NETWORK MONITOR

By

Muhammad Salman Taj

Naveed Aqdus

Salman Akram Cheema

Supervisor:

Lec Waseem Iqbal

Submitted to the Faculty of Computer Software Engineering
National University of Sciences and Technology, Islamabad in partial fulfillment
for the requirements of a B.E. Degree in Computer Software Engineering

May 2014

CERTIFICATE

It is certified that the work contained in this thesis entitled "**Network Monitor**" carried out by Muhammad Salman Taj, Naveed Aqdus and Salman Akram Cheema under the supervision of Lec Waseem Iqbal for partial fulfillment of degree of Bachelor of Software Engineering is correct and approved.

Supervisor: _____

Lec Waseem Iqbal

_____ Department

MCS

Dated: _____

ABSTRACT

Network Monitor is a network administrator tool which facilitates the administrator to know and manage the network more effectively and efficiently. It also helps save the network users time by helping the user find the available workstation even before entering the network area physically. The motivation which led us to undertake this challenging task was our network administrator/ user problems here at the institute. An issue that needed urgent attention was that of the network clients “workstations” going off line without giving any warning to the administrator. It became a problem for the administrator as well as the network users. This problem has specially been addressed here by our software product.

Network Monitor may not be a complete administrator tool but it does ensure employing of a lot of features that facilitates the administrator and the user of a network. We have developed this windows based software with a client and server module. The server module periodically evokes the clients and registers their statuses. In case of a system going off line it warns the administrator of it. The client module checks the windows registry to see if the system is occupied by a user or not (checks log in). a separate display outside the network facilitates the users of the network by allowing them to see which workstation is taken, available, offline or online etc. The server module maintains the log of each workstation and the graphs of their activities can be generated with a single click. To avoid chocking of the total bandwidth a very small time interval has been set for which the application employees the bandwidth periodically.

Various Testing and evaluation results conducted on the product are extremely promising.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and unstinting cooperation,

a work of this magnitude would not have been possible

ACKNOWLEDGEMENTS

There is no success without the will of ALLAH. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality. We are also grateful to our parents and family and well-wishers for their admirable support. We would like to thank our supervisors Lec Waseem Iqbal and AP Bilal Rauf, for their help and motivation throughout the course of our project. Without their help we would have not been able to accomplish anything. They provided us with the opportunity to polish our technical skills and guided us into this area of learning.

TABLE OF CONTENTS

Chapter 4 (Software Design Specifications)

Chapter 5 (System Implementation)

Chapter 6 (Testing)

Chapter 7 (Conclusion & Future Work)

List of Figures

Design Diagrams

Appendix A

List of Tables

Table Number **Page Number**

CHAPTER 1

INTRODUCTION

1. Introduction

1.1 Purpose

This project is aimed at developing a Network Monitoring Application for windows based systems. There are a number of universities/ institutes in Pakistan where the old traditional ways of network administration are practiced. This system will make network monitoring more efficient and effective, thereby facilitating the administrator in managing the network and also helping the users to more effectively use the network resources. Our special focus is on the network employed at the Computer Science Department in (MCS) Military College of Signals (NUST) National University of Science and Technology. The department has five labs in total under the same network. No real network administrator tool is in employed to effectively view the statuses of the workstations available on the network. This makes the job of the network administrator more cumbersome and hectic as one has to manually check each workstation when in doubt. This requires manpower and time. More over the students often find it difficult to find a workstation that has the network available on it on the first attempt. The System's purpose is to facilitate the network administrator by automatically generating erroneous messages once a workstation goes off line. It will also facilitate the network user's allowing them to find an available and connected workstation via an LCD screen outside the lab.

1.2 Project Background

Network Monitor is a windows based application that helps the network administrator and user to a great deal in administrating and using the network resources. The output of the application is basically the current status of the workstations under the network it is working for. It helps the network users to find a working and available workstation to work on.it also facilitates the network administrator in administrating the network more efficiently. There are a number of software's commercially available in the market with slightly similar but varying features.

The project tilted as "Network Monitor" will find the solution to network administrators and users problems. The core tasks involved in the project are reading statuses of workstations, generating graphs, registering and maintaining a log, generating a user's view, scanning IP's, identifying IP conflict[1] etc.

1.3 Project Scope

The project scope is the development of a network monitoring system with main focus on error generation for workstations going off line. The main features include Periodic Ping Request Manager, User Display Updater, Database Management, Lab Setup System, Administrator Console and Client and Server [2] Modules.

1.4 Objectives

1.4.1 Learning windows application development/interfacing.

1.4.2 To develop a useful application for network administrators and users.

1.5 Work Breakdown Structure

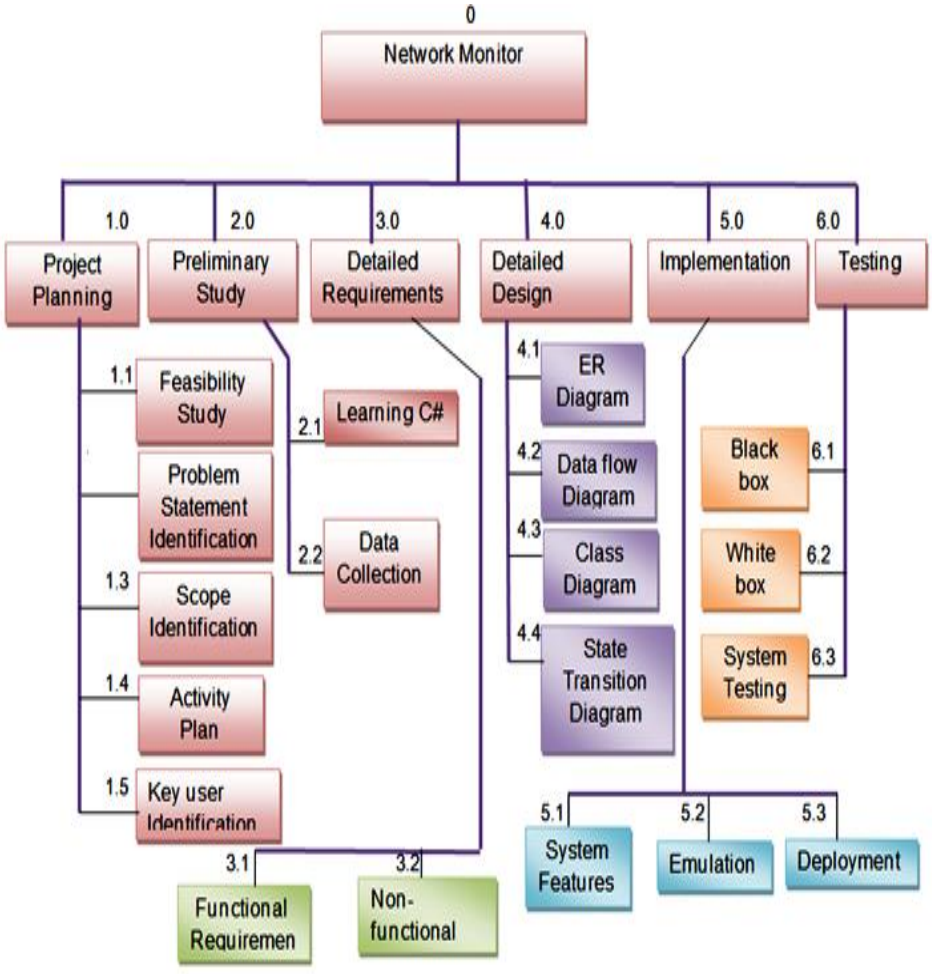

Figure 1-1 Work Breakdown Structure

1.6 **Deliverables**

Deliverable Name	**Deliverable Summary Description**
Synopsis	A proposal of a project to be undertaken is put forward for approval.
Defense	A presentation is presented to defend the proposed project.
Software Requirements Specification(SRS) Document	Complete Description of WHAT system will do, who will use it. Detailed description of functional and non-functional requirements and system features.
Analysis Document	Detailed requirement analysis and analysis models are included.
Design Document	Complete description of How the system will do. Design models are included.
Code	Complete code with the API.
Testing Document	Whole system is tested corresponding to the specifications. System is tested at all levels of Software Development Life Cycle (SDLC).
Complete System	Complete working system.

Table1: Deliverables

CHAPTER 2

LITERATURE REVIEW

2. Literature Review

2.1 Introduction

Network monitor application is essential for companies/ institutes, to ensure that their computer systems are running smoothly and that no outages occur. Our motivation was actually a consequence of the problems we face in our department labs in routine. Finding a system that is well connected to the server often becomes a search process which in turn wastes the time of the workstation user. Administrators have similar issues. An administrator is never sure which system is on line and which one has gone off line unless they are physically checked. If we go deeper in to the problem we can come up with many other issues that can be addressed in the future versions of the software under construction by us. For example a user trying to handle the registry setting of a workstation can be caught and stopped via this system, setting up of software's by any one other the administrator can be put under surveillance.

So far we are addressing the very basic but very important issue through our final year project. We are developing software that will ensure that a user of the workstation only has to view an LCD placed outside the lab to find a working vacant workstation and go straight to it rather than roaming around checking multiple systems for their status. Software that will facilitate the administrator by alerting him with erroneous messages by the client workstation that goes off line for any reason.

As mentioned that our basic motivation came from our own experiences that we gained from our department labs but once we went on to study the subject in detail we came across various software's developed for this purpose by different companies. The software's are much sophisticated and cover a lot more ground than we plan to cover in our first version of this product. This paragraph is followed by the literature review of some of the products we came across. It should be in the minds of the readers that these few products that we studied are way bigger in their scope than our product. They are all commercial products and cover various problem areas which might not be of interest in our line of work at this point and time. The undertaken project's scope is "To develop a Windows based application to make administration of networks easier and to help the network users in utilizing the network resources more efficiently".

2.2 **Problem Domain**

The problem at hand is an important one. A fast an available network environment confirms a good working atmosphere. The problem at hand is an important one. A fast and available network environment confirms a good working atmosphere. Networks always have issues when it comes to problems like workstations not alive etc. these problems needs to be resolved for a better output from the workstation and the network users. More over the administrator's job should be made easier and what can be better than software doing the same job. There are many other requirements that may be required at the networks level. For instance in a network environment where computers originality is not to

be tampered with i-e no new software's are supposed to be installed (even though the workstations have the admin rights), a simple trick can help. As soon as an installation is tried, a message containing the details of the workstation on which the software is being installed and credentials of the user on it, is sent to the administrator.

Similarly access to unauthorized websites can be denied and in case someone tries to violate this rule can be caught effectively.

2.3 Related Work

Since the employment of networks efforts have been made to adapt methods that would make the network administrating easier and will facilitate the network users as well in process. Many different applications have been designed to help the case. Some of them are included in the discussion below.

