

# **Boutique Management System**

Muhammad Yousaf

21311/411711

Introduction of Software Engineering

Mam Somia

May 22, 2023

## **Boutique Management System**

A boutique management system is a software application that helps boutique owners and managers to manage their business operations efficiently and effectively. It can handle various tasks such as inventory management, sales management, customer management, employee management, product management and more.

A boutique management system can have different actors or users who can access the system with different privileges and functionalities.

**Customers** Customers can create their accounts, sign in to their accounts and perform various operations such as book their orders, view booking, delete bookings and update their details

**Employees (Sales Staff)** Employees can sign in to their account using their employee id and perform operations such as updating delivered orders of customers, adding a new product, and deleting a product. It include all those person who work in the boutique.

**Designer** Designer is responsible to create new design, customize previous design, check the customers demand and create the design according to the customer requirements.

### **Work perform**

**Inventory Management** The system provides functionality to manage boutique inventory. This includes adding new products, updating stock levels, categorizing items, and setting prices.

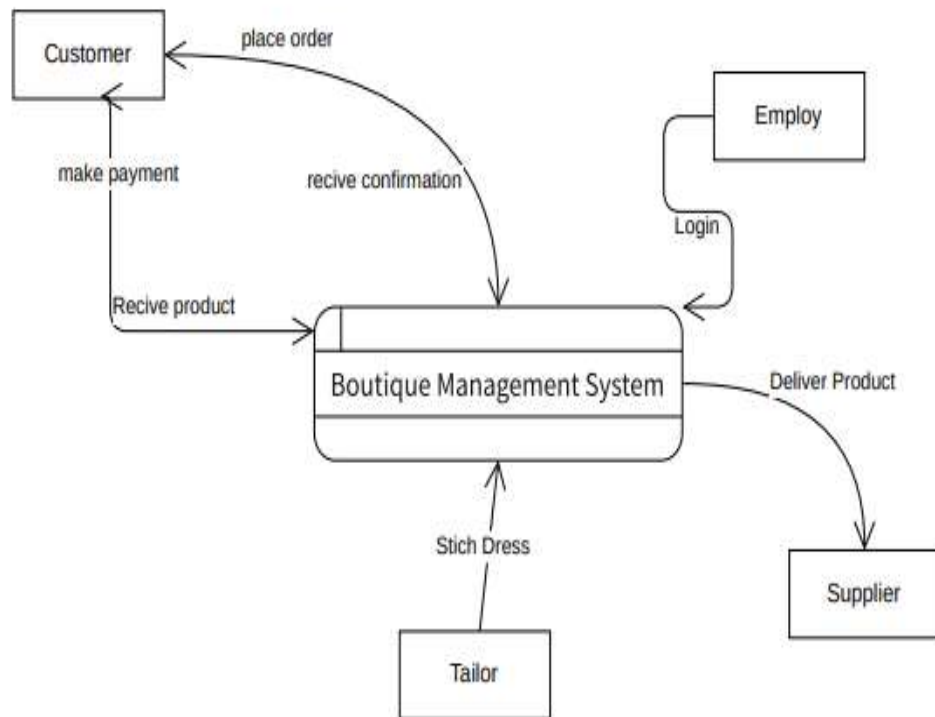
**Sales Process** When a customer visits the boutique, the sales staff can use the system to select items, check availability, and generate invoices. The system deducts sold items from the inventory and maintains a record of the transaction.

**Customer Management** The system allows boutique staff to store customer information, including contact details, purchase history, preferences, and loyalty program data. This helps in maintaining personalized customer interactions and targeted marketing.

