RK.UNIVERSITY

A PROJECT REPORT

For

An Inventory Management System

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

Bachelors in computer Application

TO

…. UNIVERSITY, RAJKOT

SUBMITTED BY

Name of Student	**Enrollment No.**
Chothani Priya	[…………………]

UNDER THE GUIDANCE OF

……………………..
Assistant Professor
….. University,
RAJKOT

……………….
Professor (HOD),
…..University,
RAJKOT

SCHOOL OF ENGINEERING
RK UNIVERSITY

SCHOOL OF ENGINEERING, RK UNIVERSITY, RAJKOT

RK UNIVERSITY

DECLARATION

The proposed Inventory Management System (IMS) simplifies the process of tracking, managing, and replenishing stock for businesses of all sizes. It offers tools for real-time stock monitoring, automated order placements, and supplier management, ensuring accurate and efficient inventory control. The IMS helps prevent overstocking or stockouts, reduces manual tasks, and lowers operational costs. Scalable and flexible, it adapts to the needs of both small businesses and large enterprises, improving overall supply chain efficiency with detailed reporting and data insights.

Project Team Member

Student Name: Chothani Priya ………. (……..)

Date: 0_/0_/-202_

Place: - Rajkot

TABLE OF CONTENTS------------------------------

1. Introduction---

1.1 Purpose

The purpose of this document is to specify the software requirements for the **Inventory Management System (IMS)**, version 1.0, a standalone desktop application built using **C#**. The system aims to help businesses efficiently manage product inventories, suppliers, purchases, and customer orders. This SRS covers the functional, non-functional, and design requirements necessary for the development and deployment of the system.

1.2 Document Conventions

Font and Style: All headings are presented in bold. Code snippets, database table names, and system terminology are presented in **monospace font**.

- **Standard Terminologies**:
  - "User" refers to system users like **Admin**, **Dealer**, or **Stock Manager**.
  - "System" refers to the **Inventory Management System (IMS)**.
  - "Stock" refers to the current inventory of products.

1.3 Intended Audience and Reading Suggestions

Intended Audience

- **Developers**: To understand the system's functionality for implementation.
- **Project Managers**: To ensure the system meets business objectives and timelines.
- **Testers**: To validate that the system functions as expected.
- **Marketing Staff**: To learn about the system's features for promotion.
- **Users**: To understand how the system operates.
- **Documentation Writers**: To develop user manuals and training materials.

Suggested Reading Sequence

- Start with **Overview and Scope** for a general understanding of the system.
- Continue with **Functional Requirements** for developers and testers.

Read **Non-functional Requirements** for project managers and system architects.

1.4 Project Scope

The **Inventory Management System (IMS)** is designed to streamline the process of managing inventory across multiple dealers, suppliers, and warehouses. It automates core processes like:

- **Product Management**: Allowing users to add, update, and delete product information.
- **Stock Management**: Providing real-time tracking of product stock levels, units, and sales transactions.
- **Order Management**: Processing sales and purchase orders, managing bill types, and tracking order details.
- **Dealer Management**: Storing and handling dealer-related information for smooth business transactions.

The system aligns with corporate goals to improve operational efficiency and minimize costs associated with poor inventory management. It is designed to be user-friendly and ensure stock levels are optimized, orders are fulfilled in a timely manner, and profitability is calculated.

1.5 References

❖ **User Interface Guidelines for C# Desktop Applications**:

- **Title**: Microsoft UI Style Guide
- **Author**: Microsoft
- **Version**: 2023
- **Location**: https://learn.microsoft.com/en-us/windows/apps/design/

❖ **System Requirements Specification for Inventory Management System**:

- **Title**: IMS Technical Documentation
- **Author**: Project Development Team
- **Version**: 1.0
- **Date**: August 2024

❖ **Vision and Scope Document**:

- **Title**: IMS Vision and Scope
- **Author**: Priya Chothani
- **Version**: 1.0
- **Date**: July 2024
- **Location**: Internal documentation repository

2. Overall Description --

2.1 Product Perspective

The **Inventory Management System (IMS)** is a standalone, self-contained desktop application built using **C#**. It is designed to provide businesses with tools to efficiently manage their product inventories, orders, purchases, and dealers. This system is not a part of any product family but serves as a new, complete solution aimed at replacing manual inventory management processes.

The IMS is a single application that operates independently and does not rely on any larger system for functionality. However, it can integrate with third-party applications, such as accounting software, via data exports (CSV, Excel) if needed.

2.2 Product Features

1) **User Registration and Authentication**: Allows users (Admins, Dealers, Stock Managers) to register, login, and access various features based on their role.
2) **Product Management**: Add, update, and delete product details, such as name, unit, and category.
3) **Stock Management**: Track product quantities in real-time and update stock levels when purchases or sales occur.
4) **Order Processing**: Manage and process both sales and purchase orders. Allows users to generate invoices, select bill types, and track order status.
5) **Dealer Management**: Store and manage information related to dealers, including company details and contact information.
6) **Purchase Management**: Log and track product purchases, calculate costs, and manage suppliers.
7) **Reporting**: Generate reports for stock levels, sales, purchases, and profitability analysis.
8) **Security**: Role-based access control, data encryption, and secure login for system integrity.

2.3 User Classes and Characteristics

The IMS will have different user classes, each with specific roles and privileges:

1. Admin:

- Full access to all system functionalities.
- Can add, update, or remove users and products.
- Responsible for managing user privileges, overseeing stock, and generating

reports.

2. **Stock Manager:**

- Focuses on managing product inventories.
- Can add or update stock levels, but cannot modify user accounts or dealer information.

3. **Dealer:**

- Has restricted access to their own transactions and orders.
- Can place orders, view order statuses, and update personal dealer details.

4. **Guest/User:**

- Limited to viewing product catalogs and basic system information without making changes.

2.4 Operating Environment

1. **Hardware:**
   - Minimum: Dual-core processor, 4 GB RAM, 500 MB storage.
   - Recommended: Quad-core processor, 8 GB RAM, 1 GB storage for handling larger datasets.

2. **Operating System:**
   - Windows 10 or higher (64-bit version).

3. **Software:**
   - .NET Framework 4.8 or higher.
   - SQL Server 2019 for database management.

4. **External Software Dependencies:**
   - Microsoft Excel/CSV for report exports.
   - Printer drivers for invoice printing.

2.5 Design and Implementation Constraints

- **Corporate Policies**: Adherence to internal security policies, such as data encryption standards and role-based access control, will be mandatory.
- **Hardware Limitations**: The system will be optimized for performance within limited memory (minimum 4GB RAM) and storage (500MB disk space).
- **Database Requirements**: **SQL Server** will be used for database operations; no alternative database management system is supported.

- **Language Requirements**: The system will be developed using **C#** and will follow standard **.NET Framework** conventions.
- **Parallel Operations**: The system should allow multiple users to perform operations concurrently without data loss or corruption.
- **Security Considerations**: User authentication and role-based authorization will ensure only authorized users can access specific functions.

2.6 User Documentation

- **User Manual**: A detailed manual explaining how to navigate and use each of the system's features.
- **Installation Guide**: Step-by-step instructions for installing the desktop application on **Windows** systems.
- **Online Help**: In-app help feature offering brief descriptions of the key features and workflows.
- **Tutorials**: A set of tutorials for new users, guiding them through typical use cases, such as adding a product, processing an order, or generating a stock report.

2.7 Assumptions and Dependencies

- **Operating Environment**: The system assumes that it will operate on Windows OS and that users have basic knowledge of Windows navigation and application use.
- **Database Availability**: It is assumed that **SQL Server** will be available and properly configured prior to system deployment.
- **Third-party Components**: The project assumes the availability of third-party components, such as Microsoft Excel, for generating reports.
- **Hardware Resources**: The system assumes that client machines meet or exceed the recommended hardware requirements.
- **External Software**: Dependencies on external software like printer drivers and Excel (for reports) are assumed to be functioning correctly; errors in these will limit the system's reporting and output functionality.

If any of these assumptions prove incorrect, project timelines and functionality could be affected. Likewise, if third-party software components change or become unavailable, alternative solutions will need to be evaluated.

3. System Features --

3.1 System Feature

1: Product Management

3.1.1 Description and Priority

The **Product Management** feature allows users (admins and stock managers) to add, update, delete, and view product details such as name, price, category, and quantity. It is a **High Priority** feature because it serves as the core functionality for managing inventory data.

- **Benefit**: 9/9 – Critical for inventory tracking.
- **Penalty**: 8/9 – Poor product management may lead to inventory mismanagement.
- **Cost**: 5/9 – Moderate development cost.
- **Risk**: 4/9 – Low risk, though errors could impact the integrity of the database.

3.1.2 Stimulus/Response Sequences

- **User Action**: Admin or Stock Manager opens the product management interface and selects "Add Product."
  - **System Response**: Displays a form for entering product details (name, category, quantity, price, supplier).
- **User Action**: Admin enters details and clicks "Save."
  - **System Response**: System validates inputs and adds the product to the inventory database.
- **User Action**: Admin selects a product and chooses "Edit Product."
  - **System Response**: Displays the product details in an editable form, allowing modifications.
- **User Action**: Admin selects "Delete Product."
  - **System Response**: System prompts for confirmation and removes the product from the database upon confirmation.

3.1.3 Functional Requirements

REQ-1: The system shall allow users to add new products with details such as name, category, price, and quantity.

REQ-2: The system shall validate the inputs to ensure no fields are left blank or have invalid data (e.g., negative quantities).
REQ-3: The system shall allow users to update product details.
REQ-4: The system shall allow users to delete a product, provided it is not associated with any pending orders.
REQ-5: The system shall display an error message if invalid data is entered during product creation or update.
REQ-6: The system shall generate a confirmation prompt before deleting a product.

System Feature 2: Stock Management

3.2.1 Description and Priority

The **Stock Management** feature enables real-time tracking of stock levels. Users can view, search, and update stock quantities. This feature is **High Priority** due to its critical role in ensuring the inventory reflects actual stock levels.

- **Benefit**: 9/9 – Prevents stock shortages and overstocking.
- **Penalty**: 7/9 – Inaccurate stock levels could lead to order failures or missed sales.
- **Cost**: 4/9 – Moderate cost to implement real-time updates.
- **Risk**: 3/9 – Low risk, but any discrepancies could disrupt sales or purchasing.

3.2.2 Stimulus/Response Sequences

- **User Action**: Stock Manager opens the "Stock Overview" screen.
    - **System Response**: Displays the current stock levels for all products, along with search and filter options.
- **User Action**: Stock Manager searches for a specific product.
    - **System Response**: Filters and displays the matching product with its stock level.
- **User Action**: Stock Manager selects "Update Stock" for a product.
    - **System Response**: Displays a form where the quantity can be adjusted.
- **User Action**: Stock Manager enters a new quantity and saves changes.
    - **System Response**: System validates the quantity and updates the database.

3.2.3 Functional Requirements

REQ-1: The system shall display the current stock levels for all products.
REQ-2: The system shall allow users to search for products by name or category.
REQ-3: The system shall allow users to update the stock quantity for any product.
REQ-4: The system shall validate stock levels to ensure they are non-negative.
REQ-5: The system shall log all stock adjustments for auditing purposes.

System Feature 3: Order Management

3.3.1 Description and Priority

The **Order Management** feature allows users to process sales and purchase orders. It tracks order statuses and generates invoices. This feature is of **High Priority** due to its direct impact on business transactions.

- **Benefit**: 8/9 – Essential for managing customer orders and purchases.
- **Penalty**: 7/9 – Failure in this feature may lead to loss of revenue.
- **Cost**: 6/9 – High development effort due to invoice generation and order tracking complexity.
- **Risk**: 5/9 – Moderate risk, as issues in order processing could impact business operations.

3.3.2 Stimulus/Response Sequences

- **User Action**: Admin or Dealer selects "Create Order."
  - **System Response**: Displays a form to enter customer details, product selection, and quantity.
- **User Action**: Admin enters order details and submits the form.
  - **System Response**: Validates the stock levels and creates a new order, generating an invoice if the stock is available.
- **User Action**: Admin selects an existing order and chooses "View Order Status."
  - **System Response**: Displays the order status (Pending, Fulfilled, Shipped).
- **User Action**: Admin selects "Generate Invoice."
  - **System Response**: Generates a PDF invoice for the order and updates order status.

3.3.3 Functional Requirements

REQ-1: The system shall allow users to create new sales and purchase orders.
REQ-2: The system shall validate stock levels before confirming an order.
REQ-3: The system shall generate an invoice for each confirmed order.
REQ-4: The system shall allow users to view and update the status of any order.
REQ-5: The system shall provide an option to export the invoice in PDF format.
REQ-6: The system shall allow users to cancel orders that are still in a pending state.

System Feature 4: Dealer Management

3.4.1 Description and Priority

The **Dealer Management** feature allows admins to add, update, and delete dealer information. This feature is **Medium Priority**, as it supports the smooth operation of the system but is not as critical as core inventory features.

- **Benefit**: 6/9 – Facilitates better dealer interaction.
- **Penalty**: 5/9 – Missing or incorrect dealer information could complicate orders.
- **Cost**: 3/9 – Low development cost.
- **Risk**: 2/9 – Low risk, but incorrect data could lead to confusion during transactions.

3.4.2 Stimulus/Response Sequences

- **User Action**: Admin selects "Add Dealer" from the dealer management screen.
  - **System Response**: Displays a form to enter dealer name, contact details, and company information.
- **User Action**: Admin enters dealer details and clicks "Save."
  - **System Response**: System validates the input and stores the dealer information in the database.
- **User Action**: Admin selects a dealer and clicks "Edit."
  - **System Response**: Displays the dealer’s current details, allowing the admin to update them.

3.4.3 Functional Requirements

REQ-1: The system shall allow users to add new dealers with information such as name, contact details, and address.
REQ-2: The system shall allow users to edit existing dealer details.
REQ-3: The system shall allow users to delete dealer information, provided it is not associated with any pending orders.
REQ-4: The system shall display an error message if invalid data is entered.

4. External Interface Requirements -------------------------------

This section outlines the interface requirements between the **Inventory Management System (IMS)** and external entities, including users, hardware, software, and communication systems.

