ISSN No: 2456-2165 https://doi.org/10.38124/ijisrt/25jul1797

IP-Based Computing Hardware Monitoring system for Laboratory Infrastructure Integrating

Muhammad Hamza Siddiqui[1]; Muhammad Talha Siddiqui[2]; Dr. Mehwish Mirza[3]

[1]Department of Computer Science, Hamdard University Karachi
[2]Department of Robotics Engineering University of Genova, Genova Italy
[3]Department of Intermech University of Modena, Modena, Italy

Publication Date: 2025/09/02

Abstract: Computer laboratory setup typically involves hardware architecture that requires continuous management within the context of academic institutions and enterprise settings in regard to performance, fault detection and peripheral validation. The proposed and designed computing-hardware monitoring system, which monitors computer networking hardware on a real-time, IP-based architecture, streamlines and automates the system-checks with a desktop client-server application. Its solution is based on the Electron with the JavaScript, MongoDB as backend storage, and local network (LAN) as the communication framework. The client application captures and relays system and hardware status to a central server and minimizes efforts in manual monitoring and works reliably, scalability and accuracy on the same. This paper describes the design, architecture, methodology and test cases that the system adopts and analyzes its applicability to any real-world laboratory settings.

Keywords: *IP-Monitoring,Hardware Specification, Client Server System,Electron,Node.Js,Mongodb.*

How to Cite: Muhammad Hamza Siddiqui; Muhammad Talha Siddiqui; Dr. Mehwish Mirza (2025) IP-Based Computing Hardware Monitoring system for Laboratory Infrastructure Integrating. *International Journal of Innovative Science and Research Technology*, 10(7), 3757-3762. https://doi.org/10.38124/ijisrt/25jul1797

I. INTRODUCTION

In universities, maintaining a large computer resource in several laboratories is a major operational problem. Lab attendants and IT administrators typically devote time to physically inspect all machines, verifying that hardware peripherals are properly installed, the system description satisfies expected standards, and there are no performance deficits. Not only this would be a time-wasting method, but human error and inefficiencies could also strike it. Currently, with the high pressure of educational and research institutions, there is a strong demand to find the effective solutions that will accomplish the tasks of maintaining buildings in an efficient manner, whereas diminishing the administrative load on the employees.

The IP-Based Computing Hardware Monitoring System is part of the solution to these issues and offers an automatic, centralized monitoring system. The system can track, in real time, the status of the system, associated peripherals, and hardware spec, and allows this to be done using the unique IP address of each workstation. Developed as a client server program, it enabled every computer to announce periodic updates to a central server which collated and presented the content. The system reduces the number of manipulations with hands, enhances the quality of monitoring and allows to fix hardware anomalies quickly.

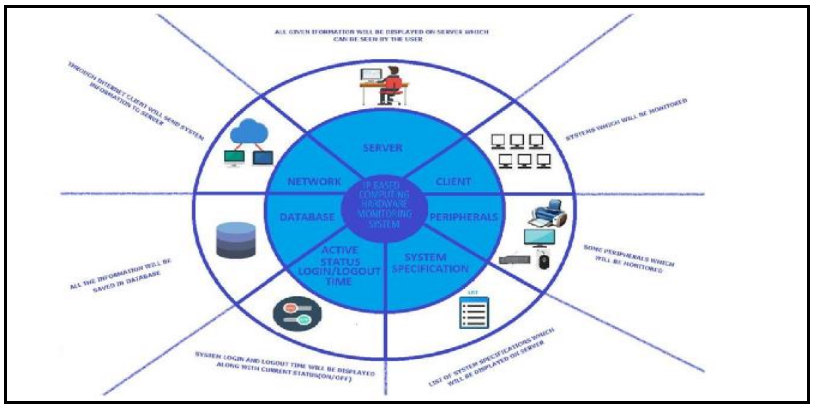

Fig 1 Big Picture

ISSN No: 2456-2165 https://doi.org/10.38124/ijisrt/25jul1797

This paper introduces the design, implementation, and evaluation of the monitoring system. We also explain its strengths, weakness, and areas that can be improved stating how this solution can revolutionize laboratory management practices in academic setting.

