SUPPLY CHAIN MANAGEMENT SYSTEM

SWENG 837 SOFTWARE SYSTEM DESIGN

BY: KEVIN MALONE

PROBLEM STATEMENT AND REQUIREMENTS

Problem Definition

- Current supply chains lack transparency and traceability, leading to inefficiencies, fraud, and reduced consumer trust.

System Functionalities

- Record Creation
- Tracking
- Verification
- Inventory Management
- Reporting

PROBLEM STATEMENT AND REQUIREMENTS (CONTINUED)

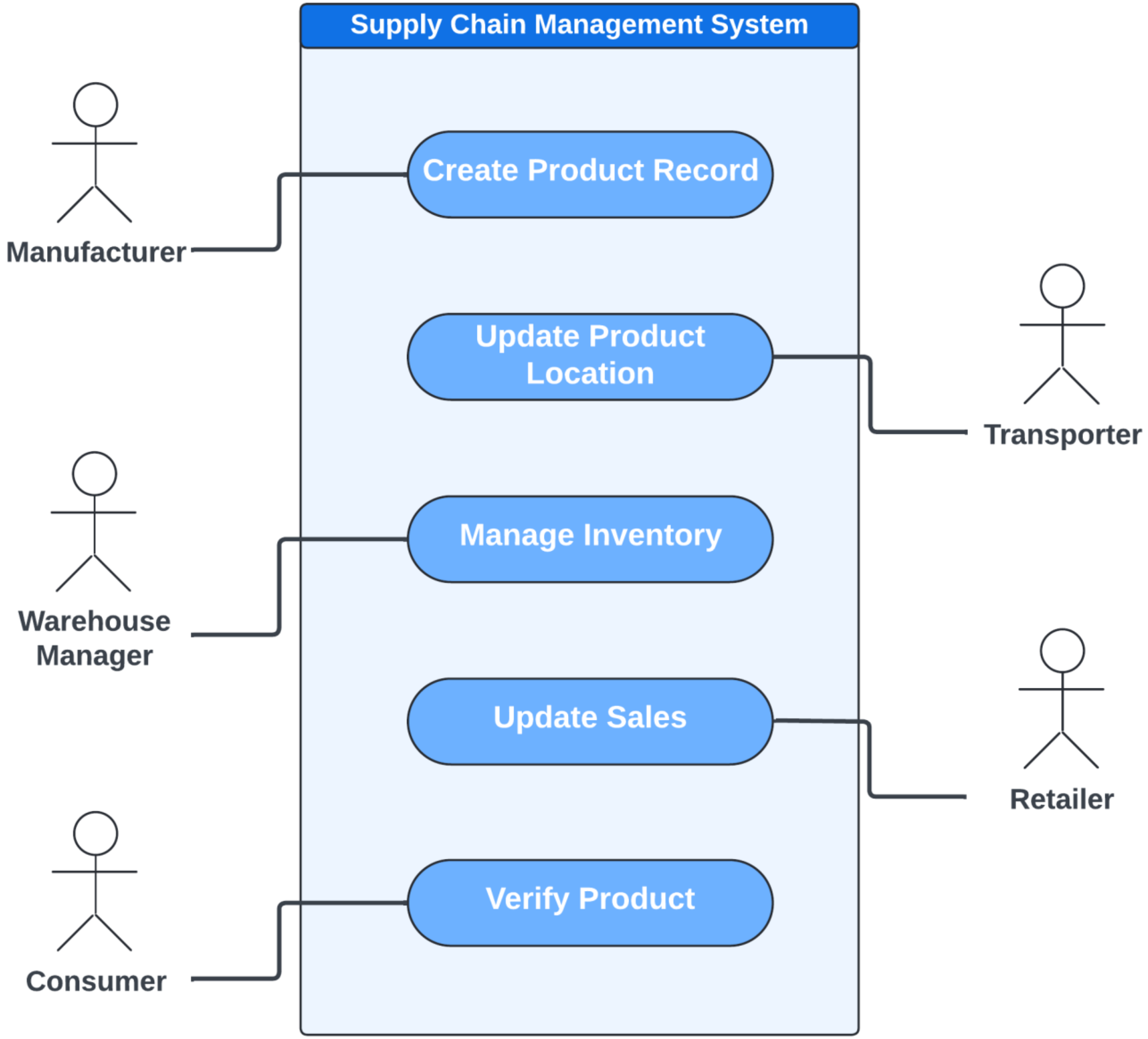
Target Users

- Manufacturers
- Transporters
- Warehouse Managers
- Retailers
- Consumers

Business Goals

- Increase transparency and trust in the supply chain.
- Improve efficiency by reducing delays and errors.
- Enhance product traceability from origin to consumer.
- Support compliance with regulations and standards.

