ASSET MANAGEMENT TOOL FOR WINDOWS

A PROJECT REPORT

Submitted by

GOPALAKRISHNAN.G (71202104011)

VIJAYAKUMAR.B (71202104050)

VINODH.K (71202104052)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCEINCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2006

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

BONAFIDE CERTIFICATE

Certified that this project report **"ASSET MANAGEMENT TOOL FOR WINDOWS"** is the bonafide work of **"GOPALAKRISHNAN.G (71202104011), VIJAYAKUMAR.B (71202104050) and VINODH.K (71202104052)"** who carried out the project work under my supervision.

SIGNATURE

Dr. S. Thangasamy
HEAD OF THE DEPARTMENT

Department of Computer Science & Engg.

Kumaraguru College of Technology
Chinnavedampatti Post
Coimbatore – 641 006

SIGNATURE

R. Dinesh
SUPERVISOR

Assistant Professor

Department of Computer Science & Engg.

Kumaraguru College Of Technology
Chinnavedampatti Post
Coimbatore – 641 006

Submitted for viva-voce examination held on 2/5/06

INTERNAL EXAMINER

EXTERNAL EXAMINER

ANNA UNIVERSITY: CHENNAI 600 025

CERTIFICATE

EVALUATION CERTIFICATE

College : **KUMARAGURU COLLEGE OF TECHNOLOGY**

Branch : **COMPUTER SCIENCE AND ENGINEERING**

Semester : **EIGHT (08)**

S.No	Name of the Student	Title Of The Project	Name Of The Supervisor
1. 2. 3.	Gopalakrishnan.G Vijayakumar.B Vinodh.K	Asset Management Tool for Windows	R. Dinesh

The report of the project work submitted by the above student in the partial fulfillment for the award of BACHELOR OF ENGINEERING degree in COMPUTER SCIENCE AND ENGINEERING of Anna University was evaluated and confirmed to be the report of the work done by the above student and then evaluated.

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

DECLARATION

We hereby declare that the project entitled **"ASSET MANAGEMENT TOOL FOR WINDOWS"** is a record of original work done by us and to the best of our knowledge; a similar work has not been submitted to Anna University or any institution, for fulfillment of the requirement of the course study.

The report is submitted in partial fulfillment for the award of the Degree of Bachelor of Computer Science and Engineering of Anna University, Chennai.

Place : Coimbatore

Date : 27 04.06

G. Gopalakrishnan

(Gopalakrishnan.G)

(Vijayakumar.B)

(Vinodh.K)

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We would like to express my heartfelt thanks to Dr. K.K.Padmanaban Principal of Kumaraguru College of Technology, Coimbatore, for having permitted me to do this project.

With profound gratitude, we express our deepest thanks to our internal guide Mr. Dinesh R, Assistant Professor, Department of Computer Science and Engineering, who guided us throughout the project and taken all measures to ensure its successful completion and has been a source of inspiration and motivation for us.

We express our sincere thanks to all the lab assistants who have been very helpful in implementing our software.

We would like to thank the Head of the Department, Computer Science and Engineering, Dr. S. Thangasamy and Mrs. Devaki for guiding us through the project.

Our sincere thanks to the Department of Computer Science and Engineering, Kumaraguru College of Technology, for extending its fullest support by all means to enabled to complete the project.

Last but not the least; we extend our gratitude to all the student peers and our parents, and all those who directly or indirectly helped us in successful completion of the project

ABSTRACT

ABSTRACT

"ASSET MANAGEMENT TOOL FOR WINDOWS" is developed using Win32 API's under VISUAL BASIC.NET. The objective of this software is to monitor all the systems in the network and to retrieve all the information of the client systems and remote systems.

The main features of this product are as follows:

- **Network Management** - will retrieve all the systems ip address and the system ids that are connected in the network, and it allows the administrator to disconnect the systems in the network.
- **File Management** - will retrieve all the Files in each and every drives of the system that are present in the network.
- **File Explorer** - will retrieve all the information's about the particular files from the drives of the system that are present in the network.
- **Event Management** - will retrieve all the events that are handled by the system like log type, log message, log time, log source, from the corresponding system in the network

- **Media Management** - will allow the administrator to play all the media files from the server to clients.
- **Device Management** – will retrieve all the device information's like Driver name, device descriptions, device state and device type their status.
- **Process Management** – will retrieve all the process information's like process name, process id, CPU usage, RAM usage, and process priority about the corresponding system in the network.
- **Service Management** – will retrieve all the service information's like service name, service descriptions, and service state, start up about the corresponding system in the network.
- **Desktop Management**

  Chatting - will allow the administrator and the users to communicate each other.

  White board - will be used to explain any working process from the server as soon as when administrator performs text function it will be displayed in the client or in the remote system monitor.

  File transfer – will be used to transfer files from one system to other system.

  Desktop monitoring - will capture all the operations that are performed in the client or remote system.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

INTRODUCTION

1. INTRODUCTION

"ASSET MANAGEMENT TOOL FOR WINDOWS" is developed using Win32 API's under VISUAL BASIC.NET. The objective of this software is to monitor all the systems in the network and to retrieve the information of the client systems and remote systems.

The main features of this project are Network Management, File Management, File Explorer, Event Management, Media Management, Device Management, Process Management, Service Management and Desktop Management. The desktop management is further divided into the parts as Chatting, White board, File transfer and Desktop Monitoring.

The Existing Systems for network management are specially designed for monitoring all the client system information. This existing Softwares will only work on Windows NT 4.0 or in Windows Server 2003 and also they do not have an interactive Graphical User Interface (GUI).

The proposed system can work in any Windows operating system higher than Windows 95. Since it uses a unique format to communicate with the other systems it can even connect different Windows operating systems in a network.

Most of the existing systems support only locally connected systems (LAN) but not remote systems. Even if some of the systems support remote operations they provide only limited access to that system. The proposed system supports all the operations in the local network in the remote system as well. The software would also provide an interactive GUI, so even a naïve user can also operate it effectively.

The proposed software performance capability can be listed as following,

1. Retrieving of all the systems ip address and the system id that is connected in the network.
2. Retrieving of all the system information's about the particular files from the drives of the system that are present in the network.
3. Retrieving of all the events that are handled by the system like log type, log message, log time, log source, from the corresponding system in the network
4. Allowing the administrator to play all the media files from the server to clients.
5. Retrieving of all the device information's like Driver name, device descriptions, device state and device type their status.
6. Retrieving of all the service information's like service name, service descriptions, and service state, start up about the corresponding system in the network
7. Allowing the administrator and the users to communicate each other.
8. Whenever the administrator performs any function it will be displayed in the client or in the remote system monitor.
9. Transferring of files from one system to other system.
10. Capturing of all the operations that are performed in the remote systems.

SYSTEM STUDY ANALYSIS

2. SYSTEM STUDY ANALYSIS

2.1 Existing System

The Existing System is specially designed for monitoring all the client system information. This software will only work on Windows NT 4.0 or in Windows Server 2003.

Limitations

1. The Existing system is developed only for the Local Area Network.
2. It is not so effective in retrieving the remote system information.
3. Normally the existing package will only run on Windows NT or in Windows Server 2003 Operating System.
4. Connecting the systems in distributed environment and Sharing data is so difficult.
5. The system information that are retrieved from the client is not highly secured in the network

2.2 System Analysis

System analysis is an activity that encompasses the most of the tasks collectively called as computer engineering. This is the most important step in the software project where we get general idea about the needs of the end-users by having human-to-human conversation with them and about the various

Conditions and restrictions that have to be taken care of while developing the software.

1. Identify the users need.
2. Evaluate the system concept for feasibility.

3. Perform economic and technical analysis.
4. Allocate functions to hardware, software, people, database and other system elements.
5. Establish cost and schedule constraints.
6. Create a system definition that forms the foundation for all subsequent engineering work.

2.2.1. Identification of need

As a first step in the analysis of the system, the end – users of the proposed project were met to get first hand information regarding their needs. Ideas from both the sides were exchanged in order to get a standard and satisfactory system.

2.2.2 Feasibility Study

The feasibility study is carried out to test if the proposed project is worth being implemented. Given unlimited resources and infinite time, all projects are feasible but unfortunately such results and time are not possible in real life situations. Hence it becomes both necessary and prudent to evaluate the feasibility of the project at the earliest possible time in order to avoid unnecessary wastage of time and effort and professional embarrassment over an ill-conceived system. The following feasibility studies were carried out for the proposed project, namely.

2.2.3. Technical Feasibility

These are the technical feasibility constructs

1. Thus a through study reveals that this project is technical feasible. The memory capacity of the existing hardware is quite sufficient for the execution of the system.
2. The speed of the existing hardware and the system is quite sufficient.
3. Technical enhancement may be needed in this system in future, and it will not force barriers to estimated budget.

2.2.4. Economic Feasibility

The cost of the system is evaluated here

- There is no extra cost needed for implementing the system.
- Since it is very easy to use, no training is needed. so training cost can be avoided
- The system is flexible so that further enhancement is possible according to the future needs of the users.

2.2.5 Behavioral Feasibility

People are inherently resistant to changes, and if the user needs a sufficient amount of training, it would result in the expenditure of the users and organisations time, which is precious enough to be saved, generally a user would reject the proposal if it were going to consume much amount of time a effort measure to be put from them .The outcome of establishing of project should bring user-convenience and satisfaction.

The suggested project is much more advantageous and requires lesser amount of time and effort from the users to use the developed software.

2.3 Proposed Systems

The aim of the proposed system is to have the system that monitors the activities of the client systems connected to the network. The proposed system is a win32 API windows based application, developed using Visual basic. NET.

2.3.1 Product Perspective

The perspective of the projects is to provide the user with an effective means for monitoring the Local & Remote client computer based systems by viewing all the system information's and controlling the network from the server. Such information can be listed as:

1. Provision for adding new user to the system and allowing the existing user to modify the user name & password.
2. Provision for selecting the inputs from a list of active systems in the network.
3. Provision to view the system information that is connected in the network.
4. Provision to kill the system process, services, and events and to disable the device driver files from the system.
5. Provision to chat between the users of the system in the network.
6. Provision to transfer the files between the systems in the network
7. Provision to capture the systems desktop from the server.

2.3.2 Product Functions

The software product functions can be listed as:

1. Retrieving of all the systems ip address and the system id that is connected in the network.

2. Retrieving of all the system information's about the particular files from the drives of the system that are present in the network.
3. Retrieving of all the events that are handled by the system like log type, log message, log time, log source, from the corresponding system in the network
4. Allowing the administrator to play all the media files from the server to clients.
5. Retrieving of all the device information's like Driver name, device descriptions, device state and device type their status.
6. Retrieving of all the service information's like service name, service descriptions, and service state, start up about the corresponding system in the network
7. Allowing the administrator and the users to communicate each other.
8. Whenever the administrator performs any function it will be displayed in the client or in the remote system monitor.
9. Transferring of files from one system to other system.
10. Capturing of all the operations that are performed in the remote systems.

2.3.3 General Constraints

The general constraints regarding this software can be listed as:

1. A time gap occurs between sending the request and displays the output.
2. Whenever a new system is added to the network, its details should be entered.
3. Except the administrator no other person should access this program excluding chat.

2.3.4 Requirements

2.3.4.1. Functional Requirements

List of Inputs

1. The users of the system should be provided with a point &click interface, i.e., the users of the software mainly users mouse to provide inputs.
2. Initially the user should authenticate themselves by means of a user name and password. Once the authentication process is over he/she is allowed to perform the required operations provided by the software.
3. The primary input is the ip address of the client system to be captured.

2.3.4.2. Performance Requirements

A. Security

Security should be implemented by means of a user name / password validation process. Only authorized users should be allowed to access the software.

B. Availability

Availability is the probability that a program is operating according to requirements at a given point of time. The availability is an indirect measure of the maintainability of software.

The successful function of the software depends on the validity of the inputs given to it. If the data entered is not appropriate or data is missing the system should indicate the possibility of an error.

C. Capacity

Capacity measures number of systems software can access. The software should manage up to 5 user sessions simultaneously.

D. Response Time

Response time is the time with in which a system identifies the instructions of the user and responds to it.The time required to capture the screen/process from the client system and to display it is assumed to be 20 seconds.

2.3.5. Design Constraints

2.3.5.1. Hardware Constraints

- A Pentium processor of 166 MHz speed (to make sure that application does not take too long to run).
- A Random access memory of 64MB.
- A mouse and keyboard.
- LAN (Ethernet Interface).
- Modem.

2.3.5.2 User interface and Screen Formats

It is required to maintain certain GUI standards during the development of this system. The user can either use a mouse or a keyboard to operate the product.

The user should be provided with menus, dialog boxes and controls such as textbox, edit box, command buttons, scroll bars, data grids, etc., the system should make the best use of the resources provided by .NET Frame work.

2.3.5.3. Operation required by the User

The users of the system should access the software after providing a user-name &password. The only operation required by the user is to provide inputs to the system. The user should provide valid inputs to the system.

2.4 User Characteristics

The system has been designed a very easy to understand point-click interface system. The whole system is necessary to retrieve the system information from the remote system.

The main users of the system are system administrators and the product is such user-friendly. A user need to have only minimum knowledge a basic knowledge on basic networking concepts is recommendable.

2.5 Final Outline of the Proposed System

This is a piece of writing that deals or shows how to use the software. A software based on this concept has been developed to manage the details in the efficient and reliable way.

The proposed software deals with friendly management methodology and how to use the actual resources in the developed system It also deals with the economical aspects like minimum memory and minimum hardware mixtures. The proposed system is economically good for every end user because it makes the system as much as easy than that of existing system

PROGRAMMING ENVIRONMENT

3. PROGRAMMING ENVIRONMENT

3.1. Hardware Configuration

(The configuration that are quoted below are minimum requirements)

Processor	: Pentium II
Processor speed	: 500 MHz
RAM	: 64 MB
Cache Memory	: 512 KB
Hard disk	: 5 GB
Display	: 14''Color Monitor
Mouse	: Microsoft 2 button mouse.
Keyboard	: Samsung 105 key.
Networking	: Intel 21040 based Ethernet controller (16-bit)

3.2 Description of software and tools used-resources for the choice

Operating System	: Windows 98, NT and XP.
Programming Environment	: Visual Basic. NET.

