DWIT COLLEGE

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Institute of Science and Technology

SERVER MONITORING SYSTEM

A PROJECT REPORT

Submitted to

Department of Computer Science and Information Technology

DWIT College

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Submitted by

Bimal Gaire

August, 2016

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by BIMAL GAIRE entitled **"SERVER MONITORING SYSTEM"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

…………………………………………

Rituraj Lamsal

Lecturer

Deerwalk Institute of Technology

DWIT College

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

LETTER OF APPROVAL

This is to certify that this project prepared by BIMAL GAIRE entitled **“SERVER MONITORING SYSTEM”** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

…………………………………… Rituraj Lamsal [Supervisor] Lecturer DWIT College	………………………………………… …Hitesh Karki Chief Academic Officer DWIT College
………………………………………….. Jagdish Bhatta [External Examiner] IOST, Tribhuvan University	………………………………………….. Sarbin Sayami [Internal Examiner] Assistant Professor IOST, Tribhuvan University

ACKNOWLEDGEMENT

First and foremost, I have to thank my project supervisors, Mr. Ritu Raj Lamsal and Mr. Sarbin Sayami. Without their assistance and dedicated involvement in every step throughout the process, this project would have never been accomplished. I would like to thank them very much for their support and understanding over the period of this project development.

I would also like to show my gratitude to Mr. Hitesh Karki for giving me the confidence and motivating me for the project development as well as for his overall support during the period of last three and half years.

I am also grateful to my teachers, Mr. Dinesh Amatya, Mr. Ramesh Maharjan, Mr. Dhiraj Kedar Pandey for their constant support and guidance throughout my learning process in DWIT.

Most importantly, none of this could have happened without my family, friends and the whole DWIT family.

Bimal Gaire
TU Exam Roll no: 1798/069

STUDENT’S DECLARATION

I hereby declare that I am the only author of this work and that no sources other than listed here have been used in this work.

... ...

Bimal Gaire

Date: August, 2016

ABSTRACT

Network Monitoring Systems are essential in running the complex computer networks at present. They ensure all faults on the network are known and assist the network operator in fixing these faults. This project aims to plan and implement a system designed to monitor a modern network and includes all tools in a single extendable system. By using today's technologies and making good design decisions, this project aims to build a unified and fresh network monitoring system that is easy to configure, maintain and use, streamlining the work flow of a network operator. The network administrator needs to know the various resource usages of the server computer. The administrator can see the resource usages such as RAM, CPU and Disk as well as the IP address of the server. This application can notify if any problems occur in the system (server failure). Network administrator can receive the alerts if the resource usages exceed the threshold value. The administrator can get the time-series graph of the resource usage of the server computer.

Keywords: Network Monitoring, resource usage, server failure, alerts, time-series graph

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SMS:	Server Monitoring System
NMS:	Network Monitoring System
ISP:	Internet Service Provider
SNMP:	Simple Network Management Protocol
CPU:	Central Processing Unit
RAM:	Random Access Memory
PC:	Personal Computer
HTTP:	Hypertext Transfer Protocol
HTML:	Hypertext Markup Language
OS:	Operating System
IP:	Internet Protocol
PHP:	Hypertext Preprocessor
CSS:	Cascading Style Sheet
SQL:	Structured Query Language
RRD:	Round Robin Database
ICMP:	Internet Control Message Protocol

CHAPTER 1: INTRODUCTION

A Server Monitoring System (SMS) is an essential aspect of running a computer network of a significant size. Network Monitoring Systems (NMS) are used to identify faults before they happen and reduce the impact they may have on users of the network. Networks range in size from small office networks to large Internet Service Provider (ISP) networks, yet a NMS needs to be able to adapt and be scalable to a network of any size, while still providing the same level of insight without overwhelming the user with unnecessary information.

SMS adds insight by employing a number of different monitoring techniques, including:

a. service polling
b. graphing
c. notification system

These different systems interact to provide insight about the network being monitored. Historic data is regularly collected and archived for later analysis. Archived data can be used to help identify the root cause of any problem occurred in the system. SMS helps in day to day process by tracking technical problems which are addressed on the network.

SMS is a web based application which allows monitoring the server with different options such as exploring resource usage, network monitor, server monitor and process monitor. It provides package of facilities which makes it easy to monitor any network. Server admin will be alerted if computer resource usage exceeds the maximum threshold value. It monitors the network for the problems caused by overloaded and/or crashed server. Commonly measured metrics are response time, availability and uptime.

1.1 Problem Statement

Generally, we are unable to access the status of server computer if the server is located far from the admin. If there is overload (CPU, RAM and Disk) in the server, it leads to improper performance of the system and the server may go down. The server may not be able to respond if too many clients try to get the response from the server at the same time. Additionally, it takes a lot of time to physically check each and every server by an admin. Furthermore, there are no simple tools for analyzing the resource usage of the server. Thus, this project aims to provide easy tools to monitor the server.

1.2 Objectives

The main objectives of developing the application "Server Monitoring System" are given below:

a. To develop a network monitoring application that is capable to send alert to network administrator when a problem is detected in the services or server
b. To detect if the problem is due to too many clients' connections to the server and overloading of CPU, RAM and Disk

1.3 Scope and Limitations

1.3.1 Scope

Many servers are required to control many types of systems such as HTTP, Database, etc. This web application could help the user to know their server's status. The security issues in the server computer could send alerts via internet and can be received by any other computer by an administrator. Resource usages (CPU, RAM, and Disk) of the server computer can be viewed by the network administrator in any other computer. The administrator can receive the alerts about the overload to the server and when the server is down unexpectedly.

1.3.2 Limitations

The limitations of this application are:

a. The server has to be run all the time to monitor its activities.
b. The graph can show the representation of an hour only.
c. This application can only be used for Linux servers.
d. External intrusion and attacks cannot be detected.

