NiS – A NETWORK MONITORING TOOL

By

Muhammad Ertza Afzal

Nasir Hussain

Subtain Ishfaq

Supervisor

A/P Waseem Iqbal

Submitted to the Faculty of Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfilment for requirements of Degree of B.E in Software Engineering

JUNE 2016

CERTIFICATE FOR CORRECTNESS AND APPROVAL

Certified that work contained in the thesis “NiS – A Network Monitoring Tool” carried out by Muhammad Ertza Afzal, Nasir Hussain and Subtain Ishfaq under supervision of A/P Waseem Iqbal for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

A/P Waseem Iqbal

ABSTRACT

NiS – A Network Monitoring Tool

Network monitoring is a difficult and demanding task that is a vital part of a Network Administrators job. If a network were to be down even for a small period of time productivity within a company would decline. In order to be proactive rather than reactive, administrators need to monitor traffic movement and performance throughout the network and verify that security breaches do not occur within the network.

Network monitoring is a very important aspect of looking after and running any computer network with any significant number of nodes and used to help identify problems before users of that network are impacted and to highlight the specific fault or problem and where it is occurring for quick diagnosis and ensuring a fast transition from the stage of identifying there is a problem to working out where that problem lies and fixing the underlying cause.

Systems Administrators and Network Administrators alike currently use a combination of various systems to meet their monitoring needs, ranging from graphing software to server monitoring software to network measurement software all in conjunction to provide enough visibility into a network to monitor it effectively when these needs should be met by one system.

The project produced is a modular system that can be extended to handle the different types of monitoring a systems/network administrator might need to look after their network in a unified fashion.

NiS successfully discovers the devices in the network automatically, maintains the database and logs of connected devices, scans the network for any changes, gets specific device parameters, graphically plots the network keeping its topology in mind and shows the bandwidth utilization of the devices. If any node is down, the system immediately updates itself and removes that specific system from the topology.

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

This project is dedicated to our parents, siblings and our encouraging supervisor without whose encouragement and support we would not have been able to complete this project.

ACKNOWLEDGEMENTS

First of all, we are thankful to **ALLAH (SWT)** Himself and we feel blessed that He guided us and provided with enough strength that we completed this project well in time.

We extend our sincerest gratitude to **A/P Mian M Waseem Iqbal** for his continuous guidance and persistent support. Sir, you are an exceptional supervisor and without you we could not have come this far. Thank you very much for everything you did for us Sir.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

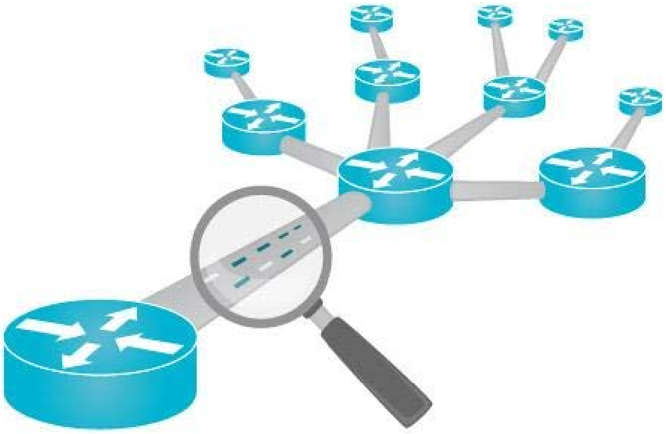
Network monitoring is a very important aspect of looking after and running any computer network with any significant number of nodes. It is used to help identify problems before users of that network are impacted; and to highlight the specific fault or problem and where it is occurring for quick diagnosis, ensuring a fast transition from the stage of identifying there is a problem to working out where that problem lies and fixing the underlying cause.

The purpose of our project is to produce a modular system that can be extended to handle the different types of monitoring a systems/network administrator might need to look after their network in a unified fashion. It includes:

1. Auto Device Discovery
2. Graphical View of Network Topology
3. Maintaining logs of devices in the network
4. Get device name and type of device
5. Displaying IP and MAC addresses of all devices
6. Visually displaying bandwidth usage of devices

The application will have a wide diversity of users which includes Network Administrators, System Administrators, general public, and anyone who has access to a PC on a network.

Figure 1-1: Network Monitoring



1.1 Problem Statement

Network Administrators are constantly striving to maintain smooth operation of their networks. If a network were to be down even for a small period of time productivity within a company would decline, and in the case of public service departments the ability to provide essential services would be compromised. Our aim thus, is the implementation of a dynamic network topology finder as a network monitoring system which will help the administrators with their job.

1.2 Scope

Table 1-1: Scope of Project

For	Network Administrators or anyone trying to have an insight in their network
What	Auto Discovery of Devices, Getting specific device parameters, Visual display of network with its topology, Bandwidth usage by devices
The	NiS – A Network Monitoring Tool
Is	A Linux Desktop Application
That	Provides convenient, rapid and accurate monitoring of network

NiS – A Network Monitoring Tool shall provide a web based GUI that would run on top of a C/C++ program on Linux operating system. It shall be satisfying the following objectives:

1. Auto Device Discovery

2. Graphical View of Network Topology

3. Maintaining logs of devices in the network

4. Get device name and type of device

5. Displaying IP and MAC addresses of all devices

6. Visually displaying bandwidth usage of devices

1.3 Objectives

The objectives of the project include:

1. To learn and implement SNMP based solutions
2. To implement the algorithms of fast auto discovery of devices
3. To implement the accurate picture portraying of network's topology
4. To develop a user friendly interface
5. To take care of all of a Network Administrators' needs of monitoring.

1.4 Deliverables

Table 1-2: Project Deliverables

Sr.	**Tasks**	**Deliverables**
1	Literature Review	Literature Survey
2	Requirements Gathering	SRS Document
3	Application Design	Design Document
4	Implementation	Implementation of the application on an android device (mobile/tablet), Test Plan
5	Testing	Evaluation plan and test document
6	Training	Deployment Plan
7	Deployment	Complete application along with necessary documentation

1.5 Achievements

1. Industrial Project
2. FICS
3. Sponsored by NESCOM

Chapter 2: Literature Review

2.1 Introduction

Every organization now-a-days has a network, and network administrators to ensure that their networks are up and running as well as working correctly. For this, they make use of a number of network monitoring tools. Our application is one of those such tools.

A common person, who knows how to use a computer will be able to use the application and will be able to familiarize himself/herself with the system within 20 minutes at most.

2.2 Background

The problem network monitoring is particularly more in large organizations but is also applicable to anywhere where networks are a part of daily life. The problem increases more when the network we need to monitor is bigger in size thus it becomes difficult to detect and furthermore locate the exact location of the issue. This is the reason why in almost every organization, administrators are using a number of network monitoring tools to help cater for this issue. Our team has come up with a novel idea of providing these people with one system that is specifically designed to meet all of their needs.

2.3 Related Work

Some links that show the work done in the field of network monitoring are:

[1] It refers to the Wiki page of SNMP and describes the overview of the protocol. SNMP stands for Simple Network Monitoring Protocol and is widely used to get device parameters of devices in the network by sending by SNMP requests and receiving their responses.

[2] This website is about an organization called Paessler which offers easy to use network monitoring tools for every network. They call themselves 'The Network Monitoring Company'. Their product PRTG is industry standard and is a great tool.

[3] This product is considered the best graphing solution for network monitoring.

[4] This is a comprehensive list of the top 20 network monitoring solutions.

2.4 Proposed Project

NiS – A Network Monitoring Tool will be a complete application that will take care of almost all of a network administrator's needs. It will provide the following functionality:

1. Auto Device Discovery
2. Graphical View of Network Topology
3. Maintaining logs of devices in the network
4. Get device name and type of device
5. Displaying IP and MAC addresses of all devices
6. Visually displaying bandwidth usage of devices

2.5 Summary

There hasn't been a single application, till date that incorporates all of the common network monitoring needs in one suite. We intend to develop an application that will cater most of the network monitoring needs so that maximum people can make use of it. All of the aforementioned tools are missing something in each of them, some have auto discovery, some provide bandwidth usage information, but none of them provides all those features in one solution and neither one of them incorporates topology discovery in them.

Chapter 3: Software Requirement Specification

3.1 Introduction

3.1.1 Purpose

The purpose of this document is to describe the project titled “NiS”. This document contains the functional and non-functional requirements of the project. It also contains the guidelines for developers and examiners of the project.

3.1.2 Intended Audience and Reading Suggestions

This document is primarily intended for the evaluators and the supervisor of the project. This document will provide guidelines to system developers and testers. Any third party who needs a basic understanding of the system may find this document helpful.

3.1.2.1 Examiners/Evaluators

The document will provide the FYP evaluators with the scope, requirements and details of the project to be built. It will also be used as basis for the evaluation of the implementation and final project.

3.1.2.2 Developers

The document will provide guidance to the developers to determine what the requirements are and how they should continue with the project.

3.1.2.3 Project Supervisor

This document will be used by the project supervisor to check whether all the requirements have been understood and in the end whether the requirements have been implemented properly and completely.

3.1.2.4 Project Testers

Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. It will help in building up test cases for the testing process. This way testing becomes more methodically organized.

3.1.2.5 Up gradation Engineers

Up gradation engineers can review projects capabilities and more easily understand where their efforts should be targeted to improve or add more features to it. It sets the guidelines for future developments.

3.1.2.6 End Users

This document can be read by the end users if they wish to know what the project is about and what requirements have been fulfilled in this project.

3.1.3 Product Scope

The scope of the project is to monitor and analyze the network and the end systems connected to it using SNMP protocol. The end system information and status information are then interpreted to usable information for administrator for security and maintenance purpose.

3.2 Overall Description

3.2.1 Product Perspective

The product is a network monitoring tool for government organization which is developed according to the specified requirements related to the domain of the organization. The aim of this project is to develop an application for monitoring end systems connected to network under consideration. The product is based on distributed application architecture to gather the end-system information using SNMP protocol and other standard monitoring practices. The information is gathered by distributed clients on the networks which are targeted by administrator. This information gathered by client-end applications will be posted to the administrator system using web services architecture (client-server approach). As we are using web-services architecture for administrator application so the multiple platforms applications are being developed for administrator to monitor the information of end-system. Using centralized server where the data is posted and the API for the server we are able to develop website and desktop application for administrator. The information of the end-systems will enable administrator to analyze the progress, state and security of the end-systems.

3.2.2 Product Functions

1. SNMP Monitor web based
2. RESTful API
3. Packet Monitor
4. Output of network in the form of visualized topology

3.2.3 User Classes and Characteristics

Following are the user classes and their brief description.

3.2.3.1 Administrator

Administrator will use this Project as a monitor the progress and status of end system as well as maintenance. This project will help administrator to identify the vulnerabilities in the network.

3.2.3.2 Tester

Tester will also use this project to test the end-system response and developing networks.

3.2.3.3 Project Evaluator/Supervisor

Project supervisor/Evaluator will also use the product to evaluate. They will use this product to analyze the monitor functionality according to the set standards.

3.2.4 Operating Environment

The operating environment required for this project is:

3.2.4.1 Hardware requirements:

Remote server: To post the collected information of all the networks
Desktop Computer: To install client-end or administrator application.

3.2.4.2 Software requirements:

PHP and MySQL Server: To host the API, Website and UI.
OS: Windows 7, 8, 8.1, Linux

3.2.4.3 Design and Implementation Constraints

Following are the constraints of design and implementation in our project:

3.2.4.3.1 SNMP Service Unavailability

If SNMP service is not installed (or enabled) on end-system, the monitoring tool cannot access any information present in MiB.

3.2.4.3.2 Ping inability to find host

Due to the firewall or user privileges ping is not able to find host sometimes which can cause undetected end-system.

3.3 User Documentation

For the user documentation, a user manual will be provided with the system. It will include the details of the system’s working. Help documents will also be a part of the system. The project report will also be available for the users which will highlight the system features, working and procedures.

3.4 Assumptions and Dependencies

1. Overall performance of this project will depend on bandwidth of the network.
2. Traffic control during monitoring is important factor in the success of this project.
3. It will also depend on the response time of the end-systems.

3.5 External Interface Requirements

3.5.1 User Interfaces

1. The first UI screen will gather the information of available devices in the network and will display it.
2. Clicking on the end-system icon will display further details.
3. Clicking on Speed Graph on further next window will open up the speed graph.