II. LITERATURE REVIEW

The need to monitor hardware in a more efficient manner has been fueled by the growing complexity of the computing environments. Current research and purchased applications have offered diverse solutions, including cloud-based services, monitoring opportunities based on Internet of Things (IoT). This segment surveys key literature responsive to building an IP-based monitoring system.

➢ *Hardware Monitoring in the Cloud*

Zhang et al. (2016) developed a cloud-based monitoring system where the data acquisition layer is separated and the data analysis layer, in which it is feasible to process large-scale distributed computing systems. Their contribution managed to feature the system reliability and maintenance that centralized data storage and processing enhance. Nonetheless, the use of cloud resources creates latency implications, bandwidth deployment, and privacy issues. Though suitable in enterprise settings, these solutions can be considered too much for smaller systems, like the university, where localized approach of monitoring can be more feasible.

➢ *Smart Monitoring Using IoT*

Ali et al. (2013) researched smart home control systems with the embedded microcontrollers and android based client. Their proposal exhibited lightweight, cost-effective monitoring using mobile-based control and observation. The case highlights advantages of flexible frameworks with friendly user systems. It is based on this that our system uses the same concepts of simplicity and economy of cost but used in a laboratory setting, involving real-time broadcasting of data between clients and servers.

➢ *Embedded Systems Runtime Verification*

In reporting correctness and reliability in embedded systems, Ernst et al. highlighted the relevancy of runtime verification. They dealt with the issue of surveillance without hindrance to the functioning of the systems. The application of these principles to the networks of institutions implies that non-intrusive monitoring tools that can operate round the clock without overloading the system are required. Our system is related to these discoveries because we make sure that the client application is running quietly in the background, sending the updates without hampering the activities of the user.

➢ *Comparison of Commercial Tools*

A category of commercial network monitoring software includes PRTG, ManageEngine, and Spiceworks that all provide comprehensive functionality, displaying performance indicators, alert mechanism, and rich reports. They however come along with expensive licenses, specialized infrastructure, and extensive configuration, which renders them unfriendly to academic institutions and small to medium enterprises (SMEs) with small budgets. Conversely, the proposed IP-based system avoided these shortcomings by providing a scalable and open-source infrastructure that takes into consideration the core monitor capabilities at a low cost.

The current literature highlights the importance of the effective and economical monitoring solutions. Our system design is based on the analyzed studies and tools, which are centralized control, low-resource characteristics, and real-time features designed with the academic context in mind.

III. METHODOLOGY

IP-Based Computing Hardware Monitoring System was developed in a highly organized manor to make the project clearly understood, accurate, and efficient as well as during the project lifecycle. Waterfall Software Development Model was chosen due to the fact that it is easy to understand the sequence of steps and any phase is to be finished before another one starts. This option was the most applicable to the project since the nature of the system needs was well-known at the very beginning of the work, and there was the least chance of reworking.

➢ *Waterfall Model: Lifecycle in Software Development*

- *This Methodology Consisted of Five Clustered Stages:*

Requirement Gathering Requirement gathering to comprehend the relevant status of monitoring requirements and establishing the system needs it was carried out through interviewing laboratory attendants as well as IT personnel. Rep.

Design- Under design stage, use-case models, process flow diagrams and system architecture was designed to outline the interactions among the clients, servers and the database. Implementation by programming the modules with JavaScript, the development team interfaced this with the Electron framework to support cross-platform functionality, and stored backend data in MongoDB.

Testing - Various test cases were also done to validate that the system is accurate, stable, and reliable under actual conditions. Deployment- The last system was implemented and evaluated in Hamdard university labs to examine the performance in the real world.

