Software Requirements Specification (SRS)

for

Retail and E-commerce: Demand Forecasting

Version 1.0

Team Members: Sachin Jadhav (202201040080)
Sahil Karne (202201040086)
Rohan Wagh (202201040090)
Viraj Mandlik (202201040102)
Vinay Karad (120200142)
Ajay Ingle (202302040021)

Date Created: 10/02/2025

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the software requirements for the **Retail and E-commerce: Demand Forecasting System**. This document outlines the functional, non-functional, and system requirements necessary for the successful implementation of the demand forecasting module within an inventory management system for retail and e-commerce businesses.

The project focuses on **demand forecasting using machine learning models** and an **inventory management system**, where inventory updates occur based on Estimated Delivery Time (ETA) provided by suppliers rather than manual stock updates. The system aims to optimize stock management, minimize stockouts and overstocking, and improve sales forecasting using **machine learning-based demand prediction models**.

This SRS serves as a reference for stakeholders, including developers, project managers, testers, and end-users, ensuring clarity and alignment on the system's objectives and functionalities.

1.2 Intended Audience and Reading Suggestions

This document is intended for the following stakeholders:

- **Developers** – To understand functional and non-functional requirements.
- **Project Managers** – To plan development timelines and allocate resources.
- **Data Scientists/ML Engineers** – To design and integrate predictive models for demand forecasting.
- **Retail Business Owners** – To understand system capabilities and benefits.
- **Testers & Quality Assurance (QA) Engineers** – To validate the system against the defined requirements.

Suggested Reading Order:

1. **Introduction** – Overview of the system.
2. **Overall Description** – Understanding the system’s purpose and user interaction.
3. **External Interface Requirements** – Interaction with users, hardware, and other software.
4. **System Features** – Core functionalities and capabilities.

5. **Non-Functional Requirements** – Performance, security, and quality attributes.
6. **Appendices** – Additional references and supporting materials.

1.3 Product Scope

The **Retail and E-commerce: Demand Forecasting System** is designed to predict future demand for products based on historical sales data, seasonal trends, and market conditions. The key objectives include:

- **Automated Demand Forecasting** – Utilizing machine learning models to predict sales and demand patterns.
- **Optimized Inventory Management** – Ensuring stock availability while reducing overstocking.
- **Supplier ETA Integration** – Automatically updating stock based on expected arrival times.
- **Data-Driven Decision Making** – Providing insights through dashboards and reports.
- **Scalability & Performance** – Supporting large datasets and real-time updates.

1.4 References

The following documents and resources have been referred to while preparing this SRS:

1. IEEE Std 830-1998 – Recommended Practice for Software Requirements Specifications.

2. K. Williams, "Agile methodologies in retail software development: A case study," Journal of Software Engineering for Retail Solutions, vol. 10, no. 2, pp. 34-42, Feb. 2025.

3. S. J. Smith, "A study on demand forecasting models in retail and eCommerce," Journal of Retail Analytics, vol. 25, no. 3, pp. 123-134, Mar. 2023.

4. M. R. Patel and L. A. Jones, "The impact of machine learning algorithms on demand forecasting accuracy," International Journal of Machine Learning and Retailing, vol. 18, no. 1, pp. 45-58, Jan. 2024.

5. R. Kumar, P. P. Desai, and S. Sharma, "Optimization of eCommerce inventory through predictive demand forecasting," IEEE Transactions on E-Commerce and Retailing, vol. 15, no. 4, pp. 202-215, Apr. 2024.

2. Overall Description

This section provides an overview of the Demand Forecasting System for the Retail and E-commerce domain. It describes the product's context, primary functions, intended users, operating environment, constraints, documentation, and dependencies.

2.1 Product Perspective

The **Demand Forecasting System** is an essential module within a larger **Retail and E-commerce Management System**, designed to predict inventory demand based on historical data and market trends. It aims to improve inventory management by reducing stockouts and overstock situations.

This system is a standalone module but can be integrated with existing enterprise resource planning (ERP) systems or inventory management software through APIs. The system employs **machine learning algorithms** to analyze past sales data, seasonal trends, and external factors to provide accurate demand forecasts.

