, it ensures that each function works correctly.

The aim of prototyping is to enable input from the end user at early stage by giving them the look and feel of the application. This is achieved by modeling the user interface while having little or no content behind that interface. Prototyping is especially valuable where requirements cannot be specified clearly.

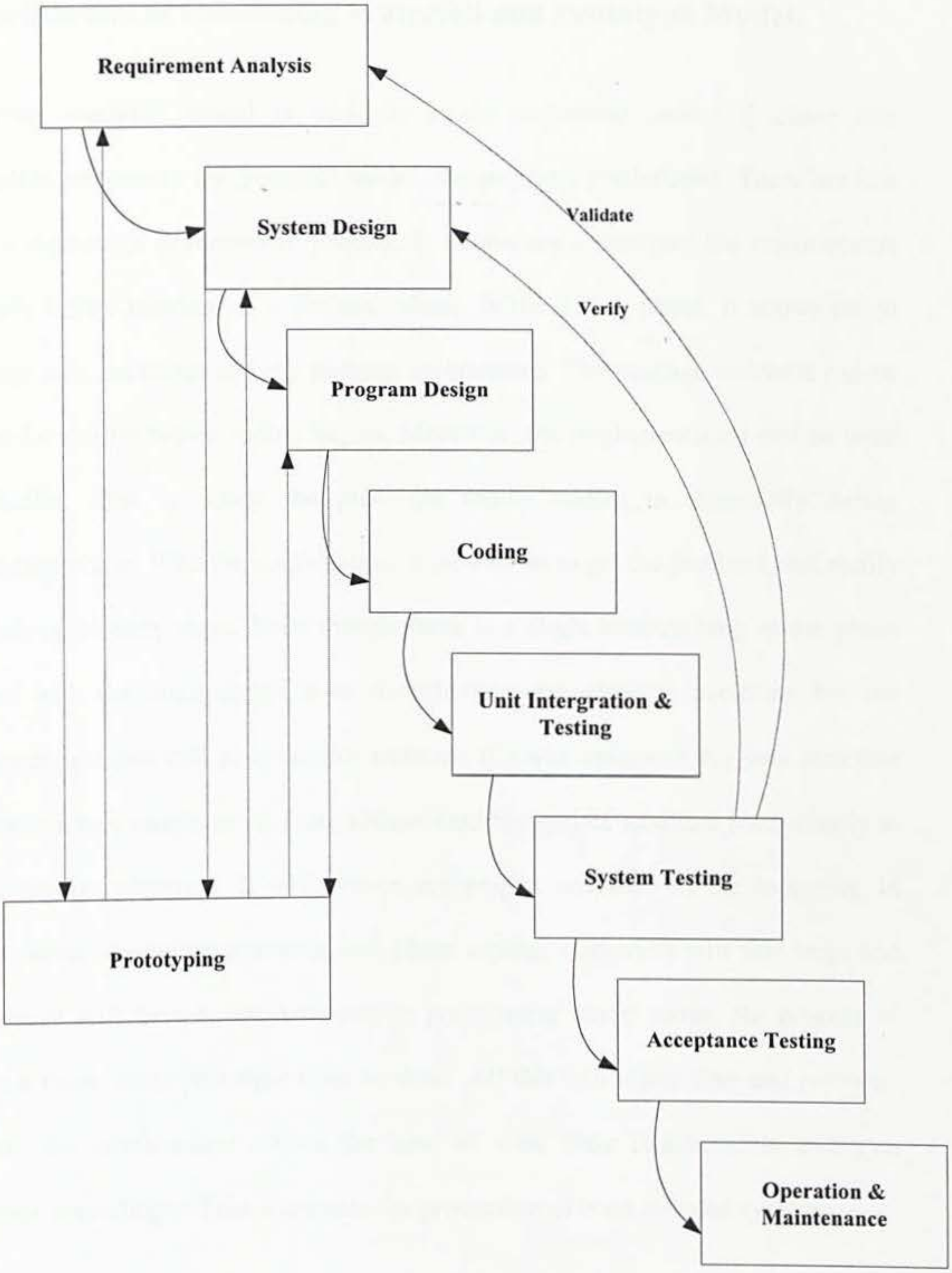


Figure 3.1 Waterfall Model With Prototyping

3.3 Justification of Combining Waterfall and Prototype Model.

As known, waterfall model is basically linear sequential model. I chose this combination because in the waterfall model, the steps are predestined. Therefore this creates a systematic development process. It allows me to analyze the requirements thoroughly before moving on to the next phase. In the design phase, it allows me to design my data structures and my package architecture. The package outcome can be accessed for quality before coding begins. Moreover, the implementation will be done mechanically. This is where the prototype model comes in, especially during requirements phase. With the combination, it permits us to get the feedback and rectify a problem at an early stage. Even though there is a slight backtracking in the phase flow but with the combination even though there are changes occurring but the development process still goes on. For instance, if I was designing my data structure and there is a new requirement, I can always modify my data structure immediately to suit my new requirement. It will reduce my project schedule in the long run. In waterfall alone, during maintenance and phase testing, customers will find bugs and enhancement will have to be done and in prototyping model alone, the process of building a throw away prototype must be done. All this will waste time and resource. Therefore the combination allows the user to state their requirements and gain satisfaction accordingly. This eliminates the production of bugs infected systems.

3.4 Global Requirements

This covers the overarching constraints that must inform and direct detailed requirements for web support. This encompasses compatibility across different implementations, as well as compatibility with current practice. Therefore,

1. Stableness of software.

The software for Asp authentication is intended to maintain the accurate operation of the website that involving both password authentication process and encryption. Since URLs may be reassigned at a server's discretion this requirement applies only for that period of time during which a URL identifies the same resource.

2. User Agent Interoperability.

All Asp authentication software should be able to work with any versioning-aware HTTP server. It is acceptable for some user agent/server combinations to provide special features that are not universally available, but the protocol should be sufficient that a basic level of functionality will be universal.

3. Style-free Versioning

The protocol like TCP/IP should not unnecessarily restrict version management style to any one paradigm. For instance, locking and version number assignment should be interoperable across servers and clients, even if there are some differences in their preferred models.

4. Separation of access to resources and access control

The protocol must separate the reservation and release of versioned resources from their access methods. Provided that consistency constraints are met before, during and after the modification of a versioned resource, no "right way" to access to a resource is enforced by the protocol. For instance, a user may request declare an intention to write after a GET, may POST a resource without releasing the lock, and might even request a lock via HTTP connection while getting the document via FTP.

5. Legacy Resource Support.

The protocol TCP/IP and the ASP authentication should enable authentication awareness for the server to work with existing resources and URLs. Special authentication information should not become a mandatory part of HTTP protocols except where it is required. Special version information that would break existing clients and servers, such as new mandatory headers, cannot therefore be required for GET (and possibly also for PUT/POST).

6.Legacy User Agent Support.

Servers should make versioned resources on Internet explorers accessible to versioning-unaware user-agents in a format acceptable to them.

3.5 Functional Requirement

Functional requirement is a statement of the service or functions that a system should provide how the system reacts to particular input and how the system should behave in particular situations. **[Sommerville, 1998]**

The following functional requirements are intended to satisfy the global requirements and enable the benefits. The mention of possible new HTTP methods is intended to make the discussion clearer and more concrete, not to rule out other methods of meeting the requirements. Though the technology depend upon the server and the windows it is important to analyze the compatibility of the ASP Authentication software to work on any platform. As far as the ASP Authentication technology is concerned, it is important to consider the protocols and the versioning support given by the operating systems like windows and also depends upon the network and type of the server used.

3.5.1 Access to specific named ASP versions via a URL

This is required for version-specific linking, and for legacy user-agent support. The ASP Authentication should work irrespective of versioning because the versioning depends upon the windows or depends upon the server also.

3.5.2 A URL to denote a versioned resource itself

This is more important if URL computations are not allowed, since an identifier is needed for queries about the versioning status of a resource. This is used to perform operations (such as adjusting attributes, changing locks, or reassigning URLs) that affect all versions of a resource, rather than any specific version. Asp Authentication software is intended to work on any windows based platform irrespective of versioning support or problem.

3.5.3 Direct Access To A Server_Defined

This is one of the simplest ways to guarantee legacy user-agent compatibility and legacy file compatibility. If no special version software's and URL's are used, the server will provide a default. This does not rule out the possibility of a server returning an error in case no such default exists. The ASP Encryption is able to work in such an environment even the versioning problem persists.

3.5.4 Way To Access Common Related URLs From A Versioned URL

Whether by server query, URL computation, or some other way:

- Root version(s) of this document
- Predecessor version(s) of this document
- Successor version(s) of this document
- Default version of this document

Some ASP versions of a resource are special. It must be possible in some way for a versioning-aware client to access common related ASP versions to the one it currently is displaying. Possible solutions include, but are not limited to: the server automatically adding header fields to a versioned URL specifying the URL of the common related versions, the server providing one or more query methods ("who is the previous version to this URL?"), or a standardized way to compute related URLs when given a versioned URL. We feel that access to the "default" version of a resource is an extremely important operation that a browser should be able to perform at any time that a versioned URL is seen.

3.5.5 Some Way To Determine ASP Version Identification

This requirement describe the ability to take the URL of a resource and determine:

URL for the resource

- A version identifier for the resource.
- ASP delimiter including the versioning.

Note that this kind of facility supports only some comparison operations: It enables the determination that two version-containing URLs designate ASP versions of the same resource. However, given the phenomenon of URL a liaising, it is insufficient to determine that they are not versions of the same resource.

3.5.6 A Way To Request Exclusive Access To ASP Version Of A Resource (LOCK)

Since not all systems implement lock-based access there is a question as how this should be implemented. Client use of this method could be optional, allowing some relatively strong guarantee on the meaning of acquiring a lock. Alternatively, clients could be expected to take a lock, but servers might implement different locking policies (possible even including implementation of LOCK and UNLOCK as NOPS).

3.5.7 A Way To Submit A New Version Of A Resource (PUT/POST)

The server should be able to attach it to the correct part of the version tree, based on the version number associated with the resource before its modification. The end user need not worry about the ASP versioning problem because it is taken care by the server and server maintenance once it is hosted on the Internet.

3.5.8 A Way For A User-Agent To Request A Version Identifier For A Checked Out ASP Version.

Such an identifier will not be used by any other user-agent in the meantime. The server may refuse the request. The end user may not know the ASP version or the ASP versioning procedure it is hidden by the server so end user need not worry about any thing.

3.6 Non-Functional Requirements.

Non-functional specifications are constrains under which a system must operate and the standard, which must be met by delivered system **[Sommerville, 1995].** The ASP Authentication Using IP must ensures certain application qualities like user-friendliness, correctness, functionality, reliability, flexibility, efficiently as well as maintainability.

3.6.1 Software Requirements

On the end user side

56Kbps modem with normal Internet connection and access to the website with Internet Explorer 3.0 or later version.

On the programmer side

Any HTML editor – Dream weaver 2.0 or later, FrontPage 98 or Later

Internet Information Server 2.0 or later / Personnel Web Server 2.0 or later

Configuring TCP/IP settings to universal port address depends on the server ID too.

Windows 95 or Later; NT 4.0 or later

3.6.2 Hardware Requirements

Pentium / Calderon 133 MHZ or faster processor

Microsoft Windows 95 or later; NT 4.0 or later

32MB Ram or more recommended for 2000; 256 MB for the server

Minimum 2GB hard disk space

Any Monitor to operate

3.6.3 User Interface Specifications

The devices with the user interacts are the keyboard, the mouse, and the computer screen. The keyboard is used to enter some textual information's, the mouse helps the user to select, draw, and connect components. The results of all the operation performed by the user are either displayed on the screen or sent to a file or a printer.

3.6.4 Interaction & Dialog Conventions

This system will act on a graphical user interface.

RATIONALE: Although this is a system constraint, using this type of interface will make it easier for the user to interact with the program, given the environment it will be designed in.

3.6.5 Key Bindings

This system has to work on the following different platforms UNIX / Win98 / WinNT / Win2000 platform. Therefore, the interfaces will include any key bindings that the platforms support such as:

- Ctrl + C to halt a running process
- Ctrl + Z to suspend a running process
- Ctrl + P to print

3.6.6 Exception Handling

3.6.6.1 Anticipated Error Conditions

These are conditions that will result in aborting. Conditions such as the following may exist:

- HTTP server not found error
- Insufficient memory
- Invalid query specification, e.g. node does not exist

RATIONALE: The system will be not able to operate correctly if one or many of these conditions are present.

3.6.6.2 Anticipated Warning Conditions

These are conditions that will result in a message, but the program will continue running. Conditions such as the following may exist: the machine doesn't accept the given language. If a node doesn't have any connections to other nodes.

RATIONALE: ASP Authentication will need to react to these conditions; however, these should not cause the processing to abort.

3.6.7 Usability

If the user has had some experience using a computer and some background in theory of final state machines, then the user should have no difficulty using the website. User documentation, including a reference guide and a short tutorial, will be provided with the final product.

3.6.8 Reliability

There are many situations that can affect the results of authentication procedures. These are situations such as incorrect user input of a text or user query. We will develop exception handling code to counteract these and situations like these. However, there are also many such situations that this software cannot predict. These include situations that involve outside factors, such as the operating system used ceasing to work. In such a case, this website on ASP Authentication Technique should not become unusable when the system is running again. This system should not fail to run after recovering from unforeseen situations.

3.6.9 Invalid Inputs

The System shall be able to handle invalid inputs in a manner that does not abnormally terminate the System, or allow any functions of the System to operate contrary to its description in this Document. The System will not accept invalid inputs - it will require any invalid inputs to be changed.

3.6.10 Power Loss

If the physical server computer loses power, or suffers physical damage while the System is operating and the System abnormally terminates, then the System will not be able to operate to its description in this Document. This will be the case even after power is restored. This is due to the System possibly having corrupted data - the administrator will need to manually replace the database of the System with a known reliable version.

3.6.11 Network Failure

A "network connection loss" is the inability of the client computer to contact the server computer via a local area network.

If a client computer loses its network connection with the server, then the client will not be able to operate to its description in this Document until such a time as the connection is restored. Any data being transferred at the time of the lost connection will be lost.

If a client application loses its connection with the server application, the server app will from that time onwards continue to operate to its description in this Document with any other client apps that did not lose their connections. The transaction in progress at the time of the connection loss will be discarded.

3.6.12 Security

The System must uphold the following aspects of security:

- The System should implement levels of access as described in the core requirements. That is, user groups and logins are expected for the System, akin to Unix or Windows 2000 user groups and logins. Lowest would be receptionist, then management, and then admin being the highest.

- If the "customer web-access" requirement is implemented, the transfer of information between the browser and the server must be secure. That is, only the information listed in the "customer web-access" requirement must be accessible to the member, and the information that is accessible to the member must not be accessible to a third party.

- If the web-access feature is implemented, the System will only allow read-only access of data over the Internet. That is, data cannot be added or changed over the Internet.

The security of the physical server computer and the database file of the System is the responsibility of the Client.

3.6.13 Response Times

ASP Authenticated result can be expected to respond to any user request within 10 seconds of the request being entered. This will depend greatly on the size of the web page and the performance of the operating system being used. It should be noted that if ASP Authentication were processing a query, it would not respond until completion of the query. During the interactive menu mode, processing will be indicated status bar.

3.6.14 Storage and Memory Size Constraints

This system is completely programmed in VBScript, ASP, and HTML/DHTML. So it just need the IIS / PWS (Internet Information Server / Personnel Web Server) and about 3MB for the system it self.

3.6.15 Portability

This program is written in pure ASP. So it will run on any platform, that supports Microsoft Windows Platform.

3.6.16 Modularity

This program will be designed in a modular fashion. Additional modules can be added without changing the rest of the system.

3.7 Chosen Platform, Database and Tools.

The ASP authentication will work only on the windows platform, which is dependent on any version of Internet access. The software is developed using ASP 3.0 on the developer platform in windows supportive technology like Notepad, Microsoft Interdev, Dream weaver etc. the effective backend used for this website is MSAccess 95 or any later version which is more compatible for ASP Authentication on windows server. The ASP program developed is hosted on the Internet information server to activate the existing ASP programs, which forms a supportive architecture to communicate the server side script.

3.7.1 Chosen Platform - Internet Information Server

3.7.1.1 Introduction

Microsoft Internet Information Server (IIS) is tightly integrated with the Microsoft Windows NT Server operating system to provide the most powerful Web server for organizations to take advantage of the opportunities of the Internet and intranets, while providing the highest levels of security for applications and information.

This integration means IIS offers the same robust security that is built into Windows NT from the very core. Windows NT was created intending to meet the security criteria for the U.S. Government's C2 Security Evaluation. The critical need for an operating system to be designed for optimum security from the ground up was noted by the NCSC, which wrote in its Final Evaluation Report of the Windows NT operating system: "When security is not an absolute requirement of the initial design, it is virtually impossible through later add-ons to provide the kind of uniform treatment to diverse system resources that Windows NT provides."

3.7.1.2 Integrated Security

The robust security architecture of Windows Server is used consistently across all system components, with authentication tied to controlled access to all system resources. IIS integrates into the Windows security model and operating system services such as the file system and directory. Because IIS uses the Windows Server user database, administrators do not need to create separate user accounts on every Web server, and intranet users need only to log on to their network once. IIS automatically uses the same file and group permissions as the existing file, print, and application servers.

Some Web servers install their own security implementations on top of the operating system, creating additional overhead and potential security exposure due to lack of integration and synchronization. Windows Server is inherently secure by design. Files and system objects can only be accessed with the proper permissions. User and group accounts are managed by a globally unique identification. When accounts are deleted, all access permissions and group memberships are deleted. So, even if a new account is created using a previous user name, none of the permissions are inherited.