Figure 3-1: User Interface I

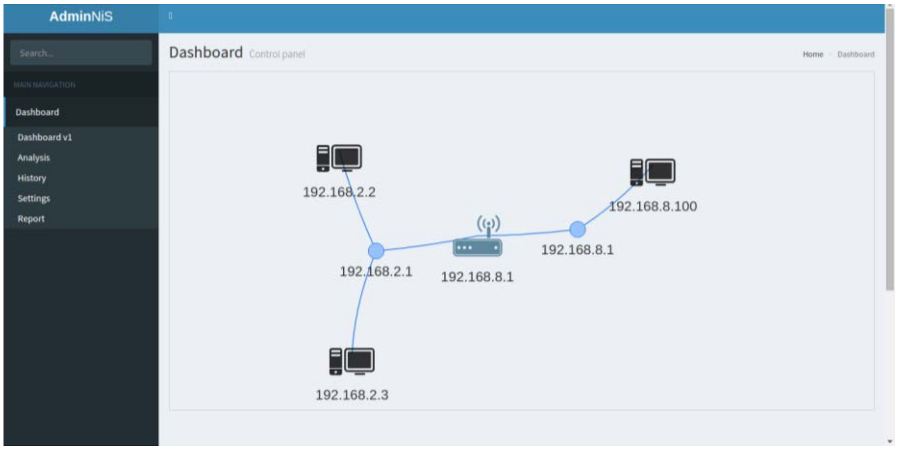
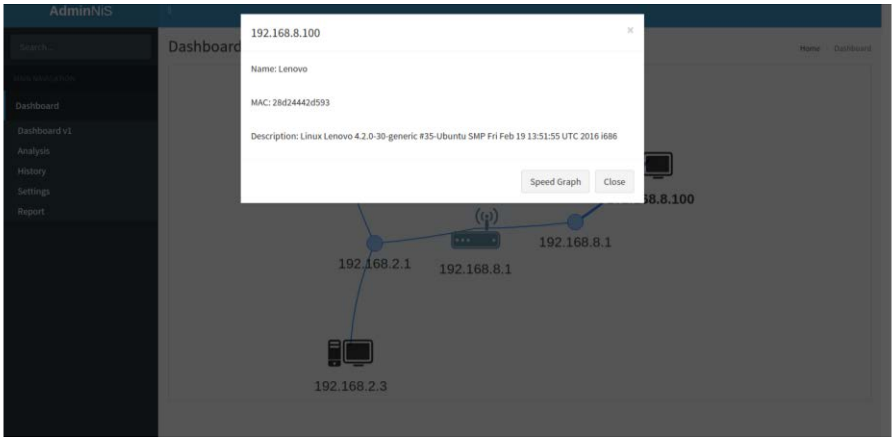


Figure 3-2: User Interface II



192.168.8.100

Name: Lenovo

MAC: 28d24442d593

Description: Linux Lenovo 4.2.0-30-generic #35-Ubuntu SMP Fri Feb 19 13:51:55 UTC 2016 i686

Figure 3-2: User Interface III

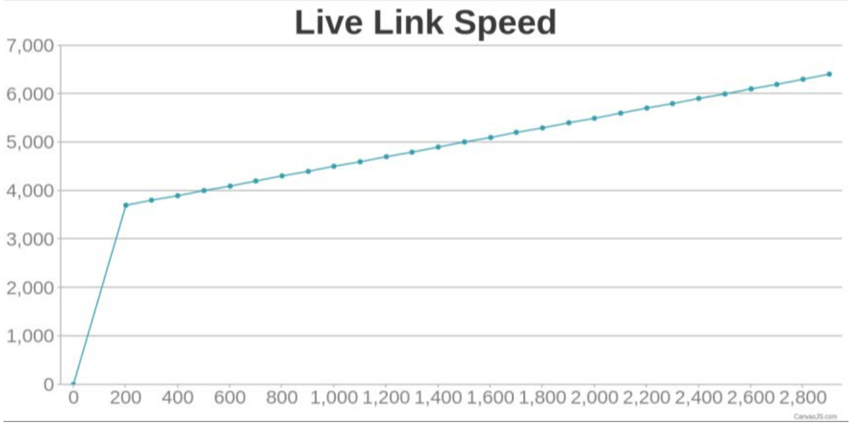
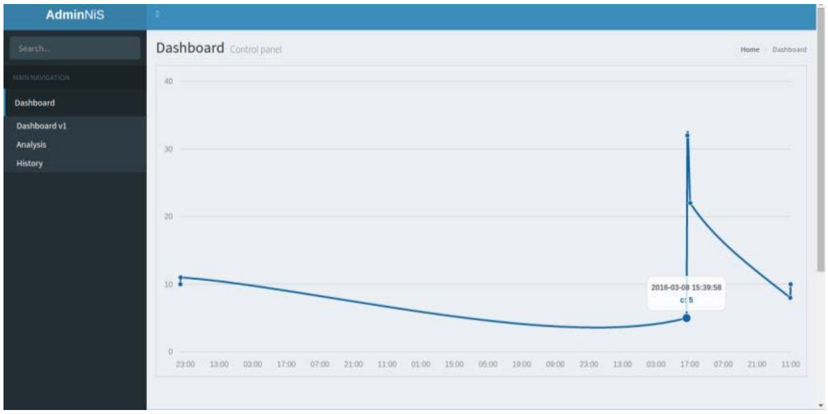


Figure 3-2: User Interface IV



3.5.2 Hardware Interfaces

The communication between end-system and administrator will be through the network.

3.5.3 Software Interfaces

SNMP Service will be use to enable the software interfacing of network monitoring tool.

1. SNMP Monitor web based
2. RESTful API
3. Packet Monitor
4. Output of network in the form of visualized topology
5. Bandwidth Monitor
6. Logs

3.6 System Features

3.6.1 Auto Discovery

The system shall automatically discover all the devices connected in the network.

3.6.2 Device Identification

The system shall identify all the devices discovered i.e. whether it is a switch, router, firewall, computer etc.

3.6.3 Device Parameters

The system shall also collect a number of device parameters (as selected by user) e.g. information about the network card, information about the processor etc.

3.6.4 Topology

The system shall find the topology of the network to display it graphically.

3.6.5 Web Services

The monitoring interface shall be a web based interface accessible through any device in the network.

3.6.6 Centralized Database

The system shall have a centralized database accessible to the administrator through the web interface.

3.7 Other Non-Functional Requirements

- System will have PHP for server side scripting.
- System will have C++ for the client end.
- System will have bootstrap admin/client implemented.
- System will use SNMP v2c
- System will use REST API.
- System will be distributed in nature.
- System will not need maintenance for at-least one year.
- System will provide authentic data.
- System will not take more than 1 minute to detect a new device in the network.
- A new user will not require more than half hour to get trained in using the system.

Chapter 4: Design and Development

4.1 Introduction

This document shows the design, components and working of ‘NiS’. It starts from higher level details for a non-technical reader to the lower level details that aid the developer to code and understand other technical details of the application.

Section 1 is Introduction to the document with a general description of what a Design Document is and what is the project whose SDD is this document.

Section 2 the System Architecture Description gives a detailed overview of the application. Section 2.1 Overview of Modules/Components overviews the main components of the application and their inter-relationships. Section 2.2 Structure and Relationships shows the higher level details system working by the means of System Block, Activity, State Transition, and Use Case diagrams. Lower level details are described using the Class, Chen’s Entity Relationship, Sequence diagrams and Structure Chart. Section 2.3 describes how the application is designed to curb the tendency of User Interface Issues and problems during User Interaction.

In Section 3, Detailed Description of Component is given to show the working of modules with low level details. It shows the purpose, function, subordinates, dependencies, interfaces, resources, processing and data of the components and their relationships with each other.

Section 4 shows the Reuse and Relationship to other Products i.e.; information about work done in the same project before and any reuse of the same work. The section also provides a key to reuse this system for further upgrades.

Section 5 Design Decisions and Tradeoffs shows the architecture style and design pattern of the application, while the Pseudo Code of the components is given in Section 6 for human reading.

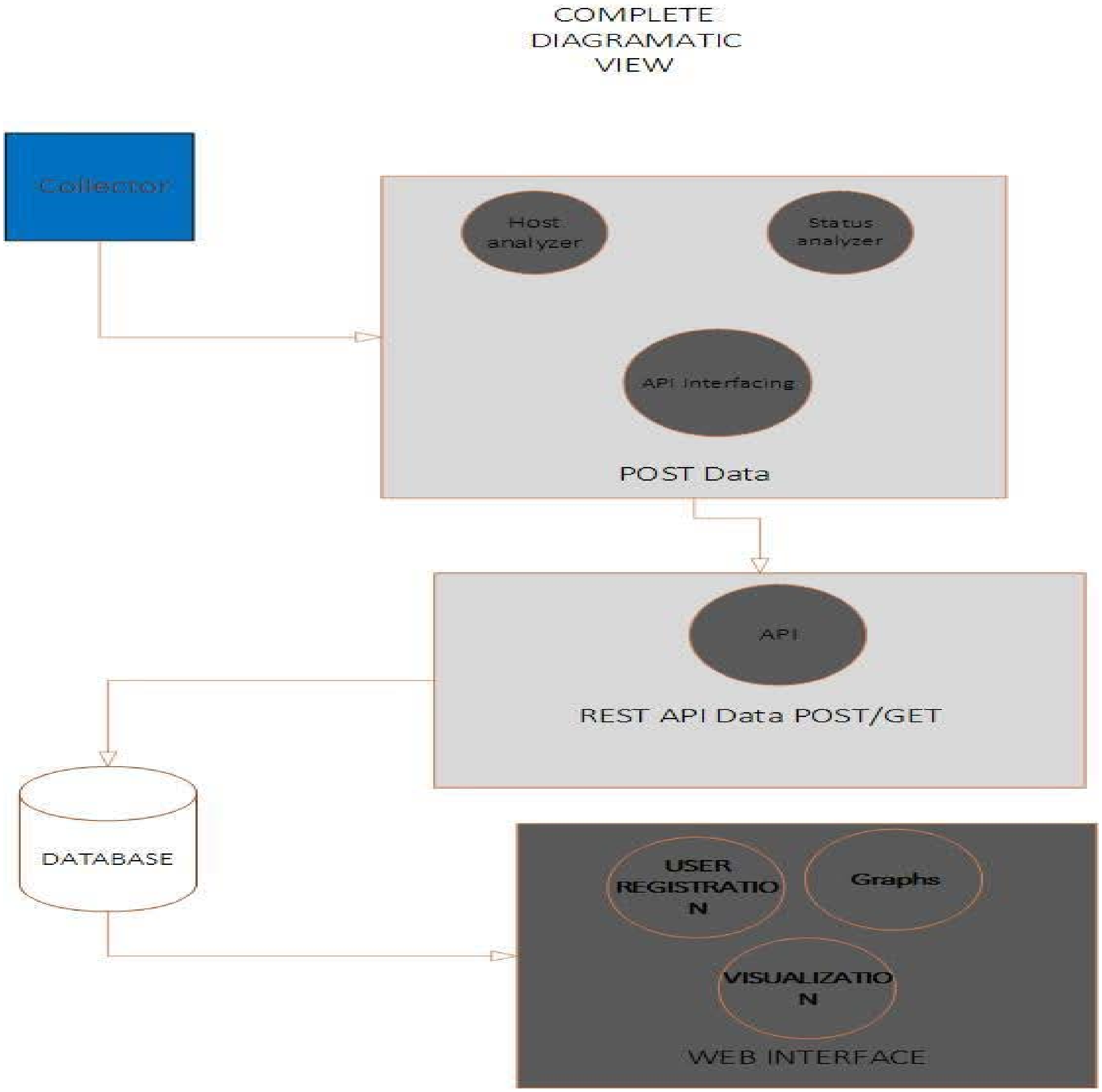
4.2 System Architecture Description

4.2.1 Overview of Modules

The product is a network monitoring tool for government organization which is developed according to the specified requirements related to the domain of the organization. The aim of this project is to develop an application for monitoring end systems connected to network under consideration. The product is based on distributed application architecture to gather the end-system information using SNMP protocol and other standard monitoring practices. The information is gathered by distributed clients on the networks which are targeted by administrator. This information gathered by client-end applications will be posted to the administrator system using web services architecture (client-server approach). As we are using web-services architecture for administrator application so a multiple platforms application is being developed for administrator to monitor the information of end-system. Using centralized server where the data is posted and the API for the server we are able to develop website and desktop application for administrator. The information of the end-systems will enable administrator to analyze the progress, state and security of the end-systems.

