JSLHU
JOURNAL OF SCIENCE
OF LAC HONG UNIVERSITY

ISSN: 2525 - 2186
Tạp chí Khoa học Lạc Hồng, **2026**, 26, 001-008

ỨNG DỤNG CÔNG NGHỆ CLIENTLESS REMOTE DESKTOP GATEWAY TRONG XÂY DỰNG HỆ THỐNG PHÒNG THỰC HÀNH ẢO

Lâm Thành Hiển, Phan Thị Hường, Lê Hoàng Anh, Trần Văn Thành, Nguyễn Hữu Lộc, Nguyễn Minh Sơn*
Trường Đại học Lạc Hồng, Số 10 Huỳnh Văn Nghệ, Phường Trấn Biên, Đồng Nai, Việt Nam
*Tác giả liên hệ: *nmson@lhu.edu.vn*

THÔNG TIN BÀI BÁO

Ngày nhận:	06/03/2026
Ngày hoàn thiện:	27/03/2026
Ngày chấp nhận:	05/04/2026
Ngày đăng:	15/04/2026

TỪ KHÓA

Phòng thực hành từ xa;
Truy cập máy tính từ xa;
Giao thức RDP/VNC;
Kiến trúc mã nguồn mở;
Công bằng giáo dục.

TÓM TẮT

Đào tạo thực hành ngành kỹ thuật tại Việt Nam đang gặp thách thức lớn về chi phí và bất bình đẳng trong tiếp cận thiết bị phần cứng. Nghiên cứu này phát triển hệ thống phòng thực hành ảo từ xa (Clientless Remote Lab) sử dụng Apache Guacamole, cho phép sinh viên truy cập 24/7 qua trình duyệt (HTML5) mà không cần cài đặt phần mềm. Hệ thống tích hợp kiến trúc mã nguồn mở (Node.js, Nuxt.js, Docker) nhằm tự động hóa toàn bộ quy trình: đặt lịch, thanh toán qua PayOS và cấp phát tài nguyên tức thời. Thử nghiệm thực tế được tiến hành trên 20 máy ảo (Windows/Ubuntu) với 30 sinh viên để đánh giá hiệu năng giao thức RDP/VNC và tính khả dụng. Kết quả cho thấy hệ thống hoạt động tối ưu với băng thông thấp (100-150 Kbps cho tác vụ văn bản), độ trễ nhỏ (30-80 ms) và đạt điểm khả dụng SUS là 76.5 (xếp hạng "Tốt"). Mô hình này cung cấp một giải pháp chi phí thấp, thúc đẩy công bằng giáo dục trong môi trường hạn chế tài nguyên, dù vẫn còn giới hạn khi xử lý đa phương tiện. Các hướng phát triển tương lai bao gồm phân tích AI và hỗ trợ thiết bị di động.

APPLICATION OF CLIENTLESS REMOTE DESKTOP GATEWAY TECHNOLOGY IN BUILDING A VIRTUAL PRACTICE LAB SYSTEM

Lam Thanh Hien, Phan Thi Huong, Le Hoang Anh, Tran Van Thanh, Nguyen Huu Loc, Nguyen Minh Son*
Lac Hong University, No. 10 Huynh Van Nghe, Tran Bien Ward, Dong Nai, Vietnam
*Corresponding author: *nmson@lhu.edu.vn*

ARTICLE INFO

Received:	Mar 6th, 2026
Revised:	Mar 27th, 2026
Accepted:	Apr 5th, 2026
Published:	Apr 15th, 2026

KEYWORDS

Apache Guacamole;
Remote Desktop;
RDP/VNC Protocols;
Open-Source Architecture;
Educational Equity.

ABSTRACT

Practical training in technical disciplines in Vietnam faces significant challenges regarding high costs and the digital device divide. This study develops a clientless remote virtual lab system using Apache Guacamole, enabling students 24/7 browser-based access via HTML5 without client-side installations. The system utilizes an open-source architecture (Node.js, Nuxt.js, Docker) to automate the entire workflow, including booking, PayOS payment, and dynamic resource provisioning. The prototype was tested on 20 virtual machines (Windows/Ubuntu) with 30 students to evaluate RDP/VNC protocol performance and system usability. Results demonstrate stable operation with low bandwidth consumption (100-150 Kbps for text-based tasks), low latency (30-80 ms), and a "Good" System Usability Scale (SUS) score of 76.5. This low-cost model effectively promotes educational equity in resource-constrained environments, despite current limitations in multimedia processing. Future enhancements will integrate AI analytics and mobile device support.

