I. INTRODUCTION

A. OVERVIEW OF THE PROJECT

- The Integrated Coffee Shop POS and Inventory Management System is a modern solution designed to streamline operations by unifying sales processing and real-time inventory tracking. It directly addresses critical inefficiencies in existing systems, such as weak security, manual inventory updates, disorganized interfaces, and lack of actionable insights.

PROBLEM STATEMENT

- The current Coffee Shop POS system suffers from critical inefficiencies:
- Weak security with no role-based access, allowing unauthorized personnel to view sensitive data.
- Manual inventory tracking, leading to errors and delays in restocking.
- No low-stock alerts, risking ingredient shortages.
- Unorganized UI (e.g., randomized size buttons) causing operational delays.
- Inability to track daily/weekly sales by product or identify non-selling items.
- System instability (crashes, lagging), disrupting order processing.

a. OBJECTIVES OF THE STUDY

The proposed system aims to:

- Implement role-based access control to secure sensitive data.
- Automate real-time inventory updates and low-stock alerts via SMS/email.
- Provide sales analytics (daily/weekly reports by product) and highlight non-selling items.
- Design a stable, user-friendly POS interface with fixed layouts and minimal lag.

b. SCOPE AND LIMITATIONS

Scope:

- POS system with order processing, role-based authentication, and sales tracking.
- Inventory management with automated alerts and real-time updates.

- Dashboard for sales reports and product performance analytics.
- Integration with payment gateways (e.g., Stripe) and SMS/email APIs.

Limitation:

- Does not include direct supplier ordering (alerts only).
- Limited to basic menu customization (e.g., sizes, images).
- Offline mode supports order processing but delays inventory sync.

B. PROJECT BACKGROUND

a. DESCRIPTION OF THE PROJECT DOMAIN

The coffee shop industry relies heavily on efficient POS systems to manage high transaction volumes and dynamic inventory needs. However, many small businesses use outdated or manual systems, leading to operational bottlenecks, financial inaccuracies, and customer dissatisfaction. This project targets coffee shop owners seeking modern, integrated solutions to streamline sales and inventory management.

b. CURRENT SYSTEM OR PROCESS

The existing system:

- Lacks role-based security, exposing financial data to all staff.
- Requires manual inventory input, increasing human error.
- Provides no low-stock notifications or sales breakdowns by product.
- Uses a disorganized UI with randomized buttons, slowing down order processing.
- Suffers from frequent crashes and lag, disrupting workflow.

c. JUSTIFICATION OF THE PROPOSED SYSTEM

The new system addresses these gaps by:

- Enhancing security through role-based access (owner, barista, manager).
- Automating inventory updates and alerts, reducing manual work and errors.

- Providing actionable insights via sales analytics to optimize menu offerings.
- Improving system reliability with cloud-based architecture and crash recovery.
- Streamlining operations with an intuitive UI, fixed layouts, and offline support.
- This solution will save time, reduce costs, and improve decision-making for coffee shop owners.

II. STAKEHOLDER AND USER ANALYSIS

A. USER PERSONA

1. Coffee Shop Owner
   - Goals: Secure system access, real-time inventory tracking, sales analytics, and automated alerts.
   - Behaviors: Monitors sales, manages staff permissions, reviews reports.
   - Pain Points: Weak security, manual inventory input, inability to track low stock or non-selling products.
   - Goals: Oversee operations, generate sales reports, optimize stock ordering.
   - Pain Points: Lack of sales breakdown by product, no low-stock alerts.

2. Barista
   - Goals: Efficient order processing, organized menu interface, reliable system.
   - Behaviors: Processes orders, updates inventory after sales, interacts with POS daily.
   - Pain Points: System crashes, lagging interface, randomized UI buttons, manual inventory updates.

B. STAKEHOLDER IDENTIFICATION

Stakeholder	**Interest in the Project**
Coffee Shop Owner	Secure system, automated inventory, sales insights.
Baristas	Stable POS, intuitive interface, reduced manual work.
Suppliers (Indirect)	Timely inventory restocking alerts.
Customers	Faster service, accurate orders.

C. USER STORIES AND REQUIREMENTS ANALYSIS

1. USER STORIES

Owner:

- As an owner, I want role-based access control so that only authorized staff can view financial data.
- As an owner, I want automated low-stock alerts so that I can reorder ingredients proactively.
- As an owner, I want daily sales reports by product so that I can identify top-selling items.
- As an owner, I want a dashboard to track non-selling products this week to adjust the menu.

Barista:

- As a barista, I want a stable POS system so that orders are processed without delays or crashes.
- As a barista, I want a fixed menu layout so that size buttons are organized and easy to locate.

Manager:

2. IDENTIFICATION OF USER REQUIREMENTS

a) FUNCTIONAL

- Role-based authentication (e.g., owner, barista).
- Image upload for products (e.g., coffee types).
- Real-time inventory updates after each sale.
- Automated low-stock alert.
- Sales tracking by product, size, and time (daily/weekly).
- Fixed UI layout for product customization (e.g., size buttons).
- Report generation for non-selling products.

b) NON-FUNCTIONAL REQUIREMENTS

- Performance: Handle 50+ concurrent transactions without lag.

- Security: Encrypted user data and HTTPS for all communications.
- Usability: Intuitive interface with <2 clicks for common tasks.
- Reliability: 99.9% uptime with crash recovery features.