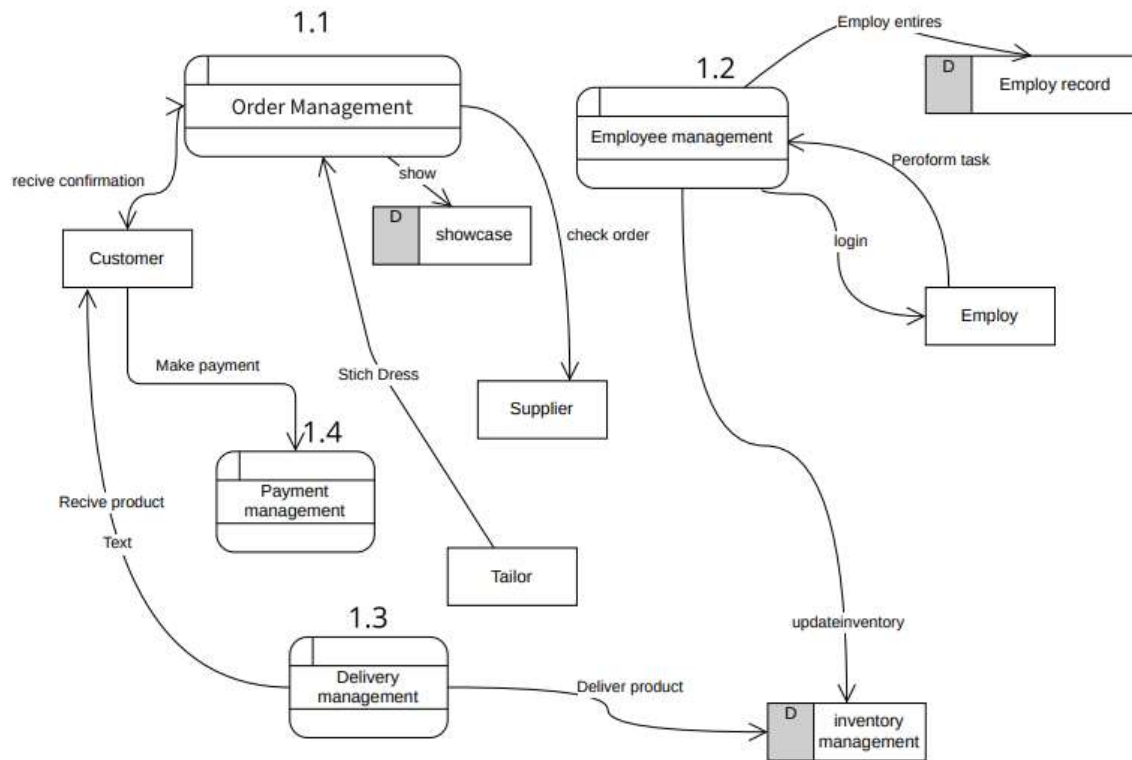
**Reporting and Analytics** The system generates reports and analytics to provide insights into various aspects of the boutique's performance. This includes sales reports, inventory status, popular products, customer trends, and financial summaries.

## Data flow Diagram

### Level 0



## Level 1



## Use Case Model

A Use Case Model describes the proposed functionality of new system. A Use Case represents a discrete unit of interaction between a user (human or machine) and the system.