3.2.1. About Windows 98

Windows 98 is a preemptive multitasking operating system that belongs to Microsoft windows family of operating system products. Microsoft windows 98 are the most powerful and reliable operating system. This operating system, which is very familiar Windows 95 user interface. Some of the other features include Multi tasking & Multi threading facilities. It provides highly reliable fault tolerance features. It has the capability to run more than one process at a time.

The major feature provided includes

Probability: -

Unlike most operating system, Windows 98 can run on variety of platforms. The flexibility can be a great advantage when implementing a computer strategy for an organization.

Multi-tasking Operation: -

From the perspective of the end user, multi-tasking means that different type of application can be running in the background.

File Systems: -

Windows 98 supports a variety of file systems, including FAT 32 File Systems.

Features of Windows 98: -

Windows and programs open faster than ever before. By using the Maintenance Wizard, you can easily improve your computer's speed and efficiency. The power management feature allows newer computer to go in to hibernation mode awaken instantly, instead of requiring you to shutdown and restart your computer. Also, you can use the FAT32 file systems to store files more efficiently and saves hard disk space.

3.2.2. About Visual Basic.NET

Visual Basic.NET

Microsoft's Visual Basic Programming language is about to take its biggest technology leap its ten-year history. Microsoft Visual Basic.NET and is a complete and total rewrite of Visual Basic Compiler.

Features of Visual Basic. NET

Object Oriented Capabilities

Visual Basic .NET is Object-Oriented, which means that it's a true Object-Oriented Programming Language. VB .NET supports all the key OOP features like Polymorphism, Inheritance, Abstraction and Encapsulation. It's worth having a brief overview of OOP before starting VB.NET

Some important features of Object Oriented programming are as follows:

- Emphasis on data rather than procedure
- Programs are divided into Objects
- Data is hidden and cannot be accessed by external functions
- Objects can communicate with each other through functions
- New data and functions can be easily added whenever necessary
- Follows bottom-up approach

. NET Framework

.NET is a "Software Platform". It is a language-neutral environment for developing rich .NET experiences and building applications that can easily and securely operate within it. When developed applications are deployed, those applications will target .NET and will execute wherever .NET is implemented instead of targeting a particular Hardware/OS combination.

The components that make up the .NET platform are collectively called the. NETFramework

The .NET Framework is a managed, type-safe environment for developing and executing applications. The .NET Framework manages all aspects of program execution, like, allocation of memory for the storage of data and instructions, granting and denying permissions to the application, managing

execution of the application and reallocation of memory for resources that are not needed.

The .NET Framework is designed for cross-language compatibility. Cross-language compatibility means, an application written in Visual Basic.NET may reference a DLL file written in C# (C-Sharp). A Visual Basic.NET class might be derived from a C# class or vice versa.

The .NET Framework consists of two main components:

- The Common Language Runtime (CLR)
- Set of Class Libraries

The Common Language Runtime (CLR)

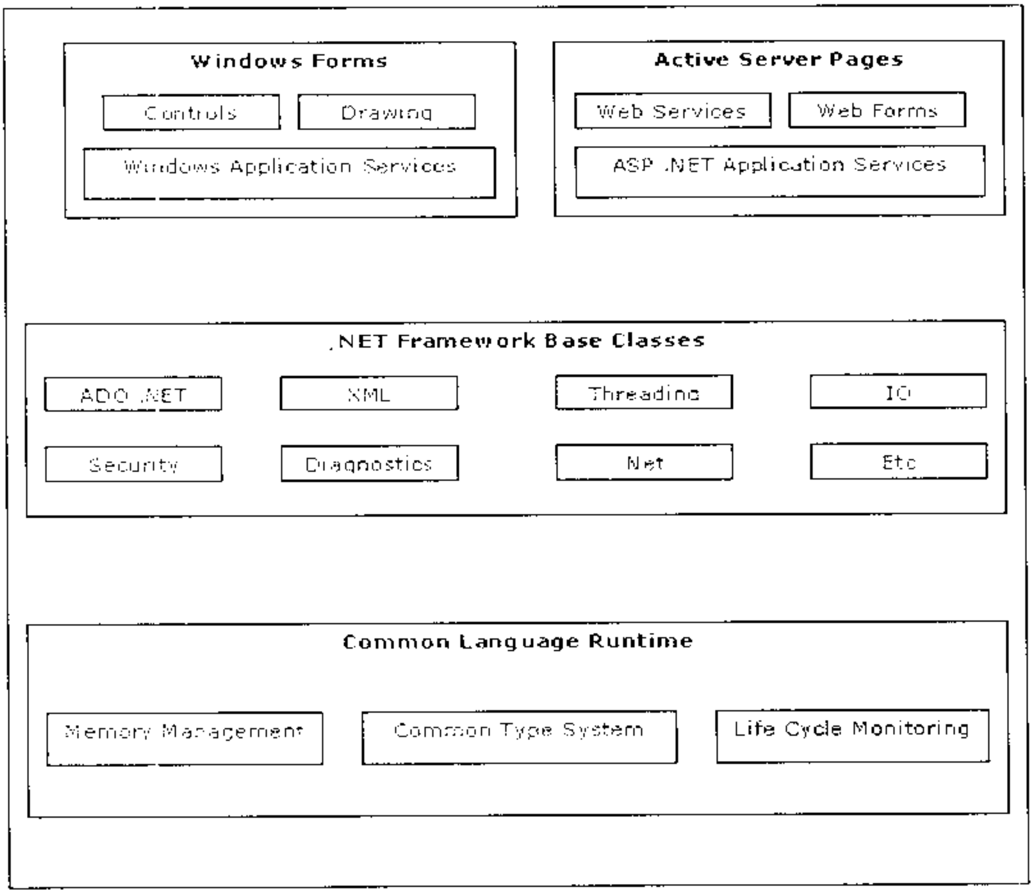
The CLR is described as the “execution engine” of .NET. It’s this CLR that manages the execution of programs. It provides the environment within which the programs run. The software version of .NET is actually the CLR version.

When the .NET program is compiled, the output of the compiler is not an executable file instead it is a file that contains a special type of code, which is the Microsoft Intermediate Language (MSIL). This MSIL defines a set of portable instructions that are independent of any specific CPU. It's the job of the CLR to translate this Intermediate code into a executable code when the program is executed making the program to run in any environment for Which the CLR is implemented. And that's how the .NET Framework achieves Portability. This MSIL is turned into executable code using a JIT (Just In Time) complier. The process goes like this, when the .NET programs are executed, the CLR activates the JIT complier. The JIT complier converts MSIL into native code on a demand basis as each part of the program is needed. Thus the program executes as a native code even though it is compiled into MSIL making the program to run as fast as it would if it is compiled to native code but achieves the portability benefits of MSIL.

Class Libraries

Class library, is the second major entity of the .NET Framework. This library gives the program access to runtime environment. The class library consists of lots of prewritten code that all the applications created in VB.NET and Visual Studio.NET will use. The code for all the elements like forms, controls and the rest in VB.NET applications actually comes from the class library.

Figure 3.1



.NET Framework

The Common Language Specification (CLS)

If we want the code, which we write in a language to be used by programs in other languages then it should adhere to the Common Language Specification (CLS). The CLS describes a set of features that different languages have in common. The CLS includes a subset of Common Type System (CTS), which define the rules concerning data types and ensures that Code is executed in a safe environment.

SYSTEM DESIGN & DEVELOPMENT

4. SYSTEM DESIGN AND DEVELOPMENT

4.1 Design and Development Process

Fundamental Design Concepts

The most creative and challenging phase of the system life cycle is system design. The term design describes a finite system and the process by which it is developed. It also includes the construction of program and testing.

System design is the process by which the detailed design of the system selected in the study phases accomplished. The system design goes through the logical and physical state of the development. In this the user oriented performance specification is expanded into design specification. While designing a new system we should have in mind a clear picture of inputs that may be required for the system.

The design covers the following:

1. Reviews the current physical system
2. Prepares output specifications
3. Prepares input specifications
4. Prepares edit, security and control specifications
5. Specifies the implementation plan
6. Prepares a logical design walkthrough of the information flow, output, input, controls and implementation plan
7. Reviews benefits, costs, target dates and system constraints

4.2 Design Notations

The analyst uses a set of techniques and graphical tools that allow the analyst to develop a new kind of system specifications that are easily understandable to the user. The analyst considers new goals and structured tools for analysis.

The new goals specify the following:

1. Use graphics wherever possible to help communicate Better with the user.
2. The analyst use the various design notations such as Data flow diagram, structured chart for analysis

4.3 Data Flow Diagram

A Data Flow Diagram also known as a “Bubble chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. A Data Flow Diagram consists of a series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flows in the system.

