© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716334

Asset Flow: A Centralized Web-Based Asset Lifecycle Management System

PROF. DR. SANJIVANI DEOKAR[1], PARTH TRIPATHI[2], SWET PANDEY[3], SAKSHI MAURYA[4], VAIBHAV DUBEY[5]

[1]*Professor, Department of Computer Science and Engineering in Artificial Intelligence and Machine Learning, Lokmanya Tilak College of Engineering, Mumbai, India*

[2, 3, 4, 5]*Students of B.E. Computer Science and Engineering in Artificial Intelligence and Machine Learning, Lokmanya Tilak College of Engineering, Mumbai, India*

Abstract- Efficient management of organizational assets is a fundamental requirement for operational stability, financial accountability, and regulatory compliance. In many small and medium enterprises (SMEs), asset tracking is still performed using manual registers or spreadsheet-based systems. These approaches are prone to human error, data redundancy, delayed updates, and limited scalability. As the number of assets increases, the absence of a centralized monitoring system results in poor visibility, inefficient allocation, and increased operational risk. Although enterprise-grade asset management platforms provide advanced lifecycle monitoring and predictive analytics, they often involve high implementation costs, complex deployment processes, and significant technical overhead. Such systems are typically optimized for large industrial environments rather than SMEs with moderate asset volumes.This research presents Asset Flow, a centralized web-based asset lifecycle management system designed to unify asset registration, allocation, monitoring, and reporting into a secure and scalable platform. The proposed system utilizes a layered architecture implemented using Spring Boot, Hibernate ORM, and a structured relational database. It supports real-time asset updates, automated maintenance scheduling, warranty tracking, depreciation monitoring, and audit-ready reporting.A role-based authentication framework ensures secure access control and data integrity. Experimental evaluation demonstrates improved data retrieval efficiency, elimination of duplication, reduced administrative overhead, and enhanced lifecycle transparency. The proposed system provides a cost-effective and extensible solution that bridges the gap between manual tracking systems and enterprise-level platforms.

I. INTRODUCTION

Organizational assets, including hardware devices, licensed software, infrastructure components, and digital resources, represent significant financial and operational investments. Effective tracking and lifecycle monitoring of these assets are critical to ensuring productivity, cost control, and regulatory compliance. Poor asset governance often results in misplaced resources, expired warranties, missed maintenance schedules, and inefficient allocation across departments.

Traditional asset tracking mechanisms rely on manual registers or spreadsheet applications. While such systems are simple to implement, they suffer from several structural limitations. Manual systems lack validation constraints, audit trails, and concurrent accessibility. Spreadsheet-based systems, although slightly more organized, do not provide strong database normalization, automated scheduling, or role-based access control. As asset volume increases, maintaining data accuracy becomes increasingly difficult.

Enterprise-level asset management solutions offer comprehensive features such as predictive maintenance and compliance analytics. However, these systems are frequently expensive, resource-intensive, and complex to configure. Small and medium enterprises often require a more streamlined solution that balances functionality, cost-efficiency, and scalability.

The primary motivation behind this research is to develop a centralized and modular system that integrates asset registration, lifecycle monitoring, reporting, and security within a unified architecture. The proposed system focuses on reducing manual intervention, enhancing transparency, and enabling

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716334

real-time decision-making through structured database design and automated backend processes.

A. Research Objectives

The objectives of this research are formulated to address the limitations of existing systems and enhance organizational asset governance:

1. To design a centralized web-based platform for comprehensive asset lifecycle management.
2. To eliminate data silos by integrating all asset records into a unified relational database.
3. To implement automated notifications for maintenance schedules, warranty expirations, and depreciation tracking.
4. To establish a secure authentication framework using role-based access control.
5. To generate structured, exportable, and audit-ready reports for compliance purposes.
6. To ensure scalability and maintainability through layered architectural design.
7. To provide a cost-effective solution optimized for SMEs without sacrificing extensibility.

II. LITERATURE REVIEW

Asset management systems have evolved significantly over the past decades. Early systems were manual and paper-based, focusing primarily on record keeping. With the advancement of digital technologies, spreadsheet applications became common tools for asset tracking. However, these solutions lack database integrity enforcement and automated lifecycle monitoring.

Modern asset management systems can be categorized into three primary types:

1. Manual and spreadsheet-based systems
2. Enterprise Asset Management (EAM) platforms
3. IT-specific asset tracking solutions

Manual systems are cost-effective but unreliable for large datasets. Enterprise platforms provide advanced analytics but involve high financial and technical costs. IT asset management tools centralize digital asset monitoring but may not integrate all asset categories in a unified view.

The literature suggests that while enterprise systems are technically robust, there remains a significant gap in affordable and modular systems designed specifically for SMEs.

A. Reviewed Papers and Analysis

Enterprise platforms such as IBM Maximo are widely recognized for their lifecycle management capabilities. These systems support predictive maintenance, regulatory compliance, and workflow automation. However, they require significant infrastructure and specialized expertise, making them less suitable for smaller organizations.

IT-focused tools such as ManageEngine AssetExplorer provide centralized dashboards and reporting features for tracking IT resources. While effective for digital asset management, these systems often operate on subscription models and may restrict customization flexibility.