4.1 User Interfaces

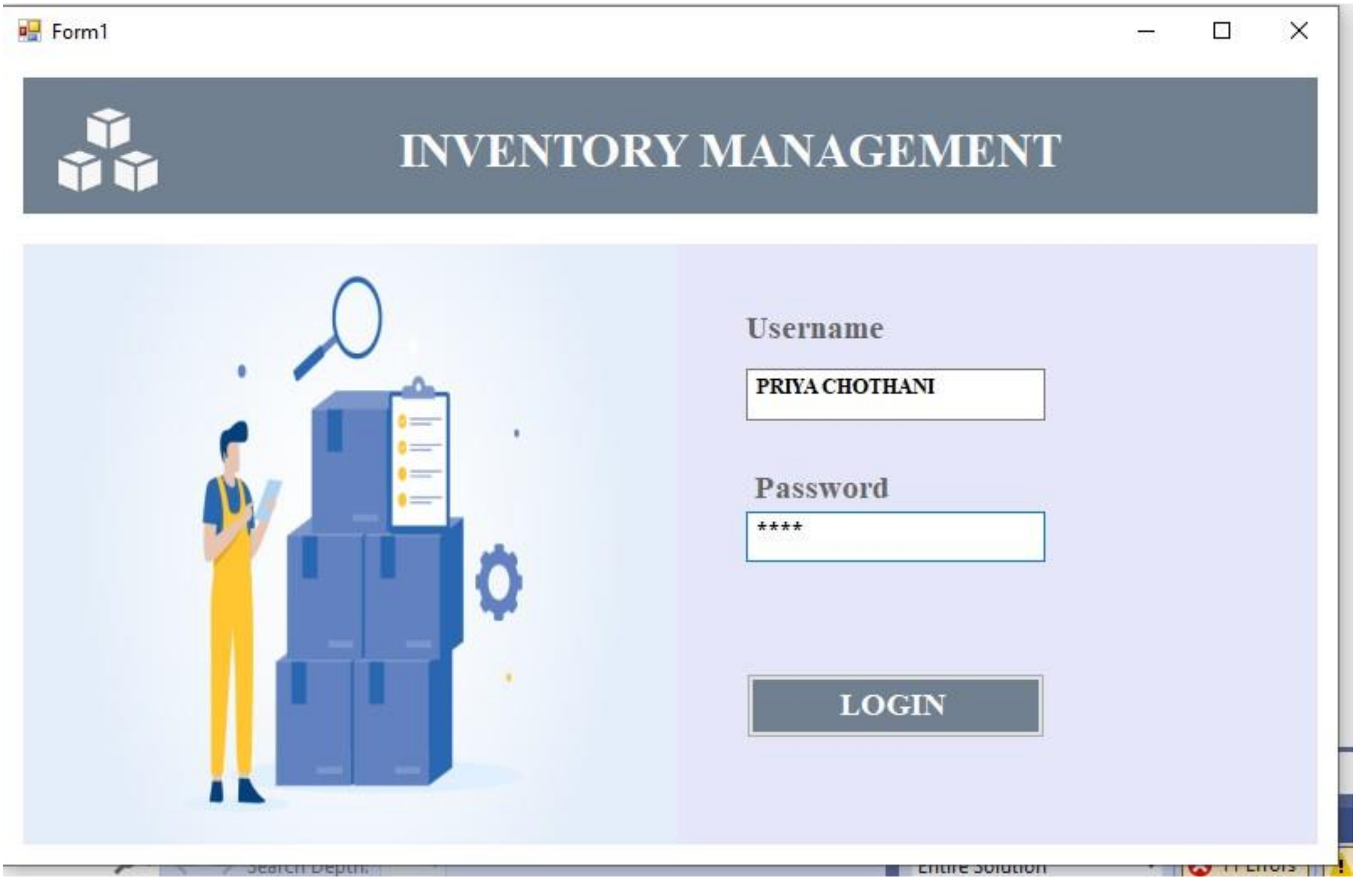
The **IMS** will have a graphical user interface (GUI) to facilitate interaction between the user and the system. The design should follow user-friendly principles, ensuring intuitive navigation and minimal training requirements for users. Key elements are described below.

- **GUI Standards and Guidelines**:

The application will follow the **Microsoft UI Style Guide** for desktop applications, ensuring consistency in design, usability, and accessibility. The UI will adhere to standard GUI conventions with consistent placement of navigation buttons, form layouts, and data presentation grids.

- **Screens**:

1) **Login Page :** The system provides a secure login interface for authorized users, including admins and managers, requiring both a username and password for access.



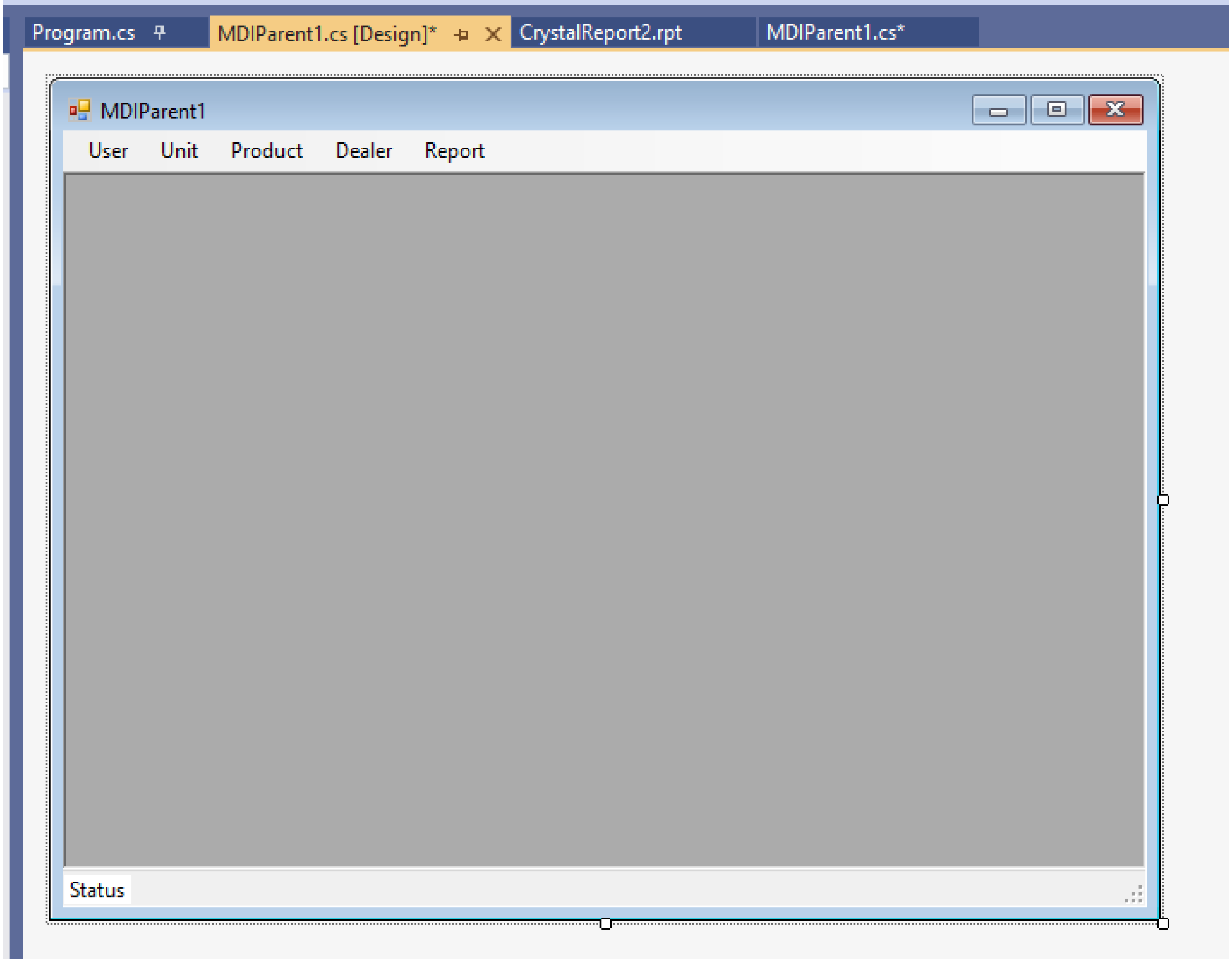
Username

PRIYA CHOTHANI

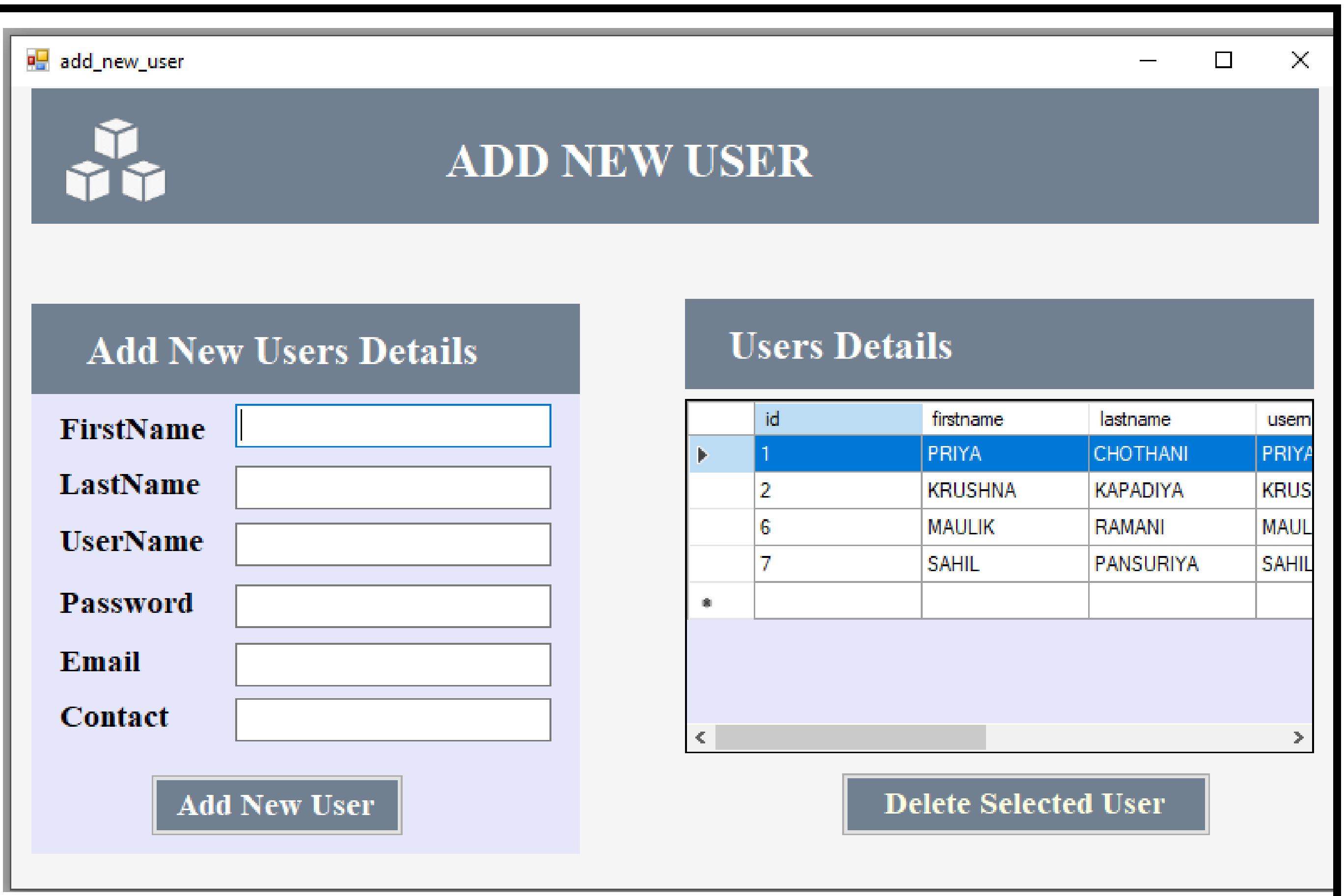
Password

****

2. **Dashboard :** Users with credentials can update information by selecting options such as User, Unit, Product, Dealer, or Report.



3. **Add New User** : Admins or managers with valid ID and password can add new users or delete selected users.



