IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, June 2025
Impact Factor: 7.67

Android Application for Real-Time PC and Server Control over Network

Ninad Bhagat[1], Pranali Jagadale[2], Sagar Bobade[3], Atul Bhagat[4], Sagar Patole[5]
Students, Computer Engineering Department[1-4]
Professor, Computer Engineering Department[5]
PES's COE, Phaltan, Maharashtra, India

Abstract: *Efficient remote control of computer systems has become an essential need in the current educational and organizational settings. This project reports a mobilebased system for remote monitoring and lab PC control via a Local Area Network (LAN). The system is designed and implemented using React Native for the front-end and Flask as the back-end framework. The system supports real-time PC status detection (ON/OFF), remote shutdown functionality. Every computer in the lab has a light-weight client running which is communicating with the master server, enabling control commands to be sent from the Android application and live updates received. This makes physical presence unnecessary, minimizes manual labour, and maximizes operating efficiency. The system is affordable, easy to scale, and best utilized in environments like colleges where it is frustrating and time-consuming to deal with multiple computers manually. Through its use of mobile and networking technologies, the project promotes a clever and adaptable style of managing the lab.*

Keywords: Remote Monitoring System, PC Power Control, Android Application, React Native, Flask Web Framework, Local Area Network (LAN), Real-time Device Management, Client-Server Architecture, Mobile-Based Network Control.

I. INTRODUCTION

As the reliance on computer networks for educational and institutional settings continues to rise, the demand for adaptable and effective system management has increased tremendously. In traditional settings, the management of computer labs necessitates manual monitoring and physical intervention for the day-to-day operations of turning systems off, checking for availability, or conducting maintenance. This is time-consuming, not very efficient, and not practical for large-scale setups.

In order to solve these drawbacks, this project introduces a remote monitoring and control system for lab PCs connected through a Local Area Network (LAN) using mobile technology. The solution utilizes the latest wireless communication and mobile software development technologies to avoid physical presence in system supervision and control.

The system includes two primary parts: a mobile app built with React Native, and a backend server built with Flask, an efficient Python web framework. Every PC in the lab executes a small client agent that dialogs with the central server. This system provides real-time status discovery (ON/OFF), and remote shutdown, facilities directly from the mobile app interface.

By combining these technologies, the system proposed guarantees quicker decision-making, greater flexibility, and better resource allocation. It is especially ideal for educational institutions where computer labs are heavily utilized and it becomes difficult to manage multiple systems manually. Additionally, the infrastructure is costeffective, scalable, and flexible for a vast variety of LANbased environments. This article addresses the system architecture, deployment, and possible applications, illustrating how mobile-led network management can revamp and automate lab administration functions.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, June 2025
Impact Factor: 7.67

II. LITERATURE REVIEW

The blending of remote device management and mobile technologies has been a focus of growing interest in recent years. A number of research projects have examined the adoption of systems whereby users can monitor and control devices using mobile interfaces and wireless communication.

In [1], a PC remote controller was created with Android phones via Bluetooth and Wi-Fi. Although efficient in localrange settings, the system had limited scalability and did not support real-time LAN-wide controls. Likewise, [2] suggested a home appliance control system based on Android and Wi-Fi modules but was limited to smart home scenarios and did not include the general PC control operation.

In [3], a remote power management device for institutional settings was presented through the web interface. While working, it needed a high-performance backend and lacked support for light mobile applications. In contrast, [4] investigated the utilization of Wake-on-LAN (WOL) to remotely activate computers, presenting the consistency of magic packets in an environment of a LANbeing the backbone of our system's wake-up functionality.

In addition, [5] introduced a network monitoring system that offered real-time visualization of the devices connected, but it was mainly for security and bandwidth statistics, not control commands like shutting down or rebooting.

All earlier solutions either focused on home automation, enterprise-focused network utilities, or needed to have centralized servers along with intricate infrastructure. Our system bridges the gap by providing a lightweight, mobile centric solution developed using React Native and Flask specifically targeting real-time monitoring and management of lab PCs over LAN. It utilizes remote shutdown capability, while maintaining platform simplicity, affordability, and scalability.

III. PROBLEM STATEMENT

In many educational institutions and organizations, managing a large number of computers manually is inefficient and resource-intensive. Lab administrators are often required to physically monitor the operational status of PCs, switch systems off, and troubleshoot network or device-level issues. This conventional method not only increases administrative workload but also introduces delays and limits flexibility, especially when quick action is needed or remote management is desirable. A more streamlined and automated approach is essential for effective lab management in modern, connected environments.

IV. METHODOLOGY

The software was created with a modular client-server architecture to provide scalabilityand integration convenience in current LAN environments. Development consisted of the following major components:

It works something like: (phone) ⇄□ (server) ⇄□□□ (client PCs)

Architecture

We divided the system into following parts:

1. Frontend Development

React Native was employed to develop the mobile app for cross-platform compatibility. It offers a neat and user-friendly interface for administrators to:

- Display the real-time ON/OFF status of PCs.
- Send shutdown commands.
- Get alerts and notifications dynamically by using HTTP API calls.

