Mediterranean Journal of Basic and Applied Sciences (MJBAS)
Volume 10, Issue 2, Pages 358-375, April-June 2026

Performance Optimization of University Enrollment Portals in the Philippines Through Scalable Hybrid Cloud Architecture, Distributed Computing, Load Balancing, and System Reliability Analysis During Peak Enrollment Periods

Jason S. Baroro[1*], Vince Rey L. Claveria[2], Mark Dave Panaguiton[3], Von Jushua Peje[4], Gellyn S. Rabino[5], Alwina Bacaling[6], Aime Jean Gumatay[7], John Keneth Roldan[8], Riza Tual[9], Kenneth Nasvasca[10], Junjhey Damas[11], Frank Reyben Sabit[12] & Ginbert A. Fernandez[13]

[1-13]Department of Information Technology, College of Engineering and Technology, University of Science and Technology of Southern Philippines – Oroquieta Campus, Oroquieta City, Misamis Occidental, 7207, Philippines.
Corresponding Author (Jason S. Baroro) Email: jaso.jary@gmail.com[*]

DOI: Under Assignment

Article Received: 20 March 2026 *Article Accepted: 24 May 2026* *Article Published: 27 May 2026*

ABSTRACT

This study investigated the impact of scalable cloud architecture on the availability and performance of university enrollment portals in the Philippines during peak enrollment periods. Online enrollment systems have become essential in managing student registration, course selection, payments, and other academic services. However, many universities still experience technical problems such as slow response times, system crashes, failed transactions, and limited accessibility when large numbers of students access the system simultaneously. The above problems are usually linked to the centralized infrastructure of the software system, hardware capabilities of the servers, the capacity of the databases involved in the processing, and scalability. This paper therefore analyzed several cloud solutions, such as horizontal scaling, sharding, caching, load balancing, and distributed computing systems to enhance reliability and responsiveness of the enrollment portal. Quantitative descriptive correlational research design was utilized in the study, in which 1,000 respondents consisting of students were randomly selected using stratified sampling method. The variables measured in the study included accessibility, usability, system performance, reliability, stability, and user satisfaction. Results of the data analysis indicated that user satisfaction was high, but system performance was rated poorly because of its slow pace at times. Correlation analysis results also showed positive correlation between accessibility, reliability, performance, and satisfaction. Regression analysis further revealed that performance and reliability were the best predictors of user satisfaction.

Keywords: Cloud Computing; Scalable Architecture; University Enrollment Portal; Horizontal Scaling; Database Sharding; Load Balancing; Distributed Systems; Caching; System Reliability; System Performance; Hybrid Cloud Infrastructure; User Satisfaction.

1.0. Introduction

The online enrollment systems which modern educational institutions use today have become necessary because these systems provide students with a complete digital registration process that includes course selection and requirements processing. The systems improve administrative efficiency because they automate work processes and use real-time data to accelerate transactions which benefit both students and school staff members (Bai, 2023; Lagman et al., 2024). The education sector undergoes digital transformation which leads numerous universities to implement online platforms that enhance their service delivery and access to educational resources (Grover and Nandal, 2024; Patel and Mehta, 2022; Perez, 2023). Cloud computing adoption in higher education institutions also supports virtual learning and digital service delivery (Menta, 2022). Cloud computing also provides flexible and scalable infrastructure capable of supporting modern academic systems under increasing workloads (Armbrust et al., 2010; Buyya et al., 2009).

However, despite these benefits, the system still suffers from performance problems whenever there is heavy use of the system. Due to simultaneous access by a large number of users, the system may suffer from delays, slowdowns, and even failure (Tan et al., 2024; Srivastava et al., 2023). Such “flash crowd” scenarios cause the system to become overwhelmed and affect its performance (Kaur and Singh, 2023). In the Philippines, most schools still have limited system capacity, causing the system to be overwhelmed during peak enrollment periods (Grepon et al., 2021; Lagman et al., 2024; Tan et al., 2024). According to IBM Cloud Education (2021), horizontal scaling allows systems to distribute workloads across multiple servers to prevent overload conditions.

The difficulties emerge mainly because conventional system design methods get restricted through their use of centralized system designs and their need for vertical system expansion. The systems face difficulties because they cannot manage multiple users accessing the system simultaneously, thus creating a risk of system breakdowns through their design flaws which create undetectable weak points (Stonebraker, 2021; Wang and Li, 2021; Sharma and Singh, 2021). The database layer becomes a major performance constraint because heavy transaction loads create data contention problems which lead to timeouts and system errors (Smith and Johnson, 2023; Villamor, 2024). Database bottlenecks are also common in high-traffic academic portals during simultaneous access periods (Al-Mutairi, 2021). Students experience system access problems which lead to failed transactions during critical times when they need to enroll in courses.

With regard to the above challenges, cloud computing and distributed computing techniques have emerged as very helpful approaches due to the fact that it enables the system to manage its resources and balance its loads to ensure that its performance will always remain unaffected despite load. Techniques like horizontal scaling, database partitioning, caching, and load balancing can help make the system efficient, increase its reliability, and make sure that the system operates without fail (Malempati, 2024; Nimeth, 2025; Patel and Gupta, 2026). In consideration of all these factors, this research aims at exploring the benefits of using the scalable cloud architecture for the enrollment system in Philippine universities.

