**CHAPTER THREE**

**SYSTEM ANALYSIS AND DESIGN**

**System Analysis**

In this phase, the goal was to gain a clear understanding of Ultracare Pharmacy’s current operations and identify the system requirements that would guide the design and development of a fully functional Pharmacy Management System. The analysis focused on both functional and non-functional requirements, which were gathered through staff feedback, direct observation of daily workflows, and a review of existing records and processes.

**Introduction**

This section provides an analysis of the current processes used in managing pharmacy operations, highlights the limitations of the existing manual system, and outlines the requirements for the proposed Pharmacy Management System. The goal is to understand the functional needs and operational context that the system must address in order to improve efficiency, accuracy, and overall service delivery.

**System Design**

**Requirement Specifications**

### ****1. Functional Requirements****

These define **what the system must do (features and modules):**

1. **User and Role Management**

Allow user registration and login for Admin, Pharmacist, Cashier, Procurement staff, and Customers.

Implement role-based access control so users only access features relevant to their roles.

1. **Prescription Management**

Record, verify, and archive prescriptions digitally.

Maintain prescription history for each customer and ensure regulatory compliance.

1. **Inventory Management**

Track stock levels in real-time after every sale or stock delivery.

Monitor batch numbers and expiry dates with automatic alerts for low stock or soon-to-expire medicines.

Generate automatic restocking notifications.

1. **Sales & Point-of-Sale (POS)**

Process sales and bulk purchases with support for cash, M-Pesa, and bank transfers.

Automatically calculate totals, taxes, and discounts, and generate digital receipts.

Maintain digital records of daily, weekly, and monthly sales.

1. **Supplier & Procurement Management**

Manage supplier information and maintain purchase orders.

Track deliveries and automatically update inventory upon confirmation.

1. **Bulk Order Handling**

Capture large orders from clinics and institutions.

Automatically generate invoices and track order progress until delivery and payment.

1. **Online Pharmacy Platform**

Allow customers to browse products, upload prescriptions, place orders, and choose delivery or pickup options.

Display real-time product availability and enable secure checkout.

1. **Reporting & Analytics**

Generate reports on sales, inventory, expiry dates, and bulk orders.

Provide dashboards for admins and management to view key performance metrics.

### ****Non-Functional Requirements****

These define **how the system should perform (quality attributes):**

1. **Performance**

Transactions and inventory updates must process in real-time with minimal delay.

Support at least 10 concurrent users without affecting performance.

1. **Security**

Enforce secure login with encrypted passwords and user authentication.

Restrict data access based on user roles (e.g., only pharmacists can approve prescriptions).

Encrypt sensitive data (customer, prescription, and payment records) during storage and transmission.

1. **Usability**

Provide a simple, intuitive interface that staff with minimal technical skills can use effectively

The system must be compatible with both desktops and mobile devices (for online customers).

1. **Reliability & Availability**

Ensure 99% uptime during pharmacy operational hours.

The system must handle unexpected failures without losing critical data.

1. **Backup & Recovery**

Perform automatic backups at least once a day.

Allow recovery of data from backups in the event of a system crash or database failure.

1. **Scalability**

The system must be designed to accommodate future expansion, including multiple branches or additional users.

1. **Compliance & Ethics**

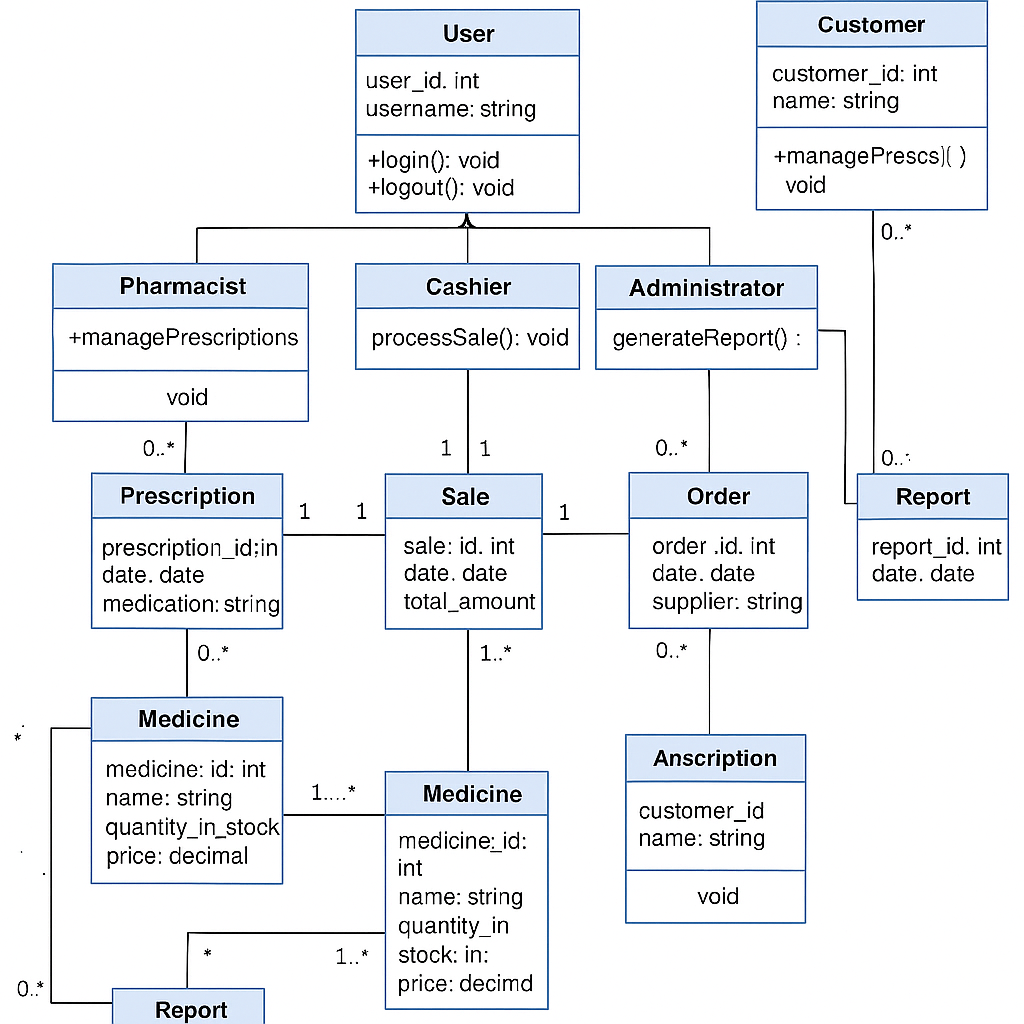
After establishing the system requirements, the design phase outlined how the Pharmacy Management System would be structured to meet those needs. The system design was divided into four key parts: Logical Design, Physical Design, Interface Design, and Database Design. Together, these components served as the blueprint for the development phase.

