Pharmacy Inventory Control System – Documentation

1. Project planning & management

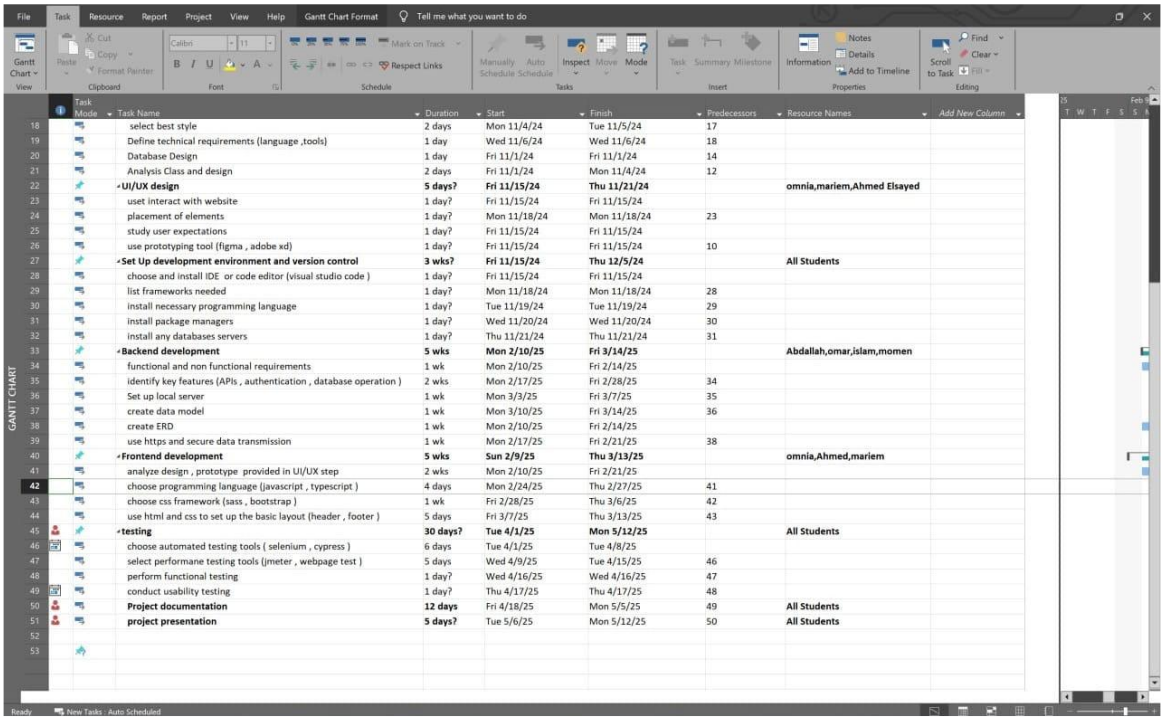
Project proposal

The "Pharmacy Management System (PMS)" aims to enhance the efficiency of pharmacy operations by automating inventory management, sales tracking, and expiration date monitoring. The system provides advanced technological solutions that reduce human errors, offer real-time data on stock levels, and improve customer experience through faster and more accurate transactions.

Project plan

The project follows a precise timeline that includes several key phases:

- System Analysis : Studying requirements and designing the initial system structure.
- Database Design: Developing a data structure to ensure efficient storage and fast data retrieval.
- Frontend & Backend Development: Creating the user interface and integrating it with the database and servers.
- Testing & Quality Assurance: Verifying performance, fixing bugs, and ensuring seamless user experience.
- Deployment & Evaluation: Launching the system and collecting user feedback for performance improvement.



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
18		select best style	2 days	Mon 11/4/24	Tue 11/5/24	17	
19		Define technical requirements (language ,tools)	1 day	Wed 11/6/24	Wed 11/6/24	18	
20		Database Design	1 day	Fri 11/1/24	Fri 11/1/24	14	
21		Analysis Class and design	2 days	Fri 11/1/24	Mon 11/4/24	12	
22		**UI/UX design**	**5 days?**	**Fri 11/15/24**	**Thu 11/21/24**		**omnia,mariem,Ahmed Elsayed**
23		uset interact with website	1 day?	Fri 11/15/24	Fri 11/15/24		
24		placement of elements	1 day?	Mon 11/18/24	Mon 11/18/24	23	
25		study user expectations	1 day?	Fri 11/15/24	Fri 11/15/24		
26		use prototyping tool (figma , adobe xd)	1 day?	Fri 11/15/24	Fri 11/15/24	10	
27		**Set Up development environment and version control**	**3 wks?**	**Fri 11/15/24**	**Thu 12/5/24**		**All Students**
28		choose and install IDE or code editor (visual studio code)	1 day?	Fri 11/15/24	Fri 11/15/24		
29		list frameworks needed	1 day?	Mon 11/18/24	Mon 11/18/24	28	
30		install necessary programming language	1 day?	Tue 11/19/24	Tue 11/19/24	29	
31		install package managers	1 day?	Wed 11/20/24	Wed 11/20/24	30	
32		install any databases servers	1 day?	Thu 11/21/24	Thu 11/21/24	31	
33		**Backend development**	**5 wks**	**Mon 2/10/25**	**Fri 3/14/25**		**Abdallah,omar,islam,momen**
34		functional and non functional requirements	1 wk	Mon 2/10/25	Fri 2/14/25		
35		identify key features (APIs , authentication , database operation)	2 wks	Mon 2/17/25	Fri 2/28/25	34	
36		Set up local server	1 wk	Mon 3/3/25	Fri 3/7/25	35	
37		create data model	1 wk	Mon 3/10/25	Fri 3/14/25	36	
38		create ERD	1 wk	Mon 2/10/25	Fri 2/14/25		
39		use https and secure data transmission	1 wk	Mon 2/17/25	Fri 2/21/25	38	
40		**Frontend development**	**5 wks**	**Sun 2/9/25**	**Thu 3/13/25**		**omnia,Ahmed,mariem**
41		analyze design , prototype provided in UI/UX step	2 wks	Mon 2/10/25	Fri 2/21/25		
42		choose programming language (javascript , typescript)	4 days	Mon 2/24/25	Thu 2/27/25	41	
43		choose css framework (sass , bootstrap)	1 wk	Fri 2/28/25	Thu 3/6/25	42	
44		use html and css to set up the basic layout (header , footer)	5 days	Fri 3/7/25	Thu 3/13/25	43	
45		**testing**	**30 days?**	**Tue 4/1/25**	**Mon 5/12/25**		**All Students**
46		choose automated testing tools (selenium , cypress)	6 days	Tue 4/1/25	Tue 4/8/25		
47		select performane testing tools (jmeter , webpage test)	5 days	Wed 4/9/25	Tue 4/15/25	46	
48		perform functional testing	1 day?	Wed 4/16/25	Wed 4/16/25	47	
49		conduct usability testing	1 day?	Thu 4/17/25	Thu 4/17/25	48	
50		**Project documentation**	**12 days**	Fri 4/18/25	Mon 5/5/25	49	**All Students**
51		**project presentation**	**5 days?**	Tue 5/6/25	Mon 5/12/25	50	**All Students**
52							
53							

Task Assignment & Roles

Lead Developer

- Responsible for designing and implementing the core system functionalities.
- Develops the backend services and APIs to ensure smooth data flow.
- Oversees system integration, ensuring seamless communication between different components.
- Troubleshoots technical issues and optimizes system performance.

Database Administrator (DBA)

- Manages the database structure, ensuring efficient data storage and retrieval.
- Implement security measures to protect sensitive pharmacy data.
- Monitors database performance and optimizes queries for faster response times.
- Performs regular backups and recovery processes to prevent data loss.

System Tester (Quality Assurance Engineer)

- Conducts comprehensive testing to identify and fix bugs before deployment.
- Performs functional, performance, and security testing to ensure system stability.
- Simulates real-world scenarios to evaluate system behavior under various conditions.
- Documents test cases, reports issues, and verify that all features meet the required standards.

UI/UX Designer

- Develops an intuitive and visually appealing user interface.
- Ensures a seamless user experience by focusing on accessibility and usability.
- Conducts user research and gathers feedback to refine the design.
- Collaborates with developers to ensure design feasibility and implementation consistency.

Risk Assessment & Mitigation plan

Potential Risks	Impact	Mitigation Measures
Data loss or errors	High	Implement automated backups and logging systems.
System integration issues between frontend and backend	Medium	Conduct regular integration testing and code reviews.
Slow system response	Low	Optimize database queries, improve caching techniques.

Key Performance Indicators (KPIs)

- **Response Time :** The system ensures that operations and data loading are completed in less than two seconds, providing a fast and seamless user experience.

- **System Uptime :** Achieves 99.9% operational reliability, ensuring stable performance without any disruptions that may hinder workflow.

- **User Satisfaction Rate :** Aims to exceed 90% user satisfaction, by offering a flexible interface and high-quality performance that meets user expectations.

2. Literature Review

Feedback & Evaluation

System Architecture

This criterion assesses the efficiency of the system's structure, ensuring a well-organized and scalable design. The system should allow seamless integration between the database and the user interface, enabling future modifications and expansions without compromising performance.

Performance

This focuses on the system's ability to process inventory and sales data efficiently without slowdowns or delays. The system's response time for searching medicines, updating stock, and generating reports is tested. Additionally, stability is assessed when multiple users access the system simultaneously.

User Experience (UX)

This criterion evaluates the ease of use for pharmacy staff, ensuring the interface is clear and intuitive. The system should allow for quick completion of daily tasks without requiring extensive training, thereby improving workflow efficiency within the pharmacy.

Suggested Improvement

- **Smart Notifications System :** Sends alerts about nearly expired drugs or low stock.
- **Billing System Integration :** Supports electronic payments and digital invoices.
- **Security Enhancements :** Restricting access to sensitive data, logging operations, encrypting inventory and sales, and securing login for authorized users only.
- **Advanced Data Analysis :** Utilizing historical data reports to identify sales trends and optimize inventory management based on expected demand.

Final grading criteria

Criterion	Evaluation Percentage (%)	Description
Documentation	20%	Quality of documentation and diagrams.
Implementation	30%	Achievement of required functions and code efficiency.
Testing	25%	Number of detected bugs and system efficiency after fixes.
Presentation	15%	Clarity of idea, delivery style, and interaction with questions.
User Experience	10%	Ease of use and user satisfaction.

