

WEB BASED REMOTE CONSOLE

A PROJECT REPORT

Submitted by

SAIKRISHNAN B. - 21108205075

SRIRAM V.S. - 21108205092

VINAYAK R. - 21108205104

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY



RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY:: CHENNAI 600 025

APRIL 2012

WEB BASED REMOTE CONSOLE

A PROJECT REPORT

Submitted by

SAIKRISHNAN B. - 21108205075

SRIRAM V.S. - 21108205092

VINAYAK R. - 21108205104

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY



RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY:: CHENNAI 600 025

APRIL 2012

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**WEB BASED REMOTE CONSOLE**” is the bonafide work of “**Saikrishnan B, Sriram VS, Vinayak R**” who carried out the project work under my supervision.

SIGNATURE

Mr.P.Kumar

HEAD OF THE DEPARTMENT

Information Technology

Rajalakshmi Engineering College

Rajalakshmi Nagar

Thandalam

Chennai-602105

SIGNATURE

Ms.K.Poornimathi

SUPERVISOR

Senior Lecturer

Information Technology

Rajalakshmi Engineering College

Rajalakshmi Nagar

Thandalam

Chennai-602105

This project report is submitted for viva voice examination to be held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We thank the Almighty God for the successful completion of our project. Sincere thanks to our Chairman **Mr. S. Meganathan B.E, F.I.E** for his sincere endeavor in educating us in his premier institution.

We would like to express our deep gratitude to our beloved chairperson **Dr. (Mrs).Thangam Meganathan, Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing our project.

We also express our gratitude to our principal, **Dr. S. Renganarayanan, B.E., M.Sc., Ph.D.,** who helped us in providing the required facilities in completing our project.

We would like to thank our Head of the Department **Mr. P. Kumar, M.E.,** for his invaluable guidance and encouragement for completing our project.

We would also like to thank our supervisor **Ms. K. Poornimathi, M.Tech.,** for constantly guiding us and motivating us throughout the course of our project.

We are very thankful to our project coordinator **Ms. L. Priya, M.Tech.,** for her invaluable guidance, ideas, advice and encouragement for the successful completion of our project.

We express our gratitude to our parents and friends for extending their full support to us.

ABSTRACT

Our web based remote console is tailored to provide an intuitive interface to manage critical and routine administration tasks on various powerful operating systems. The goal of our project is to provide a unified remote graphical user interface to manage administrative tasks. The main objective is to provide a system that allows the administration, configuration and execution of tasks on remote systems in a unified and intuitive manner. System administrators previously had to write several complex scripts. Malicious scripts could jeopardize the critical data or performance of a live environment, in scenarios where availability cannot be compromised. Our system solves this problem and also the platform dependencies. Since we use open source technology with a highly modular approach and well documented code, it shall be easy for script writers to enhance or extend our code to even their custom spun distributions. We exhaustively cover a wide range of commands, so that the power user will appreciate our suite.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	Acknowledgements	iii
	Abstract	iv
	List of Tables	ix
	List of Figures	x
	List of Abbreviations	xii
	List of Symbols	xiv
1.	Introduction	1
	1.1. Project Definition	1
	1.2. Need for proposed system	2
	1.3. Application of proposed project	2
2.	Literature Survey	3
	2.1. Brian Down, Management of Systems and Services Made Simple with the Oracle Solaris Service Management Facility, Oracle Inc, 2010	3
	2.2. Daniel Price, Solaris Zones, Sun Microsystems Inc, 2004	3
	2.3. Michael Hasenstein, SuSE LVM Management Paper, SuSE Inc, 2001	3
3.	Problem formulation	4
	3.1. Main Objective	4
	3.2. Specific Objective	4
	3.3. Methodology	4
	3.4. System Requirements	4
	3.4.1. Hardware Platform Specification	4

CHAPTER NO.	TITLE	PAGE NO.
	3.4.2. Software Platform Specification	5
4.	System Analysis and Design	6
	4.1. Architecture Diagram	6
	4.2. Entity Relationship Diagram	7
	4.3. Data Flow Diagram	8
	4.3.1. Higher-level Data Flow Diagrams	9
	4.3.2. Level 2 Data Flow Diagram	11
	4.4. Database Design	12
	4.4.1. EndUser table design	12
	4.4.2. SeniorUser table design	12
	4.4.3. Hosts table design	13
	4.4.4. Log table design	13
5.	Functional Description	14
	5.1. Sign up Module	14
	5.2. Task Management Module	14
	5.3. Host Registration Module	14
6.	System Implementation	15
	6.1. Remote Host Administrator Sign up	15
	6.2. Login	16
	6.3. Host Functions	16
	6.3.1. Host Addition	17
	6.3.2. Host Deletion	18
	6.3.3. Host Modification	19
	6.4. Navigating to task execution	20
	6.4.1. Task Panel	20
	6.5. Software package management	21
	6.5.1. Install package	22

CHAPTER NO.	TITLE	PAGE NO.
	6.6. Disk Management	23
	6.6.1. Partition listing	24
	6.6.2. Zettabyte Filesystems Management Suite	26
	6.6.2.1. Zettabyte Pool Management	26
	6.6.2.2. IO Statistics	27
	6.7. User Management	28
	6.7.1. Add user	28
	6.7.2. Modify user properties	29
	6.7.3. Delete user	30
	6.8. Group Management	31
	6.8.1 Group Listing	31
	6.8.2. Add group	31
	6.8.3. Modify group	32
	6.8.4. Delete group	33
	6.9. Automating Tasks	34
	6.9.1. Cron Management	34
	6.9.2. Anacron Management	35
7.	Testing	36
	7.1. Unit Testing	36
	7.1.1. Host Listing	36
	7.1.2. Package Installation	37
8.	Conclusion and Future Enhancements	39
	8.1. Conclusion	39
	8.2. Future Enhancements	39
9.	References	47

Appendices

Appendix A. Setup

Appendix B. Setting up cron jobs in Unix

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
4.1	EndUser	12
4.2	SeniorUser	12
4.3	Hosts	13
4.4	LogTable	13
7.1	Test case for host reachability	37
7.2	Test case for package availability	38

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.1	System Architecture	5
4.2	Entity Relationship Diagram	7
4.3	Level 0 Data Flow Diagram	9
4.4	Signup Level 1 DFD	9
4.5	Host Registration Level 1 DFD	9
4.6	Task Management Level 1 DFD	10
4.7	Service Maintenance Level 1 DFD	10
4.8	Service Maintenance Level 2 DFD	11
6.1	Sign up process	15
6.2	Login page	16
6.3	Host listing	16
6.4	Host Listing with host reachability status	17
6.5	Host Addition	18
6.6	Host Deletion	18
6.7	Host modification	19
6.8(a)	Before modification	19
6.8(b)	After modification	19
6.9	Host selection	20
6.10	Task Panel	20
6.11	Software Package Management overview	21
6.12	Install package	22
6.13	Disk management tasks	23
6.14	Partition overview	24

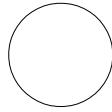
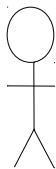
FIGURE NO.	FIGURE NAME	PAGE NO.
6.15	Toggle boot flag	25
6.16	Deleting partition	25
6.17	ZPool overview	26
6.18	ZPool management tasks	27
6.19	ZPool IOStat	27
6.20	User management overview	28
6.21	Adding a user	29
6.22	Modify user	29
6.23	Delete User	30
6.24	Group Listing	31
6.25	Add group	32
6.26	Modify group	33
6.27	Delete group	33
6.28	Cron management	34
6.29	Anacron management	35
7.1	Host listing with reachability status report	36
7.2	Successful package installation	37
7.3	Error in package installation	38
A.1	Apache installation screenshot	40
A.2	Content transfer	41

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
LVM	Logical Volume Manager
OS	Operating Systems
SPARC	Scalable Processor ARChitecture
XML	eXtensible Markup Language
HTML5	HyperText Markup Language version 5
CSS3	Cascading StyleSheet version 3
GUI	Graphical User Interface
MySQL	My Structured Query Language
HRS	Host Registration Server
PSP	Python Server Pages
RHA	Remote Host Administrator
IP	Internet Protocol
RHS	Remote Host Server
MB	Mega Byte
GB	Giga Byte
KB	Kilo Byte
B	Byte
fdisk	Partition manipulator for Linux
ZFS	Zettabyte File System
Zpool	Zettabyte Pool
I/O	Input/Output
Zfs-fuse	Zettabyte FileSystem- Filesystem in UserSpacE
IOStat	Input/Output Statistics

ABBREVIATION	EXPANSION
GID	Group Identifier
ID	Identifier
Crontab	Cron TABLE
MySQLDb	My Structured Query Language DataBase
DFD	Data Flow Diagram

LIST OF SYMBOLS

SYMBOLS	DEFINITION
	ENTITY
	USECASE
	DATA STORE
	PROCESS
	UNIDIRECTIONAL FLOW
	ACTOR

CHAPTER 1

INTRODUCTION

1.1. Project Definition

The Web Based Remote Console is tailored to provide an intuitive interface to manage critical as well as routine administration tasks on various powerful OS' including Linux, Solaris and extensible to other Unix OS as well. The goal of our project is primarily to provide a unified remote interface to manage administrative tasks. The main objective is to provide a system that allows to administer, configure and execute tasks in remote systems in a unified and intuitive manner.

In the feasibility study stage, we referred to few articles to identify the benefits and shortcomings of the existing systems in the system administration arena. It can be summarized as put forth. We found a paper which explains how to manage logical volumes in a SuSE Linux instance. It provides the overview of volume management in Linux, which provided the stepwise procedure to create physical extents, physical volumes and volume groups. It demonstrates how to create volumes with the various commands and switches that are required.

Command line offers unparalleled flexibility in implementing disk administration. Moreover, the system administrator can easily attach and detach disks even in live environment, without downtimes. But since the scripting environment is exposed to the administrator, it is possible to operate malicious activity like the fork bomb, which may totally bring the system down.

Example:

```
main()
{
    for(;;)
        fork();
}
```

Overwhelming switches have to be remembered for every command line. Solaris zones create virtualized application execution environments within a single OS. There are advantages. Firstly, it becomes simple to consolidate multiple virtual environment instances under a roof. Moreover, multiple OS' of different architecture (x86, SPARC etc) can be set under a roof. However, there are shortcomings, like zones are not easy to administer. From OS environment creation to configuration, the administrator needs to be thorough with the commands to execute them. Thus, it is all complicated. Moreover, zones cannot be administrated over a network.

Various initialization scripts are required during the boot time, that can be aggregated and stored in a simple XML file called a service.

Any service can be in online, offline, maintained or disabled modes. There are few advantages. Simple management of services using modes of service operation is possible that enables configuring it. The configuration is simplified in XML formatted open standards. However, it is not possible to classify services, to club them together. Also, services are highly interdependent-malconfigured services are sensitive and cause them to reach a maintenance mode.

1.2. Need for proposed system

The various existing systems specified above, have several disadvantages that we aim to overcome. Certain tasks require complex scripts. Imagine the complexity to add 500 users in a college environment with the permutations of switches and the loop constructs.

Scripting ports, when opened, the administrators get their hand to create malicious scripts, for example, fork bombs. To overcome this, organizations require to deploy complex life cycle patterns- to test the scripts on standalone machines, loop them and deploy in live environments. This is expensive. Lastly, the daunting tasks require highly skilled system administrators. Moreover, there are high platform dependencies.

1.3. Application of proposed project

To make system administration simple, we provide an intuitive HTML5 and CSS3 GUI that is immersive but not ostentatious. We classified tasks, with a powerful engine that converts common graphical actions into a command line. We do exhaustive coverage of every command to provide the wide range of switches and options, so that the power user will appreciate our tool.

CHAPTER 2

LITERATURE SURVEY

2.1. Brian Down, Management of Systems and Services Made Simple with the Oracle Solaris Service Management Facility, Oracle Inc, 2010

Various initialization scripts that are required during boot time can be aggregated and stored in a simple XML file called a service. Any service can be in online, offline, maintained or disabled modes. Simple management of services using modes of service operation is possible. The configuration is simplified in XML formatted open standards. However, it is not possible to classify services to club them together.

Advantages:

1. It is simple to configure system services.
2. Configuration files are in XML format, thus following open standards.

Limitations:

1. It is not possible to club multiple services together.

2.2. Daniel Price, Solaris Zones, Sun Microsystems Inc, 2004

Solaris zones create virtualized application execution environments within a single instance of the operating system. It becomes simple to consolidate multiple virtual environment instances under a roof. Multiple OS' of different architecture (x86, SPARC etc) can be set under a roof.

Advantages:

1. Virtual environments can be easily created, paving way to efficient development cycles.

Limitations:

1. Zones are difficult to administer.
2. Network based administration of zones is not possible.

2.3. Michael Hasenstein, SuSE LVM Management Paper, SuSE Inc, 2001

This paper explains how to manage logical volumes in a SuSE Linux instance. It provides the overview of volumes in Linux, the stepwise procedure to create physical extents, physical volumes and volume groups.

Advantages:

1. It is easy to attach and detach disk volumes in live environments without server downtimes.

Limitations:

1. Overwhelming number of switches are to be remembered.

CHAPTER 3

PROBLEM FORMULATION

3.1. Main Objective

The main objective of this project is to provide a system that allows to administer, configure and execute tasks in remote systems in a unified and intuitive manner. The GUI allows the RHA to directly administer the system tasks, so that he does not have to remember too many commands. The GUI is purely designed with HTML5 and CSS3 which provide a pleasant interface.

3.2. Specific Objective

The Web Based Remote Console is designed with the task panel which elaborates the various system tasks. A powerful engine that converts common graphical actions into respective commands. Log records are stored in the database which shows the various tasks performed by the RHA and it is classified such that it shows the system performances also. The SSH provides the security aspect for the remote systems.

3.3. Methodology

In this project the Object Oriented Methodology is used. Object Oriented Methodology provides reusability, inheritance, and encapsulation. OOM closely represents the problem domain. Because of this, it is easier to produce and understand designs. The system designed using this approach is closer to the real world as real world functioning of the system is directly mapped into the system designed using this approach. Object Oriented Methodology designs encourage more reuse. New applications can use the existing modules, thereby reduces the development cost and cycle time.

3.4. System Requirements

All computer software needs certain hardware components or other software resources to be present on a computer, to be used efficiently. These prerequisites are known as system requirements.

3.4.1. Hardware Platform Specification

Processor	:	Pentium IV
RAM	:	512 MB RAM
Secondary Storage	:	20 GB HDD

3.4.2. Software Platform Specification

Operating System	: Ubuntu 11.10 32 Bit
Programming Language	: Python 2.7
Front End	: Python server pages
Documentation Tool	: LibreOffice Writer

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

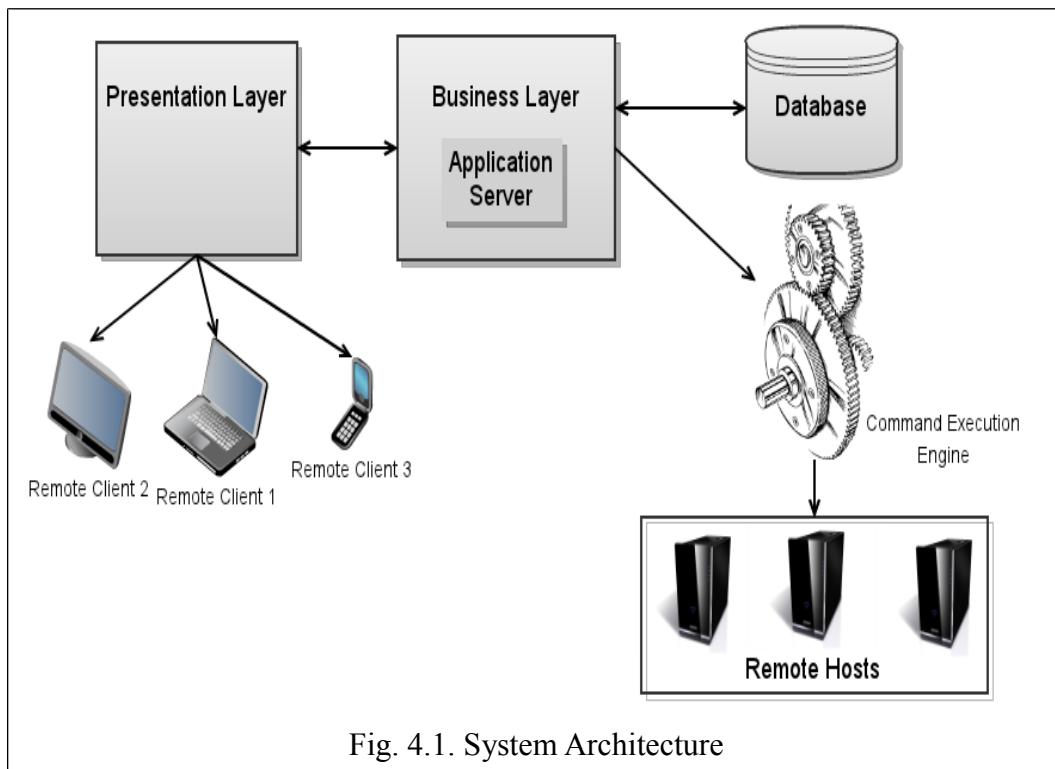
4.1. Architecture Diagram

The system is organized in a three-tier architecture, where components are designed modularly , keeping service orientation in mind. The back-end includes the database and the remote hosts. The logical layer includes Apache web server, processing logic in Python Server Pages (PSP).

The presentation layer consists of a GUI implementation in HTML5. However, the modules in the business layer is service oriented, which allows scope for front-end development in other languages as well. When remote administrators sign up and register their servers in a registration system, entries get committed into MySQL database in the back end.

The middle tier consists of a command execution engine, abstracted by the task management module. It takes graphical tasks as input, digests and renders platform-tailored commands as output. The commands are then executed in the remote hosts by PSP. The Apache server hosts the pages and provides the PSP platform using mod_python libraries.