System Context and Components

The Demand Forecasting System consists of the following key components:

- **Data Ingestion Module**: Collects data from various sources such as past sales records, supplier delivery timelines, and market trends.
- **Data Processing and Feature Engineering**: Cleans, transforms, and selects the most relevant data features for forecasting.
- **Forecasting Engine**: Uses machine learning models such as ARIMA, Prophet, LSTMs, or XGBoost to generate demand predictions.
- **User Interface**: A dashboard that visualizes demand forecasts, alerts, and recommended stock levels.
- **Integration APIs**: Interfaces for suppliers, inventory databases, and ERP systems to share and update demand data.

2.2 Product Functions

The Demand Forecasting System provides the following major functions:

- **Historical Data Analysis**: Processes and cleans past sales data, detecting trends, seasonality, and anomalies.

- **Forecast Generation**: Uses machine learning models to predict future demand for various products.
- **Automated Inventory Updates**: Updates stock levels based on supplier **Estimated Delivery Time (ETA)** and predicted demand.
- **Supplier Integration**: Allows suppliers to input ETAs for better inventory tracking.
- **Visualization and Reporting**: Provides dashboards with graphical representations of demand trends.
- **Alerts and Notifications**: Sends alerts when inventory levels are projected to be too high or too low.
- **API Support**: Enables integration with third-party ERP and inventory management systems.

2.3 User Classes and Characteristics

The system will be used by different types of users, each with specific requirements and access levels:

User Class	Description	Access Level
Retail Managers	Monitor stock levels, review demand forecasts, and place orders accordingly.	Full Access
Suppliers	Provide estimated delivery times and stock availability.	Limited Access
Inventory Analysts	Analyze forecast accuracy and optimize stock levels.	Full Access
System Administrators	Manage users, permissions, and system configurations.	Admin Access

2.4 Operating Environment

The system will be deployed as a **web-based application** hosted on a cloud platform. The key components of the operating environment include:

- **Hardware**:
  - Server: Cloud-based (AWS, Azure, or GCP)
  - Minimum client-side requirements: 4GB RAM, multi-core processor
- **Software**:
  - Operating System: Windows/Linux/macOS
  - Backend: Python (Flask/Django)
  - Frontend: React.js / Angular
  - Database: PostgreSQL / MySQL
  - ML Libraries: TensorFlow, Scikit-Learn, Prophet

2.5 Design and Implementation Constraints

The development of the Demand Forecasting System is subject to the following constraints:

- **Real-Time Processing**: The system should provide near real-time forecasting updates with minimal latency.
- **Scalability**: The system should handle large datasets (millions of records) without significant performance degradation.
- **Data Security**: The system must comply with **GDPR** and **ISO 27001** security standards.
- **Integration Restrictions**: Some third-party ERP and inventory systems may have proprietary APIs, limiting direct integration.
- **Computational Resource Constraints**: The choice of ML models should balance accuracy and computational cost to avoid excessive cloud computing expenses.

2.6 User Documentation

The following user documentation will be provided:

- **User Guide**: Explains system functionalities, forecasting results, and data visualization tools.
- **Administrator Manual**: Covers system configuration, model training, and user management.

- **API Documentation**: Provides details on integration endpoints for external ERP systems and suppliers.
- **Troubleshooting Guide**: Lists common issues and solutions related to demand forecasting.

2.7 Assumptions and Dependencies

Assumptions

- The historical sales data provided is clean, structured, and free of missing values.
- Supplier ETAs are accurate and updated in real time.
- The system will be accessed primarily through web browsers (Chrome, Firefox, Edge).
- Retailers will integrate the system into their existing inventory management workflows.
- External market factors (festivals, promotions, or economic conditions) can be incorporated into the model.

Dependencies

- **Data Sources**: The accuracy of demand forecasting relies on access to **real-time sales data** and **supplier updates**.
- **Machine Learning Models**: The system depends on **pre-trained models** that require periodic retraining with updated datasets.
- **Cloud Services**: The system will utilize **cloud computing** for data storage and processing (e.g., AWS S3, Azure Blob Storage).
- **Third-Party APIs**: The integration with ERP systems and supplier platforms depends on available APIs and their stability.

3. External Interface Requirements

This section describes the various interfaces that the **Retail and E-commerce: Demand Forecasting System** will have with users, hardware, software, and communication systems. The system will be a web-based application that interacts with users, databases, and external APIs for inventory management and demand forecasting.

3.1 User Interfaces

The system will have an interactive web-based user interface for different stakeholders, including **retailers, suppliers, and administrators**. The interface will be designed for ease of use, efficiency, and clarity.