The first example is that of "Spiceworks" [3]. Every day IT professionals and technical vendors use Spice works to do their jobs in a truly social way. IT professionals manage their networks while interacting directly with other IT professionals and technical vendors to decide what to buy. IT professionals trust the people, information and brands they meet in Spice works. And find Spice works more convenient than having to visit multiple media or vendor websites. Meanwhile, tech vendors connect with IT buyers where and when their tech offerings are most relevant. And get to build the relationships that buzz and new business are made of. Making it no surprise that Spice works is the well-loved vertical IT network among IT pros and marketers alike. Some of the features are monitoring your network and PC's, the alerts you want, when you want them,

monitoring and managing email, monitoring network bandwidth, windows performance monitoring, keeping tabs on all your data, error generation on failures etc. the software is fast and swift but too expensive for everyone to afford.

The network monitor software "PRTG" (by Paessler) [4] is again expensive but flexible to use and easy to deploy. PRTG Network Monitor Software helps you to Optimize Your Network. It is an uptime and bandwidth monitoring software that supports a broad variety of sensor types. The network monitoring software is currently being used by over 150,000 customers worldwide. The benefits are increased profits: no losses caused by undetected system failures, improved customer satisfaction by providing more reliable systems, peace of mind: As long as they do not hear from the monitoring tool they know everything is running perfectly.

Last but not least we have "SevOne" [5]. It monitors your entire infrastructure, including networks, applications, and systems. It includes support for SNMP, NetFlow, IP SLA, JMX, NBAR, WMI, VoIP, vCenter, and more, all without additional SevOne agents. Some of its features are Baseline Performance, Generate IT reports in Seconds, Receive Proactive Alerts, and View Real Time Dashboard etc.

2.4 Limitations

Network tool always pose a problem related to bandwidth. Reason is that the only way to continuously monitor your resources on the network is to continuously use the network. In process the tool takes a lot of bandwidth and in process disabling the network users to make efficient use of the bandwidth. To counter this issue to some extent we have employed Transmission Control Protocol. A client module is installed and kept hidden on the network workstations. One a user logs in the module's executable file automatically executes and on the transport layer it is communicated to the server module that the system is up and running and that it has been occupied by a user. We have to send a ping request to check the status of a system only when this executable file is not running. This reduces the need for sending ping requests by 80 percent in working hours (when the users are using the systems).

2.5 Proposed Project

The proposed project finds the solution to network administrator and users issues with the network administration and use. The core features of our project include:

2.5.1 Register and plot all workstations on a network.

2.5.2 Display administrator and users views.

2.5.3 Register and maintain log.

2.5.4 Generate graph.

2.5.5 Identify IP Conflicts.

2.5.6 Scan for IP addresses.

2.5.7 Send Erroneous messages to the administrator.

2.6 Technological requirements

Our project requires following software and hardware requirements.

2.7 Software Requirements

The software(s) required for the implementation of our project includes:

2.7.1 Visual Studio 2013

2.7.2 SQL Server 2010

2.7.3 Dot Net Framework 4.0 or above

2.8 Hardware Requirements

The Hardware required for the implementation of our project includes:

2.8.1 Server PC with Windows 7 and above.

2.8.2 Client systems with minimum of one lab constructed.

2.8.3 Necessary cable connections.

2.8.4 User display and required system attached to it.

CHAPTER 3

SYSTEM REQUIREMENT

SPECIFICATION

3. Introduction

3.1 Purpose

There are a number of universities in Pakistan where the old traditional ways of observing a network are practiced. We will be developing a system to make network monitoring more efficient and effective, thereby facilitating the administrator in managing the network and also help the users to more effectively use the network resources. Our special focus is on the network employed at the Computer Science Department in (MCS) Military College of Signals (NUST) National University of Science and Technology. The department has five labs in total under the same network. No real network administrator tool is in practice to effectively view the statuses of the workstations available on the network. More over the students often find it difficult to find a workstation that has the network available on it on the first attempt. The System's purpose is to facilitate the network administrator by automatically generating erroneous messages once a workstation goes off line. It will also facilitate the network user's allowing them to find an available and connected workstation via an LCD screen outside the lab.

3.2 Document Conventions

The format of this SRS is simple. Bold face and indentation is used on general topics and on specific points of interest.

3.3 Intended Audience and Reading Suggestions

This document is intended to be read by different type of readers such as project supervisor, project coordinator, project managers, project panel, external evaluators, developers, project managers, users, customers and testers. The document will be analyzed by the customer. Changes will be made if the suggestions proposed by the customer are feasible and the next SRS will be submitted to customer for analysis again. This process will be repeated till the satisfaction of the customer. After analysis, the SRS will be accepted or rejected. For reading suggestion this document should be read in a sequence in which it is presented.

3.4 Product Scope

To develop a network monitoring system with main focus on error generation every time a workstation is non-responsive to the servers ping request. It will alert the network administrator and pinpoint the location of the workstation that needs attention.

3.5 References

Project Synopsis- An approved document with complete information containing extended title, brief description of the project, scope of work, academic objectives, applications, previous work, material resources required, no of students and special skill required has already been submitted to the department.

3.6 Overall Description

3.6.1 Product Perspective

This product is a new, self-contained product. Our system is the replacement of manual network monitoring system. The system will allow access to authorized users only with specific roles.

3.6.2 Product Functions

3.6.2.1 Abstraction level of the system should be high.

3.6.2.2 The product must be an application of network monitoring.

3.6.3 User Classes and Characteristics

3.6.3.1 Network Administrators can use it for administrating their networks more effectively.

3.6.3.2 Network Users can benefit from it by not having their time wasted looking for available network resources.

3.6.3.3 The network monitor is divided in to three main modules.

3.6.3.3.1 The **Server Module** will be setup on the administrator system and will give a GUI to the network administrator. It will allow the administrator to ping a selected workstation/ workstations with a right click option. It will periodically ping the clients and generate error messages for the administrator in case a client does not reply. It will also give input to the user display placed outside the network.

3.6.3.3.2 The **Client Module** has no GUI associated with it. It will always run in the background. The client module will be installed on each workstation and will only execute once a user logs in to the workstation. It will be hidden from the user.

3.6.3.3.3 The **User GUI Module** will be installed on a separate system associated with the user display outside the network environment. It will be password protected and

will only be utilized to set up the lab matrix for the users to view.

3.6.4 Operating Environment

Software system will operate on Windows based systems.

3.6.5 Design and Implementation Details

The system will be developed using the C# (C Sharp) language tool. MySQL will be required at server end. Based on these technologies there can be implementation constraints.

3.6.6 User Documentation

No user manual for the network users is required. Users will only have a display which will guide them to an available work station. A document (user manual) will be available for the administrator to get to know the system better.

3.6.7 Assumptions and Dependencies

3.6.7.1 User Related

3.6.7.1.1 User knows English language as the user GUI will be provided in English.

3.6.7.1.2 The user has sufficient knowledge of computers.

3.6.7.2 **Administrator Related**

3.6.7.2.1 Administrator is qualified and has sufficient knowledge about networks. Administrator knows English language as the administrator GUI will be completely in English.

3.6.7.2.2 Bandwidth Related

3.6.7.2.3 The computers shall have an efficient LAN connectivity.

3.7 External Interface Requirements

3.7.1 User Interfaces

For the user interface the workstations are arranged in a matrix form. The view depicts the actual view of the lab layout. A legend is provided for the user's ease. The red color depicts that the workstation is not available on the network. The green color depicts that the work station is active and available for occupation and the yellow color depicts that the workstation is online but occupied.

On the other hand the administrator interface has more than once screen i-e the login screen, main screen and the a screen for each lab setup. The same three color codes are used for the administrator as well. Red represents that the system is not responding, yellow represents that the system is not responding but was responding to the recent calls and the

green represents that the system is responding to the calls and is available on the network.

3.7.2 **Hardware Interfaces**

- **Client End Minimum Configuration**
  - **Memory**

    4 GB for Windows 7/8
  - **CPU**

    Minimum CPU for client should be Intel Core i3 CPU@ 1.8 GHz
- **Server End Minimum Configuration**
  - **Memory**

    16 GB for Windows 7/8
  - **CPU**

    Minimum CPU for server should be Intel Core i7 CPU@ 2.50 GHz
  - **Hard Disk**

    One 200 GB hard disk space would be sufficient
- **User Display End Minimum Configuration**
  - **Memory**

    2 GB for Windows 7/8
  - **CPU**

Minimum CPU for server should be Intel Core i3 CPU@ 1.50 GHz

- **Hard Disk**

  One 40 GB hard disk space would be sufficient

3.7.3 Software Interfaces

- **Client End Minimum Configuration**
  - **Scripting Language**

    C Sharp
- **Server End Minimum Configuration**
  - **Scripting Language**

    C Sharp
  - **Database**

    MySQL
  - **OS**

    Windows Server 2003/ 2008

3.7.4 Communication Interfaces

The system will be connected to LAN with a speed of 100/ 1000 Mbps.

3.8 System Features

3.8.1 Making an Account

3.8.1.1 **Description and Priority**

The foremost step is for an administrator to create an account. Before the creation of the account the head of the department first logs in with his/ her special password protected login and activates one or more predefined passcodes. The passcode will be provided to the legitimate administrator and he/ she will only then be able to create an account as an administrator. The passcode earlier activated will automatically be deactivated as soon as it is used.

3.8.1.2 **Stimulus/Response Sequences**

User will request for passcode and then on receiving it he/ she will click on the signup button.

3.8.1.3 **Functional Requirements**

The user must be in possession on the passcode without which the user will not be able to proceed with the registration process. The user shall fill up all the appropriate boxes to complete the signup process.

Requirement-1: Passcode possession

Requirement-2: Fill signup details completely

3.8.2 **Set up Administrator's View**

3.8.2.1 **Description and Priority**

After signup and then sign in the administrator must first setup the network view for him/ herself. This will require scanning of all the IP addresses associated with the network section by section. The sections depict the labs in an instructional institute environment.

3.8.2.2 **Stimulus/Response Sequences**

The user creates new sections and then scans the associated IP's for each section.

3.8.2.3 **Functional Requirements**

The network must be in place with workstations attached to it.

Requirement-1: Create sections

Requirement-2: Scan IP addresses

3.8.3 **Set up User's View**

3.8.3.1 **Description and Priority**

Each lab has a separate user's view outside it with a separate CUP installed. The CUP holds a module which is password protected and only has the functionality of setting up the users view. It takes the information required from the main server module and processes it.

3.8.3.2 **Stimulus/Response Sequence**

We are setting up a view easily readable by the user. The user can then use this view to find an available resource.

3.8.3.3 **Functional Requirements**

The user view hardware must be installed and only the administrator should have access to it. The administrator should set up the lab by providing the dimensions of the lab to the module.