The presentation layer is pluggable, due to high modularity and loose coupling of the business layer. Our implementation uses HTML5 technology with CSS3 frills for cutting edge functionality, making mundane administration tasks highly intuitive.



4.2. Entity Relationship Diagram

An entity relation diagram is an abstract and conceptual representation of data. This modeling produces a type of conceptual schema or semantic data model of a system, of a relational database and its requirements is in top-down fashion.

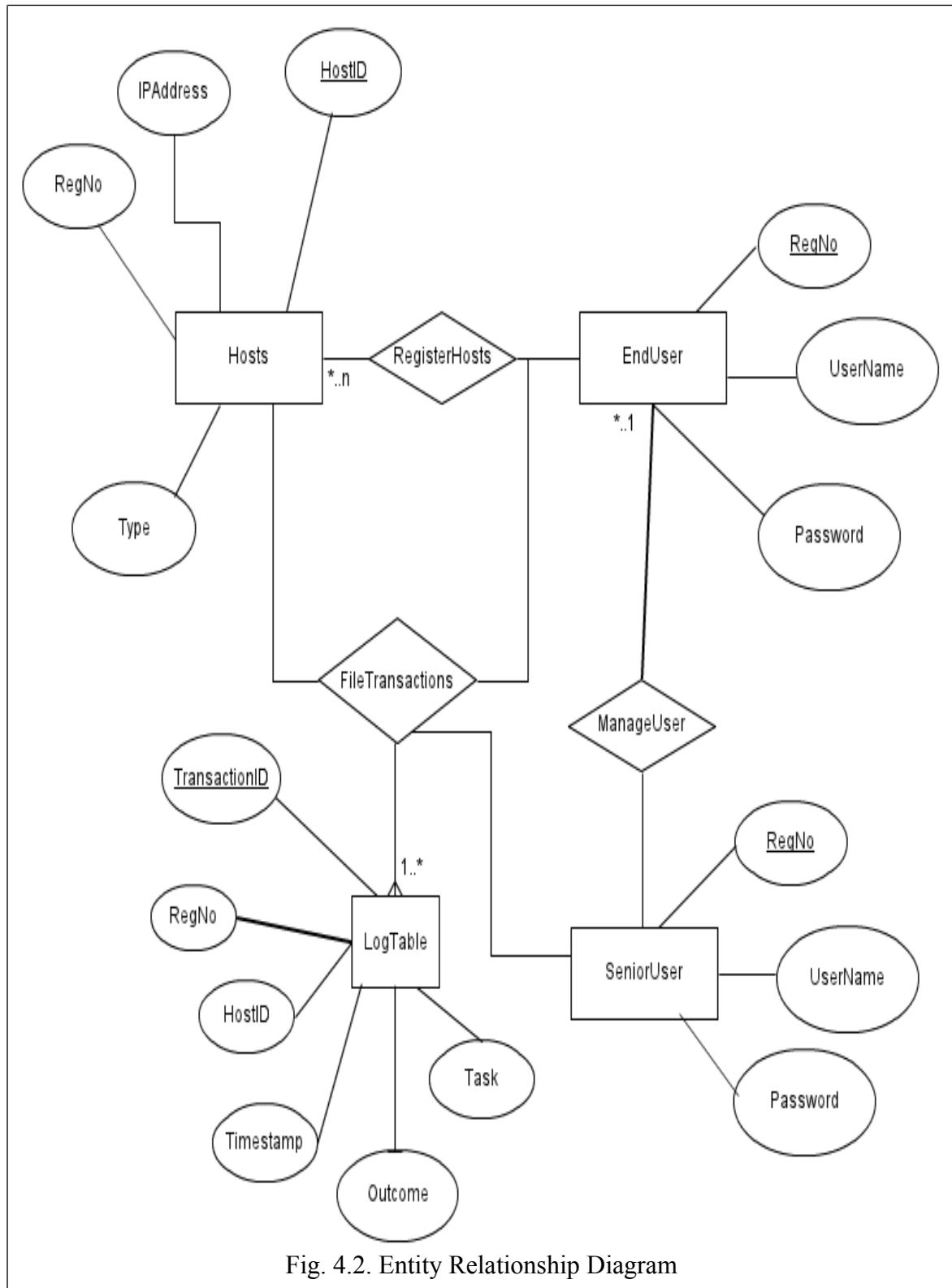
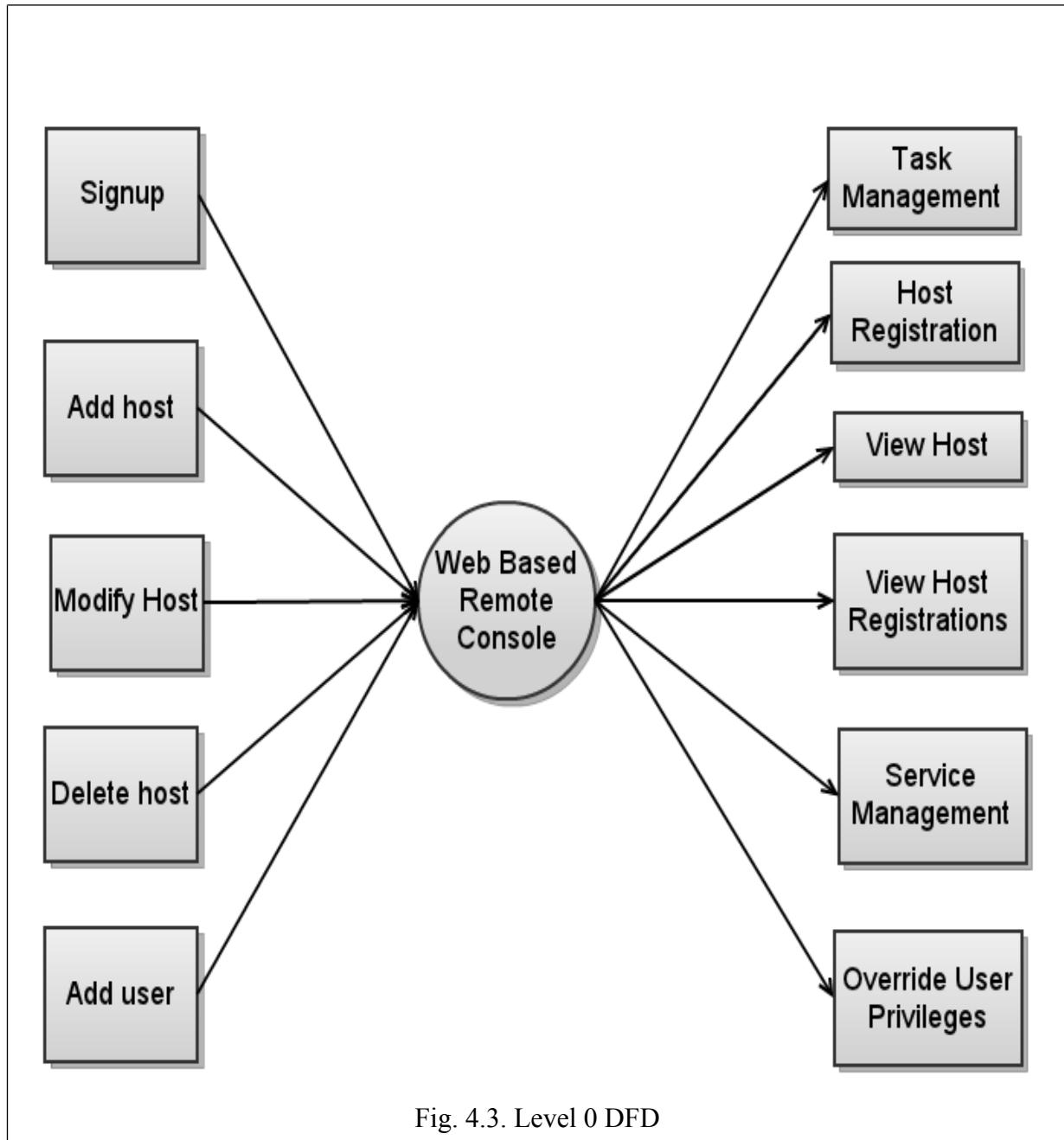


Fig. 4.2. Entity Relationship Diagram

4.3. Data Flow Diagram

Data flow diagram is a graphical representation of flow of data through an information system, modeling its process aspects. Often, they are a preliminary step used to create an overview of the system, which can later be elaborated. Data flow diagrams can also be used for the visualization of data processing. The following figure shows the level zero data flow diagram of our system. The inputs are segregated on one side, and the outputs on the right hand.



4.3.1. Higher-level Data Flow Diagrams

The data flow diagram that shows the entire system within a single diagram is the top level DFD. The DFDs that are expansion of processes at the top level are the higher-level DFDs.

