VelocityNetHub: An SNMP based Open-Source Network Management Tool Using for SMEs

Waruingi Lauryn Wanjiku

150150

CNS

Supervisor Name

Dr. Vitalis Ozianyi

Submitted in Partial Fulfillment of the Requirements of the Bachelor of Science in Computer Networks and Cybersecurity at the Strathmore University

School of Computing and Engineering Science

Strathmore University

Nairobi, Kenya

May 2024

Declaration and Approval

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the research proposal contains no material previously published or written by another person except where due reference is made in the research proposal itself.

Student Name: Waruingi Lauryn Wanjiku

Admission Number: 150150

Student Signature: __________________________ Date: _________________________

The Proposal of **Waruingi Lauryn Wanjiku** has been reviewed and approved by **Dr. Vitalis Ozianyi.**

Supervisor Signature: __________________________ Date: _________________________

Acknowledgement

I would like to express my deepest gratitude to all those who have contributed to the completion of this project proposal. First and foremost, I extend my sincerest appreciation to my supervisor, Dr. Vitalis Ozianyi, for his invaluable guidance, support, and encouragement throughout the entire process. I am also immensely thankful to my colleagues, June Ngoitsi and Arnold Ochieng', for their collaboration and helpful discussions, which greatly enriched the project. I am grateful to Strathmore University for providing the necessary resources and facilities to carry out this research. This work would not have been possible without the support and contributions of each of these individuals, and for that, I am truly grateful.

Abstract

VelocityNetHub is a network management tool using Simple Network Management Protocol (SNMP) designed specifically for small and medium-sized enterprises (SMEs). The project uses design thinking that emphasizes user-centered design to overcome the challenges of effectively managing SME networks. The project defines the most important parameters and objectives, considering the unique needs of SMEs, and ensures that the development process meets the needs of users. VelocityNetHub uses creative solutions that focus on user experience and usability through ideation and prototyping. During user evaluation sessions, feedback is collected, and the prototype is improved, resulting in a final product that meets the SME's expectations. The deliverables include a model and user interface for the network management tool, as well as a detailed proposal for the objectives, scope and expected results of the project. The importance of this project is that it can significantly increase the productivity of SMEs by providing them with customized network management solutions. The aim of the project, which meets the special needs of SMEs, is to give companies the opportunity to optimize their networking activities, which ultimately leads to better performance and competitiveness in the market.

Table of Contents

List of Figures

List of Abbreviations

CNAM: Computer and Network Asset Management

CPU: Central Processing Unit

DoD: Department of Defense

ICMP: Internet Control Message Protocol

IT: Information Technology

MIB: Management Information Base

OID: Object Identifier

PRTG Network Monitor: Paessler Router Traffic Grapher Network Monitor

SME: Small and Medium Enterprises

SNMP: Simple Network Management Protocol

TCP/IP: Transmission Control Protocol/ Internet Protocol

UDP: User Datagram Protocol

VNH: VelocityNetHub (the proposed solution)

Chapter 1: Introduction

1.1 Background Information

Network management is the foundation for the efficiency, safe use, and reliability of computer networks. It is a set of activities that includes not just the monitoring of network devices but also performance metrics' analysis, device setup, and diagnosing and resolving issues (Frontier Business Products, 2022). Introduced in 1988 (Datadog, 2024), Simple Network Management Protocol (SNMP) is one of the most important tools in network management because of its simplicity and effectiveness. SNMP enables the retrieval of data from network devices and is used to manage them from remote locations. As it may gather data on the performance of devices in real-time, such as CPU load, memory utilization, and network use, it has become a vital tool for many IT professionals. (Splunk, 2023).

Many essential gaps remain despite the progress achieved in the field of network management tools. First, the available tools are expensive, most notably the proprietary tools, which makes it impossible for small and medium-sized enterprises (SMEs) to implement efficient network management practices (Naoyuki & Taghizadeh-Hesary, 2016). Secondly, there are open-source solutions, but they do not offer intricate features required for successful network management such as intelligent alerting, real-time monitoring, and historical data maintenance models. Moreover, the lack of flexibility in the tools makes them limited in terms of supervision and adaptation.

Bridging these gaps is important for several reasons. First, the development of affordable and feature-rich network management solutions will eliminate barriers to essential management components for organizations of all sizes, democratizing network management and allowing small and medium enterprises to effectively compete on the market. In addition, enhancing the abilities of open-source resources guarantees that businesses can utilize adaptable and custom-made tools that can be adjusted to their distinctive network settings. This offers a foundation for innovation and allows organizations to tailor their network administration frameworks in line with their specific requirements (Zhu, 2021).

1.2 Problem Statement

The current available network management tools pose several challenges for organizations in different industries. Firstly, the high cost associated with many proprietary tools is a roadblock specifically for small to medium-sized enterprises (SMEs) with limited budgets. These organizations are often unable to afford the steep licensing fees required to access essential network management functionalities (Naoyuki & Taghizadeh-Hesary, 2016). As a result, they are forced to seek alternative, often less effective, solutions or give up on critical management capabilities altogether. Secondly, existing SNMP-based tools, while functional, often do not offer flexibility and customization options necessary to adapt to the different and evolving needs of modern network environments. This lack of flexibility hinders organizations from tailoring their network management systems to suit specific needs (Tarutė & Gatautis, 2013).

Unfortunately, many SNMP-based network management tools in the market today are deficient in many features required for comprehensive network monitoring and management. While they may offer basic functions like data compilation and device setup, they frequently fall short on delivering advanced features such as real-time surveillance, smart notifications, and analysis of past data (Datadog, 2024). This limitation obstructs organizations from actively identifying and addressing network problems, thus increasing downtime and security vulnerabilities. Furthermore, the issue of lack of clarity and control is prevalent in proprietary solutions, which hinders organizations from fully utilizing their network management tools (Naoyuki & Taghizadeh-Hesary, 2016). They often find themselves locked into vendor-specific ecosystems, making it difficult to integrate with other systems or customize functionalities to meet specific requirements.

Given the challenges at hand, there is an urgent requirement for a network management tool that operates on open-source principles. This tool should utilize the capabilities of SNMP while addressing the limitations of current solutions. Its primary aim would be to offer organizations a cost-effective, flexible, and feature-rich alternative to proprietary options. Furthermore, an open-source approach promotes transparency, allowing organizations to modify the underlying code as necessary. This not only enhances security but also provides greater control over their network infrastructure. Consequently, developing an open-source network management tool that utilizes

SNMP presents an opportunity to empower organizations with the necessary resources to effectively manage and optimize their networks.

1.3 Objectives

1.3.1 General Objective

The general objective of this project is to develop VelocityNetHub: An open-source network management tool that utilizes SNMP for monitoring and managing network devices.

1.3.2 Specific Objectives

The specific objectives of the project are:

i. To investigate the operation of SNMP tools.
ii. To design the VelocityNetHub Network Management tool using SNMP.
iii. To develop VelocityNetHub.
iv. To test and validate VelocityNetHub.

