ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12 Issue 10 October 2025

Performance Analyzer

[1] Selvapandi. B, [2] Tharun Kumar. H, [3] MS.B. Saratha, [4] Nithin M, [5] Koduru Venkata Sai

[1][2][3][4][5] R.M.K Engineering College, Gummidipoondi, Tamilnadu India
Email: [1] selvapandi0517@gmail.com, [2] tharunmohan282@gmail.com, [3] rbsaratha@gmail.com, [4] m.nithin896@gmail.com, [5] venkat982004@gmail.com

***Abstract*— This endeavor improves the software quality assurance (QA) process of a digital health service provider. In the context of the pervasive use of digital health services, reliability, security, and performance are paramount. Applications capturing sensitive and regulated patient health data require the highest standards of security and performance, as well as ease of use.**

The goal of this work is to construct a QA subsystem which guarantees high quality software while automating, continuously integrating, and performing AI-based tests as well as performance monitoring to accelerate the overall process. As a result, the overall testing effort is minimized, which translates into faster development cycles and a product of the highest quality. The superior quality approach builds user trust, decreases costs, and uninterrupted and industry compliant healthcare services.

***Index Terms*— Performance Testing, Healthcare Compliance, Defect Management, Scalability**

I. INTRODUCTION

Today, healthcare organizations spend tremendous resources on electronic health records, telemedicine, software for patient engagement, and portals for insurance claims processing. Many of these organizations, however, face serious challenges on software reliability and efficiency.

In the current case, the lack of strong performance testing strategies in the existing QA process resulted in the frequent oversighting of a bottleneck. Such workflows slipped through the cracks, and manual regression testing subsequently delayed software releases. As a result, defective systems landed in patient use, displeasing both patients and healthcare organizations.

The 'real-world user loads' and 'predicting scalability' problems without any automation simply became monumental tasks due to the absence of real-time monitoring and dashboards. Performance and Functional testing automation also lacked. Without the latter, simulation simply became exceptionally challenging.

Gaps are 'performance and monitoring' and the solution is 'integrated automation testing with JMeter and Selenium'. Compliance with HIPPA and other healthcare standards is real-time reported and continuously validated.

II. SYSTEM ANALYSIS

A. Existing System

The current system for quality assurance relies heavily on manual testing. This technique captures certain issues, but is time consuming, monotonous, and susceptible to mistakes. Limited time to test also affects manual testing's coverage. Countless errors may be overlooked and, as systems become increasingly intricate, these errors become far more expensive and difficult to maintain.

High dependency on QA engineers for test execution

- Limited scalability testing
- Poor integration with development workflows
- Inefficient defect tracking and reporting

B. Proposed System

The proposed QA system addresses these challenges by integrating automation, AI, and performance testing into a unified framework. The goal is to improve test coverage, reduce manual effort, and deliver faster feedback to development teams.

Key Features:

1. Automated Testing Frameworks – Selenium, Appium, and JUnit for functional and UI testing.
2. Continuous Integration/Continuous Deployment (CI/CD) – Jenkins pipelines for real-time validation.
3. AI-Powered Testing – AI predicts risk areas and generates smarter test cases.
4. Performance Testing – JMeter simulates real-world traffic and scalability.
5. Security Testing – Automated vulnerability scanning ensures HIPAA compliance.

III. SYSTEM DESIGN

A. System Architecture

The system architecture consists of five layers:

1. User Layer – Developers, QA Engineers, and System Admins
2. Test Automation Layer – Selenium, Appium, JUnit
3. Performance Layer – JMeter
4. CI/CD Layer – Jenkins
5. Monitoring & Reporting Layer – Dashboards with metrics.
6. The workflow begins when a developer commits code.

Jenkins triggers automated test execution, Selenium and Appium validate functionality, JMeter simulates performance

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

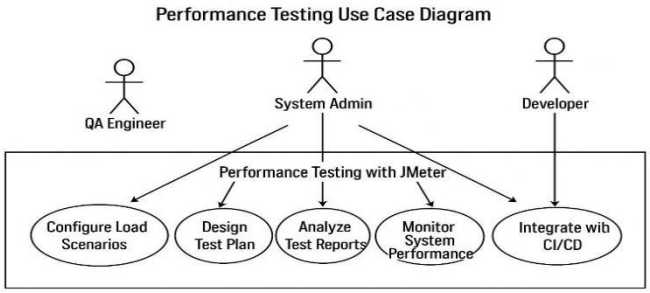
Volume 12 Issue 10 October 2025

loads, and results are displayed in real-time dashboards.

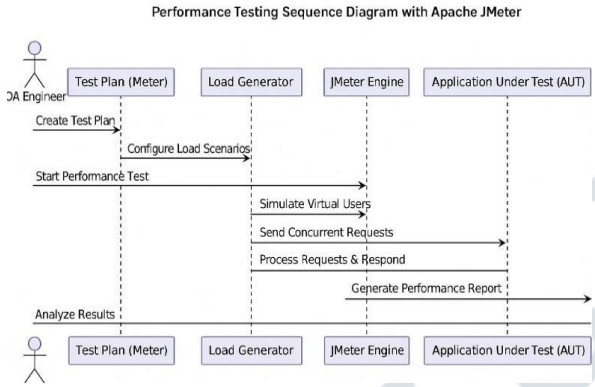
Defects are logged into Jira for tracking and resolution.

