

Software Requirements Specification (SRS)

Project: Maintenance Information System (MIS)

Version: 1.0

Date: July 27, 2025

Author: Saroj Phalke ,Nidhi Sarwe, Shradha Sonawane

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the Maintenance Information System (MIS), a centralized Progressive Web Application (PWA) designed to manage, monitor, and maintain industrial assets. The system ensures compliance with IATF 16949 and VDA 6.3 standards.

1.2 Intended Audience

- Maintenance Engineers
- Plant Managers and Maintenance Heads
- Software Developers
- Compliance Auditors
- System Integrators

1.3 Intended Use

The system will be used to:

- Track assets and their maintenance schedules
- Record and analyze breakdowns
- Manage spare parts inventory
- Monitor utility usage
- Visualize key maintenance KPIs

1.4 Product Scope

The MIS system includes modules for:

- Asset Register: Master records of all machinery, tools, and support equipment.
- Preventive Maintenance: Frequency-based scheduling with alerts, checklists, and compliance tracking.
- Breakdown Maintenance: Logging, analysis, and reporting of machine faults including MTTR and MTBF.
- Spares Inventory: Stock levels, issuance, return, reorder management, and consumption

reporting.

- Utilities Monitoring : Logs and monitors power, gas, and water usage across major auxiliary systems.
- Tool & Mold Maintenance : Lifecycle management using usage counters like shot count or run-time meters.
- Dashboard & KPI Monitoring: : Real-time metrics including downtime trends, PM compliance, and spare cost analysis.

1.5 Definitions and Acronyms

- PWA – Progressive Web Application
- KPI – Key Performance Indicator
- MTTR – Mean Time to Repair
- MTBF – Mean Time Between Failures
- IATF – International Automotive Task Force
- VDA 6.3 – German Automotive Quality Standard
- RBAC – Role-Based Access Control

2. Overall Description

2.1 User Needs

- Operators require intuitive forms to log breakdowns.
- Engineers need real-time PM schedules and reports.
- Managers require dashboards and compliance summaries.
- Admins need access control, audit logs, and backups.

2.2 Assumptions and Dependencies

- PostgreSQL is the primary DBMS.
- The app is hosted on platforms like Vercel or Render.
- Supabase/Auth/Keycloak is used for authentication.
- Tailwind CSS is the preferred UI framework.

3. System Features and Requirements

3.1 Functional Requirements

3.1.1 Numbered and Described

FR-01: The system shall allow users to log breakdowns.

FR-02: The system shall store and track preventive maintenance records. The system shall trigger PM tasks based on schedules or equipment usage.

FR-03: The system shall issue and return spare parts with stock updates.

FR-04: The system shall display KPI dashboards for uptime, downtime, MTTR, MTBF, costs

and etc.

FR-05: The system shall manage user roles: operator, engineer, admin.

FR-06: The system shall allow users to scan QR codes on machines to fetch asset details.

3.1.2 EARS Format

- When a breakdown is logged, the system shall alert the engineer.
- When PM is due, the system shall notify the responsible person.
- When spare stock drops below reorder level, the system shall raise an alert. Also, when a spare part is issued, the system update the stock balance and reorder status.

3.1.3 Specification by Example (BDD - Gherkin)

Feature: Log Breakdown

Scenario: Operator logs a breakdown

Given the operator is logged in

When they submit a breakdown form

Then the system shall mark it for engineer review

3.2 Non-Functional Requirements

- Performance: 95% of requests shall return in under 2 seconds.
- Security: Authenticated access only; RBAC enforced.
- Usability: Mobile-first responsive design (PWA).
- Reliability: Weekly backups and failure alerts.
- Compliance: Adheres to IATF 16949 and VDA 6.3.

3.3 External Interface Requirements

- Performance: Dashboard refresh within 3 seconds.
- Safety: Validations on all input forms.
- Security: Encrypted password storage using bcrypt with authentication.
- Quality: 99.5% uptime with modular and scalable architecture.
- Business Rules: PM must be completed before due date. Breakdown must be acknowledged within the same shift.

3.4 System Features

- Asset Register with full metadata
- Preventive Maintenance Scheduler
- Breakdown Logging with MTTR/MTBF
- Spare Inventory with alerts
- Utilities Monitoring and Reports
- Dashboard with KPI charts and logs.
- Backup and recovery: Weekly automated PostgreSQL dump.

4. Other Requirements

4.1 Database Requirements

Uses PostgreSQL with core tables: assets_master, pm_schedule, spare_parts_inventory, breakdown logs, daily_entry, etc. Includes views for reporting.

