NETWORK MONITORING ON LINUX ENVIRONMENT

Project work submitted on partial fulfillment of the requirements

for the award of the degree of

P-983

Master of Computer Applications
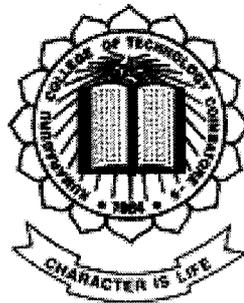
of

Bharathiar University

by

P. DHARANI SHANMUGHA PRIYA

(Reg. No. 0038M1024)

Under the guidance of

Ms. L.S. Jayashree, M.E.,

Department of Computer Science and Engineering

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)

Coimbatore – 641 006

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006.

CERTIFICATE

PROJECT REPORT 2002 - 2003

Certified that this is a bonafide report of
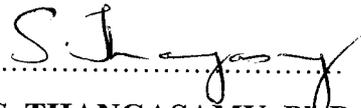
the project work done by

P. DHARANI SHANMUGHA PRIYA

(Reg. No. 0038M1024)

Ms. L.S. JAYASHREE, M.E.,

Project guide

Computer Science & Engineering
DEPARTMENT

Prof. S. THANGASAMY, PH.D.,

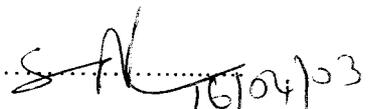Head of the Department

Computer Science & Engineering
Department

Place: Coimbatore

Date: 16 - 04 - 2003

Submitted for viva-voce examination held on

16 - 04 - 2003

....

Internal Examiner

External Examiner

Declaration

Declaration

This is to certify that this project work entitled "**NETWORK MONITORING ON LINUX ENVIRONMENT**" being submitted by **P. DHARANI SHANMUGHA PRIYA** (Reg. No. 0038M1024) for the award of degree of MASTER OF COMPUTER APPLICATIONS is a bonafide work carried under my guidance. The results embodied in this project work have not been submitted to any other university or institute for the award of any degree or diploma.

P. Dharani Shanmugha Priya
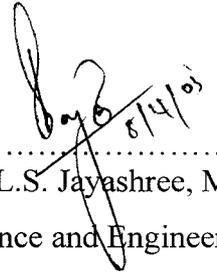
Ms. L.S. Jayashree, M.E.,
Dept. of Computer Science and Engineering,
Kumaraguru College of Technology,
Coimbatore – 641 006.

Certificate

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

P. Dharani Shanmugha Priya

undergoing Final year M.C.A, Reg No 0038M1024, in Kumaraguru College of Technology, Coimbatore has successfully completed the individual project entitled:

"Network Monitoring on Linux Environment"

during the period of December 2002 to March 2003 in our concern. During this period, her conduct was found to be good.

We wish her all the best in her future endeavours.
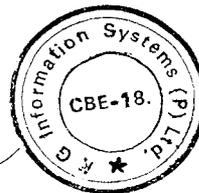
With Regards
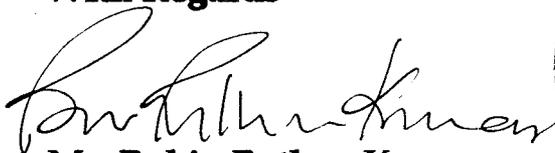
Mr. Robin Rathan Kumar
Project Guide
KG Information Systems Private Ltd.
Coimbatore-641018

We make IT happen

ISO 9001-9002
C e r t i f i e d

SW SEI - CMM
Level **4**
COMPANY

Acknowledgement

Acknowledgement

I would like to express my sincere gratitude to a few who have contributed a lot towards the completion of my project.

I express my sincere thanks to **Dr. K.K.Padmanabhan, Ph.D.,** Principal, Kumaraguru College of Technology for extending his kind cooperation.

I would like to extend my sincere thanks to **Dr.S.Thangasamy, Ph.D.,** Head of the department, Computer Science and Engineering for allowing me to take up this project at KGiSL, Coimbatore.

I extend my sincere thanks to **Mr. Muthukumar, MCA., M.Phil.,** Course Coordinator, Master of Computer Applications, Kumaraguru College of Technology, for his invaluable support and encouragement.

I owe a deep debt of gratitude to my guide **Ms. L.S. Jayashree, M.E.,** Department of Computer Science and Engineering, Kumaraguru College of Technology, for her outstanding guidance throughout the project period.

I also extend my gratitude to **Mr. Robin Rathan Kumar, M.Phil.,** and **Mr. Ananthan, M.B.A.,** of **KGiSL Systems Limited** for allowing me to carry out this project at their organization and guiding me in completing it successfully.

Above all I would like to thank the **LORD Almighty** for all His invaluable blessings and guidance.

Synopsis

SYNOPSIS

In the emerging world of networking it is a hectic task to monitor software and hardware in each and every system in the network. In order to trace their existence, hardware engineers have to manually check the entire network and then consolidate the report. This is an enormous task as our concern has around 500 systems. The whole process takes nearly a week's time in generating the required report. Similarly identify the various users in the entire network, their login and logout time irrespective of platform (Win NT /Sun Solaris / Linux) they use is not easily traceable. So there arises the need for our system that satisfies the above needs.

The Planned system comprises of Client and Server Component.

Client Component:

The required information like (current users, login, logout, software & hardware details) are fetched using Shell script and written into a text file.

Using Java Socket, File and Streams, text file transferred from the client machine to the server.

Server Component:

Using Server Socket, File, Streams and Oracle Driver Connectivity the transferred information from the client is sent to the Database. Running a Java web server / Java Servlet the stored information is viewed as HTML. This enables the NT user to view information of the Linux/Solaris counterpart and vice versa. The accessibility to this NMS should be restricted with a login and password.

Contents

CONTENTS

5. System Testing & Implementation

Introduction

1. INTRODUCTION

1.1 PROJECT OVERVIEW

NMS monitors the network activities of client organization KG Information Systems Private Limited. The objective of the NMS is to monitor systems connected to the network, which involves notifying the login & logout time, Hardware & software details, process details of each individual node in the network.

There are four different modules in NMS.

- Module1 – Fetching Information

  The information is read from log files using Shell scripts from all Linux client machines and written into a text file.

- Module2 – Transferring Information

  The stored text file from the network server is transferred to a database server. The network program is written using Java Sockets. Since the data transmission can take place at any point of time a server socket is made to execute in the background using threading.

- Module3 – Storing Information

  The information received via the server socket is stored in the database so that it can be referred at later stage. Using oracle driver for Java the received information is sent to the Oracle database.

- Module4 – Retrieval and Display of Information

  Using Java Servlet, the stored details are presented to the user in html format. The accessibility to this Network Monitoring System is restricted with a login and password.

1.2 COMPANY PROFILE:

KGiSL is a software development and IT training company, which offers IT solutions and services.

MILESTONES

- 1996 - KGISL started off - Software training & development - at KG House

- 1997 - Strategic tie-up with industry leaders - Tata Consultancy Services, Infosys, Satyam and EDS

- 1997 - HIS, US identifies KGISL as their service provider in India

- 2000 – Joint Venture with Heartland Information Services for Medical Transcription, Tie Up with ORACLE Corporation, USA.

INFRASTRUCTURE

- 1,00,000 sq. feet Software Development Campus

- 800 IBM/Acer Computers housed in the campus

- 2 mbps IPLC Earth Station in our campus - STPI Installation

SECTORS

- Automotive/Engineering

- Healthcare

- Insurance services

- Banking

- Financial Services

SERVICES

- Customized ERP solutions

- Conversion from one platform to another

- Maintenance of existing software systems

- Networking solutions

- Reengineering services

- Engineering design services

- Client Server Solutions

- Web Design, web site registration and maintenance

- Web enabling

- Web Content Development

- Web Infrastructure building & support services

CLIENTS

- TESCO Engineering, Michigan, USA - one of the biggest manufacturer of automotive door panel systems

- We have developed a customisable ERP solution for their operations

- Swift Alliance, USA

- Allied Engineering Services, USA

- TAGNET, USA

System Study and
Analysis

2. SYSTEM STUDY AND ANALYSIS

2.1 EXISTING SYSTEM

The existing system depends on the external interface Open NMS to retrieve the data hence Open NMS and the data related to them should be made available in the server.

LIMITATIONS:

Hardware Limitations:

The existing system server requires minimum of 512 MB RAM and 64 MB RAM at the client node for configuring the Open NMS external interface and client usage respectively.

Software Limitations:

The existing system is dependent on Tomcat 4 Web Server, which should be installed in Linux 7.1 platform. The reliability of the system depends on the performance of PostGreSQL7.1.3 and the Tomcat 4-web server.

Environmental Constraints:

The system is dependent on Linux 7.1 operating system.

NEED FOR CHANGE:

Since the existing system is dependent on an external interface, which is not a licensed product, the possibility for change of system is imposed.

2.2 PROPOSED SYSTEM:

ADVANTAGES:

- The new system is not dependent on any external interface.

- There are no Site Adaptation Requirements that are specific to a particular location. The client can use the system from any platform.

