GENERIC PLATFORM INDEPENDENT OPERATING SYSTEM UTILITY

V.SRIDHAR [1]

[1] Assistant Professor, ECE, Vidya Jyothi Institute of Technology, Hyderabad
[1]varadalasri@gmail.com

ABSTRACT: This Generic Platform Independent OS Utility is an interface that is related with the operating system and the end user. The role the interface plays is that of a Graphical interface that comes up with command line where a pre defined set of commands are allowed to be executed. The command line is similar to that of a DOS command line where the users are allowed to type the commands. The application also has a GUI representation where without remembering the commands the user can still run the commands. The created Virtual OS should be able to function on any OS providing the support to the end user to adapt to one environment but perform tasks that are supported by different environments. It should provide us with User friendly environment to understand and implements commands. It should carry common properties such as having a built in editor to create and edit files, history of Commands are executed, to run multiple application from a single command line and other common tasks

Keywords: OS, GUI, DOS

I . INTRODUCTION

This Virtual Operating System application is an interface that is related with the operating system and the end user. The role of the interface plays is that of a virtual OS that comes up with command line where a tree defined set of commands are allowed to be executed .The command line is similar to that of a DOS command line where the users are allowed to type the commands When compared to training a end user on a package that is available across various operating systems. The end user is initially trained in the area of the OS, and then on the package. Commands that are pertaining to the trainee are available in each OS. But has either the parameters differing are the command itself different. For example a command like Doskey that is available & functionally in the windows is unavailable on UNIX. This problem has made the end user to learn different commands for each OS. The Virtual OS addresses this problem and creates a set of common commands that are requires for the company to trained their end users using a single or common commands facility for providing online help and user guide is also provided.The created Virtual OS should be able to function on any OS providing the support to the end user to adapt to one environment but perform tasks that are supported by different environments.

It should provide us with User friendly environment to understand and implements commands. It should carry common properties such as having a built in editor to create and edit files, history of Commands are executed, to run multiple application from a single command line and other commontasks.

2. SYSTEM ANALYSIS:

2.1 EXISTING SYSTEM

This system identifies company trained trainers to trainee the end users on the OS and then on the package the company uses. The trained are delivered in groups by different trainers based on trainer's specialization of OS. As the number of trainers required is more, the present system involves more payments. The period of training also increases and normally stretches to a month before the end users work upon the package. Every end user has to be ready to adapt the changes in the OS to work on the package.

2.2 PROPOSED SYSTEM

This Virtual Operating System application is an interface that is related with the operating system and the end user. The role of the interface plays is that of a virtual OS that comes up with command line where a tree defined set of commands are allowed to be executed. The command line is similar to that of a DOS command line where the users are allowed to type the commands.

When compared to training an end user on a package that is available across various operating systems. The end user is initially trained in the area of the OS, and then on the package. Commands that are pertaining to the trainee are available in each OS. But has either the parameters differing are the command itself different. For example a command like Doskey that is available & functionally in the windows, is unavailable on UNIX. This problem has made the end user to learn different commands for each OS. The Virtual OS addresses this problem and creates a set of common commands that are requires for the company to trained their end users using a

single or common commands facility for providing online help and user guide is also provided.The created Virtual OS should be able to function on any OS providing the support to the end user to adapt to one environment but perform tasks that are supported by different environments.

It should provide us with User friendly environment to understand and implements commands. It should carry common properties such as having a built in editor to create and edit files, history of Commands are executed, to run multiple application from a single command line and other common tasks.

3. MODULES:

The system is proposed to have the following modules: 1.Environment & User Interface

2.File system maintenance Module

3.Network management Module

4.Command line surfing module

5.Command-line editor and file handler module

3.1Environment & User Interface

The module deals with providing security to the application and creating the user interface. The security section identifies a valid user who is allowed to work on the application. The user has a facility to change his password. The user interface section is responsible to provide the required screens to allow the end user to work on the application. It provides a shell from where the end user executes the commands.

3.2File System maintenance Module

This module provides and organizes commands that are pertaining to the maintenance of the files and directories. The user is allowed to execute commands that can create, change and remove directories. A user can identify and kill the files. A facility to rename and copy the file from the shell is also provided. Information about the OS is also retrieved.

3.3Network management Module

This module provides a rich set of commands that can be used to execute on the Network configuration of the system. It can locate the availability of a remote URL. It can check the internet protocol of any site/domain. A facility to provide information about a site is also included

3.4Command line Surfing Module

This module provides a facility to provide an editor from the command line that can be able to create, edit and delete files. A sort filter is created to sort the parameters specified. You can locate files using the file command, change the prompt when required and retrieve date & time. A facility to display the memory settings is also specified.

3.5Command-line editor and file handler module

This module controls the display settings. A user is allowed to switch to various combinations of supported mono colors. It also helps with a command line help so as to enable the user to identify how a command can be used. A user guide in the form of HTML is also provided.

4. REQUIREMENT SPECIFICATIONS:

Software Requirements

Operating System	:	Any 32 Bit OS
Language	:	Java

Hardware Requirements (Minimum)

|		
Processor	:	PIV
Ram	:	256 Mb
Hard Disk	:	4 Gb space
Monitor	:	VGA Color (256)

5. SOFTWARE DESIGN

5.1 DATAFLOW DIAGRAMS

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level.

The top-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.Larry Constantine first developed the DFD as a way fo expressing system requirements in a graphical form, this lead to the modular design.A DFD is 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 to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

5.2 CONTEXT LEVEL DIAGRAM:
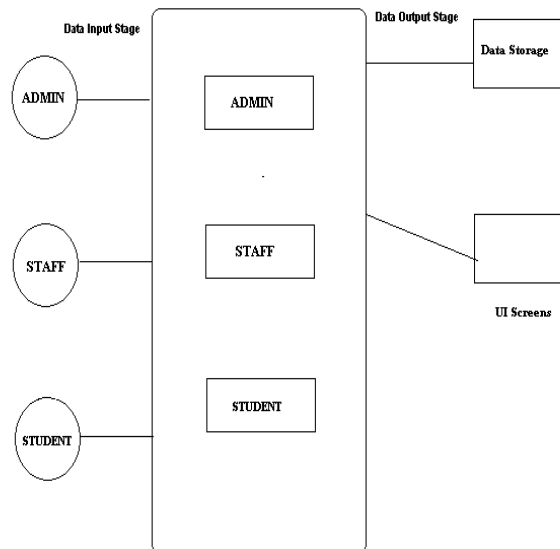
5.3 LOGIN DFD DIAGRAM
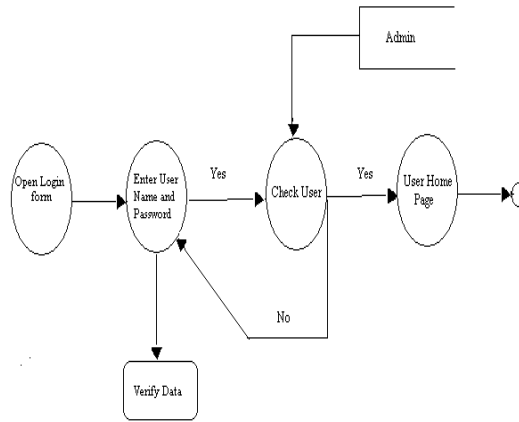
6. UML DIAGRAMS