Doi: https://doi.org/10.61591/jslhu.26.1092
Available online at: https://lhj.vn

1. INTRODUCTION

The COVID-19 pandemic has accelerated the global shift in higher education from traditional to blended and online models [1, 2]. In Vietnam, this transition is driven by Decision 749/QĐ-TTg, emphasizing digital transformation to promote inclusive education amid persistent digital divides [3].

Traditional physical computer labs expose critical limitations in this new landscape: high initial hardware investment and maintenance costs, spatial constraints limiting student numbers, and restricted access during office hours only [3]. A 2023 UNESCO report indicates that while 90% of Vietnamese schools have internet connectivity, a substantial digital divide persists in access to high-quality devices across regions and socioeconomic groups [5]. Many students lack personal computers powerful enough for resource-intensive simulations like CAD software, advanced IDEs, or virtual machines, exacerbating learning inequalities [2, 6].

In this context, "Remote Labs"—virtual environments for remote experimentation—emerge as a viable alternative. However, commercial solutions like TeamViewer, AnyDesk, or VPNs fall short for educational use: they demand client-side installations, incur steep licensing fees, and lack pedagogical features such as scheduling, real-time monitoring, or integration with learning management systems (LMS) [7, 8]. Drawing on these insights, this study investigates and develops a Remote Labs system based on a clientless remote desktop gateway, addressing the core challenge: delivering high-quality, 24/7 practice environments without user-side hardware demands, at minimal cost, and with centralized management for Vietnamese universities.

The research pursues four specific objectives. First, construct a system architecture using open-source Apache Guacamole to enable HTML5 browser access, eliminating plugins or client software. Second, automate business processes across the session lifecycle—from registration and payment to resource provisioning, monitoring, and deprovisioning [8]. Third, refine RDP and VNC protocol parameters for low latency suited to Vietnam's network conditions. Fourth, validate effectiveness through technical metrics (bandwidth, system resources) and user feedback via the System Usability Scale (SUS).

The study targets remote desktop technologies (RDP, VNC, SSH protocols), clientless gateway architectures, and user behaviors in online labs. Its scope encompasses IT practices like programming, network administration, and cybersecurity on Windows/Linux platforms within campus networks connected to public internet. Emphasis lies on software techniques and user experience, excluding in-depth physical IoT/embedded hardware design, though the system supports such extensions [4, 7].

A mixed-methods approach guides the inquiry: theoretical synthesis comparing RDP, VNC, SSH, and Guacamole architectures; experimental prototyping via Agile/Scrum with Node.js, Nuxt 3, PostgreSQL, and Docker; and quantitative assessment of performance (CPU, RAM, bandwidth using Prometheus/Grafana) alongside SUS for usability [9].

Scientifically, this work enriches knowledge on virtualization and remote desktops in education, proposing a low-cost lab model as a reference. Practically, it offers a ready-to-deploy technology for institutions, alleviating infrastructure shortages, enhancing remote training quality, and advancing educational equity in the digital era [2, 5].

The main contribution of this study lies in the practical integration and automation of open-source tools to solve resource shortages in developing regions. Specifically, the paper's key contributions are: (1) Designing an automated, end-to-end workflow (booking, payment, dynamic provisioning) tailored for educational environments; (2) Implementing a Dynamic Credential Provisioning mechanism for enhanced security without the need for complex directory services; and (3) Providing empirical performance evaluations comparing RDP and VNC protocols over Vietnam's internet infrastructure.

2. RELATED WORK

Research on remote laboratories has evolved from hardware control via proprietary protocols with security vulnerabilities [3] to web-based HTML5 solutions [11]. Virtual labs replicate physical experiments, promoting experiential learning and addressing resource constraints [12], occasionally integrating AI for personalized collaborative feedback [10]. Reviews indicate these labs predominantly support STEM fields using 2D synchronous interfaces [13]. Communication tools build common ground, though guidance mechanisms remain underrepresented [14]. Qualitative evaluations frequently report positive learning outcomes [11, 14], and hybrid models address pandemic-induced shifts by enhancing self-regulation [15].

In Vietnam, remote labs typically target subjects like automation using VPNs [16]. Post-COVID digitalization policies [2] confront device access inequalities despite 90% internet connectivity [5]. Recent studies highlight infrastructure and skill barriers to adoption [4, 17]. Comprehensive Remote Laboratory Management Systems (RLMS) with automated booking and payment remain scarce, a gap this study addresses via scalable, clientless solutions [4, 17]. Table 1 compares these remote access approaches.