3. Requirements Gathering

Stakeholder Analysis

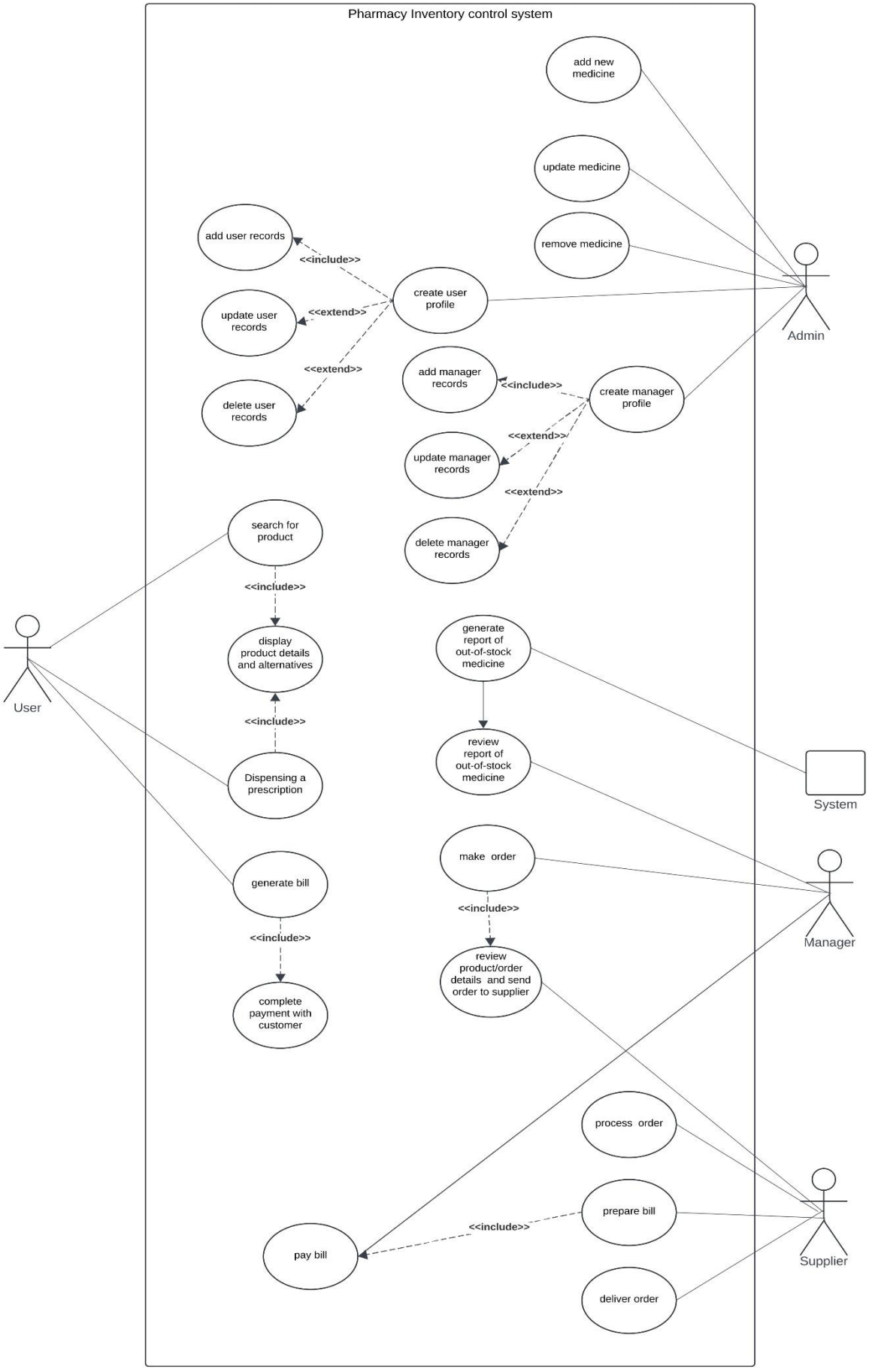
The key stakeholders in the Pharmacy Inventory Control System are:

- **Admin**: Manages profiles and data.
- **Pharmacist**: places orders and generate bill.
- **Manager**: reviews reports and sends required order to suppliers.
- **Supplier**: Supplies medicines based on pharmacy orders.
- **Customers**: Purchase medicines and complete payments.

User Stories & Use Cases

The Pharmacy Inventory Control System is designed to efficiently manage and track medicine stock, ensure availability, and prevent shortages or expired products. The system enables manager to monitor inventory levels, process sales, generate reports, and reorder stock as needed.

This system integrates multiple functionalities such as inventory management, order processing, stock tracking, report generation, and supplier coordination, ensuring seamless pharmacy operations. The main goal is to reduce human error, optimize stock levels, and enhance customer service by ensuring medicines are always available.


Pharmacy Inventory control system
add new medicine
update medicine
remove medicine
add user records
<<include>>
create user profile
update user records
<<extend>>
<<extend>>
delete user records
Admin
add manager records
<<include>>
create manager profile
<<extend>>
update manager records
<<extend>>
delete manager records
search for product
<<include>>
display product details and alternatives
<<include>>
Dispensing a prescription
generate bill
<<include>>
complete payment with customer
User
generate report of out-of-stock medicine
review report of out-of-stock medicine
System
make order
<<include>>
review product/order details and send order to supplier
Manager
process order
prepare bill
<<include>>
pay bill
deliver order
Supplier

Use case scenarios

Use Case: Add New Medicine

1. Admin logs in the system by entering username and password.
2. Admin browse through options.
3. Admin selects "Add New Medicine" option.
4. System prompts for medicine details (name, quantity, expiry date, etc.).
5. Admin enters the required details.
6. System validates and saves the new medicine information.
7. confirms the addition of the new medicine.

Use Case: Update Medicine

1. Admin logs in the system by entering username and password.
2. Admin browse through options.
3. Admin selects "Update Medicine" option.
4. System prompts for editing medicine details (name, quantity, expiry date, etc.).
5. Admin enters the required details.
6. System validates and saves the new medicine information.
7. confirms the modification of the medicine.

Use Case: Remove Medicine

1. Admin logs in to the system by entering username and password.
2. Admin browses through available options.
3. Admin selects the "Remove Medicine" option.
4. System prompts for medicine details (name, quantity, reason for removal, etc.).
5. Admin enters the required details.
6. System validates and updates the inventory by removing the specified medicine.

7. System confirms the removal of the medicine.

Use Case: Create Pharmacist Profiles

1. Admin logs in the system by entering username and password.
2. Admin browse through options
3. Admin selects "**Create Pharmacist Profiles**" option.
4. System prompts for pharmacist details (name, username, password, etc.).
5. Admin enters the required details.
6. System validates and saves the new pharmacist profile.
7. System confirms the creation of the new pharmacist profile.

Use Case: Update Pharmacist Profiles

1. Admin logs in the system by entering username and password.
2. Admin browse through options
3. Admin selects "**Pharmacist Profiles**" option.
4. Admin select the desired profile to update.
5. Admin selects the details required to edit.
6. System validates and saves the pharmacist profile.
7. System confirms the updating of the pharmacist profile.

Use Case: Delete Pharmacist Profiles

1. Admin logs in the system by entering username and password.
2. Admin browse through options
3. Admin selects "**Pharmacist Profiles**" option.
4. Admin select the desired profile to delete.
5. System validates and delete the pharmacist profile.

Use Case: Create Manager Profiles

1. Admin logs in the system by entering username and password.
2. Admin browse through options
3. Admin selects "**Create Manager Profiles**" option.
4. System prompts for manager details (name, username, password, etc.).
5. Admin enters the required details.
6. System validates and saves the new manager profile.
7. System confirms the creation of the new manager profile.

Use Case: Update Manager Profiles

1. Admin logs in the system by entering username and password.
2. Admin browse through options
3. Admin selects " **Manager Profiles**" option.
4. Admin select the desired profile to update.
5. Admin selects the details required to edit.
6. System validates and saves the manager profile.
7. System confirms the updating of the manager profile.

Use Case: Delete Manager Profiles

1. Admin logs in the system by entering username and password.
2. Admin browse through options
3. Admin selects " **Manager Profiles**" option.
4. Admin select the desired profile to delete.
5. System validates and delete the manager profile.

Use Case: Search for Medicine

1. Pharmacist logs in the system by entering username and password.
2. Pharmacist browse through options.
3. Pharmacist selects "Search for Product" option
4. System prompts for medicine name.
5. Pharmacist enters the product name.

6. System displays search results.
7. Pharmacist selects the desired product.
8. The pharmacist finds the product they were looking for.

Alternate Flow: No Results Found

- **Condition**: If no products match the search.

1. System shows a message saying "No results found".
2. Pharmacist can try searching again with a different name or alternative product.

Use Case: Dispense Prescription

1. Pharmacist logs in to the system by entering username and password.
2. Pharmacist browses through options.
3. Pharmacist selects "**Dispense Prescription**" option.
4. System prompts for prescription details.
5. Pharmacist enters the prescription details or scans an electronic prescription.
6. System verifies the prescription and checks for drug interactions.
7. Pharmacist retrieves the prescribed medication from inventory.
8. Pharmacist prepares the medication, ensuring correct dosage and packaging.
9. Pharmacist provides usage instructions and precautions to the patient.
10. System updates inventory and records the transaction.
11. Pharmacist hands over the medication to the patient.

Use Case: Generate bill (Pharmacist)

1. Pharmacist logs in the system by entering username and password.
2. Pharmacist browse through options.
3. Pharmacist selects "**Generate bill** " option.
4. System prompts for customer and product details.
5. Pharmacist enters the required details.

6. System calculates the total amount.
7. System generates the bill.
8. Pharmacist provides the bill to the customer.

Use Case: Make Order

1. Manager logs in the system by entering username and password.
2. Manager browse through options
3. Manager selects "**Review report of out-of-stock medicine**" option.
4. Manager chooses among suppliers to order the out-of-stock medicine .
5. System validates and confirms order.

Use Case: Generate Report of Out-of-Stock Medicine

1. Operator logs in the system by entering username and password.
2. Operator browse through options.
3. Operator selects "Generate Report of Out-of-Stock Medicine" option.
4. System retrieves the list of out-of-stock medicines.
5. System generates the report.
6. Operator reviews the report.