**LOGICAL DESIGN**

The logical design represents the abstract structure of the Pharmacy Management System without focusing on its physical implementation. It involves modeling user interactions, data flow, and system behavior using diagrams such as use case diagrams, class diagrams, package diagrams, object diagrams, and entity-relationship diagrams (ERDs).

This phase defined how different users—pharmacists, cashiers, procurement staff, administrators, and customers—interact with the system, including tasks such as prescription management, sales processing, inventory updates, bulk order handling, and reporting. Relationships between users and data elements were clearly mapped to ensure efficient and accurate system operations.

**Class Diagram**



It defines this classes:

1. **User**

* Attributes: userID, name, email, role, password
* Methods: login(), logout()

1. **Customer (inherits User)**

* Attributes: contact, address
* Methods: placeOrder(), viewPrescriptionHistory()

1. **Pharmacist (inherits User)**

* Methods: verifyPrescription(), manageInventory()

1. **Cashier (inherits User)**

* Methods: processPayment(), issueReceipt()

1. **ProcurementStaff (inherits User)**

* Methods: createPurchaseOrder(), manageSuppliers()

f) **Medicine**

* Attributes: medicineID, name, batchNo, expiryDate, quantity, price
* Methods: updateStock(), checkExpiry()

g) **Prescription**

* Attributes: prescriptionID, date, dosage
* Methods: validatePrescription()

h) **Order**

* Attributes: orderID, date, status
* Methods: addMedicine(), calculateTotal()

i) **OrderDetail** (association class between Order and Medicine)