1.4 Research Questions

i. What are the principles used by SNMP-based tools?
ii. How can an SNMP-based network management tool be designed?
iii. How can an SNMP-based network management tool be developed?
iv. How can an SNMP-based network management tool be tested and validated?

1.5 Justification

Several crucial reasons validate the creation of an open-source network management application utilizing SNMP. Firstly, the prohibitive cost tied to proprietary network management strategies poses a considerable hurdle for numerous entities, especially small to medium-sized enterprises (SMEs). By offering an economical substitute, an open-source application guarantees that organizations of all sizes can obtain crucial network management features without suffering financially. Secondly, the adaptability and personalization capabilities provided by open-source strategies allow entities to modify their network management systems to match their distinct needs.

This ensures that entities can establish strategies that coincide with their unique network surroundings and business goals.

Moreover, adopting an open-source methodology promotes clarity and teamwork, enabling corporations to tap into the collective wisdom of the open-source community. Additionally, by employing Simple Network Management Protocol (SNMP), a universally recognized protocol, the suggested tool guarantees compatibility with a diverse array of network devices and vendors. This effectively minimizes vendor dependency and issues related to interoperability, providing corporations with increased adaptability and governance over their network infrastructure. Consequently, the creation of an open-source network management instrument utilizing SNMP is warranted due to its potential to offer cost-efficient, adjustable, and feature-packed solutions for corporations aiming to streamline their network management procedures.

1.6 Scope, Limitations and Delimitations

1.6.1 Scope

The scope of this project is to develop a basic version of a network management tool equipped with core capabilities. These functionalities encompass gathering and storing data, analysis, and providing an interactive user interface for system engagement. The tool will facilitate communication with network devices by supporting SNMPv2 and SNMPv3 protocols, thereby ensuring its compatibility with diverse devices and manufacturers. In addition to this, the tool will be capable of managing SNMP traps, enabling it to accept and process unsolicited alerts from network devices. Nonetheless, the incorporation of advanced features like the support for alternate network protocols (for instance, ICMP or NetFlow) and the introduction of more intricate data scrutiny capabilities are aspects that will be contemplated for subsequent versions of the tool.

1.6.2 Limitations

Although VelocityNetHub holds promising advantages, there are also a few inherent limitations to consider. Firstly, there could be instances where the tool might not be compatible with certain proprietary extensions of SNMP or specific vendor features, thereby restricting its use in some network settings. In addition, the tool's scalability might not be sufficient for extensive networks comprising thousands of devices since it is primarily tailored for small to medium-sized networks. Moreover, the tool's objective is to cater to a wide array of use cases and network setups, but due to its initial stage of development, it might not be equipped to handle every conceivable situation. Lastly, the constraints related to project timelines and resources may put restrictions on the depth and variety of features that can be incorporated in the tool's initial release. Nonetheless, the future versions of this tool would aspire to overcome these challenges and enhance its capabilities to keep up with the ever-changing needs of network management.

1.6.3 Delimitations

This project has certain delimitations that define the boundaries of its scope and execution. To begin with, it operates under the presumption that standard hardware setups are available for the deployment of the network management instrument. Even though key hardware requirements like processing capacity and memory are important for peak performance, they are not catered to in this edition. Secondly, the tool's creation has considered its compatibility with widely used operating systems, for instance, Linux and Windows. Nonetheless, compatibility with less used or outdated operating systems is not assured and might necessitate further adjustments. Furthermore, VelocityNetHub caters to relatively simple network environments typically seen in small to medium scale businesses. It does tackle common circumstances, but uniquely complex network arrangements with custom configurations could call for added customization, and these are not fully catered to in this edition. These parameters help to define the author’s focus and shed light on its limitations. They also serve to manage expectations concerning what the project can deliver and what falls outside its scope. Therefore, potential users should assess their needs against these parameters to ensure the tool aligns with their specific requirements.

Chapter 2: Literature Review

2.1 Simple Network Management Protocol

2.1.1 SNMP Structure Model

SNMP consists of four components: SNMP Manager, Managed devices, SNMP agent and Management Information Database (MIB). The SNMP manager is an entity responsible for communicating with the SNMP agent implemented network devices. Its main functions include querying agents, setting variables in agents, and acknowledging asynchronous events from agents. The managed devices are the network elements that require monitoring and management (Nidhishree & Manimala, 2013). SNMP agents are programs packaged within the network element that allow them to collect the MIB from the devices locally and make it available to the manager when queried for. The agents can either be standard (Net-SNMP) or vendor specific (HP insight agent). Functions of the agents include collection of management information about its local environment, storing and retrieving information based on the MIB and signaling an event to the manager. The MIB is a collection of information for managing network elements (Safrianti et al., 2021). Figure 2.1 shows the relationship between SNMP components.

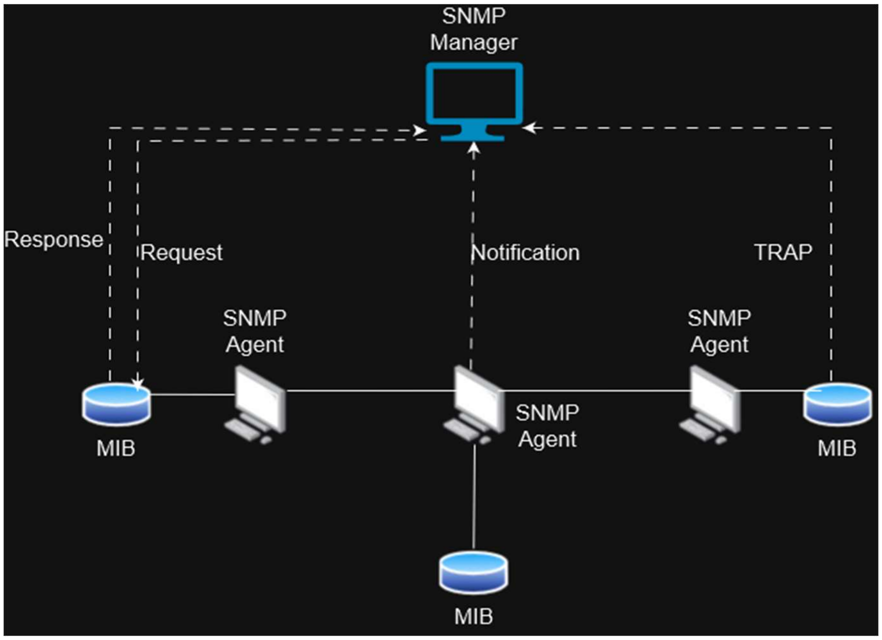


Figure 2.1: Relationship between SNMP components

2.1.2 SNMP Operation

SNMP operates on a client-server model and is part of TCP/IP protocol suite. The agent runs an SNMP server process, which collects and stores management information. The SNMP manager communicates with these agents to retrieve information requested or issue commands. SNMP uses MIBs to organize and define the structure of the data to be managed. MIBs contain a hierarchical collection of managed objects, each identified by an Object Identifier (OID). These objects represent various aspects of the device's configuration, status, and performance. The SNMP protocol operates over UDP (User Datagram Protocol) on port 161 for queries and port 162 for traps (asynchronous notifications) (Safrianti et al., 2021). Figure 2.2 is a four–layer model developed by the Department of Defense (DoD) (ManageEngine OpManager, 2024).

