



PUSL3119 Computing Individual Project

Project Interim Report

Network First Aid Box
(Network Management Sys)

Supervisor: Mr. Chamara Dissanayake

Name: Galappaththi Madubhashitha
Plymouth Index Number: 10749058
Degree Program: BSc(Hons) Computer Network

Contents

1. Intro
 - 1.1.Introduction
 - 1.2.Problem definition
 - 1.3.Project objectives
2. System analysis
 - 2.1.Facts gathering techniques
 - 2.2.Existing systems
 - 2.3.Use case diagram
 - 2.4.Drawbacks of the existing systems
3. Requirements specification
 - 3.1.Functional requirements
 - 3.2.Hardware/Software requirements
 - 3.3.Networking requirement
4. Feasibility study
 - 4.1.Operational and technical feasibilities
5. System architecture
 - 5.1.Class diagram
 - 5.2.High-level architecture
6. Development tools and technologies
 - 6.1.Development methodology
 - 6.2.Programming languages and tools
 - 6.3.Third party components and libraries
7. Discussion
8. References

1. Intro

1.1. Introduction

As the world becomes more reliant on technology, network management has become increasingly important. In this digital age, businesses and organizations depend on networks to keep their operations running smoothly. Network engineers play a crucial role in maintaining the health and security of these networks.

In modern days, network management systems are critical components of modern IT infrastructure. As networks grow in complexity and size, the management of those networks becomes increasingly challenging. Network management systems help network engineers manage their networks by providing them with tools to monitor and control various network devices, troubleshoot problems, and plan for future network upgrades.

Python has become one of the most popular and most valuable programming languages for network management systems due to its ease of use, versatility, vast number of libraries and frameworks available for network management. Python has also gained popularity due to its support for machine learning and artificial intelligence, which can be used to build smart network management tasks.

Smart network management systems take advantage of advanced technologies like artificial intelligence and machine learning to automate many of the tasks that network engineers would typically perform manually. By automating many tasks, smart network management systems can help IT professionals to reduce the time spent managing networks, increase network availability, and improve network security.

In this project, I am going to explore the benefits of using Python for network management systems, and how smart management systems can help network engineers manage their network more efficiently. And also developing network managing tools for smart network management.

So I am developing a smart network management system, some kind of network tools to check if the hosts are up or down using ping and monitoring them, gather the physical location of a specific network device using its IP address, tool to do some basic troubleshooting methods such as (ipconfig, ping, tracert etc.) within the system without going to CMD, and tool to

configure network devices. So this project is a mix of some kind of simple network tools.

In summary, this project will demonstrate the importance of Python in network management, explore the benefits of smart network management tools and provide a practical example of how these tools can be developed using Python. The project will provide an opportunity for network engineers to learn about network automation and how it can be used to improve network management efficiency.

This project is totally depending on networking automation by using python language.

1.2. **Problem definition**

The management of modern network can be a bit challenging and time consuming task. Network engineers responsible for managing an ever-growing number of network devices and applications, each with their own set of configurations and management requirements. The sheer volume of data generated by these devices can be overwhelming, making it difficult to identify and troubleshoot network problems.

In addition, the complexity of modern networks can make it difficult to plan for upgrades and changes. Network engineers must carefully balance the need for network performance and availability with the need for security and stability. This can be a difficult task, particularly in large and complex networks.

Traditional network management system can help network engineers to manage their networks by providing them with tools to monitor and control various network devices. However, these systems can be time consuming to set up and maintain, and may not be able to keep up with the pace of change in modern networks.

The problem of network management is a critical one, as the performance and availability of modern networks are essential for the success of companies in all industries. IT professionals need a better way to manage their networks, one that is both efficient and effective.

And we know there are a lot of networking management systems, tools. But the thing is those are not open source and network engineers have to spend a lot of money to buy them. Also another thing is there some companies they have

small network infrastructures and they don't need buy large scale of network management system to manage their network. They need a smart and smooth network management systems to do day-to-day troubleshooting and manage issues.