Use Case: Process Order

1. Manager sends request for required out-of-stock items from supplier.
2. Supplier reviews the order request and waits for the purchase order confirmation.
3. Operator confirms the list and purchase order.
4. Supplier prepares the order based on the request and received list.
5. Supplier confirms the order preparation.

Functional Requirements

- Add, update, and remove medicines.
- Process orders and manage inventory stock.
- Generate bills and complete payments.
- Track and generate reports for out-of-stock medicines.
- Manage user and supplier records.

Non-Functional Requirements

- **Performance**: The system should handle multiple transactions without delays.
- **Security**: Role-based access control to prevent unauthorized actions.
- **Usability**: Simple and intuitive UI for pharmacists and customers.
- **Reliability**: Ensure correct stock updates and prevent inventory mismatches.

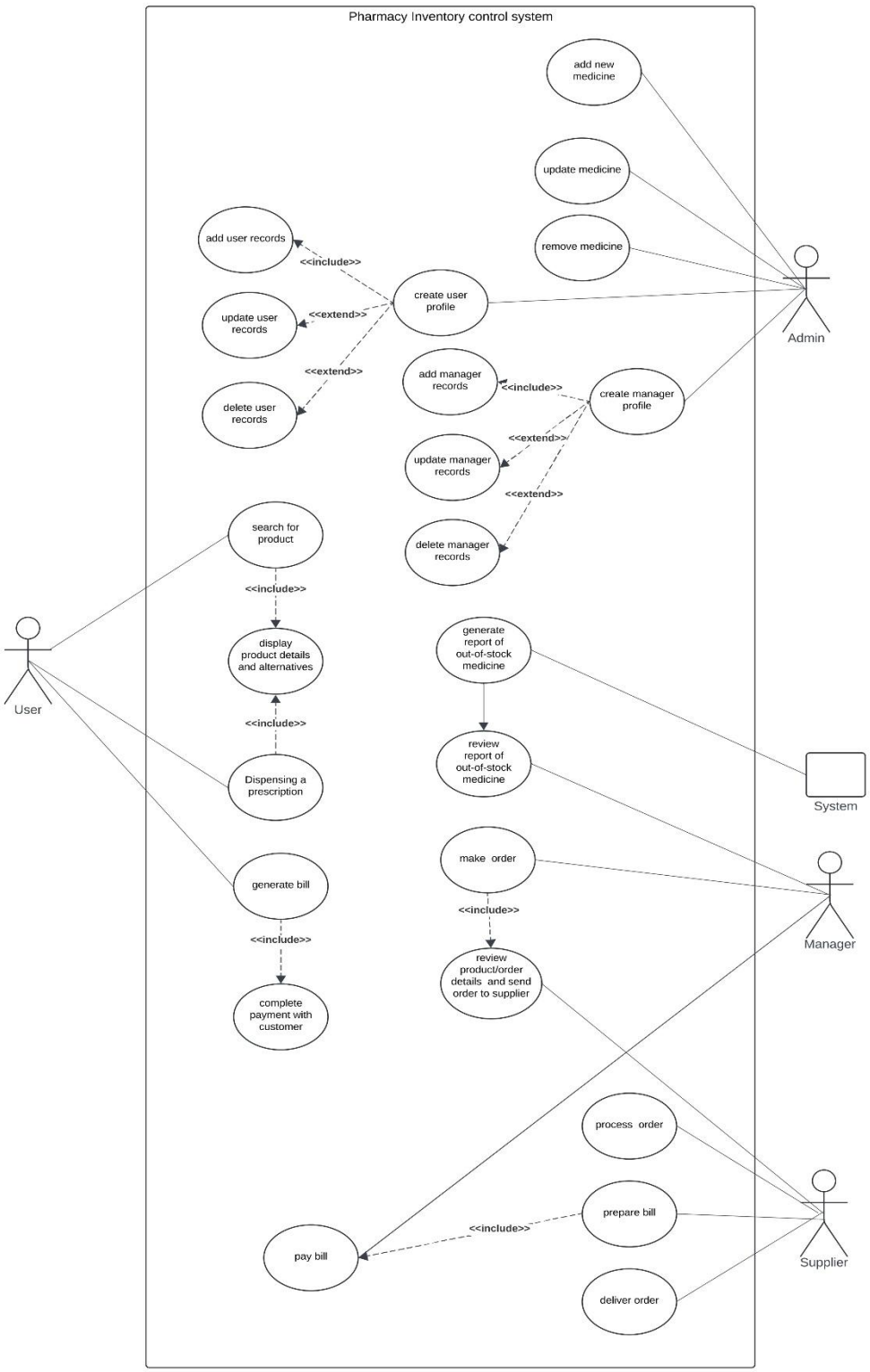
---

4. System Analysis & Design

1. Problem Statement & Objectives

The **Pharmacy Inventory Control System** aims to **streamline inventory management**, prevent stock shortages, and **enhance operational efficiency** by automating tasks such as order processing, medicine tracking, and billing.

- **Use Case Diagram**:

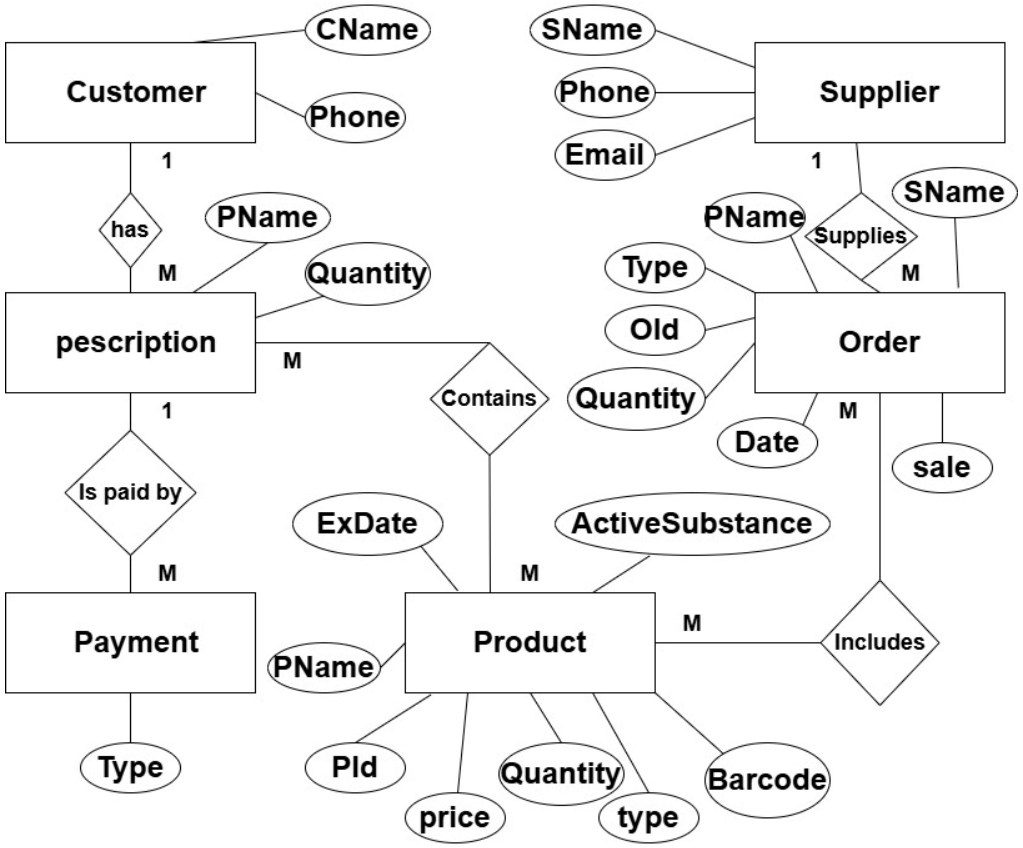


- Order Processing
- Inventory Management
- Bill generation
- Supplier Management

---

2. Database Design & Data Modeling

- **Entity-Relationship Diagram (ERD)**:



Entity-Relationship Diagram (ERD) Description – Pharmacy Inventory Control System

This **ERD (Entity-Relationship Diagram)** represents the structure and relationships of entities within the **Pharmacy Inventory Control System**, showing how different entities interact with each other.

Entities and Their Attributes:

1. Customer

- **Attributes:**
    - CName (Customer Name)
    - Phone (Contact number)
- **Relationships:**
    - A customer **has** one or more **prescriptions** (1:M).

---

2. Prescription

- **Attributes:**
    - PName (Product/Medicine Name)
    - Quantity (Number of medicines prescribed)
- **Relationships:**
    - A prescription **contains** multiple **products** (M:M).
    - A prescription **is paid by** multiple **payments** (M:1).

---

3. Payment

- **Attributes:**
    - Type (Payment method: cash, card, etc.)
- **Relationships:**
    - A **prescription** can have multiple **payments** (M:1).

---

4. Product

- **Attributes:**
    - PName (Product Name)
    - PId (Product ID)
    - Price
    - Quantity
    - Type (Medicine type)
    - Barcode (Unique identification code)
    - ActiveSubstance (Main ingredient)
    - ExDate (Expiration Date)
- **Relationships:**
    - A product is **included in** multiple **orders** (M:M).

---

5. Order

- **Attributes:**
    - PName (Product Name)
    - Type
    - Old (Previous stock level)
    - Quantity
    - Date (Order date)
- **Relationships:**
    - An **order includes** multiple **products** (M:M).
    - A **supplier supplies** multiple **orders** (1:M).

---

6. Supplier

- **Attributes:**

    - SName (Supplier Name)
    - Phone
    - Email
- **Relationships:**
    - A **supplier supplies** multiple **orders** (1:M).

Relationships Summary:

1. A **Customer** can have multiple **Prescriptions**.
2. A **Prescription** contains multiple **Products** and is paid for using multiple **Payments**.
3. A **Product** can be part of multiple **Orders** and has details like **Expiration Date, Barcode, and Active Substance**.
4. A **Supplier** provides multiple **Orders**, which contain multiple **Products**.

Key Insights from the ERD:

- **Efficient Stock Management:** The system tracks product quantities, expiration dates, and active substances.
- **Order Processing:** The supplier-relationship ensures automated ordering and tracking.
- **Customer Transactions:** Each customer prescription is linked to payments for a streamlined purchase process.