3. POSSIBLE SYSTEM REQUIREMENTS

   a) HARDWARE REQUIREMENTS
      - POS Terminals/Tablets with Touchscreen.

   b) SOFTWARE REQUIREMENTS
      - Frontend: React.js (responsive UI).
      - Backend: Node.js + Express.js.
      - Database: MySQL with tables for Users, Products, Sales, Inventory.
      - Cloud storage for product images (e.g., AWS S3).

   c) DATABASE REQUIREMENTS
      - Users Table: UserID, Role, Password (hashed).
      - Products Table: ProductID, Name, Price, ImageURL, SizeOptions.
      - Sales Table: SaleID, ProductID, Quantity, Timestamp.
      - Inventory Table: ItemID, CurrentStock, MinimumThreshold.

   d) NETWORK AND COMMUNICATION REQUIREMENTS

   e) INTEGRATION WITH EXISTING SYSTEMS (IF APPLICABLE)

V. PROJECT DEVELOPMENT PLAN

A. Development Methodology

Agile Methodology will be adopted for iterative development, allowing:

- **Sprints:** 2-week cycles for incremental feature delivery (e.g., role-based access in Sprint 1, inventory automation in Sprint 2).
- **Stakeholder Feedback:** Regular demos with coffee shop owners and baristas to refine UI/UX and functionality.
- **Flexibility:** Adjust priorities based on emerging needs (e.g., optimizing crash recovery if instability persists).

B. Timeline and Milestones

Gantt Chart Overview (Total Duration: 6 Months)

Phase	Duration	Key Tasks
1. Requirements Gathering	Weeks 1-2	Stakeholder interviews, finalize user stories, prioritize functional requirements.
2. Design	Weeks 3-4	Create UI/UX mockups, database schema, system architecture.
3. Development	Weeks 5-12	Build modules: authentication, POS interface, inventory sync, analytics dashboard.
4. Testing	Weeks 13-14	Unit, integration, and user acceptance testing (UAT) with baristas and owners.
5. Deployment	Weeks 15-16	Pilot testing at 1–2 coffee shops; full rollout post-feedback.

Phase	Duration	Key Tasks
6. Post-Launch Support	Weeks 17-24	Monitor performance, bug fixes, and staff training.

C. Risk Analysis and Mitigation Strategies

Risk	Mitigation Strategy
Delays in development	Buffer time (2 weeks) added to timeline; prioritize MVP features first.
Technical challenges	Allocate 1 sprint for prototyping complex features (e.g., real-time inventory sync).
Data security breaches	Conduct penetration testing; use HTTPS and AES-256 encryption for sensitive data.
User resistance to new system	Provide hands-on training and detailed user manuals; offer 24/7 support during rollout.
Network outages	Implement offline mode for order processing; auto-sync when connectivity resumes.

VI. Conclusion

A. Summary of Key Points

- The system addresses critical gaps in security, inventory, and usability through role-based access, automation, and analytics.
- Agile methodology ensures iterative delivery aligned with stakeholder feedback.
- Risks are mitigated via prototyping, buffer time, and robust security measures.

B. Expected Impact and Benefits of the System

- Operational Efficiency: Reduce manual inventory errors by 80% and order processing time by 50%.

- Financial Accuracy: Track daily sales by product, enabling data-driven menu adjustments.
- Customer Satisfaction: Faster service via stable UI and fewer system crashes.
- Scalability: Cloud-based architecture supports future expansion (e.g., multi-location tracking).

SYSTEM ARCHITECTURE AND DESIGN CONSIDERATIONS

A. High-level System Architecture

The system follows a layered architecture to ensure modularity, scalability, and separation of concerns:

1. Frontend Layer:
    - POS Interface: Built with React.js for responsive, touch-friendly screens.
    - Admin Dashboard: Role-based access for owners/managers to view reports and manage inventory.
2. Backend Layer:
    - Node.js/Express.js API: Handles business logic, authentication, and integrations.
    - Real-Time Sync Engine: Updates inventory and sales data across devices instantly.
3. Database Layer:
    - MySQL: Relational database for structured storage of users, products, sales, and inventory.
    - Caching: Redis for temporary storage of frequently accessed data (e.g., menu items).

4. External Integrations:
    - Built-in Notification System: Internal notification system for alerts and updates.
5. Hardware Layer:
    - POS Terminals/Tablets: Touchscreen devices for order processing.
    - Central Server: Hosts backend and database (cloud-based or on-premises).