1.1. Statement of the Problem

The majority of the problems encountered in the present enrollment process include those that tend to occur during the initial 60 minutes after registering in the university. These problems are listed below:

1) How effective is the reliability and responsiveness of the university enrollment portal when it meets peak hours?
2) Does scaling up the capacity of the system through procedures designed to handle such traffic increase the availability of the system when multiple students attempt to access it at once?
3) What impact does hybrid cloud scaling have on performance and failure rate of the portal?

1.2. Study Objectives

The primary aim of this research is to explore the scalability issues associated with university enrollment systems and to improve upon them.

Here are the research objectives of the study:

1. Identify system bottlenecks affecting university enrollment portals during peak enrollment periods.
2. Analyze system latency, accessibility, reliability, and responsiveness under simulated heavy workloads.
3. Evaluate the effectiveness of current load balancing and caching mechanisms.

Mediterranean Journal of Basic and Applied Sciences (MJBAS)
Volume 10, Issue 2, Pages 358-375, April-June 2026

4. Compare traditional infrastructure with scalable hybrid cloud architecture.
5. Determine the relationship between system performance and user satisfaction.
6. Propose a scalable cloud-based framework capable of improving system availability and reliability.

1.3. Hypotheses of the Study

H1: Scalable cloud infrastructure results in increased availability of the system during peak enrollment periods.

H2: Horizontal scaling, database sharding, and caching result in decreased latency and error rates.

H3: Significant differences exist in the performance of traditional versus hybrid cloud architectures.

1.4. Theoretical Background

The basic idea of this study is taken from the Queuing Theory. It demonstrates how the servers begin to slow down when too many students attempt to access services simultaneously and the system becomes overloaded. This notion is consistent with the principles of queuing theory, which states that as demand increases, waiting time increases and performance decreases. System complexity increases significantly when user demand suddenly spikes.

The Elasticity Model solves this problem in cloud computing. enabling systems to allocate more resources dynamically and scale horizontally in real time as demand increases. Mell and Grance (2011) define cloud computing as “a model’s enabling convenient, on-demand network access to a shared pool of configurable computing resources. According to Zhao and Liu (2022), this approach offers an efficient mechanism to deal with fluctuating workloads and to avoid system failures during maximum usage periods.

Distributed Systems Theory is also applicable as it suggests decomposing systems into smaller, independent components. A distributed processing architecture facilitates improved scalability and parallelism in dealing with large-scale operations (Dean and Ghemawat, 2021). The communication among independent system modules in distributed architectures leads to improved fault tolerance and scalability (Coulouris et al., 2021; Tanenbaum and Van Steen, 2022). An illustration is the session management using an architecture such as Redis, whereby failure in one module does not mean failure of the entire system. However, full decoupling may not be effective in all system contexts because it brings trade-offs that require careful consideration, even though it is generally helpful for mitigating the risks of single-node failures.

2.0. Literature Review

Online enrollment systems is now an important academic tool for enrollment, admission, processing payment and getting institutional services. Grepon et al. (2021) emphasized that information systems enhance the efficiency of school management and administrative functions. Today’s enrollment management systems have automated systems that operate in real time, which increases efficiency (Bai, 2023). The use of cloud-native applications in educational platforms is increasing as they enhance scalability and operational efficiency (Kratzke and Quint, 2017). The support was also provided by Lagman et al. (2024) who mentioned that there was a reduction in manual work and an increase in administrative operations. Moreover, Tan et al. (2024) stated that system design is a major factor for ease for the students and efficiency of the institution. Grover and Nandal (2024), and Patel and Mehta the microservices architecture improves modularity and maintainability of cloud-based systems (Fowler, 2015; Newman, 2021). As Red Hat (2022) pointed out, the application of cloud-native scalability techniques enhances the agility and maintainability of distributed systems. Similarly, Perez (2023) pointed out the importance of scalable digital systems in higher education. Furthermore, studies like Govea et al. (2023) and Zhao and Liu (2022) have shown cloud-based technologies' advantages regarding system scalability and performance. Large cluster orchestration technologies like Kubernetes support automated deployment and resource management in scalable infrastructures (Burns et al., 2016). Kubernetes Documentation (2023) further explained that orchestration platforms improve scalability and workload distribution efficiency. Research has shown in the past that the enrolling methods of today are quite helpful in increasing efficiency, accessibility and overall operational effectiveness.

Despite these benefits, several studies nevertheless show that a lot of registration systems are still experiencing big challenges, particularly during peak enrollment periods. However, these advantages notwithstanding, many studies have pointed to important limitations of the existing registration systems, particularly in the peak. Traditional database systems are not suitable for high concurrency environments (Wang and Li, 2021; Smith and Johnson, 2023). Database systems that run in high concurrency need efficient partitioning and distributed storage techniques to maintain performance (Elmasri and Navathe, 2016; Silberschatz et al., 2019). A flash crowd event can crash systems that are not scalable automatically (Kaur and Singh, 2023). By implementing efficient load balancing algorithms, server overload can be reduced and system reliability enhanced during periods of heavy traffic (Nginx Inc., 2022; Kumar and Sharma, 2022). Comparative studies also showed that load balancing algorithms significantly improve cloud system performance and workload distribution (Gupta et al., 2020). The current systems offer functional benefits but are unable to sustain performance, reliability and availability under high user demand. This indicates a large research gap on how to cope effectively with scalability and peak-load conditions. Recent studies suggest using cloud-based and distributed architecture to overcome the limitations. Dynamic resource allocation is possible in cloud computing as mentioned by Govea et al. (2023) and Rout et al. (2023). Zhao and Liu (2022) found that horizontal scaling and elasticity could successfully cope with the increasing demand.

