

Table of Contents

1. Introduction.....	1
1.1 System Introduction.....	1
1.2 Stakeholders.....	2
.....	3
2. Requirement Analysis.....	3
2.1 Use Case Diagram.....	3
2.2 Functional and Non-Functional Requirements.....	3
3. High Level Design.....	4
3.1 Classes.....	4
3.2 Class Diagram.....	4
4. Data Modeling.....	5
4.1 Entity Relationship Diagram.....	5
.....	5
5. Detail Design.....	6
5.1 Screen Flow Diagram.....	6
6. System Testing.....	6
6.1 Login Testing.....	6
6.2 Add Data Testing.....	7
6.2.1 Add Product.....	7
6.2.2. Add Customer.....	7
6.2.3 Add Sales.....	8
6.3 Delete Data Testing.....	8
.....	8
6.3 Delete Data.....	9
6.3.1 Delete Product.....	9
6.3.2 Delete Customer.....	9
.....	9
7. Conclusion.....	9

1. Introduction

1.1 System Introduction

The Product Sales Management System is a web-based application designed to streamline and optimize the sales and inventory management processes for a retail store. Built using HTML, PHP, and MySQL, the system enables store managers and staff to manage product information, monitor inventory levels, and track customer purchases with ease. The primary objective is to maintain an accurate, real-time view of stock levels and ensure that each sales transaction is efficiently recorded and processed.

1.2 Stakeholders

The stakeholders are people or organizations who have an interest in the development of the system architecture.

Store Managers

Store managers utilize a system to track inventory levels, analyze sales data, and analyze consumer purchasing patterns to make informed decisions about sales and restocking.

Sales Staff

Sales employees utilize the system to streamline their daily activities, including adding orders, handling client data, and updating stock levels, ensuring accurate records and seamless transactions.

Customers

Customers benefit from improved stock control and efficient sales procedures, enhancing their shopping experience and reducing the risk of out-of-stock situations.

IT Developers / Support

The people who are responsible for maintaining, troubleshooting, and upgrading the system as required. They ensure the system remains functional, secure, and up-to-date with any new requirements or technological improvements.

2. Requirement Analysis

2.1 Use Case Diagram

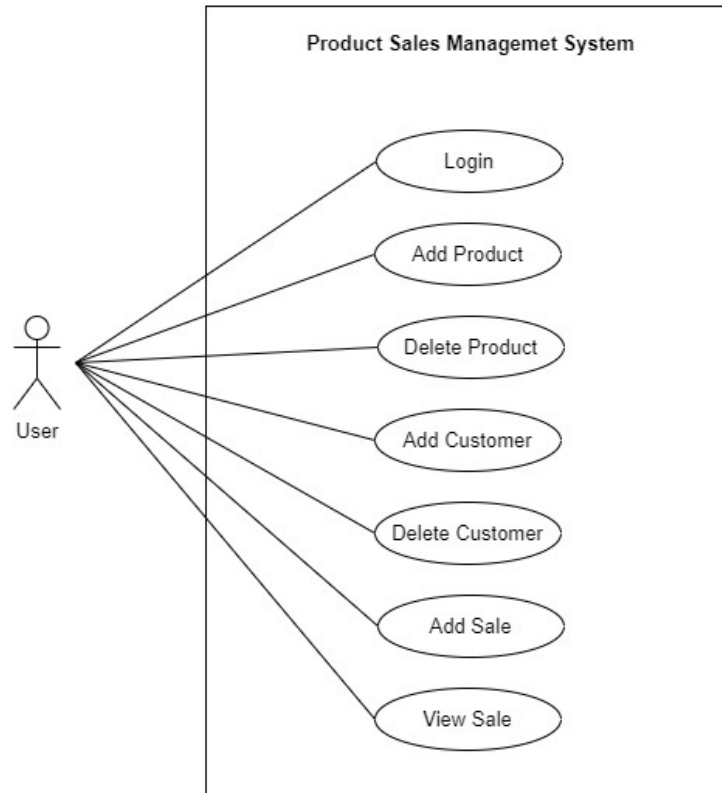


Figure 1 – Use Case Diagram

2.2 Functional and Non-Functional Requirements

Functional Requirements :

- Sales staff can log into the system using a secure username and password.
- Staff can add products, add customers, add sales and deletion operations of them respectively.
- The system links each sale to a specific product and customer.
- The system should automatically manage the stock levels when sales have been placed..