- **Logical & Physical Schema**:

The **logical schema** represents the high-level structure of the database in a conceptual manner without focusing on implementation details like storage, indexes, or data types.

Entities and Relationships in the Logical Schema:

1. **Customer (CName, Phone)**
    - A customer can have multiple prescriptions (1:M).
2. **Prescription (PName, Quantity, CustomerID (FK))**
    - A prescription belongs to one customer but can include multiple products (M:M).
3. **Payment (PaymentID, Type, PrescriptionID (FK))**
    - A payment is associated with one or more prescriptions (M:1).
4. **Product (PId, PName, Price, Quantity, Type, Barcode, ActiveSubstance, ExDate)**
    - A product is included in multiple prescriptions and orders (M:M).
5. **Order (OrderID, PName, Type, OldQuantity, Quantity, Date, SupplierID (FK))**
    - An order can include multiple products and is linked to a supplier (M:M).
6. **Supplier (SupplierID, SName, Phone, Email)**
    - A supplier supplies multiple orders (1:M).

Logical Schema Relationships:

- **Customer → Prescription (1:M)**
- **Prescription → Product (M:M)**
- **Prescription → Payment (M:1)**
- **Product → Order (M:M)**
- **Order → Supplier (M:1)**

Physical Schema

The **Pharmacy Inventory Control System** is built using a **relational database model**, ensuring data integrity and efficient management of pharmaceutical stock. The database is implemented using **SQL**, where I designed and structured the schema to support core functionalities such as inventory tracking, supplier management, and sales records. Tables are properly normalized to optimize performance and reduce redundancy, while constraints such as primary keys, foreign keys, and indexes enhance data consistency and retrieval speed. The SQL queries handle data insertion, updates, retrieval, and reporting to maintain accurate inventory control.

1. Billing Table (BillingTbl)

Stores details of sales transactions, including the seller and the transaction amount.

```
CREATE TABLE [dbo].[BillingTbl] (
   [Billid]   INT  IDENTITY (1, 1) NOT NULL,
   [BillDate] DATE NOT NULL,
   [Seller]   INT  NOT NULL,
   [Amount]   INT  NOT NULL,
   PRIMARY KEY CLUSTERED ([Billid] ASC),
   CONSTRAINT [FKs] FOREIGN KEY ([Seller]) REFERENCES [dbo].[SellerTbl] ([Sellid])
);
```

---

2. Category Table (CategoryTbl)

Stores different medicine categories, ensuring unique category names.

```
CREATE TABLE [dbo].[CategoryTbl] (
   [CatId]   INT        IDENTITY (1000, 1) NOT NULL,
   [CatName] VARCHAR (50) NOT NULL,
   PRIMARY KEY CLUSTERED ([CatId] ASC),
   CONSTRAINT [UQ_CategoryTbl_CatName] UNIQUE NONCLUSTERED ([CatName] ASC)
);
```

---

3. Manager Table (ManagerTbl)

Contains information about managers, including login credentials and personal details.

```
CREATE TABLE [dbo].[ManagerTbl] (
    [Mid]    INT          IDENTITY (1, 1) NOT NULL,
    [MName]  VARCHAR (50)  NOT NULL,
    [MEmail] VARCHAR (50)  NOT NULL,
    [MPass]  VARCHAR (50)  NOT NULL,
    [MDOB]   DATE         NOT NULL,
    [MGen]   VARCHAR (10)  NOT NULL,
    [MAdd]   VARCHAR (150) NOT NULL,
    PRIMARY KEY CLUSTERED ([Mid] ASC)
);
```

---

4. Medicine Table (MedicineTbl)

Stores medicine details, including pricing, stock levels, expiration date, and category.

```
CREATE TABLE [dbo].[MedicineTbl] (
    [MedCode]     VARCHAR (50) NOT NULL,
    [MedName]     VARCHAR (50) NOT NULL,
    [MedPrice]    INT         NOT NULL,
    [MedStock]    INT         NOT NULL,
    [MedExpDate]  DATE        NOT NULL,
    [MedCategory] INT         NOT NULL,
    [MID]         INT         IDENTITY (1, 1) NOT NULL,
    PRIMARY KEY CLUSTERED ([MID] ASC),
```

```
    CONSTRAINT [FK3] FOREIGN KEY ([MedCategory]) REFERENCES [dbo].[CategoryTbl] ([CatId])
);
```

---

5. Seller Table (SellerTbl)

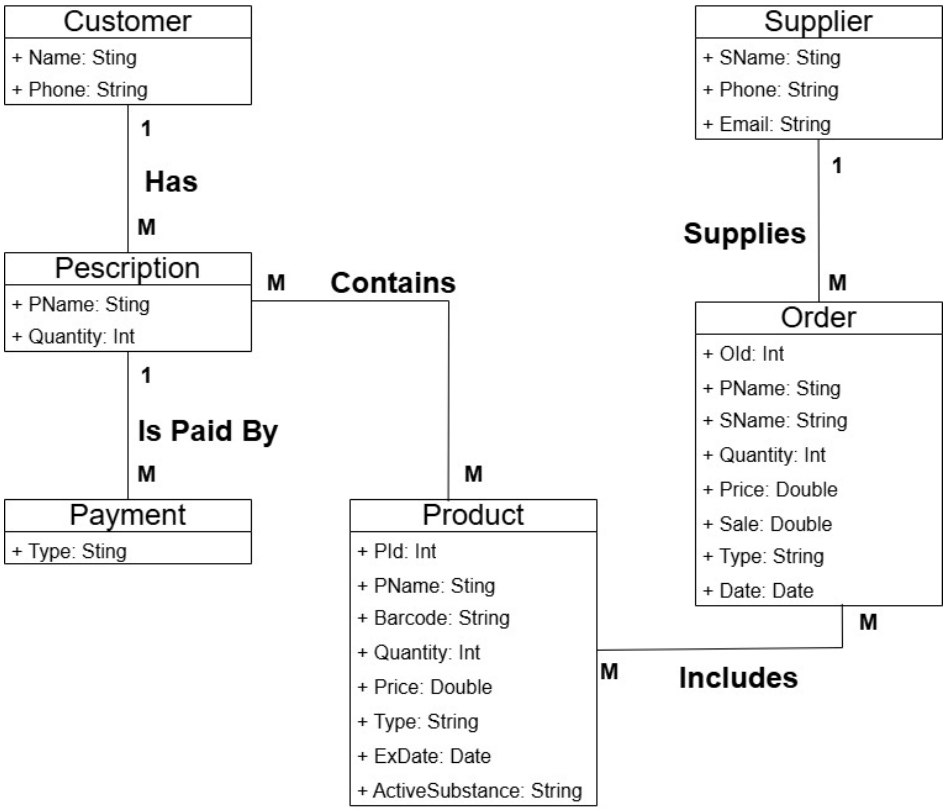
Contains seller information, including personal details and login credentials.

```
CREATE TABLE [dbo].[SellerTbl] (
    [Sellid]    INT          IDENTITY (1, 1) NOT NULL,
    [SellName]  VARCHAR (50)  NOT NULL,
    [SellEmail] VARCHAR (50)  NOT NULL,
    [SellPass]  VARCHAR (50)  NOT NULL,
    [SellDOB]   DATE          NOT NULL,
    [SellGen]   VARCHAR (10)  NOT NULL,
    [SellAdd]   VARCHAR (150) NOT NULL,
    PRIMARY KEY CLUSTERED ([Sellid] ASC)
);
```

---

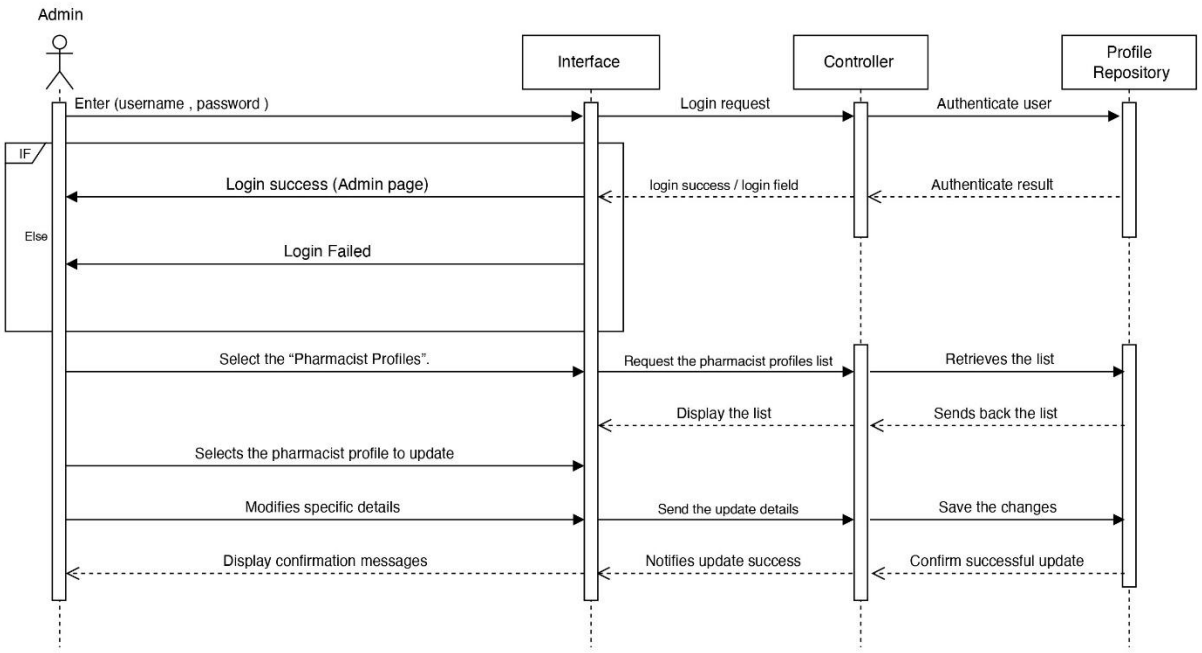
3. Data Flow & System Behavior

- **Class Diagram**:
    - Classes: Customer , Prescription , Payment , Product , Order, Supplier.
    - Attributes: Customer (Name, Phone), Prescription (PName , Quantity), Payment(Type),Product (PID,PName,Barcode,Quantity,Price,Type,ExDate,ActiveSubstance),Order (OID,PName,SName,Quantity,Price,Sale,Type,Date), Supplier(SName,Phone,Email).