4.2 Legal and Regulatory Requirements

Compliant with:

- IATF 16949
- VDA 6.3

4.3 Risk Management (FMEA Matrix)

Risk	Impact	Likelihood	Mitigation
DB crash	High	Medium	Automated backups
Auth bypass	High	Low	Secure RBAC and token expiration
Delay in PM	Medium	High	Email/SMS escalation

5. Appendices

5.1 Glossary

PM – Preventive Maintenance
RCA – Root Cause Analysis
KPI – Key Performance Indicator
RBAC – Role-Based Access Control

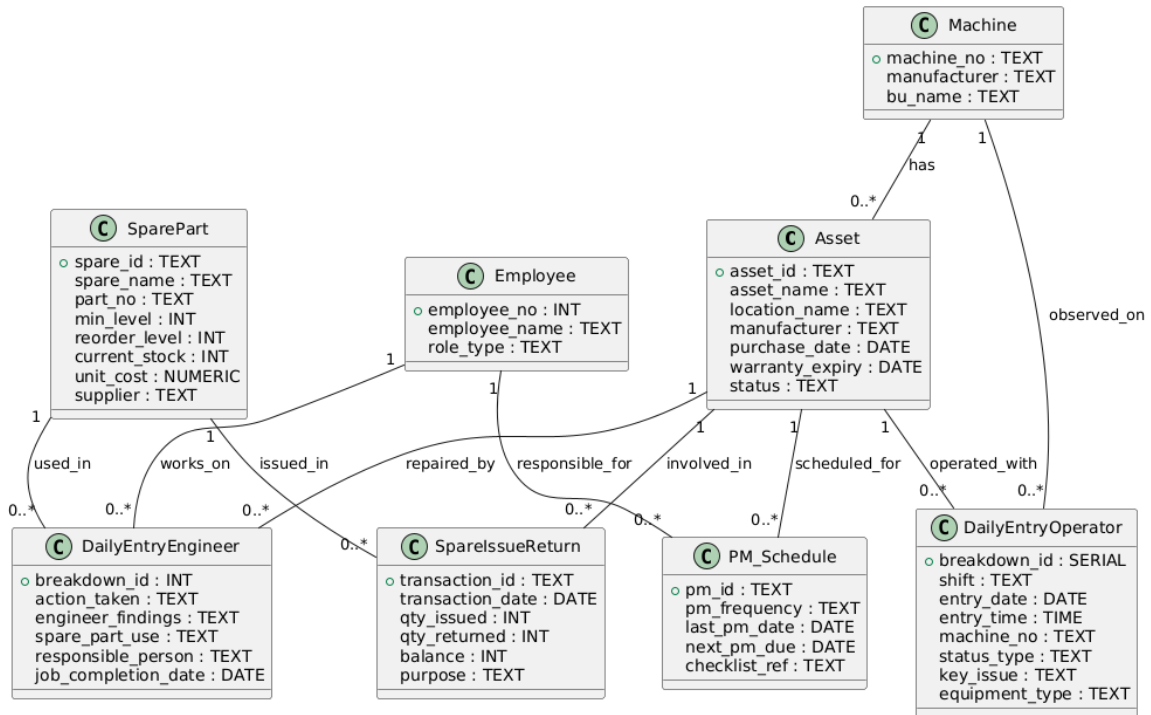
5.2 To Be Determined (TBD) List

- CI/CD Pipeline finalization
- Mobile App integration timeline
- IoT-based PM trigger roadmap

6. System Diagrams

6.1 Class Diagram

The class diagram illustrates the primary database entities and their relationships.

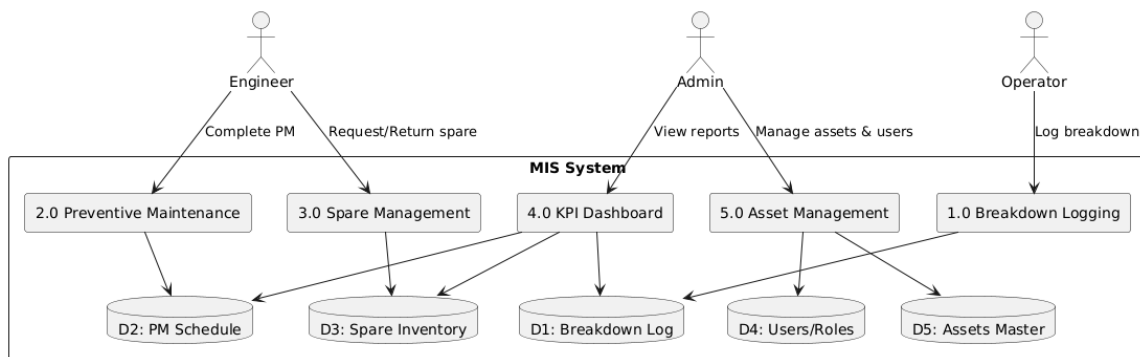


The daily_entry view can be obtained by joining daily_entry_operator table with daily_entry_engineer table on breakdown_id column.