add_new_user

ADD NEW USER

Add New Users Details

FirstName

LastName

UserName

Password

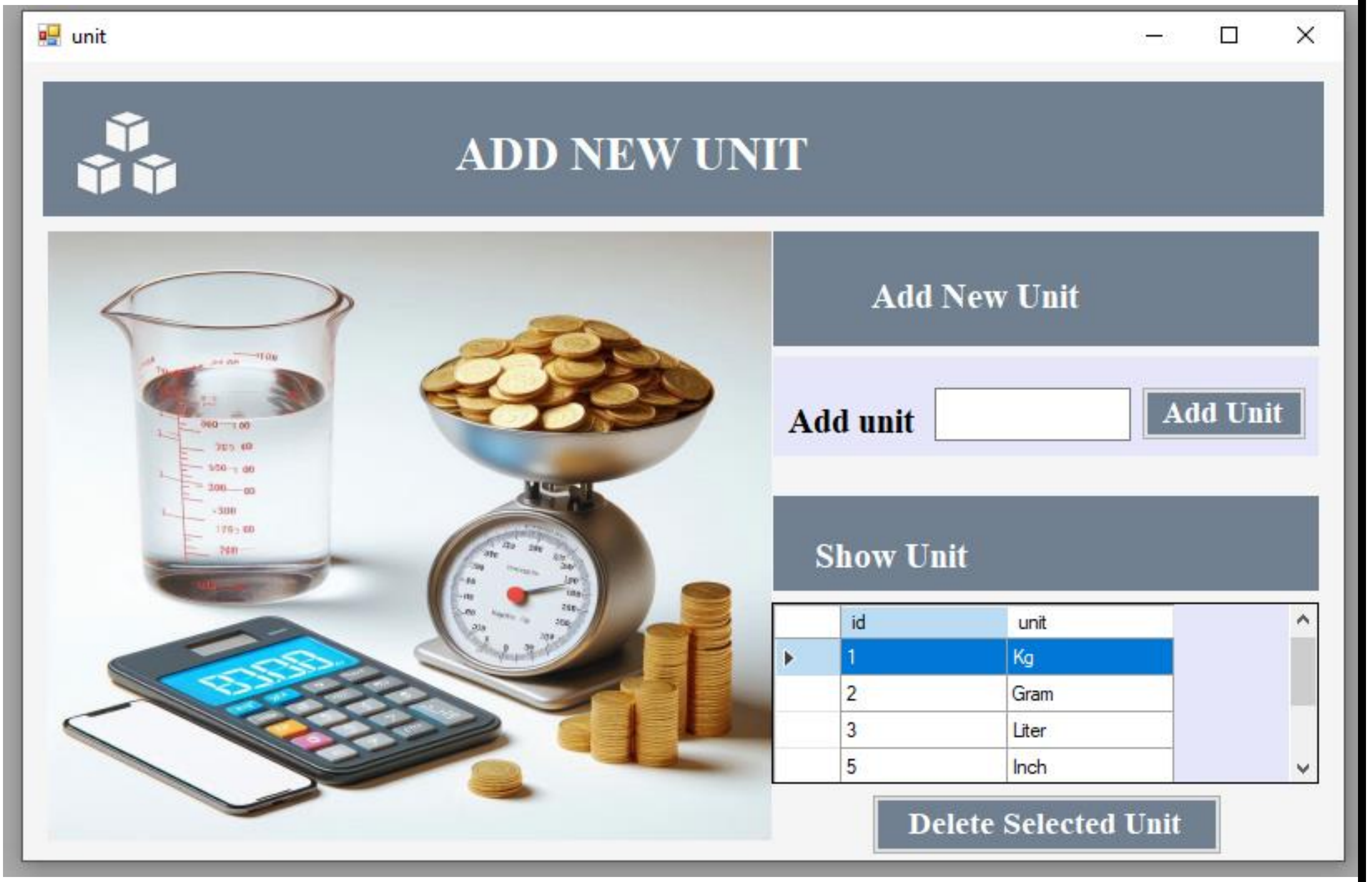
Email

Contact

Users Details

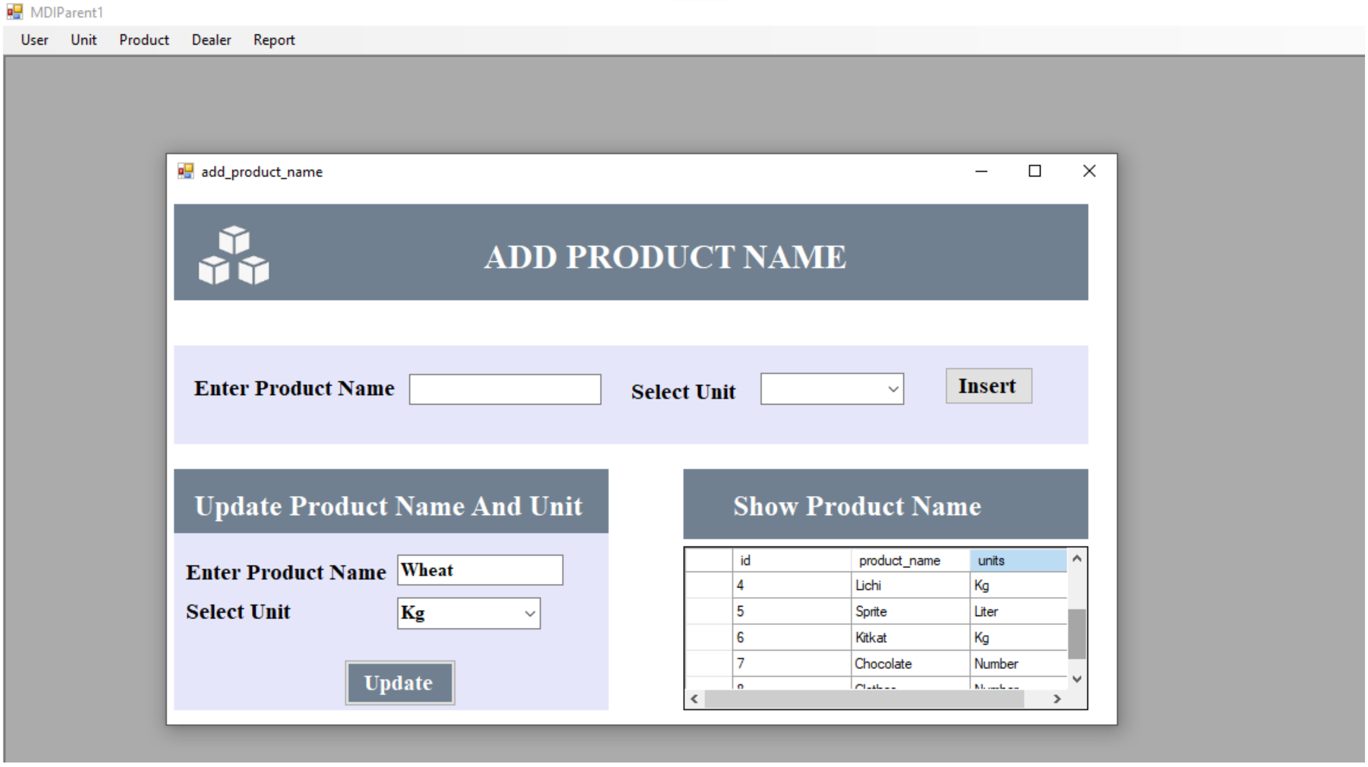
	id	firstname	lastname	usern
▶	1	PRIYA	CHOTHANI	PRIYA
	2	KRUSHNA	KAPADIYA	KRUS
	6	MAULIK	RAMANI	MAUL
	7	SAHIL	PANSURIYA	SAHIL
*				

4. **Add New Unit :** Admins or managers can delete selected units or add new units like Kg, Gm, Ltr, etc.



id	unit
1	Kg
2	Gram
3	Liter
5	Inch

5. **Add Product Name:** Admins or managers can add a new product name based on its unit or display it

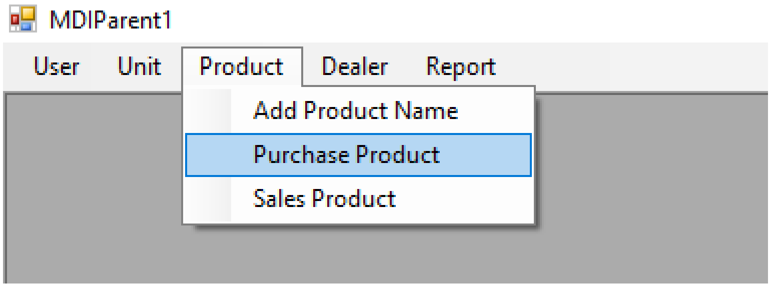


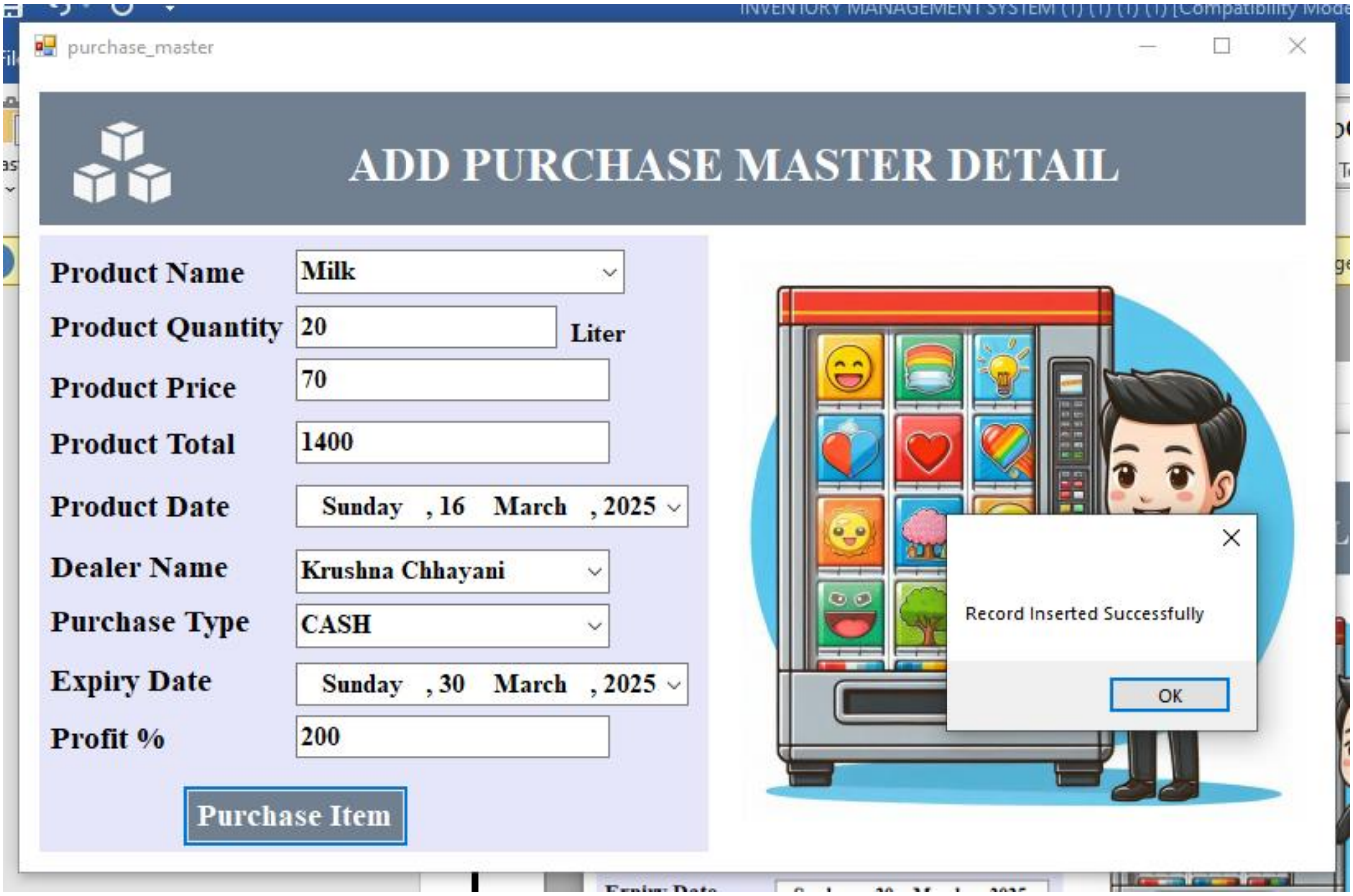
Enter Product Name: Wheat

Select Unit: Kg

id	product_name	units
4	Lichi	Kg
5	Sprite	Liter
6	Kitkat	Kg
7	Chocolate	Number
8	Clothes	Number

6. **Purchase Product:** Admins or managers can add purchased products or specify the purchase type, such as CASH or DEBIT, etc.



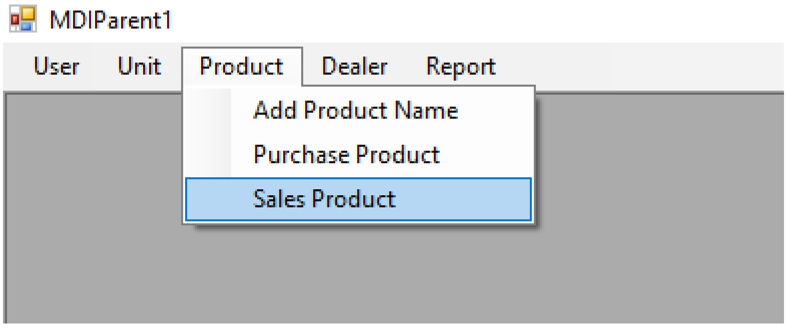


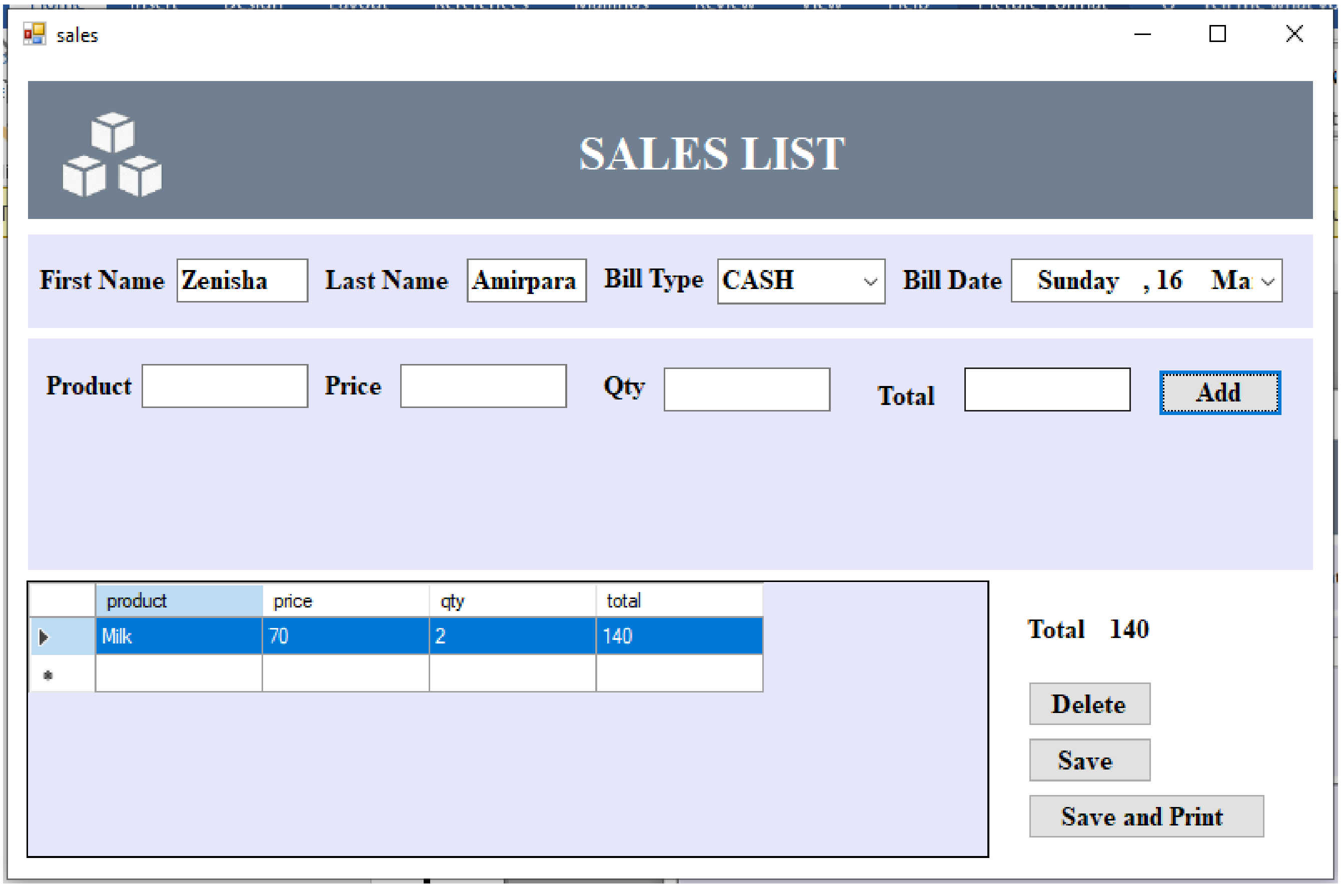
ADD PURCHASE MASTER DETAIL

Product Name	Milk	
Product Quantity	20	Liter
Product Price	70	
Product Total	1400	
Product Date	Sunday , 16 March , 2025	
Dealer Name	Krushna Chhayani	
Purchase Type	CASH	
Expiry Date	Sunday , 30 March , 2025	
Profit %	200	

Record Inserted Successfully

7. **Sales:** Admins or managers can add a sales list, save the data, or print the bill.