- NMS is web enabled, it is available for 24Hrs.

- The NMS is developed in such a way that the user can understand within 4 hours and easily operable.

- NMS is designed in a user-friendly manner so that the user of the system will be clear bout the functionality of NMS.

- The screens of NMS are designed in such a way that it attracts the user.

- The NMS is developed so that it can be modified in future for further enhancement

2.3 REQUIREMENT OF NEW SYSTEM:

2.3.1 Functional Requirement 1 – User detail View

Introduction / Purpose

NMS has provision for the system administrators and top management to view the login and logout time of each node in the network of KG Information Systems Private Limited.

Stimulation / Response Sequence

Stimulus: User detail View

Response: View of login and logout time of each individual node in the network of KG Information Systems Pvt. Ltd.

Input

The input to the NMS to view the user details is the node name.

Processing

NMS provides the user details of each individual node in the network. The data like login time, System name, Ip_address and user_id are being fetched from individual nodes. Upon fetching the data, it is sent to the database and is displayed to the end user on request.

Output

View of login and logout time of each node in the network of KG Information Systems Private Limited is displayed as follows:

- Default user

- Login time

- System name

2.3.2 Functional Requirement 2 – Hardware & Software Details

Introduction / Purpose

NMS has provision for the system administrators and top management to view the hardware and software details of each node in the network of KG Information Systems Private Limited.

Stimulation / Response Sequence

Stimulus: Hardware & Software Details

Response: View of hardware and software details of each individual node in the network of KG Information Systems Pvt. Ltd.

Input

The input to the NMS to view the hardware and software details is the node name.

Processing

NMS provides the hardware and software details of each individual node in the network. The data like system name, hardware interfaces, memory details, and software installed are being fetched from individual nodes. Upon fetching the data, it is sent to the database and is displayed to the end user on request.

Output

View of hardware and software details of each node in the network of KG Information Systems Private Limited is displayed as follows:

- System name
- Hardware Interfaces
- Memory details
- Software Installed

2.3.3 Functional Requirement 3 – Process details

Introduction / Purpose

NMS has provision for the system administrators and top management to view the process details of each node in the network of KG Information Systems Private Limited.

Stimulation / Response Sequence

Stimulus: process details

Response: View of process details of each individual node in the network of KG Information Systems Pvt. Ltd.

Input

The input to the NMS to view the process details is the node name.

Processing

NMS provides the process details of each individual node in the network. The data like processid, systems name and process names are being fetched from individual nodes. Upon fetching the data, it is sent to the database and is displayed to the end user on request.

Output

View of process details of each node in the network of KG Information Systems Private Limited is displayed as follows:

- Process id
- Process name
- System name

2.4 USER CHARECTERISTICS:

It is intended that the users of NMS should have the basic knowledge and understanding of Internet. The administrative user should have the knowledge about the networks and its functionality.

*Programming
Environment*

3. PROGRAMMING ENVIRONMENT

3.1 SYSTEM CONFIGURATION:

Hardware Specification:

Processor type	:	Intel Pentium III
RAM	:	128 MB
Hard Disk Space	:	20 GB
Network	:	Ethernet 10 MBPS

Software Specification:

Operating System	:	REDHAT LINUX
Tools used	:	Shellscript, Java Socket, Java Servlet, HTML and Java Script.
Database	:	Oracle 9i

3.2   OVERVIEW OF SOFTWARE:

Shell scripts:

A shell program is a sequence of one or more commands stored in a file. This file is executable and can be called as shell procedure. The shell program has some built in commands that can be used in a shell script. In Linux a shell script has the same status and can be executed in the same way compiled and linked program can. The permissions should be set to allow execution

Java Socket:

A socket is a communication endpoint — an object through which a socket application sends or receives packets of data across a network. Clients and servers establish connections and communicate via socket. Sockets are endpoints of Internet communication. Clients create client socket and connect them to server socket. Sockets are associated with a host address and a port address. The host address is the IP address of the host where the client or server program is located. The port address is the communication port used by the client or server program. Server programs use the well-known port number associated with their application protocol.

Socket allows a single computer to serve many different clients at once as well as serving many different types of information a port manages this, which is a numbered socket on a particular machine. A server process is said to listen to a port until a client

13

connects to it. A server is allowed to accept multiple clients connected to the same port number, although each session is unique. To manage multiple client connections, a server process must be multithreaded or have some other means of multiplexing the simultaneous input / output. Socket supports two protocols as TCP/IP and UDP. Our project uses TCP/IP protocol.

In our project a file is read in the client socket and transferred to the database via server socket. The server socket accepts the client request, reads the file and transfers then into a database.

TCP/IP protocol:

The TCP is a connection-oriented protocol. It establishes a connection link between a source port/IP address and a destination port/IP address. The ports are bound together via this link until the connection is terminated and the link is broken. TCP implements the connection as a stream of bytes from source to destination.

DATABASE:

Database is a collection of data that is organized so that it may be easily searched and updated. In our project the file contents is stored in database at specific locations for future references. A database server is a software program that manages databases, keeps them organized, and provides shared access to them. Database servers manage and organize databases at both physical and logical level.

JDBC:

JDBC provides a common database- programming API for Java programs. JDBC API is based on the X/Open CLI (Call-Level Interface), which defines how clients and servers interact with one another when using database systems

```
        JAVA APPLICATION,
        APPLET, OR SERVLET

            JDBC API

              JDBC
        DRIVER MANAGER

   JDBC         JDBC         JDBC
  DRIVER       DRIVER       DRIVER

   ODBC        ORACLE       SYBASE
```
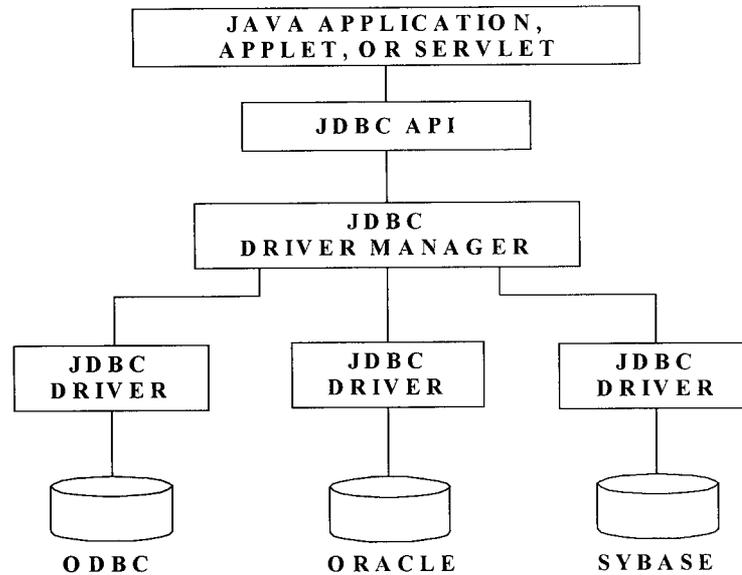
Figure 3.2.1 JDBC Interoperability.

The basic JDBC application follows the following flow:

1. Establish a connection to the database.

2. Execute a SQL statement.

3. Process the results.

4. Disconnect from the database.

15

Java Servlet:

Servlet are server side applets, which are initialized and executed by a Web Server. Servlet are fast and platform independent. The Web browser makes a request via an HTTP. The Web server receives the request and sends it to the servlet. The servlet will receive the HTTP request and perform task and return a response to the Web server. The Web server will forward the response to the client.

Servlet is chosen, since it is persistent, fast, platform independent, extensible, secure and can be used with a variety of clients.
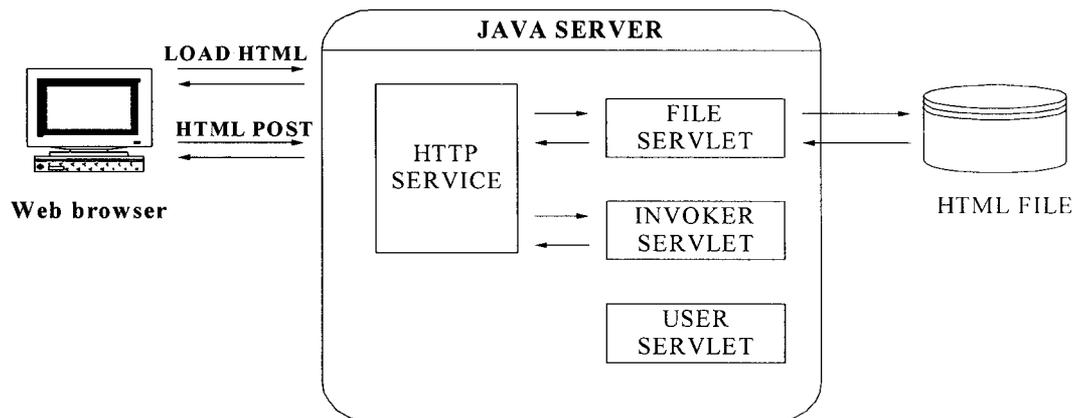

Figure 3.2.1 – Java Servlet.

