A PROJECT REPORT ON

"ADVANCED CLIENT MANAGEMENT TOOL FOR DISKLESS WORKSTATIONS ON LINUX PLATFORM OVER LTSP"

By

ANSH BAHRI (B80334204)

JAYESH CHAUDHARI (B80334220)

AKSHAY CHAVAN (B80334222)

SWAPNIL GUJAR(B80334246)

Under the Guidance of

Mrs. Sonal Gore

IN THE PARTIAL FULFILMENT OF

B.E. (COMPUTER ENGINEERING)

DEGREE OF UNIVERSITY OF PUNE,

MAY/JUNE-2013

DEPARTMENT OF COMPUTER ENGINEERING,

PCET'S PIMPRI CHINCHWAD COLLEGE OF ENGINEERING,PUNE

Sector 26, Pradhikaran, Nigdi,

Pune - 411044

Examination Approval Sheet

The Project Report entitled

"ADVANCED CLIENT MANAGEMENT TOOL FOR DISKLESS WORKSTATIONS ON LINUX PLATFORM OVER LTSP"

By

ANSH BAHRI (B80334204)

JAYESH CHAUDHARI (B80334220)

AKSHAY CHAVAN (B80334222)

SWAPNIL GUJAR(B80334246)

is approved for the degree of

B.E. Computer Engineering University of Pune

at

Pimpri Chinchwad College of Engineering

Examiners :

External Examiner : _____

Internal Examiner : _____

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING,PUNE

SECTOR 26,NIGDI,PRADHIKARAN,PUNE 411044

Department of Computer Engineering

CERTIFICATE

This is to certify the project report entitled

"ADVANCED CLIENT MANAGEMENT TOOL FOR DISKLESS WORKSTATIONS ON LINUX PLATFORM OVER LTSP"

Submitted By

ANSH BAHRI (B80334204)

JAYESH CHAUDHARI (B80334220)

AKSHAY CHAVAN (B80334222)

SWAPNIL GUJAR (B804334246)

is approved by "Mrs. Sonal Gore" for submission. It is certified further that, to the best of my knowledge, the report represents work carried out by the students as the partial fulfillment for the BE degree in Computer Engineering as prescribed by the University of Pune for the academic year 2012-13.

(Mrs. Sonal Gore)
Guide

(Prof. Dr. J S UMALE)
Head, Computer Engineering

(Dr. A. M. FULAMBARKAR)
Principal,
Pimpri Chinchwad College of Engineering Pune-44

Place : Pune

Date :

Acknowledgments

We express our sincere thanks to our Guide **Mrs. Sonal Gore**,for his/her constant encouragement and support throughout our project, especially for the useful suggestions given during the course of project and having laid down the foundation for the success of this work.

We would also like to thank our Project Coordinator **Mrs. Deepa Abin**,for her assistance, genuine support and guidance from early stages of the project. We would like to thank **Prof. Dr. J.S. Umale**, Head of Computer Department for his unwavering support during the entire course of this project work. We are very grateful to our Principal **Dr.A.M.Fulambarkar** for providing us with an environment to complete our project successfully. We also thank all the staff members of our college and technicians for their help in making this project a success. We also thank all the web committees for enriching us with their immense knowledge. Finally, we take this opportunity to extend our deep appreciation to our family and friends, for all that they meant to us during the crucial times of the completion of our project.

ANSH BAHRI
JAYESH CHAUDHARI
AKSHAY CHAVAN
SWAPNIL GUJAR

Contents

Abstract

The Indian education suffers from infrastructure deficiencies that block the technologies and knowledge. Similar conditions also exist in the other developing countries. Most of schools and colleges do not have funds to buy costly licensed software and with strict IT laws it becomes very difficult to meet both ends for the authorities. Free and Open Source Software (FOSS) is the solution for this problem, The open source movement has many solutions, one such solution is Linux Terminal Server Project (LTSP). There are two main aims of this project:

1] To provide the disk-less workstations that boot from network server running Ubuntu.

2] To provide a client management tool for Administrator to have watch on clients from single place.

The disk-less workstations can be any 486 / low end Pentium's or even any 386 systems.The Advanced client tool will benefit administrators and institutes to a large extent.

Keywords:

1. LTSP: Linux Terminal Server Project.
2. PXE: Pre eXecution Enviornment
3. NTP: Network Time Protocol.
4. DNS: Domain Name System.
5. DHCP: Dynamic Host Control Protocol.
6. ACMT: Advanced Client Management Tool.
7. TFTP: Trivial File Transfer Protocol
8. LDM: Local Data Manager
9. IDE: Integrated Development Environment
10. JVM: Java Virtual Machine
11. GUI: Graphical User Interface
12. JDK:Java Development Kit

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

1. The goal of Advanced Client Management Tool on Diskless Workstation using LTSP is to provide client management to administrator on diskless workstations which will run on LINUX utility LTSP.
2. Our goal is to reduce costing of hardware and also efforts of human beings.
3. The administrator can manage number of workstations from single Server.
4. Administrator can install / uninstall software on all workstations from its server also it can generate/kill any process on any client.

1.2 Brief Description

There are two components of a network: hardware and software. This section will give an introduction to both.

1.2.1 Hardware

Networking works by breaking files and other data into little packets of information. These packets are transferred over a network. The difference between various types of networks is how they transfer packets. There are two types of networking hardware: wired

and wireless. An important fact to remember is that a network will be only as fast as the slowest part. Making sure that your network setup matches your intended use case is an important consideration in a LTSP network.

1.2.2 Software

The most common network infrastructure services include: Each computer on a network needs a unique identifier called an IP address. The IP address allows packets to be directed to the computer, much like a street address allows mail to be delivered to the correct house. An IP address follows a specific form: four groups of digits forming a number from 0 to 255. For example, a local IP address might be 192.168.2.50. Your computer talks to a DNS server every time you refer to another computer system with a name instead of an IP address. For example: www.ltsp.org, wikipedia.org, and google.com are all DNS hostnames.

1.3 Problem Definition

Implementing Advanced Client Management Tool(ACMT) for disk-less workstation using LTSP over Edubuntu.We are making ACMT tool in Java programming language which is platform independent.This tool is heterogeneous and future compatible.

Chapter 2

Literature Survey

2.1 LTSP(Linux Terminal Server Project)

LTSP is a flexible, cost effective solution that is empowering schools, businesses, and organizations all over the world to easily install and deploy Diskless Workstations.

LTSP workstations can run applications from Linux and Windows servers. Linux based Diskless Workstations have proven to be extremely reliable because tampering and viruses are virtually non-existent.

BENEFITS OF LTSP:

1. Reduced Costs
2. No Licensing Fees
3. Less Maintenance Required
4. Secure

2.2 PXE booting

Part of the 'Wired for Management' specification from the late 1990's included a specification for a boot ROM technology known as the Preboot execution Environment, commonly abbreviated as PXE. It enables computers to boot from a network image and allows computers to function without local storage of any kind. It also enables system administrators to centrally configure how computers should boot. This is the default method that LTSP uses to boot thin and fat clients, although it is also possible to boot clients that do not support PXE from local media.

A PXE ROM is available in most wired network cards nowadays. You often have to change the settings in the BIOS of a computer to make network the first boot device. For onboard network cards you often separately have to enable the PXE ROM in the BIOS. During successful PXE boot, the network cards ROM requests a DHCP lease from the network. The DHCP server tells the client where on the network it can find an image to boot from, which the PXE environment then downloads to start the computer. Wireless network cards generally do not support PXE.

2.3 Client Management Tools

The client management tool provided in LTSP is Epoptes. This software can show you client screens and administrator can understand what a client is working on. The software has no other use like it cannot access any client machine or cannot execute/kill any process on client system

2.4 Hardware components

Wired Wired networking transfers packets over a cable that resembles a telephone cord, but with more wires. Wired networks can transfer packets at one of three possible speeds: 10 Mbit/sec, 100 Mbit/sec, or (Gigabit) 1000 Mbit/sec. A network is only useful if it can connect multiple computers. There are some pieces of hardware that allow multiple computers to be connected in a network. They look alike, but they function differently and, likewise, operate at different speeds.

Hub A hub is the simplest way to connect multiple computers. A hub has a lot of ports in the front and usually has several small lights corresponding to each port. The hub takes a message it receives on one port and re-sends it to all the ports. As a result, only one port can talk at a time.

Switch A switch looks a lot like a hub; it has a lot of ports in the front and usually has several small lights corresponding to each port. However, a switch is unlike a hub because it only makes a connection between the ports it needs to. A switch can have multiple connections at the same time. This allows a switch to be faster than a hub.

Router A router is used to make a connection between two networks. Routers are also commonly used to connect a LAN (local area network) to the Internet.

Wireless Some people may wish to try using LTSP in a wireless environment, for various reasons. This represents some challenges.

Wireless networks typically have more latency than wired networks, which generally makes interactive programs feel slow and unresponsive. As well, wireless adaptors cannot directly PXE boot, as you need to set things such as ESSID, keys, etc., which wouldnt be there in a PXE capable card.

However, for those wishing to use LTSP wirelessly, it is still possible, but requires more hardware. Wireless bridge boxes are available, which contain both an Ethernet and a wireless network connection. One can typically connect to them like a small Internet router box, and program them with the information pertinent to your network. You can then use a standard wired network card connected directly to the bridge, and the bridge itself will handle the wireless part.

This method has been used with success by users of LTSP in the past. The latency of wireless makes the experience slower; however, depending on the application you wish to use, you may find it acceptable.

2.5 Software Components

DHCP (Dynamic Host Configuration Protocol) Each computer on a network needs a unique identifier called an IP address. The IP address allows packets to be directed to the computer, much like a street address allows mail to be delivered to the correct house. An IP address follows a specific form: four groups of digits forming a number from 0 to 255. For example, a local IP address might be 192.168.2.50.

For convenience, a computer's IP address can be given by a server running the Dynamic Host Configuration Protocol (DHCP) service. DHCP automatically provides network settings to the computers on the network. With DHCP, there is no need to keep track of each computer's IP address.

DNS (Domain Name System) DNS is a service that runs on a server, and it is like a phone book for computers, except that it stores IP addresses instead of phone numbers. Your computer talks to a DNS server every time you refer to another computer system with a name instead of an IP address. For example: www.ltsp.org, wikipedia.org, and google.com are all DNS hostnames.

NTP (Network Time Protocol) NTP is a service that runs on a server and allows other computers to synchronize their clocks. The server synchronizes with an extremely accurate atomic clock, and then the clients synchronize with the server.

Web Server A Web server answers queries using protocols such as HTTP, and sends content such as web pages back to clients. Your Web browser almost exclusively talks to Web servers. Server like APACHE TOMCAT 6.0 will be used.

