Preface:

Project Name: System Requirement Specification for pharmacy management system

Date: 7 December 2024

Introduction:

Purpose:

The pharmacy management system is designed to enhance the efficiency and accuracy of day- to-day pharmacy operations. It provides a comprehensive solution for managing sales, purchases, inventory, financial transactions, and customer relations, ensuring seamless integration across all processes.

Background:

Pharmacies face challenges such as managing large inventories, tracking expiration dates, handling customer orders, processing returns, and maintaining accurate financial records.

Manual systems are prone to errors and inefficiencies. This system aims to digitize these processes, reducing errors, improving productivity, and ensuring compliance with industry standards.

Scope:

-The pharmacy management system will cover the following main functionalities:

1.**Sales and Returns**: Efficient handling of sales and processing of returns.

2.**Inventory Management**: Tracking stock levels, expiration dates, and reordering processes.

3.**Financial Accounting**: Managing daily transactions, monitoring cash flow, and generating financial reports.

4.**Customer and Supplier Management**: Keeping records of customers, suppliers, and their interactions.

5.**Reporting**: Generating detailed

Audience:

The intended audience for this document includes:

Pharmacy Owners/Managers: To understand system functionalities and ensure it meets business needs.

Developers and QA Engineers: To design, implement, and test the system effectively.

End-users (Pharmacy Staff): To operate the system efficiently.

Stakeholders: To monitor and oversee project execution.

Document Overview:

This document is structured into several sections:

Introduction: Overview of the system's goals and context.

Glossary: Definitions of key terms and acronyms used.

User: Description of user roles and interactions with the system.

System: Detailed system requirements and specifications.

Model: Visual diagrams to depict system workflows (use case and sequence diagrams).

Evolution: Future enhancements and scalability considerations.

Index: Quick reference guide to locate content.

Appendix: Supporting documentation and additional details.

Glossary:

Abbreviation	Definition
POS (Point of Sale)	A system used by the pharmacy to manage sales transactions, including payment processing, inventory updates, and receipts.
API (Application Programming Interface)	A set of protocols and tools that allows different software systems to communicate and interact with each other.
SKU (Stock Keeping Unit)	A unique identifier for each product in the inventory system, typically used for tracking and managing stock.
Barcode	A machine-readable representation of product information, typically used for quick identification and data entry in the system.

CRM (Customer Relationship Management)	A system used to manage customer interactions, data, and relationships to improve customer service and increase retention.
ERP (Enterprise Resource Planning)	A suite of integrated software applications used by businesses to manage and automate core operations like inventory, sales, and accounting.
UML (Unified Modeling Language)	: A standardized visual language used to model the design of software systems, including use case, sequence, and class diagrams.
Frontend	The part of the system that users interact with directly, typically involving the graphical user interface (GUI).

Backend	The part of the system that handles data processing, business logic, and interactions with databases or external systems.
Database	A structured collection of data stored electronically that can be accessed, managed, and updated by the system.
Encryption	The process of converting data into a secure format to prevent unauthorized access.
Authentication	The process of verifying the identity of a user or system, typically using passwords or multi-factor

User Requirement:

Overview:

The pharmacy management system will cater to various users, each with distinct roles and requirements. This section outlines the needs and expectations of these user groups to ensure the system provides value and supports their daily tasks effectively.

Detailed User Requirements:

- I want a pharmacy management system that organizes inventory, tracks sales, issues invoices, and manages customer and supplier data.
- I want the system to verify prescription details before selling medications that require a medical prescription.
- I need periodic notifications regarding medications nearing their expiration date.
- I want to record customer data, especially for those who purchase regularly
- I want to track all purchases from suppliers, including quantities and dates.
- I want the system to be available as a desktop application.
- I need detailed sales and inventory reports weekly, monthly, and annually.
- I want the system to support barcode functionality and printers to display the price and source of each product.
- I need payment options via credit cards or digital wallets.

System Architecture:

-Functional Requirements

1. Inventory Management

-Ability to add, update, and track medicines in stock.
-Automated alerts for medicines nearing expiry dates.
Integration with barcode scanning for inventory updates.