The tools and mold maintenance table is currently not added to the database.

6.2 Data Flow Diagram (DFD)

The Data Flow Diagram (Level 1) shows how information moves across modules.



Text-Based DFD Description (Level 1)

External Entities (Actors):

- Operator
- Engineer
- Admin

Processes (System Modules):

- 1.0 Breakdown Logging
- 2.0 Preventive Maintenance
- 3.0 Spare Parts Management
- 4.0 KPI Dashboard & Reporting
- 5.0 Asset Management

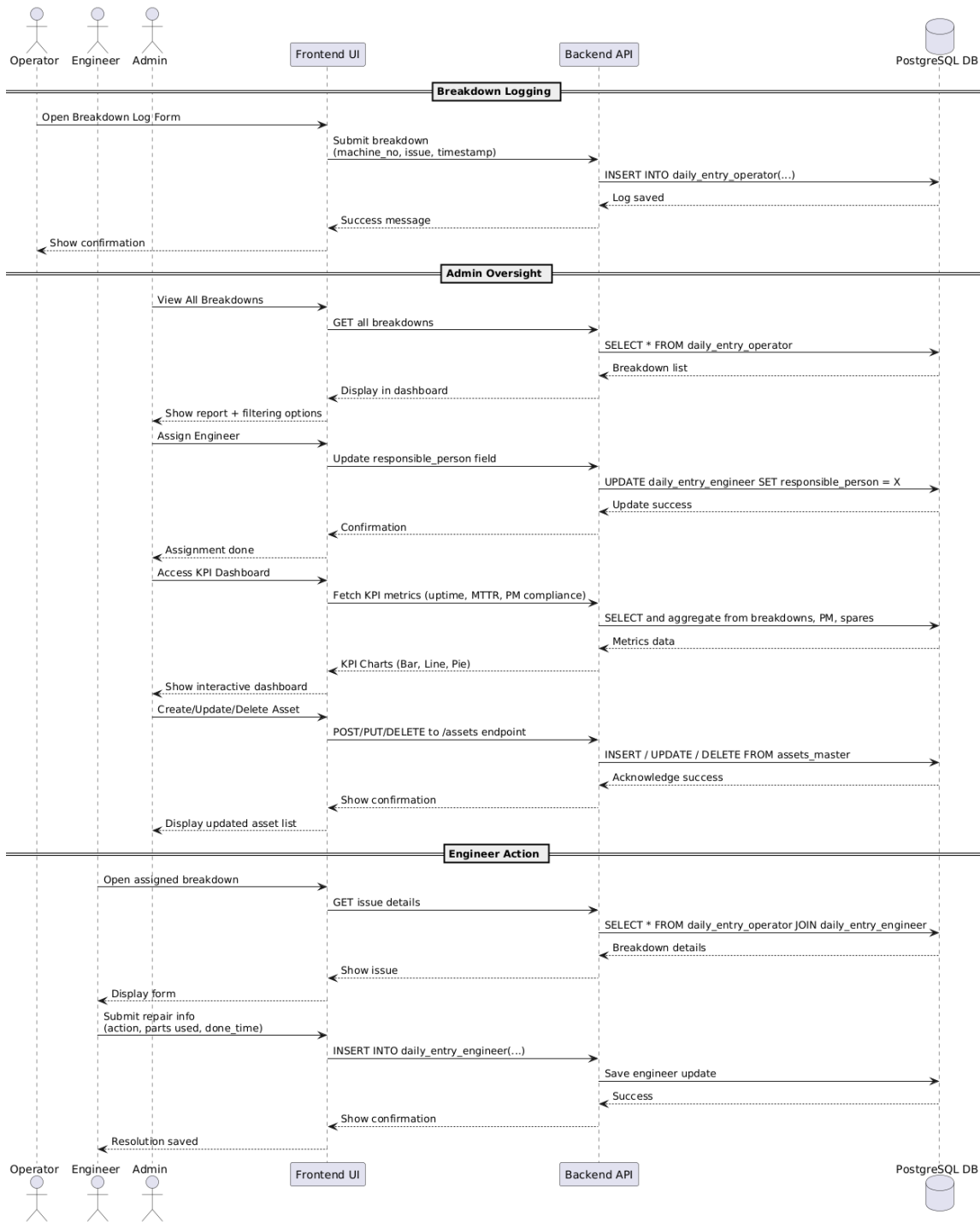
Data Stores:

- D1: Breakdown Log
- D2: PM Schedule
- D3: Spare Parts Inventory
- D4: Users and Roles
- D5: Assets Master

Actor	Process	Data Store
Operator	Logs breakdown	Breakdown Log (D1)
Engineer	Handles PM tasks	PM Schedule (D2)
Engineer	Uses spare parts	Spare Inventory (D3)
Admin	Manages system/master	Assets, Users (D4, D5)
Admin	Views KPIs & reports	D1, D2, D3