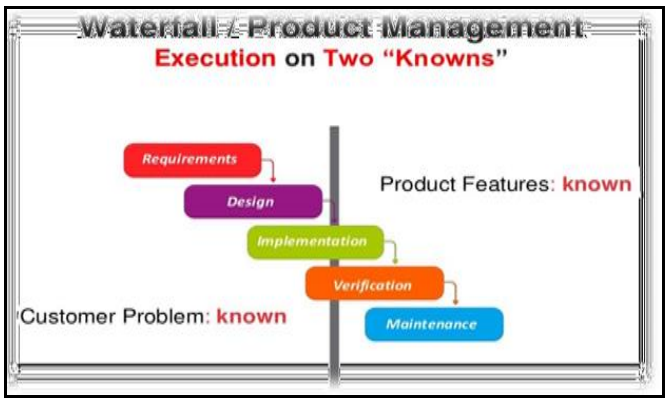

Fig 2 Waterfall Methodology Diagram

ISSN No: 2456-2165 https://doi.org/10.38124/ijisrt/25jul1797

➢ *Architecture Functional*

- *The Architecture is Divided into Three Layers:*

✓ Client Application: This program is installed on every lab PC and it retrieves the hardware information such as CPU, RAM, disk, and network details.
✓ Server Application: It runs on the system of the administrator and shows real-time data of clients in an interactive GUI.
✓ Database Layer (MongoDB): Maintains log data, system specification, and the login/logout times of the system to analyze and report.
✓ The communication is conducted within the Local Area Network (LAN) through Ethernet. Every client sends its data to the server every five seconds, guaranteeing correct and real entries.

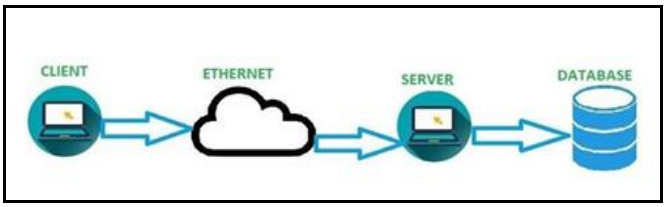

Fig 3 Architect & Design of IP Base Computing Hardware Monitoring System

IV. SYSTEM DESIGN

System design of IP-Based Computing Hardware Monitoring system enables the smooth flow of interactions between clients and a central server to provide the real-time monitoring of all the connected devices. The architecture integrates a myriad of parts that collaborate with each other to bring about efficiency, scalability, and reliability. Our workflow process starts with the data collection on the client machines, then being transmitted to the local network and lastly manipulated and stored in the server.

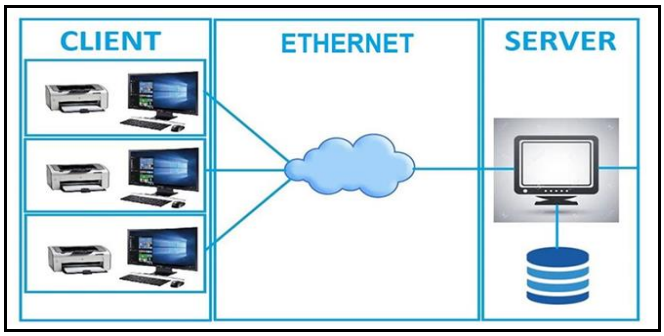

Fig 4 Process Flow Diagram

➢ *Modules and Components*
There are three major modules contained in the system:

- *Client Module:*

It is installed in every computer within the laboratory and gathers specific hardware detail about client machines including CPU, RAM, disk, network, and peripherals. It executes silently in the background during boot and automatically sends the gleaned information to the server.