Requirement-1: hardware must be installed

Requirement-2: Module must be connected to the server

Requirement-3: Lab dimensions should be accurately provided

3.9 Other Nonfunctional Requirements

3.9.1 **Performance Requirements**

The database shall be able to accommodate minimum 1000 status records of client workstations. The response time of a workstation will be directly proportional to the total number of clients registered to the server. As each client will be periodically invoked (ping will be sent) after a predefined time period in a specified order, so every client will have to wait its turn. Each time the client will be at maximum invoked three times (in case the first two pings fail).

3.9.2 **Safety Requirements**

Client modules will not be accessible to the workstation user.

3.9.3 **Security Requirements**

All client workstations will be properly authenticated before being registering to the system. By authentication we mean checked physically on ground.

3.9.4 **Software Quality Attributes**

3.9.4.1 **Availability**

The system shall be available throughout the working hour's i-e 0800 hours to 0315 hours.

3.9.4.2 **Mean Time between Failures (MTBF)**

The system will be deployed in such a way that it does not fail during the working hours.

3.9.4.3 **Mean Time to Repair (MTTR)**

Even if the system fails during the working hour, I will recover within 20 minutes.

3.9.4.4 **Maintainability**

The maintenance of the system shall be done on monthly bases.

3.9.4.5 **Design Quality**

3.9.4.5.1 **Evaluation**

The system will have the capability to evolve according to the environment of the network it is deployed in. it will automatically generate the network diagram of the workstations placed on it. It will also ask the user to specify the total number of division on the network.

3.9.4.6 **Usability**

3.9.4.6.1 **Learnability**

The administrator will not need a thorough training to work the system. Any qualified network administrator will take no more than 30 minutes to get used to the system environment himself. Although the administrator can use the administrator user guide to thoroughly learn the system in about one hour time. The users only have to know how the output screen works. It needs no time to learn. Anyone with common sense will be able to read the users screen which shows the availability of the systems.

3.9.4.6.2 **Efficiency**

The individual system is expected to reply with in the 10 seconds time period allotted to it. If it doesn't, it will be considered to a failure and an on ground check will be carried out.

3.9.4.6.3 **User Satisfaction**

We are expecting that at least 70 % of the user's will be satisfied with the system after using it. We will carry out a survey in which users will grade the system from 1-10 (where 1 is low and 10 is high).

3.10 **Business Rules**

3.10.1 Client module should be hidden from the user of the workstation.

3.10.2 Hardware failures and power failures should be handled manually.

3.10.3 All physical connections should be checked physically.

3.10.4 The first time scan should be carried out to ensure all systems are on board.

3.10.5 Administrator should be provided with the system password only.

3.11 **Other Requirements**

All the requirements have been already stated in SRS.

CHAPTER 4

SYSTEM DESIGN SPECIFICATION

4. Architectural Design

4.1 System Block Diagram

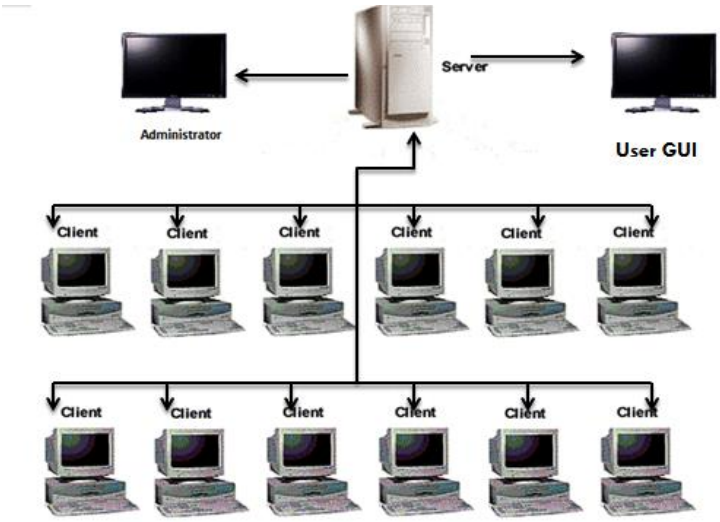

Figure 11: System Block Diagram

4.2 High Level Design Diagram (Modules Identification)

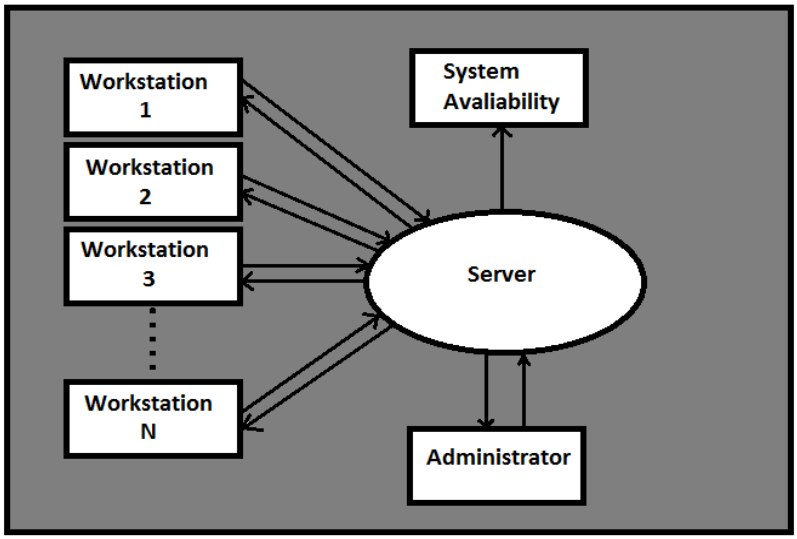

Figure 12: High Level Design

4.3 **Architectural Style**

Architecture of network monitor can be modeled using **Model-View-Controller**. Interface of the system is distinct from the application logic. Interface forms the View of the architecture. Model has the application data .Controller controls the coordination between the view and the model. There is no direct communication between model and the view all the communication is directed using the controller. The system is divided into modules as per the main functionality of the system. These functionalities can be used separately as off-the shelf components .Interface of this system has no application logic embedded in it so the system architecture can easily be made using Model View Controller approach. User class which has the functions related to the GUI forms the **View**. IP Scan Button, Ping All Button, Show Log Button, User View Setup Button, Stop Pinging Button, Show Graph Button, Help Button are the functions of User class .These function handle the processing of interface but no application logic is involved in it. **Controller** in this system is use-case handlers as the system functionality is clearly divided in modules. But for reusability of the system a façade controller will be a bad choice so use-case handler controllers are used. **Model** in this system is the User View Form, Ping Form, IP Scan Form, and Trace Form which is encapsulating the application logic. These classes are composing the other classes which have the distributed functionality. These classes process the requests made by the view through the controller. After

processing Model gives the retrieved results back to controller.

