International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12– No. 43, March 2024 – www.ijais.org

Remote Performance Monitoring System for Managed Service Providers

Adam Ball
Graduate Apprenticeship,
School of Science and Engineering,
University of Dundee, UK

Laud Charles Ochei
Graduate Apprenticeship,
School of Science and Engineering,
University of Dundee, UK

ABSTRACT

Remote Performance Monitoring Systems (RPMS) are a critical method and resource for Managed Service Providers and IT departments in businesses. These systems, however, are either non-existent or out of date in many small and medium-sized businesses. However, these systems are either non-existent or are out of date in small and medium-sized businesses. They are also difficult to read with many unresolved bugs. In addition, similar solutions on the market offer limited functionality and do not meet the stakeholder's requirements. The aim of this paper is to design a remote performance monitoring system for providing performance bandwidth history reports for small and medium-sized businesses. This paper also provides a review of related work and similar solutions used for remote performance monitoring. The system is developed with a DevExtreme design using a Golang API that reads from an InfluxDB database. The new system provides the user with a modern, readable user interface as well as a modern, readable chart with the metrics selected by the user. The new system is recommended for small and medium-sized businesses since it enables users to extend its functionality and customise several modules (e.g., dynamically altering the order and size of graphs displayed).

Keywords

Remote, Performance, Remote Performance Monitoring System, Managed Service Providers

1. INTRODUCTION

Remote Performance Monitoring Systems (RPMS) stand as indispensable tools for Managed Service Providers (MSPs) and IT departments in small and medium-sized businesses. However, a prevalent challenge persists in the form of either outdated or non-existent RPMS in many small and medium-sized businesses. Existing solutions often suffer from readability issues and unresolved bugs, while their limited functionalities fail to align with stakeholder requirements.

The absence of effective RPMS in small and medium-sized businesses underscores a pressing need for a tailored solution. Currently, available systems not only lack readability but also fail to provide comprehensive functionalities.

Motivated by this problem, this paper aims to develop and integrate a novel performance and bandwidth history reporting system into remote performance monitoring systems for managed service providers and IT departments in small and medium-sized businesses.

The main contributions of the paper are:

1. Presenting a review of related work and similar solutions for performance monitoring of remote computers and networks in the MSP industry.
2. Designing and implementing a new performance and bandwidth history reporting system for managed service providers and IT departments in small and medium-sized businesses.
3. Presenting a range of testing strategies specifically tailored for the evaluation of remote performance monitoring systems, particularly when end-user participation is limited due to the non-client-facing nature of the system.
4. Providing recommendations and guidelines for how a user would use the system to monitor and historically view the performance and bandwidth data

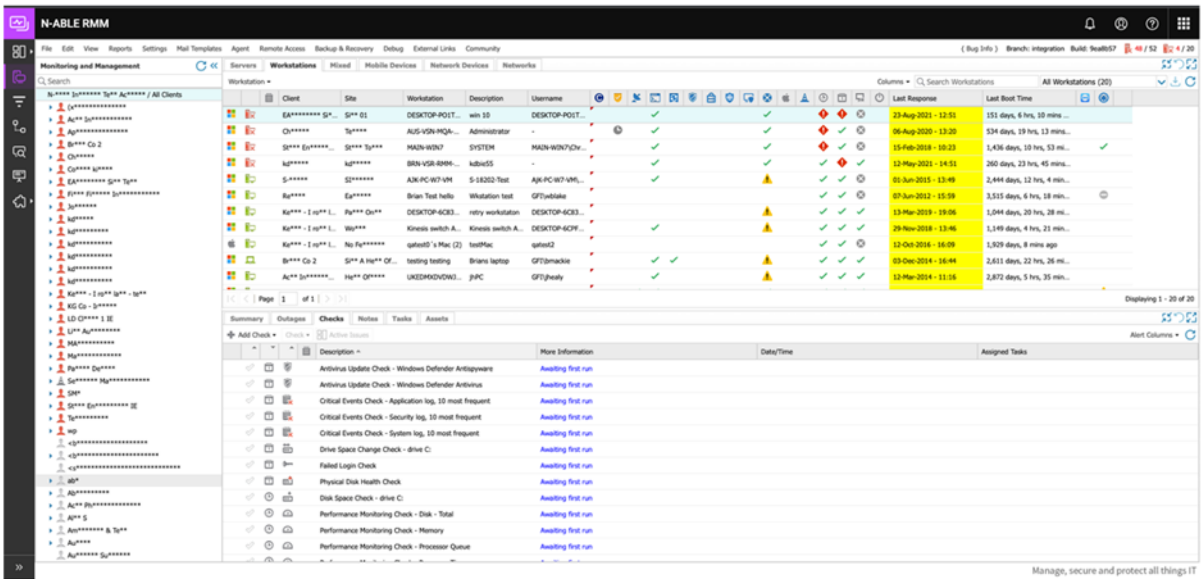
This research bridges this gap by presenting a modern, user-friendly RPMS. This system, developed using a DevExtreme design and a Golang API reading from an InfluxDB database, not only offers a contemporary user interface but also introduces a customizable reporting feature. This adaptability empowers users to extend the system's functionality and customize various modules, providing dynamic control over the display of graphs.

The rest of the chapter is organised as follows: Section two is the background of the study. Section three is the literature review and related concepts. Section four is the methodology. Section five is the analysis and design of the study while section six is the implementation and testing of the system. Section seven is the result and discussion. Section eight concludes the study with future work.

2. BACKGROUND AND STATEMENT OF THE PROBLEM

XYZ Ltd is an IT service management company, whose goal is to empower Managed Service Providers (MSPs) to help small and medium companies manage their IT infrastructure. The MSPs offer the service to manage these systems to companies that may not have the ability to do so themselves.

The XYZ company is responsible for offering the software to support this service. One of the most important components of the software by XYZ Ltd is the N-sight RMM which is the remote monitoring and management system. This system allows MSPs to manage devices and servers, remote access, and automate billing. Figure 1 shows N-sight RMM.


N-ABLE RMM
File Edit View Reports Settings Mail Templates Agent Remote Access Backup & Recovery Debug External Links Community
(Bug Info) Branch: integration Build: 9ea8b57 48 / 52 4 / 20
Monitoring and Management
Search
N-**** In****** Te** Ac***** / All Clients
Servers Workstations Mixed Mobile Devices Network Devices Networks
Workstation
Columns Search Workstations All Workstations (20)
Summary Outages Checks Notes Tasks Assets
Add Check Check Active Issues
Alert Columns
Page 1 of 1
Displaying 1 - 20 of 20
Manage, secure and protect all things IT

Client	Site	Workstation	Description	Username	Last Response	Last Boot Time
EA******** Si*...	Si** 01	DESKTOP-PO1T...	win 10	DESKTOP-PO1T...	23-Aug-2021 - 12:51	151 days, 6 hrs, 10 mins ...
Ch*****	Te****	AUS-VSN-MQA-...	Administrator	-	06-Aug-2020 - 13:20	534 days, 19 hrs, 13 mins...
St*** En*****...	St*** Ty***	MAIN-WIN7	SYSTEM	MAIN-WIN7\Ch...	15-Feb-2018 - 10:23	1,436 days, 10 hrs, 53 mi...
kd*****	kd*****	BRN-VSR-RMM-...	kdbie55	-	12-May-2021 - 14:51	260 days, 23 hrs, 45 mins...
S-*****	SI******	AJK-PC-W7-VM	S-18202-Test	AJK-PC-W7-VM\...	01-Jun-2015 - 13:49	2,444 days, 12 hrs, 4 min...
Re****	Ea*****	Brian Test hello	Wkstation test	GFI\wblake	07-Jun-2012 - 15:59	3,515 days, 6 hrs, 18 min...
Ke*** - I ro** I...	Pa*** On**	DESKTOP-6C83...	retry workstaton	DESKTOP-6C83...	13-Mar-2019 - 19:06	1,044 days, 20 hrs, 28 mi...
Ke*** - I ro** I...	Wo***	Kinesis switch A...	Kinesis switch A...	DESKTOP-6CPF...	29-Nov-2018 - 13:46	1,149 days, 4 hrs, 21 min...
Ke*** - I ro** I...	No Fe******	qatest0´s Mac (2)	testMac	qatest2	12-Oct-2016 - 16:09	1,929 days, 8 mins ago
Br*** Co 2	Si** A He** Of...	testing testing	Brians laptop	GFI\bmackie	03-Dec-2014 - 16:44	2,611 days, 22 hrs, 26 mi...
Ac** In******...	He** Of*****	UKEDMXDVDW3...	jhPC	GFI\jhealy	12-Mar-2014 - 11:16	2,872 days, 5 hrs, 35 min...

