

## RESEARCH PAPER

# Web-Based Real-Time Monitoring and KPI Management System for Engine Development Laboratory Operations

Dr. Swapnaja A Ubale<sup>1</sup>, Ashish Rangdale<sup>2\*</sup>

<sup>1</sup>HoD, Department of Information Technology, Marathwada Mitra Mandal's College of Engineering, Savitribai Phule Pune University, Pune, Maharashtra, India, Email: swapnaja.b.more@gmail.com

<sup>2\*</sup>Department of Information Technology, Marathwada Mitra Mandal's College of Engineering, Savitribai Phule Pune University, Pune, Maharashtra, India, Email: ashishrangdale@gmail.com

## ABSTRACT

The system being developed here offers an innovative approach to improving the visibility and reporting of laboratory operations. Designed around a standardized method of collecting and accessing operational data, the system addresses the issue of fragmented and delayed manual record keeping associated with many existing systems by providing a centralized database for recording laboratory operations. The platform has three key deficiencies: First, most of the data input into the system is still done manually, which makes the success of the platform dependent on the accuracy and timeliness of the operators' input. Second, due to the platform being built specifically to suit the workflow of the source lab, it may need to go through configuration changes before we can implement it in other labs that do not have the same shift schedule, number of cells, or regulatory constraints. Third, the platform is designed primarily for monitoring and reporting purposes; while the potential is present to develop analytic capabilities such as predictive modelling, anomaly detection, and automated failure classification, these capabilities have yet to be developed. These limitations are in line with the Design Science Research approach utilized in this study, where the principal objective was to solve a well-defined operational issue rather than to produce a fully functioning analytics platform.

**Keywords:** Industrial Engineering, Operations Management, Real-Time Monitoring, KPI Dashboard, Maintenance Management, Laboratory Digitalization, MTTR, MTBF, Industry 4.0, Web-Based Monitoring System.

**How to cite this article:** Ubale SA, Rangdale A. Web-Based Real-Time Monitoring and KPI Management System for Engine Development Laboratory Operations. *Int J Drug Deliv Technol.* 2026;16(47s): 779-788. DOI: 10.25258/ijddt.16.47s.87

## INTRODUCTION

The EDL (engine development laboratories) consists of multiple test cells, each with its own engine, customer, and testing protocols that run at the same time. For this study, fourteen separate test cells were operated simultaneously, resulting in a great deal of operational data generated (test runs, breakdowns, fuel consumption, and maintenance). Currently, all of these operations exist in written form—very accurate but unable to be queried or collectively aggregated as live data and therefore not usable for use in automated reports. All Excel summaries of each register have been created for monthly periods only and are typically delayed by a few weeks. Therefore, management decisions are often based on stale information, and operationally relevant details are frequently not included. However, this information is also very well documented in the TC-10 March 2026 register and includes summary information about running in cycles, breakdown events (i.e., water hose leak), emissions information, fuel log information, and cross cell communication information. Such records also must be transcribed.

A significant second limitation is the incorrect estimate of available time when there is overlapping of shifts. These General, First, Second, and Third shifts (in different

combinations) are generally performed in different cells in many labs. Aggregating shift lengths just overestimates the number of hours available—and produces a skewness of reliability analysis. Without accurate computing models, management KPIs are systematically biased.

This paper creates a web-based integrated monitoring tool aimed expressly at overcoming these shortcomings. The system consists of: (i) live browser-based monitoring of all fourteen cells; (ii) organized digital logging of test runs, breakdowns, maintenance, fuelling, and engine configuration; (iii) a shift-aware KPI computation engine, according to interval-union logic; (iv) role-based access control that takes into account laboratory responsibilities; and (v) automation of Excel-based reporting to management. As indicated in Figure 1, the system design is already elaborated on in detail.