2. Backend Development

A server for Flask was used to receive HTTP requests from the mobile application and forward them to the corresponding PC clients.

The backend:

- Keeps a record of registered PCs and their MAC/IP addresses.
- Sends shutdown commands.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, June 2025
Impact Factor: 7.67

• Queries client agents for status.

3. Client Agent (on every PC)

There is a light Python script on each PC that:

• Listens for status queries from the server.

• Accepts shutdown commands.

• Keeps the server aware of its status (ON/OFF).

4. Network Communication

The whole setup runs on a LAN and shutdown/status requests via regular HTTP calls. CORS is supported for seamless mobile-backend communication.

5. Security and Access Control

Simple request validation and IP whitelisting are used to prevent unauthorized access.

UML Diagrams

We created UML diagrams to explain:

Activity Diagram: Describing the flow of the use case step by step, for example: user logs into the app → send command to server → server sends command to the PC → PC executes command.

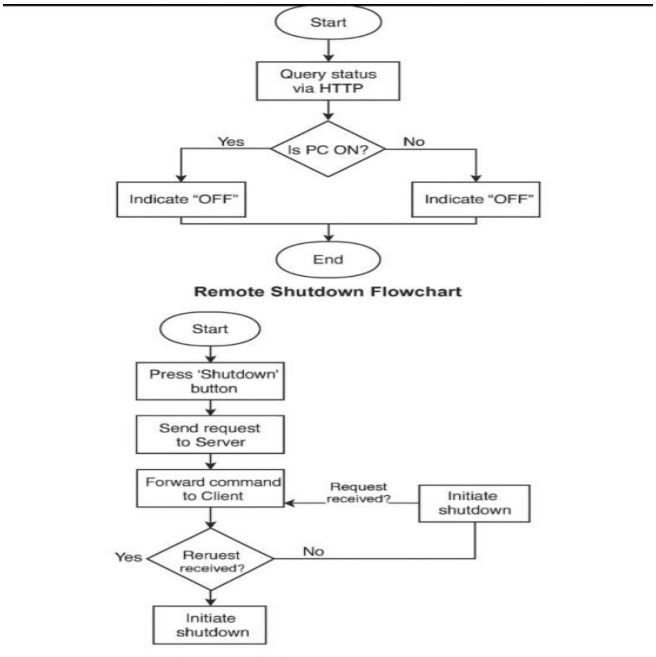

Fig. 1. Flowchart

V. PROPOSED SYSTEM

The suggested system offers a mobile based solution for monitoring and control of PCs linked via a Local Area Network (LAN), based on a client-server architecture. The system combines a React Native-constructed Android application, a central Flask server, and client agents deployed on single PCs. The system supports real-time control and monitoring without physical presence.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, June 2025
Impact Factor: 7.67

The Android app is used as the graphical user interface, through which the administrator can see the status of all PCs connected (ON/OFF), send shutdown commands. The Flask server, installed on any machine that exists on the LAN, is used as a middleware to manage communication between the PCs and the mobile app. It receives API requests, sends shutdown signals to targeted PCs.

Every PC has a lightweight client script (implemented in Python) that listens for shutdown requests and reports system status when asked. The mobile app, server, and client agents communicate with each other over HTTP via RESTful APIs secured within the LAN. This architecture of the system enables the administrators to be able to manage lab computers effectively, with the remote access flexibility, fast actions, and better power utilization efficiency.

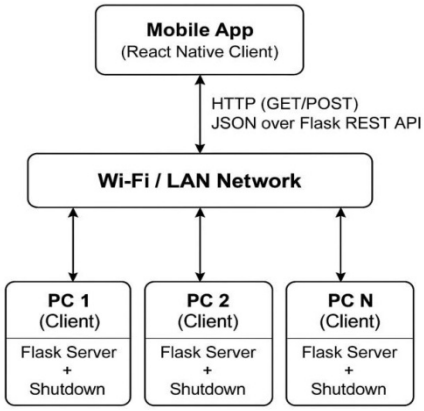

Fig. 2. Proposed System Architecture

VI. EXISTING WORKING / WORK

Numerous systems have, over the years, been designed to facilitate remote access, control, and monitoring of computers. The classic ones used to depend on desktop applications such as TeamViewer, Remote Desktop Protocol (RDP), or enterprise-level software such as Ansible and Microsoft Endpoint Configuration Manager. Although useful, they call for extensive configurations, internet connectivity, licensing, or are inefficient for lightweight, mobile-based control across local networks.

In research papers, previous research has investigated web-based interfaces for computer lab management, which were normally based on browser accessibility and manual intervention by an administrator. There were systems that implemented Raspberry Pi or Arduino-based controllers for simple device switching over Wi-Fi but were not scalable, fully integrated with software, or responsive on mobile platforms.

