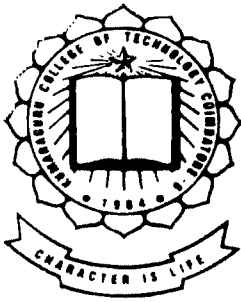


AUTHENTICATION SERVER

*Submitted in partial fulfillment of the requirements
for the award of degree of
Master of Science in
Applied Science – Software Engineering
of Bharathiar University, Coimbatore*



P-925



Submitted by

K.SARAVANAN.

Under the Guidance of

**Mr. Kannan, B.Sc (CT), M.C.A., Lecturer,
Kumaraguru College of Technology,
Chinnavedampatty Post,
Coimbatore 641006.**

Mr. Yeshwanth B.Sc., M.C.A.

**Mr. S.Arokiaswamy,
Diploma in Computer Science & Engg,
Computer Networks & Eng, from BELL LABS, USA,
Computer System Design and Analysis, MIT, USA.**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641006

APRIL 2003

Department of Computer Science and Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY
Coimbatore – 641 006


CERTIFICATE


This is to certify that the project entitled “Authentication Server” has been submitted by Mr. K.Saravanan in partial fulfillment of the award of the degree of Master of Science in Applied Science – Software Engineering of Bharathiar University, Coimbatore during the academic year 2002 – 2003


.....
Guide


..... 29/12/03
Head of the Department

Certified that the candidate was examined by us in the
Project Work Viva Voce Examination held on 5/4/2003
and the University Register Number was 9837S0063


.....
Internal Examiner

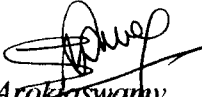

.....
External Examiner

W3 Solutions

No. 35, near 11th cross, 2nd Main, Jayanagar, Mysore 570014. Ph 0821- 561505

This is to certify that **Mr.K.Saravanan** (roll no. 98se23) doing his final year M.Sc. [Software Engineering] at Kumaraguru College of Technology, Coimbatore has been working as a project trainee here at our firm “W3 Solutions ”, Mysore, on the project entitled **Authentication Server**.

He successfully completed the final semester project **Authentication Server**. His conduct during this period (from Dec. 2002 to Feb 2003) was excellent.



Arokiaswamy.
M.D., W3 Solutions

Acknowledgements

ACKNOWLEDGEMENTS

I wish to express my sincere and heart felt gratitude to our Principal Mr. K. K. Padmanahban B.Sc. (Engg), M.Tech, Ph.D. for giving us the needed encouragement in starting of this project and carrying it out successfully.

I'm grateful to our **Prof. Dr. S. Thangasamy**. Ph.D. H.O.D./C.S.E. for helping me when I was in doubtful situations and for the overall guidance.

I also extend my gratitude and thanks to our course coordinator **Asst. Prof. Mrs. S. Devaki** B.E., M.S., for her guidelines and advice in leading my ideas through a proper channel.

I also extend my sincere thanks to my college guide **Mr. Kannan**, B.Sc (CT), M.C.A., who has taken up keen interest in the success of this project.

I also take immense pleasure to thank my project guide at **W3 Solutions**, Mysore **Mr. Yeshwanth** B.Sc., M.C.A. and **Mr. Arokiaswamy**, *Diploma in Computer Science & Engg, Computer Networks & Eng, from BELL LABS, USA, Computer System Design and Analysis, MIT, USA*, for their help and guidance throughout this project and all the staff of **W3 Solutions** for their helping hands.

This is an important moment to remember all my teachers at college who have helped me for the past five years and in the future too, in lighting my future through their caring words.

SYNOPSIS

Today in the shrinking world, technology is growing up like a giant. To keep at that speed the current technologies in use have to be updated periodically, otherwise we will stay behind. Keeping this in mind, this project has come into existence.

“*Authentication Server*” is the title of the project. This is for the firm W3 Solutions, at Mysore. It’s an I.T. firm that does Development of Software, Web Site Development, Portals, E-Commerce Packages, Network Solutions, etc.

This project is developed with the tools namely Java 1.3, Jsdk 2.0, Macromedia Flash 5, Dreamweaver 4, Dreamweaver Ultradev 4 and Adobe Photoshop.

This project of course reduces the time for duplication work for it does not permit duplication of data. Ultimately reducing the storage space of database,

The Authentication Server has 4 modules

- Administration Module
- Client Module
- Network Module
- Interface Module

The *Administrator Module*, is used by the Administrator for all administrative tasks – registering the new user/resource, modifying the user details/ resource details, deleting the user/resources, and so on.

The *Client Module* is used by the clients for making their request to the Authentication Server and to post any messages or information about their account extension. The clients can proceed with their transactions with the server and do their tasks, those are carried out under their root directory.

The Networking Module is for the server and the client interactions using sockets and handshake signaling. Sockets are mainly used for better connection purpose. This module is responsible for the communication throughout the network.

The Interface Module is particularly for the interface of the cross-platforms features and human-computer interface for displaying the interaction screens.

CONTENTS

**ACKNOWLEDGEMENT
SYNOPSIS
ABOUT ORGANISATION**

1. INTRODUCTION	1
1.1 ABOUT THE PROJECT	
1.1.1 PROJECT OVERVIEW AND SUMMARY	
1.1.1.1 ADMINISTRATION MODULE	
1.1.1.1.1 Registration of new clients	
1.1.1.1.2 Make Changes	
1.1.1.1.3 Deletion of clients	
1.1.1.1.4 Adding new resources	
1.1.1.1.5 Removing a resource	
1.1.1.2 CLIENT MODULE	
1.1.1.3 NETWORKING MODULE	
1.1.1.4 INTERFACE MODULE	
1.2 ABOUT ORGANISATION	
2. EXISTING SYSTEM	11
3. PROPOSED SYSTEM	12
4. INPUT DESIGN AND OUTPUT DESIGN	15
5. DATABASE DESIGN	26
5.1 ABOUT MICROSOFT ACCESS	
5.2 ABOUT MYSQL	
5.3 DATABASE DESIGN	
6. SYSTEM ELEMENTS	30
6.1 SYSTEM ELEMENTS – HARDWARE	
6.1.1 Server Configuration	
6.1.2 Clients Configuration	
6.1.3 Modem	
6.1.4 Soft modem	
6.1.5 Operating System	

6.2 SYSTEM ELEMENTS – SOFTWARE

- 6.2.1 Java**
 - 6.2.1.1 Introduction to Java**
 - 6.2.1.2 Java: Security**
 - 6.2.1.3 Java: Portability**
 - 6.2.1.4 Java: Server Side**
 - 6.2.1.5 Java: Servlets**
 - 6.2.1.6 Advantages of Servlets**
- 6.2.2 TCP/IP**
- 6.2.3 Sockets**
- 6.2.4 Servlet Runner**
- 6.2.5 Software used in Development**

7. SYSTEM IMPLEMENTATION AND TESTING	41
7.1 SYSTEM DESIGN	
7.2 CLIENT SERVER MODEL	
7.3 SYSTEM TESTING	
7.4 SYSTEM IMPLEMENTATION	
8. CONCLUSION	49
9. FUTURE ENHANCEMENTS	50
10. REFERENCES	51
11. APPENDIX	52
12. NOTES	68

1. INTRODUCTION

1.1 ABOUT THE PROJECT

Introduction to Authentication Server

Authentication

Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. In private and public computer networks (including the Internet), authentication is commonly done through the use of logon passwords. Knowledge of the password is assumed to guarantee that the user is authentic. Each user registers initially (or is registered by someone else), using an assigned or self-declared password. On each subsequent use, the user must know and use the previously declared password. Logically, authentication precedes authorization (although they may often seem to be combined).

Introduction to Computer Networks

The merging of computers and communications has had a profound influence on the way computer systems are organized. The concept of the “computer center” as a room with large computer to which users bring their work for processing is now totally absolute. The old model of a single computer serving all of the organization’s computational needs has been replaced by one in which a

large number of separate but interconnected computers do the job. These systems are called **computer networks**.

The term “computer networks” means *interconnected* collection of *autonomous* computers. Two computers are said to be interconnected if they are able to exchange information. The connection need not be via copper wire; fiber optics, microwaves, and communication satellites can also be used. If one computer can forcibly start, stop, or control another one, the computers are not autonomous.

There is considerable confusion in the literature between a computer network and a distributed system. The distinction is that, the user of a distributed system is not aware that there are multiple processors; it looks like a virtual uniprocessor. Allocation of jobs to processors and files to disks, movement of files between where they are stored and where they are needed, and all other system functions must be automatic.

With a network, users must explicitly log onto machine, explicitly submit jobs remotely, explicitly move files around and generally handle all the network management personally.

Uses of Computer Networks

❖ **Networks for companies:**

Many organizations have a substantial number of computers in operation, often located far apart. The issue here is **resource sharing**, and the goal is to make all programs, equipment, and especially data available to anyone on the network without regard to the physical location of the resource and the user. The second goal is to provide **high reliability** by having alternative sources of supply. The third goal is **saving money**. Small computers have a much better Price/performance ratio than large ones. The last goal is setting up a computer network has little to do with technology at all. A computer network can provide a powerful communication medium.

❖ **Networks for People:**

Starting in the 1990s, computer networks began to start delivering services to private individuals at home. Below are the more exciting ones that are starting to happen:

1. Access to remote information.
2. Person –to-person communication.
3. Interactive entertainment.

1.1.1 PROJECT OVERVIEW AND SUMMARY

An Authentication Server is a system, which provides Authentication services to other systems on a network. The classical example of this is a bank server that exchanges the data between the user and the information from the service provider. The bank server is basically designed to receive the Id or Credit card and return the information to the provider about the validity and the information so that the transaction holds goods.

Authentication is any process by which a system verifies the identity of a user who wishes to access it. Since Access Control's 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.

A set of Credentials is information used to verify the identity of a user. Normally a User ID and a Password, together, form a set of Credentials.

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

Access Control is any mechanism by which a system grants or revokes the right to access some data, or perform some action. Normally, a user must first Login to a system, using some Authentication system. Next, the Access Control mechanism controls what operations the user may or may not make by comparing the User ID to an Access Control database.