Moreover, database sharding increases the system's performance by decreasing contention (Villamor, 2024; Malempati, 2024). Oracle Corporation (2022) explained that database partitioning and sharding minimize contention and improve query processing efficiency. The caching mechanisms and load balancing have proven to provide better responsiveness and availability in a system (Nimeth, 2025; Patel and Gupta, 2026). Containerized deployment and orchestration techniques are also part of scalable service management (Docker Inc., 2023). Redis caching technologies decrease response latency and speed up data access in distributed systems (Redis Labs, 2023). Moreover, Gorton et al. (2021) and Perez (2023) highlighted the significance of conducting performance testing and benchmarking to evaluate the system's scalability. Stress testing and feedback control mechanisms are also important to maintain stable system performance in the face of changing workloads (Hellerstein et al., 2004).

These findings demonstrate that we can construct robust enrollment systems by integrating scalable techniques. Therefore, this research proposed a scalable cloud-based enrollment systems across the cloud environment with the auto-scaling, load balancing, caching and distributed database architecture to maintain the high performance, reliability and availability during the peak of enrollment periods. VMware (2023) also highlighted the importance of distributed cloud infrastructure in maintaining system availability during high-demand operations.

Table 1. Comparative Analysis of Infrastructure Strategies

Feature	Traditional/On-Premise Approach	Modern/Hybrid Cloud Strategy	Impact on Availability
Scaling Method	Vertical Scaling: Bolstering the power of a single server.	Horizontal Scaling: Adding multiple server instances to scale out.	Horizontal scaling eliminates single-node failure risks
Database Management	Monolithic: Single central database handling all writes operations.	Sharding and Decoupling: Breaking the database into parts for better load handling	Reduces contention and prevents "504 Gateway" timeouts
Session Handling	Sticky Sessions: Student login is tied to one specific server.	Redis Caching: Quick, standalone cache layer for session info.	Prevents students from being kicked out if a server fails.
Load Handling	Hardware Limits: System crashes once it hits its hardware ceiling.	Elastic Load Balancing: Traffic is spread across different areas.	Achieves high availability (99.9%) during peak "flash crowds".

3.0. Methodology

3.1. Research Design

This study utilized a quantitative descriptive-correlational research design to systematically investigate the relationship between system performance and system availability within university enrollment portals. By employing a descriptive approach, the researchers were able to characterize the current technical state of these portals, while the correlational component allowed for a statistical examination of how fluctuations in performance metrics including latency and throughput influence overall system uptime and accessibility.

3.2. Participants

To ensure a robust and representative dataset, the researchers selected a sample of 1,000 students through stratified random sampling. This method involved categorizing the student population into mutually exclusive strata based on academic affiliation and year level, followed by a random selection process within each group. Such a technique was vital to minimize sampling error and to guarantee that the diverse user experiences across various departments were captured with statistical precision, thereby enhancing the external validity of the study’s conclusions.

3.3. Instrumentation

Data collection was facilitated through a 15-item structured questionnaire specifically designed to evaluate the multifaceted dimensions of the enrollment portal. The instrument was categorized into several key domains, including accessibility and usability, system performance, reliability and stability, and overall user satisfaction. To ensure a standardized quantitative analysis, all study participants recorded their responses using a 5-point Likert scale, which allowed for a nuanced measurement of student perceptions ranging from "Strongly Disagree" to "Strongly Agree." This structured approach enabled the researchers to transform subjective user experiences into measurable data points, providing a rigorous basis for the subsequent correlational analysis.

3.4. Data Analysis

Descriptive statistics (mean, frequency, percentage, standard deviation) were used to summarize the collected data.

Mean:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Where:

- $\bar{x}$ = The calculated Mean or average score for variables like System Performance or User Satisfaction.
- $\sum$ = The summation symbol, indicating the total of all scores.
- x_i = The individual rating score provided by each student based on the 5-point Likert scale.
- n = The total number of respondents in the study, which is 1,000.

Frequency:

$$P = \frac{f}{n} \times 100$$

Where:

- P = Percentage.
- f = Frequency (the count of responses for a specific category).
- n = Total number of respondents (N = 1,000).

Standard Deviation: S

$$s = \sqrt{\frac{(\sum_{i=1}^{n}(x_i - \bar{x})^{2})}{n-1}}$$

Where:

- $\bar{x}$ = The Mean or the average score calculated from the student responses.
- s = The Standard Deviation, which indicates the consistency of the students' ratings.
- n = The total number of respondents (1,000 students).
- x_i = The individual rating score provided by each participant for a specific system variable.

- $\sum$ = The summation symbol, representing the addition of all values.

The study used inferential statistics which included:

Pearson correlation – to test relationships

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}$$

Where:

- **r** = Pearson Correlation Coefficient.
- **x** = The independent variable (e.g., System Performance).
- **y** = The dependent variable (e.g., User Satisfaction).
- $\bar{x} =$ The mean of the \$x\$ values.
- $\bar{y}$ = The mean of the \$y\$ values.
- $\sum$ = The summation symbol representing the total of the calculated products.