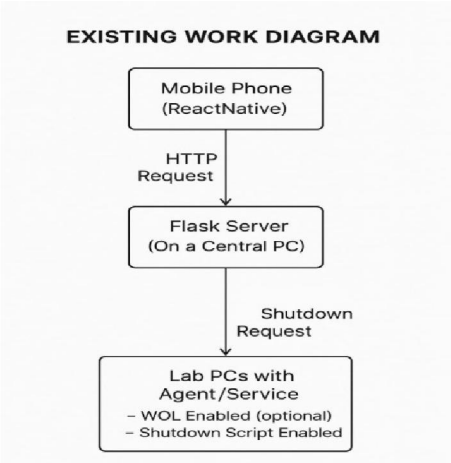

Fig. 3. Existing System Architecture for Remote PC Control

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, June 2025
Impact Factor: 7.67

Compare to that, recently mobile-based systems based on Android for IoT or device control have been popular, but the majority are restricted to smart home applications or necessitate sophisticated infrastructure like MQTT brokers, cloud servers, or costly hardware. There are apps already available on the Play Store but the majority are single-use tools that simply with no feedback mechanism, real-time monitoring, or shutdown control.

VII. RESULT AND WORKFLOW

The developed system was tested in a controlled setting within a laboratory environment. The testing was done using multiple client computers and 1 android smartphone that were connected over Wi-Fi. The server was able to discover all of the active clients that have the listener modules running successfully on them. The mobile application connected to the server as expected, and was able to perform imperative operations with accuracy and efficiency.

The system was tested in a true college lab environment under typical LAN conditions. The results show:

- Real-time PC ON/OFF status detection with negligible delay (less than 1–2 seconds).
- Successful shutdown of PCs connected through the mobile interface.
- Cross-platform compatibility with Androidbased smartphones.
- Support for scalability to 10+ PCs without performance loss.

This project represents an effective technique of being able to control networked PCs remotely over Wi-Fi with an Android device. It reduces the administrative overhead and provides a scalable, and easy to use solution when compared to existing solutions. Future enhancements may include encrypted communication, cloud hosting, and compatibility with additional platforms.

Workflow:

1. PC Status Detection:

When the app is opened, the user initiates a request for all PC statuses. The mobile app initiates a request to the Flask server, which in turn sends an inquiry to each PC's client agent. Feedbacks (ON/OFF) are returned and indicated in the app interface.

2. Remote Shutdown:

The user chooses a PC from the app interface and presses the shutdown button. This triggers an instruction to the Flask server, and it is relayed to the chosen PC's client. The client shuts down the system.

3. Status Feedback and Logging:

The system gives instant feedback on command success/failure through the app.This allows fortransparency, reliability, and rapid decision-making

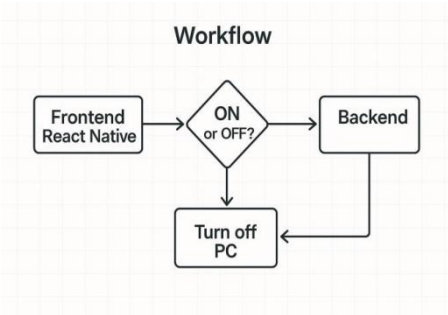

Fig. 4. System Workflow for PC Status Detection and Shutdown

VII. CONCLUSION

This project offers an effective and functional way to remotely control lab PCs through the use of mobile technology. With the integration of React Native, Flask, and LAN communication, the system supports real-time multiple PC control without the need for manual involvement. The system architecture is straightforward, affordable, and extensible to a wide range of institutional requirements. The system provides a smart substitute for conventional manual lab

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, June 2025
Impact Factor: 7.67

management, enhancing both administrative effectiveness and convenience to users. Future enhancement might encompass secure login capabilities, greater support for different OS, and cloud integration to handle remote networks.

IX. ACKNOWLEDGMENT

Special thanks to Prof. S. D. Patole, Assistant professor / guide for the project and Prof. A. A. Hipparkar, Head of Department, Computer Department.

REFERENCES

[1]. V. Parameswaran, S. Mallick, "Android-Based Remote PC Controller," International Journal of Computer Applications, vol. 125, no. 7, pp. 1–5, 2015.
[2]. R. Sharma, P. Kaur, "Home Automation System Using Android and Wi-Fi," International Journal of Scientific Research in Computer Science, vol. 6, no. 2, pp. 20–24, 2018.S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.
[3]. M. I. Sarwar et al., "Web-based Energy Management System for Computer Labs," International Journal of Emerging Technologies, vol. 11, no. 3, pp. 345–350, 2020.
[4]. A. A. Raut, P. U. Shinde, "Wake on LAN Based Power Management for PCs," Procedia Computer Science, vol. 132, pp. 591–598, 2018s.
[5]. B. Singh and R. Patel, "Network Monitoring using Mobile Apps," International Journal of Computer Networks and Wireless Communications, vol. 9, no. 1, pp. 15–22, 2019.