3.7.1.3 Easy to Manage

Permissions to control access files and directories can be set graphically, because IIS uses the same Windows Server Access Control Lists (ACL) as all other Windows services, such as file sharing or Microsoft SQL Server permissions. Permissions for the Web server are not separate from other file services, so the same files can be securely accessed over other protocols, such as FTP, CIFS/SMB, or NFS without duplicating administration.

Administrators do not need to maintain multiple sets of user databases, and all of the services for literally hundreds of intranet servers can easily be managed from a single graphical tool. IIS and Windows are the only platform that ensures that administrators, with one mouse click, can give new users access to valuable network resources, such as HTML pages, shared files, printers, corporate databases, and legacy applications on all servers.

IIS produces standard Web server access logs to analyze usage. Integration with Windows Server also means IIS can take advantage of system auditing for more secure monitoring of resource use. For example, failed attempts to access a secure file can be recorded in the Windows Event Log, and audited with the same tools used for managing existing servers.

3.7.1.4 Comprehensive Solution

IIS takes full advantage of tight integration with Microsoft Proxy Server, Certificate Server, Site Server, BackOffice®, and other applications to provide a complete, robust platform with a rich spectrum of security functionality built in.

Only IIS and Windows Server provide a comprehensive platform for rapidly building, robust, scalable Web servers that are secure for both public and intranet Web sites. By building Web services to be part of the core Windows Server operating system, Microsoft IIS delivers high performance, easier management, and excellent security. All of this makes IIS the best platform for integrating with existing solutions as well as for delivering a new generation of Web applications.

3.7.1.5 Why Security Is Important

The introduction of intranets is opening vast opportunities for providing better access to information, improving business processes, and creating new business models. However, the open nature of the Web and its role as gateway to information and other business systems underscore the absolute need for using a Web server with a solid security foundation; it is also essential that the Web server is tightly integrated into the underlying operating system the network and applications run on. For example, security is vital for:

- Application and database security
- Electronic commerce
- Business relationships and extranets

- Communicating with customers
- Database and application access

3.7.1.6 Application and Database Security

A Web browser is increasingly being used to provide access to information and applications in databases and other existing business systems. For example, many businesses are allowing employees to manage their personal information and benefits plans through Web browsers that link back to HR systems. These business systems must be protected so users are allowed to access only applications they have authorization for, and so employees can change only their personal information. This requires first identifying users, ascertaining that they are who they say they are, and determining if they have permission to view the information or to perform the task requested. This last step often requires integration with existing information systems. Moreover, the exchange between client and server must take place over a secure channel to ensure private information transfer. Windows Server and IIS provide these integrated services that enable companies to securely connect the Web with databases and business applications

3.7.1.7 Framework for Using Security

Microsoft designed IIS and Windows Server to provide administrators with a powerful framework for deploying Web servers. Above all, IIS and Windows NT Server provide administrators with a single integrated security model. In other words, IIS security is fully integrated with Windows security. This gives it a number of advantages, including the ability to:

- Take full advantage of the strong, secure underpinnings of the U.S. Government C2 and ITSEC FC2-rated Windows NT security.
- Eliminate possibilities for security weaknesses and holes by not adding redundant security layers. This sets IIS apart from other operating systems and Web servers with multiple security layers that increase their complexity and possibility for security holes.
- Take advantage of existing Window NT knowledge, making it easy to learn and configure.
- Provide better performance by eliminating unnecessary overheads of additional security and access control layers.

The framework allows the administrator to determine everything from what type of end user authentication will be used on the Web server, to how the Web server itself will be physically locked down.

3.7.1.8 Access Control

One of the most important areas of focus for IIS is providing powerful access control functionality for Web access to files and applications on the server. IIS was designed to make it easy to use a wide range of access control mechanisms to critical business data, depending on the needs of the organization. These include the following:

- Support for the Windows NT Challenge/Response (NTLM) authentication
- IP address grant/deny restrictions
- Ability to implement restrictions on virtual servers and directories
- Support for the Windows NT File System (NTFS)
- Impersonation of users when running applications
- Client and server digital certificates
- Advanced security filters

IIS provides a set of open APIs that developers can use to create filters that authenticate users based on custom rules. This gives administrators the flexibility to control access using any authentication scheme or external directories.

Once users are authenticated, IIS checks to see if they have permission to access the requested file or application.

On the Internet, each server and client (or proxy for a group of clients) has a specific Internet address called the "IP address." IIS can be configured to grant or deny access to specified IP addresses. This gives the administrator the ability to exclude users by denying access from a particular IP address, or prevent entire networks from accessing the server. Conversely, administrators can choose to allow only specific IP addresses to have access to the service.

The Windows NT File System (NTFS) was designed to provide security features required for high-end Web servers in both intranet and Internet scenarios. The NTFS file system supports discretionary access control and ownership privileges that are important for the integrity of critical business data. NTFS allows administrators to assign permission to individual files, not just to folders and directories. By using the NTFS file system for the content made available by IIS, administrators can help ensure only the right individuals have access to individual files on the Web server.

Once the user's IP address restrictions are satisfied, the user name or password is validated, and the service's virtual directory permissions are completed, IIS will then attempt to access the specified resource (based on the URL) using the security context of the authenticated user. This allows Windows NT Server to enforce access control based on NTFS permissions on the resources, offering administrators extremely granular control over sensitive resources and data.

Windows NT identifies each user by a globally unique security identification (SID), not by user name. This SID is mapped in the background to the user's account name, so file permissions and group accounts are managed using a friendly name but applied using the SID. When an account is deleted, all ACLs and group assignments for the account are also removed. SIDs and synchronization ensures that an account later created with the same user name cannot inherit permission to the old account.

3.7.1.9 User Authentication and Authorization

IIS security is integrated with the Windows Directory Service, and a user with a valid Windows user account must access every resource. This allows administrators to use the full power of the Windows Directory Service account management, including the ability to audit and log all activity, set time of day restrictions, expire passwords, and force secure password policies.

At setup, IIS creates an anonymous account for unauthenticated Web connections. When file security is not required, the server in the security context of this anonymous user account processes the request. The anonymous user account can access only files and applications for which permission has been granted.

Files and applications can be restricted to access only by specific users or groups. This requires obtaining and verifying the user name. IIS can be configured to require basic HTTP authentication. Users are prompted for a name and password, which are compared to accounts in the Windows NT Server directory. However, the name and password in basic authentication are passed as clear text over the network, and can potentially be intercepted by a network packet sniffer.

3.8 Active Server Pages (ASP)

Microsoft Active Server Pages (ASP) is a server-side scripting environment that user can use to create and run dynamic, interactive Web server applications. With ASP, user can combine HTML pages, script commands, and COM components to create interactive Web pages or powerful Web-based applications, which are easy to develop and modify. Among the various attributes of ASP are:

- Browser independent for it is a server-side scripting language.
- ASP language is six times faster to write than other conventional Web page design methods
- ASP takes less time to write + debug (no compilation), thus less "down" time for Web sites
- ASP outperforms (by 5-to-1 ratio) other conventional Web page design methods (CGI, etc.)
- ASP allows for multiple browsers, doesn't restrict a user to any "one" particular browser type
- ASP Web design is "dynamic", continuous changes can be made "on the fly" effortlessly
- ASP is inherently multi-threaded (CGI isn't) allowing a greater number of concurrent users
- ASP's memory resident engine uses minimal server resources, thus greater performance
- ASP allows amazing design power in developing database applications and e-commerce.

3.9 Chosen OS

For the ASP Authentication Using IP, Windows 2000 is chosen as the OS. Microsoft's Window 2000 is built to work with a series of microprocessors from the Intel Corporation that share the same or similar sets of instructions.

The Microsoft Windows 2000 Server operating system simplifies deployment and management of network security with Windows IP Security, a robust implementation of the IP Security Protocol (IPSec).

The need for Internet Protocol (IP)–based network security is already great and is growing. In today's massively interconnected business world of the Internet, intranets, branch offices, and remote access, sensitive information constantly crosses the networks. The challenge for network administrators and other IS professionals is to ensure that this traffic is:

- Safe from data modification while enroute.
- Safe from interception, viewing, or copying.
- Safe from being accessed by unauthenticated parties.

These issues are known as data integrity, confidentiality, and authentication. In addition, replay protection prevents acceptance of a resent packet.

Designed by the Internet Engineering Task Force (IETF) for the Internet Protocol, IPSec supports network-level authentication, data integrity, and encryption. IPSec integrates with the inherent security of the Windows 2000 Server operating system to provide the ideal platform for safeguarding intranet and Internet communications.

Microsoft Windows IP Security uses industry-standard encryption algorithms and a comprehensive security management approach to provide security for all TCP/IP communications on both sides of an organization's firewall. The result is a Windows 2000 Server end-to-end security strategy that defends against both external and internal attacks.

Because Windows IP Security is deployed below the transport level, network managers (and software vendors) are spared the trouble and expense of trying to deploy and coordinate security one application at a time. By deploying Windows 2000 Server, network managers provide a strong layer of protection for the entire network, with applications automatically inheriting the safeguards of Windows 2000 Server security. The encryption support of Windows IP Security extends to Virtual Private Networks (VPNs), as well.

Network administrators and managers benefit from integration of IPSec with Windows 2000 Server for a number of reasons, including:

- **Open industry standard**—IPSec provides an open industry-standard alternative to proprietary IP encryption technologies. Network managers benefit from the resulting interoperability.
- **Transparency**—IPSec exists below the transport layer, making it transparent to applications and users, meaning there is no need to change network

applications on a user's desktop when IPSec is implemented in the firewall or router.

- **Authentication**—Strong authentication services prevent the interception of data by using falsely claimed identities.
- **Confidentiality**—Confidentiality services prevent unauthorized access to sensitive data as it passes between communicating parties.
- **Data integrity**—IP authentication headers and variations of hash message authentication code ensure data integrity during communications.
- **Dynamic rekeying**—Dynamic rekeying during ongoing communications helps protect against attacks.
- **Secure links end to end**—Windows IP Security provides secure links end to end for private network users within the same domain or across any trusted domain in the enterprise.
- **Centralized management**—Network administrators use security policies and filters to provide appropriate levels of security, based on user, work group, or other criteria. Centralized management reduces administrative overhead costs.
- **Flexibility**—The flexibility of Windows IP Security allows policies to apply enterprise-wide or to a single workstation.

All of this is good news to network managers and other IS professionals charged with protecting the security of information. The explosive growth of intranets and the increasing integration of corporate networks with the Internet have caused an even greater need for security. Although the classic security concern is to protect data from outsiders, Windows IP Security also provides protection against attacks from what is the more likely source—unauthorized access by insiders.

Whether setting security profiles for key workgroups or the entire network, the encryption support of Windows IP Security can provide network managers with the peace of mind that comes from protecting an enterprise's communications.

3.10 Project Working Model

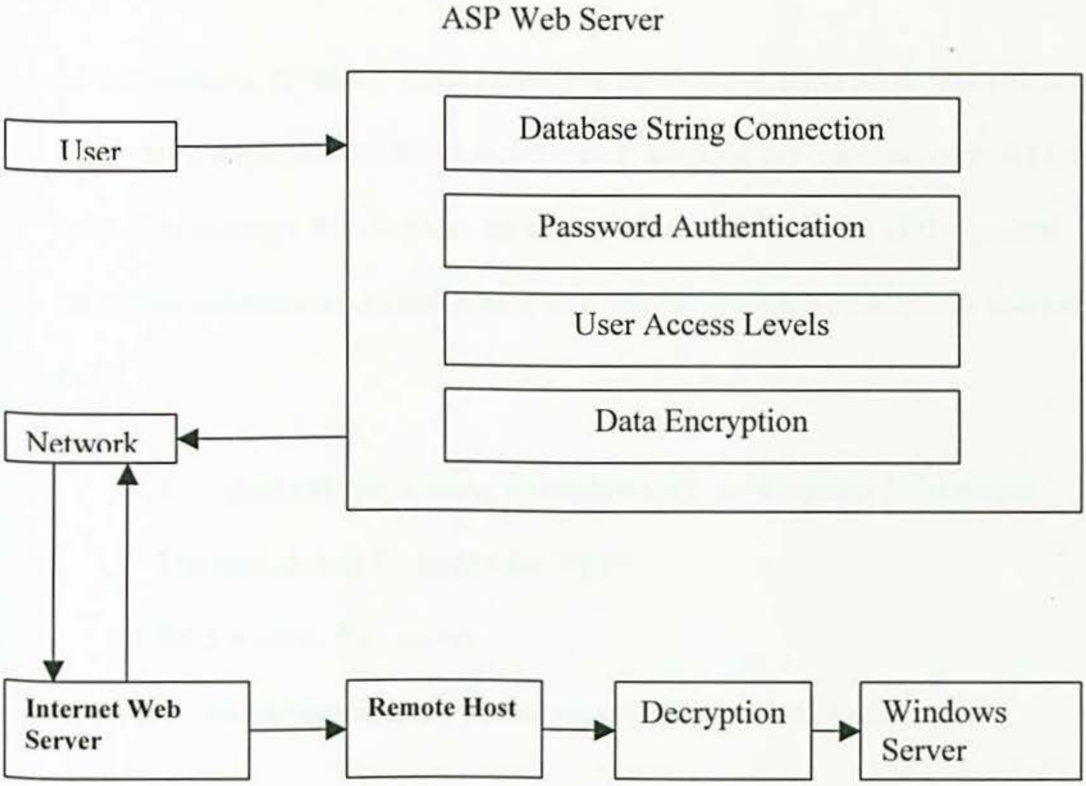


Figure 3.2 – ASP Authentication – Password Encryption

3.11 Authentication

Authentication is the process of obtaining identification credentials such as name and password from a user and validating those credentials against some authority. If the credentials are valid, the entity that submitted the credentials is considered an authenticated identity. Once an identity has been authenticated, the authorization process determines whether that identity has access to a given resource.

3.11.1 ASP Authentication Using IIS

In this scenario, an administrator is setting up an application on an intranet Web site for posting employee information. However, some of the information is for managers only. The manager information can be posted to a subdirectory of the general employee information, so that access to it can be limited. The scenario also assumes that:

- The administrator is using a Windows NT or Windows 2000 server.
- The hard disk is formatted for NTFS.
- IIS 5.0 is the Web server.
- All employees needing access are using Windows platforms.

The administrator:

1. Creates the files and directories shown in the following figure.

Files and directories

c:\inetpub

- wwwroot (Manager Information)
- User Directory\ *.asp

2. Creates a Windows group called Managers that contains all users who should have access to the ASP file.
3. Sets up Windows authentication using the IIS administration tool (window).
4. Sets the **impersonate** element in the ASP configuration file to true.
5. Sets the NTFS for the Manager Information directory to allow access to only those identities that are in the Windows Manager group. Note that the local system still needs access as well so that the ASP process itself can read the files. Settings in this directory typically resemble the following:
    a. Remove access from the everyone group, if it has access.
    b. Deny anonymous users.
    c. Add accounts that are to have access privileges.
    d. Give the system account access.

This provides the necessary security without the necessity of writing any code.

3.11.2 Basic Authentication

When IIS is configured for Basic authentication, it instructs the browser to send the user's credentials over HTTP. Passwords and user names are encoded using Base64 encoding. Although the password is encoded, it is considered insecure due its ability to be deciphered relatively easily. The browser prompts the user with a dialog box, and then reissues the original anonymous request with the supplied credentials, including the user name and password. A pop-up logon dialog box may or may not be appropriate, depending upon your user interface design requirements. Most Internet browsers support Basic authentication.

3.11.3 Forms Authentication

Forms authentication refers to a custom user interface component that accepts user credentials; for example, a user name and password. Many Internet applications used today present such forms for users to log on. It is important to note that the form itself does not perform the authentication and is provided solely as a way of obtaining the user credentials. ***The authentication is performed by accessing the user name and password database using custom code.***

When the user is authenticated, the server typically gives the client some means to indicate that it has already been authenticated for subsequent requests. If required, you can force the client to authenticate upon every request, although this impacts performance and scalability. There are two basic approaches that you should consider to identify a client who has previously logged on:

- **Cookies**. A cookie is a small piece of data initially presented by the server to the client. The client subsequently presents it back to the server within each HTTP request. This can be used as an indication that the client has already been authenticated. ASP provides a mechanism for you to use cookies for Forms authentication in the Cookies module. Cookies are supported by most Web browsers, including Internet Explorer and Netscape Navigator.
- **Custom**. You can implement your own custom mechanism to identify the client to the server. If your clients have disabled cookies, you may consider storing a unique identifier within each URL query string. You can also use hidden form fields, which are stored in a persistent top-level or non-visible frame. In either case, you need to make sure that a hacker cannot simulate being authenticated to your application programmatically.

Cookies are widely utilized by Web sites that implement Forms authentication.

3.11.3.1 Typical usage scenarios

You should consider Forms authentication when:

- User names and passwords are stored somewhere other than Windows Accounts. Note that it is possible to use Forms authentication with Windows Accounts.
- You are deploying your application over the Internet.
- You need to support all browsers and client operating systems.
- You want to provide your own user interface form as a logon page.

You should not consider Forms authentication when:

- You are deploying an application on a corporate intranet and can take advantage of Integrated Windows authentication.
- You are unable to perform programmatic access to verify the user name and password.

3.11.4 Web Service Authentication

Web services will need to accept user credentials in some manner. If the service is non-interactive, it will need to either obtain the security token of the caller, or it will need to expose the appropriate methods to allow credentials to be supplied. The following authentication methods can be easily used and do not require users to input credentials, making them good choices for programmable Web services:

- Basic, Digest, and Integrated Windows authentication
- Certificate Mapping authentication
- Application specific or custom authentication

Potentially, you could also use:

- Internet Protocol Security

3.12 Encryption and Decryption

Encryption is the process of encoding data into cipher, a form unreadable without a decoding key. *Decryption* is the reverse process of converting encoded data to its original un-encoded, *plaintext*, and form. When a user encodes a file, another user cannot decode and read the file without the decryption key. Adding a *digital signature*, a form of personal authentication, ensures the integrity of the original message.