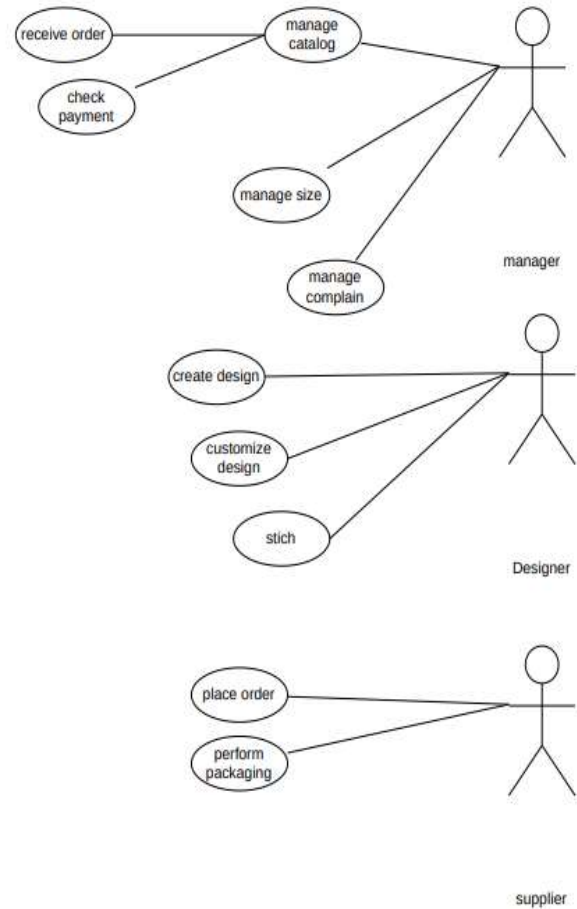
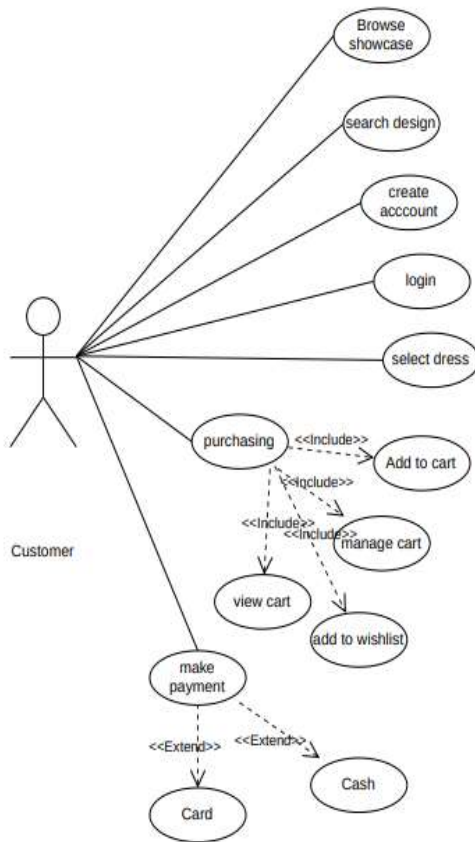
### List of Actors

- Customer; this person check items, buy them and give feedback.
- Manager; this person manages the whole system.
- Employ; the persons who work in boutique
- Supplier; the person who ship dress
- Designer & Tailor the person who design dress

## List of Use Cases

- Browse Showcase; customer select dress to buy
- Search New Design; customer finds dress according to his requirements
- Create Account; customer creates account to buy dress.
- Login/Register; customer login to the site to buy dress.
- Manage Dress; customer manages the selected dress
- Select dress; customer selects the desired item.
- Manage Cart; customer can easily manage his/her cart.
- Add to Cart; customer adds dress to cart.
- Add to Wish List; customer adds the dress to his/her wish list to buy the dress later.
- View Cart; customer views cart to see what he/she selected.
- Edit Dress Quality; customer tells the quality/brand that he/she wants.
- Place Order; customer order the selected dress.
- Make Payment; customer pays for dress.
- By Credit Card; customer can use credit card for payment.
- By Hand; customer pays by hand when he receive dress.
- Manage Catalog; administration manage the whole catalog. Add new dress for customer to search and buy.
- Receive Orders; administration receives order from customer.
- Check Payment; administration checks the payment paid by customer.
- Manage Measures; administration manages the quantity of products.
- Manage Complains; administration manage complains by customers.
- Design dress; customer design according to his requirments
- Make Payment; customer pays for item.

## UC Diagram



## Functional Requirements

Here are some functional requirements for a boutique management system:

### 1. Inventory Management

- The system should allow boutique staff to add, update, and delete product information, including details such as name, description, SKU, price, and quantity.
- It should provide functionality to track and manage stock levels, including notifications for low stock or out-of-stock items.
- The system should support categorization and organization of products for easier navigation and management.

## **2. Sales and Point of Sale (POS)**

- It should support multiple payment methods such as cash, credit/debit cards, and digital wallets.
- The system should maintain a sales history and provide reports for analyzing sales performance.

## **3. Customer Management**

- The system should allow boutique staff to create and manage customer profiles, including contact information, purchase history, loyalty program data, and preferences.
- It should provide functionality for searching and retrieving customer information quickly.
- The system should support the ability to generate customer-related reports, such as customer demographics or purchasing patterns.

## **4. Reporting and Analytics**

- The system should generate reports on various aspects of boutique operations, such as sales trends, inventory status, top-selling products, and revenue analysis.
- It should provide analytics capabilities to gain insights into key performance indicators (KPIs) like sales growth, customer retention, and product profitability.
- The system should offer customizable reporting options to cater to specific reporting needs.