sales

SALES LIST

First Name: Zenisha

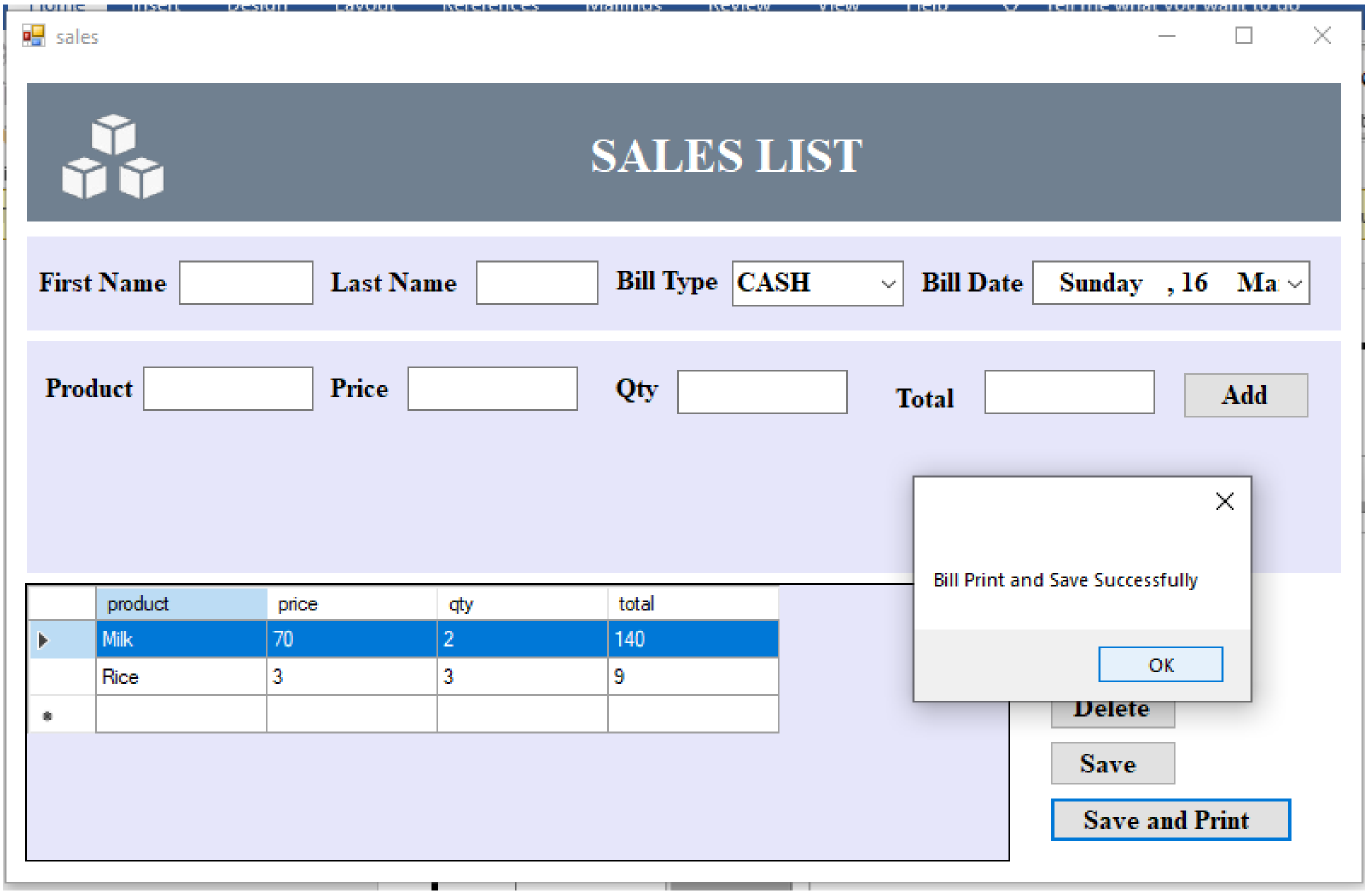
Last Name: Amirpara

Bill Type: CASH

Bill Date: Sunday , 16 Ma

	product	price	qty	total
▶	Milk	70	2	140
*				

Total 140



sales

SALES LIST

First Name:

Last Name:

Bill Type: CASH

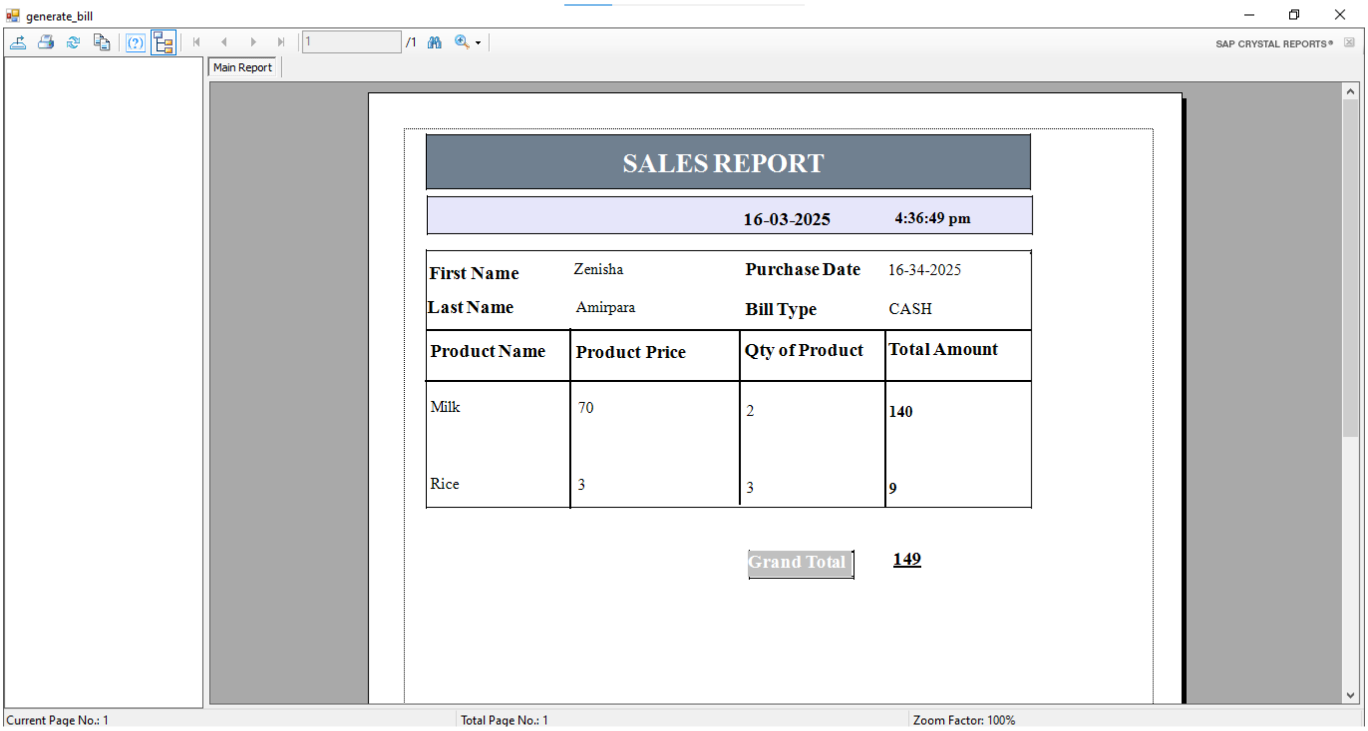
Bill Date: Sunday , 16 Ma

	product	price	qty	total
▶	Milk	70	2	140
	Rice	3	3	9
*				

Bill Print and Save Successfully

When clicking on the "Save and Print" button: The bill is saved in the system and printed for the user.

Bill



SALES REPORT

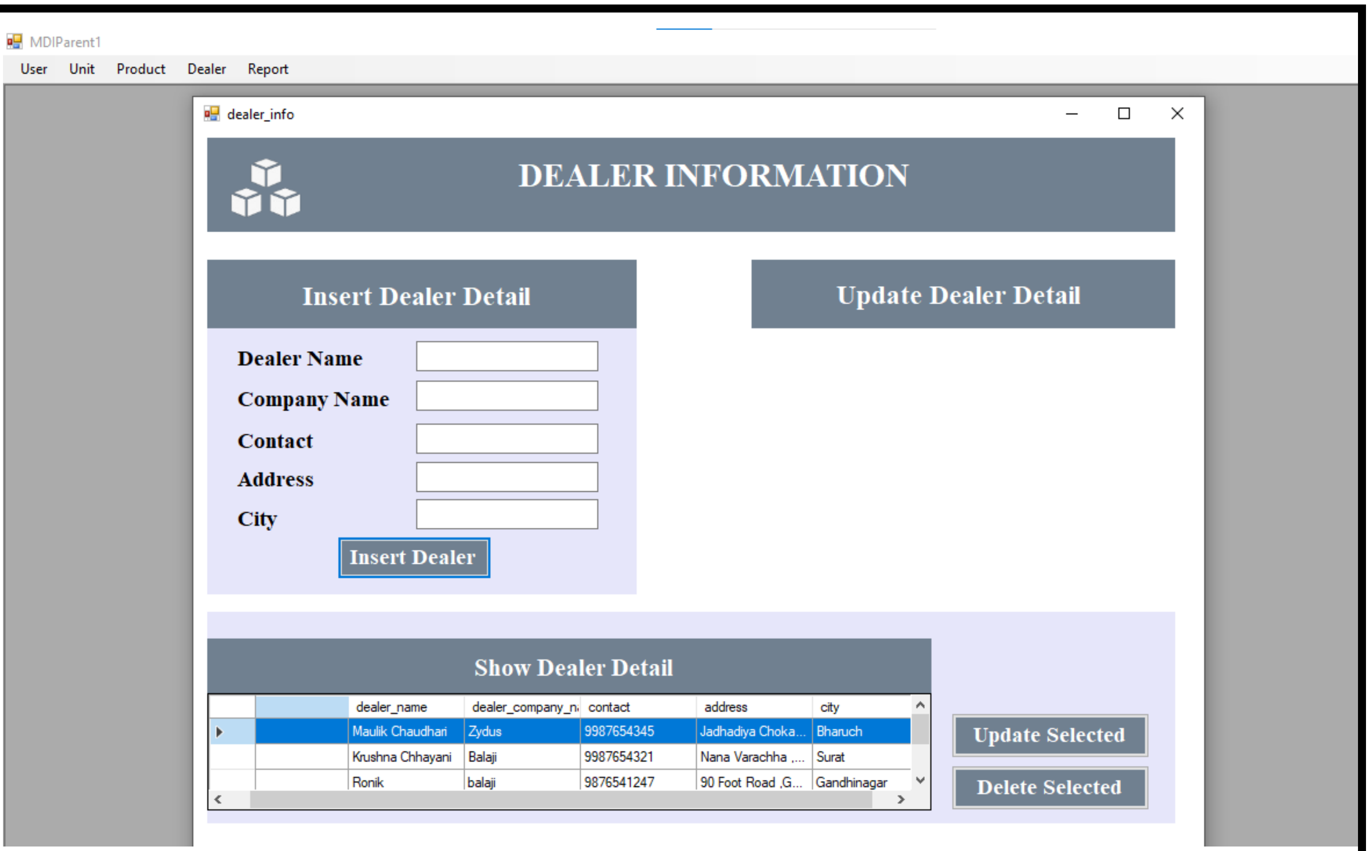
16-03-2025 4:36:49 pm

First Name	Zenisha	Purchase Date	16-34-2025
Last Name	Amirpara	Bill Type	CASH

Product Name	Product Price	Qty of Product	Total Amount
Milk	70	2	140
Rice	3	3	9

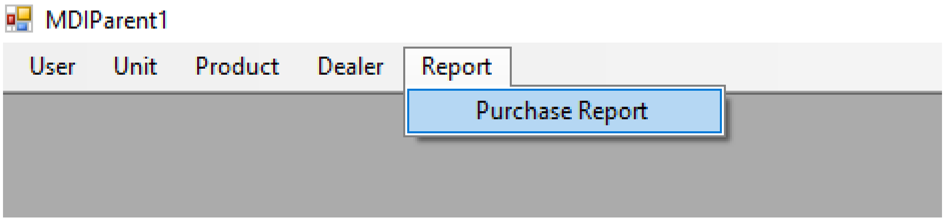
Grand Total **149**

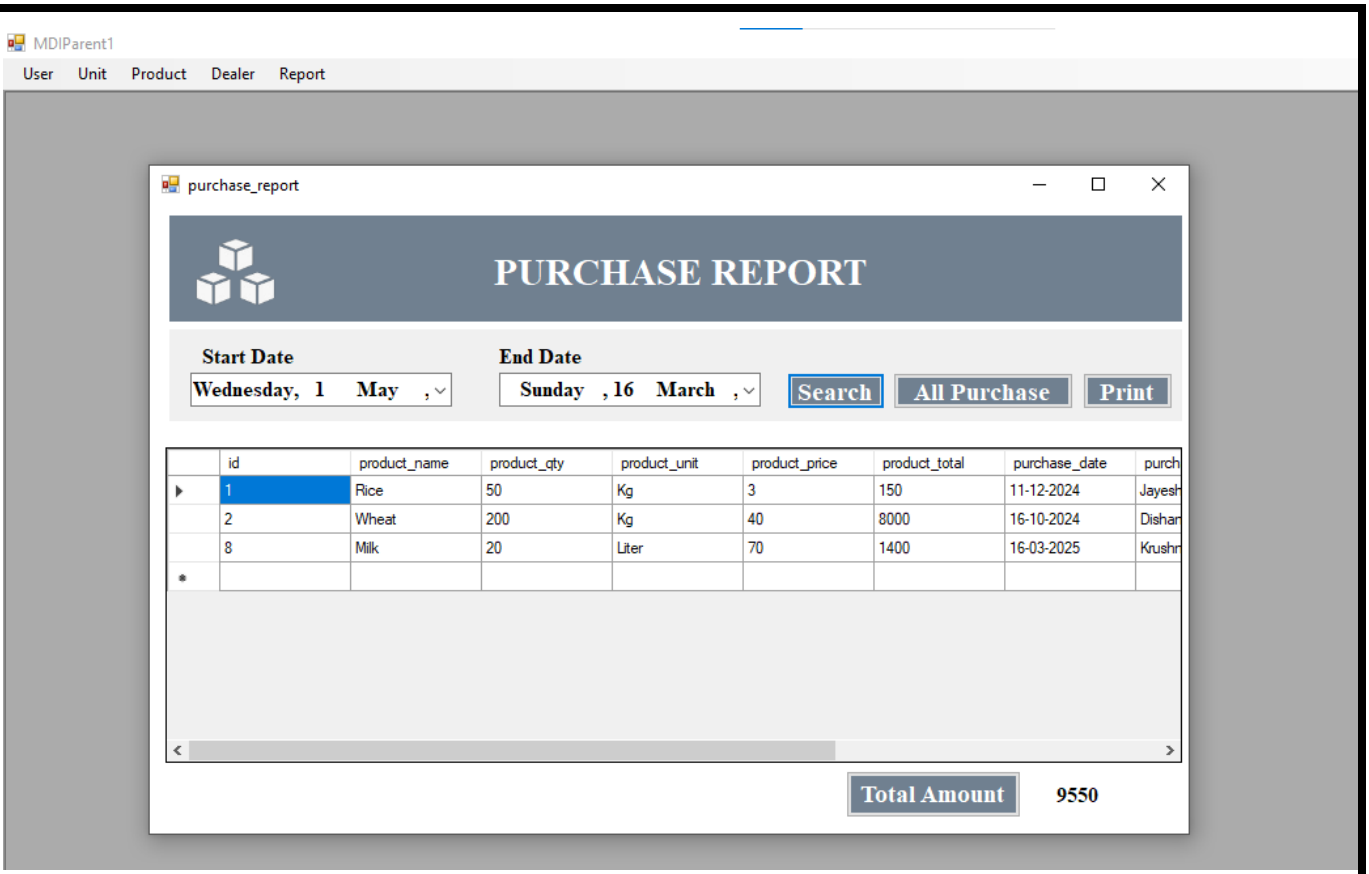
8. **Dealer Information:** Admins or managers can insert, update, or delete dealer information.



	dealer_name	dealer_company_n	contact	address	city
	Maulik Chaudhari	Zydus	9987654345	Jadhadiya Choka...	Bharuch
	Krushna Chhayani	Balaji	9987654321	Nana Varachha ,...	Surat
	Ronik	balaji	9876541247	90 Foot Road ,G...	Gandhinagar

9. **Purchase Report:** Admins or managers can select the start date and end date to filter purchase data. Clicking on **All Purchase** displays all customer purchase data, and clicking on **Print** generates and prints the bill.





