International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

An Integrated Framework for Hospital Resource and Inventory Optimization

Kashish Porwal[1], Anindhya Sharma[2], Prof. Sneha Singha[3]

[1,2,3]MIT ADT University, Pune, Maharashtra, India

ABSTRACT

Contemporary hospitals continuously accumulate substantial volumes of heterogeneous operational records—including patient intake logs, medicine dispensation histories, staff shift details, ward bed status, and financial billing data. Despite this wealth of information, the majority of healthcare facilities—particularly those of small to medium scale—lack the analytical infrastructure needed to transform raw operational records into meaningful, decision-ready insights. The result is a cascade of systemic inefficiencies: pharmaceutical stock accumulates beyond need and exceeds shelf life, clinical personnel face uneven workload distributions, and ward capacity is not utilized to its potential.

This work introduces DRIMS (Data-driven Resource and Inventory Management System for Hospitals), a comprehensive solution that unifies operational data streams and applies descriptive and diagnostic analytics to equip hospital administrators with actionable decision support. Rather than depending on computationally intensive machine learning pipelines, DRIMS is built upon robust statistical computations, SQL-driven data aggregations, configurable rule-based alerting mechanisms, and accessible visual dashboards. The system delivers key performance indicators (KPIs) spanning pharmaceutical consumption rates, automated restocking advisories, staff efficiency scores, and ward occupancy trends. System performance was assessed through a synthesized dataset modelling six months of operations at a representative mid-scale hospital (~12,500 transactions). Outcomes confirm meaningful gains in inventory turnover efficiency, a reduction in pharmaceutical waste, and enhanced administrative workflow responsiveness.

Keywords — Hospital Resource Management, Healthcare Informatics, Data Analytics, Inventory Optimization, Operational Efficiency.

I. INTRODUCTION

Effective healthcare delivery demands more than clinical expertise—it requires disciplined coordination of diverse resources, including pharmaceutical stocks, hospital infrastructure, and clinical personnel. In practice, many hospitals accumulate large volumes of operational data that remain siloed across departments and seldom leveraged for administrative improvement. Instead of analytics-driven planning, most facilities depend on periodic manual audits and informal estimation, leaving administrators ill-equipped for proactive decision-making.

This paper presents DRIMS—a pragmatic, deployable system that aggregates hospital operational data, executes periodic analytical routines, and presents findings through intuitive dashboards and configurable alert mechanisms. A deliberate design principle of DRIMS is to avoid opaque predictive modeling in favor of transparent, interpretable statistical and rule-based methods that clinical administrators can readily audit and trust. The system pursues four operational objectives: (1) curtail unnecessary pharmaceutical wastage, (2) promote equitable distribution of staff workload, (3) enhance utilization of ward bed capacity, and (4) accelerate and simplify administrative reporting workflows.

The remainder of this paper proceeds as follows: Section II surveys related work in hospital information systems and healthcare analytics; Section III explains the role of data analytics within DRIMS; Section IV details the system architecture and design decisions; Section V describes the development methodology; Section VI presents the algorithmic logic underpinning core system functions; Section VII explains implementation specifics; Section VIII reports evaluation findings; Section IX discusses practical implications and constraints; Section X outlines directions for future development; and Section XI concludes the paper.

II. LITERATURE REVIEW

Hospital Information Systems (HIS) and Electronic Medical Records (EMR) represent a well-established domain of healthcare informatics research. Open-source platforms such as OpenMRS and various commercial counterparts have achieved maturity in patient registration, clinical documentation, and billing management. Nevertheless, a persistent gap exists between the transactional capabilities of such systems and the analytical functionality necessary for operational governance.

Prior studies have explored machine learning techniques—including demand forecasting algorithms and readmission prediction models [1][2]—to extract operational value from healthcare data. However, such approaches carry significant barriers to adoption: they demand sizable labeled training corpora, specialized data science competency, and ongoing model maintenance—factors frequently unavailable in smaller healthcare institutions. Inventory management literature has similarly explored time-series forecasting and mathematical optimization to anticipate demand and minimize procurement costs, though these techniques require precise feature engineering and regular recalibration.