## **5. Promotions and Discounts**

- The system should support the management of promotional campaigns, such as discounts, coupons, or loyalty rewards programs.
- It should allow boutique staff to define and apply promotional rules, track usage, and calculate discounts accurately during sales transactions.

## **6. Vendor Management**

- The system should enable the management of vendor information, including contact details, products supplied, pricing agreements, and payment terms.
- It should support tracking vendor orders, delivery status, and vendor performance metrics.

## **7. User Access and Security**

- The system should provide role-based access control to ensure that only authorized users can perform specific actions or access certain information.
- It should include features like user authentication, password management, and audit trails to maintain data security and accountability.

## **8. Integration Capabilities**

- The system should have the ability to integrate with external systems, such as payment gateways, accounting software, or e-commerce platforms, to streamline operations and data synchronization.

## **Non-Functional Requirements**

### **1. Usability**

The system should have a user-friendly and intuitive interface, making it easy for boutique staff to navigate, perform tasks efficiently, and quickly learn how to use the system.

### **2. Performance**

The system should have fast response times, ensuring that operations like inventory lookup, sales transactions, and report generation are processed in a timely manner, even under high user loads.

### **3. Reliability**

The system should be reliable and available for use during boutique operating hours, minimizing downtime and ensuring data integrity.

### **4. Scalability**

The system should be able to handle increasing data volumes, growing product catalogs, and a larger customer base without significant performance degradation. It should scale seamlessly to accommodate future business growth.

### **5. Security**



The system should enforce proper security measures to protect sensitive data, including customer information, payment details, and business records. It should include features like encryption, access controls, and regular data backups.

## **6. Compatibility**

The system should be compatible with different hardware and software environments commonly used in boutiques. It should support various operating systems, web browsers, and mobile devices, ensuring broad accessibility.

## **7. Integration**

The system should have the capability to integrate with third-party systems, such as payment gateways, accounting software, or customer relationship management (CRM) systems, allowing seamless data exchange and process automation.

## **8. Maintainability**

The system should be designed and developed in a way that allows for easy maintenance, updates, and bug fixes. It should follow best practices and coding standards, promoting code modularity and documentation.

## **9. Data Backup and Recovery**

The system should include mechanisms for regular data backups and the ability to restore data in the event of system failures, ensuring data integrity and business continuity.

## **10. Localization and Internationalization**

If the boutique operates in multiple regions or serves customers from different countries, the system should support localization features, including multilingual support, regional date and currency formats, and compliance with relevant regulations.

## Class Diagram

### Attribute and Method of class

**Customer:** Name, Customer ID

Choose Design(), Pay bill(), Watch Showcase()

**Manager:** Name, ID

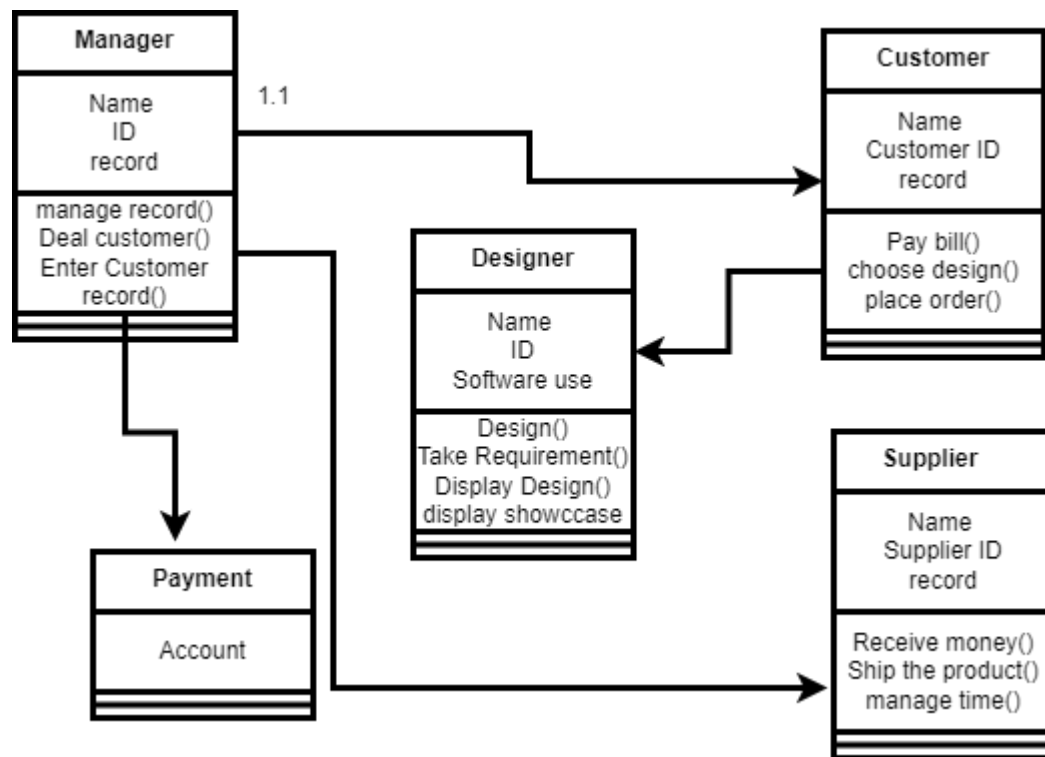
Manage Records(), Deal Customer(), Manage record()