Key Features of the User Interface:

- **Dashboard:** Provides an overview of inventory levels, forecasted demand, and supplier updates.
- **Inventory Management Panel:** Allows retailers to view, update, and manage stock.
- **Demand Forecasting Reports:** Displays predicted demand trends using graphical and tabular formats.
- **Supplier Management Module:** Allows suppliers to provide inventory restocking updates and estimated delivery times (ETA).
- **Notification System:** Alerts users about low stock levels, delayed deliveries, and significant demand shifts.

UI Design Considerations:

- **Responsive Design:** Compatible with desktops, tablets, and mobile devices.
- **Consistent Layout:** Follows industry UI/UX standards for better usability.
- **Standard Buttons & Functions:** Includes common navigation options like *Home, Dashboard, Reports, Notifications, and Settings*.
- **Keyboard Shortcuts:** Supports keyboard-based navigation for efficient data entry.
- **Error Handling & Alerts:** Displays user-friendly error messages and warnings.

3.2 Hardware Interfaces

The system will interact with the hardware components necessary for efficient functioning. These include:

Client-Side Hardware Requirements

- **Device Compatibility:** The system should run on standard desktop PCs, laptops, tablets, and smartphones.
- **Recommended Specifications:**
    - Processor: Minimum **Intel i5 / AMD Ryzen 5**
    - RAM: **8GB or higher**
    - Storage: **At least 10GB free space**
    - Display: **1280x720 resolution or higher**

Server-Side Hardware Requirements

- **Cloud Server / On-Premise Server:** The system will be deployed on cloud infrastructure (e.g., AWS, Azure, or Google Cloud) or an on-premise server.
- **Recommended Specifications:**
    - Processor: **Intel Xeon / AMD EPYC** (multi-core processing for ML computations)
    - RAM: **16GB or higher**
    - Storage: **500GB SSD or more**
    - Network: **1 Gbps Ethernet**

Peripheral Hardware Integration

- **Barcode Scanners (Optional):** Used for quick stock updates.
- **RFID Readers (Optional):** For automated inventory tracking.
- **Thermal Printers (Optional):** For generating inventory reports and invoices.

3.3 Software Interfaces

The system interacts with multiple software components, including operating systems, databases, third-party APIs, and libraries.

Operating System Compatibility

- Client-side: Windows, macOS, Linux, Android, iOS (for mobile users).
- Server-side: Linux (Ubuntu, CentOS) or Windows Server.

Database System

- **Database Type:** Relational Database Management System (RDBMS)
- **Recommended Databases:** PostgreSQL / MySQL / MongoDB
- **Data Stored:**
    - Inventory data
    - Sales history
    - Supplier updates
    - Demand forecasts
    - User authentication and access control

Third-Party APIs & Libraries

The system will integrate with external APIs for real-time updates, ML processing, and notifications.

API/Library	Purpose
Google Firebase	Real-time notifications & authentication
OpenWeather API	Weather-based demand forecasting
Pandas & NumPy	Data processing for ML models
Scikit-learn	Machine learning for demand forecasting
TensorFlow / PyTorch	Deep learning models for advanced forecasting
Twilio API	SMS notifications for inventory alerts
Email API (SMTP/Gmail API)	Automated email notifications

Web Technologies

- **Frontend:** React.js / Vue.js / Angular
- **Backend:** Django (Python) / Flask / FastAPI
- **ML Model Deployment:** TensorFlow Serving / Flask API for ML predictions

3.4 Communications Interfaces

The system must communicate securely with databases, users, and external APIs. The communication protocols and security measures are defined below.

Communication Methods

- **Client-Server Communication:**
  - **Protocol:** HTTPS (Secure Communication)
  - **Data Format:** JSON / XML
  - **Authentication:** JWT (JSON Web Tokens) for secure user sessions
- **Database Communication:**
  - **Protocol:** TCP/IP
  - **Encryption:** AES-256 for sensitive data
  - **Backup Mechanism:** Daily automated database backups
- **Supplier Communication:**
  - **Protocol:** RESTful API
  - **Data Exchange Format:** JSON
  - **API Authentication:** OAuth2.0 for secure access
- **Notifications & Alerts:**
  - **Email Notifications:** SMTP, secured with TLS/SSL
  - **SMS Alerts:** Twilio API (or similar)
  - **Web Push Notifications:** Firebase Cloud Messaging (FCM)
- **Logging & Monitoring:**
  - **Application Logs:** Stored in cloud storage (AWS S3, Azure Blob Storage)
  - **Error Tracking:** Integrated with Sentry / Logstash
  - **System Health Monitoring:** Prometheus / Grafana for real-time monitoring

4. System Features

This section details the major system features of the **Retail and E-commerce Demand Forecasting System**. Each feature is described along with its priority, stimulus/response sequences, and associated functional requirements.