Non-Functional Requirements :

- The system should handle multiple sales ,add customers information efficiently without delays.
- Secure login mechanism with password encryption.
- Clear error messages and validation to guide users.
- The system should be structured so that updates, bug fixes, or feature additions can be applied with minimal disruption.

3. High Level Design

3.1 Classes

For the Hotel Room Booking System, followings are the classes that can be identified.

- Customer
- Product
- Sale
- User

3.2 Class Diagram

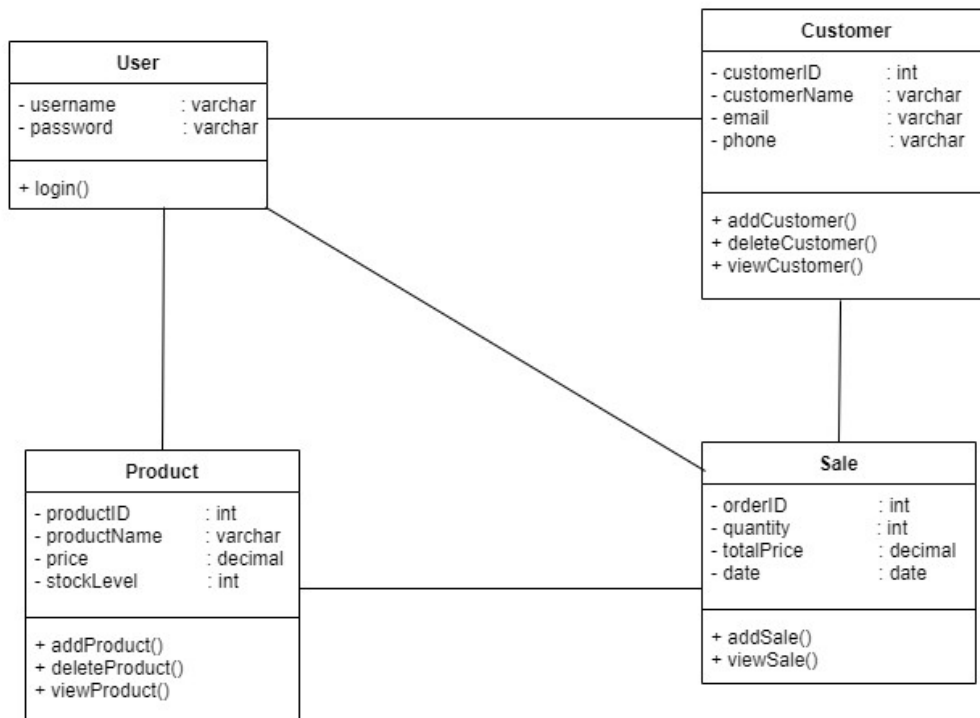


Figure 2 – Class Diagram

4. Data Modeling

4.1 Entity Relationship Diagram

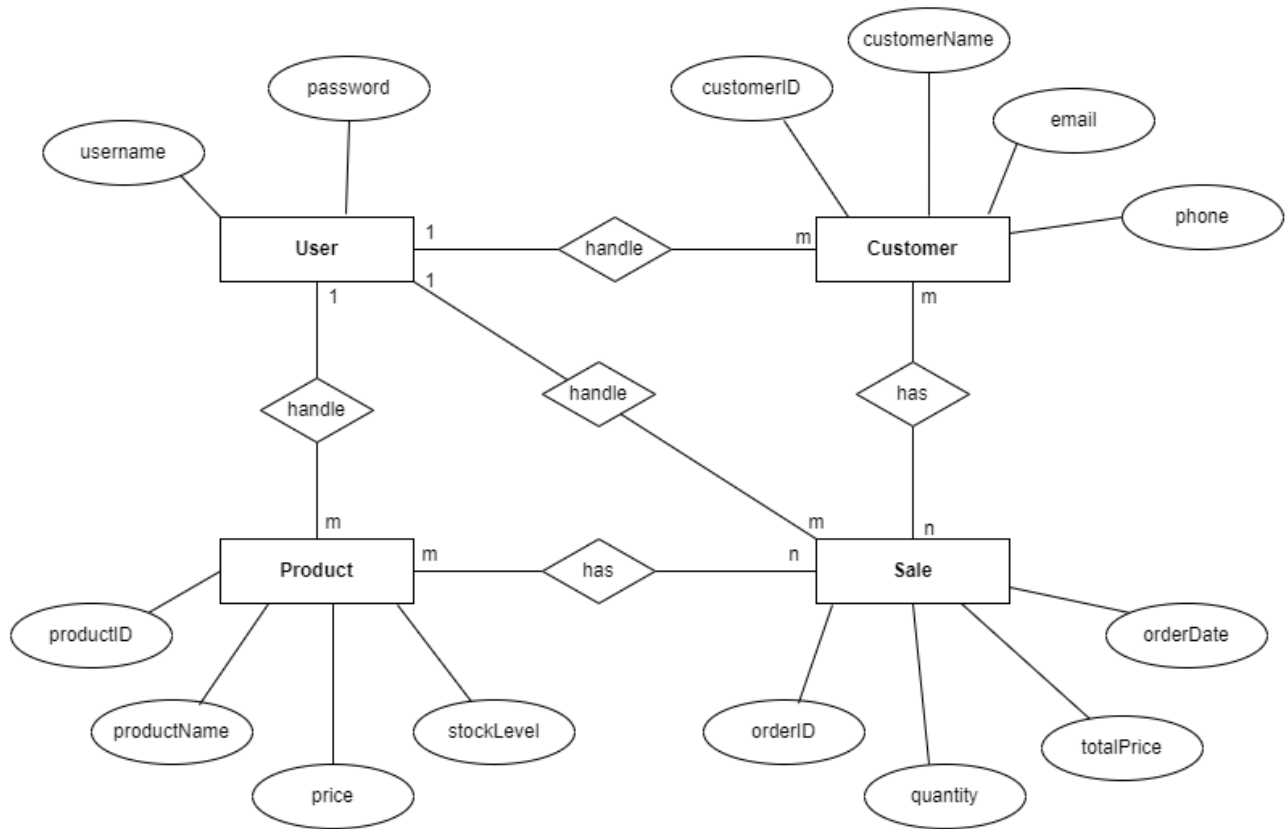


Figure 3 – ER Diagram

5. Detail Design

5.1 Screen Flow Diagram

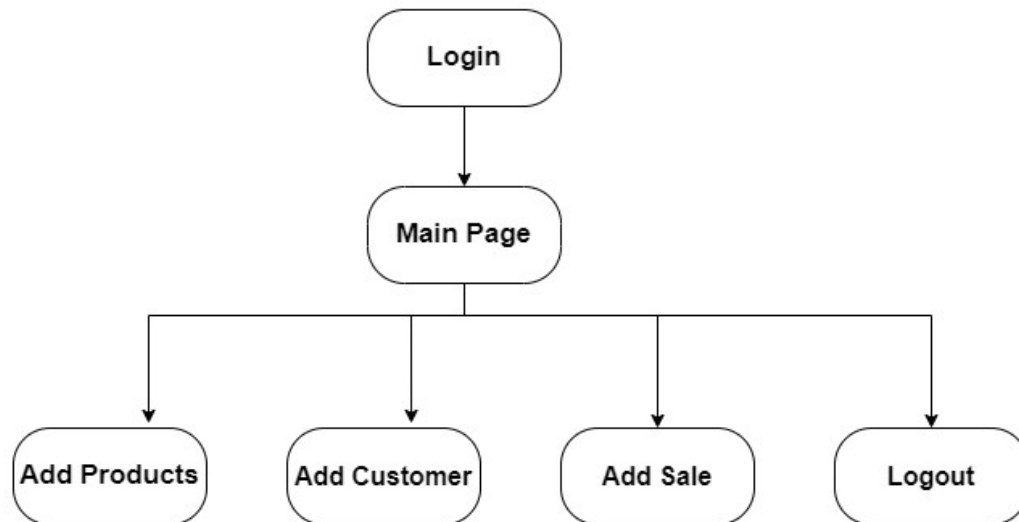


Figure 4 – Screen Flow Diagram

6. System Testing

6.1 Login Testing

- Login page will display an error message when entered incorrect credentials.