1.4 Project Features

The features of this project are:

a. Various options are available for monitoring the server computer:
    i. CPU monitor
    ii. RAM Monitor
    iii. Disk Monitor
b. It could help the users to get the alerts if any issues are detected in the server computer
    a) Due to high resource usage
    b) Due to unexpected server failure
c. The administrator can receive the alerts from the server computer to any other computers
d. Graphical analysis of the resource usage

1.5 Outline of Document

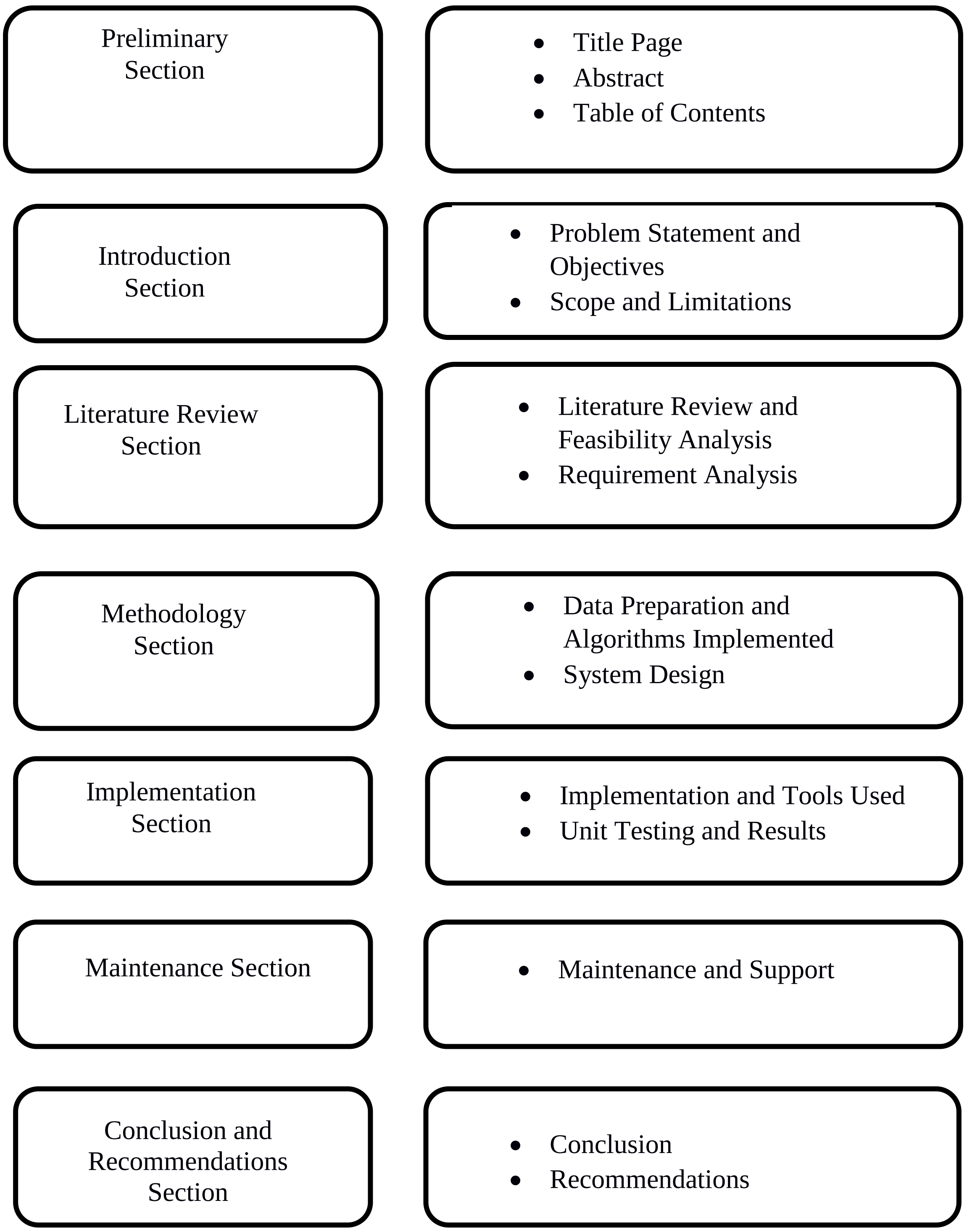


Figure 1- Outline of document

CHAPTER 2: REQUIREMENT AND FEASIBILITY ANALYSIS

2.1 Literature Review

Christopher Roblee, Vincent Berk and George Cybenko have discussed a new server monitoring method based on a new and powerful approach to dynamic data analysis: Process Query Systems (PQS) [1]. PQS enables user space monitoring of servers by using advanced behavioral models and makes accurate and fast decisions regarding server and service state. Data to support state estimation come from multiple sensor feeds located within a server network. By post-processing a system's state estimates, it becomes possible to identify, isolate and/or restart anomalous systems, thus avoiding cross-infection or prolonging performance degradation. The PQS system we use is a generic process detection software platform. It builds on the wide variety of system-level information that past autonomic computing research has studied by implementing a highly flexible, scalable and efficient process-based analytic engine for turning raw system information into actionable system and service state estimates.
Some of the related works have been discussed below:

2.1.1 Cacti

Cacti is a NMS designed for drawing time-series graphs of performance data on a monitored network [2]. Cacti typically draws a different graph for each monitored data source. A graph can be drawn from multiple data sources, but requires a suitable template created using the cacti format. The configuration system is entirely web based and there is no provided method for performing bulk configuration. The additional effort required to update the network model in Cacti often discourages the user from monitoring everything. Cacti also does not include an event detection system or a notification system and therefore is usually used to supplement another resource monitoring by providing historical graphs. These provide more visibility and insight as to why an event may have triggered in another monitoring system. Data to be graphed in Cacti is collected using SNMP at a specified rate. This defaults to 5 minutes but with some effort can be reduced to a faster rate. This data is then archived by storing each data source in separate RRD files.

2.1.2 Nagios

Nagios is a host and service monitoring application designed to inform network administrator about network problems before the clients, end-users or managers do [3]. It has been designed to run under the Linux operating system, but works fine under most *NIX variants as well. The monitoring daemon runs intermittent checks on hosts and services that have been specified using external plug-ins that return status information to Nagios [4]. When problems are encountered, the daemon can send notifications out to administrative contacts in a variety of different ways (e-mail, instant message, etc.). Current status information, historical logs, and reports can be accessed via a web browser.

2.2 Feasibility Analysis

Feasibility analysis is the measurement of how beneficial and practical the built application is to an individual or an organization. The feasibility analysis discusses the intention of the application to the client and its usefulness in the present context.

In this application any server administrator can use the application to know the resource usages of the server computer. So this project is feasible.

Three key considerations are involved in feasibility analysis.