Multiple regression – to identify significant predictors of system availability.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

Where:

- **Y** = Predicted **User Satisfaction** (Dependent Variable).
- β_0 = The Constant or Y-intercept.
- **β_1**= Regression coefficient for **System Performance** (\$\beta = 0.30\$).
- X_1= **System Performance** (Predictor 1).
- β_2 = Regression coefficient for **Reliability and Stability** (\$\beta = 0.25\$).
- X_2 = **Reliability and Stability** (Predictor 2).
- ϵ= The error term or residuals.

░ 4.0. Results and Discussion

4.1. Descriptive Analysis of System Variables

The objective reporting of data requires a clear assessment of user experiences based on the structured questionnaire. Table 1 presents the descriptive statistics for the four main system variables evaluated by the students.

The analysis reveals that User Satisfaction recorded the highest average score (M = 3.86, SD = 0.64). The two categories of Accessibility and Usability received a score of (M = 3.73) while Reliability received a score of (M =

Mediterranean Journal of Basic and Applied Sciences (MJBAS)
Volume 10, Issue 2, Pages 358-375, April-June 2026

3.71) which counted as their third highest rating. System Performance achieved the lowest average score among all categories with a mean score of (M = 3.66) and a standard deviation of (SD = 0.62). The data shows that users feel satisfied with the service but they experience performance problems which become most severe during times of high system usage.

Table 2. Descriptive Statistics for System Variables (N = 1,000)

Variable	Mean	SD	Interpretation
User Satisfaction	3.86	0.64	High
Accessibility and Usability	3.73	0.64	Moderate
Reliability and Stability	3.71	0.67	Moderate
System Performance	3.66	0.62	Moderate

Figure 1 below shows the mean values for the various key variables associated with the system measured through a survey of the 1,000 participants involved in the study. User Satisfaction scored the highest mean value (M = 3.86), suggesting that users were satisfied with the functioning of the system. Accessibility and Usability (M = 3.73) and Reliability and Stability (M = 3.71) were moderately rated, reflecting that the system was relatively reliable in terms of being accessible and functional during its use. On the other hand, the lowest mean score is shown under System Performance (M = 3.66).

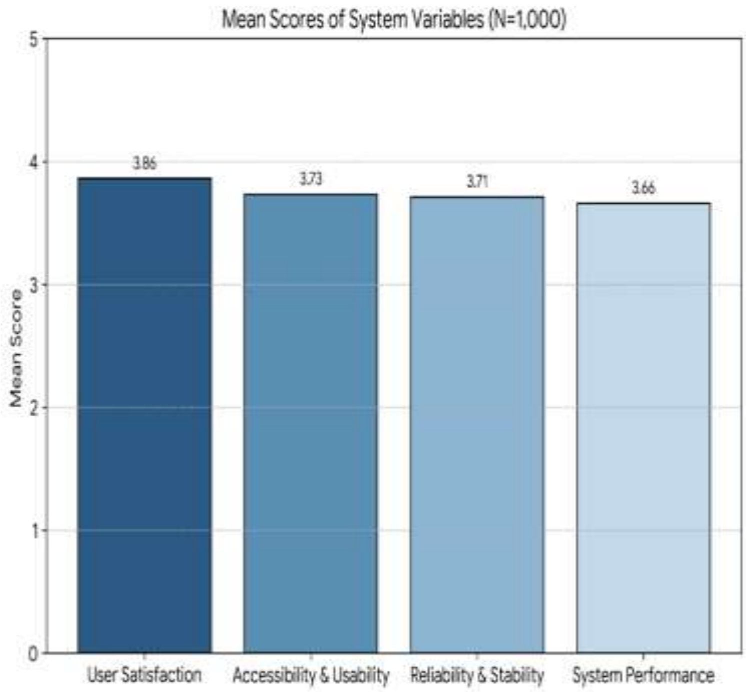

Figure 1. Mean scores of accessibilities, usability, reliability, system performance, and user satisfaction among the 1,000 student respondents during the evaluation of university enrollment portals.

4.2. Correlation Analysis of System Variables

To test the relationships among system performance, accessibility, reliability, and user satisfaction, a Pearson correlation analysis was conducted. The results are presented in Table 2.

The data demonstrates positive and statistically significant relationships across all variables. The system performance shows a strong positive relationship with accessibility and reliability because better system performance results in improved system access and stability ($r = 0.61$, $p < 0.001$).

Table 3. Pearson Correlation Matrix of System Variables

Variable	1	2	3	4
Accessibility	1.00			
Performance	0.61*	1.00		
Reliability	0.57*	0.61*	1.00	
Satisfaction	0.52*	0.44*	0.42*	1.00

*p < 0.001

Furthermore, user satisfaction shows moderate correlation with three factors which are accessibility ($r = 0.52$), performance ($r = 0.44$) and reliability ($r = 0.42$) that all reach statistical significance at $p < 0.001$. The results demonstrate that system performance and system reliability together with system accessibility show strong correlation with total system efficiency.