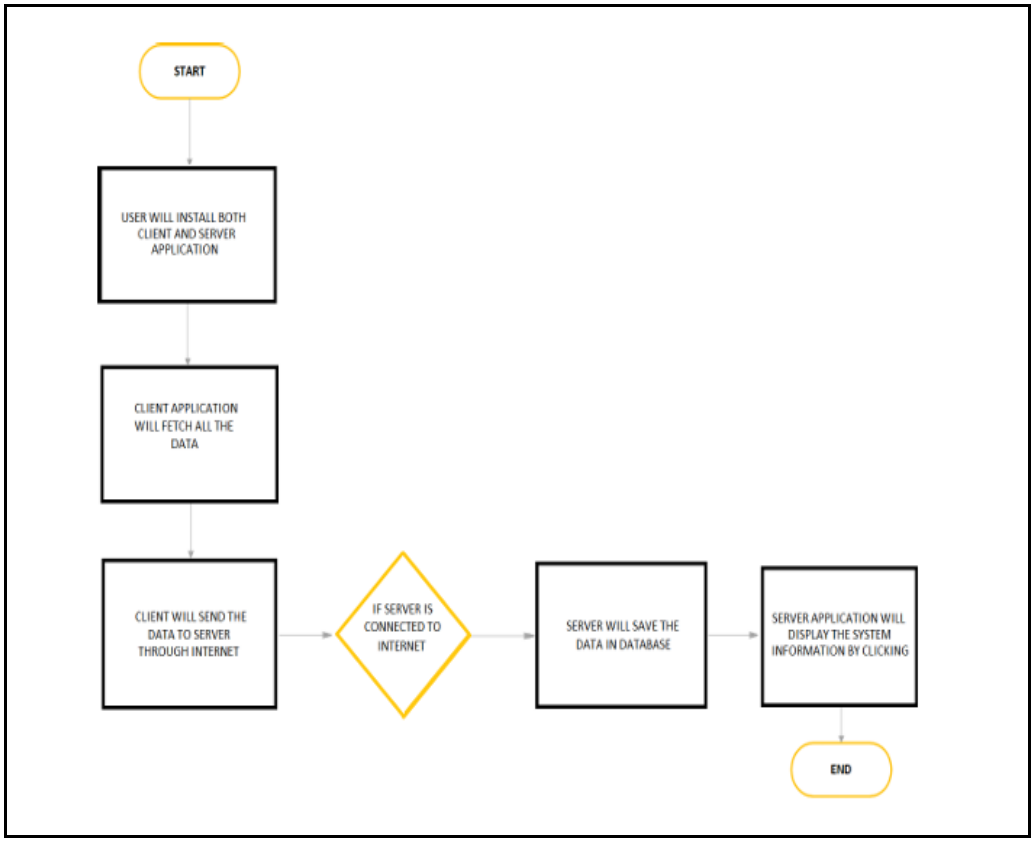

Fig 5 Flow Diagram for IP Base Computing Hardware Monitoring System

ISSN No: 2456-2165 https://doi.org/10.38124/ijisrt/25jul1797

- *Server Module:*

Receives and processes data sent by all connected clients. It offers a graphical user interface (GUI), where system administrators are able to observe the statuses of systems in real-time. It also facilitates export to Excel the data collected to enable additional reporting and analysis.

- *Backend Module:*

It is designed based on MongoDB and helps manage database operations, store logs and allows searching and retrieving the data efficiently.

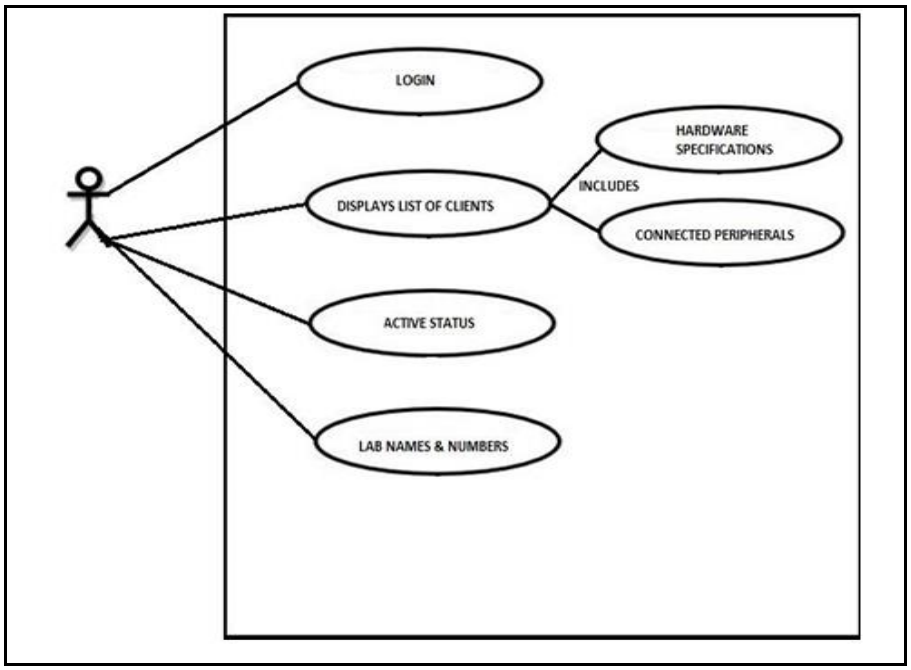

Fig 6 Use Case Diagram of IP Base Hardware Computing Monitoring System

➢ *Hardware Requirements*

It is also a lightweight system and needs only a few hardware: An Intel Pentium 4 or Core i3+ Processor, minimum 128MB of RAM, 100 MB of disk space, and an Ethernet-based LAN interface.