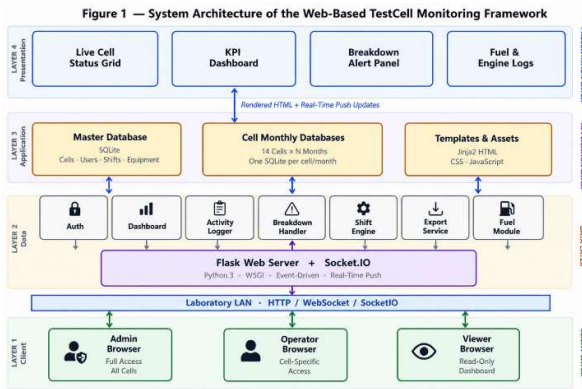


Figure 1: System Architecture of the TestCell Monitoring Framework

### LITERATURE REVIEW

The literature addressing this work includes five stream connections: Industrial IoT and Digital Monitoring, predictive maintenance in Industry 4.0/5.0, Reliability and maintenance measurement, Digital Twin & Data-Driven Maintenance, Condition monitoring of Internal combustion engines. Put together, such streams preach for real time monitoring, organized data base and predictive analytics – but, to conclude, are based on the assumption that operational data now is electronic – a not a reality in operator-controlled test-cell labs where primary data are handwritten.

#### Industrial IoT and Digital Monitoring

The first introductory survey by Xu, He, and Li on IoT in industries describes RFID, wireless sensors, and mobile technologies as enabling technologies for industrial monitoring, and frames industrial monitoring as a networked web of data rather than a problem for single devices. Though working in the context of some high-sensory factories, their framework failed to provide consideration of digitizing human-input operational knowledge – an acute challenge in the current scenario. The lab on study runs multiple cells in parallel, but operating performance is captured on paper and consolidated manually — thus, the monitoring challenge has been not just connectivity, but structured data capture.

#### Predictive Maintenance in Industry 4.0 and 5.0

Zonta et al. provide a systematic review on predictive maintenance in Industry 4.0 to show structured operational data is requisite for any kind of predictive models. Nunes, Santos, and Rocha also highlight practical problems such as noisy sensor data, the requirement for high-throughput real-time processing, and the tendency of current methods to consider only individual assets as opposed to the plant as the whole. Mallioris et al. prove that predictive maintenance

is not a single approach, but a set of methods that are directly related to the quality of the data pipeline. Murtaza et al. generalize this to Industry 5.0, arguing for future maintenance systems that have to be human-centric and sustainable—a point particularly timely in operator-driven laboratories where human judgement remains foundational to the monitoring process.

#### Reliability Assessment and Maintenance Metrics

Friederich and Lazarova-Molnar provide an overview of the major challenges of reliability assessment for manufacturing systems, with availability of data and its quality as fundamental barriers. Zyluk et al. define MTTR as the mean time taken to repair after failure and MTBF is a central availability metric. Reliability studies consistently demonstrate that the information for these metrics is meaningful only when the occurrence of failure and repair instances are recorded in a reliable and complete way – thus providing the rationale behind systematic breakdown logging in the system in place.

#### Digital Twins and Data-Driven Maintenance

Hassan et al. describe an industrial application that compares working-machine digital twins with maintenance records over 18 months and uses differences to predict maintenance requirements. Their work illustrates that you need to establish digital records before you will be able to apply advanced maintenance intelligence. In the wider Industry 5.0 literature, digital twins, with ML, IoT, and big data, are combined in condition monitoring frameworks, and each domain recommendation involves this step: Clean digital records, Operational regimes, Predictive or prescriptive analytics. Our contribution here addresses the first two stage in this chain.

#### Condition Monitoring of Internal Combustion Engines

Delvecchio et al. give a systematic review of vibro-acoustic condition monitoring for internal combustion engines and the methodological maturity of this field. Nithin et al. examine combustion and vibration monitoring techniques. Vilas Boas et al. suggests a Nelson rule based statistical process control method, proving that classical statistical methods are very effective without leveraging deep learning. Torres et al. offer a recent systematic review of non-invasive fault detection methods that has been taken in a manner in which a good data base, not just an algorithm, is essential for future research.

#### Research Gap

The compilation of these streams shows that there is a definite deficiency: no framework reviewed provides a solution for the problem of a multi-cell engine field

laboratory in which primary records are handwritten, shift overlaps are responsible for computational errors when estimating availability. Here we summarize how the present work filled some of those gaps as identified in the literature in Table 1.

