

Specialized System on Managing Virtual Machine with Proxmox for DLSU - College of Computer Studies

Jonathan Lin¹, Rafael Christian Sanchez¹, Patrick Bennett Ong¹, Angeline Shi¹ and Gregory Cu^{1,*}

¹ De La Salle University

*Corresponding Author: gregory.cu@dlsu.edu.ph

Abstract: This study addresses the challenges faced by the Technical Support Group of the College of Computer Studies at De La Salle University in managing their Proxmox Virtual Environment. The existing system faces a number of critical issues such as error-prone mass creation of virtual machines, difficulty in identifying necessary/idle VMs, unintended network accessibility, and potential performance degradation due to resource allocation inefficiencies. Some challenges arising could be addressed with the present study, developing a unique system for easy integration among multiple open-source technologies such as Ansible for automation, Guacamole for remote desktop access, InfluxDB for metric storage, and pfSense to manage firewalls. Development was done by following Scrum methodology with eight sprint cycles, containing all modules for user management, tracking, virtual machine management, monitoring, and reporting. The User Acceptance Testing conducted with both TSG and faculty users showed the system to be effective, with the majority of users rating good for such key features. All TSG users rated the automation of virtual machine creation and deletion processes as excellent, while 83.3% of the faculty users rated the completeness of information of the system as good. The results obtained indicate that this system eases virtualization management processes and can go a long way in saving money for all educational institutions by optimizing virtual infrastructure management.

Key Words: virtualization management; Proxmox VE; educational technology; open-source integration

1. INTRODUCTION

Virtual machines are now central to modern computing, revolutionizing software deployment and management. Virtualization creates independent virtual systems beyond physical hardware, allowing single servers to efficiently run multiple environments (Brush, Hashemi-Pour, & Kirsch, 2023), (Shaw, 2023).

Educational institutions increasingly adopt virtualization for cost-effectiveness and scalability. VMware reports up to 50% hardware and 80% power/cooling cost reductions (Salameh, 2023). This technology is valuable in education, enabling rapid deployment of services and adaptation to changing computational needs (Vinod, 2023).

Traditional educational deployments suffered from poor resource isolation, low hardware utilization, and high server maintenance costs. These limitations drive the shift to virtualization for better resource use

and lower operational expenses. Virtual machines are common in education (Klement, 2017), enhancing student skills in information management, industry software, and communication technologies.

De La Salle University-Manila – College of Computer Studies adopted Proxmox Virtual Environment (Proxmox VE) (Proxmox, 2025) for deploying compute resources across subjects, managed by their Technical Support Group (TSG).

This platform was crucial during the COVID-19 pandemic, facilitating online classes with consistent compute resources and remote lab access, essentially providing a private cloud infrastructure for the college.



1.1 Challenges in the Usage of Proxmox VE System

The college's TSG staff faces several challenges with Proxmox VE's default configuration, impacting efficiency, stability, and security, especially during peak periods.

Specific Challenges:

- **Error-prone mass virtual machine / user creation:** Manually edited scripts for bulk virtual machine and user creation are prone to errors and inconsistencies, especially during high-demand periods.
- **Difficulty assessing virtual machine necessity:** Proxmox VE lacks essential metadata (creator, purpose, dates), making it difficult to determine virtual machine necessity and accountability.
- **Challenges identifying idle virtual machines:** Identifying abandoned, resource-consuming virtual machines requires manual inspection of each machine.
- **Unintended network accessibility:** Unremoved firewall ports of deleted virtual machines can inadvertently expose new VMs using the same IPs.
- **Risk of excessive virtual machine creation:** Absence of centralized request tracking can lead to duplicate and wasteful VM deployments.
- **Performance degradation risk:** The lack of a centralized request system can lead to duplicate VMs and wasted resources.

Potential Performance Degradation: Reactive VM allocation and the absence of real-time resource monitoring increase the risk of overutilization and performance conflicts from ad hoc assignments.

1.2 Proposed Solution

To address the challenges in section 1.1, we propose a custom virtual machine management system aligned with the college's workflow.

This solution mitigates these challenges through a dedicated system integrating open-source technologies with Proxmox VE. The conceptual framework of this solution, illustrated in Fig. 1, outlines the key components:

- A centralized ticketing system for faculty VM requests, offering real-time tracking, automated validation, and consistent resource allocation,

eliminating duplicates.