To address these challenges, there is a growing need for open source network management systems that are flexible, scalable, and cost-effective. Open source solution can be customize to meet the specific needs of an organization and can be integrated with other tools and applications. They also offer a community of development and improvement of the system.

1.3. **Objectives**

The solution to the above problems there should be a smart network management system. These systems can automate many of the tasks involved in network management, freeing up network engineers to focus on high-level tasks. Smart network systems can also use machine leaning to analyze network data and predict potential problems before they occur, improving network availability and reducing downtime. The development of smart network management systems that leverage the power of python and machine learning is an important step forward in solving above problems.

The primary objective of this project is to create a user-friendly network management system that can automate various network management tasks and provide a simple, intuitive interface for network engineers. The system will be designed to reduce the complexity of network management tasks, enabling users to easily monitor, to troubleshoot and configure network devices.

So in my network first aid box project, am developing user friendly user interfaces. Users can create an user account and login to the system using their email and the password. And then it will display the network first aid box's main dashboard. On the dashboard users can choose whatever the tool the want to execute. In this case am developing hosts monitoring tool, users can give their IP range and quickly look which devices are up in the network and which devices are still down. And another tool is location gathering tool, users can upload a map of the location of network devices and retrieve the maps whenever they want to see that maps. There is another tool, it Is troubleshoot tool, users can use this tool to execute basic troubleshoot commands within the network first aid box system without going to the command prompt, also users can download and retrieve the information of that troubleshoot as a PDF report. Finally am going to develop a tool to do the configurations of network

switches and routers, users can easily do the configurations using this network first aid box system.

Those are the main objectives of my project and hope to use most suitable technologies to do this project.

2. System analysis

2.1. Facts gathering techniques.

System analysis is an important phase in the software development process, which involves understanding and identifying user requirements, constraints and objectives for the software system. As a part of system analysis, facts gathering techniques are employed to gather relevant information that can be used to develop a system that meets the user's needs.

There are several facts gathering techniques that can be used to collect information about user requirements. Some of these techniques include interviews, surveys, observation, document analysis and prototyping. These techniques can be used in combination to obtain a comprehensive understanding of the user's needs.

First I had no idea about what project I am going to do and what kind of project that suitable for me and related to network industry. Then I gone through some web sites by using google search engine. Then I could be able to recognized some projects that are related to computer networks. There was vast amounts of network related projects university students have done. I studied some kind of several projects and I could be able to find some special category based on network management systems. Then I did further more researches about network management systems going through some YouTube videos, because I am very friendly with watching videos rather than reading books or reading research papers. But at some points I had to go through some research papers.

Then a friend one of my friends said about networking automation. After hearing that I did some research about network automation going through online articles and YouTube videos.

On those days I went for a job as a networking assistant. It was a small IT solution company, and I had a networking team to work. I got some information from them about network management systems and network automation. From that job I could be able to deal with some network tools. In that case I got some ideas about networking tools.

And also I followed videos from YouTube, online articles and got better understanding about what is network tools, what kind of network tools that regularly network engineers use in network industry, what kind of network management systems use in industry such as Nagios, Zabbix, PRTG.

Then I had to find out how to create network tools and which languages, which technologies that I wanted to refer. So again I gone through YouTube videos and official documentation from software vendors and communities can provide in depth information about the software and tools used in network management and automation.

And the next thing is I have never learnt about python as language. So in this case again there are lot of tutorials about python on YouTube and there are lot of online research papers and online articles related to python, I gone through these kinds of online resources to get better ide about python and how to implement proper python technologies, codes to develop network tools. There was another headache on me about how to generate python GUIs. So in this case I searched about what kind of technologies, what kind of designing environments use to generate python GUIs and I used to watch again YouTube videos, tutorials and gone through some online articles.

There were some online resources that helped me very much, those are tutorials point, stack overflow, google scholar etc.