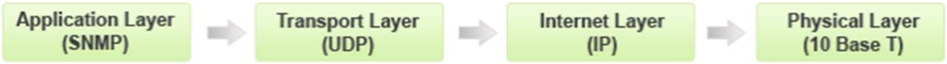


Figure 2.2: Four-layer model by DoD (ManageEngine OpManager, 2024)

The relationship between SNMP agents and managers is shown in figure 2.3. (Zhou et al., 2015)

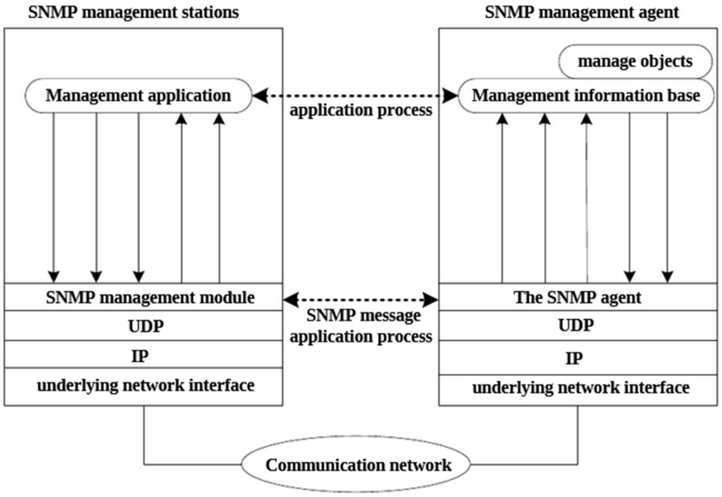
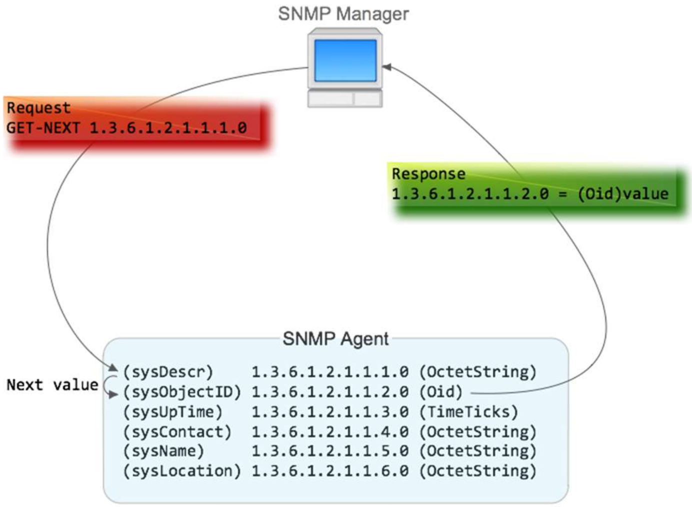


Figure 2.3: Relationship between SNMP agent and manager (Zhou et al., 2015)

2.1.3 MIB structure and Object Identifier

The SNMP MIB is a local database containing information relevant to network management and is utilized by both SNMP managers and agents. SNMP messages, created by agents and received by managers, are processed with reference to the MIB, allowing managers to interpret traps or messages from network devices. SNMP messages, utilized for monitoring network devices, rely on the SNMP protocol, and are typically generated by SNMP agents and received by SNMP managers. MIB is essentially a formatted text file that lists data objects used by network equipment. Each device's manufacturer provides a MIB file, which is loaded into SNMP managers to allow interpretation of messages (Mauro & Schmidt, 2005).

Objects in the MIB are identified by Object Identifiers (OIDs), which act as addresses for individual components in the network. The MIB translates these OIDs into human-readable text, allowing managers to understand and process SNMP messages effectively. An object ID is usually a dotted list of integers. An example of this is the OID in RFC1213 for "sysDescr" is .1.3.6.1.2.1.1.1, as shown below. The structure of the MIB is organized into a tree, with each object having a unique OID, enabling precise identification of network elements. Overall, the SNMP MIB serves as a codebook that translates numerical strings (OIDs) into human-readable text, enabling SNMP managers to effectively manage network devices (Mauro & Schmidt, 2005). Figures 2.4 and 2.5 are diagrammatic representations of the MIB tree structure and the OID as represented in the SNMP agent respectively.



```
(sysDescr)    1.3.6.1.2.1.1.1.0 (OctetString)
(sysObjectID) 1.3.6.1.2.1.1.2.0 (Oid)
(sysUpTime)   1.3.6.1.2.1.1.3.0 (TimeTicks)
(sysContact)  1.3.6.1.2.1.1.4.0 (OctetString)
(sysName)     1.3.6.1.2.1.1.5.0 (OctetString)
(sysLocation) 1.3.6.1.2.1.1.6.0 (OctetString)
```

Figure 2.4: OID as represented in SNMP Agent

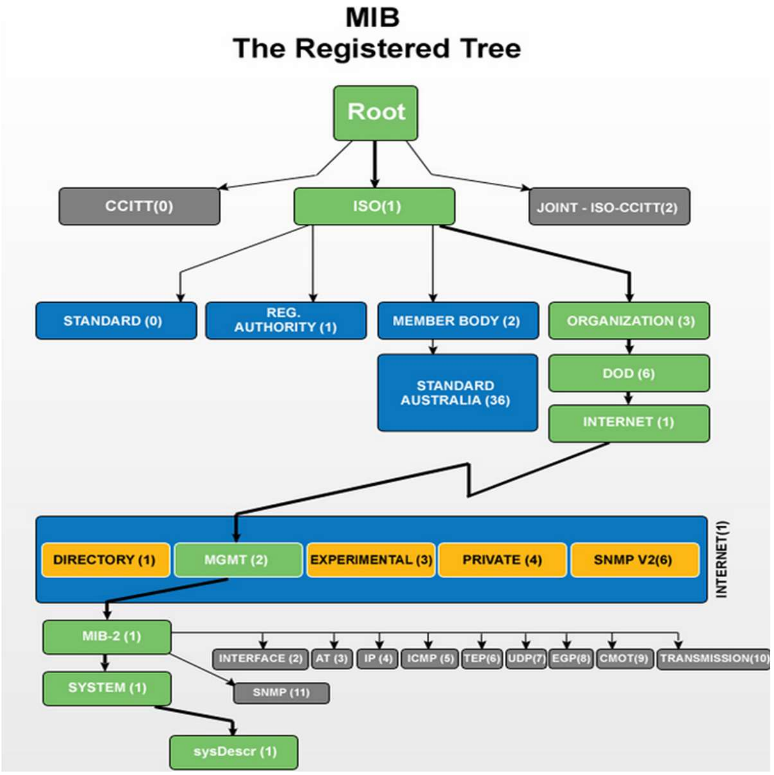


Figure 2.5: MIB Tree Structure (ManageEngine OpManager, 2024)

2.1.4 SNMP TRAPS

SNMP traps enable an agent to notify the SNMP manager of significant events, like hardware failure, by an unsolicited SNMP message. The current sysUpTime value, an OID indicating the kind of trap, and optional variable bindings are all included in SNMP Trap protocols. Destination addressing for SNMP trap port is determined in an application-specific manner typically through trap configuration variables in the MIB, but typically uses port 162. The format of the trap message was changed in SNMPv2, and the protocol data units was renamed SNMPv2-Trap (Mauro & Schmidt, 2005).

