

Project Summary & Plan of Action: Cloud-Based Multi-Branch Bakery ERP in VB.NET

1. Project Overview

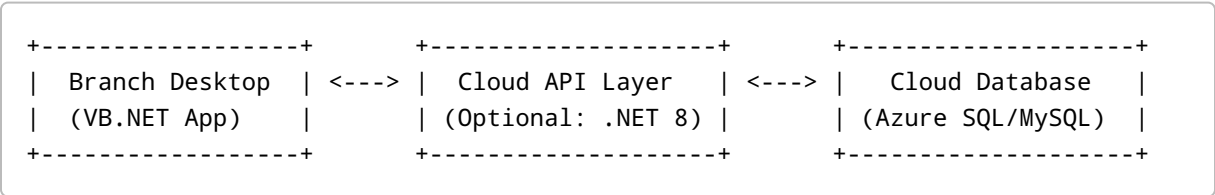
- **Application Type:** Full-stack desktop application in **VB.NET**
- **Key Modules:**
  - Warehousing / Stock Room
  - Manufacturing
  - Retail (POS)
  - Accounting
- **Multi-branch Support:** Yes
- **Backend:** Cloud-based (no local backend)

2. Cloud Backend Recommendation

Option	Pros	Cons	Estimated Cost (July 2025)
<b>Azure SQL Database</b>	Seamless Visual Studio integration, scalable, secure	Monthly cost applies	\~\~\$5-\~\$8/month (Basic Tier)
<b>PlanetScale (MySQL)</b>	Zero maintenance, serverless, free tier	Free tier limits, newer tech	Free to \~\~\$10/month
<b>Amazon RDS / DigitalOcean</b>	Scalable, predictable costs	Slightly more setup	\~\~\$15-\~\$30/month

**Recommendation:** Start with **Azure SQL Database (Basic Tier)** for smooth integration with Visual Studio and .NET.

3. Suggested Architecture



- Option A: Direct connection to Cloud DB (faster to build)
- Option B: Connect via API (better security and future scalability)

## 4. Modular Build Approach (Recommended)

Build the system in the following modular order:

**1. Warehousing / Stock Room Module:**

2. Manage raw materials, inventory, and stock levels
3. Generate **Purchase Orders (POs)** for suppliers or Manufacturing

**4. Manufacturing Module:**

5. Track production, batch management, and work-in-progress
6. Generate **Transfer Orders** for Retail (finished goods)

**7. Retail (POS) Module:**

8. Sales, invoicing, stock decrement from Manufacturing
9. Cash management and shift closure

**10. Accounting Module:**

11. Capture all financial transactions from other modules
12. Generate financial reports, manage journals, perform reconciliations

**Key Interactions:**

- Warehousing → Manufacturing (Stock Requests/POs)
  - Manufacturing → Retail (Goods Transfer)
  - Retail → Accounting (Sales & Revenue)
  - All → Accounting (Journals & Ledgers)
- 

## 5. Accounting Module Design

**Key Features:**

- **General Ledger:** Chart of accounts, double-entry journal system
- **Stock Management:** FIFO, LIFO, weighted average, real-time updates
- **Accounts Payable & Receivable:** Invoices, payments, aging reports
- **Bank Reconciliation:** Import bank statements (CSV), reconcile transactions
- **Financial Reporting:** Trial Balance, Income Statement, Balance Sheet
- **Tax / VAT Compliance:** Optional but recommended

## Best Practices:

- Enforce **double-entry** integrity
- Prevent editing of **posted journals** (use adjustments)
- Implement **financial period locking**
- Support **branch-level segregation** and **consolidated reporting**

## 6. Pastel Accounting Integration

Method	How it Works	Tools Required
<b>CSV/Excel</b>	Export transactions, import into Pastel	Simple, low-cost
<b>ODBC</b>	Read/write directly to Pastel DB	Pastel installed + ODBC drivers
<b>Pastel SDK/API</b>	Real-time automated integration	SDK license (extra cost)

**Recommendation:** Start with **CSV-based export/import** for lowest cost and fastest deployment.

## 7. Cost Summary

- **Azure SQL Database (Basic Tier):** ~\$5-\$8/month
- **Visual Studio:** Free (Community Edition) or licensed
- **Optional Azure Web API Hosting:** Starting ~\$5/month (if used)

## 8. Plan of Action

Phase	Task	Deliverable
Phase 1	Set up <b>Azure SQL Database</b> & design database schema	Cloud database ready
Phase 2	Build <b>Warehousing / Stock Module</b>	Purchase Order generation, stock management
Phase 3	Build <b>Manufacturing Module</b>	Production tracking, transfer to retail
Phase 4	Build <b>Retail (POS) Module</b>	Sales, invoicing, daily reports
Phase 5	Build <b>Accounting Module</b> with integration to Pastel (start with CSV export)	Financials, ledgers, reports
Phase 6	Implement <b>Security, Role Management, and Reporting</b>	User access controls & reports

Phase	Task	Deliverable
Phase 7	Testing, Deployment, and Training	Live system with documentation

**End of Summary & Action Plan**