

STOCK MANAGEMENT SYSTEM



A PROJECT REPORT

Submitted by

MOHAMED FIRDOUS S (2303811724321067)

in partial fulfillment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

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

SAMAYAPURAM – 621 112

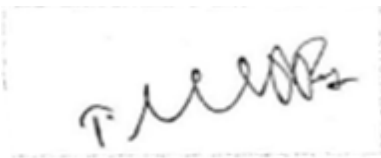
DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

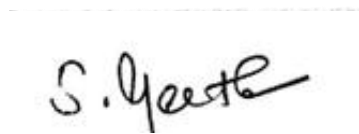
BONAFIDE CERTIFICATE

Certified that this project report on “ **STOCK MANAGEMENT SYSTEM**” is the bonafide work of **MOHAMED FIRDOUS S (2303811724321067)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



Signature


Dr.T AVUDAIAPPAN M.E.,Ph.D.,
HEAD OF THE DEPARTMENT,
Department of Artificial Intelligence,
K. Ramakrishnan College of Technology,
Samayapuram, Trichy -621 112.




Signature

Mrs. S.GEETHA M.E.,
SUPERVISOR,
Department of Artificial Intelligence,
K. Ramakrishnan College of Technology,
Samayapuram, Trichy -621 112.

Submitted for the viva-voce examination held on 3.12.24



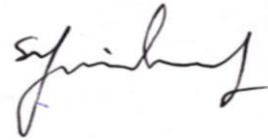
INTERNAL EXAMINER



EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**STOCK MANAGEMENT SYSTEM**” is the result of original work done by me and best of my knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING**.



SIGNATURE

MOHAMED FIRDOUS S

Place: Samayapuram

Date: 3/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, **“K. Ramakrishnan College of Technology (Autonomous)”**, for providing us with the opportunity to do this project.

I extend our sincere acknowledgement and appreciation to the esteemed and honourable Chairman, **Dr. K. RAMAKRISHNAN, B.E.**, for having provided 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 our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.**, Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D.**, Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs.S.GEETHA M.E.**, Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the 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 serve the society by offering top-notch technical education on par with global standards.

MISSION OF THE INSTITUTION

- Be a centre of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all- round personalities respecting moral and ethical values.

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conducive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

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.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

The Stock Management System is an inventory management solution designed to automate the tracking, monitoring, and updating of stock levels in real time. It addresses key challenges such as stockouts, overstocking, and inefficient manual processes, ensuring optimal inventory control and streamlined operations. Built using Java and incorporating AWT for the user interface, the system provides an interactive and user-friendly platform for managing products, processing orders, replenishing stock, and generating reports. The system organizes products using a catalog, storing essential details such as product name, ID, stock quantity, and price. Core functionalities include adding new products, updating inventory levels, placing orders with stock validation, generating invoices, and creating detailed reports for analysis. An integrated decision-making mechanism provides stock alerts, ensuring timely replenishment and maintaining inventory efficiency.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	viii
1	INTRODUCTION	1
	1.1 INTRODUCTION	1
	1.2 OBJECTIVE	1
2	PROJECT METHODOLOGY	2
	2.1 PROPOSED WORK	2
	2.2 BLOCK DIAGRAM	2
3	JAVA PROGRAMMING CONCEPTS	3
	3.1 COLLECTION FRAMEWORK (HASHMAP)	3
	3.2 EVENT-HANDLING	3
4	MODULE DESCRIPTION	4
	4.1 PRODUCT MANAGEMENT	4
	4.2 STOCK UPDATE	5
	4.3 ORDER MANAGEMENT	6
	4.4 REPORT GENERATION	7
5	CONCLUSION	8
	REFERENCES	9
	APPENDICES	10
	Appendix A – Source code	10
	Appendix B – Screen shots	16

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The **Stock Management System** is a Java-based solution designed to automate inventory processes, addressing challenges like stockouts and inefficiencies. With features like product management, order processing, stock replenishment, and report generation, it ensures real-time tracking and streamlined operations. The system offers a user-friendly GUI built with AWT, integrating core Java concepts to provide accuracy, scalability, and enhanced decision-making. This tool is ideal for businesses seeking efficient and reliable inventory management.

1.2 OBJECTIVE

- a) **Real-time Inventory Tracking:** To ensure stock levels are always up-to-date, reducing errors and discrepancies.
- b) **Streamlined Operations:** Automating tasks like product management, stock updates, and order processing to minimize manual intervention and save time.
- c) **Improved Decision-Making:** Providing detailed stock and sales reports to help businesses make informed decisions about inventory, sales trends, and replenishment needs
- d) **Enhanced Accuracy:** Eliminating manual errors by validating stock quantities before transactions and ensuring precise billing.