* Attributes: quantity, price

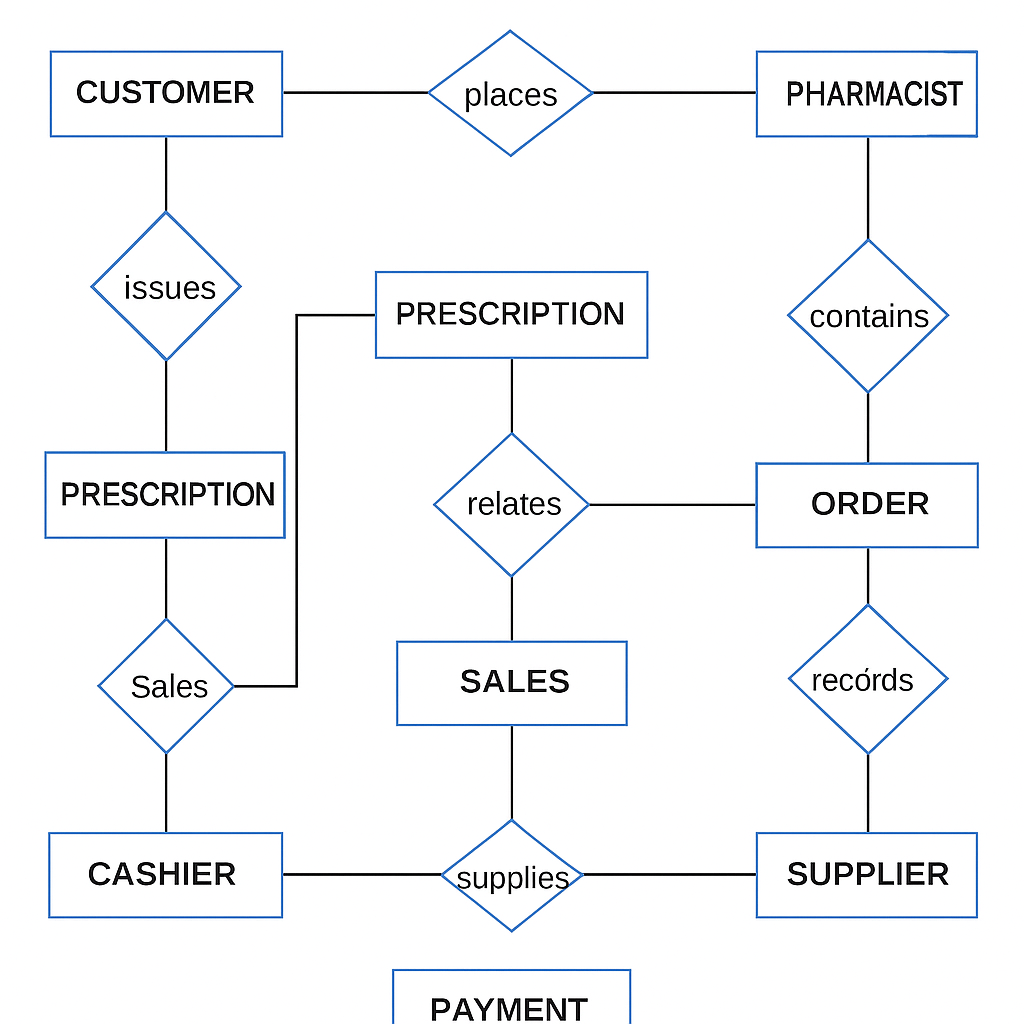
j) **Supplier**

* Attributes: supplierID, name, contact, address
* Methods: supplyMedicine()

k) **Payment**

* Attributes: paymentID, method, amount, date
* Methods: processPayment()

**Entity Relationship Diagram**

****

### ****Entities and Relationships****

The Pharmacy Management System is built around several core entities that represent real-world objects and actors involved in daily operations. These entities interact through well-defined relationships that reflect how data is shared and processed within the system.

#### **Entities**

1. **User**
   * Attributes: user\_id, name, email, role, password
   * Represents all system users including pharmacists, cashiers, procurement staff, and administrators.
2. **Customer**
   * Attributes: customer\_id, name, contact, address
   * Represents individuals who purchase medication or submit prescriptions.
3. **Pharmacist** (Inherits from User)
   * Handles prescription verification and inventory control.
4. **Cashier** (Inherits from User)
   * Processes customer payments and generates receipts.
5. **Procurement Staff** (Inherits from User)
   * Manages suppliers and oversees ordering of stock.
6. **Medicine**
   * Attributes: medicine\_id, name, batch\_number, expiry\_date, quantity, price, supplier\_id
   * Represents all medicines in the system, tracked for availability, expiry, and pricing.
7. **Prescription**
   * Attributes: prescription\_id, customer\_id, medicine\_id, dosage, date
   * Captures prescriptions issued to customers and approved by pharmacists.
8. **Order**
   * Attributes: order\_id, customer\_id, date, status
   * Represents customer orders, whether walk-in or online.
9. **OrderDetail**
   * Attributes: order\_id, medicine\_id, quantity, price
   * A linking entity between orders and medicines that details what was ordered and in what quantity.
10. **Supplier**
    * Attributes: supplier\_id, name, contact, address
    * Represents external pharmaceutical suppliers who provide stock.
11. **Payment**
    * Attributes: payment\_id, order\_id, method, amount, date
    * Tracks financial transactions for orders, including M-Pesa, cash, and bank transfers.

#### **Relationships**

* **User** has roles (Pharmacist, Cashier, Procurement Staff) that determine access permissions.
* **Customer** can place one or more **Orders**.
* **Order** consists of multiple **OrderDetails**, each linking to a specific **Medicine**.
* **Prescription** is created for a **Customer** and references specific **Medicine** records.
* **Pharmacist** verifies **Prescriptions**.
* **Cashier** processes **Payments** for **Orders**.
* **Procurement Staff** places **Orders** with **Suppliers**, who supply **Medicines**.
* **Each Payment** is associated with exactly one **Order**.

## ****Physical Design****

The Physical Design phase focuses on how the Pharmacy Management System will be implemented, specifying the inputs, outputs, data storage, processing, and system controls.

### ****1. Input Requirements****

The system will use user-friendly forms to capture accurate data:

**Login Form:** Username and password.

**Prescription Entry Form:** Customer information, medicine details, dosage, quantity, and doctor information.

**Inventory Management Form:** Medicine name, batch number, expiry date, quantity, purchase price, and supplier details.

**Sales/POS Form:** Product selection, quantity, price, discounts, payment method (cash, M-Pesa, card).

**Supplier/Procurement Form:** Supplier details, purchase order details, order date, and expected delivery date.

**Online Order Form:** Customer details, selected products, delivery address, and payment method.

### ****2. Output Requirements****

The system will generate the following outputs:

**Receipts:** Printed or digital receipts with itemized purchases, total cost, and payment method

**Reports:**

Daily, weekly, and monthly sales reports.

Inventory stock level reports.

Expiry date alerts and reports.

Supplier performance and delivery reports.

**Dashboards:** Graphical summaries of sales, stock levels, and order statuses for staff and management.

**Notifications:** Low stock alerts, payment confirmations, and online order updates.

### ****3. Storage Requirements****

The system will use a **MySQL database** that stores all information securely:

**Tables:** Users, Customers, Medicines, Prescriptions, Orders, Order\_Details, Suppliers, Payments.

**Data Relationships:** Established using primary keys (e.g., MedicineID, OrderID) and foreign keys for data integrity.

**Indexes:** Applied to frequently queried fields (e.g., MedicineName, CustomerName) to improve performance.

**Data Retention:** Historical data will be archived periodically to maintain database efficiency

### ****4. Processing Requirements****

The system will process data as follows:

**Stock Management:** Deduct quantities automatically after each sale and update stock after deliveries.

**Prescription Verification:** Validate prescription details before dispensing medicines.

**Sales Calculations:** Automatically compute totals, discounts, and taxes at the POS.

**Expiry Alerts:** Check expiry dates during inventory updates and generate notifications

for soon-to-expire products.