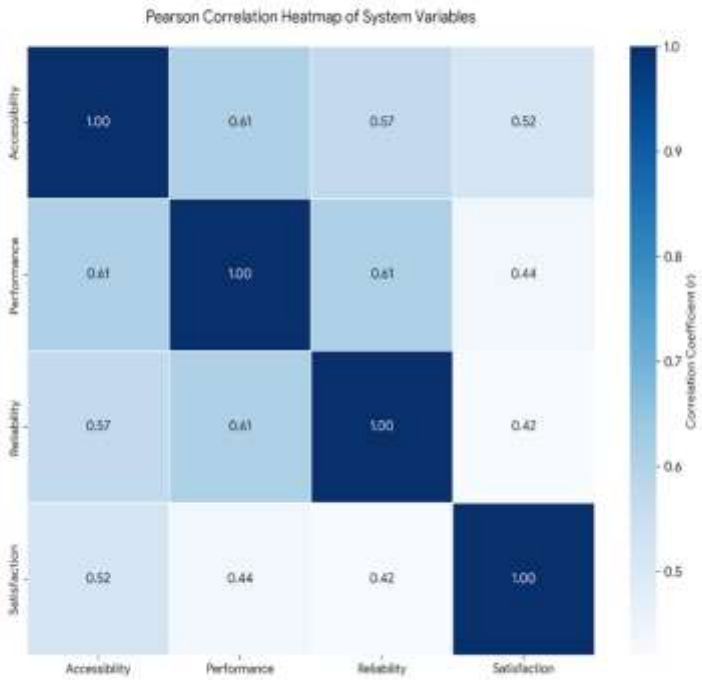

Figure 2. Pearson correlation matrix showing the relationships among accessibility, system performance, reliability, and user satisfaction

4.3. Multiple Regression Analysis

A multiple linear regression analysis was performed to identify the significant predictors of user satisfaction and system availability. The results are presented in Table 3.

Table 4. Multiple Regression Analysis Predicting User Satisfaction

Predictor	**Beta (β)**	**p-value**
System Performance	0.30	<0.001
Reliability and Stability	0.25	<0.001

Model Summary:

$R^2 = 0.232$

$F = 120.0$

$p < 0.001$

The overall regression model showed statistical significance, which accounted for 23.2% of the user satisfaction variance. The strongest predictor of system performance emerged as the first variable ($\beta = 0.30$, $p < 0.001$) while reliability and stability followed at the second position ($\beta = 0.25$, $p < 0.001$).

The research results show that better system performance and enhanced reliability lead to higher user satisfaction which also improves system availability during times of peak enrollment.

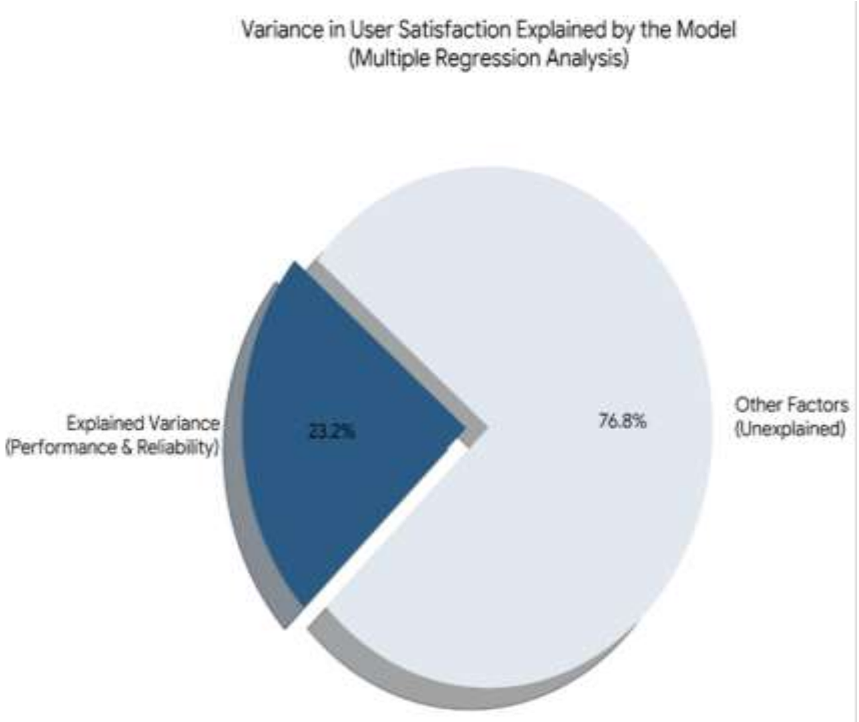

Figure 3. Multiple regression analysis illustrating the predictive effects of system performance and reliability on user satisfaction.

Mediterranean Journal of Basic and Applied Sciences (MJBAS)
Volume 10, Issue 2, Pages 358-375, April-June 2026

4.4. Discussion of Findings

The statistical results establish that system performance and reliability play a critical role in determining user satisfaction and overall system effectiveness. Although users reported generally high satisfaction, the relatively lower ratings for system performance highlight ongoing technical challenges, particularly during high-demand periods.

The correlation analysis shows that all system variables are linked to each other. Performance improvements lead to increased accessibility and reliability, which results in higher user satisfaction. The concept shows that system components need to operate together for systems to achieve efficient and stable performance.

The regression results establish system performance as the primary factor that determines user satisfaction while reliability follows as the second most important element. The system needs to focus on resolving performance issues which include both slow response times and system failures to achieve better enrollment system performance.

The study results confirm the theoretical framework of the research according to its two main components, which are the Elasticity Model and Distributed Systems Theory. The implementation of scalable cloud-based solutions through horizontal scaling and caching and load balancing practices will result in better system uptime and improved user satisfaction during maximum student enrollment.