2.1.5 SNMP Basic Commands

GET: This is a request sent by the manager to the managed device and is performed to retrieve one or more values from the managed device.

GET NEXT: This is like GET. The difference is that the GET NEXT operation retrieves the value of the next OID in the MIB tree.

GET BULK: The GETBULK operation is used to retrieve voluminous data from large MIB tables.

SET: This operation modifies or assigns the value of the managed device.

TRAPS: TRAPS are initiated by the Agents unlike the above which are initiated by the manager. It signals to the SNMP Manager when an event occurs.

INFORM: This command is like TRAP initiated by the Agent, additionally INFORM includes confirmation from the SNMP manager on receiving the message.

RESPONSE: It is the command used to carry back the value(s) or signal of actions directed by the SNMP Manager (Mauro & Schmidt, 2005)

2.1.6 SNMP versions

SNMP has evolved through several versions. The main SNMP versions are SNMPv1, SNMPv2c, and SNMPv3 as explained below.

i. **SNMPv1 (Simple Network Management Protocol version 1)**: This was the original version of SNMP and uses a simple security scheme (passwords) to authenticate and authorize access to network devices. However, it is vulnerable to attacks such as eavesdropping.

ii. **SNMPv2c (Simple Network Management Protocol version 2 community-based)**: SNMPv2c was introduced to address some of the limitations of SNMPv1. It added new features like support for 64-bit counters, improved error handling, and the ability to retrieve multiple pieces of information in a single request. However, SNMPv2c still relied on passwords for security.

iii. **SNMPv3 (Simple Network Management Protocol version 3)**: SNMPv3 is the most recent and secure version of SNMP which introduced robust security features to address the vulnerabilities of previous versions. SNMPv3 provides message encryption, authentication, and access control, making it suitable for use in secure environments. It also introduced the User-based Security Model (USM) and the View-based Access Control Model (VACM), allowing for control over access to network devices. Having explored the different versions of SNMP, let us explore how SNMP gathers data from network devices.

2.1.7 SNMP polling algorithm

SNMP gathers data from network devices using a method called self-trapping polling. In this process, the network management workstation sends GetRequest and GetNextRequest packets to the agents on managed devices, which then respond with GetResponse messages. This collected information is displayed on the console, either numerically or graphically, providing insights into the operational status of network devices and enabling analysis of network traffic. Additionally, management agents can generate Trap messages to alert administrators about significant changes in the Management Information Base (MIB) or other critical events when devices exhibit abnormal behavior. When a device generates a self-trap, administrators can use the network management station to query the device's status to get more information.

2.2 Related Works

Case 1: Studies on SNMP implementation

An application for management and monitoring the data centers based on SNMP (Roohi et al., 2024) gives insight into the use of Computer and Network Asset Management (CNAM) which is a network management software that helps large enterprises and SMEs service providers, manage, and monitor their equipment and IT structure efficiently. CNAM collects information on all hardware components of the network instruments and will start to monitor them based on SNMP. It explores the implementation of SNMP-based network management systems in SMEs.

Performance Analytics of Network Monitoring Tools (Chahal et al., 2009) is a study on 15 SNMP-based network management and monitoring tools including, but not limited to, Nagios, Zabbix, SolarWinds, Cacti and Kiwi Monitor. This study compares the tools according to license, data storage method, access control, platform, logical grouping, and distributed monitoring. This shows how extensively SNMP is utilized in network monitoring and management.

Case 2: Effectiveness of SNMP tools

Performance Analytics of Network Monitoring Tools (Chahal et al., 2009) shows the effectiveness of the reviewed tools in 2.2.1 based on the features they offer. Examples of this include Cacti, which has robust and powerful functions, can be extended for collecting queries and scripts and can also initiate SNMP polling. SNMP-based tools adopt multiple technologies like PHP, MySQL, SNMP and RDDT thus producing good interactive interfaces which are convenient to managers and provide automatic display mechanisms for viewing graphs with web interface.

An Efficient Network Monitoring and Management System (Khan et al., 2013) is an empirical study that evaluates the effectiveness of network management systems utilizing SNMP protocols. In this study, Nagios is configured to assess the performance of network devices. The conclusion of this study shows that open-source solutions such as Nagios prove effective for network management.

Case 3: Comparison of SNMP Tools with Proprietary Solutions

Comparative Study on Network Monitoring Tools (Manohar, 2020) is a comparative study that examines open-source SNMP-based network management systems against proprietary solutions in the context of SMEs. In this study, the tools Nagios, PRTG Network Monitor and SolarWinds are compared in terms of cost-effectiveness, scalability, performance, and support. From this study, the observation that open-source solutions like Nagios provide a base for the network monitoring system to be built on by the end user. However, proprietary solutions like SolarWinds have a high price point for small business owners.

2.3 Gaps in Related Works

Most SNMP-based network management programs have scalability issues when handling large-scale networks, as explained in Challenges of Implementing Network Management Solution (Rao, 2011). These tools' performance could decline dramatically as the number of devices rises, which can cause delays in the gathering and processing of data. Therefore, better algorithms and architectures are required that can manage large-scale networks effectively without sacrificing performance.

SNMP, particularly versions 1 and 2c, has well-documented security weaknesses such as weak authentication methods and vulnerability to attacks like SNMP enumeration and brute force attacks. The SNMP Vulnerabilities Frequently Asked Questions (FAQ) document by Carnegie Mellon University from 2017 emphasizes that the use of community string is a weak authentication mechanism, leaving SNMPv1 and 2c susceptible to attacks. Despite the availability of SNMPv3, which provides better security features, many tools either do not fully support it or fail to utilize its security capabilities. There is thus a need for network management tools that fully incorporate SNMPv3 to guarantee secure management of network devices.

Numerous existing SNMP management tools have complex and confusing user interfaces, posing challenges for network administrators to effectively utilize. According to Usability Matters: A Human-Computer Interaction Study on Network Management Tools (Verde et al., 2020), many existing management applications feature poor interface resources due to lack of knowledge regarding user profiles. This proves that there is a need for user-friendly interfaces that simplify network management tasks, offering better visualization of network status and intuitive configuration options.