Figure 4.1

Over all Process Diagram

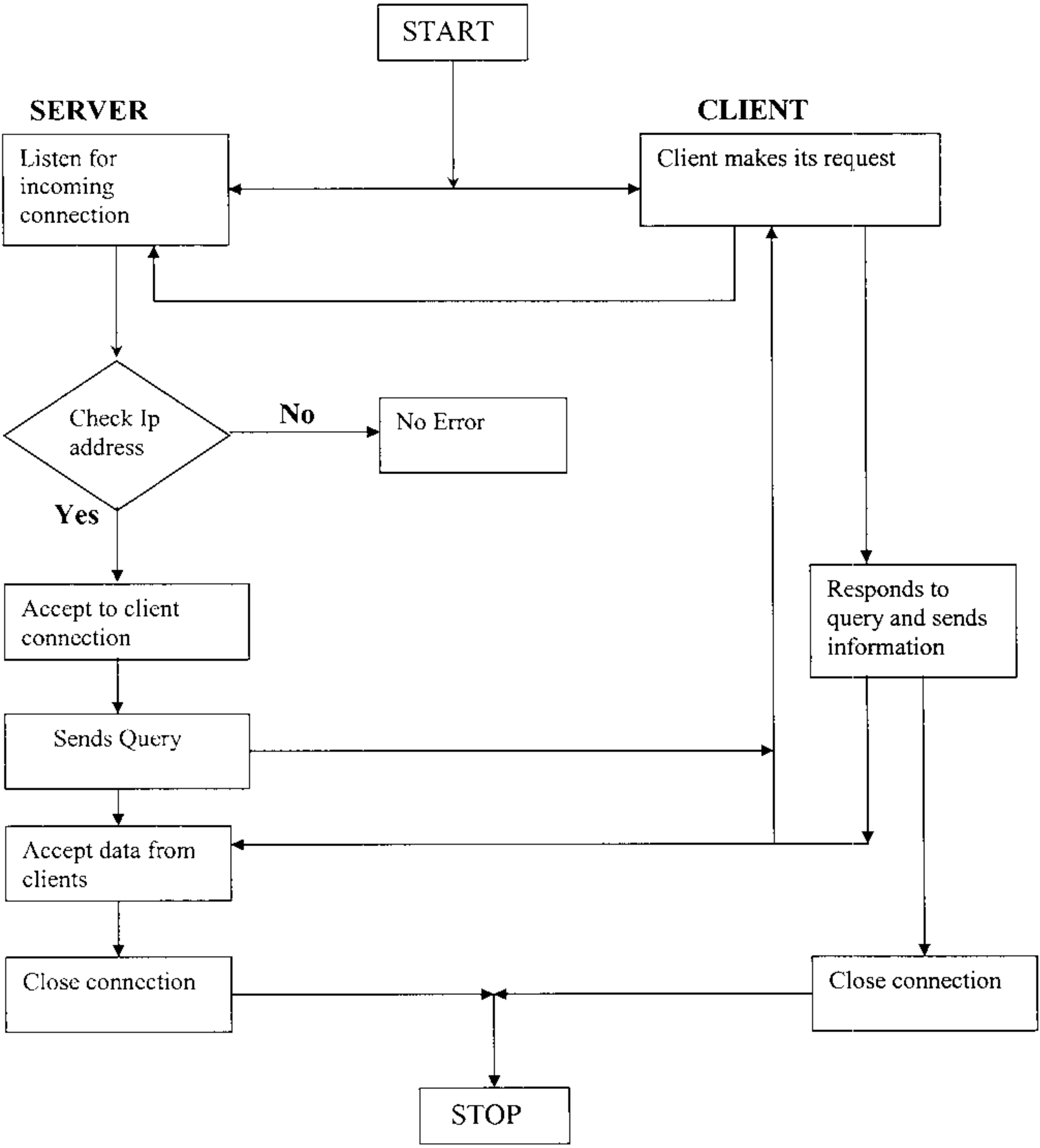


Figure 4.2

Client-Server Communication Diagram

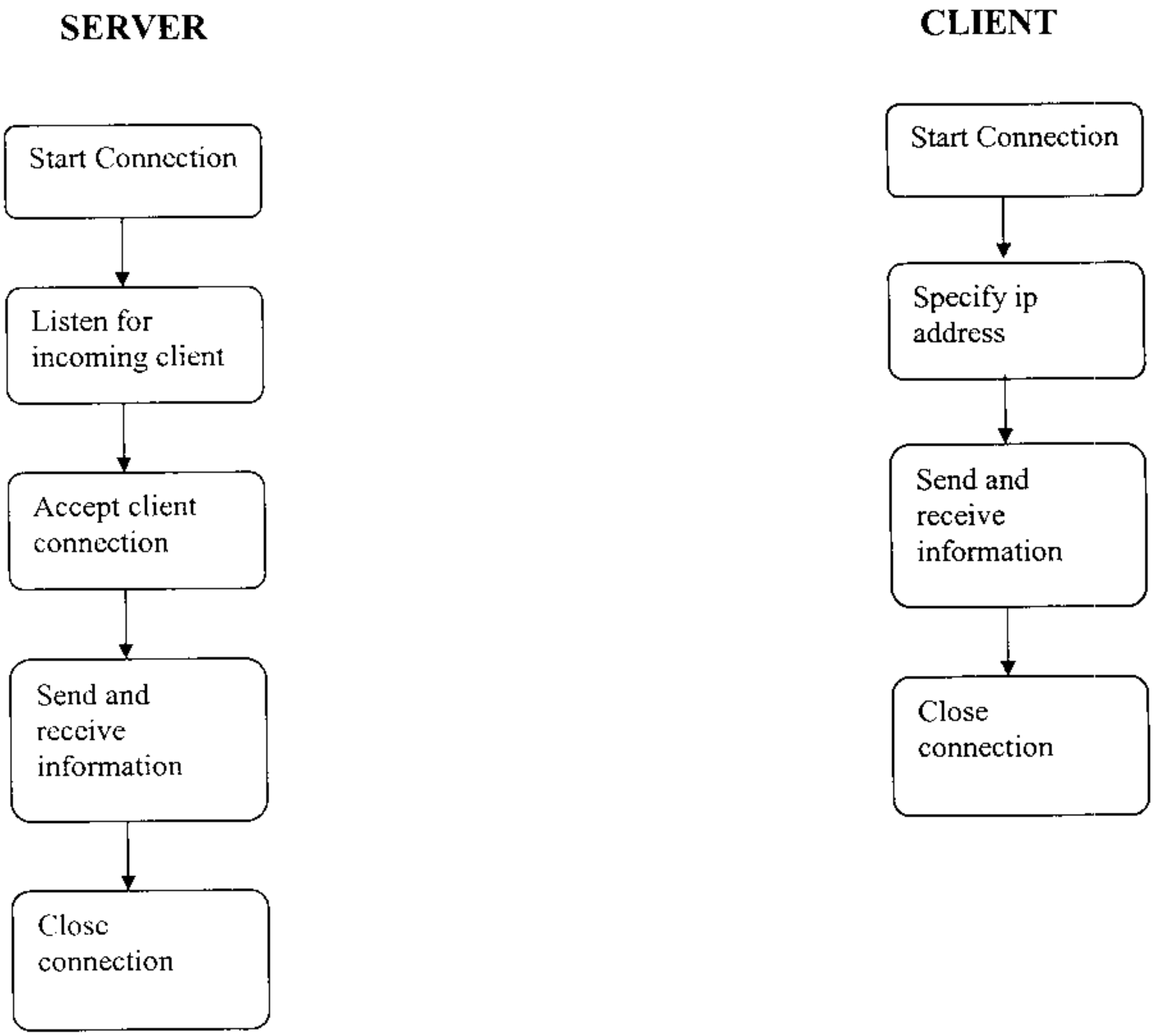


Figure 4.3

Context Flow Diagrams

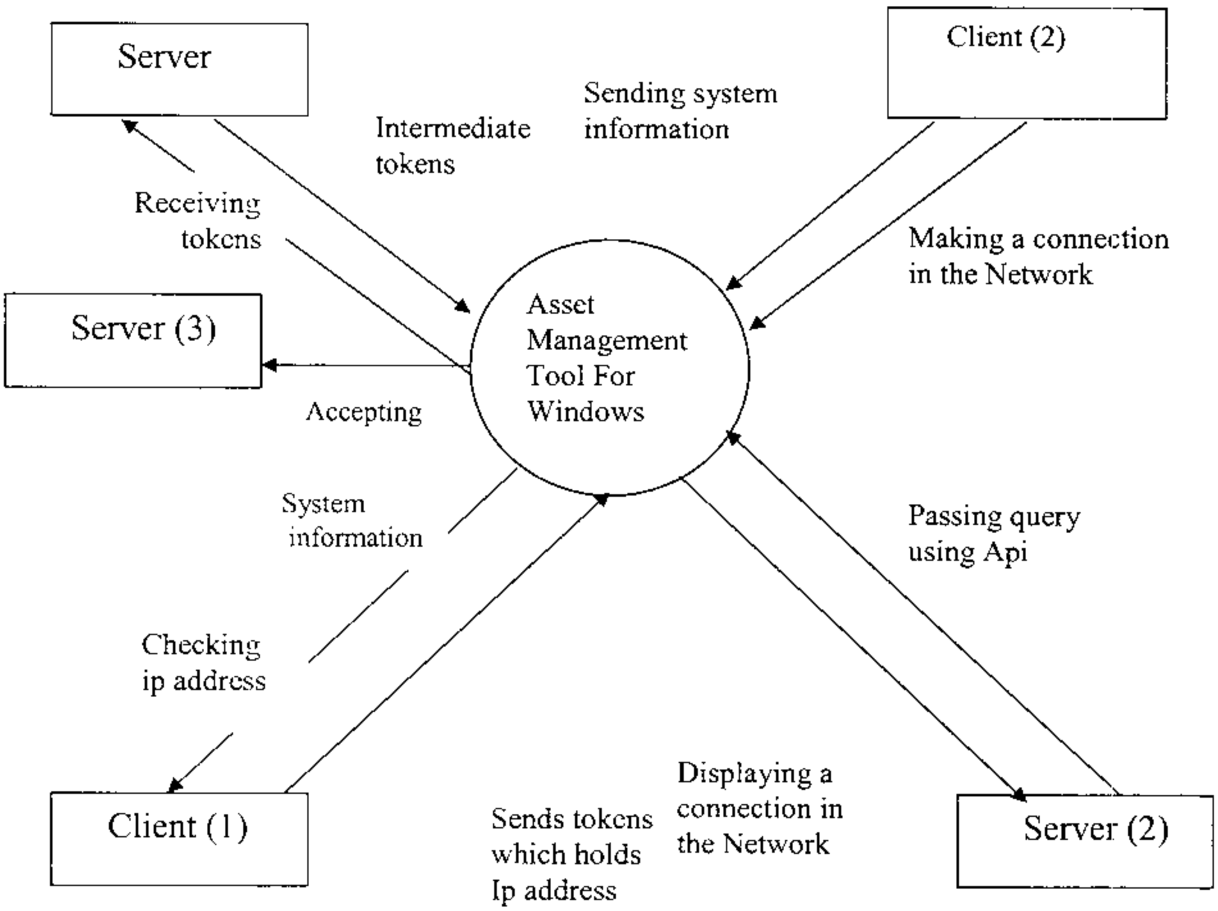

Server
Client (2)
Sending system information
Intermediate tokens
Receiving tokens
Making a connection in the Network
Server (3)
Asset Management Tool For Windows
Accepting
System information
Passing query using Api
Checking ip address
Displaying a connection in the Network
Client (1)
Sends tokens which holds Ip address
Server (2)

Figure 4.4

DATA FLOW DIAGRAM (LEVEL 0)

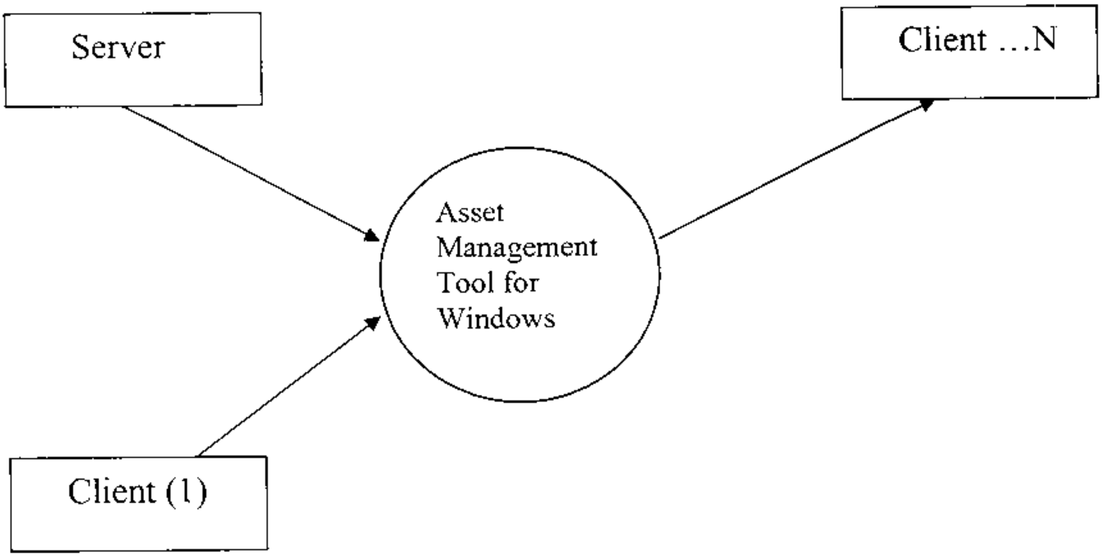
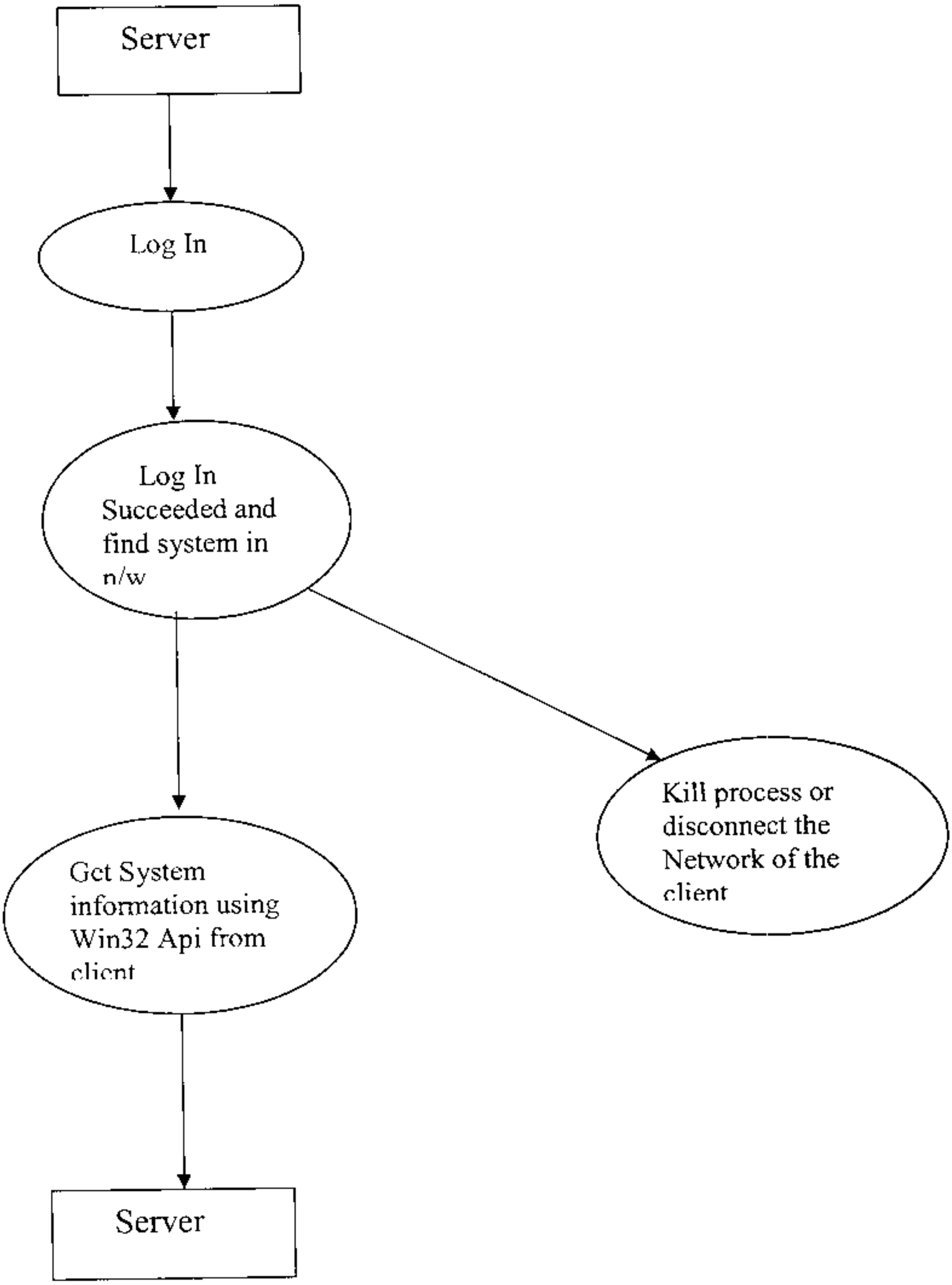


Figure 4.5

DATA FLOW DIAGRAM (LEVEL 1)


Server
Log In
Log In Succeeded and find system in n/w
Get System information using Win32 Api from client
Kill process or disconnect the Network of the client
Server

SYSTEM TESTING & DEVELOPMENT

5 SYSTEM TESTING & IMPLEMENTATION

5.1 Testing

Testing is a technique to establish in an experimental way the reliability and robustness of the software. Testing is an important process, which leads to the success of the system. System testing is mainly performed in the intention of finding error free system. System testing makes a logical assumption that all parts of the system are correct and move towards it to make the system error free. Inadequate testing of the system will lead to errors, which even can arise after long time of the system implementation.

When validating a system, number of aspect play role. First it must be determined whether the software satisfies the original requirements and global set by user, as specified during analysis. Secondly it must be established whether the system meets the specification laid down in the design document.

5.1.1 Testing Methodologies

The following are the various testing methodologies that can be applied in order to find an error free system.

Unit Testing

In Unit Testing, the various programs of the system are listed. Unit testing first locates errors on those modules that are independent of one another. This enables to detect errors on those modules that are independent of one another.

This enables to detect errors in loading and logic that are contained within the same module. Various test cases were generated to categories of transport management system processing.

- Improper or incorrect typing.
- Erroneous initialization or default values.
- Inconsistent data typing.
- Overflow or underflow of records.
- All error handling paths where tested.

Code Testing

Coding translated the detailed design specification into a program language that is ultimately transformed into machine executed instruction. The code testing examines the logic of a program.

Software Testing

Software testing represents the ultimate review of specification, design and preview. The test and integration phase is the final filter for all errors of omission and commission. Hence, testing performs a very critical role for quality assurance and for ensuring the reliability of software.

Functional Testing

The philosophy is that if the module accepts all test cases inputs and produces correct result, we do not care how it does it. This approach places burden on test design test cases and procedure to provide operational eventualities. The objective is to search for interface errors, functional errors, performance errors, and shortcomings and startup/shutdown errors.

Program Testing

A program represents the logical elements of system. For a program to run satisfactorily, it must compile and test data correctly and tie in properly with other programs. Achieving an error-free program is the responsibility of the programmer. Program testing checks for two types of errors: syntax and logic. A syntax error is a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted key words are common syntax errors. These errors are Shown through error messages generated by syntax errors. These errors are shown through error messages generated by the computer.