Description	More Information	Date/Time	Assigned Tasks
Antivirus Update Check - Windows Defender Antispyware	Awaiting first run		
Antivirus Update Check - Windows Defender Antivirus	Awaiting first run		
Critical Events Check - Application log, 10 most frequent	Awaiting first run		
Critical Events Check - Security log, 10 most frequent	Awaiting first run		
Critical Events Check - System log, 10 most frequent	Awaiting first run		
Drive Space Change Check - drive C:	Awaiting first run		
Failed Login Check	Awaiting first run		
Physical Disk Health Check	Awaiting first run		
Disk Space Check - drive C:	Awaiting first run		
Performance Monitoring Check - Disk - Total	Awaiting first run		
Performance Monitoring Check - Memory	Awaiting first run		
Performance Monitoring Check - Processor Queue	Awaiting first run		

Figure 1. N-sight RMM

Company XZY Ltd has a long history of providing this industry-leading software to MSPs they have been ranked the number 1 platform for MSPs and are used by over 25,000 MSPs. With the world's increasing demand for IT infrastructure MSPs are having to offer more and more services to the end users and the pressure is on to compete for this business. Company XYZ Ltd is working to increase the standard of its current system as well as implementing new features and device support.

One of the key components of the remote service monitoring system is the RMM device dashboard which is the area of the product that the end users such as the support staff and engineers would use. This area of the application is built up of an RMM Wrapper built in angular.

Another important component is the RMM all-devices view which is imported into the application wrapper from its own RMM server repository. Within component, there are many different features that can be provided, one of which is the reporting software, more specifically the Performance and bandwidth history report. There are several different features that can be supplied inside this component, one of which is the reporting software, particularly the Performance and bandwidth history report.

Currently, N-sight RMM offers the users a performance and bandwidth history report, which shows the results of the performance monitoring of the client's servers and workstations, or the bandwidth monitoring checks for the client's servers. This can be on individual devices over the last 24 hours and 8 days. The current charts for each check show both the average and maximum values to help easily identify when and where the performance tailbacks occur.

Currently, this report dialogue can be opened either via the report's menu or from the more information section for the checks. After the user selects the report, they can select what checks to include within the report for both the last 24 hours and or the last 8 days. These checks include the processor, memory, and network utilization. as well as the disk space check and bandwidth monitoring for the servers. Currently, this module shows serval static graphs that are not easy to read and cannot be adjusted. This has also been identified as a problemed area by our customers as they are continuing to discover many different bugs, such as incorrect data, loading issues and blank graphs being displayed. This is starting to become a bigger issue for N-able and the customers are getting progressively more frustrated.

3. REVIEW OF LITERATURE AND RELATED CONCEPTS

This presents a review of the literature and related concepts.

3.1 Overview of Related Concepts

A performance monitoring and reporting system is an application that collects the health status data and performance metrics of a system or device (SolarWinds, 2023). As described by SolarWinds these systems work "by continuously gathering data on system resources" (SolarWinds, 2023).

Network monitoring systems provide the information used by network administrators to gauge if a network is running optimally (Cisco, 2023). Margaret Rouse an award-winning technical writer talks about how these systems enable administrators, organizations, and end users to evaluate the performance of a system (Rouse, 2013)

When it comes to Network performance monitoring there are several different metrics used to access this such as bandwidth usage, latency, and throughput. Bandwidth usage is the maximum data transmission rate at a certain time. (DNSstuff, 2022)

Throughput is a metric that allows you to measure the data transmission rate through the network. Latency is the amount of time it takes to transfer data. (SolarWinds, 2023) Each application uses different metrics to monitor the network performance of certain devices. The metrics discussed all N-able to monitor network performance effectively (N-able, 2023).

3.2 Examples of Similar Solutions for Remote Performance Monitoring System (RPMS)

There are several examples of remote performance monitoring systems for managed service providers and IT departments for small and medium-sized businesses. These include NinjaOne, DataDog, SolarWinds and JAMF.

One of the significant differences between the N-sight RMM and other solutions is that it provides an exclusive offering of device support for Windows, Mac, and Linux. They allow users to manage and protect their devices without taking away from the Apple experience. (JAMF, 2022)

Another solution on the market Is SolarWinds NetFlow Traffic Analyzer, a tool that makes it easy to view bandwidth used by the application, protocol and IP address" (SolarWinds, 2023). The tool can identify bandwidth bottlenecks very quickly. It also offers the ability to set up alerts when bandwidth utilization drops and can automatically turn the usage data into charts, tables, and network bandwidth reports.

Another similar solution is DataDog, described as end-to-end visibility into on-prem, cloud, or hybrid networks (DataDog, 2023). This tool is more focused on network usage rather than network performance. This solution offers many different benefits such as an out-the-box dashboard and a feature that will autonomously find anomalies in the environment. DataDog also offers deeper visibility into the DNS performance as well as a feature to be able to analyse the network traffic between meaningful endpoints rather than just IP addresses. (DataDog, 2023)

JAMF, another similar solution is one of the most widely used Apple mobile device management software. Their goal is to help their customers empower their workforce by removing the hassles of managing technology. As of December 2022, they manage over 30 million devices. (JAMF, 2022) However, JAMF does not offer support for Windows devices and instead limits its target audience to Mac devices.

One of the main differences between the N-sight RMM and other similar solutions is in terms of compatibility. In addition to network monitoring functionality, the N-sight RMM tool provides the ability to add custom checks and patch critical updates.

The N-sight RMM does report on the performance of the device and does have the option to monitor the bandwidth performance at a client, site, or device level, and also support Windows devices. (JAMF, 2022).

3.3 Related Work on Remote Performance Monitoring Systems for MSP

The following section discusses recent research in remote performance monitoring systems. The areas that have been highlighted include - integration of AI and machine learning, cloud and IoT Integration, predictive analytics and capacity planning, security and compliance enhancements, customization and user experience, and benchmarking and comparative studies.

Incorporating artificial intelligence (AI) and machine learning (ML) techniques into RPMS has gained traction. These technologies augment anomaly detection, predictive analytics, and decision automation. Wang et al. (2022) have developed an AI-driven RPMS utilizing ML algorithms for continuous analysis of historical performance data, facilitating proactive issue resolution and system reliability.

The advent of cloud computing and the proliferation of the Internet of Things (IoT) have spurred research into seamless RPMS integration with these technologies. Huang et al. (2023) have pioneered an approach to integrate RPMS with cloud services, extending monitoring capabilities to cloud-based resources effectively. This integration affords a comprehensive view of the entire IT ecosystem.

RPMS research has honed predictive analytics and capacity planning capabilities. Chen et al. (2021) proposed a predictive analytics framework within RPMS to aid MSPs in data-driven decisions for capacity planning, optimizing resource allocation, and reducing costs.

Elevating security and compliance within RPMS has been a focal point. Kim et al. (2019) have delved into secure data handling practices within RPMS, introducing encryption and access control mechanisms to protect sensitive data, aligning with regulatory requirements.

Customization features have been augmented within RPMS to cater to diverse client needs. Gao et al. (2017) have introduced a customizable RPMS, allowing MSPs to create personalized dashboards, reports, and alerting rules to enhance the user experience.

Benchmarking and comparative studies have been instrumental in evaluating RPMS performance and reliability. Savvopoulos et al. (2019) conducted a comparative study of alerting mechanisms within RPMS, providing insights to aid MSPs in selecting the most suitable solution.