**Supplier:** Name, Supplier ID(), Convince

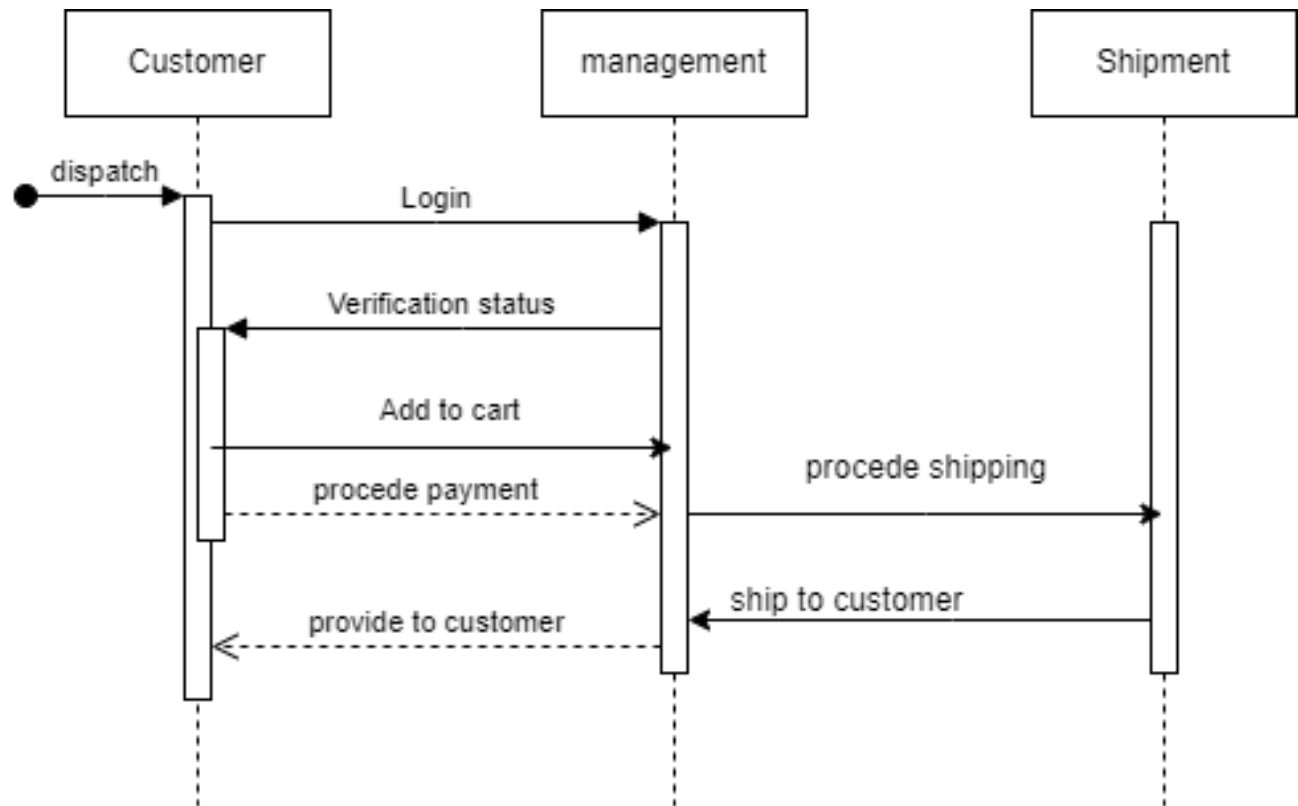
Time Taken(), supply(), Receive Money()

