Table of Contents

[1. Introduction 1](#__RefHeading___Toc188_3275986575)

[1.1 System Introduction 1](#__RefHeading___Toc190_3275986575)

[1.2 Stakeholders 2](#__RefHeading___Toc192_3275986575)

[3](#__RefHeading___Toc194_3275986575)

[2. Requirement Analysis 3](#__RefHeading___Toc196_3275986575)

[2.1 Use Case Diagram 3](#__RefHeading___Toc198_3275986575)

[2.2 Functional and Non-Functional Requirements 3](#__RefHeading___Toc200_3275986575)

[3. High Level Design 4](#__RefHeading___Toc202_3275986575)

[3.1 Classes 4](#__RefHeading___Toc204_3275986575)

[3.2 Class Diagram 4](#__RefHeading___Toc206_3275986575)

[4. Data Modeling 5](#__RefHeading___Toc208_3275986575)

[4.1 Entity Relationship Diagram 5](#__RefHeading___Toc210_3275986575)

[5](#__RefHeading___Toc212_3275986575)

[5. Detail Design 6](#__RefHeading___Toc214_3275986575)

[5.1 Screen Flow Diagram 6](#__RefHeading___Toc216_3275986575)

[6. System Testing 6](#__RefHeading___Toc218_3275986575)

[6.1 Login Testing 6](#__RefHeading___Toc220_3275986575)

[6.2 Add Data Testing 7](#__RefHeading___Toc222_3275986575)

[6.2.1 Add Product 7](#__RefHeading___Toc224_3275986575)

[6.2.2. Add Customer 7](#__RefHeading___Toc226_3275986575)

[6.2.3 Add Sales 8](#__RefHeading___Toc228_3275986575)

[6.3 Delete Data Testing 8](#__RefHeading___Toc230_3275986575)

[8](#__RefHeading___Toc232_3275986575)

[6.3 Delete Data 9](#__RefHeading___Toc675_1461596875)

[6.3.1 Delete Product 9](#__RefHeading___Toc677_1461596875)

[6.3.2 Delete Customer 9](#__RefHeading___Toc234_3275986575)

[9](#__RefHeading___Toc236_3275986575)

[7. Conclusion 9](#__RefHeading___Toc238_3275986575)