The figure 3.2.1 illustrates, the client (Web Browser) will make a request to the server to load an HTML page. The HTTP Web service within the server will receive the request, recognize that it is an HTML file read request, and invoke the file servlet to actually perform the file I/O. The HTML page will then be returned to the client and displayed within the web browser. If the web browser makes an HTML POST request,

16

the HTTP Web service will again receive the request. If the POST requires that a servlet be loaded, the request is forwarded to the invoker servlet, which will then invoke the servlet. The servlet then does some types of processing and returns data back to the client via HTTP.

HTTP protocol:

Web server implements the Hyper Text Transfer Protocol in order to retrieve Web resources identified by URLs. HTTP is an application-level protocol that is designed to be quick and efficient. It is based on the request-response paradigm. Web browsers initiate connections with Web servers and submit service requests. The servers, upon receiving a request, locate the specified resource and perform the request operation. HTTP supports several request types, referred to as methods. These include the GET, HEAD, and POST methods.

HTML & JavaScript:

HTML is a subset of SGML, which is used to embed commands that describe how different elements with in a document should be interpreted in the document. The HTML document can be read and displayed by Web browsers.

JavaScript is primarily a client side scripting language. It brings powerful, flexible scripting from the Web server to the Web client. It is supported across multiple browsers. It is used to customize Web pages for individual users.

17

System Design and Development

4.   SYSTEM DESIGN AND DEVELOPMENT:

4.1   INPUT & OUTPUT DESIGN:

The input to our system is the Information collected from the client machine, which is stored in the database. All these are hidden processes. The output screens are only visible and displayed to the Management Representative.

ENTRY SCREEN:

The management representative enters into the NMS through this screen. It is authorised with username and password fields, which are validated using JavaScript. On form submit, the details are verified with the database in a servlet, which on success leads to the detailed frame set screen. There is another option in the screen for change password.

ERROR SCREEN:

The error screen is displayed when authorization fails. This screen is same as the entry screen.

CHANGE PASSWORD SCREEN:

The screen consists of four text fields as username, old password, new password and conform password, which are validated using JavaScript. The username and old password fields are retrieved from the entry screen and is disabled. On submit the details are updated in the database.

18

FRAME SET SCREEN:

The screen is split into two frames. One screen displays the current login systems and the other displays the menu options. On click of the system name in the first frame, the menu screen is displayed.

HARDWARE DETAILS SCREEN:

On click of the hardware detail button in the menu screen, the details are retrieved from the database and displayed to the Management.

SOFTWARE DETAILS SCREEN:

On click of the software detail button in the menu screen, the details are retrieved from the database and displayed to the Management.

PROCESS DETAILS SCREEN:

On click of the process detail button in the menu screen, the details are retrieved from the database and displayed to the Management.

USER DETAILS SCREEN:

On click of the user detail button in the menu, the details are retrieved from the database and displayed to the Management.

4.2 MODULAR DESIGN:

There are four modules that are designed:

- Fetching Information
- Transferring Information
- Storing Information
- Retrieval and Display of Information

Fetching Information:

The information is collected from log files using Shell scripts from all Linux client machines and written into a text file. Information such as the hardware, software and process detail is collected when the client machine is started. The process details are fetched according to a timer that is set in the application.

Transferring Information:

The stored text file in the client machine is read in a client socket and written into a server socket. The data stream and file streams concepts of Java are utilized. The server socket accepts the information. Since the data transmission can take place at any point of time a server socket is made to execute in the background using threading. The application is multithreaded to accept any number of clients request.

Storing Information:

The information received via the server socket is stored in the database so that it can be referred at later stage. Using oracle driver for Java the received information is sent to the Oracle database. The JDBC concepts are implied in the server socket.

Retrieval and Display of Information

The stored information is read in a Java Servlet and presented to the user in HTML format. Java Script are embedded for validation. The accessibility to this Network Monitoring System is restricted with a login and password.

SYSTEM PROCESS ARCHITECTURE:

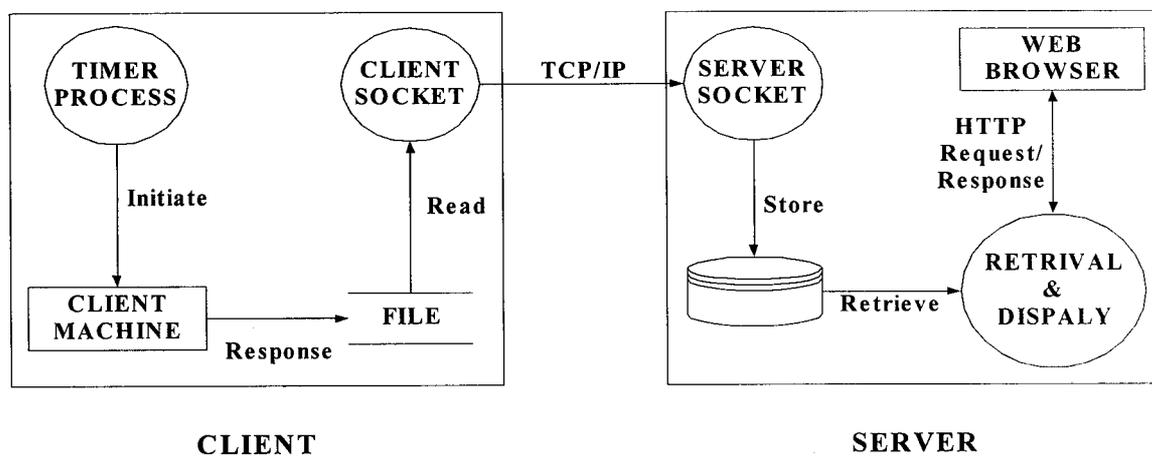CLIENT   SERVER

Figure 4.2.1 System Architecture.

The figure represents overall system architecture of the Network Monitoring System, which depicts the detailed process involved in client and server.

21

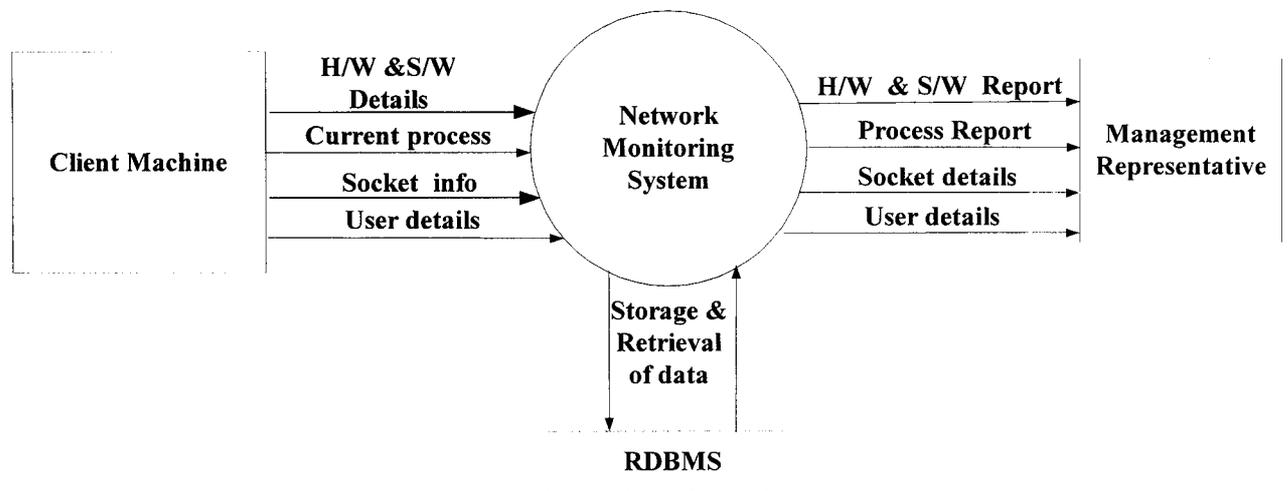## 4.3   PROCESS DESIGN:

Context Diagram:

Figure 4.3.1 – Context Diagram.

Figure 4.3.1 represents the overall process in the Network Monitoring System. The details are collected from the individual client machine and stored in the database, which is retrieved and presented to the Management Representative.