4.1 Demand Forecasting

4.1.1 Description and Priority

The system predicts future demand for products based on historical sales data, seasonal trends, and external factors. This feature is crucial for optimizing inventory levels and minimizing stockouts or overstocking.

- **Priority:** High

4.1.2 Stimulus/Response Sequences

1. The system retrieves historical sales data, product details, and any external factors affecting demand.
2. The forecasting model processes this data and generates predictions for future demand.
3. The results are displayed in a dashboard with graphical trends and numeric forecasts.
4. Users can adjust forecasting parameters (e.g., time frame, product category).
5. The system updates demand forecasts periodically and notifies users of significant changes.

4.1.3 Functional Requirements

- **REQ-1:** The system shall retrieve historical sales data from the database for forecasting.
- **REQ-2:** The system shall support demand forecasting at daily, weekly, and monthly intervals.
- **REQ-3:** The system shall apply machine learning models (e.g., ARIMA, LSTM, Prophet) for accurate predictions.
- **REQ-4:** The system shall display graphical representations (line charts, bar graphs) of predicted demand trends.
- **REQ-5:** The system shall allow users to fine-tune forecasting parameters.
- **REQ-6:** The system shall provide real-time updates if significant demand variations are detected.

4.2 Inventory Management and Automated Updates

4.2.1 Description and Priority

The system tracks product inventory levels and updates stock automatically based on supplier-provided Estimated Delivery Time (ETA).

- **Priority:** High

4.2.2 Stimulus/Response Sequences

1. The system tracks product stock levels in real-time.
2. When stock reaches a predefined threshold, an automated order request is triggered.
3. Suppliers provide an ETA for product replenishment.
4. The system updates stock levels automatically when the ETA is reached.
5. Users can manually adjust stock levels if necessary.

4.2.3 Functional Requirements

- **REQ-1:** The system shall maintain a real-time inventory database.
- **REQ-2:** The system shall allow suppliers to input an ETA for product deliveries.
- **REQ-3:** The system shall automatically update stock levels when the ETA is reached.
- **REQ-4:** The system shall notify users if stock falls below a critical threshold.
- **REQ-5:** The system shall allow manual stock adjustments by authorized users.

4.3 Supplier and Product Data Integration

4.3.1 Description and Priority

The system integrates supplier and product data for accurate inventory tracking and demand forecasting.

- **Priority:** Medium

4.3.2 Stimulus/Response Sequences

1. Suppliers update product availability and expected delivery timelines.
2. The system syncs supplier data with inventory records.
3. If supplier delays occur, the system recalculates inventory needs and notifies users.

4.3.3 Functional Requirements

- **REQ-1:** The system shall allow suppliers to update delivery schedules.
- **REQ-2:** The system shall store and sync supplier data with inventory records.
- **REQ-3:** The system shall notify users of supplier delays affecting stock levels.

4.4 Analytics and Reporting

4.4.1 Description and Priority

The system provides insights into sales trends, stock turnover, and demand forecasts through reports and dashboards.

- **Priority:** High

4.4.2 Stimulus/Response Sequences

1. Users select a reporting period and relevant data points (e.g., product category, region).
2. The system retrieves and processes the requested data.
3. Graphical and tabular reports are generated and displayed.
4. Users can export reports as PDFs or CSV files.

4.4.3 Functional Requirements

- **REQ-1:** The system shall generate reports on demand forecasts, sales trends, and inventory turnover.
- **REQ-2:** The system shall display data in charts, tables, and graphs.
- **REQ-3:** The system shall allow users to export reports in CSV and PDF formats.

4.5 User Management and Access Control

4.5.1 Description and Priority

The system provides role-based access control to ensure data security.