2.2. **Existing systems**

I early mentioned about what kind of network management systems that I was going through while I was doing my research about network management systems such as Nagios, Zabbix, PRTG. These kind of existing network management systems are essential for the efficient and effective functioning of complex computer networks. The primary function of an network management system is to monitor, manage, and optimize the network infrastructure, including hardware devices, software applications, and the data that travels across them. Traditional network management systems solutions, however, are no longer sufficient in meeting the demand of modern networks, which require greater flexibility, scalability and automation. This has led to the emergence of smart network management systems that leverage advance technologies like artificial intelligence, machine learning and automation to improve network performance, reduce downtime and enhance security.

Smart network management systems enable network engineers to proactively detect and address network issues before they become major problems. For instance, an smart management systems can use machine learning algorithms to analyze network traffic patterns and detect anomalies that may indicate a potential security breach or hardware failure. When an issue detected, the smart network management system can automatically generate alerts, assign tickets to IT staff and even initiate remediation actions like isolating affected devices or rerouting traffic. This can significantly reduce the time and effort required to address network issues and minimize downtime.

One of the key benefits of smart management systems is that it can improve network visibility and provide granular insights into network performance. Smart management systems solutions typically include advanced monitoring tools that can collect and analyze data from across the network, including device logs, traffic flow data, and performance metrics. This data can be used to identify trends, troubleshoot issues, and optimize network configurations. For example, an smart network systems can provide real-time visibility into network bandwidth usage, allowing administrators to identify and address bandwidth-hogging applications or devices.

Another important feature of smart management systems is their ability to automate routine network management tasks. Automation can help reduce human error, improve consistency, and free up network staff to focus on more strategic initiatives. Also smart network management systems can automate tasks such as device discovery, configuration management and firmware updates, reducing the burden on network staff and ensuring that network devices are always up to date and configured correctly. In addition, automation can help improve security by enforcing consistent security policies and ensuring that devices are patched and updated regularly.

In conclusion, smart network management systems are the future of network management. Developing and maintaining smart network management systems requires a team of network professionals with diverse skills, including network engineers, software developers, data scientists and cybersecurity experts. Effective communication and collaboration are also critical to ensuring that the smart network management systems meets the need of the entire organization. As networks continue to become more complex and distributed, smart network management systems will play an increasingly important role in ensuring their efficient and secure operation.

2.3. Use case diagram.



2.4. **Drawbacks of the existing systems.**

While network management systems offer numerous benefits for large and complex networks, they may not be suitable for all organizations. One major drawback of network management systems is that they can be expensive to implement and maintain. This is especially true for small companies that may not have resources or budget to invest in network management systems. In addition, network management systems typically require specialized expertise to operate effectively, which may not be available within a small company.

Another potential drawback of network management systems is that they may be overkill for smaller networks. For example, a small company with only a few devices and applications may not require the advanced monitoring and automation capabilities provided by network management systems. Instead, a simpler network management system, such as basic monitoring tools or managed services from a third-party provider, may be more appropriate.

Another challenge for small companies is that network management systems may require significant changes to existing networks infrastructure and configurations, which can be disruptive and time consuming. This may require additional resources and expertise that small companies may not have readily available.

In addition, small companies may not have the same security concerns as larger organizations, which may make the advanced security features of network management systems less relevant. For example, a small company with a limited number of devices and users may be able to manage security threats using basic security tools and protocols.

Network management systems may not be flexible enough to accommodate the unique needs of small companies. These companies may require more customized and flexible solutions that can be adapted to their specific requirements and workflows. This may not be possible with off-the-shelf network management systems, which may be designed for large and more complex networks.

Another potential drawback of network management systems is that they only have a steep curve for inexperienced users. Small companies with limited resources may not have the time or personnel to learn how to use the system effectively, which can lead to inefficiencies and errors. And network management systems may require significant hardware and software upgrades, which can further

add to the end to the cost of implementation and maintenance. Small companies may not have the budget to invest in new hardware or software, which can limit their ability to adopt network management systems.