4. METHODOLOGY

The research methodology adopted for this study is the Build methodology. This resolves around creating a model such as a software application. Easterbrook describes this methodology as the combination of practical implementation and theoretical knowledge. (Easterbrook S, et al., 2008). The Build methodology mostly focuses on the practical aspect of the implementation. However, it does still require conducting an extensive literature review, analysis of current solutions and justify their decision with evidence. This is built around the guidelines and examples of conducting case study research within the software engineering industry (Runeson, et al., 2012).

The software development methodology used is the agile methodology which involves breaking down the project into different phases and focuses on collaboration and improvement. (Atlassian, 2023) This methodology promotes practices such as daily catch-up meetings (known as stand-ups), planning sessions, sprints, and pair programming (Alliance, 2023).

The project chose the Scrum framework, an incremental and iterative software development framework that manages project development (Sachdeva, 2016).

5. ANALYSIS AND DESIGN

The section presents the analysis and design of the study.

5.1 Requirements gathering

For this project, created a Jira Board, this will be the single source of truth for the work and requirements. This will show the breakdown of all the tasks including what is known as epics. An epic is a large body of work broken down into several smaller stories Fields (Atlassian, 2022). These stories will then be displayed in an ordered list within the TODO column on the Jira Board. This board will contain several different columns such as TODO, in progress, Ready for Review, In Review, RFT, Testing and Done. This board will show the progress of each task, how long it has been at that stage and who is working on it. In this project, the primary developer responsibilities will be carried out by a designated individual who will also assist the Test Engineers employed by N-able. Figure 2 shows the project Jira board.

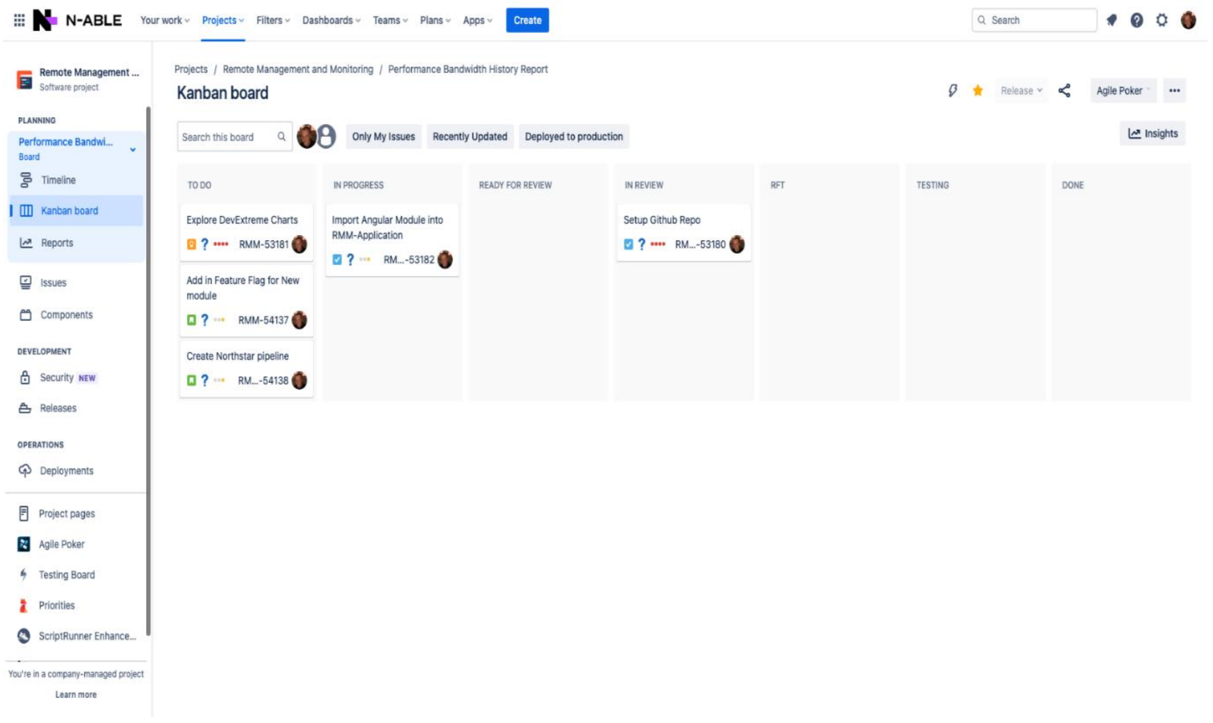


Figure 2. JIRA Board for the project.

Several functional and non-functional requirements were also gathered for this project. This is summarised in the table below.

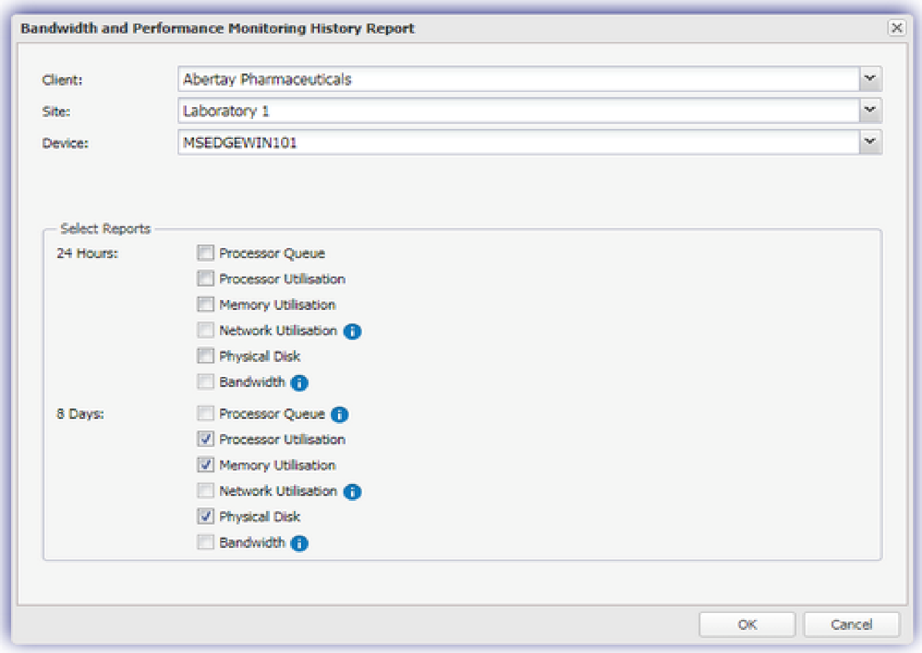
Table 1. Summary of functional and non-functional requirements of the system

Functional Requirements	Non-Functional Requirements
The user must be able to select the required metrics	The user interface must have a modern and consistent look
Graphs must load in under five seconds	The user must be able to switch between light and dark mode
The application must be module and built using the DevExtreme library	The application must be secure from unauthorized users
The Golang API must read from the InfluxDB database and present the information to the data source	The code must be readable with comments

5.2 User Interface Design

1. Viewing selected metrics over different time periods

Another feature of this application is the ability to view the selected metrics over different time periods. When the users open the application, they are presented with the option to select the client, site, or device. They can then select the report they want to view as well as the timeframe. Figure 3 shows the dialogue where the users will make this selection.



Bandwidth and Performance Monitoring History Report

Client:	Abertay Pharmaceuticals
Site:	Laboratory 1
Device:	MSEDGEWIN101

Select Reports

24 Hours:
- [ ] Processor Queue
- [ ] Processor Utilisation
- [ ] Memory Utilisation
- [ ] Network Utilisation
- [ ] Physical Disk
- [ ] Bandwidth

8 Days:
- [ ] Processor Queue
- [✓] Processor Utilisation
- [✓] Memory Utilisation
- [ ] Network Utilisation
- [✓] Physical Disk
- [ ] Bandwidth

Figure 3. Metrics Dialog for the project.

2. Displaying of reporting graphs

The design of the tools allows for splitting the graphs into separate graphs and then toggling between the timeframes. The user will have the option to tick whether to show the processor graph, memory graph, network graph or physical disk graph. Then within these graphs, the user can toggle the more detailed metrics such as the processor queue and processor utilization. Figure 4 shows the graph after selecting one of the metrics at the bottom while the others are hidden.

