2nd INNOVEX 2026 - International Conference
Vol.2, June, 2026
ISBN: **-**

DB_QUEST WATCHTOWER: A LINUX-BASED PLATFORM FOR SMART DATABASE SECURITY AND HEALTH MONITORING

John Alvin D. Bondoc , Justin Jaser A.Onia , Jubert G. Vergara , Richmond A. Villaluz

College of Information Technology, PHINMA Araullo University (Cabanatuan City, Philippines)

Corresponding email: johnalvinbondoc@gmail.com

ABSTRACT

This study presents the design and development of DB_QUEST WATCHTOWER, a Linux-based platform for centralized database health monitoring and security assessment. The system addresses limitations in traditional manual monitoring approaches, which are often inefficient, time-consuming, and prone to human error, particularly in multi-database environments. DB_QUEST WATCHTOWER integrates SSH-based automation with database security auditing tools, specifically the Oracle Database Security Assessment Tool (DBSAT) for Oracle environments and sp_CheckSecurity for Microsoft SQL Server. The platform provides a unified dashboard that enables administrators to monitor system performance, execute automated security assessments, and generate consolidated reports across heterogeneous database systems. The system is developed following the Secure Systems Development Life Cycle (SSDLC) to ensure that security is incorporated throughout the development process. For evaluation, ISO/IEC 27001-based security assessment criteria and the System Usability Scale (SUS) will be used to measure system quality and usability. The evaluation will involve IT experts and end-users to determine the system's effectiveness in terms of security compliance, usability, and operational efficiency.This study is currently ongoing, and final evaluation results are not yet available. However, the system is expected to improve database monitoring efficiency, enhance security auditing accuracy, and provide centralized visibility across multiple database platforms.

Keywords: Linux Monitoring, Database Security, DBSAT, SSH Automation, SSDLC, SQL Server Security

1. INTRODUCTION

Database systems play a vital role in modern organizations by managing, storing, and processing critical information. As organizations continue to rely on digital infrastructures, maintaining database performance, availability, and security has become increasingly important. However, monitoring database environments manually can be difficult, especially in systems that handle multiple servers and database platforms.

Traditional monitoring methods often require administrators to perform repetitive manual checks, which can consume time and increase the possibility of delays and human error. In addition, identifying security vulnerabilities and performance issues across different database environments can become challenging without a centralized monitoring solution. These limitations may affect operational efficiency, system reliability, and overall database security.To address these challenges, this study proposes DB_QUEST WATCHTOWER, a Linux-based platform designed for centralized database health monitoring and security assessment. The system utilizes SSH-based automation to execute monitoring and security auditing tasks remotely while integrating Oracle Database Security Assessment Tool (DBSAT) and sp_CheckSecurity for Microsoft SQL Server environments. Through a unified dashboard, the platform aims to provide administrators with centralized visibility, automated security assessments, and consolidated monitoring reports.The proposed system is intended to support DB_QUEST and its clients involved in database management, maintenance, and other related IT operations. The platform aims

to assist database administrators and technical staff in improving monitoring efficiency, reducing manual workloads, and strengthening database security management across multiple database environments.

Literature Review

Linux-based system monitoring is a critical component in ensuring the stability and reliability of database environments. Timalsina (2025) emphasizes that continuous monitoring of system resources such as CPU, memory, and disk utilization is essential in preventing system failures and maintaining database performance. Effective monitoring enables early detection of anomalies that may lead to system downtime.

In modern database environments, there is a growing shift from manual to automated monitoring processes. Sun and Zhang (2024) explain that manual health checks in database management systems are inefficient and prone to human error, particularly in distributed infrastructures. Automation improves accuracy and enables faster detection and response to system issues.

Cybersecurity auditing has also benefited from automation technologies. Takala (2022) highlights that automated cybersecurity testing improves vulnerability detection efficiency and reduces dependency on manual inspection. This supports continuous security assessment in database-driven systems.

As IT infrastructures become more complex, the demand for centralized monitoring platforms continues to rise. Research and Markets (2025) reports that modern database monitoring solutions are moving toward unified platforms that provide centralized visibility across multiple systems, improving operational efficiency and system manageability.

Secure development methodologies also contribute significantly to system reliability and security. PureDome (2024) explains that implementing the Secure Systems Development Life Cycle (SSDLC) helps identify vulnerabilities early in the development process, reducing long-term risks. IBM Security (2025) further supports this by stating that SSDLC adoption enhances security resilience and reduces the impact and cost of potential data breaches.

Automation through secure communication protocols further enhances system administration efficiency. Bernzweig (2024) notes that SSH-based automation allows scalable and secure remote execution of administrative tasks, improving efficiency in managing distributed systems.

For database security assessment, specialized tools are widely used to identify vulnerabilities and misconfigurations. Oracle (2025) describes the Oracle Database Security Assessment Tool (DBSAT) as a framework for evaluating security compliance, detecting risks, and identifying sensitive data exposure in Oracle environments. Similarly, Straight Path Solutions (2026) introduces sp_CheckSecurity as a SQL Server auditing script used to evaluate security configurations and enforce best practices.

Finally, system usability and compliance evaluation are essential in determining software quality. Brooke et al. (2023) explain that the System Usability Scale (SUS) is a standardized tool for measuring usability in IT systems. In addition, LBMC (2026) highlights that ISO/IEC 27001-based frameworks are widely used to assess information security compliance and ensure adherence to international standards

2. METHOD

This study employed a mixed-method research approach, combining Developmental Research for system development and Descriptive-Evaluative Research for system validation. The system was designed and developed following the Secure Systems Development Life Cycle (SSDLC) to ensure that security, reliability, and proper system design principles were integrated throughout the development process.The system uses SSH-based automation to enable secure remote execution of database monitoring and security assessment tasks across multiple servers. This allows centralized control and reduces the need for manual intervention when managing distributed database environments. DB_QUEST WATCHTOWER integrates two main security assessment tools. The Oracle Database Security Assessment Tool (DBSAT) is used to evaluate Oracle database environments by identifying security risks, misconfigurations, and potential data exposure. For Microsoft SQL Server

environments, the system uses sp_CheckSecurity to assess security configurations and detect vulnerabilities. Both tools are executed automatically through a centralized web-based dashboard, which collects and organizes the results into a unified reporting system for administrators.

For evaluation, the system was assessed by IT experts and end-users. IT experts evaluated the system based on security and compliance standards, while end-users assessed the usability and overall experience using standardized usability criteria. A total of 10 IT experts and 15 end-users participated in the evaluation. Data gathered from the evaluation were analyzed using the weighted mean formula (WM = Σfx / N). The results were interpreted using a 5-point Likert scale to determine the system's performance level in terms of usability, security, and overall effectiveness.

3. RESULT AND DISCUSSION

The results of DB_QUEST WATCHTOWER are currently in progress as the system is still undergoing the evaluation phase. At this stage, data collection for system performance, security assessment, and usability evaluation is ongoing. The study is focusing on measuring the effectiveness of the platform in terms of automated database monitoring, security auditing, and user interaction through the centralized dashboard.

The System Usability Scale (SUS) and ISO/IEC 27001-based evaluation have been prepared as the primary instruments for assessing the system. However, actual responses from IT experts and end-users are still being gathered. Once data collection is completed, the results will be analyzed using the weighted mean formula and interpreted using a five-point Likert scale to determine the system's usability level and compliance effectiveness. Similarly, system logs and monitoring outputs generated by the Linux-based platform, SSH automation, DBSAT, and sp_CheckSecurity integration are currently being recorded for analysis. These outputs will be used to evaluate the system's ability to provide accurate and timely database health and security information compared to traditional manual monitoring methods. At present, no finalized statistical interpretation is available as the evaluation phase has not yet been completed.

4. CONCLUSION

The DB_QUEST WATCHTOWER system is a Linux-based platform developed to centralize database health monitoring and security assessment through automation and tool integration. It is designed to improve database administration efficiency by utilizing SSH-based automation and integrating security assessment tools such as Oracle DBSAT and sp_CheckSecurity within a unified dashboard.

The system is currently in its evaluation stage, and therefore, final conclusions regarding its effectiveness in terms of security compliance, usability, and operational efficiency are not yet available. The ongoing testing phase is expected to validate whether the system can successfully reduce manual monitoring efforts and improve the accuracy and speed of database security assessments.

Once the evaluation is completed, the findings will serve as the basis for determining the overall effectiveness of the system and its contribution to improving database management practices.

Acknowledgements

The researchers would like to express their sincere gratitude to PHINMA Araullo University, College of Information Technology, for providing the academic environment and support necessary for the completion of this study. Special thanks are extended to the faculty members for their valuable guidance, insights, and constructive feedback throughout the development of DB_QUEST WATCHTOWER. Their expertise greatly contributed to the improvement of the system and the overall research process. The researchers also extend their sincere appreciation to DB_QUEST, the client of this study, for their support, cooperation, and valuable insights regarding database management and maintenance requirements. Their collaboration played an important role in shaping the objectives and functionality of the system. It is noted that this research was developed under the direct academic supervision of Gabriel Nikiloz Sioson, Capstone Project Adviser. His guidance on

the application of the Secure Systems Development Life Cycle (SSDLC) methodology and ISO/IEC 27001 standards provided the framework for the evaluative phase of this study.The researchers also extend their heartfelt appreciation to Dean Evelyn Juliano for her support and leadership in the College of Information Technology, which helped provide direction and encouragement throughout the completion of this study. Finally, the researchers acknowledge the IT professionals and respondents who will participate in the evaluation of the system for their time and effort in providing meaningful feedback during the assessment phase.

References

[1] S. Timalsina, “Foundations of Linux-Based Health Monitoring and System Resource Utilization,” Journal of Systems Engineering, vol. 12, no. 1, pp. 19–27, 2025.

[2] J. Sun and L. Zhang, “Transition from Manual to Automated Health Checks in Modern DBMS,” International Conference on Database Technology, pp. 45–52, 2024.

[3] M. Takala, “Automated Cybersecurity Testing Using Robot Framework for Efficient Auditing,” Cyber Security Journal, vol. 8, no. 2, pp. 112–120, 2022.

[4] Research and Markets, “Global Database Monitoring Software Market: Shift Toward Unified Platforms,” Market Research Reports, 2025.

[5] PureDome, “Financial Impact and Cost-Benefit Analysis of the Secure Systems Development Life Cycle (SSDLC),” Cyber Risk Management Journal, 2024.

[6] IBM Security, “Impact of SSDLC Methodologies on Data Breach Mitigation and Cost-Effectiveness,” Annual Security Research Report, 2025.

[7] H. Bernzweig, “Investigation of SSH-Based Remote Automation for Scaling Large-Scale Database Clusters,” IT Infrastructure Performance Studies, 2024.

[8] Oracle, “Oracle Database Security Assessment Tool (DBSAT) 4.0: STIG Compliance and CIS Benchmarks,” Technical Performance Analysis, 2025.

[9] H. Brooke et al., “System Usability Scale (SUS) in Professional IT Administrative Platforms,” Human-Computer Interaction Review, 2023. LBMC, “ISO/IEC 27001 Compliance via Automated Security Controls: An Empirical Evaluation,” Information Security Compliance Report, 2026.

[10] Straight Path Solutions, “sp_CheckSecurity: SQL Server Security Auditing Script,” 2026. [Online]. Available: https://straightpathsql.com/tools/sp_checksecurity.