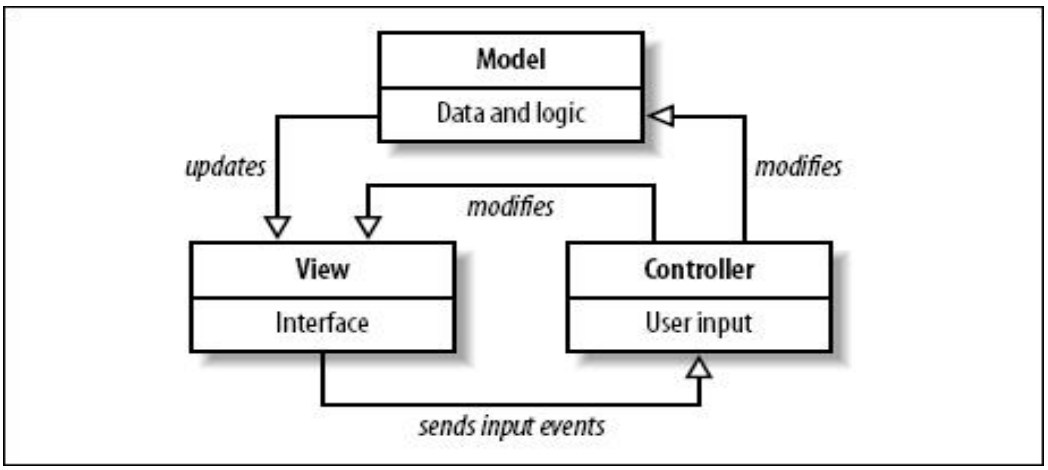

Figure 4- 1: Model View Controller – Generic Interaction

4.4 Detailed Design

4.4.1 Entity Relationship Diagram

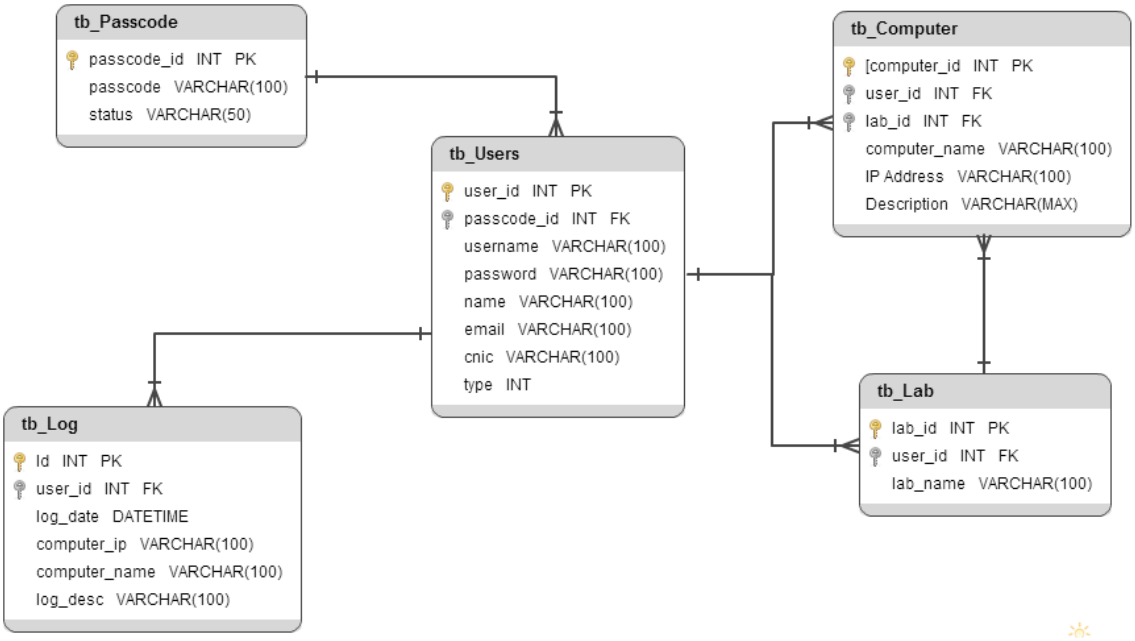

Figure 4- 2: Entity Relationship Diagram

4.5 UML Diagrams

4.5.1 Use-case Diagram

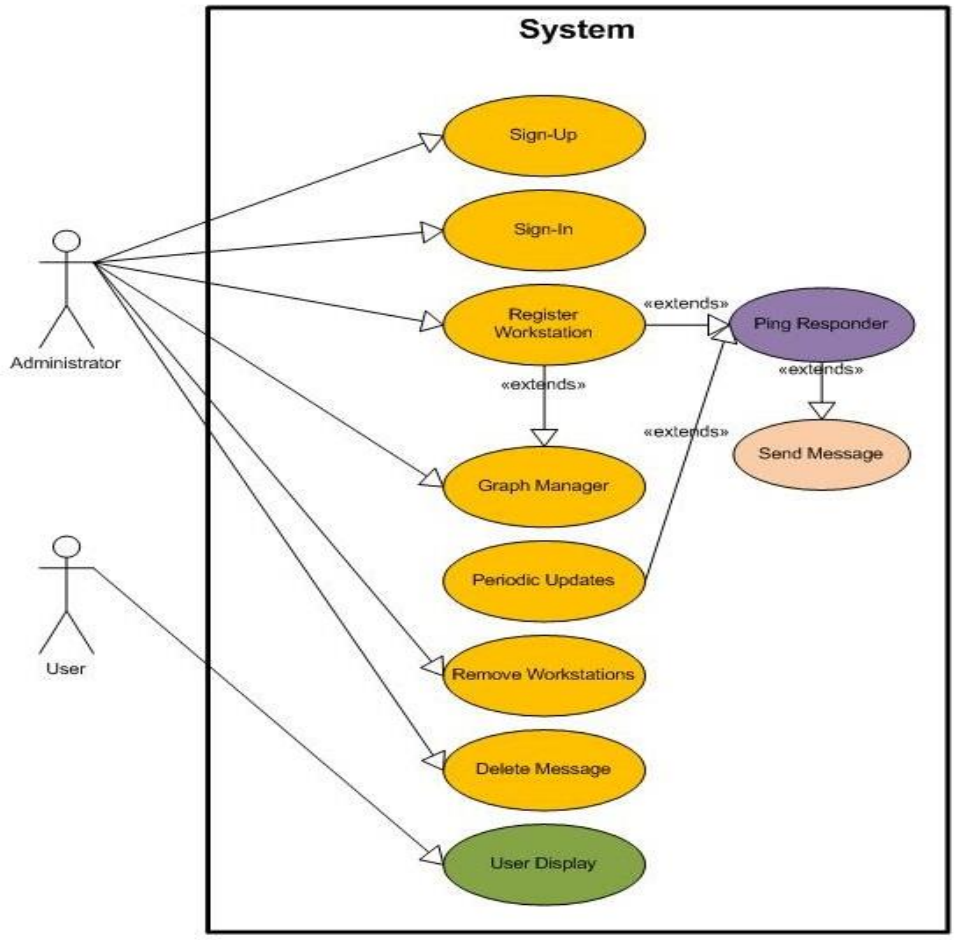

Figure 4- 3: Use-case Diagram

4.5.1.1 Use-case Description

4.5.1.1.1 Sign In

Use Case ID	1
Use Case Name	Sign In

Actors	Administrator		
Created By	Salman Taj	**Last Updated By**	Salman Taj
Date Created	17/11/2013	**Last Updated**	02/12/2013
Description	The actor tries to login to the system.		
Pre-Conditions	The actor has to open the login page first.		
Post-Conditions	If the use case is successful the actor is now logged in to the system, if it is otherwise, the state remains unchanged.		
Normal Flow(Primary Scenario)	The use case starts when an actor wants to login to the system. • The system prompts the actor to enter his/ her user name and password. • The actor enters the required information. • The system first validates the entered information and then loges the user in to the system.		
Alternative Flows	**First Alternative Flow** The use case starts when an actor wants to login to the system. • The system prompts the actor to enter his/ her user name and password. • The actor enters the required information. • The system cannot validate the entered information and generates an erroneous message.		

	• The actor goes back and re-enters his/her credentials. • The system validates the entered information and then loges the user in to the system.

4.5.1.1.2 **Sign Up**

Use Case ID	2		
Use Case Name	Sign Up		
Actors	Administrator		
Created By	Salman Taj	**Last Updated By**	Salman Taj
Date Created	17/11/2013	**Last Updated**	02/12/2013
Description	The actor wants to make an account to facilitate login to the system.		
Pre-Conditions	The actor first has to open a sign up page.		
Post-Conditions	If the use case is successful the system must record the information provided by the new user for future use (logins).		
Normal Flow(Primary Scenario)	The use case starts when the actor clicks on the sign up icon and then enters the specified data in to the system. The system takes the following steps when the actor enters submit. • Checks for the validity of the network passcode provided by the department.		

	• Checks for the availability of the user name. • Check for the correctness of the user password according to the predefined requirements. • The system generates the administrator ID. • The system acknowledges the actor of his/ her successful sign up.
Alternative Flows	**<u>First Alternative Flow</u>** The use case starts when the actor clicks on the sign up icon and then enters the specified data in to the system. The system takes the following steps when the actor enters submit. • Checks for the validity of the network passcode provided by the department. • If the passcode is not correct as contained in the systems database, the request for sign up is rejected. • The actor is provided with an error alert. • The actor can retry the process or leave the system as it is. **<u>Second Alternative Flow</u>** The use case starts when the actor clicks on the sign up icon and then enters the specified data in to the system. The system takes the following steps when the actor

	enters submit. • Checks for the validity of the network passcode provided by the department. • Checks for the availability of the user name. • If the username is already taken the system prompts the actor to enter another username. • Once a valid user name is entered the actor is asked for the password. • If the password does not meet the predefined criteria, he/ she are asked to enter a different password. • The actor enters a new password. • The system generates the administrator ID. • The system acknowledges the actor of his/ her successful sign up.

4.5.1.1.3 **Register Workstation**

Use Case ID	3		
Use Case Name	Register Workstation		
Actors	Administrator		
Created By	Salman Cheema	**Last Updated By**	Salman Cheema
Date Created	17/11/2013	**Last Updated**	02/12/2013

Description	The actor registers new workstation/ workstations to the system.
Pre-Conditions	The actor has to click on the "Register Workstation" icon to proceed with the registration process.
Post-Conditions	If the use case is successful the system must register the new workstation/ workstations and save their statuses to its database.
Normal Flow(Primary Scenario)	When the use case starts the actor clicks on the "Register Workstation" icon on the main page. The following steps take place. • The actor enters the first and the last workstation IP address that is to be registered with the system. • The actor enters submit button. • The system pings each IP address one by one. • A success message is generated by the send message use case (client) acknowledging successful registration of the requested workstation/ workstations.
Alternative Flows	**First Alternative Flow** When the use case starts the actor clicks on the "Register Workstation" icon on the main page. The following steps take place. • The actor enters the first and the last workstation

	IP address that is to be registered with the system. • The actor enters submit button. • The system pings each IP address one by one. • If the send message use case (client) does not respond the IP address is saved in a stack. • On the completion of the cycle the stack IP address are sent with another ping. • The IP addresses which respond this time is removed from the stack. • Remaining IP addresses in the stack undergo the same process one more time. • If there are still IP addresses present in the stack, they are reported to the actor as an error message. • The actor has to physically check for the problem area.

4.5.1.1.4 **Periodic Updates**

Use Case ID	4		
Use Case Name	Periodic Updates		
Actors	Administrator		
Created By	Salman Cheema	**Last Updated By**	Salman Cheema
Date Created	17/11/2013	**Last Updated**	02/12/2013

Description	The client module receives a ping from the server and replies to this ping request.
Pre-Conditions	Server initiates a ping request to each workstation registered to it periodically.
Post-Conditions	If the use case is successful the system updates the data against this workstation in its database.
Normal Flow(Primary Scenario)	When the use case starts the server initiates a ping request for each workstation in a specified order. The following actions take place. • The ping request is received by the client because it is connected to the network. • The client makes a new message for the server and adds the response time, status, system name, IP address and MAC address of the client's workstation to it. • Client sends the message back to the server. • Server receives the message and reads it. • Server updates the data against this workstation in to its database. • The new state is sent to the actor via success message.
Alternative Flows	**First Alternative Flow** When the use case starts the server initiates a ping

	request for each workstation in a specified order. The following actions take place. • The ping request is received by the client because it is connected to the network. • No reply from the client is received by the server. • The server generates an erroneous message to alert the actor. • The actor reads the message and checks the physical connection to the client and puts it right. • The actor resends the ping. • Client receives the ping request. • The client makes a new message for the server and adds the response time, status, system name, IP address and MAC address of the client's workstation to it. • Client sends the message back to the server. • Server receives the message and reads it. • Server updates the data against this workstation in to its database. • The new state is sent to the administrator via success message.

4.5.1.1.5 **Remove Workstation/ Workstations**

Use Case ID	5		
Use Case Name	Remove Workstation/ Workstations		
Actors	Administrator		
Created By	Naveed Aqdus	**Last Updated By**	Naveed Aqdus
Date Created	17/11/2013	**Last Updated**	02/12/2013
Description	The actor selects one or more workstations and clicks on the "Delete Workstations" icon to delete them from the network.		
Pre-Conditions	The actor must first click on the "Remove Workstations" to proceed with the deletion of the workstations.		
Post-Conditions	If the use case is successful the deleted workstation/ workstations data is removed from the system database and they are no more depicted as part of the network.		
Normal Flow(Primary Scenario)	When the use case starts the actor clicks on the "Remove Workstations" icon. The following actions take place after that. • The actor puts the first and the last IP addresses in the "Starting IP" and "Ending IP" slots to show which workstations are to be removed from the network (The administrator may enter the same address in both the fields if only one particular IP address is to be deleted/ alternate option for a		

	single workstation is to select the desired workstation and right click on it and then select the "Delete Workstation" option). • Actor clicks on the "Delete Workstation" icon to remove the workstations from the network. • The system searches the data against these workstations in it database and deletes all the record. • The actor is provided with a confirmation message.
Alternative Flows	**<u>First Alternative Flow</u>** When the use case starts the actor clicks on the "Remove Workstations" icon. The following actions take place after that. • The actor puts the first and the last IP addresses in the "Starting IP" and "Ending IP" slots to show which workstations are to be removed from the network (The administrator may enter the same address in both the fields if only one particular IP address is to be deleted/ alternate option for a single workstation is to select the desired workstation and right click on it and then select the "Delete Workstation" option).