Table 1: Research Gap Analysis: What the Literature Lacks vs. What This Paper Provides

Area	Literature Gap	This Paper's Contribution
<b>Multi-cell dashboard</b>	No unified system for monitoring many test cells	Live browser-based dashboard for 14 cells with equipment-wise status and alerts
<b>Shift overlap correction</b>	Existing studies do not solve overlapping shift-time computation	Interval-union shift engine that merges overlaps before computing available hours
<b>Per-day shift override</b>	No support for day-wise shift customization	Calendar-based shift overrides, Saturday-working rules, and holiday handling
<b>Structured breakdown logging</b>	Manual logs not standardized for digital analysis	Structured fields: start/end time, problem statement, solution, manual/facility flags
<b>Automated KPI calculation</b>	MTTR, MTBF, utilization computed manually	Real-time automatic KPI calculation from live database records
<b>Fuel type tracking</b>	Published systems rarely track fuel by category	Daily fuel logs with multiple fuel types and carry-forward meter logic
<b>Role-based cell access</b>	Most systems do not isolate operator access by cell	One operator account per cell plus admin/viewer roles
<b>Conflict detection</b>	Manual systems do not prevent overlapping entries	Real-time validation detects time conflicts before saving
<b>Exportable reporting</b>	Many systems lack easy monthly management reporting	Excel multi-sheet exports for activity, fuel, KPI, and breakdown reporting

**Problem Statement**

The engine development laboratory is a busy multi-cell facility in which up to fourteen test cells can operate simultaneously. Each cell has its own engine, instruments, and customer program with its own shift pattern. Currently operational details are recorded in a hand-written ledger, and later duplicated into Microsoft Excel. In general, the accuracy of these records is acceptable. However, there is no searchable or automated means to analyze these records to capture the performance in real-time. This creates significant challenges for the professionals trying to provide service.

Breakdown reports are prepared but not at or near the time of the actual event; this makes it difficult to properly assess how maintenance response and downtime was assessed. Fuel readings are recorded manually at start and end of each shift, and fuel consumption is calculated manually, which increases the probability of calculation errors and transcription errors. It is particularly challenging to quantify cell utilization and availability accurately because the laboratory does not rely on one fixed shift system. Different cells operate under various combinations of General, First, Second, and Third shifts and many of those shifts overlap. Another problem to address is traceability. The laboratory is required to determine for every test period, what was the main engine that was installed, what fuel it was using, what equipment in it was in use and whether any failures or maintenance has taken place. This will allow the lab to calculate and confirm the service level and timing to be accurate. Though this information is in narrative form from the manual register, it is not displayed in such a way that automatic KPI calculation can be performed. The resulting measures such as uptime, downtime, MTTR, and MTBF require a large amount of manual effort to reconstruct. As a result, the laboratory is not only without software support; it lacks a proper digital operating model for its lab operations.

**RESEARCH METHODOLOGY**

This approach is suitable since the main output in using a Design Science Research (DSR) is a software artifact created to solve a practical real-world operational problem. The research process will consist of a series of five steps: (1) Define the problem, (2) Translate requirements, (3) Build artifacts, (4) Design algorithms, and (5) Validate.

**Problem Diagnosis**

To understand how the current operational system works, we reviewed the existing manual registers and monthly Excel reports currently used to track the operate the system. The TC-10 March 2026 register was selected for use as the main data set for validation purposes as it contains a detailed

record of engine incidents which have occurred over a specific period of time (by date). The TC-10 register includes an incident log which captures the following operational data items: engine firing, running-in cycles, consumption (when using fuel), breakdowns with repair notes, emissions measurements, and inter-cell co-ordination of work, to illustrate that a large volume of operational data will be available from the laboratory but this data is currently fragmented and unstructured and therefore not automatically usable.

**Requirements Extraction**

From the analysis undertaken to address the problem three possible users were identified with the following main requirements: (1) Real-time monitoring of 14 cells, (2) incremental documentation of tests, breakdowns and maintenance, (3) KPI's calculated for each cell, with shift specific timekeeping, (4) Role based access control, (5) Shift schedule will be supported for calendars, holidays and overrides, (6) Allows for automated fuel logs for multiple fuel types, (7) Management's ability to export reports to Excel format.

**Artifact Construction**

API is a Flask web application that includes a master SQLite database with separate monthly SQLite files per test cell. This architecture organizes operational data along the cells and temporal distance, slices the files by each to the time of monthly reporting of the lab each month. The data model is cell function, breakdown and test run, maintenance events, fuel log, engine log tables, etc. Full flow in the data operation is written in Figure 2.