A logic error, on the other hand, deals with incorrect data fields, out-of range items, and invalid combinations. Since diagnostics do not detect logic errors, the programmer must examine the output carefully for them.

When a program is tested, the actual output is compared with the expected output. When there is a discrepancy, the sequence of instructions must be traced to determine the problem. Breaking the program down into self-contained portions, each of which can be checked at certain key points, facilitates the process. The idea is to compare program values against desk-calculated values to isolate the problem.

Validation Testing

Software validation was achieved through series of block box testing that helps to find out errors incorrect and missing functions, interface errors, errors in the data structures, performance errors, initialization errors and termination errors.

User Acceptance Testing

An acceptance test has the objective of selling the user on the validity and reliability of the system. It verifies that the system's procedures operate to system specifications and that the integrity of vital data is maintained.

Performance of an acceptance test is actually the user's show, user motivation and knowledge is critical for the successful performance of the system. Then a comprehensive test report is prepared. The report indicates the system's tolerance, performance range, error rate, and accuracy.

5.1.2 Quality Assurance

The goal of quality assurance is to provide management with the data necessary to be performed about product quality, there by gaining insight and confidence that product quality meeting its goal. The quality assurance is the activity for any business that produces to be used by others. Quality Assurance is composed of technical works for quality as planning, oversight and record keeping analysis and reporting.

Quality assurance defines the objectives of the projects and reviews the overall activities so that errors are corrected early in the development process.

5. 1.3 Generic Risks

Any large project involved certain risks, and that is true of software projects. Risk management is the area that tries to ensure that the impact of risks on cost, quality, and schedule is minimal.

Risk assessment is an activity that must be undertaken during project planning. This involves identifying the risks, analyzing them and prioritizing them on the basis of the analysis.

At a very high level, the software risks can be broadly divided into three categories. These are:

- Cost risk
- Performance risk
- Schedule risk

Risk control comprises active measures that are taken by project management to minimize the impact of risks.

5.1.4 Security Technologies & Policies

The system security problem can be divided into four related issues:

- **Security**
- **Integrity**
- **Privacy**
- **Confidentiality**

System Security

System Security refers to the technical innovations and procedures applied to the hardware and operating systems to protect against deliberate or accidental damage from a define threat.

System Integrity

System Integrity refers to the proper functioning of hardware and programs, appropriate physical security, and safety against external threats such as wiretapping and eavesdropping.

Privacy

Privacy defines the rights of the users or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair, or excessive dissemination of information about it.

Confidentiality

The term Confidentiality is a special status given to sensitive information in a database to minimize the possible invasion of privacy.

5.2. System Implementation

An important aspect of a system analyst's job is to make sure that the new design is implemented to established standards. The term implementation has different means, ranging from the conversion of a basic application to a complete replacement of a compute system.

Implementation is used here to mean the process of converting a new or a revised system design into an operational one. The other aspect is the post implementation review.

5.2.1. Implementation Procedures

Change over is the stage of moving from existing system to new proposed system. The change over from the old to new system may take place when the system has been proved to the satisfaction for the system analyst, users managers and operational staff. One strategy for change over is the parallel running.

In this methods, the old and new systems are run simultaneously for an agreed of time and result from the two systems are compared. Another strategy is the direct change over. This method is the complete replacement of the old system by the new system in one more.

5.2.2User Training

An analysis of user training focuses on two factors: user capabilities and the nature of the system being installed. Users range from the naïve to the highly sophisticated. Developmental research provides interesting insights into how naïve computer users think about their first exposure to a new system. They approach it as concrete learners, learning how to use system without trying to understand which abstract principles determine which function. The distinction between concrete and formal learning says much about what one can expect from trainees I general. Three important lessons that pertain to user training can be concluded from this case situation.

- Users are reluctant to read manuals, but they will learn from demonstrations and through visual aids. Users also tend to be natural teachers.
- Another user training element is a training demonstration. Live demonstrations with personal contact are extremely effective for

training users. In a demonstration, a new concept that is show in many ways learned.

- The third element of user training is the resident expert. In our example, one clerk read the manual carefully, spent time on her own to practice, and ended up being the resident expert a natural teachers

5.2.3. Operational Documentation

After the design phase, operational documentation was done. This document specified the user how information should be entered. The details of planning inspection and physical storage are entered. The details relating to how to use a particular control and how to generate useful reports are also specified to the user.

CONCLUSION

6. CONCLUSION

“Asset management tool for windows” is successfully designed and developed at mentor labs, this project is developed using Microsoft visual basic.Net. This project was done keeping in mind the fact it should follow all the steps of software engineering process and covers the complete software development cycle. The user interface provided by this project will be widely accepted by the user in general the software was tested thoroughly to ensure that it works effectively and efficiently. Complete documentation that is provided makes the changes and enhancements that are to be done very easy and provides the vitality of documentation. The preference of vb.net to any other GUI is also justified since any further enhancement of this project would mean that more of platform independent extensions that are fore seen with this project.

The application is tested with the user requirements and verified for validity. The software requirements have been met. Needed documents were generated and adequate documentation has been provided for maintenance and future enhancements.

APPENDIX – I

APPENDIX I

SAMPLE CODE

`DisplayRemoteServices:`

```
Imports System.Management



Public Class DisplayRemoteServices
    Inherits System.Windows.Forms.Form
    Friend compName, userName, password As String



#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
```

```
Boolean)
    If disposing Then
      If Not (components Is Nothing) Then
        components.Dispose()
      End If
    End If
    MyBase.Dispose(disposing)
  End Sub

  'Required by the Windows Form Designer
  Private components As System.ComponentModel.IContainer

  'NOTE: The following procedure is required by the Windows Form
Designer
  'It can be modified using the Windows Form Designer.
  'Do not modify it using the code editor.
  Friend WithEvents RichTextBox1 As
System.Windows.Forms.RichTextBox
  <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
    Me.RichTextBox1 = New System.Windows.Forms.RichTextBox()
    Me.SuspendLayout()
    '
    'RichTextBox1
    '
    Me.RichTextBox1.Dock = System.Windows.Forms.DockStyle.Fill
    Me.RichTextBox1.Name = "RichTextBox1"
```

```
        Me.RichTextBox1.TabIndex = 0
        Me.RichTextBox1.Text = ""
        '
        'DisplayRemoteServices
        '
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.ClientSize = New System.Drawing.Size(292, 273)
        Me.Controls.AddRange(New System.Windows.Forms.Control()
{Me.RichTextBox1})
        Me.Name = "DisplayRemoteServices"
        Me.Text = "DisplayRemoteServices"
        Me.ResumeLayout(False)

    End Sub

#End Region

    Private Sub DisplayRemoteServices_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'RichTextBox1.Text = Win32Service.EnumerateProcesses(compName,
userName, password)
        RichTextBox1.Text = Win32Service.EnumerateProcesses("giridhar",
"administrator", "jdev")
    End Sub
```

```
System.Object, ByVal e As System.EventArgs) Handles
RichTextBox1.TextChanged


    End Sub
End Class





Class Win32Service
    Public Shared Function EnumerateServices(ByVal machineName As
String, ByVal username As String, ByVal password As String) As
ManagementObjectSearcher
        Dim sc As ManagementScope
        Dim co As ConnectionOptions
        Dim oq As ObjectQuery
        Dim os As ManagementObjectSearcher
        Dim service As ManagementObject
        Dim message As String
        Try
            co = New ConnectionOptions()
            co.Impersonation = ImpersonationLevel.Impersonate
            co.EnablePrivileges = True
            If machineName.ToUpper() =
Environment.MachineName.ToUpper() Then
                sc = New ManagementScope("\ROOT\CIMV2", co)
            Else
            co.Password = password
            sc = New ManagementScope("\\" & machineName &
"\ROOT\CIMV2", co)
        End If
        sc.Connect()
        oq = New ObjectQuery("SELECT * FROM Win32_Service")
        os = New ManagementObjectSearcher(sc, oq)
        'For Each service In os.Get()
        '    message += "Name = " & service.GetPropertyValue("Name") &
", " & "State = " & service.GetPropertyValue("State") & vbCrLf
        'Next
      Catch ex As Exception
        'message += ex.StackTrace
      Finally
        'os.Dispose()
      End Try
      Return os
    End Function


    Public Shared Function EnumerateProcesses(ByVal machineName As
String, ByVal username As String, ByVal password As String) As String
'ManagementObjectSearcher
      Dim sc As ManagementScope
      Dim co As ConnectionOptions
      Dim oq As ObjectQuery
      Dim os As ManagementObjectSearcher
    Dim message As String
    Try
       co = New ConnectionOptions()
       co.Impersonation = ImpersonationLevel.Impersonate
       co.EnablePrivileges = True
       If machineName.ToUpper() =
Environment.MachineName.ToUpper() Then
          sc = New ManagementScope("\ROOT\CIMV2", co)
       Else
          co.Username = username
          co.Password = password
          sc = New ManagementScope("\\" & machineName &
"\ROOT\CIMV2", co)
       End If
       sc.Connect()
       oq = New ObjectQuery("SELECT * FROM Win32_Share")
       os = New ManagementObjectSearcher(sc, oq)
       For Each processes In os.Get()
          message += "Name = " & processes.GetPropertyValue("Name") &
", " & "State = " & processes.GetPropertyValue("State") & vbCrLf
       Next
     Catch ex As Exception
       message += ex.StackTrace
     Finally
       os.Dispose()
     End Try
     'Return os
```

```
  End Function

  Public Shared Function EnumerateShares(ByVal machineName As
String, ByVal username As String, ByVal password As String) As String
'ManagementObjectSearcher
    Dim sc As ManagementScope
    Dim shares As ManagementClass
    Dim mgmtPath As ManagementPath
    Dim co As New ConnectionOptions()
    Dim oq As ObjectQuery
    Dim os As ManagementObjectSearcher
    Dim processes As ManagementObject
    Dim message As String
    Try
      co.Impersonation = ImpersonationLevel.Impersonate
      co.EnablePrivileges = True
      If machineName.ToUpper() =
Environment.MachineName.ToUpper() Then
        sc = New ManagementScope("\ROOT\CIMV2", co)
      Else
        co.Username = username
        co.Password = password
        sc = New ManagementScope("\\" & machineName &
"\ROOT\CIMV2", co)
      End If
      sc.Connect()
      oq = New ObjectQuery("SELECT * FROM Win32_Share")
```

```
            For Each processes In os.Get()
                message += "Name = " & processes.GetPropertyValue("Name") &
", " & "State = " & processes.GetPropertyValue("State") & vbCrLf
            Next
        Catch ex As Exception
            message += ex.StackTrace
        Finally
            os.Dispose()
        End Try
        'Rcturn os
        Return message
    End Function

    Public Shared Function GetRemoteProcesses() As String


    End Function
End Class
```