	• The actor clicks on the "Delete Workstation" icon to remove the workstations from the network. • The actor forgets to fill both slots. • Erroneous message is generated prompting the actor to fill both fields. • The actor refills the field's and clicks on the "Delete Workstations" icon to remove the Workstations from the network. • The system searches the data against these workstations in it database and deletes all the record. • Administrator is provided with a confirmation message

4.5.1.1.6 **Ping Response**

Use Case ID	6		
Use Case Name	Ping Response		
Actors	Administrator		
Created By	Naveed Aqdus	**Last Updated By**	Naveed Aqdus
Date Created	17/11/2013	**Last Updated**	02/12/2013
Description	The client module on a workstation confirms to the server that it exists on the system.		

Pre-Conditions	The actor should either register a new workstation or the periodic check for workstation's status initiates a ping request.
Post-Conditions	The client responds successfully to the ping request and makes a new message for the server and adds the response time, status, system name, IP address and MAC address of the client's workstation to it. The message is then returned to the server.
Normal Flow(Primary Scenario)	The actor clicks on the "Register Workstations" icon or the periodic status check initiates ping request. The following steps take place. • The actor puts the starting and ending IP addresses in the "Starting IP" and "Ending IP" slots to show which IP addresses are to be searched from the database (The administrator may enter the same address in both the fields if only one particular IP address is to be searched) or the periodic status check initiates the ping request. • Actor clicks on the "Register Workstation" icon to register the selected workstations or the periodic status check initiates the ping request. • The client responds to the ping request and sends back the message to the server.

	• This prompts the "Send Message" use case which then generates a success message and sends it to the actor's inbox.
Alternative Flows	**<u>First Alternative Flow</u>** The actor clicks on the "Register Workstations" icon or the periodic status check initiates ping request. The following steps take place. • The actor puts the starting and ending IP addresses in the "Starting IP" and "Ending IP" slots to show which IP addresses are to be searched from the database (The administrator may enter the same address in both the fields if only one particular IP address is to be searched) or the periodic status check initiates the ping request. • Actor clicks on the "Register Workstation" icon to register the selected workstations or the periodic status check initiates the ping request. • The client does not respond to the ping request. • This prompts the "Send Message" use case which then generates a failure message and sends it to the actor's inbox.

4.5.1.1.7 **Send Message**

Use Case ID	7		
Use Case Name	Send Message		
Actors	Administrator		
Created By	Salman Taj	**Last Updated By**	Salman Taj
Date Created	17/11/2013	**Last Updated**	02/12/2013
Description	After receiving a ping request reply from the client the server sends a success message to the actor's inbox.		
Pre-Conditions	The server receives a ping request reply from the client.		
Post-Conditions	If the use case is successful the server sends a success message to the actor's inbox and the relevant of that client's workstation is also sent and saved in the database.		
Normal Flow(Primary Scenario)	When the use case starts the server receives a ping request reply from the client. The following steps take place. • The message in the reply is extracted and sent to actor's inbox and also saved in the database.		
Alternative Flows	**First Alternative Flow** When the use case starts the server does not receives a ping request reply from the client. The following steps take place. • An erroneous message is sent to actor's inbox. • The actor physically checks the workstation for		

	connection errors and puts it right. • The actor then sends a ping request to that workstation. • The server receives a ping request reply from the client. • The message in the reply is extracted and sent to actor's inbox and also saved in the database.

4.5.1.1.8 **User Display**

Use Case ID	8		
Use Case Name	User Display		
Actors	User		
Created By	Salman Cheema	**Last Updated By**	Salman Cheema
Date Created	17/11/2013	**Last Updated**	02/12/2013
Description	The server displays the current statuses of each workstation on a GUI for the network actor to view.		
Pre-Conditions	The server must be having the up-to-date statuses of each workstation and then displays it on the user GUI.		
Post-Conditions	If the use case is successful then the actor successfully reads the display and finds an available workstation to work on.		
Normal Flow(Primary	When the use case starts the server check the fresh status of the work stations. The following steps take place.		

Scenario)	• The server sends the data to the actor's GUI for display. • The actor reads the display to find and available workstation to work on. • User occupies the workstation.
Alternative Flows	**<u>First Alternative Flow</u>** When the use case starts the server check the fresh status of the work stations. The following steps take place. • The server fails to sends the data to the user GUI for display. • Old data is displayed on the user GUI. • The actor reads the display to find and available workstation which is actually not available. • The actor reports the matter to the administrator. • Administrator refreshes the system. • The server sends the data to the actor's GUI for display. • The actor reads the display to find and available workstation to work on. • The actor occupies the workstation.

4.5.1.1.9 **Delete Message**

Use Case ID	10

Use Case Name	Delete Message		
Actors	Administrator		
Created By	Naveed Aqdus	**Last Updated By**	Naveed Aqdus
Date Created	17/11/2013	**Last Updated**	02/12/2013
Description	The actor selects one or more messages and clicks on delete icon and removes them from the network.		
Pre-Conditions	The actor selects the messages to be deleted.		
Post-Conditions	If the use case is successful the deleted messages are removed from the network and their data is deleted from the system database.		
Normal Flow(Primary Scenario)	When the use case starts the actor selects one or more messages from the inbox. The following steps take place. • The actor clicks the "Delete Workstation" button. • System deletes the selected messages. • The actor is provided with a confirmation message.		
Alternative Flows	**First Alternative Flow** When the use case starts the actor forgets to selects one or more messages from the inbox. The following steps take place. • The actor clicks the "Delete Workstation" button. • The system alerts the actor to select at least one		

	message to delete. • The actor selects message/ messages and click on delete again. • System deletes the selected messages. • The actor is provided with a confirmation message.

4.5.1.1.10 **Graph Manager**

Use Case ID	11		
Use Case Name	Graph Manager		
Actors	Administrator		
Created By	Salman Taj	**Last Updated By**	Salman Taj
Date Created	17/11/2013	**Last Updated**	02/12/2013
Description	The client responds to the ping request and the result is graphically produced for the user's consumption.		
Pre-Conditions	The actor registers a workstation or the periodic ping request generates a ping request and the client responds to the request.		
Post-Conditions	If the use case is successful the ping response is graphically produced for the consumption of the actor.		
Normal Flow(Primary	When the use case starts the actor registers a workstation or the periodic ping request generates a ping request. The		

Scenario)	following steps take place. • The actor clicks the "Register Workstation" button or the periodic ping request generator sends a ping to each client. • System sends ping request to each client. • The client responds to the ping request. • The result is shown graphically to the actor.
Alternative Flows	**First Alternative Flow** When the use case starts the actor forgets to fill both or either of the starting and ending IP address slots. The following steps take place. • The actor clicks the "Register Workstation" button. • The actor forgets to fill both or either of the starting and ending IP addresses slots. • The system alerts the actor to fill both slots. • The actor again fills the slots and clicks the "Register Workstation" button. • System sends ping request to each client. • The client responds to the ping request. • The result is shown graphically to the actor.