Access Control systems include:

- File permissions, such as create, read, edit or delete on a file server.
- Program permissions, such as the right to execute a program on an application server.
- Data rights, such as the right to retrieve or update information in a database.

A **Login** is the act, made by a user, of connecting to a system or network service. Usually, a user must enter some Credentials, such as his User ID and Password, in order to successfully Login.

A **User ID** is the code used by a user to identify him when he logs into a system and starts a **Login Session**. It is used by the system to uniquely identify this user. A User ID is one-half of a set of Credentials.

Authentication Server is a server for managing the clients and the resources the request for, in the firm. This system is designed to manage the resources and clients; these two are under the administrator.

There are four major modules they are as follows

Administration Module

Client Module

Networking Module

Interface Module

These four modules serve different purpose in real time. The *W3 Solutions* Staff and *W3 Solutions* officials use the Authentication Server for the usage of necessary resources.

The **Administrator Module**, is used by the Administrator for all administrative tasks – registering the newuser/resource, modifying the user

details/ resource details, deleting the user/resources, granting privileges to the clients, extending the period of usage, denying the clients access on resources, etc.

The **Client Module** is used by the clients for making their request to the Authentication Server and to post any messages or information about their account extension.

The **Networking Module** is for the server and the client interactions using sockets and handshake signaling. Sockets are mainly used for better connection purpose.

The **Interface Module** is particularly for the interface of the cross-platforms and human-computer interface for displaying the interaction screens.

These modules elaborated and explained in a brief manner in the following few pages.

1.1.1.1 ADMINISTRATION MODULE

There are five basic functions in this module they are

- i) Registering the new applied client
- ii) Changing Rights and Privileges
- iii) Deletion of the clients
- iv) Adding new resources
- v) Removing resources

1.1.1.1.1 Registration of the new clients

During Registration of new clients, the clients must submit his details and the resources, which he/she requires, access with the privileges to read, write and execute.

The given new login is checked in the database whether it already resides or not. If yes then it asks for another login name else it accepts the given name.

Login is the Primary Key, since there cannot be two logins with the same name. All the required details are recorded about the resources and the client.

The details recorded are:

- Login (Client login name)
- Password
- Confirm Password
- Name
- Sex
- Resource
- Privileges to the resources

All these details are recorded in the temporary database, and then the administrator views this alone, if the client is satisfactory the client is accepted and transferred to the main database, which is permanent.

If the client is rejected then he is left in the temporary database, he could be available in the rejected list, which can be viewed by anyone.

1.1.1.1.2 Changing rights and privileges

This is a module used to change or modify the registered details of a client and sometimes resource. All the details can be modified but not the Login Name because it is the Primary Key.

Here not only the client details but also the client's account expiry date can be changed or modified.

1.1.1.1.3 Deletion of clients

In this Deletion Module a registered client can be deleted, the result is that all the information regarding that client will be removed from the main database.

1.1.1.1.4 Adding new resources

This is a module used to add new resources to the authentication server. Any number of resources can be added to the authentication server. Only the administrator can add any new resource to the authentication server.

1.1.1.1.5 Removing a resource

This is a module used to remove any available resources from the authentication server. Any number of resources can be removed from the authentication server. Only the administrator can remove any resource from the authentication server.

1.1.1.2 CLIENT MODULE

This module is designed for the clients; they must login with their respective Login name and Password. The login information is first verified whether such a user exists in the database or not. If such a user exists, then it checks for the accuracy and correctness of the password. If the login information is incorrect then the system displays the re-login information showing that the login information is incorrect.

If the login information is correct then it moves to the first page where it shows the root directory under which all that clients' transactions will take place and will be recorded. Here the client can call for the required file and he/she can append, modify, erase the contents of the file if the client has got rights to do so. If the client has a privilege of read, the client can only read that existing file.

1.1.1.3 NETWORK MODULE

This module is responsible for the connection establishment between the client and the server. Here Sockets re used for connection between the client and the server.

1.1.1.4 INTERFCE MODULE

This module is responsible for all the interactive screens between the systems and human. This is basically the GUI (Graphical User Interface) that is handled in this system. Since the development is in Java on Linux platform, there is a vast support for GUI.

1.2 ABOUT ORGANISATION

W3 Solutions provides high quality Web Design and Development, Web Hosting, Website Registration, Search Engine Submission, Software Design and Development, Network Engineering, Office Solutions, Offshore Projects with excellent customer support, great customer support at prices that can't be beat!

Through our commitment, experience, and expertise W3 Solutions has established a business relationship with our customers that will last a lifetime!

W3 Solutions has the e-business experience, expertise and vision to help companies launch their services on the Internet and increase their business opportunities several folds with very less investment.

We utilize standardized and proven methods, applying specialized skills and knowledge to create business value through applied technology.



Existing System

2. EXISTING SYSTEM

The problem of allowing authorized users to access licensed information products from anywhere at anytime is crucial.

Security for most of files, programs, databases or systems is currently provided via either IP filtering, login/password combinations when passing through a local online catalog are also a crucial job. Although each networking system is working individually to provide authentication solutions for particular products, there currently are not comprehensive solutions allowing authorized users to access these resources from anywhere at anytime. In addition, a great deal of duplication is occurring in software and hardware to address these issues. User authentication and authorization have long been issues that have been dealt with by the computing community. However other organizations have begun to develop a more comprehensive networked vision of information access over the Internet it has become clear that flexible new solutions need to be implemented.

Now in the organization there are basically three servers and a few resources. The three servers are *the file server*, *the mail server* and *the print server*. In each server there is a database and an authentication system where each personnel of the firm are registered for access rights on each server. This task holds more duplication of data and more time is wasted on duplication of the data. As for the whole firm all personnel must be registered thrice for access.

3. PROPOSED SYSTEM

An Authentication Server can provide a solution for the existing duplication of data. Here in the proposed system, there is single database and a single authentication system for all the available servers and resources in the firm. By doing so the duplication of data is reduced and the time spent on duplication of data is also reduced. Not only the time but also the storage space and eventually the cost of duplication and storage.

All the clients (firm personnel) are registered in the authentication server with certain privileges and permissions on particular resources / server as decided by the administrator.

Authentication server allows third parties to completely customize the access control performed by the server. It allows the server to delegate its access control policies for portions of its document tree to a separate authentication server. The interface utilizes the TCP/IP socket abstraction, allowing the authentication server to be running on a physically different machine from the server to share a centralized authentication server.

It also allows the authentication server implementation to be in any programming language, which can interact with sockets.

Sites running online subscription magazines, for example, have very complicated access control policies, requiring external database lookups on every request. An authentication server checks whether the request should be satisfied or requires a password etc. The authentication server can do whatever is required to validate the client, for example perform some SQL query into a central database on a mainframe where all of the subscription records are kept. Such a design allows for complete flexibility.

For simple connections between a client and a server, *ServerSocket* and *ClientSocket* is all that you will probably need. *ServerSocket* represents the socket on a server that waits and listens for requests for service from a client. *Socket* represents the endpoints for communication between a server and a client. When a server gets a request for service, it creates a *Socket* for communication with the client and continues to listen for other requests on the *ServerSocket*. The client also creates a *Socket* for communication with the server. The sequence is shown below:

□ **Security**

Security includes authentication- and permissions. Authentication relates to user authentication and involves username and password checking. Authentication of a user may be required in a number of situations, such as when a user tries to access a URL. Permissions relate to what actions may be performed.

□ **Authenticator**

Authenticator has methods for authenticating a network connection. Authenticator a security manager, he checks to see that the security policy permits the network permission. Password Authentication is simply a data holder for a user name and a password.

□ **Permissions**

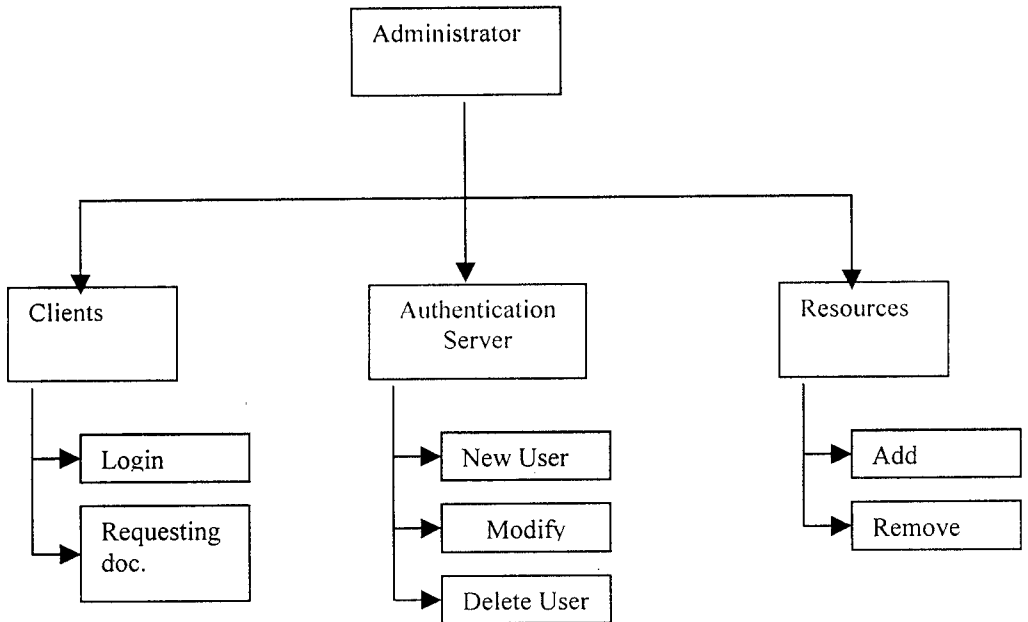
- `SocketPermission`
- `NetworkPermission`

A *socketpermission* consists of a host, with optional port range, and a set of actions that may be performed on that host -connect, bind, listen, and accept. It includes methods to determine if one *socketpermission* is equal to another or implies