2. Sales Management

-Issue invoices for purchases.
-Validate prescription data before selling prescription-only medicines.
-Maintain a sales log for tracking sold items and quantities.

3. Customer and Supplier Management

-Store and manage customer details, including frequent buyers.

-Track supplier details, including purchase history (quantities, dates,prices).

4. Reporting

-Generate detailed yearly, monthly and daily reports on:

-Inventory status.

-Sales trends.

-Expired stock.

5. Notifications

-Provide periodic alerts for low-stock medicines.
-Notifications for medicines nearing expiry.

6. Platform Support

-Desktop application for primary operations.

7. Barcode and Printing Support

-Support for barcode scanners to streamline inventory and sales processes.

-Integration with printers for invoice and report printing.

Non-Functional Requirements

1. Performance

-The system should handle up to 1,000,000product records without performance degradation.

2. Scalability

-Easily extendable to accommodate new branches or users.

3. Availability

-99.9% uptime for the desktop version

4. Usability

-User-friendly interface for pharmacists and staff with minimal training required.
-Responsive design for the mobile application.

5. Data Integrity and Validation

-Ensure prescriptions are accurately validated before allowing sales.
-Robust database schema to prevent data loss or corruption.

6. Security

-Role-based access control (e.g., admin, staff).
-Data encryption for sensitive information (customer details, supplier,info)

7. Languages and Tools

-**Backend**: Python (Django/Flask)
orJava (Spring Boot).
-**Frontend**: react.js
-**Desktop**: Electron.js or WPF (C#).
-**Database**: PostgreSQL or MySQL
for structured data.

8. Hardware Requirements

-Barcode scanner compatible with the system.
-Printer support for A4-size and
receipt formats.

9. System Compatibility

-**Desktop**: Windows, macOS, or Linux support.

10. Data Backup and Recovery

-Automatic daily backups to local or cloud storage.

-Recovery mechanism for accidental deletions or failures.

11. Suggested Development Phases

Phase 1: Implement core functionalities, including inventory management, sales tracking, and reporting.
Phase 2: Add prescription validation and notification features.
Phase 3: Develop the mobile application for enhanced accessibility.
Phase 4: Integrate barcode scanning and printer support.
Phase 5: Add advanced analytics and additional features to improve decision-making and forecasting.
This development roadmap ensures the system evolves systematically, meeting immediate priorities first while allowing for future expansion and scalability.

Model:

1. Use Case Diagram

The use case diagram provides a high-level overview of the interactions between users and the system.