**Reports & Dashboards:** Aggregate and display real-time data for decision-making.

### ****5. System Controls & Backup****

To ensure data security and reliability, the system will include:

**User Access Control:** Secure login with encrypted passwords and role-based permissions (Admin, Pharmacist, Cashier, Procurement Staff).

**Data Validation:** All inputs validated both on the client-side (frontend) and server-side (backend) to prevent errors.

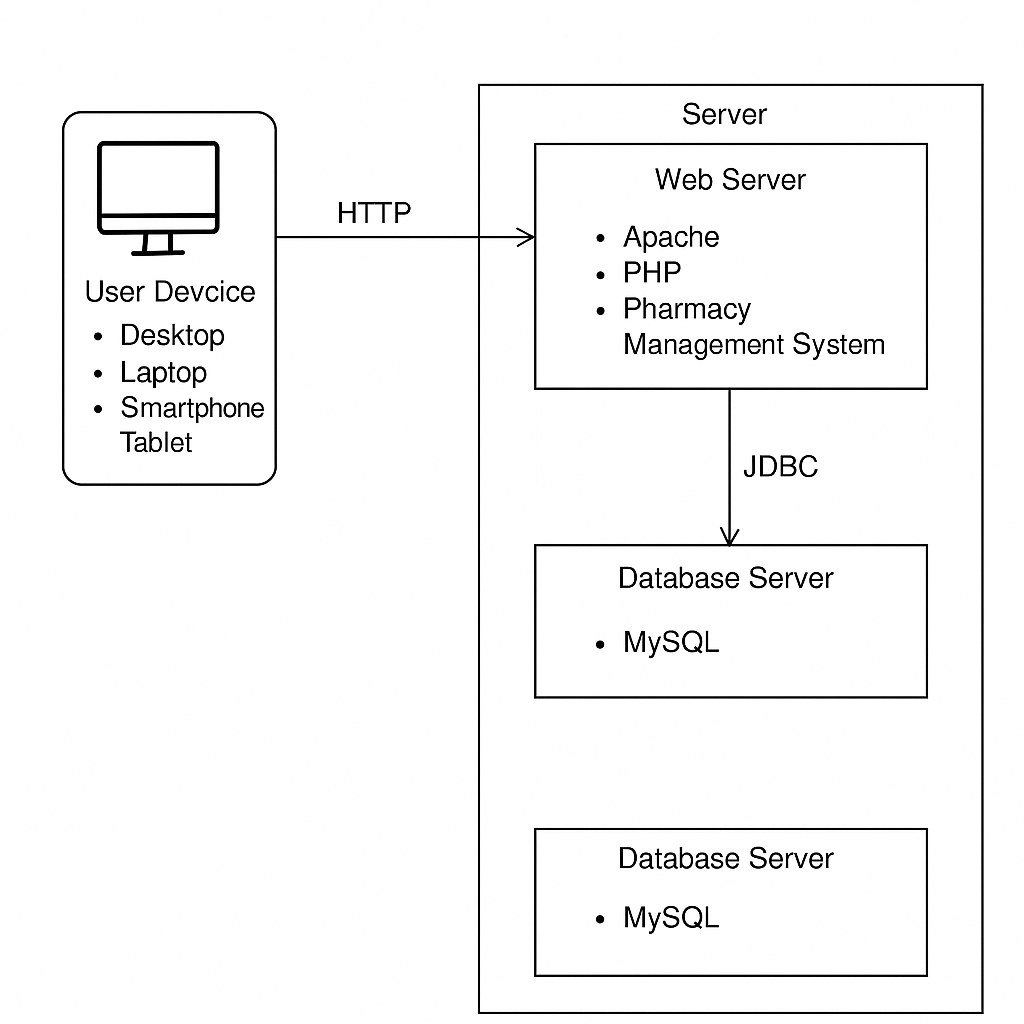
**Error Handling:** Friendly error messages for users and error logging for administrators.

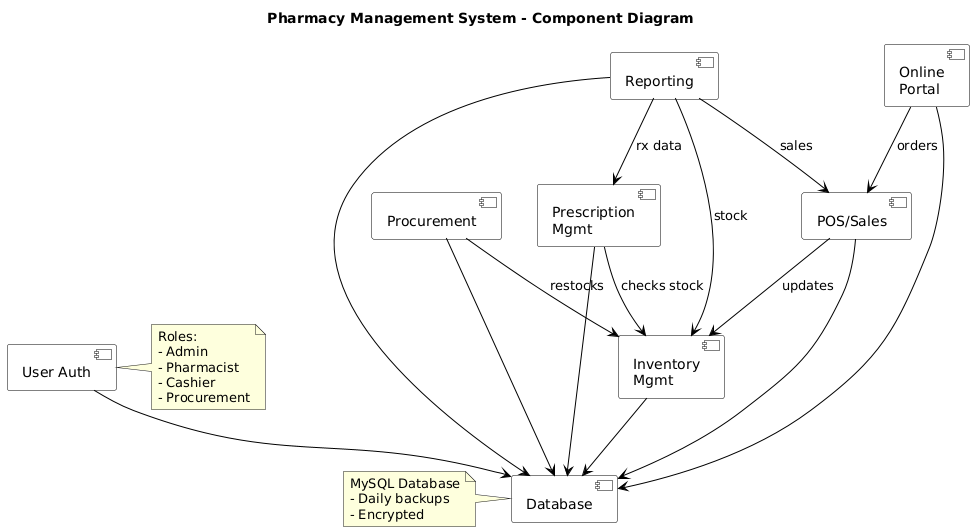
**Backups:** Automatic daily database backups with a restore option in case of data loss.

