# Project Report: Project1-InventoryManagement

# 

**Submitted By:** GROUP NO 12

Group Members:

Zain ul abideen Ali 233604 BSCS B

Arfa Imran 233596 BSCS B

Fatima Khalid 233526 BSCS B

**Course Title:** VISUAL PROGRAMMING

**Semester:** 3RD

**Session:** 2023-2027

**Department:** COMPUTER SCIENCE

**Submitted To:** MAM AATKA ALI

Table of Contents

[Project Report: Project1-InventoryManagement 1](#_Toc185975013)

[1. Introduction 3](#_Toc185975015)

[2. Objectives 3](#_Toc185975016)

[3. Database Design 3](#_Toc185975017)

[3.1 Database Structure 3](#_Toc185975018)

[3.2 Relationships 4](#_Toc185975019)

[4. Application Development 4](#_Toc185975020)

[4.1 Technology Stack 4](#_Toc185975021)

[4.2 Key Features 5](#_Toc185975022)

[4.2.1 Admin Features 5](#_Toc185975023)

[4.2.2 Manager Features 5](#_Toc185975024)

[4.2.3 Staff Features 5](#_Toc185975025)

[4.3 User Interface 6](#_Toc185975026)

[5. Challenges Faced 6](#_Toc185975027)

[6. Conclusion 6](#_Toc185975028)

[7. Recommendations 7](#_Toc185975029)

[8.Testing 7](#_Toc185975030)

[Testing Strategies 7](#_Toc185975031)

[Test Cases 7](#_Toc185975032)

[9.Challenges and Solutions 7](#_Toc185975033)

[Challenges 7](#_Toc185975034)

[Solutions 7](#_Toc185975035)

# 1. Introduction

The Inventory Management System (IMS) aims to streamline the management of inventory-related tasks within an organization. This project involves creating a robust database and a user-friendly desktop application that enables users to manage suppliers, products, orders, and stock movements efficiently.

# 2. Objectives

* To develop a relational database that handles various inventory management tasks.
* To create a WPF application that provides an intuitive interface for users to interact with the database.
* To ensure data integrity and security through proper validation and user roles.

## 3. **Database Design**

## 3.1 Database Structure

The InventoryManagement database consists of the following tables:

1. **Suppliers** - Stores information about suppliers including contact details.
2. **PurchaseOrders** - Tracks orders placed with suppliers, including order status and total amount.
3. **PurchaseOrderDetails** - Contains details about products included in each purchase order.
4. **SalesOrders** - Manages customer orders, their status, and total amounts.
5. **SalesOrderDetails** - Details products associated with sales orders.
6. **Products** - Maintains product information including SKU, category, and pricing.
7. **StockMovements** - Records movements of stock (in, out, adjustments) for inventory tracking.
8. **Users** - Manages user accounts and roles (Admin, Manager, Staff).

**9) Categories** - Categorizes products for better organization.

## 3.2 Relationships

* Suppliers to PurchaseOrders: One-to-Many
* PurchaseOrders to PurchaseOrderDetails: One-to-Many
* SalesOrders to SalesOrderDetails: One-to-Many
* Products to PurchaseOrderDetails and SalesOrderDetails: One-to-Many
* Users to AuditLogs: One-to-Many

# 4. Application Development

## 4.1 Technology Stack

* **Frontend:** XAML
* **Backend:** C#
* **Database:** Microsoft SQL Server
* **Framework:** .NET Framework
* **Libraries:** Entity Framework for data access
* **Development Tools:** Visual Studio, SQL Server Management
* **Hashing :** BCrypt

## 4.2 Key Features

### 4.2.1 Admin Features

* **User Management:** Ability to add, update, and delete user accounts and assign roles.
* **Product Management:** Add, update, and delete products in the inventory.
* **Supplier Management:** Manage suppliers, including adding new suppliers and updating existing ones.
* **Role-Based Access Control:** Define permissions for different user roles to enhance security.

### 4.2.2 Manager Features

* **Sales Order Management:** Create, update, and view sales orders.
* **Stock Management:** Track stock movements and maintain accurate inventory levels.
* **Reporting:** Generate reports on sales, stock levels, and supplier performance.
* **Forecasting:** Analyze historical data to predict future stock requirements and trends.

### 4.2.3 Staff Features

* **Basic Inventory Management:** View products, suppliers, and stock movements.
* **Stock Movement Recording:** Input stock movements for products and update quantities.
* **Sales Order Processing:** Manage customer sales orders and update their statuses.

### 4.3 User Interface

The UI is designed to be intuitive, with separate tabs for managing:

* **Products**
* **Suppliers**
* **Sales Orders**
* **Stock Movements**

Each section allows for easy navigation and data manipulation.

# 5. Challenges Faced

* **Data Integrity:** Ensuring referential integrity between tables posed initial challenges, which were addressed through careful design of relationships.
* **User Experience:** Balancing functionality with ease of use required multiple iterations on the UI design.
* **Security:** Implementing secure user authentication and authorization was critical for protecting sensitive data.

# 6. Conclusion

The Inventory Management System successfully meets the project objectives by providing a comprehensive solution for managing inventory. The combination of a well-structured database and a user-friendly WPF application facilitates efficient operations within an organization. Future enhancements may include integration with external systems and advanced reporting features.

# 7. Recommendations

* **User Training:** Conduct training sessions for users to maximize the system's potential.
* **Regular Backups:** Implement a backup strategy to prevent data loss.
* **Feedback Mechanism:** Establish a method for users to provide feedback for continuous improvement.
* **System Updates:** Regularly update the system to include new features and improvements based on user feedback

# 8.Testing

## Testing Strategies

* Manual testing for input validation and cart functionalities.

## Test Cases

* Ensure valid input handling for price and quantity.
* Verify correct cart operations for adding, viewing, and removing products.

# 9.Challenges and Solutions

## Challenges

* Handling incorrect user inputs gracefully.

## Solutions

* Implemented robust input validation with prompts for re-entry.