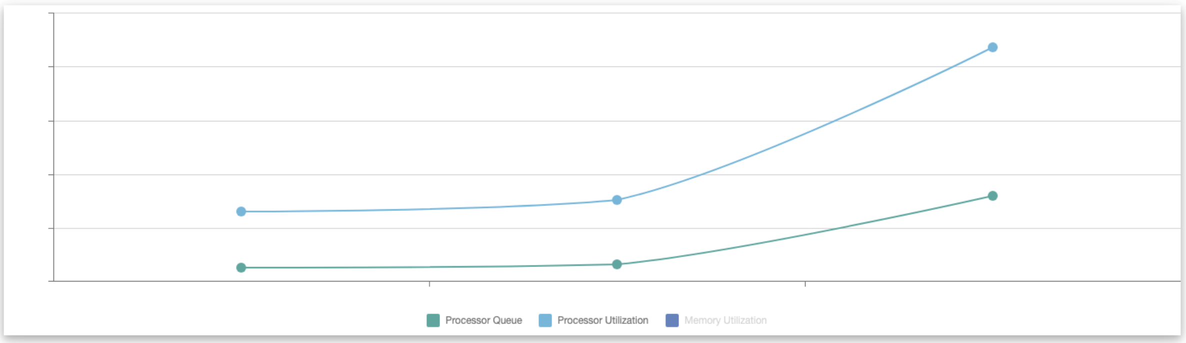


Figure 4. Graph displayed with One metric hidden.

5.3 Architecture Design

The Performance and bandwidth history report follows a Client-Server architecture built up of multiple components. Joseph Molloy describes Client-Server architecture as a “network where multiple clients request and receive files and services from a centralized server over a local or internet connection" he then moves on to discuss how typically the client will use an application as an interface to connect to the server (Molloy, 2023).

This form of software architecture offers many benefits such as increased scalability, flexibility, and a high level of performance. Vikram Joshi from Forbes discusses how this type of architecture means that it is easier to scale the application. He also talks about how this separation can offer faster deployments and optimization of resources (Joshi, 2018). These reasons were previously discussed and to keep this application consistent with others at N-able, it was decided that Client-Server architecture be used. Figure 5 shows the software’s architecture.

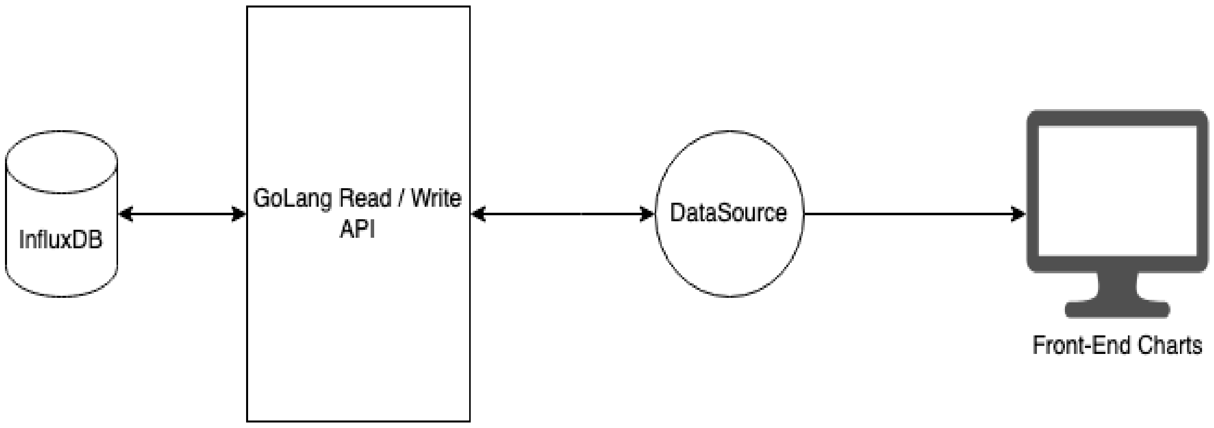


Figure 5. Client-Server architecture for the system

As shown in the diagram above the Backend of the application is built up of a read-and-write API written in Golang, an open-source, compiled, and statically typed programming language designed by Google. It is built to be simple, high-performing, readable, and efficient (Chris, 2023). In this project, they are responsible for pushing and pulling the required data to and from the Influx database. They are stored within the main directory within the CMD folder in a main.go file. The Angular Application is built up of several different components. The first is the main components, which show the features of the main page and then call the other components when requested such as the metric dialogue and each of the charts.

The application also contains a package.json file. This is responsible for all the dependencies and imports for the application, such as the Angular Dev kit and the nebular common repository that holds the styles for each of the Apex Charts.

5.4 Application structure

The application structure is split up into front-end and back-end folders (Figure 6). Within the CMD folder is the Golang read/write APIs. Following on from this within the source folder are the main Angular components, The app components hold the main application page, and the "performance-history-line-graph components" hold the graphs.

The software structure also contains a package.json file where the dependencies are imported as well as a docker-compose.yml file which contains the docker configurations.

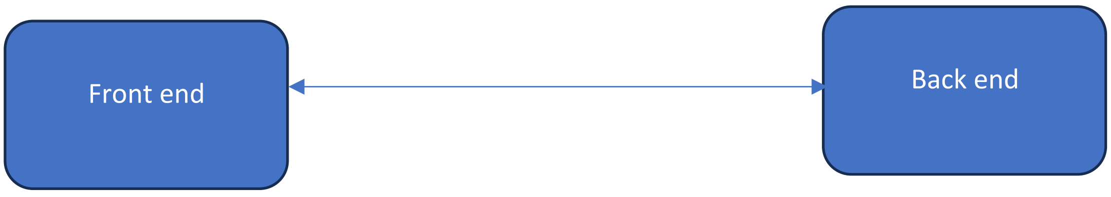


Figure 6. Application structure with front-end and back-end folders.