1.9.2 **Sequence Diagrams of key use cases**

4.5.1.2 **Sign-Up**

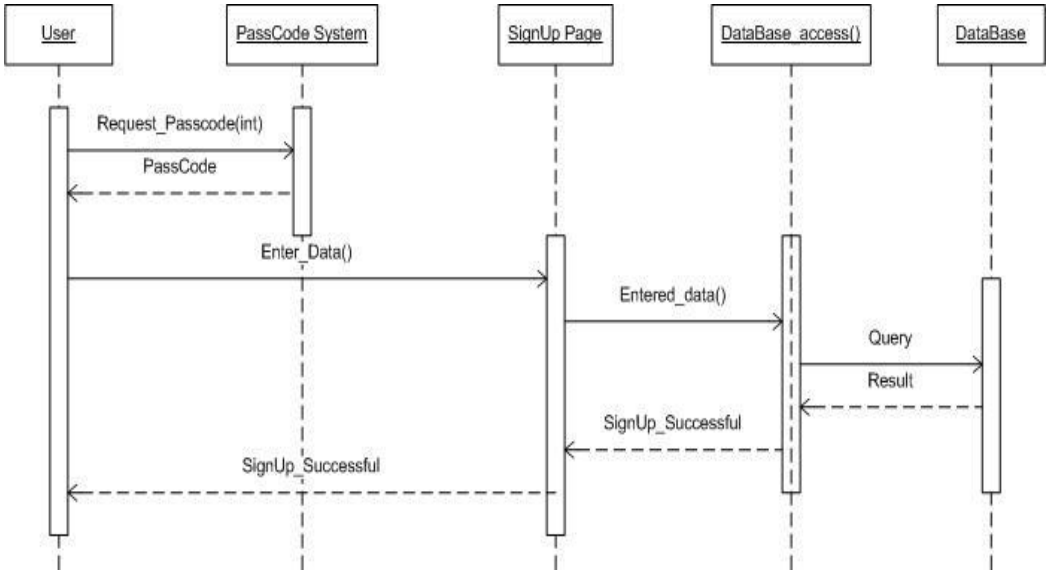

Figure 4- 4: Sign-Up Sequence Diagram

4.5.1.3 **Sign-In**

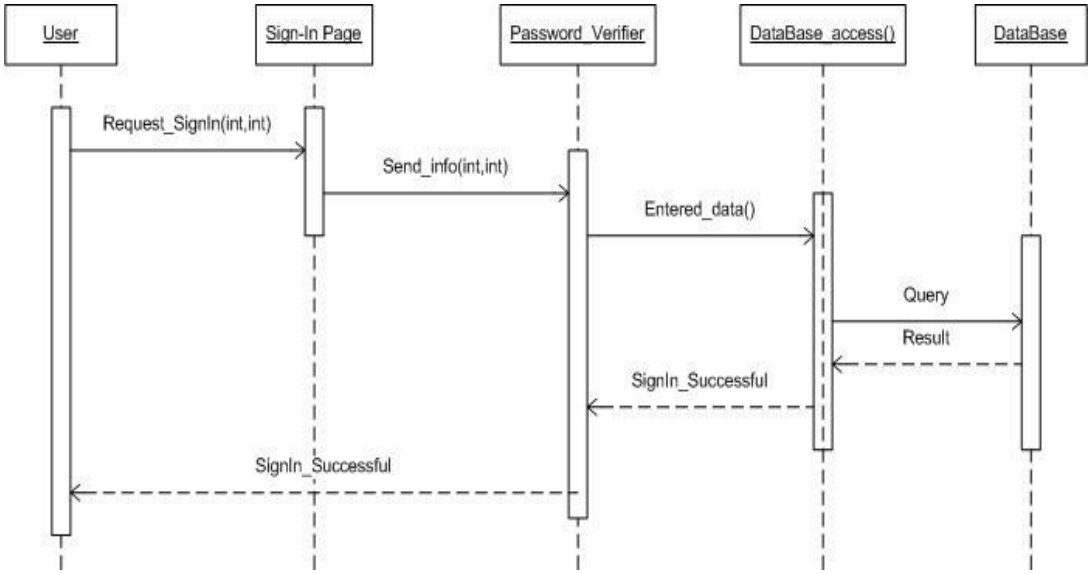

Figure 4- 5: Sign-In Sequence Diagram

4.5.1.4 **Register Workstation**

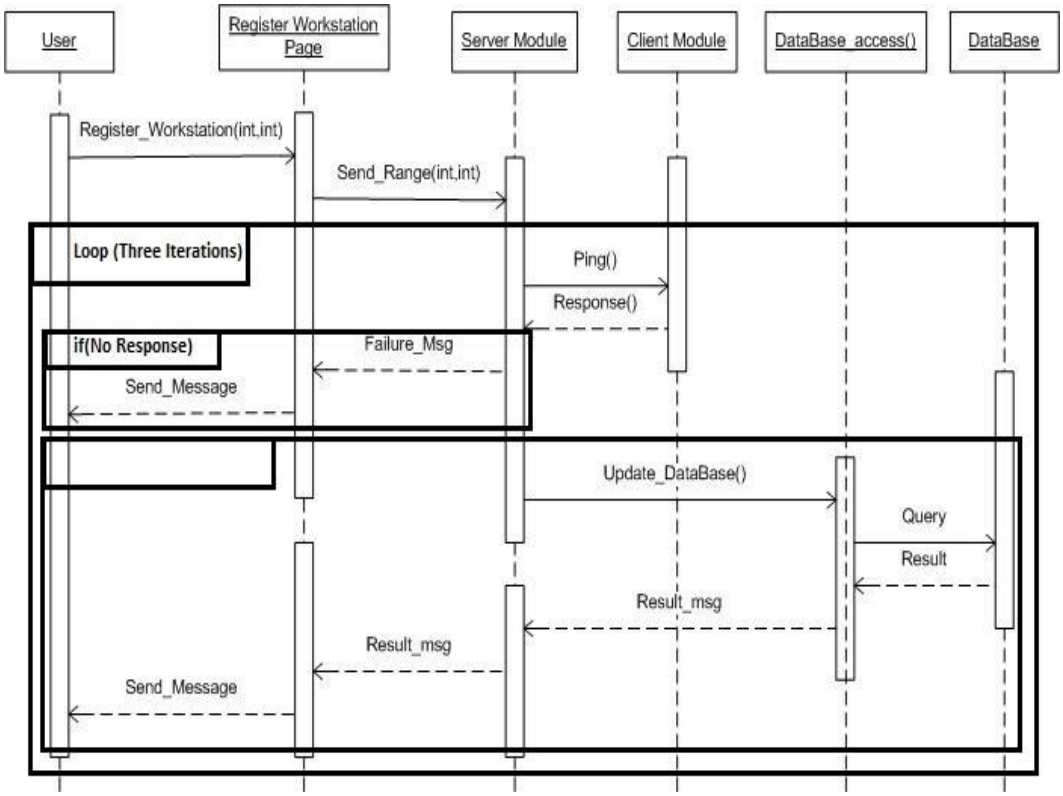

Figure 4- 6: Register Workstation Sequence Diagram

4.5.1.5 **Periodic Updates**

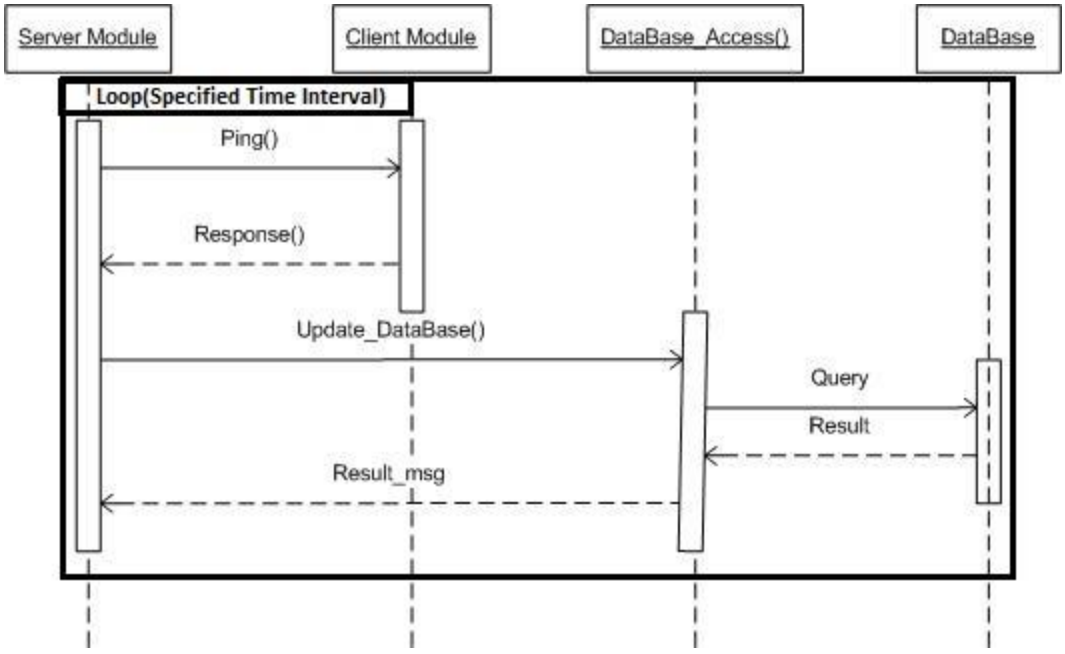

Figure 4- 7: Periodic Updates Sequence Diagram

4.5.1.6 **Graph Manager**

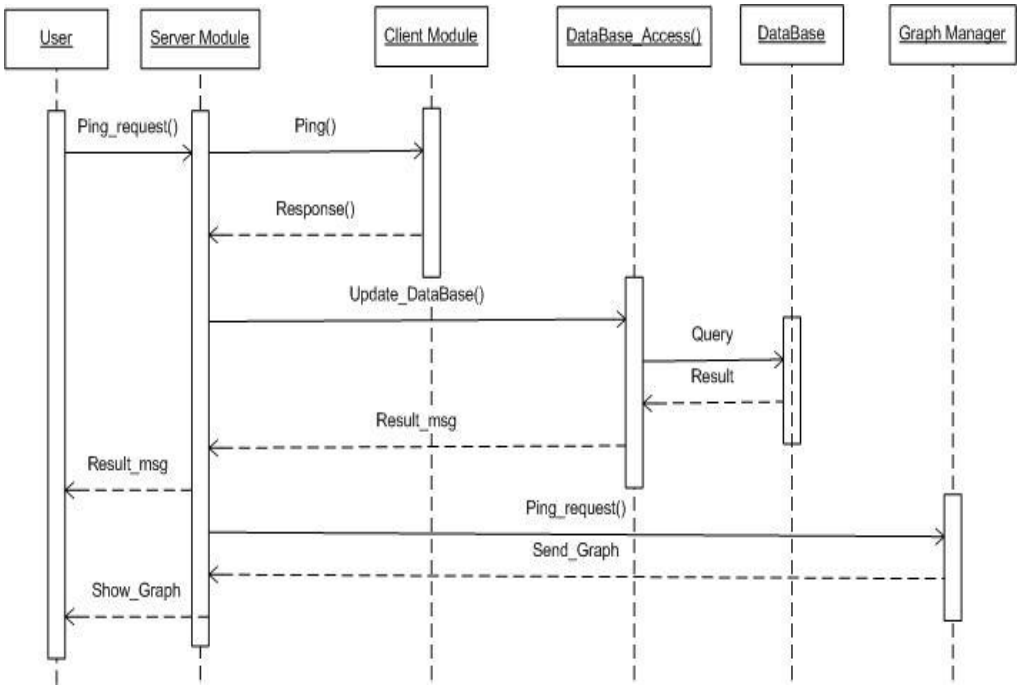

Figure 4- 8: Graph Manager Sequence Diagram

4.5.1.7 **Delete Message**

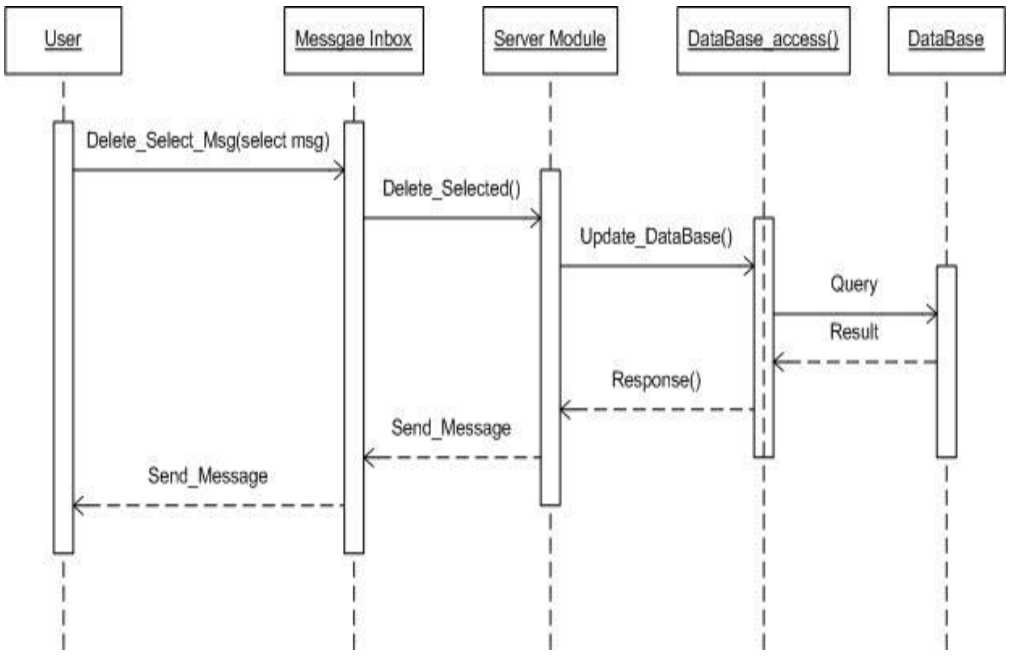

Figure 4- 9: Delete Message Sequence Diagram

4.10 Dynamic View

4.10.1 **Activity Diagrams**

4.5.2.1 **Sign-Up**

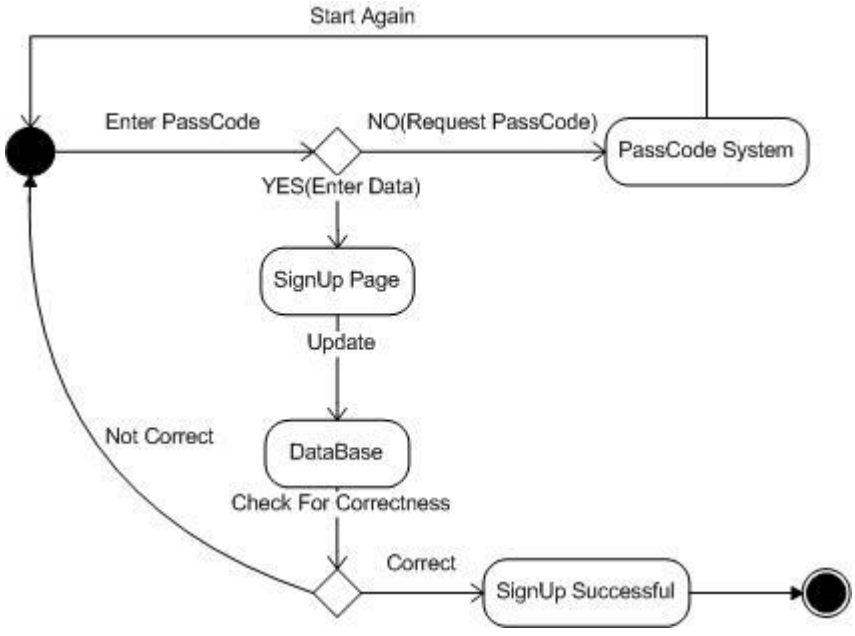

Figure 4- 10: Sign-Up Activity Diagram

4.5.2.2 **Sign-In**

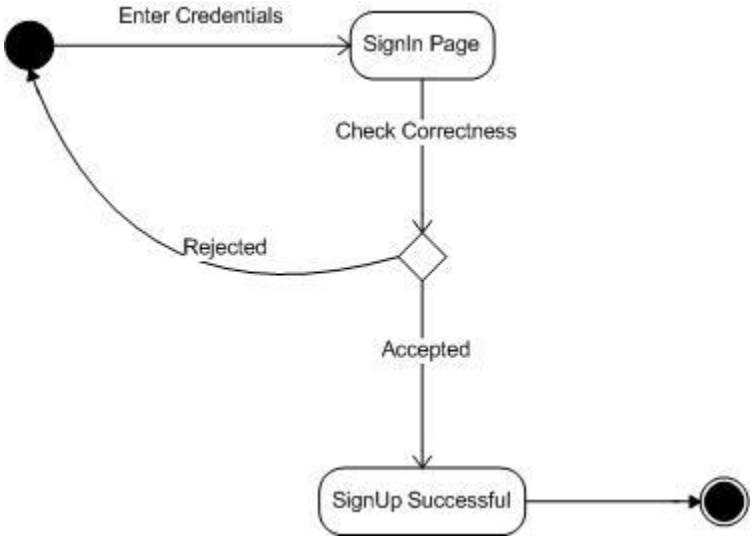

Figure 4- 11: Sign-In Activity Diagram

4.5.2.3 **Register Workstation**

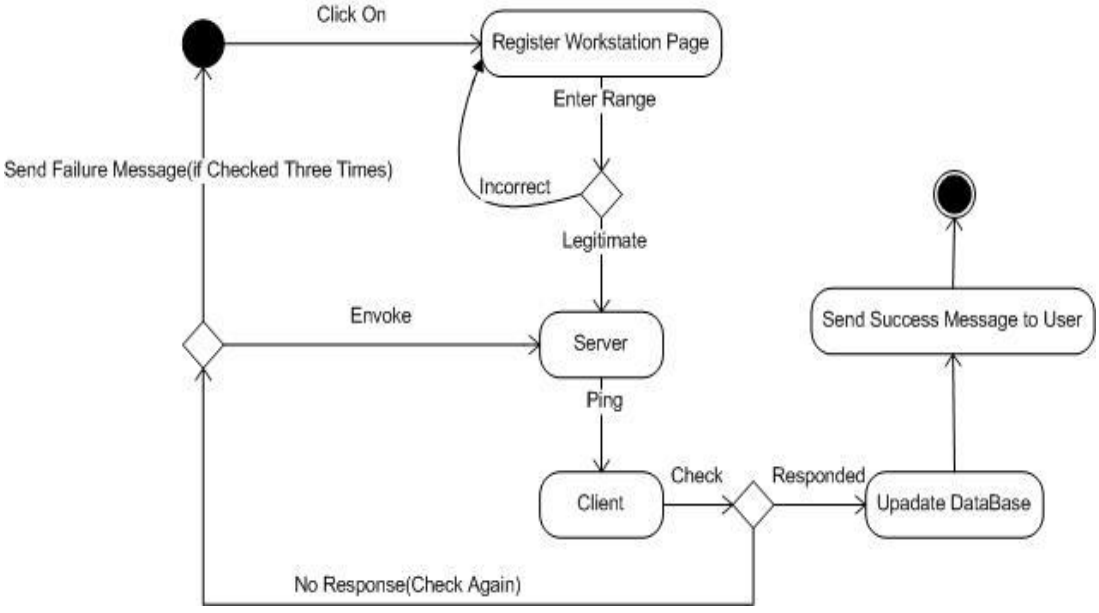

Figure 4- 12: Register Workstation Activity Diagram

4.5.2.4 **Periodic Update**

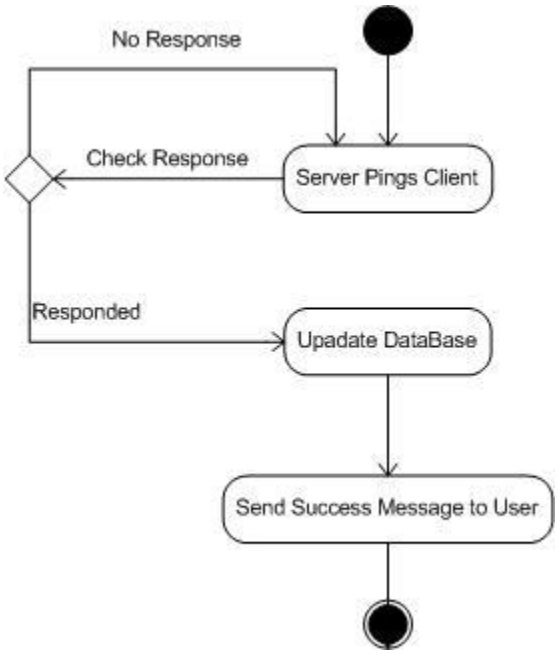

Figure 4- 13: Periodic Updates Activity Diagram

4.5.2.5 **Graph Manager**

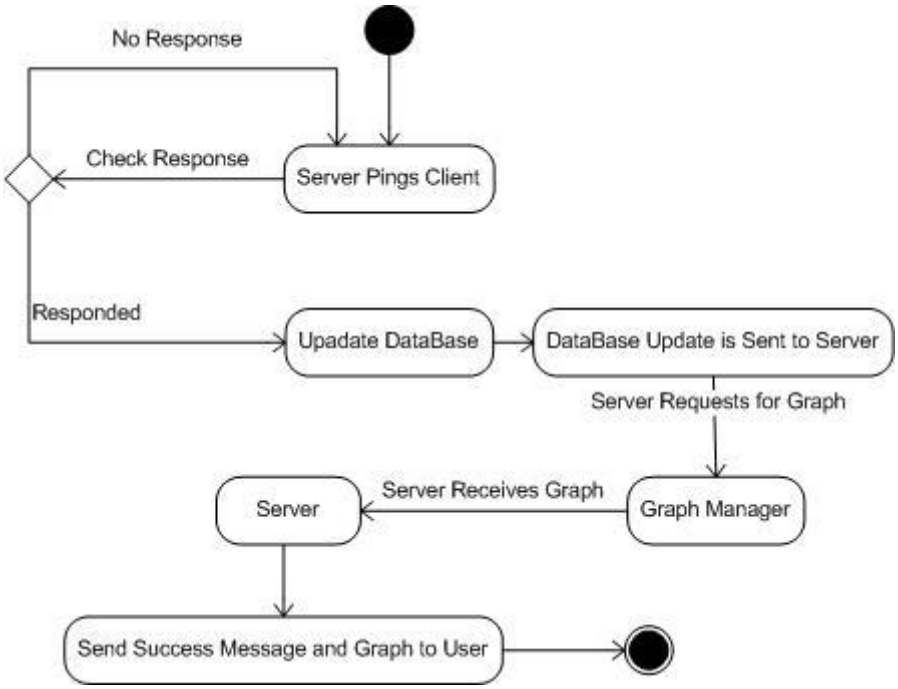

Figure 4- 14: Graph Manager Activity Diagram

4.5.2.6 **Delete Workstation**

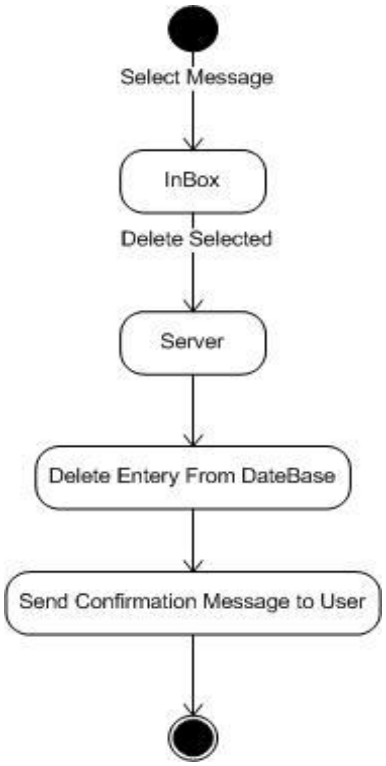

Figure 4- 15: Delete Workstation Activity Diagram

4.11 Implementation View

4.12.1 System Class Diagram

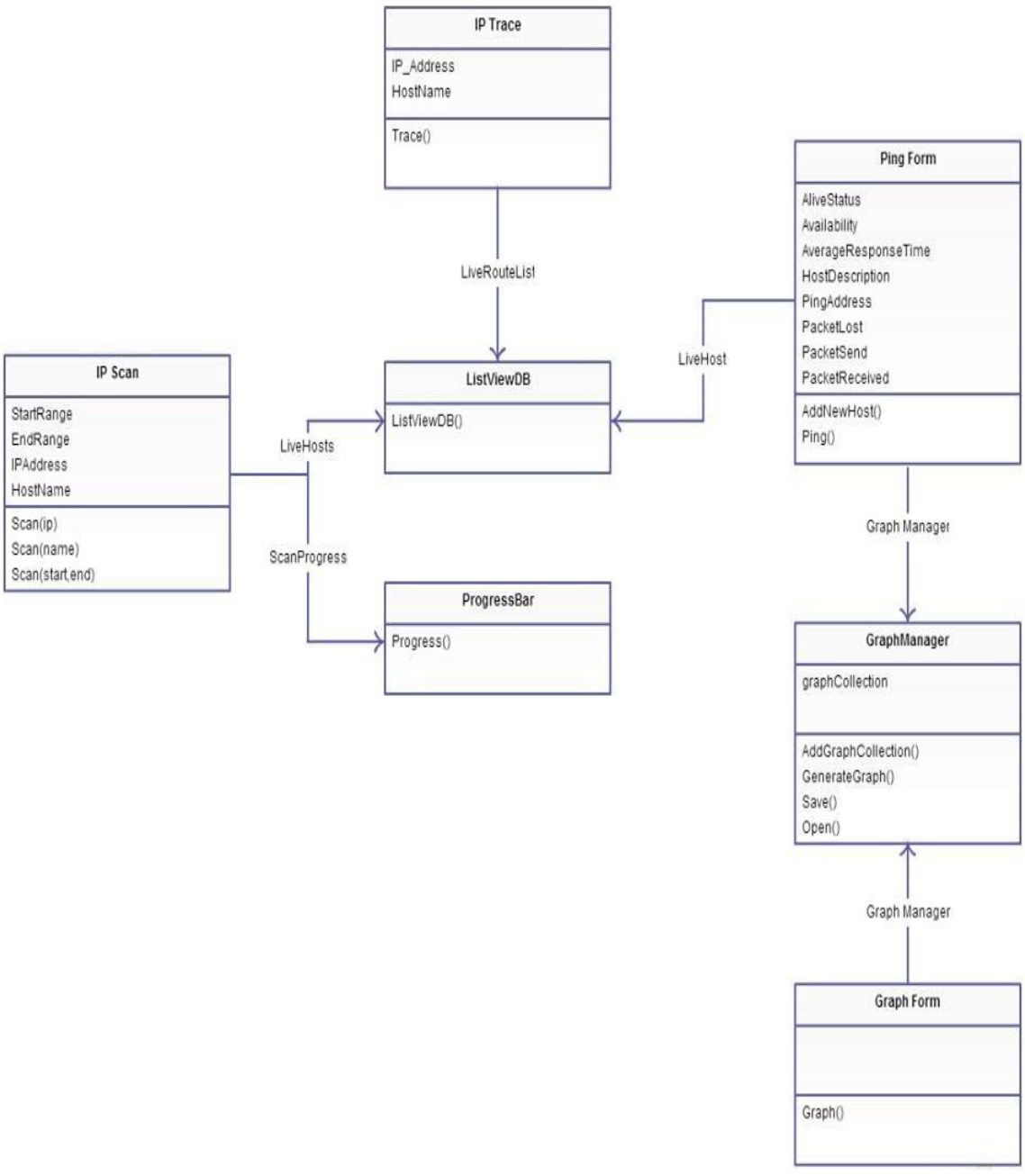

Figure 4- 16: Class Diagram

CHAPTER 5

SYSTEM IMPLEMENTATION

5. System Implementation

5.1 Socket Programming: An Overview

Inter-Process Communication i.e. the capability of two or more physically connected machines to exchange data, plays a very important role in enterprise software development. TCP/IP is the most common standard adopted for such communication. Under TCP/IP each machine is identified by a unique 4 byte integer referred to as its IP address (usually formatted as 192.168.0.101). For easy remembrance, this IP address is mostly bound to a user-friendly host name. The program below (showip.cs) uses the System.Net.Dns class to display the IP address of the machine whose name is passed in the first command-line argument. In the absence of command-line arguments, it displays the name and IP address of the local machine.

Dns.GetHostName() returns the name of the local machine and Dns.Resolve() returns IPHostEntry for a machine with a given name, the AddressList property of which returns the IPAdresses of the machine. The Resolve method will cause an exception if the mentioned host is not found.

Though IPAddress allows identifying machines in the network, each machine may host multiple applications which use network for data exchange. Under TCP/IP, each network oriented application binds itself to a unique 2 byte integer referred to as its port-number which identifies this

application on the machine it is executing. The data transfer takes place in the form of byte bundles called IP Packets or Datagrams. The size of each datagram is 64 KByte and it contains the data to be transferred, the actual size of the data, IP addresses and port-numbers of sender and the prospective receiver. Once a datagram is placed on a network by a machine, it will be received physically by all the other machines but will be accepted only by that machine whose IP address matches with the receiver's IP address in the packet. Later on, this machine will transfer the packet to an application running on it which is bound to the receiver's port-number present in the packet.

TCP/IP suite actually offers two different protocols for data exchange. The Transmission Control Protocol (TCP) is a reliable connection oriented protocol while the User Datagram Protocol (UDP) is not very reliable (but fast) connectionless protocol.