- Automation of VM creation, configuration, and deletion using Ansible and the Proxmox VE REST API, enabling uniform, large-scale deployments and reducing errors.
- Performance monitoring via InfluxDB for real-time resource utilization and system health insights, facilitating proactive resource management and identification of idle VMs.
- Enhanced network security through automated pfSense (pfSense, 2025) firewall management, including rule configuration, tracking, and automatic removal upon VM deletion.
- Secure remote access via a web-based Guacamole (Apache Software Foundation, 2025) interface, eliminating the need for specialized clients.
- A reporting system providing detailed analytics on resource use, user activity, and system performance to inform administrative decisions on resource allocation and capacity planning.

This integrated solution simplifies private cloud infrastructure operation and management for the college, ensuring security, efficiency, and performance.



Fig. 1 Conceptual framework of the proposed solution

2. SYSTEM ARCHITECTURE

The system is a centralized management platform that integrates various open-source technologies to automate and streamline VM management. As shown in the system architecture in



Fig. 2, it uses Django for web development, MySQL for the database, and includes modules for ticketing, user management, VM management, notifications, and monitoring. Guacamole provides remote desktop access to Proxmox VMs.

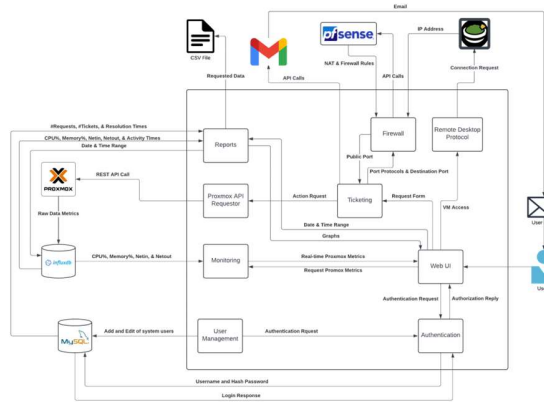


Fig. 2 System Architecture

2.1 Key Integration Points

- Data Flow and Management:** The system processes data from faculty VM request forms, including ISO image requirements, VM quantity, course code, hardware specifications, software needs, and internet accessibility. This information is used to automate VM provisioning via Proxmox APIs.
- Request Processing and Management:** The Ticketing module logs all submissions with essential information and tracks their status.
- Network Configuration:** pfSense manages IP and port assignments.
- Database Architecture:** The relational database stores user data, system data, audit trails, IP address assignments, and user credentials.
- Monitoring and Resource Management:** The monitoring module uses Proxmox's API and InfluxDB to collect and process resource utilization metrics.

- Reporting Capabilities:** The reporting module fetches statistics from InfluxDB to provide reports on resource utilization, course-specific patterns, and system performance.

Fig. 3 shows a sample dashboard for the monitoring module of the system. The monitoring module shows the resources being used in the system.

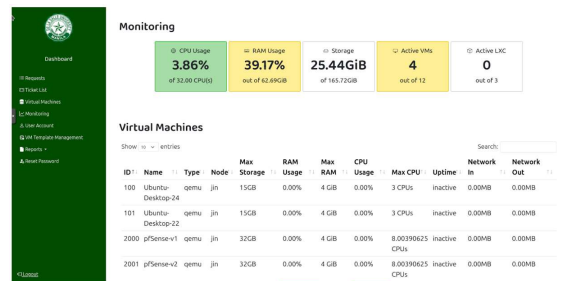


Fig. 3 Dashboard of the monitoring module

3. BUSINESS PROCESSES

The process begins with faculty submitting virtual machine requests. During this initial stage (Fig. 3), TSG verifies the specifications, approving them or requesting adjustments from the faculty. Upon approval, the system proceeds to the VM creation phase (Fig. 4), where a test VM is generated. This test VM allows faculty and TSG to install and verify course-specific software and configurations before mass deployment. Mass provisioning occurs only after the test VM is validated. In the final phases, created VMs are deployed for classroom use, and usage is monitored. Faculty are consulted about VM retention upon completion or term end. Confirmed VMs are then systematically removed, freeing resources for future use (Fig. 5).