# 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

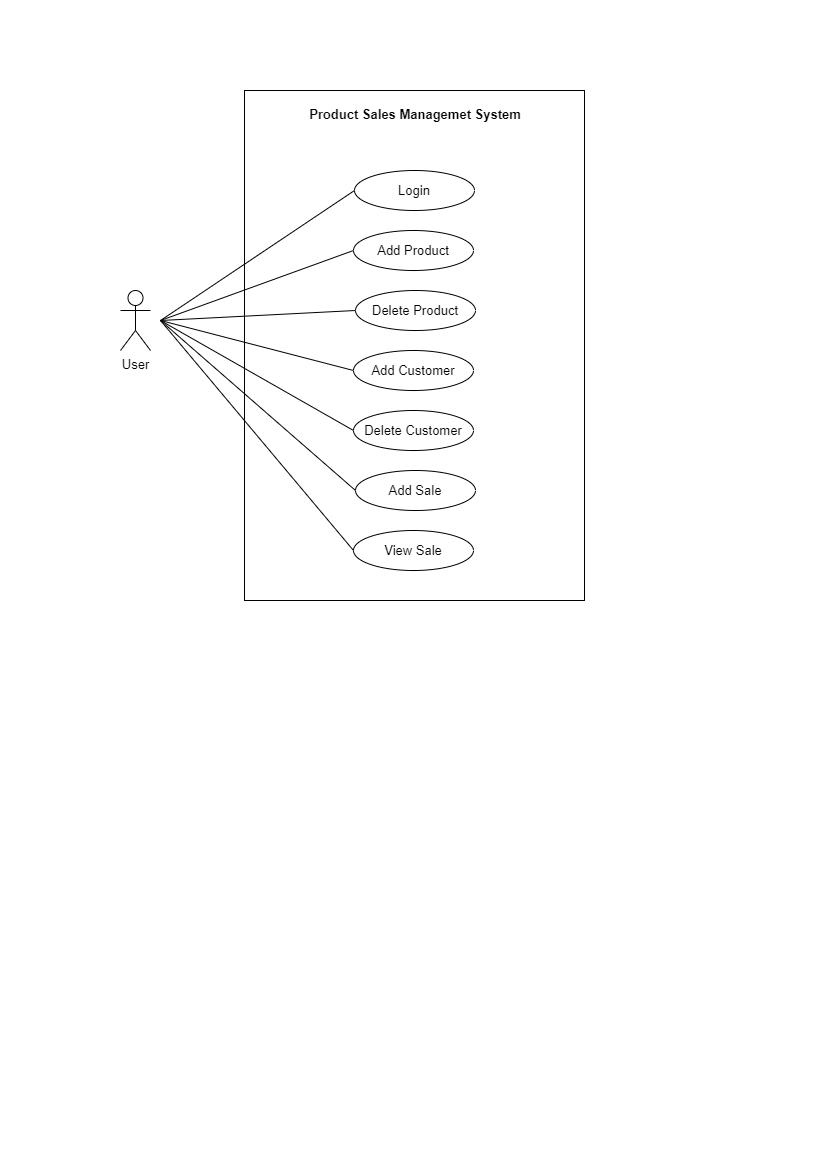


Figure 1 – Use Case Diagram

## 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

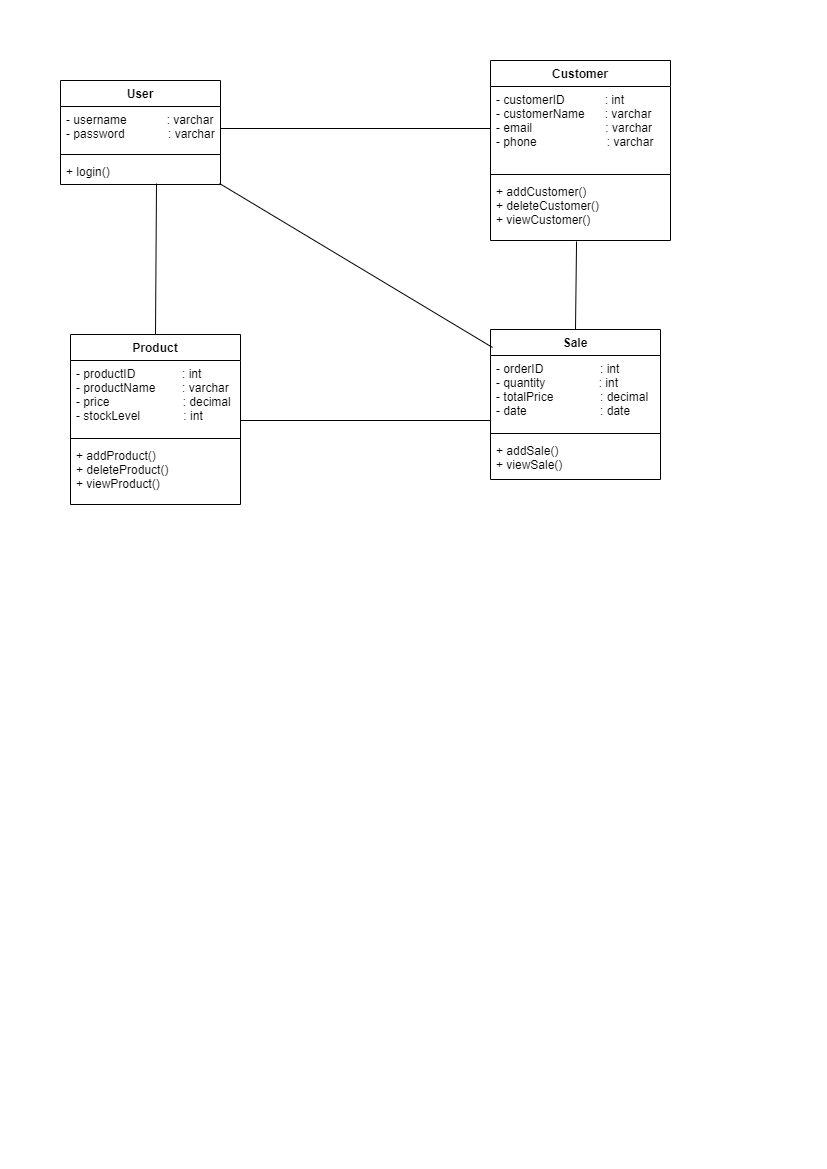


Figure 2 – Class Diagram

Figure 2 – Class Diagram

# 4. Data Modeling

## 4.1 Entity Relationship Diagram

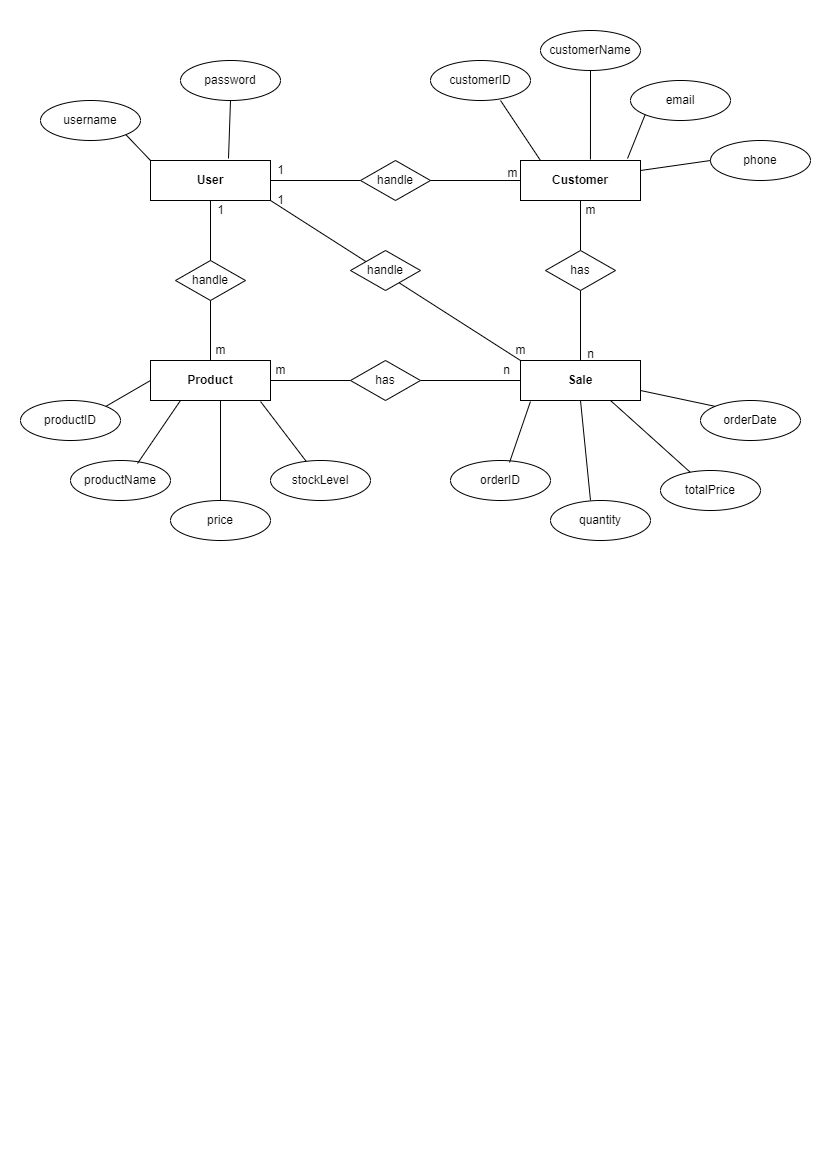


Figure 3 – ER Diagram

Figure 3 – ER Diagram