PURCHASE REPORT

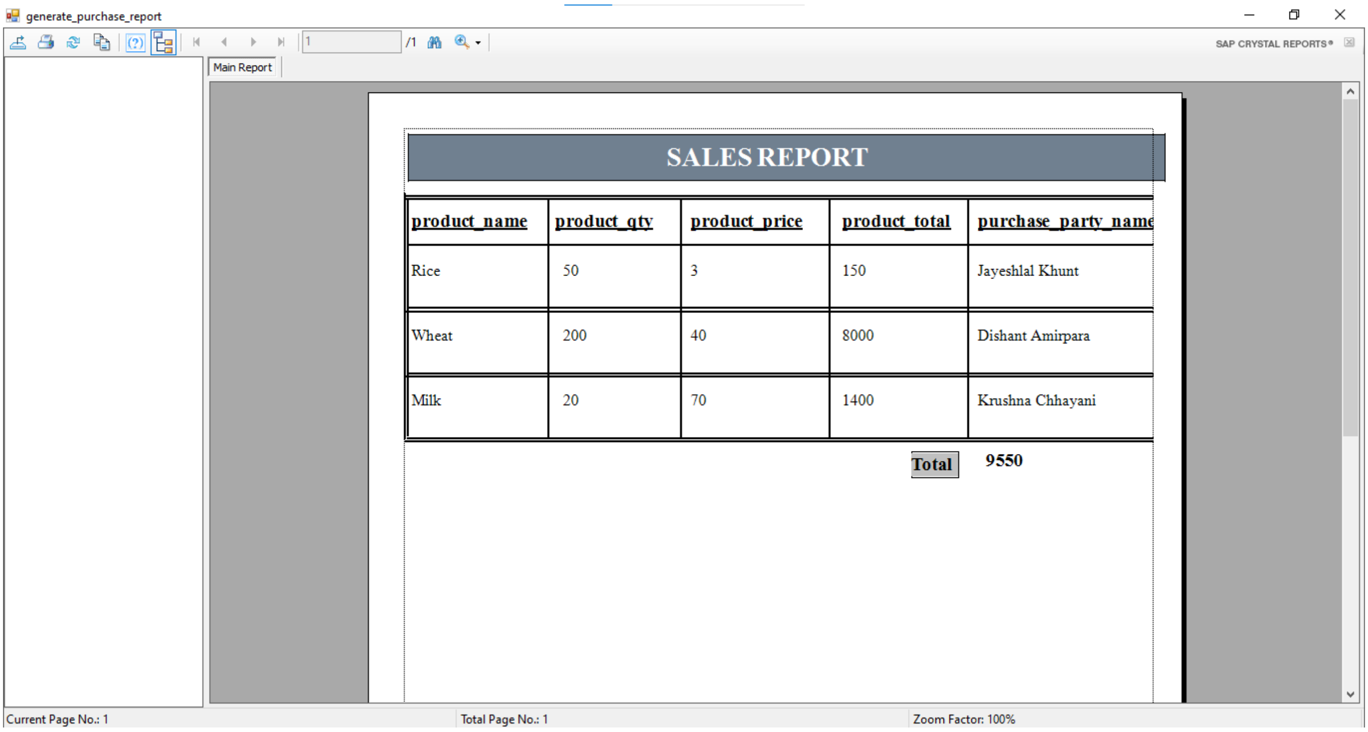
Start Date: Wednesday, 1 May ,

End Date: Sunday , 16 March ,

id	product_name	product_qty	product_unit	product_price	product_total	purchase_date	purch
1	Rice	50	Kg	3	150	11-12-2024	Jayesh
2	Wheat	200	Kg	40	8000	16-10-2024	Dishan
8	Milk	20	Liter	70	1400	16-03-2025	Krushn

Total Amount 9550

When the **Print** button is clicked, the bill is generated and printed.



SALES REPORT

product_name	product_qty	product_price	product_total	purchase_party_name
Rice	50	3	150	Jayeshlal Khunt
Wheat	200	40	8000	Dishant Amirpara
Milk	20	70	1400	Krushna Chhayani

Total 9550

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

4.2 User Interfaces

Computers:

- The system will operate on standard desktop computers with Windows OS.
- Minimum specifications: Intel i3 or above, 4 GB RAM, 500 GB HDD or SSD, monitor with 1024x768 resolution.

Bar Code Scanners:

- The system will support barcode scanners to automate the process of adding and managing products.
- Bar code scanner input will directly populate product fields when a product is scanned.

Printers:

- The system will interface with standard printers to print invoices, reports, and other documentation.
- Supported printing formats include PDF and standard paper sizes (A4, Letter).

4.3 Software Interfaces

Operating System:

- The system will be built to run on Windows 10 and later versions.

Database:

- The system will connect to a SQL Server database for persistent storage of product, stock, order, and dealer information.
- The database schema will include tables such as Products, Orders, Dealers, Stock, and Transactions.
- Data will be accessed via ADO.NET or Entity Framework, ensuring secure and efficient database operations.

Libraries:

- The application will use the System.Windows.Forms library for user interface elements.
- The System.Drawing library will be utilized for generating graphical reports and invoices.
- The iTextSharp library will be used for generating PDFs (e.g., invoices, reports).

Data Interchange:

- Data such as invoices and reports will be exported in formats like CSV and PDF.
- The system will also allow data imports from CSV files to bulk update products or stock.

Integration with External Software:

- The system may integrate with external accounting software or ERP systems for synchronizing order and transaction data.

- APIs for data sharing between IMS and external systems will be REST-based.

4.4 Communications Interfaces

The IMS will support communication between different system components and external systems:

Network Protocols:

- HTTP/HTTPS will be used for secure web-based communication if the system is integrated with online services (e.g., supplier APIs for real-time inventory updates).

Message Formats:

- If integrated with an external system, the communication will utilize JSON format for data exchange.
- REST APIs will follow common standards for request and response (GET, POST, PUT, DELETE).

Security and Encryption:

- All communications between the system and external systems will be secured using TLS/SSL encryption.
- Sensitive data, such as transaction details and customer information, will be encrypted using industry-standard AES-256 encryption.

Email Notifications:

- The system will send email notifications for critical events such as low stock alerts, order confirmations, and invoices.
- Emails will be sent using SMTP protocol with support for custom email templates and attachments.

5. Other Nonfunctional Requirements-----------------------------

5.1 Performance Requirements

The Inventory Management System (IMS) should provide optimal performance across various usage scenarios. The following performance requirements apply:

System Response Time:

- The system must respond to user actions (e.g., navigating between screens, saving or updating data) within 2 seconds under normal operating conditions.
- For more complex operations such as generating reports or processing large datasets, the system should complete tasks within 5 seconds.

Scalability:

- The system should handle up to 10,000 products and 5,000 orders without significant degradation in performance.
- The system must be able to manage 100 concurrent users accessing the system without impacting performance.

Data Processing Speed:

- Stock updates and order processing should reflect changes in the database in real-time, with no more than a 1-second delay for inventory adjustments.
- Importing bulk product or stock data (up to 5,000 records) via CSV should complete within 30 seconds.

Reporting:

- Generating standard reports (e.g., monthly sales, product inventory) should take no more than 10 seconds.

5.2 Safety Requirements

Data Backup:

- The system must provide automatic daily backups of all critical data, including product details, stock levels, orders, and dealer information.
- Manual backup options must also be available for users, with warnings displayed to confirm the backup process.

Error Handling:

- Any system errors or unexpected failures should be logged and reported to the system admin immediately.
- In the event of a power failure or sudden shutdown, the system must auto-save ongoing operations to prevent data loss.

Transaction Integrity:

- The system must ensure the atomicity of transactions (all operations within a transaction must either complete successfully or fail as a whole). For instance, when processing an order, all stock updates and invoicing should occur as a single, indivisible transaction.

Compliance:

- The system must comply with any relevant safety standards for data handling and processing in a retail or wholesale inventory environment.

5.3 Security Requirements

User Authentication:

- All users must authenticate with username and password credentials.
- User passwords must be encrypted using a minimum of SHA-256 encryption.
- A multi-factor authentication (MFA) option should be available for administrative users and managers to enhance security.

Access Control:

- The system must implement role-based access control (RBAC). Different roles (e.g., Admin, Stock Manager, Dealer) should have appropriate permissions. For example:
  - Admin: Full access to all system features.
  - Stock Manager: Access to stock and order management but not to system settings or user management.
  - Dealer: Access to view product lists and place orders, but no access to stock management or other administrative tasks.

Data Encryption:

- All sensitive data, such as financial transactions, order details, and customer/dealer information, must be encrypted both at rest and in transit using AES-256 encryption.

Audit Trails:

- The system must log all user actions and data modifications (e.g., product updates, stock level changes, orders) to provide an audit trail for security monitoring and issue resolution.

Session Management:

- User sessions should expire after 30 minutes of inactivity, and users should be required to log back in.

Data Privacy:

- The system must comply with any relevant data privacy laws (e.g., GDPR), ensuring that users' personal data and business-sensitive information are protected.

5.4 Software Quality Attributes

Reliability:

- The system must achieve **99.9% uptime**, ensuring it is available for users at all times during business hours.
- The system must handle any failures or crashes gracefully, with the ability to recover from critical errors without data loss.

Usability:

- The system’s interface must be intuitive and easy to use, minimizing the need for extensive user training.
- The application must include **tooltips** and **help documentation** accessible from the user interface, ensuring users can quickly resolve common issues.

Maintainability:

- The system codebase must be modular and well-documented, allowing future developers to easily understand and modify the system.
- **Error logs** and **performance monitoring** tools should be integrated to facilitate debugging and maintenance.

Interoperability:

- The system should support integration with external software like accounting tools or ERP systems through **standardized APIs**.

Portability:

- The system should be designed to run on multiple versions of Windows (Windows 10 and later) without modification.
- The system should be **easily deployable** on other Windows machines by non-technical users.

Scalability:

- As business grows, the system should be able to scale horizontally (adding more users, more products) without needing significant architectural changes.

Testability:

- The system should have built-in **unit tests** and **integration tests** to verify functionality.
- Every major system function should be **testable** in isolation, with the ability to simulate data inputs for testing purposes.

6. Other Requirements-----------------------------------

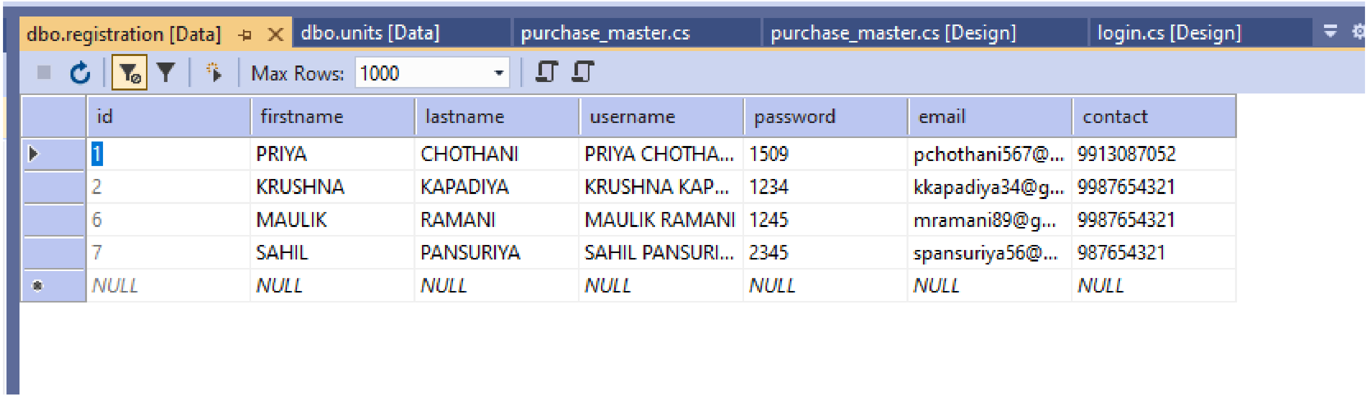
This section outlines any additional requirements not explicitly covered elsewhere in the Software Requirements Specification (SRS). These requirements are essential for the completeness and proper functioning of the Inventory Management System.

Database Requirements:

The system shall use Microsoft SQL Server as the backend database. The database should be capable of storing inventory-related data, such as product details, stock levels, suppliers, and transaction records. It must ensure ACID (Atomicity, Consistency, Isolation, Durability) compliance and support concurrent access by multiple users.