2.4 Conceptual framework

The conceptual framework for VelocityNetHub focuses on creating a comprehensive user-friendly, network management system. The framework is divided into three main layers: Data collection layer, management layer and the user interface layer. Figure 2.5 shows these three layers, their components and how they interact with one another.

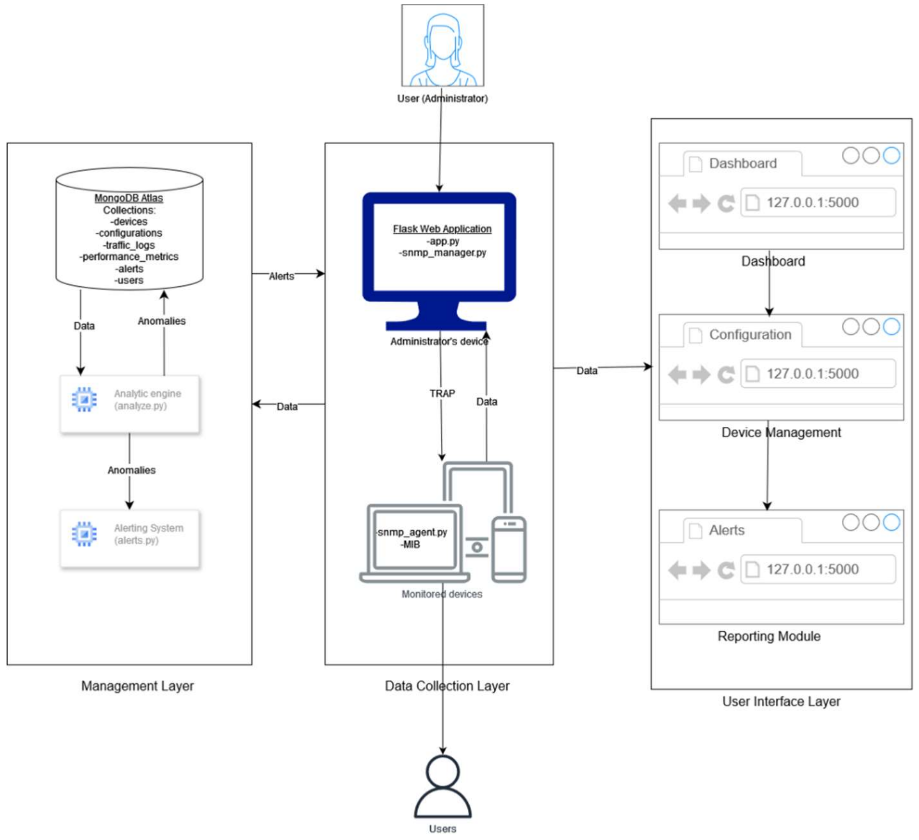


Figure 2.6: Conceptual framework

The components in the data collection layer are the SNMP agents, managers and MIBs. The MIB is a database schema found on all network devices that defines management data. The agent gathers data from the networking devices and the manager communicates with the agents to gather and store management information. The management layer is comprised of data storage, which stores collected SNMP data, analytic engine, which processes and analyses data to find patterns and anomalies from collected data, and an alerting system which monitors the network for anomalies which trigger alerts based on set conditions. The user interface layer consists of an admin dashboard, device management interface and a reporting module. The dashboard gives an overview of the network status and alerts. The device management interface allows the administrator to configure network devices and manage settings. The reporting module gives reports on the network performance and usage.

The interactions between the components give this model its functionality and enables the proper management of a network. Figure 2.5 highlights the different functionalities of VelocityNetHub.

For device discovery, the manager sends discovery requests periodically to identify new devices with agents. The flow of data starts from the manager to the agent which then responds to the manager, then the manager stores the information. Data collection involves the self-polling process where the manager polls the agents to collect performance metrics and status information. The data flow is similar to the discovery process. Alerting involves the continuous monitoring of data in storage and when the data exceeds predefined limits, an alert is triggered. The flow of data is from the storage to the analytic engine, which then triggers the alerting system which makes the manager send an asynchronous message (TRAP) to the agent. This alert is also shown on the dashboard. Device Configuration allows users to remotely configure and manage network devices. Through the device management interface, users can change device settings and apply configuration changes. These user inputs are sent to the SNMP Manager, which then communicates the configuration commands to the SNMP Agents on the devices.

Chapter 3: Methodology

3.1 Introduction

This chapter outlines the methodology adopted for the development of VelocityNetHub using SNMP for SMEs. It provides a summary of the contents covered in each section.

3.2 Methodology

Design Thinking has been selected as the project's methodology. According to Brown (2008), design thinking is a human-centered, non-linear, iterative approach to innovation that emphasizes understanding people, questioning presumptions, reframing issues, thinking of original solutions, and quickly developing and testing ideas. The five steps are: Define, Ideate, Prototype, Test, and Empathize.

3.2.1 Justification of the Methodology

Given that it prioritizes ongoing input, iterative problem-solving, and empathy for end users, Design Thinking is a good fit for this project. Design Thinking promotes a thorough comprehension of customer requirements and difficulties, resulting in creative and approachable solutions (Interaction Design Foundation, 2024). This is in line with the project's objectives, which include creating a network management tool that is customized to meet the unique needs of SMEs.

3.2.2 Methodology Diagram

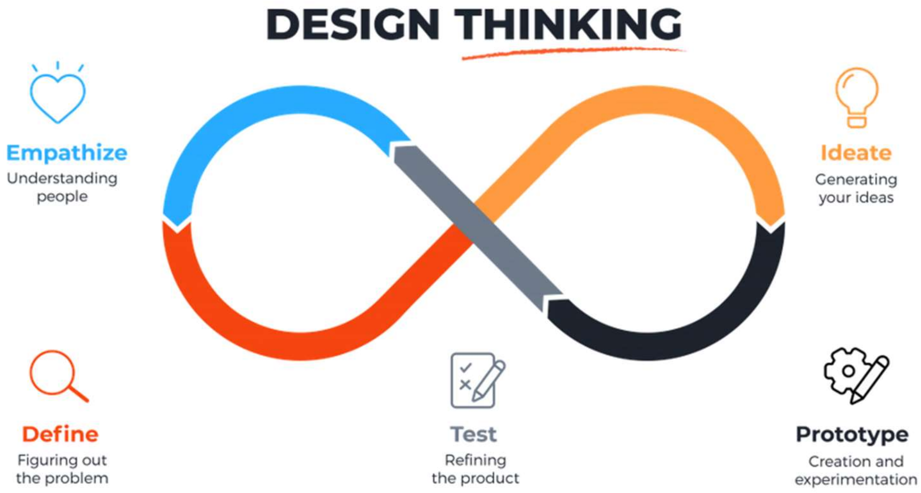


Figure 3.1: Design Thinking Diagram (Karl, 2020)