B. Diagram Representation

Use Case Diagram

Actors:

- Owner
- Barista
- Manager
- System (automated processes)

Key Use Cases:

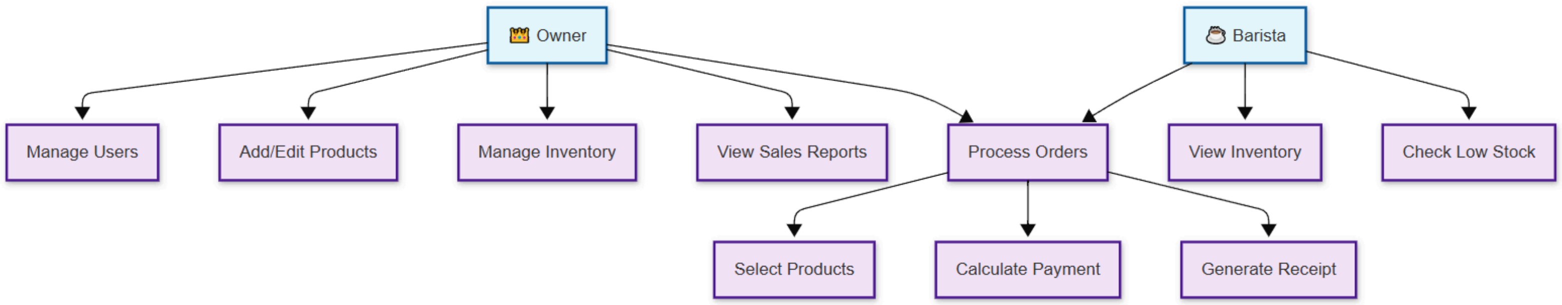
- Owner: Manage user roles, view financial reports, set low-stock thresholds.
- Barista: Process orders, update inventory post-sale, view menu.
- Manager: Generate sales analytics, track non-selling products, restock inventory.
- System: Send low-stock alerts, sync data in real-time, recover from crashes.

Context Diagram

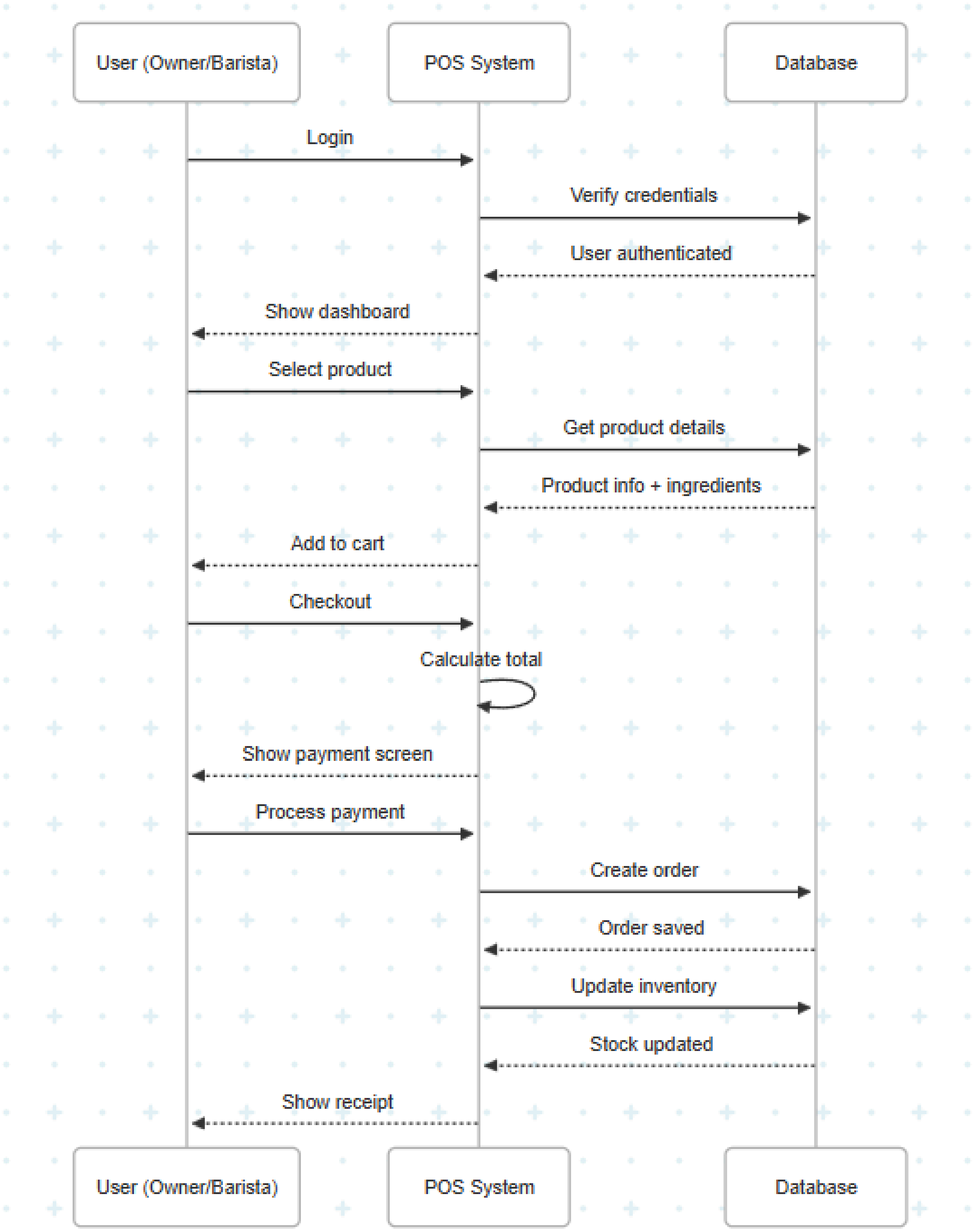
System Preferences:

- Built-in Notification System.
- Cloud Storage (AWS S3) for product images.