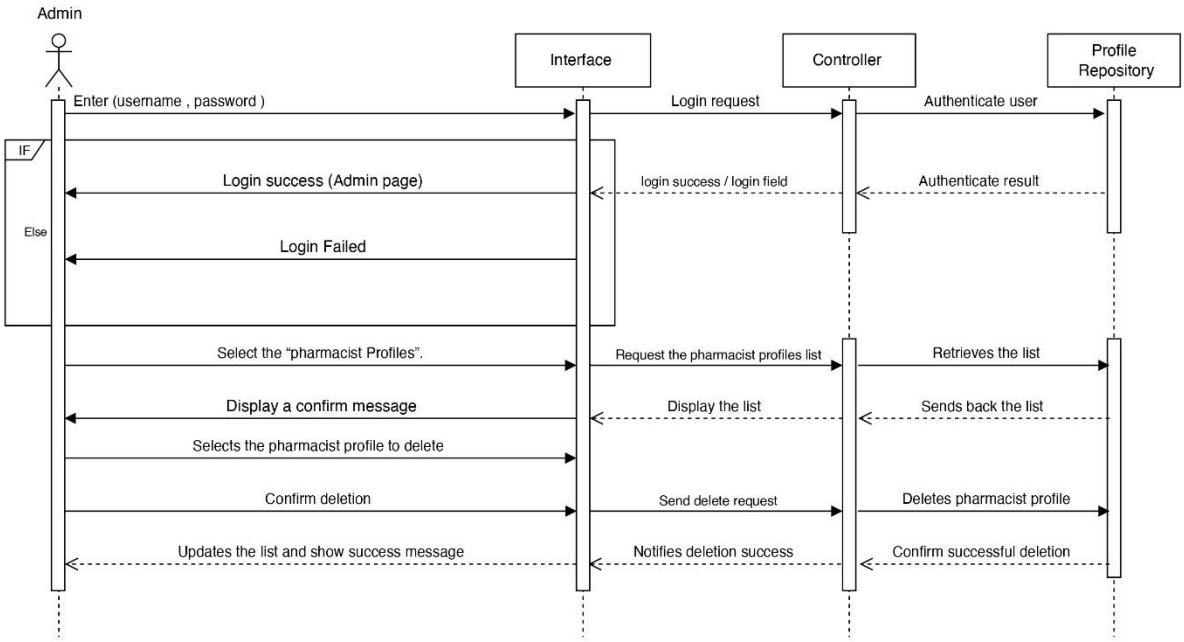


- **Sequence Diagrams**:
    - Customer searches for a product → System fetches details → Customer proceeds to payment.