➢ *Software Stack*

The software environment would include desktop application packaging (Electron.js), backend logic (Node.js), structured data storage (MongoDB) and GUI design (Bootstrap and HTML/CSS). This pairing provides cross platform support and high performance as well as easy scalability.

V. IMPLEMENTATION

IP-Based Computing Hardware Monitoring System was developed as a desktop solution based on JavaScript and the Electron framework, which guarantees the cross-platform compatibility and the responsive user interface requirements. The backend was created using Node.js and connected to MongoDB, which we use as the database that can store real-time hardware metrics, such as CPU manufacturer, processing speed, the number of cores, memory capacity and usage, disk serial numbers and sizes, network IP addresses, and statuses of connected peripherals. This architecture facilitates effective communication between clients and the server as well as keeping the architecture scalable to many connected systems. Implementation is strictly based on modular.

➢ *Client Flow:*

The client application will automatically start when the system starts. It repeatedly sends the hardware specifications to the system information API and sends it to the server through HTTP POST requests at intervals to keep it up-to-date.

➢ *Server Flow:*

The server application listens to the broadcast of all client machines and displays the data in real time updating the dashboard as well. It includes watching functionalities about the active/inactive status of systems and has an export feature which can save the collected data in excel format so as to be reported and analyzed.

➢ *Testing and validation*

The testing phase involved testing the functionality of the system, reliability, and usability of the IP-Based Computing Hardware Monitoring System. They created nine pieces of structured test cases, which addressed key areas of the client and server modules. Such tests were installation processes and background operations, user interface verification and correctness of data exports. The conduct of the tests was done in a controlled laboratory setup in an attempt to resemble real life conditions. The results showed that the client application correctly broadcasted system data, and the server always showed accurate and real-time information. There was also validation of the Excel export feature, securing data integrity. Their test cases were all successful and they were very robust and stable.

ISSN No: 2456-2165 https://doi.org/10.38124/ijisrt/25jul1797

Table 1 Test Cases

Test #	Scenario	Result
TC#1	Installation of Client App	Pass
TC#2	Client runs in background	Pass
TC#3	Lab name field appears correctly	Pass
TC#4	Installation of Server App	Pass
TC#5	Server lists all systems	Pass
TC#6	Server displays correct client info	Pass
TC#7	Server shows real-time updates	Pass
TC#8	Export to Excel feature works	Pass
TC#9	Excel file contains accurate client data	Pass

VI. RESULTS AND DISCUSSION

The IP-Based Computing Hardware Monitoring System implementation resulted in great efficiencies on how the laboratories were managed. The drastic decrease of the monitoring time was one of the most evident outcomes. Lab attendants traditionally took hours to manually go through all the machines in six labs. The new system automatically does this, and the server is updated, every five seconds, by all the client machines. This real-time information keeps the administrators in a prime position to detect and correct hardware anomalies promptly, thus reducing the cost of downtimes and boosting the productivity of key operations.

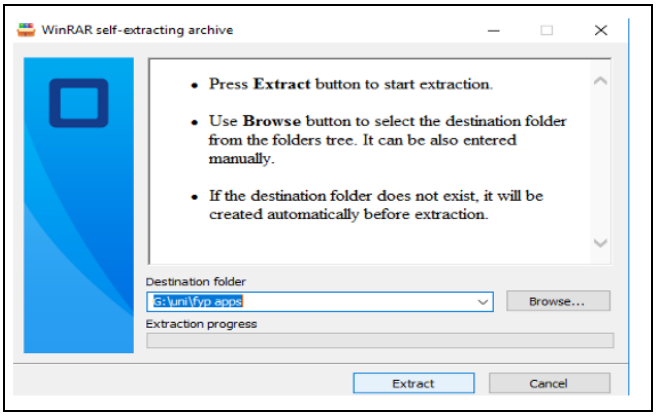

Fig 9 Installation of Client Application

Another key advantage was the centralization of data. System specifications and peripheral connectivity information were scattered before deployment, and could only be checked with each system separately. The client-server framework that was worked out combined this information and allowed administrators to get the access to all hardware information with the help of one convenient element and interface. This top-down perspective enhances not only surveillance but also choice making pertaining maintenance, upgrading and assigning resources.