Use Case Diagram



UML Domain Model

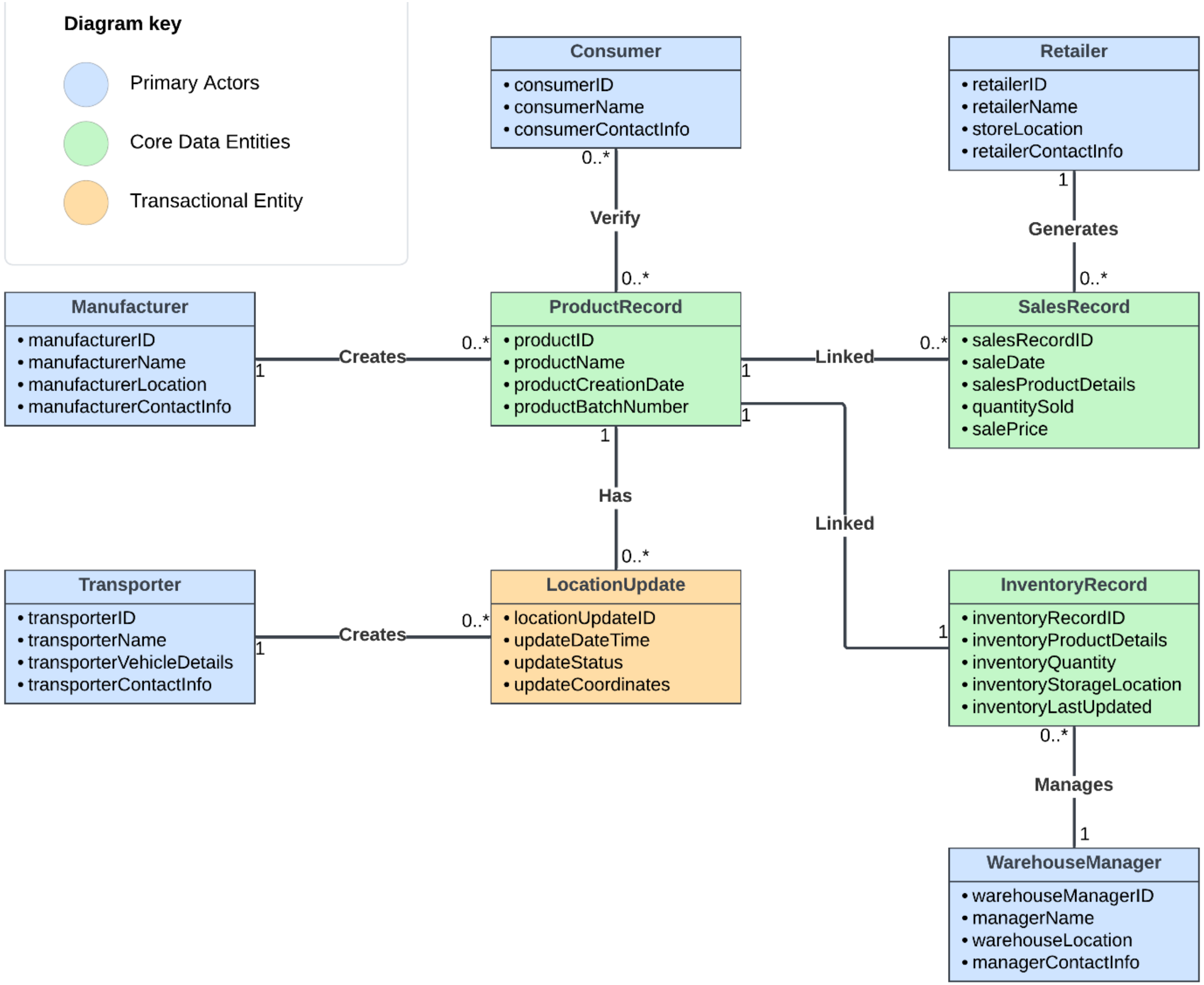


Diagram key

- ● Primary Actors
- ● Core Data Entities
- ● Transactional Entity

Consumer
- consumerID
- consumerName
- consumerContactInfo

Retailer
- retailerID
- retailerName
- storeLocation
- retailerContactInfo

Manufacturer
- manufacturerID
- manufacturerName
- manufacturerLocation
- manufacturerContactInfo

ProductRecord
- productID
- productName
- productCreationDate
- productBatchNumber

SalesRecord
- salesRecordID
- saleDate
- salesProductDetails
- quantitySold
- salePrice

Transporter
- transporterID
- transporterName
- transporterVehicleDetails
- transporterContactInfo

LocationUpdate
- locationUpdateID
- updateDateTime
- updateStatus
- updateCoordinates

InventoryRecord
- inventoryRecordID
- inventoryProductDetails
- inventoryQuantity
- inventoryStorageLocation
- inventoryLastUpdated

WarehouseManager
- warehouseManagerID
- managerName
- warehouseLocation
- managerContactInfo