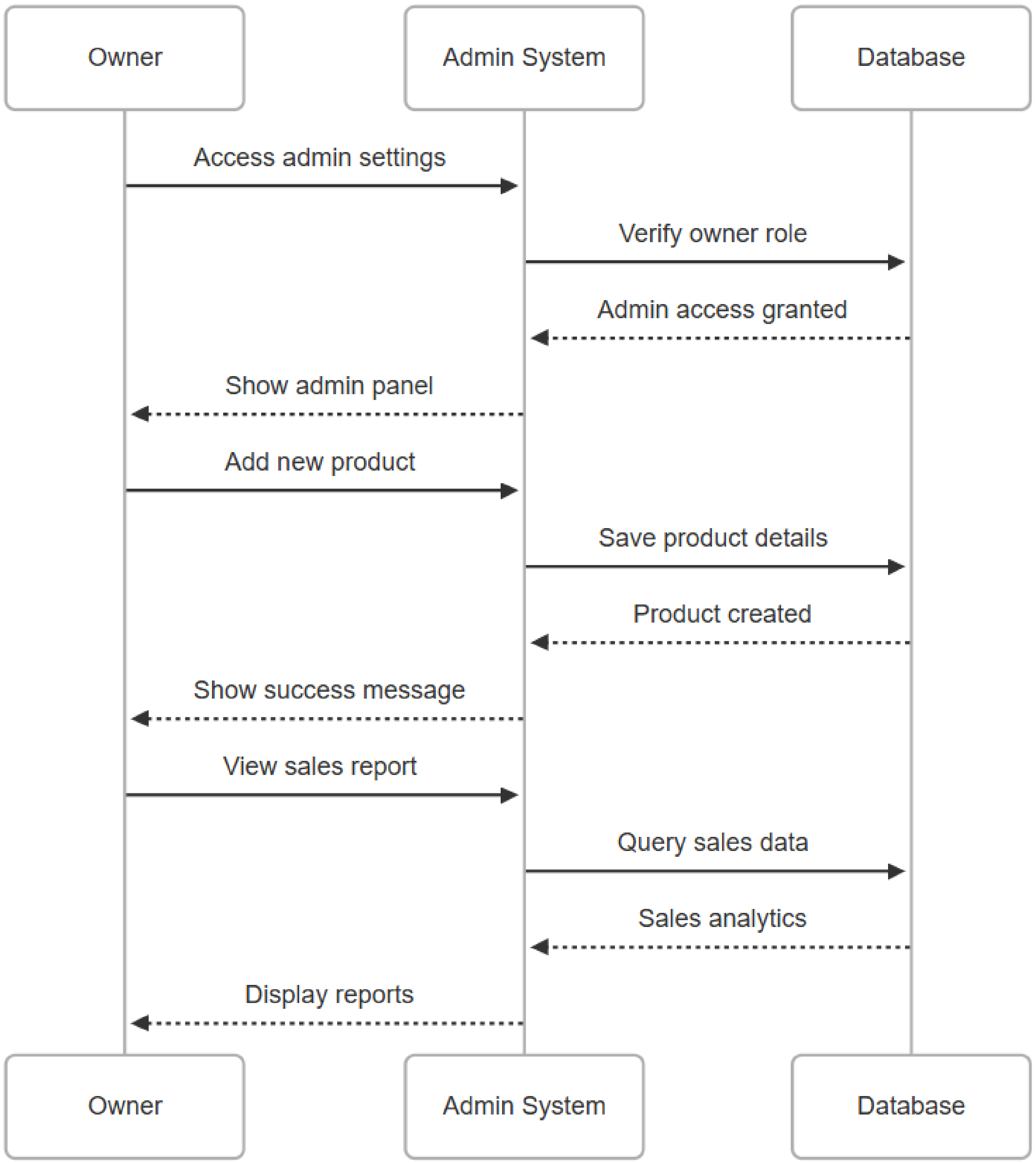
1.Use Case Diagram



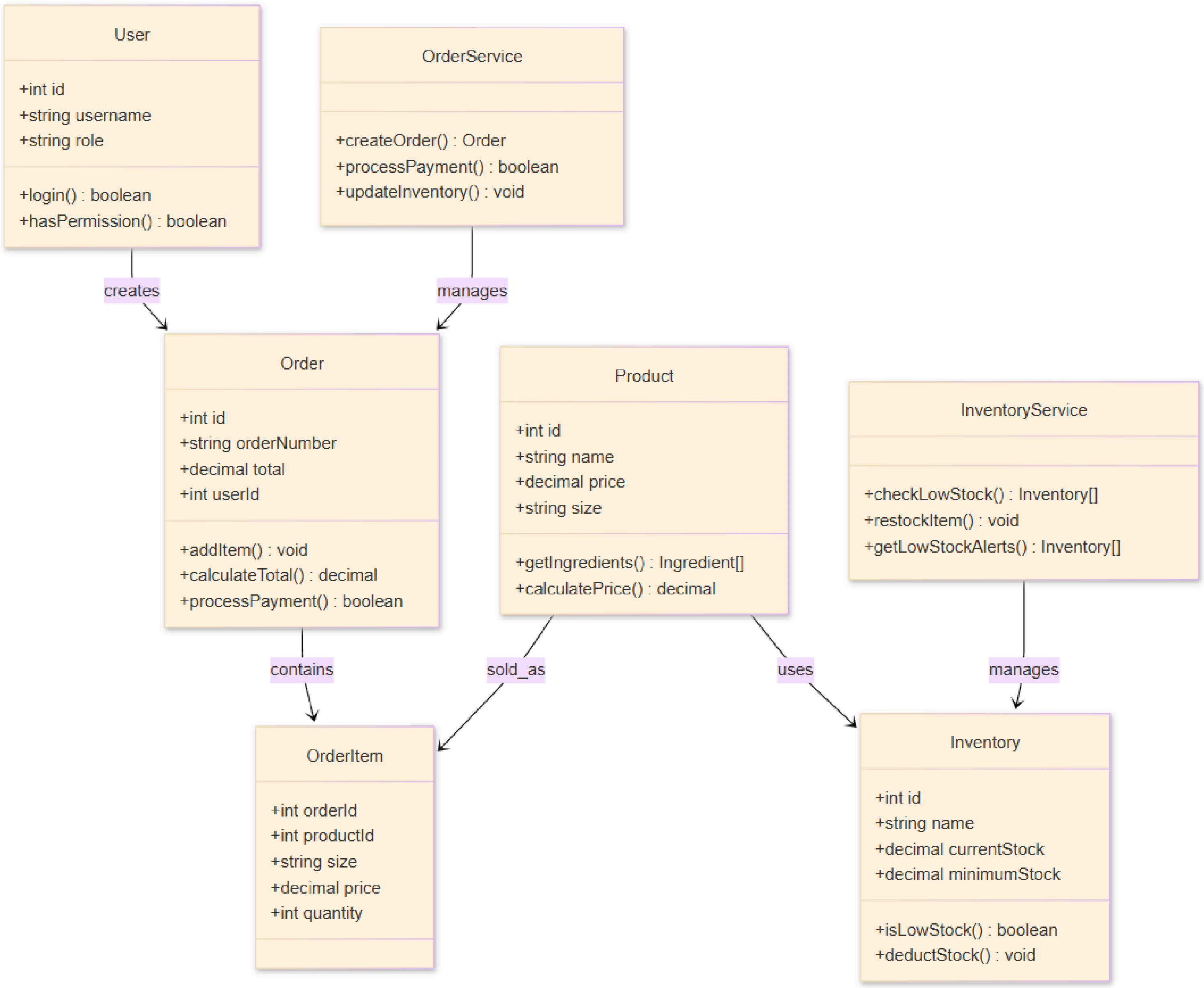