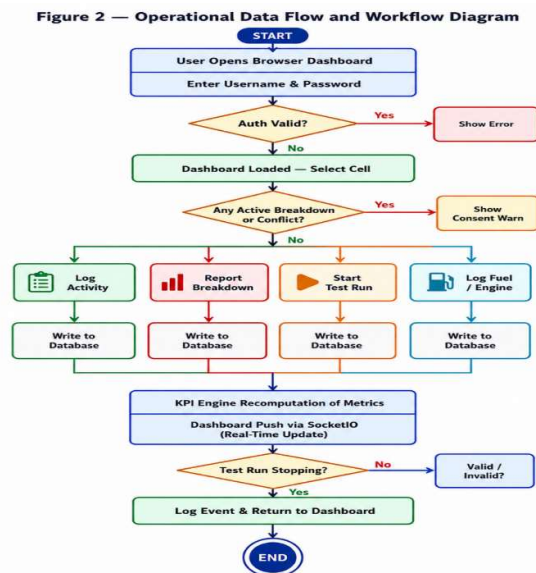


Figure 2 — Operational Data Flow and Workflow Diagram

### Algorithm Design

In computations, the shift engine is the heart. Algorithm does not directly sum up the shift lengths (dependent): (i) gives the intervals of time for each active shift for a specific day; (ii) clips those times to the desired date range; (iii) aggregates all overlapping intervals using an interval-union procedure; (iv) sums the periods of merged non-overlapping intervals. So use it as you have the actual available times to calculate utilization and uptime. It also provides Saturday working rules, declared holidays and shift-specific day overrides.

### Validation

The TC-10 March 2026 register was used to validate the proposed system’s accuracy concerning the calculations for the time and fuel used. By manually entering the same data from the register into the dashboard outputs, and comparing those values with the system’s calculations (e.g., downtime, uptime, mean time to repair (“MTTR”), mean time between fall (“MTBF”)), it was possible to confirm the dual validation of (1) functional correctness and (2) numerical accuracy of the artifact. The results of this validation process confirm that the artifact accurately reflects the laboratory’s operational records.

### System Design

#### Architecture Overview

This framework is a complete monitoring system developed within the laboratory’s local area network (LAN) and can work with a standard web browser. It consists of two databases: a master database and a cell database. The master database has data on the configuration including cell, equipment, user accounts, and shift rules. Cell DB maintains monthly SQLite files for each test cell. This design separates and compiles records individually against both a cell, and at the time period, which minimizes file size and aids in accessing and maintaining old files. This can be explained as shown in Figure 3.

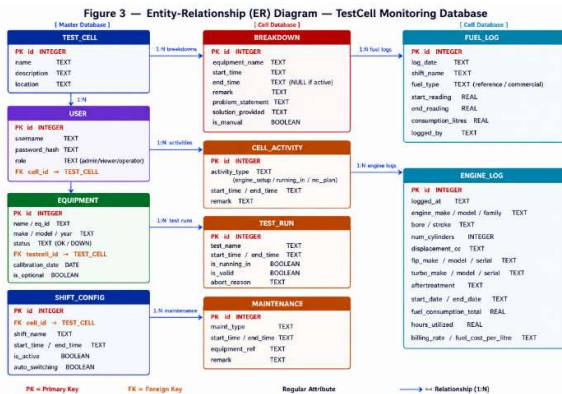


Figure 3: Entity-Relationship Diagram of the TestCell Monitoring System

### User Access Model

It is a Role-Based Access Control (RBAC) based system with three user types. The administrator can access all cells and set up the settings and view reports as expected by all cells. The operator of this code is connected to only a single cell, so that cells cannot enter incorrect information from cells and is consistent with an actual lab working process. The viewing experience: Only views the dashboard viewable, and is primarily for management only -- cannot enter or modify operational data.

### Functional Modules

The system consists of six function modules, each corresponding to one of the laboratory operational entities:

- Breakdown Module: for storing failure events with start time, end time, remark, problem statement, solution provided, flag for manual/facility etc. Live alerts to the dashboard are triggered after active breakdown.
- Activity Module: stores cell level activities such as engine setup, running-in, no-plan time and maintenance periods in a typing for start time and end time.
- Test Run Module: logs test sessions with test name, start/end time, running-in flag, validity status and abort reason. The system enforces that engine setup and test run cannot run simultaneously.
- Maintenance Module: logs scheduled and unplanned maintenance with equipment references, category on type, time.
- Fuel Log Module: stores start-of-shift and end-of-shift meter reading, fuel type (reference or commercial), computed consumption amount for each shift, and the identification of such shifts. Carry-forward logic keeps track for the continuity of more than one shift.
- Engine Log Module: preserves configuration traceability including engine make, model, family, bore, stroke, cylinders, displacement, FIP data, turbocharger data, aftertreatment type and billing/fuel-cost information.