DRIMS diverges from these precedents by anchoring its analytical layer in descriptive statistics, diagnostic pattern recognition, and rule-based operational heuristics. This choice prioritizes accessibility and transparency over predictive sophistication, yielding a system that is lighter in infrastructure requirements, easier to sustain, and more readily understood by the staff who must act on its recommendations—while still generating substantive operational value.

III. ROLE OF DATA ANALYTICS

Within DRIMS, data analytics is organized into three complementary layers: descriptive analytics, diagnostic analytics, and rule-based operational logic.

Descriptive analytics consolidate historical activity records—daily pharmacy dispensation volumes, hourly patient arrival rates, monthly revenue summaries—through SQL-based aggregation routines and Python-driven data transformation pipelines. Diagnostic analytics probe anomalies and inter-variable correlations, for instance linking a surge in antibiotic consumption to seasonal respiratory illness peaks. Rule-based logic encodes domain expertise through configurable thresholds: reorder level calculations,

safety stock policies, and expiry-date prioritization rules that convert raw metrics into concrete, clinically relevant recommendations.

The deliberate preference for statistical over predictive methods reflects a fundamental consideration in clinical settings: healthcare administrators require metrics that can be audited and understood. For example, the restocking threshold is computed as:

Reorder Level = (Average Daily Consumption × Lead Time) + Safety Stock

where Safety Stock = k × σ(Daily Consumption), and k represents a configurable service level multiplier (typically k = 1.65 for a 95% service level). These computations are expressible as concise SQL queries or compact Python scripts, making them accessible to analysts with standard technical skills.

IV. SYSTEM DESIGN AND ARCHITECTURE

DRIMS adopts a modular three-tier architecture comprising Presentation, Application, and Data layers. The Presentation layer provides role-differentiated web interfaces for system actors—administrators, pharmacists, physicians, and nursing staff—constructed with HTML/CSS/JavaScript and React.js component libraries. The Application layer, implemented in Node.js with Express, exposes RESTful API endpoints servicing both transactional CRUD operations and analytical data retrieval. The Data layer employs MySQL for relational transactional storage alongside a lightweight analytical warehouse following a star schema (fact and dimension tables).

Principal system components include:

- Transactional Module — captures patient registrations, medication prescriptions, pharmacy dispensation events, ward admissions and discharges, staff scheduling entries, and billing records.
- Analytics Pipeline — scheduled ETL processes implemented in Python that extract and transform transactional records into consolidated metrics, populating the analytical warehouse tables.
- Dashboard and Reporting Layer — Power BI reports or embedded Chart.js visualizations presenting KPIs with interactive filtering and drill-down capabilities.
- Notification Engine — a rule-based microservice that generates email and SMS alerts for stock-out risks, near-expiry pharmaceutical batches, and critical ward occupancy thresholds.

This architectural separation ensures optimized performance for high-frequency transactional workloads while permitting resource-intensive analytical queries to execute against the dedicated warehouse without degrading operational responsiveness.

V. DATA MODEL AND SCHEMA DESIGN

The data architecture employs normalized relational tables for operational data capture alongside a denormalized star schema for analytical processing. Core transactional tables are:

- Patients (patient_id, full_name, date_of_birth, gender, contact_details, address)
- Staff (staff_id, full_name, role, department, assigned_shift)
- Inventory (item_id, item_name, batch_number, expiry_date, current_quantity, unit_cost, supplier_id)

- Prescriptions (prescription_id, patient_id, prescribing_doctor_id, issue_date, prescribed_items)
- Admissions (admission_id, patient_id, ward_id, bed_id, admission_date, discharge_date)
- Transactions (transaction_id, transaction_type, amount, transaction_date, reference_id)

Analytical fact tables—MedicineUsageFact and AdmissionsFact—support time-based aggregations over shared dimension tables: DateDim, DepartmentDim, ItemDim, and StaffDim. This structure enables computationally efficient queries for rolling averages, seasonal trend indices, and period-over-period comparative metrics.