There are two primary approaches to encryption: *symmetric* and *public-key*. Symmetric encryption is the most common type of encryption and uses the same key for encoding and decoding data. This key is known as a *session key*. Public-key encryption uses two different keys, a public key and a private key. One key encodes the message and the other decodes it. The public key is widely distributed while the private key is secret.

3.12.1 Common Encryption Algorithms

The encryption algorithms available to an application depend on the CSP used. Each encryption algorithm described here is supplied with the Microsoft RSA Base Provider.

The following table shows several encryption algorithms along with some performance benchmarks. These figures are for comparison purposes only. Your setup time and encryption speed may vary.

Table 3.0 Algorithms

Cipher	Cipher Type	Key Setup Time (Microseconds)	Encryption Speed (Bytes/Second)
DES	64-bit block	460	1.1 MB
RC2	64-bit block	40	290 KB
RC4	Stream	151	2.4 MB

RC2 and RC4 are variable-key-length ciphers. However, when using Crypto API with the Microsoft RSA Base Provider, these key lengths are hard-coded to 40 bits.

3.12.2 Data Encryption

SNA Server allows you to encrypt data for client-to-server and server-to-server communications, as shown in Figure 3.3

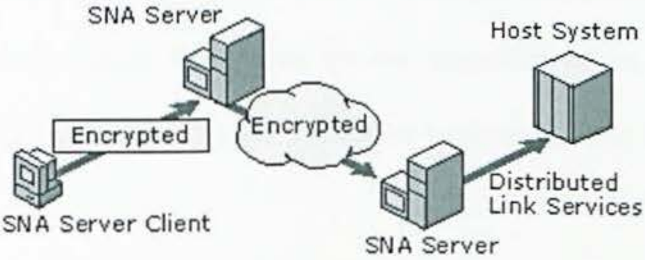


Figure 3.3 Model of Client-to-Server and Server-to-Server Data Encryption

Client-to-server encryption prevents information from being sent in plaintext between computers running SNA Server Client software and computers running SNA Server. Data encryptión enhances network security on the client-to-server communications path for all applications using SNA Server Client connections, including 3270/5250 emulators and APPC logon IDs and passwords. Data encryption can be enabled on a user-by-user basis using the SNA Server Manager.

Server-to-server encryption can be used to provide secure communications across your network, the Internet, or any other wide area network (WAN). If a user enables data encryption, information transferred through the Distributed Link Services is secure.

3.12.3 Symmetric Key Encryption

Encryption algorithms that use the same key for encrypting and for decrypting information are called symmetric-key algorithms. The symmetric key is also called a secret key because it is kept as a shared secret between the sender and receiver of information. Otherwise, the confidentiality of the encrypted information is compromised. Figure 3.4 shows basic symmetric key encryption and decryption.

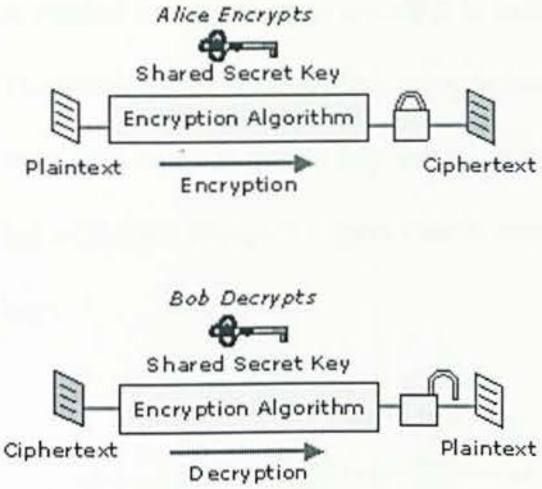


Figure 3.4 Encryption and Decryption with a Symmetric Key

Symmetric key encryption is much faster than public key encryption, often by 100 to 1,000 times. Because public key encryption places a much heavier computational load on computer processors than symmetric key encryption, symmetric key technology is generally used to provide secrecy for the bulk encryption and decryption of information.

Symmetric keys are commonly used by security protocols as *session keys* for confidential online communications.

3.12.4 Public Key Encryption

Encryption algorithms that use different keys for encrypting and decrypting information are most often called public-key algorithms but are sometimes also called *asymmetric key algorithms*. Public key encryption requires the use of both a private key (a key that is known only to its owner) and a public key (a key that is available to and known to other entities on the network). A user's public key, for example, can be published in the directory so that it is accessible to other people in the organization. The two keys are different but complementary in function. Information that is encrypted with the public key can be decrypted only with the corresponding private key of the set. Figure 3.5 shows basic encryption and decryption with asymmetric keys.

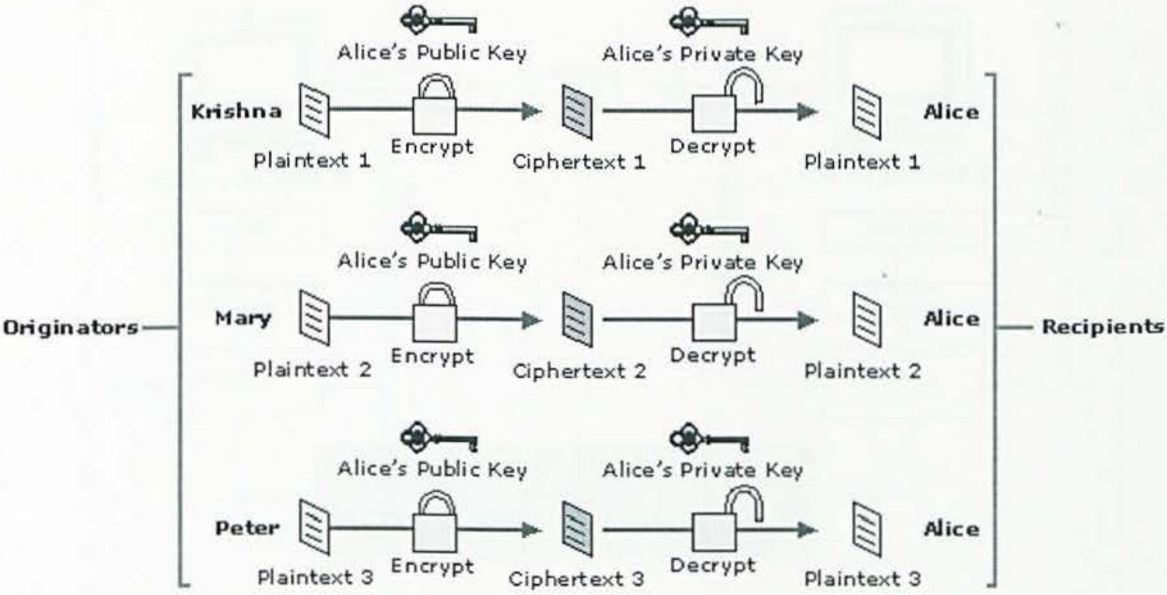


Figure 3.5 Encryption and Decryption with Asymmetric Keys

3.12.5 Diagramming a Basic Deployment (ASP Encryption Model)

In the example in Figure 3.7, a user on Host A is sending data to a user on Host B. Windows IP Security has been implemented for both computers.

At the user level, the process of securing the IP packets is transparent. User 1 launches an application that uses the TCP/IP protocol, such as FTP, and sends the data to User 2.

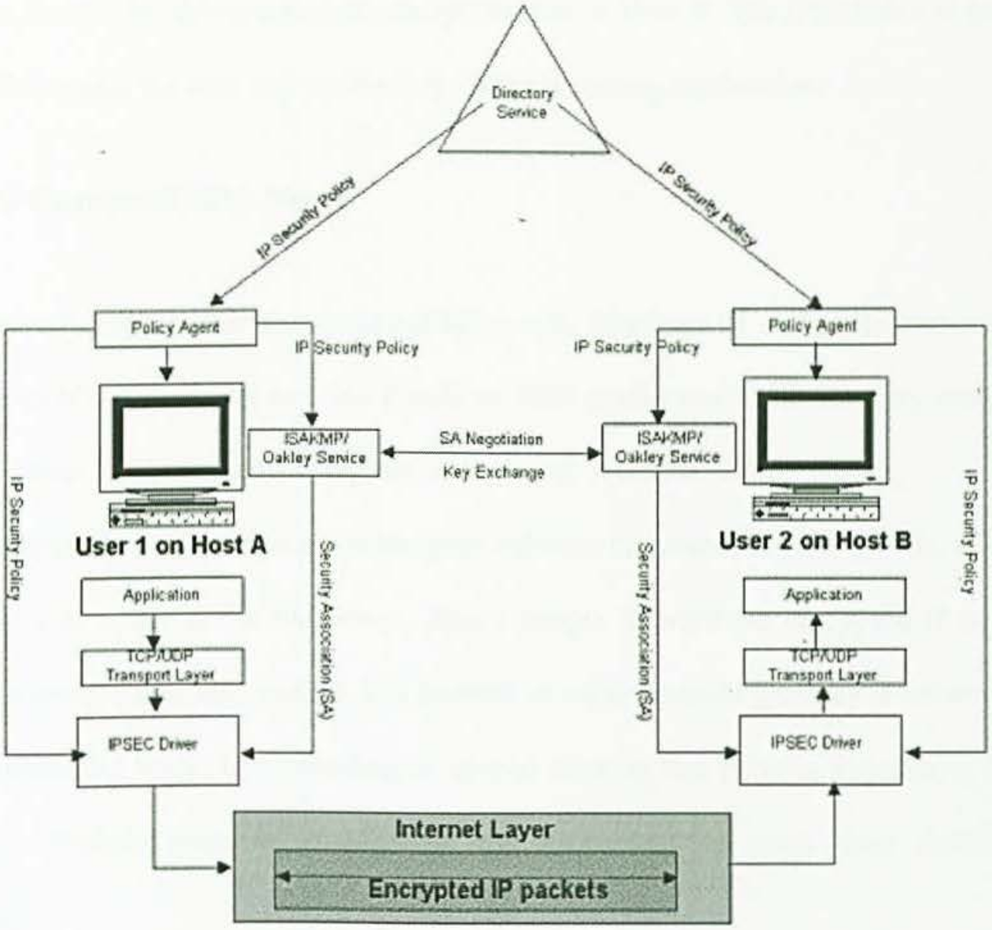


Figure 3.6. Example of a Windows IP Security deployment

The security policies assigned to Host A and Host B by the administrator determine the level of security for the communication. These are picked up by the policy agent and passed to the ISAKMP/Oakley service and IPSEC driver. The ISAKMP/Oakley service on each computer uses the negotiation policies associated with the assigned security policy to establish the key and a common negotiation method (a security association). The results of the ISAKMP policy negotiation between the two computers are passed to the IPSEC driver, which uses the key to encrypt the data.

Finally, the IPSEC driver sends the encrypted data to Host B. The IPSEC driver on Host B decrypts the data and passes it up to the receiving application.

3.12.6 Compatibility Notes

To ensure full communications compatibility with Windows 9x and earlier versions of Windows NT, a computer running Windows 2000 configured for IP Security sends the data without encryption to a computer not running Windows 2000.

Any routers or switches that are in the path between the communicating hosts, whether two users or a user and a file server, should simply forward the encrypted IP packets through toward their destination. If a firewall or other security gateway is between the communicating hosts, IP forwarding or special filtering that permits forwarding for IP Security Packets must be enabled for the IP packets to reach their destination correctly.

3.12.7 Enabling SSL Encryption on Windows 2000

To enable SSL encryption for your IIS virtual directory on Windows 2000 with IIS 5.0

1. Start the IIS snap-in by clicking the **Start** button, choosing **Settings**, and then clicking **Control Panel**. In Control Panel, double-click **Administrative Tools**. In Administrative Tools, double-click **Internet Services Manager**.

2. On Windows 2000 Server, start the IIS snap-in by clicking the **Start** button, choose **Programs, Administrative Tools**, and then clicking **Internet Services Manager**.

3. In the left pane, expand the computer containing your Web site, and then expand the list of virtual directories for your Web site.

4. From the list of virtual directories, right-click the IIS virtual directory you configured for SQL Server CE replication or RDA. Click **Properties** and click the **Directory Security** tab.

5. Under **Secure Communications**, click **Edit** to display the **Secure Communications** sheet.

6. Select **require secure channel (SSL)** check box to enable SSL encryption. You can select **require 128-bit encryption** if this option is provided. Export restrictions and national laws might limit the availability of this option.

7. Click **OK** to close the **Secure Communications** property sheet.

8. Click **OK** to close the **Directory Security** property sheet.

9. Close Internet Services Manager.

Chapter 4 - System Design

4.1 Introduction

The design phase in this project is the important stage, which gives the entire information about the project. This phase is done after the analysis stage. Here in this project ASP authentication using IP is design to cater the needs of a common website where the login and system authentication becomes very important. Design focuses both on the logical and physical or technical aspects of the system.

The design includes the various aspects and features of the ASP Authentication using IP. The design displays the information on website design, database structure, functional and other logical aspects of the system.

At the same time, to make this system more interesting and attractive, its content graphics, pictures, suitable page layout, fonts, color and other elements make the system more interesting and attractive.

4.2 System Functionality Design

4.2.1 System Structure Chart

The objective of system structure chart is to show how the modules in ASP Authentication Using IP are related to each other.

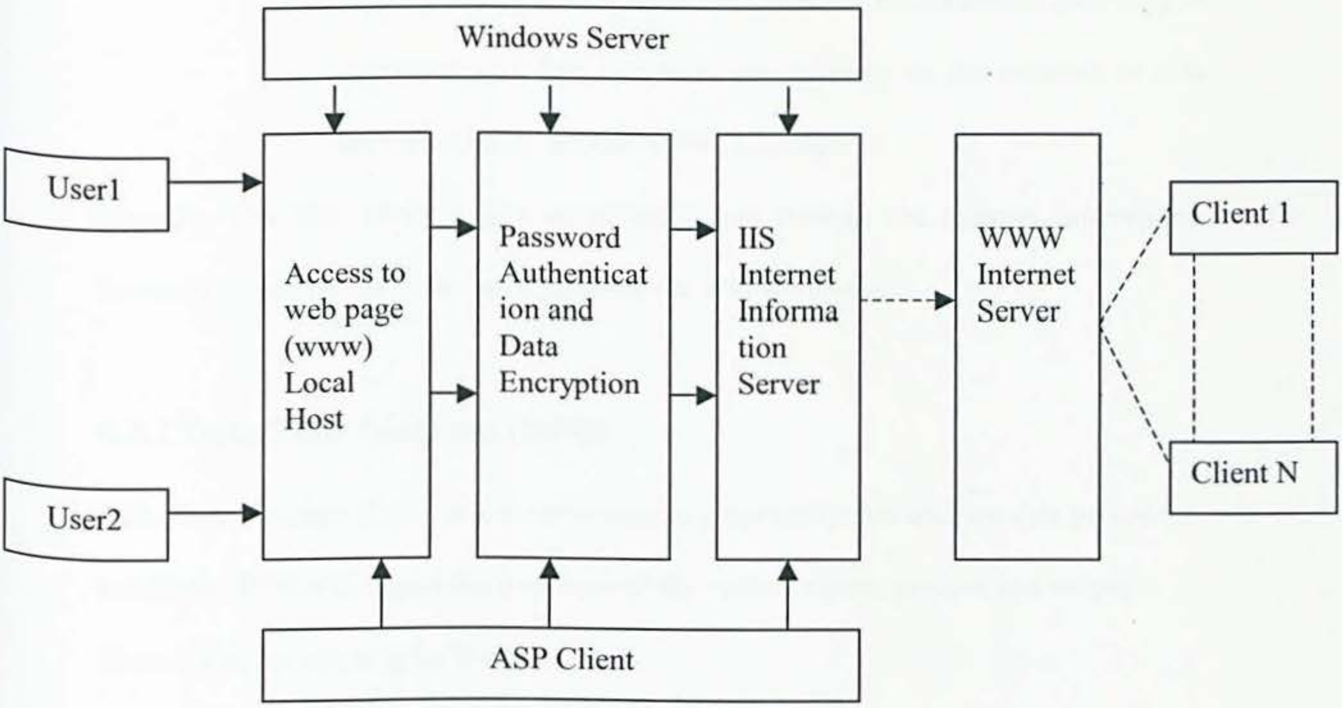

Windows Server
User1
User2
Access to web page (www) Local Host
Password Authenticat ion and Data Encryption
IIS Internet Informa tion Server
WWW Internet Server
Client 1
Client N
ASP Client

Figure 4.0 – ASP Authentication – Password Encryption – Internet Operation.

The above figure describes the working operation of ASP Authentication using IP addressing. Basically this system is designed to interact with any number of users at the same time. For instance user 1 and user 2 login at same time, their login status is detected by the local host and redirects the program response to the Internet Information Server. The overall operation is described in the steps below.

Step 1: When the local host recognizes user 1 and user 2-login immediately, the local host gets the user status from the Windows Server and the Asp Client program.

Step 2: The ASP client program starts its operation

a. Password Authentication: The user password is verified from the redirected response from the server, which displays the user existence check.

b. Data Encryption: In this process first the authenticated password is encrypted and then the user data existing on the network is also encrypted using the encryption techniques.

Step 3: The ASP client finally sends the output through the Internet Information Server, then redirected to the open gateway for WWW services.

4.2.2 Data Flow Diagram (DFD)

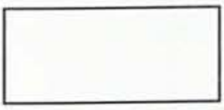
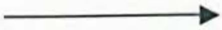
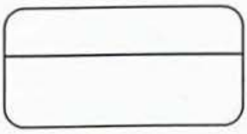
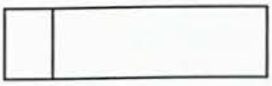
Data Flow Diagram (DFD) is a method used to graphically characterize data processes and flows. DFD will depict the overview of the system inputs, process and outputs.