The database design will include tables such as:

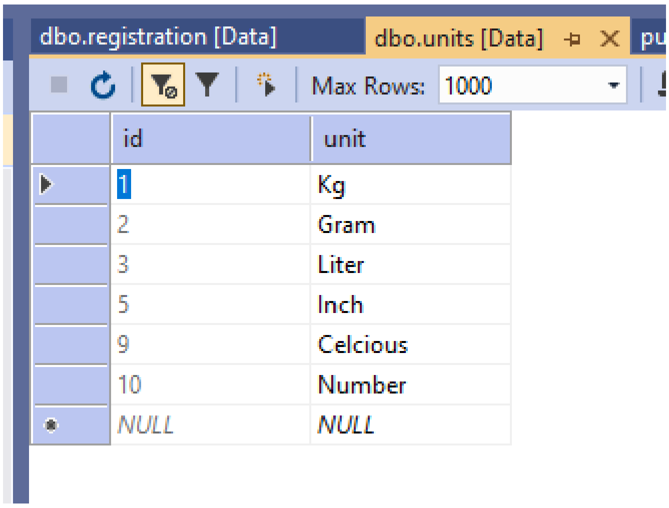
1) Registration



Max Rows: 1000

	id	firstname	lastname	username	password	email	contact
▶	1	PRIYA	CHOTHANI	PRIYA CHOTHA...	1509	pchothani567@...	9913087052
	2	KRUSHNA	KAPADIYA	KRUSHNA KAP...	1234	kkapadiya34@g...	9987654321
	6	MAULIK	RAMANI	MAULIK RAMANI	1245	mramani89@g...	9987654321
	7	SAHIL	PANSURIYA	SAHIL PANSURI...	2345	spansuriya56@...	987654321
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

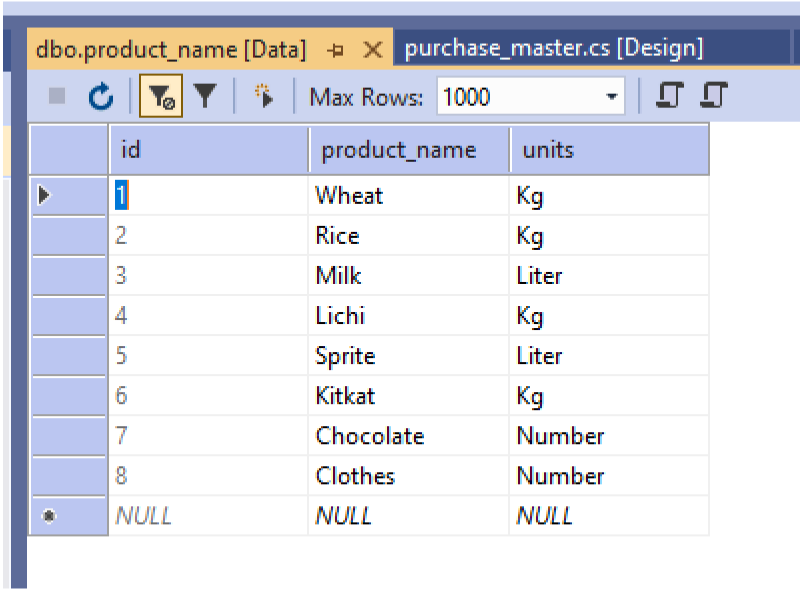
2) Unit



Max Rows: 1000

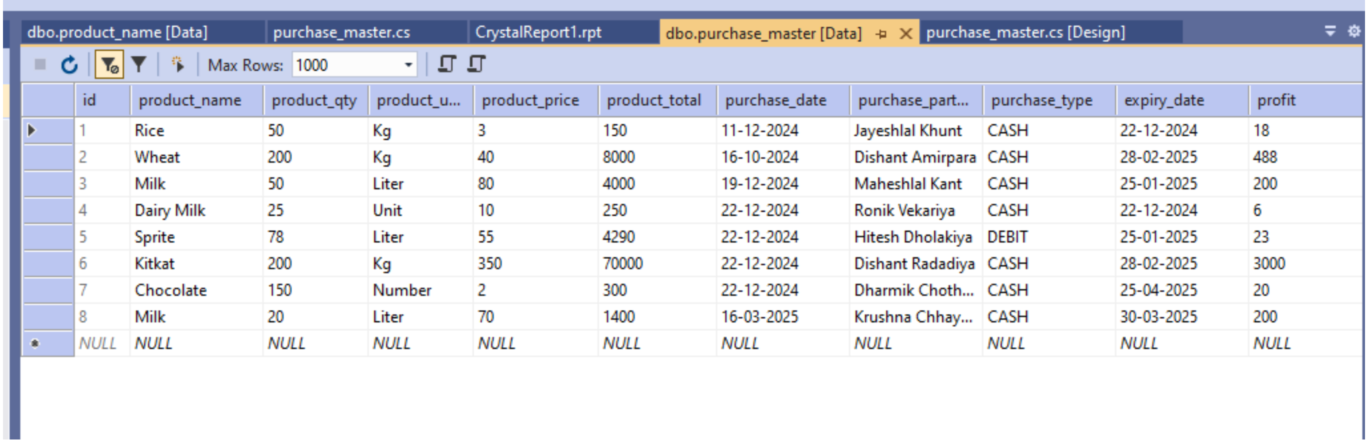
	id	unit
▶	1	Kg
	2	Gram
	3	Liter
	5	Inch
	9	Celcious
	10	Number
*	NULL	NULL

3) Product_name



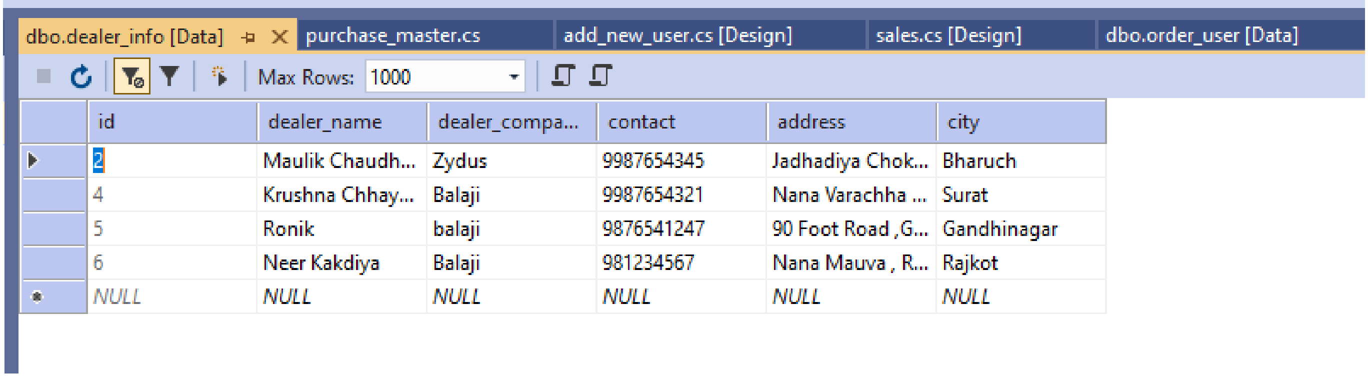
id	product_name	units
1	Wheat	Kg
2	Rice	Kg
3	Milk	Liter
4	Lichi	Kg
5	Sprite	Liter
6	Kitkat	Kg
7	Chocolate	Number
8	Clothes	Number
NULL	NULL	NULL

4) Purchase_Master



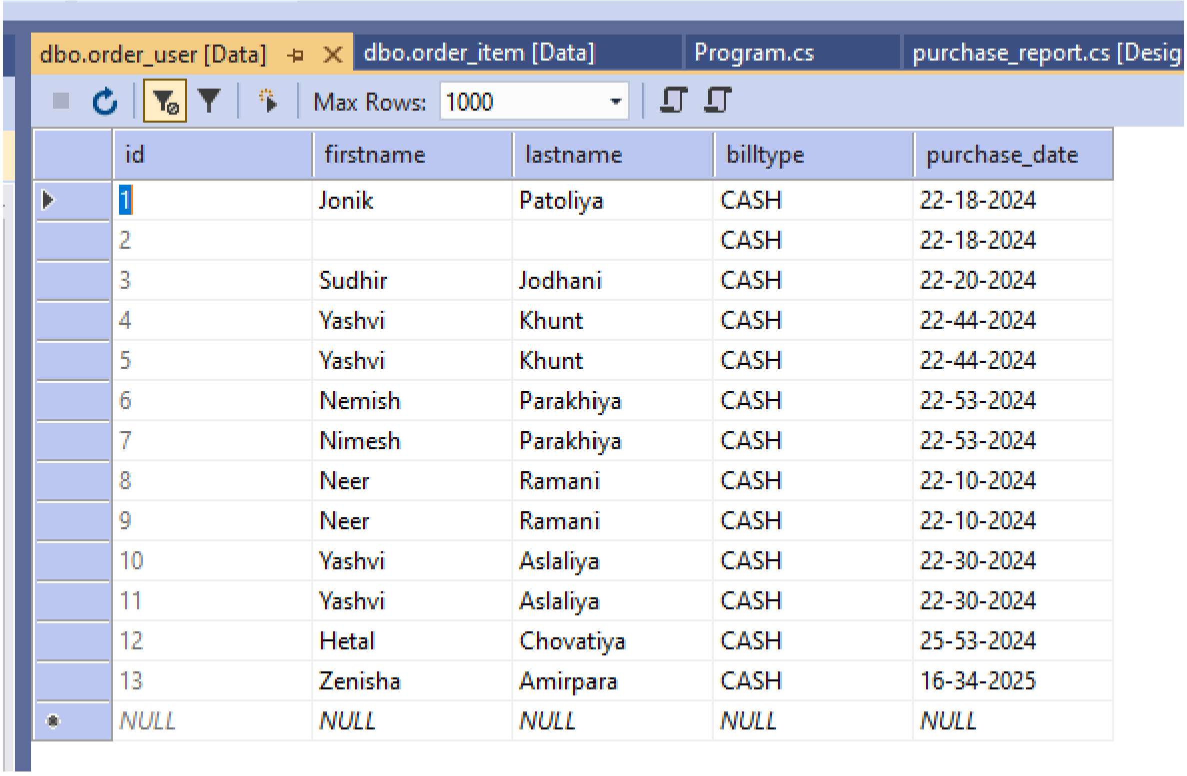
id	product_name	product_qty	product_u...	product_price	product_total	purchase_date	purchase_part...	purchase_type	expiry_date	profit
1	Rice	50	Kg	3	150	11-12-2024	Jayeshlal Khunt	CASH	22-12-2024	18
2	Wheat	200	Kg	40	8000	16-10-2024	Dishant Amirpara	CASH	28-02-2025	488
3	Milk	50	Liter	80	4000	19-12-2024	Maheshlal Kant	CASH	25-01-2025	200
4	Dairy Milk	25	Unit	10	250	22-12-2024	Ronik Vekariya	CASH	22-12-2024	6
5	Sprite	78	Liter	55	4290	22-12-2024	Hitesh Dholakiya	DEBIT	25-01-2025	23
6	Kitkat	200	Kg	350	70000	22-12-2024	Dishant Radadiya	CASH	28-02-2025	3000
7	Chocolate	150	Number	2	300	22-12-2024	Dharmik Choth...	CASH	25-04-2025	20
8	Milk	20	Liter	70	1400	16-03-2025	Krushna Chhay...	CASH	30-03-2025	200
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5) Dealer_Info



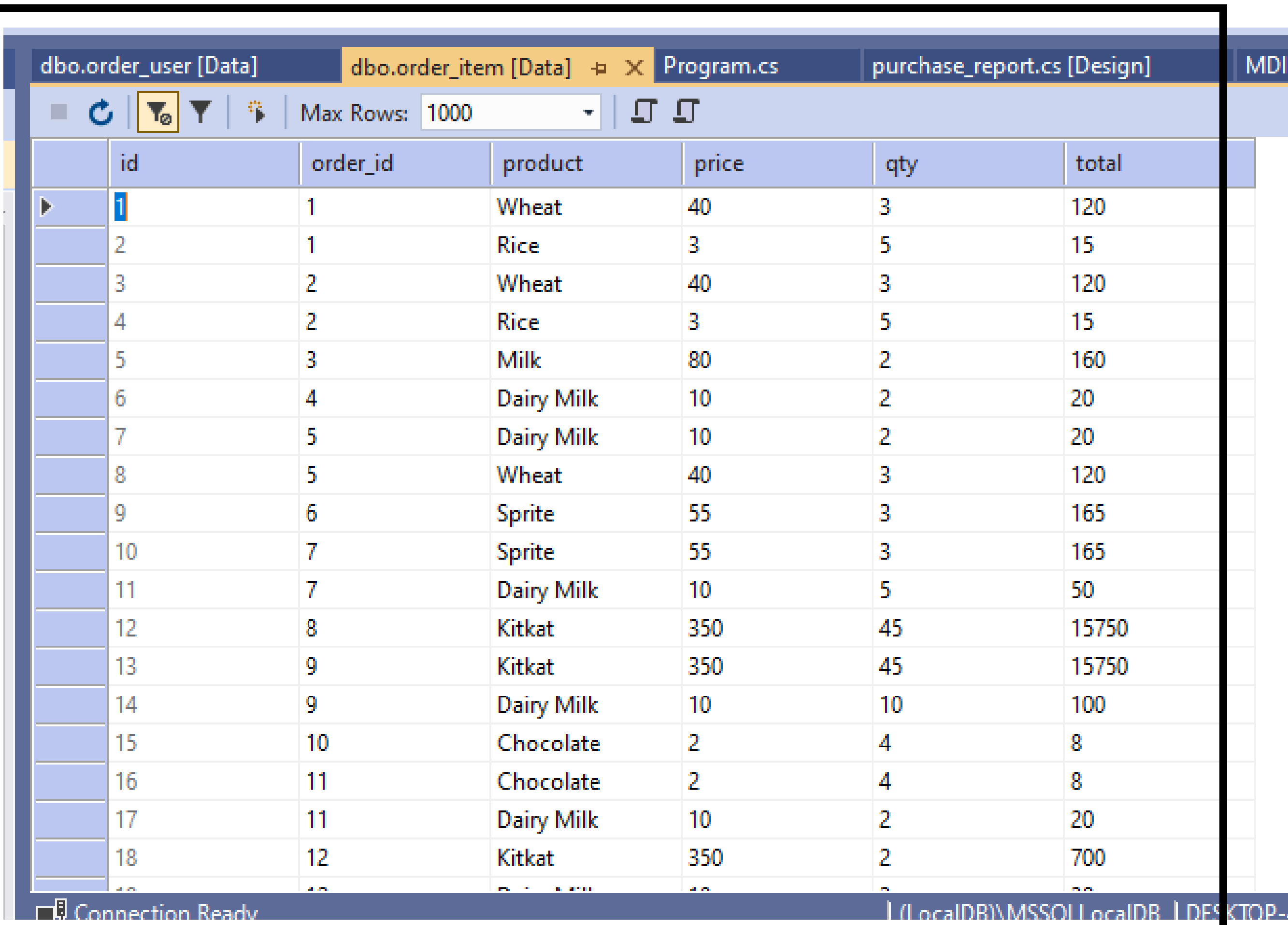
id	dealer_name	dealer_compa...	contact	address	city
2	Maulik Chaudh...	Zydus	9987654345	Jadhadiya Chok...	Bharuch
4	Krushna Chhay...	Balaji	9987654321	Nana Varachha ...	Surat
5	Ronik	balaji	9876541247	90 Foot Road ,G...	Gandhinagar
6	Neer Kakdiya	Balaji	981234567	Nana Mauva , R...	Rajkot
NULL	NULL	NULL	NULL	NULL	NULL