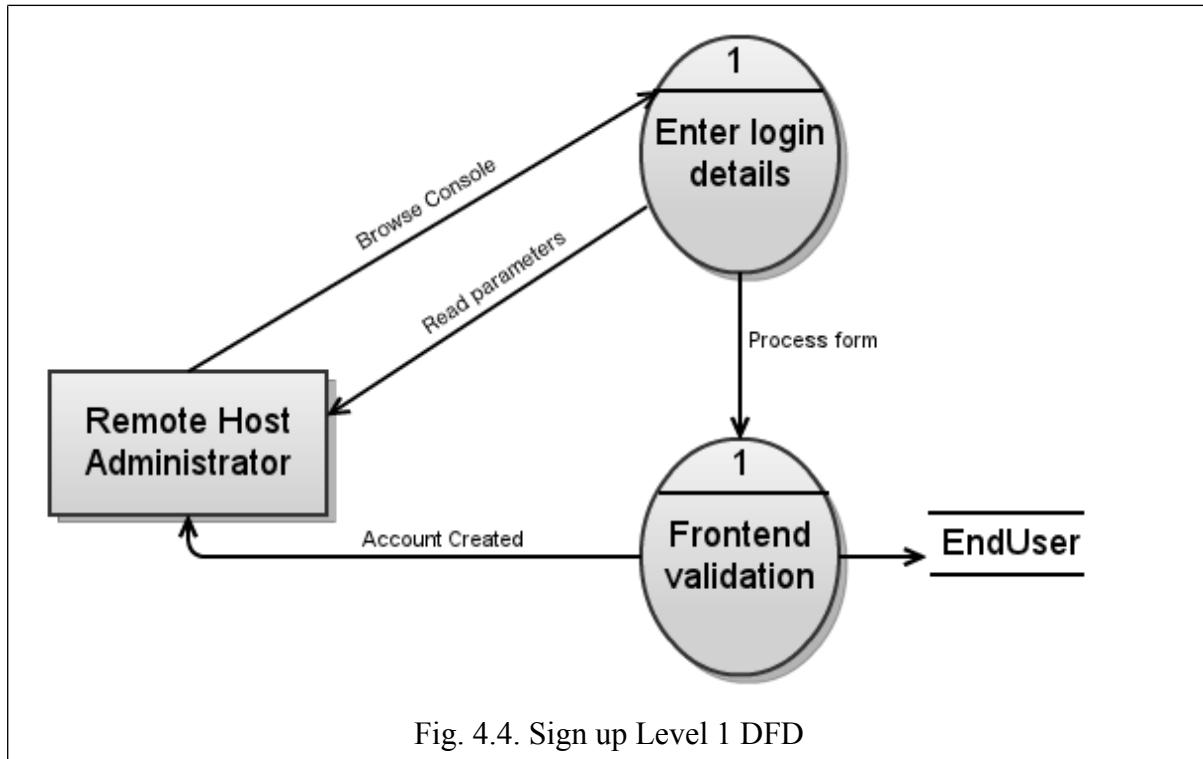


Fig. 4.4. Sign up Level 1 DFD

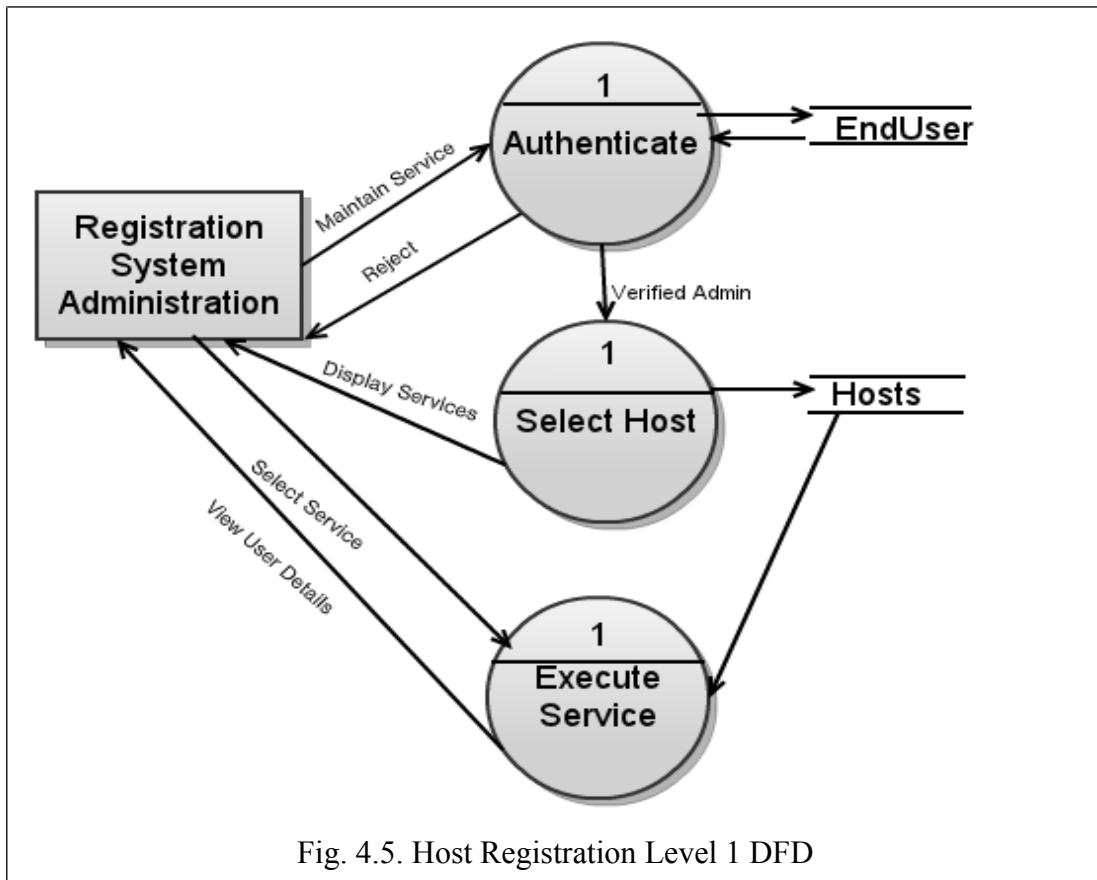


Fig. 4.5. Host Registration Level 1 DFD

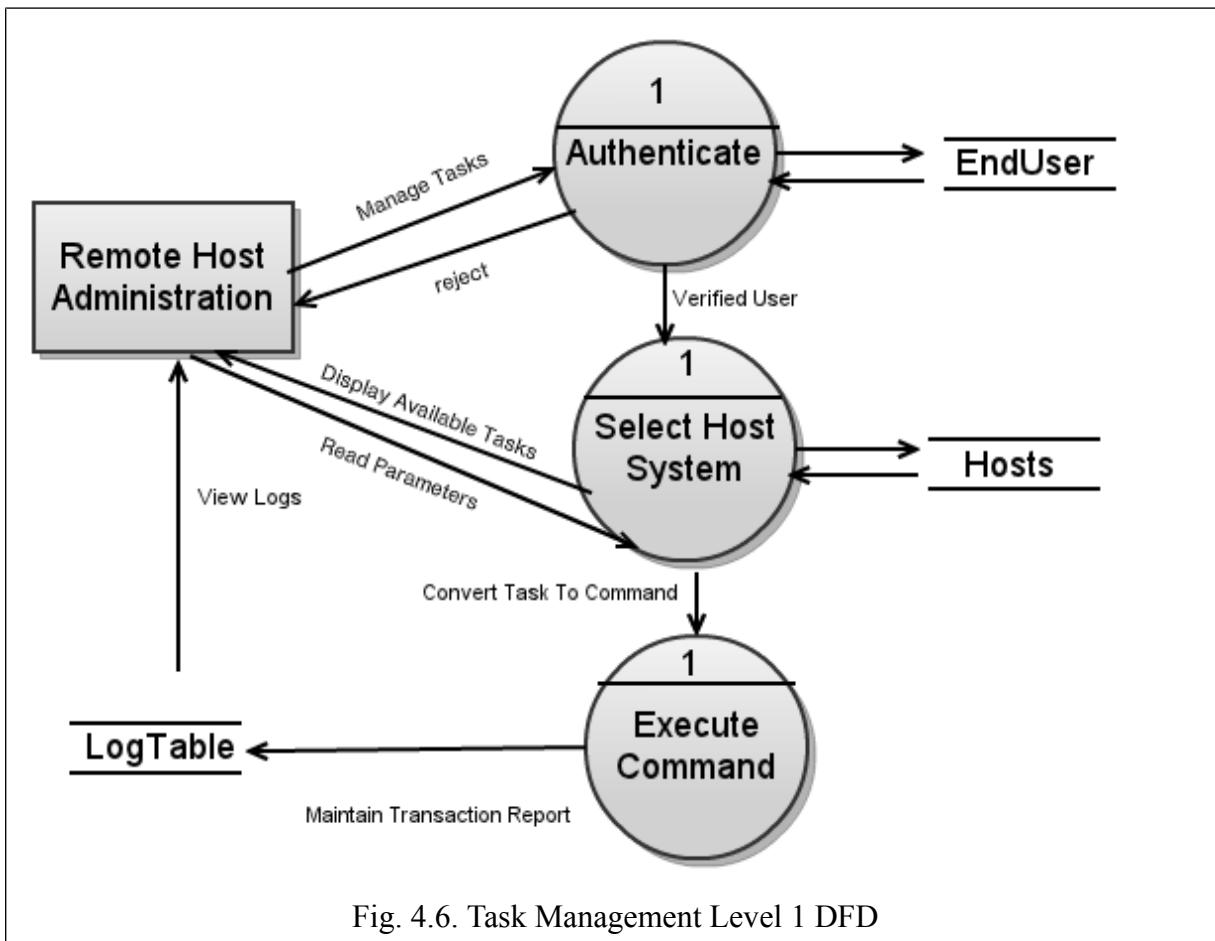


Fig. 4.6. Task Management Level 1 DFD

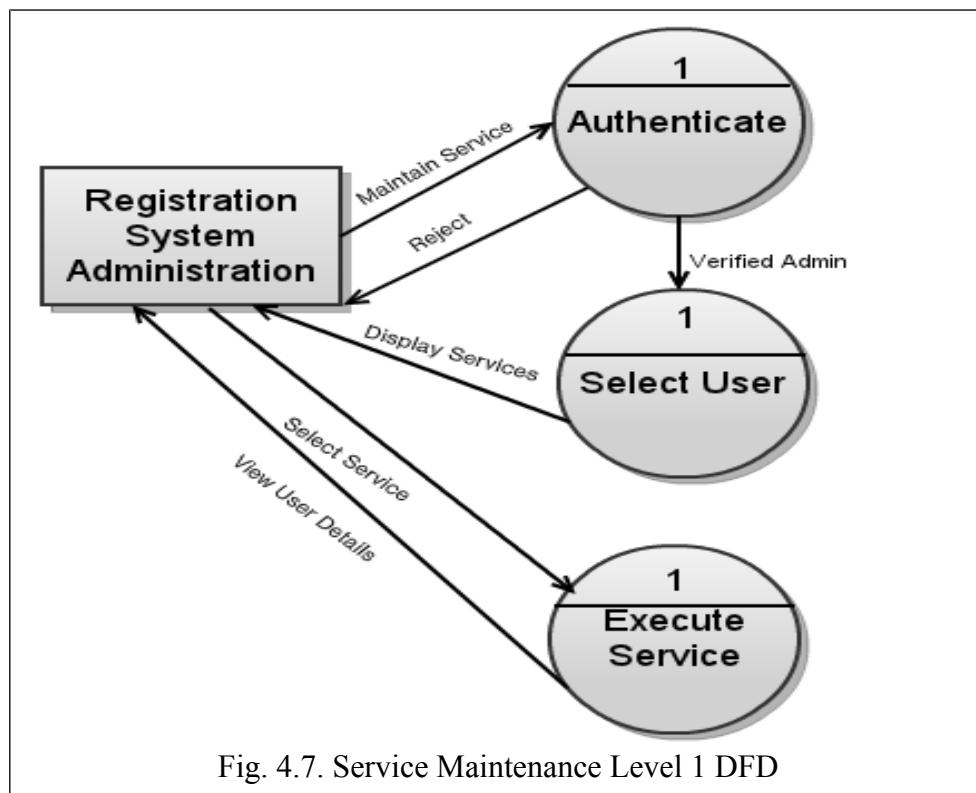


Fig. 4.7. Service Maintenance Level 1 DFD

4.3.2. Level 2 Data Flow Diagram

Processes from the top level DFD may be broken down or decomposed into number of levels, if they are complex, or may not be broken down, if they are simple. The DFD that are expansions of processes at the top level are level 2 DFDs. The level 2 DFD of our system is shown in the following figure.

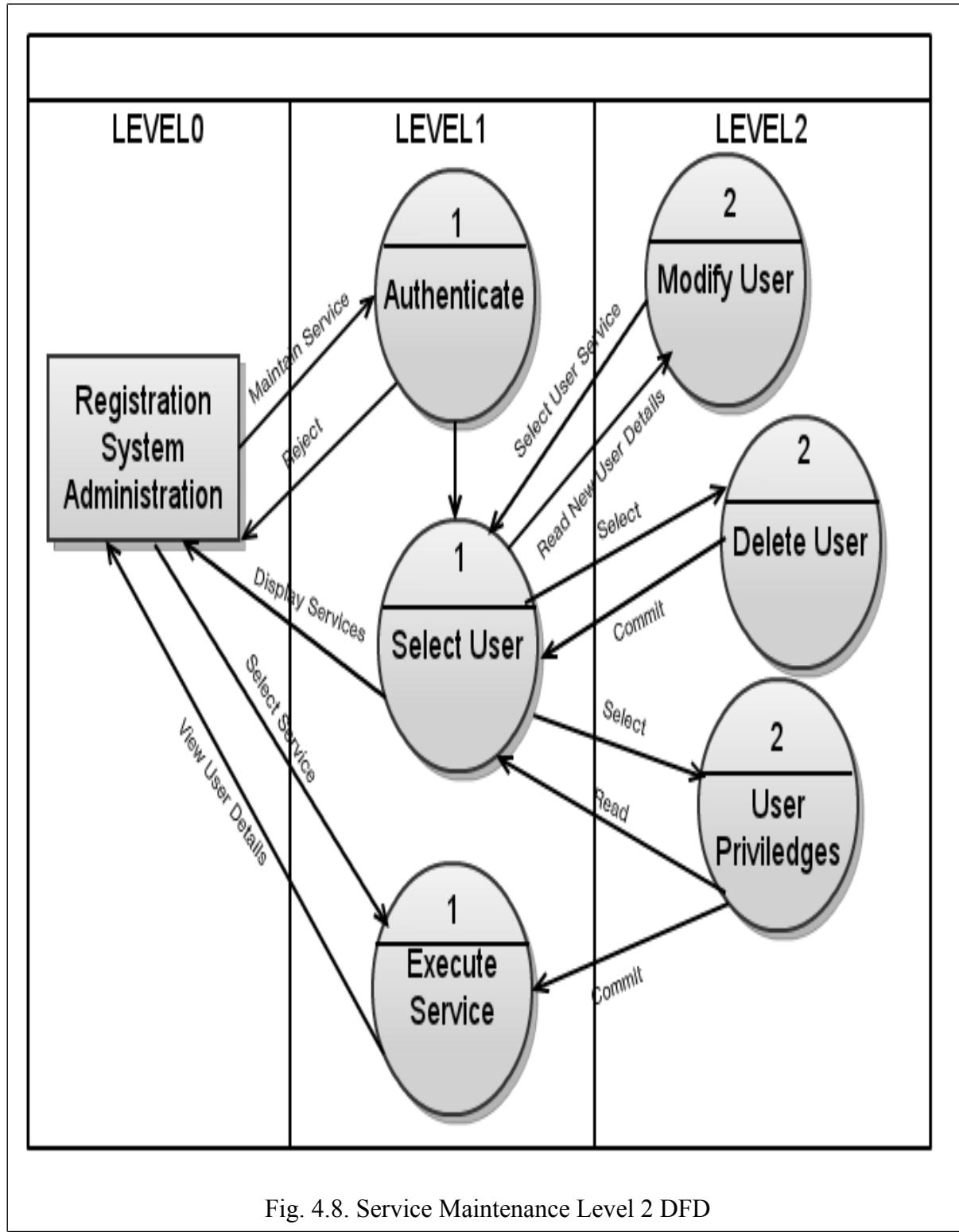


Fig. 4.8. Service Maintenance Level 2 DFD

4.4. Database Design

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and the physical storage parameters, needed to generate a design in data definition language, which can be used to create a database. A fully attributed data model contains detailed attributes for each entity. The database design steps include determining the relationships between the different data elements. It also includes superimposition of a logical structure upon the data on the basis of these relationships.

4.4.1. EndUser table design

This table contains the details that are used for authenticating remote host administrators. The primary key is the register number, regno, which is generated automatically by the system. The userID and pwd combination provides the authentication mechanism for RHA to login. The fields in the table are shown below.

COLUMN NAME	DATATYPE	DESCRIPTION
regno	Integer	Register number
userID	varchar	User identifier
pwd	varchar	Password

Table 4.1. EndUser Table

4.4.2. SeniorUser table design

This table shares the same structure as the EndUser table. The primary key is the register number, regno, which is generated automatically by the system. The userID and pwd combination provides the authentication mechanism for RHA to login. The main reason for this table is to enforce higher level of security, such that the authenticating mechanism for the highest user of the system is isolated from the remaining mechanism. The fields in the table are shown below.

COLUMN NAME	DATATYPE	DESCRIPTION
Regno	Integer	Register number
UserID	varchar	User identifier
Pwd	varchar	Password

Table 4.2. SeniorUser Table

4.4.3. Hosts table design

This table stores all the hosts that are registered under a remote host administrator. The primary key is the host identifier, hostID, which is generated automatically by the system. The regno is a foreign key, which references the EndUser table. The IPAddr is used to connect to the remote host. The platform specifies the operating system which the remote host operates in. The hostname is the human way to identify the host. The fields in the table are shown below.

COLUMN NAME	DATATYPE	DESCRIPTION
HostID	Integer	Host identifier
Regno	Integer	Register number
IPAddr	Varchar	Internet Protocol address
Hostname	Varchar	Host identification string
Platform	Varchar	Operating system of remote host

Table 4.3. Hosts Table

4.4.4. Log table design

This table logs all the tasks that are performed by various remote host administrators. The transaction identifier, TxnID, is the primary key which is used to uniquely identify a task. HostID is used to identify on which remote host the task was performed. The regno serves as a foreign key to identify which administrator performed an activity. Timestamp records the time when the task was performed. Task indicates what task was performed. Outcome indicates the standard output, error and command that was executed in background to accomplish the task. The fields in the table are shown below.

COLUMN NAME	DATATYPE	DESCRIPTION
TxnID	Integer	Transaction identifier
HostID	Integer	Host identifier
Regno	Integer	Register number
Timestamp	Date	Time when task was run
Task	Varchar	String depicting the task
Outcome	Varchar	Output, error and command executed

Table 4.4. LogTable

CHAPTER 5

FUNCTIONAL DESCRIPTION

5.1. Sign up Module

This module enables the user to register for login and authentication. The user is provided with a form which has user-id and password as the main input fields. JavaScript validation is done when a user does not fill in a particular field- he is prompted with a pop-up window. Registration number is generated by the system and all the fields are committed to the database with the registration number taken as the primary key. It is used as the reference in the hosts and log tables to maintain the identification.

5.2. Task Management Module

The task management module is the focal point of the Web Based Remote Console. It consists of a task acceptance unit, task to command translation engine and a command retrieval unit. The task acceptance unit is designed with service orientation in mind. It is an endpoint that collects well defined sequence of user level actions, that can be passed in a string-tuple format to the command translation engine.

The command translation engine is a processing unit, that takes in predefined tasks and platform (OS) as input from the task acceptance unit. It converts user-level tasks into commands, concatenated with the switches and looping constructs, tailored to the OS that is passed as an argument to the engine. The command retrieval unit is used to present the output script of the command translation engine in an encoded format that is compatible with the target language that uses this entire module. This unit modifies escape characters such as “\n” or “\r” so that they are not mistakenly interpreted by the target language. The task management module does not feature a GUI, this design fosters loose coupling, re-usability and promotes platform diversity.

5.3. Host Registration Module

The host registration module is available to the remote host administrator (RHA) and registration system administrator (RSA) in a supervisor-level. The RHA can attach/detach hosts and configure them. To perform any of the administrator tasks, login is necessary. The privileges are determined based on the registration number from the EndUser database. To attach a host, the RHA selects the registration form with a hostname and the IP Address. He also provides the access type, to identify whether the host uses a private network or it is internet based. On successful completion of the form, validated by Javascript, the host is registered in the database with his identity and also modify and delete those hosts.

CHAPTER 6

SYSTEM IMPLEMENTATION

System development is the process of defining, designing, testing, and implementing a new software application or program.

6.1. Remote Host Administrator Sign up

Authorized persons can sign up as a Remote Host Administrator (RHA). The prospective RHA provides his actual first name, last name, userid, password with confirmation, security question and answer and contact details. In the event of an abnormal task that is executed, the RHA would be notified.

The screenshot shows a "Signup Form - Google Chrome" window. The URL in the address bar is "localhost/trial2/signup/signup.psp". The form is divided into sections:

- Name**: Fields for First Name ("John") and Last Name ("Smith").
- UserID**: Field containing "john".
- Password**: Field containing ".....".
- Confirm Password**: Field containing ".....".
- Security Question**: Fields for "Your birth place" ("California") and "Contact details".
- Contact details:** Fields for "Mobile Number" ("+919600026311") and "Email Address" ("johnsmith@wrc.com").
- Terms and Conditions**: A checked checkbox labeled "I accept to the Terms And Conditions".
- Buttons**: "Submit" and "Reset" buttons at the bottom.

Fig. 6.1. Sign up process

6.2. Login

Once the RHA signs up, he can login. He provides his userid and password to perform this action. Once the RHA is logged in, the host listing page opens.

Signup Form - Google Chrome
localhost/trial2/login.psp

Enter login details

UserID: john

Password: *****

Submit Reset

Fig. 6.2. Login Page

6.3. Host Functions

On successful login, the RHA is provided with a host listing screen. He can click on the hosts that are reachable, provided he registered them with the correct IP Address and root password. Once he clicks, he can navigate through the task panel and perform his tasks.

Template for our Project

localhost/PROJ/ver3.1/host_reg/host_list.psp

Add Host Delete Host Modify Host

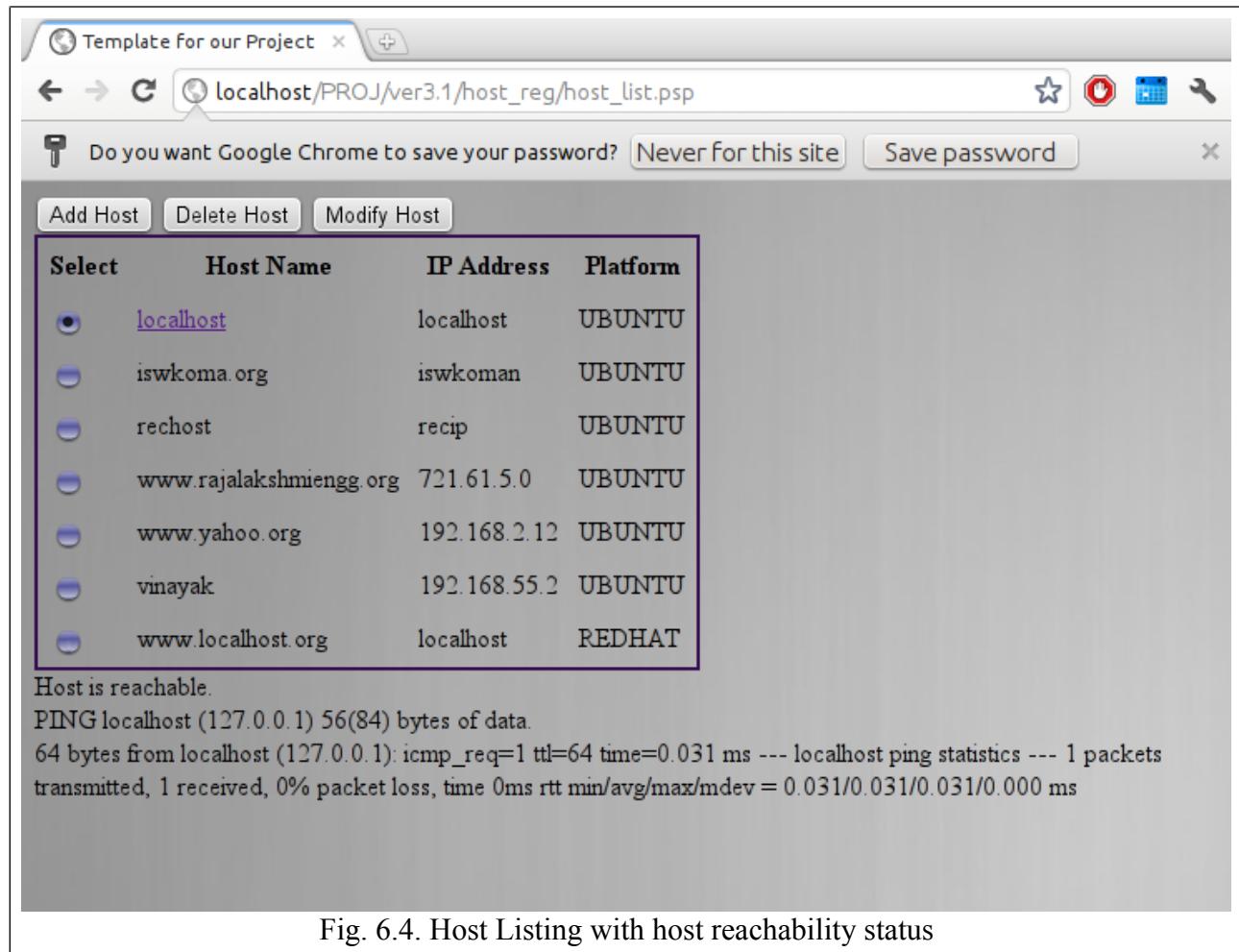
Select	Host Name	IP Address	Platform
<input checked="" type="radio"/>	localhost	localhost	UBUNTU
<input type="radio"/>	iswkoma.org	iswkoman	UBUNTU
<input type="radio"/>	rechost	recip	UBUNTU
<input type="radio"/>	www.rajalakshmiengg.org	721.61.5.0	UBUNTU
<input type="radio"/>	www.yahoo.org	192.168.2.12	UBUNTU
<input type="radio"/>	vinayak	192.168.55.2	UBUNTU
<input type="radio"/>	www.localhost.org	localhost	REDHAT

Host is reachable.
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_req=1 ttl=64 time=0.031 ms --- localhost ping statistics --- 1 packets transmitted, 1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.031/0.031/0.031/0.000 ms

Fig. 6.3. Host Listing

6.3.1. Host Addition

On login, the RHA is taken to the host listing page. This page lists all the hosts that are registered under him with specific details of the IP address and platform. The Remote Host Administrator (RHA) clicks on one of the radio buttons to perform an action. He can, here, either modify or delete the host. Addition of a Host into the HRS. To add a host into the HRS, the RHA clicks on the Add Host Button.