4.2.2 Structure and Relationships

Figure 4-1: Complete Architectural View



4.2.3 User Interface Issues

Figure 4-2: Use Case: Data Collector

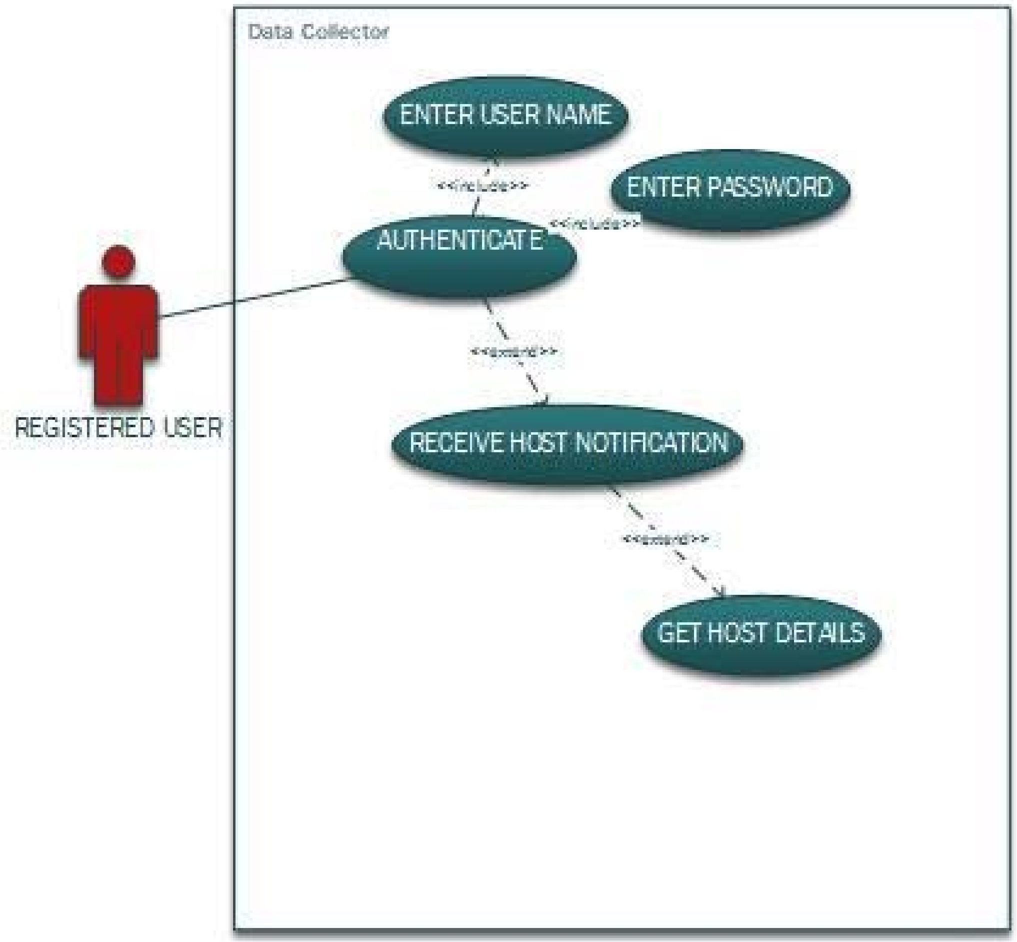


Figure 4-3: Use Case: Network Monitor

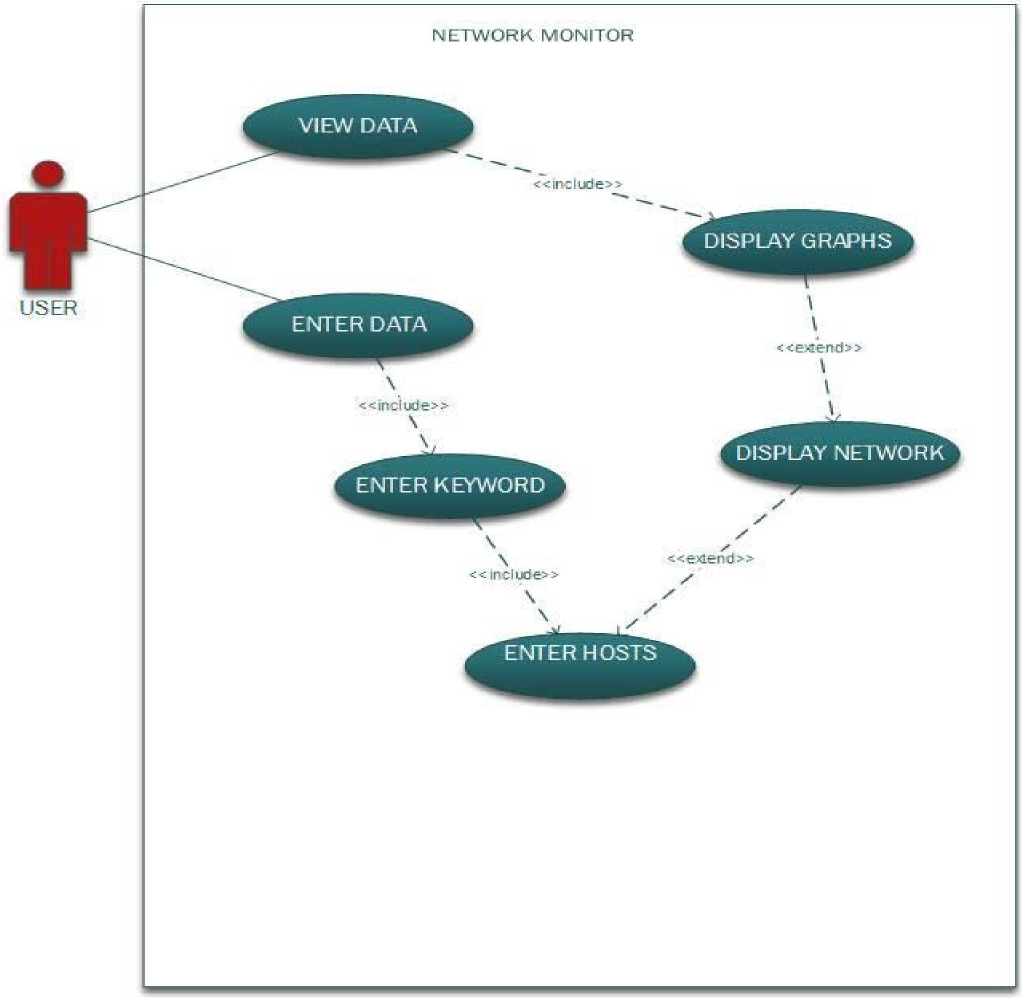


Figure 4-4: Activity Diagram: User Interface I

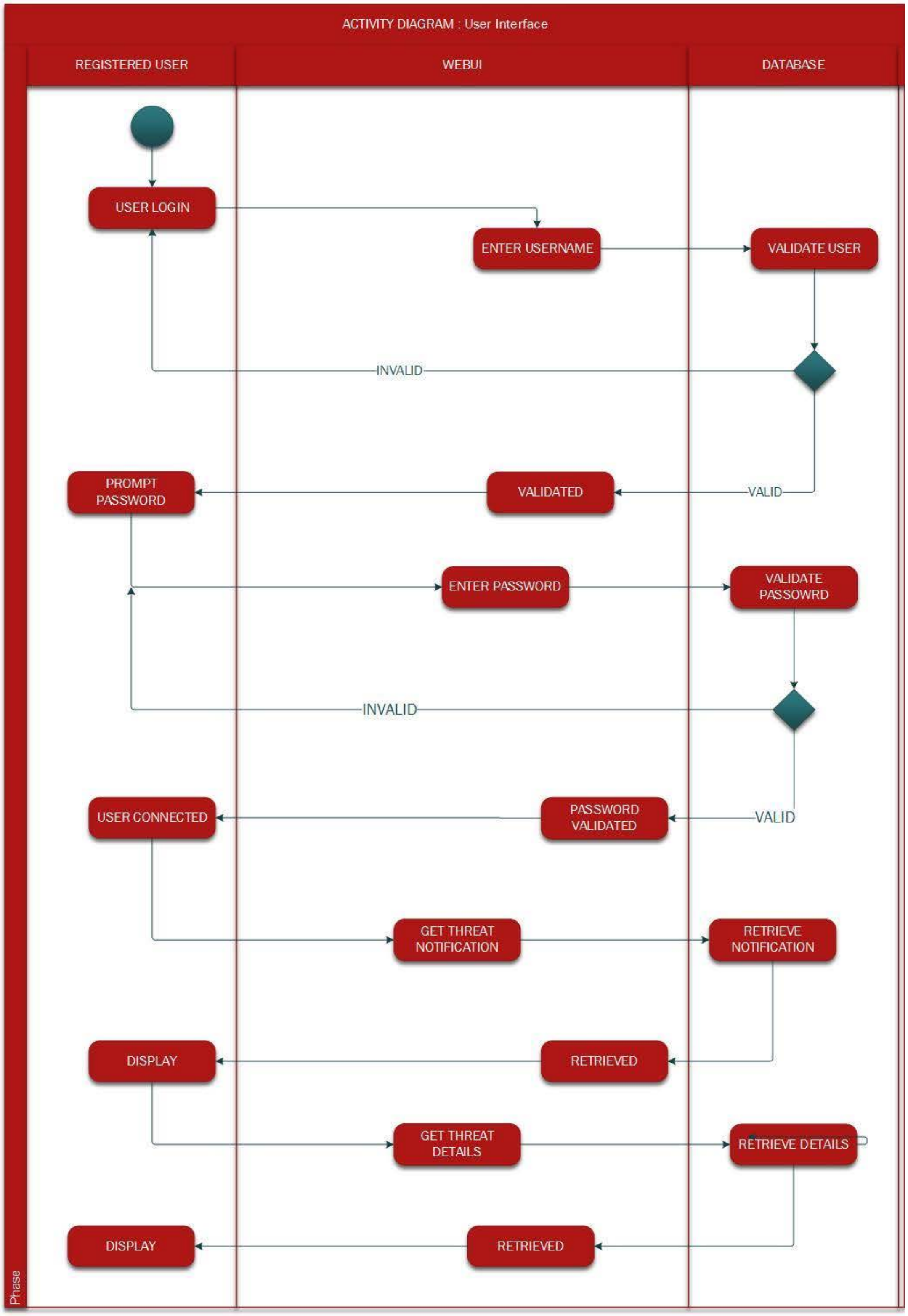


Figure 4-5: Activity Diagram: User Interface II

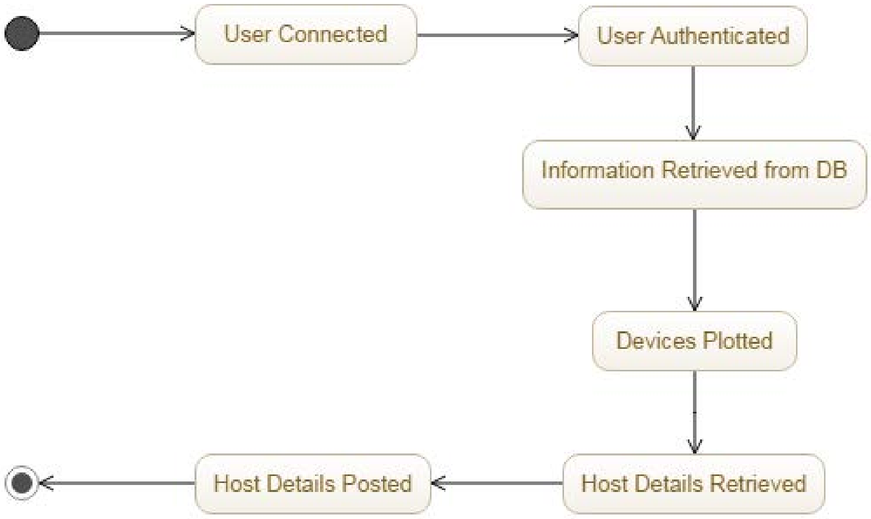


Figure 4-6: Activity Diagram: User Interface III

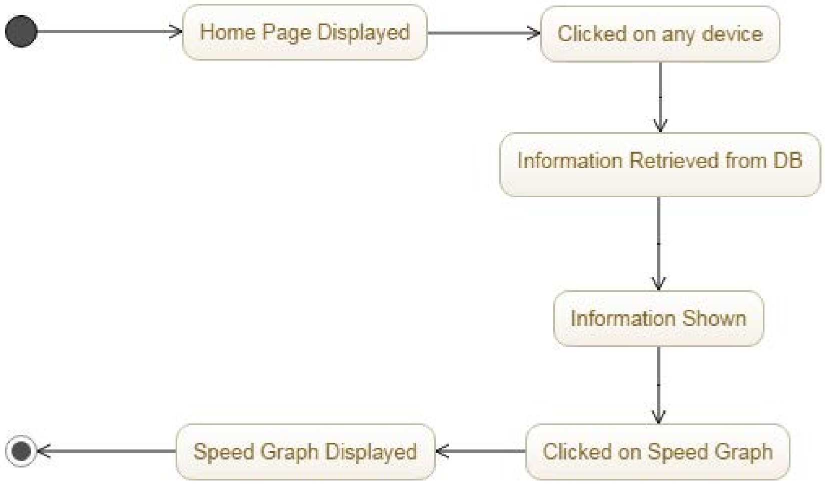


Figure 4-7: Sequence Diagram: User Interface

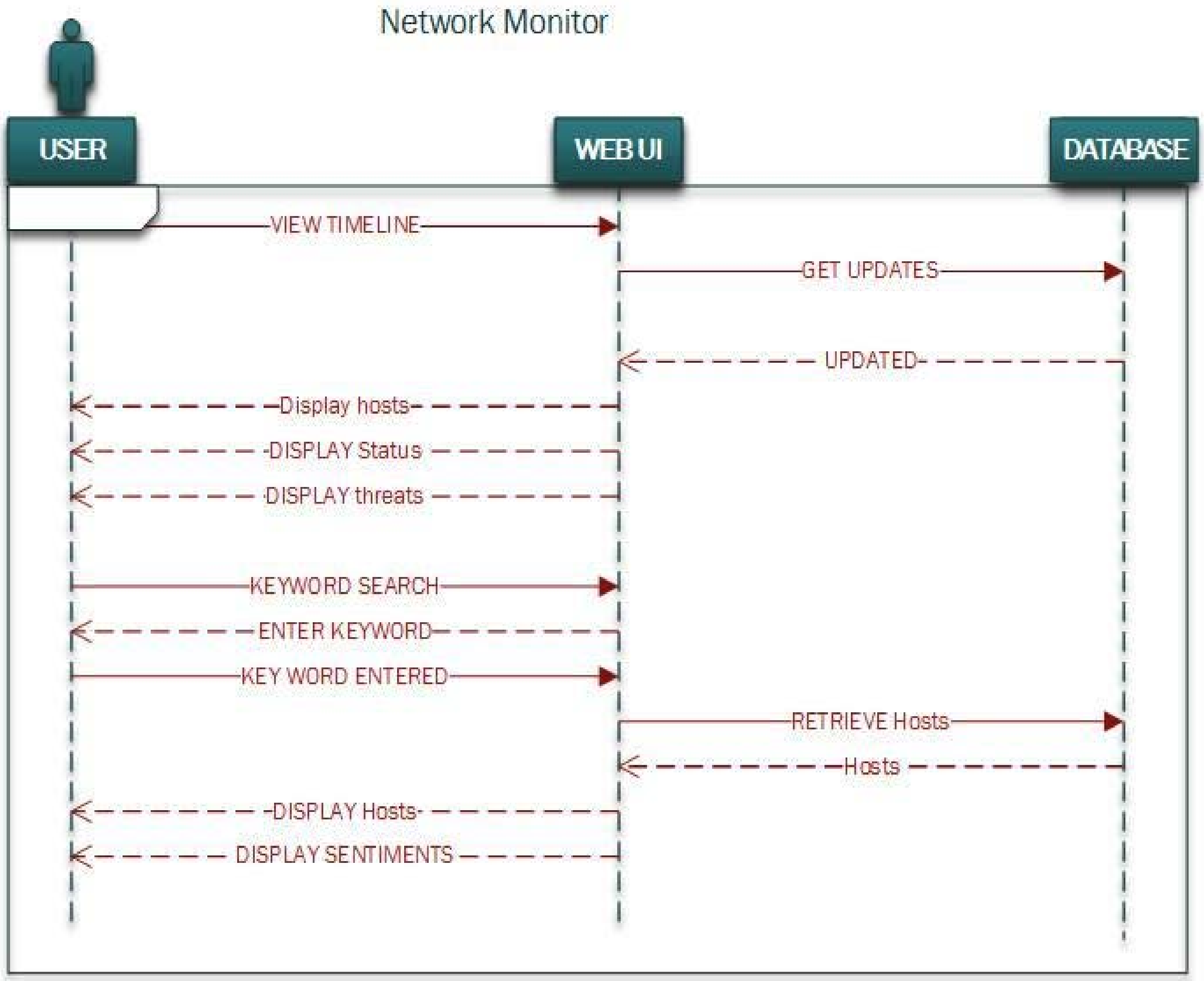
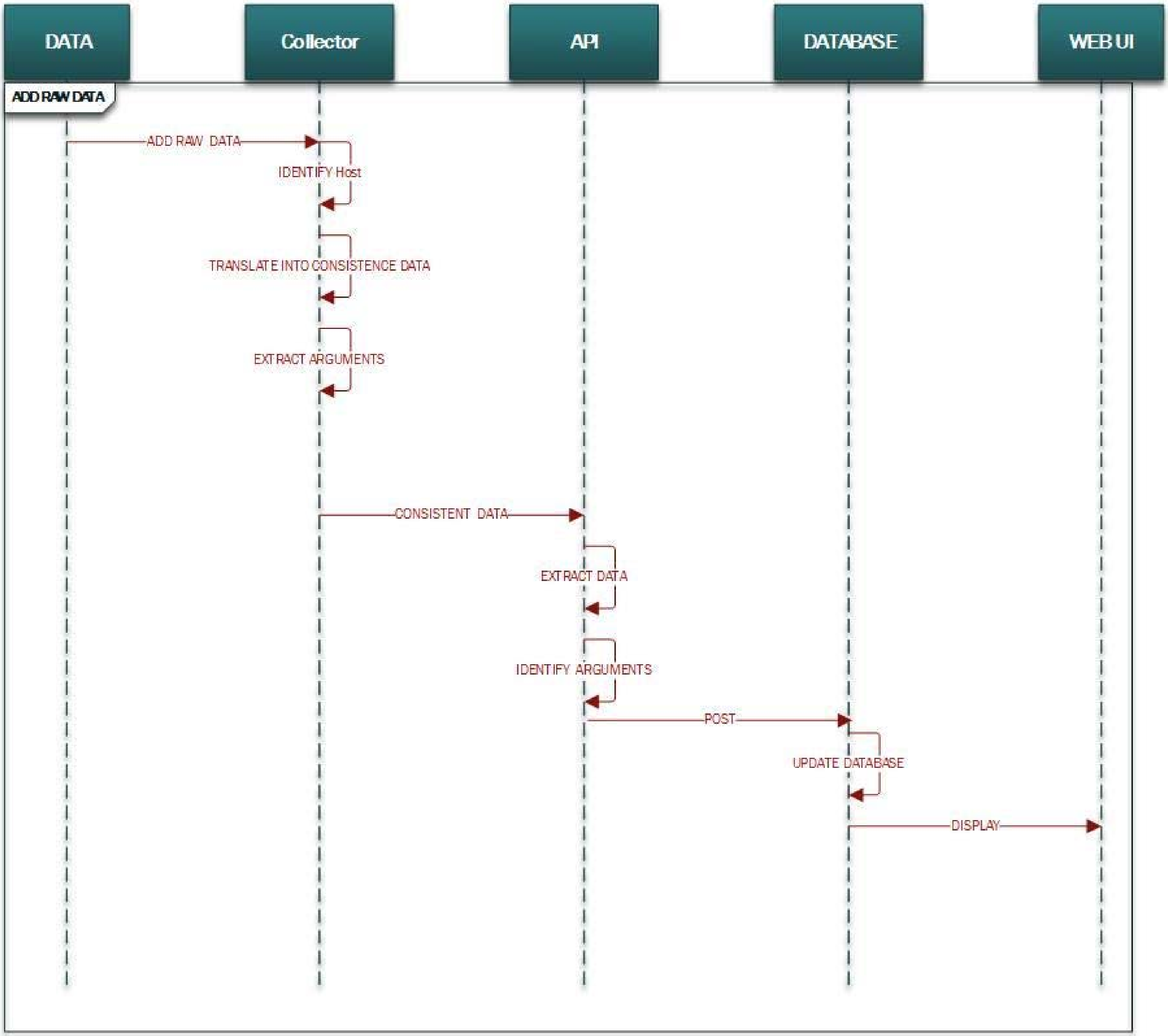


Figure 4-8: Sequence Diagram: NiS



4.3 Detailed Description of Components

4.3.1 Data Collector

Table 4-1: Description of components: Data Collector

Identification	*Name:* Data Collector *Location:* Application Logic layer of the system architecture
Type	Sub-component
Purpose	Following functional requirements are fulfilled by this sub-component: **REQ-1:** The system should scan the network after every time interval. **REQ-2:** The system should get basic computer info on network and identification of computers using SNMP. **REQ-3:** The system should set SNMP traps to get reports. **REQ-4:** The system should report the info the API to store into Database.
Function	The function of this sub-component is to do network discovery and scan the Network nodes via SNMP protocol. The scanned the then will be sent to API to store into the database.
Subordinates	It has two subordinates, *update data* and *retrieve data*. Update data performs functions to report to API satisfy these functional requirements: REQ-4. Whereas retrieve data will satisfy these requirements: REQ-1, REQ-2 and REQ-3.
Dependencies	This dependent is dependent on *no one.* But the Function API depends on the successful execution of this function.
Interfaces	Network information are fetched and sent to the API component.
Resources	Hardware: RAM and Processor of the system will be utilized and the Network. Software: C/C++ and SNMP libraries.
Processing	The component handles the network. It discovers the network and send out put to API component.
Data	SNMP Packet

4.3.2 Notification System

Table 4-2: Description of components: Notification System

Identification	*Name:* Processing Data *Location:* Application Logic layer of the system architecture
Type	Sub-Component
Purpose	Following functional requirements are fulfilled by this sub-component: **REQ-1:** The system should recognize if some new Device is connected in the network by processing the logs in from the API. **REQ-2:** The System should register the listeners which wants to get notified. **REQ-3:** The system should notify all the listeners
Function	The function of this sub-component is to register the listeners and get and process the data from the API and notify the changes to all of the listeners.
Subordinates	It has two subordinates, Listener Register and Notification System.
Dependencies	This subcomponent provides service to the component 'Application UI' i.e. these components depends on this component and the functionality of this component depends upon the API.
Interfaces	It provides external interface to Application GUI component.
Resources	Hardware: RAM and Processor of the system will be utilized. Software: C++ core libraries
Processing	Take input from other components (refer Fig 2.1.1) in the form of data and forward to some other components if useful changes are detected.
Data	SNMP Data

4.3.3 API

Table 4-3: Description of components: API

Identification	*Name:* API *Location:* Database layer of the system architecture
Type	Database component