6.1 USE CASE DIAGRAM
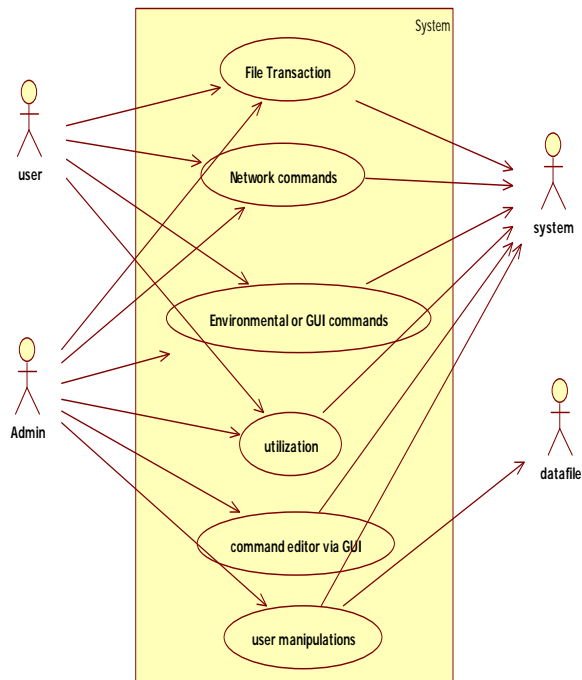
6.2. CLASS DIAGRAM
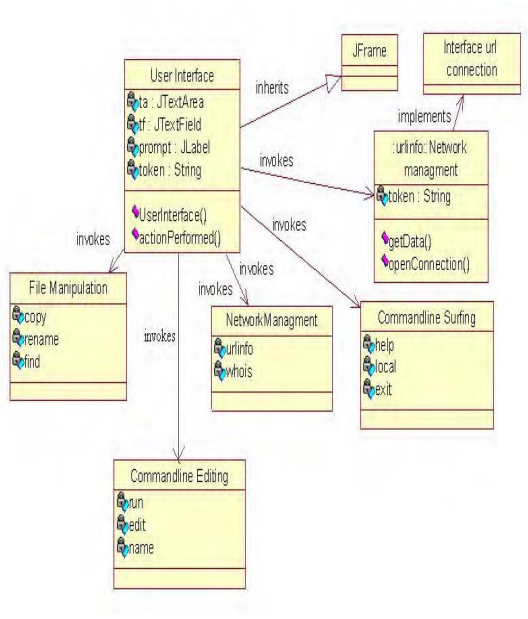
6.3 SEQUENCE DIAGRAMS:

6.3.1 Sequence diagram for Login:
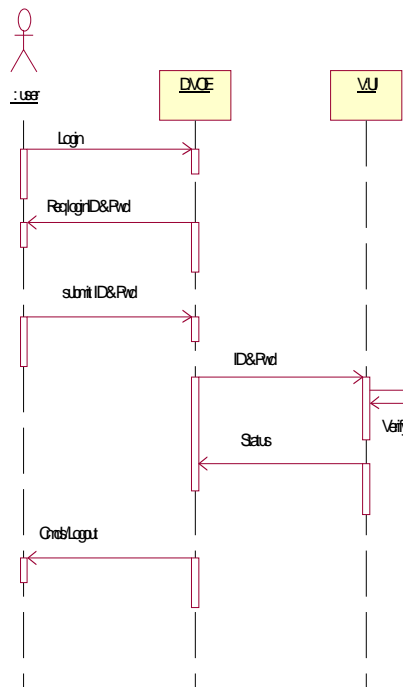
6.3.2Sequence Diagram for Basic Commands:
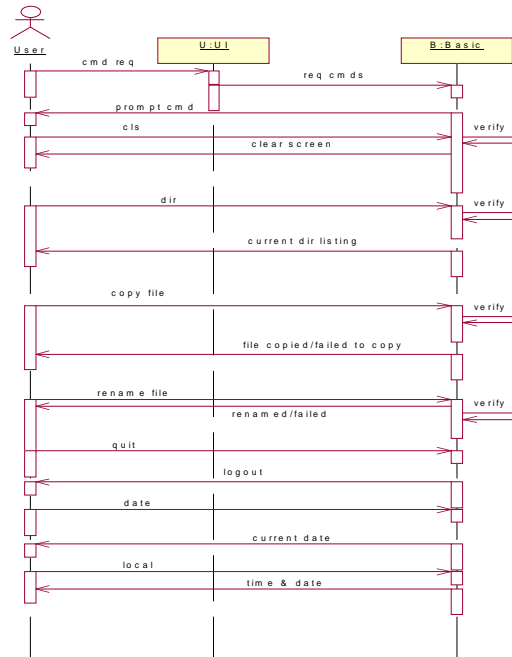
6.3.3Sequence Diagram for File Management:
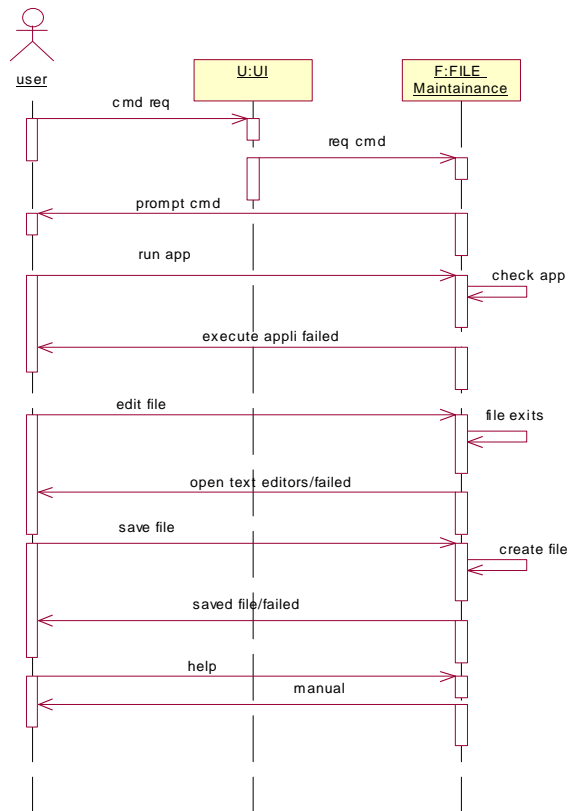
6.3.4 Sequence Diagram for Network Commands:
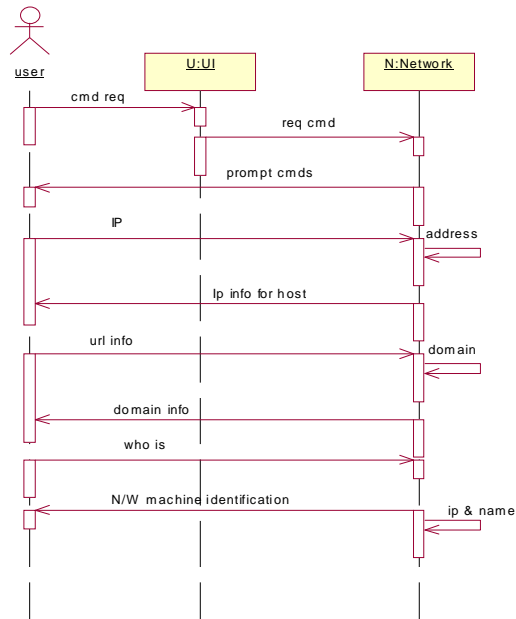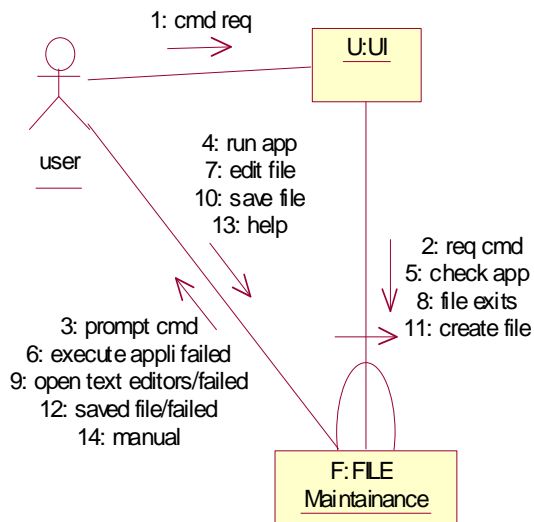

