Inventory Management System (IMS)

---

1. Preface

This document specifies the **Software Requirements Specification (SRS)** for the **Inventory Management System (IMS)**.
It is intended to provide a comprehensive description of system functionality, constraints, and behavior for all stakeholders including system users, developers, testers, and evaluators to understand what the system should do.

The document serves as a baseline agreement between stakeholders and the development team regarding system expectations and deliverables.

---

2. Introduction

2.1 Purpose

The purpose of this document is to define the **functional and non-functional requirements** of the Inventory Management System based on stakeholder needs and system modeling using **use-case diagrams, activity diagrams, and sequence diagrams**.

This SRS ensures clarity, consistency, and traceability across system design, implementation, and validation phases.

2.2 Scope

The Inventory Management System (IMS) is a **centralized, analytics-driven platform** designed to support retailers in managing:

- Product catalogs
- Inventory levels
- Supplier coordination
- Consumer availability
- Multi-store inventory operations

The system integrates multiple stakeholders and external POS systems to ensure **real-time stock visibility**, **optimized replenishment**, and **efficient supply chain operations**.

IMS is not responsible for low-level POS hardware control but relies on standardized interfaces for data synchronization.

2.3 Definitions, Acronyms, and Abbreviations

Term	Description
IMS	Inventory Management System
Retailer	Business entity managing inventory and sales
Supplier	Entity responsible for product supply and fulfillment
Consumer	End user checking availability and providing feedback
POS	Point of Sale system
Admin	System administrator with governance privileges
Stockout	Situation where inventory reaches zero
Reorder Point	Threshold triggering replenishment

---

3. Overall Description

3.1 System Users

The system supports the following user roles:

- Retailer
- Supplier
- Consumer
- System Administrator
- POS System (External)

3.2 Product Perspective

IMS is a **web-based centralized system** that integrates:

- Inventory tracking
- Supplier management
- Consumer interaction
- POS data synchronization
- Analytics and reporting

The system follows a modular architecture with clearly defined role-based access control and supports horizontal scaling for multi-store deployments.

3.3 Operating Environment

- Web Browser (Chrome, Edge, Firefox)
- Backend Server (Cloud or On-Premise)
- Relational Database (MySQL / PostgreSQL/MongoDB)
- REST-based POS Integration APIs

3.4 Design and Implementation Constraints

- POS data depends on third-party API availability
- Real-time updates require stable network connectivity
- Data consistency must be preserved across multiple store locations

---

3.5 Assumptions and Dependencies

- Retailers maintain accurate POS configurations

- Suppliers provide timely fulfillment updates
- POS systems push transactional data reliably

---

4. User Requirements (High-Level)

ID	User Requirement
UR1	The system shall allow retailers to manage products and inventory
UR2	The system shall provide real-time stock visibility
UR3	The system shall generate intelligent reorder recommendations
UR4	The system shall support multi-store inventory management
UR5	The system shall integrate with POS systems
UR6	The system shall allow suppliers to manage fulfillment
UR7	The system shall notify consumers of stock availability
UR8	The system shall provide analytics and audit reports

---

5. Functional Requirements

5.1 Retailer Functions

- **FR1:** The system shall allow retailers to add, update, and remove product details.
- **FR2:** The system shall maintain real-time inventory levels per product and store.
- **FR3:** The system shall track inventory movement from procurement to sale and return.
- **FR4:** The system shall recommend reorder quantities based on demand trends.
- **FR5:** The system shall forecast future demand using historical sales data.
- **FR6:** The system shall support centralized inventory management across multiple stores.

5.2 Supplier Functions

- **FR7:** The system shall allow suppliers to confirm order fulfillment.
- **FR8:** The system shall track supplier delivery status in real time.
- **FR9:** The system shall evaluate supplier performance based on delivery metrics.
- **FR10:** The system shall allow suppliers to manage delivery schedules and capacity.

5.3 Consumer Functions

- **FR11:** The system shall allow consumers to view real-time product availability.
- **FR12:** The system shall notify consumers when out-of-stock products are restocked.
- **FR13:** The system shall allow consumers to submit product feedback.

5.4 Administrator Functions

- **FR14:** The system shall allow administrators to configure multi-store structures.
- **FR15:** The system shall manage user roles and access permissions.
- **FR16:** The system shall control global system configurations.
- **FR17:** The system shall generate system-wide analytics and audit reports.