LaunchConsole:

```
Public Class LaunchConsole
    Inherits System.Windows.Forms.Form
    Dim clrBackcolor, clrForecolor As Color
    Dim fntConsolefont As Font
    Dim frmMapDrive As New MapNetworkDrive()

#Region " Windows Form Designer generated code "
```

```
    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form
Designer
    'It can be modified using the Windows Form Designer.
    Friend WithEvents niTrayicon As System.Windows.Forms.NotifyIcon
    Friend WithEvents MainMenu1 As System.Windows.Forms.MainMenu
    Friend WithEvents MenuItem9 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem11 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem12 As System.Windows.Forms.MenuItem
    Friend WithEvents mnuSystem As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem14 As System.Windows.Forms.MenuItem
    Friend WithEvents cmAMC As System.Windows.Forms.ContextMenu
    Friend WithEvents cmiHide As System.Windows.Forms.MenuItem
    Friend WithEvents cmiMinimize As System.Windows.Forms.MenuItem
    Friend WithEvents cmiMaximize As System.Windows.Forms.MenuItem
    Friend WithEvents cmiExit As System.Windows.Forms.MenuItem
    Friend WithEvents cmiShow As System.Windows.Forms.MenuItem
    Friend WithEvents mniExit As System.Windows.Forms.MenuItem
    Friend WithEvents mniPing As System.Windows.Forms.MenuItem
    Friend WithEvents mniFindcomp As System.Windows.Forms.MenuItem
    Friend WithEvents mniAMC As System.Windows.Forms.MenuItem
    Friend WithEvents cdColordialog As
System.Windows.Forms.ColorDialog
    Friend WithEvents mniFont As System.Windows.Forms.MenuItem
    Friend WithEvents fdFont As System.Windows.Forms.FontDialog
    Friend WithEvents MenuItem4 As System.Windows.Forms.MenuItem
    Friend WithEvents mniArrenge As System.Windows.Forms.MenuItem
    Friend WithEvents mniCascade As System.Windows.Forms.MenuItem
    Friend WithEvents mniHorizontal As System.Windows.Forms.MenuItem
    Friend WithEvents mniVertical As System.Windows.Forms.MenuItem
    Friend WithEvents mniList As System.Windows.Forms.MenuItem
    Friend WithEvents mniAbtAMC As System.Windows.Forms.MenuItem
    Friend WithEvents fdOpen As System.Windows.Forms.OpenFileDialog
    Friend WithEvents mniMapDrive As System.Windows.Forms.MenuItem
    Friend WithEvents mnuColor As System.Windows.Forms.MenuItem
    Friend WithEvents mniBackColor As System.Windows.Forms.MenuItem
    Friend WithEvents mniForecolor As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem1 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem6 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem17 As System.Windows.Forms.MenuItem
    Friend WithEvents mniViewLogReport As
System.Windows.Forms.MenuItem
    Friend WithEvents mniViewSvcReport As
System.Windows.Forms.MenuItem
    Friend WithEvents mniViewProReport As
System.Windows.Forms.MenuItem
    Friend WithEvents mniDisconnect As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem3 As System.Windows.Forms.MenuItem
    Friend WithEvents mniRemoteAMC As
System.Windows.Forms.MenuItem
    Friend WithEvents mniBrowseNetwork As
System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem5 As System.Windows.Forms.MenuItem
    Friend WithEvents mniBackImage As System.Windows.Forms.MenuItem
    Friend WithEvents mniViewDevReport As
System.Windows.Forms.MenuItem
    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
```

```
    Dim resources As System.Resources.ResourceManager = New
System.Resources.ResourceManager(GetType(LaunchConsole))
    Me.niTrayicon = New
System.Windows.Forms.NotifyIcon(Me.components)
    Me.cmAMC = New System.Windows.Forms.ContextMenu()
    Me.cmiShow = New System.Windows.Forms.MenuItem()
    Me.cmiHide = New System.Windows.Forms.MenuItem()
    Me.cmiMinimize = New System.Windows.Forms.MenuItem()
    Me.cmiMaximize = New System.Windows.Forms.MenuItem()
    Me.MenuItem14 = New System.Windows.Forms.MenuItem()
    Me.cmiExit = New System.Windows.Forms.MenuItem()
    Me.MainMenu1 = New System.Windows.Forms.MainMenu()
    Me.mnuSystem = New System.Windows.Forms.MenuItem()
    Me.mniAMC = New System.Windows.Forms.MenuItem()
    Me.mniRemoteAMC = New System.Windows.Forms.MenuItem()
    Me.MenuItem3 = New System.Windows.Forms.MenuItem()
    Me.mniBrowseNetwork = New System.Windows.Forms.MenuItem()
    Me.MenuItem6 = New System.Windows.Forms.MenuItem()
    Me.mniFindcomp = New System.Windows.Forms.MenuItem()
    Me.mniPing = New System.Windows.Forms.MenuItem()
    Me.MenuItem17 = New System.Windows.Forms.MenuItem()
    Me.mniMapDrive = New System.Windows.Forms.MenuItem()
    Me.mniDisconnect = New System.Windows.Forms.MenuItem()
    Me.MenuItem9 = New System.Windows.Forms.MenuItem()
    Me.mniExit = New System.Windows.Forms.MenuItem()
    Me.MenuItem1 = New System.Windows.Forms.MenuItem()
    Me.mniViewLogReport = New System.Windows.Forms.MenuItem()
    Me.mniViewProReport = New System.Windows.Forms.MenuItem()
    Me.mniViewDevReport = New System.Windows.Forms.MenuItem()
    Me.MenuItem12 = New System.Windows.Forms.MenuItem()
    Me.mniBackImage = New System.Windows.Forms.MenuItem()
    Me.MenuItem5 = New System.Windows.Forms.MenuItem()
    Me.mniFont = New System.Windows.Forms.MenuItem()
    Me.MenuItem4 = New System.Windows.Forms.MenuItem()
    Me.mnuColor = New System.Windows.Forms.MenuItem()
    Me.mniBackColor = New System.Windows.Forms.MenuItem()
    Me.mniForecolor = New System.Windows.Forms.MenuItem()
    Me.MenuItem11 = New System.Windows.Forms.MenuItem()
    Me.mniArrenge = New System.Windows.Forms.MenuItem()
    Me.mniCascade = New System.Windows.Forms.MenuItem()
    Me.mniHorizontal = New System.Windows.Forms.MenuItem()
    Me.mniVertical = New System.Windows.Forms.MenuItem()
    Me.mniList = New System.Windows.Forms.MenuItem()
    Me.mnuAbout = New System.Windows.Forms.MenuItem()
    Me.mniAbtAMC = New System.Windows.Forms.MenuItem()
    Me.cdColordialog = New System.Windows.Forms.ColorDialog()
    Me.fdFont = New System.Windows.Forms.FontDialog()
    Me.fdOpen = New System.Windows.Forms.OpenFileDialog()
    '
    'niTrayicon
    '
    Me.niTrayicon.ContextMenu = Me.cmAMC
    Me.niTrayicon.Icon = CType(resources.GetObject("niTrayicon.Icon"),
System.Drawing.Icon)
```

```
        Me.niTrayicon.Visible = True
        '
        'cmAMC
        '
        Me.cmAMC.MenuItems.AddRange(New
System.Windows.Forms.MenuItem() {Me.cmiShow, Me.cmiHide,
Me.cmiMinimize, Me.cmiMaximize, Me.MenuItem14, Me.cmiExit})
        '
        'cmiShow
        '
        Me.cmiShow.DefaultItem = True
        Me.cmiShow.Index = 0
        Me.cmiShow.Text = "Show"
        '
        'cmiHide
        '
        Me.cmiHide.Index = 1
        Me.cmiHide.Text = "Hide"
        '
        'cmiMinimize
        '
        Me.cmiMinimize.Index = 2
        Me.cmiMinimize.Text = "Minimize"
        '
        'cmiMaximize
        '
        Me.cmiMaximize.Index = 3
```

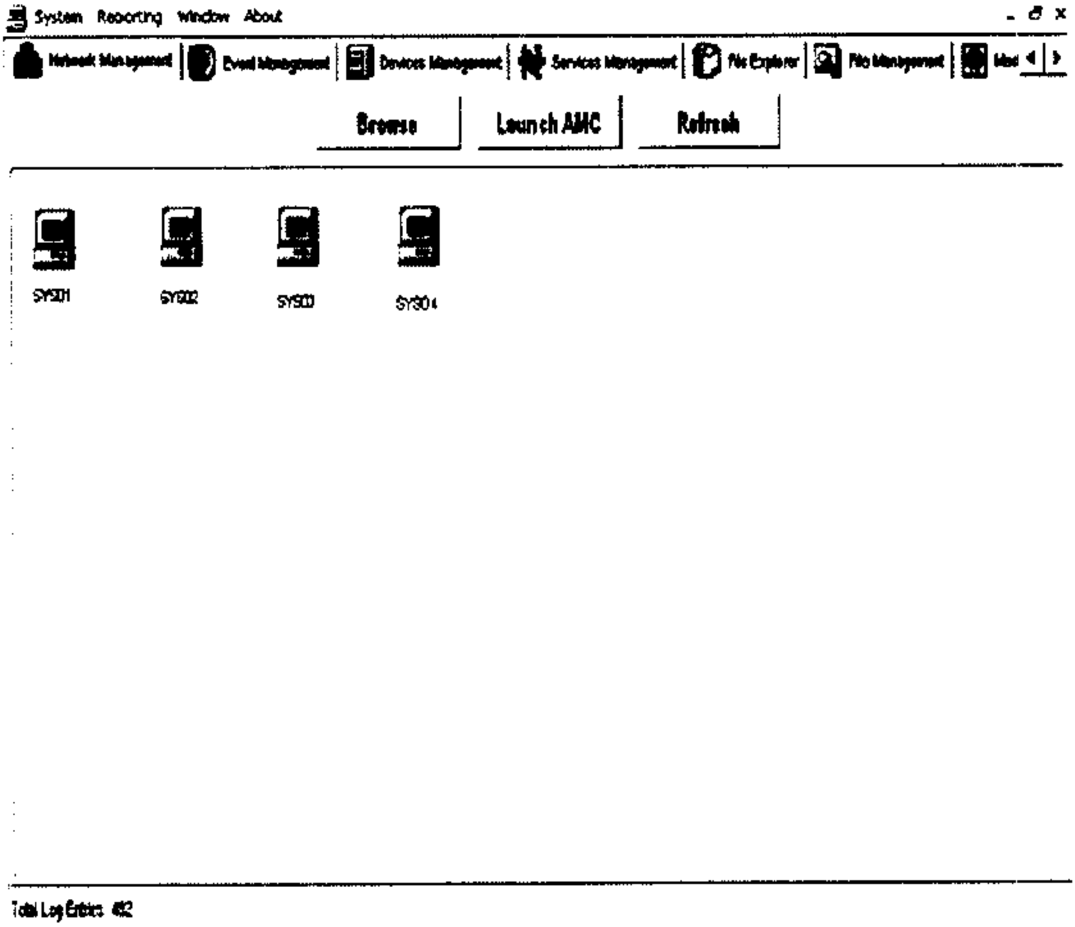
```
    '
    'MenuItem14
    '
    Me.MenuItem14.Index = 4
    Me.MenuItem14.Text = "-"
    '
    'cmiExit
    '
    Me.cmiExit.Index = 5
    Me.cmiExit.Text = "Exit"
    '
    'MainMenu1
    '
    Me.MainMenu1.MenuItems.AddRange(New
System.Windows.Forms.MenuItem() {Me.mnuSystem, Me.MenuItem1,
Me.MenuItem12, Me.MenuItem11, Me.mnuAbout})
    '
    'mnuSystem
    '
    Me.mnuSystem.Index = 0
    Me.mnuSystem.MenuItems.AddRange(New
System.Windows.Forms.MenuItem() {Me.mniAMC, Me.mniRemoteAMC,
Me.MenuItem3, Me.mniBrowseNetwork, Me.MenuItem6,
Me.mniFindcomp, Me.mniPing, Me.MenuItem17, Me.mniMapDrive,
Me.mniDisconnect, Me.MenuItem9, Me.mniExit})
    Me.mnuSystem.Text = "&System"
    '
```