Select	Host Name	IP Address	Platform
<input checked="" type="radio"/>	localhost	localhost	UBUNTU
<input type="radio"/>	iswkoma.org	iswkoman	UBUNTU
<input type="radio"/>	rehost	recip	UBUNTU
<input type="radio"/>	www.rajalakshmiengg.org	721.61.5.0	UBUNTU
<input type="radio"/>	www.yahoo.org	192.168.2.12	UBUNTU
<input type="radio"/>	vinayak	192.168.55.2	UBUNTU
<input type="radio"/>	www.localhost.org	localhost	REDHAT

Host is reachable.
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_req=1 ttl=64 time=0.031 ms --- localhost ping statistics --- 1 packets transmitted, 1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.031/0.031/0.031/0.000 ms

Fig. 6.4. Host Listing with host reachability status

A modal window titled “Host Addition” pops up. The RHA is required to enter IP address, host name, platform and the root password of the remote host. Once he clicks the submit button, a page opens indicating the success or failure of the host addition. This final page contains a link that redirects the RHA back to the host listing page. The host addition module provides the remote host to add a host. Once the RHA makes that selection, he gets the modal window where he is requested to enter the IP address, host name, platform and the password and the host is added.

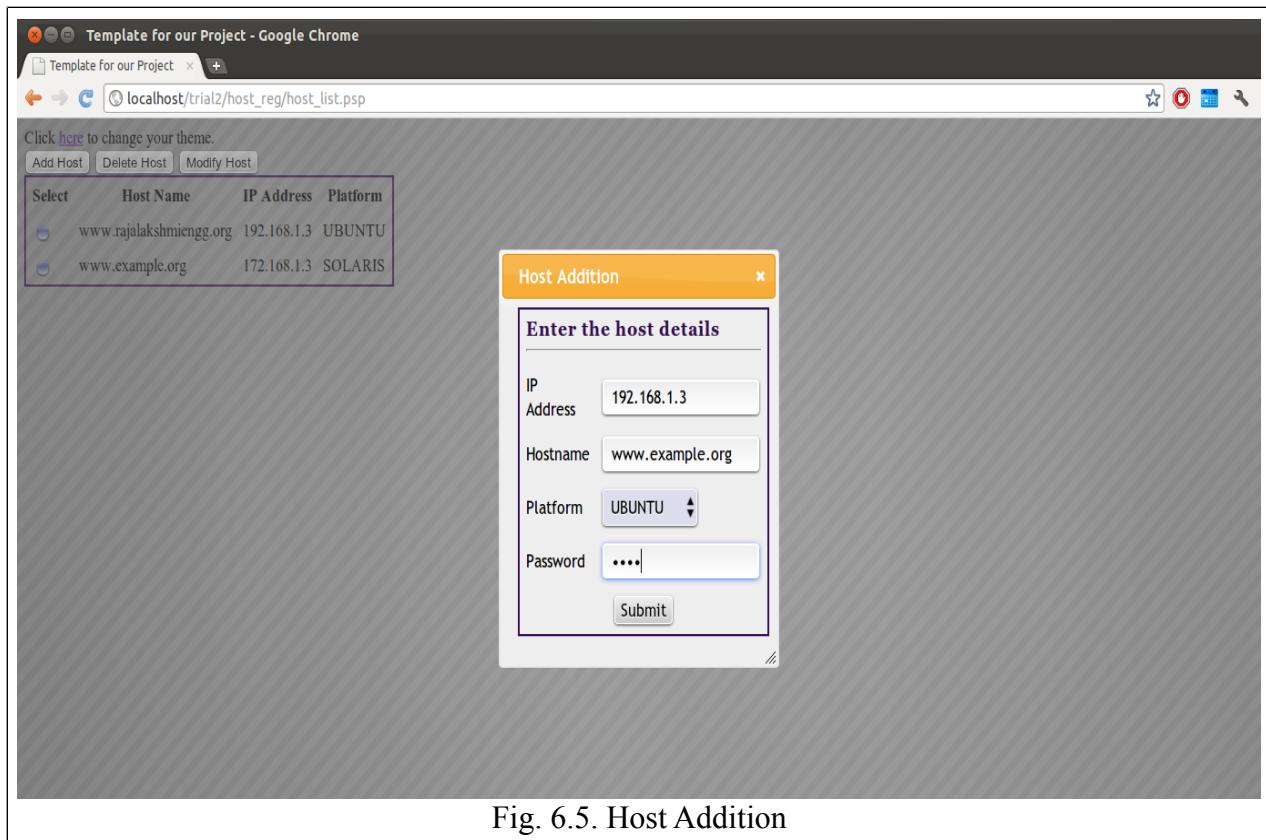


Fig. 6.5. Host Addition

6.3.2. Host Deletion

The host deletion module enables a remote host administrator to delete a host permanently. The RHA clicks on a host on the listing screen and clicks the delete host button to delete the host.

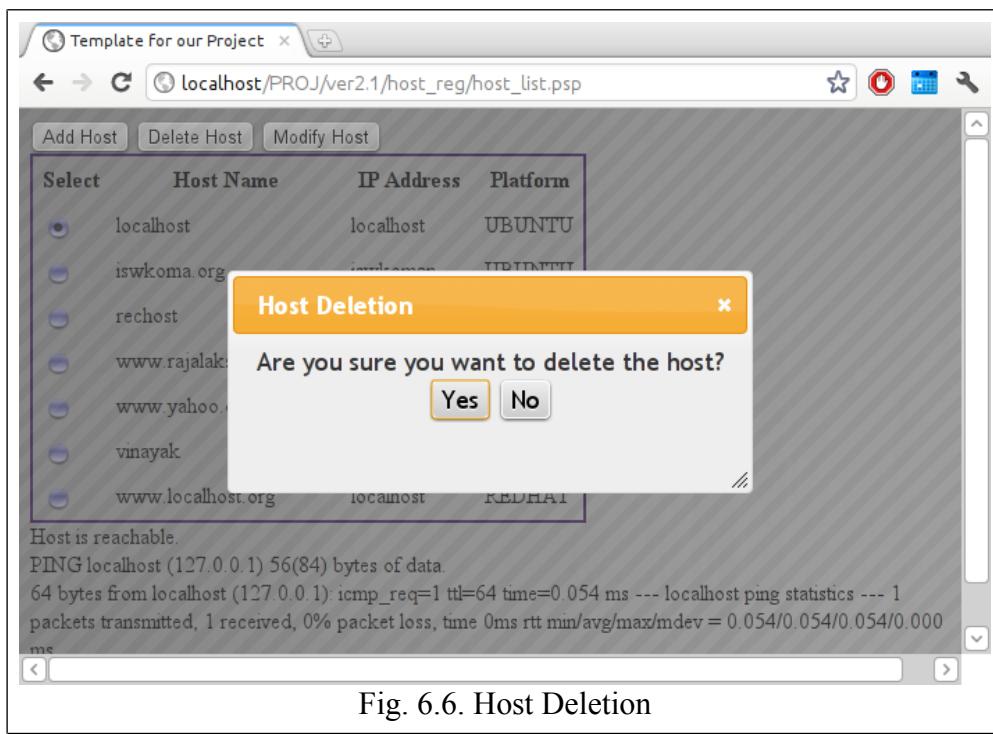


Fig. 6.6. Host Deletion

6.3.3. Host Modification

This allows the RHA to modify the IP address, host name, platform and the root password of that host. It does not update the root password on the RHS, but only on the HRS.

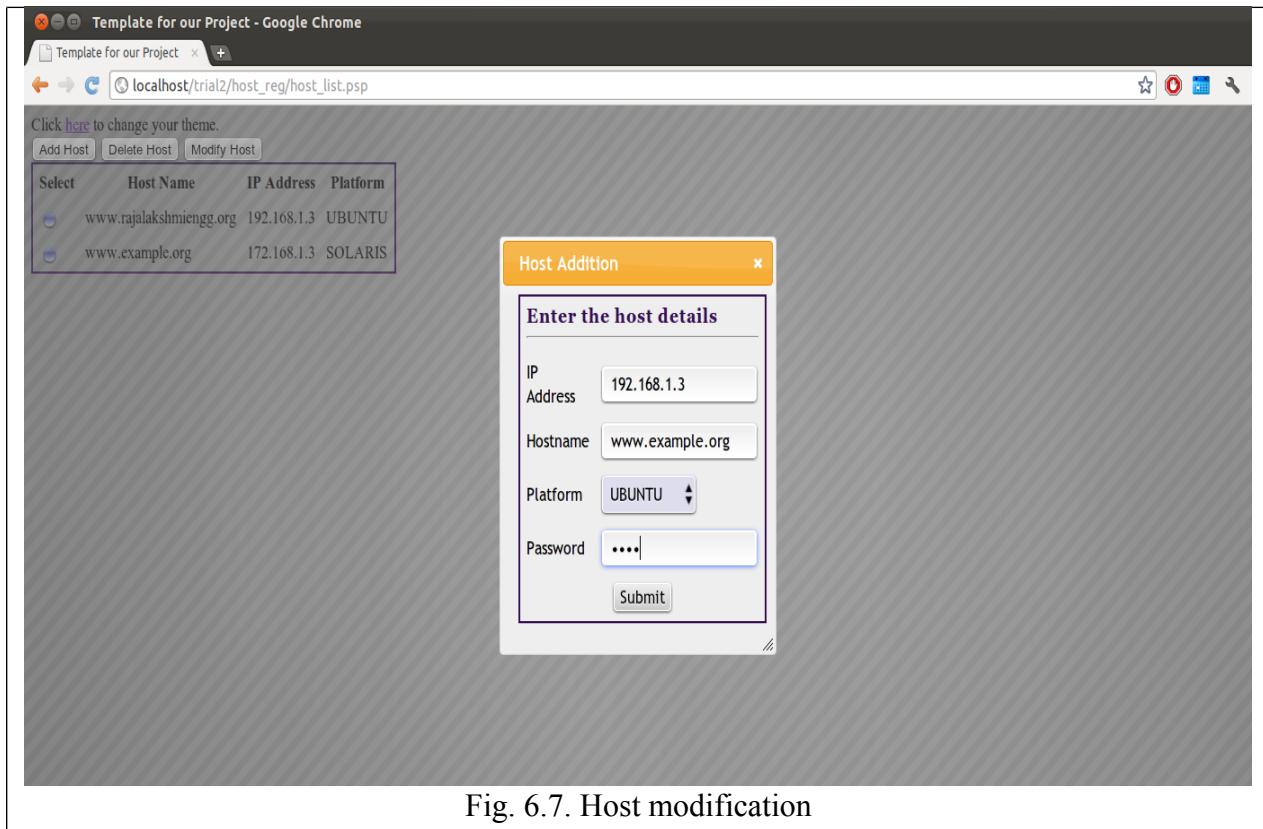


Fig. 6.7. Host modification

The image contains two side-by-side screenshots of a web browser. Both screenshots show a table of hosts with columns: Select, Host Name, IP Address, and Platform. In the left screenshot (a), the host "www.example.org" has an IP of 172.168.1.3 and a Platform of SOLARIS. In the right screenshot (b), after modification, the host "www.example.org" now has an IP of 202.168.1.3 and a Platform of REDHAT. The "Host Modification" dialog box is visible in both screenshots, showing the updated values: IP Address (202.168.1.3), Hostname (www.example.org), Platform (REDHAT), and Password (*****).

Fig. 6.8.(a) Before modification

Fig. 6.8.(b) After modification

6.4. Navigating to task execution

The RHA selects one of the hosts that he registers under himself. If the host is not reachable, he is confronted with a warning message. If the host is reachable, a hyperlink immediately appears. On clicking the hyperlink, the RHA navigates to the task panel.

A screenshot of a web browser window titled "Template For our Project". The address bar shows "localhost/PROJ/ver3.1/host_reg/host_list.psp". Below the address bar, there is a password save dialog: "Do you want Google Chrome to save your password? Never for this site Save password". The main content area has tabs: "Add Host", "Delete Host", and "Modify Host". A table lists hosts with columns: "Select", "Host Name", "IP Address", and "Platform". The hosts listed are:

Select	Host Name	IP Address	Platform
<input checked="" type="radio"/>	localhost	localhost	UBUNTU
<input type="radio"/>	iswkoma.org	iswkoman	UBUNTU
<input type="radio"/>	rehost	recip	UBUNTU
<input type="radio"/>	www.rajalakshmiengg.org	721.61.5.0	UBUNTU
<input type="radio"/>	www.yahoo.org	192.168.2.12	UBUNTU
<input type="radio"/>	vinayak	192.168.55.2	UBUNTU
<input type="radio"/>	www.localhost.org	localhost	REDHAT

Below the table, a message says "Host is reachable." followed by a ping command output:

```
PING localhost (127.0.0.1) 56(84) bytes of data.  
64 bytes from localhost (127.0.0.1): icmp_req=1 ttl=64 time=0.031 ms --- localhost ping statistics --- 1 packets transmitted, 1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.031/0.031/0.031/0.000 ms
```

Fig. 6.9. Host selection

6.4.1. Task Panel

The task panel provides the RHA with a common grid of icons that lets him perform system administration tasks graphically on the RHS. If the task is valid, it is available, else it is absent in the grid. The RHS clicks on any of the tasks to proceed with administration.

A screenshot of a web browser window titled "Template for our Project - Google Chrome". The address bar shows "localhost/ver2.1/task_mgt/table.psp". The main content area displays a grid of icons representing system management tasks. The icons are:

- Manage Computer
- Disk Management (highlighted)
- Power Resources
- Backup Management
- User Administration
- Process Monitor...
- Server Health A...
- Software Package...
- Network Time Pr...

At the bottom of the grid, there is a link: "localhost/ver2.1/task_mgt/disk_mgt/disks.psp".

Fig. 6.10. Task Panel

6.5. Software package management

The software package management module allows to install a new package, upgrade the package, clean and check the package.

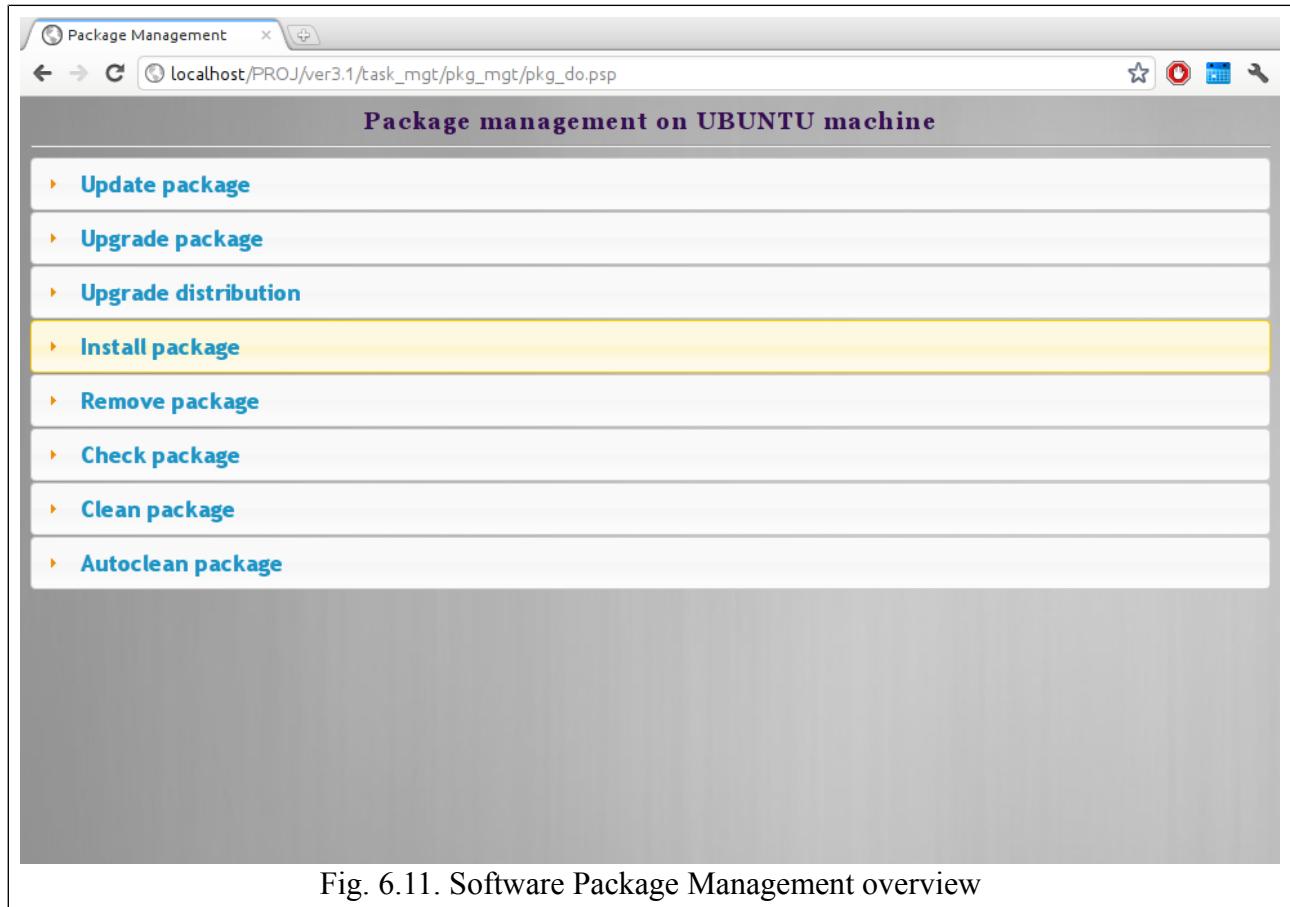


Fig. 6.11. Software Package Management overview

Installation of a particular package requires internet connection where it checks for the package and installs manually. The downloads only is used when it is directly downloaded through the internet, fix broken is used to break the dependencies, ignore missing is used when the particular options are ignored, no message is used to denote there is no message to be displayed, simulate option, yes is used to conform to install package that are huge. Check package is required to find whether the package is available and using the various options allows the package to be checked and it redirects back to same page after checking package details. Update package provides the various newer versions of the particular packages available and by selecting that option it gets updated and redirects back to same page.