2.2.1 Technical feasibility

The Server Monitoring System will be developed using the Python and PHP environment. The reason for using Python, as the development platform is that, Python could help for the better searching and consists libraries which are very helpful to acquire the required operations. PHP will be used for the user interface for our web application.

2.2.2 Economic feasibility

This is the most frequently used method for evaluating the effectiveness of a system. It is also called as a cost analysis. This project requires a computer for its support and one could easily use it for knowing the resource usages of the server computer.

2.2.3 Operational feasibility

The Server Security and Alert System is a user-friendly web application developed in order to make the user know about the server's status and to get the alerts from the server computer. The server administrator could know the resource usages of the server computer graphically in any computer by using this web application.
Most of the mobile phones and computers nowadays supporting web browsers can view the resource usages of the server computer by using this application.

2.2.4 Schedule feasibility

The total estimated time for the development of the application was about 5 month and the application development was completed in a time period of 4 months with additional 1 month for testing and bug fixes.

2.3 Requirements Analysis

Server Monitoring System will require a computer with browser and internet connection to run. The web application is tested on Mac OS, Microsoft Windows OS, Linux (Ubuntu, Mint and Fedora) for viewing the resource usages. But the server computer must have Linux installed on it. It requires a web browser (tested on Chrome, FireFox, Safari, Microsoft Edge and Internet Explorer) with an internet connection.
The functional requirements and the non-functional requirements of the application are discussed below:

2.3.1 Functional requirements

The functional requirements of the application are listed below:

a. Display list of servers and their status
b. Display the graph of the server status
c. Send notification to the admin about the state of the server (warning and criticality notification via email)

2.3.2 Non-functional requirements

The non-functional requirements of this project are:

a. Fetch the usage data from the server in every one minute
b. Send the fetched data from the server to the web database.

The Use Case Diagram of Server Monitoring System is given below:

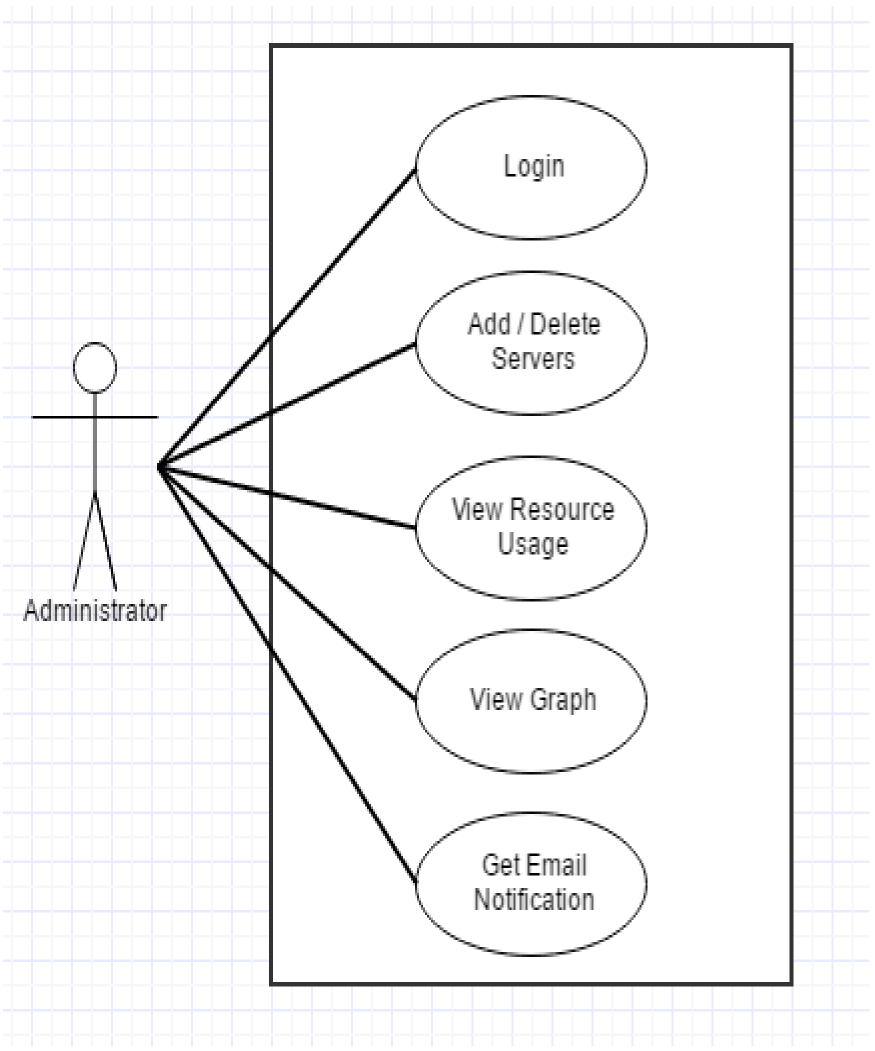


Figure 2- Use case diagram of SMS

The administrator can login to the system and can add or delete new servers for monitoring. Also, the administrator can view the resource usage of different servers and can see a graph of the resource usage. The administrator gets email notification in case of high resource usage in the server.

CHAPTER 3: METHODOLOGY

Effective management of the software project depends on thoroughly planning with the progress of the project. The problems which may arise be anticipated and tentative solution to that problem should be prepared. A plan drawn up at the start of the project will be used as the driver of the project. Planning is an iterative process which is only complete when the project itself is complete.

The software development model used for building this application is the spiral model because of its extended capability and benefits of the traditional waterfall model as well as the incremental development strategies.

Regular design mock ups have been made and regular testing have been done for each module to build a complete product within a defined interval of time.