Network management systems may also be prone to technical issues and downtime, which can impact network performance and productivity. Small companies may not have the resources and expertise to quickly resolve these issues, which can lead to longer downtime and loss of revenue.

Finally small companies may need to carefully evaluate their network management needs and consider simpler and more cost-effective and open source simple network management systems with simple tools that can meet their requirements.

3. Requirement specification.

3.1. Functional requirements.

- The user must be able to create an account by entering their proper email address and valid password. The system must validate the user's email address and password and then create the account.
- The user must be able to enter their email and password to access the system. The system must authenticate the user's email and password before granting access to the other main features.
- After the user has logged in, the system must display a welcome page with options for the choose from.
- The user must be able to input a IP range to scan for active hosts. The system must scan the range of IP address and display the list of active hosts.
- The user must be able to enter an IP address and the system must display the approximate location of the IP address.
- The user must be able to access the troubleshoot feature, which will help diagnose and solve network issues.
- The user must be able to navigate between the different pages of the system using the back button.
- The system must handle errors gracefully, displaying clear error message to the user when something goes wrong. For example, if the user enters an invalid email address, the system must display a clear message telling the user what they did wrong.
- Users must be able to download and retrieve the troubleshoot reports whenever they needed from the troubleshoot tool.

- Users must be able to download and retrieve the location maps whenever they needed from the location gathering tool.

3.2. **Non-functional requirements.**

- The system should load quickly and provide a responsive user interface even when there is high network traffic.
- The user's email address and password should be stored securely and protected from unauthorized access. There should be an authentication method using firebase. The system should also be protected from attacks such as SQL injection, cross-site request forgery.
- The user interface should be intuitive and easy to use, with clear instructions and feedback for the user. The system should be accessible and should follow common user interface conventions.
- The system should be reliable and stable, with minimal downtime and no data loss or corruption. It should also be handle errors gracefully and provide helpful error messages to user.
- When coding, the code should be well-structured and easy to maintain, with clear documentation and comments. It should also be easy to modify or extend the system in the future.
- The system should be compatible with a range of platforms and device.
- The system should be able to handle large amounts of data and users, and should be designed to scale up as the user base grows.
- The system should be easily portable to different environments and platforms, with minimal configuration or customization required.

3.3. **Hardware/Software requirements.**

- Network equipment for a simple local area network.
- A windows operating system that supports the system to run.
- Python language to develop the system.
- PyCharm IDE to develop the python codes and get python libraries to create the system.
- Firebase to authenticate the user logins and firebase storage to store the location maps.
- Also need internet connectivity for some troubleshooting functions and for firebase authentication.

3.4. **Network requirements.**

- The system needs to be connected to a LAN to enable communication between the client and server. The LAN should support TCP/IP protocol and have reliable connections.
- The system needs to be connected to the internet to enable communication with firebase. A internet connection with adequate bandwidth is required to ensure smooth communication.
- The network should be ensured to prevent unauthorized access to the system. This can be achieved by implementing appropriate security protocols such as encryption, access control and authentication mechanisms.

4. **Feasibility study.**

4.1. **Operational and technical feasibilities.**

Operational feasibility and technical feasibility are important consideration for any network management system implementation. This involves assessing whether the network management system can be operated effectively within the existing organizational structure and whether the human resources for its operation are available. There are several factors must be considered, including human requirements, their skills, the type processes they have to access and backup plans.

It is important to consider the human requirements for the system, such as the number of people needed to operate it, their training needs, and their familiarity with technologies that am going to implement. The system should be designed with ease of use in mind, and training should be provided to users to ensure that they are comfortable with the system's features and functionalities. There should be user guidance.

The system should be design to user-friendly, with a simple and intuitive interface that is easy to navigate. Users should be able to perform common tasks quickly and easily, without requiring extensive technical knowledge. The system should also provide detailed documentation and support to users, ensure that they are able to resolve any issues they encounter.