- **Priority:** High

4.5.2 Stimulus/Response Sequences

1. Users log in with credentials.

2. The system verifies access permissions based on user roles.
3. Users access only authorized features and data.

4.5.3 Functional Requirements

- **REQ-1:** The system shall support multiple user roles (admin, supplier, analyst).
- **REQ-2:** The system shall restrict access to sensitive data based on user roles.
- **REQ-3:** The system shall log user activities for security audits.

5. Other Nonfunctional Requirements

This section outlines the nonfunctional requirements of the **Retail and E-commerce: Demand Forecasting System** to ensure optimal performance, security, and maintainability.

5.1 Performance Requirements

- The system should be able to handle a minimum of **100 concurrent users** without significant performance degradation.
- Forecasting models should process new demand data and generate updated predictions within **30 seconds** for a dataset of **100,000 records**.
- The system should support a **response time of less than 2 seconds** for user interactions, such as viewing demand forecasts, reports, or analytics dashboards.
- The database should be optimized to handle at least **500,000 inventory records** and **1 million transactions** efficiently.
- The system should be capable of running batch forecasting jobs on **historical sales data (spanning up to 5 years) within 10 minutes**.
- The demand forecasting API should return predictions within **500 milliseconds** for real-time requests.

5.2 Safety Requirements

- The system must ensure **data integrity** by preventing accidental data loss or corruption.
- Regular **automated backups** of critical data should be taken every **24 hours**, and backup recovery should be **tested monthly**.
- The system should prevent **unauthorized data modifications** by enforcing proper access control mechanisms.
- If a machine learning model fails to generate a forecast, the system should **fallback to the last known prediction** and log the error for further analysis.
- The system should have an **automated rollback mechanism** to revert to a stable version in case of failure during updates.

5.3 Security Requirements

- Users must authenticate using a **secure login system** (e.g., OAuth 2.0, JWT-based authentication).
- Role-based access control (RBAC) should be enforced:
    - **Admins**: Full access to system configurations and reports.
    - **Suppliers**: Limited access to inventory updates and delivery schedules.
    - **Retailers**: Access to demand forecasts and inventory status.
- All data must be encrypted using **AES-256** for stored data and **TLS 1.3** for data in transit.
- The system should log all user activities related to data access and modifications for **audit purposes**.
- If a user account detects **5 failed login attempts within 10 minutes**, the account should be **temporarily locked** for **30 minutes** and notify the user.
- The system should comply with **GDPR and industry standards** for handling user and transaction data.

5.4 Software Quality Attributes

- **Reliability**: The system should have an uptime of **99.9%** with redundancy in case of failures.
- **Scalability**: The system should support **scaling up to 500,000 inventory items and 1 million transactions** without performance degradation.
- **Maintainability**: Code should follow **modular architecture** with documentation for each module, ensuring ease of updates and debugging.
- **Interoperability**: The system should expose RESTful APIs to **integrate with external inventory management and e-commerce platforms**.
- **Usability**: The UI should be intuitive, supporting mobile and desktop views with **a minimal learning curve**.
- **Extensibility**: The system should allow for **future model upgrades**, such as incorporating deep learning-based forecasting.
- **Testability**: The system should include **unit tests (90% coverage), integration tests, and load testing** to ensure stability.

5.5 Business Rules

- **Stock Updates**: Inventory levels should be **automatically updated** when the Estimated Delivery Time (ETA) is reached.
- **Reorder Threshold**: If forecasted demand exceeds current stock by **20%**, a **restock alert** should be triggered.
- **Supplier ETA Compliance**: If a supplier fails to deliver stock **3 times consecutively**, they should be flagged for **review by the admin**.
- **Historical Data Retention**: Demand forecasting should be based on **at least 2 years of historical sales data**.
- **Report Generation**: The system should generate **weekly and monthly sales forecast reports** that can be exported in **CSV, PDF, and Excel** formats.
- **System Notifications**: Users should receive alerts via **email and dashboard notifications** for low stock levels and forecast updates.

6. Other Requirements

This section outlines additional requirements that are crucial for the **Retail and E-commerce: Demand Forecasting** system but are not covered in the previous sections. These include database requirements, internationalization, legal and compliance aspects, and system scalability considerations.