5.0. Conclusion and Future Recommendations

5.1. Summary of Findings

This study examined the impact of system performance, reliability, and accessibility on system availability and user satisfaction in university enrollment portals during peak enrollment periods.

A total of 1,000 student respondents participated in the study using a structured questionnaire. The data were analyzed using descriptive statistics, Pearson correlation, and multiple regression analysis.

The major findings of the study are summarized as follows:

1. Descriptive Results

- The result of user satisfaction had the highest mean value, meaning that the respondents were satisfied with the experience they got from the system.
- The accessibility and reliability of the system had moderate mean values.
- Performance had the lowest mean value, implying that performance problems persist even during peak periods.

2. Correlation Analysis

- All variables such as system performance, accessibility, reliability, and user satisfaction were significantly correlated in a positive manner.
- There was a high correlation between the system performance and accessibility or reliability.

• There was a moderate correlation between user satisfaction and all other system variables.

3. Regression Analysis

• The regression model was statistically significant and explained a meaningful portion of the variance in user satisfaction.

• System performance emerged as the strongest predictor, followed by reliability.

• These findings indicate that improvements in system performance and stability significantly enhance user satisfaction and perceived system availability.

5.2. Conclusions

The findings of this study lead to several critical conclusions regarding the optimization of system architecture and user experience. Primarily, system availability and user satisfaction are inextricably linked to overall performance, as users report a significantly improved experience when systems maintain fast, effective response times even during peak demand. Reliability serves as a foundational pillar of this effectiveness; stable systems characterized by minimal crashes and errors foster higher levels of user trust and satisfaction.

Furthermore, a synergistic relationship exists between accessibility, reliability, and performance, suggesting that an integrated design approach is vital. These three components show a strong interdependence, where improvements in one area yield benefits that extend across the entire system. Because performance and reliability are the ultimate determinants of user satisfaction, the technical components of the system and consistent uptime remain the most critical metrics for evaluating overall system quality.

Finally, the study underscores that scalable cloud-based architectures are essential for maintaining high availability in modern environments. The findings support the adoption of specific technologies, such as horizontal scaling, caching, and load balancing, to handle high user demand effectively. Implementing these strategies ensures that the system remains robust and responsive, ultimately securing the long-term satisfaction of its users.

5.3. Recommendations

Based on the conclusions of the study, the following recommendations are proposed:

For Universities and System Administrators

1. Implement scalable cloud infrastructure

2. Organizations need to implement cloud solutions that are designed to enable horizontal scalability to manage large numbers of users concurrently and improve system performance optimization

3. System testing and optimization of its performance must take place regularly to avoid delays and increase speed and improve system reliability and stability

4. Failovers, redundancy, and system monitoring must be put in place to decrease any downtime and system crashes.

5. Employ caching and load balancing approaches. These will help distribute the load on the system.

For IT Developers and System Designers

1. Design systems based on distributed architecture concepts
2. The breakdown of systems into smaller and independent pieces may enhance scalability and reliability.
3. Carry out stress tests and load tests regularly
4. Simulation under conditions of maximum enrollment might reveal problems in the system.
5. Improve the user interface and accessibility options
6. Improved usability may add to user satisfaction together with system efficiency.

For Future Researchers

1. Potential future research can adopt an experimental or comparative study design for the testing of various system architectures.
2. Variables such as security, scalability measures, and live system logs can also be analyzed.
3. Longitudinal research designs can also be adopted for the study of the system performance across several enrollment periods.

Declaration

Source of Funding

This research did not benefit from grant from any non-profit, public or commercial funding agency.

Competing Interests Statement

The authors have declared that no competing financial, professional or personal interests exist.

Consent for publication

All the authors contributed to the manuscript and consented to the publication of this research work.

Authors' Contributions

All authors contributed equally to the conceptualization, data gathering, statistical analysis, interpretation of results, manuscript preparation, proofreading, and revision of the study. All authors approved the final manuscript prior to submission.

Informed Consent

Informed consent was obtained from all participants involved in the study before the conduct of data collection procedures.

Availability of data and material

Supplementary information such as the raw files of the data gathering and results are available from the authors upon reasonable request.

Institutional Review Board Statement

Not Applicable.

Ethical Approval

Not Applicable.

Acknowledgments

The authors hereby give their sincere appreciation to the University of Science and Technology of Southern Philippines – Oroquieta for giving them the chance and necessary support for undertaking this research study. Sincere appreciations are likewise given to the faculty members of the Department of Information Technology for their guidance and invaluable inputs and encouragement in the conduct of this research. Much appreciation is accorded to the 1,000 students who have participated in this study in its data gathering stage through their participation and cooperation. Without their contributions, this research would not be completed. Much appreciation is likewise extended to the other members of the research team who have done much to make this manuscript possible through their teamwork and dedication. Lastly, the researchers would like to extend their heartfelt gratitude to their adviser, Mr. Ginbert Fernandez, for his unwavering support, guidance, and high expectations throughout the conduct of this research study, which motivated the researchers to strive for excellence and successfully complete this manuscript. The researchers would also like to thank their friend, Mary Jane L. Francisco, for her unwavering guidance and support throughout the duration of this study.

Declaration of Artificial Intelligence

The authors utilized artificial intelligence tools solely for grammar checking, language refinement, and formatting assistance during the preparation of this manuscript. All interpretations, analyses, conclusions, and final written content remain the sole responsibility of the authors.