The advantages of using DFD are:

- Further understanding of the interrelated of modules and sub modules.
- Analysis of a proposed system to determine if the necessary data and processes have been defined.

DFD is easy to be understood as it has symbols that specify the physical aspects of implementation. There are four symbols in DFD: entity, flow of data, process and data stores (See Table 4.0).

Table 4.0 DFD Symbols

Symbols	Attribute
	**Entity**
	**Flow of Data**
	**Process**
	**Data Store**

C.Gane and T.Sarson base the convention, which is used to design DFD on the work. The data flow is conceptualized with a top-down perspective. So, the Context Level diagram will be drawn, followed by the Diagram 0. Diagram 0 is an overview process of all the major modules in the ASP Authentication Using IP that includes all the data stores, entities and process involved.

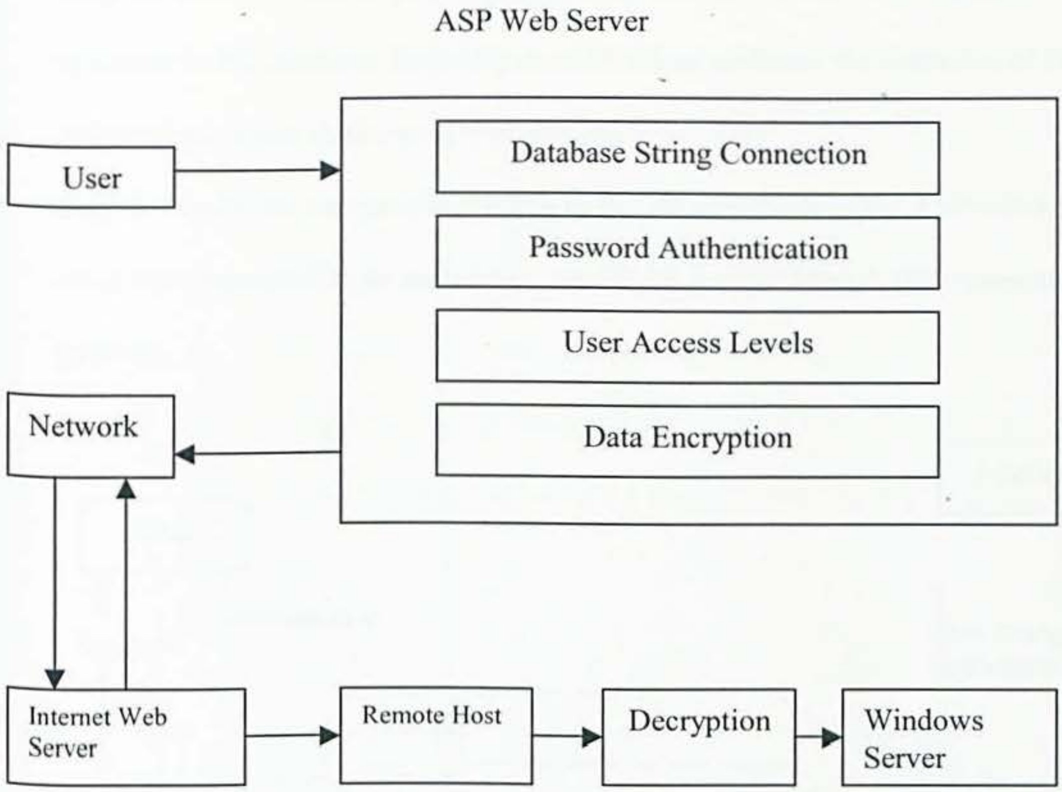


Figure 4.1: Context Analysis Diagram

The figure 4.1 gives the overall layout for the internal data flow for the entire model. The process of data flow could be explained in steps.

Step 1: The user enters the systems through the login services offered in the website. Immediately when the user logins, the user status is detected by the local host and it sends the user existence status to the windows server where the user is connected to the database in server through a database connection string which detects the type and mode of the database operation.

Step 2: The ASP client gets the response from the windows server about the user existence in the database. Secondly the ASP client performs the operation of Password Authentication and Data encryption process.

Step 3: Finally the outcome is sending to the IIS server where the ASP client program takes the responsibility to redirect to the WWW Server through the common existing gateway.

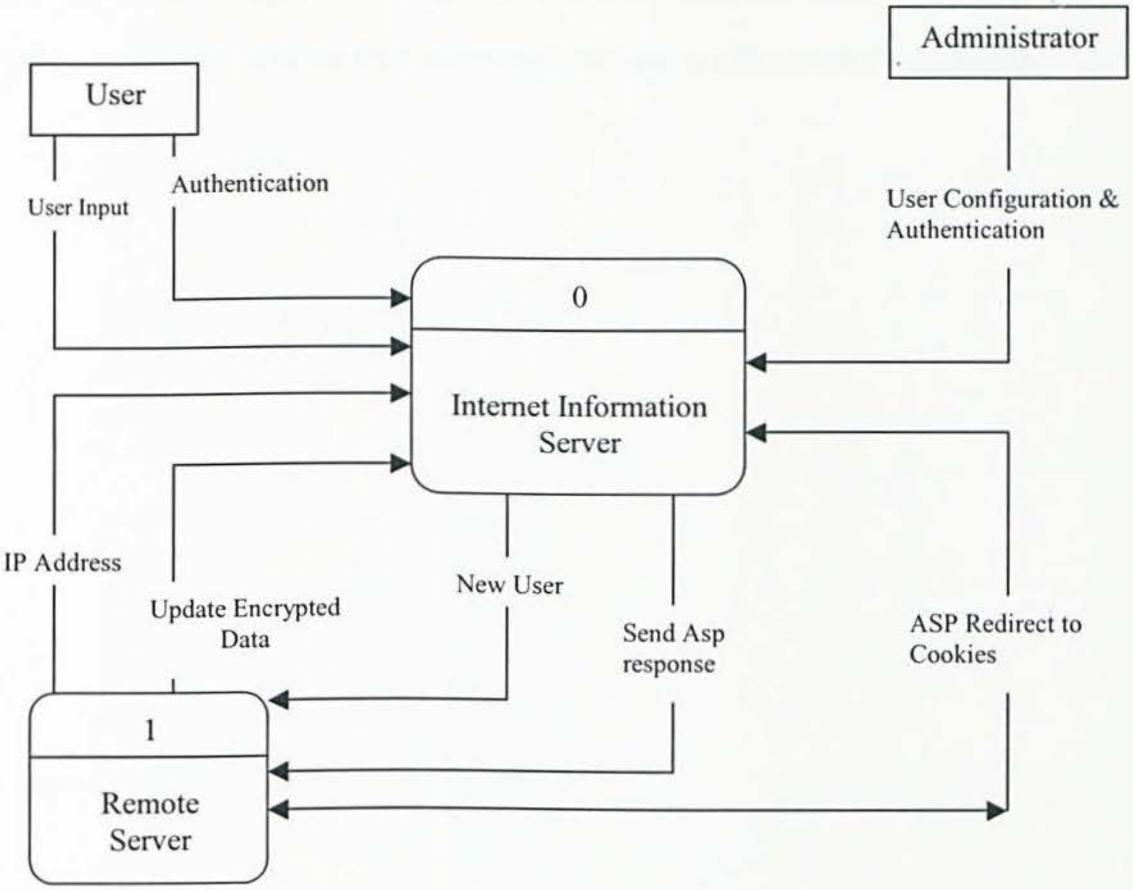


Figure 4.2: Diagram 0

The figure 4.2 shows the first stage of the data flow diagram where it is assumed on the real time implementation basis that ASP clients are already hosted in IIS server and IIS server is already installed in Windows Server, which has the rights to access all the clients in the network.

In the above figure 4.2, the user is authenticated and sends user information to IIS; on the other part the Administrator from the windows server detects the user existence check and directs the response of user status to the IIS Server. The IIS server takes the responsibility to connect to the remote host by transmitting the user data through the Internet or common gateway to any other network. Finally the remote server redirects the user information to the IIS Server where the user is authenticated permanently.

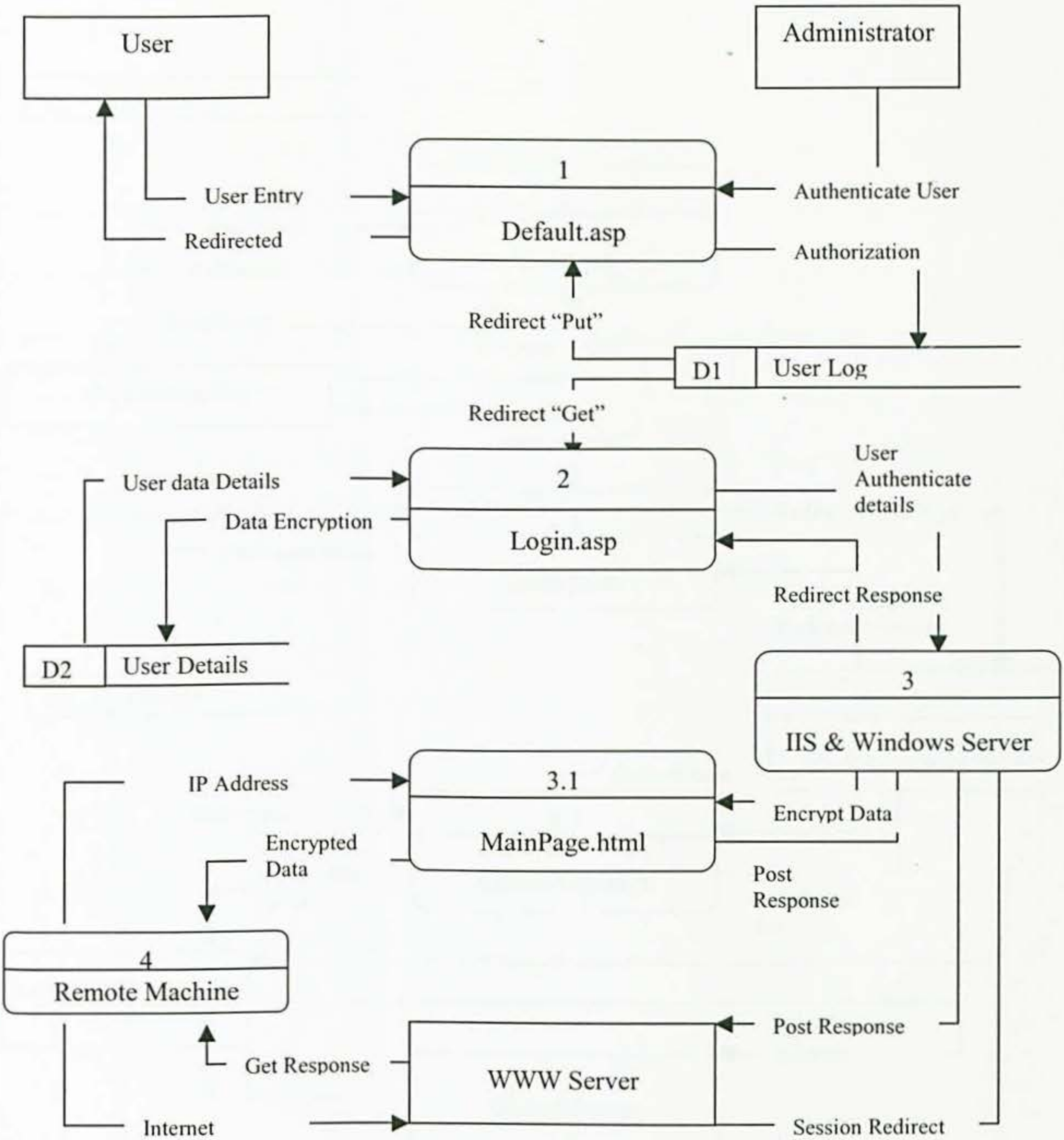


Figure 4.3: Diagram 1

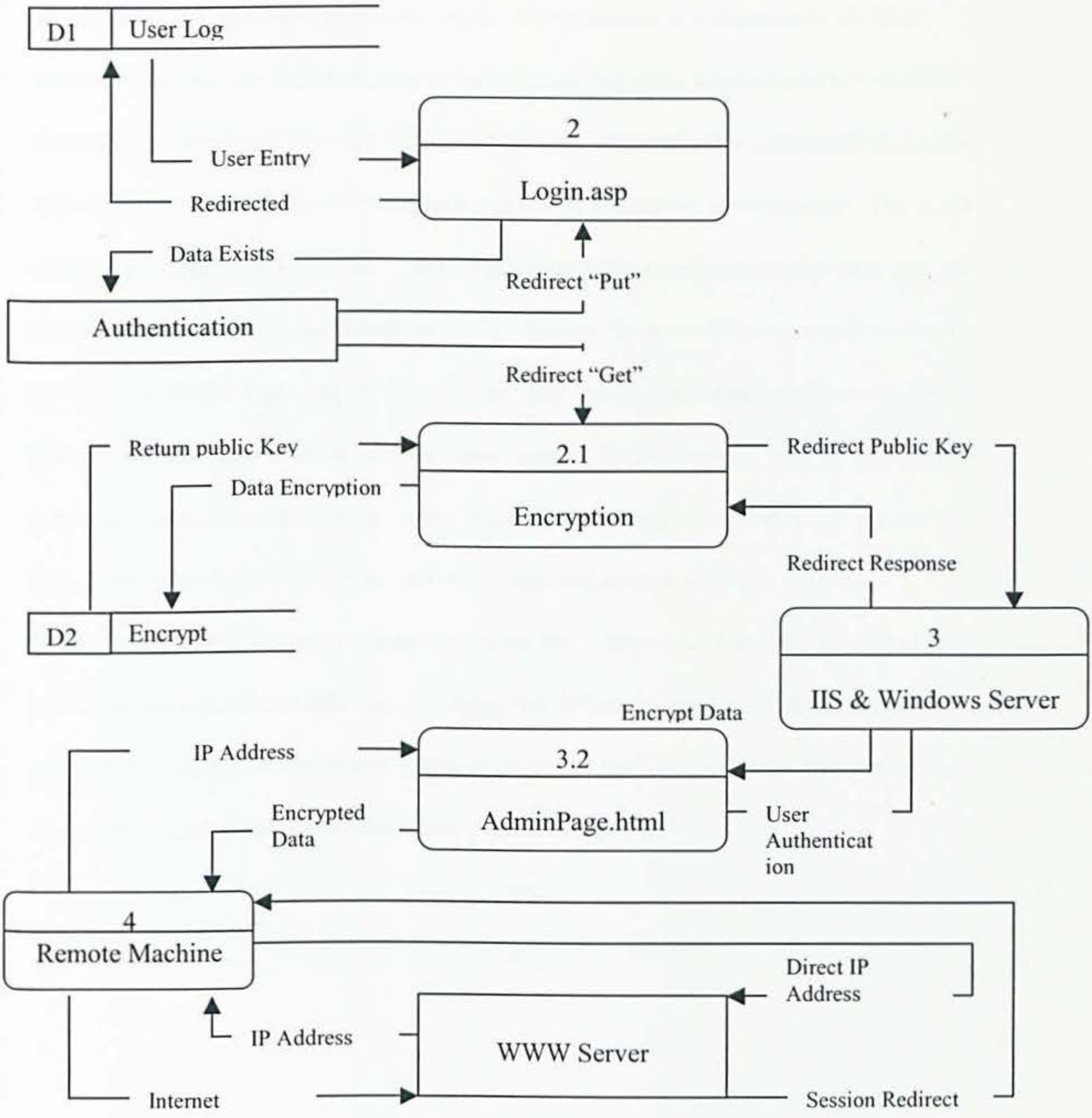


Figure 4.4: Diagram 2

The figure 4.4 shows the user entry into the website. When the user access the website first the Default. asp file opens which gives the way for user to login into the system. The Default. asp provides the access to the Administrator to authenticate the user. When the user has entered the system first the user log from windows server database checks the existence of the user then gives the authorization. The authorization for the user access depends on the authentication given by the server administrator. The login .asp is a program of ASP client, which is programmed to authenticate the user and user password and send the user details to the IIS Server. Now the IIS server will redirect the entry of user to login. asp where the user gets into the network permanently. Then the user has all rights to enter into the main page of the network.

Any communication made on the main page will be encrypted on behalf of the user and connects to the remote server for other user end communication purpose.

In figure 4.4 it shows when the user enters the main page, user has the rights to check with administrator about other user's status too. When the communication process is enabled the encrypted data is decrypted again to the end user terminal, this forms the strong security for data communication over the network.

4.3 User Interface Diagram

4.3.1 Prototype Illustrator

The website is developed using the Microsoft FrontPage 2000. The resulting illustrator artwork is displayed below here the menu items change the content showed hereby redirecting the hidden frame.