Academic research on web-based inventory systems using frameworks such as Spring Boot and Hibernate demonstrates efficient CRUD operations and structured database management. However, many such studies focus primarily on stock management rather than full asset lifecycle tracking, including maintenance scheduling and depreciation monitoring.
The analysis reveals that most existing solutions either prioritize enterprise-scale features or focus narrowly on inventory management, leaving a gap for lightweight yet comprehensive lifecycle management platforms.

B. Synthesis and Research Gap Identification

From the reviewed systems and literature, the following research gaps are identified:

- Absence of SME-optimized centralized lifecycle management systems
- Limited automation for maintenance and depreciation tracking
- Weak integration between reporting modules and authentication mechanisms
- Lack of unified platforms covering hardware, software, and digital assets

The proposed system addresses these gaps by combining lifecycle automation, structured database

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716334

design, secure authentication, and audit-friendly reporting within a modular and scalable architecture.

III. SYSTEM ARCHITECTURE

The system follows a layered architectural model to ensure modularity, scalability, and maintainability. The primary layers include:

1. Presentation Layer (User Interface)
2. Controller Layer (REST APIs)
3. Service Layer (Business Logic)
4. Data Access Layer (DAO using Hibernate ORM)
5. Database Layer (Relational SQL Database)

The frontend is developed using HTML, CSS, and JavaScript to provide a responsive and intuitive interface. The backend is implemented using Spring Boot, which facilitates RESTful API development and dependency management. Hibernate ORM is used to map Java entities to relational database tables, reducing manual SQL complexity.

This layered structure ensures separation of concerns, enabling independent updates and easier debugging. The architecture also supports scalability by allowing horizontal expansion if deployed in distributed environments.

A. Core Components

The system consists of the following major components:

1. Asset Registration Module – Handles creation, modification, and deletion of asset records.
2. Allocation Module – Manages assignment of assets to users or departments.
3. Maintenance and Warranty Scheduler – Monitors upcoming deadlines and triggers alerts.
4. Depreciation Tracking Module – Calculates asset value reduction over time.
5. Reporting Module – Generates structured reports with CSV export functionality.
6. Dashboard Module – Provides real-time visualization of asset status and lifecycle states.
7. Audit Log Module – Records transactional history for accountability and traceability.

Each component interacts with the centralized database, ensuring data consistency and synchronization across modules.

B. Authentication Framework

Security is implemented using Spring Security with role-based access control. Users are categorized into roles such as Administrator, Manager, and Employee. Key features of the authentication framework include:

- Encrypted password storage using BCrypt hashing
- Role-based endpoint authorization
- Session management with timeout control
- Restricted database access through secure API layers
- Audit logging of login attempts and critical transactions

This framework ensures confidentiality, integrity, and availability of sensitive asset information.

IV. IMPLEMENTATION DETAILS

The system is implemented using:

- Backend: Java 17 and Spring Boot
- ORM: Hibernate
- Database: MySQL
- Frontend: HTML, CSS, JavaScript
- Security: Spring Security

The database schema is normalized to reduce redundancy and enforce referential integrity. Primary and foreign key relationships ensure structured linkage between assets, users, and allocation records.

Indexes are created on frequently queried fields such as Asset ID and User ID to optimize search performance. The scheduler module periodically scans the database to identify maintenance or warranty deadlines and updates dashboard notifications accordingly.

V. RESULTS AND EVALUATION

The system was evaluated using simulated asset datasets representing real-world SME environments. Testing focused on performance, accuracy, and functional reliability.

Observed improvements include:

- Significant reduction in asset retrieval time compared to spreadsheet search.
- Elimination of duplicate entries through unique constraints.
- Automated generation of maintenance alerts.
- Accurate depreciation tracking over defined intervals.
- Structured CSV export for audit compliance.

The centralized database approach improved consistency and minimized manual errors. The layered architecture ensured stable performance even under concurrent user operations.

VI. TECHNICAL CHALLENGES AND FUTURE DIRECTIONS

During development, several technical challenges were encountered:

- Designing scalable schema structures
- Managing concurrent database transactions
- Ensuring secure session handling
- Optimizing query performance

Future enhancements include:

- Integration with IoT or RFID systems for automated tracking
- AI-based predictive maintenance models
- Cloud-native deployment for distributed organizations
- Blockchain-based immutable audit logging

These advancements can further enhance automation and intelligence within the system.

VII. CONCLUSION

This research presented a centralized web-based asset lifecycle management system designed to enhance organizational efficiency and transparency. By integrating structured database architecture, automated lifecycle monitoring, and secure authentication mechanisms, the system eliminates data silos and reduces manual intervention.

The modular design ensures scalability and adaptability, making the platform suitable for SMEs while retaining extensibility for future technological integration. The evaluation results validate the system's effectiveness in improving operational control, accountability, and decision-making.

REFERENCES

[1] IBM Corporation, *IBM Maximo Asset Management Documentation*, 2023.

[2] Zoho Corporation, *ManageEngine IT Asset Management Guide*, 2023.

[3] P. S. Reddy and K. Kumar, "Design of Web-Based Inventory Systems Using Spring Boot and Hibernate," International Journal of Computer Applications, 2022.

[4] J. F. Nunamaker Jr., R. H. Sprague, and M. Chen, *Information Systems Management and Decision Support Systems*, Springer, 2020.