### Shift-Calculation Engine

The shift engine comprises the largest components of the system that is algorithmically powerful. It enables four standard definitions of shift: General (06:30–15:00), First (07:00–15:00), Second (15:00–23:00) and Third (23:00–07:00), while an adjustment rule modifies the General shift start time when the First shift is working simultaneously. The algorithm proceeds as follows:

1. Generate time intervals. Each active shift is converted into a (start, end) interval on the date of request, and to accommodate midnight-spanning shifts is split into 2 segments.

2. Clip to date range: in multi-day queries, every interval is clipped to the start and end dates as requested for queries to avoid double-counting boundary days.
3. Sort and merge: intervals are sorted by start time and merged pairwise — overlapping intervals are replaced by an interval that spans from the earliest start time to the latest end time.
4. Sum durations: time with the merged non-overlapping intervals is total time.
5. The overlap problem and its resolution is shown in Figure 4. This naive sum will overstate the number of available hours if there are General, First, and Second shifts in operation; the interval-union would indeed reflect the true covered time.

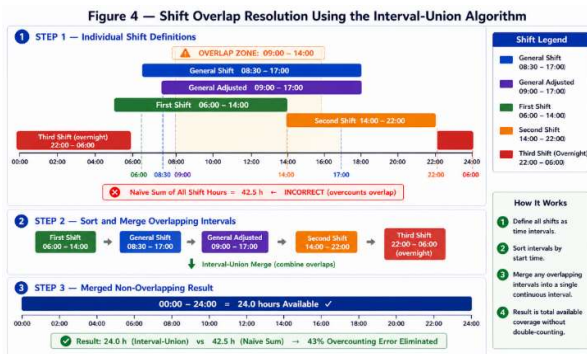


Figure 4: Shift Overlap Resolution using the Interval-Union Algorithm

**Dashboard and Reporting Design**

The dashboard is built up into three levels. At strategic level, we take KPI cards that display the breakdown count, total downtime, percentage of equipment uptime, MTTR, and MTBF data for a certain date series. On operational level, there is active breakdown alert strip with live failure events, time elapsed for response, cell number, and a straight path to equipment information page. Downtime data in the form of screens displaying downtime tables similar to those used to analyse the main reasons behind down time and the types of cell failures which occur are provided to engineers. The

dashboard presenting this information; see fig 5 of appendix A to view the dashboard. In addition to the reports are data generated by the reporting module that generate Excel files with activity records, log for downtime, total fuel use, test run validation and cell by cell summaries. All of the generated Excel files can be exported to be compatible with the existing Excel report system of the laboratory that will allow a bigger audience to see the data and minimise time spent on manual data preparation.

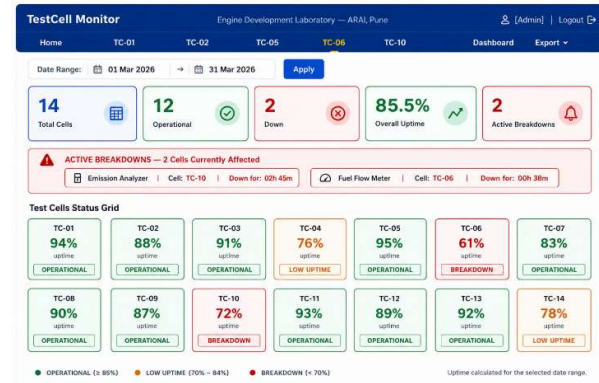


Figure 5: Dashboard UI Layout with KPI Cards, Active Alerts, and Cell Status Grid

**RESULTS AND DISCUSSION**

**Operational Event Digitization**

We are able to convert handwritten test cell registers into a structured data format. A database schema is created consisting of six separate tables for operational events (breakdowns, cell activity, test runs, maintenance, fuel logs and engine logs) from the full TC-10 March 2026 register. Each record is created with the time of an event as well as an explanation for a failure and a response taken for analytical purposes to assist in discovering the cause of a problem and assist in maintenance learning. The major differences between using a paper versus an electronic format are summarised in Table 2.

