# Inventory Management System

A complete inventory management application built with VB.NET and DevExpress WinForms components.

## Features

* **Dashboard** - Real-time metrics with interactive tiles and charts
* **User Authentication** - Role-based login with SHA256 password hashing
* **Catalog Management** - Items, Categories, Suppliers, Units of Measure
* **Order Processing** - Create, edit, and track customer orders with line items
* **Purchase Management** - Purchase orders with supplier integration
* **Inventory Control** - Stock levels, physical counts, stock history audit trail
* **Reports** - Inventory valuation, order reports, reorder alerts
* **Administration** - User management, payment terms, system settings

## 

## Dashboard

The Dashboard provides an at-a-glance overview of key business metrics using DevExpress TileControl and ChartControl.

### Live Metric Tiles

| Tile | Description | Click Action |
| --- | --- | --- |
| **Inventory Value** | Total stock value across all items | Opens Inventory Report |
| **Items** | Count of active items in catalog | - |
| **Low Stock** | Items below reorder level (red when > 0) | Opens Reorder Report |
| **Pending Orders** | Orders in Draft/Submitted status | Opens Order List |
| **Pending PO** | Purchase orders awaiting receipt | Opens Purchase List |
| **Today’s Sales** | Total value of orders delivered today | - |

### Stock Value by Category Chart

A pie chart showing inventory value distribution across product categories with: - Interactive segments (click to explode) - Percentage labels - Category legend

### Quick Actions

Shortcut buttons for common tasks: - New Order - New Purchase Order - Inventory Count - Reorder Report

### Auto-Refresh

Dashboard metrics automatically refresh every 60 seconds to show current data.

## Technology Stack

* **Framework:** .NET 8.0 Windows Forms
* **Language:** VB.NET
* **UI Components:** DevExpress WinForms 24.2
* **Database:** Microsoft SQL Server
* **Architecture:** 3-tier (Presentation, Business Logic, Data Access)

## 

## Default Login Credentials

| Username | Password | Role |
| --- | --- | --- |
| admin | password123 | Administrator |
| manager | password123 | Manager |
| sales | password123 | Sales |
| warehouse | password123 | Warehouse |
| purchasing | password123 | Purchasing |

## 

## Project Structure

InventoryManagement/  
├── Database/  
│ ├── 01\_CreateDatabase.sql # Database schema (13 tables)  
│ └── 02\_SampleData.sql # Sample data for testing  
├── Common/  
│ ├── AppSettings.vb # Singleton settings manager  
│ ├── DatabaseHelper.vb # ADO.NET database wrapper  
│ ├── SessionManager.vb # User session management  
│ └── Enums.vb # Enumerations  
├── Entities/ # 14 entity classes  
│ ├── User.vb  
│ ├── Item.vb  
│ ├── Category.vb  
│ ├── Supplier.vb  
│ ├── OrderHeader.vb  
│ ├── OrderLine.vb  
│ ├── PurchaseHeader.vb  
│ ├── PurchaseItem.vb  
│ └── ...  
├── DataAccess/ # Data Access Objects (DAO)  
│ ├── BaseDAO.vb  
│ ├── UserDAO.vb  
│ ├── ItemDAO.vb  
│ ├── OrderDAO.vb  
│ └── ...  
├── BusinessLogic/ # Business layer managers  
│ ├── AuthenticationManager.vb  
│ ├── InventoryManager.vb  
│ ├── OrderManager.vb  
│ └── PurchaseManager.vb  
└── Forms/ # UI Forms  
 ├── LoginForm.vb  
 ├── MainForm.vb # Ribbon-based main form  
 ├── Admin/ # User, Settings, Payment Terms  
 ├── Catalog/ # Items, Categories, Suppliers  
 ├── Orders/ # Order list and edit forms  
 ├── Purchases/ # Purchase order forms  
 ├── InventoryControl/ # Stock levels, counts, history  
 └── Reports/ # Reporting forms

## Database Setup

1. Open SQL Server Management Studio (SSMS)
2. Connect to your SQL Server instance
3. Execute Database/01\_CreateDatabase.sql to create the database and tables
4. Execute Database/02\_SampleData.sql to insert sample data

## Configuration

Update the connection string in Common/AppSettings.vb (line 13) to match your SQL Server:

' Default (SQL Express):  
Private Const DEFAULT\_CONNECTION\_STRING As String = "Server=.\SQLEXPRESS;Database=InventoryManagement;Trusted\_Connection=True;TrustServerCertificate=True;"  
  
' Local default instance:  
Private Const DEFAULT\_CONNECTION\_STRING As String = "Server=.;Database=InventoryManagement;Trusted\_Connection=True;TrustServerCertificate=True;"  
  
' Named instance:  
Private Const DEFAULT\_CONNECTION\_STRING As String = "Server=.\YourInstanceName;Database=InventoryManagement;Trusted\_Connection=True;TrustServerCertificate=True;"

## Build and Run

### Prerequisites

* .NET 8.0 SDK
* DevExpress WinForms 24.2 license
* SQL Server (Express or higher)

### Command Line

cd InventoryManagement  
dotnet build  
dotnet run

### Visual Studio

1. Open InventoryManagement.sln
2. Ensure DevExpress license is installed
3. Press F5 to build and run

## Key Business Logic

### Stock Validation

* **In\_Stock\_Check**: Validates stock availability before order processing
* **Negative\_Stock\_Allowed**: Controls whether stock can go negative

### Moving Average Cost

When receiving purchase orders, item costs are recalculated using weighted moving average:

New Cost = ((Current Qty × Current Cost) + (Received Qty × Purchase Cost)) / (Current Qty + Received Qty)