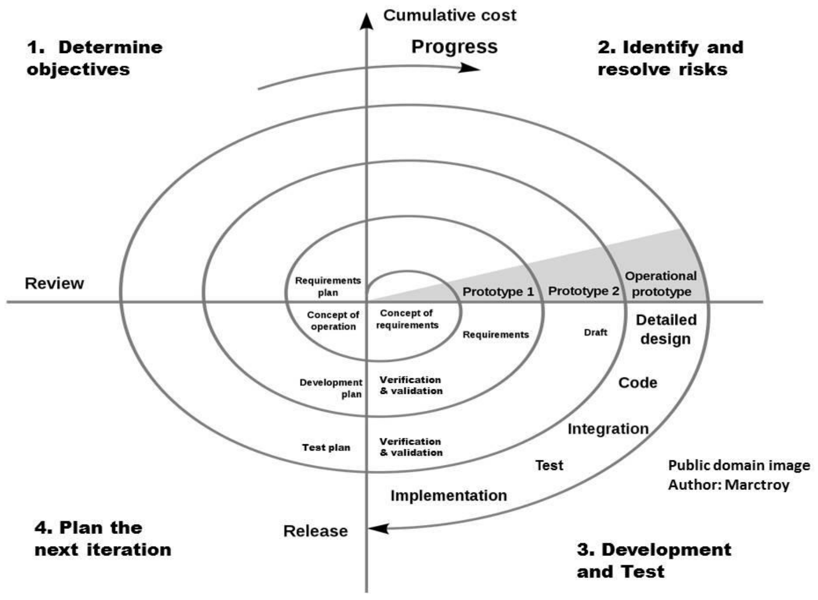


Figure 3- Spiral model

For monitoring the server's status, the data has been fetched from the server and sent to the web database. From the data, a graph for individual server is plotted, which can be easily viewed by the administrator.

3.1 Data Preparation

3.1.1 Data collection

The data has been fetched from the server computer in real time fashion. The data is fetched in every minute and sent to the database through Json dump to display it in visual form. Data for new server is manually entered by the admin in the form of UID (user id) and passkey (unique key identifier for the server). The data obtained from the server computer is shown in APPENDIX.

3.1.2 Data selection

After the data has been fetched from the server computers, all the data is sent to the server from which data about CPU load, RAM usage and Disk usage is selected to represent in the visual form and in the graph (RAM and Disk).

3.2 Algorithms Studied and Implemented

The algorithms used in this project are discussed below:

3.2.1 Server polling algorithm

This algorithm fetches the data from the server computer with the help of various system commands and collects the data in the form of dictionary and sends it to the database through Json dump.

The steps in this algorithm are:

a. Fetch the resource data from the server computer through system commands.
b. Store the data in the dictionary.
c. Send the data to the database through Json dump

3.2.2 Thresholding algorithm

This algorithm is primarily used for send the email to the admin in case of critical state of the server. For this, the threshold has to be set by the admin at first.

The steps of this algorithm are:

```
a. Set the threshold as TW and TC
b. Read the RAM usage as R and Disk usage as D
c. If R > TW or D > TW
          Send Warning Email to the admin
               Else If R > TC or D > TC
          Send Critical Email to the admin
               Else
           Don’t send any Email.
```

3.3 System Design

The system design has been described in the form of data model and process model as given below:

3.3.1 Data model

The data model has been represented as class diagram below. It represents the relationship between the classes and shows the attributes and methods of the classes.

Hostname is the super class and the child classes are Uptime, RAM, Disk and User.

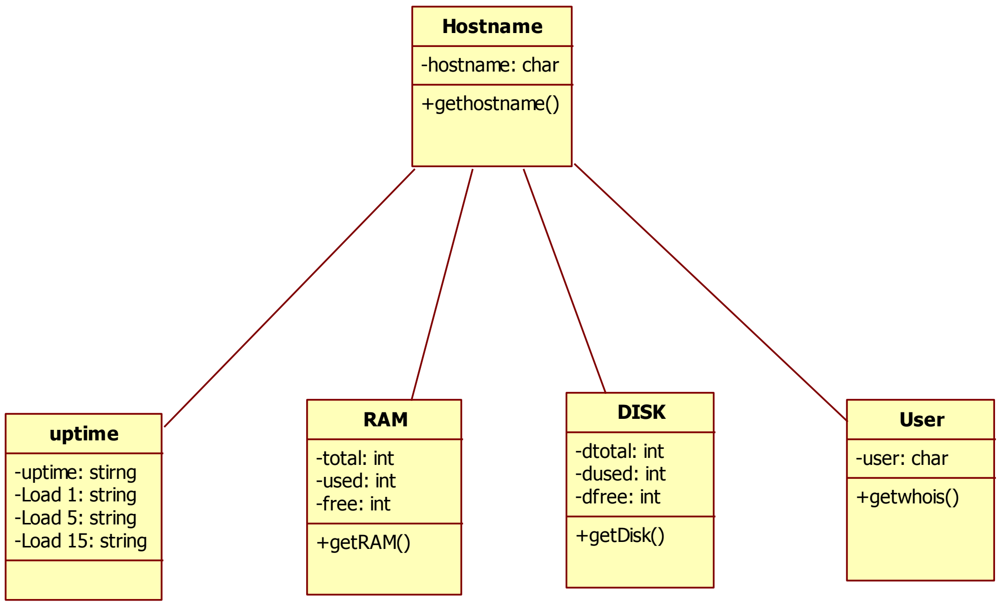


Figure 4- Class diagram of SMS

3.3.2 Process model

The process diagram can be represented in the form of activity diagram and state diagram.

a. Activity Diagram

Figure 5 shows the activity diagram of SMS. The admin can login to the application. Upon incorrect login attempt, multiple options are given for login. After successful login, the admin can view the servers along with their resource usage status and graphs.