Table 1. *Comparison of remote access solutions in education*

Criterion	TeamViewer / AnyDesk	VPN (Virtual Private Network)	Remote Lab (Apache Guacamole)
Installation	Requires agent on both client & host	Needs client and complex configuration	Clientless, browser-only
Security	Relies on third-party measures	Exposes internal networks if client compromised	Isolates via proxy, no direct host access
Cost	High for commercial licenses	Hardware gateway and licensing fees	Low, open-source with server infrastructure
Performance	Optimized for WAN	Tunneling overhead	Matches RDP/VNC internally, gateway-dependent
Management	Limited API integration	Network access only, no session control	Deep REST API for sessions, recording, permissions

This comparison reveals Guacamole's superiority in scalability and network isolation via open-source code [7, 8]. Outperforming VPNs and agent-based tools (e.g., TeamViewer), it enables installation-free browser access, aligning with web-based lab trends [14].

Clientless nature: Browsers receive display data directly via HTML5 canvas without understanding native protocols, eliminating the need for client-side plugins [18],[19], supporting diverse protocols: RDP is highly efficient as it transmits high-level graphics primitives (drawing commands), making it suitable for low-bandwidth environments. In contrast, VNC via the RFB protocol transmits raw pixel rectangles, making it highly bandwidth-intensive and sensitive to latency, especially during multimedia playback [20], [21].

Serving as an HTML5 gateway, Apache Guacamole optimizes canvas drawing and translates native RDP/VNC to the Guacamole Protocol via the guacd daemon[18]. Its clientless nature delivers display data without browser plugins [19, 25], enabling seamless educational integration to address existing accessibility gaps [10, 11].

3. METHODOLOGY

Developing clientless remote labs requires a structured methodology to address accessibility and equity in higher education. This section details the approach, drawing from recent studies on educational technology prototyping.

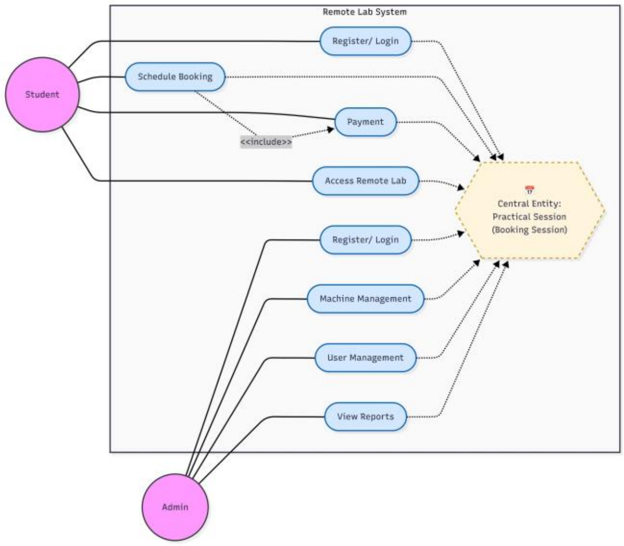

Figure 1. *Use Case Diagram illustrating student and admin interactions*

3.1. Requirement Analysis

The system serves students and administrators with specific needs [8]. Students register via email, view machine status (available/busy), book slots, pay through online gateways if required, and access labs directly in browsers without setup. Administrators manage machines (add, edit, delete, maintain), monitor active sessions, disconnect violations or overtime, and generate usage/revenue reports.

Figure 1 illustrates the use case diagram, showing interactions between actors and system functions.

3.2. System Architecture Design

A 3-tier architecture integrates microservices for flexibility.

Presentation Layer (Frontend): Built on Nuxt 3 (Vue.js framework), running on port 3000. Responsive interfaces use TailwindCSS. Functions include dashboards, registration forms, and Guacamole client integration via guacamole-common-js.

Application Layer (Backend): Express.js on Node.js, port 8000. Provides RESTful APIs for frontend, interacts with Guacamole REST API, handles PayOS payment logic, and runs cron jobs for session lifecycle management.

Data Layer (Database): PostgreSQL on Supabase Cloud. Manages user data, workstations, booking history. Knex.js serves as query builder for secure DB interactions.

Service Layer (Remote Gateway): Apache Guacamole server (guacd + Tomcat) acts as proxy, connecting to physical/virtual labs via RDP/VNC, streaming visuals to frontend.