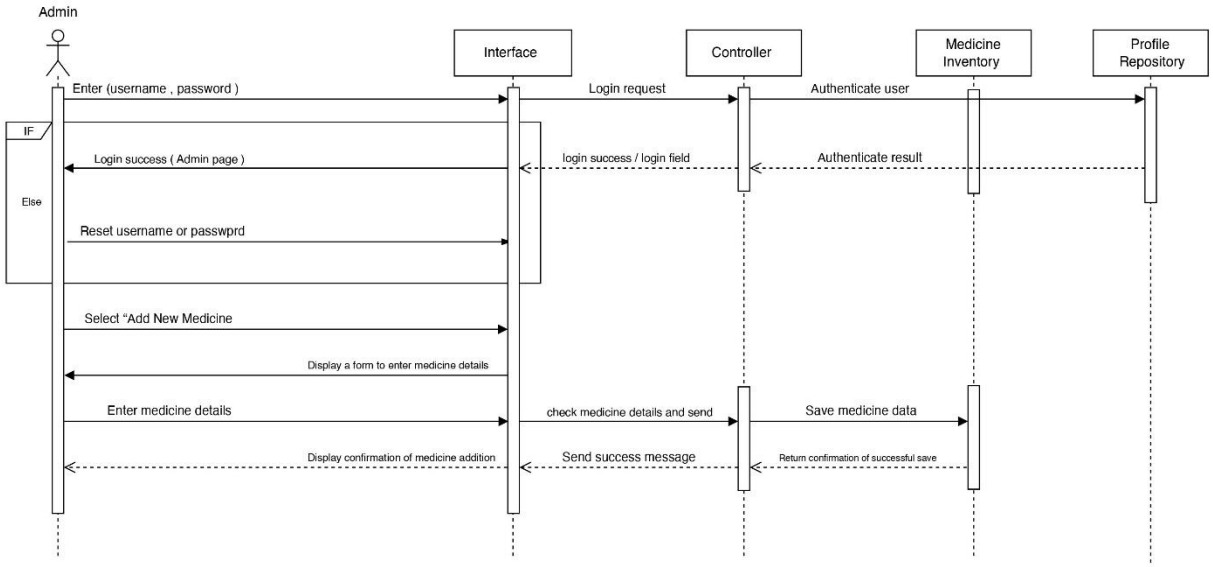
Update Pharmacist Profiles



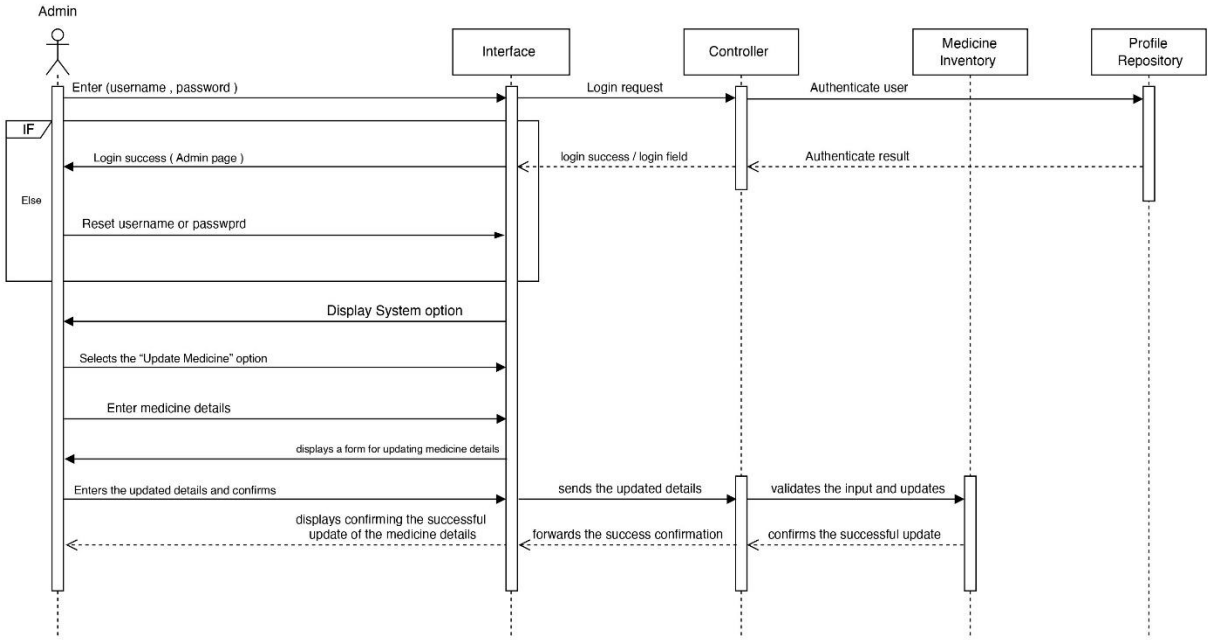
Delete Pharmacist Profiles



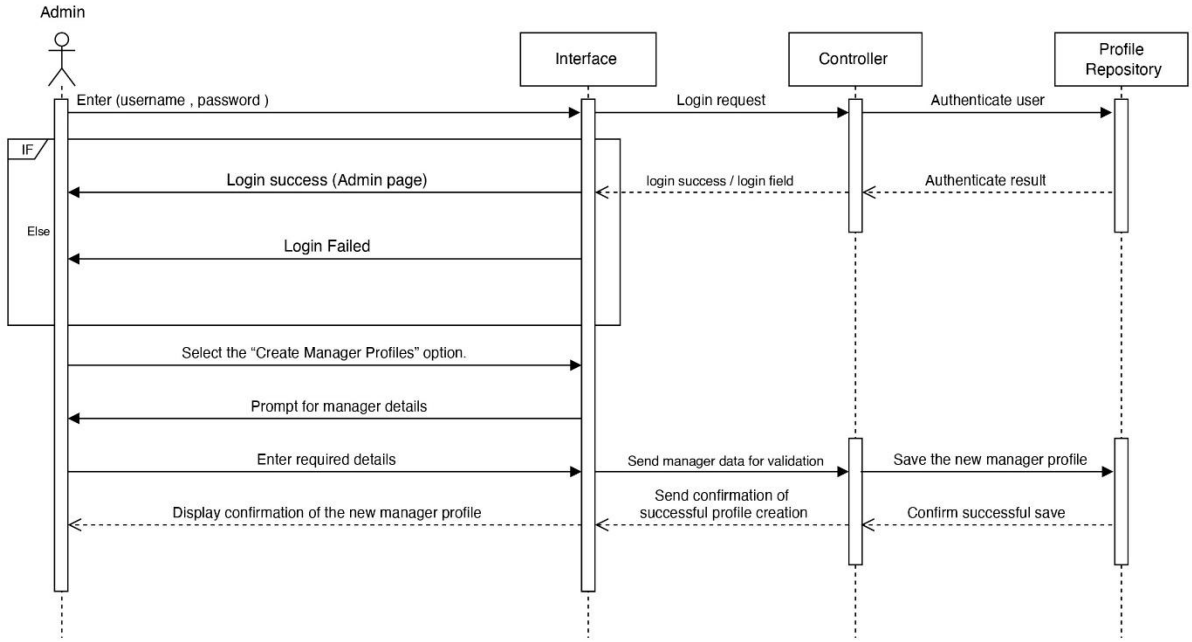
Add New Medicine



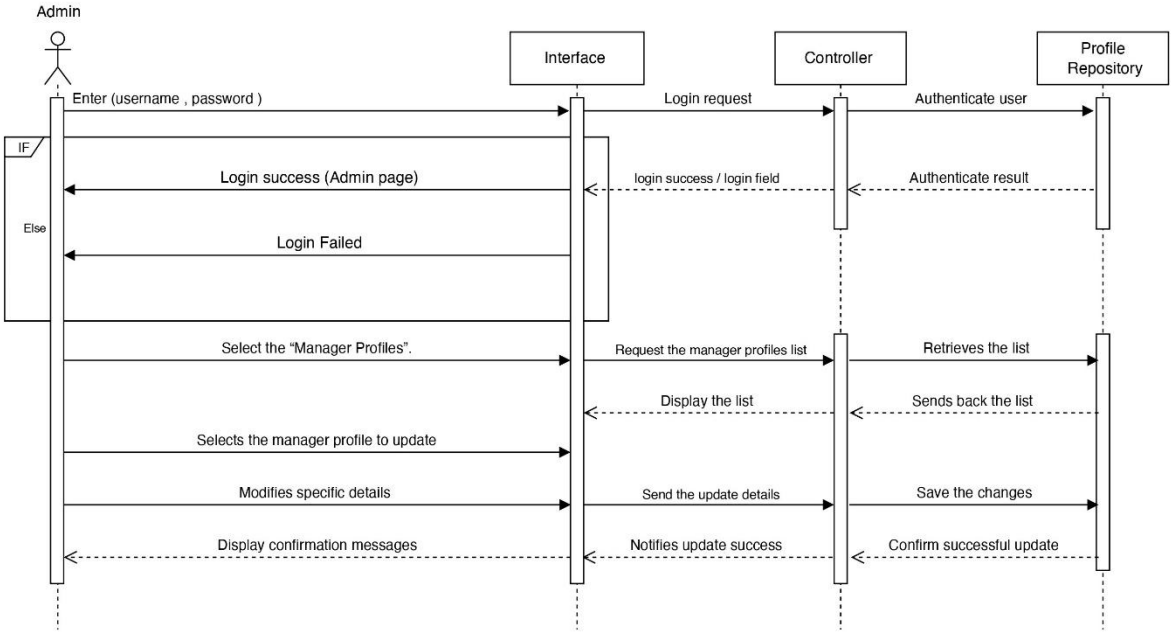
Update Medicine



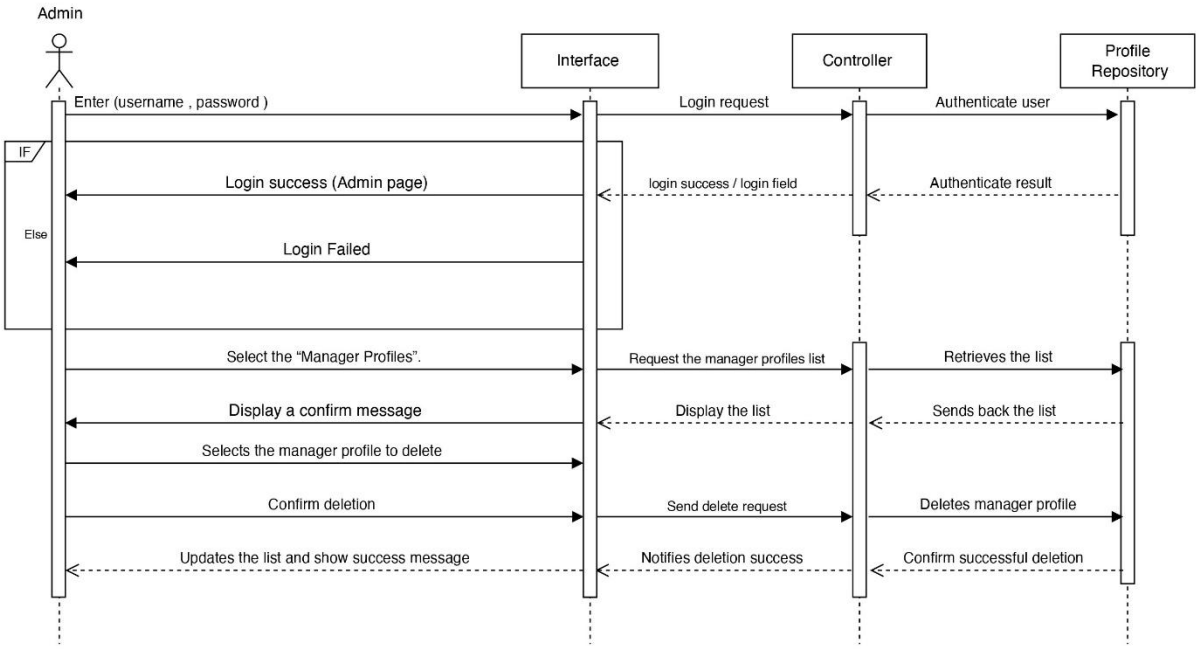
Create Manager Profiles



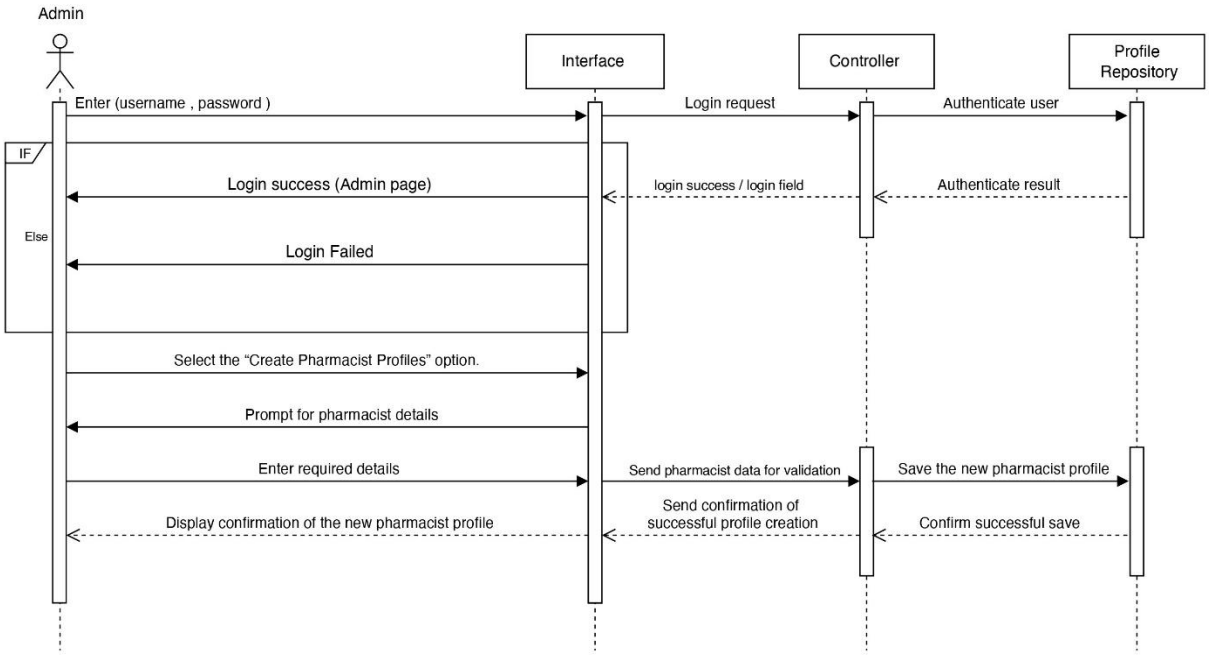
Update Manager Profiles



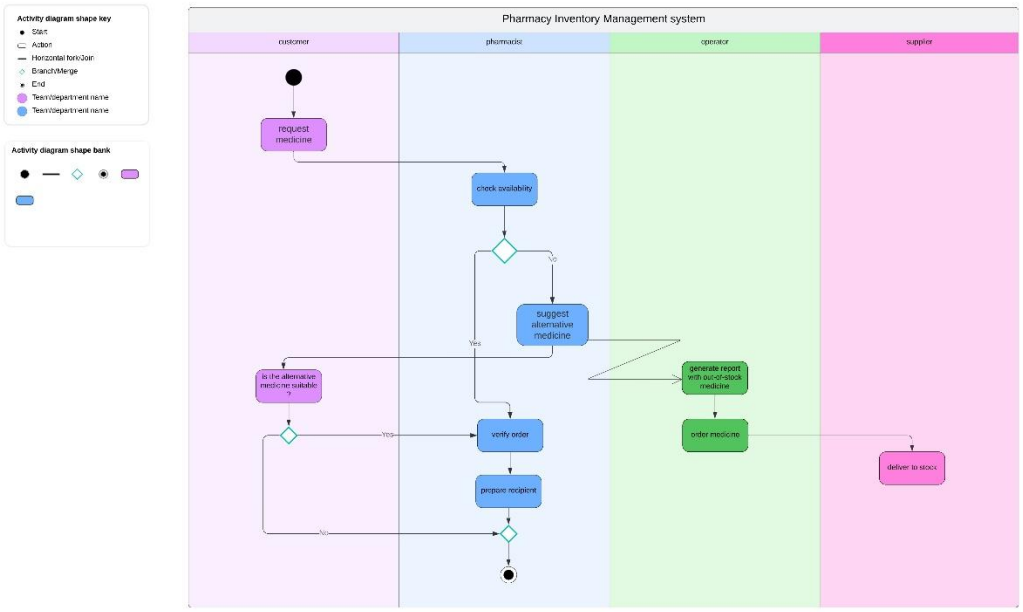
Delete Manager Profiles



Create Pharmacist Profiles



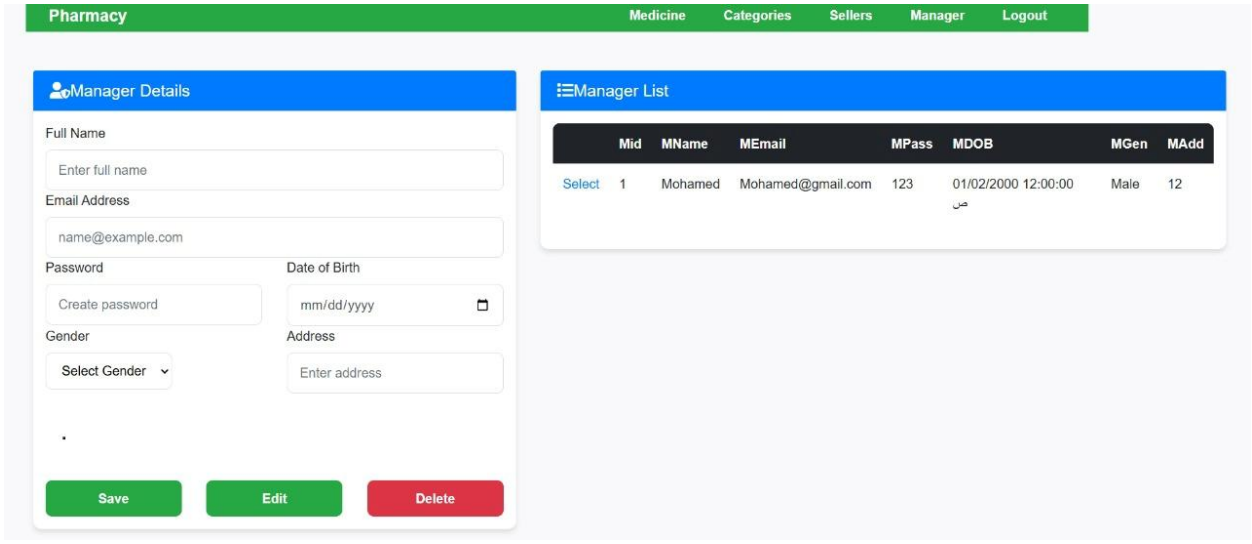
- **Activity Diagram**:
    - Steps for **ordering medicine** and **generating reports**.