VI. DEVELOPMENT METHODOLOGY

System development follows Agile methodology, structured into iterative delivery sprints that progressively build core functional modules—pharmacy management, ward admissions, and staff scheduling. Analytical capabilities were introduced subsequent to validation of the minimum viable product (MVP) for transactional operations, following a principle of analytics-on-top-of-solid-data-foundations.

Each sprint incorporated unit testing of RESTful API endpoints, integration testing of ETL pipeline components, and User Acceptance Testing (UAT) conducted with domain-specialist participants representing pharmacist and administrative roles within a controlled testbed environment.

Statistical methodologies applied within the analytical layer include: rolling moving averages over 7-day and 30-day windows, coefficient of variation metrics for characterizing demand volatility, seasonal index derivation using the ratio-to-moving-average technique, and safety stock formulations as described previously. ETL processing leverages Python's Pandas library for data transformation tasks, with performance-critical aggregations delegated to MySQL stored procedures.

VII. SYSTEM ALGORITHMS AND PROCESS FLOWS

Three principal operational workflows are formalized below:

A. Inventory Monitoring Workflow (Rule-Based)

1. For each pharmaceutical item, calculate Average Daily Consumption (ADC) across the trailing N-day window.
2. Retrieve the supplier-defined Lead Time (in days) for the item.
3. Compute the Reorder Level: RL = ADC × Lead Time + Safety Stock, where Safety Stock = k × σ(Consumption).
4. If the current Quantity-on-Hand falls at or below RL, issue a Reorder Alert specifying the suggested procurement volume: (RL + Buffer Quantity) − Quantity-on-Hand.
5. Scan inventory batches: flag any batch with an expiry date within a configurable threshold window for priority dispensing to minimize pharmaceutical waste.

B. Staff Utilization Measurement

6. Aggregate patient encounter counts per staff member per shift period.
7. Compute the Utilization Ratio: Utilization = Actual Patient Encounters ÷ Standard Encounters per Shift.

8. Classify each staff member as Underutilized, Within Normal Range, or Overutilized based on administrator-defined threshold bands.

C. Bed Occupancy Reporting

9. For each ward, determine the count of occupied beds relative to total available beds at each reporting interval.
10. Calculate Occupancy Rate (%): (Occupied Beds ÷ Total Beds) × 100.
11. Trigger critical occupancy alerts when the ward occupancy rate exceeds a configurable upper safety bound (e.g., 90%).

All three algorithms are designed for maximum transparency and configurability, permitting hospital administrators to calibrate threshold values and safety factors in accordance with institutional policies and clinical standards. Detailed pseudo-code is provided in Appendix A.

VIII. IMPLEMENTATION DETAILS

The DRIMS prototype was developed using a coherent full-stack technology selection. The frontend employs React.js with Bootstrap, delivering responsive and accessible user interfaces across device types. Backend services are implemented in Node.js (Express), providing RESTful API endpoints secured via JSON Web Token (JWT) authentication. Persistent storage uses MySQL configured for two distinct roles: an OLTP transactional database and a separate OLAP analytics warehouse hosting aggregated metrics for reporting. Python 3.9 scripts, orchestrated via system cron scheduling, execute ETL workflows and publish derived metrics to the analytics warehouse, from which Power BI consumes data for dashboard rendering.

Security design incorporates role-based access control (RBAC) ensuring personnel access only data pertinent to their function. Transport security is enforced via HTTPS. All database interactions use parameterized queries to eliminate SQL injection vulnerabilities, and comprehensive server-side input validation is applied throughout. Automated nightly backup and recovery procedures maintain data integrity, and structured audit logging supports compliance and traceability requirements.

