

Project Proposal: AI-Powered Retail Management & Marketing System

Document Version: 1.0

Date: August 12, 2025

1. Project Idea & Executive Summary

This project proposes the development of an intelligent management system designed for small to medium-sized retail businesses, such as local supermarkets and shops. The core of this initiative is to build a system that transcends traditional sales tracking by integrating Artificial Intelligence to solve two critical business challenges: **inventory optimization** and **personalized customer marketing**.

The system will leverage historical sales data to provide actionable intelligence. For inventory, it will employ predictive models to forecast optimal stock levels, aiming to significantly reduce wastage from overstocking perishable goods while preventing lost sales from stockouts. For marketing, it will analyze customer purchasing behavior to enable highly personalized and efficient notification campaigns, thereby increasing customer loyalty, retention, and lifetime value.

This Proof of Concept (PoC) will deliver a fully functional, end-to-end solution demonstrating the powerful synergy between modern data collection at the Point of Sale (POS), sophisticated AI-driven analysis, and automated business operations. The architecture is designed to be flexible, accommodating businesses with existing POS systems through integration or providing a complete, new system for those who require one.

2. Core Objectives for the Proof of Concept (PoC)

1. **Develop a Functional POS Client:** Create a modern, intuitive, and cross-platform desktop application using JavaFX for processing daily sales transactions efficiently.
2. **Build a Web-Based Admin Dashboard:** Provide a centralized, accessible web interface for managers to view key performance indicators, interact with AI-driven analytics, and manage marketing campaigns.
3. **Implement a Robust Backend Integration Hub:** Utilize Ballerina to build the central API gateway that will handle all business logic, orchestrate communication between all system components, and manage data flow securely.
4. **Create a Dedicated Analytics Service:** Develop a standalone Python-based microservice to house the machine learning models responsible for stock forecasting and customer segmentation.
5. **Demonstrate End-to-End Data Flow:** Showcase the complete data lifecycle, from a sale being registered at the POS to its direct impact on a stock prediction.

chart and its use in defining a targeted marketing campaign.

3. System Architecture

The system is designed with a modern, decoupled microservices-oriented architecture to ensure scalability, maintainability, and flexibility. Each component has a clearly defined responsibility.

- **1. Client Layer (User-Facing Applications):**
 - **POS Client:** The primary interface for cashiers to record transactions.
 - **Admin Dashboard:** The primary interface for managers to gain insights and control operations.
- **2. API & Integration Layer (The Central Hub):**
 - This layer, built with Ballerina, acts as the API Gateway. It is the single point of entry for all client applications and is responsible for authenticating requests, enforcing business rules, and orchestrating the backend services.
- **3. Analytics Layer (The Intelligence Engine):**
 - A specialized microservice dedicated to machine learning tasks. It is kept separate to allow for independent scaling and development and to leverage the best tools for data science.
- **4. Persistence Layer (The System's Memory):**
 - A central database that serves as the single source of truth for all application data, including transactions, customer profiles, product catalogs, and prediction results.

4. Finalized Technology Stack

Component	Technology Chosen	Rationale & Key Responsibilities
POS Client	JavaFX (using Java 17+)	A modern, cross-platform framework for building rich desktop UIs. Responsible for capturing sales data and communicating with the Ballerina API.
Admin Dashboard	React or Vue.js	Industry-standard JavaScript libraries for creating interactive, responsive, and data-rich web UIs. Will visualize data from the Ballerina API.

Backend API	Ballerina	Natively designed for integration and creating robust, network-aware services. It will expose the REST API, manage database connections, and orchestrate calls to the Python service.
Analytics Service	Python 3.x with FastAPI	Leverages Python's unparalleled ML ecosystem (pandas, scikit-learn, Prophet). Responsible for all predictive modeling and customer segmentation.
Database	Supabase (PostgreSQL)	Provides a robust, enterprise-grade SQL database with excellent BaaS features (like authentication and auto-generated APIs) that can accelerate PoC development.

5. Detailed Implementation Approach

5.1 AI-Powered Stock Management

1. **Data Capture:** The **JavaFX POS** captures every sale (item ID, quantity, timestamp) and sends the transaction data to a dedicated endpoint on the **Ballerina Backend**.
2. **Prediction Trigger:** A manager logs into the **Web Dashboard**, navigates to the inventory section, and requests an updated stock forecast for a specific item or category.
3. **Orchestration:** The **Ballerina Backend** receives the request and calls the appropriate endpoint on the **Python Analytics Service**, passing the relevant item ID and historical date range.
4. **Analysis:** The **Python Service** uses a time-series model (e.g., Prophet or ARIMA) to analyze the historical sales data for that item, identifying trends, weekly/seasonal patterns, and producing a forecast for the optimal stock quantity for the upcoming period.
5. **Response & Visualization:** The prediction data (e.g., predicted sales for the next 7 days or next month) is returned to Ballerina, which then forwards it to the **Web Dashboard** to be displayed as an interactive chart.

6. and for seasonal items there is a separate tab called seasonal analytics that only analyze seasonal items and show last sales and predict for next season like that. Seasonal item analysis is handled through a dedicated "Seasonal Analytics" tab. This feature provides comprehensive insights into seasonal product performance by analyzing past sales data. Furthermore, it leverages predictive analytics to forecast demand and trends for upcoming seasons, enabling better inventory management and marketing strategies for seasonal offerings.

5.2 Personalized Notification System

1. **Data Collection:** All customer transactions are continuously collected and stored in the **Database** via the **Ballerina Backend**.
2. **Customer Segmentation:** Periodically (e.g., via a nightly batch job) or on-demand, the **Ballerina Backend** triggers the **Python Service** to re-evaluate customer segments.
3. **Analysis:** The **Python Service** performs **RFM (Recency, Frequency, Monetary) analysis** on the entire customer transaction history. It then uses a clustering algorithm (e.g., K-Means) to group customers into meaningful segments like "Champion," "Loyal," "At-Risk," "New," etc.
4. **Data Update:** The resulting segment label for each customer is sent back to Ballerina and updated in the customer's profile in the **Database**.
5. **Campaign Targeting:** A manager uses the **Web Dashboard** to create a marketing campaign. They select a target segment (e.g., "At-Risk") and define a message template (e.g., "We've missed you! Here's 15% off.").
6. **Notification Dispatch:** Upon campaign launch, the **Ballerina Backend** queries the database for all customers in the target segment. It then iterates through this list, personalizes the message for each customer, and uses its native HTTP client to connect to a third-party messaging service API (e.g., Twilio, SendGrid) to dispatch the final notifications.

6. High-Level User Workflows

6.1 Cashier Workflow

1. Logs into the **JavaFX POS** application at the start of their shift.
2. Scans items, applies customer loyalty information, and processes payments for each transaction.
3. The sale is seamlessly sent to the **Ballerina** backend and recorded in the database in real-time.

6.2 Manager Workflow

1. Logs into the **Web Dashboard** from any device with a web browser.
2. Views a summary of the day's performance: total sales, number of customers,

and top-selling products.

3. Navigates to the "Inventory Analytics" page to check the AI-generated stock forecasts for perishable items that need to be reordered.
4. Navigates to the "Customer Relations" page, reviews the customer segmentation dashboard, and notices a growing "At-Risk" segment.
5. Creates and launches a targeted discount campaign to re-engage these specific customers, all from within the dashboard interface.