6.5.1. Install package

To install a package, the RHA is required to provide the name of the package in the input box and choose appropriate options. Install package allows to install any new packages by giving the package name and by selecting the various options the packages are installed and it redirects back to same page.

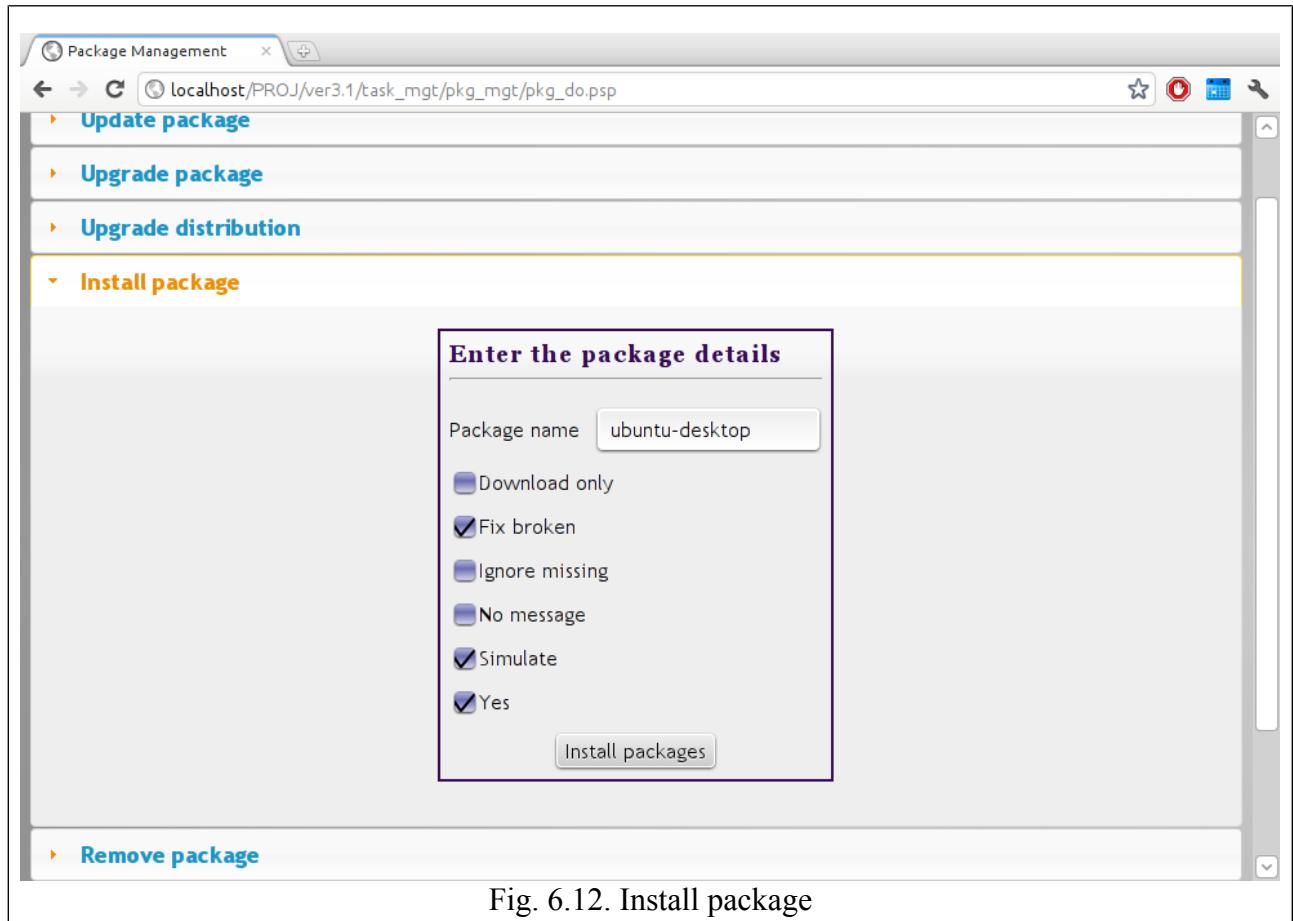


Fig. 6.12. Install package

Remove package is required to remove the package which has been installed by clicking on remove package button it is removed and redirects back to original page. Upgrade distribution is required to upgrade the whole system configuration like for instance a new version of Linux OS is available it directly upgrades. Upgrade Package provides with a newer version of the package and it will be upgraded automatically by selecting the option upgrade package and redirects back to original page.

6.6. Disk Management

The disk management tile provides the RHA local disk management and ZFS utilities. In the task management grid, the RHA selects Disk Management to navigate.



Fig. 6.13. Disk management tasks

Manage file systems is a suite that provides the RHA ways to configure partitions on the RHS. Its equivalent to a simple disk management utility that does not involve any logical volumes or RAID. Once the RHA navigates to the file system management, he is required to choose the physical drive. This page lists all valid physical drives that can be managed on the RHS. Once the RHA chooses the drive, he clicks toggle drive button. If he feels that any new drive was inserted in the RHS, he clicks the Refresh drives button.

6.6.1. Partition Listing

This page allows the RHA to perform partitioning on the physical disk that was chosen in the toggle drives page. The operations include toggling the boot flag, creating physical, extended and logical partitions, deleting partitions and creating a fresh partition map.

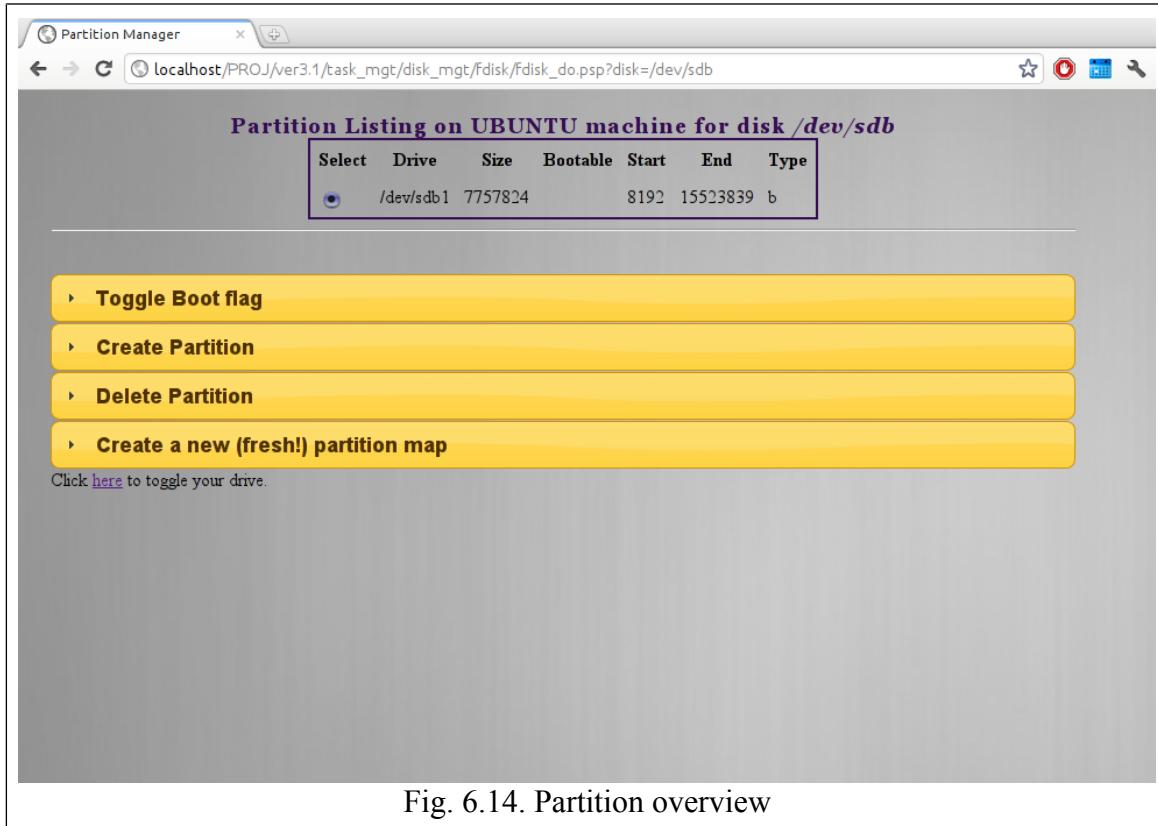


Fig. 6.14. Partition overview

Create partition allows to create new partitions on the physical drive. The RHA chooses the partition type between physical, extended and logical. He provides the start and end position of the partition and clicks the create button. If either positions are invalid, the position snaps to the immediate valid position. For instance, if the disk has reserved sectors till the 2nd MB, even if the RHA chooses 0MB as start position, the partition start-position gets snapped to 2MB. Such snaps occur without warning, so RHA is required to exercise caution.

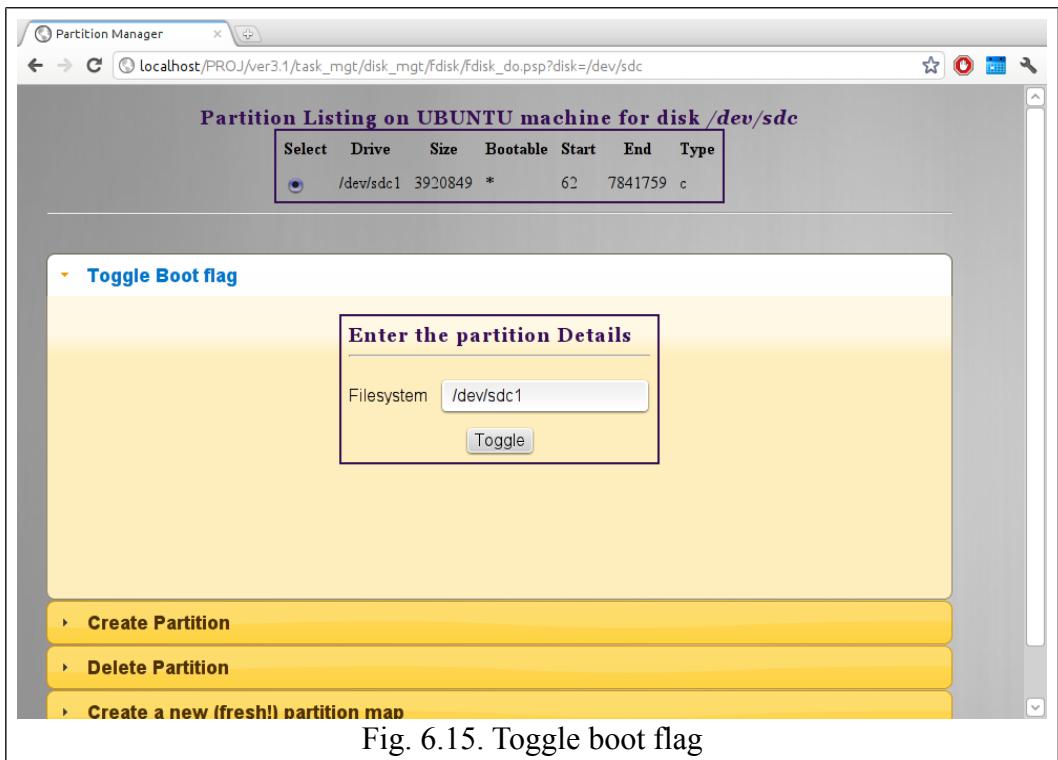


Fig. 6.15. Toggle boot flag

Toggle boot flag operation is used to toggle the boot flag of the disk. In delete partition the RHA selects one of the partitions on the disk and clicks the delete button to delete the partition. He is required to observe caution, to backup any critical data before exercising this step. Create new partition map operation is used to delete all the existing partitions on a disk and create new partitions from scratch. He should observe severe caution when exercising this step.

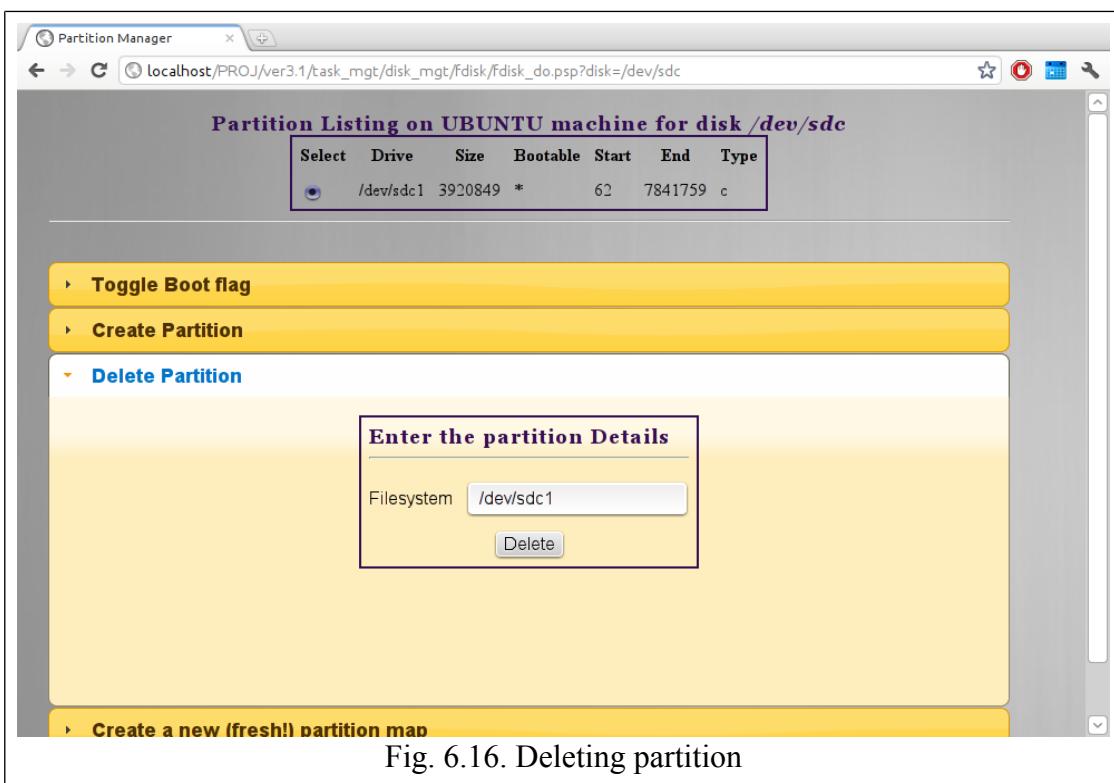


Fig. 6.16. Deleting partition

6.6.2. Zettabyte Filesystems Management Suite

The zettabyte file systems management suite consists of utilities to manage zettabyte pools and the zettabyte file systems themselves. It consists of a module called manage zpools that is used to manage the logical volumes, manage zfs that is used to manage the individual filesystems within the logical volumes and a module to manage time slider.

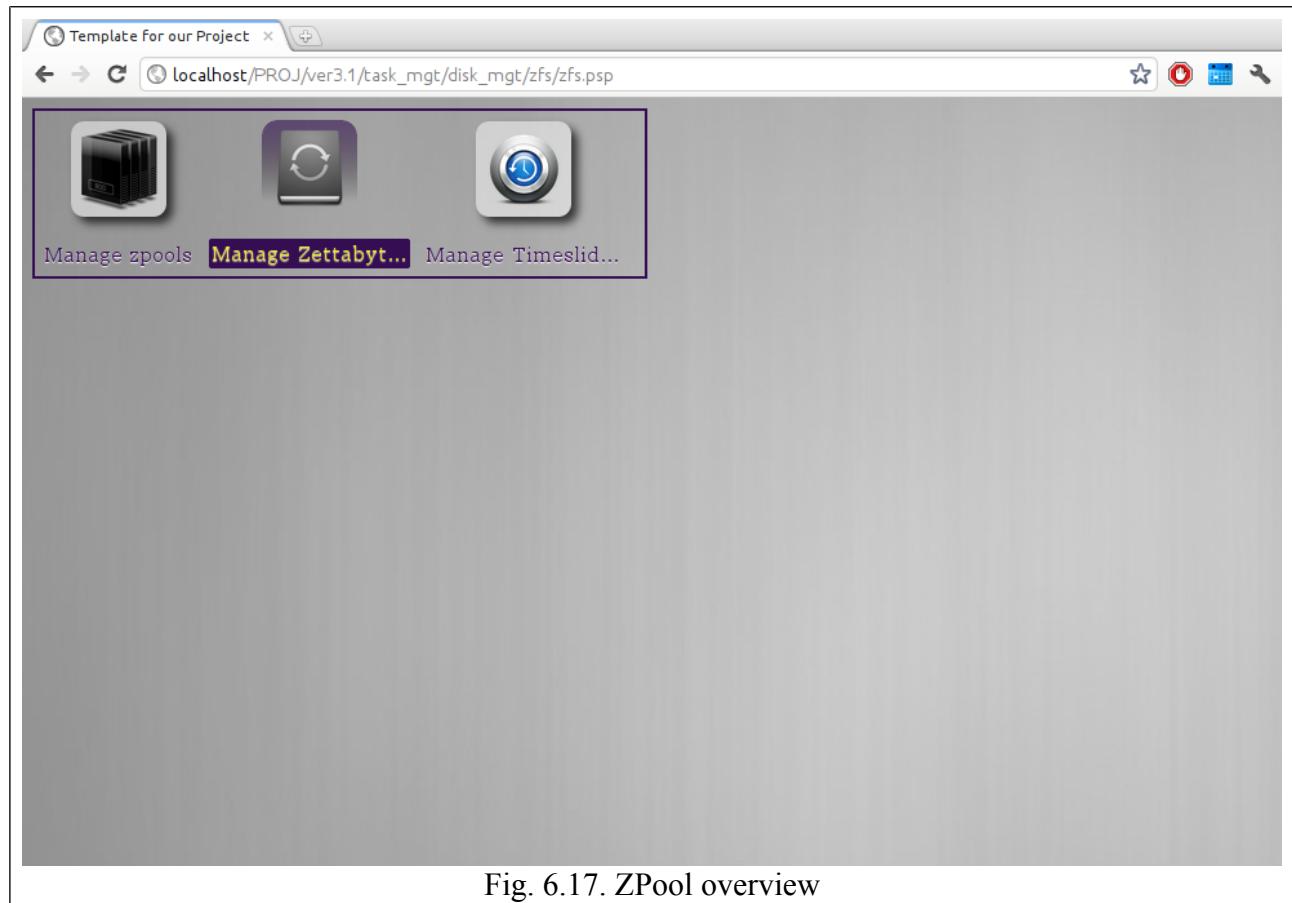


Fig. 6.17. ZPool overview

6.6.2.1. Zettabyte Pool Management

Zettabyte Pool Management allows the RHA to administer zpool in an RHS that has zfs or zfs-fuse installed. The administration tools include pool creation, attachment, detachment, online, offline, export, import, statistics, IO Statistics, history, error management, scrubbing, validation and destruction.