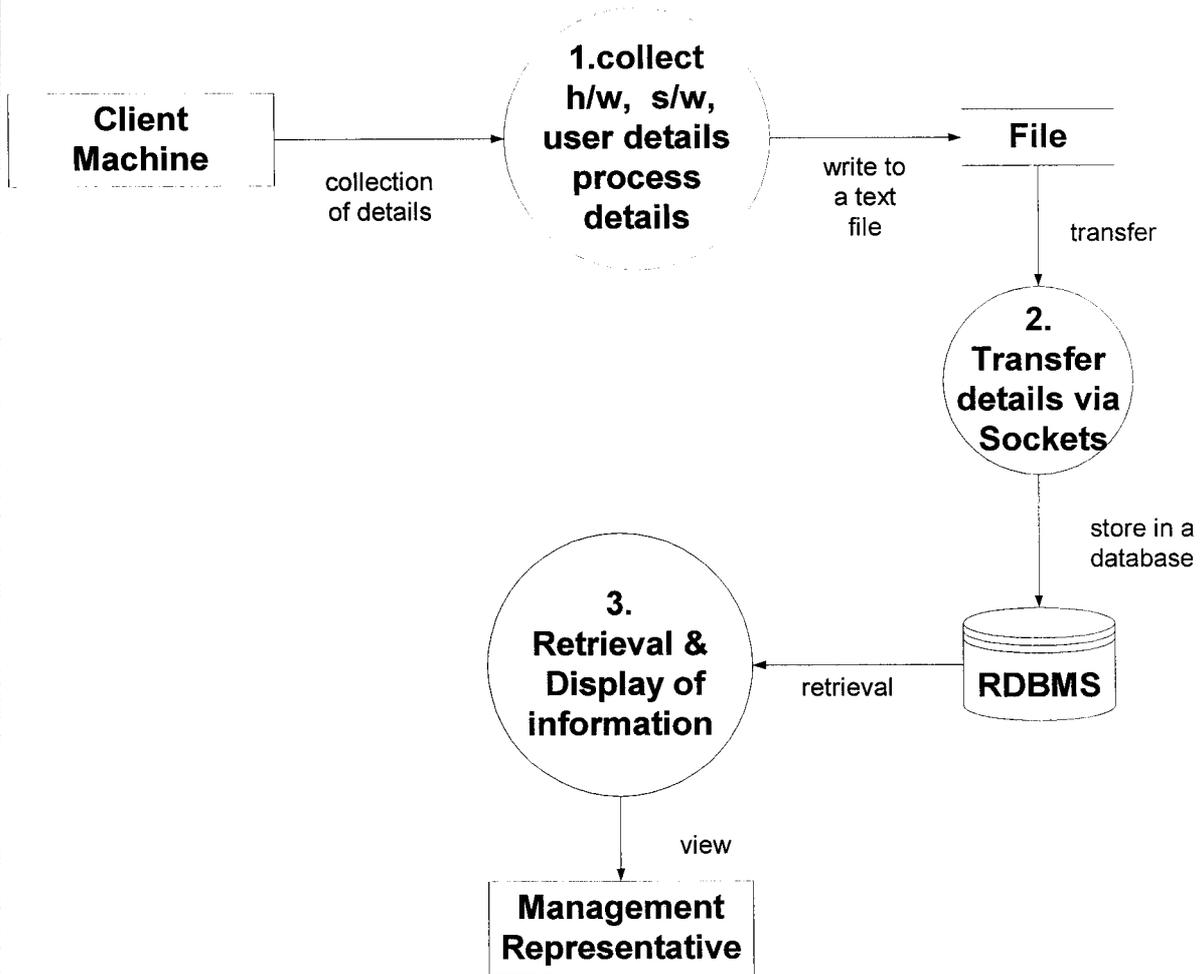**Data Flow Diagram (Level 1):**

Figure 4.3.2 - Level 1 DFD

Figure 4.3.2 depicts the flow of process in the system. When the user logs in into the client machine, the details are collected and stored as a text file. This text file is transferred and stored into a server database. On the management representative request, the details are displayed. Process and Snap shot details are collected with respect to the timer that is set.

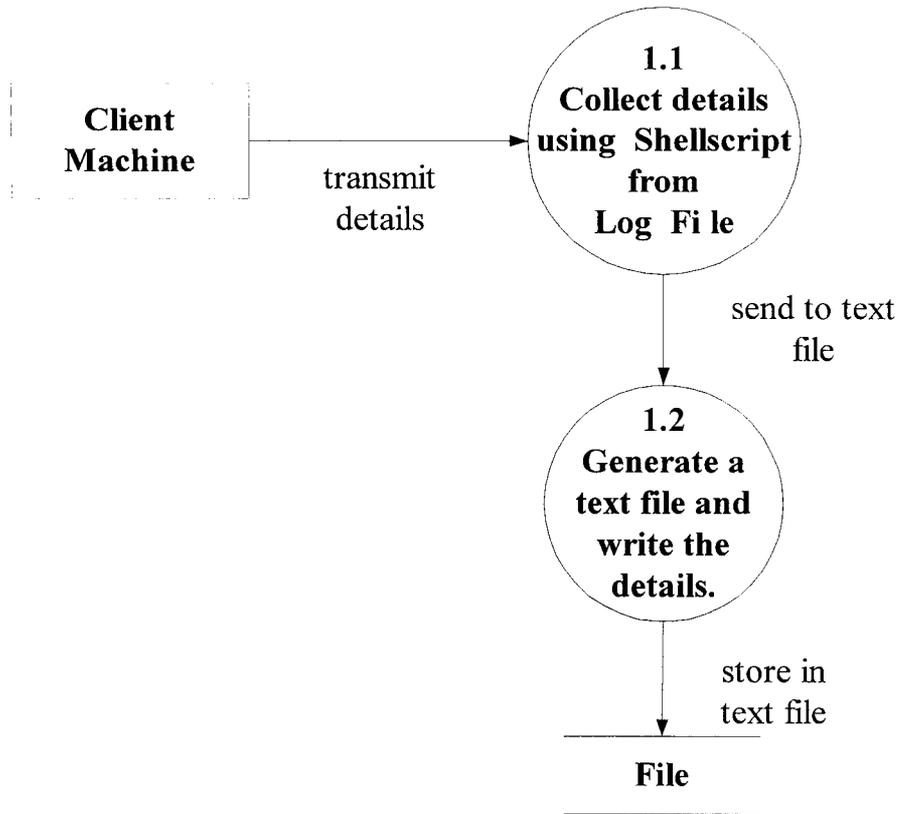**Data Flow Diagram (Level 2.1)**

Figure 4.3.3 – Level2.1 DFD

Figure 4.3.3 depicts the process involved in fetching the details in the client machine and storing it as a file.
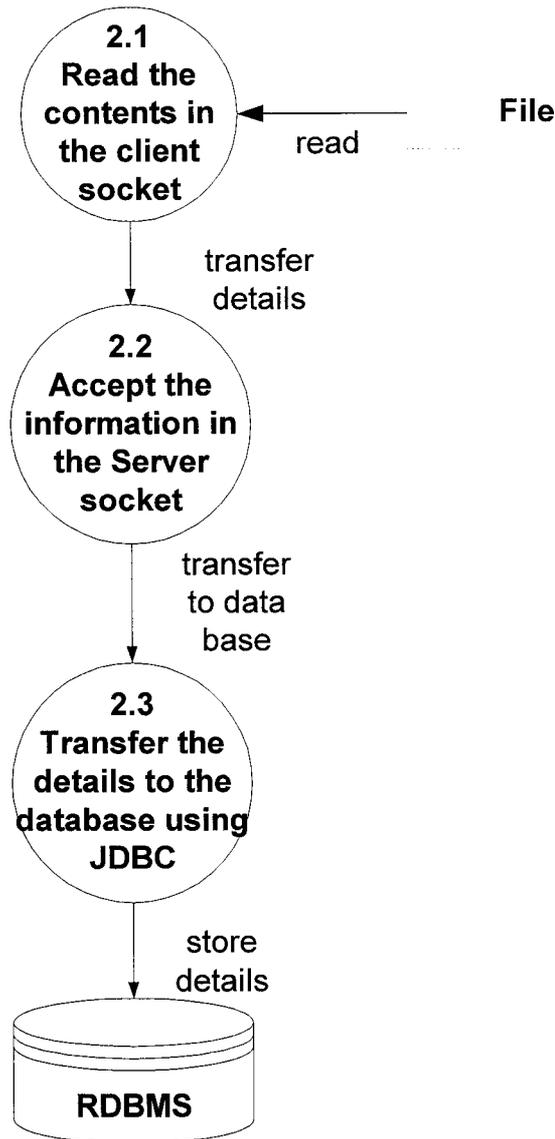
Data flow Diagram (Level 2.2)

Figure 4.3.4 - Level 2.2 DFD

Figure 4.3.4 depicts process involved in transferring the file from client to server machine using sockets. The details are the stored the server database for future reference.
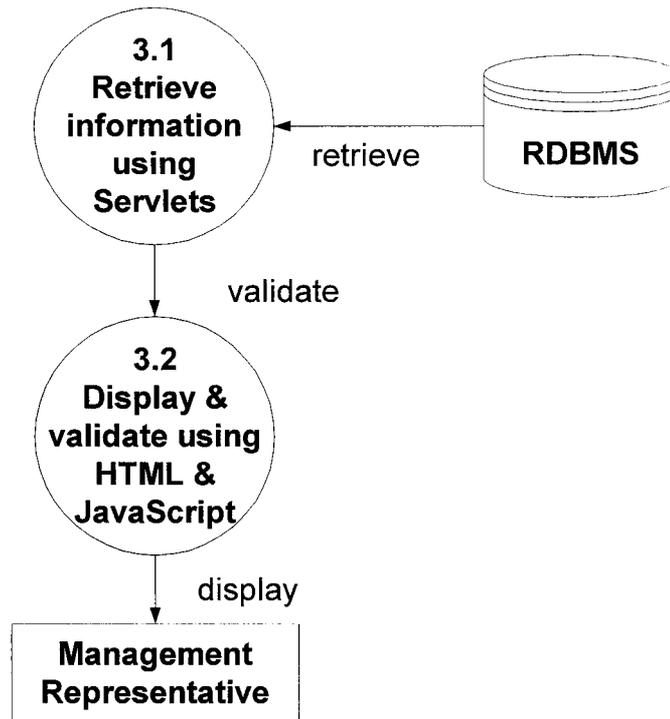
Data Flow Diagram (Level 2.3)

Figure 4.3.5 – Level 2.3 DFD

Figure 4.3.5 depicts process involved in retrieving the details from server database and displaying it as a HTML page to the management representative using Servlet concept, which includes validation.