3.2.3 Applying Stages to the Project

i. Empathize

The Empathize step is crucial in developing a network management tool for SMEs, as it involves understanding their unique network administration challenges, such as limited technical expertise, constrained financial resources, and the need for user-friendly solutions. By identifying common issues like connectivity problems, slow performance, and security vulnerabilities, and by designing cost-effective, easily operable tools that can scale with SME growth, developers can address these pain points. Gathering insights from secondary data sources such as market research reports (Brown, 2008) case studies, and industry analyses highlights prevalent issues like the lack of automated monitoring and difficulties in managing configurations. By integrating these insights, developers can ensure the tool includes proactive monitoring and simplified configuration management. Ultimately, understanding SMEs' challenges through thorough research enables the creation of a functional, accessible, and cost-effective network management solution.

ii. Define

The Define step is crucial in the development process of a network management tool which involves establishing key model or solution parameters and factors based on secondary data analysis. This stage helps set precise project objectives and goals, ensuring that the final product meets the specific needs of SMEs. By thoroughly studying market research reports, case studies, and industry analyses, developers can identify essential features and requirements, such as proactive monitoring, simplified configuration, and reporting. This step ensures that the tool is designed to address the unique challenges faced by SMEs, such as limited technical expertise and constrained financial resources. Establishing clear parameters and goals at this stage guarantees that the developed product effectively enhances network management capabilities for SMEs.

iii. Ideate

The Ideate stage entails ideation and brainstorming for the network management tool's layout and features. The needs and preferences of SMEs inform the design approach selection process. A variety of perspectives on the suggested solution are investigated, taking user experience and scalability into account. Because it prioritizes satisfying end users’ needs and preferences, user-centered design has been selected as the design paradigm.

To illustrate the overall system structure and the flow of network management processes, sequence diagrams and system architecture diagrams are drawn. These diagrams guarantee that the system design is efficient and coherent and help in the understanding of the interactions between various modules. Figures 3.1 and 3.2 depict the system architecture and the sequence.

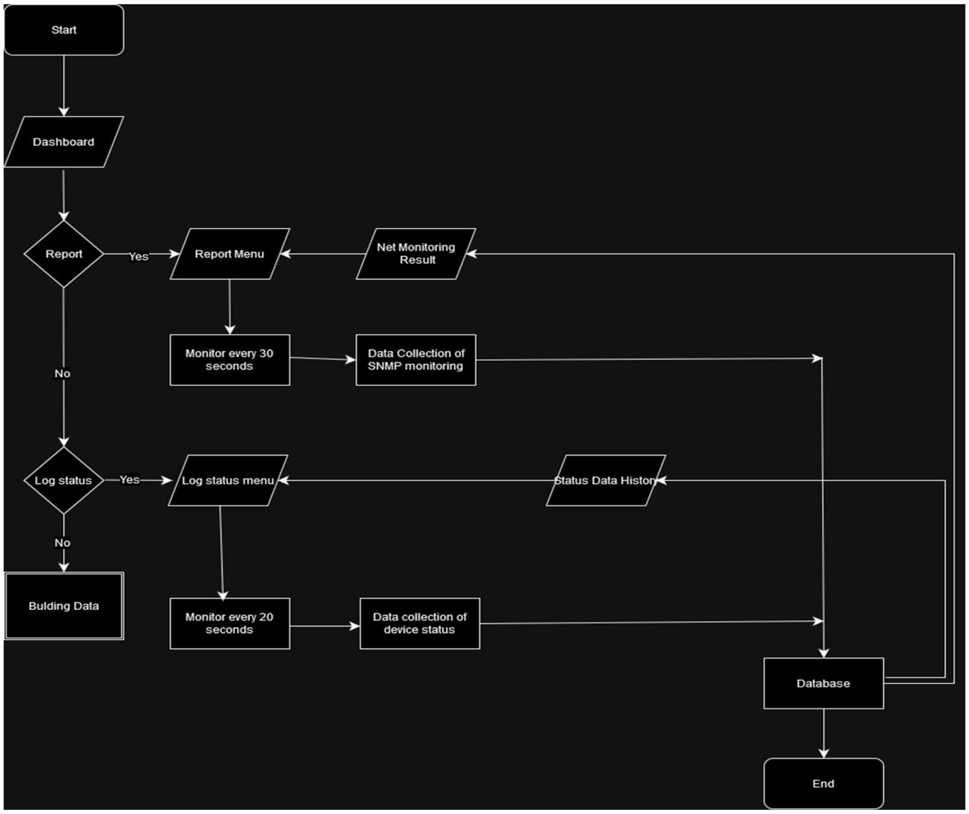


Figure 3.2: VNH Sequence design.

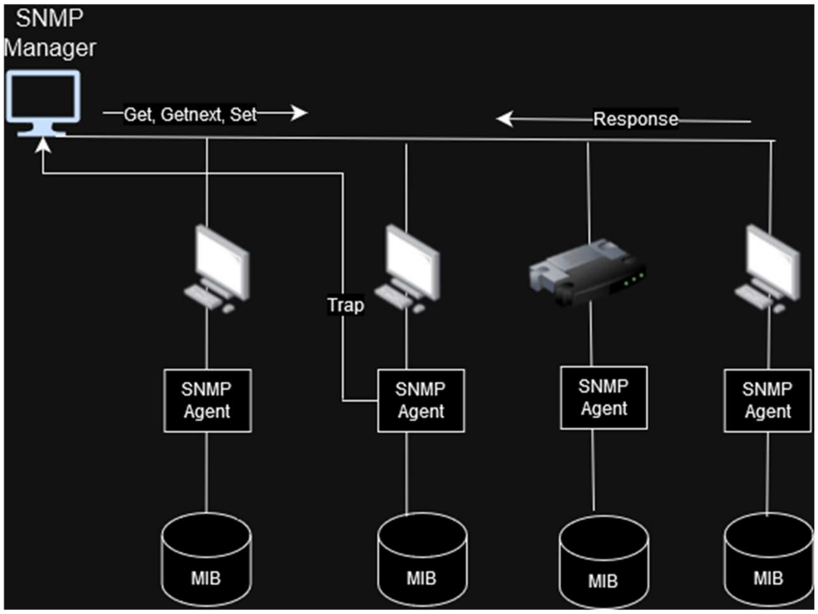


Figure 3.3: VNH System Architecture

iv. Prototype

The prototype stage entails creating a network management tool prototype based on the defined specifications and design ideas. Iterative and incremental development is the chosen methodology, enabling ongoing enhancement and integration of feedback. Python, Flask, and MongoDB are just a few of the development tools and methodologies that have been chosen to make the prototype development and testing process more productive. These tools facilitate efficient development, testing, and iteration of the prototype, ensuring that it aligns closely with the identified requirements and design concepts.

v. Test

During the Test stage, the prototype undergoes testing to ensure it meets the requirements and functions as intended. Testing techniques, including unit, blackbox, and whitebox testing, are employed to ensure the reliability of the prototype (Geeks for Geeks, 2024). Multiple testing scenarios and types, such as accuracy testing and model prediction performance testing, are conducted to identify and address any problems or bugs. Test cases are created to cover a range of use cases, ensuring comprehensive testing of the prototype's functionality and performance. These test cases include verifying accurate device discovery to detect all network devices and active connections, testing monitoring alerts to ensure timely detection and notification of network issues, validating configuration changes to confirm proper application and reflection in device settings, assessing reporting accuracy to ensure generated reports reflect the current network state accurately, and evaluating user interface usability to guarantee an intuitive and error-free user experience. This thorough testing process aims to validate the prototype's reliability, accuracy, and adherence to the defined specifications before proceeding to the next stage of development.