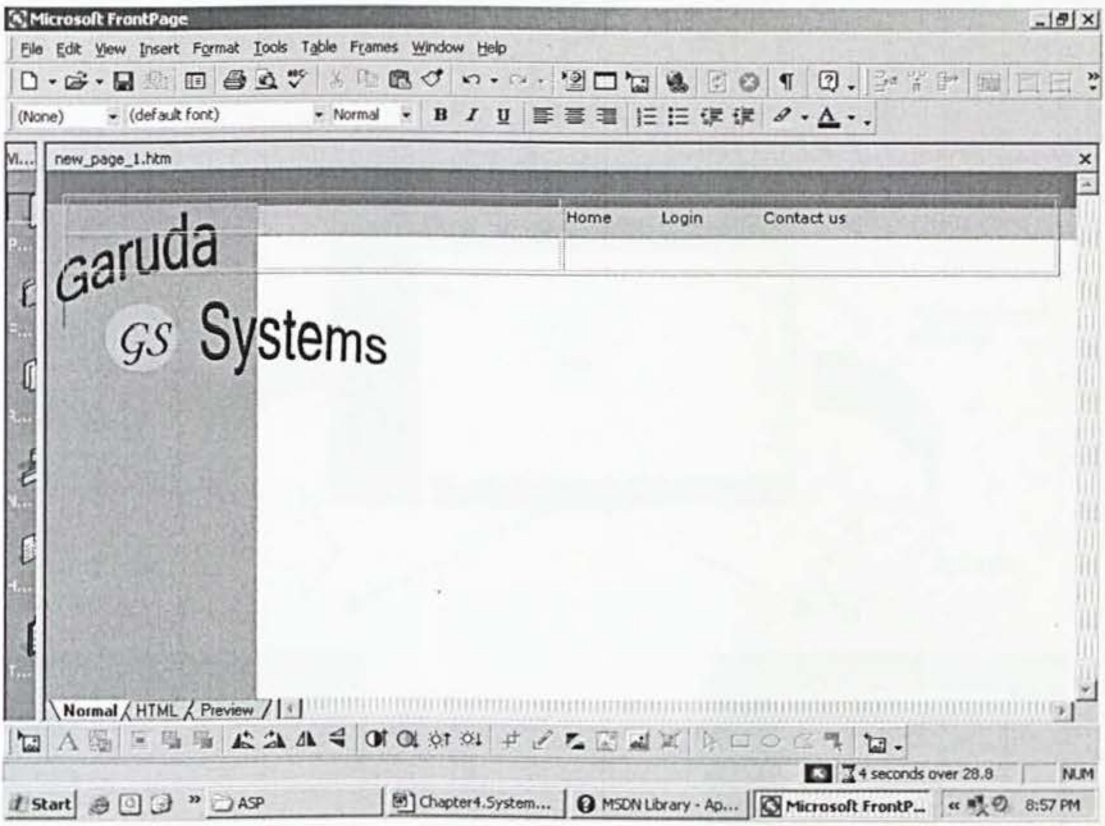


Figure 4.5: Prototype

4.3.2 Web Site Architecture

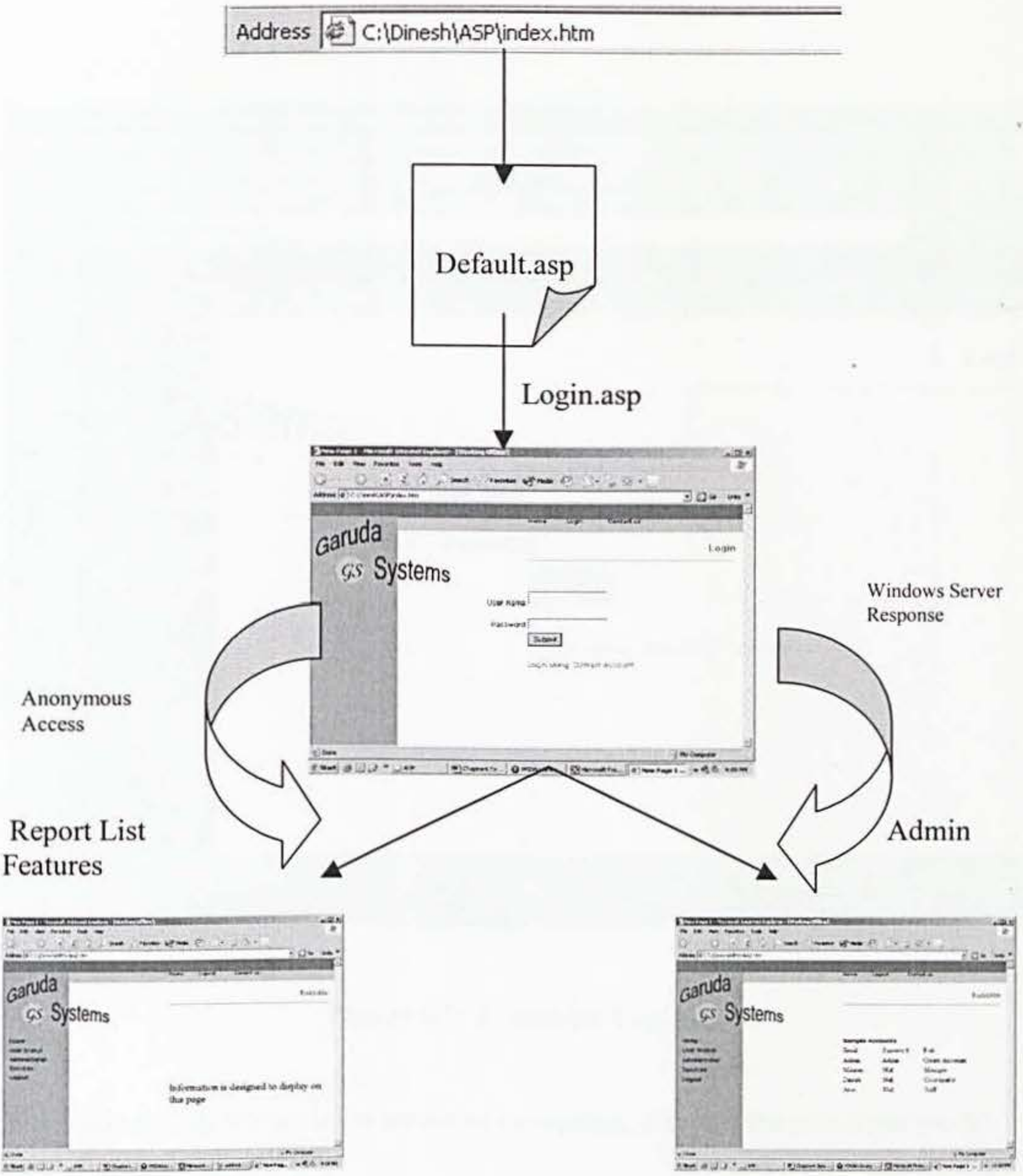


Figure 4.6: Illustrates The Garuda Web Site Navigation and Architecture.

The figure 4.6 gives a graphical design view for the user to understand the real view of the website operation. When authenticated user enters the system the user has the rights to see the report list and the admin features based on the authorization given by the administrator.

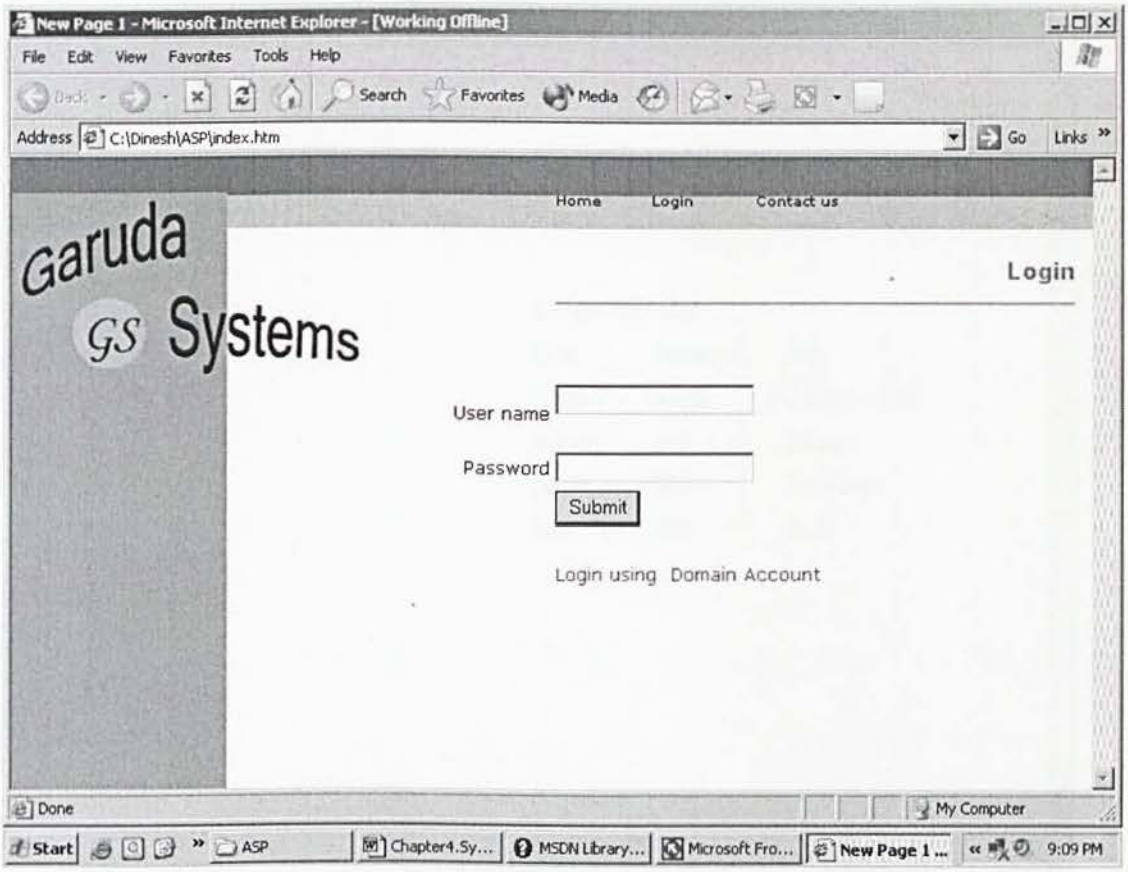
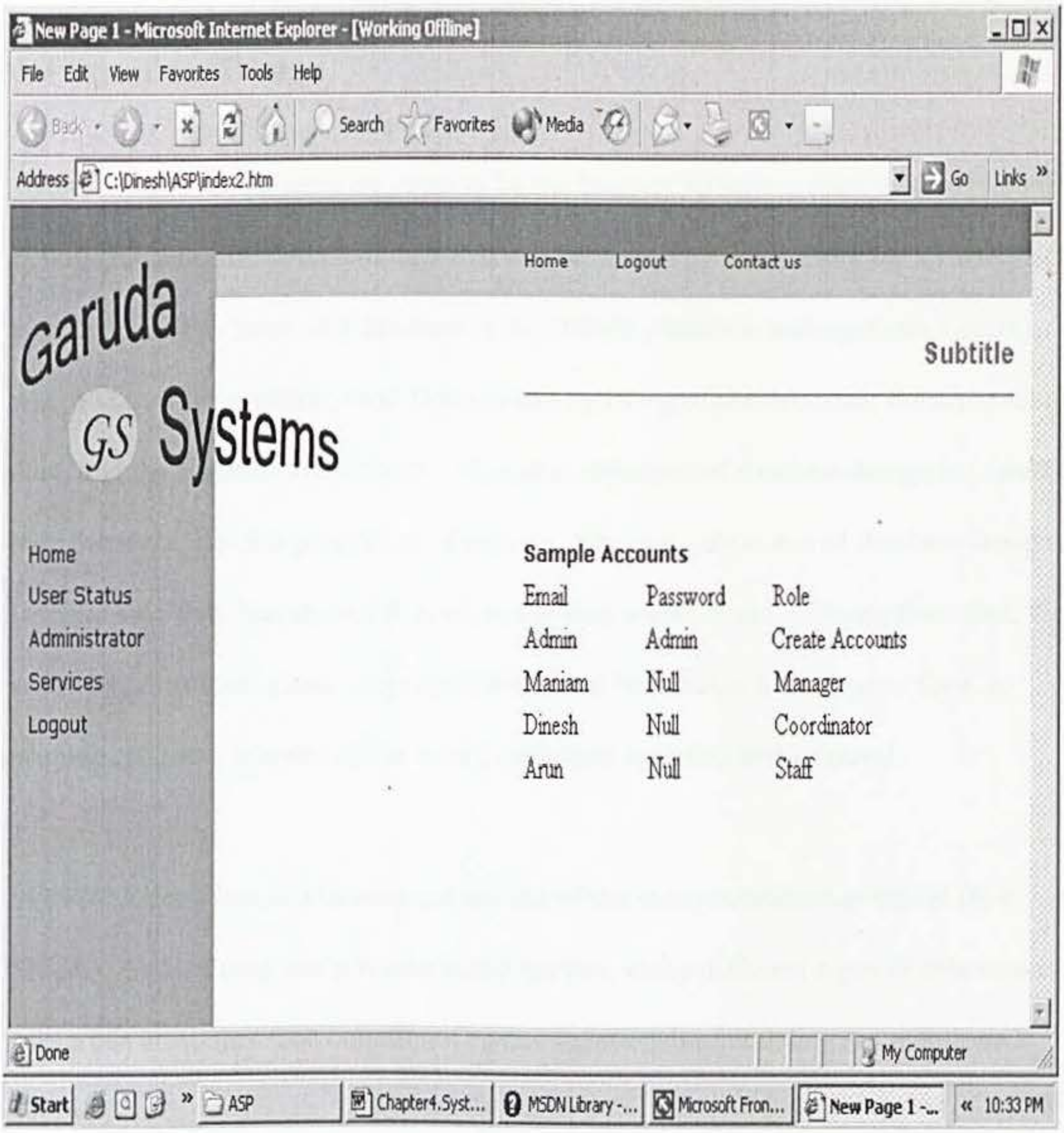

New Page 1 - Microsoft Internet Explorer - [Working Offline]
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\Dinesh\ASP\index.htm
Go Links
Home Login Contact us
Garuda
GS Systems
Login
User name
Password
Submit
Login using Domain Account
Done
My Computer
Start ASP Chapter4.Sy... MSDN Library... Microsoft Fro... New Page 1 ... 9:09 PM

Figure 4.7: Prototype Login.asp

The figure 4.7 shows the login screen of the system, it is just the prototype model showing the pre-designed layout.



Address C:\Dinesh\ASP\index2.htm

Sample Accounts

Email	Password	Role
Admin	Admin	Create Accounts
Maniam	Null	Manager
Dinesh	Null	Coordinator
Arun	Null	Staff

Figure 4.8: Prototype Admin Features

The figure 4.8 shows the user access to the administrator's information of the system; it is just the prototype model showing the pre-designed layout.

4.4 Database Design

Data Storage is considered by some to be the heart of an information system (Kendell, 1996). It is a central source of data that can be shared by many users for a variety of applications. The heart of a database is the DBMS (database management system), which allows the creation, modification and updating of the database; the retrieval of data; and the generation of reports. The main objective of database design is to make sure that data; and the generation of reports. The main objective of database design is to make sure that data is available when the user wants to use it. Apart from that, the accuracy, consistency and integrity of data must be assured from time to time, to provide efficient data storage as well as efficient updating and retrieval.

In 1976, Peter Chen had introduced the use of the entity-relationship model (E-R Model). An E-R diagram contains many entities, many different types of relations and numerous attributes. The benefits of Entity Relationship modeling are mentioned below:

- Databases need to be designed and entity relationship (ER) modeling is an aid to design. An ER model is a graphical representation of the system and is a high-level conceptual data model.
- Supports a user's perception of data and is independent of the particular DBMS and hardware platform.

4.4.1 Database name: GarudaDB

The database has one master table, which has one primary key field, which is related to other four tables as shown in the diagram below.

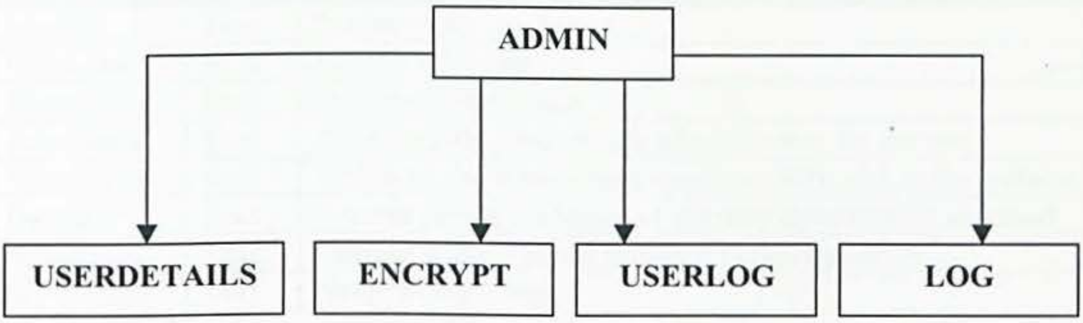


Figure 4.9: Database Design Flow

The database structure in figure 4.9 just shows the simple relation between the master and the transaction tables. In the current system the database is designed to have one master table ADMIN and other four transaction tables USERDETAILS, ENCRYPT, USERLOG and LOG. From the above structure it is clearly shown that the ADMIN table has one primary key field, which related to the similar secondary key field in all the tables as explained in the data dictionary explained in detail.

4.4.2 Data Dictionary

Data Dictionary of metadata can be described as descriptions of the database structure as contents. Data dictionary defines the field, field type and descriptions of each table.

In this system, a database is created defined as GarudaDB.

Database Name: GarudaDB

Table Name: Admin

Table 4.1: Table of Admin

Field	Type	Description
Loginid	Text	Primary Key
Username	Text	Name of the user
Password	Text	Password for the user
Accessrole	Text	Access rights given by the administrator for the user
Restriction	Text	Which all the areas where user has restriction in the website
Userip	Text	Internet protocol address of the user dynamically allocated
Serverip	Text	Internet address of the standard company server
Servername	Text	Name of the server

The above table has the information of user Login ID , the User Name, Password and their Access Role, type of Restriction given to them, the table also records the information of IP Address of the server and user for statistical purpose. The above table is exclusively meant for the Administrator to have pre allocated restriction on the user. The purpose of this table is that each time the user enters the system the above table is verified for user existence check and the user is allowed only when the user exist in the database.

Table Name: Userlog

Table 4.2: Table of Userlog

Field	Type	Description
Loginid	Text	Secondary Key
Username	Text	Name of the user
Password	Text	Password of the user
Hitcount	Text	Number of times user has entered

The above table records the information of the user and how many times the user entered into the system just for security reasons.