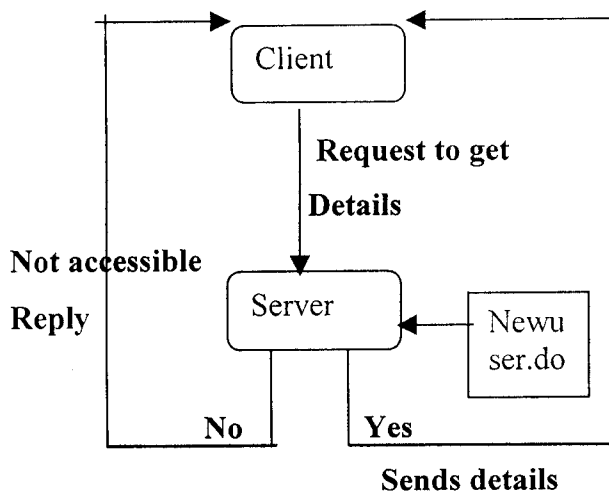
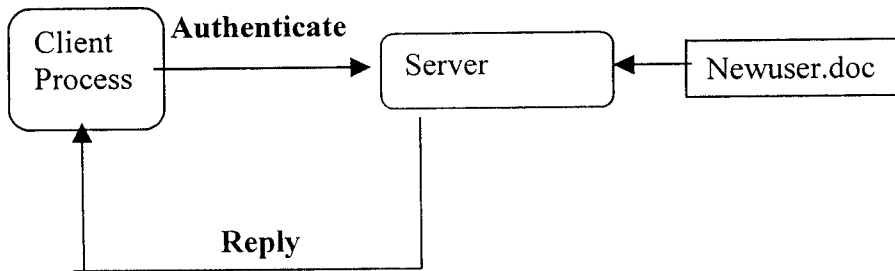
permission. A *socketpermission* is for easy checking if permission exists.
NetPermission is for various named network permissions.

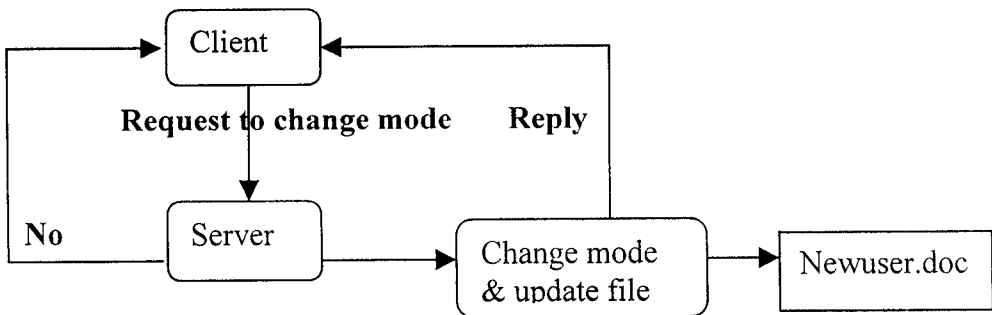
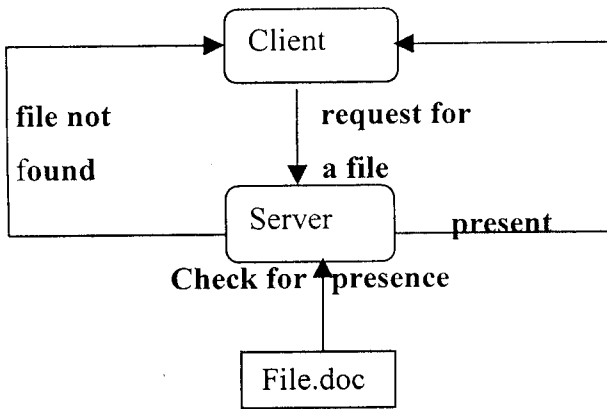
4. INPUT DESIGN AND OUTPUT DESIGN

HIPO CHART FOR AUTHENTICATION SERVER

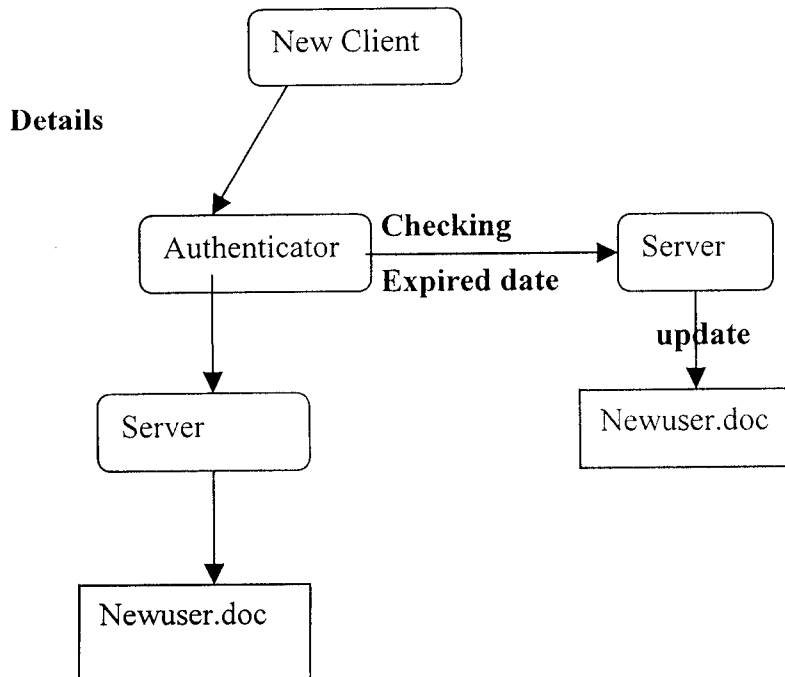


DATAFLOW DIAGRAM

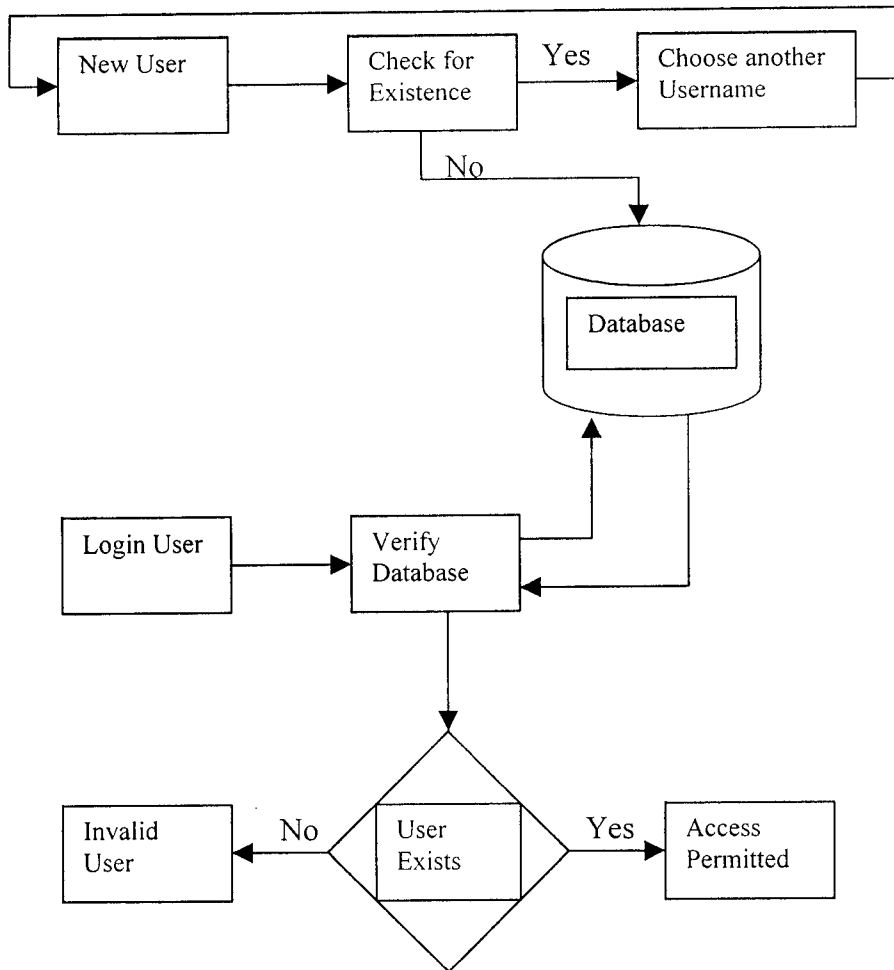




AUTHENTICATOR JOB:

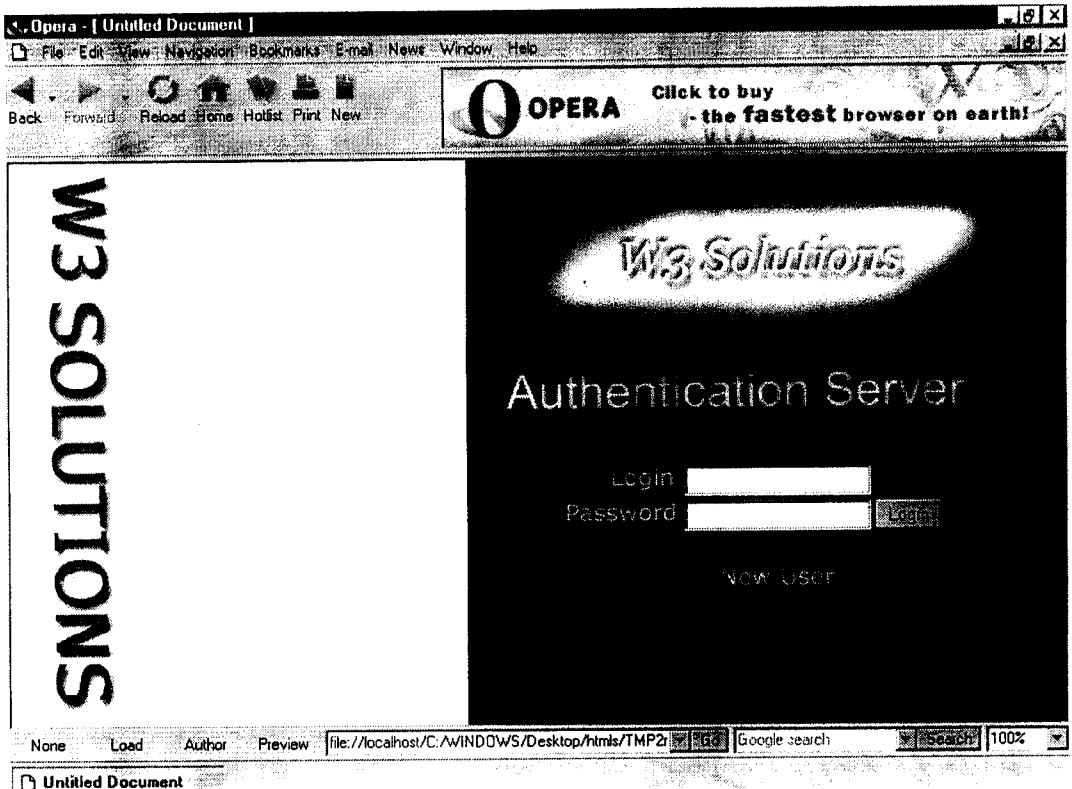


USER



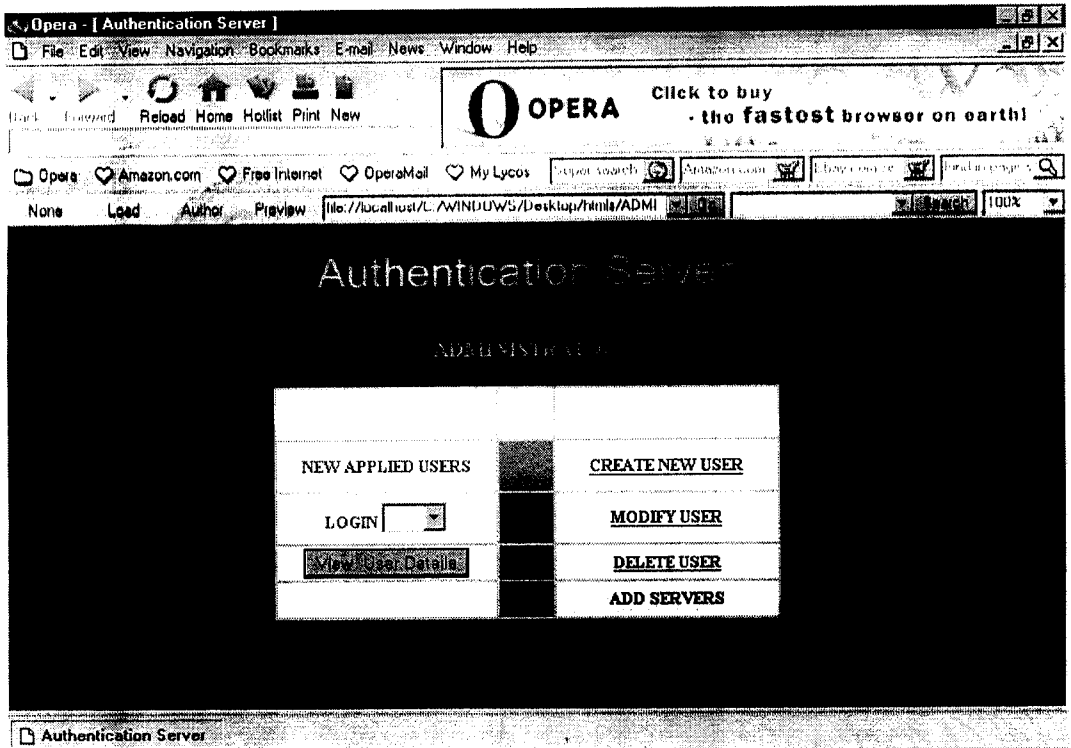
INPUT AND OUTPUT SCREENS

Login screen



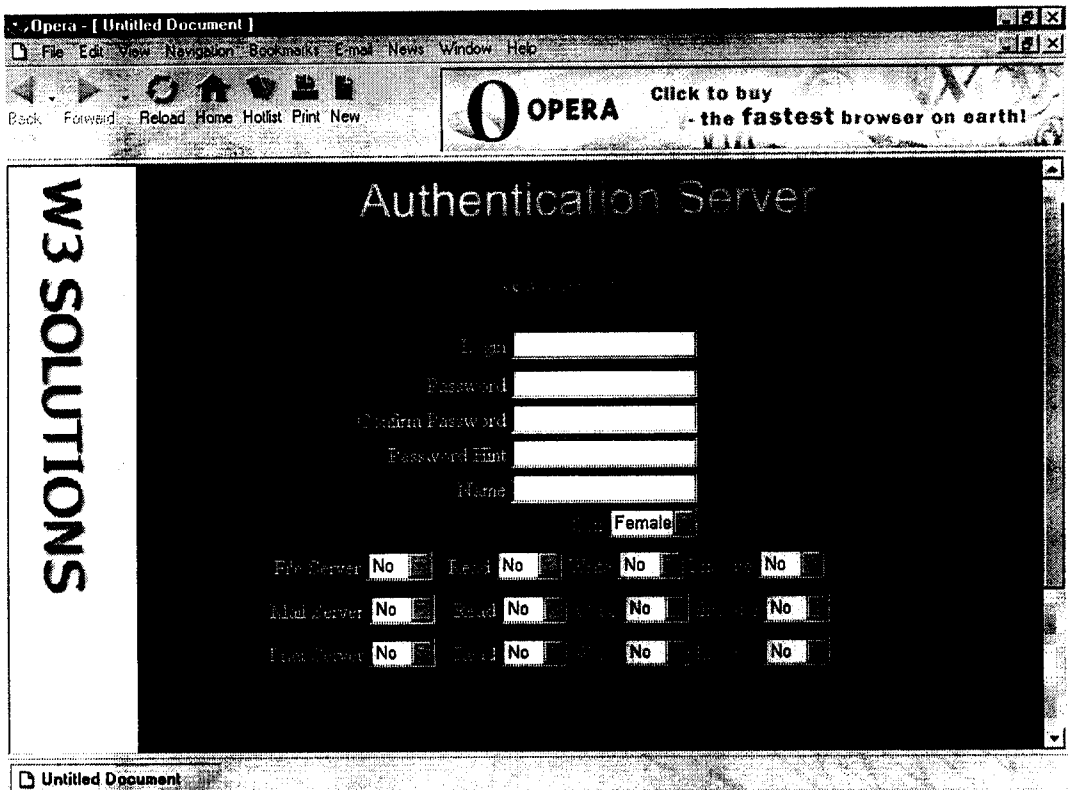
This is the first page of the Authentication Server and it is the **Login screen**.

Administrator Module Screen



This is the main page of the Administration Module.

New Client Application Screen



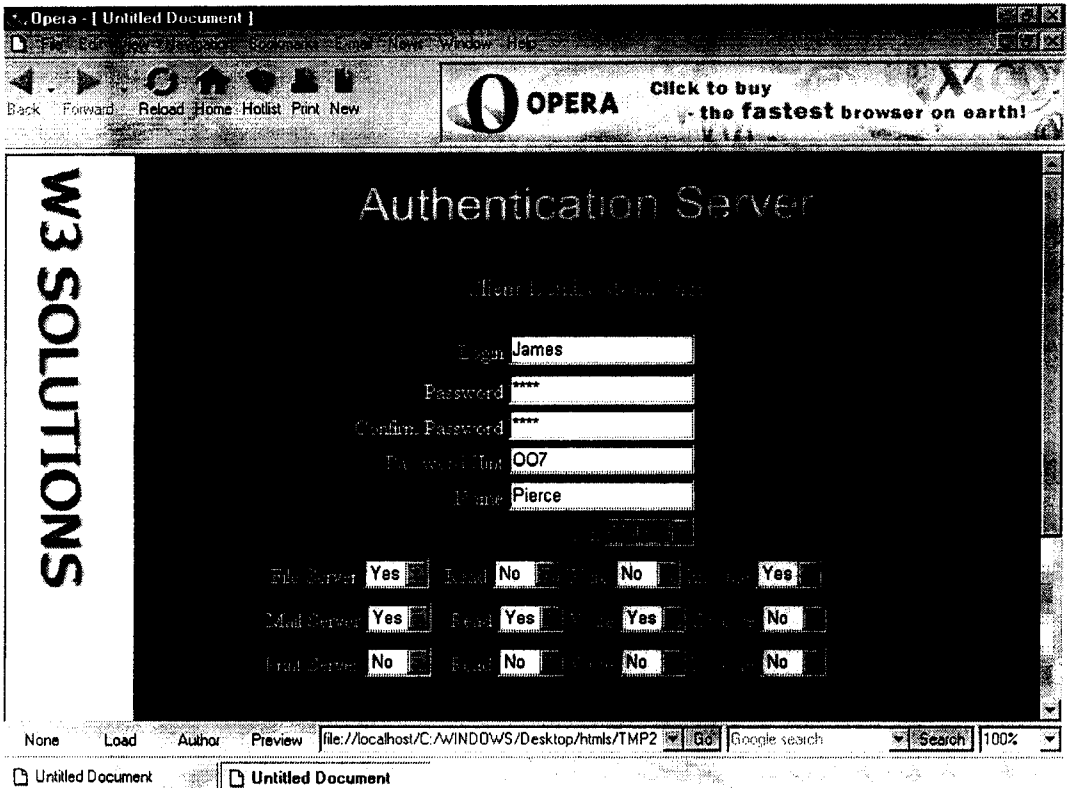
This is the New Client Application Page of the Authentication Server.

Selection Screen



This is the **selection screen** for choosing the clients' name whose registered details have to be modified.

Client Details Modification Screen



This is the **Client Details modification** screen that shows the selected client's details that can be modified.

Client Deletion Screen

Opera - [Untitled Document]

File Edit View Navigation Bookmarks E-mail News Window Help

Back Forward Reload Home Home Hotlist Print New

OPERA Click to buy the fastest browser on earth!

W3 SOLUTIONS

Authentication Server

Client Deletion

Email

Password

Confirm Password

Password Hint

Name

Sex

File Server Yes Read No Write No Execute Yes

Mail Server No Read No Write No Execute No

Print Server Yes Read No Write No Execute Yes

None Load Author Preview file://localhost/C:/WINDOWS/Desktop/html/TMP2 Google search 100%

Untitled Document

This is the **Client Deletion Screen** here the confirmation is made and by clicking the 'Delete' Button the current client who has been chosen will be erased from the database.