Invalid username or password!

- Correct credentials will direct user to the dashboard.

Welcome to Product Sales Management System

6.2 Add Data Testing

6.2.1 Add Product

User has to fill the Add New Product form with relevant data types, in order to add new product.



The form is titled "Add New Product" in a large, bold, black serif font. Below the title, there are three input fields: "Product Name:" followed by a text box containing "Product Name", "Price:" followed by a text box containing "Price", and "Stock Level:" followed by a text box containing "Stock Level". At the bottom of the form is a button labeled "Add Product". The entire form is set against a light gray background.

After entering the product, it will show in the Available Product table.



The table is titled "Available Products" in a bold, black serif font. It contains three rows of data. The first row has Product ID 1, Product Name Biscuit, Price 50.00, Stock Level 4, and a blue "Delete" link. The second row has Product ID 2, Product Name Face wash, Price 150.00, Stock Level 9, and a blue "Delete" link. The third row has Product ID 5, Product Name Oil, Price 100.00, Stock Level 50, and a purple "Delete" link. This third row is highlighted with a red rectangular border.

Product ID	Product Name	Price	Stock Level	Action
1	Biscuit	50.00	4	Delete
2	Face wash	150.00	9	Delete
5	Oil	100.00	50	Delete

6.2.2. Add Customer

User has to fill the Add Customer form with relevant data types, in order to add new customer.



The form is titled "Manage Customers" in a bold, black serif font, with a subtitle "Add Customer" below it. There are four input fields: "Customer ID:" followed by a text box, "Customer Name:" followed by a text box, "Email:" followed by a text box, and "Phone:" followed by a text box. At the bottom is a button labeled "Add Customer". The form has a light gray background.

Added Customer will show in the Existing Customers table.

Existing Customers				
Customer ID	Customer Name	Email	Phone	Actions
100	Zara	zara@example.xyz	741-852-7896	Delete
101	James	james@example.com	789-582-9631	Delete
102	John	john@example.com	856-852-9632	Delete
103	Tina	tina@example.com	741-789-8596	Delete

6.2.3 Add Sales

User has to fill the Add A New Sale form with relevant data types, in order to add new sale. And added sales will show in the Sales Records table.

Add a New Sale					
Customer: <input type="text" value="Select a customer"/>					
Product: <input type="text" value="Select a product"/>					
Quantity: <input type="text"/>					
<input type="button" value="Add Sale"/>					
Sales Records					
Order ID	Customer Name	Product Name	Quantity	Total Price	Order Date
1	Zara	Biscuit	1	\$50.00	2024-11-02
2	Zara	Biscuit	2	\$100.00	2024-11-02
4	Tina	Face wash	1	\$150.00	2024-11-03

6.3 Delete Data

6.3.1 Delete Product

User can easily delete a product by clicking on Delete Button in the Actions column in Available Products table.

6.3.2 Delete Customer

User can easily delete a customer by clicking on Delete Button in the Actions column in Existing Customers table.

7. Conclusion

A complete solution for handling inventory, product details, and sales transactions in a retail setting is offered by the Product Sales Management System. This system, which was created with HTML, PHP, and MySQL, maintains real-time data on sales orders and automates stock level adjustments to guarantee accurate and effective record-keeping. Store managers and employees can avoid stock discrepancies, expedite customer encounters, and make data-driven choices by centralizing vital data. All things considered, this solution improves operational effectiveness, reduces errors, and helps the retail establishment provide a dependable and smooth client experience.