### Audit Trail

All stock movements are logged to the STOCK\_HISTORY table including: - Purchase receipts - Sales/order deliveries - Inventory adjustments - Physical count variances

## 

## User Roles and Permissions

| Role | Access |
| --- | --- |
| Administrator | Full access to all modules |
| Manager | All except user administration |
| Sales | Orders, Catalog (view), Reports |
| Warehouse | Inventory Control, Catalog (view) |
| Purchasing | Purchases, Suppliers, Catalog |

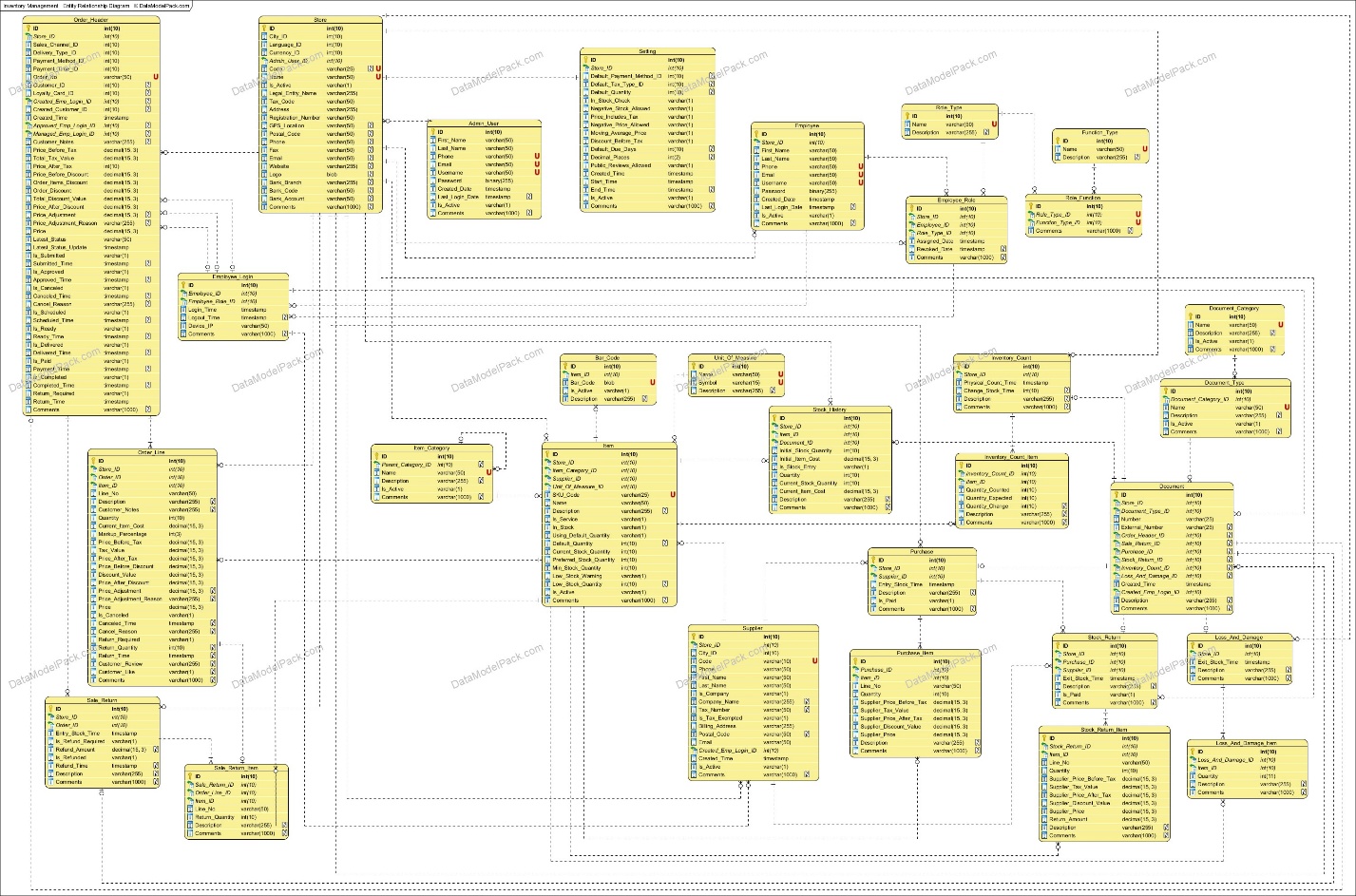
## 

## Database Schema

The system uses 13 main tables: - USER - System users - STORE\_SETTINGS - Application configuration - LOV - List of values (lookups) - CATEGORY - Product categories (hierarchical) - UNIT\_OF\_MEASURE - Units for items - SUPPLIER - Vendor information - PAYMENT\_TERM - Payment terms - ITEM - Product catalog - ORDER\_HEADER / ORDER\_LINE - Customer orders - PURCHASE\_HEADER / PURCHASE\_ITEM - Purchase orders - STOCK\_HISTORY - Stock movement audit - INVENTORY\_COUNT - Physical count records

## License

* This project requires a valid DevExpress WinForms license.
  + https://www.devexpress.com/products/net/controls/winforms/
* This project requires a valid DataModelPack license.
  + https://datamodelpack.com/data-models/inventory-data-model.html



CLAUDE.AI Prompt

I want to build a complete inventory management application. It needs to be a complete solution that follows the class and database schema diagrams as outlined in the attached mermaid files and coded using VB.Net and DevExpress winforms components if needed. I have also attached a readme.md file as an outline. This solution should start with a simple user login and include a way for an administrator to handle users. Add CRUD vb.net forms to interact with this data that are called from a main system form with menu that includes report options. Include MSSQL database scripts to create the database tables. Also scripts to add limited sample data for initial database testing.