UML Class Diagram

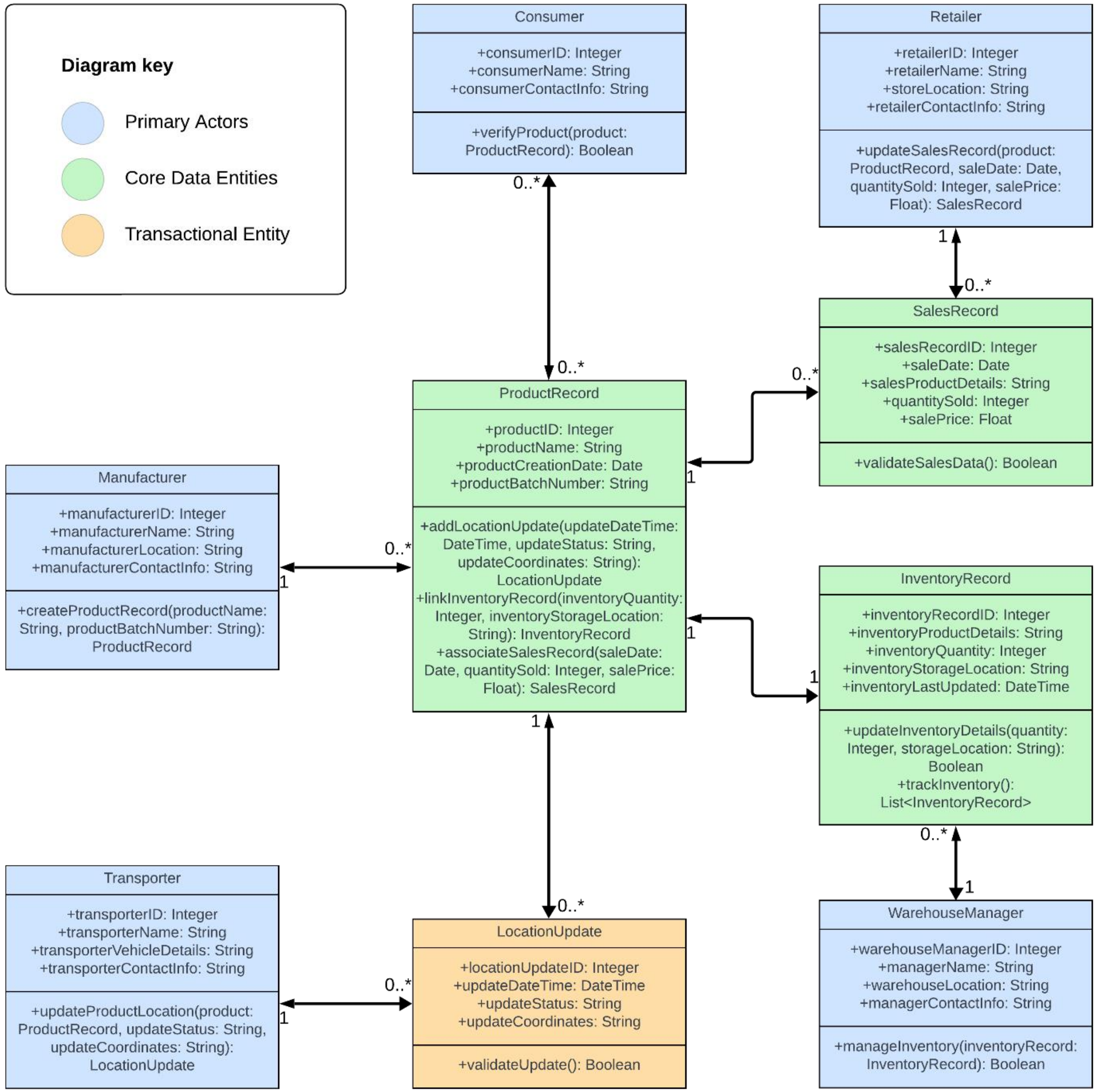


Diagram key

🔵 Primary Actors

🟢 Core Data Entities

🟠 Transactional Entity

Consumer

+consumerID: Integer
+consumerName: String
+consumerContactInfo: String

+verifyProduct(product: ProductRecord): Boolean

Retailer

+retailerID: Integer
+retailerName: String
+storeLocation: String
+retailerContactInfo: String

+updateSalesRecord(product: ProductRecord, saleDate: Date, quantitySold: Integer, salePrice: Float): SalesRecord

SalesRecord

+salesRecordID: Integer
+saleDate: Date
+salesProductDetails: String
+quantitySold: Integer
+salePrice: Float

+validateSalesData(): Boolean

ProductRecord

+productID: Integer
+productName: String
+productCreationDate: Date
+productBatchNumber: String

+addLocationUpdate(updateDateTime: DateTime, updateStatus: String, updateCoordinates: String): LocationUpdate
+linkInventoryRecord(inventoryQuantity: Integer, inventoryStorageLocation: String): InventoryRecord
+associateSalesRecord(saleDate: Date, quantitySold: Integer, salePrice: Float): SalesRecord

Manufacturer

+manufacturerID: Integer
+manufacturerName: String
+manufacturerLocation: String
+manufacturerContactInfo: String

+createProductRecord(productName: String, productBatchNumber: String): ProductRecord

InventoryRecord

+inventoryRecordID: Integer
+inventoryProductDetails: String
+inventoryQuantity: Integer
+inventoryStorageLocation: String
+inventoryLastUpdated: DateTime

+updateInventoryDetails(quantity: Integer, storageLocation: String): Boolean
+trackInventory(): List<InventoryRecord>

Transporter

+transporterID: Integer
+transporterName: String
+transporterVehicleDetails: String
+transporterContactInfo: String

+updateProductLocation(product: ProductRecord, updateStatus: String, updateCoordinates: String): LocationUpdate

LocationUpdate

+locationUpdateID: Integer
+updateDateTime: DateTime
+updateStatus: String
+updateCoordinates: String