6.3.5 Sequence Diagram for Display change Commands:
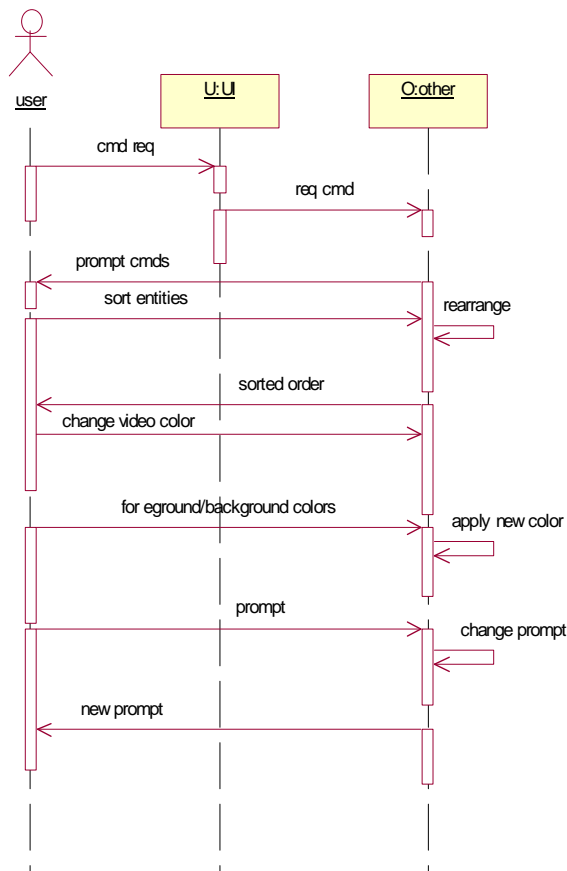
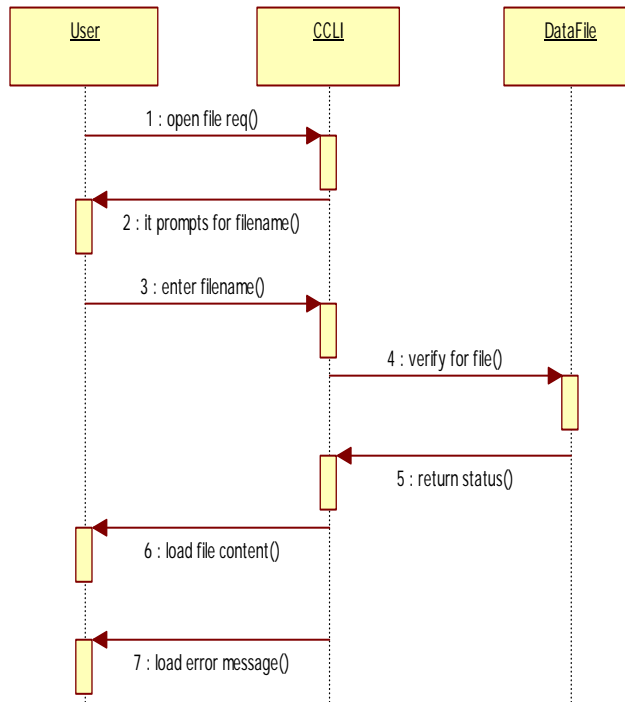## 6.4 COLLABORATION DIAGRAMS

6.4.1 Collaboration Diagram for Login:
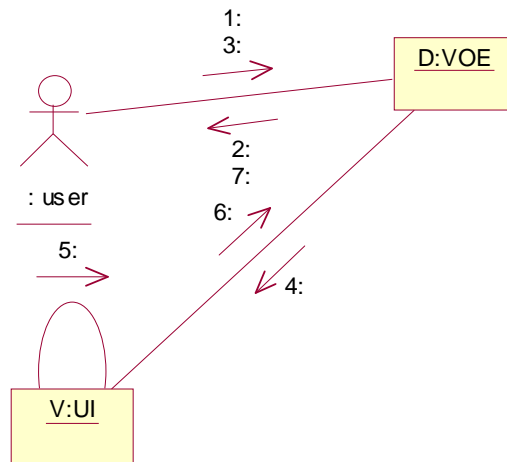
1. Login
2. Req LoginID & Pwd
3. Submit ID & Pwd
4. ID & pwd
5. Verify
6. Status
7. Cmds/Logout