APPENDIX – II

APPENDIX II

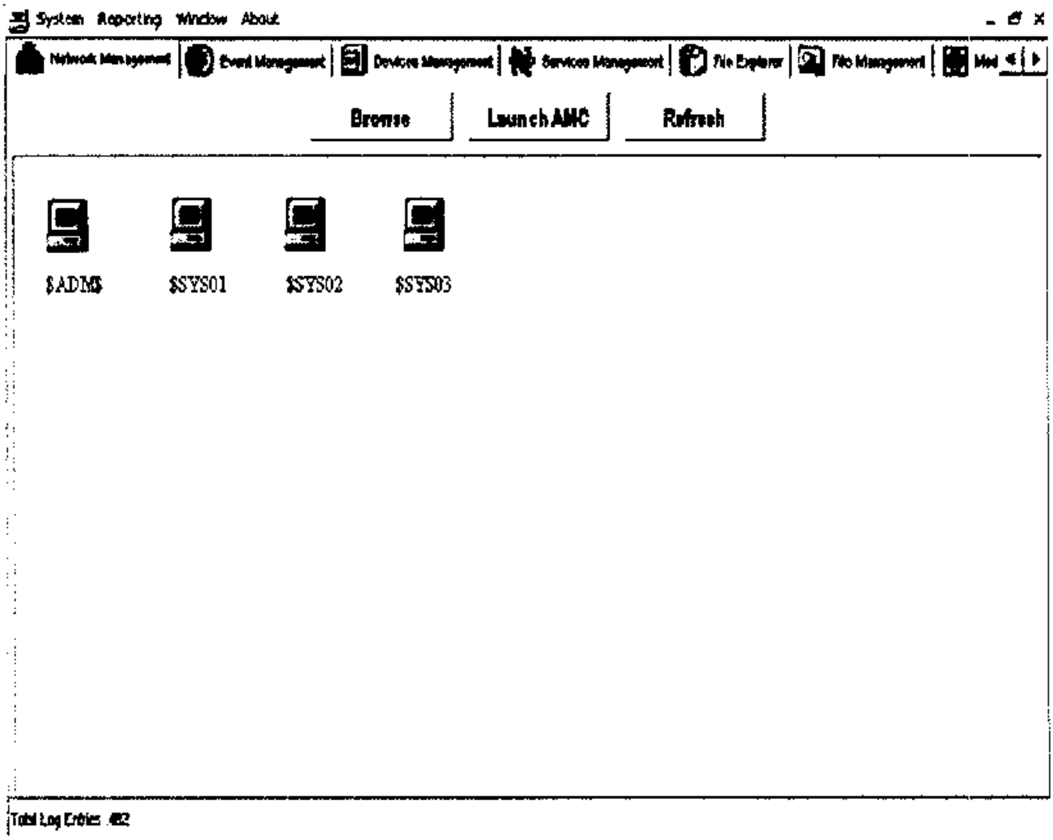
SCREEN SHOTS

RETRIEVING LOCAL SYSTEM IN THE NETWORK

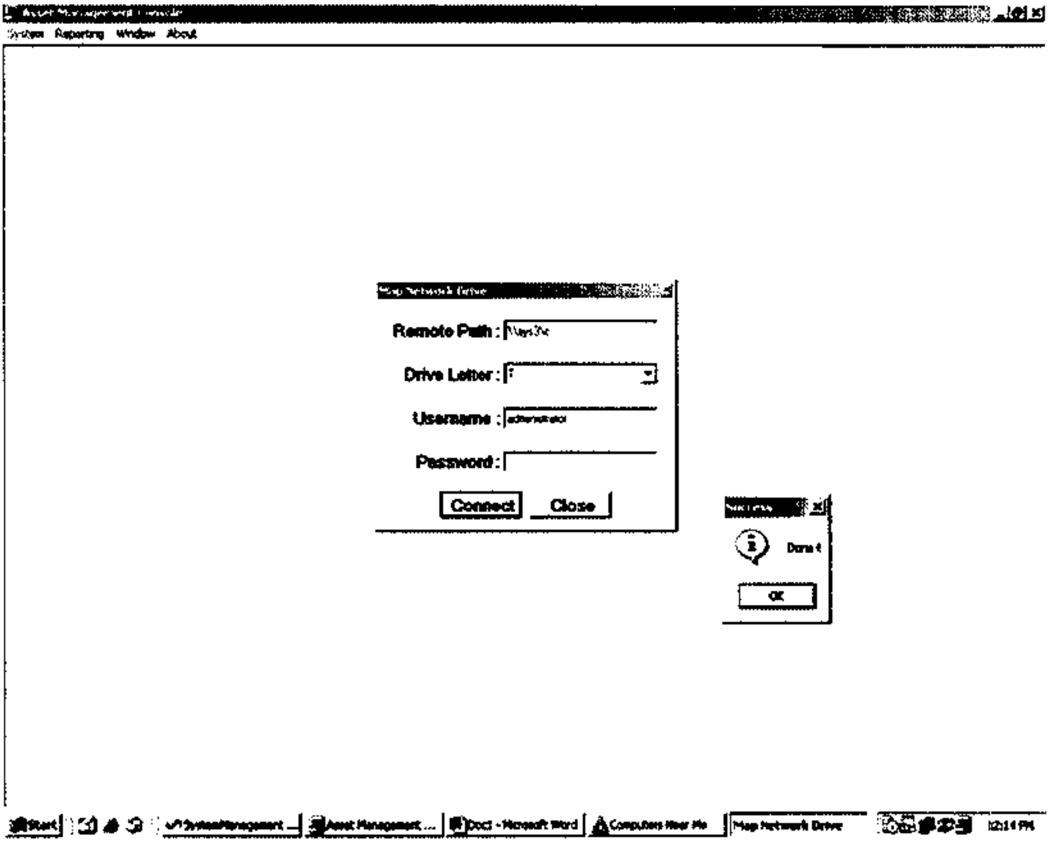


Total Log Entries: 482

RETRIEVING REMOTE SYSTEMS IN THE NETWORK



MAPPING NETWORK DRIVE



Remote Path : \\sys3\c

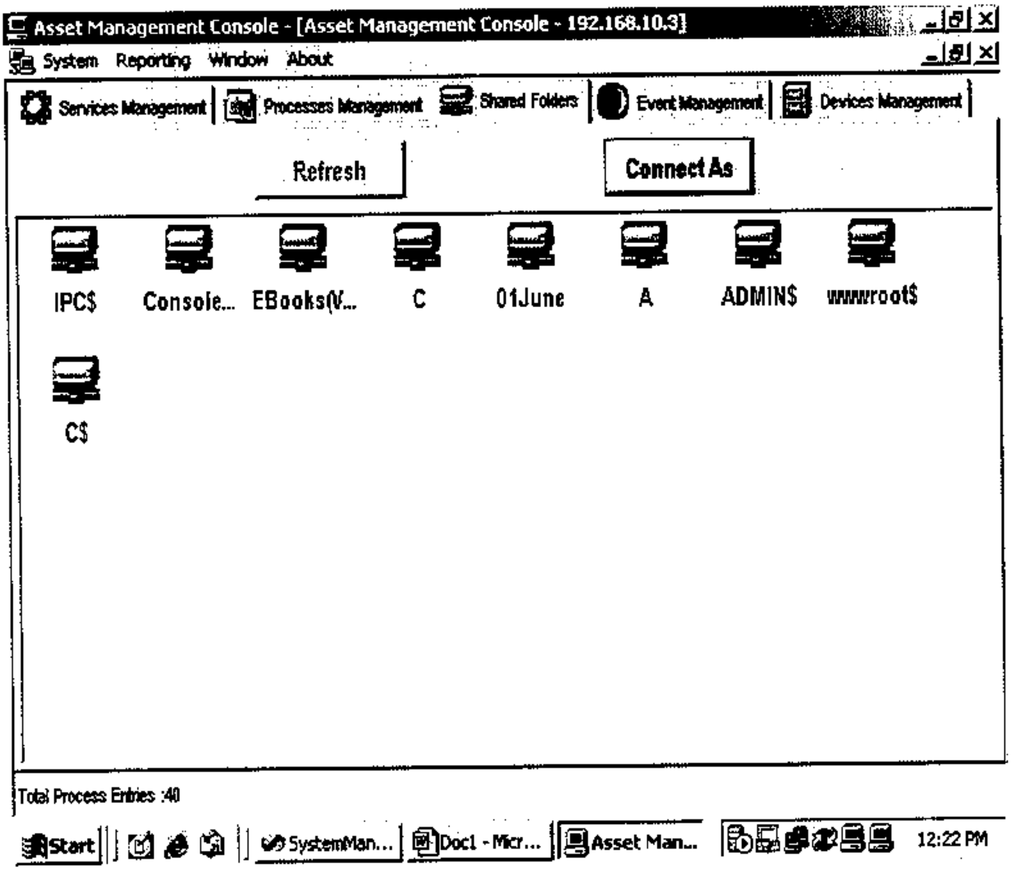
Drive Letter : [illegible]

Username : administrator

Password :

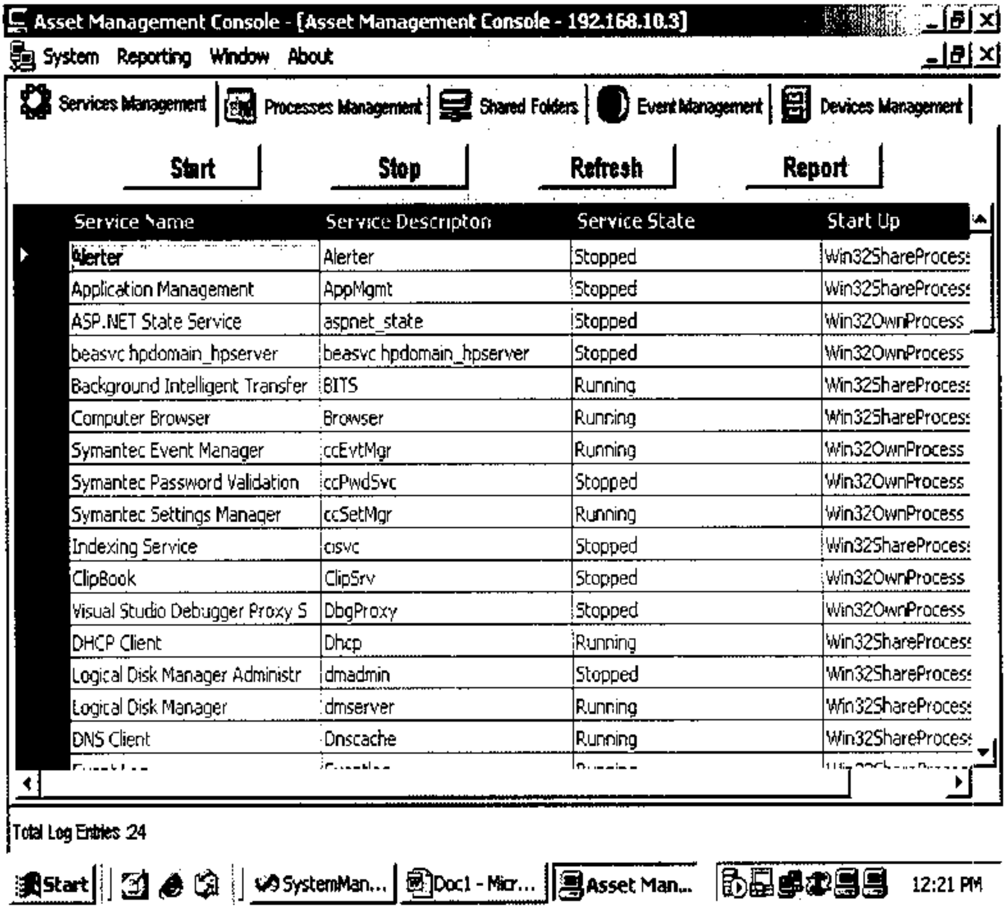
Done !

DISPLAYING SHARED FOLDERS & FILES



Total Process Entries :40

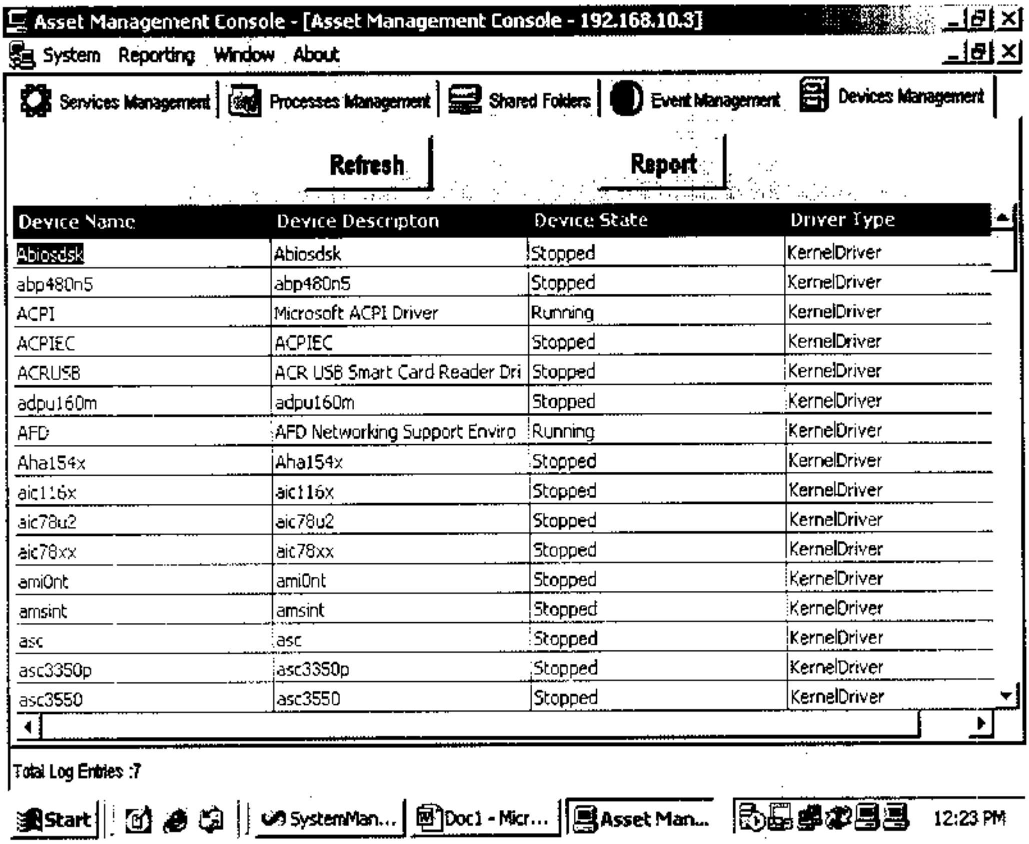
DISPLAYING THE SERVICE OF THE SYSTEM



Service Name	Service Descripton	Service State	Start Up
Alerter	Alerter	Stopped	Win32ShareProcess
Application Management	AppMgmt	Stopped	Win32ShareProcess
ASP.NET State Service	aspnet_state	Stopped	Win32OwnProcess
beasvc hpdomain_hpserver	beasvc hpdomain_hpserver	Stopped	Win32OwnProcess
Background Intelligent Transfer	BITS	Running	Win32ShareProcess
Computer Browser	Browser	Running	Win32ShareProcess
Symantec Event Manager	ccEvtMgr	Running	Win32OwnProcess
Symantec Password Validation	ccPwdSvc	Stopped	Win32OwnProcess
Symantec Settings Manager	ccSetMgr	Running	Win32OwnProcess
Indexing Service	cisvc	Stopped	Win32ShareProcess
ClipBook	ClipSrv	Stopped	Win32OwnProcess
Visual Studio Debugger Proxy S	DbgProxy	Stopped	Win32OwnProcess
DHCP Client	Dhcp	Running	Win32ShareProcess
Logical Disk Manager Administr	dmadmin	Stopped	Win32ShareProcess
Logical Disk Manager	dmserver	Running	Win32ShareProcess
DNS Client	Dnscache	Running	Win32ShareProcess

Total Log Entries :24

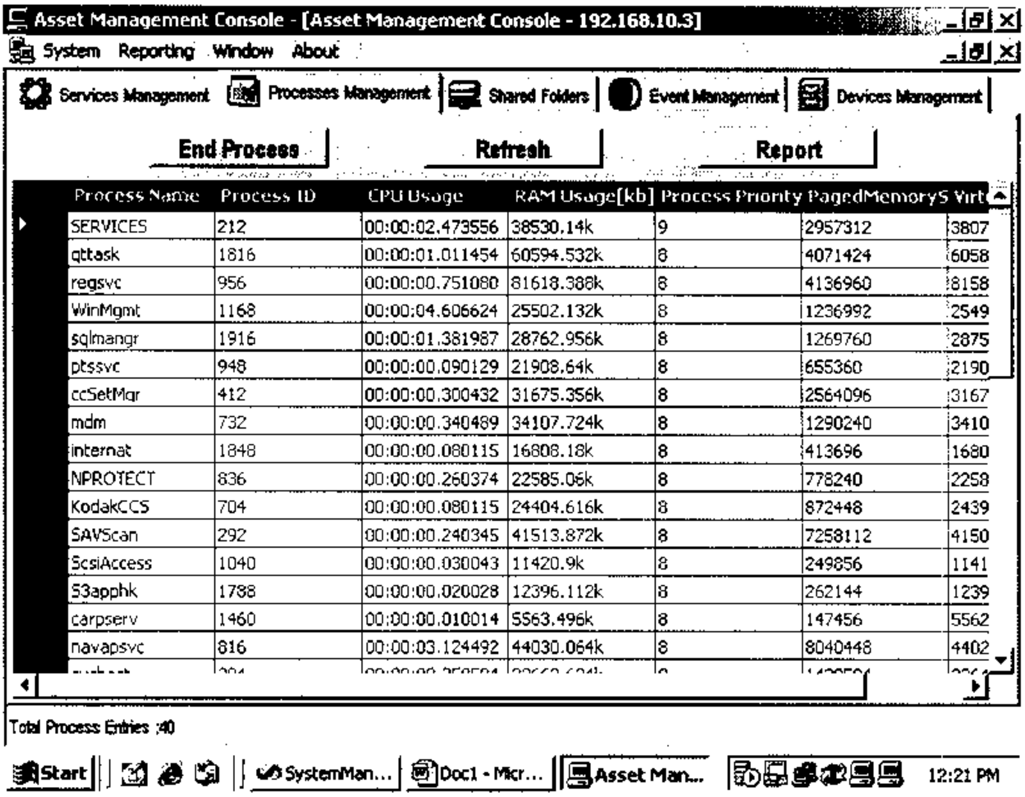
DISPLAYING DEVICE INFORMATION



Device Name	Device Descripton	Device State	Driver Type
Abiosdsk	Abiosdsk	Stopped	KernelDriver
abp480n5	abp480n5	Stopped	KernelDriver
ACPI	Microsoft ACPI Driver	Running	KernelDriver
ACPIEC	ACPIEC	Stopped	KernelDriver
ACRUSB	ACR USB Smart Card Reader Dri	Stopped	KernelDriver
adpu160m	adpu160m	Stopped	KernelDriver
AFD	AFD Networking Support Enviro	Running	KernelDriver
Aha154x	Aha154x	Stopped	KernelDriver
aic116x	aic116x	Stopped	KernelDriver
aic78u2	aic78u2	Stopped	KernelDriver
aic78xx	aic78xx	Stopped	KernelDriver
ami0nt	ami0nt	Stopped	KernelDriver
amsint	amsint	Stopped	KernelDriver
asc	asc	Stopped	KernelDriver
asc3350p	asc3350p	Stopped	KernelDriver
asc3550	asc3550	Stopped	KernelDriver