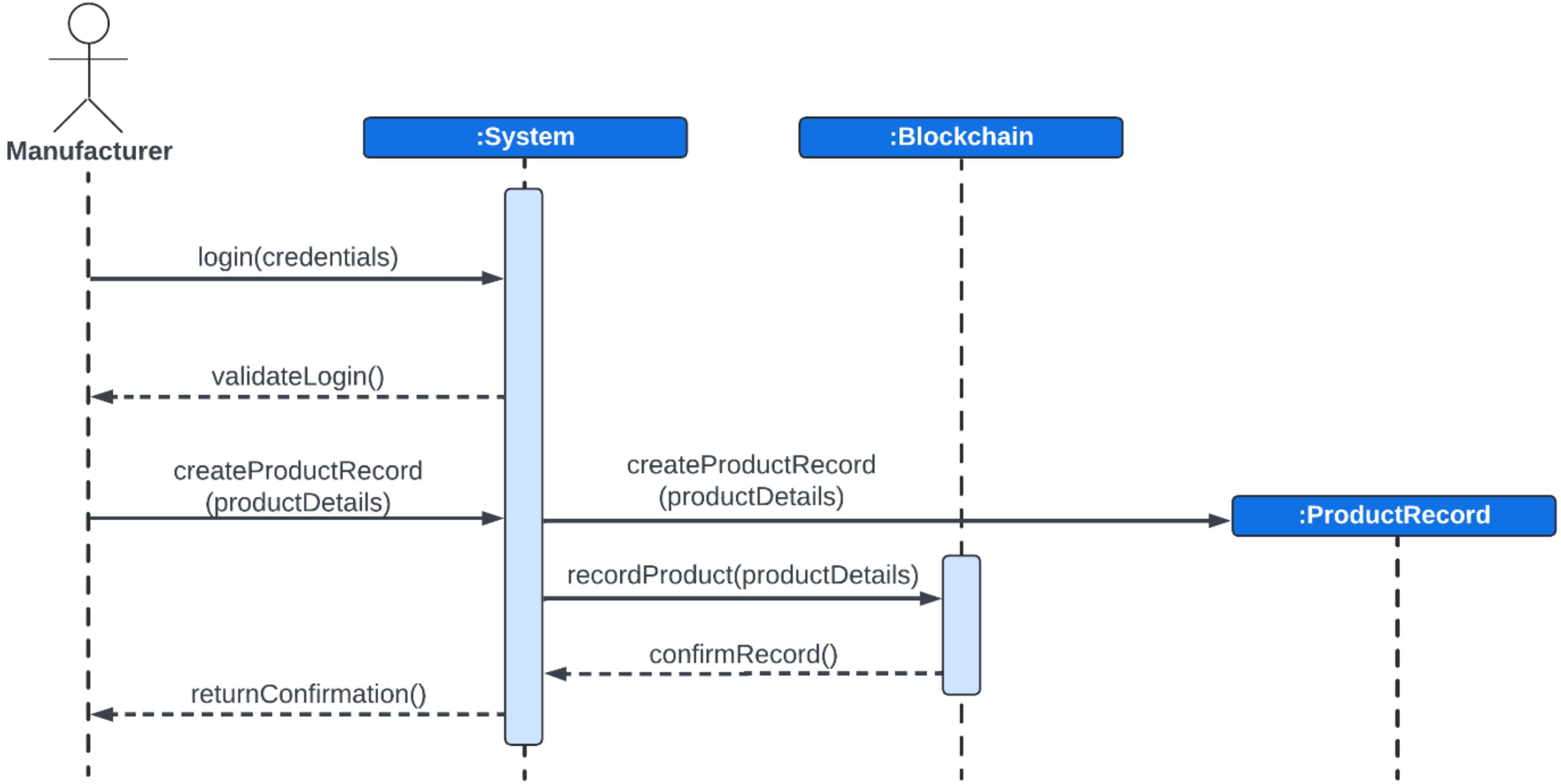
+validateUpdate(): Boolean

WarehouseManager

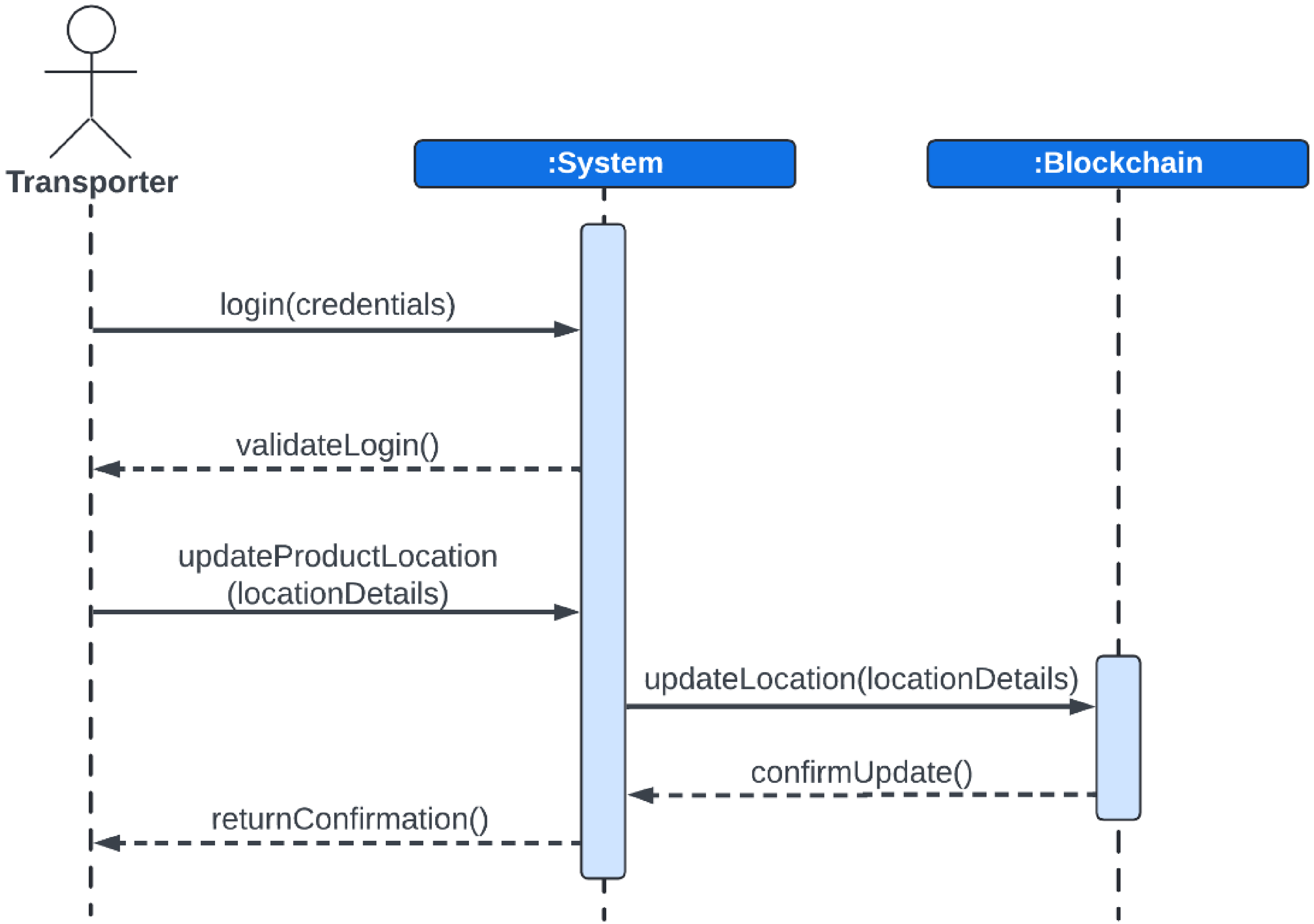
+warehouseManagerID: Integer
+managerName: String
+warehouseLocation: String
+managerContactInfo: String