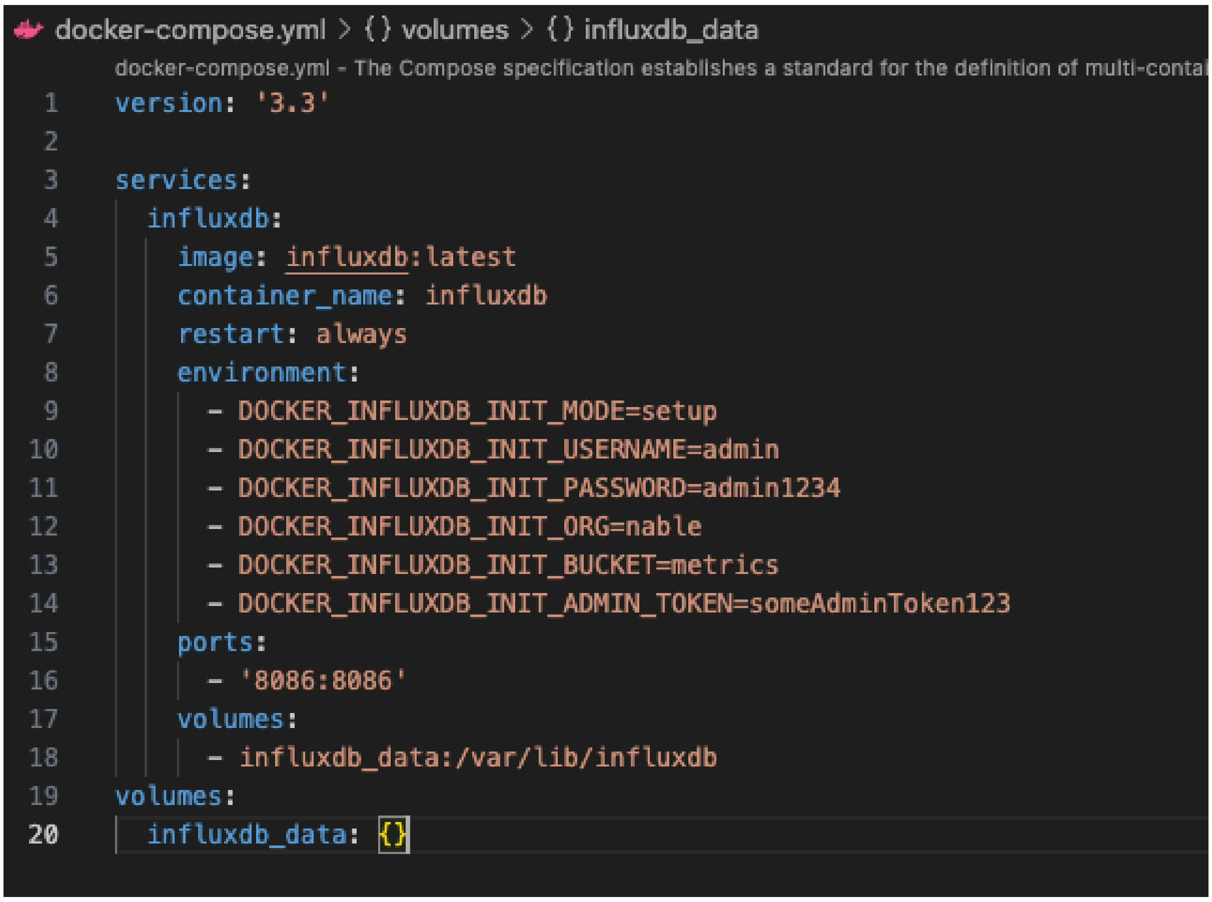
6. IMPLEMENTATION AND TESTING

6.1 System Implementation

This application is built up of a front-end Angular-based module built with the Apex Design system this will be the user interface that allows the user to view the performance and history data of their devices. It is built up of the Apex-styled DevExtreme graphs that read data from that data source.

In terms of the backend, the system has a read-and-write API that is written in Golang. The write API will push the data and store the performance and bandwidth history report metrics within the database. The Read API will be responsible for running the query to pull the requested data from the database and will store them within the data source that will be used to show the data within the Apex chart.

Another development tool used during the implementation of the application is Docker. Docker is an open platform for developing, shipping, and running various applications in a faster way. Docker enables the applications to run separately from the host infrastructure and treat the infrastructure like a managed application. Docker does this by combining a lightweight container virtualization platform with workflows and tooling that help to manage and deploy applications (Preeth et al, 2015). In this project, a docker-compose file was created within the main file directory. Via one command this will then build the container based on the configuration within the docker compose YAML. The Docker compose file created runs the InfluxDB service. The YAML files define the contains name, ports, and environment variables such as the database username, password, organisation, bucket name and also admin token. Figure 7 shows the configuration within the Docker compose YAML file.


docker-compose.yml > {} volumes > {} influxdb_data

docker-compose.yml - The Compose specification establishes a standard for the definition of multi-conta

```
version: '3.3'

services:
  influxdb:
    image: influxdb:latest
    container_name: influxdb
    restart: always
    environment:
      - DOCKER_INFLUXDB_INIT_MODE=setup
      - DOCKER_INFLUXDB_INIT_USERNAME=admin
      - DOCKER_INFLUXDB_INIT_PASSWORD=admin1234
      - DOCKER_INFLUXDB_INIT_ORG=nable
      - DOCKER_INFLUXDB_INIT_BUCKET=metrics
      - DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=someAdminToken123
    ports:
      - '8086:8086'
    volumes:
      - influxdb_data:/var/lib/influxdb
volumes:
  influxdb_data: {}
```

Figure 7 - Docker composes YAML configuration.

The database used for this application is the Influx DB. This is a database built specifically for time series data. It is written in Go and is responsible for storing and retrieving time series data (Influx, 2023). Influx DB also offers several different benefits such as a UI that allows the developers to edit scripts and view the collected metrics. This type of database also offers low-latency queries and acts as a single data source for all-time series data.

6.2 Testing

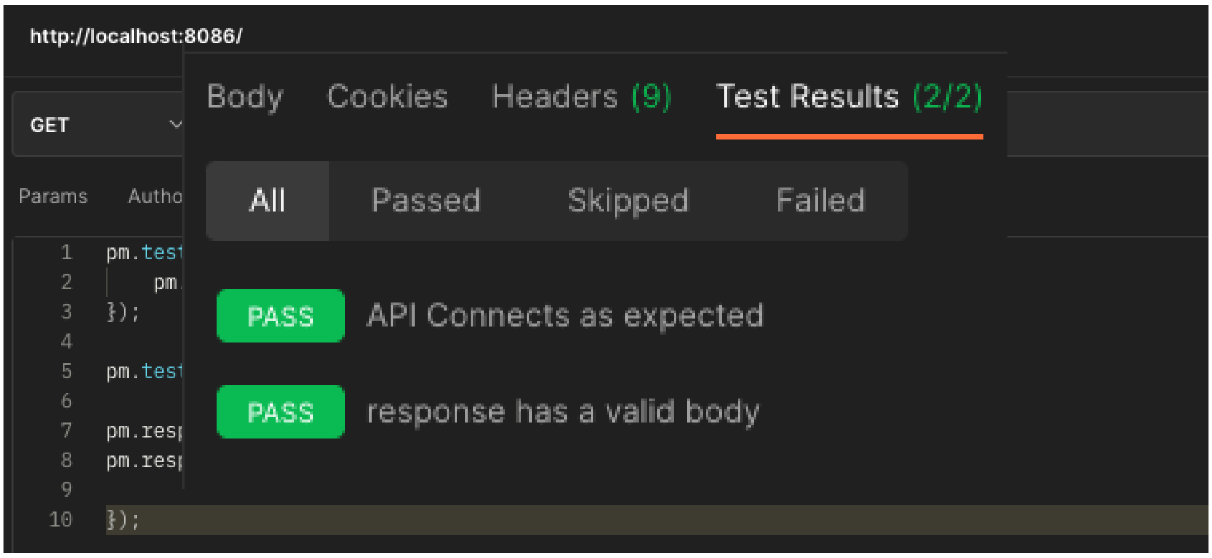
The application is not client-facing, and therefore end users were not involved in its testing. As a result, the application was tested using several software testing tools. The performance testing tool known as Apache JMeter was employed during the execution of the test. This tool provides a comprehensive feature set for assessing the behaviour of software applications under various load conditions. This is designed to load test functional behaviour and measure performance. (Apache, 2023) This testing tool will simulate multiple users sending requests to the system.

The Performance and Bandwidth reporting system is not client-facing, and therefore end users were not involved in its testing. As a result, the system was tested using different techniques and software testing tools. This testing includes API testing to ensure that the read and write APIs connect to the InfluxDB successfully. Data Validation testing was also performed to check that the data pulled via the API was accurate.

In addition to this, performance testing was carried out to ensure that the API can handle high-volume requests and that the response time is as expected (below five seconds). The user interface was tested to ensure that it displays the expected results correctly and that the application can scale to different screen sizes and orientations. As well as checking the application is compatible with different browsers. Several functional tests were carried out to ensure that the application meets the requirements.

6.3 API tests

One of the tests that were conducted ensured that the read and write APIs connected successfully to the Application Programming Interface (API). For this test, Postman was utilized to successfully test the API connections. Two tests were created to run against the local port for the API and database connection, testing that the API connected and that the API body was valid. See Figure 8 for the tests and Figure 9 for the test results.



```
pm.test
    pm.
});

pm.test

pm.resp
pm.resp

});
```