Chapter 3

Software Requirements Specification

3.1 Introduction

The product will be very helpful in large Institutes/Organizations/IT Industries where large numbers of computers are to be maintained. Large amount of cost is spent on maintenance of such systems and is very hard to check every computer for security weaknesses, application updates, OS updates, and Local hardware access. Our product will check only one system i.e. the Server as all clients are diskless and will access Operating System via Server which means the maintenance cost and time is saved by a big margin. All systems can be monitored by the administrator, No issue of virus attack. This kind of architecture ensures data security and will prevent users from copying information out of the system.

3.1.1 Purpose

This document has complete system information about Advanced Client Management Tool for Disk-less Workstations on Linux Platform over LTSP. The system is about different client systems that have no pre-installed OS in it and these Virtualized Clients accessing the Server for its Operating System and Application plus memory storage.

3.1.2 Intended audience and reading suggestion

The audience benefited by our project will be Institutes/Organizations/Industries where data security, user management, application management and backup are crucial focus/points large number of computers is to be maintained.

3.2 Overall Description

3.2.1 Product Perspective

Its a new project wherein an advanced client management tool is developed for especially educational institutions; it could be used in offices and cyber cafes as per need. The server machine can monitor and manage (open and close applications) at runtime. The server is virtualised, runs the diskless client (clients do not have any physical hard disk with no flash) from there with high speed network. The diskless clients will access the server to fetch Linux based OS over the network. The diskless client can perform all normal tasks it performs on normal Linux Operating System. The administrator can monitor every activity of each client. Installation of any software is to be done on server only once and updates are automatically made over the client machines. Additionally all the advantages of using a Linux based system are collectively provided with it. In short, the project saves two very essential components Time and Money.

3.2.2 Product Features

1. The Server can monitor and manage access right for the client on runtime.

2. The administrator can add and remove a diskless client. Automatic shutdown of client machines is also provided for inactive clients to save power and conserves important resources.

3. Backup needs to be taken only once on server as all user data reside on server side only.

4. The client activities are stored on a log which is maintained over the server machine. All the currently running tasks of the client machine are visible over the server in a task manager format.

3.2.3 Assumptions and Dependency

The test cases can be generated as fast as possible by the user only if the bandwidth is maximum. i.e. if 1 test case require 10 ms to generate then for 1000 test cases it requires 10ms * 1000 so number of test cases is directly proportional to bandwidth.

3.3 System Description

3.3.1 Process Flow

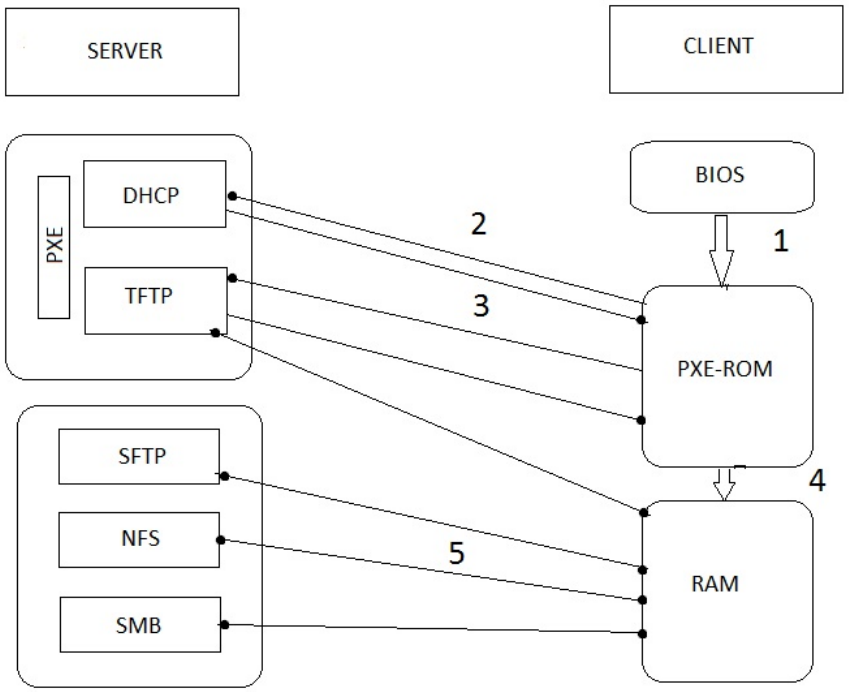


Figure 3.1: Process Flow

Process Flow for Static Testing

1. Diskless Workstation boot via a protocol called PXE (Pre-execution Environment).
2. PXE requests an IP address from a local DHCP server.
3. The DHCP server passes additional parameters to the thin-client and downloads a Linux file system image via TFTP into a RAM disk on the client itself.
4. The Diskless Workstation then boots the downloaded Linux image, detects hardware, and connects to the LTSP server's X4 session (normally handled by LDM).
5. Pre-Execution Environment or, more commonly, "Network Boot" Trivial File Transfer Protocol.

6. X Window System, the default Linux graphical windowing system.

7. LTSP Display Manager, the default login manager for LTSP.

3.3.2 Component Description

Graphical user Interface: Swing-Netbeans:

The Netbeans IDE is written in Java and can run anywhere a JVM is installed, including Windows, Mac OS, Linux, and Solaris. A JDK is required for Java development functionality, but is not required for development in other programming languages. The Netbeans platform allows applications to be developed from a set of modular software components called modules. Applications based on the Netbeans platform (including the Netbeans IDE) can be extended by third party developers. The GUI design-tool enables developers to prototype and design Swing GUIs by dragging and positioning GUI components.

Data library/ Meta Data:- JAVA Class library

The Java Class Library is a set of dynamically loadable libraries that Java applications can call at run time. Because the Java platform is not dependent on any specific operating system, applications cannot rely on any of the existing libraries. Instead, the Java platform provides a comprehensive set of standard class libraries, containing much of the same reusable functions commonly found in modern operating systems. The Java class libraries serve three purposes within the Java platform:

1. Like other standard code libraries, they provide the programmer a well-known set of useful facilities, such as container classes and regular expressions.

2. In addition, the class libraries provide an abstract interface to tasks that would normally depend heavily on the hardware and operating system. Tasks such as network access and file access are often heavily dependent on the native capabilities of the platform.

3. Finally, some underlying platforms may not support all of the features a Java application expects. In these cases, the class libraries can either emulate those features using whatever is available, or provide a consistent way to check for the presence of

a specific feature.

Main features of Java Class Library:

Features of the Class Library are accessed through classes grouped by packages.

1. **java.lang** contains fundamental classes and interfaces closely tied to the language and runtime system.
2. **I/O and networking:** access to the platform file system, and more generally to networks, is provided through the java.io, java.nio, and java.net packages. For networking, SCTP is available through com.sun.nio.sctp.
3. **Mathematics package:** java.math provides regular mathematical expressions, as well as arbitrary-precision decimals and integers numbers.
4. **Collections and Utilities:** provide built-in Collection data structures, and various utility classes, for Regular expressions, Concurrency, logging and Data compression.
5. **GUI and 2D Graphics:** the AWT package (java.awt) supports basic GUI operations and binds to the underlying native system. It also contains the 2D Graphics API. The Swing package (javax.swing) is built on AWT and provides a platform independent widget toolkit, as well as a Pluggable look and feel. It also deals with editable and non-editable text components.
6. **Applets:** java.applet allows applications to be downloaded over a network and run within a guarded sandbox.
7. **Databases:** access to SQL databases is provided through the java.sql package.

I/O:- JAVA I/O file and Serialization

In computer science, in the context of data storage and transmission, serialization is the process of converting a data structure or object state into a format that can be stored (for example, in a file or memory buffer, or transmitted across a network connection link) and "resurrected" later in the same or another computer environment. When the resulting

series of bits is reread according to the serialization format, it can be used to create a semantically identical clone of the original object. For many complex objects, such as those that make extensive use of references, this process is not straightforward. Serialization of object oriented objects does not include any of their associated methods with which they were previously inextricably linked. This process of serializing an object is also called deflating or marshalling an object. The opposite operation, extracting a data structure from a series of bytes, is deserialization (which is also called inflating or unmarshalling).

Uses of Serialization:

Serialization provides:

1. a method of persisting objects which is more convenient than writing their properties to a text file on disk, and re-assembling them by reading this back in.
2. a method of remote procedure calls, e.g., as in SOAP
3. a method for distributing objects, especially in software componentry such as COM, CORBA, etc.
4. a method for detecting changes in time-varying data.

3.4 External Interface

3.4.1 User Interface

- Login Screen.
- Manage Commands.
- Activate System Monitor.
- Daemon Tool Window.
- View Log.
- Client Side console for communication with Server(s).

3.4.2 Hardware Requirements

- Server:4 GB RAM ,320 GB HDD,NIC
- Client:128 MB RAM.

3.4.3 Software Interfaces

- Operating System: Linux Operating System
- HTTP (Hyper text Transfer Protocol)
- Language: J2EE and Java
- Front End : Java Swing

3.4.4 Software Quality Attributes

1. Reliability.
2. Effectiveness.
3. Flexibility.
4. Usability.
5. Availability.
6. Interoperability.

3.5 Other Non-Functional Requirements

3.5.1 Performance Requirements

Any Transaction for small application should not exceed for 6-7 seconds.* For Complex Applications the transaction time should not exceed for 12-15 seconds.

3.5.2 Safety and Security Requirements

1. Backup copy of user folder on server should be created every 2 hrs. 2. Server must be fully accused with backup power supply. 3. Servers interface facing the client network usually should not be connected directly to internet. 4. Clients are not using local hardware, so external drives (USB drives) cannot be used to steal information. 5. User authentication is done on server and not on local m/c.

3.6 Analysis Models

3.6.1 Data Flow Diagrams

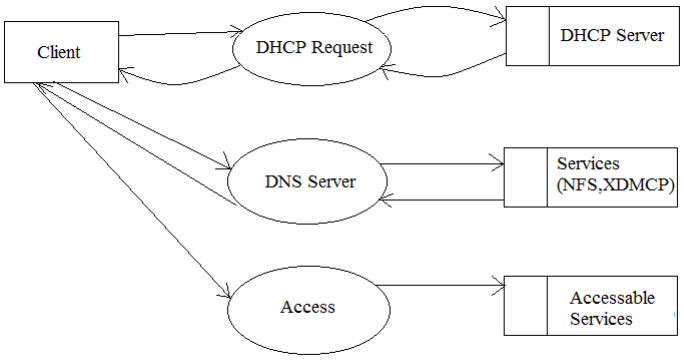


Figure 3.2: Data Flow Diagram

3.6.2 Class Diagrams

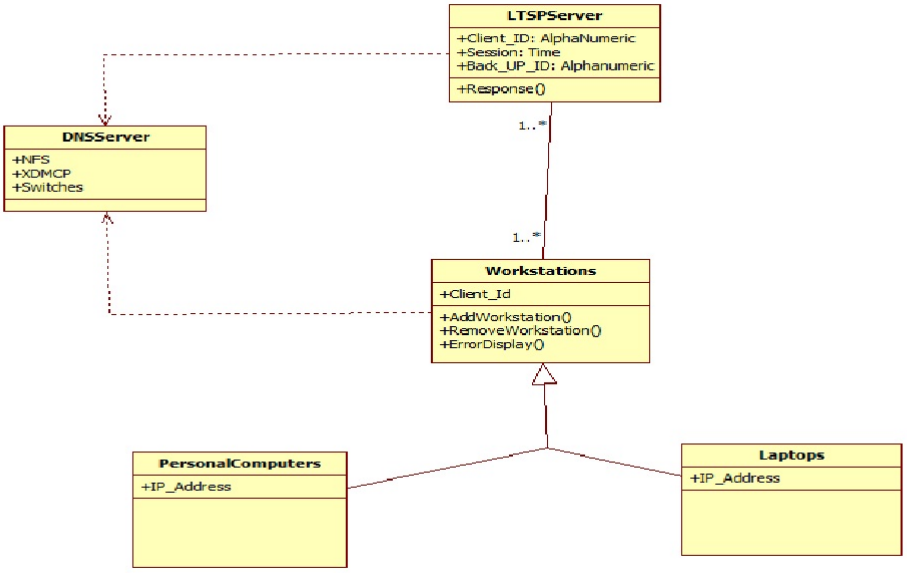


Figure 3.3: Class Diagram

3.7 System Implementation Plan

The development model that shall be followed for implementing the software is Incremental model.