Actors

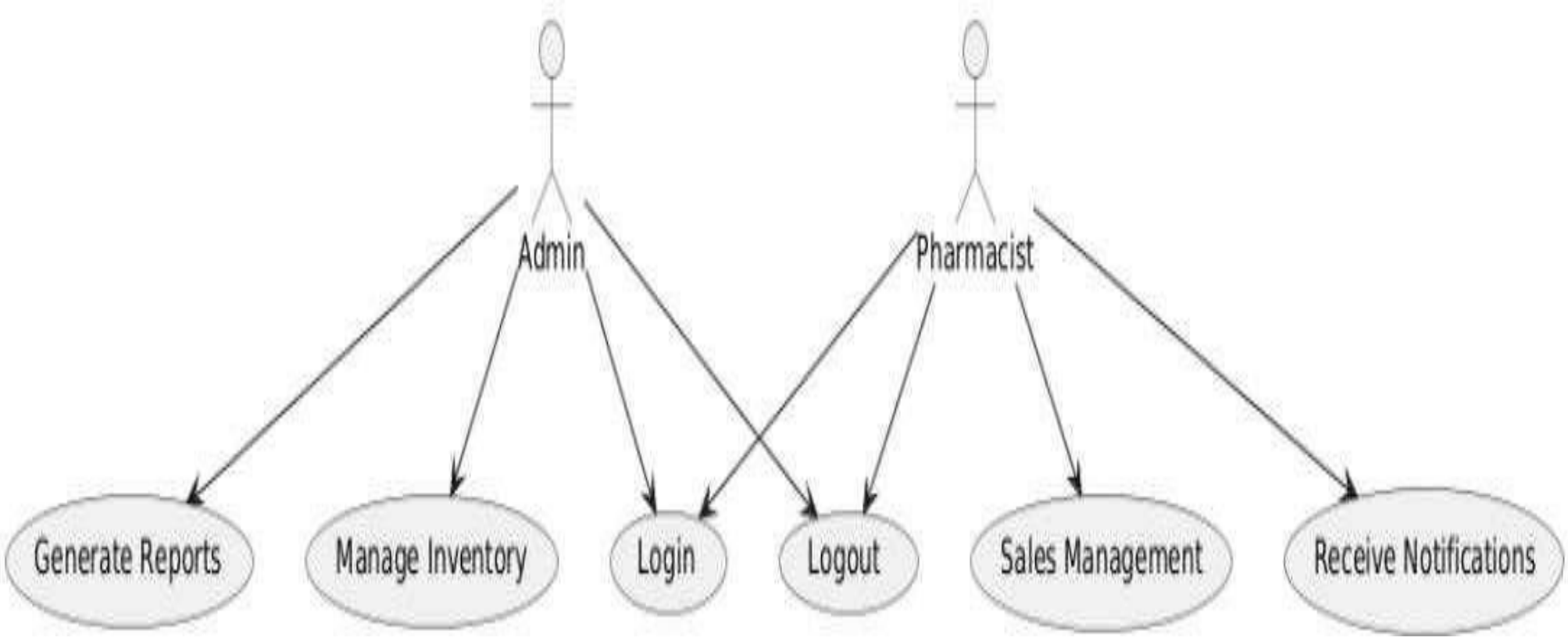
1. **Pharmacist**
   - The primary user responsible for daily sales operations and management.
   - Can log into the system and receive notifications related to sales or inventory.
1. **Admin**
   - Responsible for the overall management of the system.
   - Handles report generation and inventory management, in addition to login and logout.

Use Cases

1. **Login**
   - Enables all users to access the system.
2. **Logout**
   - Allows users to end their session within the system.
3. **Sales Management**
   - Used by the pharmacist to track and manage sales.

4. **Receive Notifications**
   - Allows the pharmacist to monitor notifications related to inventory or operations.
5. **Generate Reports**
   - Used by the admin to create analytical reports on pharmacy performance.
6. **Manage Inventory**
   - Used for managing products and inventory within the pharmacy.

Use Case Diagram:



2. Sequence Diagram

This **Sequence Diagram** illustrates the interactions between the **Pharmacist** and the **Pharmacy Management System** during key processes. The diagram shows the sequence of steps involved in each process to ensure smooth and efficient operations.

1. Login*:*

- **Pharmacist** enters credentials.
- **Pharmacy Management System** verifies the credentials.
- Successful login grants access to the system.

2. Inventory Management:

- **Pharmacist** adds or updates medicines.
- **Pharmacy Management System** updates the inventory.
- **Pharmacy Management System** checks expiry dates.
- If necessary, **Pharmacy Management System** sends expiry alerts.

3. Sales Management:

- **Pharmacist** validates prescriptions.
- **Pharmacy Management System** logs sales.
- **Pharmacy Management System** updates sales records.

4. Report Generation:

- **Pharmacist** requests a report.

- **Pharmacy Management System** fetches relevant data from the database.
- **Pharmacy Management System** displays the report to the pharmacist.