3.2.1. Database Design

Databases are normalized for data integrity. Table 2 describes key tables:

Table 2. *Detailed description of the database tables*

Table Name	Role	Key Columns	Relations
users	Account management	id (PK, nanoid), username, email (Unique), password (bcrypt hash), role (student/admin)	Parent table, 1-1 with students
students	Student info	id (PK), email (FK users), full_name, phone, course_id	Holds training details
lab_machines	Workstation list	id (PK), guac_identifier (Guacamole connection ID), name, status (AVAILABLE/BOOKED), ip_address	Manages hardware status
booking_sessions	Booking sessions	id (PK), user_email (FK), machine_id (FK), guac_username, guac_password, start_time, end_time, status	Central link between users and machines, stores temporary credentials
payments	Transactions	id (PK), order_code (Unique PayOS), amount, status, payos_response	Tracks payments for reconciliation

Figure 2 shows the ER diagram, depicting relationships like 1:N from users to booking_sessions.

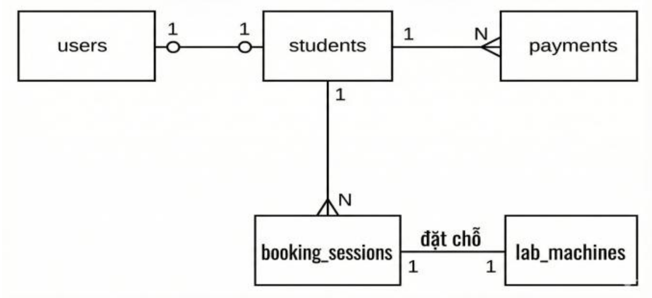

Figure 2. *ER Diagram Description*

3.2.2. UI Design

Interfaces follow minimalism to prioritize user experience [26]. Home page includes introduction and login/registration. Student dashboard features calendar views for available slots, with "Connect Now" buttons active only during booked times. Remote console maximizes screen space for desktops, with hidden toolbars for virtual keyboards or file uploads via Guacamole features [18]

3.2.3. System Implementation

Deployed on Ubuntu 22.04 LTS. Technologies: Node.js v18.x for backend runtime; Apache Guacamole v1.6.0 for gateway; PostgreSQL v15.x for data; Docker containerizes guacd [18]. Backend libraries: express, knex, node-schedule, bcryptjs, jsonwebtoken, axios for API calls.

Registration & Payment Module

Integrates PayOS for automated account creation. Workflow: User submits form; backend creates PayOS link; user scans QR; PayOS webhooks to backend; verifies checksum; inserts user in DB; sends confirmation email (Figure 3).

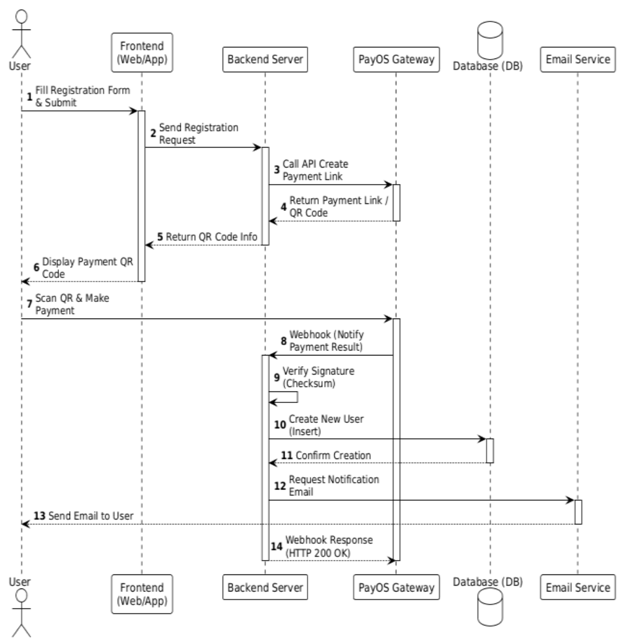

Figure 3. *Payment process Diagram*

Provisioning Module & Security Mechanisms

To ensure strict security without relying on complex, static directory services like LDAP, the system employs a "Dynamic Credential Provisioning" mechanism. It generates temporary, randomized Guacamole credentials via REST APIs just-in-time when a student books a slot. To prevent unauthorized access or token sharing, the system issues short-lived authentication tokens tightly mapped to the active session ID. Furthermore, the backend enforces IP-binding by verifying that the client IP requesting the WebSocket tunnel strictly matches the IP used during the booking process. Once provisioned, the machine status is locked to "BOOKED" to prevent race conditions see Figure 4.