2.Sequence Diagram: Order Processing Flow



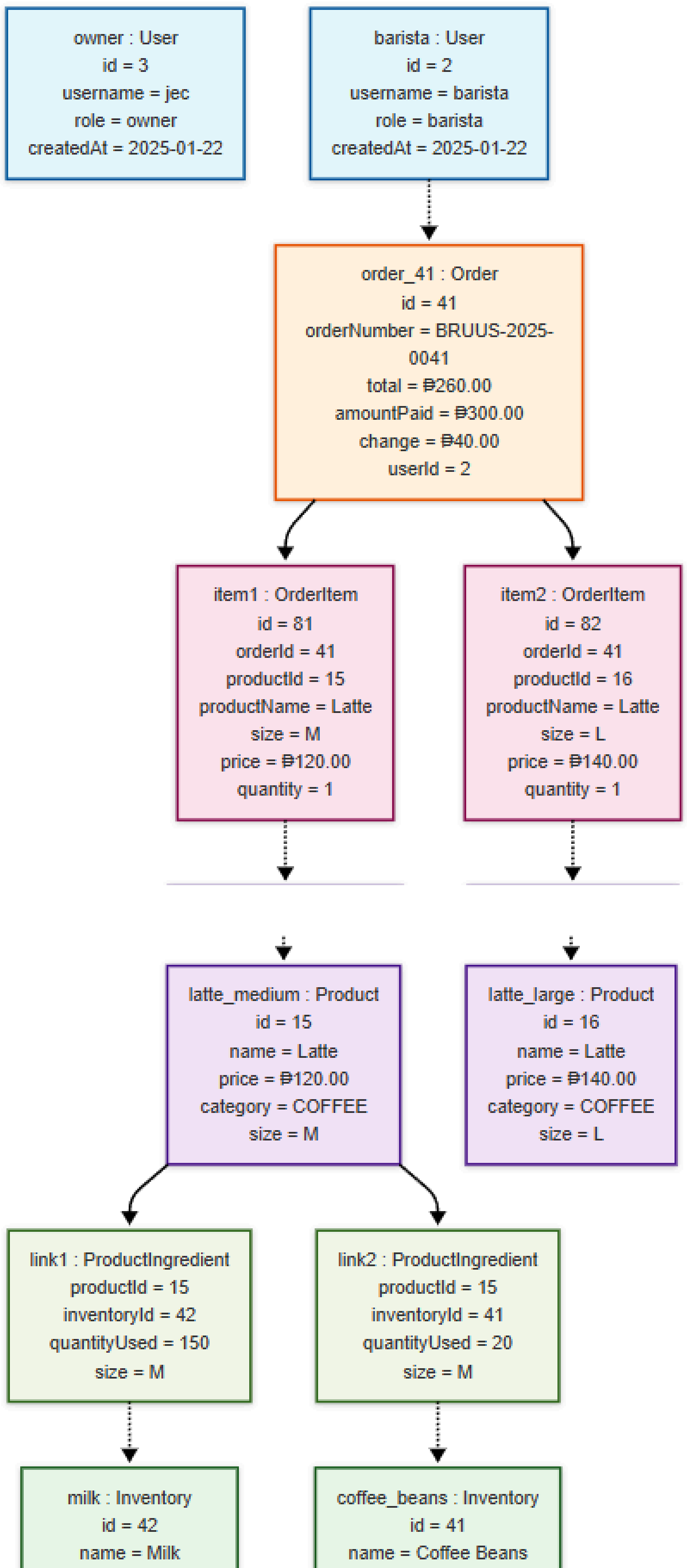
2.Sequence Diagram: Owner Management Flow