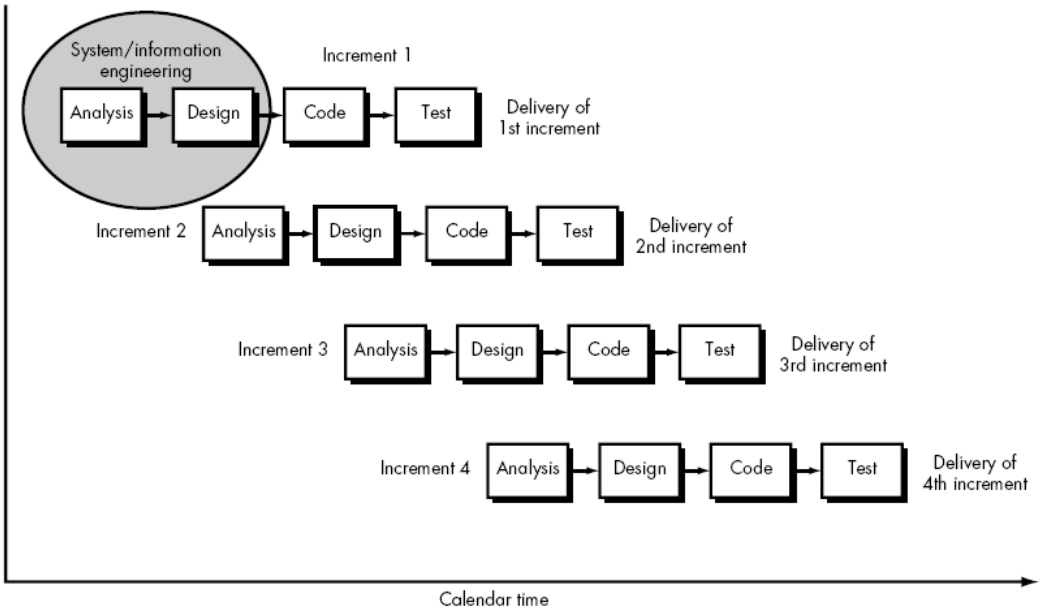


Figure 3.4: Incremental Model

Incremental model combines elements of the linear sequential model with the iterative philosophy of prototyping. The above shows, the incremental model applies linear sequences in a staggered fashion as calendar time progress. Each linear sequence produces as deliverable incremental of the software. For example word processing software developed using the incremental paradigm might deliver basic file management, editing, and document production functions in the first increment; more sophisticated editing and document production capabilities in the second increment; spelling and Software Engineering Concepts and Implementation Centre for Information Technology and Engineering, Monomania Sundaranar University 24 Grammar checking in the third increment; and advanced page layout capability in the fourth increment. It should be noted that the process flow for any increment can incorporate the prototyping paradigm. When an incremental model is used, the first increment is often a core product. That is basic requirements are addressed, but many supplementary features, some known, others unknown) remain undelivered. The core product is used by the customer (or undergoes detailed review.

As a result of use and/or evaluation, a plan is developed for the next increment. The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of additional features and functionality. This process is repeated following the delivery of each increment, until the complete product is produced. Early increments are stripped down versions of the final product but they do provide capability that serves the user and also provide a platform for evaluation by the user.

We are using Incremental Model in this project as it is easier to test and debug than other methods of Software Development because relatively smaller changes are made during each iteration. This allows for more targeted and rigorous testing of each element within the overall product.

Incremental development slices the system functionality into increments (portions). In each increment, a slice of functionality is delivered through cross-discipline work, from the requirements to the deployment. The unified process groups increments/iterations into phases: inception, elaboration, construction, and transition.

1. Inception identifies project scope, requirements (functional and non-functional) and risks at a high level but in enough detail that work can be estimated.

2. Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.

3. Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.

4. Transition delivers the system into the production operating environment.

Each of the phases may be divided into 1 or more iterations, which are usually time-boxed rather than feature-boxed. Architects and analysts work one iteration ahead of developers and testers to keep their work-product backlog full.

Chapter 4

System Design

4.1 System Architecture

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structure of the system which comprises system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them, and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system.

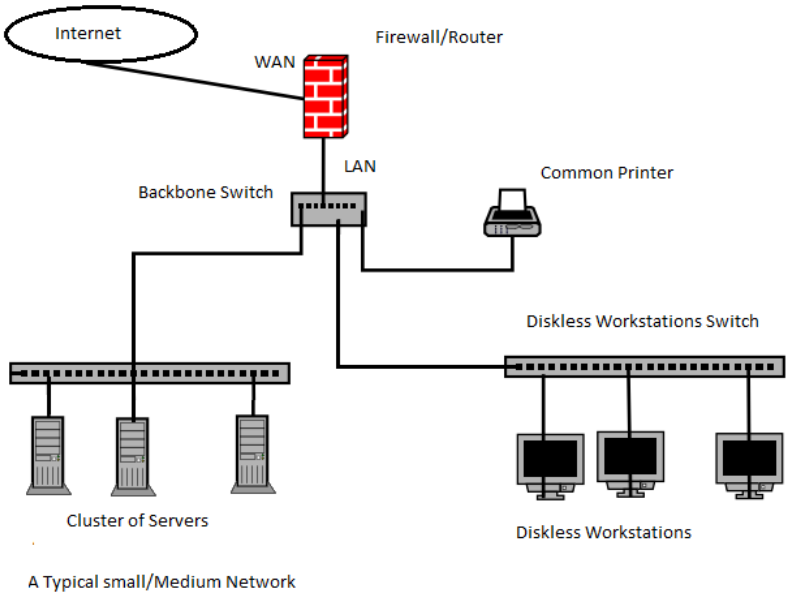


Figure 4.1: System Architecture

4.2 UML Diagrams

4.2.1 Use-case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

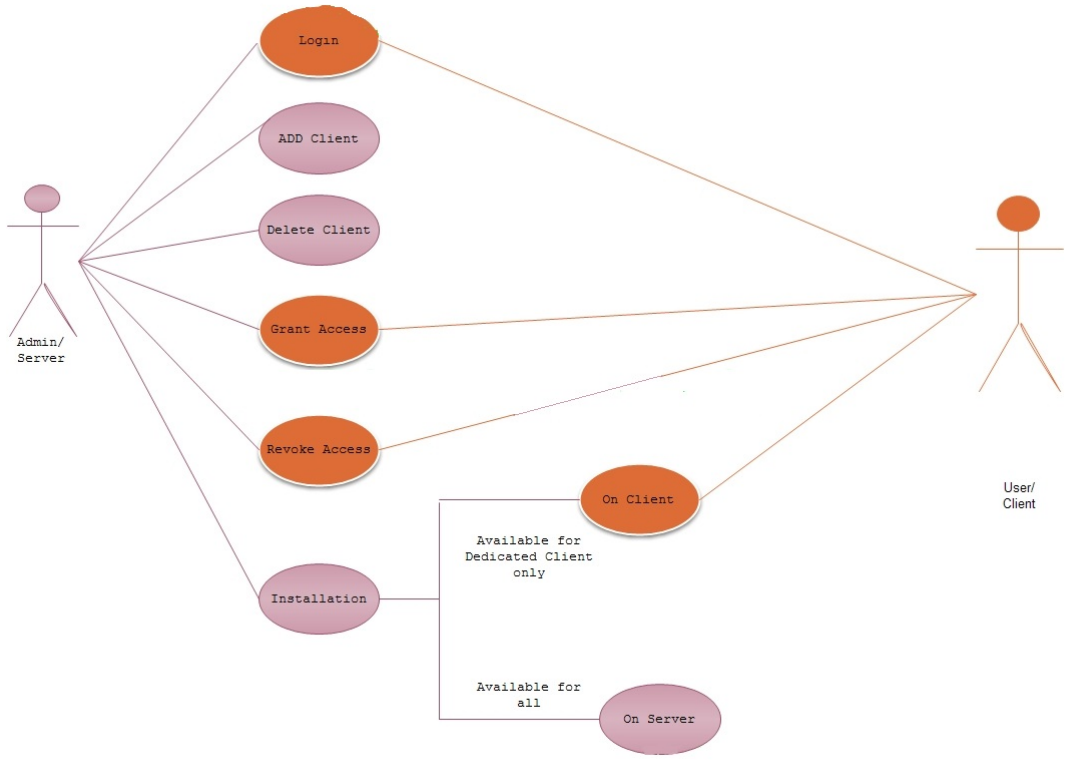


Figure 4.2: Use-case Diagram

4.2.2 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development.