6.4.2 Collaboration Diagram for Basic Commands

6.4.3 Collaboration Diagram for File Management:

6.4.5 Collaboration Diagram for Network Commands:
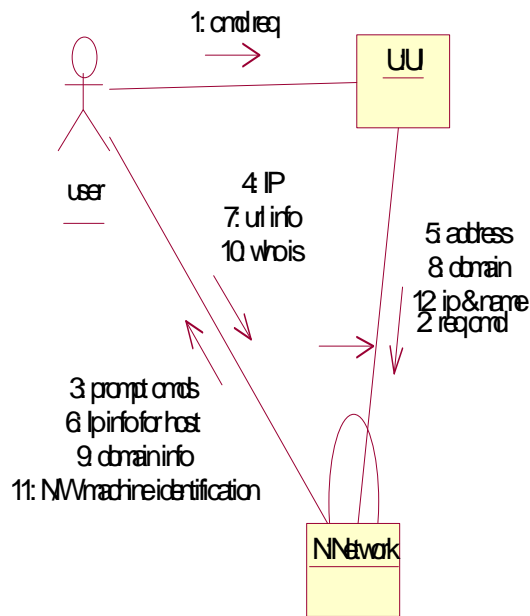
6.4.6 Collaboration Diagram for Display change Commands
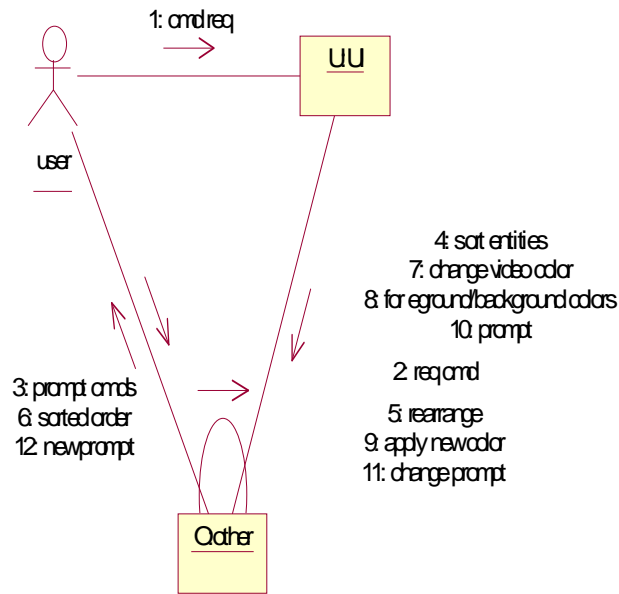
6.4.7 ACTIVITY DIAGRAM
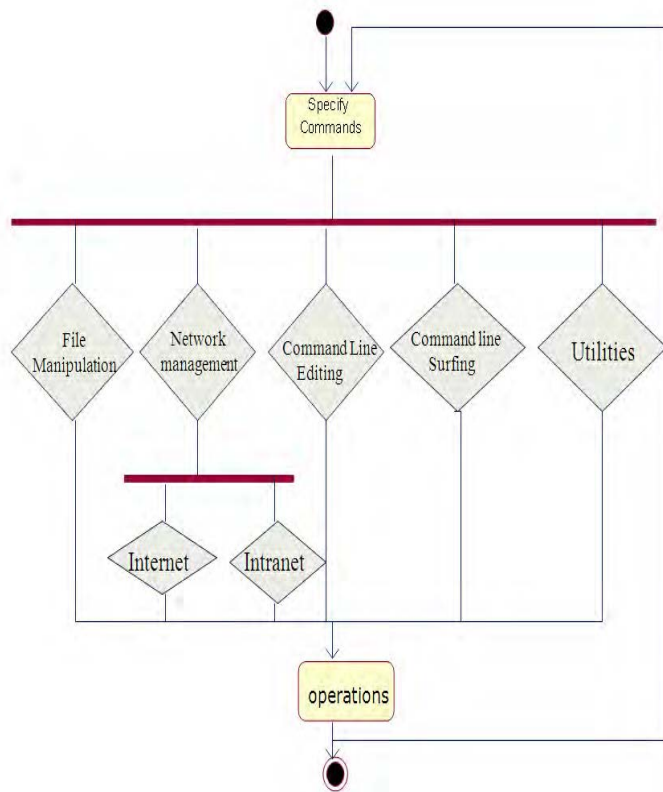
8.3 TEST CASES

Test Case 1: Login Module

File Name : Login.java
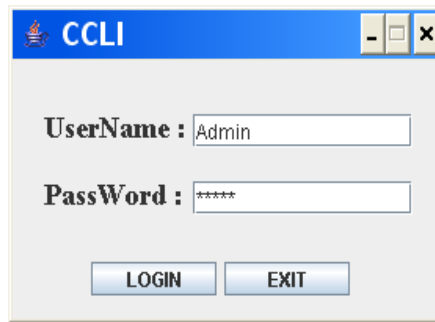
Test no	Inputs	Expected Behavior	Observed Behavior	Status P-passed F-Failed
1	Empty Requirement Field	An appropriate error message should be displayed and user should not allow to login.	do-	p
2	Wrong Username and/or wrong Password	An appropriate error message should be displayed and user should not allowed to login.	-do-	p
3	Correct Username and Password	User should be allowed to login	-do-	P

Test Case 2: New User

Test no	Inputs	Expected Behavior	Observed Behavior	Status
1	Enter wrong password at confirm password	An appropriate error message should be displayed and new user will not be created	-do-	p
2	Enter correct details	A new user gets created successfully	-do-	p

9. OUTPUT SCREENS:

9.1 Admin login:
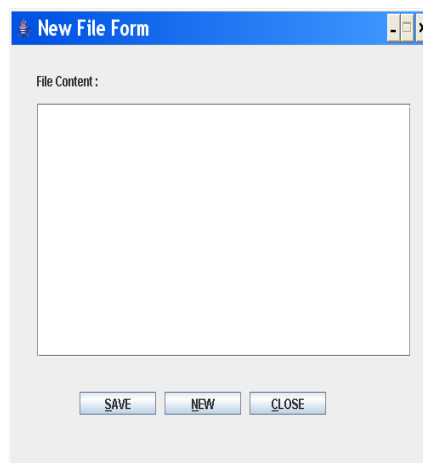
9.2 Admin Main Form:
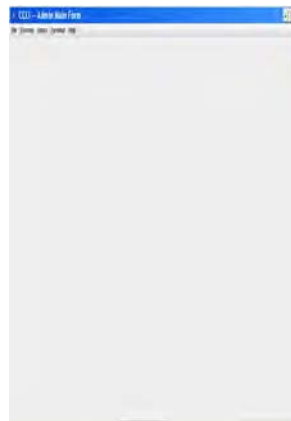
9.3 User Login Form:

9.4 User Login Form:

9.5 To Create A New File:
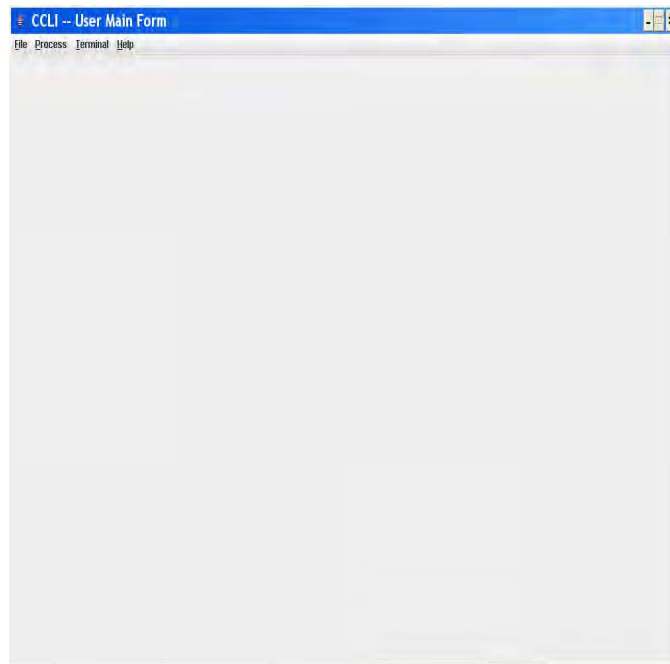
9.6 To Open A File:
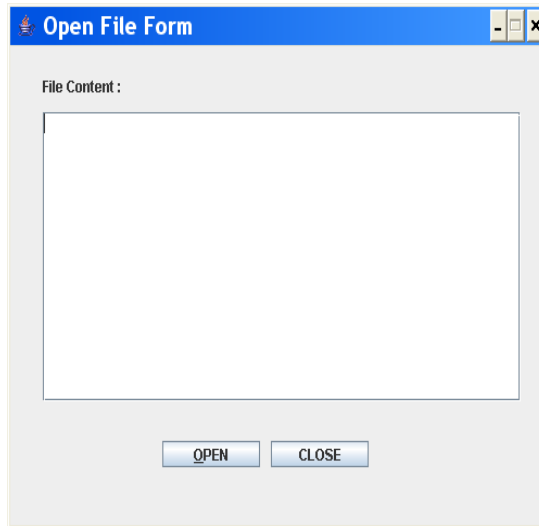
9.7 Copy /Move a File:
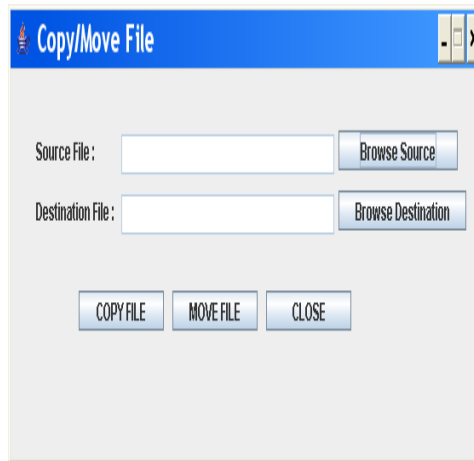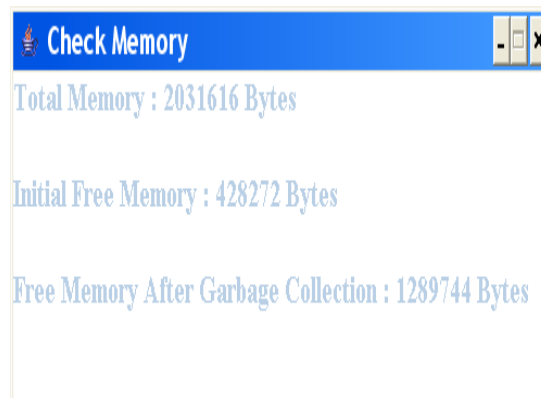

9.8 To Kill a File:
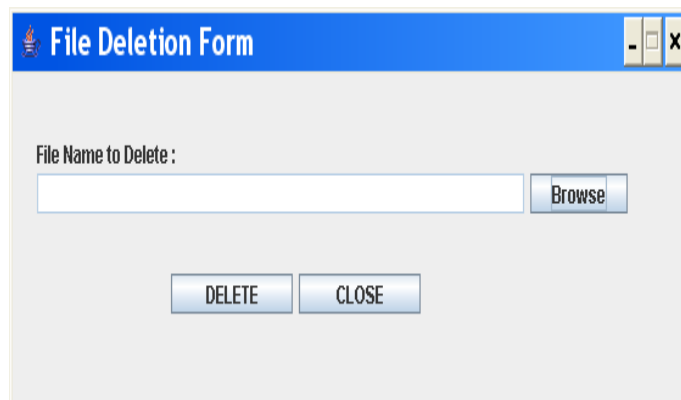
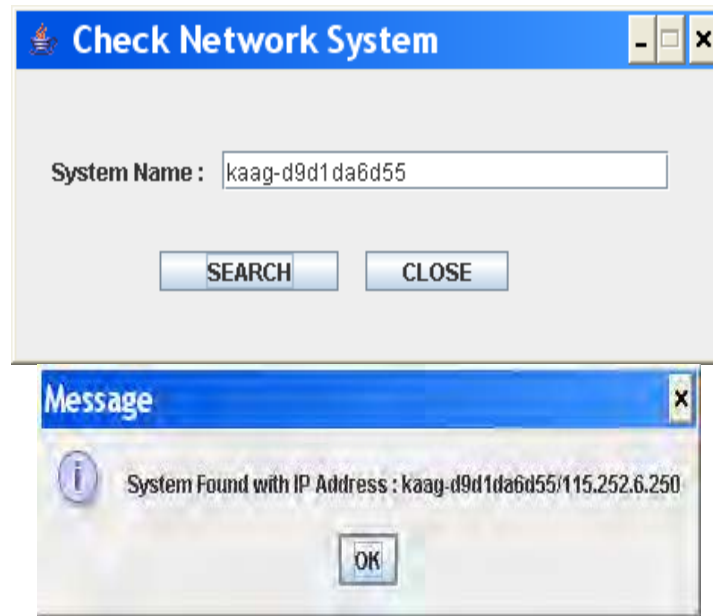**9.9 Displays Files:**
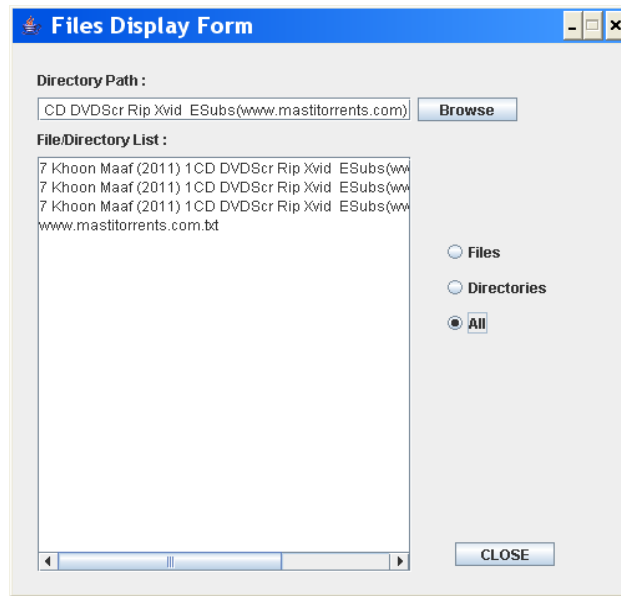
9.10 Start a New process:
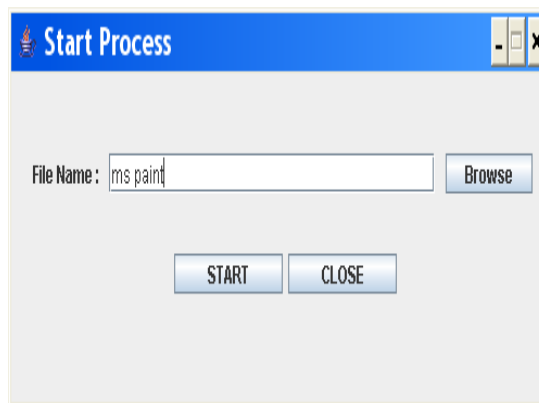
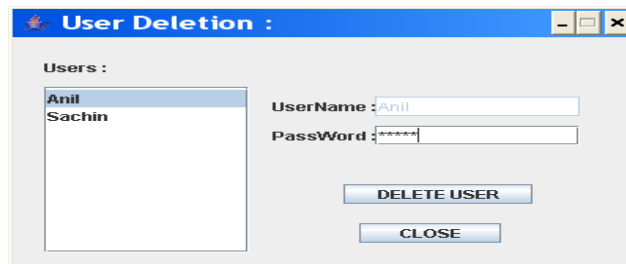**9.11 Checks System Memory:**