Table Name: Log

Table 4.3: Table of Log

Field	Type	Description
Loginid	Text	Secondary Key
Timein	Text	Time user has entered into website
Timeout	Text	Time user has logged out website
Date	Text	Date on which user has accessed

The above table records the information of the user namely the Time user has entered IN into the system and the time user checks OUT of the system, date of entry, all these fields in the table helps the administrator to have complete statistics of the user and amount of usage into the system.

Table Name: Encrypt

Table 4.4: Table of Encrypt

Field	Type	Description
Loginid	Text	Secondary Key
Password	Text	Encrypted password
Keyused	Text	Type of key used for encryption

The above table is just used to record the encrypted information just for security reasons. This table also records the type of key used for encryption process.

Table Name: Userdetails

Table 4.5: Table of Userdetails

Field	Type	Description
Loginid	Text	Secondary Key
Name	Text	Name of the user
Designation	Text	Designation of the user
Companyadd	Text	Company address
CPostcode	Text	Postcode for the company
CState	Text	Which state the company belongs
CTelephone	Text	Company Contact number
CHandphone	Text	Company Contact number
CEmail	Text	Company Email address of the user
Homeadd	Text	Home address
HPostcode	Text	Home postcode
HGState	Text	Where the user lives
HTelephone	Text	Home contact number
HHandphone	Text	Home contact number
Email	Text	Personnel email of the user

The above table describes the complete user details this table helps to have a look on the user personnel data completely so that the administrator can have list of registered users and their contact address.

4.4.3 Relationship – The Class Diagram

There are three types of established inter-table relationships, which are one: one (1:1), One: many (1:N) and many: many (M: N). The diagrammatic representation of the system's database relationship is illustrated in the class diagram in figure 4.11.

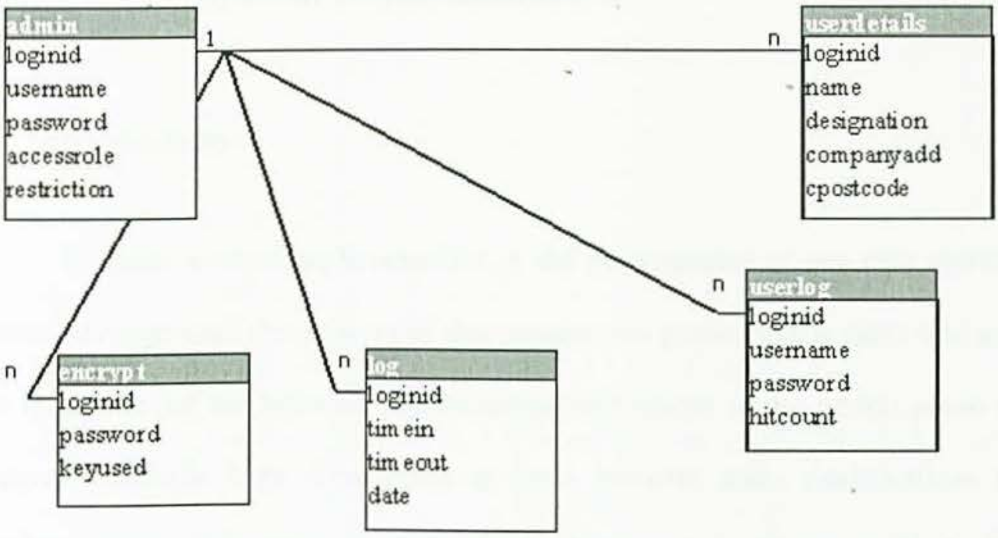


Figure 4.10: Relationship - Class Diagram

The above figure 4.10 shows the relationship – class diagram that gives the complete view of the system database operation and their relations. The above figure is primarily targeted to show the master and the transaction table's details.

Chapter 5 – System Implementation

5.1 Introduction

In short, system implementation is the development of any new system and follows through until the delivery of that system into production. It takes into account the translation of the software representation and layout in the design phase into a computer-readable form. This phase at times involves some modifications to the previous design. This comes as to no surprise because it is near impossible to stick to the requirements in the design phase for changes are inevitable.

One of the interesting aspects taken into consideration in implementation is the quality of data. By definition, it is the measurement of how consistently pertinent that data is within certain preset limits. Effectively coded data facilitates accurate data entry by cutting down on the quality of data required. This in turn reduces the chance of inputting incorrect data.

5.2 Implementation Principles

As a web application, "ASP Authentication Using IP Address" follows the following principles during the implementation phase:

- **Works continuously**

Just as a web's development process often is continuous, so is a web's implementation. Because of this, web-implementation procedures should be designed with process orientation, allowing for replication, improvement, and reliability in file management and coding techniques.

- **Separation of tasks.**

All web-development processes involve separating the processes of web development so that decisions about specific HTML structure are allowed to be made "just in time". It is during the implementation phase that decisions about the web are made based on tolerances and instructions provided.

- **Involves layering of detail**

It is most efficient to generic web components or software that works with templates for creating HTML or ASP files. This same template idea can be used to design file systems as well as page layout to achieve the goals of a consistent web.

5.3 Development Environment

There are certain changes made during the implementation of the system, which contradicts the proposed design. These changes cover the areas of tools used; web development tool as well as minor changes in system modules. These changes took to effect as the condition and simulation of the system changed and justification will be duly given below.

5.3.1 Web Development Tools

One of the changes made is the decision to use Macromedia Dreamweaver 6.0 MX instead of the proposed Macromedia Dreamweaver 4.0. Even though the former is slightly difficult to utilize, it was choosed as implementation was underway. This is because there are more features in Dreamweaver such as embedding interactive images and media (swf, .fla format files) compared to previous Dreamweaver versions, which is relatively bland. It took quite a bit of getting used to, but it was a definite betterment.

5.3.2 System Modules

The system modules include sets of HTML documents and ASP pages, each and very ASP page is programmed to get redirection from the HTML pages and other interlinked ASP pages.

The home page allows the user to login to access the webpage. The initial reason for user login makes a clear demonstration for ASP Authentication using IP Address.

When the user logs into the webpage, user is checked for the basic password authentication and user is also simultaneous check for his IP address, the form is validated in such a way when the user assigned IP from the local machine database matches the IP address of the remote machine. The entire operation of user login is checked by a click of a button, instead of waiting for the administrator to check their results, which causes delay. All of these came after the decision of integrating both HTML and ASP modules into one concise module.

The changes were made on the ASP pages to show the working model and their output results. Typically the output stays behind the screen when webpage is uploaded over Internet so that the user cannot view the background operations for security reasons.

As the project working model has to be demonstrated the output results for the working concepts or ASP Authentication, IP Authentication, Encryption and Decryption are shown when the user logs into the webpage.

5.4 Code Documentation

Code documentation is a set of written descriptions that explain to a reader what the programs do and how they do it. Internal documentation is descriptive material written directly within the code. All other documentation is external documentation. Code documentation begins with the selection of identifier (variables and labels) names, continues with connecting and ends with the organization of the program.

5.4.1 Internal Documentation

Internal documentation contains information directed at the person who will be reading the source code of the program and might possibly enhance the application. Thus in ASP Authentication summary code is provided to identify the program and describes its data structures, algorithms and control flow. A statement of purpose dictating the function of the module and descriptive comments are embedded within the body of the source code to describe processing functions.

5.4.2 External Documentation

External documentation is intended to read by those who never look at the actual source code of a program. External documentation gives the programmer a chance to explain more broadly than might be reasonable within the program's comments.

In ASP Authentication the external documentation consist of a documents displayed on profiles, Security and Encryption Demo webpage links, which explains to the user how to use ASP Authentication Using IP Addressing. The use of screen shots and pictures gives the user a clearer picture of what is used for how to use it.

5.5 Input/ Output

The style of input, output adheres to the following guidelines:

- Validate all input data
- Keep the input format simple
- Label interactive input requests, specifying available choices or bounding values.
- Keeps the input format uniform
- Label all output and design all reports

5.6. Database Implementation

There is a single database for the system named as "garudaDB" which consists of two masters and 3 transaction tables to store the related information's from the website through the ADO Objects (ActiveX Data Objects) programmed through server using ASP. The database is located on the server where MS Access 2002 is installed. Any data creation and data retrieval is accessed where the IIS 5.0 server software via ADO objects with direct connection to access 2002 database. Here the database is shared for read only mode and update mode from IIS 5.0 through which client access the database through IIS 5.0.

5.7 Program Implementation

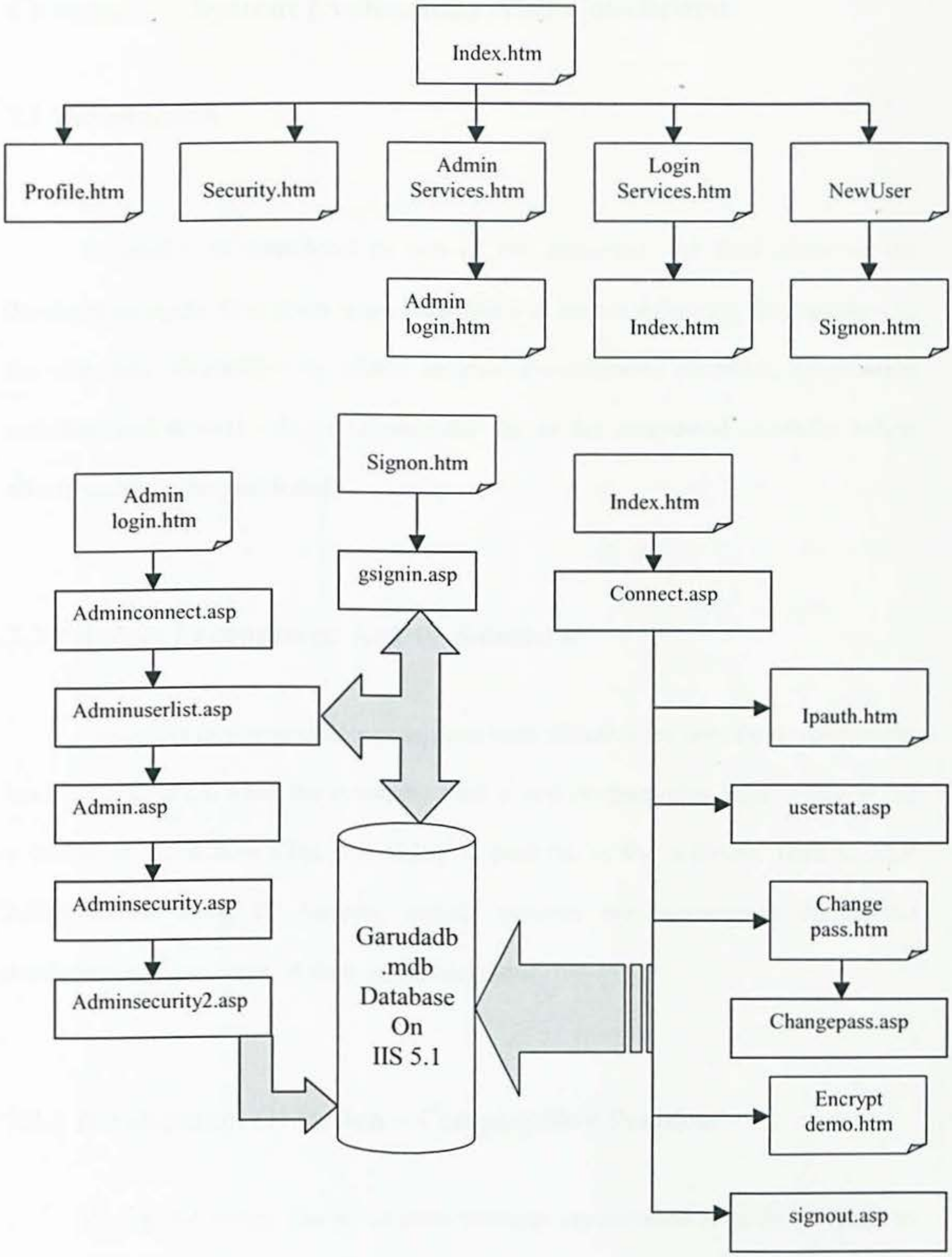


Figure 5.1 Program Flow and Structure

Chapter 7 – System Evaluations And Conclusion

7.1 Introduction

Evaluation is considered as one of the important and final phase of the development cycle. It is given more importance to before delivering final product to the end user. Evaluation is related to user environment, attributes, information priorities and several other concerns that are to be considered carefully before effectiveness can be concluded.

7.2 Problem Encountered And Its Solutions

Normally any programming language with effective features for developments lands up in problem when the developer finds a new programming logic, which is not a feature of the software but can ability to perform by the software. Here in ASP Authentication using IP Address, certain problem are encountered during the development phase, some of them are given below

7.2.1 Development Of Design – Compatibility Problem

During the design phase the main problem encountered is to fit the page to different Internet explorers based on versions, and different explorer platform like Netscape navigator, so here it is clearly solving the versioning problem first. The versioning problem is solved by developing a simple Java Script program, which can auto, adjust the width or makes the program compatible to run on any type of browser.

7.2.2 Development Of Programming Logic – Logic Problem

This phase is normally the second phase after the development, this phase the most critical problem is to suite the design to the logic that is developed, first the question arises whether to suite the logic based on the design or to suite the design based on the logic, it is typically the easy way to take or to perform the job more effectively is to use the design according to the logic that is developed, this solution makes the end user to realize the technological advancement make user easier to use the website.

7.2.3 Development Cycle

The third critical phase of development is the development cycle, after the complete logic is developed the development cycle or the programming cycle starts to encounter many problems during execution of the program. At each and every stage the testing is performed using the top to bottom approach. Each time the program is revised to suite the best performance level, the ASP Authentication using IP Address is tuned up to 3 times of revision in codings to suite the application best operated by the end user, and this has made the end user easy to access and enter into the website.

Chapter 6 – System Testing

6.1 Introduction

Why system testing? Mainly because it is a necessity in system development as defined in system methodology as well as SDLC (System Development Life Cycle). Testing is crucial because it is a process that is focused on finding faults in order to create a fault-free application. Because the goal of software testing is to discover faults, a test is considered successful only when a fault is discovered or a failure occurs during the testing process. Testing does not prove the absence of errors but only shows that software errors are present. Fault identification is the process of determining what faults caused the failure, and fault correction is the process of making subsequent changes to the system to rid (unfortunately, not as a whole) of the faults and defects.

6.2 Input Validation

The input data in the program is validated and checked at each and every stage of ASP redirection in order to ensure the smooth flow of the program. There are several possible ways to ensure data validity. Among them are as follows.

6.2.1 Test For Invalid Data

Testing can be branched into two processes; verification and validation. Verification refers to the set of activities, which indicates a successful execution of a targeted function or sub program conducted by the software. Validation, on the other hand, refers to a set of activities conducted to ensure that the software built is traceable and tailored to the customer's wants and needs.

This validity test examines data to see if there are any missing items. Additionally the record should include the key data that distinguishes one record from all others and the function code telling the computers what to do with the data. This is very important to authorize access to admin module in this system. If a wrong user name or password has been entered the system will terminate the login process and an error "Invalid name or password" message will appear.

```
<%
usr=request.form("t1")
'request the form object namely the "user" value and stores in declared variable "usr"
pwd=request.form("t2")
'request the form object namely the "password" value and stores variable "pwd"

flag=0
rs.movefirst 'moving the recordset of the table "userlog" to first
do until rs.eof 'performing Do Loop to validate the table
g_CryptThis = rs(1) 'encrypted password from the database is assigned

if rs(0) = usr and DeCrypt(g_CryptThis) = pwd then
'converting the encrypted password and then comparing with the original string
lip = request.ServerVariables("REMOTE_ADDR")
'request for seeking the local machine address
if rs(2) = lip then
'comparing the local machine address to the Remote machine assigned address
'it the above comparision is ok then the output written using "response.Write(string)"

response.Write("<font size='2' color='#ff0000' face='Verdana, Arial, Helvetica, sans-
serif'>This Message is for DEMO purpose<hr>")

response.write("<br><font size='2' color='#003399' face='Verdana, Arial, Helvetica,
sans-serif'>This Shows the 'Password' Decryption Demo along with IP Authentication,
The IP Statistics are Given Below.")

response.Write("<br>The Encrypted 'Password' from the database / admin alloted
'Password' is : <font size='2' color='#c00dff' face='Verdana, Arial, Helvetica, sans-
serif'>"& g_CryptThis &".</font>")

response.Write("<br>The Decrypted 'Password' using asp decrypting method extracts
the hidden 'Password' : <font size='2' color='#c00dff' face='Verdana, Arial, Helvetica,
sans-serif'>"& DeCrypt(g_CryptThis)&".</font><hr>")
```

```
flag = 1
exit do ' exiting the Do Loop
end if
end if
rs.movenext
' moving the recordset to next position to revalidate the cursor pointing to next data
loop

%>
```

6.2.2 Test For Range Or Reasonableness

These validity tests are really a common -sense measure of input that answers the question of whether data fall within an acceptable range or whether they are reasonable. Within predetermined parameters. A reasonableness test ascertains whether the item makes sense for the transaction.

The reasonableness test was written into the coding of ASP Authentication using IP Addressing to ensure that users did not accidentally key in wrong data. For example, during the course of implementation and testing, a simple test was coded in order to ensure that the data entered by the user were entered in the right sequence.

6.2.3 Test For Comparison With Stored Data

Another test for validating input data that is used in comparison of the data to the data that is already stored in database. As an example, when requesting for a word term of an user the input of users will be compared to the data in database the system will first perform a search to see if the code number exists in the database.

6.3 Testing Techniques

During the testing phase, the test object can be viewed as either a **black box** or a **white box**. These are the two most commonly used techniques. If the test object is viewed from the outside as a closed box or black box whose contents are unknown, the testing involves feeding input to the closed box and taking note of what output is produced. Black box testing, in this case focuses on the functional requirements of the software. Among the errors black box testing locates are:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external data base access
- Performance errors
- Initialization and termination errors.