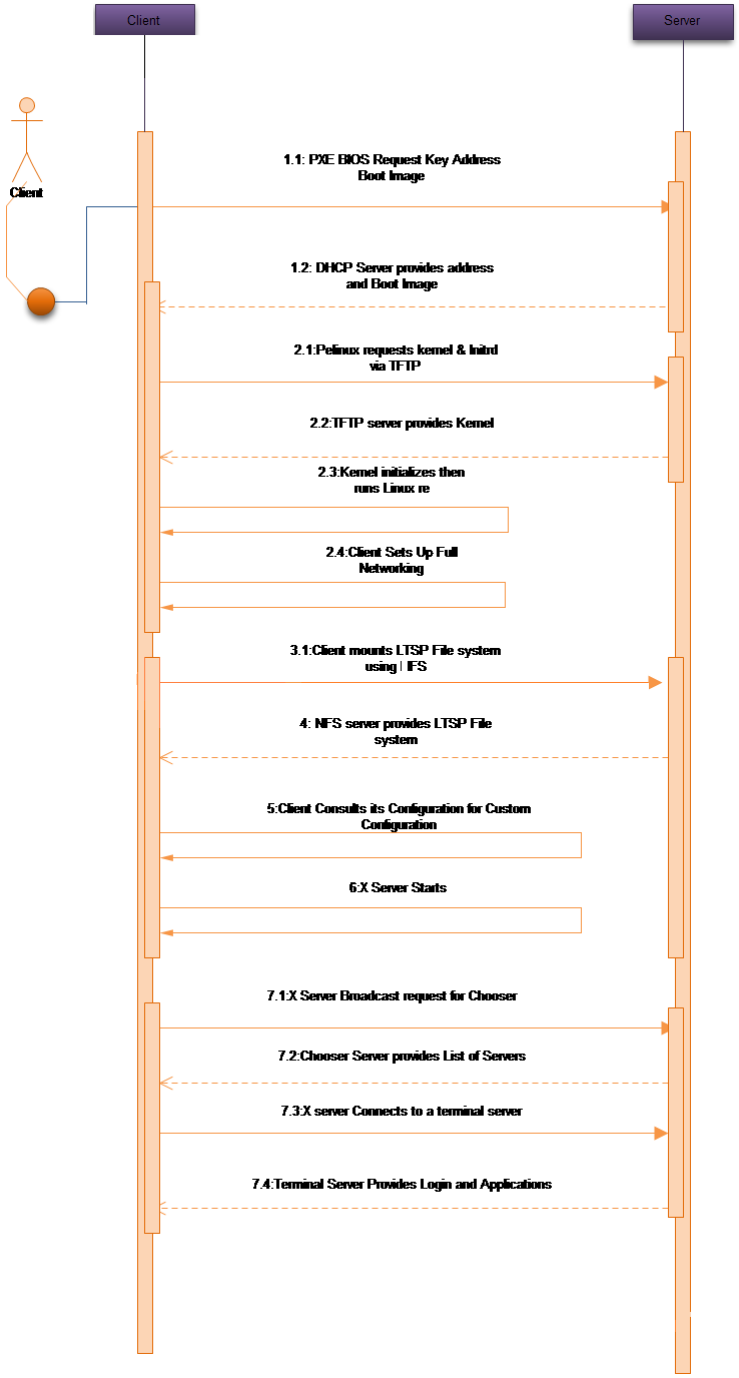


Figure 4.3: Sequence Diagram

4.2.3 Deployment Diagram

In UML, deployment diagrams model the physical architecture of a system. Deployment diagrams show the relationships between the software and hardware components in the system and the physical distribution of the processing.

Deployment diagrams, which you typically prepare during the implementation phase of development, show the physical arrangement of the nodes in a distributed system, the artifacts that are stored on each node, and the components and other elements that the artifacts implement. Nodes represent hardware devices such as computers, sensors, and printers, as well as other devices that support the runtime environment of a system. Communication paths and deploy relationships model the connections in the system.

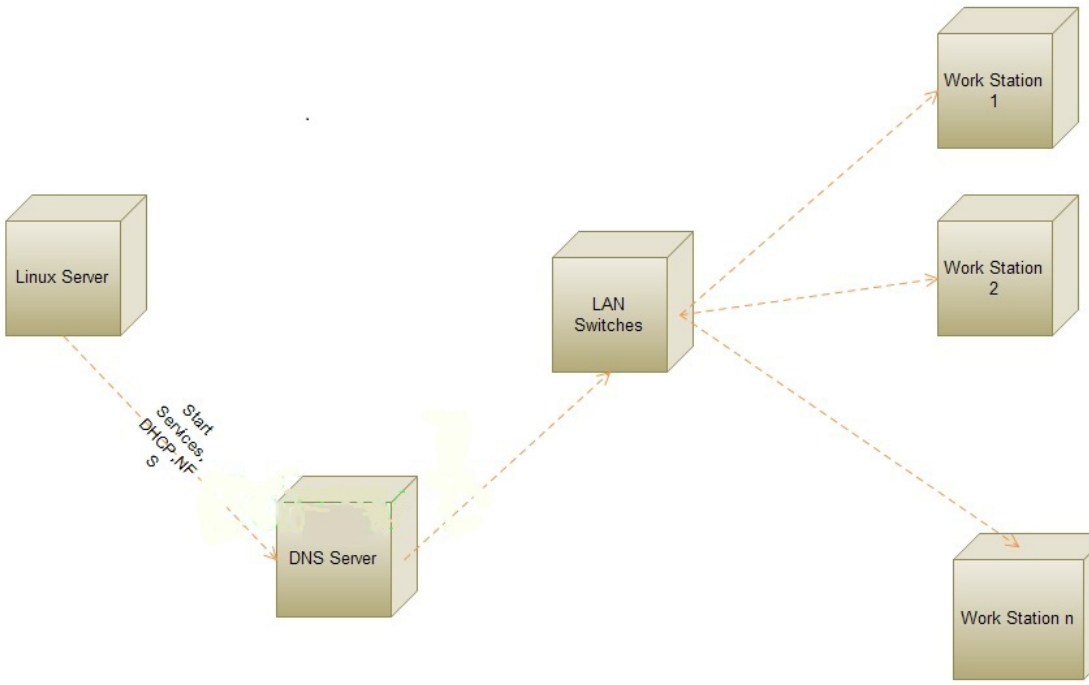


Figure 4.4: Deployment Diagram

Chapter 5

Technical Specifications

5.1 Advantages

1. This System is an OPEN SOURCE System so no licensing issues.
2. Client workstations need very low configuration thus old and out-dated machines can be reused.
3. No local hardware accesses so any threat of data piracy via USB Drives is avoided.
4. Monitoring all workstations via single server monitoring.
5. It saves TIME and MONEY.

5.2 Limitations

1. It is a LAN based system so LAN connection is always required.
2. Good quality Switches and high Bandwidth Connection.

5.3 Applications

1. It can be used in any large Institutes/Organizations/IT Industries where large numbers of computers are to be maintained.
2. It can be used where usage of money must be economical.

Chapter 6

Project Estimate, Schedule and Team Structure

1. Effort Estimate Table

Task	Effort Weeks	Deliverables
Analysis of Existing system and compare with proposed one	4 weeks	-
Literature Survey	1 Week	-
Designing and Planning	1+2 weeks	Modules
Implementation	9 weeks	Primary System
Testing	3 weeks	Test Report
Documentation	1 week	Complete Project Report

2. Phase Description Table

Phase	Task	Description
Phase 1	Analysis	Analyzed all the information on our topic
Phase 2	Literature Survey	Collected raw data on LTSP ,PXE-boot, existing Client Tools,etc.
Phase 3	Design	Modules were assigned and the process flow control was designed
Phase 4	Implementation	Implemented the code for all modules and also integrated them
Phase 5	Testing	Tested the code and overall process work properly
Phase 6	Documentation	Prepared Project Report

3. Project Plan

This is the project plan that was followed for implementation of Advanced Client Management Tool for Diskless Workstations on Linux Platform over LTSP.

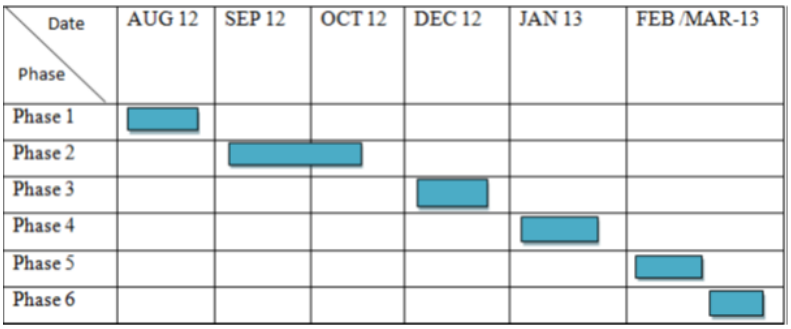


Figure 6.1: Project Plan

4. Team Structure

Figure 6.2: Team Structure

Chapter 7

Software Implementation

7.1 Introduction

Installing the Edubuntu 12.04 on a server and booting clients from network with use of LTSP and create User-Friendly GUI based Advanced Client Management Tool which can keep watch on clients and execute/kill any process on any/all client/s.

7.2 Databases

We have two databases viz: Query Database and Client Database. For both we have used Java Serialization. We store all the data in a .dat file.

7.3 Important Module and Algorithms

There are total 5 modules:

1. Configuring Disk-less Workstations
2. Client Side GUI
3. Server Side GUI
4. Communicator
5. Client Management Tool

7.4 Business logic and Architecture

For processing of our Client Management Tool we are using the Incremental Model.Incremental development slices the system functionality into increments (portions). In each increment, a slice of functionality is delivered through cross-discipline work, from the requirements to the deployment. The unified process groups increments/iterations into phases: inception, elaboration, construction, and transition.

1. Inception identifies project scope, requirements (functional and non-functional) and risks at a high level but in enough detail that work can be estimated.

2. Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.

3. Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.

4. Transition delivers the system into the production operating environment.

Chapter 8

Software Testing

8.1 Introduction

Software testing can be stated as the process of validating and verifying that a computer program application/product:

- meets the requirements that guided its design and development,
- works as expected,
- can be implemented with the same characteristics,
- and satisfies the needs of stake-holders.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. Traditionally most of the test effort occurs after the requirements have been defined and the coding process has been completed. A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software

testing may be used to correct the process by which software is developed. Every software product has a target audience. For example, the audience for video game software is completely different from banking software. Therefore, when an organization develops or otherwise invests in a software product, it can assess whether the software product will be acceptable to its end users, its target audience, its purchasers, and other stake-holders. Software testing is the process of attempting to make this assessment.

8.2 Test Cases

Test Case ID	Description	Precondition	Test Data	Test Procedure	Expected Result	Actual Result	Test Result
T001	Disk-less Workstation	Clients should boot from server	boot option must be "Boot from Network" and DHCP must be configured	Boot Machine	System should provide access to login GUI	User gets Login Screen	Pass
T002	Access to Network OS	Network OS must be accessed	Boot from Network must be selected	Enter Login Credential	System should provide access to Network OS	User get the access to Network OS	Pass
T003	User Verification	Network OS must be accessed	Boot from Network must be selected	Enter wrong Login Credential	System should not provide access to Network OS	User get message "Invalid Details"	Pass

Table 8.1: Test Table1

Test Case ID	Description	Precondition	Test Data	Test Procedure	Expected Result	Actual Result	Test Result
T004	Connect to Server	Servlets are activated	Server connected	Enter IP Address or Server Name	Server connected	Client is connected to Server	Pass
T005	Wrong URL	Servlets are activated	Server connected	Enter IP Address or Server Name	Server not connected	"Error Sending Info" message is displayed	Pass
T006	Disconnect from Server	Server is connected	Server is connected and Enable button is disabled	Press Disable Button	Server Disconnected	Client is Disconnected from Server	Pass

Table 8.2: Test Table2

Test Case ID	Description	Precondition	Test Data	Test Procedure	Expected Result	Actual Result	Test Result
T007	Wrong Directory	Manage Command form should be available	Add button	Enter directory of command	System should not add command	User gets message "Invalid Directory"	Pass
T008	Add Single Command	Manage Command form should be available	Add button	Enter Command type and directory properly	System should Add command to query database	User Adds a single command	Pass
T009	Add Shell Scripts	Manage Command form should be available	Add button	Enter Command type, shell script and directory properly	System should Add Shell Script to query database	User Adds a Shell Script command	Pass

Table 8.3: Test Table3

Test Case ID	Description	Precondition	Test Data	Test Procedure	Expected Result	Actual Result	Test Result
T010	View Log	Pre-Execution of ACM tool is to be performed	View Log button on Server Monitor Screen	Button press Event	System should provide Activity Log Form	User get the Activity Log Form	Pass
T011	Clear Log	Server Monitor form to be displayed	Clear Log button on Server Monitor Screen	Button press Event	System should clear Log	User get the Cleared Activity Log Form	Pass
T012	Remove Single Command	Manage Command form should be available	Remove button	Select the command you want to remove	System should Remove command from query database	User removes a single command	Pass

Table 8.4: Test Table4

Test Case ID	Description	Precondition	Test Data	Test Procedure	Expected Result	Actual Result	Test Result
T013	Stop Server	Activated Server	Stop button on Activate Server Monitor Screen	Button press Event	Server should stop	Server Stops	Pass
T014	Start Server	Activate Server	Start button on Activate Server Monitor Screen	Button press Event	Server should start	Server Starts	Pass
T015	Fire Query	Activate Server Monitor form to be displayed with fire button on it	Fire query on single client	Button press Event	System should fire a query on selected client	Query fired on selected client	Pass

Table 8.5: Test Table5

Test Case ID	Description	Precondition	Test Data	Test Procedure	Expected Result	Actual Result	Test Result
T016	Fire Query on All	Activate Server Monitor form to be displayed with fire all button on it	Fire query on all clients	Button press Event	System should fire a query on all connected client	Query fired on all connected clients	Pass
T017	Wrong Command Type	Activate Server Monitor form should be available	Fire Button	select execute query and read command type and Press Fire Button	System should give error for wrong type	User gets NULL Pointer Exception	Pass
T018	Unexpected Server Stop	Is server stops unexpectedly	Client Machine	Server Stops	Client should display error for server	Client gets "error sending info" message is displayed	Pass

Table 8.6: Test Table6

Chapter 9

Results

9.1 Snap Shots of the test cases and Test Plans

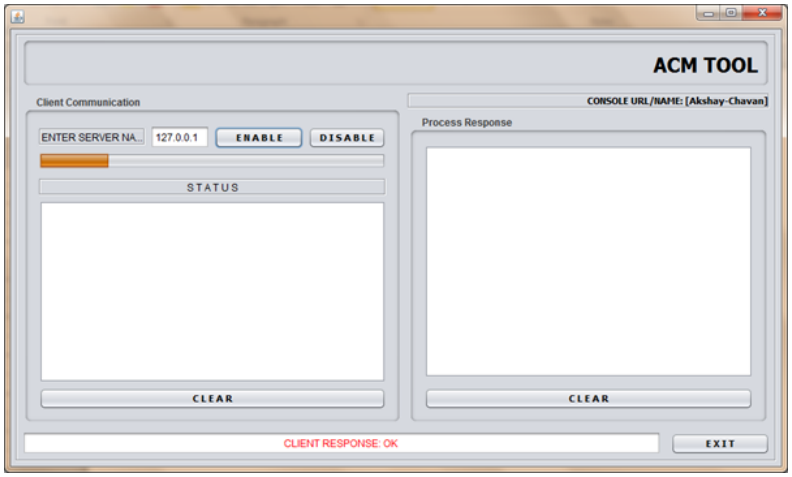


Figure 9.1: Client Console Before Connecting Server

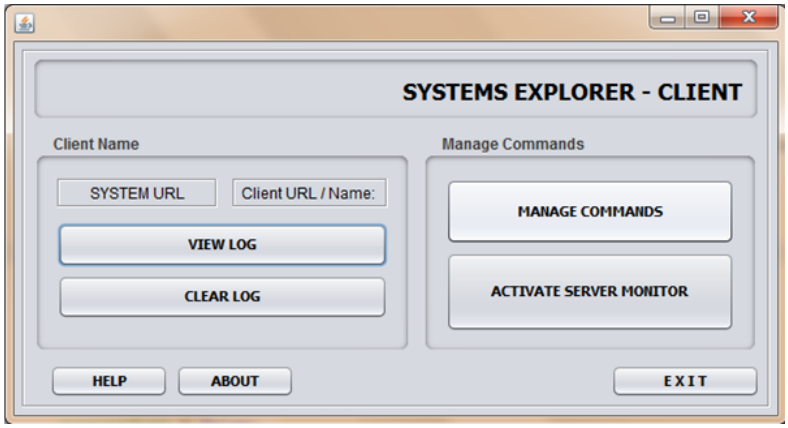


Figure 9.2: Server-Side System Explorer

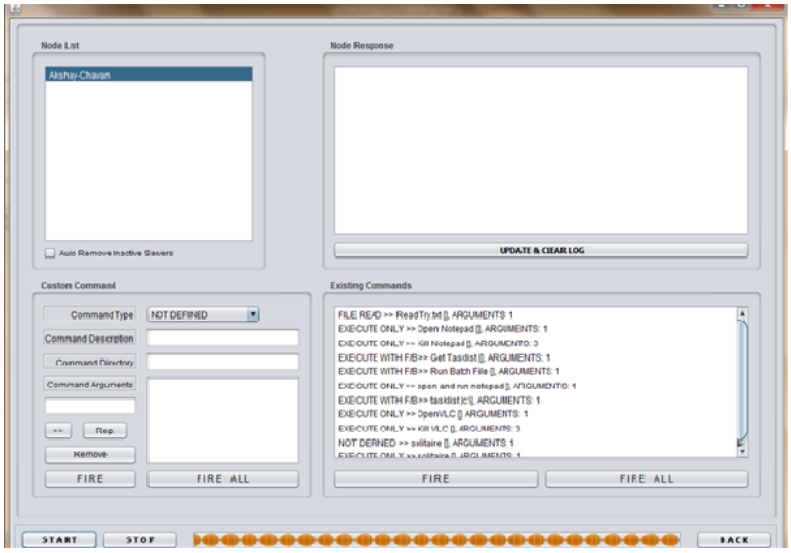


Figure 9.3: Server Monitor

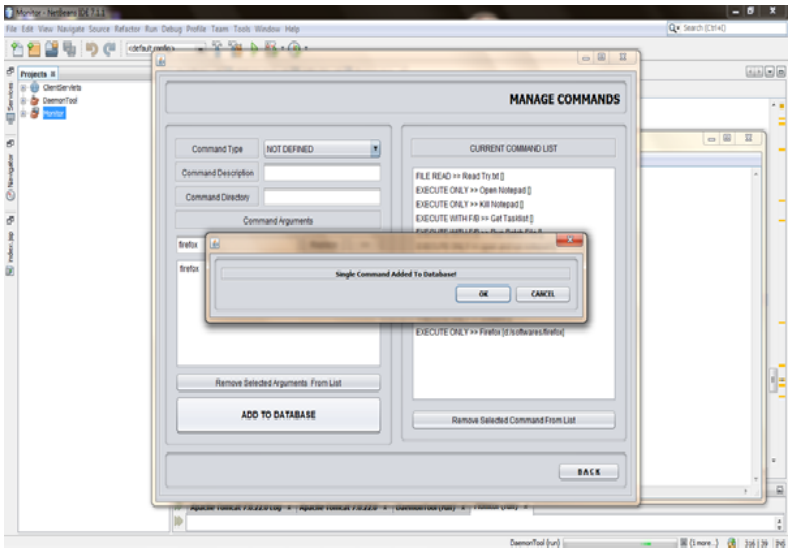


Figure 9.4: Manage Commands Windows "Single Command Added"

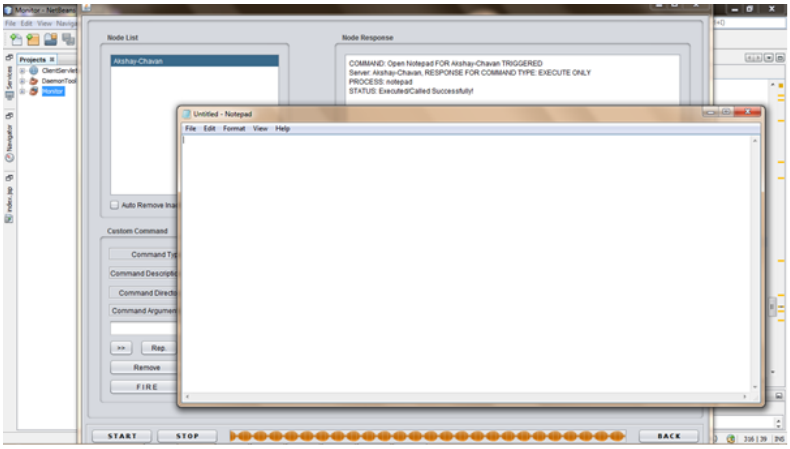


Figure 9.5: Query Executed via Server Monitor

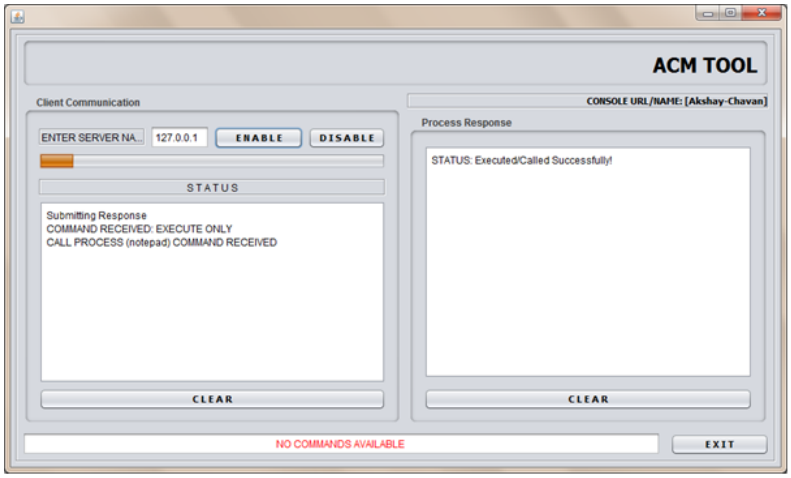


Figure 9.6: Clients notification for Query Executed by Server

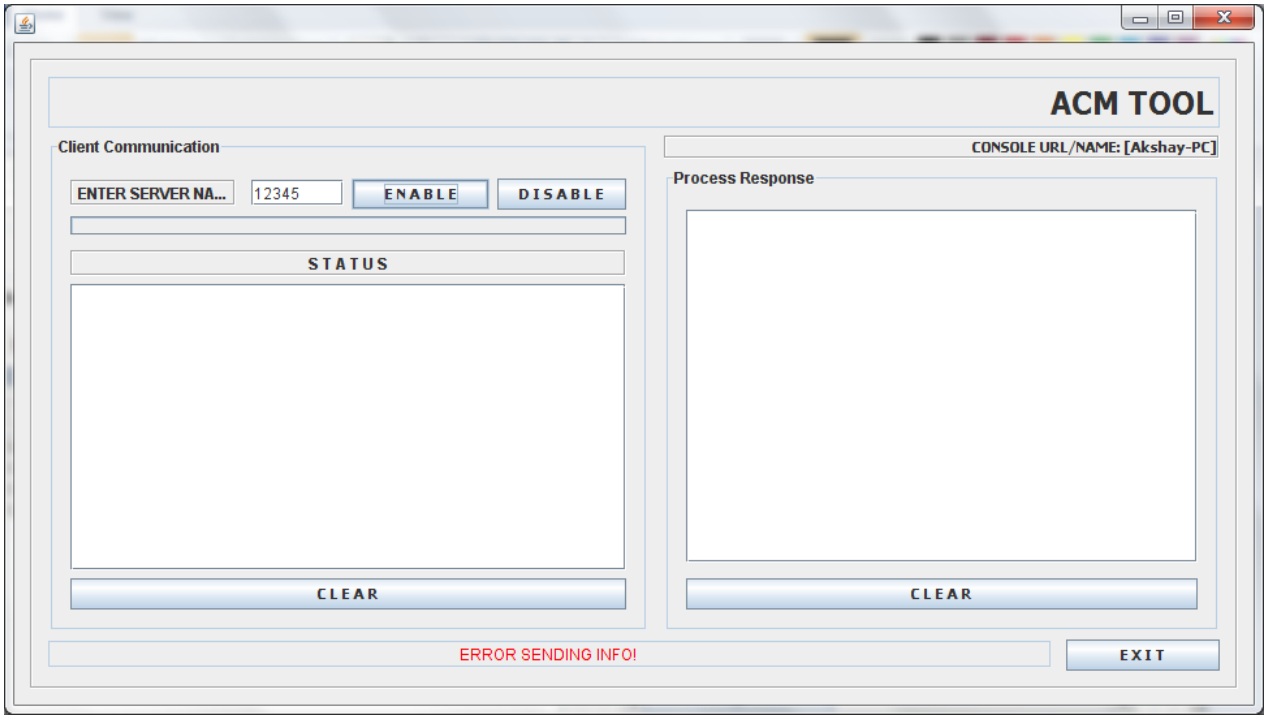


Figure 9.7: Invalid URL Number

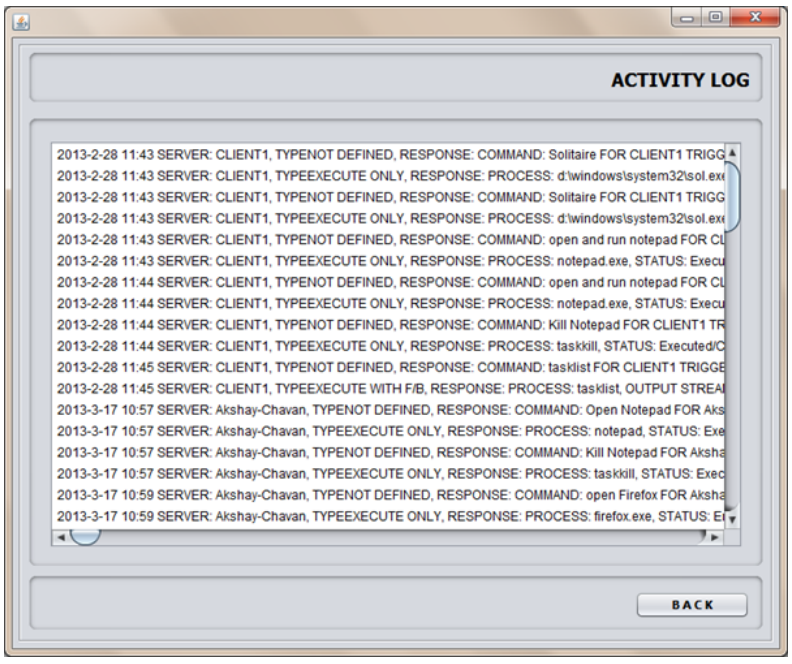


Figure 9.8: Activity Log

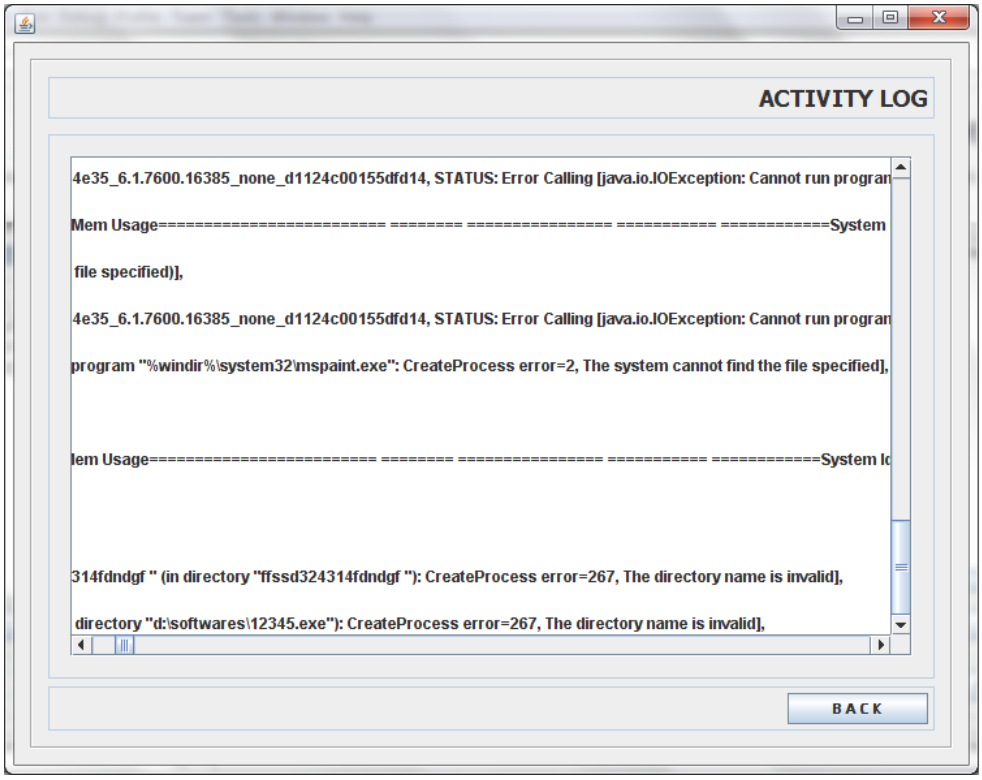


Figure 9.9: Log about Failed queries

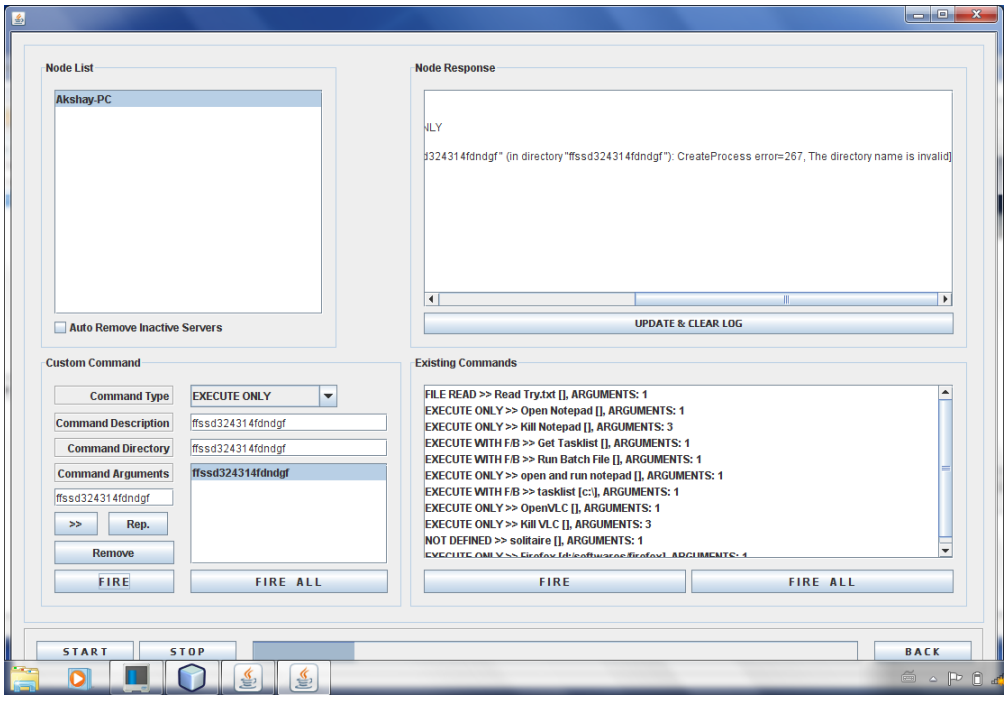


Figure 9.10: Invalid directory during adding a command

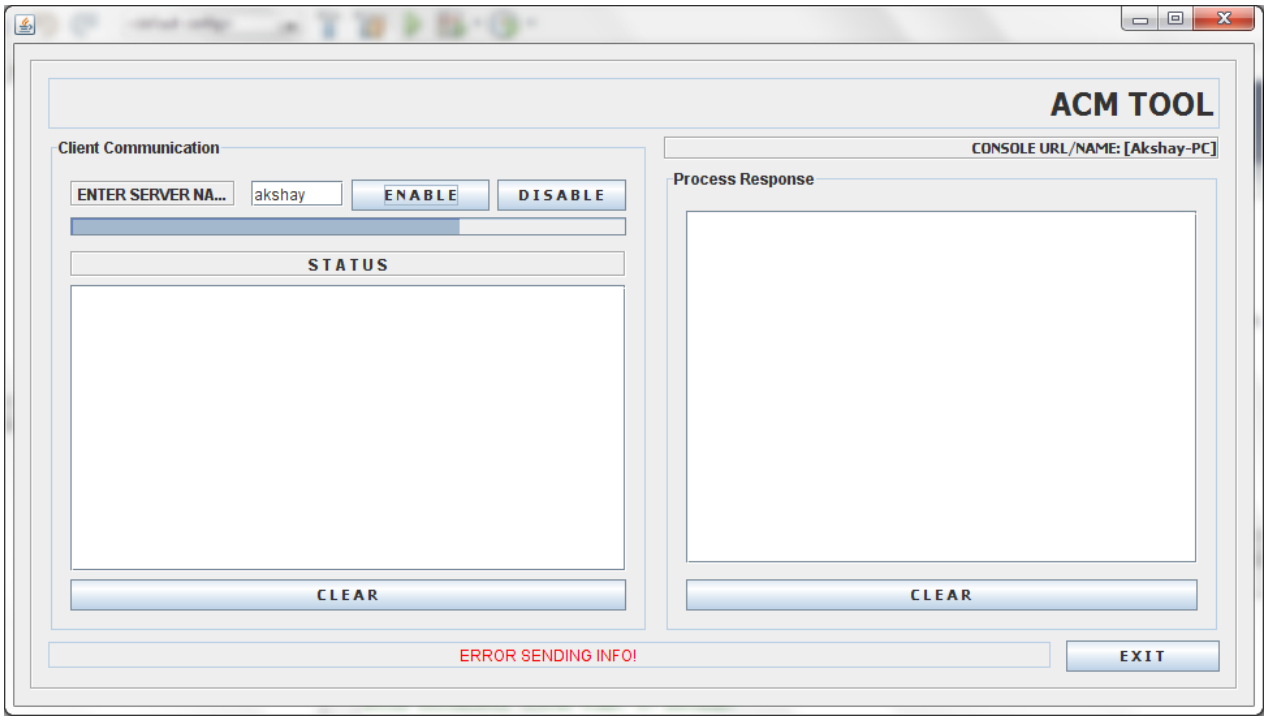


Figure 9.11: Wrong Url Name while connecting to a Server

Chapter 10

Deployment and Maintenance

10.1 Installation and Uninstallation

1. To execute Java programs you only need a Java Runtime Environment (JRE) but to build a Java project, you are going to need a Java Development Kit (JDK). As an example, We downloaded the file: jdk-7-windows-i586.exe. After downloading we double clicked it to get the following screen.

Figure 10.1: JDK1

2. Simply press the "Next ¿" button.
   You should now see the next screen:

3. On the screen above, you can customize setup. We just accepted all the defaults and clicked on the "Next ¿" button.

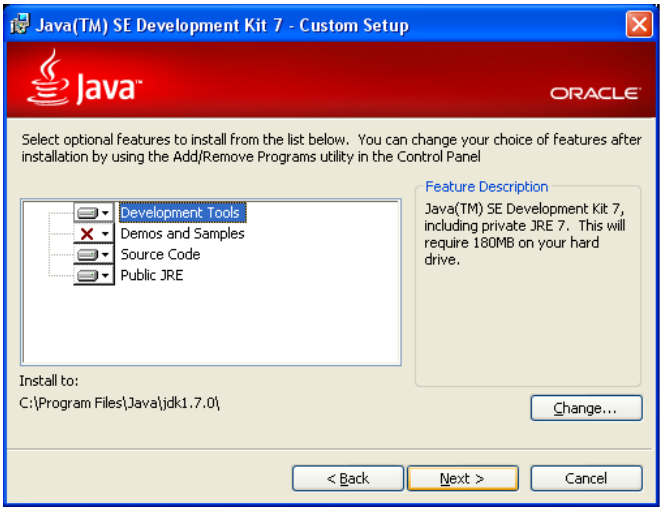


Figure 10.2: JDK2

You should now see the screen below.

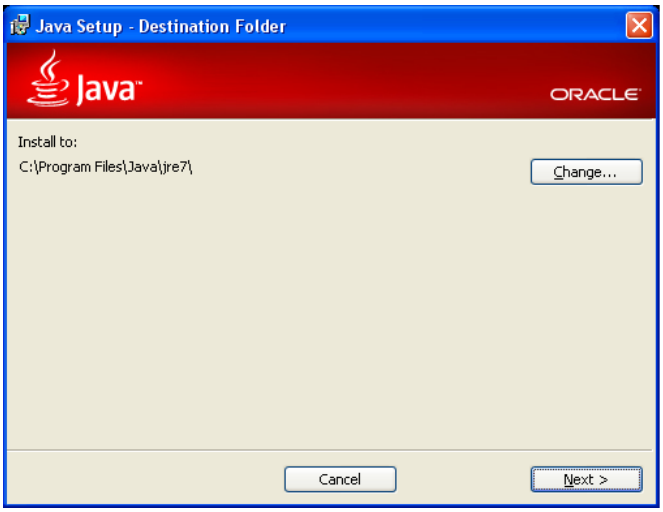


Figure 10.3: JDK3

4. On the screen above, you can change the destination folder of the jre. Again, We accepted the defaults and clicked on the "Next ¿" button.
   This should result in the screen below.

5. On Microsoft Windows platforms, you can pick from one of the following downloads:
   Self-extracting installer
   Archive distribution

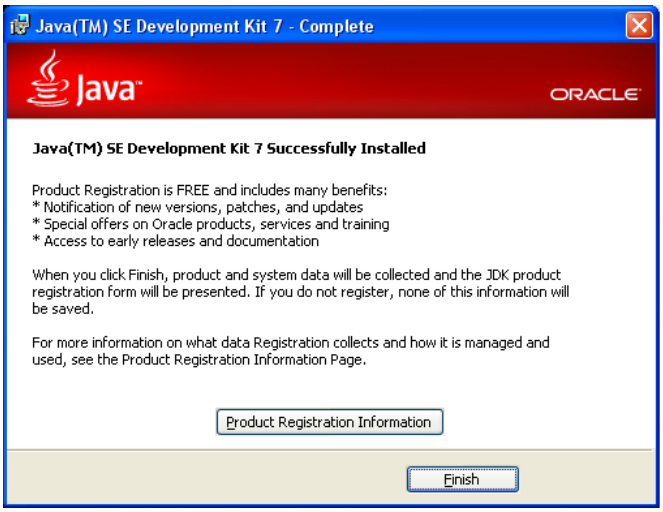


Figure 10.4: JDK4

Figure 10.5: NETBEANS

Chapter 11

Conclusion and Future Scope

We have successfully created an Advanced Client management tool". The tool is developed over a linux platfrom which works on diskless workstations over LTSP. It is useful in executing/killing a process on client machine and also keeps a log of the access made from the client and to the client. The work is innovative in several aspects. The server/admin can keep a track of all the client from one place and execute/kill task on any number of clients available. The client can request a server for specific task also. Administrator can add or remove new queries to and from the query database respectively on runtime which makes updation of the software very easy."

As the tool is basically for LAN-based systems it cannot work for remote machines. The Disk-less System can be operated from remote machine via wifi or Cloud is the future scope of our project.

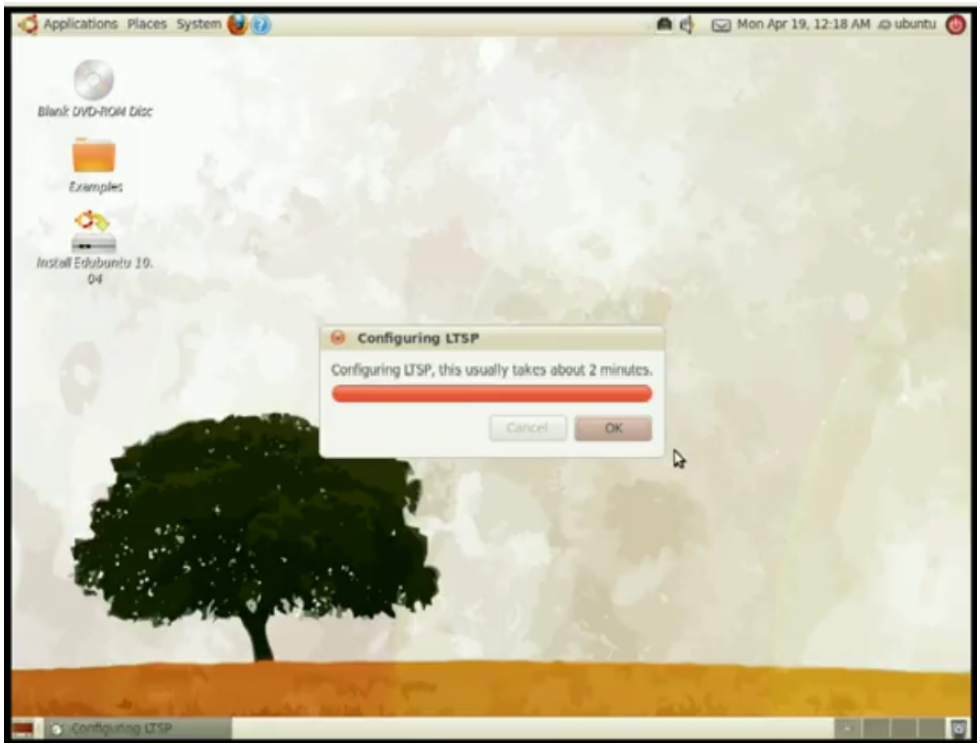
Appendix A

MODULES OF THE PROJECT

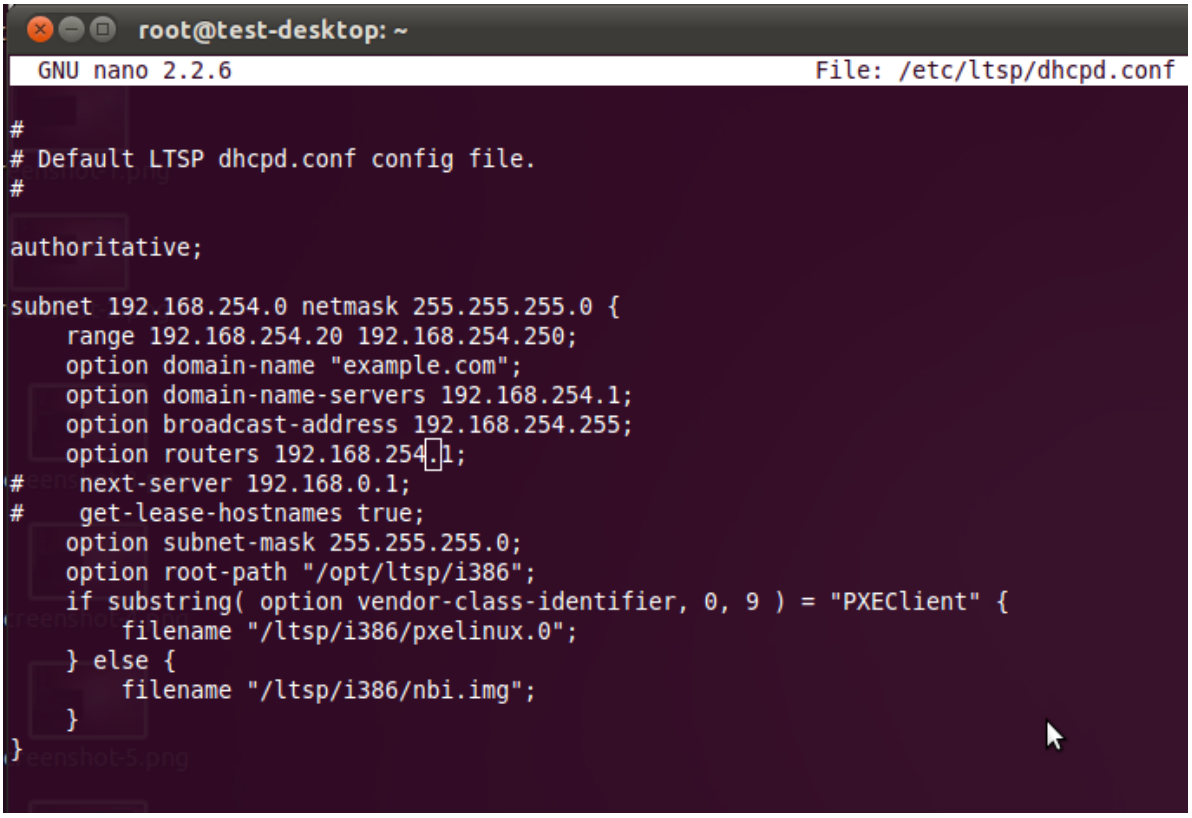
MODULES IN Disk-less workstations Booting