The processes that employees can access for the system would depend on their job roles and responsibility. For example, software developers may need to processes related to system developing and repairing. And the network admins, network engineers may need access to processes related to handle and do the functionalities and access the features, through the system. IT support staff member may need access to processes related to system maintenance.

In addition, access to sensitive processes and information such as PDF reports of some troubleshooting, should be restricted to authorized personnel only to ensure data security and privacy. Proper access control mechanisms, such as user authentication and authorization, can be implemented to prevent unauthorized access to critical processes and data.

There are several risks associated with the network management systems, such as security breaches, data loss and system failures.

Security breaches can occur due to hacking attempts, phishing attacks or unauthorized access mention before. To mitigate this risk, can implement strong authentication mechanism, such as firebase authentication.

Data loss can occur due to hardware failures, software errors, or human error. to mitigate this risk, the organization can implement a backup and recovery system. such as using OneDrive to automatically backup data. regular backups can be taken and start offsite in case of disasters or hardware failure.

Hardware requirements include the necessary network equipment for a simple LAN, as well as a windows operating system that supports the system's operation. The system will also require a computer or server to run the Python code and adequate storage capacity to store any data or occurrence generated by the system. It is important to ensure that the hardware is reliable and able to handle the expected load.

Software requirements Include Python language and the PyCharm IDE to develop the system's code, as well as firebase to prevent dictate user logins and store location maps. The system should also be compatible with a range of platforms and devices to ensure accessibility for all users.

Network requirements include connectivity to remand that supports TCP/IP protocol and has reliable connections, as well as an Internet connection with adequate bandwidth to ensure smooth communication with firebase. Security

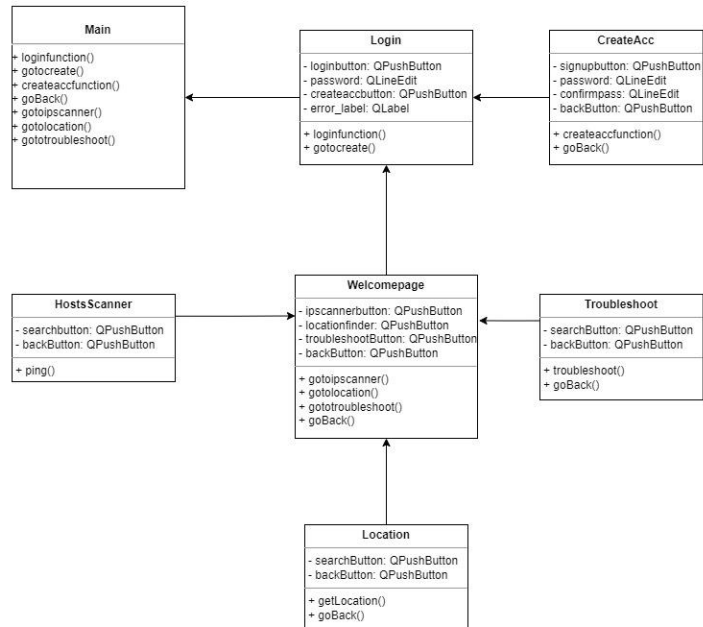
protocols that have encryption, access control and authentication mechanisms should also be implemented to prevent a rubberized access to the system.

In this project Python is the main technology that I'm going to use and by then provides libraries monitor network performance. Information there are user information and location maps to be stored in a database for reporting. Python provides libraries such as Pyrebase to connection between Python and firebase. and Python provides libraries like PyQt for GUI development.