Table 2 — Comparison of Manual vs. Digital Operational Data Management

Parameter	Manual Method	Proposed System
Reporting lag	3–6 weeks (monthly Excel cycle)	Real-time (< 1 second from entry)
Data searchability	Not searchable — handwritten	Fully queryable by cell, date, type

Parameter	Manual Method	Proposed System
<b>KPI computation</b>	Manual arithmetic from registers	Automated from structured database
<b>Shift overlap handling</b>	Naïve sum — overcounts available time	Interval-union — mathematically correct
<b>Fuel error rate</b>	Susceptible to transcription errors	Automated carry-forward and calculation
<b>Breakdown traceability</b>	Narrative form — not standardized	Structured fields with solution logging
<b>Export effort</b>	Manual compilation (hours/month)	One-click Excel multi-sheet export
<b>Access control</b>	Physical register — no restriction	Role-based per-cell authentication

**Breakdown Analysis and Pareto Results**

The breakdown tracking platform systematically analyzes failure trends by form of equipment. Figure 6 presents a Pareto analysis of downtime by failure category for TC-10. Results corroborated the 80/20 principle: The three main failure types of emission analyzers, fuel flow meters, and dynamometer controllers make up about 60% of our downtime. The data presented here offer relevant evidence for decisions on the prioritization of maintenance, while planning for equipment procurement.

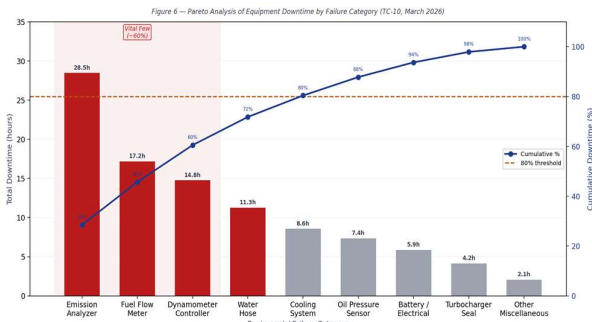


Figure 6: Pareto Analysis of Equipment Downtime by Failure Category (TC-10, March 2026)

**KPI Computation Results**

The automated KPI calculation engine generates MTTR, MTBF, and uptime numbers for all 14 cells in the chosen

date range. These KPI metrics are plotted in Figure 7 for all cells. It is worthy to note that: TC-06 and TC-10 show the greatest MTTR values (5.2 h, 3.4 h), which show the slowest repair reaction times, TC-11 and TC-05 have the most MTBF (158 h, 132 h), indicating the highest reliability, and uptime throughout the laboratory is 85.5%, with six cells out of the 90% target. These results take the role of manual KPI reconstruction, which would normally cost a few hours of the Engineer's time per reporting cycle.

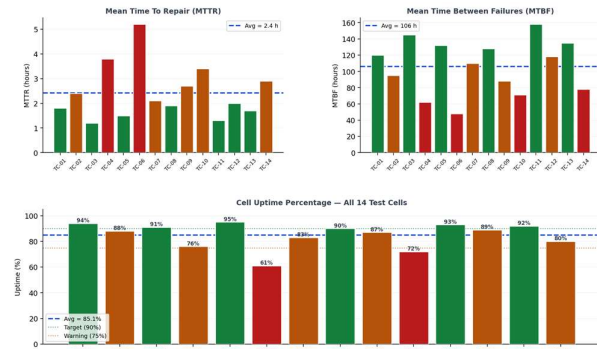


Figure 7 — KPI Visualization: MTTR, MTBF and Uptime Across All 14 Test Cells

Figure 7 : KPI Visualization: MTTR, MTBF, and Uptime Across All 14 Test Cells

**Fuel Accounting Results**

The fuel logging module eliminates manual calculations of fuel that could involve the necessary mathematical error and transcription. For TC-10, daily and cumulative fuel consumption trends in March 2026 for reference and commercial fuel types are represented in Figure 8. As a result, the chart identifies the four days during which the cell suffered from the breakdown, which is shown by zero consumption of the daily trend when the cell shows zero consumption. The aggregate view is the month-total figure obtained automatically, not by manual summation, needed for management fuel accountability reporting.