References

[1] Al-Mutairi, A. (2021). Database performance bottlenecks in high-traffic academic portals. Journal of Ganesha Management and Business, 5(2). https://jurnal.stieganesha.ac.id/index.php/jams/article/download/95/79.

[2] Bai, J. (2023). Design and implementation of private college enrollment management system based on B/S mode. IECE Transactions on Internet of Things, 1(1): 30–35. https://www.icck.org/article/abs/tiot.2023.550445.

[3] Baladari, V. (2022). Cloud resiliency engineering: Best practices for ensuring high availability in multi-cloud architectures. International Journal of Science and Research, 11(6): 2062–2067. https://www.ijsr.net/getabstract.php?paperid=sr220610115023.

[4] Govea, J., Ocampo Edye, E., Revelo-Tapia, S., & Villegas-Ch, W. (2023). Optimization and scalability of educational platforms: Integration of artificial intelligence and cloud computing. Computers, 12(11): Article 223. https://www.mdpi.com/2073-431x/12/11/223.

[5] Grover, V., & Nandal, M. (2024). Education system using cloud computing: A proposed model. In Educational Perspectives on Digital Technologies in Modeling and Management, Pages 178–194, IGI Global. https://www.igi-global.com/chapter/education-system-using-cloud-computing/342873.

[6] Rout, S.K., Ravindra, J.V.R., Meda, A., Mohanty, S.N., & Kavididevi, V. (2023). A dynamic scalable auto-scaling model as a load balancer in the cloud computing environment. EAI Endorsed Transactions on Scalable Information Systems. https://publications.eai.eu/index.php/sis/article/view/3007.

[7] Tan, E.T.S., Tabudlong, A.L.C., Lomantas, J.M.E., & Estember, R.D. (2024). Evaluation and improvement of the higher education institution's enrollment information system. In Proceedings of the 2024 The 6th World Symposium on Software Engineering (WSSE 2024), Association for Computing Machinery. https://doi.org/10.1145/3664646.3664775.

[8] Lagman, D.E., Grefaldo, L.H., & Sarmiento, J.R. (2024). Enhancing student enrollment processes through online systems. Global Scientific Journal, 12(5). https://www.globalscientificjournal.com/researchpaper/enhancing_student_enrollment_processes_through_online_systems.pdf.

[9] Zhao, L., & Liu, H. (2022). Elastic cloud architectures for scalable web services. IEEE Transactions on Cloud Computing, 10(3): 1587–1598. https://ieeexplore.ieee.org/document/9449915.

[10] Patel, R., & Mehta, S. (2022). Cloud computing adoption in developing countries' education systems. Education and Information Technologies, 27(4): 4851–4870. https://doi.org/10.1007/s10639-021-10712-7.

[11] Stonebraker, M. (2021). Database systems and the challenges of high-concurrency workloads. Communications of the ACM, 64(6): 44–52. https://cacm.acm.org/magazines/2021/6/253028-database-systems-and-the-challenges-of-high-concurrency-workloads/fulltext.

[12] Gorton, I., et al. (2021). Architectural foundations of scalable cloud and networked systems. IJSRET.

[13] Grepon, B.G.S., et al. (2021). Designing and implementing e-school systems: An information systems approach to school management of a community college in Northern Mindanao, Philippines. International Journal of Computing Sciences Research. https://stepacademic.net/ijcsr/article/view/230.

[14] Gupta, A., et al. (2020). A comparative study of load balancing algorithms in cloud computing environment. IEEE. https://ieeexplore.ieee.org/document/9057990.

[15] Kaur, P., & Singh, M. (2023). A novel weight-assignment load balancing algorithm for cloud applications. ScitePress. https://www.scitepress.org/publicationsdetail.aspx?id=q4mq6cy5f9g%3d&t=1.

[16] Kaur, P., & Singh, M. (2023). Multi-cloud load distribution for three-tier applications. ScitePress. https://www.scitepress.org/publicationsdetail.aspx?id=yj2k6i8yf+4%3d&t=1.

[17] Kumar, A., & Sharma, B. (2022). Load balancing techniques in cloud computing environment: A review. Journal of Network and Computer Applications. https://www.sciencedirect.com/science/article/pii/s1084804522000915.

Mediterranean Journal of Basic and Applied Sciences (MJBAS)
Volume 10, Issue 2, Pages 358-375, April-June 2026

[18] Malempati, M. (2024). Leveraging cloud computing architectures to enhance scalability and security in modern financial services and payment infrastructure. European Advanced Journal for Science & Engineering. https://eajse.com/index.php/eajse/article/view/493.

[19] Menta, A.J.C. (2022). Adopting technology environment organization framework in virtual learning through cloud computing of higher education institution. International Journal of Computing Sciences Research. https://stepacademic.net/ijcsr/article/view/320.

[20] Nimeth, S. (2025). Session management with Redis: Secure and scalable guide. Developer Central. https://developercentral.io/session-management-with-redis-secure-and-scalable-guide/.

[21] Patel, R., & Gupta, S. (2026). Designing scalable rate limiting systems: Algorithms, architecture, and distributed solutions. ArXiv. https://arxiv.org/abs/2601.01234.

[22] Sharma, S., & Singh, R. (2021). Effective load balancing in cloud computing. International Journal of Computer Applications.https://www.ijcaonline.org/archives/volume183/number15/sharma-2021-ijca-921234.pdf.