Deprovisioning Module

A background cron job (implemented via node-schedule) runs every minute to query the database for sessions where the end time has passed. For each expired session, the system automatically invokes the Guacamole API to terminate the active connection and permanently delete the temporary user credentials. Simultaneously, the database updates the machine status back to "AVAILABLE" and marks the session as "EXPIRED." This ensures strict time-based access control and immediately frees up resources for the next user. This simplifies user interaction, handling RDP complexities internally [19].

Frontend Integration

Instead of native client software, the Vue.js/Nuxt frontend utilizes the guacamole-common-js library. It establishes an HTTP/WebSocket tunnel using the generated token and machine ID, rendering the remote desktop directly onto an HTML5 canvas element (Guacamole.Client). This handles all keyboard and mouse event abstractions natively within the browser.

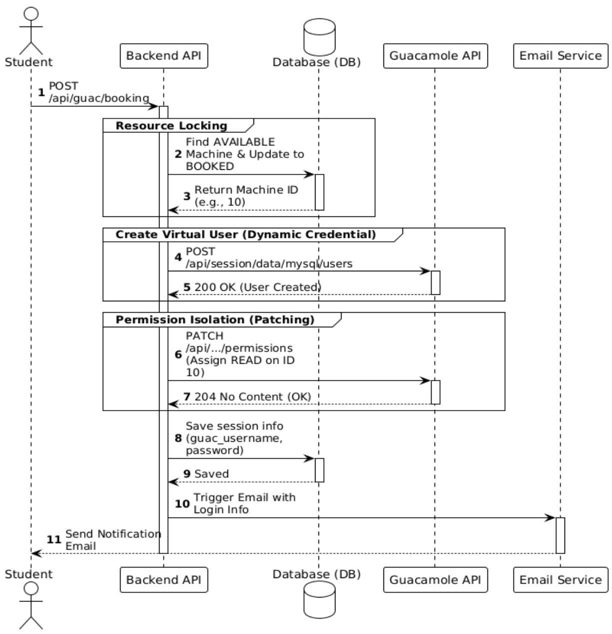

Figure 4. *Automated resource allocation process Diagram.*

4. RESULTS

In our pilot with 30 students, we observed that SUS responses varied by network quality, with rural participants reporting slightly lower scores (SD = 8.2), highlighting the need for further optimization. Pilot testing in educational technology frequently involves small-scale deployments to measure performance and usability, as seen in recent virtual lab studies. Bandwidth and latency metrics provide insights into real-world applicability, while SUS offers standardized user feedback.

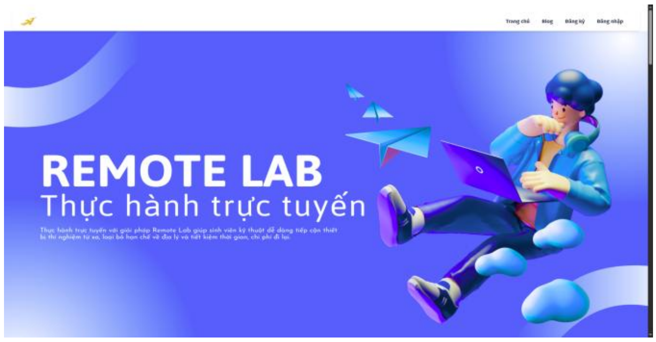

Figure 5. *The system's website interface*

4.1. System Demo

The system underwent pilot testing in a simulated lab with 20 Virtual Machines (VMs) hosted on an on-premise hypervisor—10 running Windows 10 and 10 on Ubuntu Desktop. User interfaces functioned reliably, allowing students to log in, view machine lists and statuses (available/busy), book slots, and connect in under 2 seconds via the "Connect" button. Remote desktops displayed fully, supporting bidirectional copy-paste text through clipboard redirection, enhancing practical tasks like coding or document editing [7].

4.2. Performance Evaluation

To ensure statistical reliability, each experimental scenario (text-based and multimedia) was repeated 10 times under uniform testing conditions using standard 4G and household Wi-Fi clients connecting to the on-premise server. Results are reported as Mean ± Standard Deviation (SD).