6) Order_user



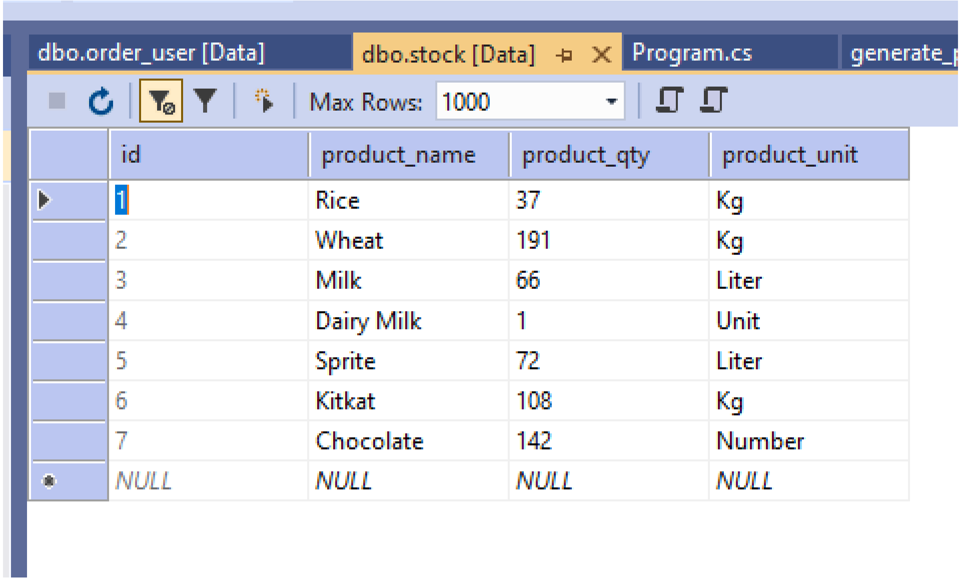
id	firstname	lastname	billtype	purchase_date
1	Jonik	Patoliya	CASH	22-18-2024
2			CASH	22-18-2024
3	Sudhir	Jodhani	CASH	22-20-2024
4	Yashvi	Khunt	CASH	22-44-2024
5	Yashvi	Khunt	CASH	22-44-2024
6	Nemish	Parakhiya	CASH	22-53-2024
7	Nimesh	Parakhiya	CASH	22-53-2024
8	Neer	Ramani	CASH	22-10-2024
9	Neer	Ramani	CASH	22-10-2024
10	Yashvi	Aslaliya	CASH	22-30-2024
11	Yashvi	Aslaliya	CASH	22-30-2024
12	Hetal	Chovatiya	CASH	25-53-2024
13	Zenisha	Amirpara	CASH	16-34-2025
NULL	NULL	NULL	NULL	NULL

7) Order_Item



id	order_id	product	price	qty	total
1	1	Wheat	40	3	120
2	1	Rice	3	5	15
3	2	Wheat	40	3	120
4	2	Rice	3	5	15
5	3	Milk	80	2	160
6	4	Dairy Milk	10	2	20
7	5	Dairy Milk	10	2	20
8	5	Wheat	40	3	120
9	6	Sprite	55	3	165
10	7	Sprite	55	3	165
11	7	Dairy Milk	10	5	50
12	8	Kitkat	350	45	15750
13	9	Kitkat	350	45	15750
14	9	Dairy Milk	10	10	100
15	10	Chocolate	2	4	8
16	11	Chocolate	2	4	8
17	11	Dairy Milk	10	2	20
18	12	Kitkat	350	2	700

8) Stock



id	product_name	product_qty	product_unit
1	Rice	37	Kg
2	Wheat	191	Kg
3	Milk	66	Liter
4	Dairy Milk	1	Unit
5	Sprite	72	Liter
6	Kitkat	108	Kg
7	Chocolate	142	Number
NULL	NULL	NULL	NULL

Internationalization Requirements:
The system should support localization for different regions and languages. Although the initial version will be in English, the system must be designed to handle multiple languages in the future (for example, by using resource files for UI

text). Currency formats must be adaptable based on the user's locale.

Legal Requirements:
The system must comply with data privacy laws, such as GDPR (General Data Protection Regulation) if deployed in the EU, and similar regulations in other regions. Proper measures should be taken to encrypt sensitive data like user credentials and transaction details.

Reuse Objectives:
The system will be built with reusable components for scalability and ease of maintenance. Modules such as Authentication, Reporting, and Stock Management will be designed as separate, loosely coupled components that can be reused across

different projects or systems in the future.

Appendix A: Glossary

This section defines all the key terms and acronyms necessary to interpret the SRS properly for the Inventory Management System.

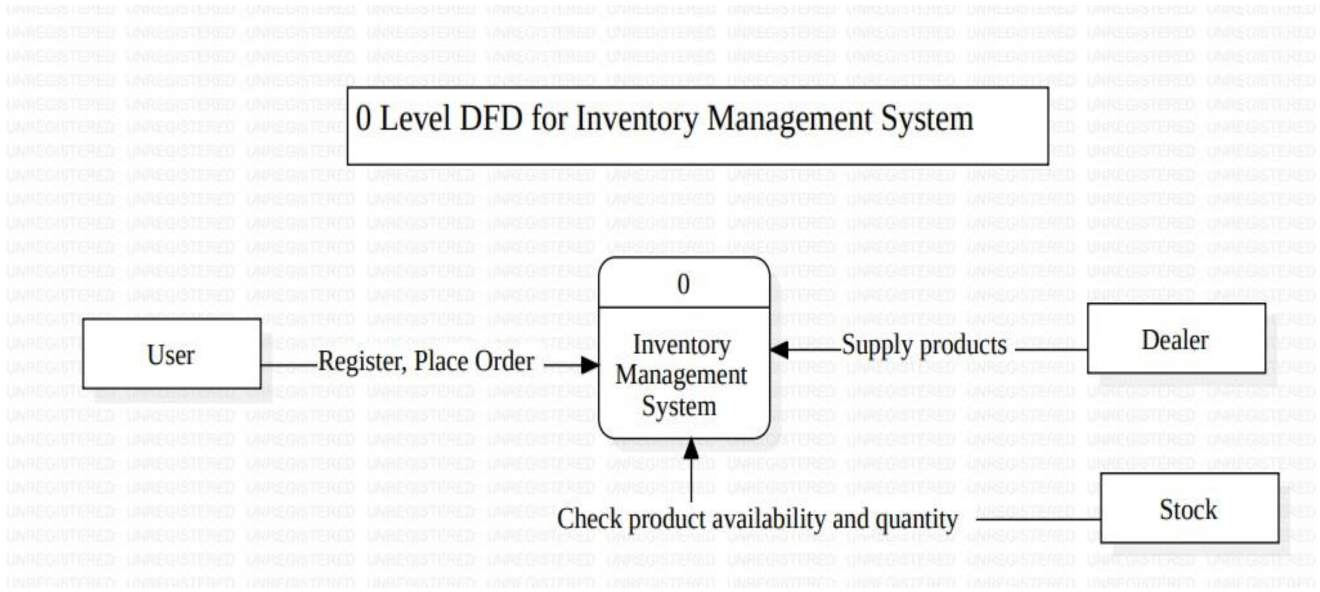
- **API:** Application Programming Interface – A set of functions that allow the system to interact with external applications.
- **CRUD:** Create, Read, Update, Delete – Operations that are typically performed on data in a database.
- **ERP:** Enterprise Resource Planning – A type of software used by organizations to manage day-to-day business activities such as accounting, procurement, and supply chain operations.
- **GUI:** Graphical User Interface – The interface through which the user interacts with the system.
- **SQL:** Structured Query Language – A programming language used to manage and manipulate relational databases.
- **Stock Keeping Unit (SKU):** A unique identifier for each distinct product and service that can be purchased.

Appendix B: Analysis Models

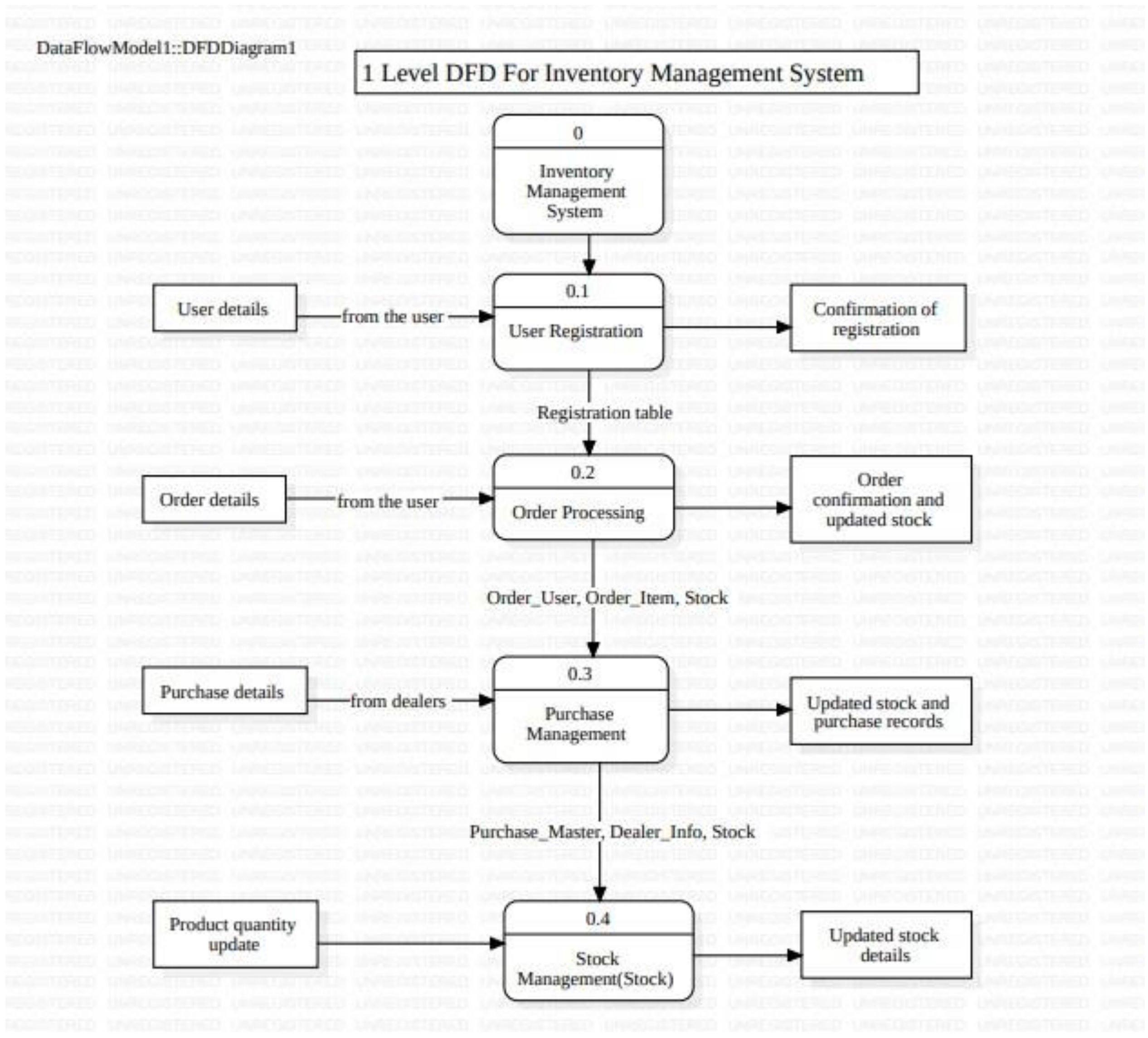
This section contains relevant analysis models that provide a visual representation of the system.

1. **Data Flow Diagram (DFD):**

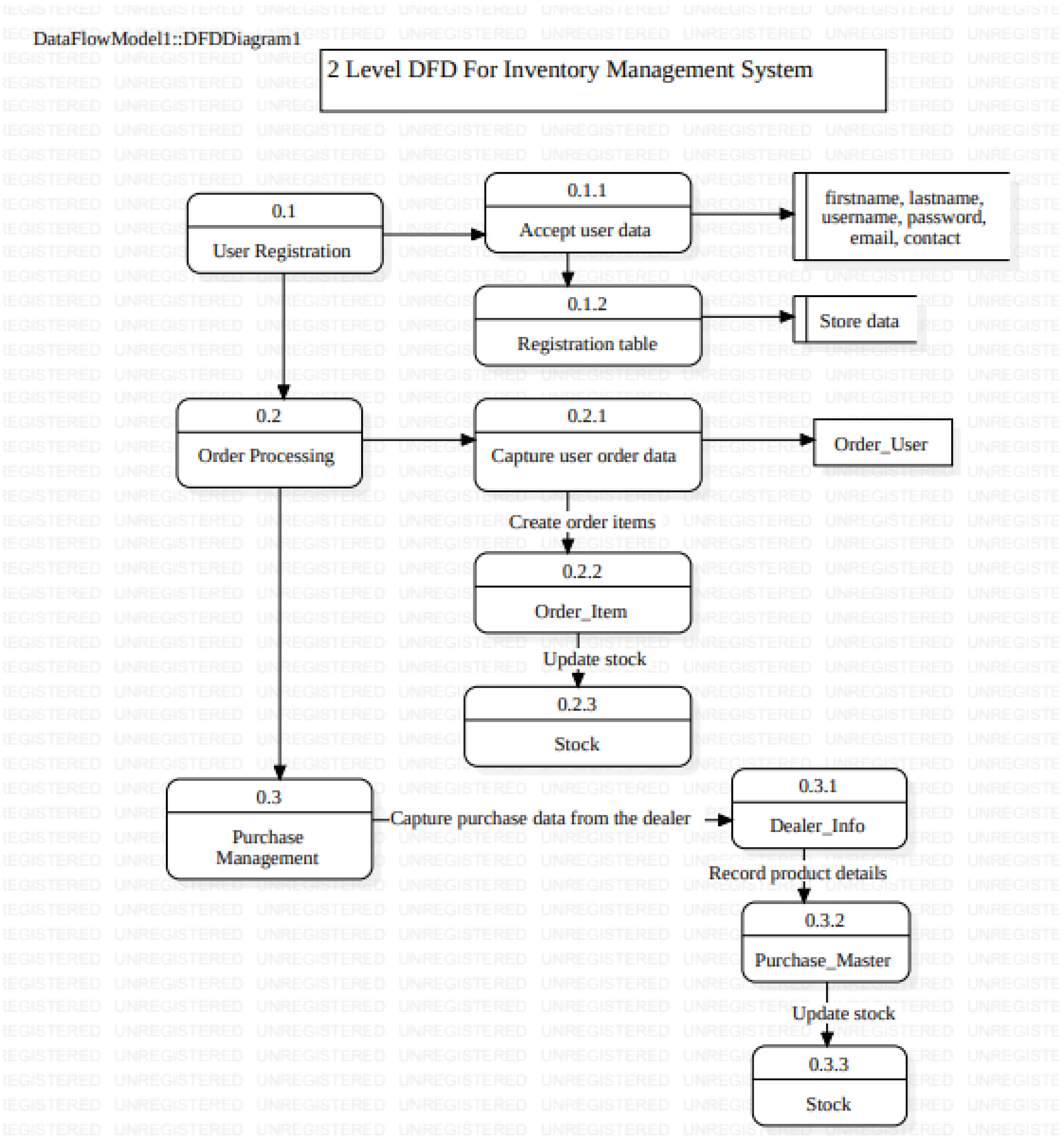
    - **Level 0:** The Inventory Management System interacts with Users, Suppliers, and Administrators. It processes input data such as product information and stock levels and generates output in the form of reports and alerts.