+manageInventory(inventoryRecord: InventoryRecord): Boolean

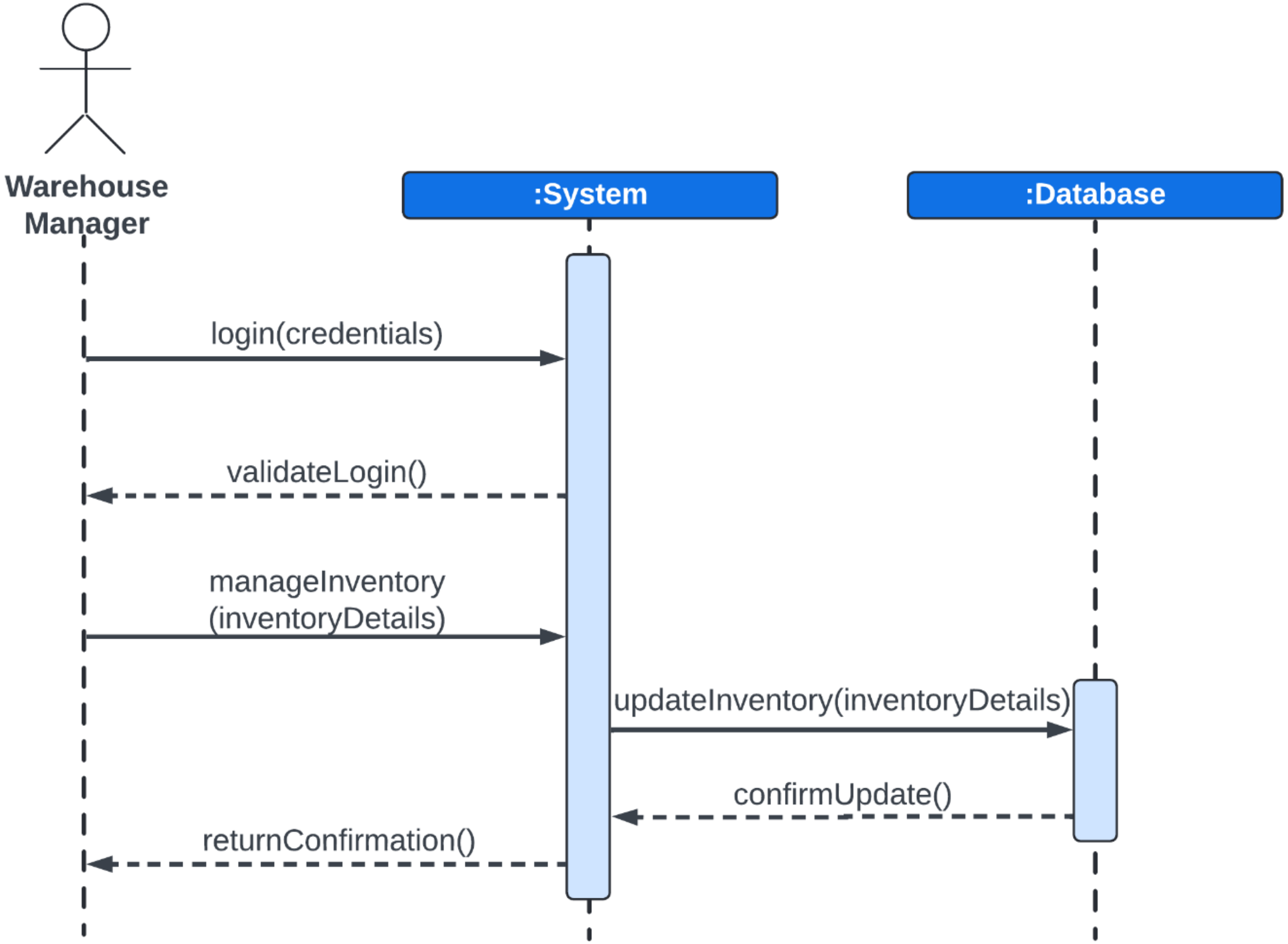
Create Product Record Sequence Diagram



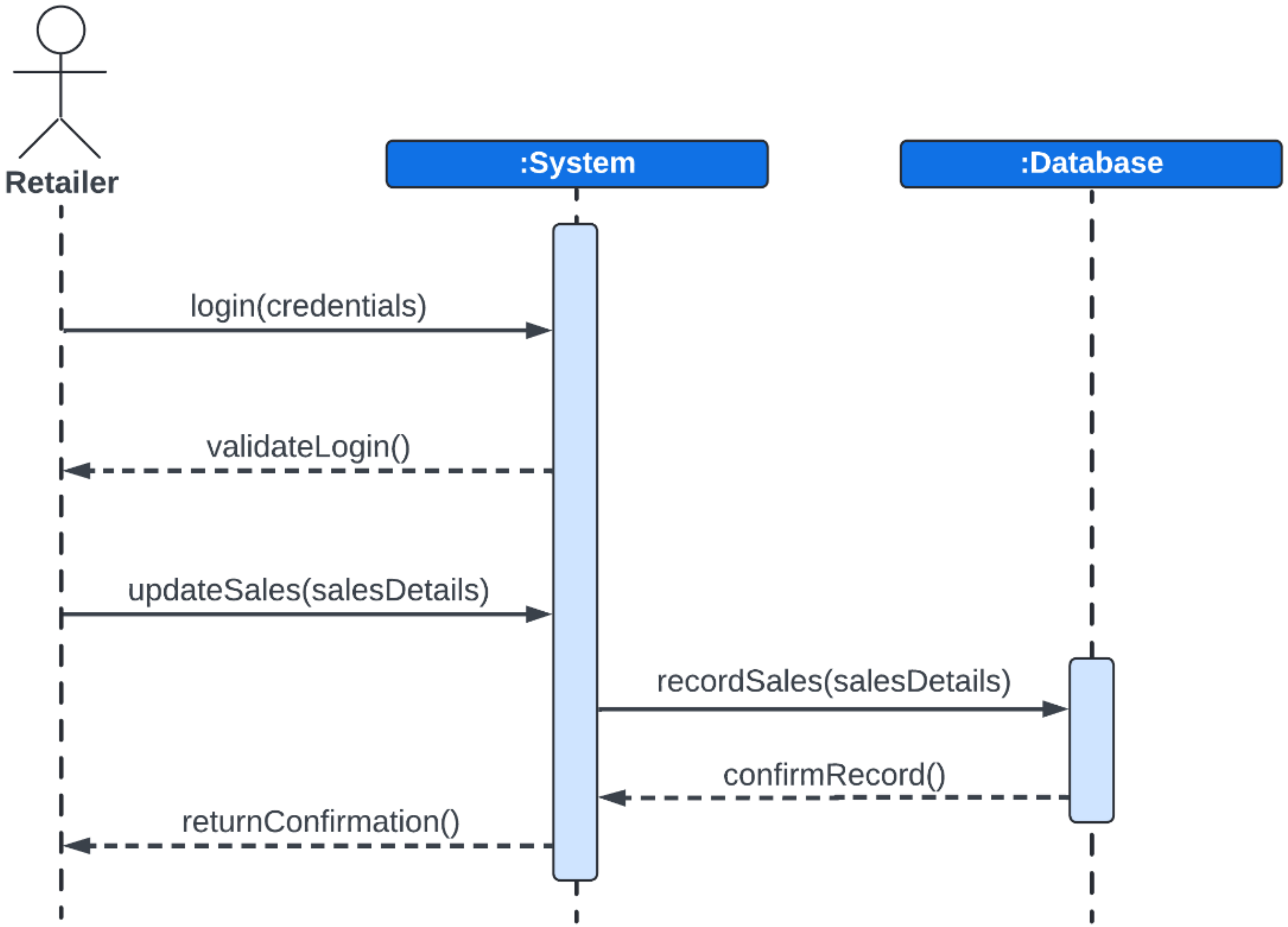
Update Product Location Sequence Diagram