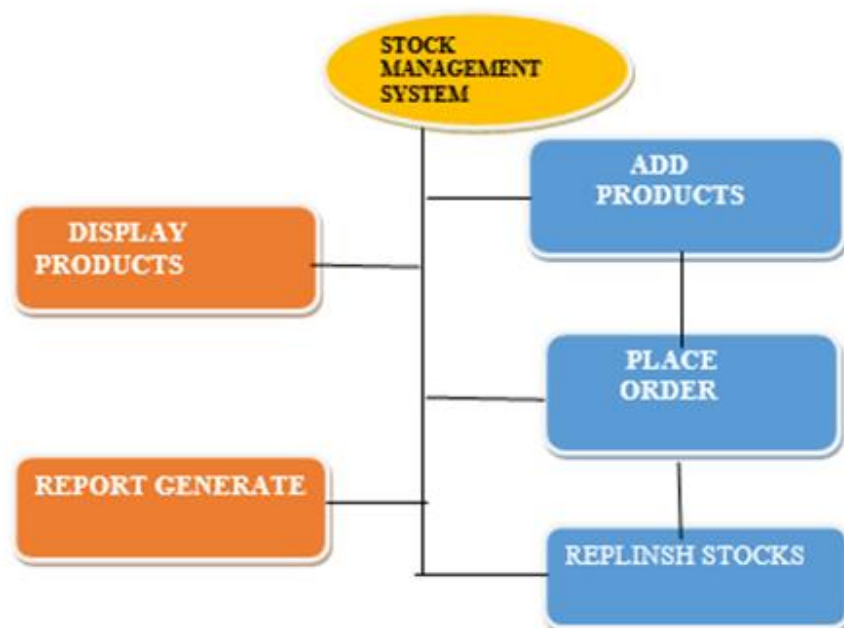
CHAPTER 2

PROJECT METHODOLOGY

2.1 PROPOSED WORK

The proposed **Stock Management System** automates inventory operations using Java and AWT. It includes modules for real-time inventory tracking, order management, stock replenishment, and report generation. The system ensures accurate stock updates, prevents stockouts, and simplifies decision-making with detailed analytics. Designed for scalability and efficiency, it provides a user-friendly interface, enabling businesses to streamline inventory processes, enhance accuracy, and improve operational efficiency.

2.2 BLOCK DIAGRAM



CHAPTER 3

JAVA PROGRAMMING CONCEPTS

3.1.COLLECTION FRAMEWORK (HASHMAP):

The **HashMap** collection is utilized in the system to efficiently store and manage the catalog of products. Each product is uniquely identified by a **product ID**, which serves as the **key**, while the **Product** object itself is stored as the **value**. This provides an effective way to store, retrieve, and update product information based on their IDs. The **HashMap** allows for constant-time complexity ($O(1)$) for operations like searching for products, which makes the system responsive even when dealing with a large number of products. This approach ensures that inventory management is fast and scalable.

3.2. EVENT-HANDLING :

The **Stock Management System** employs **event-driven programming** through the use of **AWT (Abstract Window Toolkit)** components like buttons, text fields, and labels. **ActionListener** interfaces are used to handle user interactions, such as clicking buttons to add products, place orders, or generate reports. When an event (such as a button press) occurs, the corresponding method is executed, ensuring that the system responds dynamically to user input. This approach allows for an interactive and user-friendly experience, where each action triggers specific functionality within the system.

CHAPTER 4

MODULE DESCRIPTION

4.1 PRODUCT MANAGEMENT:

Purpose:

The Product Management Module is responsible for managing the details of all products in the stock management system. It allows users to add, edit, and view product details, ensuring that the catalog is up-to-date and accurate.

Description:

This module maintains a catalog of all products, including attributes such as product name, ID, stock quantity, and price. It ensures that product information can be added or updated easily. Product objects are created to encapsulate these details, enabling efficient management of inventory data. For instance, adding a new product involves specifying its attributes, while editing allows users to modify existing records. The module also facilitates viewing the product catalog, enabling businesses to have a clear picture of their inventory.

Key Functions:

1. **Add New Products:** Insert new product details into the system with attributes like name, ID, and price.
2. **Edit Product Information:** Update existing product details in case of changes in price, name, or other attributes.
3. **Display Catalog:** View a list of all available products in the system.

4.2 STOCK UPDATE:

Purpose:

The Stock Update Module ensures real-time updates to stock levels, preventing overstock or stockout situations.

Description:

This module handles stock replenishment and deduction based on business operations like sales and purchases. For each product, the module tracks stock levels and ensures accuracy. When a sale is made, the system decreases the product's stock. Similarly, when new stock arrives, it updates the inventory. For instance, if 10 units of a product are sold, the stock quantity is automatically reduced by 10. This real-time update mechanism helps businesses maintain accurate stock levels and prevents discrepancies.