3.3 Deliverables

The deliverables for this project include:

i. **Model and User Interface**: A network management tool with a user-friendly interface for monitoring and managing network devices.

ii. **Proposal**: A detailed proposal outlining the project objectives, scope, methodology, and expected outcomes.

iii. **Detailed final report**: A final report describing the development of the proposed tool.

iv. **System diagrams**: Diagrammatic depictions of the processes that the proposed tool undergoes.

3.4 Tools and Techniques

Tools and techniques to be used in this project include:

i. **Python**: Python is chosen for backend development due to its simplicity and extensive libraries. It is used to implement the core functionalities of the network management tool, such as data collection from network devices, analysis of network traffic, and generation of reports.

ii. **Flask**: Flask is a lightweight web framework that simplifies the development of web-based user interfaces for the network management tool. It provides tools for routing HTTP requests, rendering templates, and handling user authentication. It is used to create a user-friendly web interface for interacting with the network management tool, allowing users to view network status, configure devices, and generate reports through a web browser.

iii. **MongoDB**: MongoDB is a NoSQL database management system that offers flexibility and scalability for storing network monitoring data. It is used to store network monitoring data, such as device configurations, traffic logs, and performance metrics.

iv. **Git**: Git is used to track changes in the source code of the network management tool, create branches for feature development.

v. **Pytest**: Pytest is a testing framework for Python that simplifies the process of writing and running automated tests. It is used to write and execute automated tests for the various components of the network management tool, including backend functionalities and web interface interactions.

vi. **Docker**: Docker is used for containerization, packaging, and deploying the network management tool in a consistent and isolated environment. It is used to package the network management tool and its dependencies into lightweight, portable containers which can then be deployed on any Docker-compatible host, simplifying deployment, scaling, and management of the tool.

References

Brown. (2008). *Design Thinking.* Harvard Business Review.

Carnegie Mellon University. (2017). *Simple Network Management Protocol (SNMP) Vulnerabilities Frequently Asked Questions (FAQ).* Carnegie Mellon University. Retrieved from https://insights.sei.cmu.edu/documents/541/2003_019_001_497195.pdf

Chahal D., Kharb L., &Choudhary D. (2009). *Performance Analytics of Network Monitoring Tools.* Blue Eyes Intelligence Engineering & Sciences Publication. Retrieved from https://www.researchgate.net/publication/348325775_Performance_Analytics_of_Network_Monitoring_Tools

Datadog. (2024). How SNMP Works: Simple Network Management Protocol. *Knowledge Center*. Retrieved from https://www.datadoghq.com/knowledge-center/network-monitoring/snmp-monitoring/

Frontier Business Products. (2022). What Is Network Management and Why It Is Important? *News*.

Tarutė & Gatautis (2013). *ICT impact on SMEs performance.* Kaunas, Lithuania: Elsevier Limited. Retrieved from https://pdf.sciencedirectassets.com/277811/1-s2.0-S1877042814X00042/1-s2.0-S1877042813056085/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEPL%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCXVzLWVhc3QtMSJGMEQCIGGm%2BILCA%2FwGJqWHWwRD3nQ01F75vJr91GAE0T3sP4i9AiBvKmR6rprf

Geeks for Geeks. (2024, May 09). *geeksforgeeks.org*. Retrieved from geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/: https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/

Interaction Design Foundation. (2024). *interaction-design.org*. Retrieved from interaction-design.org/literature/topics/design-thinking: https://www.interaction-design.org/literature/topics/design-thinking

Karl. (2020, April 20). *maqe.com*. Retrieved from maqe.com/insight/the-design-thinking-process-how-does-it-work/: https://www.maqe.com/insight/the-design-thinking-process-how-does-it-work/

Khan R., Khan S., Zaheer R., & Babar M. (2013). *An Efficient Network Monitoring and Management.* International Journal of Information and Electronics Engineering. Retrieved from http://www.ijiee.org/papers/280-N011.pdf

ManageEngine OpManager. (2024). *SNMP Protocol*. Retrieved from www.manageengine.com: https://www.manageengine.com/network-monitoring/what-is-snmp.html

Nidhishree & Manimala. (2013). *Design and Implementation of SNMP based Network Device Monitoring System.* Mysore, India: International Journal of Trend in Research and Development. Retrieved from https://www.ijtrd.com/papers/IJTRD9632.pdf

Manohar, V. (2020). *Comparative Study on Network Monitoring Tools.* Visakhapatnam, India: International Research Journal of Engineering and Technology (IRJET). Retrieved from https://www.irjet.net/archives/V7/i4/IRJET-V7I464.pdf

Rao, U. (2011). *Challenges of Implementing Network Management Solution.* Bhubaneswar: International Journal of Distributed and Parallel Systems. Retrieved from https://www.airccse.org/journal/ijdps/papers/0911ijdps06.pdf

Roohi A., Raeisifard K., & Ibrahim S. (2024). *An application for management and monitoring the data centers based on SNMP.* Penang, Malaysia: IEEE Student Conference on Research and Development. Retrieved from https://ieeexplore.ieee.org.ezproxy.library.strathmore.edu/document/7072941

Safrianti E., Sari L.,& Sari A. (2021). *Real-Time Network Device Monitoring System with Simple Network Management Protocol (SNMP) Model.* Surabaya, Indonesia: 3rd International Conference on Research and Academic Community Services (ICRACOS). doi:10.1109/ICRACOS53680.2021.9701973

Sathyan & Jithesh. (2016). *Fundamentals of EMS, NMS and OSS/BSS.* CRC Press Taylor&Francis Group. Retrieved from https://books.google.co.ke/books?id=jAbLBQAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Schmidt & Mauro. (2005). *Essential SNMP: Help for System and Network Administrators.* O'Reillt Media, Inc. Retrieved from https://books.google.co.ke/books?hl=en&lr=&id=65_0d25EpB4C&oi=fnd&pg=PT7&dq=SNMP&ots=H0pWyqUdgO&sig=G4oPqnFmxUduL7v--69l_VjfweE&redir_esc=y#v=onepage&q=SNMP&f=false