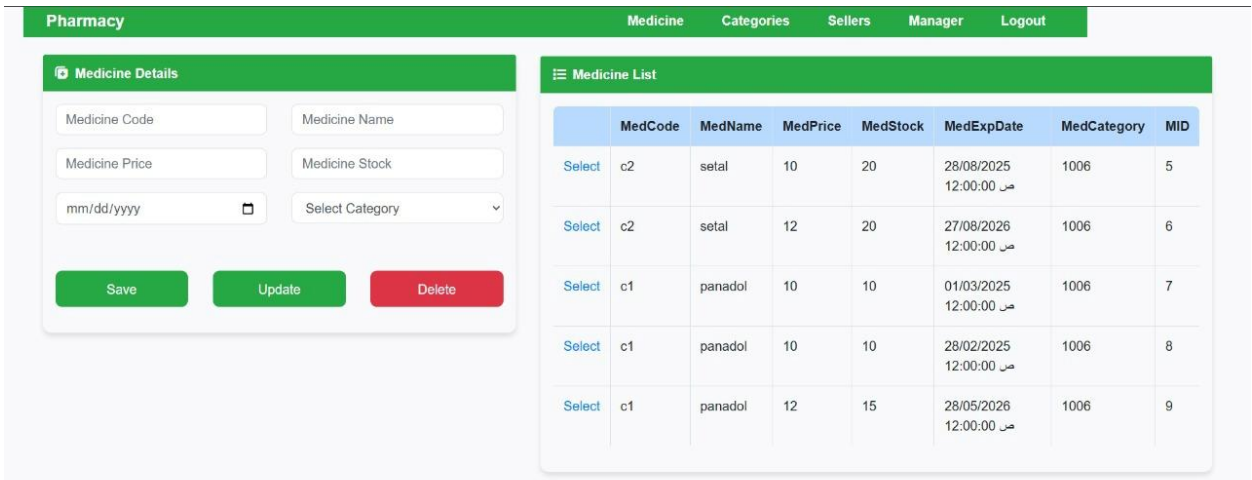


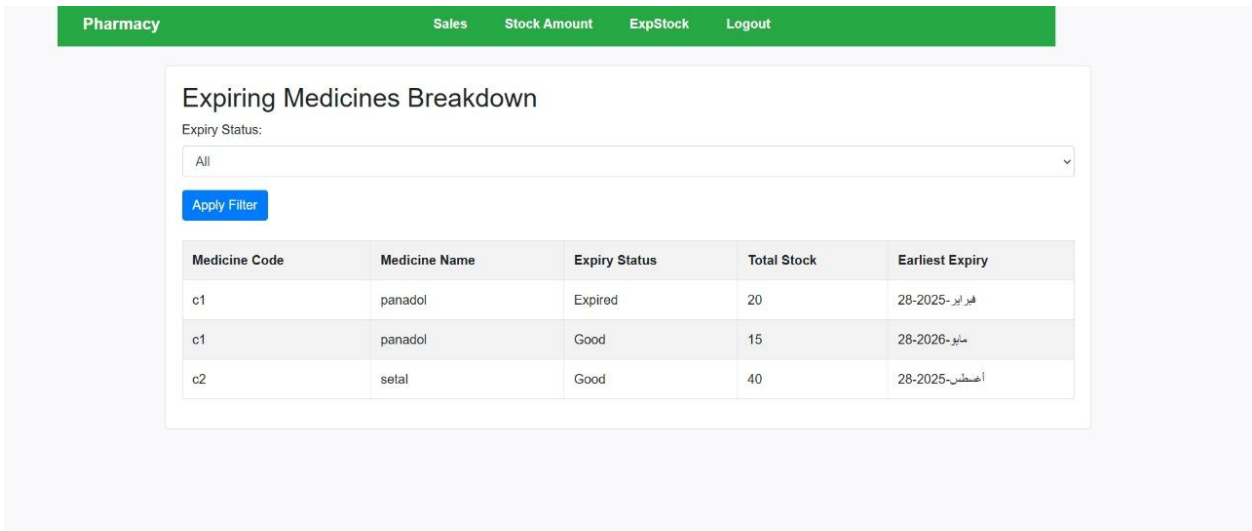
---

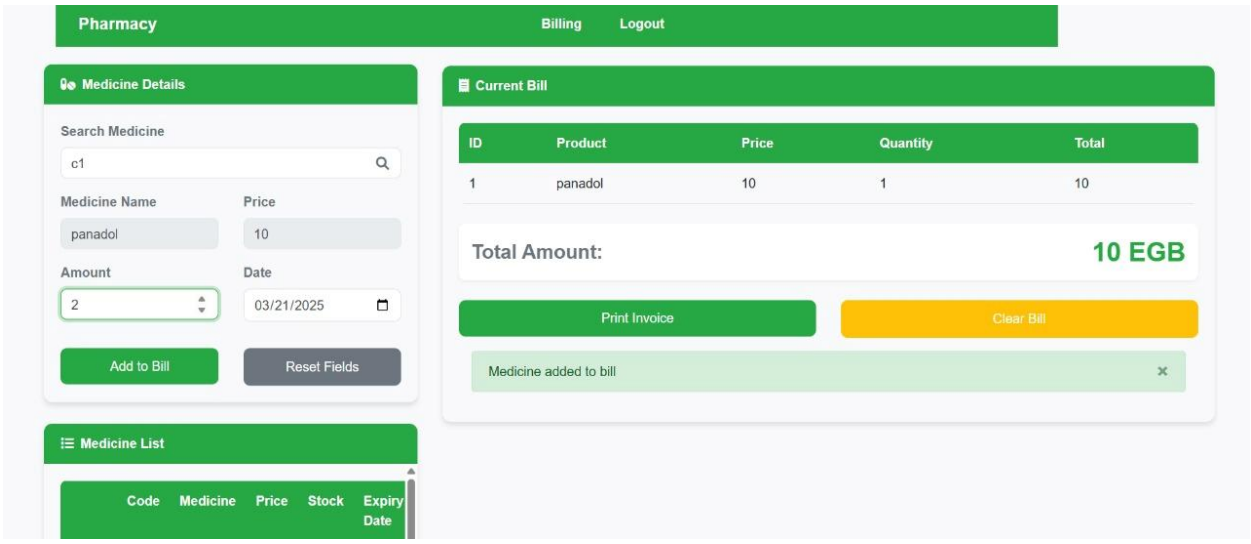
4. UI/UX Design & Prototyping

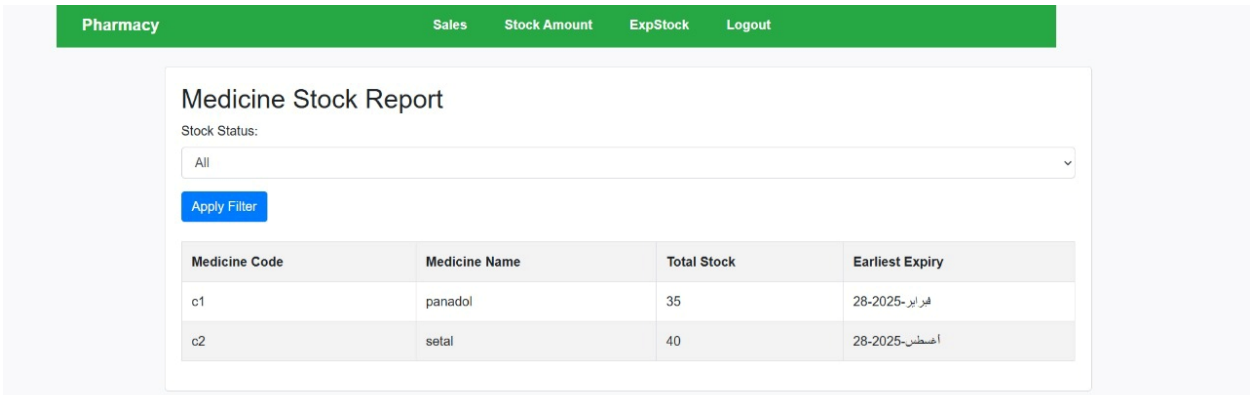
- **UI/UX Guidelines**:
    - **Minimalist design** with easy navigation.
    - **Color scheme** based on a medical/pharmacy theme.
    - **Typography** for easy readability.