5.5 POS Integration Functions

- **FR18:** The system shall automatically update inventory based on POS sales data.
- **FR19:** The system shall ensure synchronization consistency between POS and inventory records.

---

6. Non-Functional Requirements

6.1 Performance Requirements

- **NFR1:** The system shall respond to user actions within 3 seconds.
- **NFR2:** Inventory updates from POS shall reflect within 2 seconds.

6.2 Security Requirements

- **NFR3:** The system shall enforce role-based access control.
- **NFR4:** Only authenticated users shall access system functions.

6.3 Reliability Requirements

- **NFR5:** The system shall ensure data consistency across stores.
- **NFR6:** The system shall recover gracefully from system failures.

6.4 Usability Requirements

- **NFR7:** The system shall provide a simple and intuitive interface.
- **NFR8:** Minimal training shall be required for system usage.

6.5 Scalability Requirements

- **NFR9:** The system shall support onboarding of additional stores without downtime.

---

7. System Models

The system is modeled using:

- Use Case Diagrams – Actor interactions
- Activity Diagrams – Workflow logic
- Sequence Diagrams – Inventory lifecycle interactions

---

8. System Evolution

8.1 Anticipated Enhancements

- AI-driven demand prediction
- Automated supplier ranking
- Mobile application support
- E-commerce platform integration

---

9. Appendices

9.1 Hardware Requirements

- Server: Minimum 8-core CPU, 32GB RAM, 1TB SSD
- Client: Standard web-enabled devices

9.2 Database Requirements

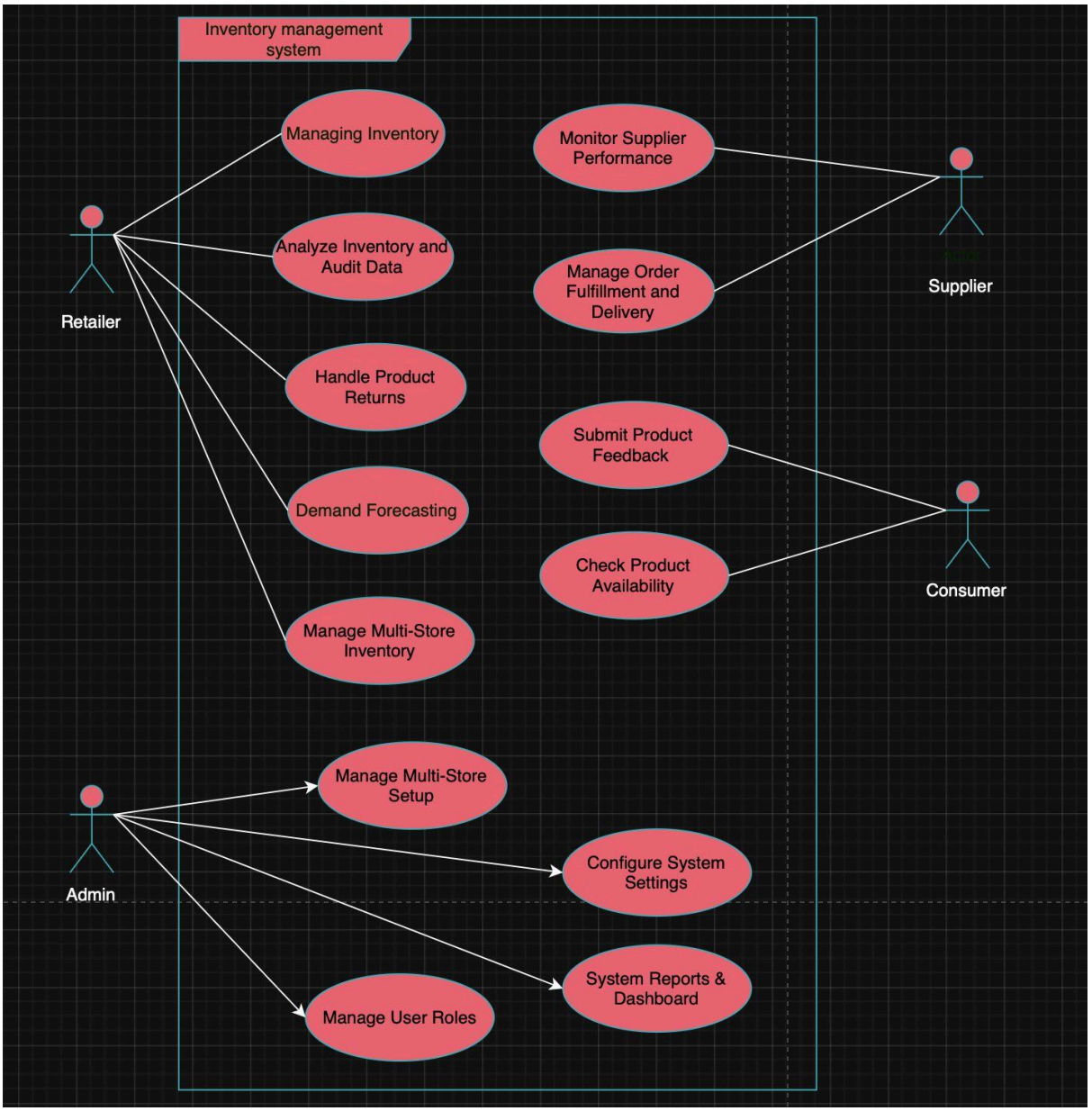
- Persistent storage of inventory, sales, and audit logs
- Historical data retention for analytics and forecasting