Total Log Entries :7

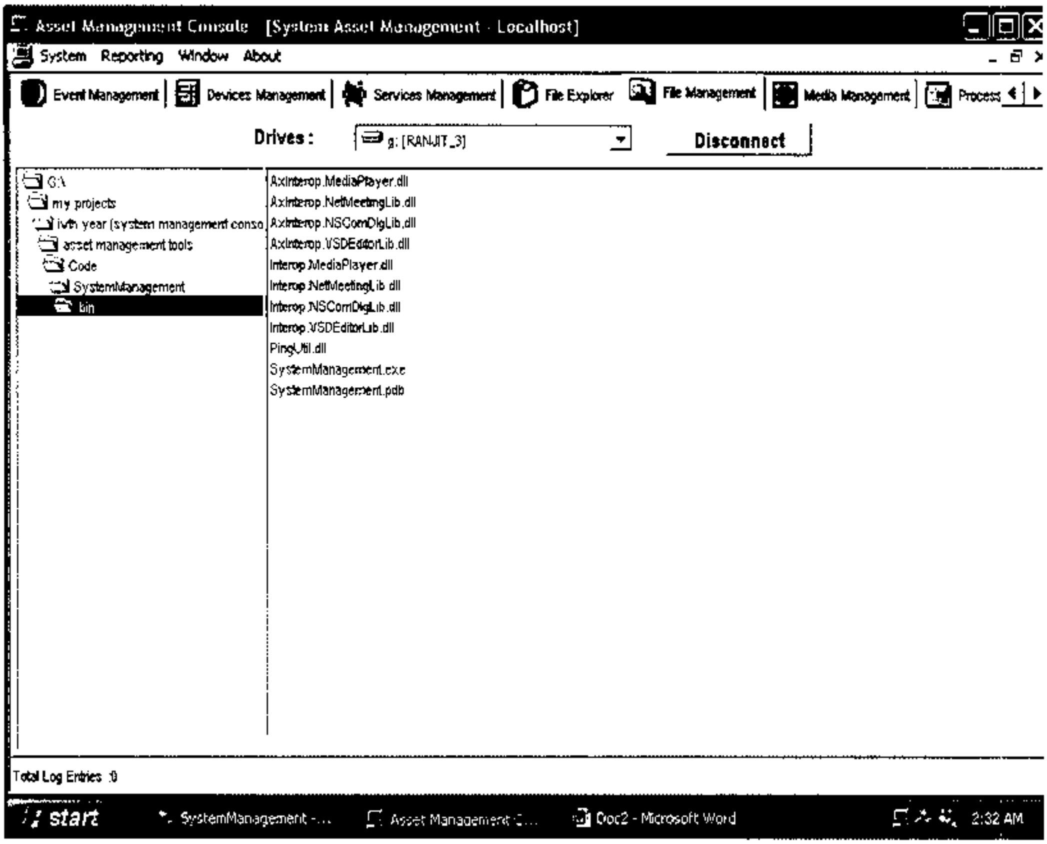
DISPLAYING PROCESS INFORMATION



Process Name	Process ID	CPU Usage	RAM Usage[kb]	Process Priority	PagedMemoryS	Virt
SERVICES	212	00:00:02.473556	38530.14k	9	2957312	3807
qttask	1816	00:00:01.011454	60594.532k	8	4071424	6058
regsvc	956	00:00:00.751080	81618.388k	8	4136960	8158
WinMgmt	1168	00:00:04.606624	25502.132k	8	1236992	2549
sqlmangr	1916	00:00:01.381987	28762.956k	8	1269760	2875
ptssvc	948	00:00:00.090129	21908.64k	8	655360	2190
ccSetMgr	412	00:00:00.300432	31675.356k	8	2564096	3167
mdm	732	00:00:00.340489	34107.724k	8	1290240	3410
internat	1848	00:00:00.080115	16808.18k	8	413696	1680
NPROTECT	836	00:00:00.260374	22585.06k	8	778240	2258
KodakCCS	704	00:00:00.080115	24404.616k	8	872448	2439
SAVScan	292	00:00:00.240345	41513.872k	8	7258112	4150
ScsiAccess	1040	00:00:00.030043	11420.9k	8	249856	1141
S3apphk	1788	00:00:00.020028	12396.112k	8	262144	1239
carpserv	1460	00:00:00.010014	5563.496k	8	147456	5562
navapsvc	816	00:00:03.124492	44030.064k	8	8040448	4402

Total Process Entries :40

FILE MANAGEMENT



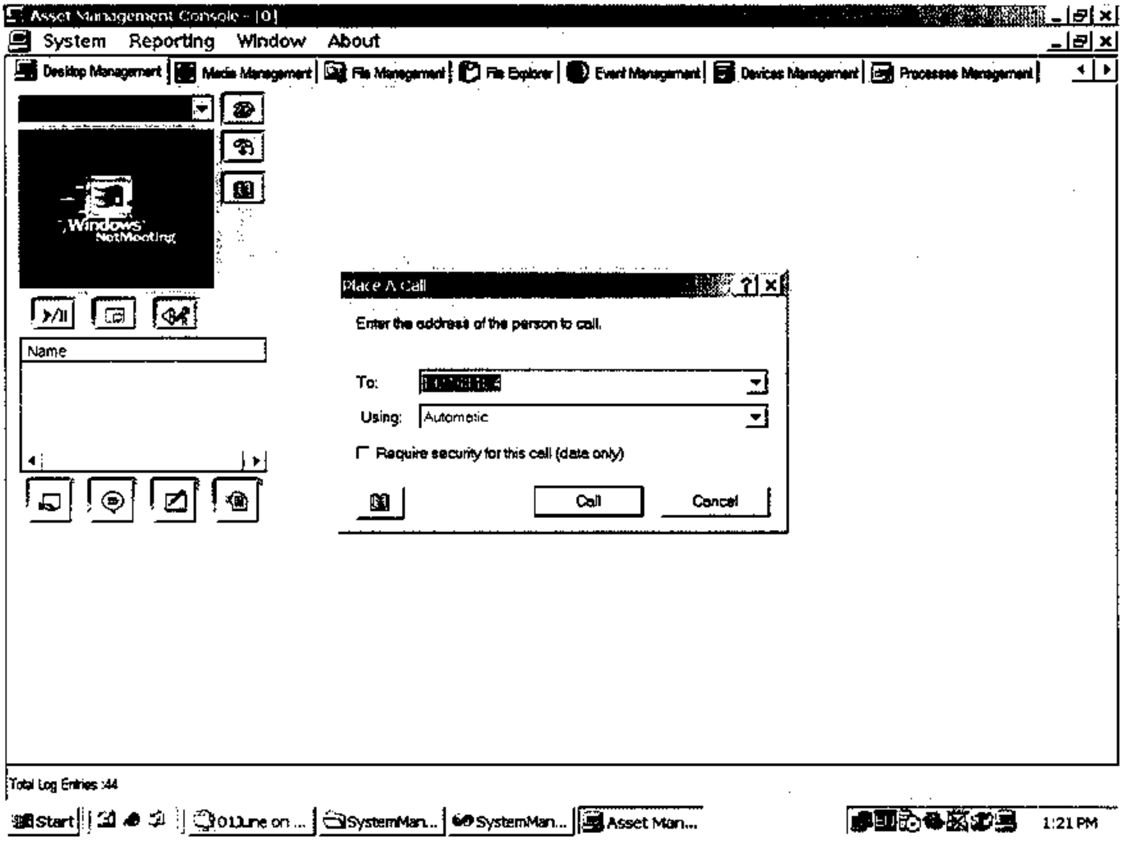
Drives : g: [RANJIT_3]

```
G:\
 my projects
  ivth year (system management conso
   asset management tools
    Code
     SystemManagement
      bin
```

```
AxInterop.MediaPlayer.dll
AxInterop.NetMeetingLib.dll
AxInterop.NSComDlgLib.dll
AxInterop.VSDEditorLib.dll
Interop.MediaPlayer.dll
Interop.NetMeetingLib.dll
Interop.NSComDlgLib.dll
Interop.VSDEditorLib.dll
PingUtil.dll
SystemManagement.exe
SystemManagement.pdb
```

Total Log Entries :0

MAKING A CALL TO REMOTE SYSTEM



Place A Call

Enter the address of the person to call.

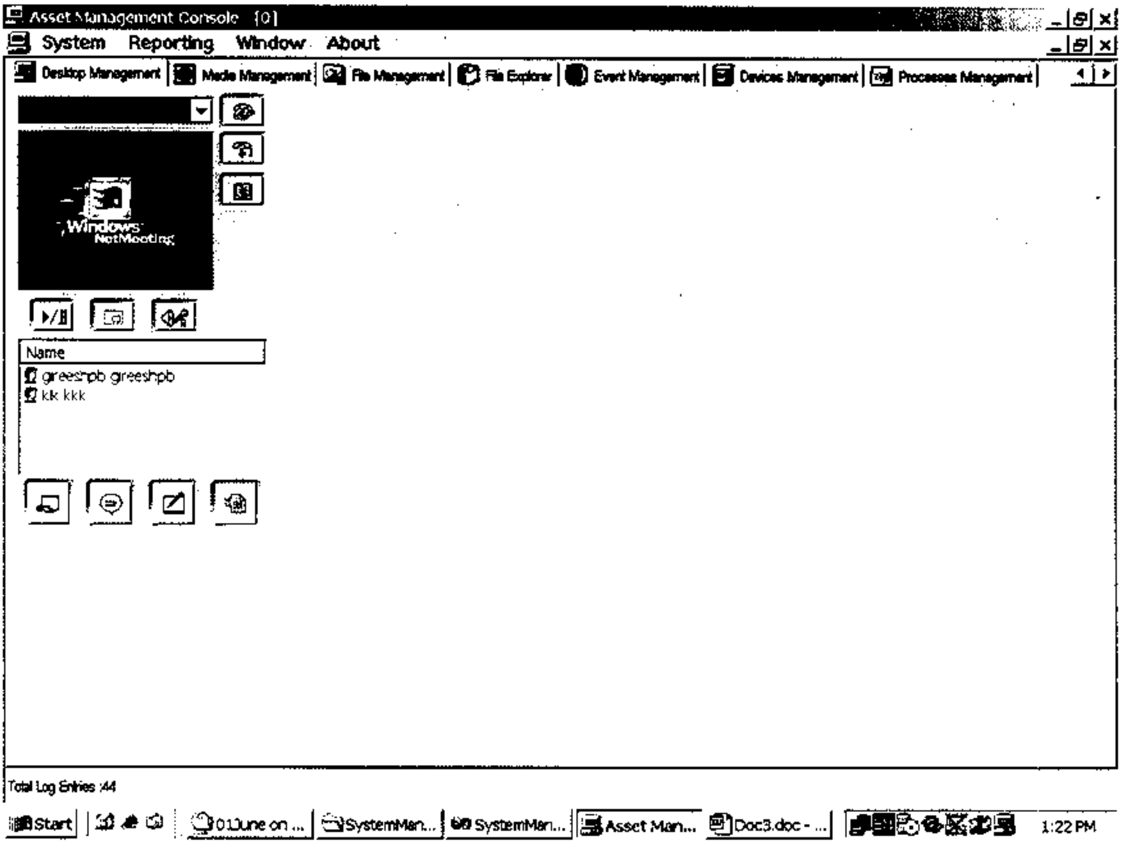
To: [illegible]

Using: Automatic

□ Require security for this call (data only)

