PHARMACY MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

PRIYANKA I (8115U23EC078)

in partial fulfillment of requirements for the award of the course

EGB1201 - JAVA PROGRAMMING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

DECEMBER - 2024

K. RAMAKRISHNAN COLLEGE OF ENGINEERING (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **“PHARMACY MANAGEMENT SYSTEM ”** is the bonafide work of **PRIYANKA I (8115U23EC078)** who carriedout the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE	**SIGNATURE**
Dr. T. M. NITHYA, M.E.,Ph.D.,	MR.K.SWAMINATHANM.E.,MBA.,(Ph.D.,)
HEAD OF THE DEPARTMENT	**SUPERVISOR**
ASSOCIATE PROFESSOR	ASSISTANT PROFESSOR
Department of CSE	Department of CSE
K.Ramakrishnan College of Engineering (Autonomous)	K.Ramakrishnan College of Engineering (Autonomous)
Samayapuram–621112.	Samayapuram–621112.

Submitted for the viva-voce examination held on …………….

INTERNAL EXAMINER EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **“PHARMACY MANAGEMENT SYSTEM ”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **EGB1201 - JAVA PROGRAMMING.**

.

Signature

____

PRIYANKA I

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Engineering (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. D. SRINIVASAN, B.E, M.E., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. M. NITHYA, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **MR.K.SWAMINATHAN,M.E.,MBA.,(Ph.D.,),Department of COMPUTER SCIENCE AND ENGINEERING,** for his incalculable,suggestions,creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To achieve a prominent position among the top technical institutions.

MISSION OF THE INSTITUTION

- M1: To bestow standard technical education par excellence through state of the art infrastructure, competent faculty and high ethical standards.
- M2: To nurture research and entrepreneurial skills among students in cutting edge technologies.
- M3: To provide education for developing high-quality professionals to transform the society.

VISION OF DEPARTMENT

To create eminent professionals of Computer Science and Engineering by imparting quality education.

MISSION OF DEPARTMENT

M1: To provide technical exposure in the field of Computer Science and Engineering through state of the art infrastructure and ethical standards.

M2: To engage the students in research and development activities in the field of Computer Science and Engineering.

M3: To empower the learners to involve in industrial and multi-disciplinary projects for addressing the societal needs.

PROGRAM EDUCATIONAL OBJECTIVES

Our graduates shall

PEO1: Analyse, design and create innovative products for addressing social needs.

PEO2: Equip themselves for employability, higher studies and research.

PEO3: Nurture the leadership qualities and entrepreneurial skills for their successful career.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Apply the basic and advanced knowledge in developing software, hardware and firmware solutions addressing real life problems.

- **PSO2:** Design, develop, test and implement product-based solutions for their career enhancement.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Pharmacy Management System is a Java-based application designed to automate and optimize essential pharmacy operations, such as inventory management and prescription processing. It enables users to add new medicines, update stock levels, and efficiently manage prescription dispensing, significantly reducing manual errors. By leveraging object-oriented programming and Java collections, the system enhances accuracy, ensures data integrity, and improves the overall workflow within the pharmacy.The system is designed to provide an intuitive user interface that simplifies daily tasks, such as tracking stock levels, generating alerts for low inventory, and maintaining a detailed record of prescriptions. These features not only streamline operations but also enhance customer satisfaction by ensuring timely and accurate service.A key strength of the project is its scalability. The modular architecture supports future enhancements, such as the integration of online prescription orders, automated billing, and real-time notifications. This flexibility makes the Pharmacy Management System a reliable, cost-effective solution for modern pharmacy management, laying the groundwork for advanced features that can further improve operational efficiency and customer experience.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The Pharmacy Management System is a Java-based application designed to automate inventory management and prescription processing for pharmacies. It enables users to add medicines, track stock, and dispense prescriptions efficiently. By leveraging object-oriented programming and collections, the system improves accuracy, reduces errors, and enhances customer satisfaction. This project lays the groundwork for scalable solutions with potential for advanced features like online orders and billing integration.	**PO 3** **PO 4** **PO 7** **PO 9**	**PSO3** **PSO5** **PSO8**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	viii
1	**INTRODUCTION**	1
	1.1 Objective	1
	1.2 Overview	1
	1.3 Java Programming concepts	2
2	**PROJECT METHODOLOGY**	3
	2.1 Proposed Work	3
	2.2 Block Diagram	3
3	**MODULE DESCRIPTION**	4
	3.1 Main Menu Module:	4
	3.2 Inventory Management Module	4
	3.3 Dispense Medication Module	4
	3.4 View Inventory Module	4
	3.5 Transaction Record Module	4
4	**CONCLUSION & FUTURE SCOPE**	5
	4.1 Conclusion	5
	4.2 Future Scope	5
	APPENDIX A (SOURCE CODE)	6
	APPENDIX B (SCREENSHOTS)	10
	REFERENCES	11

CHAPTER - 1
INTRODUCTION

1.1 Objective

1. Automate the process of managing medicines and prescriptions.
2. Track and update stock accurately.
3. Make it easy for users to add, view, and dispense medicines.
4. Keep a clear record of all transactions.
5. Reduce errors in prescription handling and stock management.
6. Provide a simple and user-friendly interface.
7. Ensure the system can be expanded with new features in the future.

1.2 Overview

The Pharmacy Management System is a Java-based application designed to automate key pharmacy operations such as managing medicines, verifying prescriptions, and tracking inventory. It simplifies the process by allowing users to add medicines, check stock, and dispense prescriptions efficiently.Using object-oriented programming concepts like classes, objects, and collections, the system ensures accurate inventory management and reliable record-keeping. The system’s modular design includes features like an interactive menu, stock validation, and transaction logging. It enhances pharmacy efficiency, reduces manual errors, and provides a foundation for future enhancements such as online orders and billing integration.This project demonstrates Java's ability to create user-friendly, scalable solutions for modern pharmacy management needs.

1.3 Java Programming Concepts

1. Classes and Objects:

A class defines the structure of objects. The Medicine class represents medicines with attributes like name, quantity, and price. Each medicine is an object.

2. ArrayList:

Used for storing and managing medicines dynamically, allowing for easy additions, removals, and updates.

3. Menu-Driven Programming:

The system presents a menu with options like dispensing medicines or viewing inventory, allowing users to interact with the program.

4. Loops:

do-while loop keeps showing the menu until the user exits.

for-each loop iterates over the medicines to display or update their details.

5. Conditional Statements:

if-else statements check conditions like stock availability and user input validit**y.**

6. Methods:

Methods perform specific tasks such as adding medicines or dispensing prescriptions, keeping the code organized.

7. Exception Handling:

Java's exception handling (try-catch) can be used to manage errors like invalid input or stock issues.

CHAPTER 2

PROJECT METHODOLOGY

2.1Proposed Work

1. Inventory Management: Tracks and updates medicine stock.

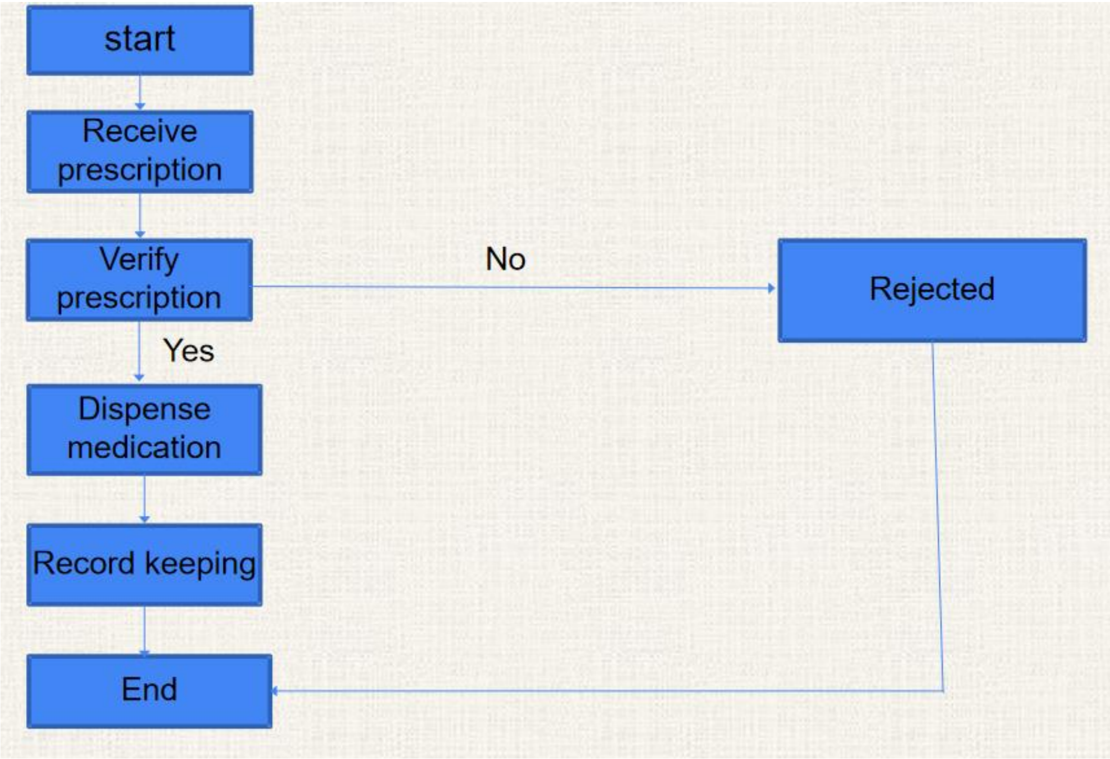
2. Prescription Verification: Ensures valid prescriptions and availability of medicines.

3. Medicine Dispensing: Dispenses medicines and updates stock levels.

4. Transaction Logging: Records all dispensing activities for future reference.

5. User Interface: A menu-driven interface for easy navigation.

2.2 Block Diagram



CHAPTER- 3

MODULE DESCRIPTION

3.1 Main Menu Module:

Description: Displays the main options to the user, such as dispensing medicines, viewing inventory, or managing transactions. It directs the user to the appropriate functionality based on their selection.

3.2 Inventory Management Module:

Description: Manages the list of medicines in the pharmacy. It allows the user to add, remove, and update the medicines, keeping track of stock levels.

3.3Dispense Medication Module:

Description: Verifies the requested medicine and quantity availability in the inventory. If available, it dispenses the medication, updates the stock, and logs the transaction.

3.4 View Inventory Module:

Description: Displays all available medicines along with their name, quantity, and price. It also informs the user if the inventory is empty or insufficient.

3.5Transaction Record Module:

Description: Keeps a record of all dispensed medicines and their respective details. It allows the user to view past transactions.

CHAPTER - 4

CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION:

The pharmacy management system successfully implements basic functionalities for managing pharmacy operations, such as dispensing medications, tracking inventory, and recording transactions. By allowing users to interact with the system to check medicine availability, dispense medication, and view transaction history, the program ensures an efficient, straightforward approach to pharmacy management. Error handling, such as rejecting invalid requests for out-of-stock medicines or incorrect inputs, ensures smooth and predictable operation. This system is an excellent starting point for small-scale pharmacies or as a learning tool for understanding inventory and transaction management.

4.2 FUTURE SCOPE:

1. Inventory Expansion: Add features to update or add new medicines to the inventory dynamically.

2. **Report Generation**: Implement detailed transaction reports, including daily, weekly, or monthly summaries.

3. **User Authentication:** Introduce roles like pharmacist and admin with access controls for security.

4. **Batch and Expiry Tracking:** Track medicine batches and expiry dates to prevent dispensing expired medicines.

5. **Graphical User Interface (GUI)**: Develop a user-friendly GUI for better interaction and usability.

APPENDIX A (SOURCE CODE)

```
import java.util.ArrayList;
import java.util.Scanner;
class Main {
    static class Medicine {
        String name;
        int quantity;
        double price;
        Medicine(String name, int quantity, double price) {
            this.name = name;
            this.quantity = quantity;
            this.price = price;
        }
@Override
        public String toString() {
            return "Name: " + name + ", Quantity: " + quantity + ", Price: ₹" + price;
        }
    }
public static void main(String[] args) {
        ArrayList<Medicine> inventory = new ArrayList<>();
        ArrayList<String> transactions = new ArrayList<>();
        Scanner scanner = new Scanner(System.in);
// Preloaded inventory for testing
        inventory.add(new Medicine("Paracetamol", 50, 10.0));
        inventory.add(new Medicine("Ibuprofen", 30, 15.0));
        inventory.add(new Medicine("Amoxicillin", 20, 25.0));
        do {
```

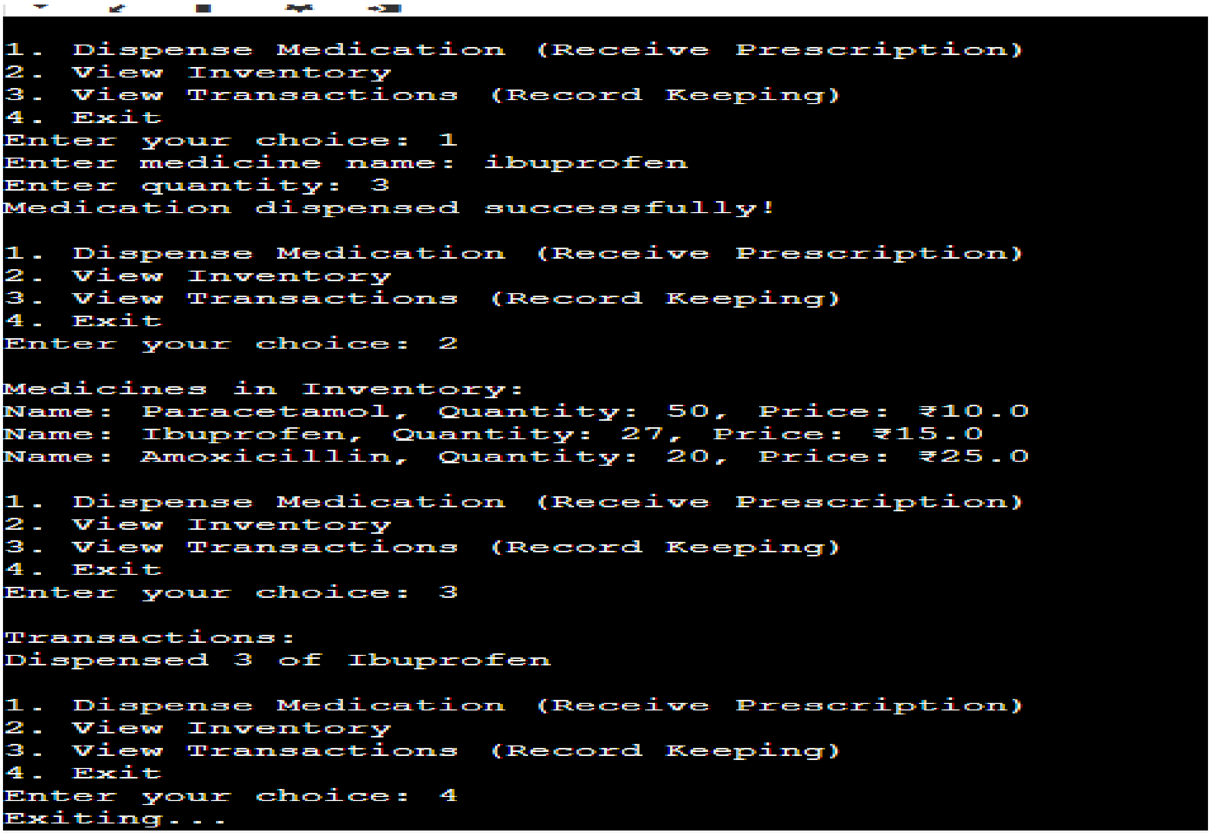
```
      System.out.println("\n1. Dispense Medication (Receive
    Prescription)");
      System.out.println("2. View Inventory");
      System.out.println("3. View Transactions (Record Keeping)");
      System.out.println("4. Exit");
      System.out.print("Enter your choice: ");
    choice = scanner.nextInt();
      scanner.nextLine(); // Consume newline character
      switch (choice) {
        case 1:
          // Receive and verify prescription
          System.out.print("Enter medicine name: ");
          String name = scanner.nextLine();
          System.out.print("Enter quantity: ");
          int quantity = scanner.nextInt();
          boolean found = false;
          for (Medicine med : inventory) {
        if (med.name.equalsIgnoreCase(name)) {
       ound = true;
                     if (med.quantity >= quantity) {
                       // Dispense medication
                       med.quantity -= quantity;
       System.out.println("Medication dispensed successfully!");
       String transaction = "Dispensed " + quantity + " of " +
med.name;
        transactions.add(transaction);
          } else {
         System.out.println("Insufficient stock. Request rejected.");
                     }
```

```
   break;
                       }
                   }
                   if (!found) {
 System.out.println("Medicine not found. Request rejected.");
                   }
                   break;
 case 2:
                   // View inventory
                   System.out.println("\nMedicines in Inventory:");
                   if (inventory.isEmpty()) {
System.out.println("No medicines available.");
                   } else {
                       inventory.forEach(System.out::println);
                   }
                   break;
 case 3:
                   // View transactions (record keeping)
                   System.out.println("\nTransactions:");
                   if (transactions.isEmpty()) {
                 System.out.println("No transactions recorded.");
  } else {
         transactions.forEach(System.out::println);
                 } break;
case 4:
                   // Exit the program
                   System.out.println("Exiting...");
                   break;
               default:
```

```
                                System.out.println("Invalid choice. Please try
again.");
                         }
                     } while (choice != 4);

                     scanner.close();
                 }
              }
```

APPENDIX B(SCREEN SHOTS)



```
1. Dispense Medication (Receive Prescription)
2. View Inventory
3. View Transactions (Record Keeping)
4. Exit
Enter your choice: 1
Enter medicine name: ibuprofen
Enter quantity: 3
Medication dispensed successfully!

1. Dispense Medication (Receive Prescription)
2. View Inventory
3. View Transactions (Record Keeping)
4. Exit
Enter your choice: 2

Medicines in Inventory:
Name: Paracetamol, Quantity: 50, Price: ₹10.0
Name: Ibuprofen, Quantity: 27, Price: ₹15.0
Name: Amoxicillin, Quantity: 20, Price: ₹25.0

1. Dispense Medication (Receive Prescription)
2. View Inventory
3. View Transactions (Record Keeping)
4. Exit
Enter your choice: 3

Transactions:
Dispensed 3 of Ibuprofen

1. Dispense Medication (Receive Prescription)
2. View Inventory
3. View Transactions (Record Keeping)
4. Exit
Enter your choice: 4
Exiting...
```

REFERENCE:

1. **"Java Documentation." Oracle, https://docs.oracle.com/javase/8/docs/api/.**

- Used to understand Java classes like ArrayList and methods such as nextInt() and nextLine().

2. "**ArrayList in Java." GeeksforGeeks, https://www.geeksforgeeks.org/arraylist-in-java/.**

- Explains how to use ArrayList for managing inventory and transaction records.

3**. "Scanner Class in Java." W3Schools, https://www.w3schools.com/java/ref_java_util_scanner.asp.**

- Describes how to use the Scanner class for taking user input.

4. **"Introduction to Java Programming." Daniel Liang, Pearson Education, 10th edition, 2019.**

- A textbook offering a comprehensive introduction to Java programming concepts.

5. "**Java Programming Basics." Tutorialspoint, https://www.tutorialspoint.com/java/index.htm.**

- A resource covering fundamental Java programming concepts and syntax.