Also, the system helped in better quality control. Caber log: It brings authenticity by recording specific hardware requirements and suitable peripherals, which may be easily checked in the event or maintenance schedule. Any form of deviations related to expected configurations are known immediately thus ensuring a parallel flavor in all systems.

In general, the system performed accurately when tested in real-world conditions and there were no remarkable network delays or processing. The overall client modules integration with the server was successful and proved its scalability making it applicable to other environments besides the university deployment as it was deployed.

VII. FUTURE WORK

The existing IP-based monitoring system has been an inspiration to many additions that can enhance its capabilities so much. A significant enhancement is protecting against operating system power-off with detection of power-off switch-ons via a DLL that restores the system through recording of full activity logs. Furthermore, a transition of the backend to Java Spring Boot would enable the introduction of multi-threading, which would result in a faster speed of processing the data and the next scale to allow operating in larger networks. To maximize accessibility to administrators, a special Android mobile app might be created, which would make it possible to control in real time even outside the office. The use of chatbot interface as another desirable quality is also expected which will allow immediate contact between clients and administrators to help in troubleshooting and monitoring of the system. Moreover, it is possible to extend the system in order to record not only the activity of hardware elements but also the software-level activity, providing an extensive picture of each machine status and utilization. Such developments would make the current system dynamically more flexible and smarter, and one able to support changing IT infrastructure demands in educational establishments and beyond.

VIII. CONCLUSION

This paper described the design and implementation, along with an evaluation of an IP-Based computing hardware monitoring system that was introduced to maximize the administration of computer laboratories within an academic facility. The idea behind the system was to overcome the drawbacks of manual monitoring that is time and error-consuming. It automates gathering and presenting hardware specifications, peripheral states, system-activity in a client-server architecture, thus alleviating manually intensive work while enhancing data integrity.

The proposed system has been made non-intrusive and affordable, thus appealing to institutions whose budgetary

ISSN No: 2456-2165 https://doi.org/10.38124/ijisrt/25jul1797

allocations are limited. High reliability, easy operation, and smooth performance in the live laboratory environment were observed during its deployment. Centralized general monitoring as well as real-time reporting has improved the efficiency of operations meaning that administrators are in a position to detect a problem and act upon it in time.

In addition to its direct effect, the system provides an infrastructure basis to longer-term trends. Possible improvements are the inclusion of mobile monitoring applications, the utilization of artificial intelligence to manage predictive maintenance and compatibility with a variety of operating environments. These enhancements allow the solution to become a full-scale platform of hardware lifecycle in educational and enterprise network systems with big networks.

In general, the IP-based monitoring system is effective, scalable, and sustainable solution to the laboratory infrastructure of the modern laboratory that can be used to open up the door to a smarter management of IT resources.

REFERENCES

[1]. “https://www.matecconferences.org/articles/matecconf/pdf/2016/07/matecconf_iceice2016_02080.pdf".
[2]. "https://www.researchgate.net/publication/257075607_Internet_of_Things_Ubiquitous_Home_Control_and_Monitoring_System_using_Android_based_Smart_Phone".
[3]. "https://pdfs.semanticscholar.org/5374/c5cf03310a790d19b0f5871ee6523a6b1d92.pdf".
[4]. "https://hlassets.paessler.com/common/files/pdf/system-requirements-en.pdf".
[5]. https://jumpcloud.com/?rs=Capterra&cs=ITmgmt&utm_medium=reviews&utm_source=capterra&utm_content=ITmgmt".
[6]. "https://www.manageengine.com/products/desktop central/?utm_medium=ppc&utm_campaign=DC&utm_source=capterra".
[7]. "https://www.device42.com/solutions/it management/?utm_medium=capterrappc&utm_source=capterra&utm_source=capterra".
[8]. "https://www.capterra.com/p/79191/Spiceworks-IT-Desktop/".
[9]. "https://www.capterra.com/p/70929/System-Center/".