5.2 Client Server programming with TCP/IP

Under TCP there is a clear distinction between the *server process* and the *client process*. The server process starts on a well-known port (which the clients are aware of) and listens for incoming connection requests. The client process starts on any port and issues a connection request.

The basic steps to create a TCP/IP server are as follows:

1. Create a System.Net.Sockets.TcpListener with a given local port and start it:

```
TcpListener listener = new TcpListener(local_port);
listener.Start();
```

2. Wait for the incoming connection request and accept a System.Net.Sockets.Socket object from the listener whenever the request appears:

```
Socket soc = listener.AcceptSocket(); // blocks
```

3. Create a System.Net.Sockets.NetworkStream from the above Socket:

```
Stream s = new NetworkStream(soc);
```

4. Communicate with the client using the predefined protocol (*well established rules for data exchange*):
5. Close the Stream:

```
s.Close();
```

6. Close the Socket:

```
s.Close();
```

7. Go to **Step 2**.

Note when one request is accepted through step 2 no other request will be accepted until the code reaches step 7. (Requests will be placed in a queue or *backlog*.) In order to accept and service more than one client concurrently, steps 2 – 7 must be executed in multiple threads

Or, we can create a client program. Basic steps for creating a TCP/IP client are as follows:

1. Create a System.Net.Sockets.TcpClient using the server's host name and port:

```
TcpClient client = new TcpClient(host, port);
```

2. Obtain the stream from the above TCPClient.

```
Stream s = client.GetStream()
```

3. Communicate with the server using the predefined protocol.
4. Close the Stream:

```
s.Close();
```

5. Close the connection:

```
client.Close();
```

CHAPTER 6

TESTING AND EVALUATION

6. Testing

Testing any software project/product/program is essential to check and ensure the provision of intended functionality and quality of software product. We have tested our software product on two levels:

- Interface Testing
- Functional Testing
    - Test case driven Testing
    - Application Testing

6.1 Interface Testing

Test Case 1

Test Case ID	01
Test Case name	Ping all computers button testing
Input(s)	Press the ping all button
Output	Starts pinging to all computers that are present in the current lab. And check their response.
Sequence of Action(s)	Press Ping all button from menu strip of Application.
Result	Success

Table 2: Test Case 1

Test Case 2

Test Case ID	02
Test Case name	Continuous Preview testing
Input(s)	Computer present in the Grid.
Output	Continuously pinging and show their results in Grid
Sequence of Action(s)	Click the ping all button to start pinging to computer.
Result	Success

Table 3: Test Case 2

Test case 3

Test Case ID	03
Test Case name	IP Scan button testing
Input(s)	Click on IP scan button
Output	The computer that are present in the current ip range and provide response, will show with their signal status.
Sequence of Action(s)	Select the ip range or ip subnet and click on Ip Scan button to show the computers that are within the range and provide response.
Result	Success

Table 4: Test Case 3

Test Case 4

Test Case ID	04
Test Case name	Live Graph testing
Input(s)	Computer ping status
Output	Graph is shown having time at x-axis and lost packet, packet sent etc values at y-axis.
Sequence of Action(s)	Click the graph button and choice computer to show graph of it.
Result	Success

Table 5: Test Case 4

Test Case 5

Test Case ID	05
Test Case name	User's Log Testing
Input(s)	User do ping to computer
Output	Log of that activities are maintain against for user.