When the test object is viewed as a white box,(also known as *glass-box testing)* the structure of the test object is used to test in different ways; test case design method that uses the control structure of the procedural design to derive test cases. For example, test cases can be devised to execute all the statement or all the control paths within the components to be sure that the test object is working properly. With white box testing, the following can be achieved:

- Guarantee that all independent paths within a module have been exercised at least once;
- Exercise all logical decisions on their True and False sides;
- Execute all loops at their boundaries and within their operational bounds;
- Exercise internal data structures to assure their validity.

ASP Authentication was tested with a combination of black box and white box testing. Black box testing and white box testing needs to be mutually exclusive. Any test philosophy can lie somewhere in between. The choice of test philosophy depends on many factors, including:

6.3.1 The Number Of Possible Logical Paths.

Since ASP Authentication using IP Address is a relatively medium sized implementation, there is limited number of logical paths in the program coding. Therefore white box testing is a feasible option when testing the application. If there were a large number of logical paths, the component would be difficult to test thoroughly.

6.3.2 The Nature Of Input Data

The input data involved in ASP Authentication using IP Address is fairly generic, that is there are few instances where the data that can be entered is limited.

6.3.3 The Amount Of Computation Involved

The computation involved in ASP Authentication using IP Address is kept to a minimum.

6.4 Unit testing

Unit testing verifies the correctness of the smallest unit of the application - the module. The tests are conducted in order to uncover errors within the boundary of the module. Unit testing includes the testing of the listed areas. Figure 6.1 included below showing its flow:

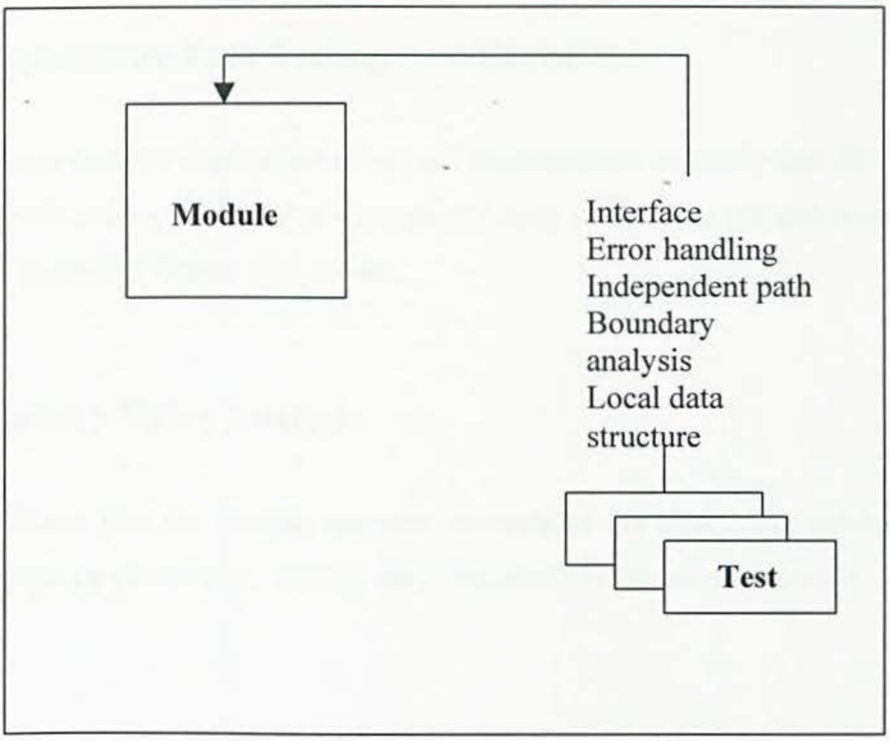


Figure 6.1 Unit Testing

6.4.1 Interface

The module interface in ASP Authentication Using IP Address is tested to ensure that the data is received from another part of the program or from the user is correct. Interface testing also tests to ensure that the data that flows out of the module into the other parts of the application is correct.

6.4.2 Local Data Structure

The unit testing conducted on ASP Authentication Using IP Address also serves to examine the data structure of the module to ensure that the data stored in the module temporarily maintains its integrity during the execution of the module.

6.4.3 Independence Path Testing And Execution

Ensures that the control structures are implemented correctly and all branches of the module's coding in ASP Authentication Using IP Address are executed at least once to see whether it works, as it should.

6.4.4 Boundary Value Analysis

It ensures that the module operates properly at the boundaries established to limit or restrict its processing. This is very important to ensure the security in admin module.

6.4.5 Error Handling

Testing of the error handling capabilities of the modules in ASP Authentication Using IP Address. The module should be able to detect and recover from any errors that occur during its execution. For example the system should display an error message when the user does not fill in a blank or inputs the wrong answer for a question.

6.5 Integration Testing

Once unit testing has been completed all the individual units are combined into a working system. The integration is planned and coordinated so that when an error occurs, there is some idea of where the error could have occurred. Integration testing is a systematic approach for constructing the program structure while simultaneously conducting tests to uncover errors by the interfacing.

Integration testing for ASP Authentication Using IP Address was done using the top-down style of coding approach. Components at the highest level of the hierarchy is tested individually first and then all the individually tested components are jointly tested. This approach was done repeatedly until all the components are tested.

This system allows faults to be discovered in each unit before combining them, which facilitates the tracking of faults when they occur.

6.6 Environment Setup Testing

The environment setup testing was aimed to ensure the integrated environment has been implemented correctly and performs its function correctly. This part of testing is akin to the unit testing for the individual modules.

Among the most important aspects of the environment setup testing was to test for Internet Information Server (IIS) Local host testing and security settings. This section was tested to ensure that the basic security was not breached. In ASP Authentication, the databases are the main components that need to be tested. The search for the definition of a word, attribute of a word and an example of the sentence that uses the word all involves the use of database as all this information is retrieved from the database.

6.7 Debugging

Debugging is performed as a consequence of successful testing. When a test case uncovers an error, debugging is the process of attempting to match symptom with the cause and if successful, leads to the correction of the error. The debugging approach employed was causes elimination. Data related to the error occurrence was organized to isolate potential causes. A list of all possible causes was developed and tests conducted to eliminated each. If initial tests indicate that a particular cause hypothesis promise, the data are refined in an attempt to isolate the error.

Chapter 7 – System Evaluations And Conclusion

7.1 Introduction

Evaluation is considered as one of the important and final phase of the development cycle. It is given more importance to before delivering final product to the end user. Evaluation is related to user environment, attributes, information priorities and several other concerns that are to be considered carefully before effectiveness can be concluded.

7.2 Problem Encountered And Its Solutions

Normally any programming language with effective features for developments lands up in problem when the developer finds a new programming logic, which is not a feature of the software but can ability to perform by the software. Here in ASP Authentication using IP Address, certain problem are encountered during the development phase, some of them are given below

7.2.1 Development Of Design – Compatibility Problem

During the design phase the main problem encountered is to fit the page to different Internet explorers based on versions, and different explorer platform like Netscape navigator, so here it is clearly solving the versioning problem first. The versioning problem is solved by developing a simple Java Script program, which can auto, adjust the width or makes the program compatible to run on any type of browser.

7.2.2 Development Of Programming Logic – Logic Problem

This phase is normally the second phase after the development, this phase the most critical problem is to suite the design to the logic that is developed, first the question arises whether to suite the logic based on the design or to suite the design based on the logic, it is typically the easy way to take or to perform the job more effectively is to use the design according to the logic that is developed, this solution makes the end user to realize the technological advancement make user easier to use the website.

7.2.3 Development Cycle

The third critical phase of development is the development cycle, after the complete logic is developed the development cycle or the programming cycle starts to encounter many problems during execution of the program. At each and every stage the testing is performed using the top to bottom approach. Each time the program is revised to suite the best performance level, the ASP Authentication using IP Address is tuned up to 3 times of revision in codings to suite the application best operated by the end user, and this has made the end user easy to access and enter into the website.

7.2.4 Final Stage

At this stage final testing and implementation of the programs are done. At this stage each program individually developed are integrated to one single project and complied at one shot and tested at last for final stage presentation, here the major problem occurred in the IIS 5.1 IP Addressing and IP Accessing, once a particular system was used as server the IIS has to configured to suite the environment of the network, in case of Windows XP the use of administrative tools helps the developers to fit the ASP pages to host on IIS and set the lost host page on the server. This makes the end user to access the multiple copies of websites from the server. This feature also gives the end user to access same website 'N' number of times in a single PC and also in the network.

7.3 Future Enhancements And System Constraints

The future enhancements is one of the important feature focused in this project, this project gives many ideas on programming as well as the in the design and also in project design, in many ways this project can be enhanced. In future this project can be enhanced mainly on the logic of program or on the technology, even the IP Accessing can be enhanced through the improved version of IIS and IP address can also be improved through top-bottom approach in programming. More over new technologies like blowfish algorithms can be used for enhancing the encryption and decryption logics. When user over WAN (Wide Area Network) the major enhancements can be done for converting the existing database to MS-SQL Server or to the Oracle platform.

7.3.1 IP Spoofing

The major enhancements can be done for solving the new problems coming up today that is IP spoofing; IP spoofing is one of the most common forms of on-line camouflage. In IP spoofing, an attacker gains unauthorized access to a computer or a network by making it appear that a malicious message has come from a trusted machine by "spoofing" the IP address of that machine.

7.3.2 Spoofing Attacks

There are a few variations on the types of attacks that successfully employ IP spoofing. Although some are relatively dated, others are very pertinent to current security concerns.

7.3.3 Non-Blind Spoofing

This type of attack takes place when the attacker is on the same subnet as the victim. The sequence and acknowledgement numbers can be sniffed, eliminating the potential difficulty of calculating them accurately. The biggest threat of spoofing in this instance would be session hijacking.

7.3.4 Blind Spoofing

This is a more sophisticated attack, because the sequence and acknowledgement numbers are unreachable. In order to circumvent this, several packets are sent to the target machine in order to sample sequence numbers. While not the case today, machines in the past used basic techniques for generating sequence numbers. It was relatively easy to discover the exact formula by studying packets and TCP sessions.

7.3.5 Man In The Middle Attack

Both types of spoofing are forms of a common security violation known as a man in the middle (MITM) attack. In these attacks, a malicious party intercepts a legitimate communication between two friendly parties. The malicious host then controls the flow of communication and can eliminate or alter the information sent by one of the original participants without the knowledge of either the original sender or the recipient.

7.4 Misconceptions Of IP Spoofing

While some of the attacks described above are a bit outdated, such as session hijacking for host-based authentication services, IP spoofing is still prevalent in network scanning and probes, as well as denial of service floods. However, the technique does not allow for anonymous Internet access, which is a common misconception for those unfamiliar with the practice. Any sort of spoofing beyond simple floods is relatively advanced and used in very specific instances such as evasion and connection hijacking.

7.5 Defending Against Spoofing

There are a few precautions that can be taken to limit IP spoofing risks on your network, such as:

7.5.1 Filtering At The Router

Implementing ingress and egress filtering on your border routers is a great place to start your spoofing defensing. You will need to implement an ACL (access control list) that blocks private IP addresses on your downstream interface. Additionally, this interface should not accept addresses with your internal range as the source, as this is a common spoofing technique used to circumvent firewalls. On the upstream interface, you should restrict source addresses outside of your valid range, which will prevent someone on your network from sending spoofed traffic to the Internet.

7.5.2 Encryption And Authentication

Implementing encryption and authentication will also reduce spoofing threats. Both of these features are included in Ipv6, which will eliminate current spoofing threats. Additionally, you should eliminate all host-based authentication measures, which are sometimes common for machines on the same subnet. Ensure that the proper authentication measures are in place and carried out over a secure (encrypted) channel.

IP Spoofing is a problem without an easy solution, since it's inherent to the design of the TCP/IP suite. Understanding how and why spoofing attacks are used, combined with a few simple prevention methods, can help protect your network from these malicious cloaking and cracking techniques.

7.6 Conclusion

ASP Authentication using IP addressing, this project was done aimed at providing the one of the best ways of security for the end user, the concept behind the website can be applied for various website developers and server side programming developers to create new security concepts which can reach end user flexibility and make them to use the website services at full extent. The concept and demonstration for the website www.garuda.com demonstrates the end user is capable of using the website more effectively and concept embedding of encryption logics makes the end user to ensure the usage of his or her transactions made are safe through the internet are effective. The two end security logs implemented in this website makes clear that security concepts can also be developed and controlled through software's.

APPENDIX:

ASP Authentication Using IP Address

By Alain Trottier

Introduction

This article explains how to control application access by validating the user's login and password against a database. Once validated, the IP address of the machine they are using is checked at the top of every page. What a given user can or can't do (that is, the security levels) is now handled easily.

> **Note:**
>
> Whitfield Diffie and Martin Hellman invented the first public key exchange protocol at Stanford in 1976. Public key cryptography was launched with this seminal work. The Diffie-Hellman cryptosystem is currently used with the Digital Signature Standard (DSS). Diffie-Hellman is based on the "discrete logarithm" hard problem.

Overview

The security method presented here comes close to the simplicity of cookie security without the headaches of losing a cookie - or cookies being turned off by the user. The one thing that the user browser will always give you is its IP address. Otherwise, where would the Web server send its response? You verify the person with an ID and password against a predefined list in ASP or a database. I recommend verifying against a database and I demonstrate how to do this with code.

An IP address-based schema presented here assumes the IP address for a given machine remains constant. For most networks it is constant. For a few networks IP addressing is dynamic where the address will change for a machine with each start up, and sometimes even during operation. The security schema presented won't work if the IP address changes while the user machine is on; however, this is rare (i.e., I think AOL proxy servers do this for dial-in clients).

Once you verify that the user has permission to access the application, you save the IP address to the database with a time stamp. You have accomplished the key step - creating a traceable identifier (think of the IP address as a security badge) for a validated user that is accessible throughout the ASP application. Remember that date stamp? You can set an expiration date or a time span on a given traceable identifier. Check each requester's IP address using:

```
        Request.ServerVariables("REMOTE_HOST")
```

Now check that IP address' time stamp. If it has lapsed beyond the limit you set (i.e., midnight) send him to the logon page for revalidation using Response.Redirect ("logon.asp"). I allow user validation to last only one day. Midnight is a fair compromise between necessary periodic user revalidation and user annoyance with high logon repetition. Below, notice that at midnight the difference between the datestamp associated with an IP address (actually all logons for that day), and the computer system time becomes 1 or TRUE for the IF-THEN test:

```
If Rs.EOF OR DATEDIFF("d",rs("DateTime"),NOW()) Then
  Response.Redirect ("logon.asp")
ELSE
  Authorization = TRUE
END IF
```

Analysis

How do you use the Data Accessibility Based Valuation Theory and the Data Stress Law to help analyze your security? You reduce the stress on your system's sensitive data by locking out everyone but a chosen few. This action reduces the number of times a given data set is accessed, reducing the possibility of change in data (read corruption). You can do this at the database level or at the ASP level. While I recommend doing it at the database level, it is easier to do it at the ASP level.

The Accessibility Based Valuation Theory uncovers the need to get data in the system only once and let many people see it many times: a one-to-many relationship. Use this theory to design your ASP application so that a data set is entered into the system once. The permission to do this should be kept to a short list of users. Enforcing this policy is where the security code comes in. Then, allow many folk to see it. For example, a development bug system I built allowed few people to enter bugs, fewer still to change data already in the system, but everyone to view and search the bug database.

Encryption Overview

The distance between the user and the Web server is a security risk. Someone in between the two can snoop. When the user submits her ID-password it must travel along the network wires. If a bad person intercepts the keys then he could use them to gain unauthorized access. The network loop represents a security hole – a snoop hole.

The Secure Sockets Layer (SSL) is a technique where the browser and Web server encrypt information before sending it, and decrypt messages that they receive. Anyone snooping between them sees gibberish. There are also third party tools that do the same. The browser mashes and bashes data so no one trashes it. Only the Web server can unmash and unbash it.

How can you, an ASP guru, stop snoopers? Encryption! While I advocate SSL or third party security solutions, you can build your own encryption process with ASP. You secure data with mirroring encryption/decryption functions across the browser and Web server. Like SSL, you encrypt form information from the browser before sending form contents to your ASP pages that decrypt the data. In reverse, your ASP page could encrypt information before sending it to the browser and have the browser decrypt it with a script that decrypts the data upon load. While you need to use a strong routine, the following is a simple example (popular XOR algorithm) of what you could have in ASP. Passing a data and password pair the first time encrypts the data. Passing the encrypted data and same password pair the second time decrypts it. The elegant, although weak, XOR method has made the routine a popular starter for many to get acquainted with the encryption concept.

An ASP Castle - A Metaphor

Security is a block wall. A select group easily gets beyond it, but no one else. It is a way to let good people in and bad people out. The following ASP castle is sturdy, but, alas, the walls are scalable.

You build the walls from the username, password and IP address blocks. An ASP guard verifies a unique- hopefully secret - user name and password combination. If verified, the ASP guard gives your guest a security badge with the IP address on it and allows passage. Once in, your guest can get into some rooms depending on his privileges, but not the others. Anyone caught without an approved badge is banished in a water clock second.

Some ASP builders use session blocks, but in that case, the ASP guards will not come to work unless you feed them cookies. Other builders use NT security boulders. They are strong, but once in the castle the guests can go into every room, even if you want to restrict their access to only certain rooms. This is because all of the NT security happens at the gate. Once past the gate, the ASP guards can't distinguish one guest from another.

Method Details

Here is the code to implement ASP security on your application without using NT security or session objects. Simply, you check the user ID and password combination against a table in the database. If there is a match, then you update the IP address and date stamp associated with that ID/Password combination. Now grant access to the user. From now on, check the requesting IP address against the database. If there is a match, grant access. If there is no match then redirect him to the logon page. The only other detail you need to observe is the date stamp attached to the IP address. If the request is made after midnight, force the user to logon again. They will have to do this once a day to preserve some validation. You can change this to any time period with the "datediff" function in the security check section of the code.