PASS API Connects as expected

PASS response has a valid body

Figure 8 - Postman Test Configuration

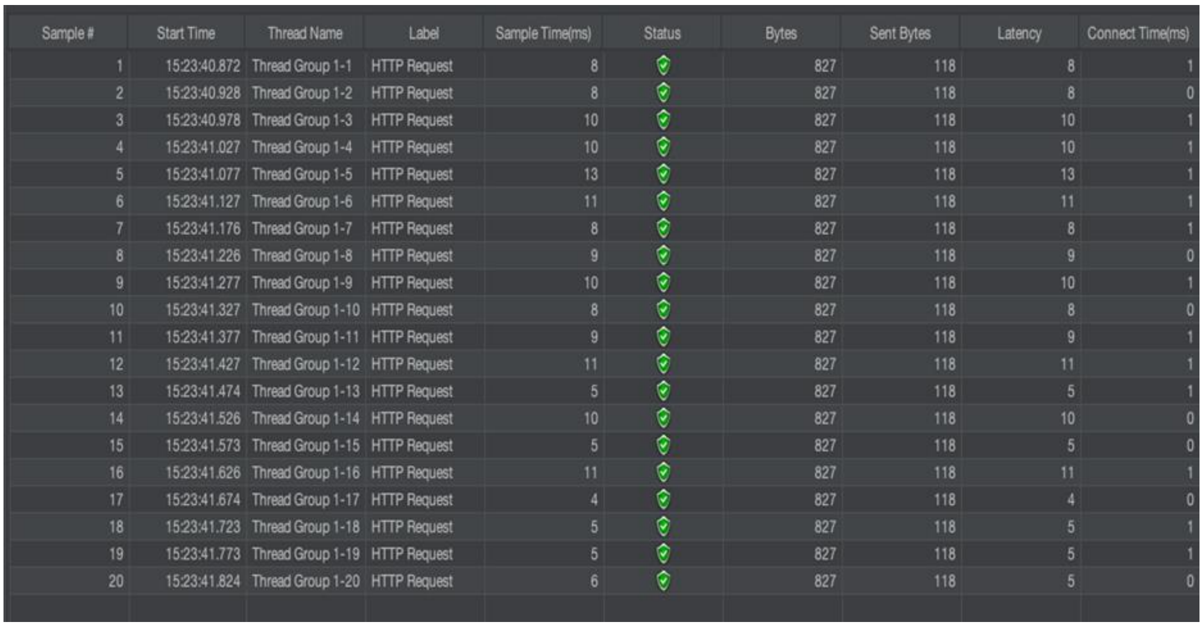
6.4 Data Validation

As part of the data validation testing process, tests were conducted using the write API written in Golang. These tests involved pushing data to the InfluxDB database, followed by using the read API to pull the same data from the database into the data source for the front end.

To verify the accuracy of the data being pulled, a data comparison tool was employed to compare the pushed and pulled data. This approach ensured that the data being displayed on the front end was an accurate representation of the data stored within the database.

6.5 Performance Testing

In the context of the testing plan, performance assessments were conducted to ascertain the ability of the application's APIs to manage a considerable volume of data input and output simultaneously. This evaluation was carried out utilizing JMeter as the testing tool. The objective was to gauge the performance of the system in handling requests from twenty virtual users and to measure response times under this load. Figure 9 shows the results of the JMeter Tests.



Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	15:23:40.872	Thread Group 1-1	HTTP Request	8	✓	827	118	8	1
2	15:23:40.928	Thread Group 1-2	HTTP Request	8	✓	827	118	8	0
3	15:23:40.978	Thread Group 1-3	HTTP Request	10	✓	827	118	10	1
4	15:23:41.027	Thread Group 1-4	HTTP Request	10	✓	827	118	10	1
5	15:23:41.077	Thread Group 1-5	HTTP Request	13	✓	827	118	13	1
6	15:23:41.127	Thread Group 1-6	HTTP Request	11	✓	827	118	11	1
7	15:23:41.176	Thread Group 1-7	HTTP Request	8	✓	827	118	8	1
8	15:23:41.226	Thread Group 1-8	HTTP Request	9	✓	827	118	9	0
9	15:23:41.277	Thread Group 1-9	HTTP Request	10	✓	827	118	10	1
10	15:23:41.327	Thread Group 1-10	HTTP Request	8	✓	827	118	8	0
11	15:23:41.377	Thread Group 1-11	HTTP Request	9	✓	827	118	9	1
12	15:23:41.427	Thread Group 1-12	HTTP Request	11	✓	827	118	11	1
13	15:23:41.474	Thread Group 1-13	HTTP Request	5	✓	827	118	5	1
14	15:23:41.526	Thread Group 1-14	HTTP Request	10	✓	827	118	10	0
15	15:23:41.573	Thread Group 1-15	HTTP Request	5	✓	827	118	5	0
16	15:23:41.626	Thread Group 1-16	HTTP Request	11	✓	827	118	11	1
17	15:23:41.674	Thread Group 1-17	HTTP Request	4	✓	827	118	4	0
18	15:23:41.723	Thread Group 1-18	HTTP Request	5	✓	827	118	5	1
19	15:23:41.773	Thread Group 1-19	HTTP Request	5	✓	827	118	5	1
20	15:23:41.824	Thread Group 1-20	HTTP Request	6	✓	827	118	5	0

Figure 9. JMeter Table Results

6.6 User Interface Testing

As previously agreed upon the testing strategy encompasses user interface checks. While carrying out the UI tests, several checks were selected from the list on the left-hand side and tested in different combinations of time frames. Additionally, it was verified that the toggles would display or conceal the chosen graphs.

(I) Displaying the correct result

To ascertain the accuracy of the information displayed on the user interface, the local dashboard of InfluxDB was accessed. Subsequently, a specialized tool known as Winmerge was employed to compare the data presented on the interface with the data stored in the database. This approach ensured that the displayed data accurately reflected the underlying database information.

(II) Scaling to different sizes and orientations

Another type of test carried out during the verification process of this application was functional testing. For this test, the size of the browser windows was changed across different screens and resolutions as well as changing the orientations from horizontal to landscape to ensure the application displayed as expected.

(III) Browser testing

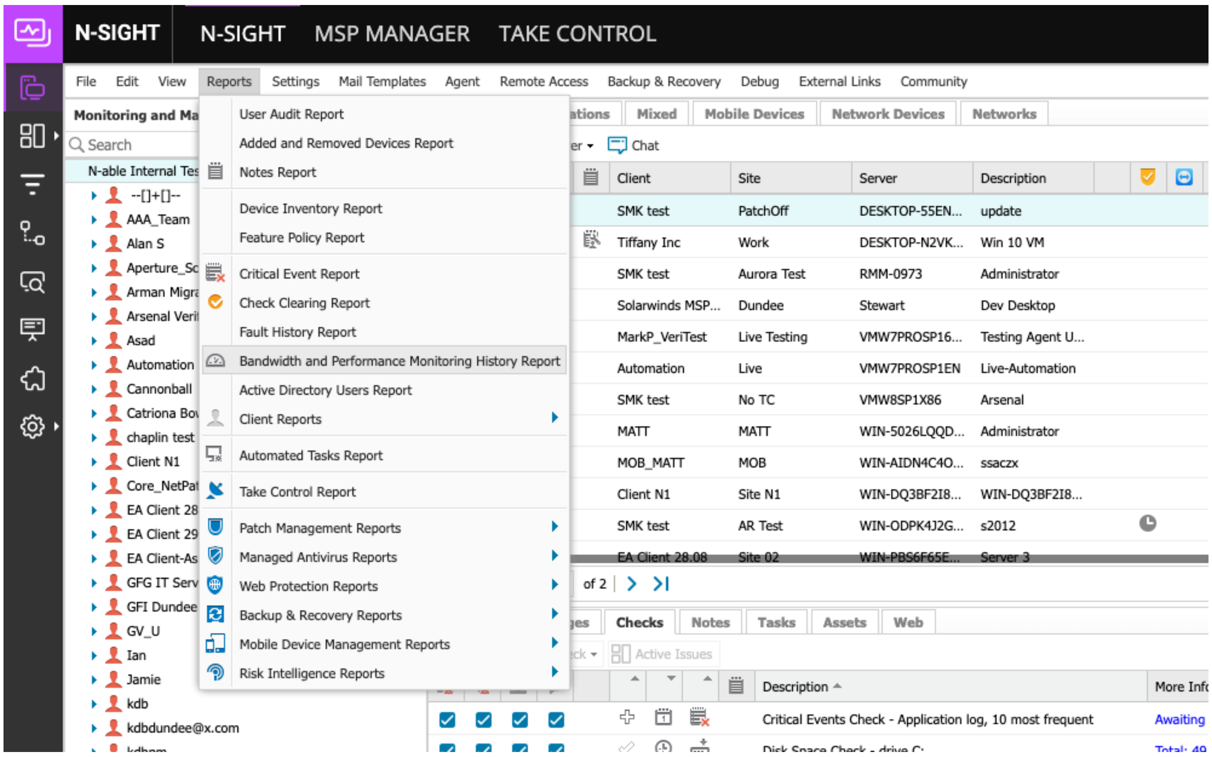
During the testing process, the application was accessed using various web browsers, including Google Chrome, Microsoft Edge, Firefox, and Safari. Subsequently, multiple tests were conducted to verify the functionality of the check selection feature. The goal of these tests was to ensure that users could successfully select the necessary checks and that the corresponding data would be displayed accurately. Additionally, the functionality of the toggle feature was evaluated to confirm that the checks would be hidden when unchecked.