Table 3. *Results of bandwidth consumption measurement (10 repetitions/scenario)*

Scenario	Protocol	Average Bandwidth (Mean ± SD)	Performance Gap (RDP vs VNC)	Remarks
Text Editing / Coding	RDP	125 ± 15 Kbps	~65% less bandwidth than VNC	Low, suitable for 3G/4G; RDP sends graphics commands only
Text Editing / Coding	VNC	380 ± 45 Kbps	Baseline	Higher due to pixel updates; less efficient
YouTube Video (480p)	RDP	3.2 ± 0.5 Mbps	~40% less bandwidth than VNC	Spikes from constant screen changes; manageable
YouTube Video (480p)	VNC	5.4 ± 0.8 Mbps	Baseline	Severe lag; Guacamole's JPEG compression strains performance

The underlying architectural difference explains this performance gap: RDP is highly efficient as it transmits high-level graphics primitives (drawing commands), whereas VNC (via RFB protocol) transmits raw pixel rectangles, making it highly bandwidth-intensive and latency-sensitive. Furthermore, stress testing revealed that the theoretical bottleneck of the Guacamole Gateway (guacd) lies in the CPU rather than RAM. This is due to the heavy computational overhead required for real-time image compression (re-encoding VNC/RDP frames into WebP/JPEG for the browser) and TLS/SSL encryption. RAM usage remained relatively low and stable per connection.

The system excelled in core learning activities (programming, network configuration, Office software), requiring under 200 Kbps—fitting household internet in Vietnam [5] Latency ranged from 30-80 ms, fully acceptable for mouse/keyboard operations, as levels below 150 ms meet RDP standards without noticeable delays [20].

4.3. User Survey - SUS

Usability was quantified using the System Usability Scale (SUS) on 30 participating students. Results yielded an average score of 76.5/100, corresponding to "Good" usability and a B-grade rating. This exceeds the industry benchmark of 68, indicating the clientless design and streamlined processes reduced technical barriers effectively [23].

Qualitative feedback highlighted strengths: easy access anytime/anywhere without complex VPN setups. Drawbacks included blocked system shortcuts (e.g., Alt+Tab by browsers) and suboptimal video smoothness on remote machines [24].

Table 4. *The pilot aligned closely with goals.*

Objective	Achievement	Assessment
24/7 Clientless Access	Achieved; browser-based (Chrome/Edge/Firefox)	Success
Automated Provisioning	Achieved; full booking/ provisioning automation	Success
Equity Assurance	Achieved; time limits and auto-kick mechanisms	Success
Stable Performance	Achieved for learning tasks; limited for multimedia	Acceptable

These outcomes validate the system's readiness for broader deployment in non-graphics-intensive courses [9].

5. DISCUSSION

Our clientless remote lab system effectively addresses post-COVID educational shifts by enabling browser-based access, thereby promoting equity in resource-constrained Vietnamese universities [2, 5]. Pilot results show low resource demands suit Vietnam's infrastructure, with SUS feedback highlighting user-friendly design. However, multimedia and shortcut issues suggest needs for protocol enhancements.

Pilot Implementation Insights

The 20-VM setup (10 Windows, 10 Ubuntu) demonstrated reliable interfaces: logins, bookings, and <2-second connections. Clipboard redirection aided practical tasks, aligning with collaborative trends in virtual labs [14, 15].

Performance Analysis

Bandwidth for text/coding via RDP averaged 100-150 Kbps, fitting 3G/4G networks and exceeding VNC's 300-500 Kbps due to graphics primitives over pixel transmission [20]. Video at 480p spiked to 2.5-4 Mbps (RDP) and >5 Mbps (VNC), causing lag from compression—common in web gateways but limiting multimedia in education [22]. Latency of 30-80 ms supported inputs without delays, below 150 ms benchmarks for responsive remoting [22]. These metrics confirm viability for IT practices like programming, matching low-bandwidth needs in unequal settings [2, 5].

Table 4 highlights efficiencies for core activities, underscoring RDP's edge in Vietnam's variable internet [6].

Usability Assessment

SUS mean of 76.5 rated "Good" (B-grade), surpassing the 68 benchmark for industrial systems [23]. This reflects clientless design's barrier reduction, with feedback praising anytime access sans VPN complexity—key for equity [1] Shortcuts blocked by browsers and video choppiness echo QoE challenges in virtual education, where qualitative issues like motivation affect adoption [15, 24].

Objective Alignment

24/7 access succeeded via HTML5, automation streamlined provisioning, equity via time limits, and performance held for learning tasks—acceptable despite media limits [12]. This positions the model as a low-cost reference for developing regions, advancing SDG 4 amid Vietnam's digital gaps [5].

