PHARMACY MANAGEMENT SYSTEM

A MINI-PROJECT REPORT

Submitted by

SAI PRAPANCH.H	**241901093**
VIJAY VARSITH.I	**241901125**

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

(CYBER SECURITY)

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI - 602105

An Autonomous Institute

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project “PHARMACY MANAGEMENT SYSTEM” is the bonafide work of “SAI PRAPANCH.H (241901093), VIJAY VARSITH.I (241901125) who carried out the project work under my supervision.

SIGNATURE

Ms. R. Rupmala

ASSISTANT PROFESSOR

Department of Computer Science

and Engineering (Cyber Security)

Rajalakshmi Engineering College

Chennai

This mini project report is submitted for the viva voce examination to be held on __________

INTERNAL EXAMINER   EXTERNAL EXAMINER

DECLARATION

We hereby declare that the mini project report Pharmacy Management System, submitted as part of the curriculum requirements for the Bachelor of Engineering (B.E) degree affiliated to Anna University, is a bonafide work carried out by us under the supervision of Ms. R. Rupmala, Assistant Professor, Department of Computer Science Engineering and Cyber Security, Rajalakshmi Engineering College, Chennai.

This submission represents our ideas in our own words, and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data, idea, fact, or source in our submission. We understand that any violation of the above will be grounds for disciplinary action by the institute and/or the University and may also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been obtained. This report has not previously formed the basis for the award of any degree, diploma, or similar title of any other University.

Rajalakshmi Engineering College,

SAI PRAPANCH.H

Chennai
November 2025

VIJAY VARSITH.I

ABSTRACT

The Pharmacy Management System is a desktop-based application developed using Java (Swing GUI) for the frontend and MySQL as the backend database. The main objective of this project is to digitalize the traditional pharmacy operations and provide a simple, secure, and efficient system to manage medicines, customers, and sales transactions.

The system enables the pharmacist to maintain complete records of available medicines, track stock quantities, update expired products, manage customer information, and process daily sales efficiently. It provides a user-friendly graphical interface with a modern FlatLaf Dark Theme, ensuring better usability and a professional appearance.

The application uses JDBC (Java Database Connectivity) to establish seamless communication between the user interface and the database. The modular structure of the project is divided into key components such as Medicine Management, Customer Management, and Sales/Billing, each handling a specific functionality. The sales module automatically calculates the total amount for each transaction and updates the inventory in real time.

By automating pharmacy operations, this system minimizes manual errors, saves time, and provides accurate, up-to-date information. The project demonstrates practical implementation of object-oriented programming, database integration, and graphical user interface development in Java. This solution can be further enhanced to include features like user authentication, barcode scanning, and cloud-based data synchronization for large-scale pharmaceutical use.

Keywords: Java, MySQL , JDBC, Swing GUI, FlatLaf, Database Integration, Inventory Management, Sales Billing.

ACKNOWLEDGEMENT

We like to convey our sincere appreciation to all who have supported and mentored us during the successful completion of this project work.

We express our profound gratitude to Mr. Benedict J.N., Associate Professor (SG) and Head of the Department of Computer Science Engineering and Cyber Security at Rajalakshmi Engineering College, for furnishing us with essential resources, support, and an enabling environment to execute this project.

We express our profound gratitude to Ms. R. Rupmala, Assistant Professor in the Department of Computer Science Engineering and Cyber Security at Rajalakshmi Engineering College, for her unwavering support, invaluable insights, and collaboration throughout our endeavor.

We would like to convey our gratitude to our faculty members and colleagues for their valuable feedback and encouragement.

We express our gratitude to our families and friends for their steadfast support, patience, and encouragement, which were instrumental in the effective execution of this seminar.

SAI PRAPANCH.H

VIJAY VARSITH.I

TABLE OF CONTENTS

LIST OF ABBREVIATION

ABBREVIATION	FULL TERM
CRUD	Create, Read, Update, Delete
SQL	Structured Query Language
JDBC	Java Database Connectivity
Java Swing	Java Framework (for GUI)
GUI	Graphical User Interface
UI	User Interface
ER	Entity-Relationship

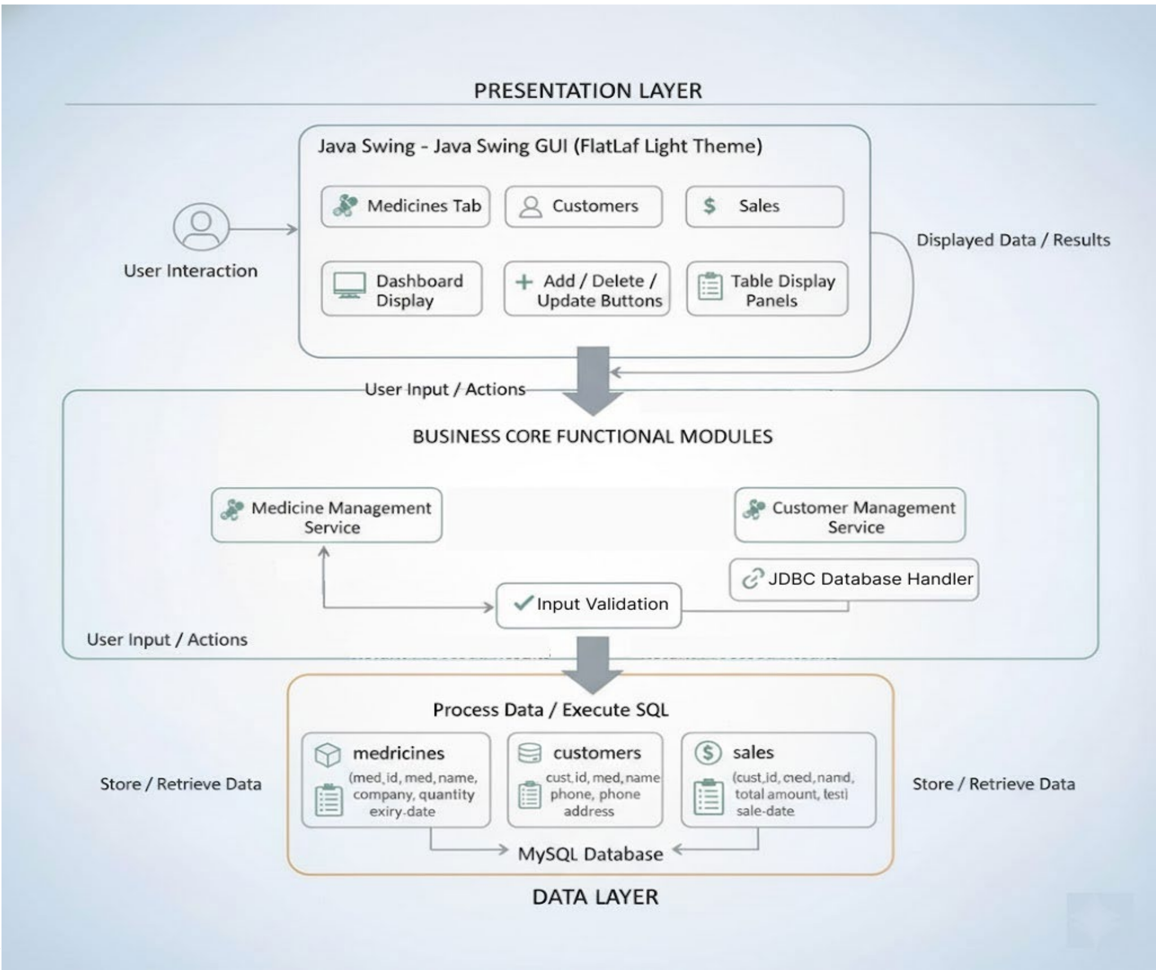
LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Overview

The Pharmacy Management System is a desktop application developed using Java and MySQL to help manage the daily operations of a pharmacy. It allows the pharmacist to easily maintain records of medicines, customers, and sales in a single system instead of using paper or manual registers. The system provides a simple and user-friendly interface built using Java Swing with a modern FlatLaf theme. Users can add, edit, or delete medicine and customer details, check stock levels, and record sales. Whenever a sale is made, the total bill is automatically calculated, and the medicine stock is updated in the database.


PRESENTATION LAYER
Java Swing - Java Swing GUI (FlatLaf Light Theme)
Medicines Tab
Customers
$ Sales
Dashboard Display
+ Add / Delete / Update Buttons
Table Display Panels
User Interaction
Displayed Data / Results
User Input / Actions
BUSINESS CORE FUNCTIONAL MODULES
Medicine Management Service
Customer Management Service
JDBC Database Handler
✓ Input Validation
User Input / Actions
Process Data / Execute SQL
medricines
(med, id, med, name, company, quantity exiry-date
customers
cust id, med, name phone, phone address
sales
(cust id, cned, namd, total amount, (est) sale-date
MySQL Database
Store / Retrieve Data
Store / Retrieve Data
DATA LAYER

/

Fig.1.1 High-Level System Architecture

1.2 Scope of the Work

The Pharmacy Management System is designed to improve and automate the workflow of a pharmacy by managing medicines, customers, and sales operations efficiently. The scope of this project covers the core features required to run a pharmacy digitally, while also allowing room for future enhancements as the business grows.

This system provides an all-in-one solution that enables the pharmacist to handle tasks such as inventory control, billing, and customer record management without the need for manual registers or spreadsheets. All data is stored securely in a MySQL database, which can be updated and retrieved easily using Java's JDBC connectivity.

The system performs the following key functions:

- Maintain a complete list of medicines, including details such as name, company, price, quantity, and expiry date.
- Add new medicines and remove or update existing ones when stock changes.
- Manage customer details, including name, phone number, and address for record-keeping and billing.
- Handle sales transactions by selecting medicines and quantities, automatically calculating total costs, and updating stock levels in real time.
- Display all data in an organized tabular format through a graphical user interface built with Java Swing and styled using the FlatLaf theme for a modern look and smooth user experience.

Beyond the current features, the system can be expanded to include advanced capabilities such as:

- Barcode scanning for fast and error-free billing.
- Expiry and low-stock alerts to help manage inventory effectively.
- Report generation for daily, weekly, or monthly sales analysis.
- User authentication and role-based access control for better security.
- Cloud synchronization to allow access across multiple branches of a pharmacy.

The system is primarily intended for small and medium-sized pharmacies, but its modular and scalable design makes it adaptable for larger setups with minor upgrades. By providing a simple, secure, and efficient way to manage data, this system not only reduces human error and time consumption but also improves customer service and overall business performance.

1.3 Problem Statement

In most small and medium-sized pharmacies, the management of medicines, customers, and sales is still carried out manually using traditional methods such as notebooks, ledgers, or spreadsheets. This manual system often leads to several problems such as data inconsistency, human errors, loss of records, and inefficient stock tracking. As the volume of data increases, it becomes very difficult to retrieve information quickly or generate reports when needed.

A pharmacist must frequently perform repetitive tasks such as recording medicine details, checking stock availability, calculating total bills, and tracking expiry dates — all of which are prone to human error when done manually. There is also no effective way to automatically update medicine quantities after a sale or generate accurate summaries of transactions. Furthermore, without automation, it becomes difficult to maintain customer records for repeat purchases or to handle a large number of transactions during busy hours.

The lack of a computerized system results in:

- Wastage of time in searching for medicine details or verifying stock levels.
- High chances of mistakes in billing or data entry.
- Difficulty in identifying expired or nearly expired medicines.
- Inaccurate stock and sales reports.
- Poor customer service due to slow processing.

To overcome these challenges, there is a need for a **computerized Pharmacy Management System** that automates all key operations of a pharmacy. Such a system should provide an easy-to-use interface for data entry, accurate billing, automatic stock updates, and a secure database for storing all information. The new system must ensure reliability, accuracy, and speed in managing pharmacy operations while reducing the dependency on manual paperwork.

By implementing this automated system, pharmacies can streamline their business operations, maintain accurate inventory, and improve overall efficiency and customer satisfaction.

1.4 Aim and Objectives of the Project

The main aim of the Pharmacy Management System is to create a simple and efficient software application that helps pharmacies manage their daily work. It is designed to replace manual record-keeping with a digital system that can store, update, and organize data about medicines, customers, and sales quickly and accurately. The goal is to make pharmacy operations faster, reduce human errors, and improve overall management through an easy-to-use computer application.

The specific objectives of the Pharmacy Management System are:

1. To build a system that stores and manages information about medicines, customers, and sales.

2. To automate billing and stock updates after every sale.

3. To reduce manual work and human errors in data entry and calculation.

4. To provide a user-friendly graphical interface for easy navigation and operation.

5. To use MySQL for storing data securely and JDBC for connecting it with the Java application.

6. To display all data in tables for better viewing and quick access.

7. To help the pharmacist easily track stock levels, expiry dates, and sales records.

8. To design the project so it can be expanded in the future, adding features like login security, barcode scanning, and automatic alerts.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Component	Minimum Specification
Processor	Dual-core 2.0 GHz or higher
Memory (RAM)	4GB (Minimum), 8GB (Recommended)
Storage	200MB free space
Display	1366x768 resolution

2.2 SOFTWARE SPECIFICATIONS

Component	Specification
Operating System	Windows 10/11, macOS 10.15+, Linux
Front-End	Java Swing
Back-End	MySQL 8.0 or above

Core Language	Java SE 17
Dependencies	MySQL JDBC Driver (connect JAVA to MYSQL), FlatLaf (theme), JDK 17 (Core Java Runtime)

CHAPTER 3

MODULE DESCRIPTION

The application is built on a modular architecture, with a clear separation of concerns across five primary modules:

3.1 Medicine Stock Module

This module handles all the activities related to managing the details of medicines available in the pharmacy.

Functions:

- Add new medicines with details such as name, company, price, quantity, and expiry date.
- Update medicine details whenever changes occur.
- Delete expired or unavailable medicines from the list.
- Display all medicines in a tabular format for easy viewing.
- Automatically update the medicine stock after a sale is made.

3.2 Customer Management Module

This module manages all information related to customers.

Functions:

- Add new customer details like name, phone number, and address.
- Update customer information when necessary.
- Delete customer details when no longer needed.
- View customer information during sales or billing.

3.3 Sales and Billing Module

This module handles sales transactions and billing operations.

Functions:

- Select the customer and medicine for the sale.
- Enter the quantity to calculate the total amount automatically.
- Deduct the sold quantity from the available stock.

- Record each sale in the database with total amount and date.
- View all past sales transactions in a table.

3.4 Database Module

This module manages all interactions between the Java application and the MySQL database.

Functions:

- Establish and close connections to the database.
- Execute SQL commands (INSERT, UPDATE, DELETE, SELECT).
- Retrieve and return data to other modules.
- Handle exceptions related to database operations.

3.5 User Interface Module

This module provides a graphical interface for user interaction. It is built using Java Swing with FlatLaf for modern styling.

Functions:

- Display medicines, customers, and sales using tabs.
- Provide buttons for add, update, and delete operations.
- Show dialog boxes for notifications and confirmations.
- Present data in neatly formatted tables.

3.6 ER Diagram

The Entity-Relationship (ER) Diagram shown above represents the logical structure of the database used in the Pharmacy Management System. It illustrates the relationships between the three main entities — Customers, Sales, and Medicines — and defines how data flows between them. Each Customer can make multiple Sales, but every sale record is linked to one specific customer, forming a one-to-many relationship between *Customers* and *Sales.* Similarly, each Medicine can be sold in multiple Sales, but every sale record corresponds to a single medicine, creating another one-to-many relationship between *Medicines* and *Sales*.

The Sales entity acts as a bridge between Customers and Medicines, storing the transaction details such as quantity, total amount, and sale date. Each table has a unique Primary Key (PK) to identify records, and the Foreign Keys (FK) in the *Sales* table maintain referential integrity between the related entities.

This diagram ensures a clear understanding of the database design and helps maintain consistency, accuracy, and data normalization throughout the system.

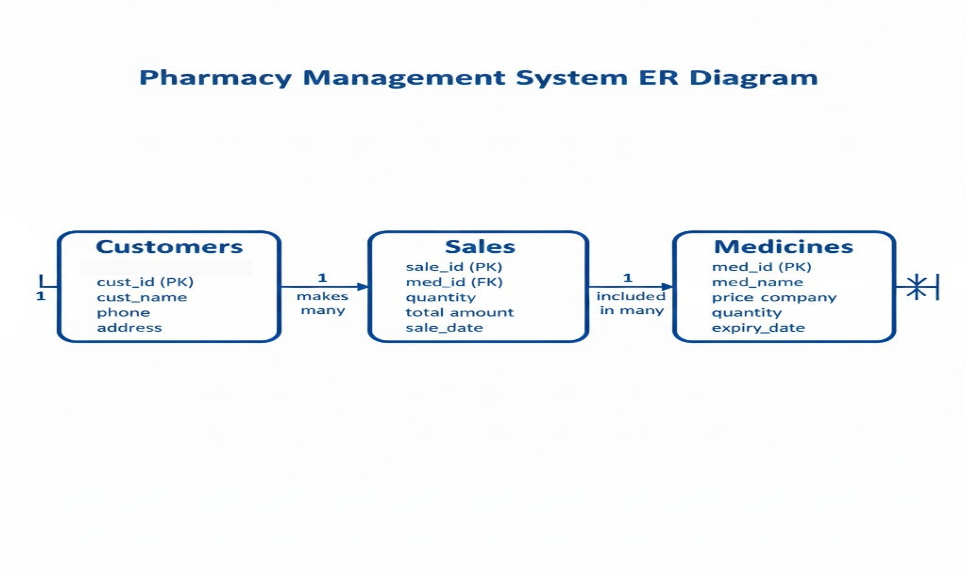
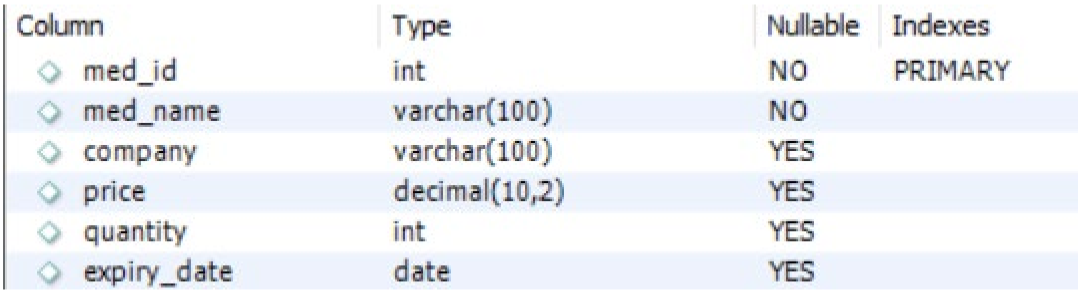


Fig.3.1 ER Diagram

3.7 Database Schema

Explanation: Database Schema of Medicines

The medicines table stores all the details of medicines available in the pharmacy. Each record contains information such as medicine name, company, price, available quantity, and expiry date. This table is the core of the system as it helps in inventory management and stock tracking.

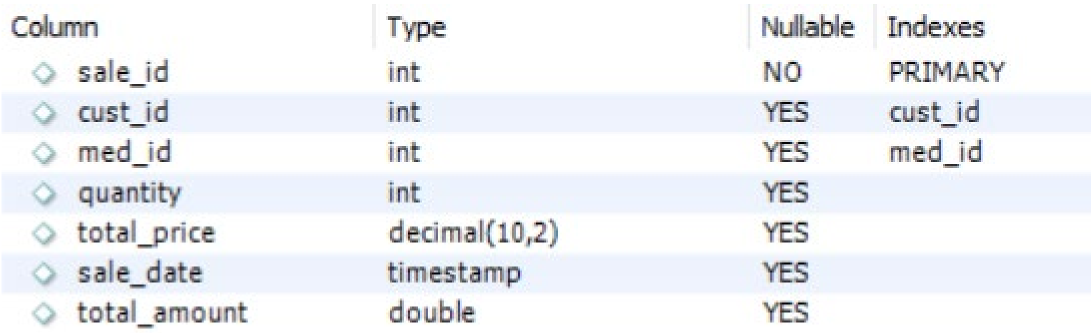


Column	Type	Nullable	Indexes
◇ med_id	int	NO	PRIMARY
◇ med_name	varchar(100)	NO	
◇ company	varchar(100)	YES	
◇ price	decimal(10,2)	YES	
◇ quantity	int	YES	
◇ expiry_date	date	YES	

Fig.3.2 Database Schema of Medicines

Explanation: Database Schema of Sales

The sales table stores information about all sales transactions made at the pharmacy. Each sale entry records which customer purchased which medicine, the quantity sold, total bill amount, and the sale date. This table is linked with both customers and medicines tables using foreign keys to maintain data integrity.

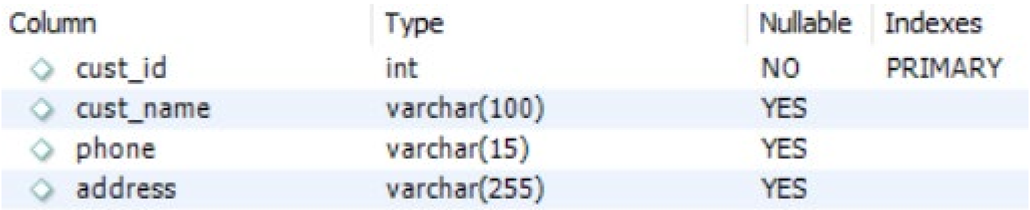


Column	Type	Nullable	Indexes
sale_id	int	NO	PRIMARY
cust_id	int	YES	cust_id
med_id	int	YES	med_id
quantity	int	YES	
total_price	decimal(10,2)	YES	
sale_date	timestamp	YES	
total_amount	double	YES	

Fig.3.3 Database Schema of Sales

Explanation: Database Schema of Customers

The customers table stores the details of all customers who purchase medicines from the pharmacy. This helps maintain customer records for future reference or repeat purchases. Each customer is assigned a unique ID, and details such as name, phone number, and address are recorded.



Column	Type	Nullable	Indexes
cust_id	int	NO	PRIMARY
cust_name	varchar(100)	YES	
phone	varchar(15)	YES	
address	varchar(255)	YES	

Fig.3.4 Database Schema of Customers

CHAPTER 4

CODING

4.1 Introduction

This chapter presents the source code implementation of the Pharmacy Management System developed using Java and MySQL. The program is divided into multiple modules such as database connection, medicine management, customer management, sales and billing, and user interface. The application uses Java Swing for the GUI and JDBC for database connectivity. The goal of the coding phase is to integrate all modules to work together efficiently and ensure smooth pharmacy operations.

4.2 Database Connection Code

This module connects the Java application to the MySQL database using JDBC. It ensures that the database is accessible whenever data is inserted, deleted, or updated.

```
private Connection getConnection() {

  try {

      String url = "jdbc:mysql://localhost:3306/pharmacydb";

      String user = "root";

      String pass = "your_password"; // replace with your MySQL password

      return DriverManager.getConnection(url, user, pass);

  } catch (SQLException e) {

      JOptionPane.showMessageDialog(this, "Database connection failed: " +
e.getMessage());

      return null;

  }

}
```

4.3 Medicine Management Module

This module handles all operations related to medicines, such as adding, updating, deleting, and displaying them in the table.

```
private void addMedicine(String name, String company, double price, int qty, String expiry)
{

   try {

       PreparedStatement ps = conn.prepareStatement(

           "INSERT INTO medicines (med_name, company, price, quantity, expiry_date)
VALUES (?, ?, ?, ?, ?)");

       ps.setString(1, name);

       ps.setString(2, company);

       ps.setDouble(3, price);

       ps.setInt(4, qty);

       ps.setString(5, expiry);

       ps.executeUpdate();

       JOptionPane.showMessageDialog(this, "Medicine Added Successfully!");

   } catch (Exception e) {

       JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());

   }

}

private void deleteMedicine(int id) {

   try {

       PreparedStatement ps = conn.prepareStatement("DELETE FROM medicines WHERE
med_id = ?");

       ps.setInt(1, id);
```

```
        ps.executeUpdate();

        JOptionPane.showMessageDialog(this, "Medicine Deleted Successfully!");

    } catch (Exception e) {

        JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());

    }

}
```

4.4 Customer Management Module

This module manages customer records such as adding new customers, viewing details, and deleting records.

```
private void addCustomer(String name, String phone, String address) {

    try {

        PreparedStatement ps = conn.prepareStatement(

            "INSERT INTO customers (cust_name, phone, address) VALUES (?, ?, ?)");

        ps.setString(1, name);

        ps.setString(2, phone);

        ps.setString(3, address);

        ps.executeUpdate();

        JOptionPane.showMessageDialog(this, "Customer Added Successfully!");

    } catch (Exception e) {

        JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());

    }

}

private void deleteCustomer(int id) {
    try {
        PreparedStatement ps = conn.prepareStatement("DELETE FROM customers WHERE cust_id = ?");
        ps.setInt(1, id);
        ps.executeUpdate();
        JOptionPane.showMessageDialog(this, "Customer Deleted Successfully!");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());
    }
}
```

4.5 Sales and Billing Module

This module manages all sales transactions. It calculates the total amount, updates stock, and records the sale in the database.

```
private void makeSale(String customer, String medicine, int quantity) {
    try {
        PreparedStatement ps1 = conn.prepareStatement(
            "SELECT med_id, price, quantity FROM medicines WHERE med_name = ?");
        ps1.setString(1, medicine);
        ResultSet rs = ps1.executeQuery();
        if (rs.next()) {
            int medId = rs.getInt("med_id");
            double price = rs.getDouble("price");
            int stock = rs.getInt("quantity");
            if (quantity > stock) {
```

```
        JOptionPane.showMessageDialog(this, "Insufficient Stock!");

        return;

      }

      double total = price * quantity;

      PreparedStatement ps2 = conn.prepareStatement("INSERT INTO sales (cust_id,
      med_id, quantity, total_amount, sale_date) VALUES ((SELECT cust_id FROM
      customers WHERE cust_name = ?), ?, ?, ?, CURDATE())");

      ps2.setString(1, customer);

      ps2.setInt(2, medId);

      ps2.setInt(3, quantity);

      ps2.setDouble(4, total);

      ps2.executeUpdate();

       PreparedStatement ps3 = conn.prepareStatement("UPDATE medicines SET
      quantity = quantity - ? WHERE med_id = ?");

      ps3.setInt(1, quantity);

      ps3.setInt(2, medId);

      ps3.executeUpdate();

      JOptionPane.showMessageDialog(this, "Sale Completed Successfully! Total: ₹" +
total);

    }

  } catch (Exception e) {

    JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());

  }

}
```

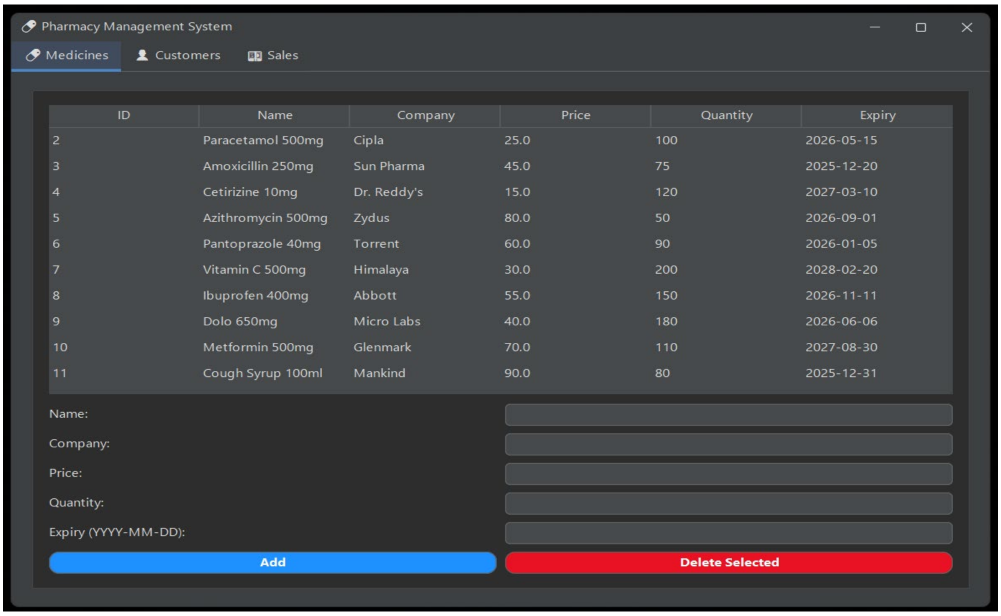
4.6 User Interface (UI) Module

This module provides a simple graphical interface using Java Swing. It includes tabbed sections for medicines, customers, and sales.

```
public PharmacyApp() {

   setTitle("Pharmacy Management System");

   setSize(950, 600);

   setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   setLocationRelativeTo(null);

   JTabbedPane tabs = new JTabbedPane();

   tabs.add("Medicines", medicinePanel());

   tabs.add("Customers", customerPanel());

   tabs.add("Sales", salesPanel());

   add(tabs);

   setVisible(true);

}
```

CHAPTER 5

SCREENSHOTS



ID	Name	Company	Price	Quantity	Expiry
2	Paracetamol 500mg	Cipla	25.0	100	2026-05-15
3	Amoxicillin 250mg	Sun Pharma	45.0	75	2025-12-20
4	Cetirizine 10mg	Dr. Reddy's	15.0	120	2027-03-10
5	Azithromycin 500mg	Zydus	80.0	50	2026-09-01
6	Pantoprazole 40mg	Torrent	60.0	90	2026-01-05
7	Vitamin C 500mg	Himalaya	30.0	200	2028-02-20
8	Ibuprofen 400mg	Abbott	55.0	150	2026-11-11
9	Dolo 650mg	Micro Labs	40.0	180	2026-06-06
10	Metformin 500mg	Glenmark	70.0	110	2027-08-30
11	Cough Syrup 100ml	Mankind	90.0	80	2025-12-31

Name:

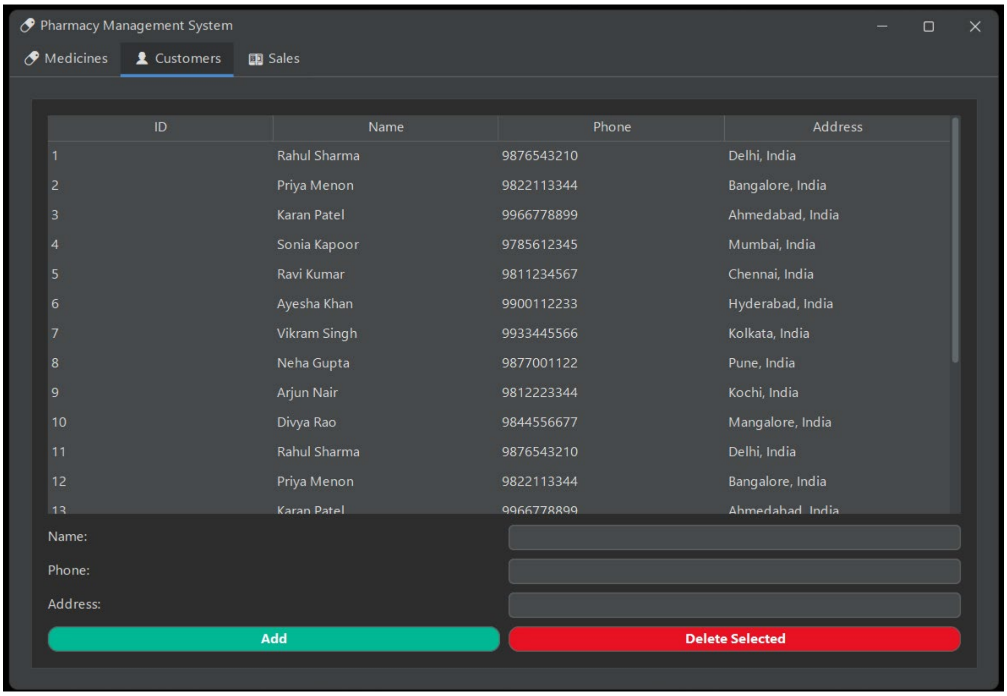
Company:

Price:

Quantity:

Expiry (YYYY-MM-DD):

Fig.5.1 Medicine Stock Interface Screenshot



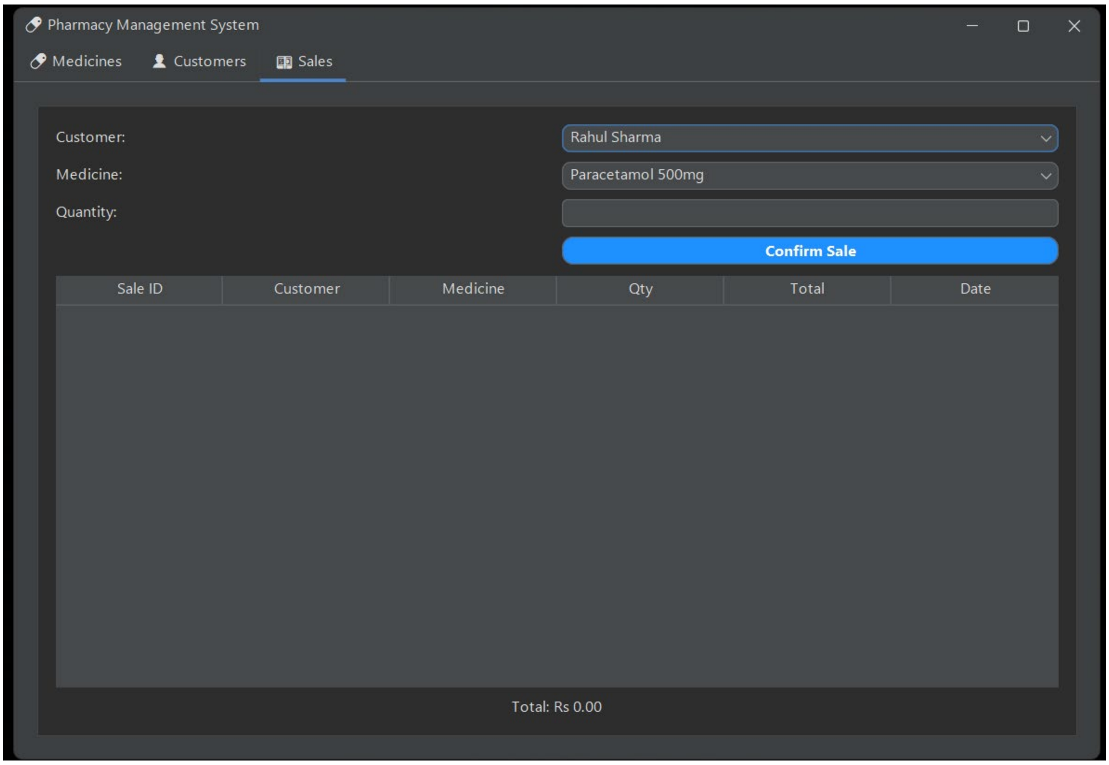
ID	Name	Phone	Address
1	Rahul Sharma	9876543210	Delhi, India
2	Priya Menon	9822113344	Bangalore, India
3	Karan Patel	9966778899	Ahmedabad, India
4	Sonia Kapoor	9785612345	Mumbai, India
5	Ravi Kumar	9811234567	Chennai, India
6	Ayesha Khan	9900112233	Hyderabad, India
7	Vikram Singh	9933445566	Kolkata, India
8	Neha Gupta	9877001122	Pune, India
9	Arjun Nair	9812223344	Kochi, India
10	Divya Rao	9844556677	Mangalore, India
11	Rahul Sharma	9876543210	Delhi, India
12	Priya Menon	9822113344	Bangalore, India
13	Karan Patel	9966778899	Ahmedabad, India

Name:

Phone:

Address:

Fig.5.2 Customer List Screenshot



Customer: Rahul Sharma

Medicine: Paracetamol 500mg

Quantity:

Sale ID	Customer	Medicine	Qty	Total	Date

Total: Rs 0.00

Fig.5.3 Sales Dashboard Screenshot

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

The Pharmacy Management System was successfully developed and implemented using Java Swing for the front-end and MySQL for the back-end database. The system fulfills its main objective of simplifying and automating the daily tasks of a pharmacy, such as managing medicines, customer records, and sales transactions. Through the use of JDBC, the system ensures smooth and secure communication between the user interface and the database.

This project demonstrates how database-driven desktop applications can efficiently handle real-time data operations. The system minimizes manual errors, reduces workload, and helps in maintaining accurate records of stock and sales. It provides an effective solution for small and medium-sized pharmacies to manage their operations in an organized and reliable way.

6.2 Future Enhancements

Although the system meets all the basic functional requirements, there is still room for improvement and extension.

Future versions of the Pharmacy Management System can include the following enhancements:

1. User Authentication:
   Add a secure login system with different roles such as Admin, Pharmacist, and Staff.
2. Barcode Scanning:
   Integrate barcode or QR code scanning to make billing faster and error-free.
3. Automated Notifications:
   Generate alerts for low stock levels or medicines that are about to expire.
4. Report Generation:
   Include features to generate daily, weekly, and monthly sales and stock reports in PDF or Excel format.
5. Cloud Integration:
   Host the database on the cloud to allow access from multiple devices or pharmacy branches.
6. Data Backup and Restore:
   Implement automatic database backups to prevent data loss due to unexpected failures.

REFERENCES

1. Oracle Corporation, “Java Swing Tutorial”, https://docs.oracle.com/javase/tutorial/uiswing/
2. MySQL AB, “MySQL 8.0 Reference Manual”, https://dev.mysql.com/doc/refman/8.0/en/
3. FormDev Software, “FlatLaf Look and Feel for Swing”, https://www.formdev.com/flatlaf/
4. GeeksforGeeks, “JDBC – Java Database Connectivity”, https://www.geeksforgeeks.org/jdbc-java-database-connectivity/
5. TutorialsPoint, “Java Swing Components and Layouts”, https://www.tutorialspoint.com/java_swing/
6. W3Schools, “MySQL Tutorial”, https://www.w3schools.com/mysql/
7. Oracle University, “Java Database Programming Using JDBC”, https://www.oracle.com/java/technologies/javase/jdbc.html
8. Stack Overflow Discussions, “Common JDBC and Swing Implementation Queries”, https://stackoverflow.com/
9. Official MySQL Connector/J Documentation, https://dev.mysql.com/doc/connector-j/8.0/en/