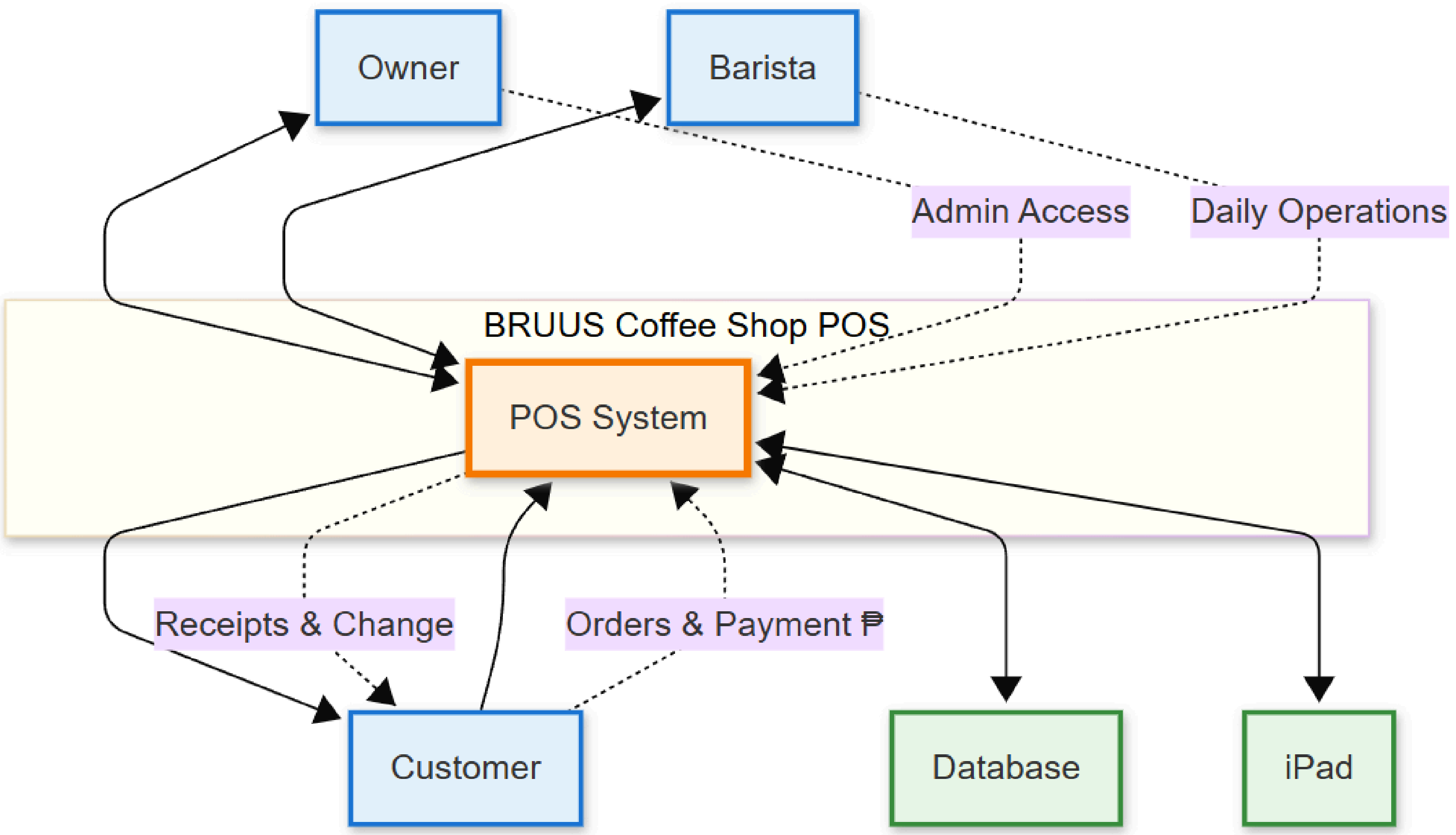
3.Class Diagram



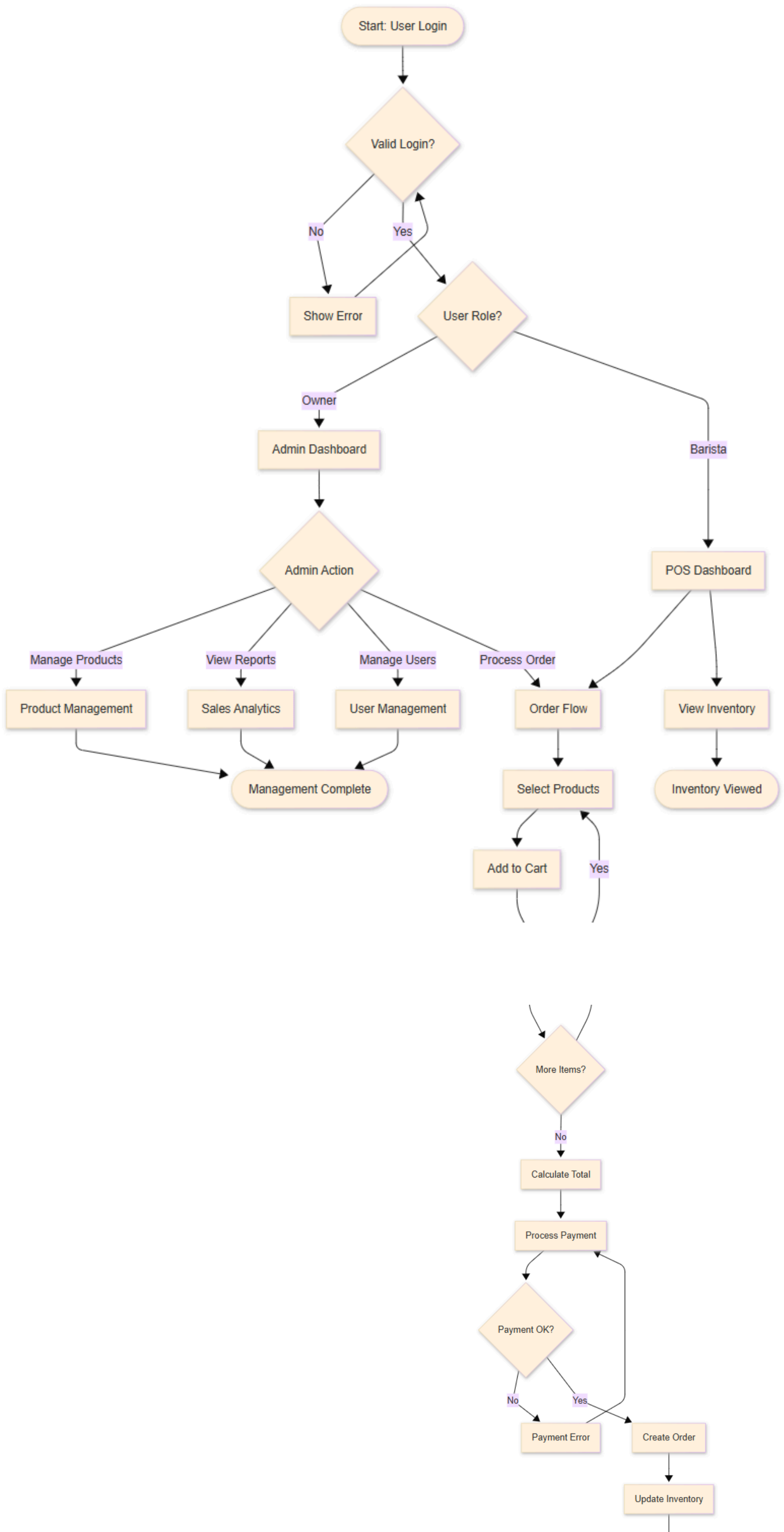
4.Object Diagram



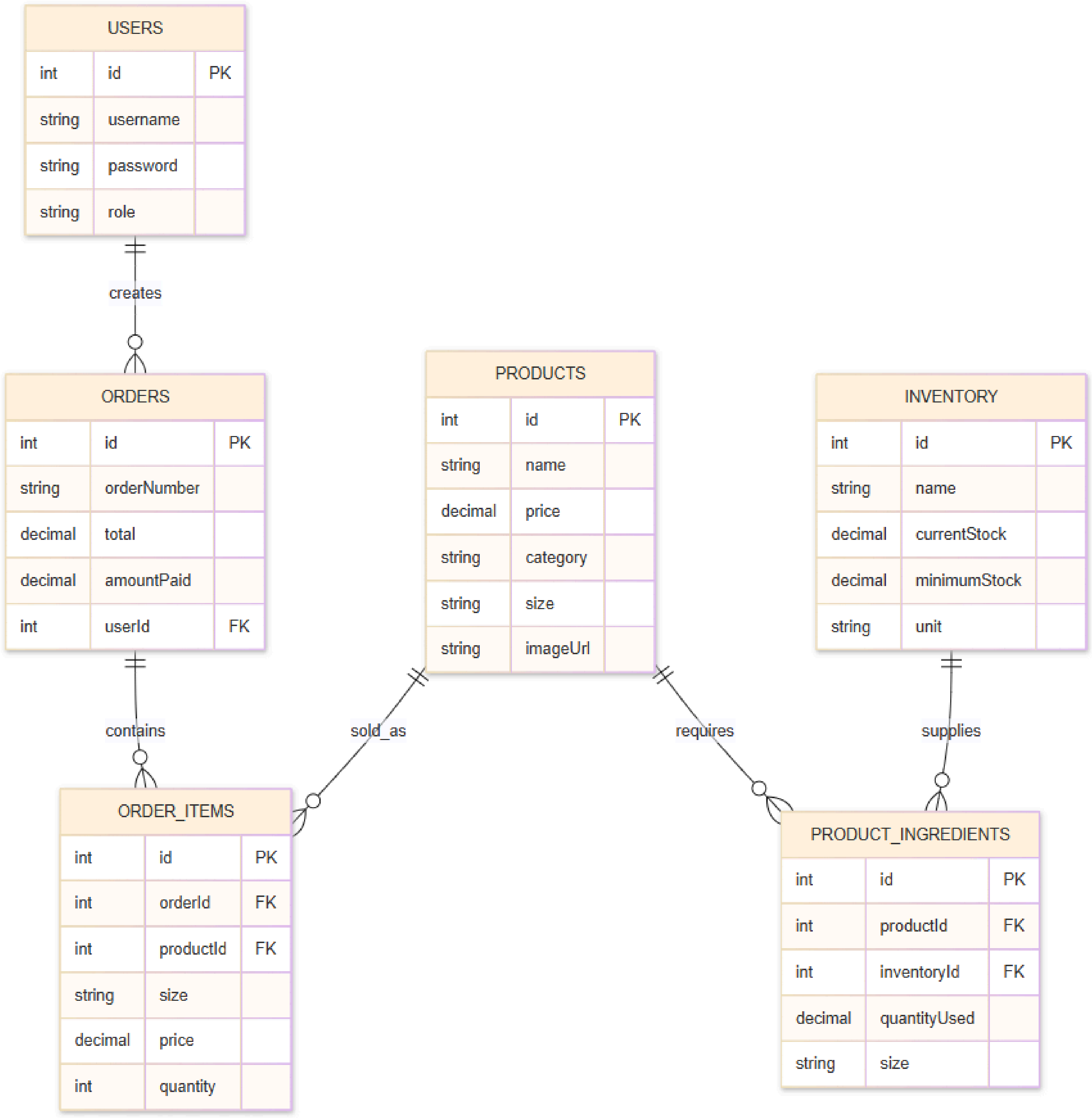
5.Context Diagram



Flowchart


Start: User Login
Valid Login?
No
Yes
Show Error
User Role?
Owner
Barista
Admin Dashboard
POS Dashboard
Admin Action
Manage Products
View Reports
Manage Users
Process Order
Product Management
Sales Analytics
User Management
Order Flow
View Inventory
Management Complete
Select Products
Inventory Viewed
Add to Cart
Yes
More Items?
No
Calculate Total
Process Payment
Payment OK?
No
Yes
Payment Error
Create Order
Update Inventory

Entity Relationship Diagram (for relation database)



C. Overview of the Technologies to be Used

Category	**Technologies**	**Purpose**
Frontend	React.js, HTML/CSS	Responsive UI with fixed layouts for POS and dashboard.
Backend	Node.js, Express.js	Scalable API development and real-time data sync.
Database	MySQL, Redis	Structured data storage and caching for performance optimization.
Cloud Storage	AWS S3	Secure storage of product images and backups.
Authentication	JWT, Bcrypt	Role-based access control and password hashing.
Security	HTTPS, AES-256 Encryption	Secure data transmission and storage.