**Data Security:** Sensitive data (passwords, customer information) encrypted during storage and transmission.

**Deployment diagram**

A deployment diagram illustrates how the Pharmacy Management System’s software components are distributed across hardware devices, such as staff computers, servers, and customer devices, and how they interact with each other.





The component diagram illustrates how the Pharmacy Management System is organized into distinct functional modules and how they interact with each other. It shows the **User Authentication** module enforcing role-based access control for Admin, Pharmacist, Cashier, and Procurement staff. The **Procurement** module handles supplier orders and restocking, while the **Prescription Management** module manages prescription entry and verification. **Inventory Management** tracks real-time stock levels, receiving updates from Procurement and **POS/Sales**, which processes both in-store and online transactions. The **Online Portal** allows customers to place orders that are directly linked to the POS/Sales module. The **Reporting** module compiles data from sales and prescriptions to generate analytical reports for management. All components interact with the centralized **MySQL database**, which is encrypted and backed up daily for security and reliability. This modular structure ensures efficient data flow, accurate real-time updates, and improved scalability of the system.

## ****Interface Design****

The Interface Design focuses on how users will interact with the Pharmacy Management System (Ultracare PMS). It defines the appearance, layout, and navigation of the system’s screens to ensure usability, efficiency, and accuracy.

### ****1. Purpose of the Interface Design****

The main objective of the interface design is to provide users (pharmacy staff, customers, and suppliers) with an intuitive and user-friendly environment. Interfaces are designed to:

Minimize input errors through clear prompts and validations.

Enable quick access to core functions such as inventory management, sales, and reporting.

Ensure consistent layout, colors, and font styles across all modules.

Support both desktop and mobile devices for flexibility.

### ****2. Key Interfaces****

The system will have the following major interfaces:

**Login Page**

Fields: Username and Password.

Buttons: Login, Reset Password.

Features: Displays error messages if invalid credentials are entered.

**Dashboard**

Displays a summary of stock levels, sales performance, pending orders, and alerts.

Provides quick navigation links to other modules (POS, Inventory, Suppliers, Reports).

**POS (Point-of-Sale) Screen**

Allows staff to process customer sales and payments.

Fields: Customer details, selected products, quantity, payment method (cash, M-Pesa, card).

Generates and prints receipts after a successful transaction.

**Inventory Management Screen**

Used by staff to add, edit, or delete medicine records.

Displays stock levels, batch numbers, and expiry dates.

Shows automatic alerts for low stock or expired products.

**Suppliers & Procurement Screen**

Manages supplier information and purchase orders.

Tracks pending deliveries and updates stock levels upon receipt.

**Reports & Analytics Screen**

Generates detailed reports on sales, inventory, and suppliers.

Allows export to PDF or Excel for further analysis.

**Online Orders Screen (Customer Interface)**

Displays the product catalog with a search feature.

Allows customers to place orders, provide delivery details, and make online payments.