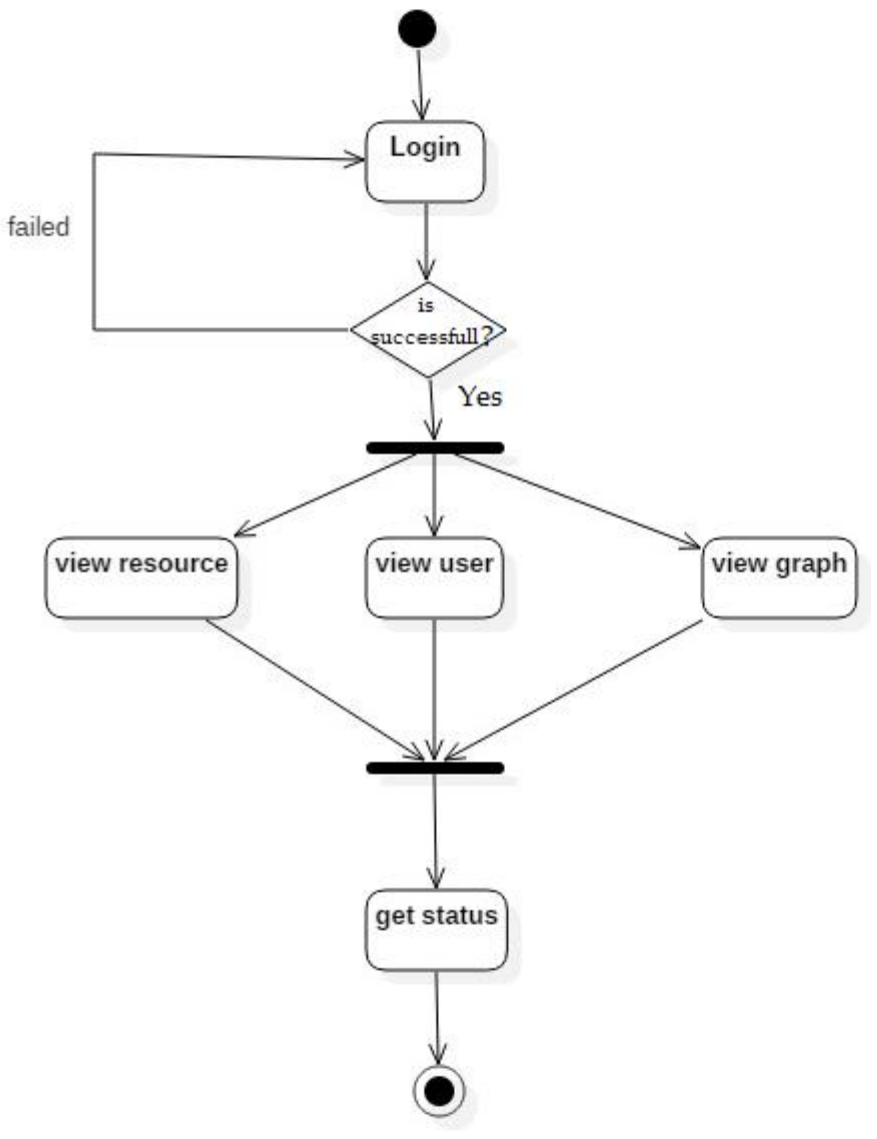


Figure 5- Activity diagram of SMS

b. Sequence Diagram

The sequence diagram models the collaboration of objects based on the time sequence. It shows how the object interact with others in a particular scenario of a use case. The sequence diagram for our project is shown below in Figure 3.4. The user can login to the web application and get the server status through the database.

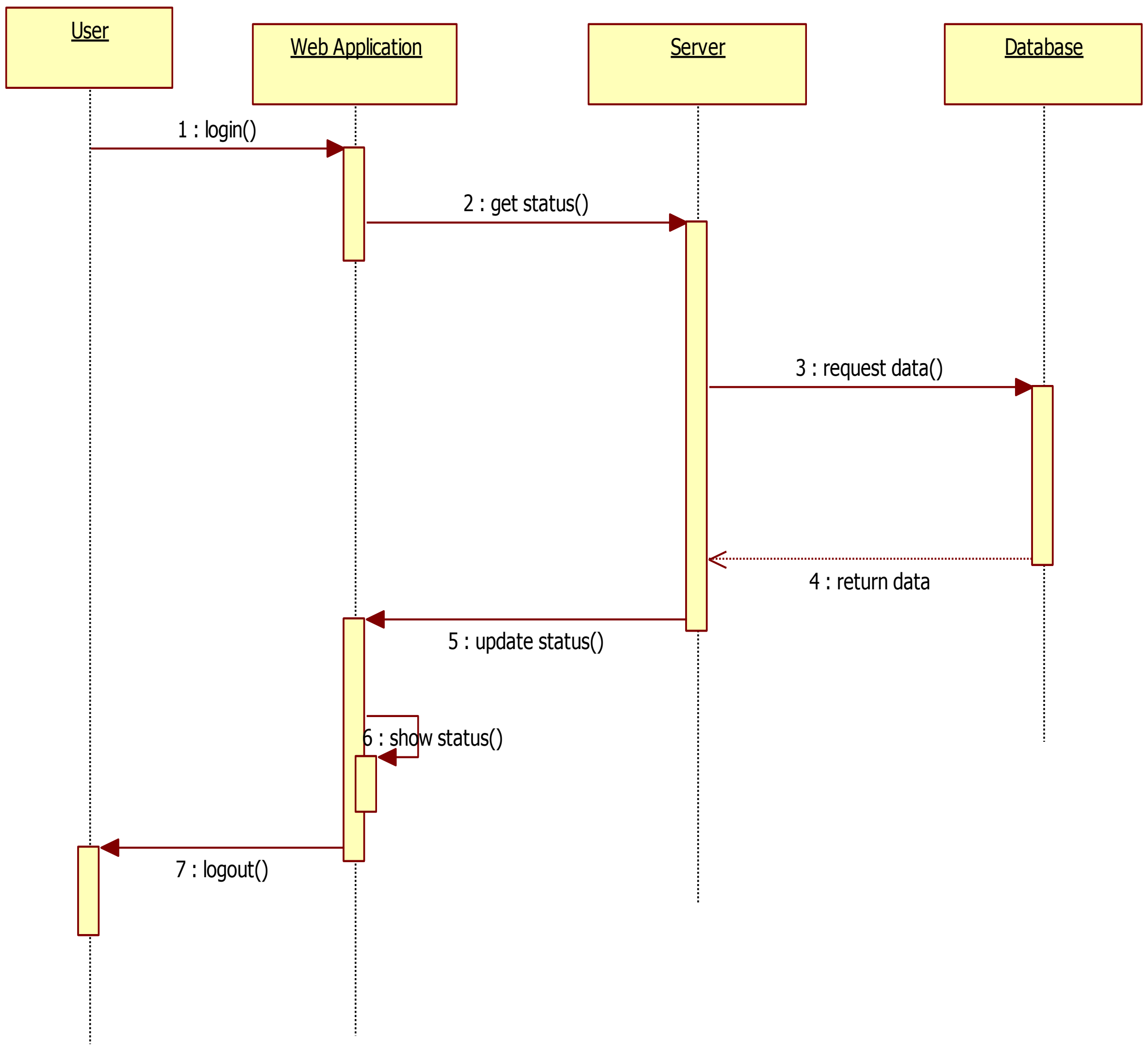


Figure 6- Sequence diagram of SMS

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

This application has been developed using Python programming language along with PHP, HTML and CSS. Server agent was developed in python programming language which retrieves the system information and pushes it to the database server. To view the system information, the web application is developed using PHP, HTML and CSS which extracts data from database server and displays the information of server computer.