IX. EVALUATION AND RESULTS

DRIMS was evaluated against a synthetically generated dataset designed to represent six months of activity at a representative mid-scale hospital: approximately 12,500 inventory transactions, 7,800 patient encounters, and 3,200 ward admission events. Seasonal demand variation was embedded in the dataset—for example, elevated cold and respiratory medication usage during monsoon-equivalent periods—to test system responsiveness under realistic fluctuation conditions.

Performance was assessed across four key metrics:

- Pharmaceutical Wastage Reduction: percentage decrease in expired pharmaceutical units relative to a fixed periodic reorder baseline.
- Inventory Turnover Ratio: measure of how frequently stock is cycled through relative to the baseline policy.
- Staff Workload Balance: standard deviation of per-staff workload indices across evaluated periods.

- Average Time-to-Reorder: mean elapsed days between a reorder trigger event and the generation of a procurement recommendation.

Outcomes were as follows: pharmaceutical wastage fell by approximately 32% relative to the baseline periodic reorder approach; inventory turnover improved by approximately 18%, attributable to more timely and demand-responsive restocking recommendations; staff workload standard deviation declined by approximately 25%, reflecting enhanced shift balance; and the mean time-to-reorder contracted from 10 days under the baseline to 4 days under DRIMS, substantially improving procurement responsiveness.

These findings derive from controlled simulation conditions and should be interpreted as indicative of potential real-world gains rather than definitive performance guarantees. Deployment in operational hospital environments will necessitate calibration against site-specific supplier lead times, procurement logistics, and clinical protocols.

X. DISCUSSION

DRIMS demonstrates that streamlined, interpretable analytics are sufficient to realize meaningful efficiency improvements across hospital operations. The rule-based inventory management component successfully reduced pharmaceutical waste and accelerated stock turnover without recourse to sophisticated predictive modeling. Staff utilization metrics similarly enabled administrators to make data-informed scheduling adjustments, improving workload equity across shifts.

A central value proposition of DRIMS is its commitment to transparency and operational accessibility. Healthcare administrators and clinical staff are more inclined to trust and adopt systems whose outputs are explainable and auditable. DRIMS achieves this through reliance on established statistical measures and administrator-adjustable rules, thereby reducing the adoption barrier that opaque machine learning systems frequently encounter in clinical governance contexts.

Limitations

The current prototype carries several acknowledged limitations. Evaluation is grounded in simulation data, and field deployment may surface operational edge cases not represented in the synthetic dataset. The absence of predictive analytical modules constrains DRIMS's ability to anticipate sudden, atypical demand surges—such as those arising from localized disease outbreaks—compared to advanced forecasting solutions. Additionally, integration with legacy hospital information systems may present non-trivial technical challenges, potentially requiring bespoke ETL connector development.

It is further recognized that technical deployment alone is insufficient; organizational change management, staff training programs, and institutional data quality assurance are essential enablers of practical system impact and fall outside the technical scope of the current work.

XI. FUTURE ENHANCEMENTS

Planned development trajectories are designed to incrementally extend capability while preserving the interpretability principles central to DRIMS:

- Incorporation of lightweight time-series forecasting methods (e.g., Facebook Prophet) to introduce short-horizon demand prediction while maintaining output explainability.

- Integration with IoT sensor networks for automated real-time bed occupancy sensing and temperature-controlled pharmaceutical storage monitoring (cold chain compliance).
- Development of standardized interoperability connectors for major EMR and HIS platforms, enabling automated data ingestion without manual ETL configuration.
- Delivery of a mobile application to support real-time push notifications and mobile approval workflows for administrators and pharmacists.

Further strategic extensions—including supplier lead-time profiling and multi-supplier procurement optimization—are envisaged to strengthen supply chain resilience. Critically, any future predictive modules will be accompanied by interpretable supporting visualizations to preserve user confidence and institutional governance alignment.