Purpose	Following functional requirements are fulfilled by this sub-component: **REQ-1:** Application should be able to save data to database. **REQ-2:** It should return latest logs to the requesting components.
Function	This component handles database transactions i.e.; add, update, delete and select statements. It provides a bridge between the application and the database, i.e.; it can take input from the *Data Collector* component and update the database. Also, it can fetch data from database and send it to the *Notification Manager and User Interface* component.
Subordinates	It has 2 subordinates; *set* and *get*. *Set* does the modifying part of the database and it will fulfill REQ-1. *Get* does the fetching part from the database and fulfills REQ-2.
Dependencies	This component depends on the Data Collector. And the other components which depends on this component are Notification Manager and User Interface.
Interfaces	It uses SQLite and provide service to All other subcomponents to save and retrieve the data.
Resources	**Hardware:** RAM, Processor **Software:** SQLite, C/C++ core libraries.
Processing	Transaction is performed in order to retrieve data from the database. The component receives a query in form of an input SNMP packet from other components. The query is then executed and transaction is performed.
Data	SNMP Packet

4.3.4 User Interface

Table 4-4: Description of components: User Interface

Identification	*Name:* User Interface *Location:* Presentation layer of the system architecture
Type	GUI Component
Purpose	The user directly interacts with this component. He/she provides an input for the required action (through this component) and it displays its output respectively. This component fulfills following functional requirements related to user interaction in the application: **REQ-1:** Application should be able to show the connected Nodes on the network. **REQ-2:** The Application should be able to map the network. **REQ-3:** The application should be able to show info for each connected node on Selecting that node. **REQ-4:** The application should be able to get notifications from notification manager and map the changes to the screen.
Function	This component has two major functions; take input from the user and display all application screens. It takes input from API and displays the network topology on the screen. It then further takes the notifications from the notification manager and updates the GUI.
Subordinates	It has two subordinates, *one is for input and other is for output*. The input part fulfills the REQ-3 and Req-4 and the output part fulfills the REQ-1, REQ-2 and REQ-3 of the functional requirements.
Dependencies	This dependent is dependent on *notification manager and API. No other module depends on this one.*
Interfaces	Displaying the data.

Resources	Hardware: RAM and Processor of the system will be utilized and the Network. Software C/C++ and Open GL libraries.
Processing	The component draws the network on the screen then further handles the user inputs to shows network info.
Data	SNMP Packets

4.4 Reuse and Relationship to Other Products

The undertaken project is like a hybrid system which implements all the advantageous functionality based on any previous systems neither it is an extension of any other application at any level, but it can be evolved into a bigger and more complex system with more features and functionality. The practical usage of the system can be increased by monitoring and recording the types of packets being sent, bandwidth usage of each system, and raising an alarm if there is any abnormality in the above mentioned to provide a better and more accurate monitoring of the network.