4.4 DATABASE DESIGN:

4.4.1 TABLE DESIGN:

- *MASTER TABLES:*

    1. SYSTEM MASTER

    2. USER MASTER

- **TRANSACTION TABLES**

    1. H/W DETAIL

    2. S/W DETAIL

    3. LOGIN TRANSACTION

    4. PROCESS TRANSACTION

4.4.2   DETAILED TABLE DESIGN:

System Master:

Field name	Type	Length	Description
1. IP address	numeric	15	Primary key
2. Room No.	varchar2	05	
3. System No.	varchar2	05	

User Master:

Field name	Type	Length	Description
1. User Id	Varchar2	15	Primary Key
2. Password	Varchar2	06	

Transaction table

Hardware Details:

Field name	Type	Length	Description
1. IP address	numeric	15	Foreign key
2. Description	varchar2	50	

Process Transaction:

Field name	Type	Length	Description
1. IP address	numeric	15	Foreign key
2. PID	numeric	05	Primary key
3. Login	numeric	25	Foreign key
4. Starttime	date/time		
5. Endtime	date/time		
6. Uptime	date/time		
7. Memused	numeric	05	
8. Cpuused	numeric		
9. Terminalno	varchar2	05	
10. Command	varchar2	10	

Software Details:

Field name	Type	Length	Description
1. IP address	numeric	15	Foreign key
2. Description	varchar2	50	

Login Transaction:

Field name	Type	Length	Description
1. IP address	numeric	15	Foreign key
2. Login	varchar2	50	Primary key
3. workingshell	varchar2	05	
4. Directory	varchar2	10	
5. Quotaused	varchar2	10	
6. Intime	date/time		
7. Outtime	date/time		

4.4.3   ENTITY RELATION DIAGRAM:
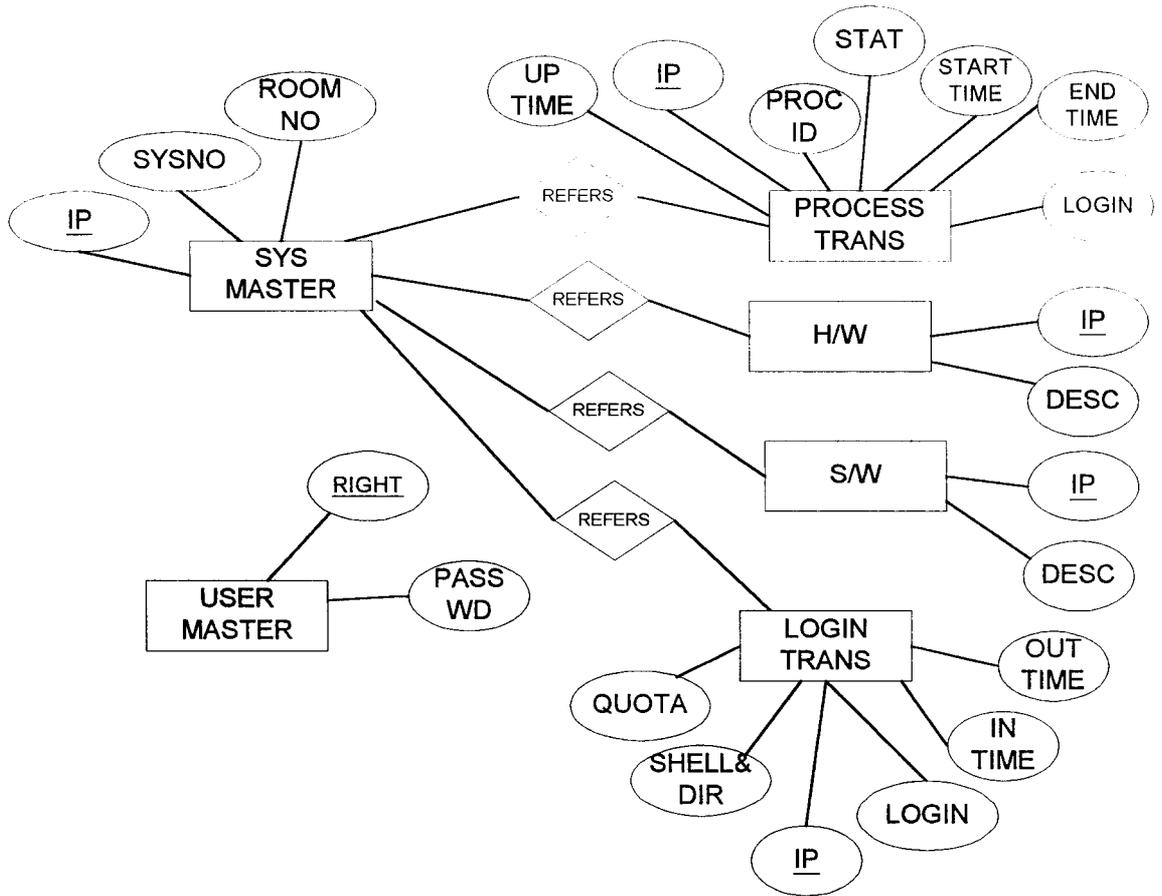

Figure 4.4.3 ERD

Figure 4.4.3 depicts the entity relation ship diagram of the system.

System Testing and
Implementation

5.   SYSTEM TESTING & IMPLEMENTATION

5.1   SYSTEM TESTING:

Testing is a process of executing a program with intent of finding errors. During testing, the program to be tested is executed with a set of test cases and the output of the program is evaluated to determine if the program is performing as expected.

TEST CASES:

Computer software considers two types of test cases as one to test the specific function of the product called as black box testing and the other to test the internal working of the product called as white box testing.

Black box testing examines some functional aspects of a system with little regard for the internal logical structure of the software. It enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. It attempts to find errors in categories as incorrect or missing function, interface errors, errors in data structures or external data base functions, performance errors and initialization / termination errors. Black box testing is applied during the later stages of testing, since it disregards control structure and focus on information domain.

White box tests focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been exercised. White box testing uses the control structure of the procedural design to derive test cases. It can derive test cases that guarantee that all independent paths with in a module have been exercised at once,

exercise all logical decisions on their true and false sides, execute all loops at their boundaries and within their operational bounds and exercise internal data structure to assure their validity.

These test cases are applied to achieve more complete testing by uncovering and correcting the highest number of errors in the program.

TEST TECHNIQUES:

Having test cases that are good at revealing the presence of false is central to successful testing. Ideally a set of test cases is to be determined such that successful execution of all implies that there are no errors in the program. Each test case needs more effort, machine time to evaluate the results. One possible ideal set of test cases is one that includes all the possible inputs to the program.

Unit Testing:

Unit testing focuses on verification effort on the smallest unit of software design of the module. Using the procedural design description as a guide, important control path are tested to uncover errors within the boundary of the module. The relative complexity of the tests and uncovered errors in limited by the constrained scope established for unit testing. The unit test is normally white box oriented. It is considered as an adjunct to the coding step.

Integration Testing:

Integration testing is a systematic technique for constructing the program structure while conducting test to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design.

Validation Testing:

It is a final series of software testing. Validation succeeds when software functions in a manner that can be reasonably expected by the customer. It is achieved through a series of black box test that demonstrate conformity with requirements. The test plan and test procedure are designed to ensure that all the functional requirements are satisfied, all performance requirements are achieved, documentation is correct and other requirements are met.

System Testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all system elements have been properly integrated and performed allocated functions.

Testing Client Server Environment:

The nature of client server systems poses a set of unique problems. Testing of client server environment includes client GUI consideration and target environment. The

client software was tested in disconnected mode the network was manually disconnected and the client was run to test whether proper error message is displayed.

Next the client and associated server applications are tested in concrete, but network operations are not exercised explicitly. Finally the client server architecture was tested including network operation and performance was tested.

Apart from these tests, a series of tests are conducted to ensure that each class of transaction is processed according to requirements. Also tests are conducted to verify the communication among the nodes of network occurs correctly and the message passing, transactions and related network traffic occurs without errors.

The software is allowed in a network environment of 10 clients where all the clients transmit data at a time to the server. The system performs well with some increased latency.

5.2   SYSTEM IMPLEMENTATION:

The server software is installed on the machine designated as server for the system and the client software on another machine. Installation of both the software does not require any additional components. This software uses only the existing components of the operating system. The server software is started when the machine is turned on but can be posted any time. The client software also starts like the server but cannot be stopped explicitly. When the server application is closed, the user might not be able to monitor and consequently manage the clients. The server software can be made to run in background but client run only in background.

Conclusion

6. CONCLUSION

Network Management System for Window Environment is developed to achieve major network management goals in a nutshell and provide the user with an interactive user interface.

The designed tool helps in providing the needed parameters on a single click without verbosity. It eliminates the tedious work of maintaining and updating the configuration details of the network nodes manually and the threat of its confidentiality is discarded by using login restrictions for the designed tool. The system tool if hosted on web server provides location independent monitoring of network, which may server as a great boon for management executives in this shrinking world. Most of the disadvantages of the existing system are overcome with these added features.

Scope for Enhancement

7. SCOPE FOR ENHANCEMENT

1. The system can be incorporated with

   - Help desk management in the system.

   - Online Tracking system.

   - Problem posting and solving capabilities.

2. Provide facilities for online execution of an application in the individual client machine.

3. Provide restricted view facilities for all the company employees.

4. Provision for online Reporting facility of job details and job scheduling constraints.

So that to ensure a good utilization of NMS. In this area further work is to be carried out.

Glossary

8.   GLOSSARY

Client is an entity of client-server architecture that will request for a service.

Server is an entity of client- server architecture that will respond to a service.

Client-Server is architecture for developing application. It is a request-reply based system where client will request to a server and the server will respond.

Host name is the name given to all client machines in the Windows NT network.

IP Address is a unique 32-bit identifier for a client in the TCP/IP network. It is also called as Dotted Decimal Number (DDN).

TCP (Transmission Control Protocol) is another protocol in TCP/IP suite, this protocol provide a connection oriented, reliable transmissions of data packet.

IP (Internet Protocol) is a protocol in the TCP/IP Protocols suit. This provides a connectionless, unreliable transmission of data packets.

HTTP (Hyper Text Transfer Protocol) is a protocol used to link various HTML pages in a Web page.

Web Browsers provide a window on the World Wide Web and support the display of Web pages, including Java programs.

Work Group Name is the name given to a group of clients in the network environment. Every client will have a Work Group Id to identify the group to which it belongs.

Bibliography

9. BIBILOGRAPHY

Books

1. Douglas.E.Comer, *"Internetworking with TCP/IP II "*, Second Edition, Prince-Hall, 2001.

2. Jamie Jaworski, *"Java 2 Unleashed "*, Techmedia, 1999.

3. Richard C. Leinecher, *"The Visual C++ Programmers Reference "*, Comdex Publishing.

4. Karl Moss, *"Java Servlet II Edition"*, Tata McGraw-Hill Edition.

5. Shelly Powers, et al, *"Dynamic Web Publishing "*, Techmedia.

6. Roger.S.Pressman, *"Software Engineering – A Practical Approach"*, Tata McGraw-Hill Edition.

7. James Goodwill, *"Developing Java Servlet"*, Third Edition, Techmedia.

Web Sites

8. http://www.java.sun.com, Sun Micro Systems.

9. http://www.jguru.com, Jguru.

10. http://www.allapi.net, Microsoft.

11. http://www.servlets.com, Sun Micro System.

12. http://www.coolservlet.com, Cool Servlet.

13. http://www.webreference.com, Techmedia.

14. http://www.javaworld.com, Java World.

15. http://www.osborne.com, OsBorne.

16. http://www.applicationdevelopmentadvisor.com, Lycos.

17. http://www.winapi.conclase.net, Microsoft.

Appendix

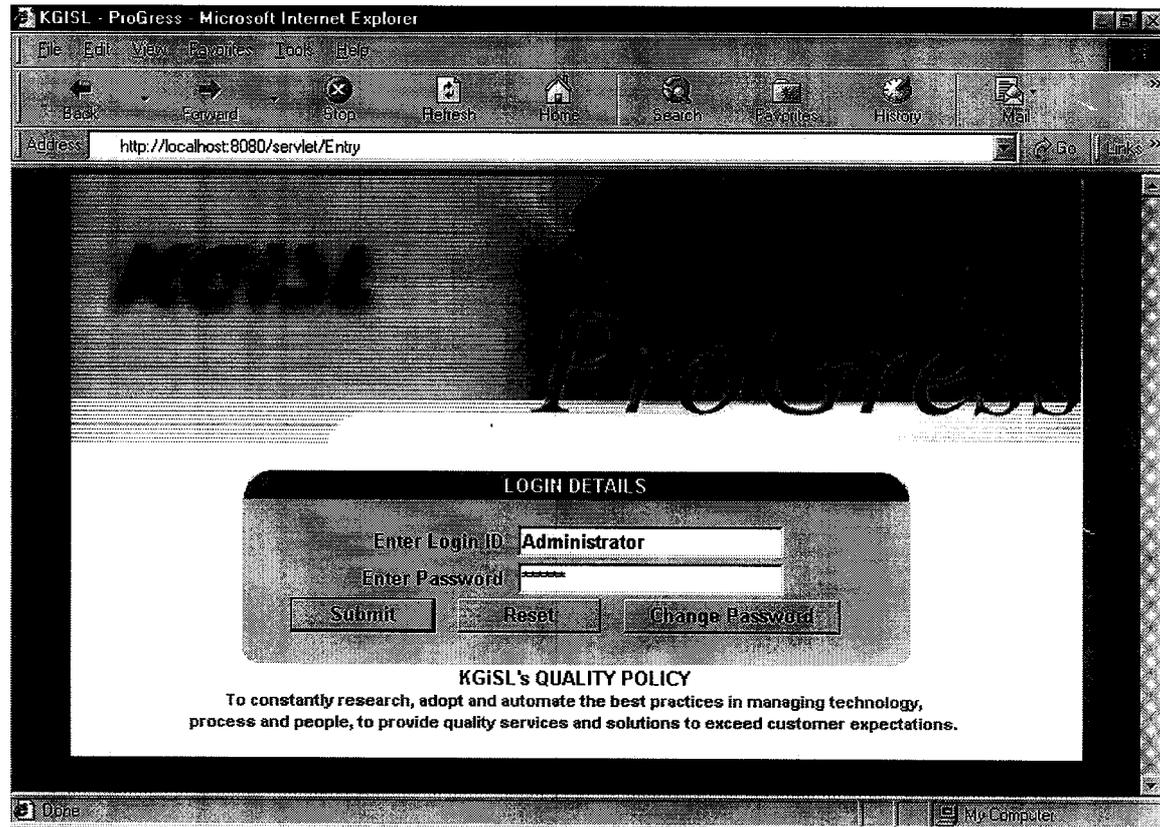# 10. APPENDIX

LOGIN SCREEN:

Figure 10.1 – Entry Screen.

Figure 10.1 depicts the entry screen of Network Management System, which is the main entry for the Management Representative.

ERROR SCREEN:

Figure 10.2 - Error screen

Figure 10.2 screen is displayed once there is an error in the user name or the password of the Entry screen.

CHANGE PASSWORD SCREEN:
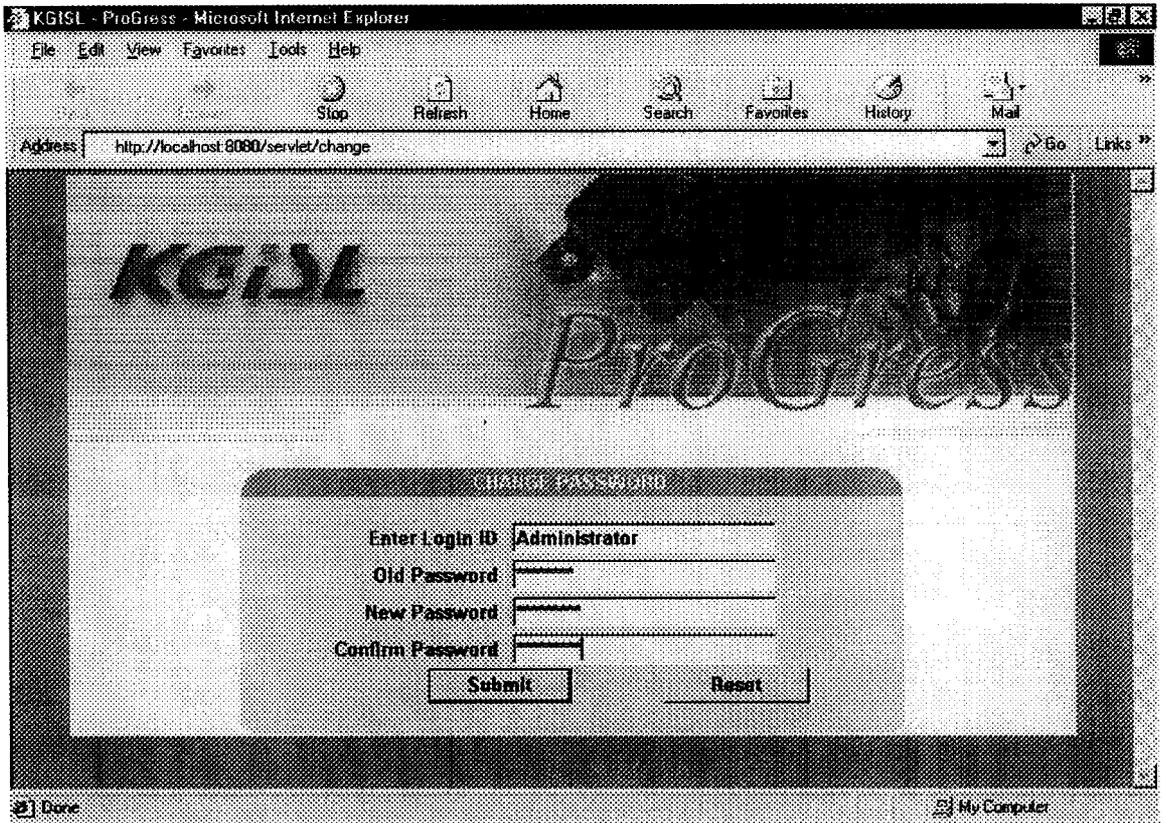
Figure 10.3 - Change Screen.