Schönwälder J., Pras A., Harvan M., Schippers J. & Meent R. (2007). *SNMP Traffic Analysis: Approaches, Tools, and First Results.* Munich, Germany: IEEE. doi:http://dx.doi.org/10.1109/INM.2007.374797

Splunk. (2023). SNMP & SNMP Monitoring, Explained. *Learn*. Retrieved from https://www.splunk.com/en_us/blog/learn/snmp-monitoring.html

Taghizadeh-Hesary & Naoyuki (2016). *Major Challenges Facing Small and Medium-sized Enterprises.* Tokyo: Asian Development Bank Institute. Retrieved from https://www.adb.org/sites/default/files/publication/182532/adbi-wp564.pdf

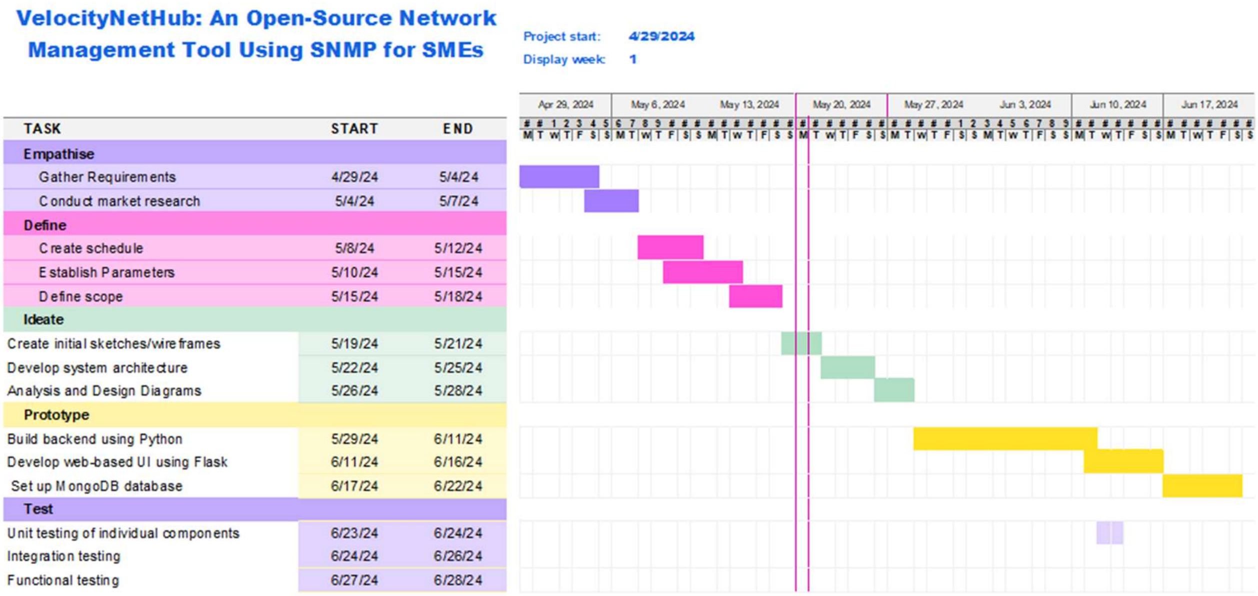
Verdi F., Oliveira H., Sampaio N., & Zaina L. (2020). *Usability Matters: A Human-Computer Interaction Study on Network Management Tools.* IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMEN. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9063640

Zhou Y., Deng M., Ji F., He X. & Tang Q. (2015). *Discovery Algorithm for Network Topology Based on SNMP.* SiChuan China: International Conference on Automation, Mechanical Control and Computational Engineering. doi:http://dx.doi.org/10.2991/amcce-15.2015.290

Zhu. (2021). Self-Organized Network Management and Computing of Intelligent Solutions to Information Security. *Journal of Organizational and End User Computing*. Retrieved from https://www.igi-global.com/viewtitle.aspx?TitleId=285515&isxn=9781799867494

Appendices

Appendix 1: Gantt Chart



VelocityNetHub: An Open-Source Network Management Tool Using SNMP for SMEs

Project start: 4/29/2024

Display week: 1

TASK	START	END
Empathise		
Gather Requirements	4/29/24	5/4/24
Conduct market research	5/4/24	5/7/24
Define		
Create schedule	5/8/24	5/12/24
Establish Parameters	5/10/24	5/15/24
Define scope	5/15/24	5/18/24
Ideate		
Create initial sketches/wireframes	5/19/24	5/21/24
Develop system architecture	5/22/24	5/25/24
Analysis and Design Diagrams	5/26/24	5/28/24
Prototype		
Build backend using Python	5/29/24	6/11/24
Develop web-based UI using Flask	6/11/24	6/16/24
Set up MongoDB database	6/17/24	6/22/24
Test		
Unit testing of individual components	6/23/24	6/24/24
Integration testing	6/24/24	6/26/24
Functional testing	6/27/24	6/28/24

Appendix 2: Marking Guide

Strathmore University

School of Computing and Engineering Sciences

Project Proposal Assessment Guide

Student Number:	150150
Working Title:	VelocityNetHub: A Network Management Tool using SNMP for SMEs

Evaluation Areas	Weight	Score	Notes
Title page: Informative, concise, and appropriate	2 pts		
Abstract To have background, problem, solution, methodology (approach data and tools) outcomes and expectations	2 pts		
Introduction Background (2) *A clear illustration of issue, context and audience* Problem Statement (2) *Pain points, audience, who is affected and how solution comes in to fix the pain.* Objectives (S.M.A.R.T and Linked to Problem Statement) (2) Research questions (1) *Alignment of questions with objectives* Justification (2) *Should be research supported.* Scope of Project (2)	(13 pts)		

Specify boundaries of people process, HW/SW, data etc. Limitations (1) *Challenges Expected* Delimitation (1) *To do to counter anticipated challenges.*			
Literature Review/Related Work Objectives mapping to Literature Review (2) Critique of Theoretical framework and content adequacy (2) *Principles, parameters of consideration* Discussion of technologies contextualization for the proposed work (2) Citations of content and alignment to work (2) Review of at least 3 systems comprehensively the working behind it (2) Gaps identification, analysis relative to the proposed solution (1) Conceptual Framework clear to communicate how it works, data flows, processing, actors (3) *Diagram that's clear; discussion of diagram.* *Describe input process output storage boundaries.*	(14 pts)		
Methodology Methodology and justification (2) Correct Methodology Application (1), Design and Development tools (2) Deliverables and milestones (2) Examinable bits from ideation Proposal, design, test cases documentation doc	(8 pts)		

Proof of concept- modules Gantt Chart that makes sense relative to the project (1)			
Proposal Presentation Table of Contents and List of Figures (2) Are relevant references provided and formatted correctly? (2) Is there a clear and proper use of language? (1) Effective report structure (chapters and sections) and layout (2)	(6 pts)		
Total Marks	45		

Verdict (Please tick) □ Accept □ Reject

Comments (Reasons for Reject/Accept)