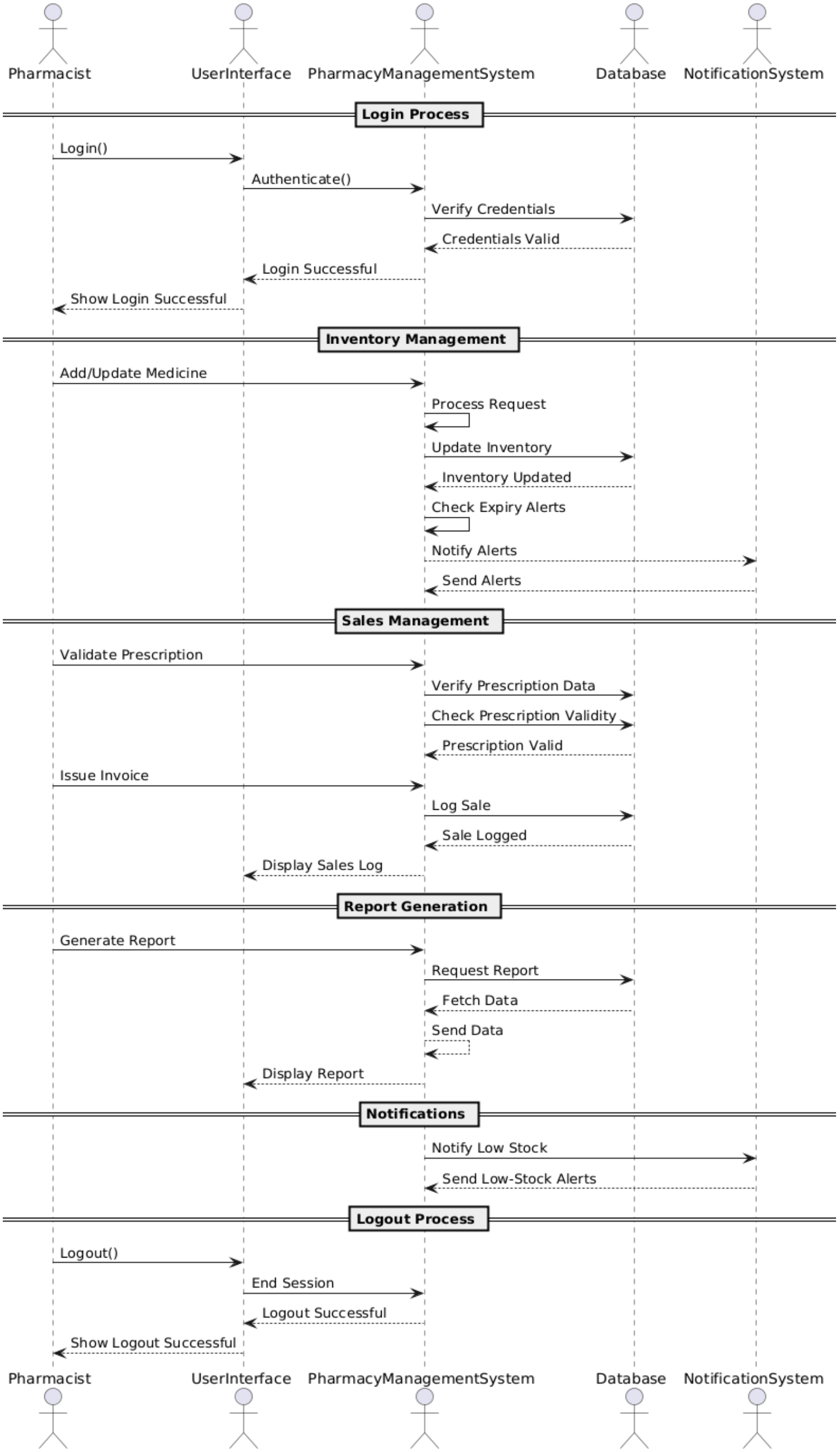
5. Notifications:

- **Pharmacy Management System** sends alerts for low stock.
- **Pharmacy Management System** sends alerts for expired medicines.

6. Logout:

- **Pharmacist** ends the session.
- **Pharmacy Management System** successfully logs the user out.

Sequence Diagram:


Pharmacist
UserInterface
PharmacyManagementSystem
Database
NotificationSystem
Login Process
Login()
Authenticate()
Verify Credentials
Credentials Valid
Login Successful
Show Login Successful
Inventory Management
Add/Update Medicine
Process Request
Update Inventory
Inventory Updated
Check Expiry Alerts
Notify Alerts
Send Alerts
Sales Management
Validate Prescription
Verify Prescription Data
Check Prescription Validity
Prescription Valid
Issue Invoice
Log Sale
Sale Logged
Display Sales Log
Report Generation
Generate Report
Request Report
Fetch Data
Send Data
Display Report
Notifications
Notify Low Stock
Send Low-Stock Alerts
Logout Process
Logout()
End Session
Logout Successful
Show Logout Successful
Pharmacist
UserInterface
PharmacyManagementSystem
Database
NotificationSystem

Evolution

The pharmacy management system is designed with scalability in mind, allowing it to evolve and adapt to future business needs. Below are key areas for future enhancements and improvements:

1.Automatic Price Updates

Current State: Prices are manually updated by system administrators or inventory managers.