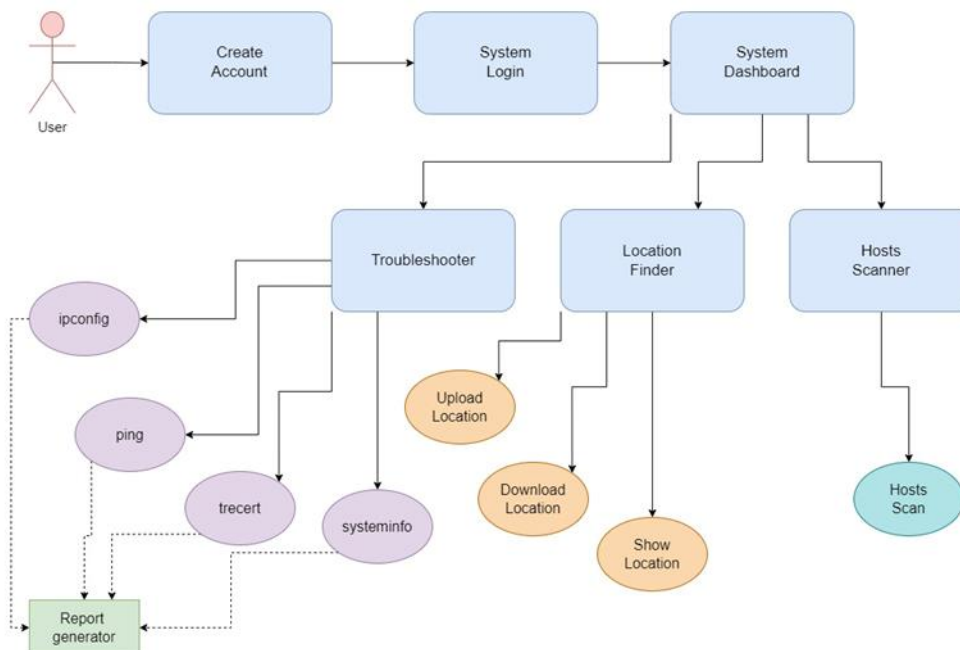
Now in terms of technical feasibility the proposed system appears feasible given the available hardware software and network infrastructure requirements.

5. System architecture

5.1. Class diagram of proposed system.



5.2. High-level architectural diagram.



6. Development methodology.

6.1. Development methodology.

Following waterfall methodology it can be good fit for the development of the network management system because it is a structured approach to software development that emphasizes planning and documentation, and is best suited for projects where the requirements are well-defined and no likely to change over the course of the project. The waterfall methodology for the network management system, the development process would follow a linear sequence of phases, where each phase must be completed before moving on to the next one. Here is an overview of the methodology,

- Requirement gathering -: gathering all the requirements for the network management system by researching.
- Design - : once the requirements are gathered, the design phase can begin. In this phase, design system architecture, the network requirements, user interfaces for the network management system.
- Implementation -: writing the code for the system by python and python libraries such as, pyrebase, os, PyQt, fpdf, webbrowser. And testing, debugging required to create the system.
- Testing -: after the implementation phase, the system is tested to ensure that it meets all the requirements and works correctly. This phase includes both functional and non- functional testing.
- Deployment -: once the system is tested, it is deployed to the production environment. This phase involves installing the system on the targeted machines and configuring it to work with the existing network infrastructure.
- Maintenance -: after the system is deployed, ongoing maintenance and support is required to keep the system running smoothly. This phase includes monitoring system for errors, making updates and improvements, and addressing any issues that arise.

6.2. Programming languages and tools.

This project is totally developing by using python language. Python is an high-level, general-purpose programming language that is widely used in a variety of fields, including network management tool development and network automation. Python has become one of the most popular programming languages due to its simplicity, ease of use, readability, and availability of

numerous libraries and modules. Python is an interpreted language, which means that it does not need to be compiled before execution. Python is an excellent language for developing network management tools and network automation due to its simplicity, flexibility and the availability of several libraries and modules specially designed for networking. Some of the main reasons why Python is widely used for network management, network developing tools and network automation are, easy to learn and use, large standard library, third party libraries, platform independent, automation and scripting. In this case with its easy to learn syntax, network administrators and engineers can quickly develop custom network management tools and automate routing tasks, thereby improving efficiency and productivity.