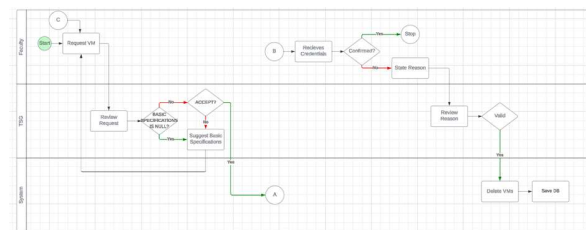


Fig. 4 Virtual machine creation request process

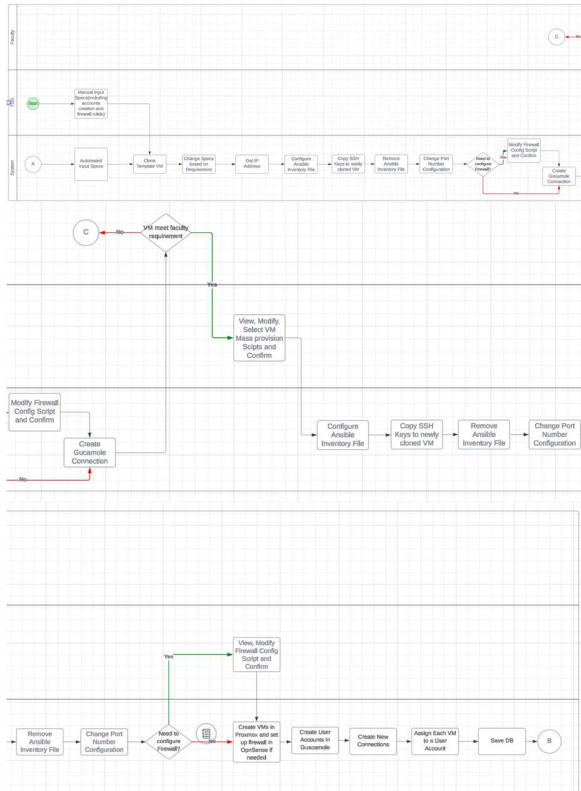


Fig. 5 VM creation process

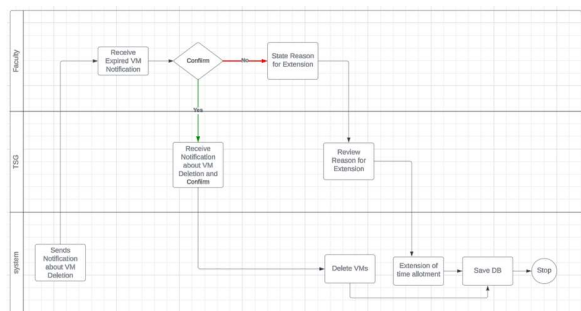


Fig. 6 VM deletion process

4. VALIDATION RESULTS

Detailed test cases were developed to ensure the software's robustness and reliability. Pre-defined scenarios are defined by these test cases, which the

system must handle under both normal and unexpected conditions.

Table 1 to Table 3 confirms the virtual machine request workflow from submission to processing. Table 1 shows faculty successfully submitting various virtual machine requests (Research, Thesis, Test, Class Course) via the web application. Table 2 validates request submissions, focusing on port forwarding and form editing. It confirms faculty can submit requests with or without port forwarding. Table 3 shows TSG processing requests through status changes ('Pending,' 'For Revision,' 'Processing,' 'Completed,' 'Deleted') and their ability to accept or delete requests.

Table 1. Result of requesting virtual machines in the web application functionality

Module	User	Scenario	Action	Expected Result	Actual Result
Ticketing Module	Faculty Members	Request form for Research	Complete the form and input 'Research' for use case	Successfully submits and saves data in the database	Pass
		Request form for Thesis	Complete the form and input 'Thesis' for use case	Successfully submits and saves data in the database	Pass
		Request form for Test	Complete the form and input 'Test' for use case	Successfully submits and saves data in the database	Pass
		Request form for Class Course	Complete the form, input 'Class Course' for use case, and enter correct section format	Successfully submits and saves data in the database	Pass
		Request form for Class Course with incorrect section format	Complete the form, input 'Class Course' for use case, and enter incorrect section format	Form does not submit	Pass

Table 2. Result of requesting virtual machines in the web application functionality

Module	User	Scenario	Action	Expected Result	Actual Result
Ticketing Module	Faculty Members	Request form without Port Forwarding	Complete the form and input 'No' for Port Forwarding with any use case.	Successfully submits and saves data in the database.	Pass
		Request form with Port Forwarding	Complete the form and Enter one port protocol	Successfully submits and saves data in the database.	Pass
		Request form with Port Forwarding	Complete the form and Enter multiple port protocols	Successfully submits and saves data in the database.	Pass
		Edit a request form	Edit anything within the form	Successfully edits the form and saves new data in the database.	Pass
		Unsuccessfully edit a request form	Leave some inputs blank and use wrong section format for 'Class Course' use case	Form does not submit	Pass