The screenshot shows a web-based ZPool management interface. At the top, a navigation bar includes a logo, the text "Zettabyte Pool Manager", and a URL "localhost/PROJ/ver3.1/task_mgt/disk_mgt/zfs/zpool/zpool_do.psp". Below the navigation is a title "Pool Listing on UBUNTU machine" followed by a table showing two pools:

Select	Pool Name	Size	Allocation	Free Space	Used Space	Health status
<input type="radio"/>	g62	48.8G	41.6G	7.19G	85%	ONLINE
<input checked="" type="radio"/>	sample	121M	117K	121M	0%	ONLINE

To the right of the table is a vertical sidebar with a list of management tasks:

- > Export pool
- > Attach disk
- > Replace disk
- > Upgrade
- > Detach disk
- > Check for pool errors
- > Validate pool data
- > Provide IO statistics
- > Provide pool statistics
- > Display ZPool History

Fig. 6.18. ZPool management tasks

6.6.2.2. IO Statistics

In IO statistics, the RHA selects one of the pools and chooses provide IO statistics. He then clicks the button and the statistics are displayed in the page. If no pool is selected, then general IO statistics pertaining to the RHS are displayed.

This screenshot shows the same ZPool management interface as Fig. 6.18, but with a different focus. The "Provide IO statistics" option in the sidebar is now highlighted with a yellow background. A modal dialog box titled "Enter the ZPool details" is open in the foreground, containing a "Poolname" input field with "sample" typed into it and a "Provide Input/Output Statistics" button below it.

Fig. 6.19. ZPool IOStat

6.7. User Management

The screenshot shows a web-based user management interface titled "User Management". The URL in the address bar is "localhost/PROJ/ver3.1/task_mgt/usr_mgt/usr_list.psp". The interface includes a header with buttons for "Add User", "Delete Single User", "Delete Multiple User", and "Modify User". Below the header is a table listing 15 system users. The columns are: Select, User Name, Comment, Directory, Shell, and Groups. The data is as follows:

Select	User Name	Comment	Directory	Shell	Groups
root	root	root	/root	/bin/bash	[]
daemon	daemon	daemon	/usr/sbin	/bin/sh	[]
bin	bin	bin	/bin	/bin/sh	[]
sys	sys	sys	/dev	/bin/sh	[]
sync	sync	sync	/bin	/bin/sync	[]
games	games	games	/usr/games	/bin/sh	[]
man	man	man	/var/cache/man	/bin/sh	[]
lp	lp	lp	/var/spool/lpd	/bin/sh	[]
mail	mail	mail	/var/mail	/bin/sh	[]
news	news	news	/var/spool/news	/bin/sh	[]
uucp	uucp	uucp	/var/spool/uucp	/bin/sh	[]
proxy	proxy	proxy	/bin	/bin/sh	[]
www-data	www-data	www-data	/var/www	/bin/sh	['admin']
backup	backup	backup	/var/backups	/bin/sh	[]
list	list	Mailing List Manager	/var/list	/bin/sh	[]
irc	irc	ircd	/var/run/ircd	/bin/sh	[]

Fig. 6.20. User management overview

6.7.1. Add user

The add user button, once clicked, opens a modal window. The RHA is required to enter the username, comment, directory, shell, group, limit and choose to create a home directory. Comment describes the role of the user. Directory specifies the home directory path for the user, shell is the default login shell of the user and group specifies the system group of the user. The limit values are used to create multiple users ranging from the two input limit values. For instance, if username is given as *usr* and limits are 0 to 2, the users created are:

1. usr0
2. usr1
3. usr2

The properties of all the created users will be same.

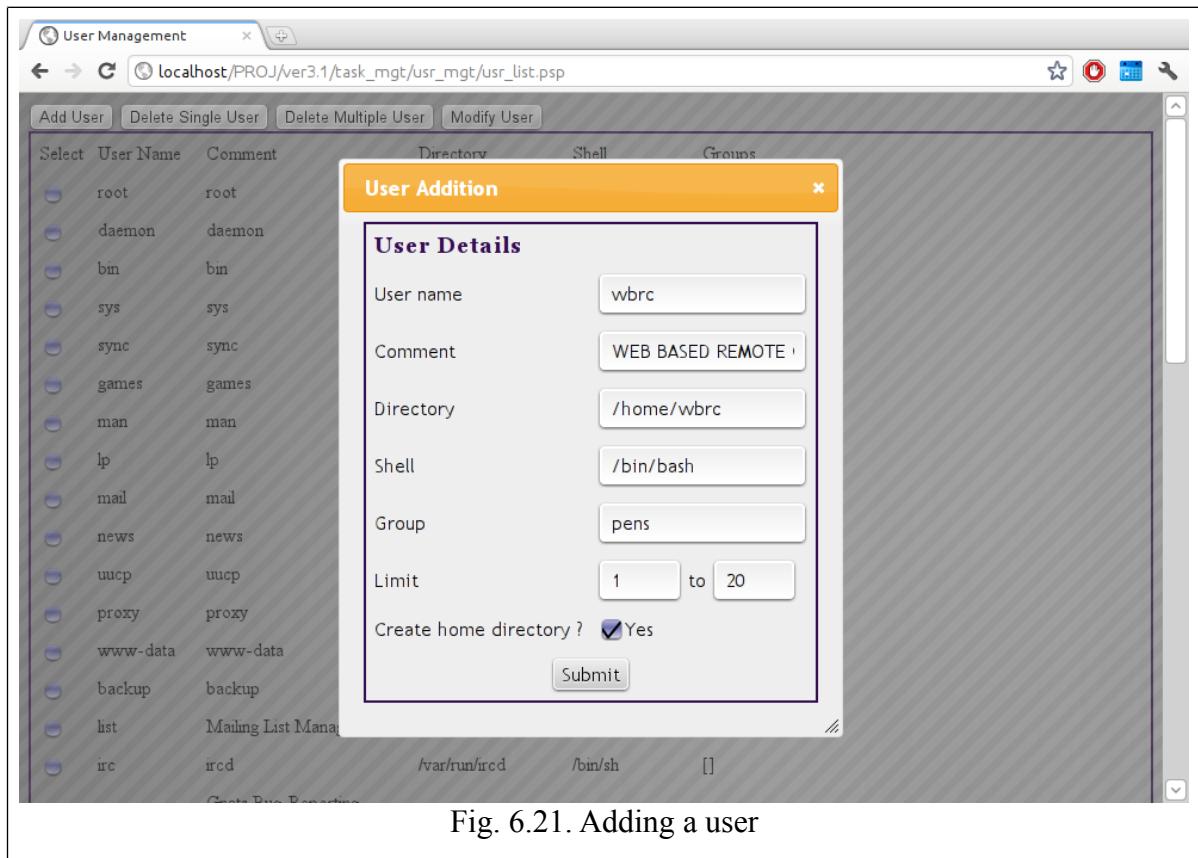


Fig. 6.21. Adding a user

6.7.2. Modify user properties

Clicking the modify user button will open a modal window to modify a user. One selected user can be modified and the comment, directory, shell, group can be modified. Since the username is the unique way to identify a user, the interface does not allow to modify the same.

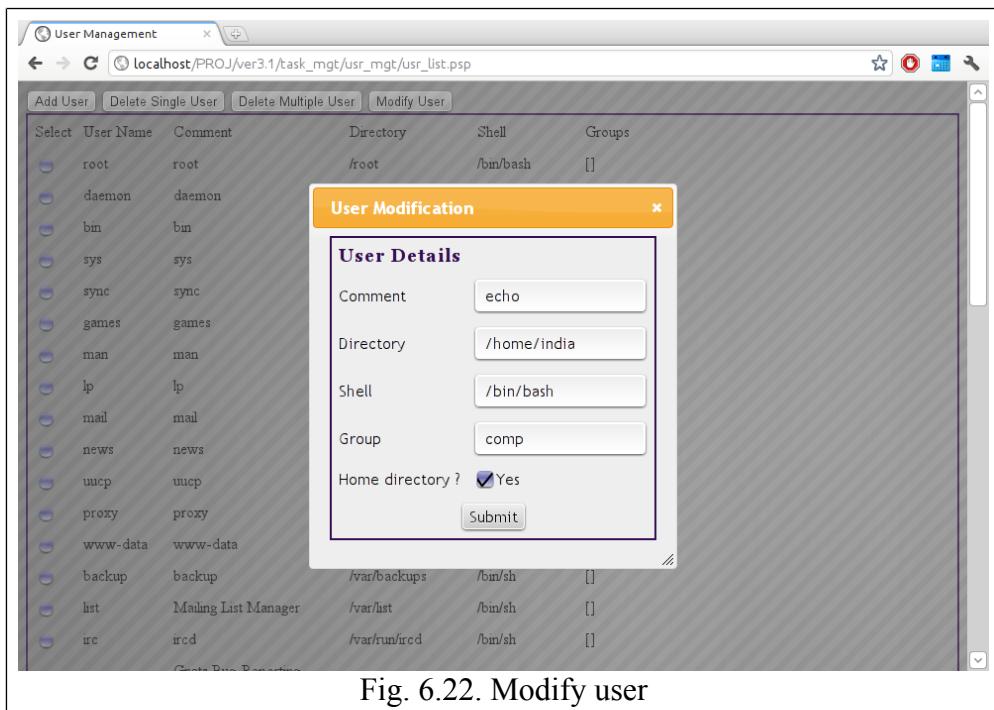
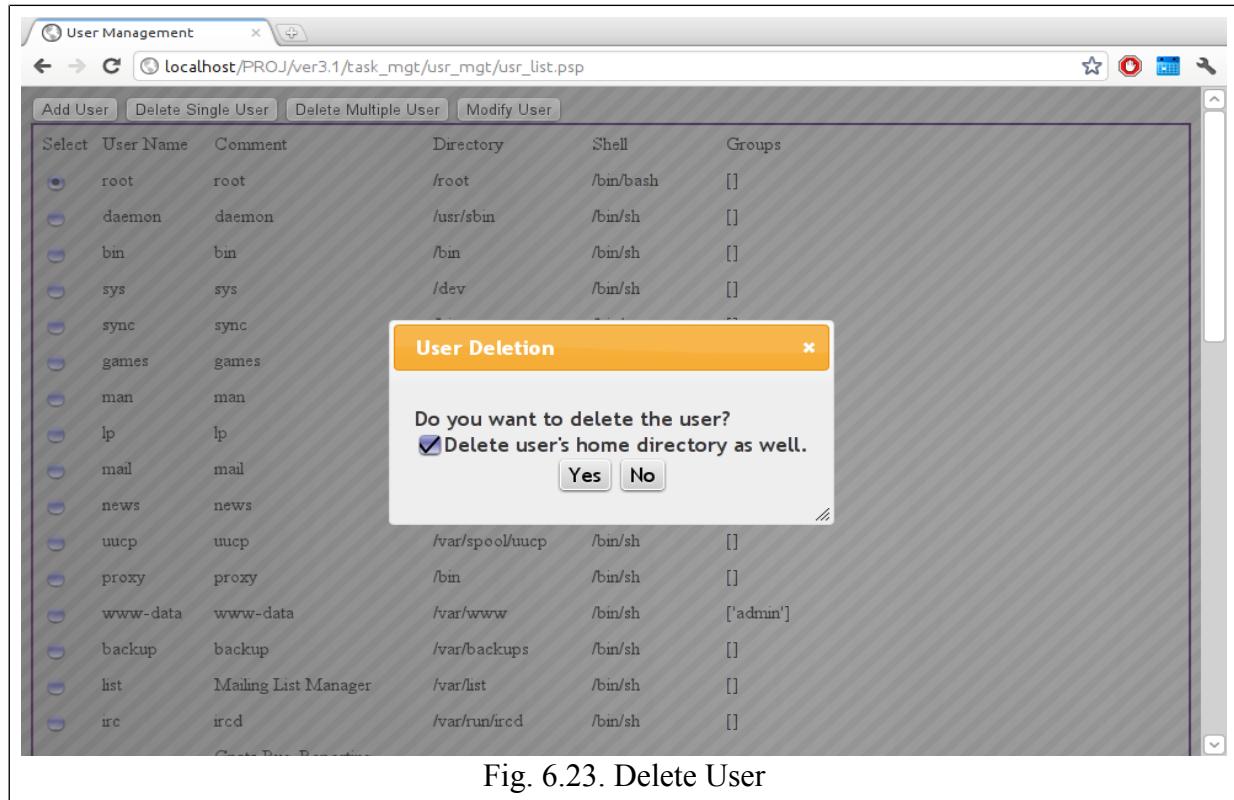


Fig. 6.22. Modify user

6.7.3. Delete user

The modal window to delete a user presents a simple interface. If the administrator wishes to make up his mind again, he can simply choose no.

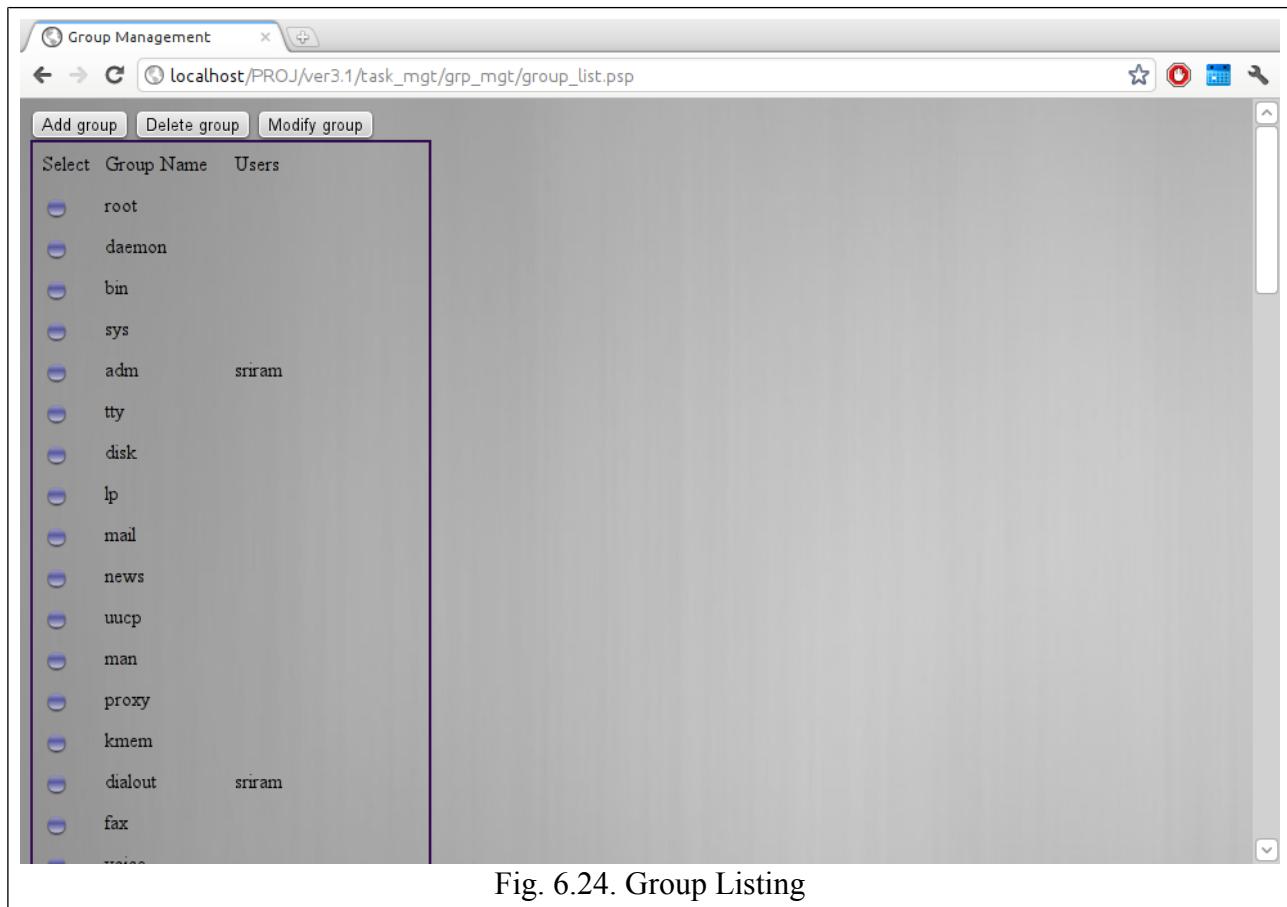


6.8. Group Management

Users in the RHS are divided into groups for the purposes of easier administration and those groups can have different levels of privileges. For example, the RHA may have a group of users who work in the accounting department. In such an environment he may wish to create an accounts group and assign all the Accounting department personnel to that group.