XII. CONCLUSION

This paper has introduced DRIMS, a deployable, data-driven Hospital Resource and Inventory Management System engineered to enhance operational efficiency in healthcare institutions through transparent statistical analytics and configurable rule-based automation. The system's modular architecture, practical analytical methods, and commitment to interpretability collectively address the core operational challenges—pharmaceutical wastage, staff imbalance, and suboptimal bed utilization—faced by small and medium-scale hospitals without imposing excessive infrastructure demands.

Simulation-based evaluation substantiates meaningful improvements in pharmaceutical waste reduction, inventory responsiveness, and staff workload equity. Deployed thoughtfully and supported by appropriate organizational change processes, DRIMS offers immediate operational value and provides a sound architectural foundation upon which future, more advanced analytical capabilities can be developed incrementally.

REFERENCES

1. OpenMRS Project. Open-source medical records system for developing countries. Available at: https://openmrs.org
2. G. Hripcsak and D. J. Albers, 'Next-generation phenotyping of electronic health records,' JAMA, 2013.
3. J.-W. Bi et al., 'Forecasting daily tourism demand for tourist attractions with big data: an ensemble deep learning method,' Journal of Travel Research, 2022.
4. S. Behera and S. Mohanty, 'Digital Transformation in Hospital Management,' Journal of Healthcare Informatics, 2023.
5. K. Sharma et al., 'Data Analytics in Healthcare Management Systems,' IJCSIT, Vol. 12, No. 5, 2023.
6. World Health Organization (WHO), 'Health Data Management and Digital Transformation,' 2021.
7. M. Singh et al., 'Efficient Data Analytics in Healthcare Operations,' International Journal of Computer Applications, 2022.

8. H. W. Bierman and I. J. Fleming, 'Inventory Control and Management in Healthcare,' Health Systems Journal, 2019.
9. S. Elkington et al., 'Hospital bed occupancy and its effect on patient flow,' International Journal of Healthcare, 2020.
10. T. Hastie, R. Tibshirani, and J. Friedman, 'The Elements of Statistical Learning,' Springer, 2009.
11. Facebook (Meta) AI Research, Prophet — Forecasting at Scale. Available at: https://facebook.github.io/prophet/
12. Open Data Kit, 'Best Practices for Data Collection in Healthcare Settings,' 2018.

APPENDIX A: SAMPLE SQL QUERIES AND PSEUDO-CODE

SQL query — compute Average Daily Consumption (ADC) over the trailing 30 days for a specified inventory item:

```
SELECT item_id,
    SUM(quantity_issued) / 30.0 AS avg_daily_consumption
FROM   medicine_usage
WHERE  usage_date >= DATE_SUB(CURDATE(), INTERVAL 30 DAY)
  AND  item_id = <TARGET_ITEM_ID>
GROUP  BY item_id;
```

SQL query — identify pharmaceutical batches approaching expiry within the next 30 days:

```
SELECT item_id, batch_number, expiry_date, quantity_on_hand
FROM   inventory
WHERE  expiry_date <= DATE_ADD(CURDATE(), INTERVAL 30 DAY)
ORDER  BY expiry_date ASC;
```

Pseudo-code — Inventory Monitoring Routine:

```
FOR EACH item IN active_inventory:
   adc          = compute_average_daily_consumption(item, window_days=30)
   lead_time    = retrieve_supplier_lead_time(item)
   safety_stock = k_factor * std_deviation_consumption(item)
   reorder_level = adc * lead_time + safety_stock
   IF item.quantity_on_hand <= reorder_level:
      suggested_qty = (reorder_level + target_buffer) - item.quantity_on_hand
      raise_reorder_alert(item, suggested_qty)
   IF item.expiry_date <= TODAY + expiry_alert_threshold:
      flag_for_priority_dispensing(item)
```