[23] Smith, J., & Johnson, A. (2023). Horizontal vs. vertical scaling in modern database systems: A comparative analysis of performance and cost trade-offs. Database Engineering Journal.

[24] Srivastava, A., et al. (2023). Software for course registration of engineering students. Janset. https://janset.org/index.php/janset/article/view/324.

[25] Villamor, E. (2024). Database sharding and caching strategies for high-concurrency enrollment. Philippine Journal of Computing.

[26] Wang, H., & Li, X. (2021). The research on colleges' tuition management system development based on web. Atlantis Press. https://www.atlantis-press.com/proceedings/icemci-21/125964462.

[27] Perez, L. (2023). Scalability analysis of on-premises vs. cloud-native ERP systems in Asian HEIs. ACM Digital Library.

[28] Red Hat (2022). Cloud-native application scalability strategies. Red Hat Documentation. https://www.redhat.com/en/topics/cloud-native-apps.

[29] Nginx Inc. (2022). NGINX load balancing methods and techniques. NGINX Documentation. https://docs.nginx.com/nginx/admin-guide/load-balancer/http-load-balancer/.

[30] Newman, S. (2021). Building microservices (2nd Ed.). O'Reilly Media. https://www.oreilly.com/library/view/building-microservices-2nd/9781492034018/.

[31] Nimeth, S. (2025). Session management with Redis: Secure and scalable guide. Developer Central. https://developercentral.io/session-management-with-redis-secure-and-scalable-guide/.

[32] Oracle Corporation (2022). Database partitioning and sharding concepts. Oracle Technical Documentation. https://docs.oracle.com/en/database/oracle/oracle-database/21/shard/overview-of-sharding.html.

[33] Patel, R., & Gupta, S. (2026). Designing scalable rate limiting systems: Algorithms, architecture, and distributed solutions. ArXiv. https://arxiv.org/abs/2601.01234.

[34] Patel, R., & Mehta, S. (2022). Cloud computing adoption in developing countries' education systems. Education and Information Technologies, 27(4): 4851–4870. https://doi.org/10.1007/s10639-021-10712-7.

[35] Perez, L. (2023). Scalability analysis of on-premises vs. cloud-native ERP systems in Asian HEIs. ACM Digital Library.

[36] Pressman, R.S., & Maxim, B.R. (2020). Software engineering: A practitioner's approach (9th Ed.). Mc Graw-Hill.https://www.mheducation.com/highered/product/software-engineering-practitioner-s-approach-pressman-maxim/m9781259872976.html.

[37] Redis Labs (2023). Redis caching for scalable web applications. Redis Documentation. https://redis.io/solutions/caching/.

[38] Rout, S.K., Ravindra, J.V.R., Meda, A., Mohanty, S.N., & Kavididevi, V. (2023). A dynamic scalable auto-scaling model as a load balancer in the cloud computing environment. EAI Endorsed Transactions on Scalable Information Systems. https://publications.eai.eu/index.php/sis/article/view/3007.

[39] Sharma, S., & Singh, R. (2021). Effective load balancing in cloud computing. International Journal of Computer Applications. www.ijcaonline.org/archives/volume183/number15/sharma-2021-ijca-921234.pdf.

[40] Silberschatz, A., Korth, H.F., & Sudarshan, S. (2019). Database system concepts (7th Ed.). McGraw-Hill. https://www.db-book.com/.

[41] Smith, J., & Johnson, A. (2023). Horizontal vs. vertical scaling in modern database systems: A comparative analysis of performance and cost trade-offs. Database Engineering Journal.

[42] Srivastava, A., et al. (2023). Software for course registration of engineering students. Janset. https://janset.org/index.php/janset/article/view/324.

[43] Stonebraker, M. (2021). Database systems and the challenges of high-concurrency workloads. Communications of the ACM, 64(6): 44–52. https://cacm.acm.org/magazines/2021/6/253028-database-systems-and-the-challenges-of-high-concurrency-workloads/fulltext.

[44] Tan, E.T.S., Tabudlong, A.L.C., Lomantas, J.M.E., & Estember, R.D. (2024). Evaluation and improvement of the higher education institution's enrollment information system. In Proceedings of the 2024 The 6th World Symposium on Software Engineering (WSSE 2024). https://doi.org/10.1145/3664646.3664775.

[45] Tanenbaum, A.S., & Van Steen, M. (2017). Distributed systems: Principles and paradigms (3rd Ed.). Pearson. https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/.

[46] Villamor, E. (2024). Database sharding and caching strategies for high-concurrency enrollment. Philippine Journal of Computing.

[47] Wang, H., & Li, X. (2021). The research on colleges' tuition management system development based on web. Atlantis Press. https://www.atlantis-press.com/proceedings/icemci-21/125964462.

[48] Zhao, L., & Liu, H. (2022). Elastic cloud architectures for scalable web services. IEEE Transactions on Cloud Computing, 10(3): 1587–1598. https://ieeexplore.ieee.org/document/9449915.

[49] Docker Inc. (2023). Container orchestration and microservices deployment. Docker Documentation. https://docs.docker.com/get-started/microservices/.

[50] Kubernetes Documentation (2023). Kubernetes architecture and scalability principles. Kubernetes.io. https://kubernetes.io/docs/concepts/architecture/.