Figure 10.3 screen is displayed when the change Password option is opted in the entry screen.

FRAMESET SCREEN:
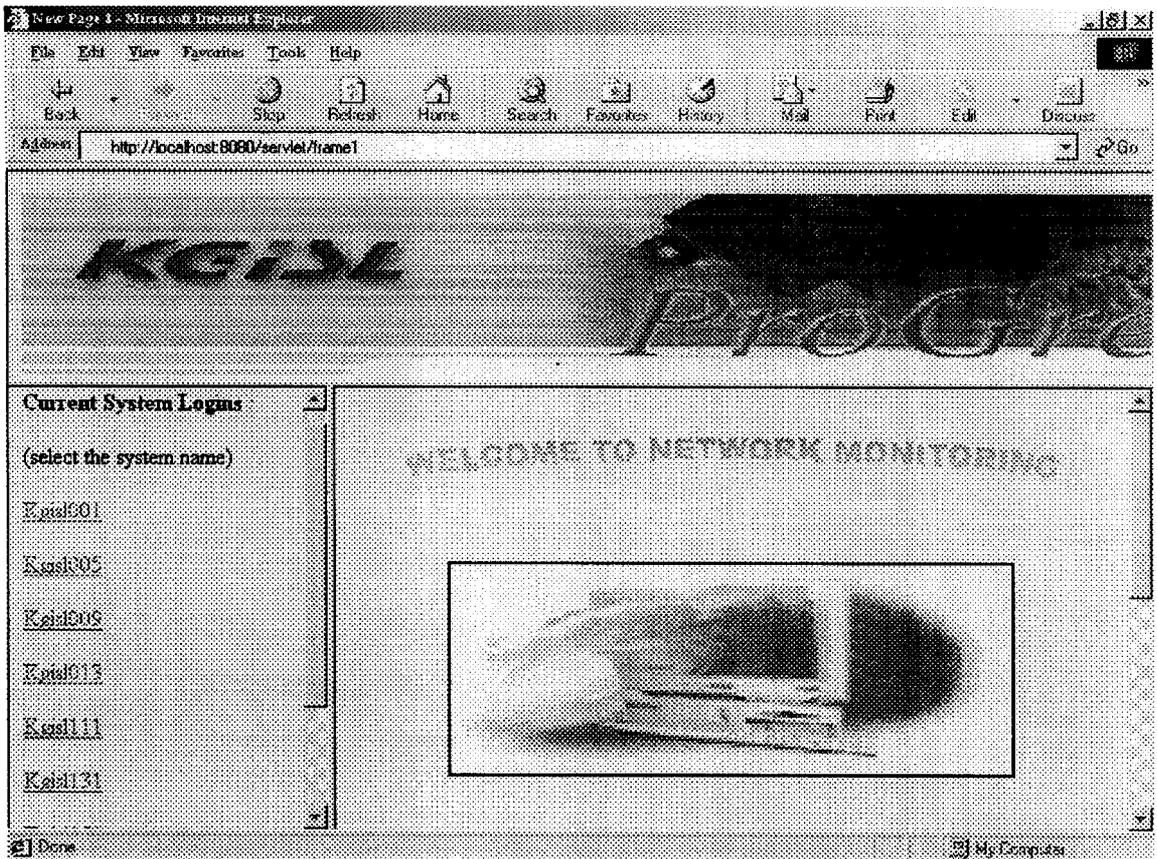
Figure 10.4 - Frameset screen.

Figure 10.4 screen represents the frame set which displays the currently logon systems names.
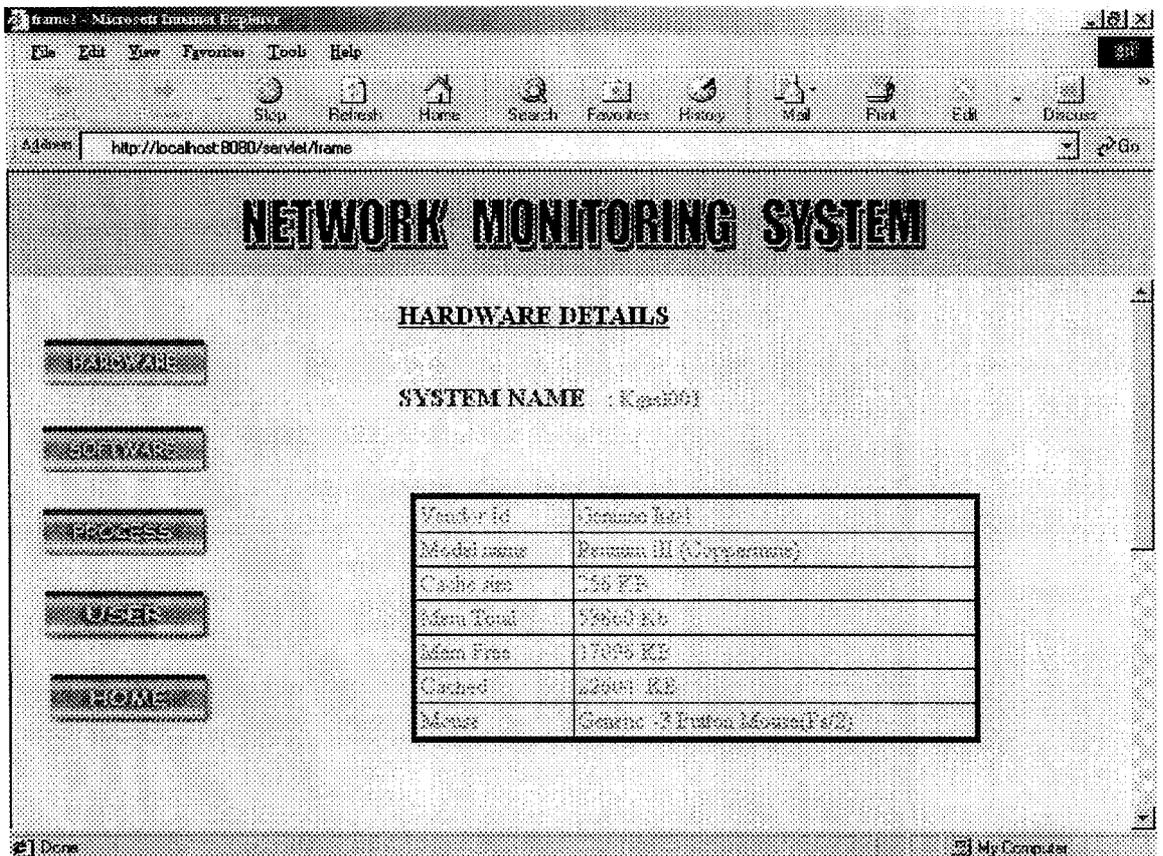
HARDWARE DETAILS SCREEN:

Figure 10.5 - Hardware details screen.