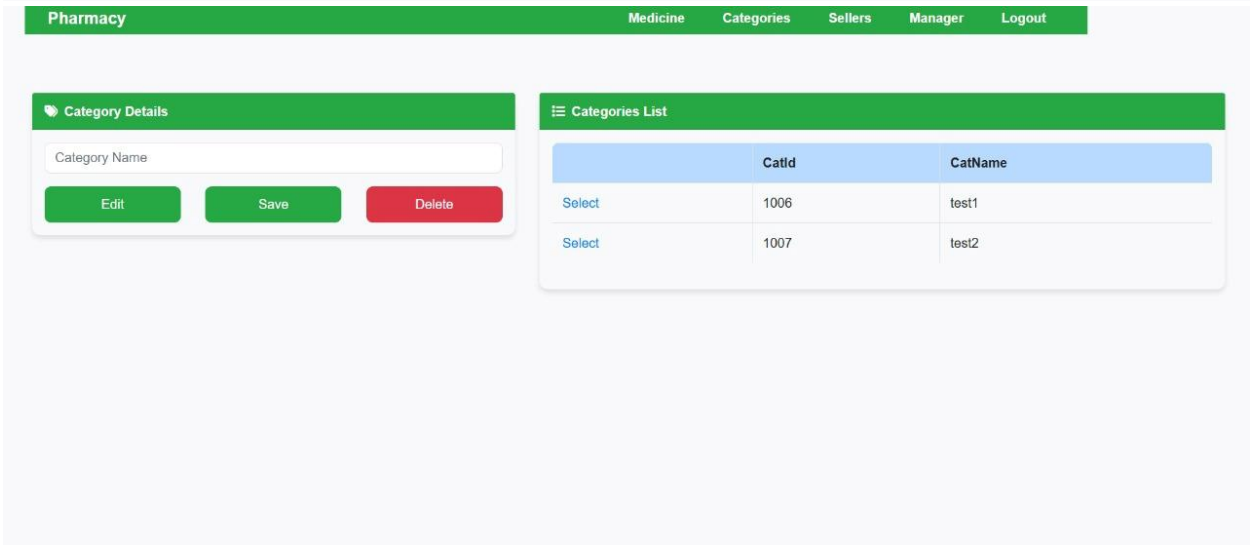

Pharmacy
Medicine
Categories
Sellers
Manager
Logout
Manager Details
Full Name
Enter full name
Email Address
name@example.com
Password
Create password
Date of Birth
mm/dd/yyyy
Gender
Select Gender
Address
Enter address
Save
Edit
Delete
Manager List
Mid
MName
MEmail
MPass
MDOB
MGen
MAdd
Select
1
Mohamed
Mohamed@gmail.com
123
01/02/2000 12:00:00 ص
Male
12

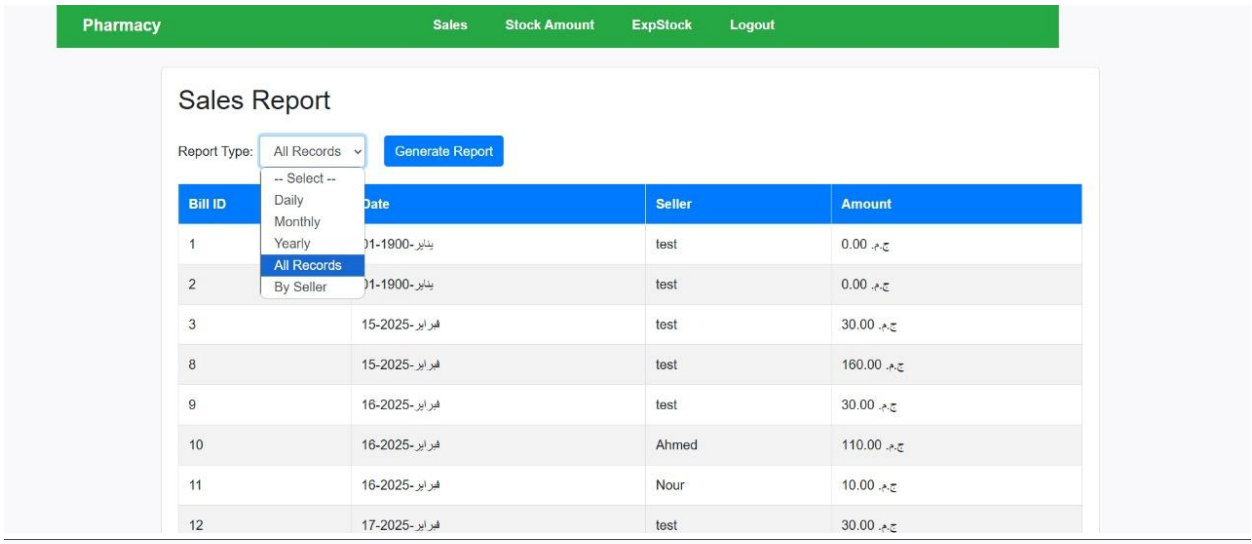

Pharmacy
Medicine
Categories
Sellers
Manager
Logout
Medicine Details
Medicine Code
Medicine Name
Medicine Price
Medicine Stock
mm/dd/yyyy
Select Category
Save
Update
Delete
Medicine List
MedCode
MedName
MedPrice
MedStock
MedExpDate
MedCategory
MID
Select
c2
setal
10
20
28/08/2025 12:00:00 ص
1006
5
Select
c2
setal
12
20
27/08/2026 12:00:00 ص
1006
6
Select
c1
panadol
10
10
01/03/2025 12:00:00 ص
1006
7
Select
c1
panadol
10
10
28/02/2025 12:00:00 ص
1006
8
Select
c1
panadol
12
15
28/05/2026 12:00:00 ص
1006
9

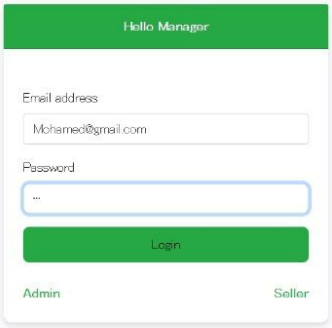

Pharmacy
Sales
Stock Amount
ExpStock
Logout
Expiring Medicines Breakdown
Expiry Status:
All
Apply Filter
Medicine Code
Medicine Name
Expiry Status
Total Stock
Earliest Expiry
c1
panadol
Expired
20
28-2025-فبراير
c1
panadol
Good
15
28-2026-مايو
c2
setal
Good
40
28-2025-أغسطس

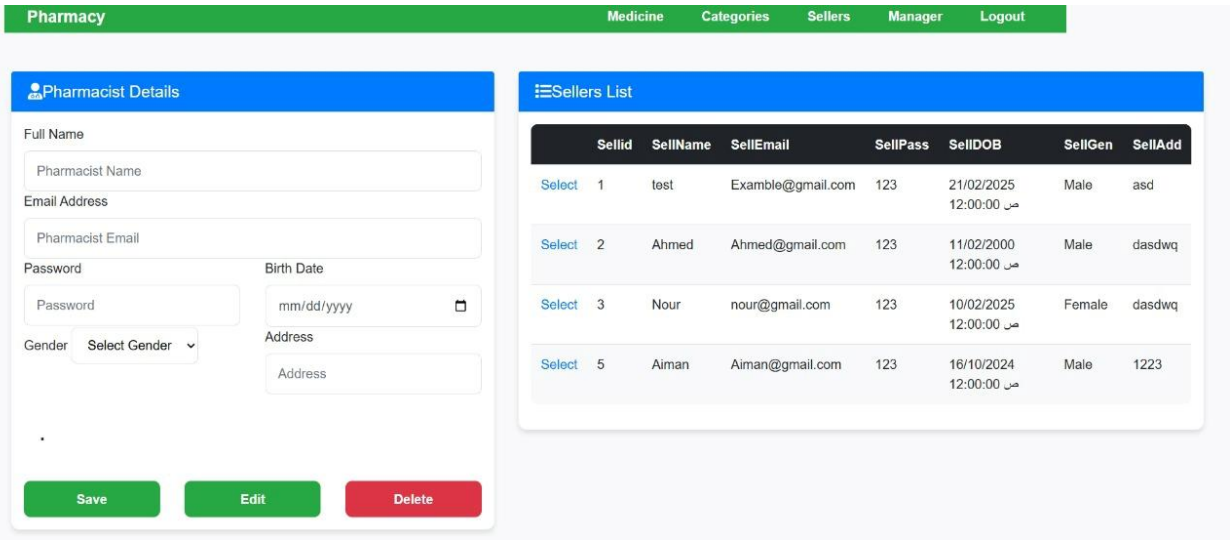

Pharmacy
Billing
Logout
Medicine Details
Search Medicine
c1
Medicine Name
Price
panadol
10
Amount
Date
2
03/21/2025
Add to Bill
Reset Fields
Current Bill
ID
Product
Price
Quantity
Total
1
panadol
10
1
10
Total Amount:
10 EGB
Print Invoice
Clear Bill
Medicine added to bill
Medicine List
Code
Medicine
Price
Stock
Expiry Date


Pharmacy
Sales
Stock Amount
ExpStock
Logout
Medicine Stock Report
Stock Status:
All
Apply Filter
Medicine Code
Medicine Name
Total Stock
Earliest Expiry
c1
panadol
35
28-2025-فبراير
c2
setal
40
28-2025-أغسطس


Pharmacy
Medicine
Categories
Sellers
Manager
Logout
Category Details
Category Name
Edit
Save
Delete
Categories List
CatId
CatName
Select
1006
test1
Select
1007
test2


Pharmacy
Sales
Stock Amount
ExpStock
Logout
Sales Report
Report Type:
All Records
Generate Report
-- Select --
Daily
Monthly
Yearly
All Records
By Seller
Bill ID
Date
Seller
Amount
1
test
0.00 ج.م
2
test
0.00 ج.م
3
15-2025-فبراير
test
30.00 ج.م
8
15-2025-فبراير
test
160.00 ج.م
9
16-2025-فبراير
test
30.00 ج.م
10
16-2025-فبراير
Ahmed
110.00 ج.م
11
16-2025-فبراير
Nour
10.00 ج.م
12
17-2025-فبراير
test
30.00 ج.م


Hello Manager
Email address
Mohamed@gmail.com
Password
Login
Admin
Seller


Pharmacy
Medicine
Categories
Sellers
Manager
Logout
Pharmacist Details
Full Name
Pharmacist Name
Email Address
Pharmacist Email
Password
Password
Birth Date
mm/dd/yyyy
Gender
Select Gender
Address
Address
Save
Edit
Delete
Sellers List
Sellid
SellName
SellEmail
SellPass
SellDOB
SellGen
SellAdd
Select 1 test Examble@gmail.com 123 21/02/2025 12:00:00 ص Male asd
Select 2 Ahmed Ahmed@gmail.com 123 11/02/2000 12:00:00 ص Male dasdwq
Select 3 Nour nour@gmail.com 123 10/02/2025 12:00:00 ص Female dasdwq
Select 5 Aiman Aiman@gmail.com 123 16/10/2024 12:00:00 ص Male 1223