# 

# 5. Detail Design

### 5.1 Screen Flow Diagram

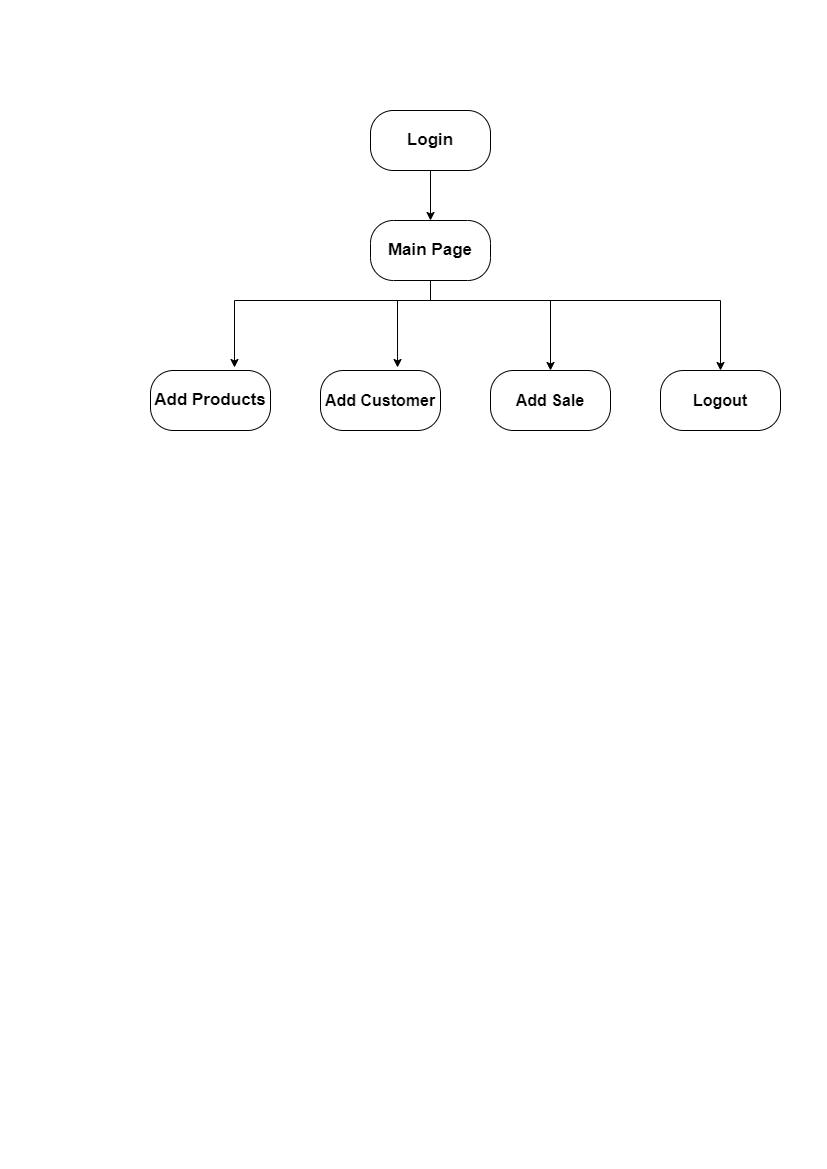


Figure 4 – Screen Flow Diagram

# 6. System Testing

## 6.1 Login Testing

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



* Correct credentials will direct user to the dashboard.



## 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.



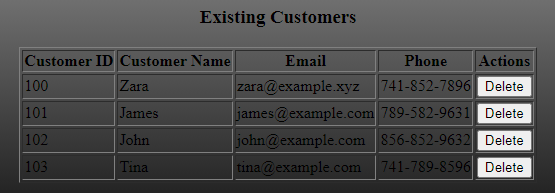
After entering the product, it will show in the Available Product table.



### 6.2.2. Add Customer

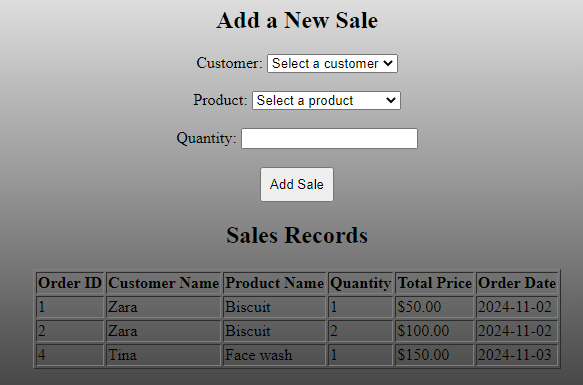
User has to fill the Add Customer form with relevant data types, in order to add new customer.

Added Customer will show in the Existing Customers table.



### 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.



## **6.3 Delete Data Testing**

### 

## 

## 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.