Figure 10.5 screen represents the hardware details of the respected system selected.
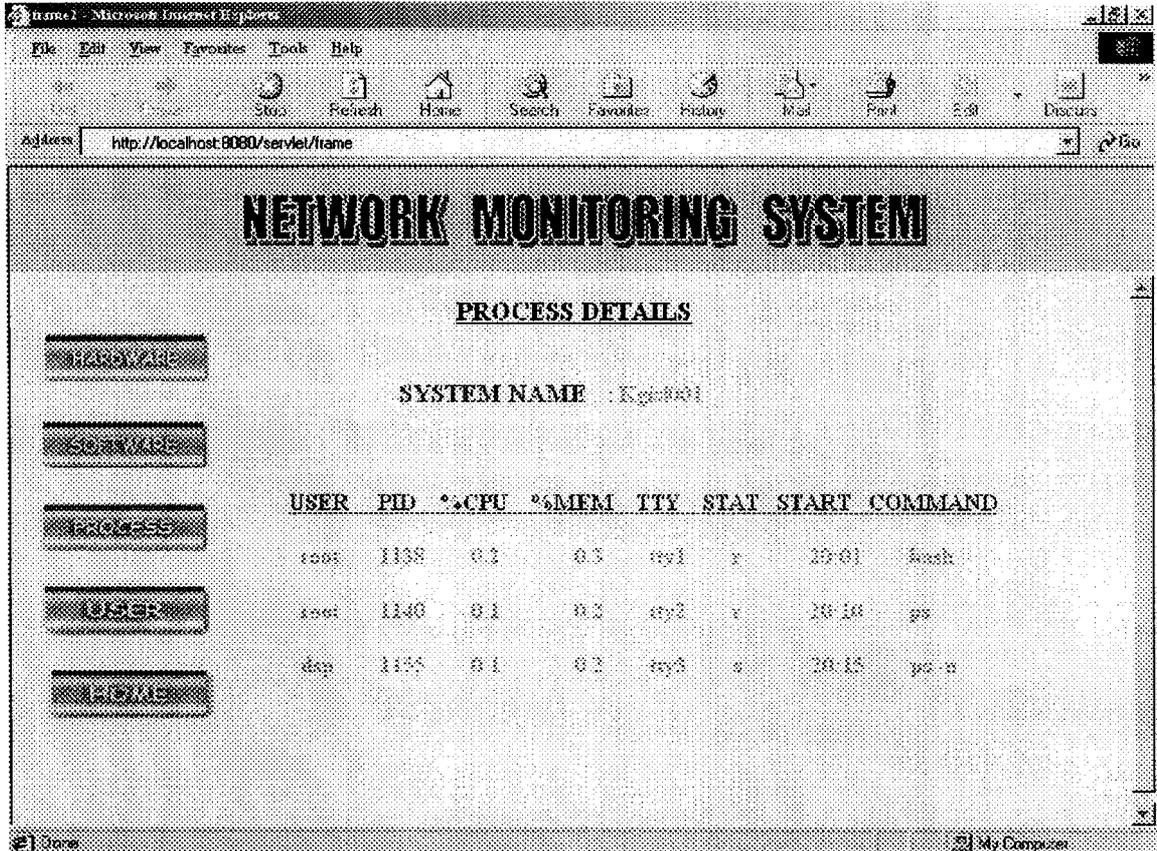
PROCESS DETAILS SCREEN:

Figure 10.6 - Process details screen.

Figure 10.6 screen represents the process details of the respected system selected.
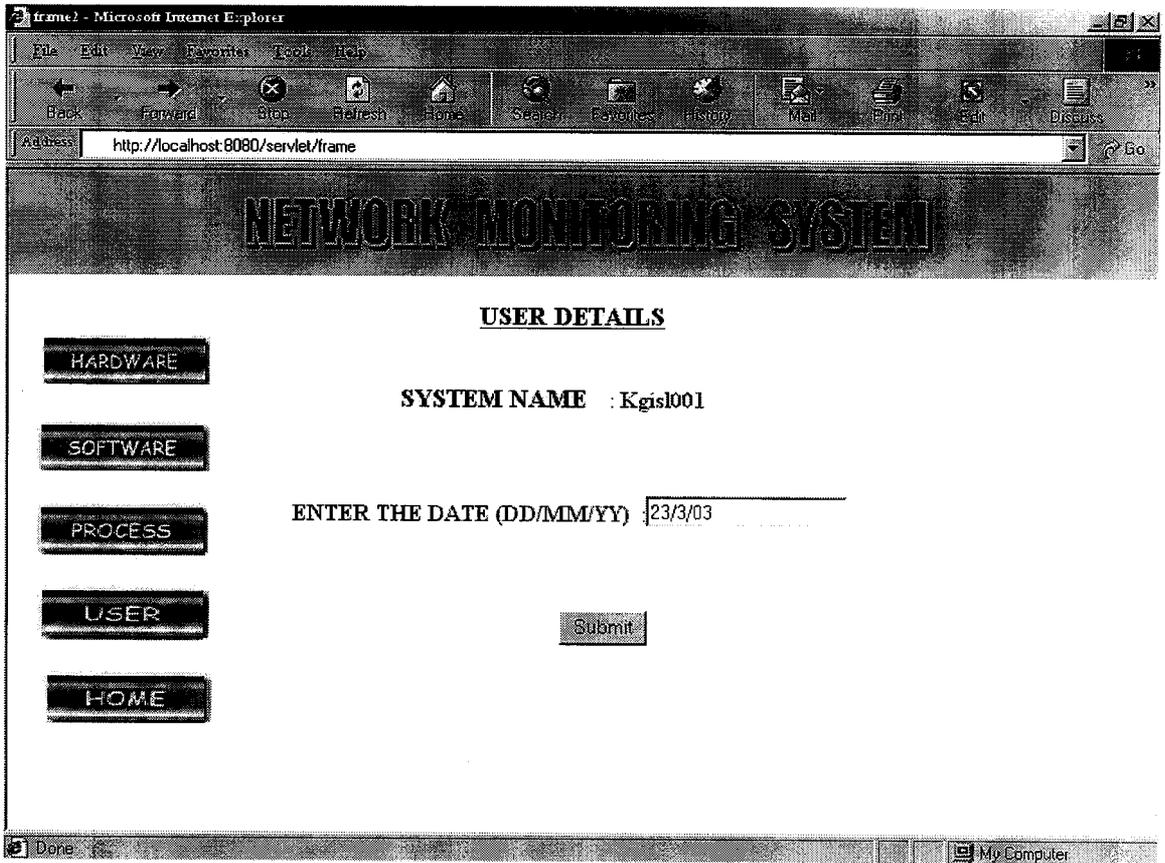
USER DETAILS SCREEN:

Figure 10.7 – User details screen.

Figure 10.7 - screen represents the user details of the respected system selected.
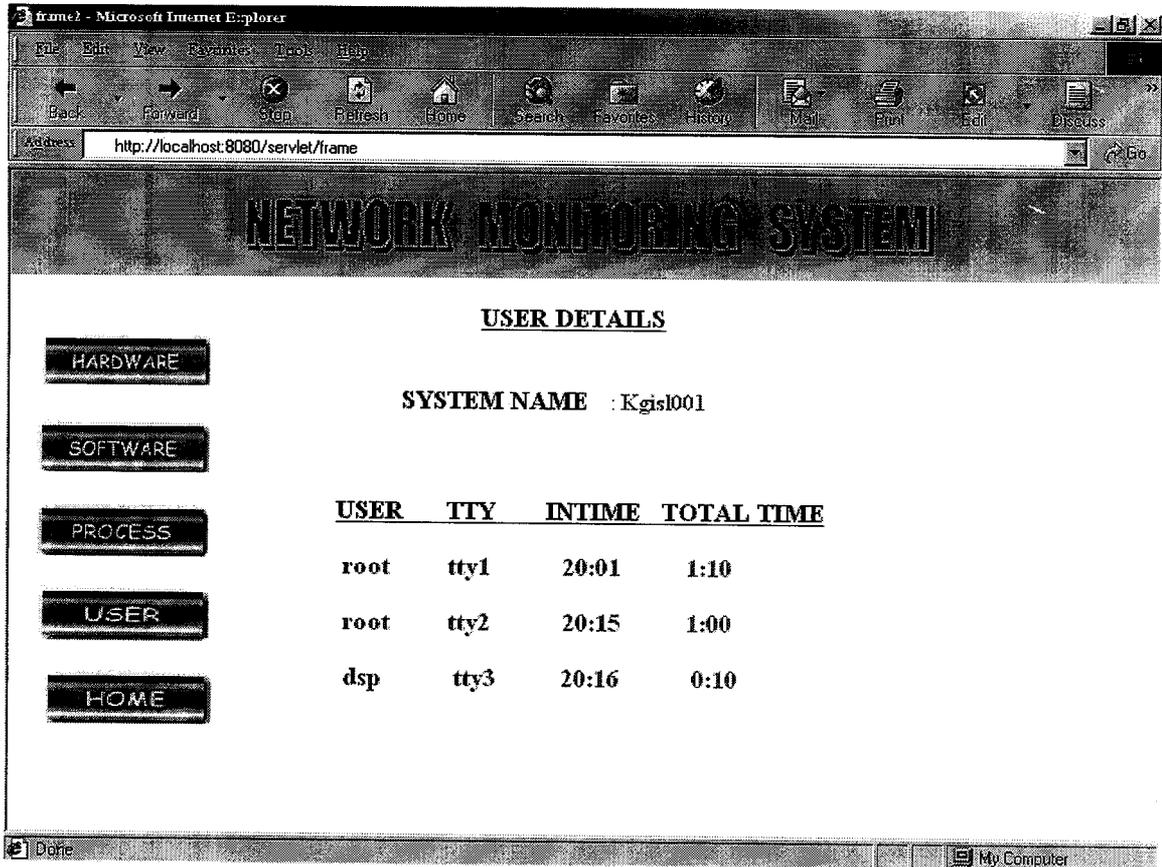
USER DETAILS SCREEN:

Figure 10.8 – User details screen.

Figure 10.8 - screen represents the user details of the respected system selected.