Sequence of Action(s)	User try to do ping and log is shown in log table.
Result	Success

Table 6: Test Case 5

Test Case 6

Test Case ID	06
Test Case name	Passcode button testing
Input(s)	Press Passcode button
Output	Form having passcodes is shown
Sequence of Action(s)	Press the passcode button and a form is shown having passcodes in it.
Result	Success

Table 7: Test Case 6

Test Case 7

Test Case ID	07
Test Case name	User View setup Testing
Input(s)	Choice the lab and orientation for the lab.
Output	Lab is created in GUI for having computers in it.
Sequence of Action(s)	Click on UserViewSetup button, a form is shown having some input values for the user, user choices the lab name and orientation and a GUI is shown.
Result	Success

Table 8: Test Case 7

Test Case 8

Test Case ID	08
Test Case name	Client Module is connected (User is logged in)
Input(s)	Client Module is connected
Output	Red computer is shown in server GUI against the client having the same ip as the client has.
Sequence of Action(s)	Server is listening the client response when the client is connected, screen of that computer is shown in red state.
Result	Success

Table 9: Test Case 8

Test Case 9

Test Case ID	09
Test Case name	Client Module is disconnected (User is logged in)
Input(s)	Client Module is disconnected
Output	Green computer is shown in server GUI against the client having the same ip as the client has.
Sequence of Action(s)	Server is listening the client response when the client is disconnected, screen of that computer is shown in green state.
Result	Success

Table 10: Test Case 9

Test Case 10

Test Case ID	10
Test Case name	Create New lab testing
Input(s)	User clicks on new lab button and type the lab name
Output	Lab is added against the user that is logged in.
Sequence of Action(s)	User clicks on new lab button and type lab name in the input form after that the lab is added against the user that is currently online.
Result	Success

Table 11: Test Case 10

CHAPTER 7

CONCLUSION AND FUTURE WORK

7. Conclusion & Future Work

Networks are employed at all levels and their administration is a point of concern for all administrators. Efficient monitoring of networks makes the working environment more professional and effective. Tools such as Network Monitor can come in handy in a situation like the one stated. If an administrator knows exactly what is going on in his network while sitting on his system, it will ensure uninterrupted connectivity of the resources to the network. On the other hand, with software like the one presented here it will make the life of a user easier. In institutions such as colleges and universities, finding a workstation that is for sure online is at times an issue. Users try multiple systems before finding such a workstation. This software and its future versions will ensure that such issues are addressed. For our future work we also intend to develop a smart phone application for network monitoring to ease the life of network administrators and users.

One major issue with this version that we have developed is that despite the fact that we have employed TCP; still the ping sender is consuming a lot of bandwidth. This slows the overall working speed on the network and as a result becomes a hurdle for the users. We are planning to reduce the need of ping even more and at some level we are planning to study other protocols and totally replace the ping sender. This would rectify this problem and a better working environment would prevail as a consequence.

REFERENCES

[1] http://www.compnetworking.about.com/od/workingwithipaddresses/f/ip_conflict.htm

[2] http://en.wikipedia.org/wiki/Client%E2%80%93server_model

[3] http://www.paessler.com/network_monitor_software

[4] http://www.spiceworks.com/

[5] http://www.sevone.com/