4.5 Software Development Plan

Figure 4-9: Software Development Plan

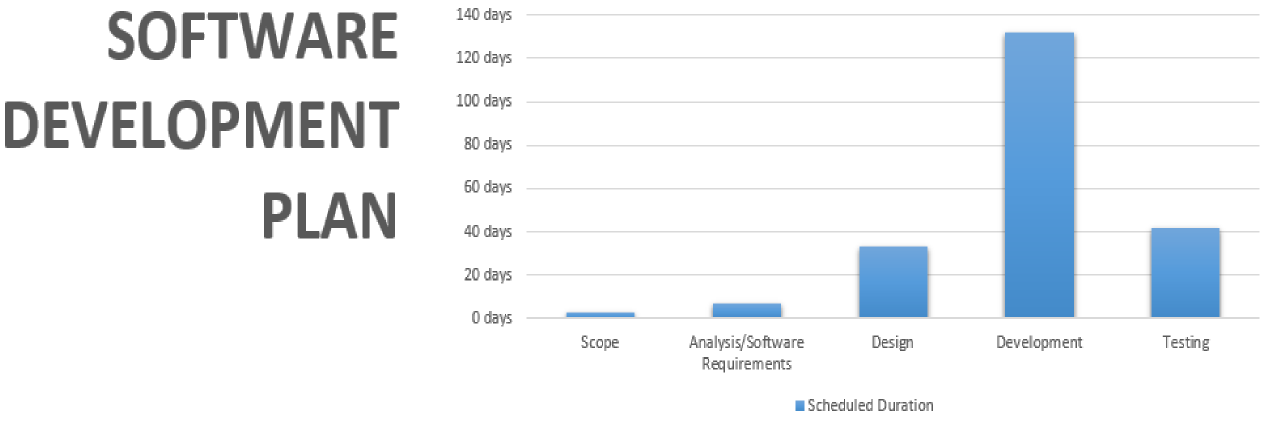
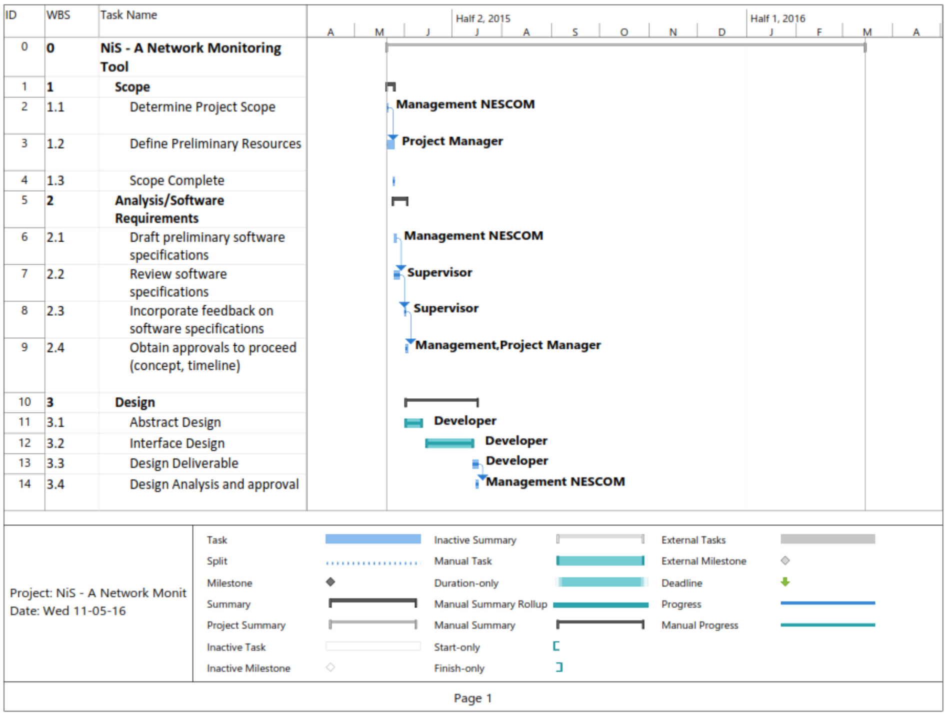


Figure 4-10: WBS I

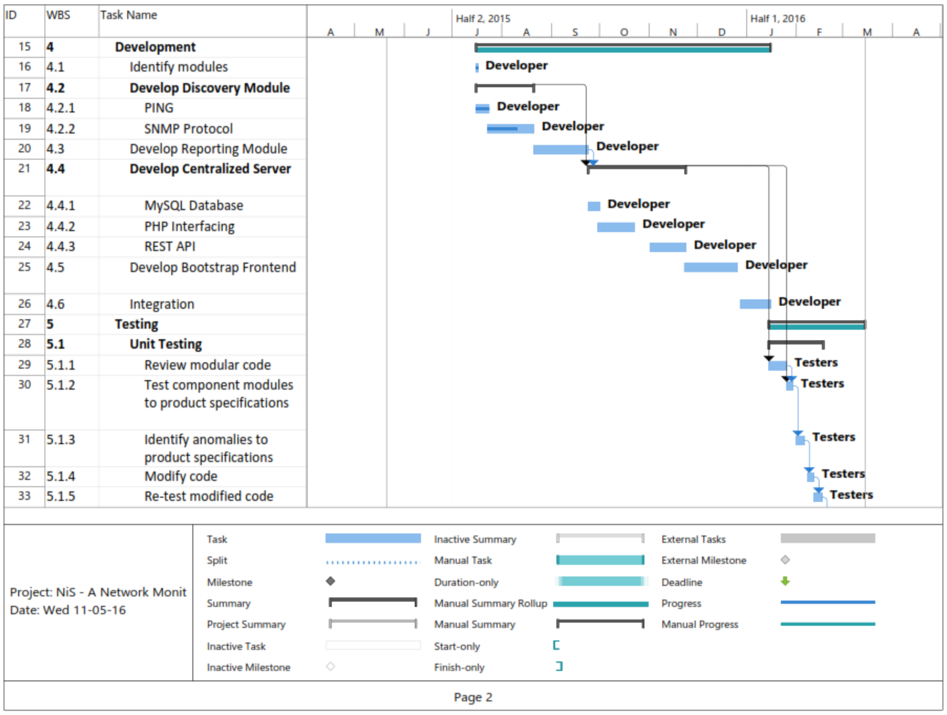


ID	WBS	Task Name
0	**0**	**NiS - A Network Monitoring Tool**
1	**1**	**Scope**
2	1.1	Determine Project Scope
3	1.2	Define Preliminary Resources
4	1.3	Scope Complete
5	**2**	**Analysis/Software Requirements**
6	2.1	Draft preliminary software specifications
7	2.2	Review software specifications
8	2.3	Incorporate feedback on software specifications
9	2.4	Obtain approvals to proceed (concept, timeline)
10	**3**	**Design**
11	3.1	Abstract Design
12	3.2	Interface Design
13	3.3	Design Deliverable
14	3.4	Design Analysis and approval

Project: NiS - A Network Monit
Date: Wed 11-05-16

Page 1

Figure 4-11: WBS II

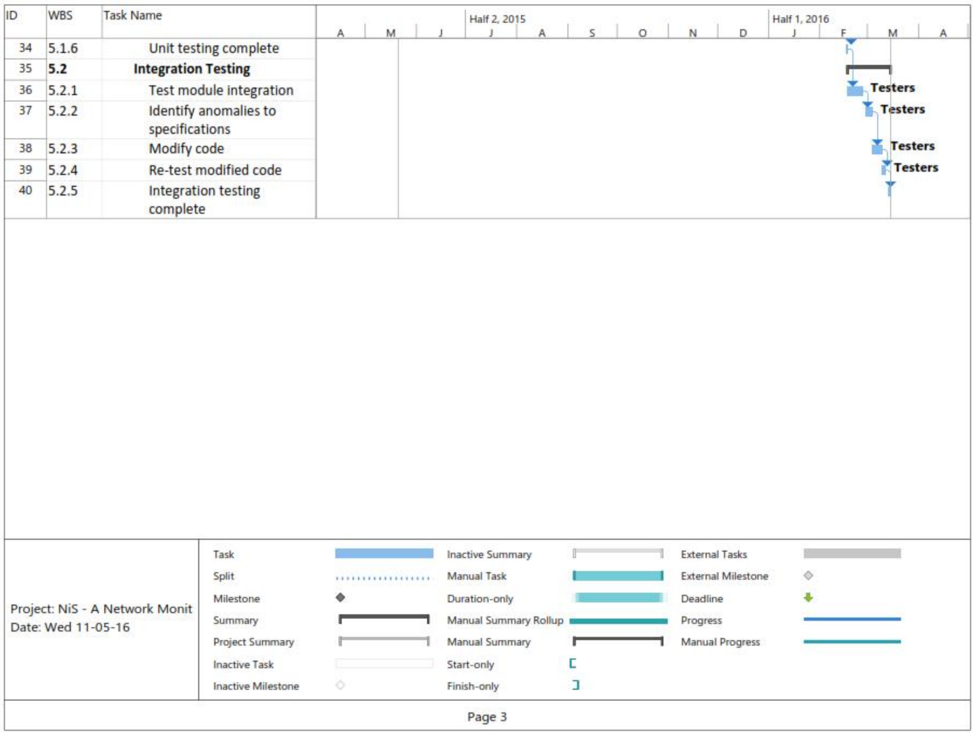


ID	WBS	Task Name
15	**4**	**Development**
16	4.1	Identify modules
17	**4.2**	**Develop Discovery Module**
18	4.2.1	PING
19	4.2.2	SNMP Protocol
20	4.3	Develop Reporting Module
21	**4.4**	**Develop Centralized Server**
22	4.4.1	MySQL Database
23	4.4.2	PHP Interfacing
24	4.4.3	REST API
25	4.5	Develop Bootstrap Frontend
26	4.6	Integration
27	**5**	**Testing**
28	**5.1**	**Unit Testing**
29	5.1.1	Review modular code
30	5.1.2	Test component modules to product specifications
31	5.1.3	Identify anomalies to product specifications
32	5.1.4	Modify code
33	5.1.5	Re-test modified code

Project: NiS - A Network Monit
Date: Wed 11-05-16

Page 2

Figure 4-12: WBS III



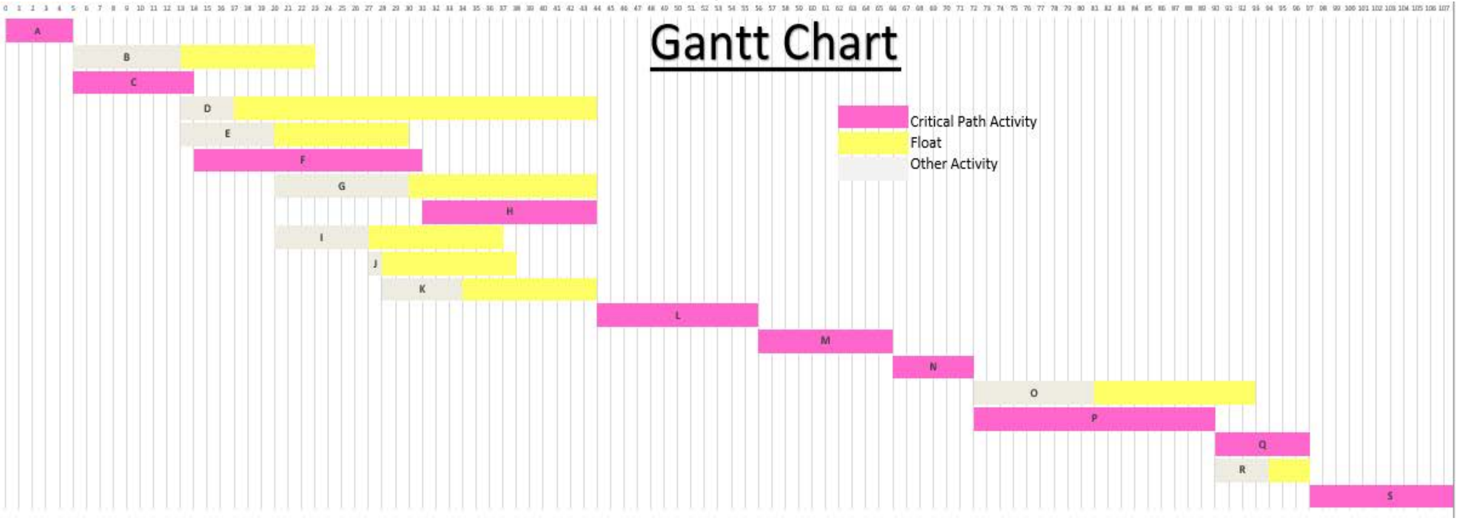
ID	WBS	Task Name
34	5.1.6	Unit testing complete
35	**5.2**	**Integration Testing**
36	5.2.1	Test module integration
37	5.2.2	Identify anomalies to specifications
38	5.2.3	Modify code
39	5.2.4	Re-test modified code
40	5.2.5	Integration testing complete

Project: NiS - A Network Monit
Date: Wed 11-05-16

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	◇
Milestone	◆	Duration-only		Deadline	⬇
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only	[		
Inactive Milestone	◇	Finish-only	]		