Security Check Include at Top of Each ASP Page Requiring Security

```
'include the following at top of each ASP page requiring security
<!-- #include virtual="/SCRIPTS/BUG/SECURITY.INC" -->
```

Security Include File

```
strFileName = Trim(Server.MapPath ("\scripts\bug")) &
"\userinfo.mdb"
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" &
strFileName & ";"
Set rs = Server.CreateObject("ADODB.Recordset")

LOGIP=Request.ServerVariables("REMOTE_HOST") SQL =
"SELECT * FROM tUsers " &_
  "WHERE (((tUsers.UserIP) = "' & LOGIP & "')); " rs.Open sql, conn,
3, 3

If Rs.EOF OR DATEDIFF("d",rs("DateTime"),NOW()) Then
Response.Redirect ("logon.asp")
ELSE
Authorization = TRUE
userid = rs("userid")
```

```
END IF
```

Logon Page Code

```
' Form field check
id = Request.Form("Userid")
pwd = Request.Form("Password")

'if forms are filled then proceed
if len(id)>0 and len(pwd)>0 then
 '---- CursorTypeEnum Value ----
Const adOpenKeyset = 1
'---- LockTypeEnum Values ----
Const adLockOptimistic = 3 '
---- CommandTypeEnum Values ----
Const adCmdText = &H0001

'get path of database
strFileName = Trim(Server.MapPath ("\scripts\bug")) &
"\userinfo.mdb"

'setup database connection
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" &
strFileName & ";"
Set rs = Server.CreateObject("ADODB.Recordset")
SQL = "SELECT * FROM tUsers WHERE (((tUsers.Password) = '" &
Pwd
SQL = SQL & "') AND ((tUsers.Userid) = '" & id & "')); "
rs.Open sql, Conn, adOpenKeyset, adLockOptimistic, adCmdText

' If the user is valid the recordset will have a record
' otherwise call up the form again.
If NOT Rs.EOF THEN
IF request("enter") = "Log Out" THEN
rs("DateTime")=now()-1
On Error Resume Next
rs.Update
msg = "You are logged out. You will need to log "
msg = msg & " in again to update or add data."
response.write msg
%>
 <META HTTP-EQUIV="Refresh"
CONTENT="3;URL=http:default.asp">
<%
response.end
```

```
END IF
NewPassword1 = request("NewPassword1")
NewPassword2 = request("NewPassword2")
IF LEN(NewPassword1) > 0 AND LEN(NewPassword2) > 0 THEN
IF LCASE(NewPassword1) = LCASE(NewPassword2) THEN
rs("Password") = LCASE(NewPassword1)
Password = LCASE(NewPassword1)
END IF
END IF

NewUserName1 = request("NewUserName1")
NewUserName2 = request("NewUserName2")
 IF LEN(NewUserName1) > 0 AND LEN(NewUserName2) > 0 THEN
IF LCASE(NewUserName1) = LCASE(NewUserName2) THEN
rs("Userid") = LCASE(NewUserName1)
Userid = LCASE(NewUserName1)
END IF
END IF
LOGIP=Request.ServerVariables("REMOTE_HOST")
rs("Permission")=TRUE rs("UserIP")=LOGIP
rs("DateTime")=now()
On Error Resume Next
rs.Update
IF LEN(Userid) > 0 OR LEN(Password) > 0 THEN
IF LEN(Password) > 0 THEN
msg = "Your <B>Password</B> has been changed to: <B>"
msg = msg & Password & "</B><P>
" response.write msg
END IF
IF LEN(Userid) > 0 THEN
msg = "Your <B>User Name</B> has been changed to: <B>"
msg = msg & Userid & "</B><P>"
response.write msg
END IF
%>
 <META HTTP-EQUIV="Refresh"
CONTENT="3;URL=http:default.asp">
<%
response.end
END IF
response.redirect "default.asp"
END IF
rs.close 'close the records
conn.close 'close the connection
END IF
```

```
%>
<html>
 <head> <title> Bug Manager Logon</title></head>
<BODY BGCOLOR="#31639C" TEXT="ffffff" LINK="#00008B"
ALINK="#FF0000" VLINK="#8B008B">
<center><h1> Bug Manager</h1>
<h3>Registered Users</h3>
<form name="passwordform" action="/scripts/bug/logon.asp"
method="POST">
<table width=415 border=0 cellspacing=0 cellpadding=0>
<tr>
 <td>User Name:</td>
<td></td>
<td>Password:</td>
<td align="left">
<input type="submit" name="enter" value="Log In"></td>
</tr>
 <tr>     <td><input type="text" name="userid" size="16"
maxlength="16"></td>
<td></td>
<td><input type="password" name="password" size="16"
maxlength="16"></td>
<td align="left"><input type="submit" name="enter" value="Log
Out"></td>
</tr>

</table>
</CENTER>
<HR>
<P>
You may change your login name or password at any time.
After correctly providing your current user name and password above,
type in your new user name or password below.
<CENTER>
<table width=415 border=0 cellspacing=0 cellpadding=0>
<tr>
<td>New Password:</td>
<td> </td>
 <td>Confirm New Password:</td>
<td></td>
</tr>

<tr>
<td><input type="password" name="NewPassword1" size="16"
maxlength="16"></td>
```

```
<td></td>
<td><input type="password" name="NewPassword2" size="16"
maxlength="16"></td>
<td align="center" valign="top"></td>
</tr>
<tr>
 <td>New User Name:</td>
<td></td>
<td>Confirm New User Name:</td>
<td></td>
</tr>

<tr>
<td><input type="text" name="NewUserName1" size="16"
maxlength="16"></td>
<td></td>
<td><input type="text" name="NewUserName2" size="16"
maxlength="16"></td>
 <td align="center" valign="top"></td>
</tr>
</table>
</form>
</center>
</body>
</html>
```

Authorization Level Controls Program Flow

```
<%' checks password in session
IF NOT Authorization THEN response.redirect
"/scripts/bug/logon.asp" %>
'continue if authorized and branch based on user id
' retrieved in security.inc include
SELECT CASE userid
CASE "jtronp"
'perform jtronp specific function
CASE "kholland"
'perform kholland specific function
CASE "lroberts"
'perform lroberts specific function
CASE ELSE
'perform everyone else specific function
END SELECT
%>
```

MS Access ID - Password Table Definition

```
<TABLE border=2 width=600>
  <CAPTION align=top><BIG><A name=accessidtabledef>MS
Access ID - Password
  Table Definition </BIG></A></CAPTION>
  <TBODY>
  <TR >
    <TD><B>Field Name</B></TD>
    <TD><B>Data Type</B></TD>
    <TD><B>Note</B></TD></TR>
  <TR >
    <TD>ID</TD>
    <TD>AutoNumber</TD>
    <TD>Always a good idea</TD>
  <TR bgColor=#ffffe0>
    <TD>Userid</TD>
    <TD>Text (10)</TD>
    <TD>Restrict size in HTML Input tag too</TD></TR>
  <TR bgColor=#ffffe0>
    <TD>Password</TD>
    <TD>Text (10)</TD>
    <TD>Restrict size in HTML Input tag too</TD></TR>
  <TR bgColor=#ffffe0>
    <TD>DateTime</TD>
    <TD>Date/Time</TD>
    <TD>Updated every logon</TD></TR>
  <TR bgColor=#ffffe0>
    <TD>UserIP</TD>
    <TD>Text (20)</TD>
    <TD>Most important data chunk</TD></TR>
  <TR bgColor=#ffffe0>
    <TD>SecurityLevel</TD>
    <TD>Number</TD>
    <TD>Use this to enforce level-feature policy</TD></TR>
  <TR bgColor=#ffffe0>
    <TD>Permission</TD>
    <TD>Yes/No</TD>
    <TD>Provides a way to turn some
away</TD></TR></TBODY></TABLE></CENTER>
<P>
<CENTER>
<TABLE border=2 width=600>
  <CAPTION align=top><BIG><A name=usertracertabledef>MS
Access User Tracer
  Table Definition </BIG></A></CAPTION>
  <TBODY>
  <TR bgColor=#ffbb00>
    <TD><B>Field Name</B></TD>
    <TD><B>Data Type</B></TD>
    <TD><B>Note</B></TD></TR>
  <TR bgColor=#ffffe0>
    <TD>ID</TD>
    <TD>AutoNumber</TD>
    <TD>Always a good idea</TD>
  <TR bgColor=#ffffe0>
    <TD>Userid</TD>
    <TD>Text (10)</TD>
    <TD>Optional: helps in dynamic IP allocation
networks</TD></TR>
  <TR bgColor=#ffffe0>
    <TD>DateTime</TD>
    <TD>Date/Time</TD>
    <TD>Set default to Now()</TD></TR>
  <TR bgColor=#ffffe0>
    <TD>UserIP</TD>
    <TD>Text (20)</TD>
    <TD>Most important data chunk</TD></TR>
  <TR bgColor=#ffffe0>
    <TD>URLFrom</TD>
    <TD>Text (100)</TD>
    <TD><% = Request.ServerVariables("HTTP_REFERER")
%></TD></TR>
  <TR bgColor=#ffffe0>
    <TD>URLCurrent</TD>
    <TD>Text (100)</TD>
    <TD><% = Request.ServerVariables("PATH_INFO")
%></TD></TR></TBODY></TABLE></CENTER>
```

Equivalent SQL Server Script

```
/* Microsoft SQL Server - Scripting */
/* Database: SecureUser */
/* Creation Date 31/9/98 10:22:09 */
PRINT '--------------------------------'
PRINT 'Starting execution of [place script name here]'
PRINT '--------------------------------'
PRINT "
GO

/* Recommend creating database with SQL Enterprise Manager */
/* If you want to do it with SQL use the top portion of the script */
```

```
print "
print getdate()
print "
print 'Creating database securitydb'
print "
GO
use master GO

set nocount on

if not exists (select name from master.dbo.sysdatabases where name =
'securitydb')
create database securitydb on [Data device] = 6 log on [Log device] = 2
GO

use securitydb
GO
/* Check that we're in securitydb... */
if (db_name() <> 'securitydb')
   raiserror('A problem was encountered accessing securitydb. Script
terminating.',
  20, 127) with log
GO

checkpoint
GO
dump tran securitydb with no_log
GO
print "
print 'Created database securitydb'
print "
GO

use track_master

/****** Object: Table dbo.Users Script Date: 31/9/98 10:22:09
******/
if exists (select * from sysobjects where id = object_id('dbo.Users')
  and sysstat & 0xf = 3)
drop table dbo.Users
GO

/****** Object: Table dbo.UserPath Script Date: 31/9/98 10:22:09
******/
if exists (select * from sysobjects where id = object_id('dbo.UserPath')
and sysstat & 0xf = 3) drop table dbo.UserPath
```

```
GO

/**********************************************************
*******************/
/* This table designed to identify users with */
/* their ID and password allowing an upgrade to the newer ones. */
/**********************************************************
******************/

/****** Object: Table dbo.Users Script Date: 31/9/98 10:22:09
******/
CREATE TABLE dbo.Users (
ID smallint IDENTITY (1, 1) PRIMARY KEY CLUSTERED, /*
Unique record identifier */
Userid char (10) NOT NULL , /* Aliase of user */
Password char (10) NOT NULL , /* Password of user */
UserIP char (20) NOT NULL, /* IP address of user machine */
SecurityLevel int NULL , /* Allows more robust access policy */
Permission tinyint NULL DEFAULT (1), /* This allows you to prevent
access wholesale */
DateTime datetime NOT NULL, /* Time user logged on */
Note char (150) NULL /* often a good idea */
)
GO

ALTER TABLE
Users
ADD
CONSTRAINT CK_UserIP CHECK (UserIP LIKE
  '[0-9][0-9][0-9].[0-9][0-9][0-9].[0-9][0-9][0-9].[0-9][0-9][0-9]'
  )

print ''
print 'created CK_UserIP CHECK'
GO
ALTER TABLE Users
ADD
DEFAULT getdate() FOR DateTime

print ''
print 'Added DEFAULT getdate() FOR DateTime'
GO

/*
** Permission Check:
** The value can be:
```

```
**
** Permission status
** ====== ========
** 0 inactive
** 1 active */

/*********** Example UserID Constraint
**************************/
/* CONSTRAINT CK_Userid CHECK (Userid LIKE                    */
/* '[A-Z,0-9][A-Z,0-9][A-Z,0-9][A-Z,0-9][A-Z,0-9][A-Z,0-9]') */
/********************************************************
*******/

/********** Example Insertion Statement *******************/

/* insert Users (Userid, Password, UserIP) */
/* values('atrottier', 'mysecret', '209.232.078.004') */
/********************************************************
*/

/****** Object: Table dbo.UserPath Script Date: 31/9/98 10:22:09
******/
CREATE TABLE dbo.UserPath (
ID smallint
IDENTITY (1, 1)
PRIMARY KEY CLUSTERED,
UserIP char (20) NOT NULL,
URLFrom char (100) NOT NULL ,
URLCurrent char (100) NULL,
DateTime datetime NOT NULL,
)
GO

print ''
print 'created table UserPath'
GO

ALTER TABLE UserPath
ADD
DEFAULT getdate() FOR DateTime

print ''
print 'Added DEFAULT getdate() FOR DateTime'
GO

/****** Can find out how many users logged on with: ******/
```

```
/* select Number_users = count(*) */
/* from Users */
/****************************************************/
/* Can play with expiration date with these two statements: */
/* select @dayofweek = (datepart(weekday,@now)+@firstday)%7 */
/* select @monthyear = substring(convert(varchar(12),getdate()),1,12)
*/

GRANT SELECT , INSERT , DELETE , UPDATE ON Users TO
public
GO

GRANT SELECT , INSERT , DELETE , UPDATE ON UserPath TO
public
GO

CHECKPOINT /* optional, but I do it to clean up loose ends */
GO

/* all done, so leave now */
print " select 'Current Date: ' = getdate()
print "
print 'Script Completed!'
print "
GO
```

User Activity Tracker

You can record the movements of your users. In the above, the security server side include file code tracks the IP, page from which the user came and the current page. There are several approaches you can take. Keep it simple. The second table (UserPath) acts as an event log. Each event is actually a page hop. Each time the user hits a page, add a record of the hit to the table. You may decide to dump it to storage periodically if space becomes a problem. Use this table for usage analysis and to help clarify matters regarding who did what that inevitably occur.

Advantages of an IP based Security Schema

1. You can set conditionals for what is displayed based on the user logon. If a user tries to go circumvent the security they will be sent to the logon page automatically.
2. Logons last until a preset date-time (i.e., midnight) and last as long as the user machine keeps its IP address. (Some networks role out IP addresses dynamically from a predefined pool, while others are permanently assigned.)
3. The user can change his user id or password from the logon page at any time.
4. The user can log on from any machine that has access to our intranet including other locations. There are no license issues.
5. The user can log out at any time. After logging out he will have to log on, again, from any machine.
6. Any user can search the bug database without logging on.
7. Security is browser independent.
8. The user can completely close and restart his browser or use a different browser on the same machine and still retain the ability to update and add bugs without needing to log on again.
9. The user can log on from different machines at will, but the last machine the user logs on from is the only valid login.

REFERENCES

- A Primer on Encryption and PKI, "*Network computing*", Asian Edition, May 2003
- Michael P. Levy, "*ASP and Web Session Management", Microsoft Source Developer Network*, 1997.
- Dougles E.Cormez, "*Internetworking With TCP/IP Principles, Protocols and Architecture* ", Prentice Hall International Inc.
- Steve Steinke, "*Network Tutorial, A Complete Introduction To Network*", Jan 2000.
- Stuart Lee, Paul Groves, Chris Stephens, Susan Armitage, "*Online-Teaching : Tools and Projects*", Oxford University
- Douglas Hamilton , "*Web Authoring Tools*", Prentice Hall International Inc.
- Ed Tittel & Bill Brodgen, "*Discover Java*", IDG Books Worldwide, 1997
- Brooks Jr. F.P. , "*The Mythical Man-Month. Essays on Software Engineering*",Anniversary Edition, Reading, MA: Addision- Wesley, 1995
- Jeffrey L. Whitton, Lonnie D. Bently abd Kevin C. Diffman, "*System Analysis and Design Methods*" , Mcgraw Hill, 5 Edition.
- Francois Fluckiger, "*Understanding Networked Multimedia: Applications and Technology*", Prentice Hall.

Website References:

- "Concepts of TCP/IP", http://www.introcomp.co.uk/networking/tcpip.html
- "Concepts of Internet Security", http://www.securityfocus.com/infocus'
- "ASP Authentication Using IP Address",

  http://m-tech.ab.ca/concepts/authentication_server.html
- http://www.bris.ac.uk/is/services/reqister/auth-info/history.html
- http://www.bris.ac.uk/is/services/reqister/auth-info/why-auth.html
- http://www.bris.ac.uk/is/services/reqister/auth-info/authentication.html
- www.asp-zone.com/aspfaq.asp
- http://www.metris.com/toolsweuse.html
- www.geogetoen.edu/crossroads/mitmedia.html
- "EdTec 650 Technology Demonstration – Fall 1997"

  http://tcp.cal/1997/9703/9703trai/web/we.html
- "Multimedia authoring Systems FAQ",

  http://www.tiac.net/users/jasiglar/MMAFAQ.html
- http://www.15seconds.com
- http://www.cgvak.com
- http://www.securitymetrics.com
- http://www.hushmail.com
- http://www.aaemail.com
- http://www.fortunecity.com

Thesis References:

- Ng Yin Chern , "Integrated Information System Via WIFI Technology", University Malaya, 2002/2003.
- Zuraini Binti Karia, "Web-Based Internet Security", University Malaya, 2002/2003.