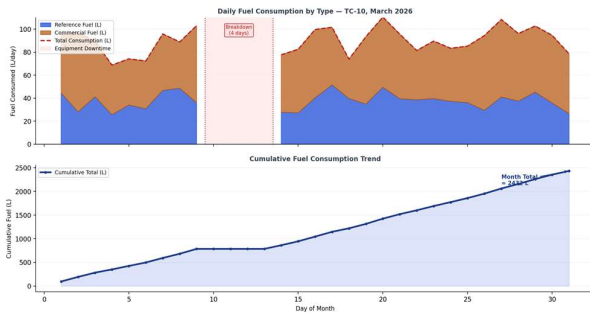


Figure 8: Fuel Consumption Trend Chart (Daily and Cumulative, TC-10, March 2026)

**Multi-Cell Utilization Analysis**

Breakdown of time utilization across all fourteen test cells by stacking test run, engine setup, servicing time, no-plan period, and unaccounted time during the available shift window is also given in Figure 9. The visualization showcases wide ranges in productive utilization: TC-01, TC-05, and TC-11 use productive utilization (test run + setup) > 50%, while TC-06 and TC-04 show substantially greater maintenance and breakdown time. Such results only occur as the interval-union shift engine appropriately calculates the available time of each cell's individual shift configuration, thus preventing the overestimation of utilization that would occur with naïve summation.

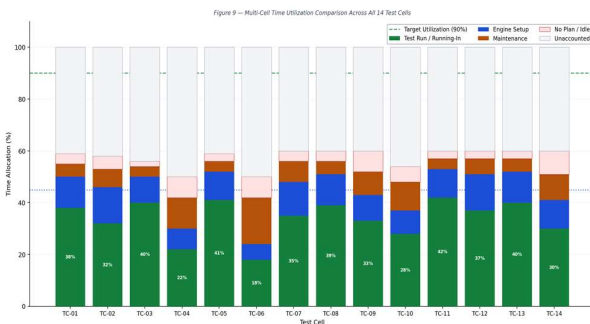


Figure 9 : Multi-Cell Time Utilization Comparison Across All 14 Test Cells

**DISCUSSION**

All findings show that we achieved our fundamental research aim — to develop a system where disconnected paper records are transformed into an operational monitoring tool. The important thing is not just digitization, but structuring in the form: every operational event is stored in such a way that the system can be queried and quickly aggregated with KPI data, aggregated and analyzed for trend without any more user activity.

The contribution of shift calculation engine is worth the focus as a technical contribution. The laboratory may use as many as four overlapping shift definitions simultaneously. In the absence of interval-union merging a cell using General as the base shift and First as the second shift at the same time would report 15.5 available hours per day while the true number is much closer to 8.5 hours, an overcounting error of 82%. This distortion is methodically corrected by the implemented algorithm and thus all downstream KPIs are more reliable.

A related practical consideration is the design of the data layer. So in short, this will save costs in contention as well as enlargement of file size, which can become much larger if the database is accessed by the whole system, because separate monthly SQLite files are written for every cell! This can be done at a fourteen-cell laboratory with 24/7 monthly operation. The application has cross-month query helpers that help keep date-range reports valid for several months.

More generally, this work shows that formal digital foundations can be built in operator-driven industrial settings: not just sensor-laden factories. The literature of predictive maintenance and reliability always assumes events in the digital realm are recorded; this exercise demonstrates how they might be confirmed in practice, building the datascape for advanced, future analytics.

**CONCLUSION**

In this paper we present a web-based monitoring tool for a fourteen-cell engine development laboratory. Using a structured digital system replaces handwritten registers and delayed Excel summaries enabling real-time operational monitoring. This project will provide a significant advancement in the software systems and build a structured database to store run-testing, breakdown maintenance events, fuel consumption data, and engine configuration. a new method for calculating the work time based on shift overlap and a shift calculation model provides reliable estimates of actual time worked by utilizing historical and actual shift duration data. The system is configured and will provide an automatic calculation of key performance indicators (KPIs), including mean time to repair (MTTR), mean time between failure (MTBF), and up/down time, as

well as utilization, for any number of users via a browser-based dashboard. system is validated against TC-10 operational data (March 2026), and confirms that both the tracking and calculation of time and fuel consumption are correct. Advantages provided by the new framework include eliminating delays in reporting, no estimation of available time, and generating management-focused reports in Excel format.