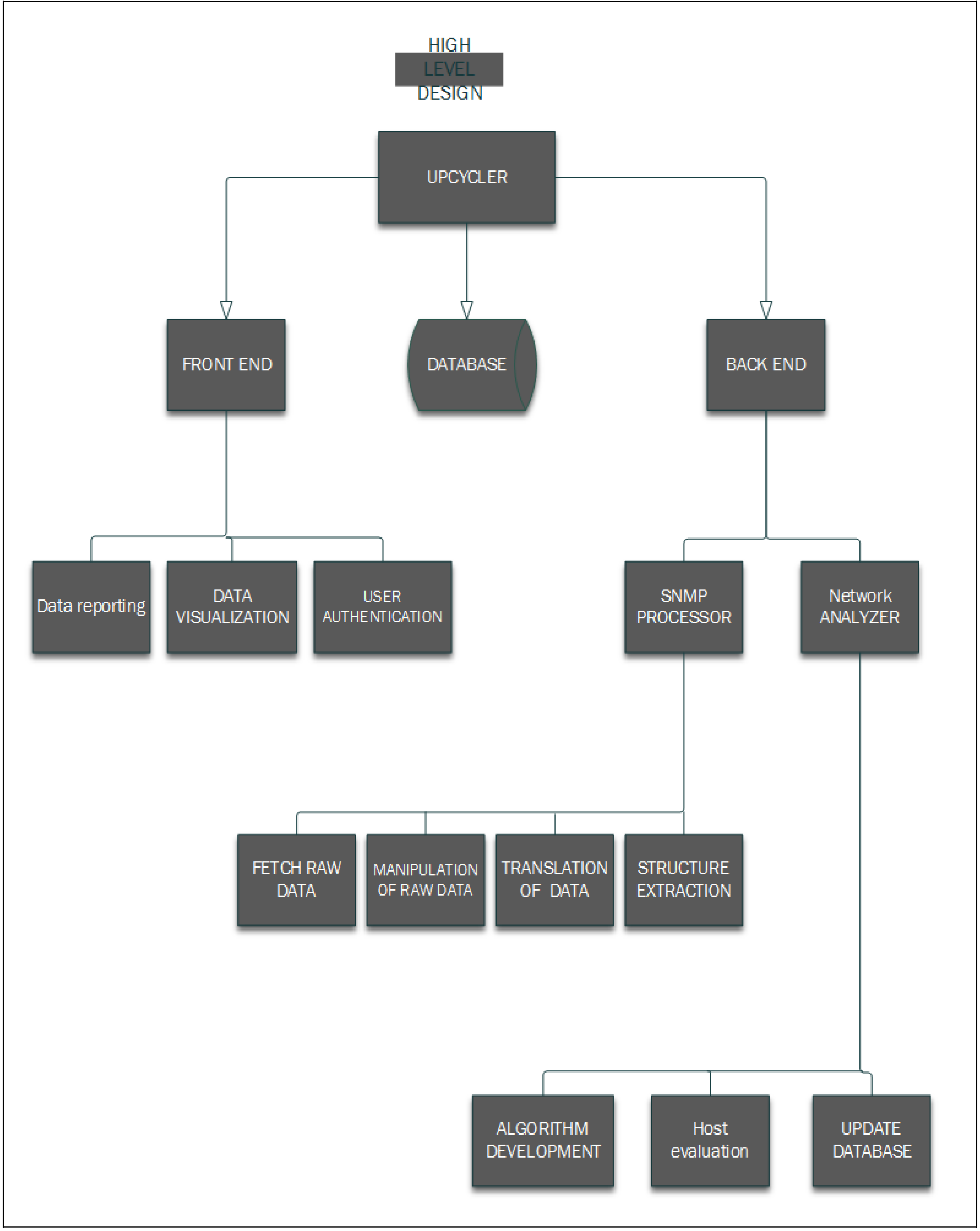
Page 3

Figure 4-13: Gantt Chart



4.6 High Level Design

Figure 4-14: High Level Design



4.7 Pseudo Code for Components

4.7.1 Data Collector

```
if some Device is Found
begin
        Show Device Found Message
        Get Data from device via SNMP
        Report data to API
end
else
    Move to next IP
```

4.7.2 Notification System

```
begin
    Get Data from API
    Check Data from Logs
    If Change in Data
    Begin
        Notify all Listeners
    end
end
```

4.7.3 API

```
if Request is Get Data
begin
    Get Logs from Data base
    Return the logs
end
else if Request is Set Data
begin
    Get Data from request
    Add Data to Database
End
```

4.7.4 User Interface

```
begin
    Log In screen
    if user authenticated
        then show live graph
        onClick
            Show related data
    if user does not exist
        then Error message print

end
```

Chapter 5: Project Analysis and Evaluation

5.1 Test Cases

The following table 5-1 below checks if the Login feature works accordingly and takes the user to the home screen as it was expected as an outcome of the login.

Table 5-1: Test Case: Testing Login Feature I

Test Case Number	1
Description	Log in
Preconditions	Login page must be open
Input	Correct Username and Password
Steps	Type the registered username and password
Expected output	User Authenticated
Results	Home Page is Displayed

The following table 5-2 below checks if the Login feature works accordingly and doesn’t give the user access in case of wrong credentials.

Table 5-2: Test Case: Testing Login Feature II

Test Case Number	2
Description	Log in
Preconditions	Login page must be open
Input	Incorrect Username and Password
Steps	Input wrong username and password
Expected output	User not authenticated
Results	Login Failed

The following table 5-3 below checks if the Parameters Panel works.

Table 5-3: Test Case: Testing Parameters Panel

Test Case Number	3
Description	Clicking on any device displayed on home screen
Preconditions	Home screen must be displaying devices
Input	Click on Device
Steps	Move the mouse and click on any device
Expected output	Device parameters are displayed
Results	Parameters are successfully displayed

The Table 5-4 and Table 5-5 below are similar tests applied to the application. They test the discovery module to check the time it takes and if at all it discovers devices not affected if they are SNMP enabled or not.

Table 5-4: Test Case: Testing the Discovery

Test Case Number	4
Description	Adding another device to network
Preconditions	Application must be running
Input	Connecting a new device
Steps	Connect a new device using Wi-Fi or wire
Expected output	New device gets displayed within 20 seconds
Results	New device gets correctly detected and displayed

Table 5-5: Testing the Discovery for Non-SNMP device

Test Case Number	5
Description	Adding a device on which SNMP is not enabled
Preconditions	Application must be running
Input	Connecting new device without SNMP
Steps	Connect a new device without SNMP
Expected output	New device gets displayed
Results	Device get successfully displayed

The following table 5-6 below checks if the Show Bandwidth function works.

Table 5-6: Testing the Bandwidth Feature

Test Case Number	6
Description	Testing functionality of Bandwidth button
Preconditions	Parameters Panel must be open and device should be SNMP enabled
Input	Opening Parameters Panel and clicking button
Steps	Open the Parameters Panel and click on the show bandwidth button
Expected output	Speed Graph
Results	Live Speed graph is shown

The following table 5-7 below checks if application gets device name.

Table 5-7: Testing Data Collector I

Test Case Number	7
Description	Testing functionality of Data Collector
Preconditions	Device must be connected
Input	Connecting the device and clicking on it on home page
Steps	Click the device shown on home page, it will open parameters panel which will contain device’s name
Expected output	Device Name
Results	Device Name is correctly displayed

The following table 5-8 below checks if application gets device type.

Table 5-8: Testing Data Collector II

Test Case Number	8
Description	Testing functionality of Data Collector
Preconditions	Device must be connected
Input	Connecting the device and clicking on it on home page
Steps	Click the device shown on home page, it will open parameters panel which will contain device’s type
Expected output	Device Type
Results	Device Type is correctly displayed

The following table 5-9 below checks if application gets device’s IP and MAC.

Table 5-9: Testing Data Collector III

Test Case Number	5
Description	Testing functionality of Data Collector
Preconditions	Device must be connected
Input	Connecting the device and clicking on it on home page
Steps	Click the device shown on home page, it will open parameters panel which will contain device’s IP and MAC
Expected output	IP Address and MAC Address
Results	IP Address and MAC Address is correctly displayed

The following table 5-10 below checks if the Settings Page works.

Table 5-10: Testing the Settings Page

Test Case Number	10
Description	Testing functionality of Settings Page
Preconditions	Web Page must be open
Input	Click on Settings button on left hand side
Steps	Click on the Settings button on left hand side
Expected output	Settings Page is opened
Results	Settings Page is opened

The following table 5-11 below checks if the Report Generation works.

Table 5-11: Testing Report Generation

Test Case Number	11
Description	Testing functionality of Report generation
Preconditions	Opening Reports generation page
Input	Clicking on reports button
Steps	Clicking the button to generate reports
Expected output	Report is generated
Results	Report is generated

The following table 5-12 below checks if the Logs works.

Table 5-12: Testing the Logs feature

Test Case Number	12
Description	Testing functionality of Logs
Preconditions	Application window must be open
Input	Clicking on Logs
Steps	Click on logs button
Expected output	Logs are shown
Results	Logs are shown

Chapter 6: Conclusion

6.1 Conclusion

‘NiS – A Network Monitoring Tool’ is a complete package comprising of multiple network monitoring modules that are beneficial for normal people as well as administrators. Not only the people’s time and cost would be saved but with the introduction of this application administrators will have less work load as well effective network monitoring.

6.2 Future Work

The product developed can be re used in the future to proceed with evolution. The number of parameters that are extracted from devices can even be extended in the next versions of this application. A new section in this project shall be the addition of the capability of showing not just amount of network traffic per device but also the traffic type. This will further help identify any potential threats in the network, generating unusual type of traffic. Another dimension can be setting up policies and getting notified on cell in case any policy’s condition is met. Furthermore, Smartphone apps can be integrated with the system to do on-the-go monitoring as well.

Appendix A: Bibliography

[1] https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol

[2] https://www.paessler.com/network_monitoring_tool

[3] http://www.cacti.net/

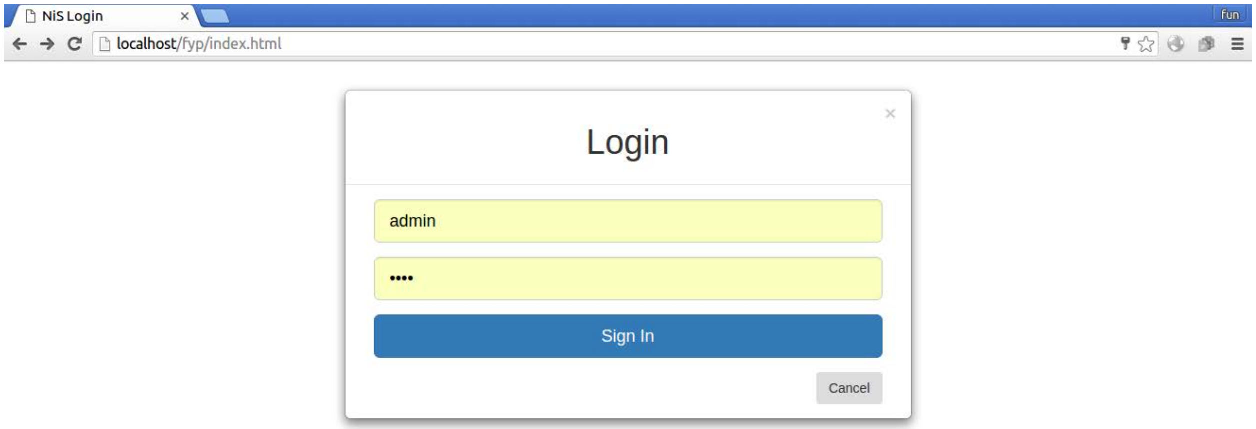
[4] http://www.gfi.com/blog/the-top-20-free-network-monitoring-and-analysis-tools-for-sys-admins/

Appendix B: User Manual

1.1 Login Page

Upon typing the URL of the Web Server, a login page appears, enter username and password to get access to the monitoring system