6. LIMITATIONS AND FUTURE WORK

6.1. Limitations

Remote lab implementations encounter several constraints that affect their broader application. Network dependency stands out as a primary issue: user experience hinges on internet stability, which can falter in rural or low-bandwidth areas of Vietnam, leading to inconsistent access and potential dropouts during sessions [2, 5]. This aligns with findings from recent reviews, where inadequate infrastructure limits participation and exacerbates digital divides, with only 90% of schools connected but device access uneven [11].

Multimedia handling presents another limitation. Protocols like RDP and VNC, when channeled through web gateways, struggle with video streaming or 3D graphics, resulting in lag during tasks such as watching instructional videos (bandwidth >5 Mbps for VNC at 480p). This restricts the system to non-intensive courses, as compression overheads degrade performance without dedicated acceleration [20, 21]. Student feedback echoed this, noting choppy video and browser-blocked shortcuts (e.g., Alt+Tab), issues common in virtual environments where foundational hardware knowledge gaps amplify usability problems [14, 15].

Additionally, the absence of a queuing mechanism leads to "out of machines" notifications during peak times, without automated waitlists or notifications for availability. This overlooks collaborative demands, where over half of remote labs lack coordination tools, potentially reducing engagement in group-based learning [12, 31]. Privacy and security in cloud-based setups also warrant caution, as data storage raises concerns in shared educational platforms [14, 19].

Furthermore, while the pilot with 30 students is sufficient for identifying preliminary UI/UX issues via the SUS survey, this sample size lacks the broad statistical power required for a comprehensive impact analysis. Future deployments will involve larger cohorts across multiple semesters to ensure statistical significance.

6.2. Future Directions

To address these gaps, several enhancements are proposed. First, implement smart queuing with WebSocket for real-time waitlists, automatically notifying users of freed machines and holding spots—improving scalability in high-demand scenarios [14, 19].

Second, integrate AI analytics to process log data for optimizing schedules, suggesting off-peak slots, and detecting anomalies like unusual behaviors. This builds on recent AI trends in higher education, where data-directed tools personalize learning and automate administrative tasks [10].

Third, optimize protocols through hardware acceleration or advanced codecs like H.264 in RDP, reducing latency for multimedia and enabling 3D/graphic applications. This could extend the system's scope to fields like engineering simulations, aligning with hybrid lab advancements [12, 15, 22].

Fourth, expand mobile support with touch input abstractions, refining controls for tablets and ensuring responsive interfaces. This taps into growing mobile learning integration, enhancing accessibility in resource-limited settings [4].

These directions position the system for wider adoption, contributing to Vietnam's digital transformation by fostering equitable, flexible education [16].

Finally, to address the critical security risk of students installing malware or deleting system files during their practice, future versions will integrate Hypervisor APIs (e.g., Proxmox or VMware API). When a session expires or is revoked, the system will automatically trigger a 'Revert to Snapshot' command, ensuring a clean, malware-free environment for the next user."

7. REFERENCES

[1] Pham, H. T., Nguyen, T. T., & Le, T. T. (2023). Digital transformation in Vietnamese higher education: Opportunities and challenges. Journal of Education and e-Learning Research, 10(2), 123–130. https://www.researchgate.net/publication/390655837_Digital_transformation_in_Vietnam%27s_education_Opportunities_challenges_and_development_strategies

[2] Thu, L. T. T. (2024). Digitalization of higher education in Vietnam. Journal of Comparative & International Higher Education, 16(2), 23. DOI: 10.32674/jcihe.v16i2.5710

[3] Nickerson, J. V., Corter, J. E., Esche, S. K., Chassapis, C., Im, S., & Ma, M. J. (2007). A model for evaluating the effectiveness of remote engineering laboratories and simulations in education. Computers & Education, 49(3), 708–725. DOI: https://doi.org/10.1016/j.compedu.2005.11.019

[4] Son, P. (2024). The current state of virtual reality and augmented reality adoption in Vietnamese education: A teacher’s perspective on teaching natural sciences. International Journal of Information and Education Technology, 14(3), 476–485. DOI: https://doi.org/10.18178/ijiet.2024.14.3.476-485

[5] Antoninis, M., Alcott, B., Al Hadheri, S., April, D., Fouad Barakat, B., Barrios Rivera, M., ... & Weill, E. (2023). Global Education Monitoring Report 2023: Technology in education: A tool on whose terms?. DOI: https://doi.org/10.54676/UZQV8501

[6] Georgakopoulos, I., Piromalis, D., Ntanos, S., Zakopoulos, V., & Makrygiannis, P. (2023). A Prediction Model for Remote Lab Courses Designed upon the Principles of Education for Sustainable Development. Sustainability, 15(6), 5473. DOI: https://doi.org/10.3390/su15065473

[7] S. Jacobs et al., "Design and Development of a Multi-Purpose Collaborative Remote Laboratory Platform," 2024 IEEE Global Engineering Education Conference (EDUCON), Kos Island, Greece, 2024, pp. 1-3.