Total Log Entries :44

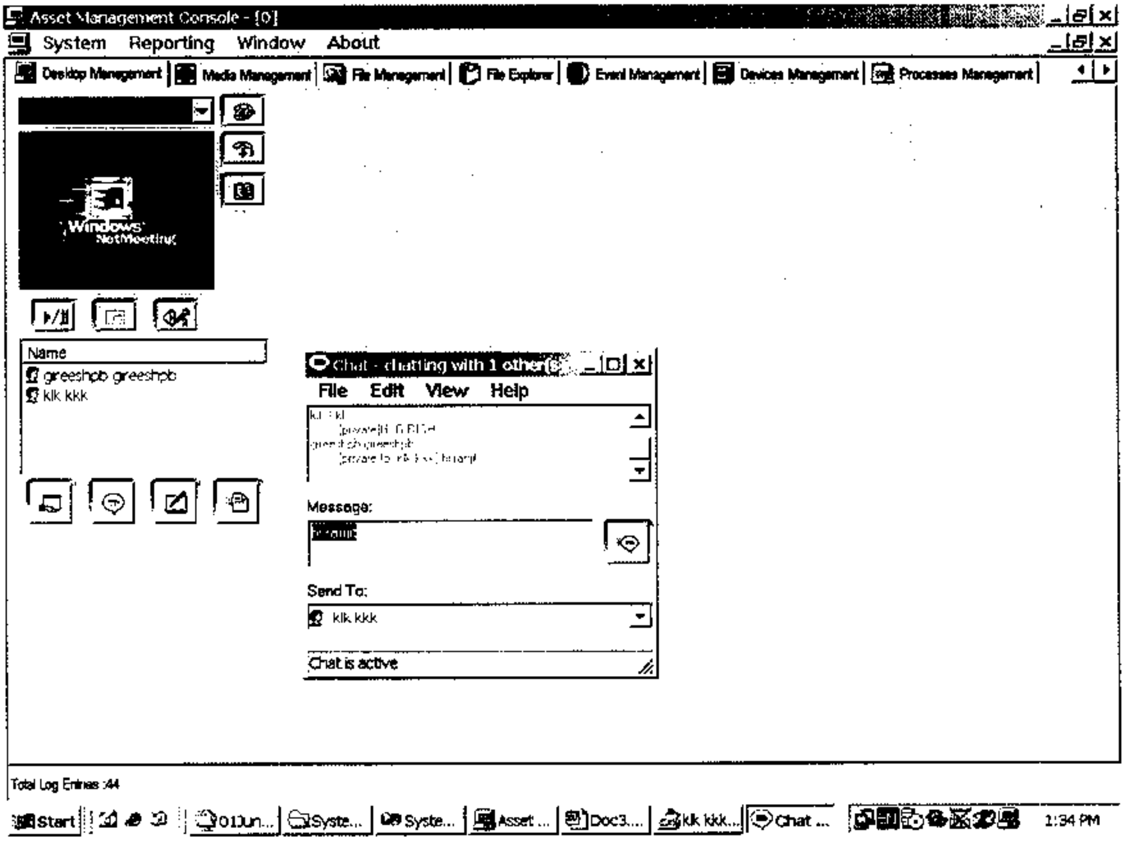
USERS IN THE LIST



greeshpb greeshpb
kk kkk

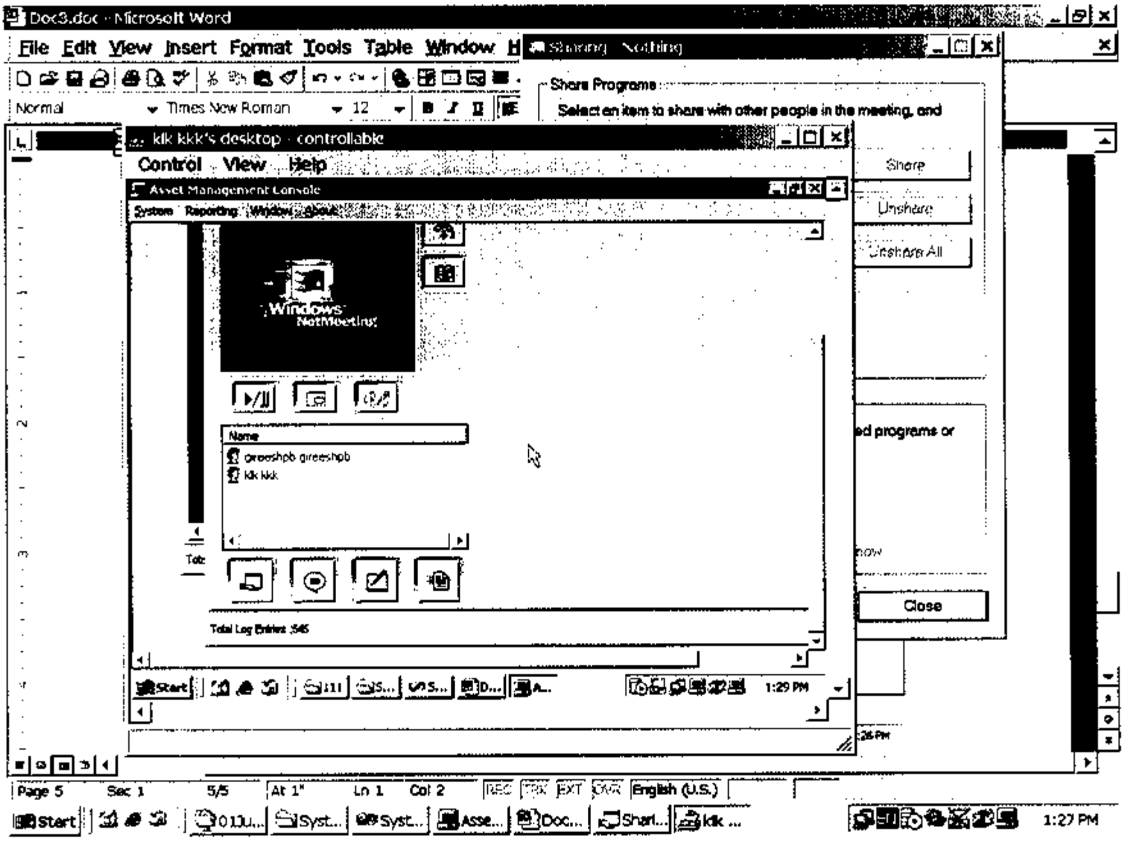
Total Log Entries :44

CHATTING BETWEEN TWO USER



[illegible]

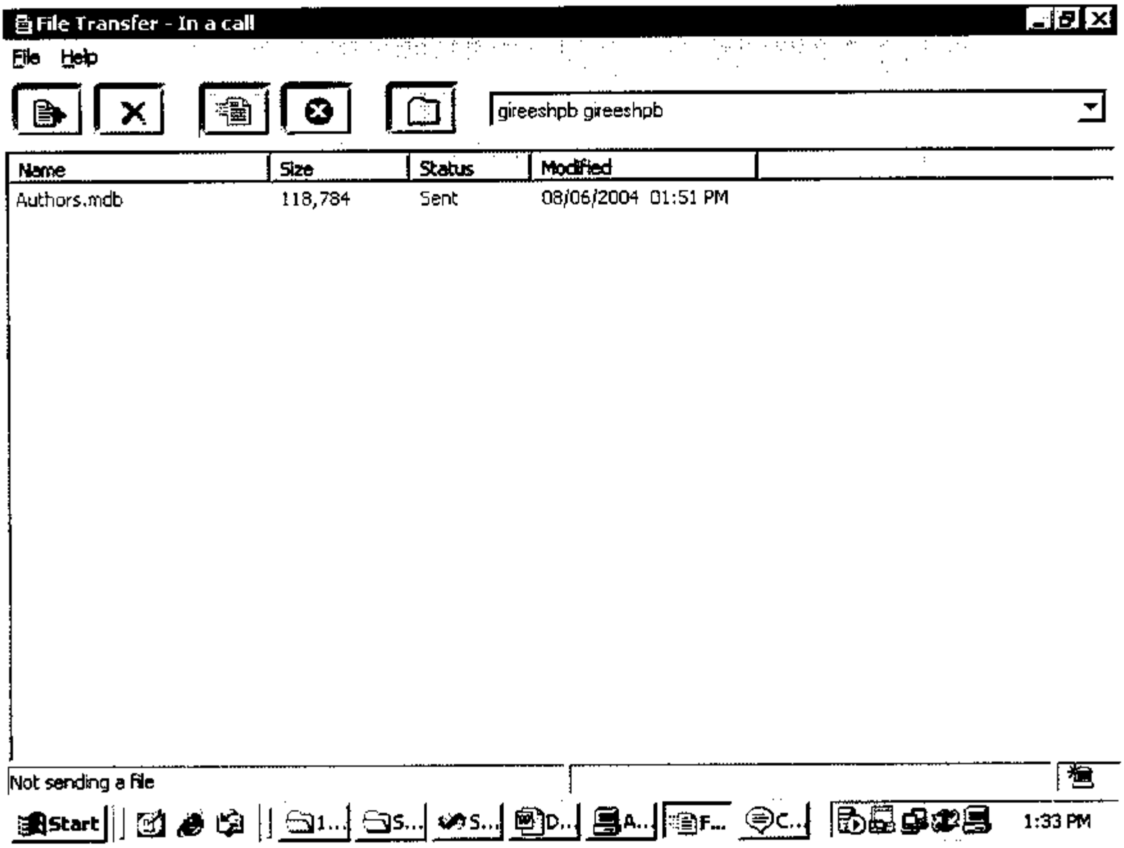
CAPTURING CLIENT DESKTOP


Doc3.doc - Microsoft Word
File Edit View Insert Format Tools Table Window H
Normal
Times New Roman
12
Sharing - Nothing
Share Programs
Share
Unshare
Unshare All
Close
klk kkk's desktop - controllable
Control View Help
Asset Management Console
System Reporting Window About
Windows NetMeeting
Name
greeshpb greeshpb
klk kkk
Start
1:29 PM
Page 5 Sec 1 5/5 At 1" Ln 1 Col 2 REC TRK EXT OVR English (U.S.)
1:27 PM

Select an item to share with other people in the meeting, and

Total Log Entries :545

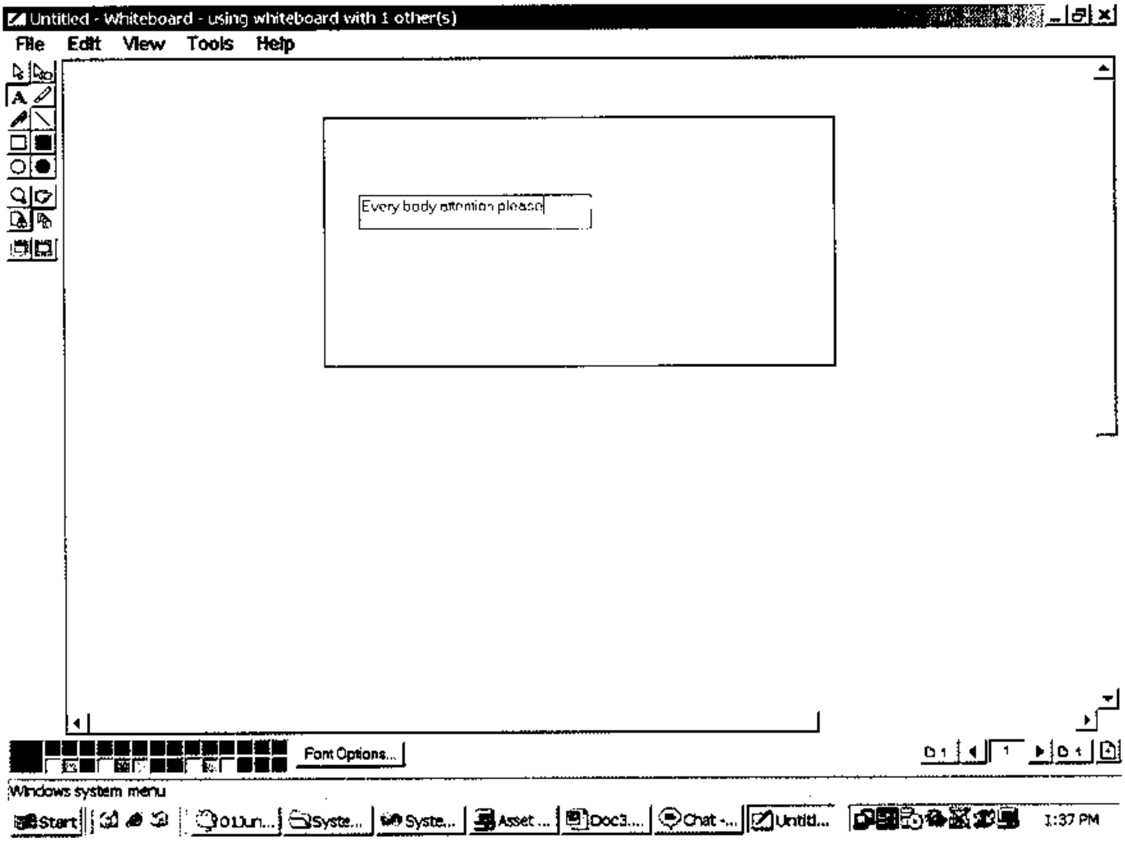
FILE TRANSFER



Name	Size	Status	Modified
Authors.mdb	118,784	Sent	08/06/2004 01:51 PM

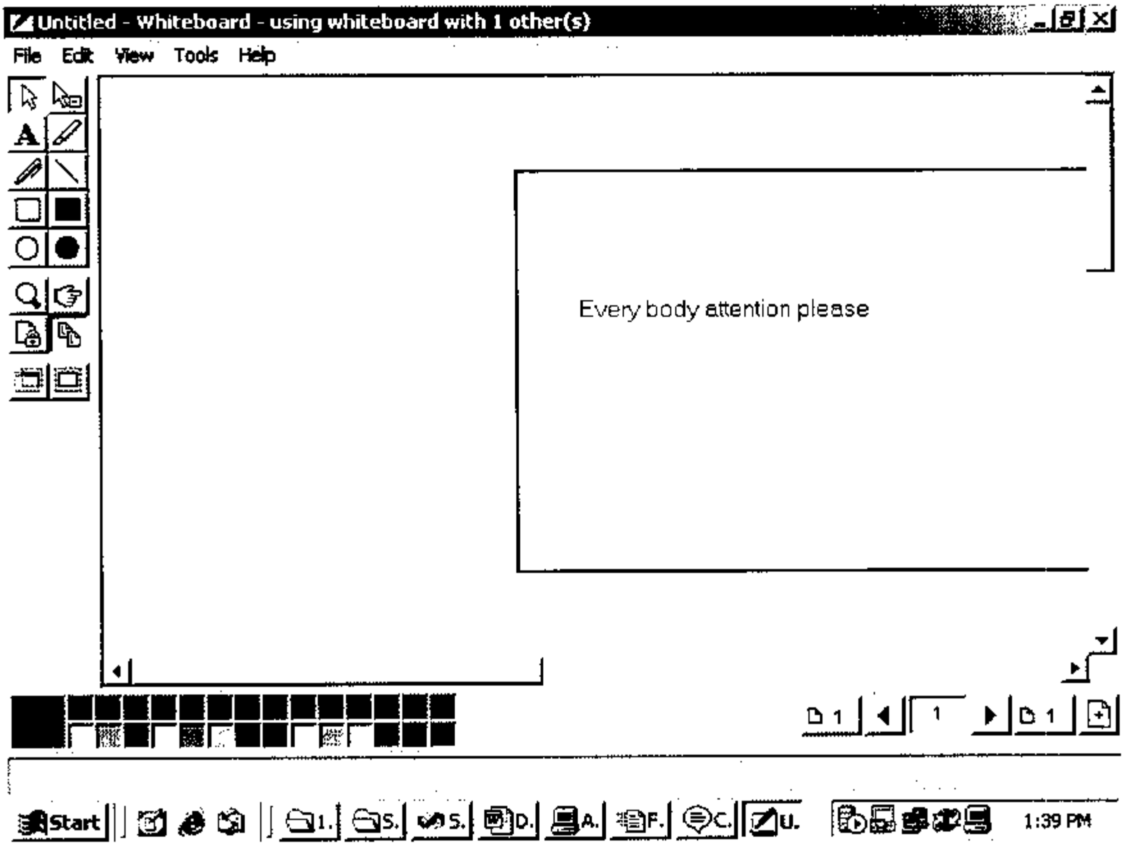
Not sending a file

WHITE BOARD USED IN SERVER



Every body attention please

WHITE BOARD USED IN CLIENT



Every body attention please

REFERENCES

REFERENCES

1. Roger S.Pressman, “Software Engineering-A Practioners Approach”, McGraw Hill International Editions, Fouth Edition, 1990.

2. Elias M.Award, “System Analysis and Design”, Galgotia Publications, Second Edition, 1995.

3. Lee, “Introducing System Analysis and Design”,Galgotia Book Source,1980.

4. Fred Barewell, Richard Blair, Richard Case. “Professional VB.NET “, Wrox Publication, second Edition, 2004