The project contains three blocks, the first block is the server block which has to be monitored. From the node agent (server) the data is collected and sent to the database (second block) and it is displayed in the web page (third block).

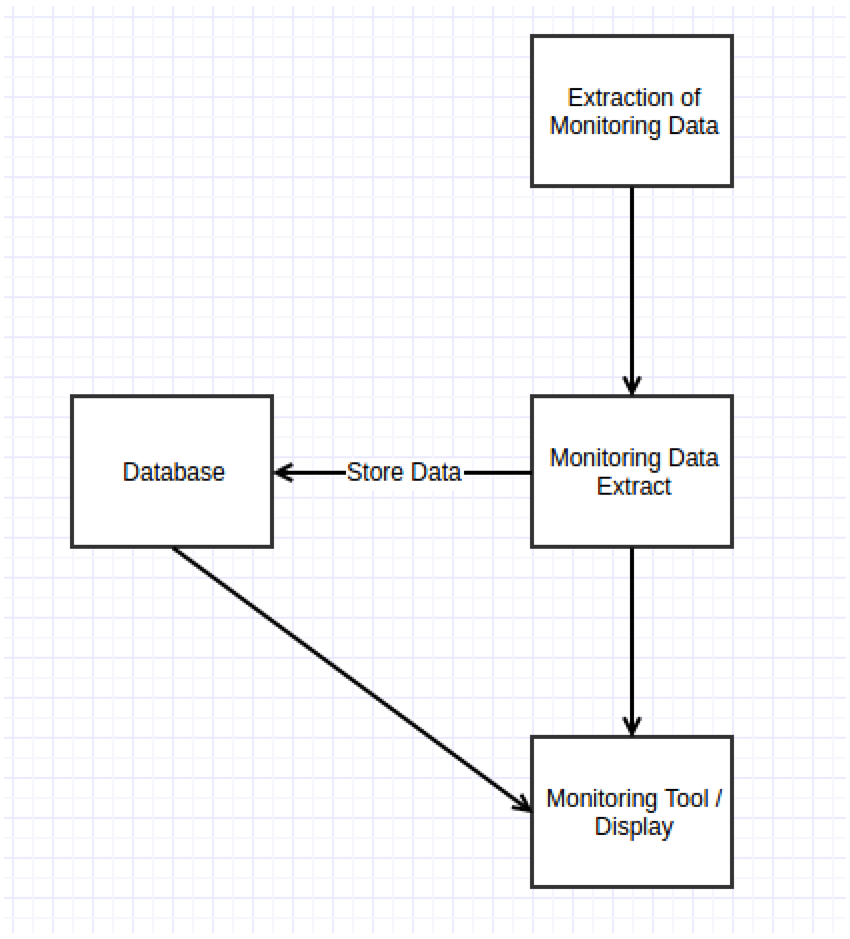


Figure 7- Block diagram of SMS

Using the alert option, the admin can set the threshold value and if any resource usage value exceeds the threshold value then notification will be sent to the admin via email. Also if any server goes then also the admin will be notified as shown in Figure below:

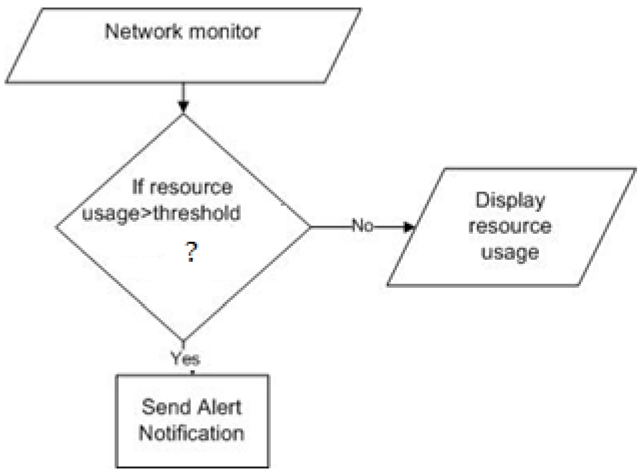


Figure 8- Flowchart of alert system for SMS

4.2 Tools Used

4.2.1 Python

The core part of this application has been developed in python. The Server agent, which retrieves the system information has been developed in python programming language. Python dictionary has been used to store the system data which has been sent to the database through Json dump.

4.2.2 PHP

PHP is especially suited for web development. Since HTML can be easily embedded, it has been used in this application. It has been used for fetching the data from the database. The admin view has been designed in PHP. Also, the graphs have been designed in PHP too.

4.2.3 MySQL

MySQL has been used to store the data collected from the server to the database. This data is then fetched to analyze the status of the servers and to represent in graph. Adminer [5] (Database Client) has been used to store and manipulate the data. Adminer is especially suitable for using with Open Shift, a Red Hat Web Hosting Platform [6].

4.3 Testing

4.3.1 Unit Testing

Unit Testing has been done to check the correctness of various modules. Different test cases have been listed below:

Test Case 1: Email Notification to the Admin:

By increasing the resource usage of the server, the email notification module was cheeked and the test case was successfully validated.

Table 1- Test result for email alert

S.N	Threshold	RAM	Disk	Email Status	Pass/Fail
1.	50	40	35	Not Sent	Pass
2.	50	60	55	Sent	Pass
3.	50	60	30	Sent	Pass
4.	50	45	55	Sent	Pass

Test Case 2: Match the status data of the table to the server’s status data

The obtained server status data through this application was checked by using the system commands for disk usage and RAM usage in the server computer and the test result obtained has been shown in the table below:

Table 2- Result of RAM and disk usage

S.N	Data through Application				Data through System Commands				Pass/Fail
	RAM Total (MB)	RAM Used (MB)	Disk Total (GB)	Disk Used (GB)	RAM Total (MB)	RAM Used (MB)	Disk Total (GB)	Disk Used (GB)	
1.	3847	2343	289	215	3847	2343	289	215	Pass
2.	5870	269	34	21	5870	269	34	21	Pass

Hence, the test was validated successfully.

Similarly, the load of the CPU was also tested and the test result has been tabulated below:

Table 3- Result of CPU load

S.N	Data through Application			Data through System Commands			Pass/Fail
	Load1	Load5	Load15	Load1	Load5	Load15	
1.	0.46	0.64	1.31	0.46	0.64	1.31	Pass
2.	1.23	0.87	0.56	1.23	0.87	0.56	Pass

Hence, the application passed the test performed through various test cases.