6.3 Sequence Diagram

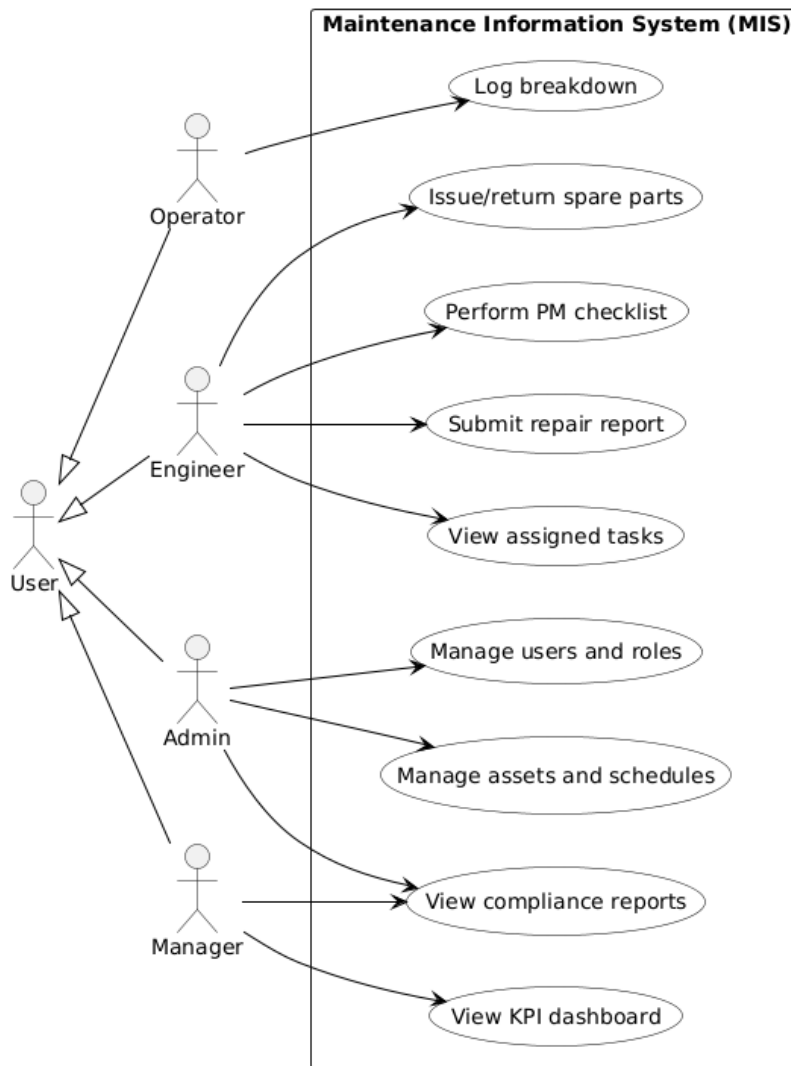
This sequence diagram shows how a breakdown logging operation flows between user, system, and database.



Flow	Actor	Modules Involved
Breakdown Logging	Operator	daily_entry_operator
Breakdown Resolution	Engineer	daily_entry_engineer

Flow	Actor	Modules Involved
Oversight and Assignment	Admin	all breakdown tables
KPI Dashboard Monitoring	Admin/Manager	Reporting layer KPi_log
Asset & Schedule CRUD	Admin	assets_master, pm_schedule, spare_parts_inventory, employee

6.4 Usecase Diagram



Actor Use Case

Operator Log Breakdown

Engineer View Tasks, Submit Repair, Perform PM

Admin Manage Assets, PM, Users, View Reports

Manager View KPI Dashboards and Compliance Data

7. Technology Stack and Architecture

7.1 Frontend (PWA)

- React + Vite – Fast and modular web UI
- Axios – REST API communication
- Tailwind CSS – Utility-first styling
- React Query – Data synchronization and caching
- PWA Support – Offline access(using PWA manifest and service workers) and push notifications.

7.2 Backend

- Node.js + Express – RESTful API backend
- PostgreSQL + Prisma – Scalable and type-safe database access
- bcrypt – Secure password hashing
- JWT – Stateless authentication
- dotenv – Environment configuration
- cors – CORS policy enforcement

7.3 DevOps & Hosting

- Vercel or Render – Hosting (free tiers supported)
- Supabase (optional) – Managed backend services
- GitHub Actions – Automation and CI/CD
- pg_dump/supabase inbuilt tool – Scheduled PostgreSQL backups

7.4 Optional & Future Enhancements

- Docker – Containerized deployment
- IoT Integration – Equipment sensors (Phase 2)
- AI/ML – Predictive maintenance (Phase 3)