5. DATABASE DESIGN

The database has become an integral part of almost every human's life. Without it, many things we do would become very tedious, perhaps impossible tasks. Banks, universities, and libraries are three examples of organizations that depend heavily on some sort of database system. On the Internet, search engines, online shopping, and even the website naming convention (<http://www...>) would be impossible without the use of a database. A database that is implemented and interfaced on a computer is often termed a database server.

In this project MySQL is the database because it is compatible with the Linux Mandrake 8.2.

5.1 ABOUT MySQL

One of the fastest SQL (Structured Query Language) database servers currently on the market is the MySQL server, developed by T.c.X. DataKonsultAB. MySQL, available for download at <http://www.mysql.com/>, offers the database programmer with an array of options and capabilities rarely seen in other database servers. MySQL is free of charge for those wishing to use it for private and commercial use.

These capabilities range across a number of topics, including the following:

- Ability to handle an unlimited number of simultaneous users.
- Capacity to handle 50,000,000+ records.
- Very fast command execution, perhaps the fastest to be found on the market.

- Easy and efficient user privilege system.

However, perhaps the most interesting characteristic of all is the fact that it's **free**. That's right, T.c.X offers MySQL as a free product to the general public.

5.3 DATABASE TABLE-STRUCTURES SAMPLES

5.3.1 Table name: temptable

Field Name	Type	Length
login	Text	50
password	Text	15
password confirm	Text	5
password hint	Text	15
name	Text	15
sex	Text	8
resource_1	Text	15
res_1_r	Text	1
res_1_w	Text	1
res_1_e	Text	1
resource_2	Text	15
res_2_r	Text	1
res_2_w	Text	1
res_2_e	Text	1
resource_3	Text	15
res_3_r	Text	1
res_3_w	Text	1
res_3_e	Text	1
status	Text	1

5.3.2 Table name: **maintable**

Field Name	Type	Length
login	Text	50
password	Text	15
password confirm	Text	5
password hint	Text	15
name	Text	15
sex	Text	8
resource_1	Text	15
res_1_r	Text	1
res_1_w	Text	1
res_1_e	Text	1
resource_2	Text	15
res_2_r	Text	1
res_2_w	Text	1
res_2_e	Text	1
resource_3	Text	15
res_3_r	Text	1
res_3_w	Text	1
res_3_e	Text	1
dir	Text	15

6. SYSTEM ELEMENTS

6.1 SYSTEM ELEMENTS - HARDWARE

Environment: Local Area Network

6.1.1 Server Configuration

Brand	:	DELL
Processor	:	Pentium IV
Processor Speed	:	1 Giga Hertz.
RAM	:	128 MB
Clock Speed	:	133 Hz
Network Card	:	Real tech.

6.1.2 Clients Configuration

For development purpose	:	8
Brand	:	HEWLETT PACKARD
Model	:	Brio
Processor	:	Pentium III
Processor Speed	:	550 MHz
RAM	:	128 MB

6.1.3 Modem

Brand	:	Dax
Type	:	External
Speed	:	128 Kbps.

6.1.4 Soft modem

Brand	:	NOKIA
Cell Phone Brand	:	NOKIA
Model	:	5110
Data Suite Cable	:	9DAW
Data Suite	:	Nokia Data Suite 3

6.1.5 Operating Systems

Windows 98 / NT
Linux Mandrake 8.2

6.2 SYSTEM ELEMENTS – SOFTWARE

6.2.1 Java

6.2.1.1 Introduction to Java

Java is related to C++, which is a direct descendent of C. Much of the characteristics of Java are inherited from these two languages. From C, Java derives its syntax. Many of Java's object-oriented features were influenced by C++. In fact, several of Java's defining characteristics come from its predecessors. Moreover, the creation of Java was deeply rooted in the process of refinement and adaptation that has been occurring in computer programming languages for the past three decades.

6.2.1.2 Java: Security

Java provides a “firewall” between a networked application and our computer. When using a Java compatible browser, we can safely download Java applets without fear of vital infection or malicious intent.

6.2.1.3 Java: Portability

Many types of computers and operating systems are in use throughout the world and many are connected to the Internet. For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed. As you will soon see the same mechanism that helps ensure security also helps create portability. Indeed, Java's solutions to these two problems are both elegant and efficient.

6.2.1.4 Java: Server Side

Server-side Java applications is one of the latest and most exciting trends in Java programming. The cross platform nature of Java is extremely useful for organizations that have a heterogeneous collection of servers running various flavors of the Unix and Windows operating systems.

6.2.1.5 Java: Servlets

Java Servlets are a key component of server-side Java development. A Servlet is a small, pluggable extension to a server that enhances the server's functionality. Servlets allow developers to extend and customize any Java-enabled server – a web server, a mail server, an application server, or any custom server – with a hitherto unknown degree of portability, flexibility, and ease.

6.2.1.6 Advantages of Servlets

Portability

Servlets are highly portable across operating systems and across server implementations. Servlet portability is not mandatory. Servlets avoid the most error - prone and inconsistently implemented portion of Java language: the Abstract Windowing Toolkit (AWT) that forms the basis of Java graphical user interfaces.

Power

Servlets can harness the full power of the core Java APIs: networking and URL access, multithreading, image manipulation, data compression, database connectivity, internationalization, remote method invocation (RMI), CORBA connectivity, and object serialization, among others. If you want to write a web application that allows employees to query a corporate legacy database, you can

take advantage of all of the Java Enterprise APIs in doing so. Or, if you need to create a web- based directory lookup application, you can make use of the JNDI API.

Efficiency and Endurance

Servlet invocation is highly efficient. Once a servlet is loaded, it generally remains in the server's memory as a single object instance. Thereafter, the server invokes the servlets to handle a request using a simple, lightweight method invocation. Unlike with CGI, there's no process to stand or interpreter to invoke so the servlet can begin handling the request immediately. Separate threads handle multiple concurrent requests so servlets are highly scalable.

Safety

Servlet support safe programming practices on a number of levels. Because they are written in Java, Servlets inherit the strong type safety of the Java language. Servlets can handle errors safely, due to Java's exception-handling mechanism. If a servlet divides by zero or performs some other illegal operations, it throws an exception that can be safely caught and handled by the server, which can politely log the error and apologies to the user.

Integration

Servlets are tightly integrated with the server. This integration allows a servlet to cooperate with the server in ways that the CGI program cannot.

6.2.2 TCP/IP

TCP and IP were developed by a Department of Defense (DOD) research project to connect a number different networks designed by different vendors into a network of networks (the "Internet"). It was initially successful because it delivered a few basic services that everyone needs (file transfer, electronic mail, remote logon) across a very large number of client and server systems. Several computers in a small department can use TCP/IP (along with other protocols) on a single LAN. The IP component provides routing from the department to the enterprise network, then to regional networks, and finally to the global Internet. On the battlefield a communications network will sustain damage, so the DOD designed TCP/IP to be robust and automatically recover from any node or phone line failure. This design allows the construction of very large networks with less central management. However, because of the automatic recovery, network problems can go undiagnosed and uncorrected for long periods of time.

As with all other communications protocol, TCP/IP is composed of 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.

There are three levels of TCP/IP knowledge. Those who administer a regional or national network must design a system of long distance phone lines, dedicated routing devices, and very large configuration files. They must know the IP numbers and physical locations of thousands of subscriber networks. They must also have a formal network monitor strategy to detect problems and respond quickly.

Each large company or university that subscribes to the Internet must have an intermediate level of network organization and expertise. A half dozen routers might be configured to connect several dozen departmental LANs in several buildings. All traffic outside the organization would typically be routed to a single connection to a regional network provider.

However, the end user can install TCP/IP on a personal computer without any knowledge of either the corporate or regional network. Three pieces of information are required:

1. The IP address assigned to this personal computer
2. The part of the IP address (the subnet mask) that distinguishes other **machines on the same LAN (messages can be sent to them directly)** from machines in other departments or elsewhere in the world (which are sent to a router machine)
3. The IP address of the router machine that connects this LAN to the rest of the world.

In the case of the PCLT server, the IP address is 130.132.59.234. Since the first three bytes designate this department, a "subnet mask" is defined as 255.255.255.0 (255 is the largest byte value and represents the number with all bits turned on). It is a Yale convention (which we recommend to everyone) that the router for each department has station number 1 within the department network. Thus the PCLT router is 130.132.59.1. Thus the PCLT server is configured with the values:

- My IP address: 130.132.59.234
- Subnet mask: 255.255.255.0
- Default router: 130.132.59.1

The subnet mask tells the server that any other machine with an IP address beginning 130.132.59.* is on the same department LAN, so messages are sent to it directly. Any IP address beginning with a different value is accessed indirectly by sending the message through the router at 130.132.59.1 (which is on the departmental LAN).

6.2.3 Sockets

Sockets provide point-to-point, two-way communication between two processes. Sockets are very versatile and are a basic component of interprocess and intersystem communication. A socket is an endpoint of communication to which a name can be bound. It has a type and one or more associated processes. Sockets exist in communication domains. A socket domain is an abstraction that provides an addressing structure and a set of protocols. Sockets connect only with sockets in the same domain. Twenty three socket domains are identified (see `<sys/socket.h>`), of which only the UNIX and Internet domains are normally used. Solaris 2.x Sockets can be used to communicate between processes on a single system, like other forms of IPC.

The UNIX domain provides a socket address space on a single system. UNIX domain sockets are named with UNIX paths. Sockets can also be used to communicate between processes on different systems. The socket address space between connected systems is called the Internet domain. Internet domain communication uses the TCP/IP Internet protocol suite.

Socket types define the communication properties visible to the application. Processes communicate only between sockets of the same type. There are five types of socket.

◆ A stream socket

Provides two-way, sequenced, reliable, and unduplicated flow of data with no record boundaries. A stream operates much like a telephone conversation. The socket type is `SOCK_STREAM`, which, in the Internet domain, uses Transmission Control Protocol (TCP).

◆ **A datagram socket**

Supports a two-way flow of messages. A on a datagram socket may receive messages in a different order from the sequence in which the messages were sent. Record boundaries in the data are preserved. Datagram sockets operate much like passing letters back and forth in the mail. The socket type is `SOCK_DGRAM`, which, in the Internet domain, uses User Datagram Protocol (UDP).