### Future Scope

The development of such a robust and well-constructed database will also facilitate the future development of reliability trend analysis, root cause classification, predictive maintenance, and other high-value applications. Consequently, the new system is not only an upgrade to existing reporting processes; it is also a key step toward the transformation of the laboratory into a digitally-oriented entity.8. Future Scope The digital environment created by the system allows for a few areas of improvement in the upcoming versions. However, it will be possible to use the stored event data to create predictive maintenance algorithms to anticipate failures, detect patterns in recurrent shutdowns, and to recommend the proactive approach regarding maintenance planning. In addition, the addition of sensors and Internet-connected devices to the system, such as dynamometers, emission analyzers, fuel flow meters, and temperature sensors, will permit more efficient automated data collection, thus avoiding the need for manual entries. Eventually, the dashboard of the project will display more robust data analysis tools such as: analysis of previous historical data trends; supplier reliability ratings by equipment; repeated failure analysis on specific equipment; operator workload comparisons with other operators in other cells over time; and provision of mobile alerts for active failures, delayed maintenance response, and missing fuel logs. Since this same basic framework can be applied to any other laboratory or test facility, future work may also incorporate benchmarking across multiple sites.

### Limitations

The present system has three substantial limitations. The quality of the system relies on how accurately and quickly operators document the events, because most operational data is still entered manually. Second, the system is tailored to the workflow of the source laboratory, and thus any changes are likely to be required in laboratories that have distinctive shift patterns, cell counts, regulatory requirements, etc. before the system can be used. Third, the system primarily enables operational monitoring and reporting. It also calculates and presents KPIs, but features on prediction, anomaly detection, and automatic failure classification are not in place. Such limitations are tolerable

in the context of the Design Science Research methodology, a method in which the major focus is to address a defined operational problem before transitioning to a high-level analytics platform. Immediate success criterion is the resolution of a defined operational problem, rather than delivering a fully developed analytics platform as yet.

### REFERENCES

- [1] Xu, L.D., He, W., and Li, S. (2014). Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233–2243.
- [2] Zonta, T., da Costa, C.A., da Rosa Righi, R., de Lima, M.J., da Trindade, E.S., and Li, G.P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, 150, 106889.
- [3] Nunes, P., Santos, J., and Rocha, E. (2023). Challenges in predictive maintenance — A review. *CIRP Journal of Manufacturing Science and Technology*, 40, 53–67.
- [4] Mallioris, P., Aivazidou, E., and Bechtsis, D. (2024). Predictive maintenance in Industry 4.0: A systematic multi-sector mapping. *CIRP Journal of Manufacturing Science and Technology*, 50, 80–103.
- [5] Murtaza, Q., Haidri, R.A., Haleem, A., Javaid, M., Rab, S., and Suman, R. (2022). Applications of Machine Learning and Internet-of-Things (IoT) in Industry 5.0. *International Journal of Innovative Research in Science, Engineering and Technology*.
- [6] Friederich, J., and Lazarova-Molnar, S. (2023). Reliability assessment of manufacturing systems: A comprehensive overview, challenges, and opportunities. *Journal of Manufacturing Systems*, 69, 509–530.
- [7] Żyluk, A., Kuźma, K., Grzesik, N., Zieja, M., and Tomaszewska, J. (2021). Fuzzy Logic in Aircraft Onboard Systems Reliability Evaluation — A New Approach. *Sensors*, 21(23), 7913. MDPI.
- [8] Hassan, M.U., Rehmani, M.H., and Chen, J. (2019). Privacy Preserving in Blockchain Based on Partial Homomorphic Encryption: Challenges and Future Research Directions. *IEEE Access*, 8, 77–91.
- [9] Delvecchio, S., Bonfiglio, L., and Pompoli, F. (2018). Vibro-acoustic condition monitoring of Internal Combustion Engines: A critical review with focus on the application of non-invasive methodologies. *Mechanical Systems and Signal Processing*, 107, 509–536.
- [10] Nithin, S.K., Hemanth, K., and Shamanth, V. (2022). Condition monitoring of IC engine — A review. *Materials Today: Proceedings*, 54(1), 346–355.
- [11] Vilas Boas, J.W., Panziera, M., Marques, C.H.A., Panziera, T.M., Santos, R.P., and Aguiar, P.R. (2022). Condition monitoring of internal combustion engine based on statistical process control and Nelson rules. MDPI.
- [12] Torres, J., Jove, E., Casteleiro-Roca, J.L., Fernandez-Francos, J., Méndez-Pérez, J.A., and Calvo-Rolle, J.L. (2024). A systematic review of non-invasive methods for monitoring and fault detection in internal combustion engines. *MDPI Applied Sciences*.
- [13] Hevner, A., March, S., Park, J., and Ram, S. (2004). Design Science in Information Systems Research. *MIS*

Quarterly, 28(1), 75–105.