---

10. Conclusion

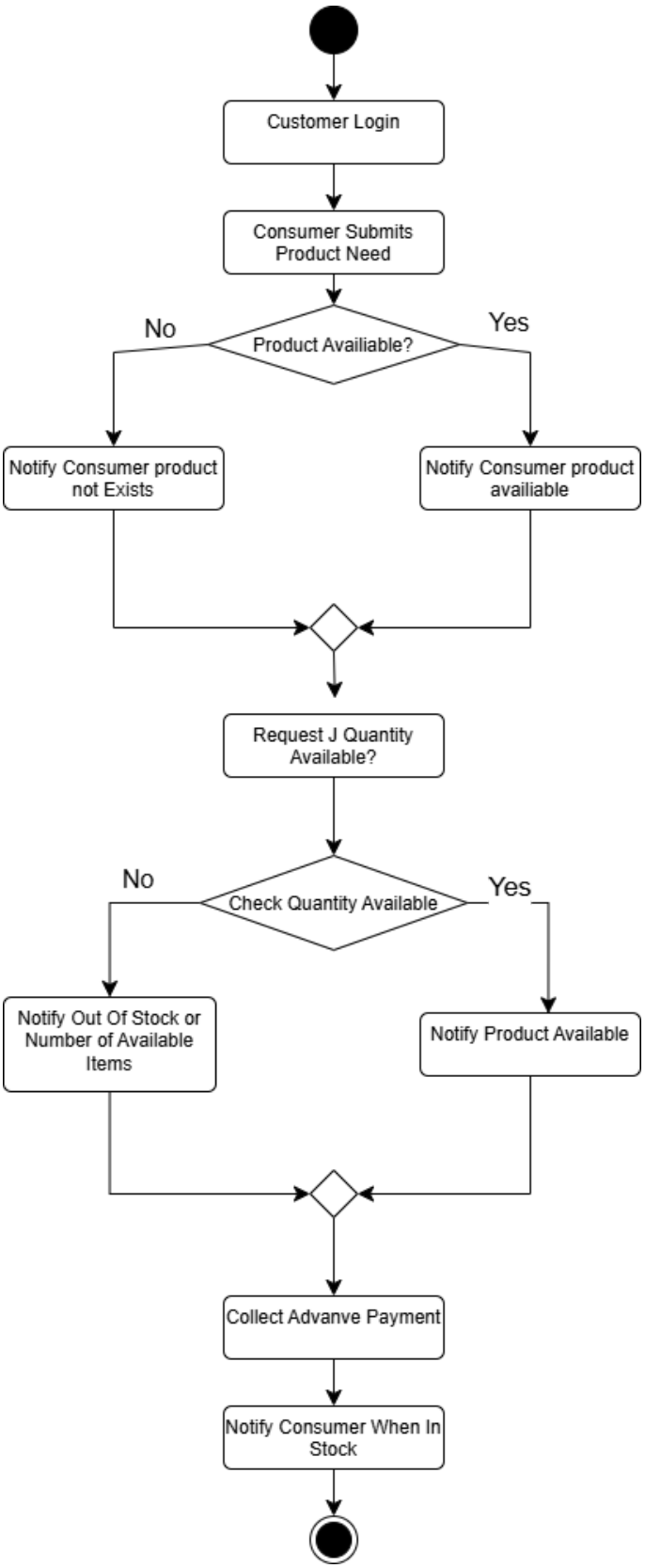
This SRS defines the complete functional scope, constraints, and future direction of the Inventory Management System, ensuring alignment between stakeholders and providing a solid foundation for design, development, and validation.