◆ **A sequential packet socket**

Provides a two-way, sequenced, reliable, connection, for datagrams of a fixed maximum length. The socket type is `SOCK_SEQPACKET`. No protocol for this type has been implemented for any protocol family.

◆ **A raw socket**

--Provides access to the underlying communication protocols.

These sockets are usually datagram oriented, but their exact characteristics depend on the interface provided by the protocol.

▪ **Datagram sockets**

A datagram socket does not require that a connection be established. Each message carries the destination address. If a particular local address is needed, a call to `bind()` must precede any data transfer. Data is sent through calls to `sendto()` or `sendmsg()`. The `sendto()` call is like a `send()` call with the destination address also specified. To receive datagram socket messages, call `recvfrom()` or `recvmsg()`. While `recv()` requires one buffer for the arriving

data, `recvfrom()` requires two buffers, one for the incoming message and another to receive the source address.

Datagram sockets can also use `connect()` to connect the socket to a specified destination socket. When this is done, `send()` and `recv()` are used to send and receive data.

`accept()` and `listen()` are not used with datagram sockets

▪ **Socket Options**

Sockets have a number of options that can be fetched with `getsockopt()` and set with `setsockopt()`. These functions can be used at the native socket level (`level = SOL_SOCKET`), in which case the socket option name must be specified. To manipulate options at any other level the protocol number of the desired protocol controlling the option of interest must be specified (see `getprotoent()` in `getprotobyname()`).

6.2.4 Servlet Runner

If you would like to run your servlet in a web server, please see that server's documentation for instructions. This section explains how to run the servlet in the `servletrunner` utility that comes with this release.

The `servletrunner` is a small utility, intended for testing. It is multithreaded, so it can run more than one servlet. It can be used, therefore, to run multiple servlets simultaneously, or to test one servlet that calls other servlets in order to satisfy client requests. Unlike some web servers, it does not automatically reload servlets when they are updated.

The *Servlet Runner* is ideal for intranets, homes, schools, small business workgroups and anyone who wants to set up a personal Web server. In corporations, any department or individual can post servlets documents and share information with anyone else in the company via the corporate intranet. Users of Windows 95/98/98SE/ME/NT, Linux can now share Web content as easily as they share folders on a network.

It is also a great cost-saving solution for small businesses and schools because they no longer need to dedicate a PC to a Web server. Instead, they can leverage their existing PCs simply by enhancing them with the *Servlet Runner* software. Finally, home users who want to experiment with Web publishing can set up their personal computer as a Web server and easily publish information on the Internet.

The *Servlet Runner* software is now available to download free (other than the cost of connect time, if applicable) from the *Sun Microsystems* Web site.

6.2.5 Software used in Development

Front end:

JAVA
Java Script
Java Servlets
Java Networking

Back end:

MySQL

Browser:

Netscape Navigator
HotJava

Development Tools:

Macromedia Flash 5
Macromedia Dreamweaver Ultradev 4
Adobe Photoshop 6

7. SYSTEM IMPLEMENTATION AND TESTING

7.1 SYSTEM DESIGN

This is a basic design of Client-Server Architecture. All transactions of data are done between the client and the server. In this project the Authentication Server is at W3 Solutions, Mysore.

This system is designed to reduce much work, in the input design; the system's interaction with the client and the officials is clearly done. The Client interaction with the system is made easier and secured by **authorized entry only**.

The Administration module is accessed by the Administrator, this design is easier so that they don't find any difficulty in the interaction. Each function is clearly defined and designed.

The output design should be more reliable because it is more dependent for the success of the system. The output is made clear and precise to meet the expected situations. Finally the system is designed to fulfill the requirements of the firm.

7.2 CLIENT SERVER MODEL

➤ Client Information Description

- All data passed between the Client and the Servers will be sent via a common network layer.
- Data requests from Client to Server are transferred and Response information from Server to Client is also transferred.
- Information handled by the Client will be one of the following types
 1. Registration information
 2. Login information
 3. Search Request/result information
 4. Data retrieval request/response information

➤ Client Interface Description

1. Will provide users with a means to register themselves as users of the system.
2. Will provide users with a means of logging on to the system .
3. Will provide users with the an interface from which they can perform searches of the data being served across the system.

➤ Network Information Description

1. Data is bundled into a predetermined format
2. Data is passed to the network
3. The network sends the data
4. The network receives the data

➤ **Network Interface Description**

The network layer will provide an interface for all servers and clients such that communication and information passage may occur. Other project team members, namely those from the client and server groups, will use the network layer.

➤ **Server Information Description**

1. All data, which is passed between clients and servers, will be sent via a common network layer.
2. Data is passed between clients and servers.
3. All data stored on the server will be kept in a relational database, and accessed through a database module.

The information handled by the server will be one of three types:

1. Registration/Authentication Information.
2. Search Query's/Results.
3. Data.

➤ **Server Interface Description**

The Servers will be interfaced to an external database system and a common network layer. The interface to the external database will be contained within a database module.

➤ **Client Functional Description**

1. Registration

- 1.1 must take place before the client can login to system.
- 1.2 A GUI will request information from the user
- 1.3 Request will be sent to a server.
- 1.4 Server will return a response containing the security level granted to the user, and a list of Servers.
- 1.5 These Servers are the ones which the client will forward its searches to.

2. Login

- 2.1 Must take place before the client can access and use the system.
- 2.2 A Login GUI will request login Information (user name, password)
- 2.3 Login Information sent to Server
- 2.4 Response received from Server granting or denying access to the system
- 2.5 Access is granted, user has use of main GUI

3. Search

- 3.1 Searches will consist of some/all of the following:
Keywords describing the data, general limits on acceptable data source formats (file or Database), and specific data source requirements.
- 3.2. Search is Sent to Server when Search button if pressed
- 3.3. Server Responds with a List of matches to users Search value.
- 3.4. Each record in List will be displayed in GUI for client to optionally download from a Server.

➤ **Network Functional Description**

Accessibility

1. The layer shall be solely accessible via objects.
2. Layers entities shall be accessible by all network entities, including servers and clients.

Data Transferral

1. The layer shall provide a mode of data transferral.
2. The layer shall not examine data for content.
3. The layer shall not guarantee the persistence of connections created.
4. The layer shall encrypt all data prior to transferral and unencrypted all data following transferral.
5. The layer shall not detect any data corruption that may occur prior to, during, or following data transferral.
6. The layer shall realize all network timeouts and attempt to connect a set number of times prior to failing.

Control

1. Layer objects shall provide the sole interface to all networking activity.
2. The layer shall maintain and control all port numbers and server IPs and be the sole storage repository for this information.

General

1. Layer objects shall never thread.
2. Layer objects shall be abstract enough that the underlying networking implementation could be replaced with minimal effort.

➤ **Server Functional Description**

Registration

Registration consists of functionality that manages user and server information on the system.

These functions are: User Registration, Retrieval and updating of user information, removal of a user, adding and removing user security levels, and adding and removing of servers.

Authentication

Authentication consists of the functionality which will ensure that the user exists on the system, and had the necessary permissions to access the information on the system.

Searching

Searching consists of the functionality which will search through the data stored on the system, based on a request from the client. This will come in two forms: a simple search and an advanced SQL search.

Data Storage

Data Storage consists of the functionality which will store all of the data on the system, whether it be a database or discrete files. It also contains the functionality which will index this data for searching. A variety of general solutions for authentication have been commonly employed... each with their own benefits and limitations.

7.3 SYSTEM TESTING

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently before line operation commences. To assure that quality of the system the following are to be considered:

- Scope of testing
- Tools used
- Test platforms
- Methods used

Here the scope is designed as the application including all the modules. Debugging tools are provided in Visual Interdev and hence they are used.

The test method used is the hierarchical test method whose root is the application and whose leaves are the individual data elements.

The system has been tested for the following criteria:

- *Validation of inputs for each available data.*
- *Sequential tests which ensure that the input for the current stage is done, only after the completion of the previous stage.*
- *Consistency of the application, which ensures that the system works properly in the long run.*

7.4 SYSTEM IMPLEMENTATION

System Implementation refers to implementation of the project to the real time. This project is installed at W3 Solutions, Mysore. The personnel of the firm are registered in this developed Authentication Server.

The Authentication Server holds a common Database for storage of client details regarding their access privileges and permissions on the available resources.

The administrator has the entire responsibility on the Authentication Server and all tasks of the Authentication Server. He is permitting the clients to access the resources they ask for. The administrator also gives the privileges to the clients on the resources the access. He is responsible for adding up a new resources / server to the authentication server, if not required he is the only person to remove the resource form the authentication server.

The Authentication Server is implemented and it is working in better way than what was really expected.

8. CONCLUSION

This project named “Authentication Server” helps to manage the administration tasks in registration of clients and resources, making changes in the registered information, deleting the unwanted information.

It also helps to record the access period for clients on the resources with privileges.

User authentication and authorization issues are becoming increasingly complex as systems move into a distributed networked information environment. No solution is universal and different approaches are needed by different organizations to meet local needs. It is important that a system clearly define its needs. The goal in user authentication and authorization is to let valid parties access databases and information services from anywhere at anytime without making the process too difficult for the system, information provider.

Flexibility is needed on both the part of information providers to allow different solutions to meet organizational needs and by networked system that need to incorporate reasonable security.

9. FUTURE ENHANCEMENT

- * Protect all types of files not just ASP scripts .gif .jpg .htm .zip .pdf .mdb .anything!

- * Users / groups / authentication Independent from NT users

- * Users and groups stored in a database backend with high performance access via native OLEDB.

- * Email functionality: Send email to any selected users using full search criteria.

- * User Password reminder: Users can request a password reminder by entering their userid. If they have a valid email address on record their password will be mailed to them.

- * User Change Password: Users can change their own password through a web interface by providing their username, old password, new password, and new password confirmation.

10. REFERENCES

Ref #1: www.linux.org “Linux”

Ref #2: Kerman “Macromedia Flash 5”

Ref #3: Sumitabu dos-Unix programming

Ref #4: www.mysql.com. “MySQL”

11. APPENDIX

SAMPLE CODE FOR REGISTRING A NEW PROJECT

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class newuser extends HttpServlet
{
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {

        res.setContentType("text/html");
        PrintWriter pw=res.getWriter();

pw.println("<html>");
pw.println("<head>");
pw.println("<title>Authentication Server - New User Sign Up</title>");
pw.println("</head>");
pw.println("<body bgcolor=#FFFFFF text=#000000>");
pw.println("<p align=center><font color=#0000CC size=4>Authentication Server
- New User");
    pw.println("</font></p>");
pw.println("<p align=center><font color=#0000CC size=4>Sign
Up</font></p>");
pw.println("<form name=frm_newuser method=post
action=http://localhost:8080/servlet/newuser_reg>");
    pw.println("<table width=77% border=0 align=center
bordercolor=#CCCCCC>");
    pw.println("<tr >");
        pw.println("<td height=29 width=63%> ");
            pw.println("<div align=right><font size=3 color=#000000>Login</font> ");
                pw.println("<input type=text name=txt_login>");
            pw.println("</div>");
        pw.println("</td>");
        pw.println("<td height=29 width=37%>User Exist?</td>");
    pw.println("</tr>");
    pw.println("<tr >");
        pw.println("<td height=19 width=63%> ");
            pw.println("<div align=right>Password ");
                pw.println("<input type=password name=txt_pass>");
            pw.println("</div>");
```



```

pw.println("<td height=5 width=63%> ");
pw.println("<table width=87% border=1 align=right>");
pw.println("<tr> ");
pw.println("<td width=45%> ");
pw.println("Mail Server ");
pw.println("<select name=cmb_ms>");
pw.println("<option value=0>No</option>");
pw.println("<option value=1>Yes</option>");
pw.println("</select>");
pw.println("</td>");
pw.println("<td width=72%> ");
pw.println("<select name=cmb_ms_r>");
pw.println("<option value=0>No</option>");
pw.println("<option value=1>Yes</option>");
pw.println("</select>");
pw.println("<select name=cmb_ms_w>");
pw.println("<option value=0>No</option>");
pw.println("<option value=1>Yes</option>");
pw.println("</select>");
pw.println("<select name=cmb_ms_e>");
pw.println("<option value=0>No</option>");
pw.println("<option value=1>Yes</option>");
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</td>");
pw.println("<td height=5 width=37%>&nbsp;  </td>");
pw.println("</tr>");
pw.println("<tr> ");
pw.println("<td height=5 width=63%> ");
pw.println("<table width=87% border=1 align=right>");
pw.println("<tr> ");
pw.println("<td width=45%> ");
pw.println("Print Server");
pw.println("<select name=cmb_ps>");
pw.println("<option value=0>No</option>");
pw.println("<option value=1>Yes</option>");
pw.println("</select>");
pw.println("</td>");
pw.println("<td width=72%> ");
pw.println("<select name=cmb_ps_r>");
pw.println("<option value=0>No</option>");
pw.println("<option value=1>Yes</option>");
pw.println("</select>");
pw.println("<select name=cmb_ps_w>");

```

```

        pw.println("<option value=0>No</option>");
        pw.println("<option value=1>Yes</option>");
    pw.println("</select>");
    pw.println("<select name=cmb_ps_e>");
        pw.println("<option value=0>No</option>");
        pw.println("<option value=1>Yes</option>");
    pw.println("</select>");
    pw.println("<td width=72%> ");

        pw.println("</td>");
    pw.println("</tr>");
    pw.println("</table>");
    pw.println("</td>");
    pw.println("<td height=5 width=37%>&nbsp;</td>");
    pw.println("</tr>");

    pw.println("<tr> ");
    pw.println("<td height=27 width=63%> ");
    pw.println("<p align=right> ");
        pw.println("<input type=submit name=Submit value=Apply>");
        pw.println("<input type=reset name=Reset value=Reset>");
    pw.println("</p>");
    pw.println("</td>");
    pw.println("<td height=27 width=37%> ");
    pw.println("<p>&nbsp;</p>");
    pw.println("</td>");
    pw.println("</tr>");
    pw.println("</table>");
    pw.println("<p>&nbsp;</p>");
    pw.println("</form>");
    pw.println("</body>");
    pw.println("</html>");

    }
}

```

```

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class newuser_reg extends HttpServlet
{
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException

```

```
{
```

```
res.setContentType("text/html");  
PrintWriter pw=res.getWriter();
```

```
String str_nu_login=req.getParameter("txt_login");  
String str_nu_pass=req.getParameter("txt_pass");  
String str_nu_compass=req.getParameter("txt_compass");  
String str_nu_pass_hint=req.getParameter("txt_passhint");  
String str_nu_name=req.getParameter("txt_name");  
String str_nu_sex=req.getParameter("cmb_sex");
```

```
String str_nu_files_r=req.getParameter("cmb_fs");  
String str_nu_files_r=req.getParameter("cmb_fs_r");  
String str_nu_files_w=req.getParameter("cmb_fs_w");  
String str_nu_files_e=req.getParameter("cmb_fs_e");  
int fs=Integer.parseInt(str_nu_files_r);  
int fs_r=Integer.parseInt(str_nu_files_r);  
int fs_w=Integer.parseInt(str_nu_files_w);  
int fs_e=Integer.parseInt(str_nu_files_e);
```

```
String str_nu_ms=req.getParameter("cmb_ms");  
String str_nu_ms_r=req.getParameter("cmb_ms_r");  
String str_nu_ms_w=req.getParameter("cmb_ms_w");  
String str_nu_ms_e=req.getParameter("cmb_ms_e");  
int ms=Integer.parseInt(str_nu_ms);  
int ms_r=Integer.parseInt(str_nu_ms_r);  
int ms_w=Integer.parseInt(str_nu_ms_w);  
int ms_e=Integer.parseInt(str_nu_ms_e);
```

```
String str_nu_ps=req.getParameter("cmb_ps");  
String str_nu_ps_r=req.getParameter("cmb_ps_r");  
String str_nu_ps_w=req.getParameter("cmb_ps_w");  
String str_nu_ps_e=req.getParameter("cmb_ps_e");  
int ps=Integer.parseInt(str_nu_ps);  
int ps_r=Integer.parseInt(str_nu_ps_r);  
int ps_w=Integer.parseInt(str_nu_ps_w);  
int ps_e=Integer.parseInt(str_nu_ps_e);
```

```
String status="0";  
int available_int=0,pass_chk_int=0,blank_chk_int=0,login_blk=0;  
String verify_name;  
Connection con=null;
```

```
try  
{
```

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

```

        con=DriverManager.getConnection("jdbc:odbc:dsname");
        Statement s=con.createStatement();

//====DONE=====// Checking
user existence
        ResultSet result;

        result=s.executeQuery("select * from temp_table");

        if((str_nu_login).equals(""))
        {
            login_blk=1;
        }

        while(result.next())                //=== Checking user for
Existence of User !!
        {
            if(result.getString("login").equals(str_nu_login))
            {
                available_int=1;
            }
        }

        if(available_int==1)
        {
            pw.println("<p><B> Sorry this user already exists
!!</B></p>");
            pw.println("<p><a href=javascript:window.history.back()
target=_self><b>Click here to correct</b></a></p>");
        }
//=====// Checking user
existence End.....DONE

        int int_fs_error=0;
        int int_ms_error=0;
        int int_ps_error=0;

//=====File Server CHK
        if(fs==0)
        {
            if((fs_r==1)|(fs_w==1)|(fs_e==1))
            {
                int_fs_error=1;
            }
        }
        if(fs==1)
        {
            if((fs_r==0)&(fs_w==0)&(fs_e==0))

```

```

        {int_fs_error=1;    }
    }
//=====Mail Server CHK
if(ms==0)
{
    if((ms_r==1)|(ms_w==1)|(ms_e==1))
        { int_ms_error=1; }
}
if(ms==1)
{
    if((ms_r==0)&(ms_w==0)&(ms_e==0))
        {int_ms_error=1;    }
}
//=====Print Server CHK
if(ps==0)
{
    if((ps_r==1)|(ps_w==1)|(ps_e==1))
        { int_ps_error=1; }
}
if(ps==1)
{
    if((ps_r==0)&(ps_w==0)&(ps_e==0))
        {int_ps_error=1;    }
}

//=====// BLANK CHECK !!

int name_nul_int=0;
    if((str_nu_name).equals("")) // NAME BLANK
    {
        name_nul_int=1;
    }
int compass_nul_int=0;
    if((str_nu_compass).equals("")) // COMPASS BLANK
    {
        compass_nul_int=1;
    }
int pass_nul_int=0;
    if((str_nu_pass).equals("")) //PASS BLANK
    {
        pass_nul_int=1;
    }

//=====// Blank Check End

```

```
//=====// Password & Confirm  
Pass CHECK !!
```

```
if((str_nu_pass).equals(str_nu_compass))  
{pass_chk_int=0; }  
else  
{pass_chk_int=1;}
```

```
//=====// Password & Confirm  
Pass CHECK END !!.....DONE
```

```
if((available_int==0)&(pass_chk_int==0)&(blank_chk_int==0)&(pass_nu  
l_int==0)&(login_blk==0)&(compass_nul_int==0)&(name_nul_int==0)&(int_fs_  
error==0)&(int_ms_error==0)&(int_ps_error==0))
```

```
{  
    pw.println(" In EXECUTE DECISION !!");
```

```
        s.executeUpdate("insert into temp_table  
values(""+str_nu_login+"",""+str_nu_pass+"",""+str_nu_compass+"",""+str_nu_pass_  
hint+"",""+str_nu_name+"",""+str_nu_sex+"",""+str_nu_fileser+"",""+str_nu_fileser_  
r+"",""+str_nu_fileser_w+"",""+str_nu_fileser_e+"",""+str_nu_ms+"",""+str_nu_ms_r  
+"",""+str_nu_ms_w+"",""+str_nu_ms_e+"",""+str_nu_ps+"",""+str_nu_ps_r+"",""+str_  
_nu_ps_w+"",""+str_nu_ps_e+"",""+status+"");
```