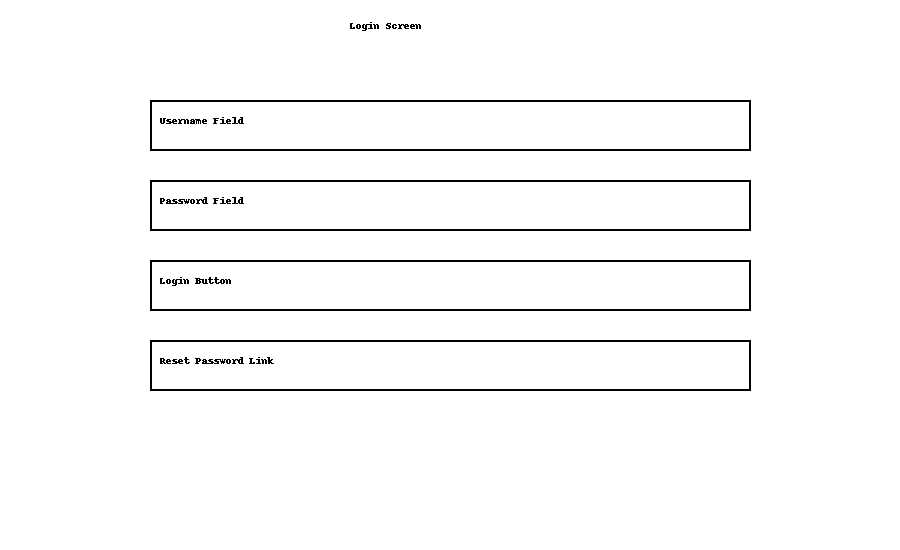
### ****3. Design Considerations****

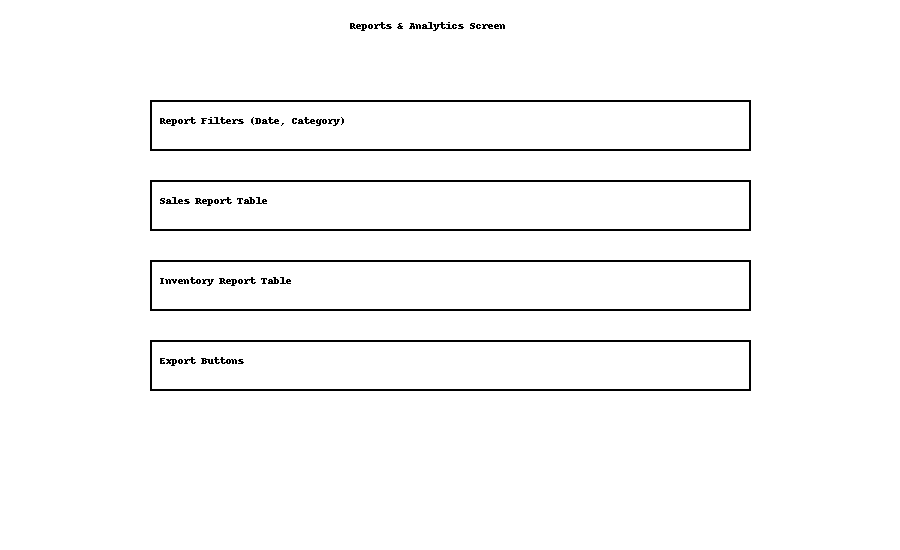
**Consistency:** Each screen will use the same color scheme, typography, and button styles to provide a unified experience.

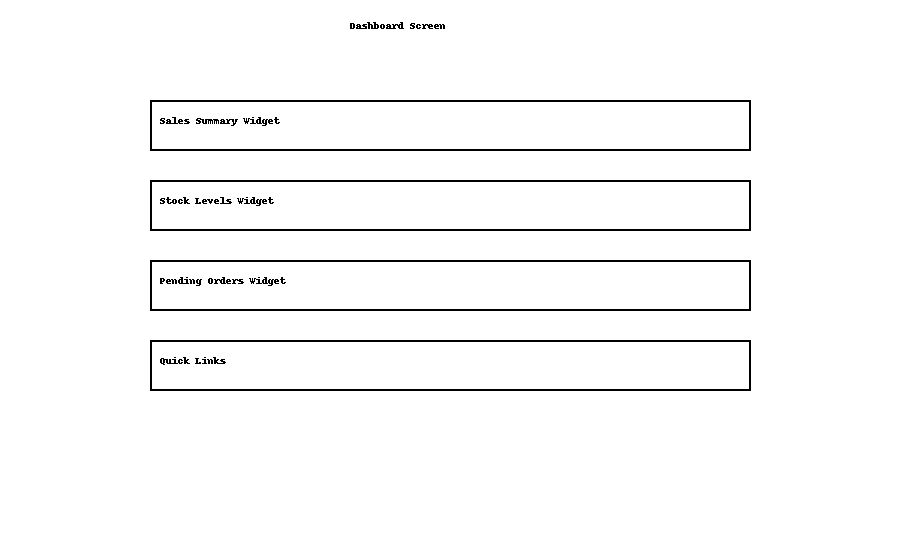
**Navigation:** A sidebar or top navigation bar will be included for quick access to different modules.

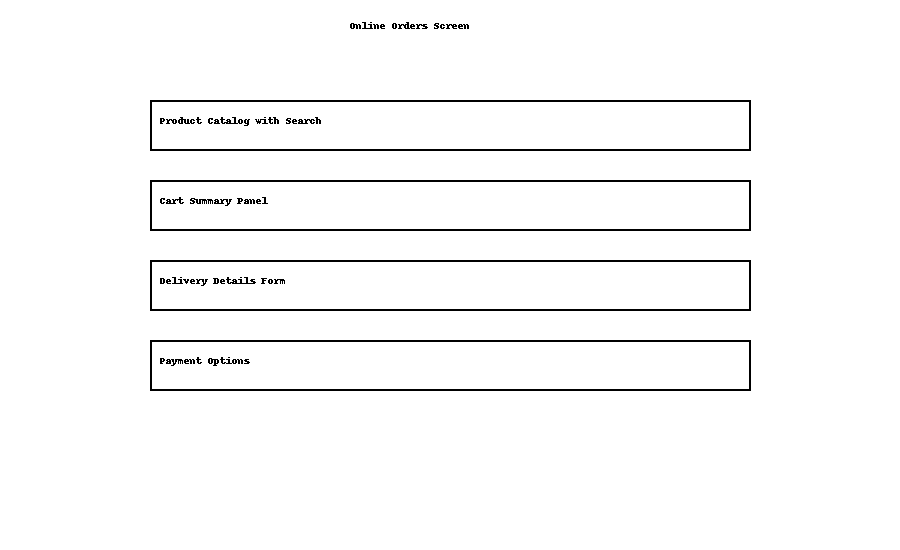
**Validation & Feedback:** Forms will have field validation (e.g., required fields, correct data formats). Users will receive confirmation or error messages after actions.