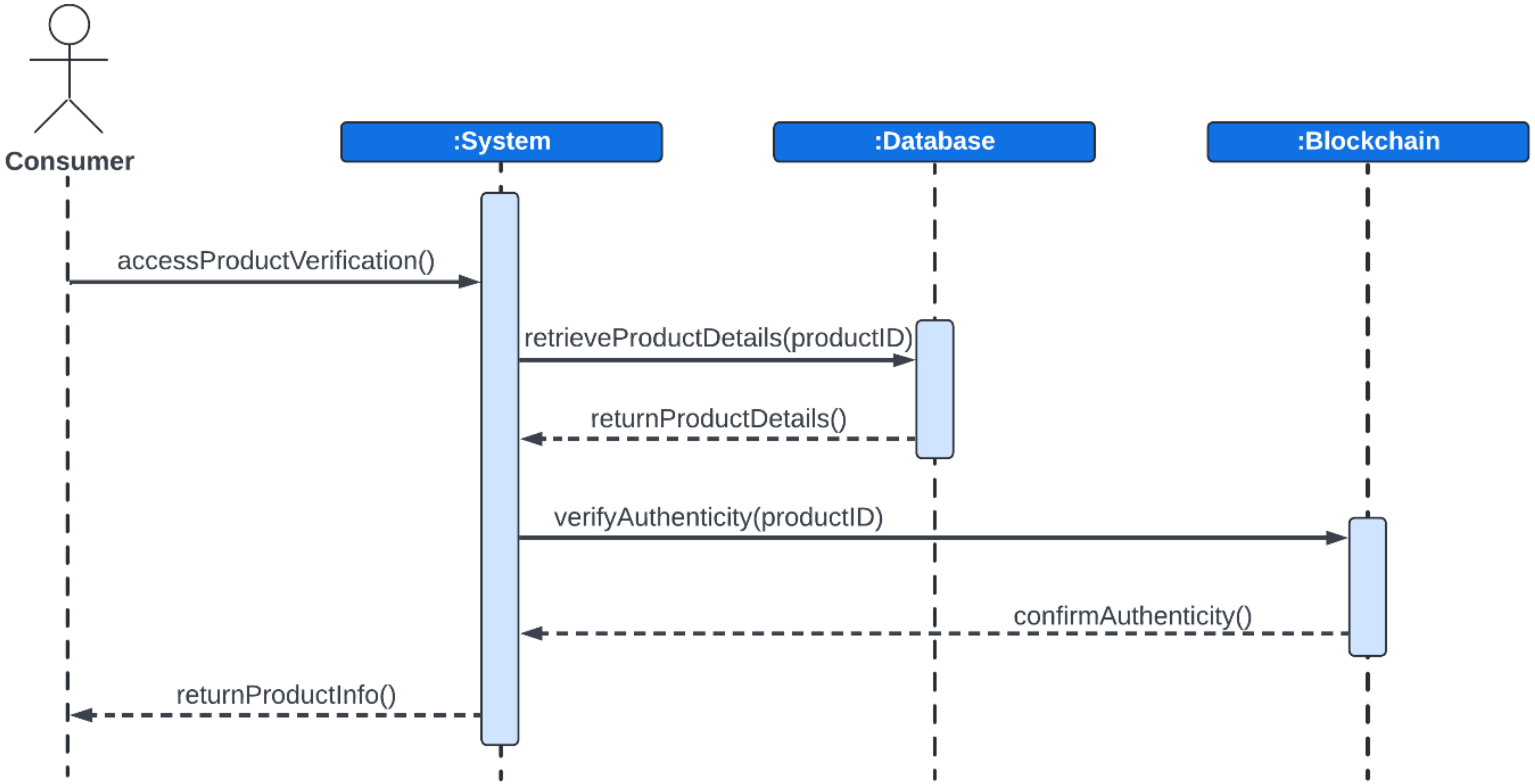
Manage Inventory Sequence Diagram



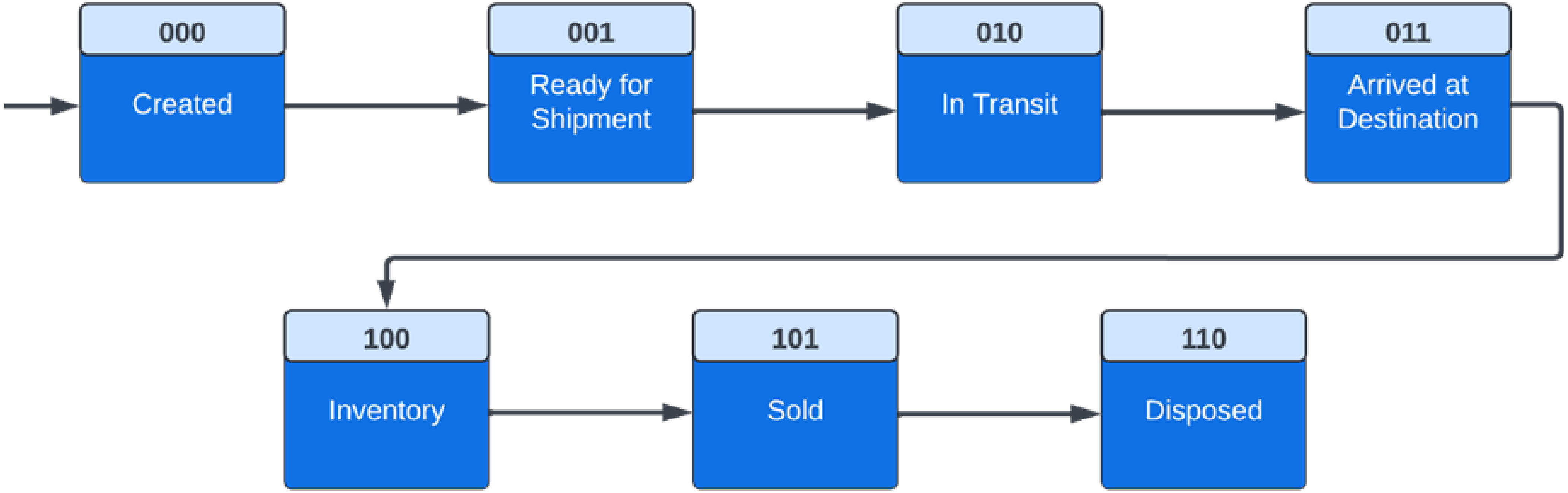
Update Sales Sequence Diagram



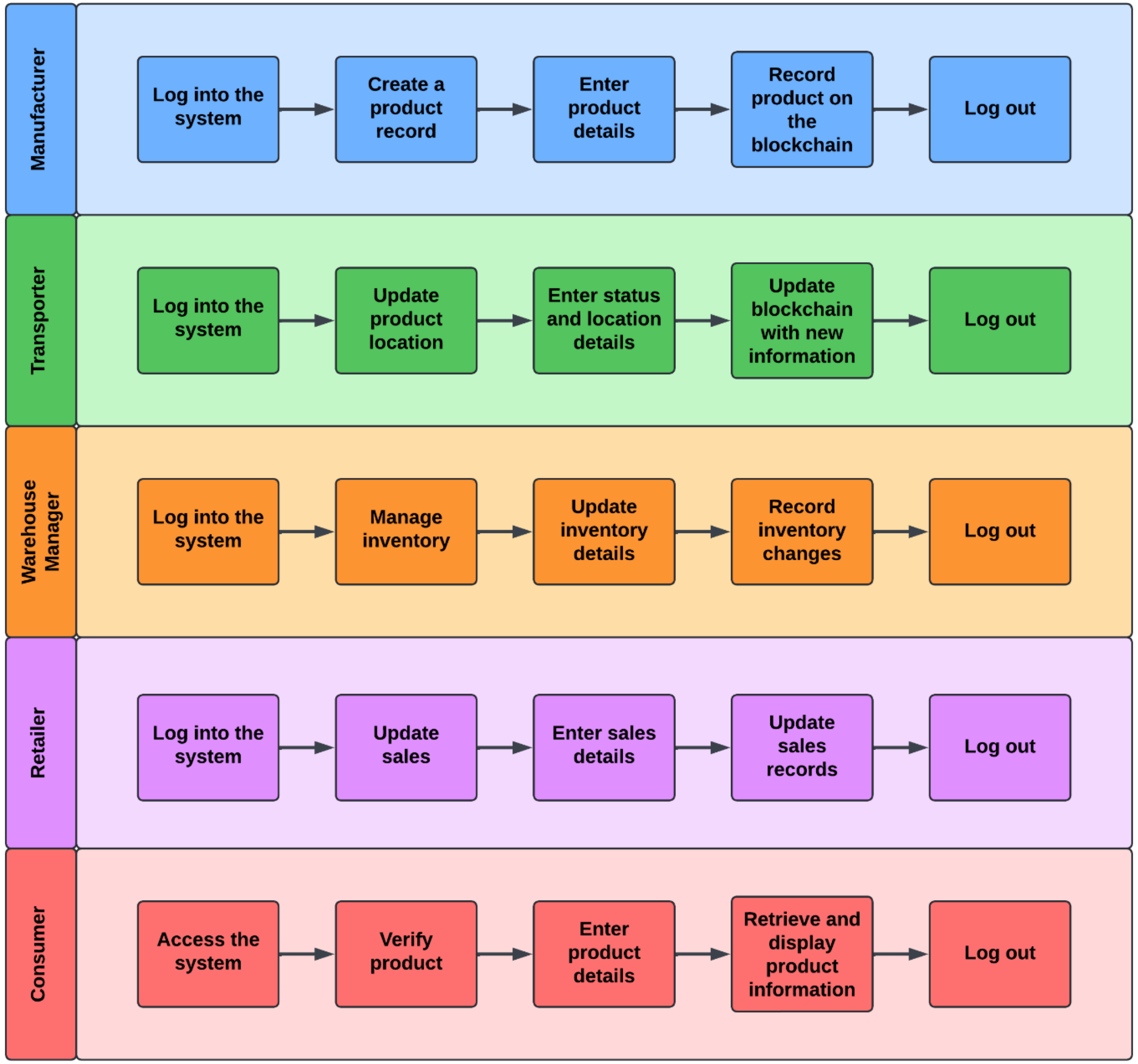
Verify Product Sequence Diagram



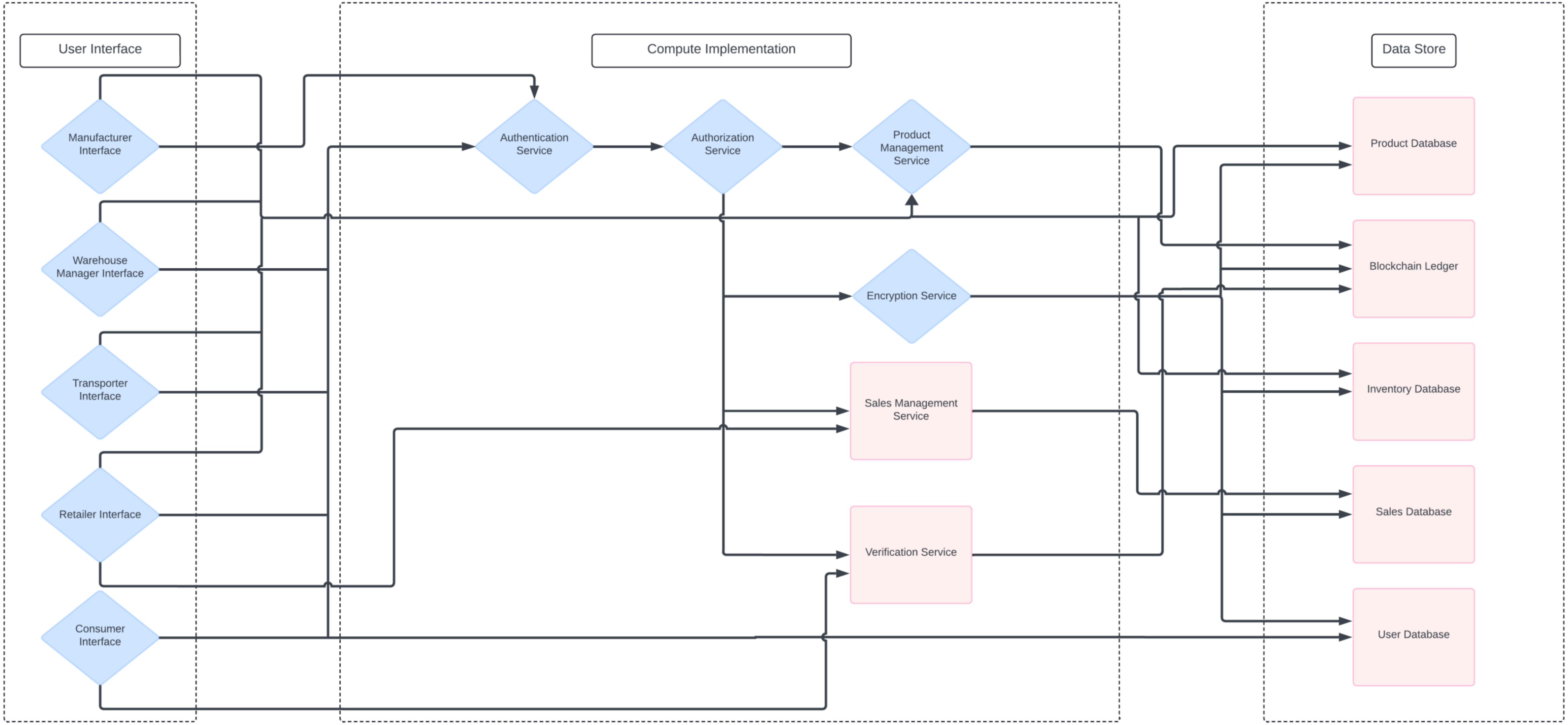
UML State Diagram



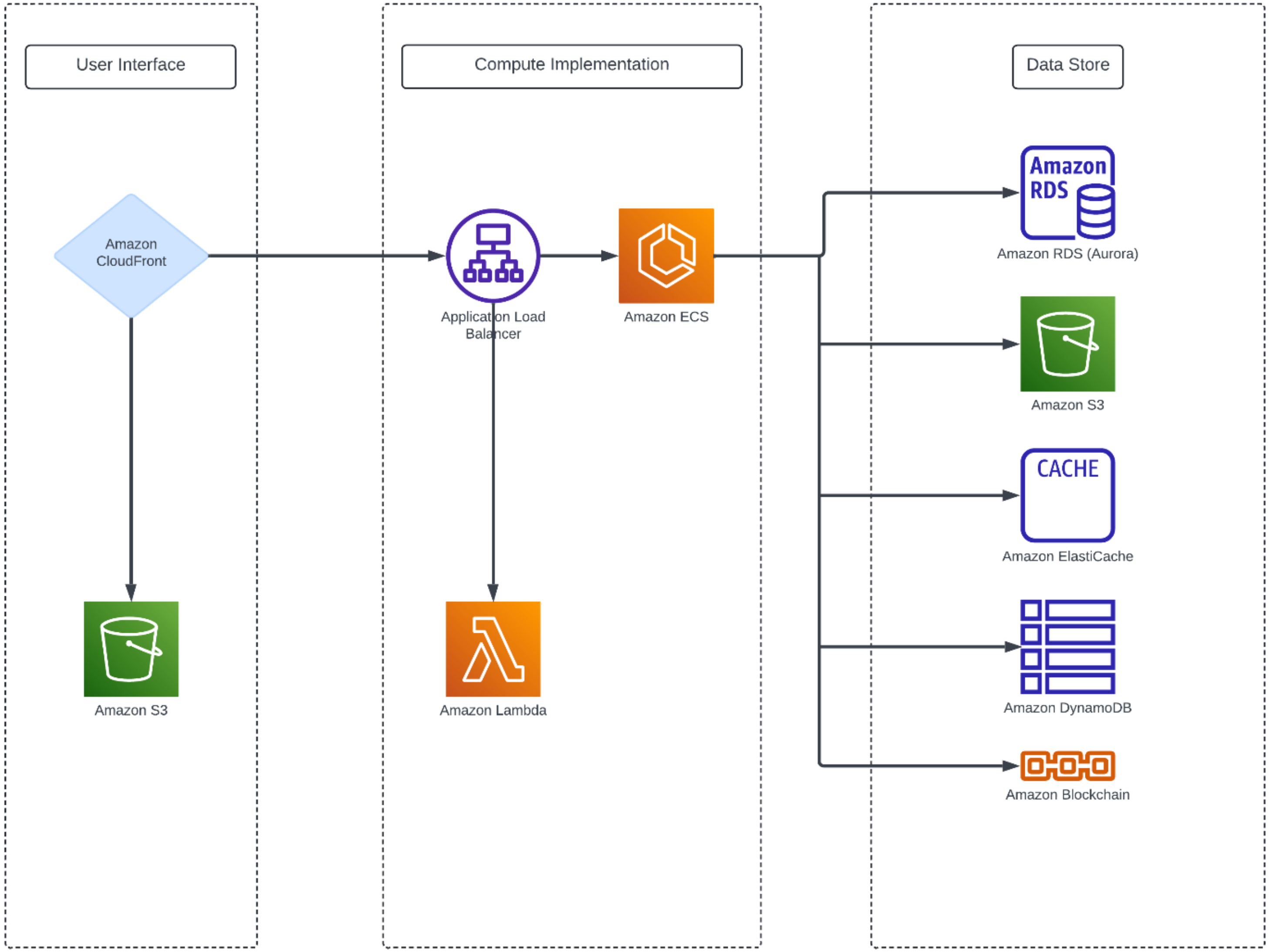
UML Activity Diagram (Swimlane Diagram)



UML Component Diagram



Cloud Deployment Diagram



Design Patterns

Design Pattern	Usage	Justification
Controller (GRASP)	Centralizes control in managing complex operations related to product records, inventory management, and location updates	By using a SupplyChainController, which coordinates actions like creating product records, updating locations, and verifying products, we maintain a clear separation of concerns, making the system easier to manage and extend.
Single Responsibility Principle (SRP) (SOLID)	Ensures that each class, such as ProductRecord, InventoryRecord, and SalesRecord, has a specific responsibility.	This principle is applied across the system to ensure that each class handles only one part of the business logic, like tracking product locations or managing inventory, which makes maintenance and future enhancements more straightforward.
Factory Method (GoF)	A factory method could be used to create different types of objects needed in the system, such as different types of database connections (e.g., for blockchain interaction or regular SQL databases).	This pattern supports the flexibility required to adapt to various backend services, enhancing the system's modularity and making it easier to switch or extend database services as needed.
Observer (GoF)	Could be used for monitoring changes across the supply chain, such as when a product's location is updated or when inventory levels change.	By implementing observers for key events like location updates, stakeholders (e.g., Warehouse Managers, Retailers) can be notified in real-time, improving responsiveness and data accuracy.
Repository (DDD)	Centralizes data access logic for entities such as ProductRecord, InventoryRecord, and SalesRecord.	This pattern encapsulates data access and simplifies testing by decoupling the system's business logic from the underlying data sources, making the system more resilient to changes in data storage technology.

THANK YOU

VIDEO LINK

https://youtu.be/cvqxdHpqlgw