1. Server Configuring LTSP

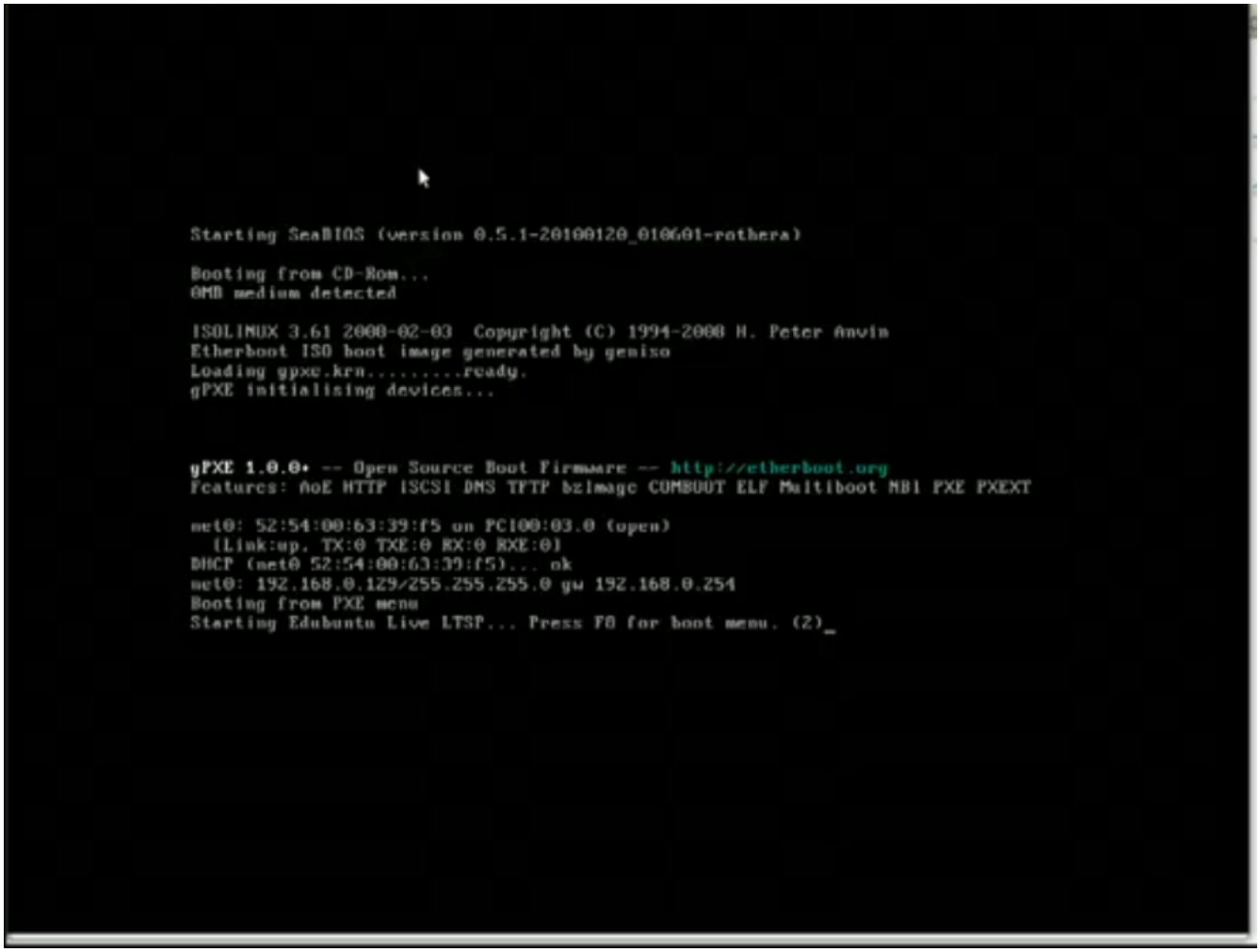
    - The Linux platform is configured with LTSP



2. Configure DHCP and Network Interfaces



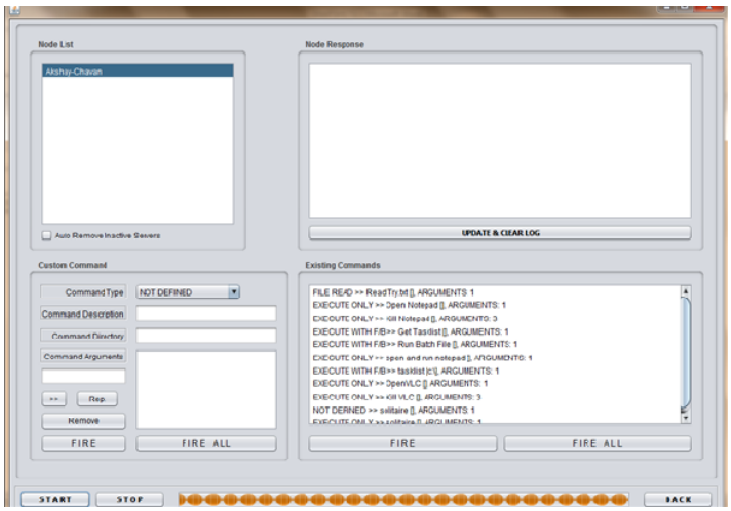
3. BOOT Client by using booting option as "Boot from Network"



MODULES IN ACM Tool

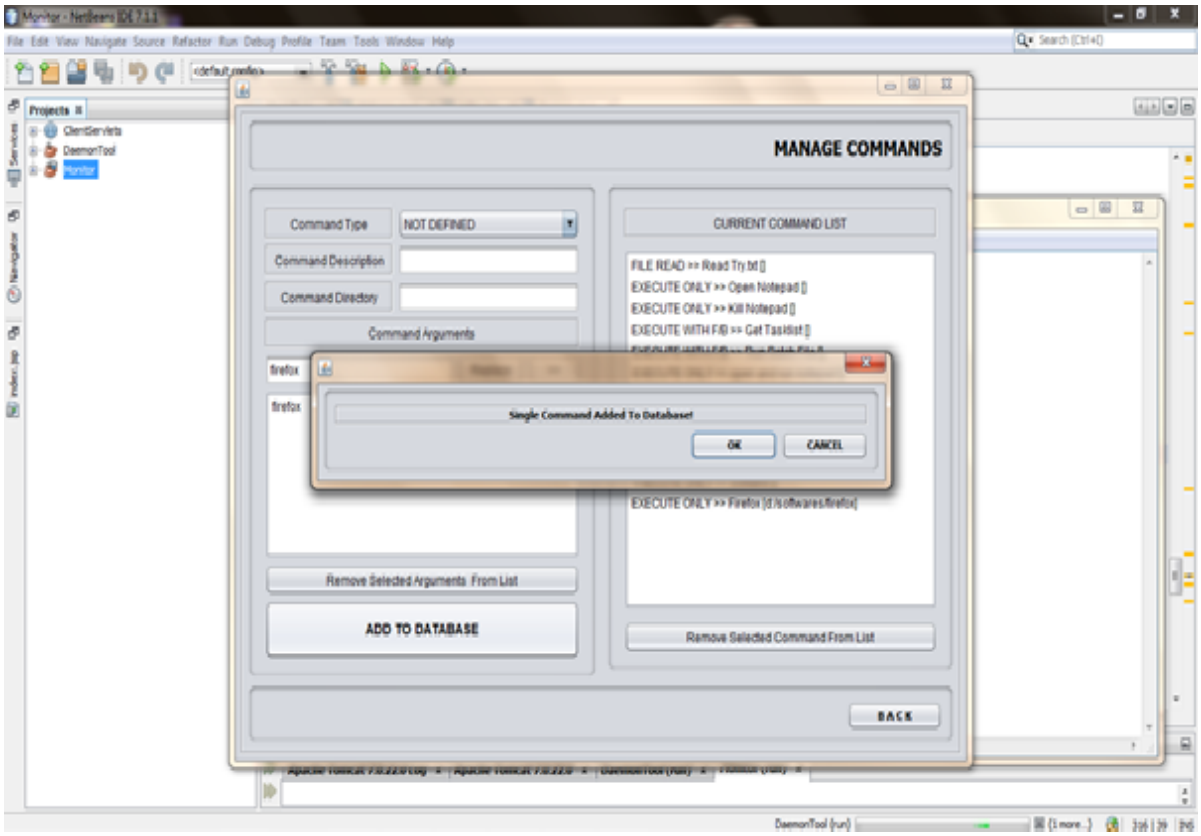
1. Client Side GUI

2. Server Side GUI



3. Communicator i.e servlets

4. Query Database



Appendix B

DOMAIN SPECIFIC ANALYSIS

Client Management Tool Epoptes is an open source computer lab monitoring tool. It allows for screen broadcasting and monitoring, message sending.
It can be installed in Ubuntu, Debian and openSUSE based labs ONLY that may contain any combination of the following: LTSP servers, thin and fat clients, non LTSP servers, standalone workstations, NX or XDMCP clients etc

Thin Client Generally, terminals are low-powered, lack a hard disk and are quieter and more reliable than desktop computers because they do not have any moving parts.Applications run on the server with a terminal known as a thin client (also known as an X terminal) handling input and output.
Pupils can access to computers without purchasing or upgrading expensive desktop machines.Improving access to computers becomes less costly as "new" thin client machines can be older computers that are no longer suitable for running a full desktop OS. By extending the useful life of obsolescent computers, costs can be cut. Even a relatively slow CPU with as little as 128MB of RAM can deliver excellent performance as a thin client. In addition, the use of centralized computing

References

[1] "Jim mcquillan, october 2000. the linux terminal server project: Thin clients and linux."

[2] "Brian zammit, 2008. linux terminal server project: Server configuration guidelines."

[3] "W. richard stevens, addison-wesley, a. chander, d. dhurjati, h. inamura, and z. su, 1994. tcp/ip illustrated volume 1."

[4] "James s. plank, youngbae kim, jack j. dongarra, 2006. fault tolerant matrix for networks of workstations using diskless check pointing. volume 6."