9.12 Checks for a system in the network:
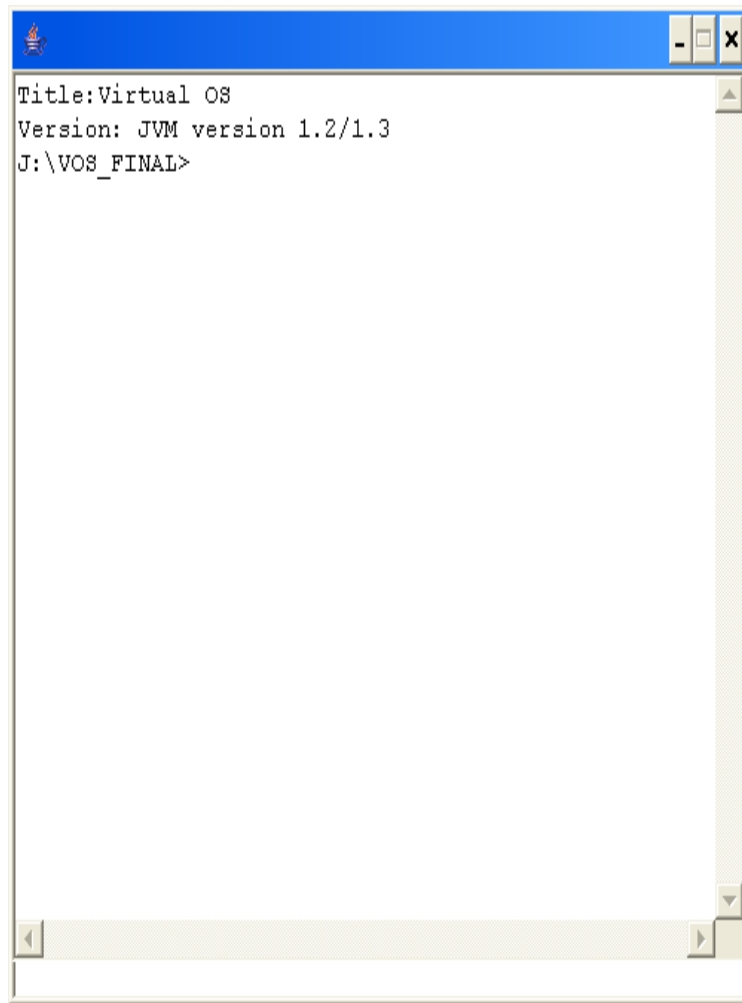

9.13 Create a new user:
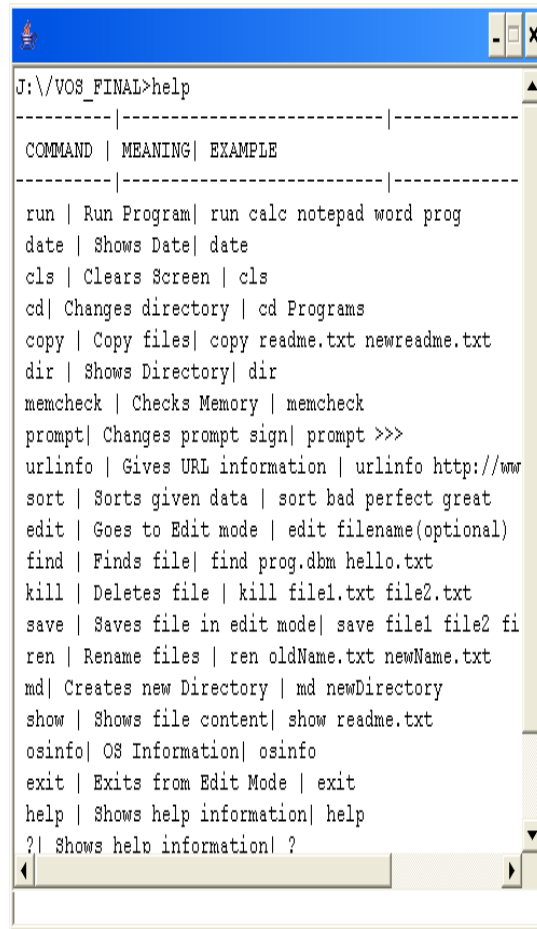
9.14 Change password:

9.15 Deleting a user

9.16Terminal prompt:

9.17 Command list:

9.18 Help command:

```
J:\/VOS_FINAL>help
----------|--------------------------|-------------
 COMMAND | MEANING| EXAMPLE
----------|--------------------------|-------------
 run | Run Program| run calc notepad word prog
 date | Shows Date| date
 cls | Clears Screen | cls
 cd| Changes directory | cd Programs
 copy | Copy files| copy readme.txt newreadme.txt
 dir | Shows Directory| dir
 memcheck | Checks Memory | memcheck
 prompt| Changes prompt sign| prompt >>>
 urlinfo | Gives URL information | urlinfo http://ww
 sort | Sorts given data | sort bad perfect great
 edit | Goes to Edit mode | edit filename(optional)
 find | Finds file| find prog.dbm hello.txt
 kill | Deletes file | kill file1.txt file2.txt
 save | Saves file in edit mode| save file1 file2 fi
 ren | Rename files | ren oldName.txt newName.txt
 md| Creates new Directory | md newDirectory
 show | Shows file content| show readme.txt
 osinfo| OS Information| osinfo
 exit | Exits from Edit Mode | exit
 help | Shows help information| help
 ?| Shows help information| ?
```
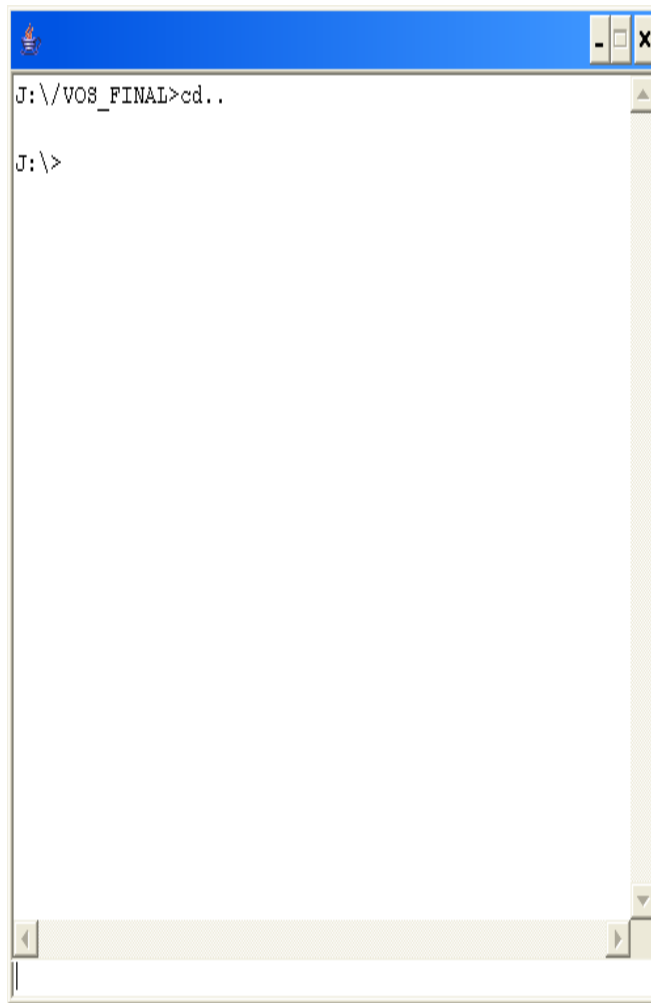
9.19 cd command:

```
J:\/VOS_FINAL>cd
J:\/VOS_FINAL

J:\/VOS_FINAL>
```
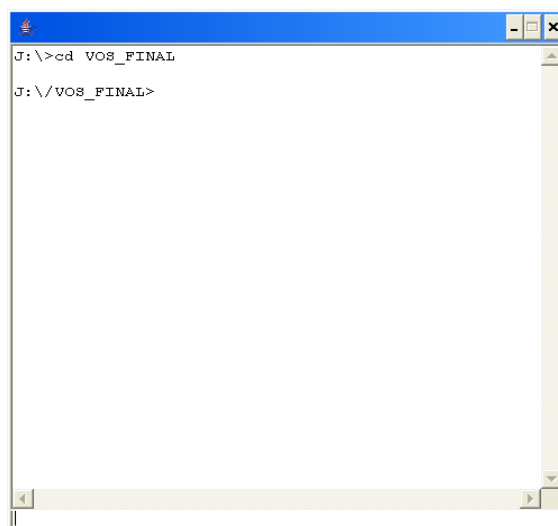
9.20 Changing to the root directory:
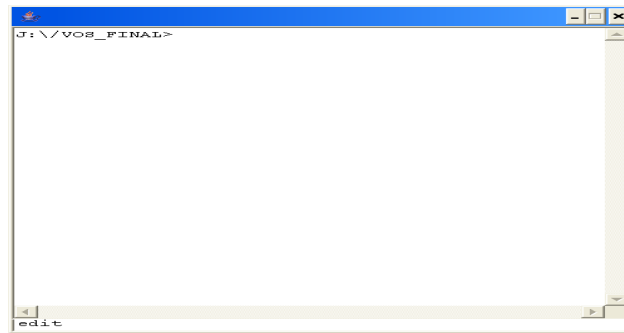
```
J:\/VOS_FINAL>cd..

J:\>
```
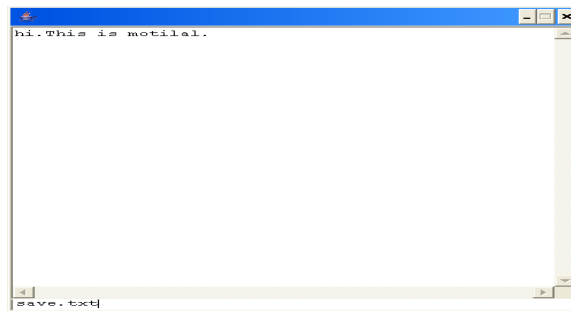
9.21 Changing to a sub directory:
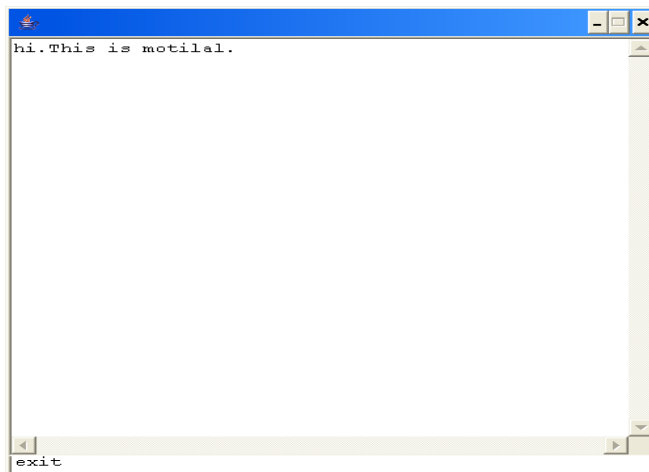
```
J:\>cd VOS_FINAL

J:\/VOS_FINAL>
```