Future Enhancement:

Integrate with external APIs from suppliers or regulatory bodies to receive real-time price updates.

Automate periodic price synchronization to maintain consistent and accurate pricing.

2.Enhanced Internet Connectivity

- **Current State**: The system operates locally with minimal reliance on the internet.
- **Future Enhancement**:

Optimize the system for cloud-based deployment, enabling remote access.

Introduce offline-mode capabilities, allowing the system to sync with the cloud once internet connectivity is restored.

3.Advanced Reporting and Analytics

- **Current State**: Basic reporting features are available for sales, inventory, and financial data.
- **Future Enhancement**:

Integrate AI-driven analytics for trend predictions, customer behavior analysis, and inventory optimization.

Develop interactive dashboards for real-time insights to aid in decision-making.

4. Multi-Branch Support

- **Current State**: The system is designed for single-location pharmacies.
- **Future Enhancement**:

Extend the system to support multi-branch operations, with centralized control and branch-specific reporting.
Implement functionality for inter-branch inventory transfers.

5. AI-Based Recommendations

- **Future Enhancement**:

Implement AI-powered modules to suggest related products during sales interactions.
Use predictive analytics to forecast demand and recommend stock replenishment.

6. Enhanced Security Measures

- **Current State**: Role-based access control and basic encryption are in place.
- **Future Enhancement**:

Integrate advanced security measures, such as two-factor authentication and biometric login options.
Enhance monitoring tools to identify and mitigate potential security risks.

7. Integration with External Systems

- **Future Enhancement**:

Enable integration with hospital systems, health insurance providers, and government health programs for seamless data exchange.
Support electronic prescriptions and automated compliance tracking.

8. Support for IoT Devices

- **Future Enhancement**:

Connect IoT-enabled devices for automatic stock monitoring and environmental control (e.g., temperature tracking for sensitive medications).

Implementation Strategy

Prioritize enhancements based on user feedback and evolving business needs.
Implement changes in incremental stages to minimize disruption to ongoing operations.
Ensure comprehensive testing and documentation for each update to guarantee smooth transitions.

Appendix

1. System Architecture Diagram

- Include a diagram showing the structure of the pharmacy management system. Illustrate the interactions between:
  - Frontend (User Interface)
  - Backend (Business Logic and Servers)
  - Database (Data Storage)
  - External Interfaces (e.g., Barcode Scanners, Payment Gateways)

2. System Workflow and User Flow Diagrams

- Provide detailed flow diagrams for:
  - **Sales Processing and Returns**: Illustrate how sales transactions are handled and how returns are processed.
  - **Inventory Management and Reordering**: Show the workflow for managing inventory and automated reordering.

- **User Management**: Diagram role assignments, permissions, and login/logout processes.
- **Report Generation**: Depict how reports are created and exported.

3. Data Flow Diagrams (DFD)

- Include:
  - **Level 1 DFD**: A high-level overview showing data movement through the system.
  - **Level 2 DFD**: Detailed processes like inventory tracking or order management.

4. Software Requirements Specification (SRS) Template

- Provide a reusable SRS template for future software development projects. Include sections for functional and non-functional requirements.

5. API Documentation

- Include information for integrating external systems:

- **Endpoints**: API URLs
- **HTTP Methods**: Supported methods (GET, POST, etc.)
- **Request/Response Parameters**: Details of what data to send and expect.
- **Sample Requests and Responses**

6. Security Considerations

1. Summarize key security measures implemented in the system:
   - **Encryption**: Explain encryption methods (e.g., AES-256, SSL).
   - **Authentication**: Role-based access and two-factor authentication.
   - **Authorization**: Describe restrictions based on user roles.

Index:

Table of Contents