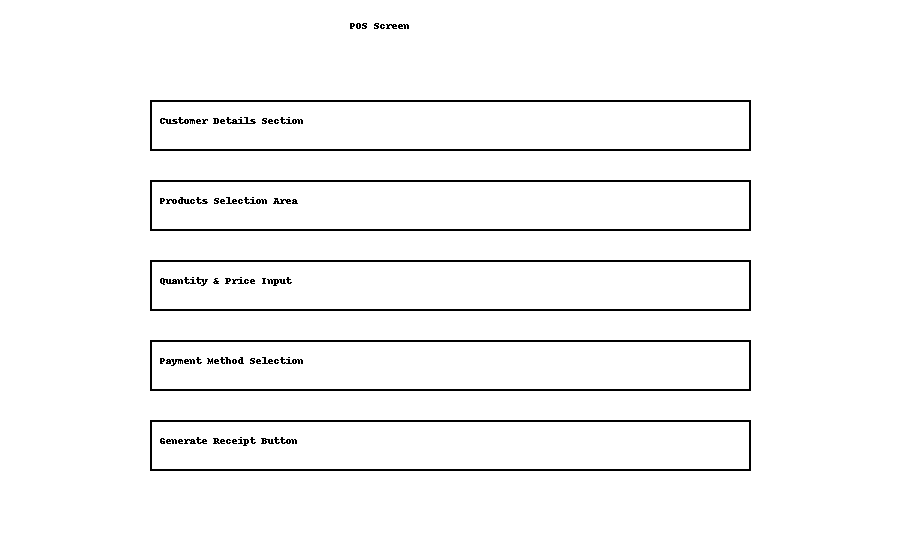
**Responsive Design:** Interfaces will be optimized for desktop and mobile devices.

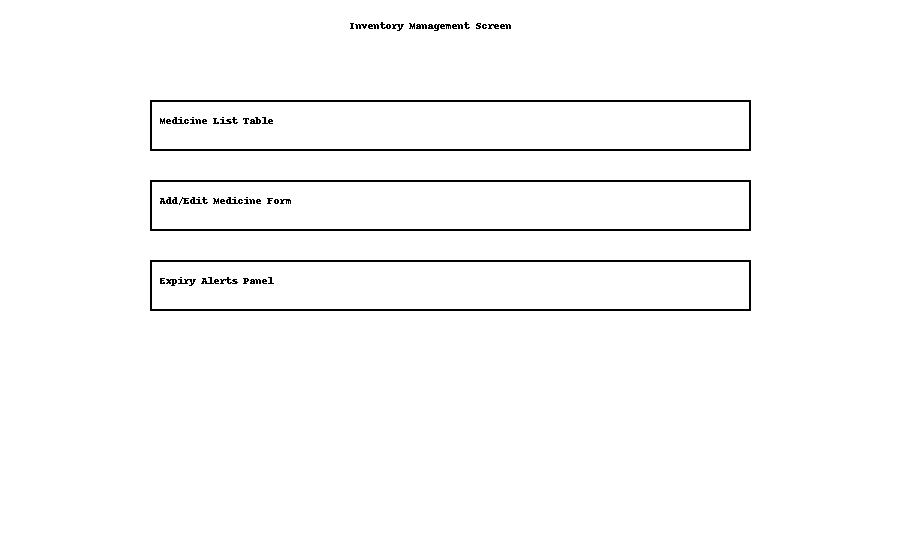




****

****





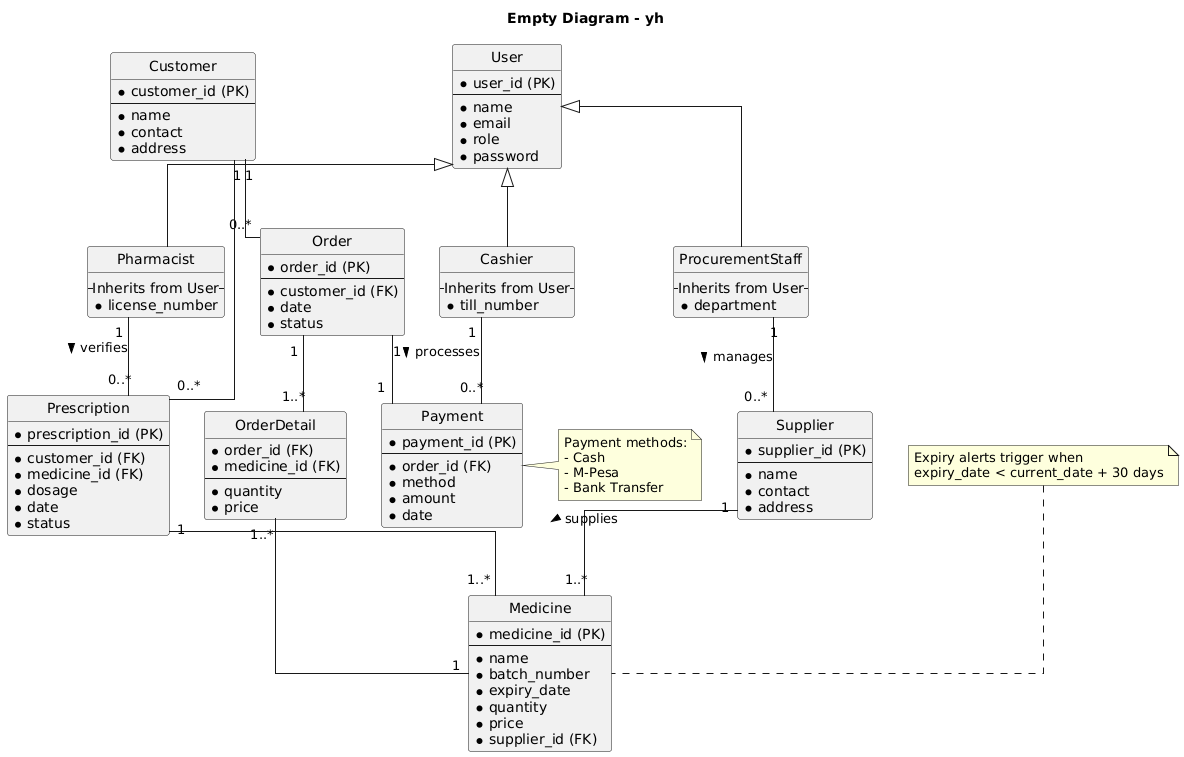
**DATABASE DESIGN**

The database design defines how data is structured and organized to ensure secure storage, efficient access, and consistent management. For the Pharmacy Management System, a relational database model was implemented because of its reliability, support for structured queries (SQL), and ability to enforce data relationships effectively.