CHAPTER 5: MAINTENANCE AND SUPPORT

The application will be maintained and updated over the period of time and necessary support will be provided to adapt the system to the future needs. Some of the strategies are:

5.1 Adaptive Maintenance

The admin needs to add or delete the servers as per the requirements. Also, the design and layout of the User Interface can be changed according to the need.

5.2 Corrective Maintenance

When the application malfunctions, the related issues have to be resolved for smooth running of the application. If the servers malfunction, then the system admin needs to take care of the settings of the servers in the database.

CHAPTER 6: CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

This project has shown the benefits and established how necessary network monitoring is on a large computer network. Without monitoring, a network is a black hole and faults can go unnoticed for extended periods of time. The major issues on server can be traced by using server security and alert system. Service polling checks regularly whether a device or a service is available and is working within normal parameters. It ensures that the web server is operational but also verifies that the web server software on that machine is running correctly. It shows status of CPU load, disk load and memory usage.
Time-series graphs of performance data has been plotted which is useful for identifying trends and anomalies. It shows the time series graphs of disk load, memory usage of one hour history. Such graphs are frequently used for identifying changes given a large quantity of data values. System sends notification to the user if some problem encountered in normal running of system, someone should be notification of the change.

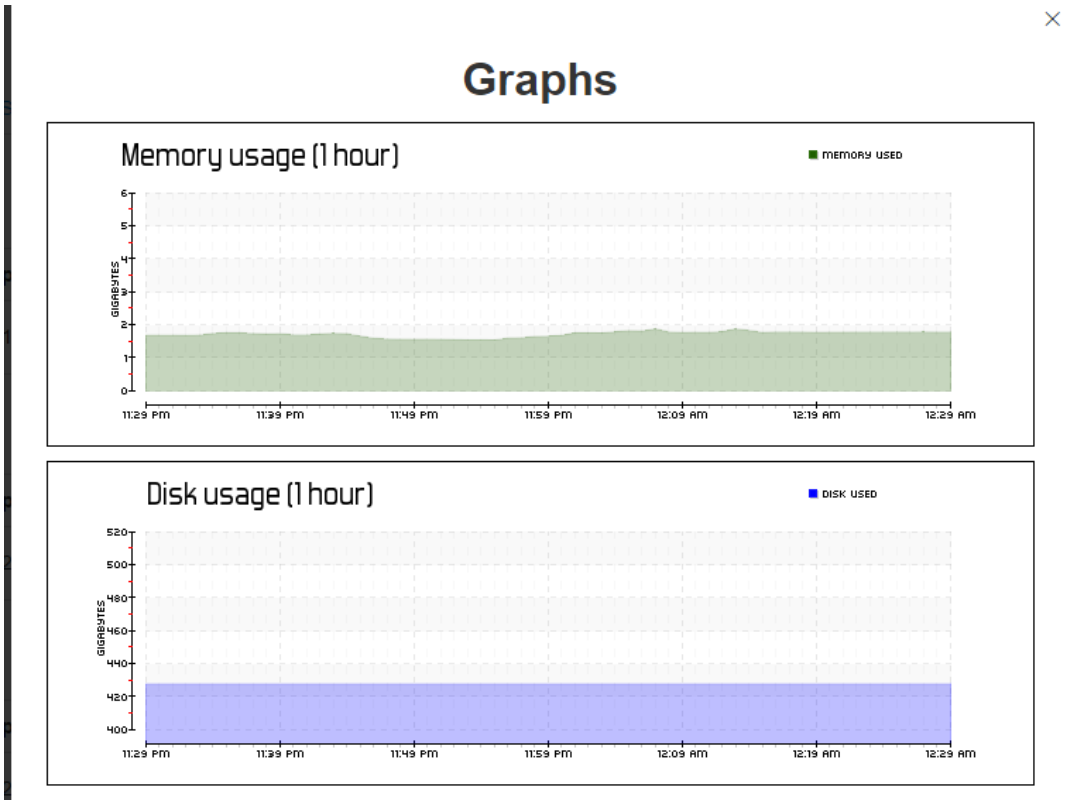
6.2 Recommendations

The system has some limitations which can be overcome with more in-depth study about the implementation. Network monitors and process monitors can be implemented more efficiently. The time series data can be plotted to show the overall usage rather than only hour usage.
Since the current system doesn’t notify about external intrusion, changes can be made to identify attacks and hack threats. Also, the server application can be made to work in the windows servers too.

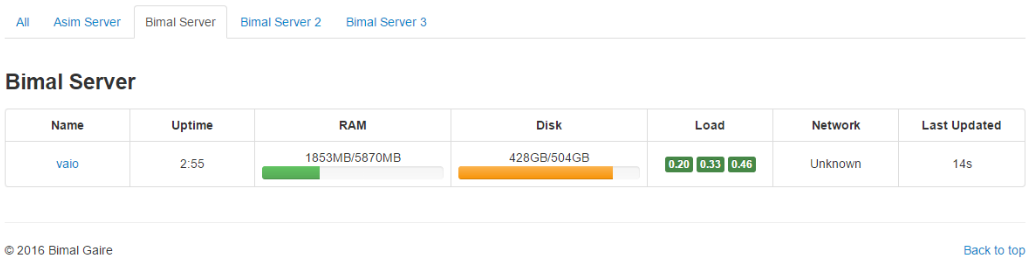
APPENDIX

Some of the snapshots of the application are listed below:

Graph of Resource Usage



Individual Server Status



All | Asim Server | Bimal Server | Bimal Server 2 | Bimal Server 3

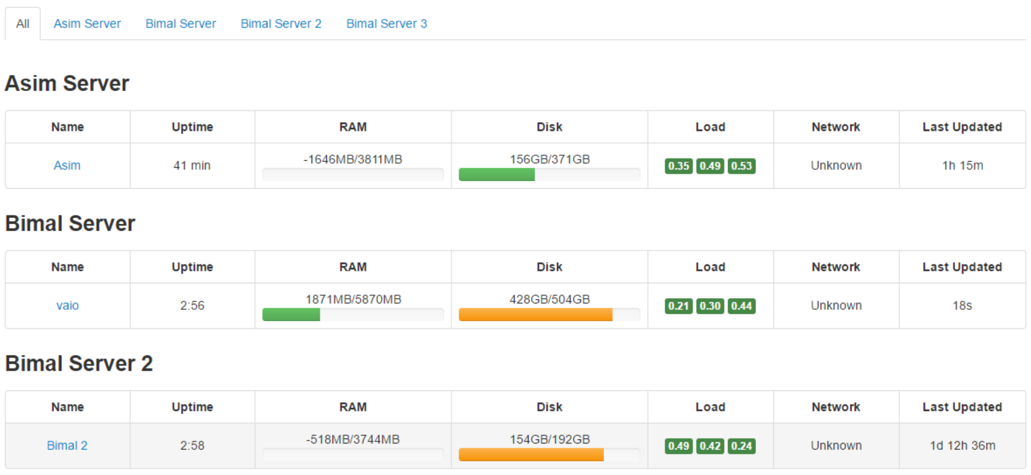
Bimal Server

Name	Uptime	RAM	Disk	Load	Network	Last Updated
vaio	2:55	1853MB/5870MB	428GB/504GB	0.20 0.33 0.46	Unknown	14s

© 2016 Bimal Gaire

Back to top

List of Servers



All | Asim Server | Bimal Server | Bimal Server 2 | Bimal Server 3

Asim Server

Name	Uptime	RAM	Disk	Load	Network	Last Updated
Asim	41 min	-1646MB/3811MB	156GB/371GB	0.35 0.49 0.53	Unknown	1h 15m

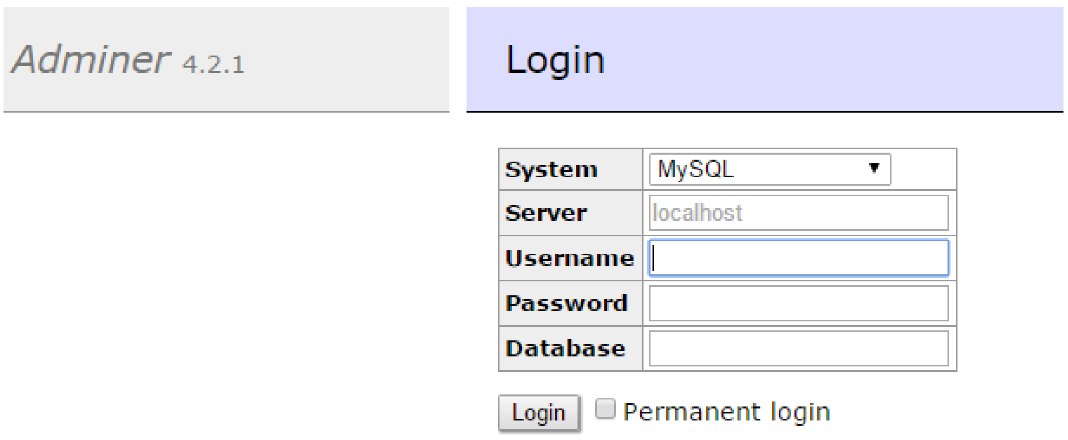
Bimal Server

Name	Uptime	RAM	Disk	Load	Network	Last Updated
vaio	2:56	1871MB/5870MB	428GB/504GB	0.21 0.30 0.44	Unknown	18s

Bimal Server 2

Name	Uptime	RAM	Disk	Load	Network	Last Updated
Bimal 2	2:58	-518MB/3744MB	154GB/192GB	0.49 0.42 0.24	Unknown	1d 12h 36m

Adminer Login Page

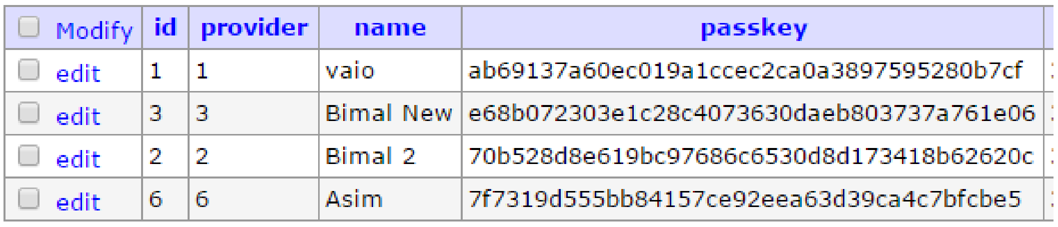


Adminer 4.2.1

Login

System	MySQL
Server	localhost
Username	
Password	
Database	

Server Information Table



☐ Modify	id	provider	name	passkey
☐ edit	1	1	vaio	ab69137a60ec019a1ccec2ca0a3897595280b7cf
☐ edit	3	3	Bimal New	e68b072303e1c28c4073630daeb803737a761e06
☐ edit	2	2	Bimal 2	70b528d8e619bc97686c6530d8d173418b62620c
☐ edit	6	6	Asim	7f7319d555bb84157ce92eea63d39ca4c7bfcbe5

REFERENCES

[1] R. Christopher, B. Vincent, C. George, “Implementing Large-Scale Autonomic Server Monitoring Using Process Query Systems”, Dartmouth College, Hanover, 2013. [Accessed: 11-July, 2016].

[2] “Cacti: The complete rrdtool-based graphing solution”. The Cacti Group, Inc. [Online] Available: http://www.cacti.net. [Accessed: 11-June, 2016].

[3] “Nagios - the industry standard in IT infrastructure monitoring”. Nagios Enterprises. Available: http://www.nagios.org/. [Accessed: June-12, 2016].

[4] “Nagios plugins - the home of the official plugins”. Nagios Plugins Development Team. [Online] Available: http://nagiosplugins.org/. [Accessed: 23-June, 2016].

[5] "Using Adminer to Manage Your Databases." LinuxCareer's Documentation, 2016. [Online]. Available: http://how-to.linuxcareer.com/using-adminer-to-manage-your-databases/. [Accessed: 29-July, 2016].

[6] "Adminer - Why is better than phpMyAdmin?", Adminer.org, 2016. [Online]. Available: https://www.adminer.org/en/phpmyadmin/. [Accessed: 12- Aug-2016].

BIBLIOGRAPHY

Pousty, S. (2014). *Getting Started with OpenShift.* Washington D.C.:O’Reilly Media, Inc.

Ryder, T. (2016). *Nagios Core Administration Cookbook*, New York: Pckt
Publishing