6.1 Database Requirements

The system relies on a well-structured and optimized database to store, process, and retrieve data efficiently. The following database requirements must be fulfilled:

- **Database Type**: A **relational database** (e.g., **PostgreSQL, MySQL, or SQLite**) will be used to store transactional data, demand forecasting models, and supplier details. NoSQL databases (e.g., **MongoDB**) may be used for storing unstructured data like logs.
- **Data Storage**: The system must store:
    - Historical sales data (at least 2 years for training ML models)
    - Inventory records with supplier Estimated Delivery Time (ETA)
    - Product details (SKU, category, supplier, pricing)
    - User roles and access control data
    - Forecasting results and accuracy metrics
    - Supplier-provided estimated delivery updates
- **Data Access**: The database must support role-based access control (RBAC) to ensure only authorized personnel (admins, suppliers, store managers) can modify or view relevant data.
- **Data Integrity**: The database must enforce constraints like primary keys, foreign keys, and indexing for efficient querying.
- **Backup and Recovery**: Automated daily backups must be configured to prevent data loss in case of failure.

6.2 Internationalization and Localization

The system should be designed to support multiple regions and languages for future scalability:

- **Currency Support**: The system must allow pricing and forecasting calculations to be done in different currencies (e.g., USD, EUR, INR).
- **Date and Time Format**: The system should handle different formats based on regional preferences.

- **Language Support**: Initially, the system will support English. It should have a framework that allows adding other languages later.
- **Timezone Handling**: The Estimated Delivery Time (ETA) should be converted to the user's local timezone when displayed.

6.3 Legal and Compliance Requirements

The system must adhere to industry standards and legal requirements:

- **Data Privacy**:
    - The system must comply with **GDPR (General Data Protection Regulation)** and **CCPA (California Consumer Privacy Act)** if deployed internationally.
    - Personally Identifiable Information (PII) such as user credentials must be encrypted.
- **Data Retention Policy**:
    - Sales and inventory data should be retained for **5 years** for analytics and auditing.
    - Supplier-related data should be retained indefinitely unless a supplier requests deletion.
- **E-commerce Compliance**:
    - The system must comply with e-commerce laws such as **Consumer Protection Act, 2019 (India)** and **FTC guidelines (USA)** if integrated into a full retail system in the future.
- **Accessibility Standards**:
    - The system should comply with **WCAG 2.1 (Web Content Accessibility Guidelines)** to ensure usability for people with disabilities.

6.4 Scalability and Performance Considerations

As the system grows, it must be able to handle increasing data loads without significant performance degradation.

- **Data Scalability**:
    - The system must be able to handle **millions of inventory records** and **high-frequency forecasting calculations**.
    - Demand forecasting models must be designed to handle increasing datasets efficiently (e.g., batch processing for historical data, streaming for real-time predictions).
- **High Availability**:

    - The system should support load balancing to distribute requests evenly across multiple servers.
    - It should have **auto-scaling mechanisms** to allocate more resources when traffic increases.
- **Response Time**:
    - Forecast generation should not exceed **5 seconds** for a standard query.
    - Inventory updates should be processed **in real-time or within 2 seconds** of receiving supplier updates.
- **Cloud Deployment**:
    - The system should be **cloud-ready** (e.g., AWS, Google Cloud, Azure) for easy deployment and scaling.

6.5 Integration with External Systems

The system should be designed to integrate seamlessly with other platforms:

- **Supplier API Integration**:
    - The system must support integration with supplier APIs to fetch Estimated Delivery Time (ETA) automatically.
- **Point of Sale (POS) Integration**:
    - If required, the system should connect with POS systems to sync sales data for demand forecasting.
- **E-commerce Platform Compatibility**:
    - The system should have the ability to integrate with e-commerce platforms such as Shopify, Magento, or WooCommerce in the future.
- **Payment Gateway (Optional for Future Scope)**:
    - If the system is expanded to include direct retail functionalities, it must integrate with payment gateways like Stripe, Razorpay, or PayPal.

6.6 Logging and Monitoring

The system should have a robust logging and monitoring framework:

- **Error Logging**:
    - All errors, warnings, and exceptions must be logged for debugging purposes.
- **Audit Logs**:
    - A record of all inventory updates, demand forecasting model changes, and supplier interactions should be maintained.
- **Performance Monitoring**:

    - Real-time monitoring tools (e.g., **Prometheus, Grafana**) should be used to track system health and performance.
- **Alert System**:
    - If demand forecasting accuracy falls below **80%**, or if inventory reaches critical low levels, an alert should be triggered to administrators.