

ORACLE BASED PHARMACY MANAGEMENT SOFTWARE FOR STOCK CONTROL, PRESCRIPTION VALIDATION, AND SUPPLIER COORDINATION

Lucia Fernandez
Spain

ABSTRACT

Oracle-based pharmacy management software supports medicine stock control, prescription validation, and supplier coordination through a centralized digital system. The platform records drug quantities, batch numbers, expiry dates, storage locations, purchase details, dispensing history, and supplier information. Automated alerts notify pharmacy teams about low stock, near-expiry medicines, delayed deliveries, and urgent replenishment needs. Prescription validation tools can detect incorrect dosages, duplicate medicines, allergies, and potential drug interactions before dispensing. Integration with patient records, hospital billing, and procurement systems improves coordination among doctors, pharmacists, administrators, and suppliers. Oracle database technology enables reliable storage, rapid retrieval, secure transactions, and real-time reporting. Overall, the software can reduce medication errors, prevent shortages, minimize wastage, strengthen supplier management, and improve hospital pharmacy operations.

keywords: Oracle Pharmacy Software; Drug Stock Control; Prescription Validation; Supplier Coordination; Medication Management.

1. Introduction

Enterprise applications depend on reliable diagnostics to detect runtime errors, failed workflows, API issues, slow transactions, and integration faults [1]. Logs are the main evidence source for understanding what happened before, during, and after an incident [2]. Effective application diagnostics require structured logging, traceability, contextual metadata, error classification, and operational visibility [3]. However, fixed logging strategies often create two problems. Low-detail logs may miss important causes [4], while high-detail logs may generate excessive storage cost and alert noise [5].

The main challenge is that diagnostic needs change during runtime. A normal transaction may require only basic logs, but a failing workflow, repeated exception, or security-sensitive event may need deeper traces [6]. Adaptive logging improves this process by changing log depth according to system condition, transaction risk, and error behavior [7]. AI-assisted monitoring can identify when additional diagnostic detail is needed and when logging can return to normal levels [8]. This article proposes an adaptive logging framework for enterprise application diagnostics.

2. Methodology

The proposed framework captures log metadata from application servers, API gateways, databases, background jobs, authentication modules, and integration services [9]. It records event type, severity, timestamp, user role, transaction ID, request path, error frequency, response time, affected component, and workflow state [10]. These features are converted into diagnostic context profiles. Adaptive logging requires context because the same event may be low-risk in one workflow but critical in another [11].

The logging controller classifies runtime states as normal, warning, error-prone, performance-risk, security-sensitive, workflow-critical, or incident-active [12]. Based on this classification, it adjusts log level, trace depth, parameter capture, correlation ID tracking, and diagnostic enrichment. Lightweight scoring and policy rules are used so logging decisions remain fast and explainable [13]. Sensitive fields are masked before storage [14], and transaction-critical events are linked with workflow and database identifiers for easier diagnosis.

The framework is evaluated using simulated enterprise applications involving APIs, database workflows, authentication events, scheduled jobs, and service integrations. Static logging is compared with adaptive logging under repeated exceptions, slow requests, failed jobs, API timeout, suspicious login, and workflow interruption scenarios [15]. Evaluation metrics include diagnosis time, log volume reduction, root-cause evidence quality, missed-event rate, storage overhead, alert usefulness, and administrator acceptance rate. Feedback from resolved incidents, ignored logs, and administrator notes is used to refine logging policies over time.

3. Results and Discussion

The results show that adaptive logging improves diagnostics by capturing richer evidence only when runtime risk increases. In the baseline condition, static logs either lack enough detail or produce large volumes of low-value records. With the proposed framework, detailed traces are activated for high-risk events and reduced during normal execution. The strongest improvement appears in intermittent failures, where deeper logs are needed only around the failure window.

The discussion shows that adaptive logging must balance diagnostic depth with privacy and storage cost. Excessive logging can expose sensitive data or overload monitoring systems, while weak logging may hide failure causes. The main limitation is that log-level decisions depend on accurate runtime classification. Regular policy review and masking controls are necessary for safe long-term use.