6.8.1. List Groups

Opening the groups tile from the task management console takes you to the group listing page. Here, all the groups in the RHS that was selected in the host listing page, are displayed. The listing is ordered chronologically.



The screenshot shows a web-based Group Management interface. The title bar says "Group Management". The URL in the address bar is "localhost/PROJ/ver3.1/task_mgt/grp_mgt/group_list.psp". Below the address bar are three buttons: "Add group", "Delete group", and "Modify group". A horizontal menu bar contains "Select", "Group Name", and "Users". The main content area displays a list of groups. Each group entry consists of a small blue circular icon followed by the group name. Some entries also show a user name next to the group name. The groups listed are: root, daemon, bin, sys, adm (with user sriram), tty, disk, lp, mail, news, uucp, man, proxy, kmem, dialout (with user sriram), fax, and voice.

Fig. 6.24. Group Listing

6.8.2. Add group

In the group listing page, click the Add group button. The RHA is required to provide a mandatory group name field, and an optional group ID. Leaving this optional field blank, is the typical case, and in which case the RHS automatically allot a sequential group ID (GID). If the RHA wishes to override it, he provides the GID. This should be exercised with caution. However, the safe design of the system ensures that conflicting GIDs prevents addition of the group, thus bringing the overall system to a stable state.

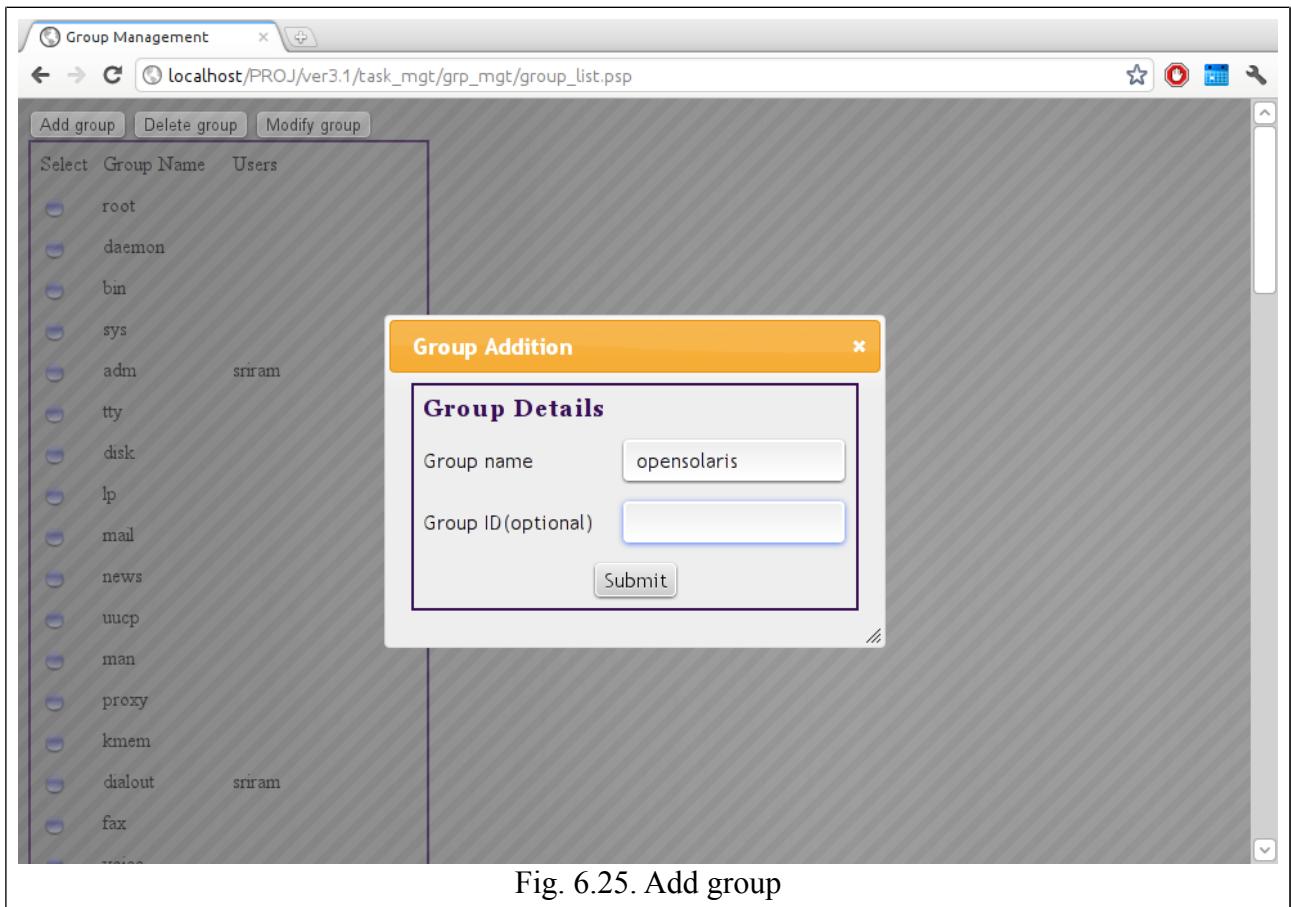


Fig. 6.25. Add group

6.8.3. Modify group

For modification of a group, the RHA selects the group via the radio button and click the Modify group button. This prompts the RHA to give the new group name. This is dynamic- it ensures that the users who were under the old group name will come under the same group and benefits, with just the group name updated.

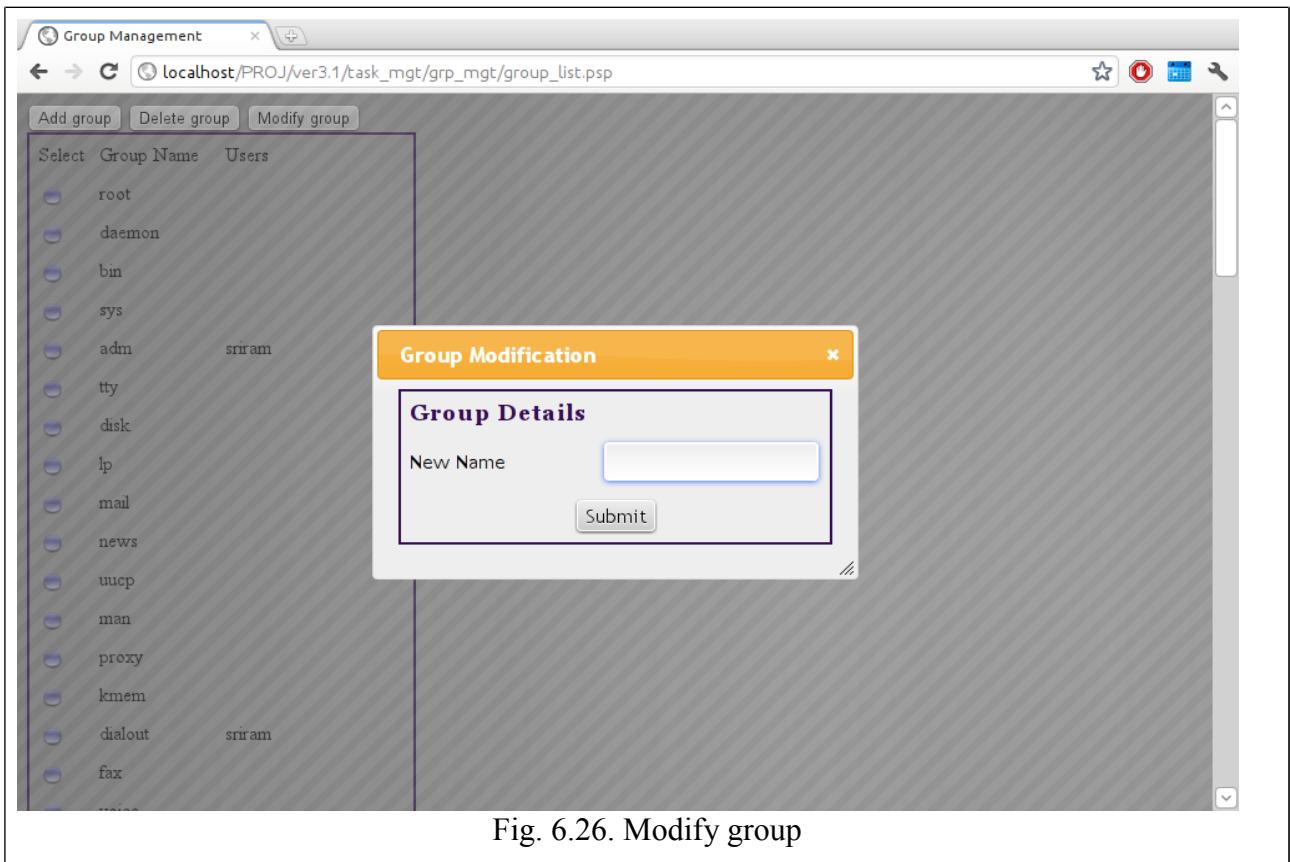


Fig. 6.26. Modify group

6.8.4. Delete group

The RHA selects the group to delete a group and presses the delete group button. He confirms the prompt in the modal window and the group gets deleted in the RHS.

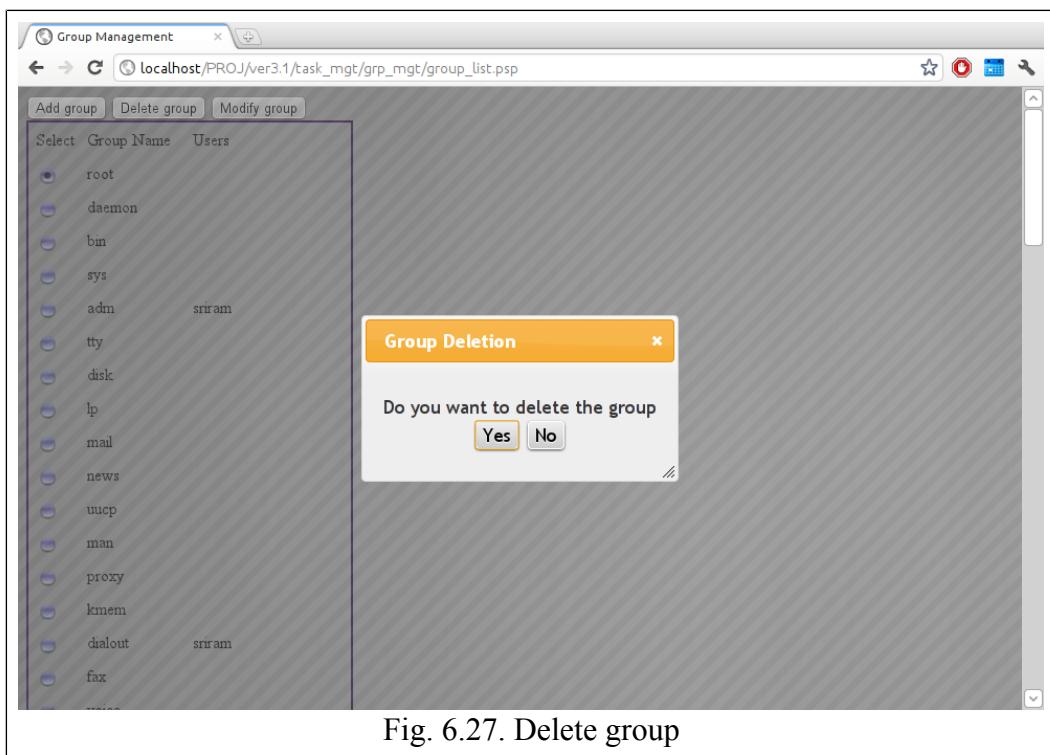


Fig. 6.27. Delete group

6.9. Automating Tasks

This task set consists of cron and anacron, both of which are used to schedule one or more commands in a Unix or Linux operating system. They differ mainly in the environment where they are used- cron offers finer granularity in the time specification than anacron. However, anacron has the advantage that the scheduled command will execute in the RHS at the next boot, even if it was not in powered on state when the command was scheduled to run.

6.9.1. Cron Management

A crontab file has five fields for specifying day, date and time followed by the command to be run at that interval. Command field is what you want run in the RHS. This field may contain multiple words or spaces. Minute controls what minute of the hour the command will run on, and is between '0' and '59'.Hour field controls what hour the command will run on, and is specified in the 24 hour clock, values must be between 0 and 23 (0 is midnight). Day of month field is what you want the command run on, for example to run a command on the 19th of each month, the value would be 19. Month specifies which month the command will run on, it should be specified numerically.

The screenshot shows a web browser window titled "Cron Management". The URL in the address bar is "localhost/ver3.1/task_mgt/automated_tasks/crontab.psp". The main content area is a form titled "Task Details" with the following fields:

Command	apt-get update
Minute(0-59)	8
Hour(0-23)	22
Day of Month(1-31)	28
Month(1-12)	6
Day of Week(0-6)	5

At the bottom of the form is a "Submit" button.

Fig. 6.28. Cron management

6.9.2. Anacron Management

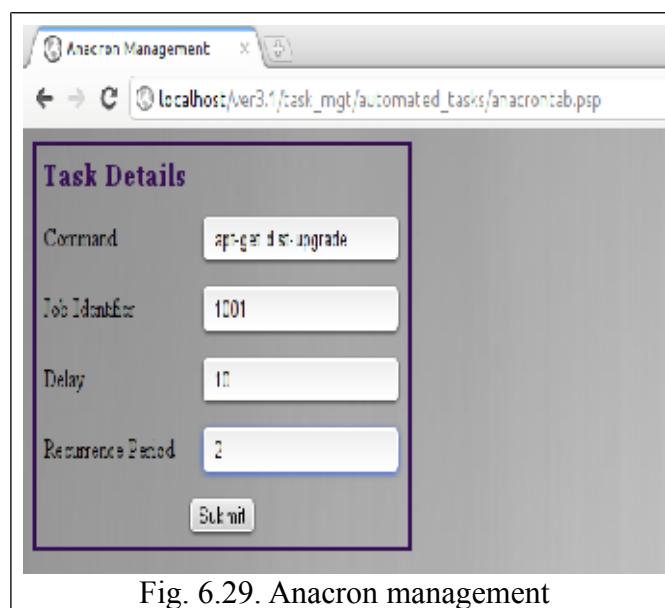
Anacron is the cron for desktops and laptops. Anacron does not expect the system to be running 24 x 7 like a server. When you want a background job to be executed automatically on a machine that is not running 24 x 7, you should use anacron. Command is the shell script that needs to be executed. Job identifier is the name for the job's time stamp file. It should be unique for each job. This will be available as a file under the /var/spool/anacron directory. This file will contain a single line that indicates the last time when this job was executed. Delay indicates the delay in minutes. i.e. the number of minutes anacron should wait before executing the job. Recurrence period is a numeric value that specifies the number of days.

1 – daily.

7 – weekly.

30 – monthly.

N – This can be any numeric value. N indicates number of days.



CHAPTER 7

TESTING

7.1. Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results. Units tests are written from a programmers perspective. In most cases, front end validation ensures that no error creeps into the remote host. The scope of error in the system is limited to network snags.

7.1.1. Host Listing

The host listing page allows the remote host administrator to view host information and double click available hosts to administer them. This page shows a specimen list of hosts. The hosts that are reachable would have a green label, whereas the unreachable hosts would have a red label with a solid border, indicating its unavailability.

The screenshot shows a web browser window titled "Template for our Project" displaying a host listing page. The URL in the address bar is "localhost/ver5.0/host_reg/host_list.psp". The page features a grid of host icons and their names:

Host Name
localhost
guruji.rocks
rechost
VirtualRedhat
VirtualUbuntu
www.solaris.org
college

On the right side of the grid, there is a "Host Information" panel for the host "www.solaris.org". The panel displays the following details:

Host name : www.solaris.org
IP Address : 192.168.56.12
Platform : SOLARIS
Available : no

Below the host listing, there are buttons for "Add Host", "Delete Host", and "Modify Host".

Fig. 7.1. Host listing with reachability status report

Project Name	Web Based Remote Console		
Test Case	1	Test type	Unit Test
Test description		Host reachability	Version
Test Environment		Ubuntu Linux 11.10 32 bit	
S.No.	Input	Test Result	Action
1.	Remote host disconnected from Host Registration Server	Invalid	Host unavailable
2.	Remote host connected to Host Registration Server	Valid	Host available

Table 7.1. Test Case for host reachability

7.1.2. Package Installation

In the package installation module, when the exclusive lock is available in the remote host server, the package installs successfully. This page shows the screen when a package was attempted to be installed with an active internet connection on the RHS and with no other package installation instance running.