Table 3. Result of requesting virtual machines in the web application functionality

Module	User	Scenario	Action	Expected Result	Actual Result
Ticketing Module	Faculty Members	Change status of request to 'Pending'	Create a request form with valid inputs	Request status changes to 'Pending'	Pass
		Change status of request to 'For Revision'	Add a comment to the request form	Request status changes to 'For Revision'	Pass
	TSG Members	Change status of request to 'Processing'	Accept the request form	Request status changes to 'Processing'	Pass
		Change status of request to 'Accepted'	Click the test virtual machine	Request status changes to 'Accepted'	Pass
	TSG Members	Change status of request to 'Ongoing'	Click the Mass Provision button and wait for the virtual machines to be cloned	Request status changes to 'Ongoing'	Pass
		Change status of request to 'Deleted'	Click the Delete button on a recurring request	Request status changes to 'Deleted'	Pass
	TSG Members	Change status of request to 'Completed'	Wait for the expiration date of the non recurring request	Request status changes to 'Completed'	Pass



In Table 4, when a VM request is processed by TSG staff, a test virtual machine is first created. Test VM creation and mass provisioning are shown to be successful in Proxmox VE. Additionally, NAT and Firewall rules are correctly configured in pfSense, and remote desktop access is functional for all users (TSG, faculty, and students).

Table 4. Result of managing virtual machine provision and related configurations in the web application functionality

Module	User	Scenario	Action	Expected Result	Actual Result
Virtual Machine Module	TSG Members	Create test virtual machines	Accept a Pending request form	A test virtual machine is created in Proxmox	Pass
		Mass produce all the needed virtual machines	Click Mass Provision button in a Accepted request form	All the virtual machines are created in Proxmox	Pass
Firewall Module	TSG Members	Creates all the NAT and Firewall rules for the virtual machines	Click Mass Provision button in a Accepted request form with Port Forwarding	NAT and Firewall rules are created in pfSense dashboard and public port is displayed in UI	Pass
		Deletes all the NAT and Firewall rules associated with the request	Click the Delete button on a recurring request	NAT and Firewall rules are deleted in pfSense dashboard	Pass
Remote Desktop Gateway	TSG Members	TSG can access the test virtual machine	Click the Open VM button on a request details with the status of 'Processing'	Opens a new tab to connect to the virtual machine	Pass
	Faculty Members	Faculty can access the test virtual machine	Click the Open VM button on a request details with the status of 'Processing'	Opens a new tab to connect to the virtual machine	Pass
	TSG/Faculty Members	Can access the mass provisioned virtual machines	Go to virtual machine list, click on any virtual machine, and click the Open VM button	Opens a new tab to connect to the virtual machine	Pass
	Student	Can access the virtual machine	Log in to the website and click the Open VM button	Opens a new tab to connect to the virtual machine	Pass

Table 5 shows the Monitoring Module effectively tracks server performance. TSG members can view aggregated server performance metrics (with color-coded indicators) and access detailed metrics and hardware specifications for individual virtual machines. The virtual machine list is sortable by clicking column headers.

Table 5. Result of metrics monitoring in the web application functionality

Module	User	Scenario	Action	Expected Result	Actual Result
Monitoring	TSG Members	Can view the aggregated performance metrics of all the servers	Click the Monitoring in the sidebar and scroll up	Aggregated performance metrics are displayed at the top with color coded based on percentage levels	Pass
		Can view performance metrics and hardware specs of each virtual machines	Click the Monitoring in the sidebar and scroll down	Performance metrics are displayed for active virtual machines and hardware specs are displayed for all	Pass
		Can filter the virtual machine list	Click on a head label column in the virtual machine list	View will filter descending or ascending based on the clicked metric	Pass

As detailed in Table 6, the Reports Module offers comprehensive reporting capabilities. TSG members can access various report types (Ticketing, System, Subject, and Individual) with customizable filters and date ranges. The reports accurately display graphs and tables based on selected filters.

Furthermore, System, Subject, and Individual Reports can be downloaded as CSV files for further analysis.