**Designer:** Name, ID, Software use

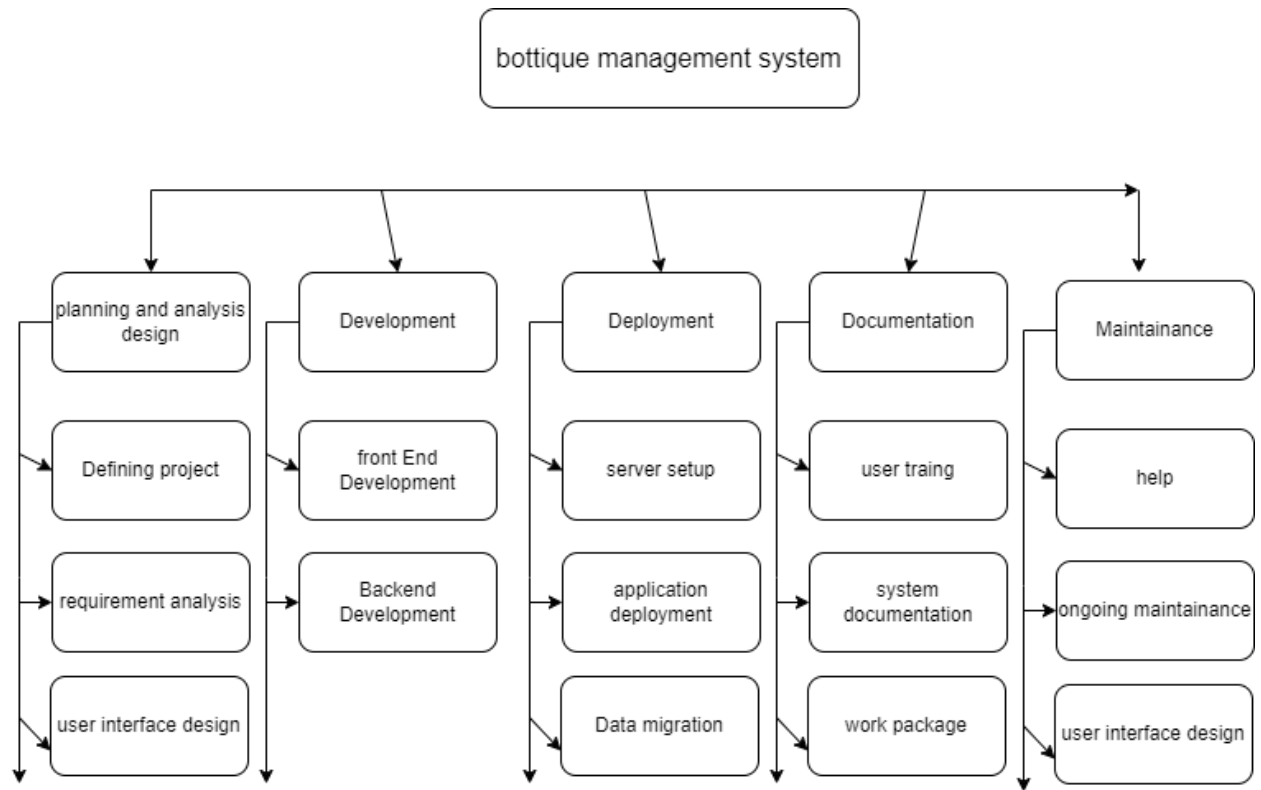
Design(), Take Requirements(), Display Design()



## Sequence Diagram



## WBS Diagram



## ER Diagram