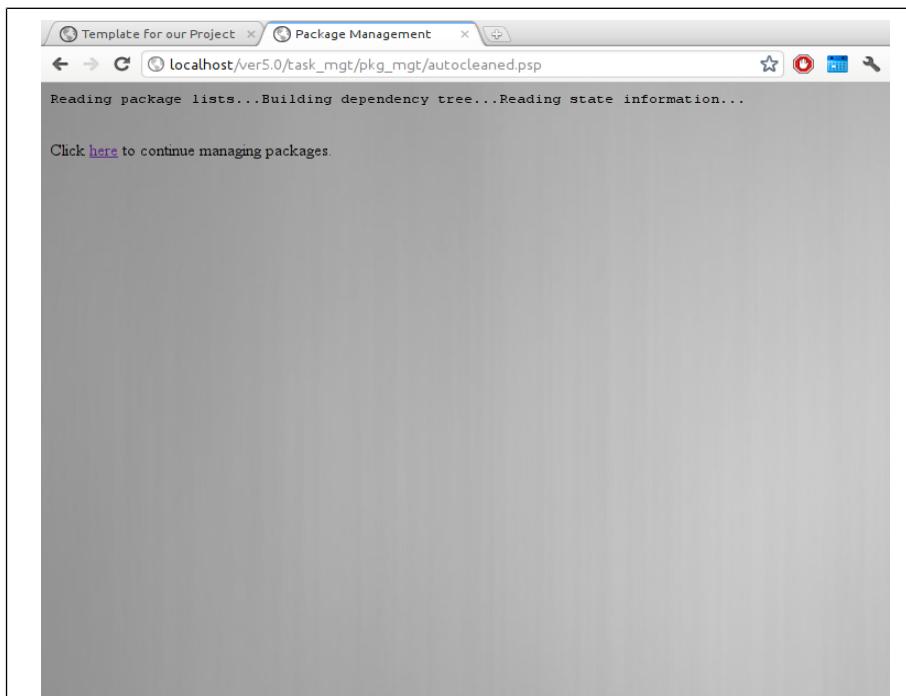


Fig. 7.2. Successful package installation

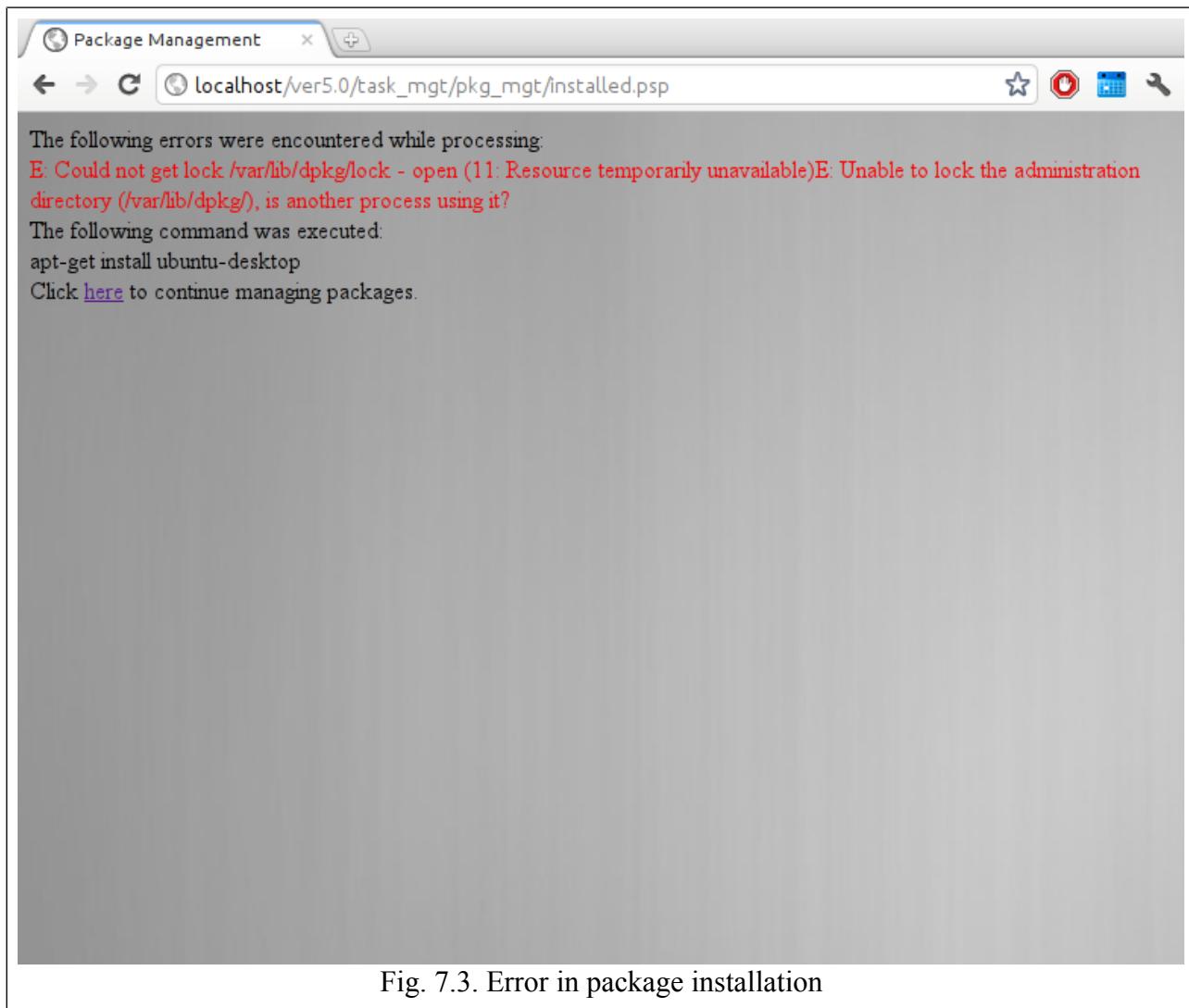


Fig. 7.3. Error in package installation

Project Name	Web Based Remote Console		
Test Case	2	Test type	Unit Test
Test description	Pkg availability	Test Unit	Package Manager
Test Environment	Ubuntu Linux 11.10 32 bit		
S.No.	Input	Test Result	Action
1.	Another package installation running	Invalid	Package lock unavailable
2.	Unique package instance running	Valid	Package installed

Table 7.2. Test case for package availability

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

8.1. Conclusion

In this project we provide a system that allows to administer, configure and execute tasks in remote systems in a unified and intuitive manner. The GUI allows the user to directly administer the system tasks, so that he does not have to remember too many commands. The system aims to remove the complexity from system administration.

Thus, the Web Based Remote Console has been implemented according to the design of the project. It was also tested against the test cases planned for acceptance of this project as a proof of concept.

8.2. Future Enhancements

The Web Based Remote Console is used to administer in remote systems. This can be further improved by providing support for more platforms and improving the efficacy by using bleeding edge technology such as multi-threading.

APPENDIX A

A.1. Setup

A.1.1. Installation Procedure

Install the following packages in your Host Registration Server (HRS):

mod_python from Ubuntu Software Package Manager- MySQLdb module for database connectivity, configure /etc/apache2/apache2.conf Add the PSP handler in the file as seen in the screenshots.

A.1.2. Installation screen shots

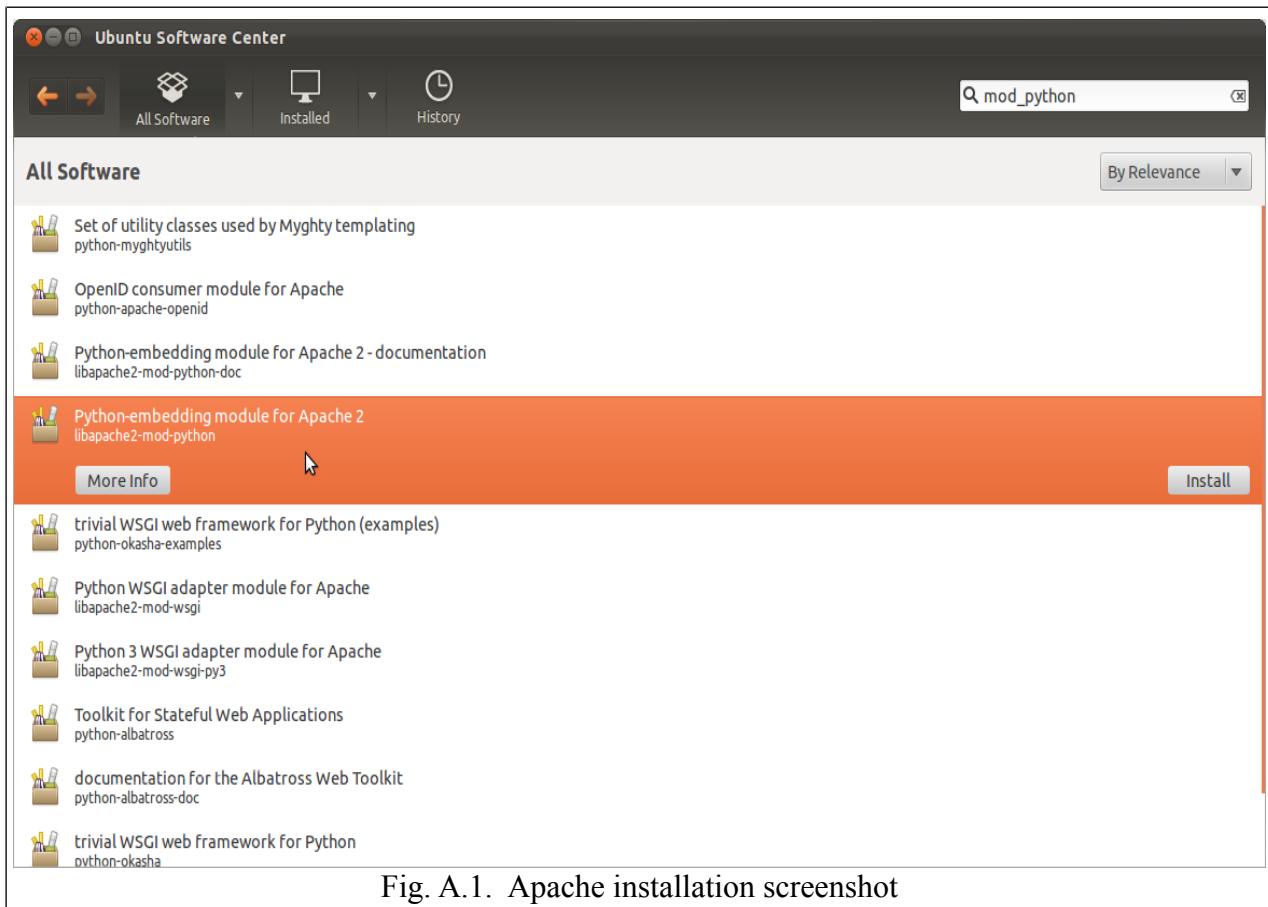


Fig. A.1. Apache installation screenshot

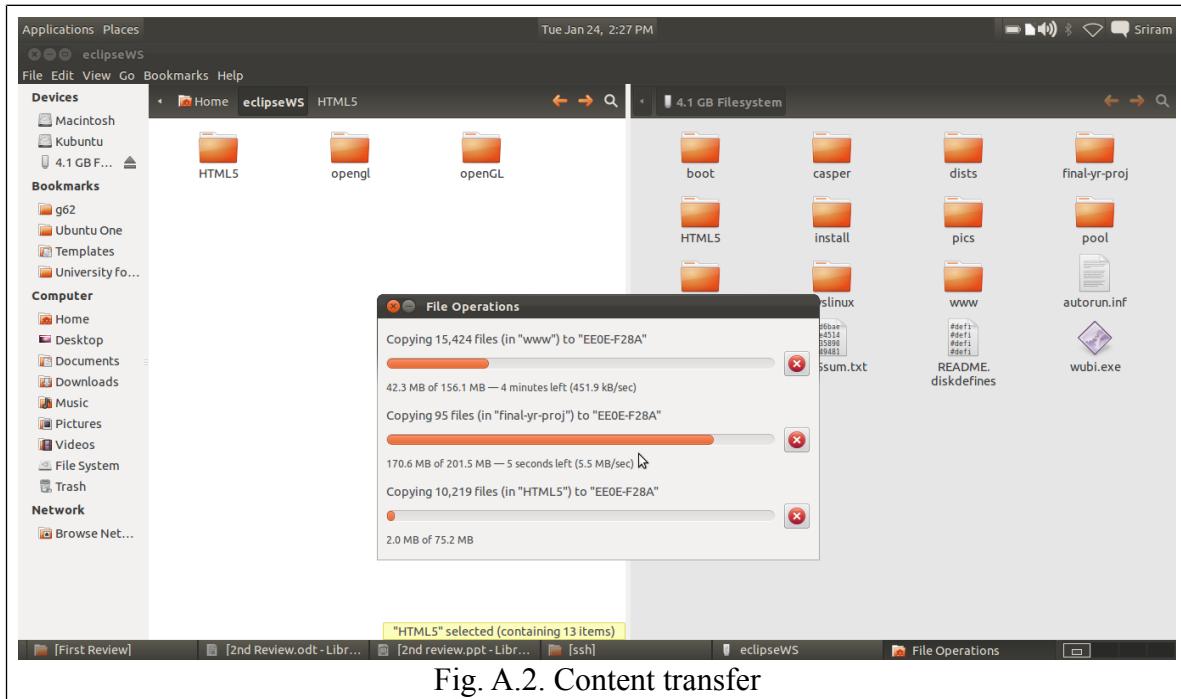


Fig. A.2. Content transfer

A.1.3. Apache Configuration File

```

#
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See http://httpd.apache.org/docs/2.2/ for detailed information
LockFile ${APACHE_LOCK_DIR}/accept.lock

PidFile ${APACHE_PID_FILE}
# Timeout: The number of seconds before receives and sends time out.
Timeout 300
# KeepAlive: Whether or not to allow persistent connections (more than one request per
connection). Set to "Off" to deactivate.
#
KeepAlive On
#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection.

#
MaxKeepAliveRequests 100
#

```

```

# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5
##
## Server-Pool Size Regulation (MPM specific)
##
# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_prefork_module>
    StartServers      5
    MinSpareServers  5
    MaxSpareServers 10
    MaxClients       150
    MaxRequestsPerChild 0
</IfModule>
<IfModule mpm_worker_module>
    StartServers      2
    MinSpareThreads  25
    MaxSpareThreads  75
    ThreadLimit      64
    ThreadsPerChild  25
    MaxClients       150
    MaxRequestsPerChild 0
</IfModule>
<IfModule mpm_event_module>
    StartServers      2
    MinSpareThreads  25
    MaxSpareThreads  75
    ThreadLimit      64

```

```
ThreadsPerChild    25
MaxClients        150
MaxRequestsPerChild 0
</IfModule>
# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
#
# AccessFileName: The name of the file to look for in each directory
# for additional configuration directives. See also the AllowOverride
# directive.
AccessFileName .htaccess
#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ~ "^\\.ht">
    Order allow,deny
    Deny from all
    Satisfy all
</Files>
DefaultType text/plain
#
HostnameLookups Off
# Include module configuration:
Include mods-enabled/*.load
Include mods-enabled/*.conf
# Include all the user configurations:
Include httpd.conf
# Include ports listing
Include ports.conf
#
# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
```

```
# If you are behind a reverse proxy, you might want to change %h into %{X-Forwarded-For}i  
#Modified by ITB11P04  
LoadModule python_module /usr/lib/apache2/modules/mod_python.so  
AddHandler mod_python .psp .psp_  
PythonHandler mod_python.psp  
PythonPath "[/var/www/modules'] + sys.path"  
PythonDebug On
```

APPENDIX B

B.1. Setting up cron jobs in Unix

Cron is a unix utility that allows tasks to be automatically run in the background at regular intervals by the cron daemon. These tasks are often termed as cron jobs in unix , solaris. Crontab (CRON TABLE) is a file which contains the schedule of cron entries to be run and at specified times.

This document covers following aspects of Unix cron jobs-

1. Crontab Restrictions
2. Crontab Commands
3. Crontab file – syntax
4. Crontab Example
5. Crontab Environment
6. Disable Email
7. Generate log file for crontab activity

1. Crontab Restrictions

You can execute crontab if your name appears in the file /usr/lib/cron/cron.allow. If that file does not exist, you can use crontab if your name does not appear in the file /usr/lib/cron/cron.deny. If only cron.deny exists and is empty, all users can use crontab. If neither file exists, only the root user can use crontab. The allow/deny files consist of one user name per line.

2. Crontab Commands

export EDITOR=vi ;to specify a editor to open crontab file.

crontab -e Edit your crontab file, or create one if it doesn't already exist.

crontab -l Display your crontab file.

crontab -r Remove your crontab file.

crontab -v Display the last time you edited your crontab file. (This option is only available on a few systems.)

3. Crontab file

Crontab syntax : A crontab file has five fields for specifying day , date and time followed by the command to be run at that interval.

* * * * * command to be executed

- - - - -
| | | | |
| | | | +---- day of week (0 - 6) (Sunday=0)
| | | | +---- month (1 - 12)
| | | +----- day of month (1 - 31)
| | +----- hour (0 - 23)
+----- min (0 - 59)

* in the value field above means all legal values as in braces for that column. The value column can have a * or a list of elements separated by commas. An element is either a number in the ranges shown above or two numbers in the range separated by a hyphen (meaning an inclusive range).

Note that repeat pattern like /2 for every 2 minutes or /10 for every 10 minutes is not supported by all operating systems. If you try to use it and crontab complains it is probably not supported. The specification of days can be made in two fields: month day and weekday. If both are specified in an entry, they are cumulative meaning both of the entries will get executed .

4. Crontab Example

A line in crontab file like below removes the tmp files from /home/someuser/tmp each day at 6:30 PM.

30 18 * * * rm /home/someuser/tmp/*

Changing the parameter values as below will cause this command to run at different time schedule below :

min	hour	day/month	month day/week	Execution time
30	0 1	1,6,12 *	- 00:30 Hrs on 1st of Jan, June & Dec.	
0	20 *	10 1-5	-8.00 PM every weekday (Mon-Fri) only in Oct.	
0	0 1,10,15	*	*	- midnight on 1st ,10th & 15th of month
5,10	0 10	*	1	- At 12.05,12.10 every Monday & on 10th of every month.

Note that if you inadvertently enter the crontab command with no argument(s), do not attempt to get out with Control-d. This removes all entries in your crontab file. Instead, exit with Control-c.

REFERENCES

1. Brian Down (2010) “Management of Systems and Services Made Simple with the Oracle Solaris Service Management Facility, Oracle Inc”
2. Daniel Price (2004) “Solaris Zones, Sun Microsystems Inc”
3. Dr Paul A. Walters (2005) “Solaris 10 Complete reference”
4. Michael Hasenstein (2001) “SuSE LVM Management Paper, SuSE Inc”
5. SuSE Linux unleashed by Mike McCallister