Table 6. Result of system reporting in the web application functionality

Module	User	Scenario	Action	Expected Result	Actual Result
Reports	TSG Members	Can view the Ticketing Report with different filters	Click the Ticketing Reports in the sidebar under Reports and click different filters to modify the view of the report	Graphs and tables are displayed according to the filter chosen	Pass
		Can view the System Report with different filters	Click the System Reports in the sidebar under Reports and choose a date range and click different filters to modify the view of the report	Graphs and tables are displayed according to the filter chosen	Pass
		Can download CSV file from System Report	Click the System Reports in the sidebar under Reports and choose a date range and click CSV button	Download a file containing the data shown in the table of the web-user interface	Pass
		Can view the Subject Report with different filters	Click the System Reports in the sidebar under Reports and choose a date range and click different filters to modify the view of the report	Graphs and tables are displayed according to the filter chosen	Pass
		Can download CSV file from Subject Report	Click the Subject Reports in the sidebar under Reports and choose a date range and click CSV button	Download a file containing the data shown in the table of the web-user interface	Pass
		Can view the Individual Report with different filters	Click the Individual Reports in the sidebar under Reports and choose a date range and click different filters to modify the view of the report	Graphs and tables are displayed according to the filter chosen	Pass
		Can download CSV file from Individual Report	Click the Individual Reports in the sidebar under Reports and choose a date range and click CSV button	Download a file containing the data shown in the table of the web-user interface	Pass

4.1 User Acceptance Test (UAT)

The User Acceptance Test (UAT) at De La Salle University's College of Computer Studies yielded very positive results from both TSG and faculty users. Testing, conducted face-to-face and via Zoom, showed that all TSG users rated label legibility, system navigation, and data metric layout as excellent. Among faculty users, 83.3% rated the Request Form and Request Details functionality as 'excellent,' and 66.7% gave the same rating to virtual machine automation features. Guacamole access to virtual machines was rated 'excellent' by 83.3% of faculty, while the comment section for TSG communication received 'very good' ratings from 66.7%. Overall, the ticketing system and email notifications performed exceedingly well, meeting the needs of both user groups.

5. CONCLUSION

The study in the system successfully created a virtualization-based system that follows academic requirements by automating virtual machine management, enabling students to access software applications, and monitoring resource allocation effectively. Based on the User Acceptance test, the

system had surmounted certain challenges identified, where system features concerned with automatic VM creation attained ratings of very good, completeness of request details attained excellent ratings by 83.3% of respondents.

This study demonstrated that the need for the provision of virtualized environments, while monitoring system resources and VM utilization, is a dire need at the educational institution level. With the universities depending mainly on the provision of technical resources for their students, the capability to manage and monitor the usage of virtual machines accordingly facilitates smooth operations and brings costs down by availing open-source solutions. Besides, the centralized request process and the monitoring within the system prevent superfluous resource allocation and thus provide better overviewing over virtual machine usage, proving to be effective in meeting technical and educational needs.

User feedback on this would provide some key enhancements: search functionality to better control the VMs, network segmentation to better provide separation of duties based on VM purposes, and a snapshot capability for fast system state recovery. Needed features for faculty users include enhanced visual cues like color-coded status updates and tooltips. This will be further complemented by the integration of the NoVNC module for access to virtual machines without installed operating systems.

6. REFERENCES

- Brush, K., Hashemi-Pour, C., & Kirsch, B. (2023). *Virtualization*. Retrieved from TechTarget: <https://www.techtarget.com/searchitoperations/definition/virtualization>
- Klement, M. (2017). Models of integration of virtualization in education: Virtualization technology and possibilities of its use in education. *Computers and Education*, 31-43.
- pfSense. (2025). *pfSense*. Retrieved from pfSense Overview: <https://www.pfsense.org/about-pfsense/>
- Proxmox. (2025). *Proxmox*. Retrieved from Proxmox Virtual Environment: <https://www.proxmox.com/en/products/proxmox-virtual-environment/overview>
- Salameh, J. (2023). *VMware: A Quantitative Look at its Impact on Business Efficiency and Innovation*. Retrieved from Linked-In: <https://www.linkedin.com/pulse/vmware-quantitative-look-its-impact-business/>
- Shaw, K. (2023). *Network World*. Retrieved from What is a virtual machine, and why are they so useful?: <https://www.networkworld.com/article/969185/what-is-a-virtual-machine-and-why-are-they-so-useful.html>
- Vinod, N. (2023). *Linked-In*. Retrieved from The Benefits of Using Virtualization Technologies in the Education Sector: <https://www.linkedin.com/pulse/benefits-using-virtualization-technologies-education-sector-vinod-sbyxc/>