```
        pw.println(" login :"+str_nu_login);
```

```
        pw.println("Recorded !!<p><p>");
```

```
        pw.println("<a
```

```
href="+res.encodeUrl("admin")+"><b>Administrator</b></a>");
```

```
    }
```

```
    else
```

```
    {
```

```
        if((login_blk==1)|(pass_nul_int==1)|(compass_nul_int==1)|(name_nul_int  
==1))
```

```
        {
```

```
            pw.println("<p> Sorry <b> Blank field exists !!
```

```
</b></p>");
```

```
            pw.println("<p><a
```

```
href=javascript:window.history.back() target=_self><b>Click here to  
correct</b></a></p>");
```

```
        }
```

```
    else
```

```
    {        if(pass_chk_int==1)
```

```
        {
```

```

        pw.println("<p> Sorry <b>Password &
Confirm Password are not same !!</b></p>");
        pw.println("<p><a
href=javascript:window.history.back() target=_self><b>Click here to
correct</b></a></p>");
    }
    if(int_fs_error==1)
    {
        pw.println("<p> Sorry <b> File Server
Error !! </b></p>");
        pw.println("<p><a
href=javascript:window.history.back() target=_self><b>Click here to
correct</b></a></p>");
    }
    if(int_ms_error==1)
    {
        pw.println("<p> Sorry <b> Mail Server
Error !! </b></p>");
        pw.println("<p><a
href=javascript:window.history.back() target=_self><b>Click here to
correct</b></a></p>");
    }
    if(int_ps_error==1)
    {
        pw.println("<p> Sorry <b> Print Server
Error !! </b></p>");
        pw.println("<p><a
href=javascript:window.history.back() target=_self><b>Click here to
correct</b></a></p>");
    }
}
}

}catch(Exception e){}
}throws
}class

```

Client Details Modify

```

import java.io.*;
import javax.servlet.*;

```

```

import javax.servlet.http.*;
public class usermodify extends HttpServlet
{
public void service(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException
{

        res.setContentType("text/html");
        PrintWriter pw=res.getWriter();

pw.println("<html>");
pw.println("<head>");
pw.println("<title>Authentication Server - New User Sign Up</title>");
pw.println("</head>");
pw.println("<body bgcolor=#FFFFFF text=#000000>");
pw.println("<p align=center><font color=#0000CC size=4>Authentication Server
- User Modify");
    pw.println("</font></p>");
//pw.println("<p align=center><font color=#0000CC size=4>Sign
Up</font></p>");
pw.println("<form name=frm_newuser method=post action=>");
    pw.println("<table width=77% border=0 align=center
bordercolor=#CCCCCC>");
    pw.println("<tr> ");
        pw.println("<td height=29 width=63%> ");
            pw.println("<div align=right><font size=3 color=#000000>Login");
            pw.println("<select name=select>");
            pw.println("</select>");
            pw.println("</font> </div>");
        pw.println("</td>");
        pw.println("<td height=29 width=37%>User Exist?</td>");
pw.println("</tr>");
    pw.println("<tr> ");
        pw.println("<td height=19 width=63%> ");
            pw.println("<div align=right>Password ");
            pw.println("<input type=password name=txt_pass>");
            pw.println("</div>");
        pw.println("</td>");
        pw.println("<td height=19 width=37%>&nbsp;&nbsp;&nbsp;</td>");
pw.println("</tr>");
    pw.println("<tr> ");
        pw.println("<td height=8 width=63%> ");
            pw.println("<div align=right>Confirm Password ");
            pw.println("<input type=password name=txt_conpass>");
            pw.println("</div>");
        pw.println("</td>");

```



```

pw.println("<td height=8 width=37%>Diff Pass?</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td height=14 width=63%> ");
pw.println("<div align=right>Password Hint ");
pw.println("<input type=text name=txt_passhint>");
pw.println("</div>");
pw.println("</td>");
pw.println("<td height=14 width=37%>&nbsp;</td>");
pw.println("</tr>");
pw.println("<tr> ");
pw.println("<td height=2 width=63%> ");
pw.println("<div align=right>Name ");
pw.println("<input type=text name=txt_name>");
pw.println("</div>");
pw.println("</td>");
pw.println("<td height=2 width=37%>&nbsp;</td>");
pw.println("</tr>");
pw.println("<tr> ");
pw.println("<td height=6 width=63%> ");
pw.println("<div align=right>Sex ");
pw.println("<select name=cmb_sex>");
pw.println("<option value=Male>Male</option>");
pw.println("<option value=Female>Female</option>");
pw.println("</select>");
pw.println("</div>");
pw.println("</td>");
pw.println("<td height=6 width=37%>&nbsp;</td>");
pw.println("</tr>");
pw.println("<tr> ");
pw.println("<td height=5 width=63%> ");
pw.println("<table width=87% border=1 align=right>");
pw.println("<tr> ");
pw.println("<td width=28%> ");
pw.println("<input type=checkbox name=chk_fileser
value=checkbox>");
pw.println("File Server</td>");
pw.println("<td width=72%> ");
pw.println("<input type=checkbox name=chk_fileser
value=checkbox>Read ");
pw.println("<input type=checkbox name=chk_fileser
value=checkbox>Write ");
pw.println("<input type=checkbox name=chk_fileser
value=checkbox>Execute");
pw.println("</td>");
pw.println("</tr>");

```



```

    pw.println("</table>");
    pw.println("</td>");
    pw.println("<td height=5 width=37%>&nbsp;&nbsp;&nbsp;</td>");
    pw.println("</tr>");
    pw.println("<tr >");
    pw.println("<td height=5 width=63%> ");
    pw.println("<table width=87% border=1 align=right>");
    pw.println("<tr >");
    pw.println("<td width=28%> ");
    pw.println("<input type=checkbox name=chk_mailser
value=checkbox>");
    pw.println("Mail Server </td>");
    pw.println("<td width=72%> ");
    pw.println("<input type=checkbox name=chk_mailser
value=checkbox>Read ");
    pw.println("<input type=checkbox name=chk_mailser
value=checkbox>Write ");
    pw.println("<input type=checkbox name=chk_mailser
value=checkbox>Execute");
    pw.println("</td>");
    pw.println("</tr>");
    pw.println("</table>");
    pw.println("</td>");
    pw.println("<td height=5 width=37%>&nbsp;&nbsp;&nbsp;</td>");
    pw.println("</tr>");
    pw.println("<tr >");
    pw.println("<td height=5 width=63%> ");
    pw.println("<table width=87% border=1 align=right>");
    pw.println("<tr >");
    pw.println("<td width=28%> ");
    pw.println("<input type=checkbox name=chk_printser
value=checkbox>");
    pw.println("Print Server</td>");
    pw.println("<td width=72%> ");
    pw.println("<input type=checkbox name=chk_printser
value=checkbox>Read ");
    pw.println("<input type=checkbox name=chk_printser
value=checkbox>Write ");
    pw.println("<input type=checkbox name=chk_printser
value=checkbox>Execute");
    pw.println("</td>");
    pw.println("</tr>");
    pw.println("</table>");
    pw.println("</td>");
    pw.println("<td height=5 width=37%>&nbsp;&nbsp;&nbsp;</td>");
    pw.println("</tr>");

```

```

pw.println("<tr> ");
pw.println("<td height=14 width=63%> ");
pw.println("<div align=right>Directory ");
pw.println("<input type=text name=txt_dir>");
pw.println("</div>");
pw.println("</td>");
pw.println("<td height=14 width=37%>&nbsp;&nbsp;&nbsp;</td>");
pw.println("</tr>");
pw.println("<tr> ");
pw.println("<td height=27 width=63%> ");
pw.println("<p align=right> ");
pw.println("<input type=submit name=Submit value=Apply>");
pw.println("<input type=reset name=Reset value=Reset>");
pw.println("</p>");
pw.println("</td>");
pw.println("<td height=27 width=37%> ");
pw.println("<p>&nbsp;&nbsp;&nbsp;</p>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("<p>&nbsp;&nbsp;&nbsp;</p>");
pw.println("</form>");
pw.println("</body>");
pw.println("</html>");

    }
}

```

Administrator Page

```

import java.io.*;
import javax.servlet.*;
import java.sql.*;
import javax.servlet.http.*;
public class admin extends HttpServlet
{
public void service(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException
{
    res.setContentType("text/html");
    PrintWriter pw=res.getWriter();
    String a;
    Connection conn=null;
    try

```

```

        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            conn=DriverManager.getConnection("jdbc:odbc:dsname");
            Statement s=conn.createStatement();
            ResultSet result;
            result=s.executeQuery("select * from temp_table");

pw.println("<html>");
pw.println("<head>");
pw.println("<title>Authentication Server</title>");
pw.println("</head>");

pw.println("<body bgcolor=#FFFFFF text=#000000>");
pw.println("<p align=center>Authentication Server</p>");
pw.println("<p align=center><b>ADMINISTRATOR</b> </p>");
pw.println("<form name=form_admin method=post action=admin_nu>");
    pw.println("<table width=50% border=0 align=center bordercolor=#0000FF
bgcolor=#CCCCCC height=183>");
        pw.println("<tr >");
            pw.println("<td width=44% height=34>");
                pw.println("<div align=left>=====</div>");
                pw.println("</td>");
            pw.println("<td width=12% bgcolor=#FFFFFF height=34>=====</td>");
            pw.println("<td width=44% height=34>=====</td>");
        pw.println("</tr>");
        pw.println("<tr >");
            pw.println("<td width=44% height=33>");
                pw.println("<div align=center><font color=#0000FF><font size=2>NEW
APPLIED USERS</font></font></div>");
                pw.println("</td>");
            pw.println("<td width=12% height=33 bgcolor=#999999>&nbsp;  </td>");
            pw.println("<td width=44% height=33>");
                pw.println("<div align=center><font size=2><b><a href=newuser
target=_self>CREATE");
                pw.println(" NEW USER</a></b></font></div>");
                pw.println("</td>");
            pw.println("</tr>");
            pw.println("<tr>");
            pw.println("<td width=44% height=34>");
                pw.println("<div align=center><font color=#666666><b><font
size=2>LOGIN </font></b></font>");
                pw.println("<select name=cmb_admin_login>");
                    while(result.next())
                        {
                            a=result.getString("login");
                            pw.println("<option value="+a+">"+a+"</option>");

```