The database is built around key entities such as users, customers, medicines, prescriptions, orders, suppliers, and payments. Each entity is represented as a table with specific fields designed to handle the data it manages.

To maintain data integrity and reduce redundancy, the tables were normalized and designed with primary and foreign key constraints. Relationships between tables (e.g., one-to-many or many-to-one) were established to support efficient querying, accurate reporting, and secure access control across the system.

The Database Design defines how data will be stored, related, and retrieved in the Pharmacy Management System. It ensures data integrity, avoids redundancy, and supports efficient queries.

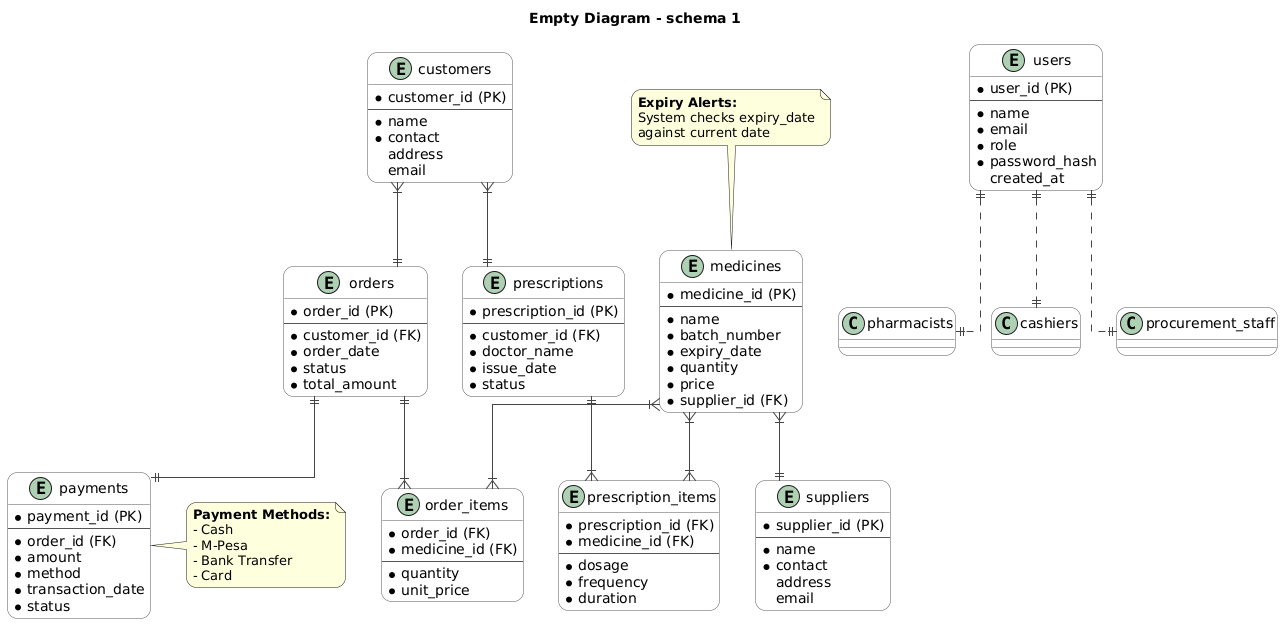


The Entity Relationship Diagram (ERD) illustrates the key entities and relationships that form the foundation of the Pharmacy Management System’s database.

**Entity Relationships**

* A **User** (Admin, Pharmacist, Cashier, Procurement) can perform multiple actions such as managing inventory, processing sales, or verifying prescriptions.
* A **Customer** can place multiple **Orders**, but each order belongs to a single customer.
* Each **Order** can have multiple **OrderDetails**, and each order detail refers to a specific **Medicine**.
* A **Supplier** can supply multiple **Medicines**, but each medicine is linked to one supplier.
* A **Prescription** belongs to one **Customer** and can include multiple **Medicines**.
* Each **Payment** is linked to a single **Order**, but an order must have at least one payment record.
* **Medicines** are updated in real-time through sales, prescriptions, and procurement processes, ensuring accurate stock levels.

**Schema diagram**

****

The Pharmacy Management System schema is designed to efficiently manage customers, prescriptions, medicines, suppliers, orders, payments, and user roles. It captures detailed customer and prescription information, links prescriptions and orders to specific medicines, and maintains inventory with supplier details. A built-in expiry alert system ensures medicines are safe for use by checking expiry dates. The system supports multiple payment methods (Cash, M-Pesa, Bank Transfer, Card) and logs transactions for each order. Role-based access control is implemented to manage different user types, including pharmacists, cashiers, and procurement staff. Overall, the schema supports streamlined pharmacy operations, inventory control, secure user management, and reliable transaction tracking.