Use case:

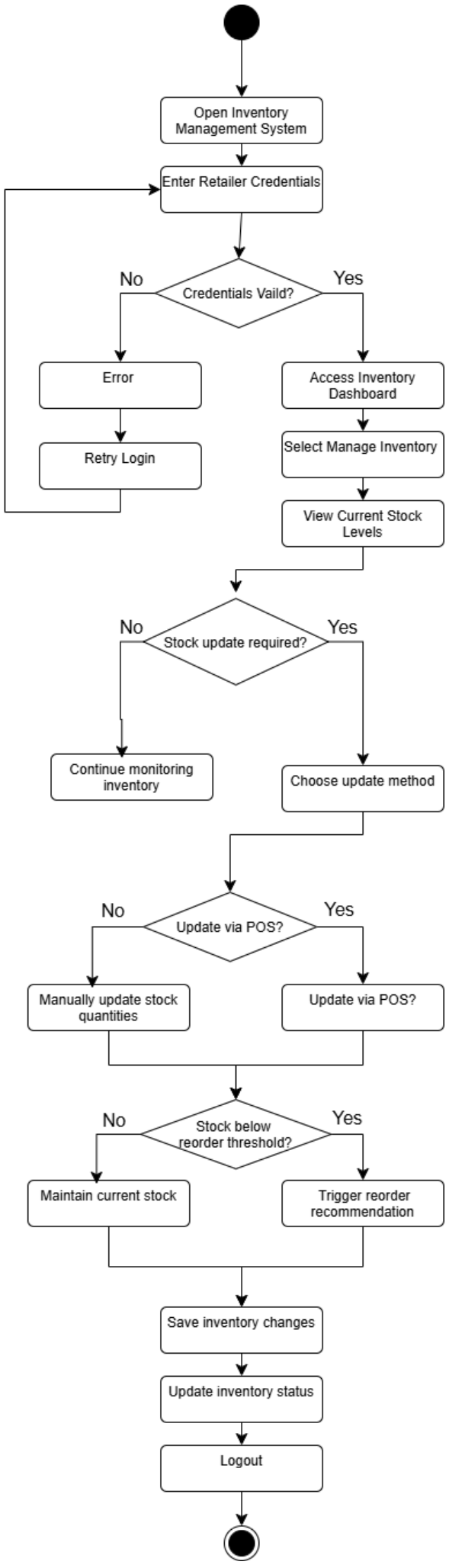


Activity Diagrams:

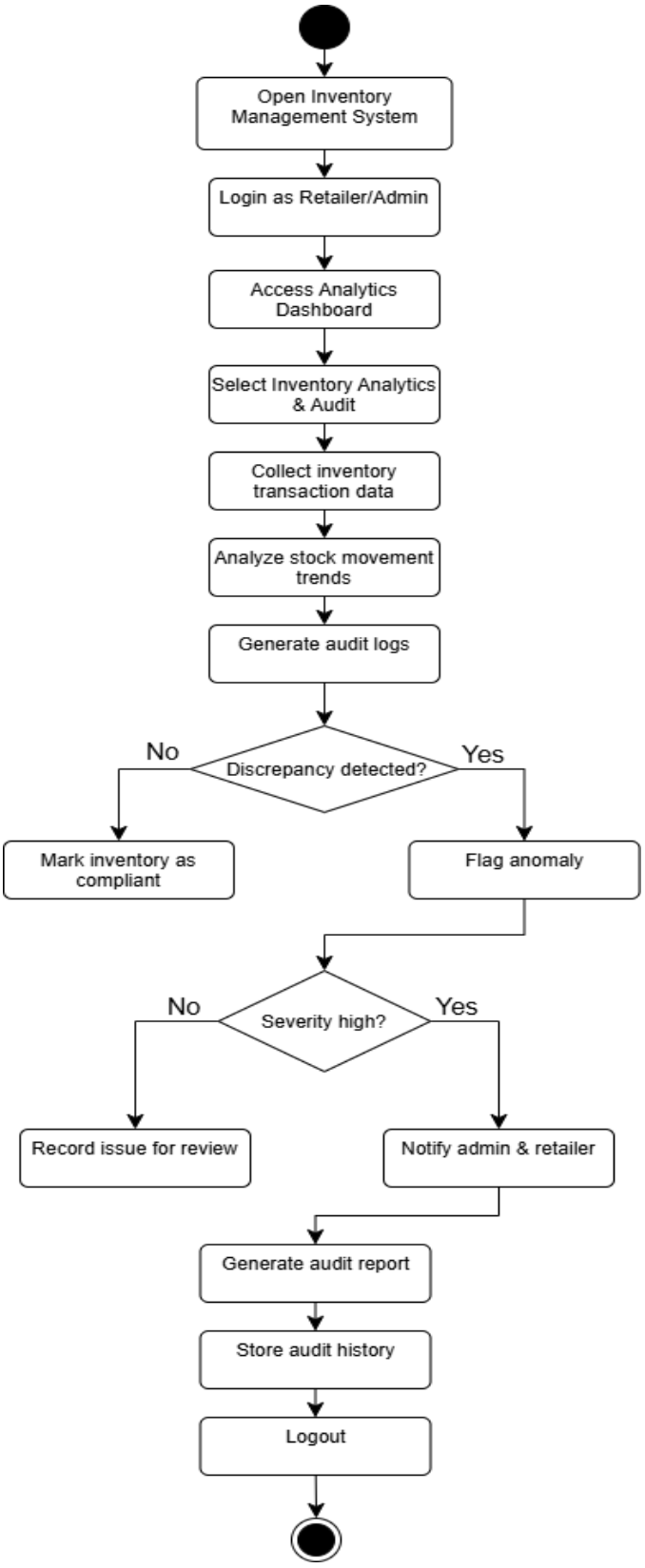
1) Product Availiability:



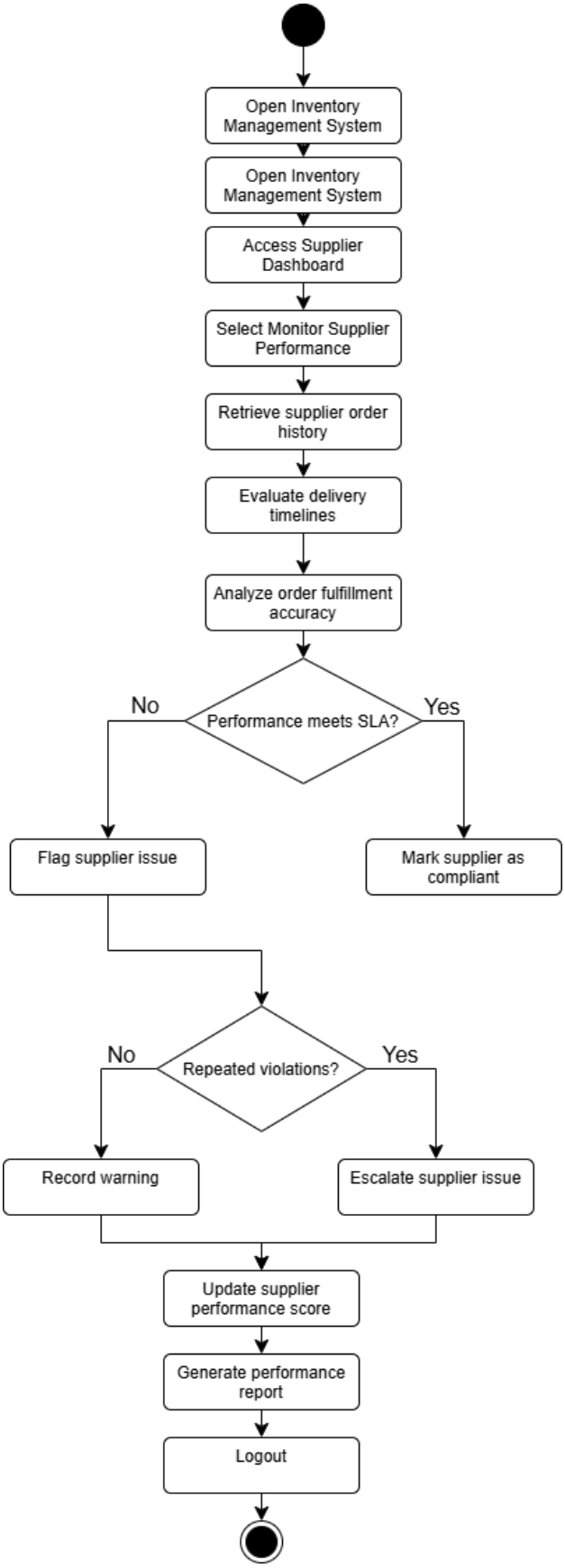
2) Manage Inventory:


Open Inventory Management System
Enter Retailer Credentials
Credentials Vaild?
No
Yes
Error
Retry Login
Access Inventory Dashboard
Select Manage Inventory
View Current Stock Levels
Stock update required?
No
Yes
Continue monitoring inventory
Choose update method
Update via POS?
No
Yes
Manually update stock quantities
Update via POS?
Stock below reorder threshold?
No
Yes
Maintain current stock
Trigger reorder recommendation
Save inventory changes
Update inventory status
Logout

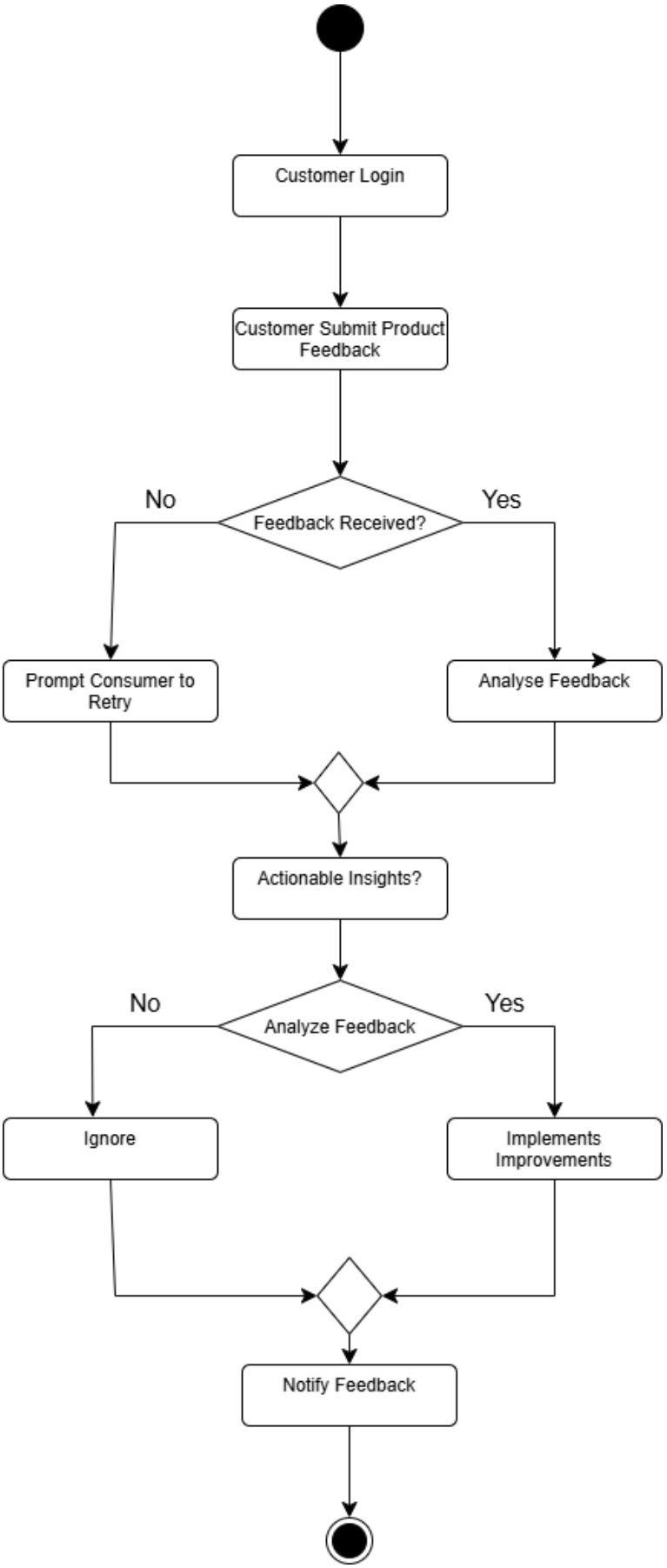
3) Audit Data:


Open Inventory Management System
Login as Retailer/Admin
Access Analytics Dashboard
Select Inventory Analytics & Audit
Collect inventory transaction data
Analyze stock movement trends
Generate audit logs
Discrepancy detected?
No
Yes
Mark inventory as compliant
Flag anomaly
Severity high?
No
Yes
Record issue for review
Notify admin & retailer
Generate audit report
Store audit history
Logout

4) Supplier Performance:



5) Product Feedback:



6) Order fullfilement and delivery: