SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS) FOR SMART MANUFACTURING ANALYTICS DASHBOARD

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to outline the software requirements for the Smart Manufacturing Analytics Dashboard. This project aims to develop an interactive dashboard that visualizes and aggregates manufacturing data to monitor key performance indicators (KPIs), identify bottlenecks, and optimize production efficiency.

1.2 Scope

The Smart Manufacturing Analytics Dashboard is designed for manufacturing companies to monitor real-time data, generate reports, and analyze production performance. The system will integrate with SQL Server for data storage and will allow users to interact with live and historical data through visual charts and graphs.

1.3 Definitions, Acronyms, and Abbreviations

KPI: Key Performance Indicator

SQL: Structured Query Language

SRS: Software Requirements Specification

IoT: Internet of Things

1.4 Overview

This document contains the system's functional and non-functional requirements, describes the key modules of the system, and provides models and diagrams to explain the system's flow and relationships.

2. System Description

2.1 Product Perspective

The Smart Manufacturing Analytics Dashboard will act as a web-based or desktop application that interacts with the manufacturing data stored in SQL Server. It will gather data from various production lines and machines, display them in an easy-to-understand format, and help decision-makers optimize production.

2.2 Product Features

- Real-time monitoring of production data.
- Data visualization through charts and graphs.
- Bottleneck detection and analysis.
- Alerts and notifications for production issues.
- Report generation and export in multiple formats.
- Integration with multiple data sources (IoT devices, ERP systems).
- User role management and permissions.
- Data storage in SQL Server.

2.3 User Classes and Characteristics

- Admin: Manages user roles, system settings, and data sources.
- Production Manager: Views production performance, generates reports, and tracks bottlenecks.
- Operator: Monitors live data and responds to alerts.

2.4 Operating Environment

- Software: SQL Server, WPF for UI, .NET Framework for backend, and possibly REST APIs for integration.
- Hardware: Standard desktop or laptop with internet connection for real-time updates.
- OS: Windows (for WPF desktop application).

2.5 Design and Implementation Constraints

- The system should be scalable to handle large data volumes.
- The application will need a connection to SQL Server for data storage.
- All communications with external systems should use secure protocols (HTTPS, SSL/TLS).

3. Functional Requirements

3.1 Home Dashboard

- Description: Displays an overview of KPIs and recent alerts.
- Inputs: None (data is auto-fetched).
- Outputs: Summary of production efficiency, downtime, and active alerts.
- Functionalities:
    - View current KPIs.
    - Check unresolved alerts.
    - Monitor real-time data feeds.

3.2 Data Visualization

- Description: Provides interactive charts and graphs for production data.
- Inputs: Time period, production line, metric type.
- Outputs: Graphs of production rate, efficiency, quality, and downtime.
- Functionalities:
    - Filter data by time, production line, and metric.
    - Generate trend graphs and comparison charts.

3.3 Production Performance Monitoring

- Description: Detailed metrics for each production line or machine.
- Inputs: Line ID, date range.
- Outputs: Production rate, efficiency, quality rate, downtime.
- Functionalities:
  - ➢ View production metrics by line.
  - ➢ Access historical performance data.

3.4 Bottleneck Analysis

- Description: Identifies production bottlenecks.
- Inputs: Line ID, performance thresholds.
- Outputs: Areas with the most delays and downtime.
- Functionalities:
  - ➢ Highlight bottlenecks in the production process.
  - ➢ Provide recommendations for improving flow.

3.5 Alerts and Notifications

- Description: Lists all alerts generated by the system.
- Inputs: None.
- Outputs: Alerts by severity, date, and type.
- Functionalities:
  - ➢ Mark alerts as resolved.
  - ➢ Filter alerts by severity and status.

3.6 Reporting and Analytics

- Description: Generates reports on production data.
- Inputs: Date range, production line, metrics.
- Outputs: Reports in PDF, Excel, or CSV format.
- Functionalities:

- ➢ Generate on-demand reports.
- ➢ Schedule automated report generation.

3.7 Integration Settings

- Description: Manages connections to external data sources (IoT devices, ERP).
- Inputs: API credentials, data refresh schedules.
- Outputs: Status of data source connections.
- Functionalities:
  - ➢ Connect or disconnect data sources.
  - ➢ Set up API integrations.

3.8 User Management

- Description: Administers user roles and permissions.
- Inputs: User details (name, role).
- Outputs: User list with roles.
- Functionalities:
  - ➢ Add, edit, or remove users.
  - ➢ Assign user roles (Admin, Manager, Operator).

3.9 Data Sources Overview

- Description: Lists all data sources connected to the system.
- Inputs: None.
- Outputs: Status of each data source.
- Functionalities:
  - ➢ View status and last sync time for data sources.
  - ➢ Manually refresh data.

3.10 Help and Support

- Description: Provides documentation and technical assistance.
- Inputs: Search query or contact form.
- Outputs: FAQs, troubleshooting guides.
- Functionalities:
  - Search for help articles.
  - Submit a support request.

4. Non-Functional Requirements

4.1 Performance Requirements

- The system should handle up to 1000 concurrent users with minimal lag.
- Data queries should execute within 5 seconds under normal load.

4.2 Security Requirements

- User data must be encrypted, both at rest and in transit.
- Authentication should be enforced via role-based access control.
- Data integrations with external sources must use secure APIs (HTTPS, OAuth).

4.3 Usability Requirements

- The dashboard must be intuitive and accessible to non-technical users.
- All reports and alerts should be easily understandable with minimal training.

4.4 Reliability and Availability

- The system should maintain 99.9% uptime.
- Data syncing should occur without data loss, even during downtimes or system restarts.

Glossary

KPI: Key Performance Indicator

IoT: Internet of Things

SQL: Structured Query Language

References

SQL Server Documentation

WPF User Interface Guidelines

.NET Framework Best Practices