There are mainly three kinds of simple network tools, hosts scanner, location finder and troubleshoot tool, and there is another tool that is network device configuration tool, it is a further development.

6.3. **Third party components and libraries.**

In addition to the core Python libraries, this network management tools development project uses several external libraries and modules to enhance its functionality. This project uses PyQt5, fpdf, pyrebase, subprocess, webbrowser libraries.

The PyQt5 library is used for designing the user interfaces of the tools, providing a modern and intuitive interface that is both easy to use and visually appealing. PyQt5 is a popular toolkit for building cross-platform graphical user interfaces and allows the development of desktop applications with a modern and responsive interface.

The Pyrebase is used for authentication of user details and for accessing the Firebase storage, providing secure storage of user data and making it easily accessible from any device. This enables the tool to store user preferences, map locations and other important data and information securely and access it from anywhere, providing a seamless user experience.

The Fpdf library is used to create PDF files that capture the troubleshooting steps generated by the tool. This feature allows network administrators to easily and share the troubleshooting steps taken, providing a useful record of the problem and its resolution.

The subprocess library is used to execute CMD commands on the system, enabling the tool to run various network commands and retrieve information about the network and its devices.

7. Discussion.

The increasing complexity and size of modern networks have made network management challenging, and manual management tasks can be time consuming and error-prone. Additionally, existing network management systems and tools are for large size of network infrastructures, they are not suitable for simple companies who has simple network infrastructures and making them less efficient and effective. Therefore, the need for smart network management systems.

The primary objective of this project is to develop a smart network management system that leverages Python programming language to automate and simplify network management tasks. The system will have the following features such as, hosts monitoring, physical location gathering, basic troubleshooting, network device configuration (furthermore development).

The facts gathering techniques used in this project include research on existing network management systems, analysis of network management requirements, and surveys with network engineers to gather insights on their network management challenges.

Existing network management systems were analyzed to identify their strengths and weaknesses. Most of the existing systems are not open source, and not suitable for simple networks and simple companies.

All the functional and non-functional requirements are fulfilled with proper tools and programming languages. And also hardware/ software requirements are briefly explained above. This network management system is operationally feasible and technically feasible as it uses reality available hardware and software. Additionally, Python programming language provides the necessary tools and libraries to develop the system.

And all diagrams have created to represent different components and functionalities of the system such as, use case diagram, class diagram, high-level architectural diagram.

The waterfall methodology that I was used to do this project. And used libraries such as, PyQt5, subprocess, Pyrebase etc.

When am developing this project I was faced several errors. My pip installation didn't execute early, PyQt5 design toolkit didn't installation errors, Pyrebase installation errors, Firebase configuration errors, code errors.

8. References

- ManageEngine, communications@manageengine.com (no date) *Switch Port & IP Address Management software by ManageEngine OpUtils, ManageEngine OpUtils*. Available at: <https://www.manageengine.com/products/oputils/> (Accessed: February 28, 2023).
- *Network Automation Solutions* (2022) *Network to Code*. Available at: <https://www.networktocode.com/> (Accessed: February 28, 2023).
- Networktocode (no date) *Networktocode/netutils: Python Library that is a collection of functions and objects for common network automation tasks.*, *GitHub*. Available at: <https://github.com/networktocode/netutils> (Accessed: February 28, 2023).
- (2021) *YouTube*. Available at: <https://youtu.be/DCaH4bQ4DxA> (Accessed: February 28, 2023).
- (2020) *YouTube*. YouTube. Available at: https://www.youtube.com/watch?v=tlhFIAymKnQ&list=PL3JVwFmb_BnRpvOeIh_To4YSiebiggyXS (Accessed: February 28, 2023).
- (2020) *YouTube*. Available at: <https://youtu.be/sHShBZIR9JA> (Accessed: February 28, 2023).