Infrastructure	Docker, AWS EC2	Containerized deployment and scalable cloud hosting.

Design Considerations:

- Offline Mode: Orders are cached locally during network outages and synced when connectivity resumes.
- Scalability: Cloud-based architecture supports future expansion to multiple coffee shop branches.
- Usability: Fixed UI layouts and <2-click navigation to reduce barista training time.
- Reliability: Crash recovery features and automated backups ensure minimal downtime.

CONCLUSION

The proposed Bruus: Unified POS and Inventory Platform with Automated Alerts

offers a comprehensive solution to modernize coffee shop operations by seamlessly integrating point-of-sale (POS) functionality with real-time inventory management. Designed to address critical inefficiencies in existing systems—such as weak security, manual inventory tracking, disorganized interfaces, and lack of actionable insights—this platform leverages cutting-edge technologies (React.js, Node.js, MySQL) and agile development practices to deliver a robust, user-centric tool.

Key benefits of the system include:

1. Enhanced Security: Role-based access control ensures sensitive financial and inventory data are accessible only to authorized personnel.
2. Operational Efficiency: Automated inventory updates, and real-time low-stock alerts eliminate manual errors and reduce restocking delays by 80%.
3. Data-Driven Decision-Making: Daily/weekly sales analytics and non-selling product reports empower owners and managers to optimize menus and inventory orders.
4. Improved Customer Experience: A stable, intuitive UI with fixed layouts minimizes order processing time by 50%, reducing wait times and system crashes.
5. Scalability: Cloud-based architecture and offline mode support future expansion to multi-location operations.

By bridging the gap between sales and inventory management, the system not only streamlines workflows but also positions coffee shops to thrive in a competitive market. Its emphasis on automation, security, and usability ensures long-term sustainability, cost savings, and customer satisfaction.

Final Impact: This project transforms outdated, error-prone processes into a cohesive digital ecosystem, empowering coffee shop owners to focus on growth while delivering consistent, high-quality service.

Closing Statement:
The Bruus: Unified POS and Inventory Platform with Automated Alerts represents a strategic leap forward for coffee shops, combining innovation with practicality to redefine how small businesses manage sales and inventory in the digital age.