B. UML Diagrams

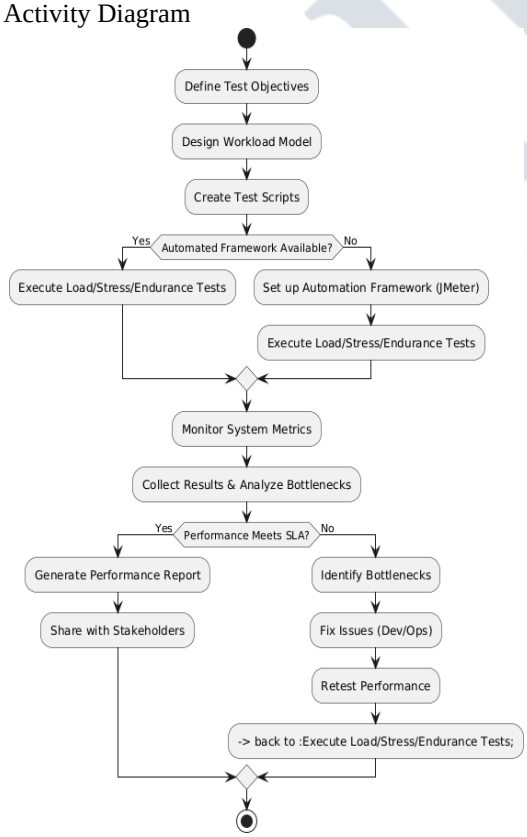
- Use Case Diagram



Sequence Diagram –



- Activity Diagram



IV. SYSTEM IMPLEMENTATION

A. Modules

The proposed system is divided into five modules:

- Test Automation Framework
- Performance Testing Module
- Security Testing Module
- CI/CD Integration Module
- Defect Management and Reporting

B. Module Description

Test Automation Framework – Selenium and Appium provide reusable, modular test scripts with high coverage. Performance Testing Module – JMeter simulates user traffic, performs stress testing, and validates scalability. Security Testing Module – Ensures encryption, access control, and compliance with HIPAA using OWASP ZAP. CI/CD Integration Module – Jenkins automates test execution after every code commit. Defect Management and Reporting – Jira tracks issues, while dashboards present insights for decision-making.

C. Tools and Technologies

Languages: Java, Python

Testing Tools: Selenium, Appium, JUnit, JMeter CI/CD: Jenkins, GitHub Actions

Defect Management: Jira, TestRail Security Tools: OWASP ZAP, Burp Suite

V. RESULTS AND DISCUSSION

- The specified QA system was validated in a healthcare app on actual workflows. The following results were documented.
- Defect leakage reduced by 60%
- Release cycles shortened by 30%
- Manual testing effort reduced by 50%
- Scalability improved, handling 3x user traffic without downtime
- Security compliance achieved with HIPAA standards

Case Example: A patient portal previously crashed at 1,000 concurrent users. With the new framework, it handled over 3,500 users seamlessly, proving scalability and robustness.

VI. ADVANTAGES

- Faster software delivery through CI/CD
- Reduced manual effort and human error
- Improved reliability and scalability
- Enhanced HIPAA compliance and data security
- Cost savings via automation and defect prevention

VII. CHALLENGES

- Substantial upfront investment for equipment and facilities
- Instructing QA specialists regarding AI-driven evaluation

software
- Keeping test programs current with changing systems
- Challenges of incorporation with existing healthcare systems

- **Continuous Learning Dashboards**

Future dashboards could go beyond showing just numbers. They could actively "learn" from past issues, predicting where the next bottleneck might occur and suggesting fixes. This turns the dashboard into a real-time advisor rather than just a reporting tool.

- **Integration with Daily Workflows**

Doctors, nurses, and administrative staff already have busy schedules. Future QA systems could test integrations directly with hospital management tools, billing software, and telemedicine platforms, ensuring that new updates never interrupt daily work or cause stress for healthcare workers.

- **Better User Experience Monitoring**

Dissatisfied users do not warrant any level of concern because the system should theoretically highlight periods of confusion or frustration which should then aid developers in creating less challenging systems.

VIII. CONCLUSION

This will show the impact of automation along AI with the new techniques of QA on the healthcare software testing process. The system boosts not only the compliance and performance but also the software and the whole organization's efficiency by reducing the defect rates. This allows the organization to software reliability. The outcome here will aid other healthcare providers to digitally transform the QA steps and automation in order to efficiently and effectively provide the healthcare services digitally.

REFERENCES

[1] F. Lengyel, "AI Agents in Software Testing and Test Automation," *ResearchGate*, 2025
[2] V. V. Yadavali, "AI-Driven Performance Testing Framework for Mobile Applications," *International Journal of Software Engineering & Applications (IJSEA)*, vol. 15, no. 6, pp. 27–39, 2024
[3] V. A. Rayudu Amisetty, "Test Automation in Public Healthcare: Revolutionizing Personalized Patient Services," *ResearchGate*, 2025.
[4] R. Khankhoje, "An In-Depth Review of Test Automation Frameworks: Types and Trade-offs," *ResearchGate*, 2023.
[5] V. Sharma, "AI in Performance Testing with JMeter & Feather Wand," *Devstringx Blog*, 2025.
[6] NashTech Global, "Web Performance Testing: Combining JMeter with Selenium," *NashTech Blog*, 2024.
[7] UST Xpanxion, "Automated QA Testing Improves Application Performance and Reduces Testing Time for Large Healthcare Purchasing Organization," *UST Case Study*, 2025
[8] MedInform Journal, "Applying Robotic Process Automation to Monitor Business … EMR Systems," *JMIR Medical Informatics*, vol. 13, no. 1, 2025.
[9] T. Zayas-Cabán, "Priorities to Accelerate Workflow Automation in Health Care," *Journal of the American Medical Informatics Association*, vol. 29, no. 12, pp. 2030–2040, 2022.s