7. RESULTS AND DISCUSSION

7.1 Summary of Results

A test plan was during the testing and verification stage of the application, which contains various test scenarios. The performance requirement specified that the graphs should load data within five seconds. During testing, the graphs were found to load significantly faster, taking less than one second. Additionally, JMeter testing revealed that the application could seamlessly handle multiple users connected to the database, demonstrating its scalability and stability under increased load. The API testing using Postman showed that the API connected successfully and returned a valid result. To ensure that the application met the requirements of the project, it was evaluated by comparing the requirements and features to the requirements list.

The performance and bandwidth reporting system will be deployed to the cloud for use on desktops and laptops via the browser. The application can be accessed by selecting the performance and bandwidth history report from the report menu as shown below (see Figure 10).



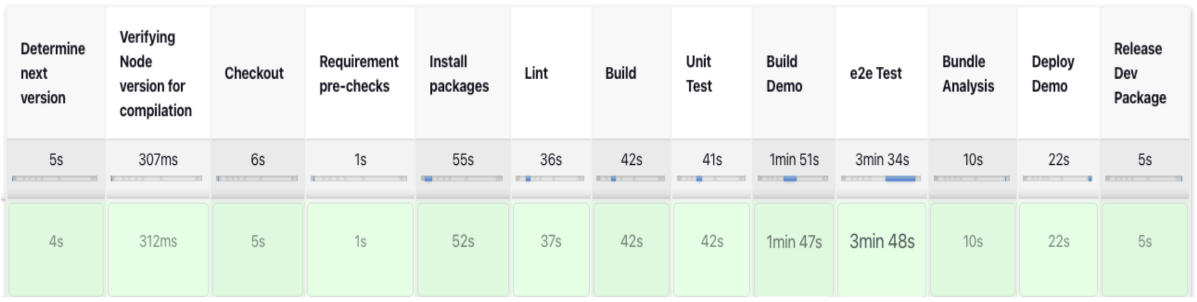
Client	Site	Server	Description
SMK test	PatchOff	DESKTOP-55EN...	update
Tiffany Inc	Work	DESKTOP-N2VK...	Win 10 VM
SMK test	Aurora Test	RMM-0973	Administrator
Solarwinds MSP...	Dundee	Stewart	Dev Desktop
MarkP_VeriTest	Live Testing	VMW7PROSP16...	Testing Agent U...
Automation	Live	VMW7PROSP1EN	Live-Automation
SMK test	No TC	VMW8SP1X86	Arsenal
MATT	MATT	WIN-5026LQQD...	Administrator
MOB_MATT	MOB	WIN-AIDN4C4O...	ssaczx
Client N1	Site N1	WIN-DQ3BF2I8...	WIN-DQ3BF2I8...
SMK test	AR Test	WIN-ODPK4J2G...	s2012
EA Client 28.08	Site 02	WIN-PBS6F65E...	Server 3

--[]+[]--
AAA_Team
Alan S
Aperture_Sc
Arman Migr
Arsenal Veri
Asad
Automation
Cannonball
Catriona Bo
chaplin test
Client N1
Core_NetPa
EA Client 28
EA Client 29
EA Client-As
GFG IT Serv
GFI Dundee
GV_U
Ian
Jamie
kdb
kdbdundee@x.com

Critical Events Check - Application log, 10 most frequent

Figure 10. How to access the performance and bandwidth history report.

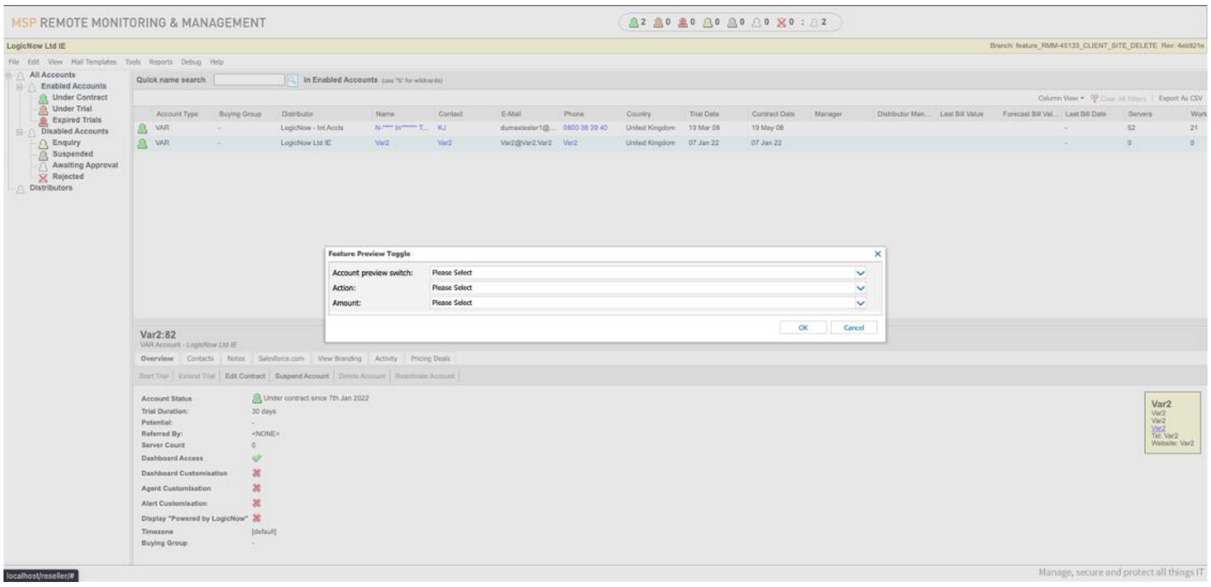
In undertaking this project, the decision was made to adhere to the principles of continuous integration and continuous delivery, thereby ensuring the efficient and seamless development and deployment of the software. The “developer's changes are validated by creating a build and running automated tests against the build”. (Pittet, 2023). Jenkins was used for the builds and deployment. The Jenkins file within the root of the application is detected when changes are pushed to the repository and then Jenkins will run the configuration from that file. This file is used to configure the build pipeline, Figure 11 shows the build pipelines and the steps it takes throughout the build process.



Determine next version	Verifying Node version for compilation	Checkout	Requirement pre-checks	Install packages	Lint	Build	Unit Test	Build Demo	e2e Test	Bundle Analysis	Deploy Demo	Release Dev Package
5s	307ms	6s	1s	55s	36s	42s	41s	1min 51s	3min 34s	10s	22s	5s
4s	312ms	5s	1s	52s	37s	42s	42s	1min 47s	3min 48s	10s	22s	5s

Figure 11. Build pipeline and stages.

An in-house feature flag service was used to manage the deployment of new features and releases. Brian Rinaldi describes a feature flag as a "concept that allows you to enable or disable a feature without modifying the source code or requiring a redeploy" (Rinaldi, 2020). Figure 12 shows the feature flag toggle used to control these feature flags.