4. Conclusion

This article presented an adaptive logging framework for enterprise application diagnostics. The framework adjusts log detail using runtime context, risk scoring, workflow state, and error behavior. It improves troubleshooting by increasing diagnostic evidence during abnormal conditions while reducing unnecessary log noise during normal operation. The study shows that adaptive logging can strengthen enterprise diagnostics when combined with structured metadata, privacy controls, and continuous incident feedback.

References

1. Kavuluri, H. V. R. (2023). Version-Controlled Transformation Logic for Reproducible Enterprise Workflows. *Journal of Grid, Machines and Drives*, 10, 9-16.
2. Kavuluri, H. V. R. (2021). Distributed Feature Stores for Machine Learning Pipelines with Versioned Synchronization. *Cross-Disciplinary Knowledge Systems*, 8, 9-16.
3. Kavuluri, H. V. R., & Keshireddy, S. R. (2019). Migrating Sensitive Government Databases to AWS RDS: Security, Compliance, and Risk Mitigation. *Emerging Digital Systems*, 6, 26-32.
4. Kavuluri, H. V. R. (2020). Clustering Analysis of User Behavior in Web-Based Software Applications. *Cross-Disciplinary Knowledge Systems*, 7, 7-12.
5. Kande, R., Kotla, C., & kanth Thottempudi, K. (2023). Data Quality Firewall for Real-Time Schema Validation and Automated Quarantine in Analytics Pipelines. *Transactions on Smart Electrical and Computational Systems*, 10, 29-36.
6. Kavuluri, V. V. R., Gorumutchu, M. R., Jagadhabhi, N., Mandapatti, J. K., & Bandla, S. (2021). Schema Volatility Propagation in AI-Driven Data Architecture Pipelines. *High-Performance Distributed Computing Review*, 8, 1-7.

7. Kotla, C., kanth Thottempudi, K., & Kande, R. (2022). AI Risk Management for Cloud Platforms with Quantitative Scoring Aligned with NIST AI RMF. *Journal of Privacy-Aware Computing Systems*, 9, 41-48.
8. Kande, R., Kotla, C., & kanth Thottempudi, K. (2022). Zero Trust Architecture with MLDriven Behavioral Analytics for Healthcare ERP on Microsoft Azure. *Journal of Grid, Machines and Drives*, 9, 29-36.
9. Kavuluri, H. V. R. (2023). PostgreSQL for AI-Driven Applications: Performance Tuning for Machine Learning Pipelines. *Perception, Navigation and Intelligent Devices*, 10, 37-48.
10. Keshireddy, S. R. (2019). Oracle APEX Integration with RESTful Services for Lightweight Enterprise Data Exchange. *Journal of Grid, Machines and Drives*, 6, 7-12.
11. Kavuluri, H. V. R. (2019). Bug Severity Classification in Issue Tracking Systems with Support Vector Machines. *High-Performance Distributed Computing Review*, 6, 7-12.
12. Keshireddy, S. R. (2022). Low-Latency Data Lake Ingestion Using Adaptive Partitioning and Query-Aware Layouts. *Cross-Disciplinary Knowledge Systems*, 9, 8-14.
13. Bandla, S., Kavuluri, V. V. R., Gorumutchu, M. R., Jagadhabi, N., & Mandapatti, J. K. (2021). Memory Pressure Amplification in Declarative Runtimes. *Transactions on Smart Electrical and Computational Systems*, 8, 39-45.
14. Keshireddy, S. R. (2021). Role-Based Access Control in Oracle APEX with Database Authentication. *Emerging Digital Systems*, 8, 7-12.
15. Kotla, C., kanth Thottempudi, K., & Kande, R. (2021). Deep Learning Anomaly Detection for HIPAA-Regulated Azure Cloud Threat Monitoring. *Journal of Privacy-Aware Computing Systems*, 8, 28-34.