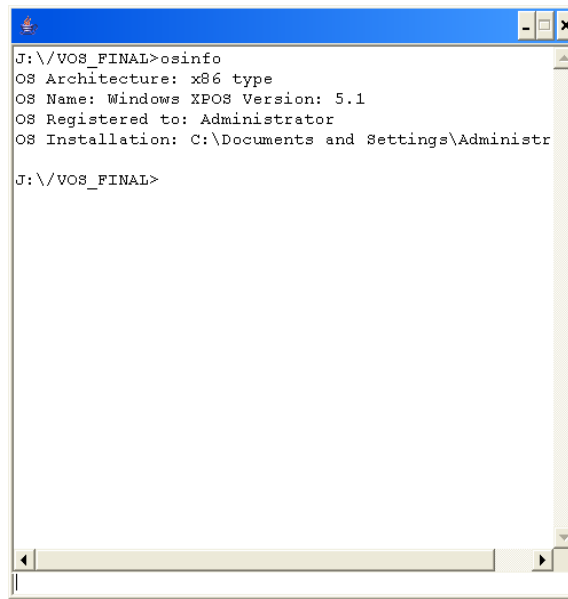**9.22 Converting to edit mode:**

9.23 Saving the text file:
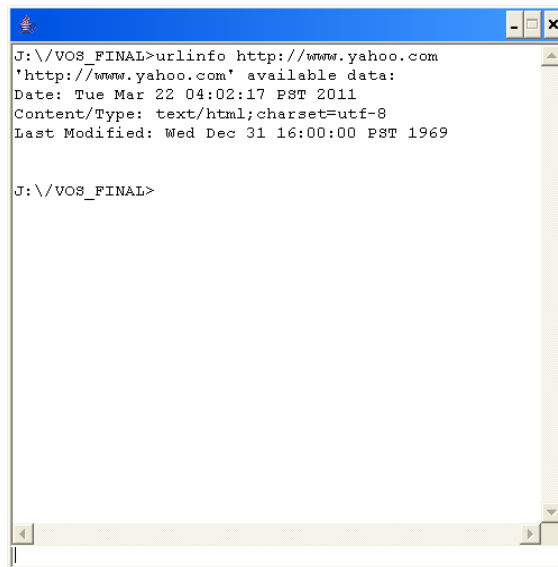

9.24 Exiting from edit mode:

9.25 Getting information about OS:

```
J:\/VOS_FINAL>osinfo
OS Architecture: x86 type
OS Name: Windows XPOS Version: 5.1
OS Registered to: Administrator
OS Installation: C:\Documents and Settings\Administr

J:\/VOS_FINAL>
```
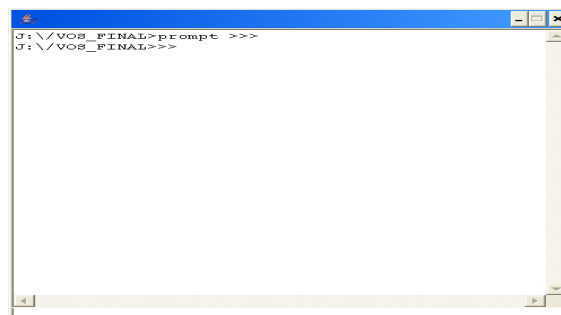
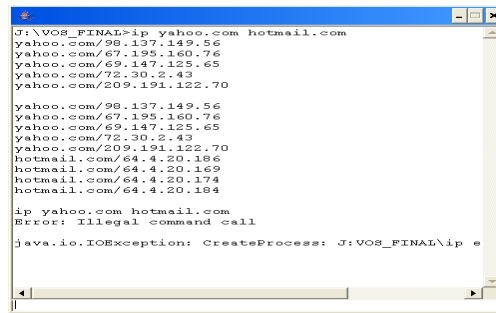**9.26 Getting information about a particular domain:**

```
J:\/VOS_FINAL>urlinfo http://www.yahoo.com
'http://www.yahoo.com' available data:
Date: Tue Mar 22 04:02:17 PST 2011
Content/Type: text/html;charset=utf-8
Last Modified: Wed Dec 31 16:00:00 PST 1969


J:\/VOS_FINAL>
```

9.27 Changing the prompt:

```
J:\/VOS_FINAL>prompt >>>
J:\/VOS_FINAL>>>
```

9.28 Getting the ip address of a domain:

9.28 Exit message:

10. CONCLUSION:

It has been a great pleasure for me to work on this challenging paper. I hope this application can be used extensively in the IT/BPO companies to train the people who works on different operating systems. As it has got very user friendly look and feel, it will be definitely succeeded in the market as a product. As it has got very user friendly look and feel, it will be definitely succeeded in the market as a product. This application can be extended so that it will give maximum performance including the security related aspects, resulting releasing of this application as a product in the open market. As it has got very user friendly look and feel, it will be definitely succeeded in the market as a product.Virtual operating system can be used in various colleges and universities instead of the OS they use. It can be extended to big industries where training for the fresher is mandatory.

REFERENCES:

[1] Core Java2 Volume I – Fundamentals; SunSoft Press; S. Horstmann&GaryCornell
[2] Core Java2 Volume II – Advanced Features SunSoftPress;S.Horstmann&GaryCornell
[3] Professional Java Server Programming; Wrox publication;12Authors
[4] An integrated approach to Software Engineering.ByPankajJalote.SecondandThirdEdition.
[5] Java Server Pages in 24 hours. Jose Annunziato dso and StepanieFeslerKaminaris
[6] www.bhartisoftland.com/technologies-skill-set (mvc architechture)

Authors Biography:

[1]VARADALA SRIDHAR is from HYDERABAD, ANDHRAPRADESH, and BORN on 25th JAN 1985. Completed M.TECH in ECE with specialization (WIRELESS AND MOBILE COMMUNICATION SYSTEMS) from vardhaman college of engineering affiliated by JNTUH in 2011.he has completed M.Sc (IT)from Nagarjuna University, guntur, AndhraPradesh.and B.TECH in ECE from vidya jyothi institute of technology affiliated by JNTUH in 2007. Currently he is working as an Assistant professor in ECE department at Vidya Jyothi Institute of Technology, Hyderabad from 2010. His areas of research interests include Wireless and Mobile communication ssystems,Digitalsignalprocessing,Imageprocessing,
Telecommunications,communication systems, Signal processing,Embedded systems. He has published more than 20 international research journals papers.He is Lifetime Membership of ISTE, IETE, IAENG, SDIWC, IACSIT, CSTA, UACEE, and AND MCDM. He is reviewer of SDIWC, IJARCET, SSRGJ-IJCTT; He is Editorial board member of IJCIT, IJARCET, IJOART, IJARECE, IJARCSEE, AND IJSETR.