MSP REMOTE MONITORING & MANAGEMENT

2 0 0 0 0 0 0 : 2

LogicNow Ltd IE

Branch: feature_RMM-45133_CLIENT_SITE_DELETE Rev: 4eb921e

File Edit View Mail Templates Tools Reports Debug Help

All Accounts
Enabled Accounts
Under Contract
Under Trial
Expired Trials
Disabled Accounts
Enquiry
Suspended
Awaiting Approval
Rejected
Distributors

Quick name search in Enabled Accounts (use '%' for wildcards)

Column View ▾ Clear All Filters | Export As CSV

	Account Type	Buying Group	Distributor	Name	Contact	E-Mail	Phone	Country	Trial Date	Contract Date	Manager	Distributor Man...	Last Bill Value	Forecast Bill Val...	Last Bill Date	Servers	Work
	VAR	-	LogicNow - Int Accts	N-**** In****** T...	KJ	dumaxtester1@...	0800 38 39 40	United Kingdom	19 Mar 08	19 May 08					-	52	21
	VAR	-	LogicNow Ltd IE	Var2	Var2	Var2@Var2.Var2	Var2	United Kingdom	07 Jan 22	07 Jan 22					-	0	0

Feature Preview Toggle

Account preview switch:	Please Select
Action:	Please Select
Amount:	Please Select

OK Cancel

Var2:82
VAR Account - LogicNow Ltd IE

Overview Contacts Notes Salesforce.com View Branding Activity Pricing Deals

Start Trial | Extend Trial | Edit Contract | Suspend Account | Delete Account | Reactivate Account

Account Status	Under contract since 7th Jan 2022
Trial Duration:	30 days
Potential:	-
Referred By:	<NONE>
Server Count	0
Dashboard Access	✓
Dashboard Customisation	✗
Agent Customisation	✗
Alert Customisation	✗
Display "Powered by LogicNow"	✗
Timezone	[default]
Buying Group	-

Var2
Var2
Var2
Var2
Tel: Var2
Website: Var2

localhost/reseller/#

Manage, secure and protect all things IT

Figure 12 - RMM feature preview toggle.

7.2 Product Evaluation

The application meets the requirement of allowing the user to select the checks to be displayed and toggle between viewing the eight-hour or twenty-four-hour timeframe. The API correctly reads the information from the database and outputs it to the user interface via a DevExtreme chart. The application as required can be accessed via the N-able sight application by logging in and selecting the performance and bandwidth history report. The application has been designed in a way that will allow it to be extended to further improve the graphs make it clearer and allow the user to customize the layout of each chart individually.

The source code has also been designed to make it easier to refactor, maintain and improve the code quality, fox ample, by renaming some of the components and classes to improve the ease of maintenance throughout the application. The source code of the application has been developed to facilitate easier refactoring and maintenance, with a focus on enhancing overall code quality. For example, various components and classes have been renamed to streamline maintenance tasks and improve the application's manageability.

7.3 Process Evaluation

The process of developing this application using agile methodology was appropriate. The stakeholders were kept informed and valuable support was gained from others in the company. Concerning the techniques used for gathering requirements, there is an opportunity for improvement. Employing different methods, such as user observation or surveys, may lead to better results. The testing of the application was conducted successfully. By utilizing tools such as JMeter and Postman, vital data was collected, including the maximum load of the application and its load time. The use of these tools also proved helpful in verifying that the API returned accurate data.

8. CONCLUSION AND FUTURE WORK

In this study, we have developed and integrated a novel performance and bandwidth history reporting system into remote performance monitoring systems for managed service providers and IT departments in small and medium-sized businesses. This research contributes to the literature on remote performance monitoring systems.

The study revealed a prevalent deficit of effective RPMS in small and medium-sized businesses (SMBs), characterized by either their absence or outdated and challenging-to-read interfaces fraught with unresolved bugs. Existing solutions in the market further compound this issue by offering limited functionality and falling short of stakeholder requirements.

The primary objective of this paper was to address this void by designing a novel RPMS tailored explicitly for SMBs, with a focus on providing comprehensive performance and bandwidth history reports. The developed system, utilizing a DevExtreme design and a Golang API interfacing with an InfluxDB database, presents users with a modern and readable interface. Notably, users can customize the system, dynamically altering the order and size of displayed graphs, enhancing its adaptability to specific user needs.

This research not only addresses the identified gaps in RPMS but also sets a precedent for user-centric, customizable solutions. It is anticipated that the outcomes of this study will not only benefit SMBs by offering an effective RPMS but will also contribute significantly to the broader field of remote performance monitoring, fostering innovation and improvement in industry practices.

9. REFERENCES

[1] Alliance, A. (2023). [Online] Available at: https://www.agilealliance.org/agile101/.

[2] Apache. (2023). Apache JMeter. [Online] Available at: https://jmeter.apache.org/. [Accessed 22 July 2023].

[3] Atlassian. (2022). What is an Epic? [Online] Available at: https://support.atlassian.com/jira-software-cloud/docs/what-is-an-epic/.

[4] Atlassian. (2023). Jira. [Online] Available at: https://www.atlassian.com/software/jira.

[5] Chen, W., et al. (2021). A Novel Remote Performance Monitoring System for Managed Service Providers. International Journal of Network Management, 31(2), e2139.

[6] Chris, K. (2023). What is GO? Golang Programming Language Explained. [Online] Available at: https://www.freecodecamp.org/news/what-is-go-programming-language/.

[7] Cisco. (2023). What is Network Monitoring. [Online] Available at: https://www.cisco.com/c/en_uk/solutions/automation/what-is-network-monitoring.html. [Accessed 22 July 2023].

[8] DataDog. (2023). Network Monitoring. [Online] Available at: https://www.datadoghq.com/dg/monitor/network/network-new/?utm_source=advertisement&utm_medium=search&utm_campaign=dg-google-network-emea-networkcapacity&utm_keyword=network%20bandwidth%20monitoring&utm_matchtype=p&utm_campaignid=15832880555&utm_adgroupid=.

[9] DNSstuff (2022). What Is Bandwidth Usage – How to Check, Measure, and Monitor it. [Online] Available at: https://www.dnsstuff.com/bandwidth-usage. [Accessed 18 July 2023].

[10] Easterbrook, S., Storey, J., & M.A, D. (2008). "Selecting Empirical Methods for Software Engineering Research." Guide to Advanced Empirical Software Engineering. Springer Science & Business Media.

[11] Gao, Y., et al. (2017). Customizable Remote Performance Monitoring System for MSPs. Journal of Cloud Computing: Advances, Systems and Applications, 6(1), 1-14.

[12] Huang, L., et al. (2023). Integrating RPMS with Cloud Services for Scalable Managed Services. Future Generation Computer Systems, 129, 112-125.

[13] Influx. (2023). Influx Data. [Online] Available at: https://www.influxdata.com/.

[14] Influxdata. (2023). Comparison to SQL. [Online] Available at: https://docs.influxdata.com/influxdb/v1.3/concepts/crosswalk/#.

[15] Joshi, V. (2018). Seven Reasons Why A Website's Front-End And Back-End Should Be Kept Separate. [Online] Available at: https://www.forbes.com/sites/forbestechcouncil/2018/07/19/seven-reasons-why-a-websites-front-end-and-back-end-should-be-kept-separate/?sh=2e27d1144fca [Accessed 12 July 2023].

[16] Kim, J., et al. (2019). Secure Data Handling in Remote Performance Monitoring for MSPs. Journal of Information Security and Applications, 47, 1-10.

[17] Molloy, J. (2023). A Comprehensive Overview of the Client-Server Model. [Online] Available at: https://www.liquidweb.com/blog/client-server-architecture/ [Accessed 12 July 2023].

[18] N-able. (2023). [Online] Available at: https://www.n-able.com/lp/it-solutions?utm_medium=cpc&utm_source=google-brand&utm_campaign=rm-glbl-lt-dgd-google_brand_core-2021-01-01&utm_term=n%20able_kwd-319841733353_e_575867562472&utm_content=g_15939739634_132568308419&cq_src=GOOGLE&cq_cmp=7170000008. [Online].

[19] Preeth, E. N., Mulerickal, F. J. P., Paul, B., & Sastri, Y. (2015, November). Evaluation of Docker containers based on hardware utilization. In 2015 international conference on control communication & computing India (ICCC) (pp. 697-700). IEEE.

[20] Savvopoulos, I., et al. (2019). Alerting Mechanisms in RPMS: A Comparative Study. Computers & Electrical Engineering, 76, 175-188.

[21] Wang, Y., et al. (2022). Leveraging Machine Learning for Anomaly Detection in Remote Performance Monitoring Systems. IEEE Transactions on Network and Service Management, 19(1), 454-466.