0 Level DFD for Inventory Management System
0
Inventory Management System
User
Register, Place Order
Dealer
Supply products
Stock
Check product availability and quantity

- **Level 1:** Details of specific processes, such as Add Product, Update Stock, Generate Report, and Handle Supplier Orders.


DataFlowModel1::DFDDiagram1
1 Level DFD For Inventory Management System
0
Inventory Management System
User details
from the user
0.1
User Registration
Confirmation of registration
Registration table
Order details
from the user
0.2
Order Processing
Order confirmation and updated stock
Order_User, Order_Item, Stock
Purchase details
from dealers
0.3
Purchase Management
Updated stock and purchase records
Purchase_Master, Dealer_Info, Stock
Product quantity update
0.4
Stock Management(Stock)
Updated stock details

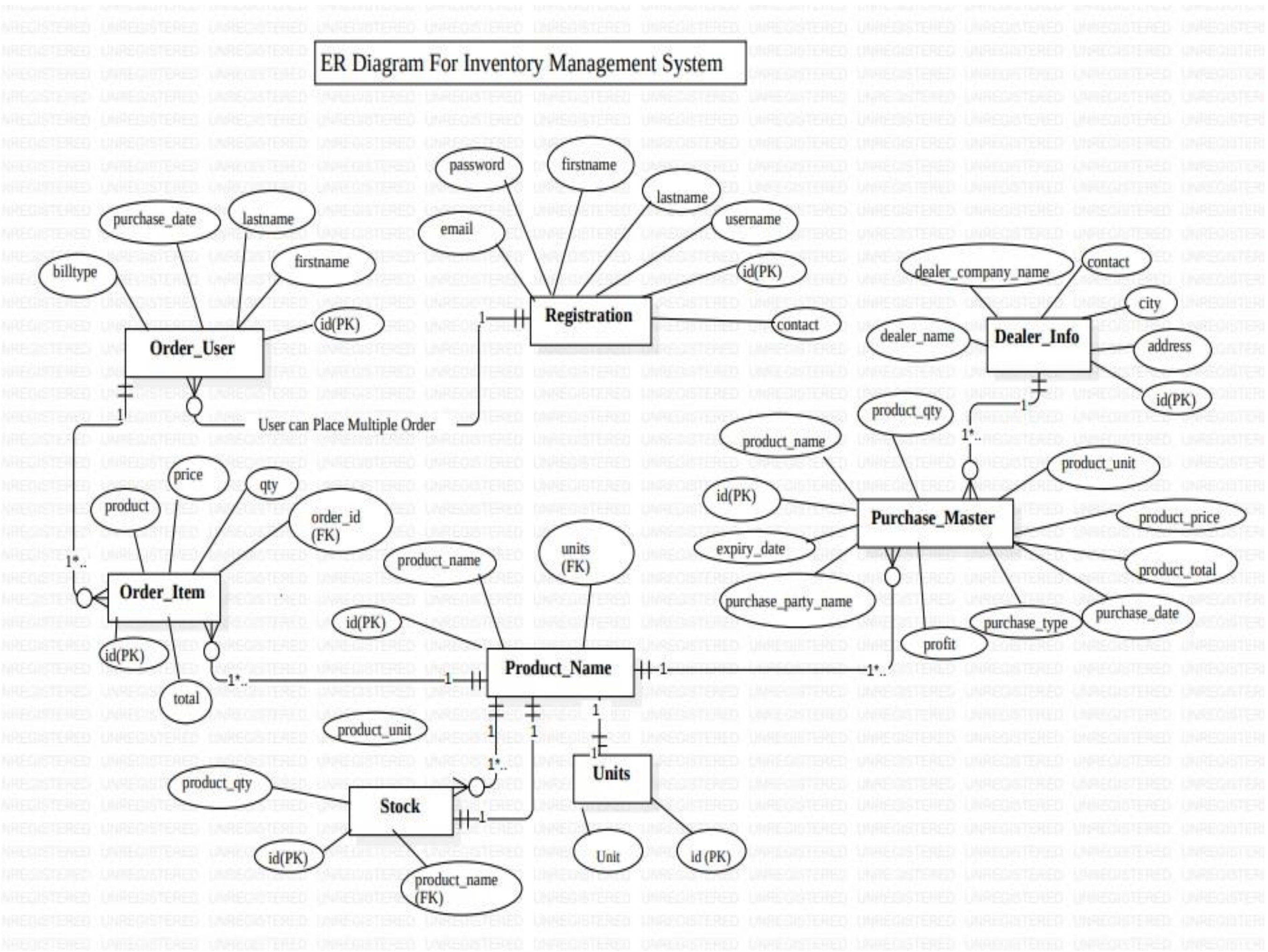
Level 2: The 2-level DFD for the Inventory Management System shows the flow of data through processes like User Registration, Order Processing, and Purchase Management, with updates to stock and records.



2. Entity-Relationship Diagram (ERD):

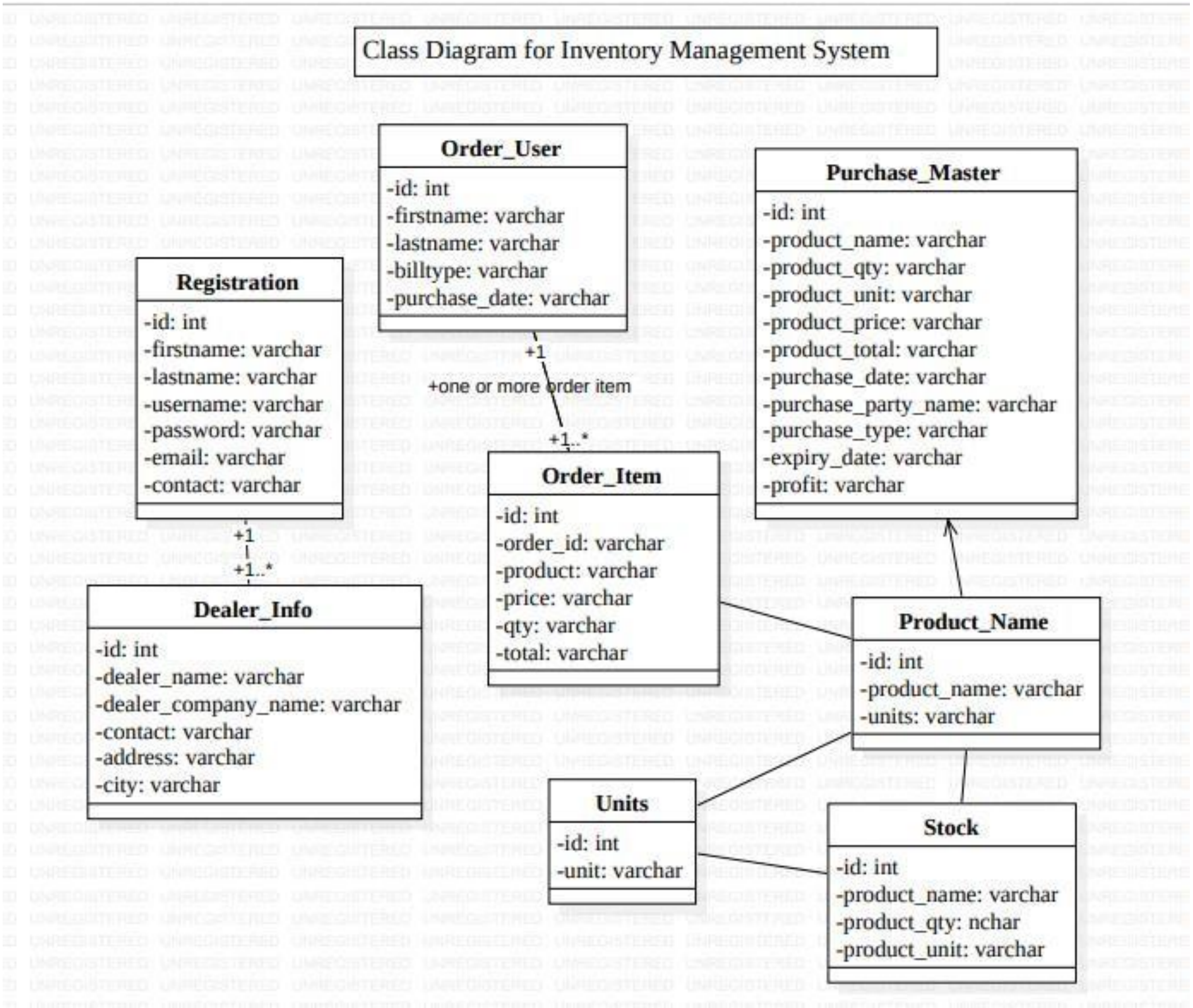
- Entities: Products, Suppliers, Users, Transactions.
- Relationships: Each Product has one or more Transactions; Suppliers

provide one or more Products; Users (with roles such as admin, manager) manage transactions and products.



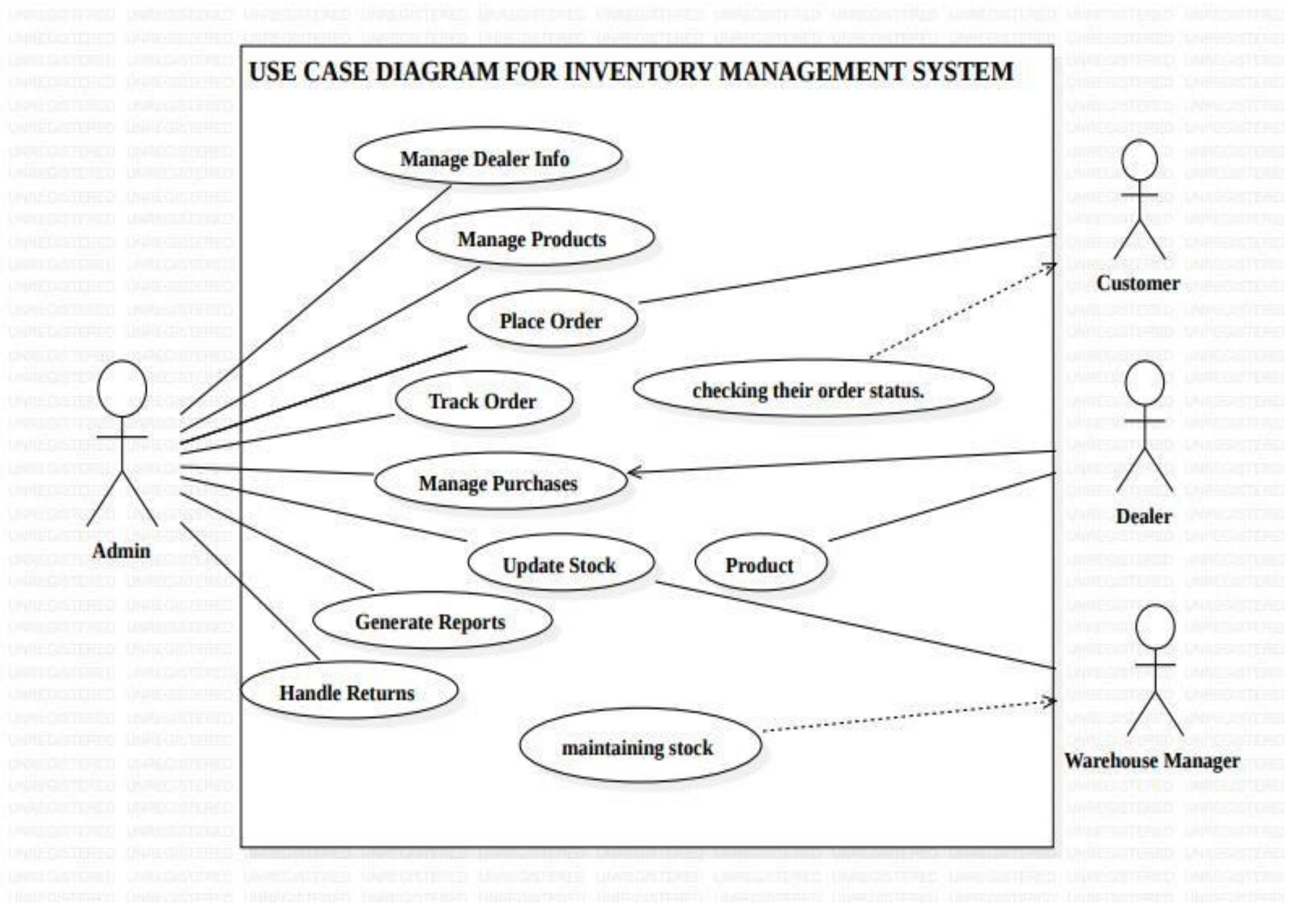
3. **Class Diagram:**

   - Classes: Product, Supplier, Transaction, User, ReportGenerator.
   - Attributes and methods for each class, such as ProductID, ProductName, GenerateReport(), AddProduct().



4. Use Case Diagram:

The **SRS (Software Requirements Specification)** for the **Inventory Management System** focuses on use cases like placing and tracking orders, managing products, updating stock, and generating reports. It defines user roles such as Admin, Customer, Dealer, and Warehouse Manager, specifying their interactions for tasks like maintaining stock, managing purchases, and checking order status.



Appendix C: Issues List

This section maintains a dynamic list of open issues and pending decisions. It will be updated regularly during the project lifecycle.

- **TBD (To Be Determined) Items:**
    1. **Authentication Method:** The type of authentication system (e.g., OAuth, JWT) to be integrated for user management is yet to be decided.
    2. **Localization Support:** Pending a decision on the number of languages to support in the initial release.
- **Pending Decisions:**
    1. **API Integration:** Whether to integrate with third-party APIs for automatic stock reordering from suppliers.
    2. **Cloud Deployment:** The decision regarding whether the system will be deployed on the cloud or on-premises is still under consideration.
- **Conflicts:**

1. **Role Permissions:** There is a conflict in the defined roles and permissions, especially concerning inventory management privileges between Store Managers and Admins.