Figure 7-1: Login Page



localhost/fyp/index.html

Login

admin

••••

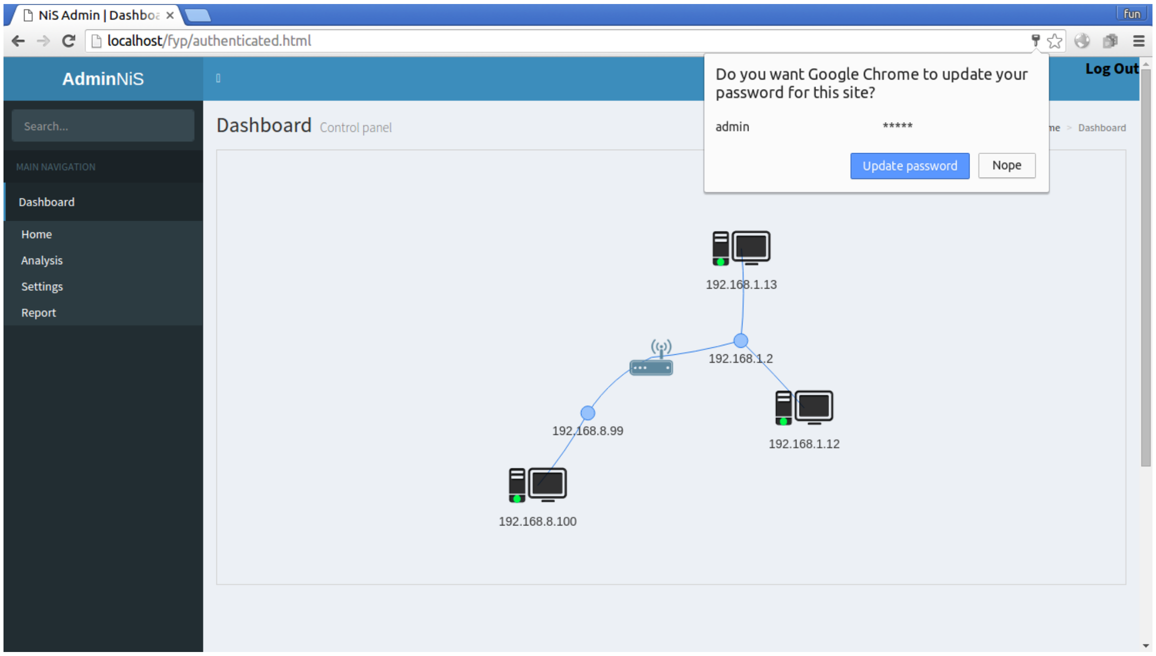
Default Username: admin

Default Password: admin

1.2 Home Page

After successful entering his/her credentials, user is greeted by the home page. It displays the current network topology automatically and user does not have to give any kind of input

Figure 7-2: Home Page



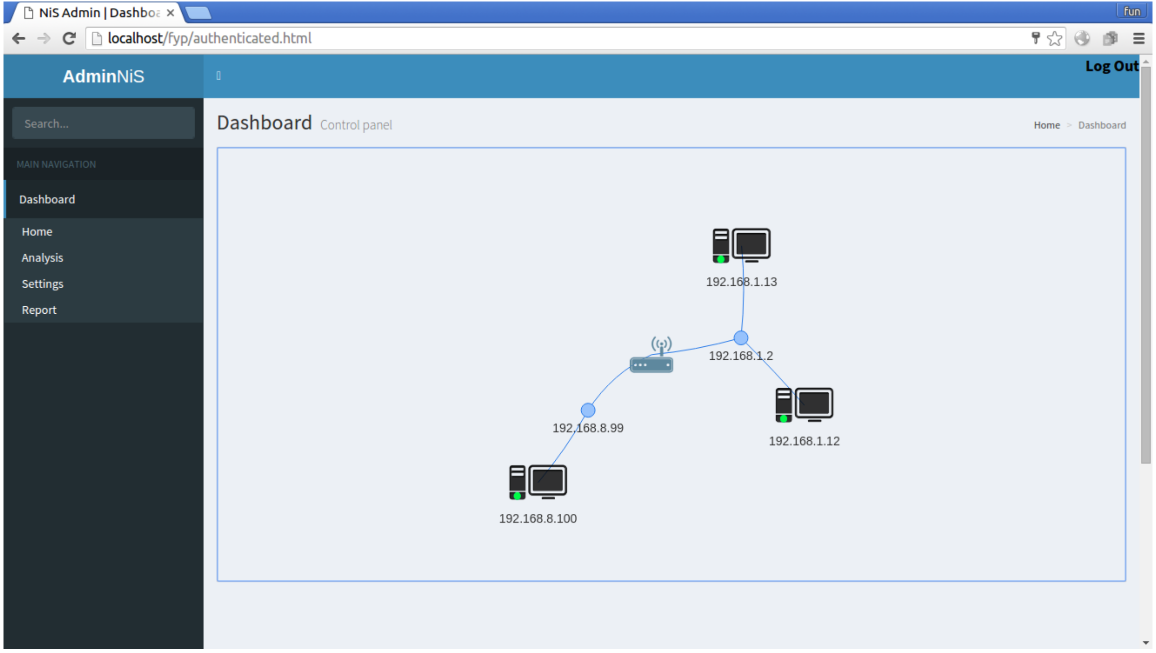
localhost/fyp/authenticated.html

Do you want Google Chrome to update your password for this site?

admin *****

- Green light indicates that device is online

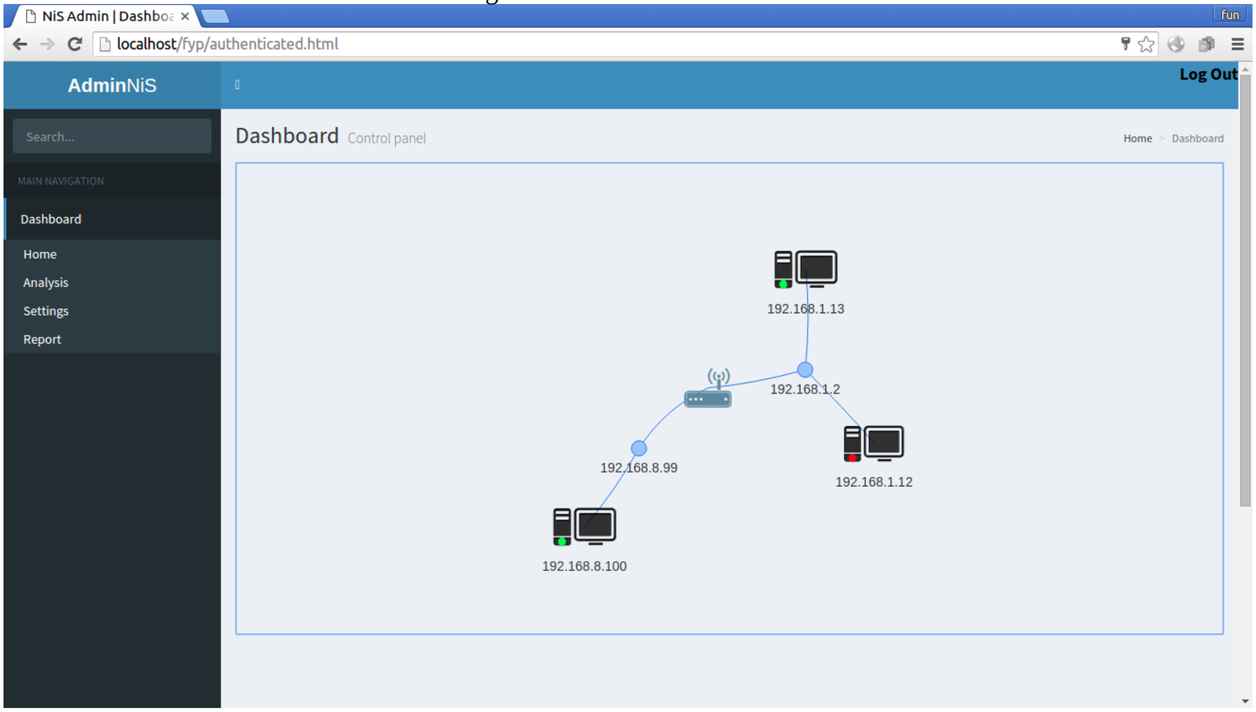
Figure 7-3: Online Devices



localhost/fyp/authenticated.html

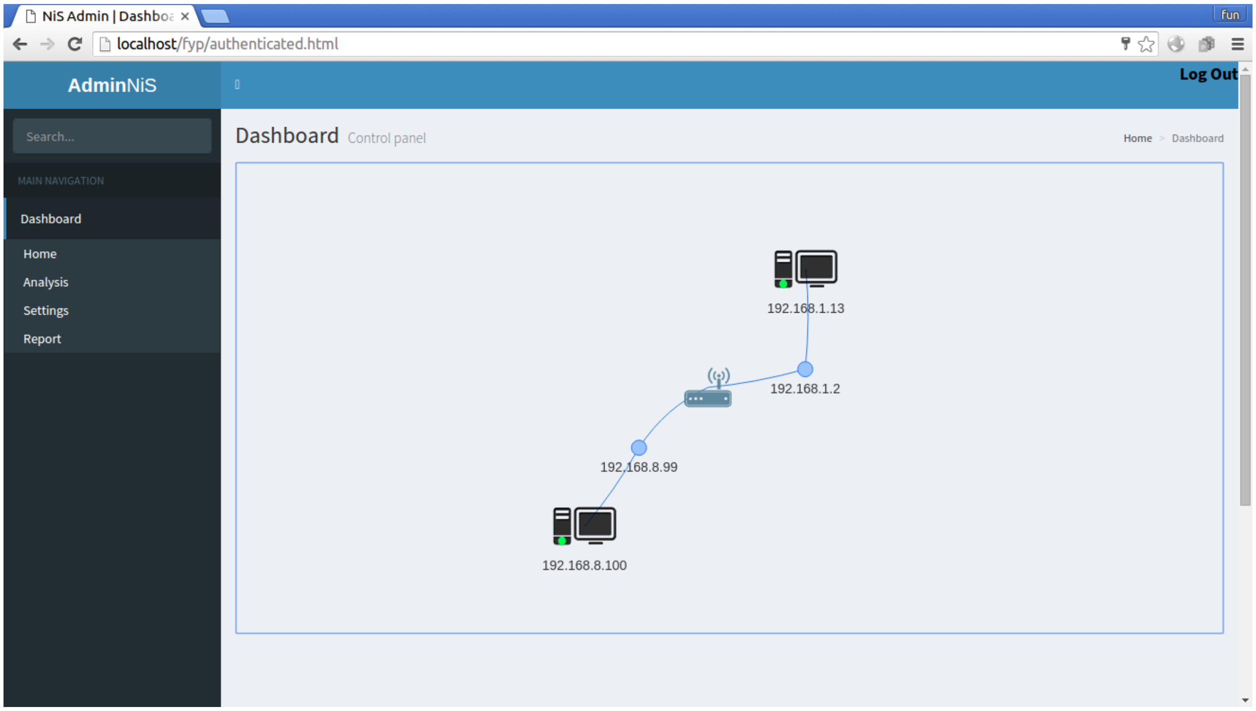
- Red light indicates that system is offline

Figure 7-4: Offline Device



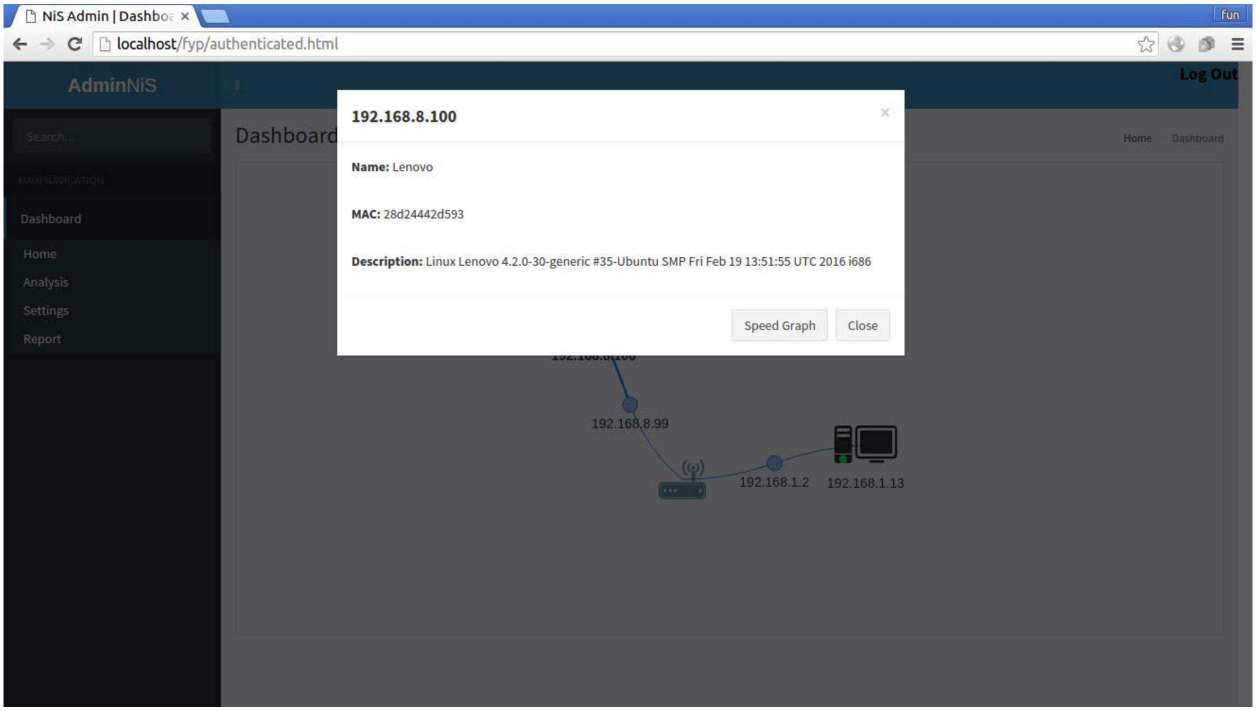
- Offline systems are removed from network after a time period of 20 seconds

Figure 7-5: Device removed from Network



- You can click on any device to get its Name, MAC, IP etc.