DOI: 10.1109/EDUCON60312.2024.10578743.

[8] Hassan, I. (2022, October). Levereging Apache Guacamole, Linux LXD and Docker Containers to Deliver a Secure Online Lab for a Large Cybersecurity Course. In 2022 IEEE Frontiers in Education Conference (FIE) (pp. 1-9). IEEE.

[9] Kanetaki, Z., Stergiou, C., Bekas, G., Jacques, S., Troussas, C., Sgouropoulou, C., & Ouahabi, A. (2022). Grade Prediction Modeling in Hybrid Learning Environments for Sustainable Engineering Education. *Sustainability*, *14*(9), 5205. DOI: https://doi.org/10.3390/su14095205

[10] Kovari, A. (2025). A systematic review of AI-powered collaborative learning in higher education: Trends and outcomes from the last decade. Social Sciences & Humanities Open, 11, 101335.

[11] Hanine, H., Farajy, N., & Moumen, A. (2025). Virtual Laboratories in STEM Education: A Scoping Literature Review on E-Learning Innovation. Engineering Proceedings, 112(1), 17.

[12] May, D., Terkowsky, C., Varney, V., & Boehringer, D. (2023). Online laboratories in higher engineering education–solutions, challenges, and future directions from a pedagogical perspective. European Journal of Engineering Education, 48(5), 779-782.

[13] Chen, F., Lin, Z. & Li, X. Research on the emerging technological intervention models in design education from a strategic perspective of global design education institutions. Sci Rep 15, 41366 (2025). DOI: https://doi.org/10.1038/s41598-025-25272-1

[14] Kurtz, M., Benabbou, A., Pons, C., & Broisin, J. (2025). Collaboration in virtual and remote laboratories for education: A systematic literature review: M. Kurtz et al. International Journal of Computer-Supported Collaborative Learning, 20(4), 549-603.

[15] Zhang, Y., Stöhr, C., Jämsvi, S. S., Kabo, J., & Malmqvist, J. (2025). Student Experiences of Hybrid and Online Engineering Labs in a Logic Control Course. Computer Applications in Engineering Education, 33(3), e70032.

[16] Huong, P. T., My, N. T., Nga, N. T. H., & Van, P. D. (2021). Current situation of natural sciences laboratories and factors affecting the frequency of natural science laboratory teaching at some lower secondary schools in the north central region of Vietnam. Journal of Management Information & Decision Sciences, 24(3).

[17] Son, P. N. (2024). The current state of virtual reality and augmented reality adoption in Vietnamese education: A teacher's perspective on teaching natural sciences. International Journal of Information and Education Technology, 14(3), 476-485.

[18] Apache Software Foundation. (2025). Apache Guacamole manual (Version 1.6.0). https://guacamole.apache.org/doc/gug/

[19] E. Magaña, I. Sesma, D. Morató, and M. Izal, "Remote access protocols for Desktop-as-a-Service solutions," PLoS ONE, vol. 14, no. 1, p. e0207512, Jan. 2019. DOI: doi: 10.1371/journal.pone.0207512.

[20] T. Song et al., "FastDesk: A remote desktop virtualization system for multi-tenant," Future Generation Computer Systems, vol. 81, pp. 478–491, Apr. 2018.

[21] Y. Piao et al., "AgentBay: A Hybrid Interaction Sandbox for Seamless Human-AI Intervention in Agentic Systems," arXiv preprint arXiv:2512.04367, Dec. 2025.

[22] J. Arellano-Usón, D. Morató, and M. Izal, "Protocol-agnostic method for monitoring interactivity time in remote desktop services," Multimedia Tools and Applications, vol. 80, no. 13, pp. 19513–19539, May 2021. DOI: 10.1007/s11042-021-10708-3.

[23] Sauro, J. (2018). 5 ways to interpret a SUS score. https://measuringu.com/interpret-sus-score/

[24] Tan, C., & Lin, J. (2021). A new QoE-based prediction model for evaluating virtual education systems with COVID-19 side effects using data mining. Soft Computing, 27(3), 1699–1713. DOI: https://doi.org/10.1007/s00500-021-05932-w