Figure 7-6: Device Parameters



192.168.8.100

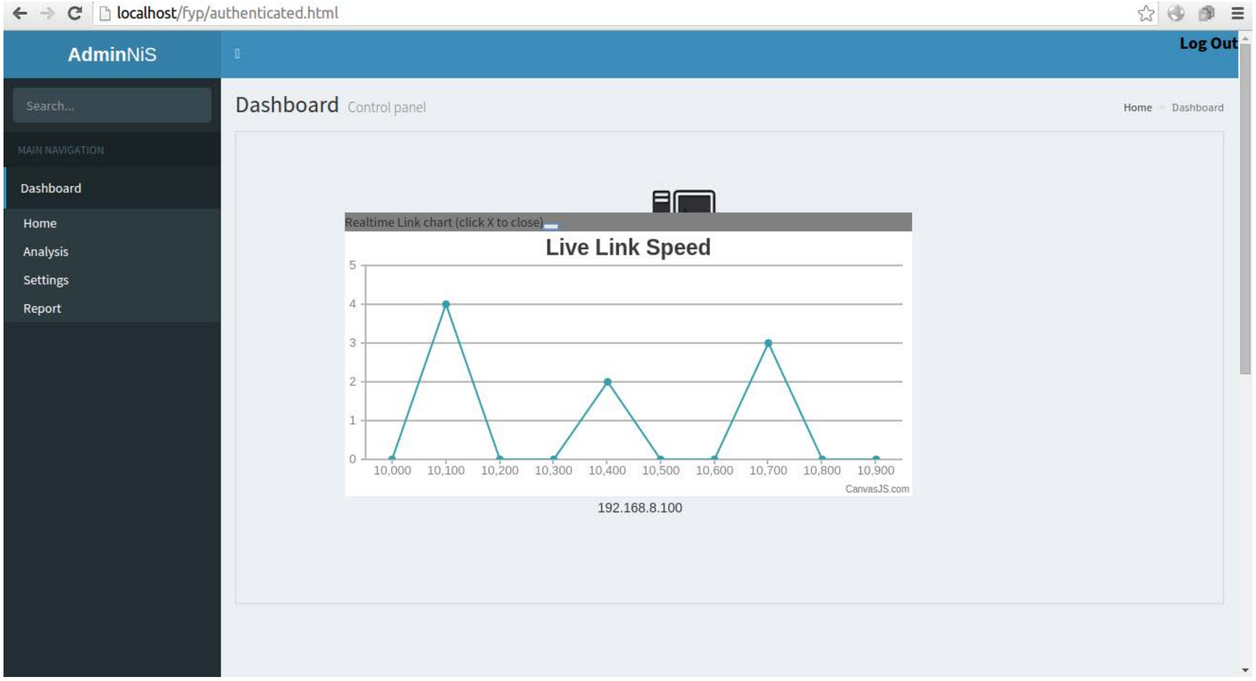
Name: Lenovo

MAC: 28d24442d593

Description: Linux Lenovo 4.2.0-30-generic #35-Ubuntu SMP Fri Feb 19 13:51:55 UTC 2016 i686

- You can also click the Speed Graph button in that dialog to open up the Speed Graph

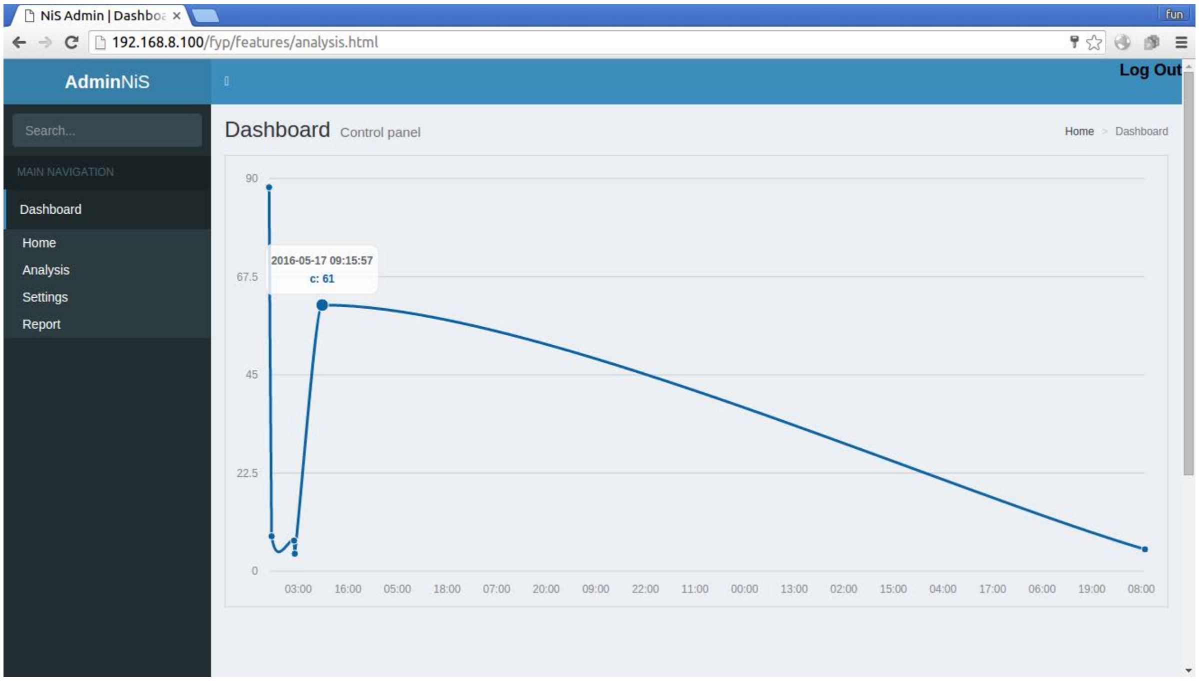
Figure 7-7: Speed Graph



1.3 Analysis Page

This page shows a graphical representation of the network activity history

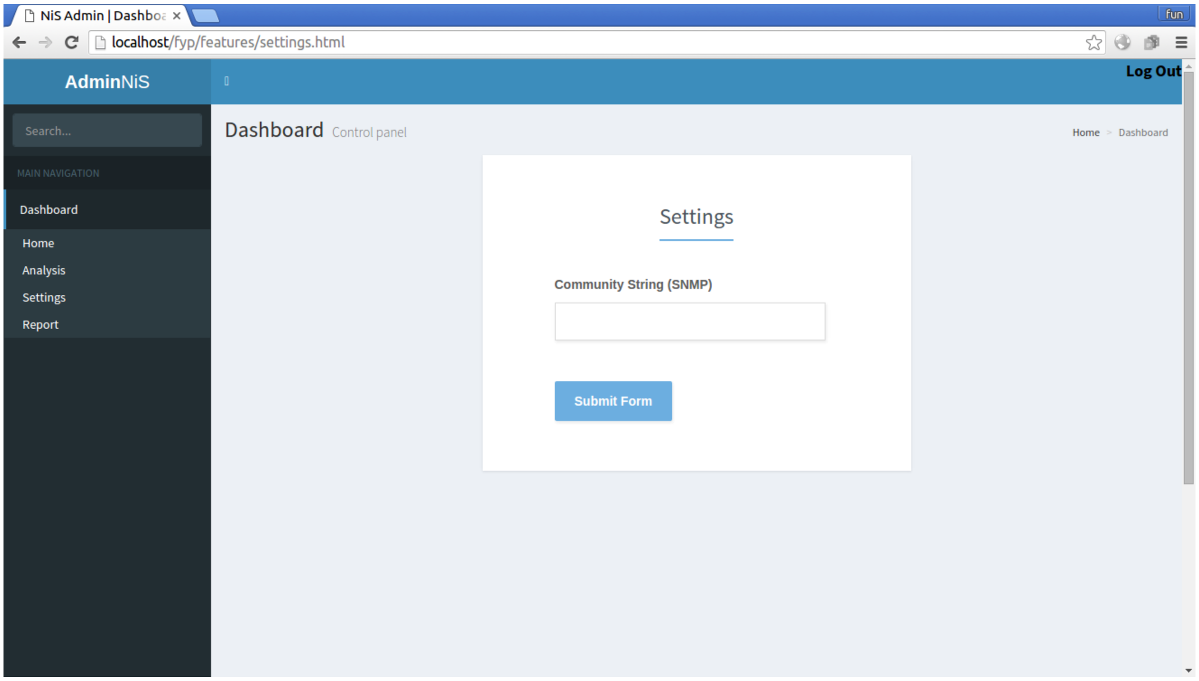
Figure 7-8: Analysis Page



1.4 Settings Page

You can set the community string using Settings page, default community string is ‘public’

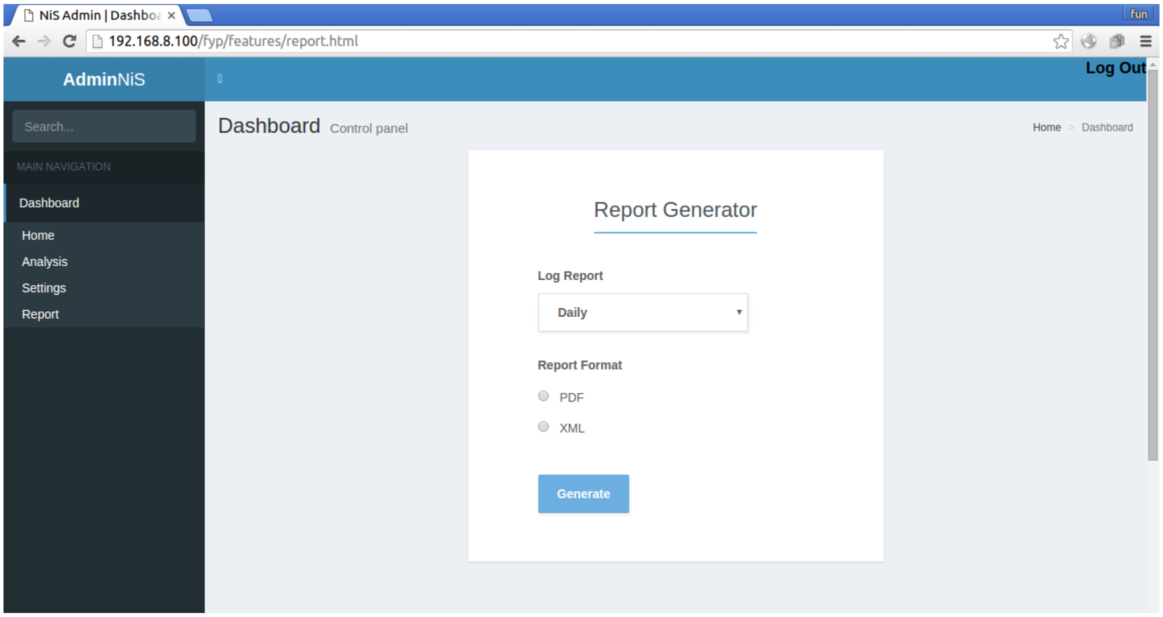
Figure 7-9: Settings Page



1.5 Report Page

You can generate reports of the network activity in XML or PDF format using the report page

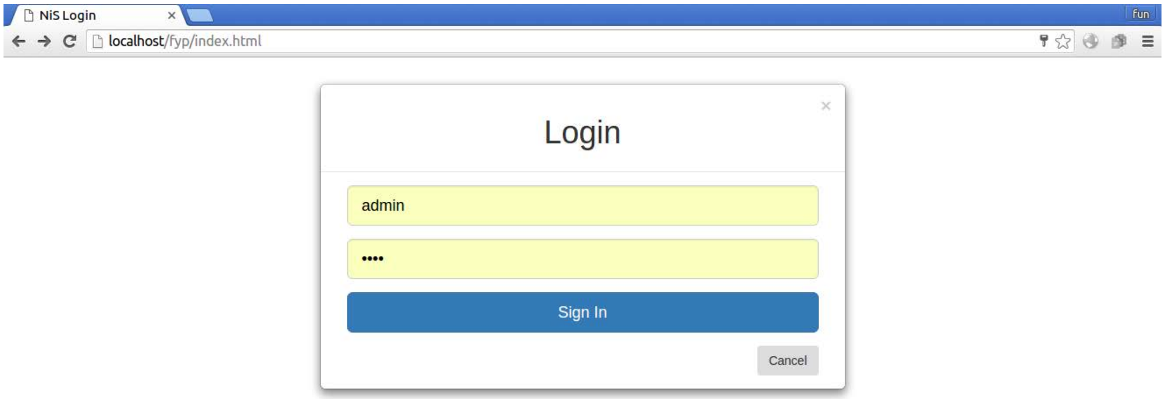
Figure 7-10: Reports Page



1.6 Log Out

User can log out from the Admin Panel by clicking the Log Out button at the upper right corner of any page

Figure 7-11: Log Out


NiS Login
fun
localhost/fyp/index.html
Login
admin
••••
Sign In
Cancel