Key Functions:

1. **Stock Deduction:** Automatically reduce stock levels when orders are placed.
2. **Stock Replenishment:** Update stock levels when new inventory is added.
3. **Track Stock Movements:** Record and display stock changes for auditing and decision-making.

4.3 ORDER MANAGEMENT:

Purpose:

The Order Management Module facilitates the smooth processing of customer orders while maintaining inventory accuracy.

Description:

This module ensures that customer orders are processed efficiently. It verifies if the requested quantity of a product is available before confirming an order. If sufficient stock is present, the module deducts the stock quantity and updates the system. If stock is insufficient, it notifies the user, preventing order errors. For example, if a customer requests 5 units of a product, the system checks availability and either confirms or rejects the order.

Key Functions:

1. **Order Verification:** Check if sufficient stock is available for a customer order.
2. **Process Orders:** Deduct stock and confirm successful order placement.
3. **Notification System:** Notify users in case of insufficient stock.

4.4 REPORT GENERATION:

Purpose:

The Report Generation Module provides detailed insights into stock and sales performance to support business decisions.

Description:

This module generates comprehensive reports on inventory and sales. It provides a summary of product details, including stock levels, prices, and sales figures. The module also tracks historical stock changes and generates reports to identify trends. For example, a monthly report might show which products are top-selling and which need replenishment.

Key Functions:

1. **Inventory Reports:** Summarize current stock levels and prices.
2. **Sales Reports:** Provide insights into sales performance by product and date.
3. **Trend Analysis:** Identify patterns in stock and sales data for forecasting.

CHAPTER 5

CONCLUSION

The **Stock Management System** built using Java provides a comprehensive solution for managing inventory, placing orders, and generating reports in real-time. By utilizing the Object-Oriented Programming (OOP) principles, the system ensures high modularity and reusability, making it easy to scale and maintain. The **Product Management** module efficiently stores and manages product details, while the **Stock Update** and **Order Management** modules handle inventory tracking and order processing, ensuring smooth operations.

The **Report Generation** module offers valuable insights into the system's performance, helping businesses track sales and stock levels, which is crucial for informed decision-making. Furthermore, the **User Interface** module enhances the user experience by offering an intuitive and interactive platform for customers and administrators to manage products and view reports.

By leveraging **event-driven programming** and **collections** such as **HashMap**, the system ensures quick retrieval and updating of product information, while **exception handling** ensures robust error management and smooth execution. The use of Java's AWT components for the graphical user interface (GUI) further adds to the system's usability, providing a responsive and dynamic environment for users.

REFERENCES:

- a) **GeeksforGeeks - Java Programming** (<https://www.geeksforgeeks.org/java/>).
 - Provides practical examples and in-depth articles on Java programming, including collections and event handling.
- b) **Head First Java** by Kathy Sierra and Bert Bates.
 - Provides a solid foundation in Java programming, including OOP concepts.
- c) **Java: The Complete Reference** by Herbert Schildt.
 - Covers detailed explanations of Java programming features and libraries used in the project.
- d) **Java Programming Tutorials - YouTube Channel** (<https://www.youtube.com/user/thenewboston>).
 - Offers video tutorials on Java programming concepts and GUI development.
- e) **Oracle Java Documentation** (<https://docs.oracle.com/en/java/>).
 - Official documentation for Java APIs, including AWT and collections.

APPENDICES

APPENDIX A – SOURCE CODE

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;

class Product {
    String productName;
    int productId;
    int stockQuantity;
    double price;

    public Product(String productName, int productId, int stockQuantity, double
price) {
        this.productName = productName;
        this.productId = productId;
        this.stockQuantity = stockQuantity;
        this.price = price;
    }

    public void updateStock(int quantity) {
        this.stockQuantity += quantity;
    }

    public boolean isAvailable(int quantity) {
        return stockQuantity >= quantity;
    }
}
```

```

@Override
public String toString() {
    return "Product: " + productName + " | ID: " + productId + " | Stock: " +
stockQuantity + " | Price: $" + price;
}
}

```

```

class StockManagementSystem extends Frame implements ActionListener {
    Map<Integer, Product> productCatalog = new HashMap<>();
    TextArea displayArea = new TextArea(20, 50);
    TextField productIdField = new TextField(10);
    TextField quantityField = new TextField(10);
    TextField nameField = new TextField(15);
    TextField priceField = new TextField(10);
    Button displayButton = new Button("Display Products");
    Button orderButton = new Button("Place Order");
    Button replenishButton = new Button("Replenish Stock");
    Button reportButton = new Button("Generate Report");
    Button addButton = new Button("Add Product");

    public StockManagementSystem() {
        setLayout(new FlowLayout());
        setTitle("Stock Management System");

        // Add UI components
        add(new Label("Product ID:"));
        add(productIdField);
        add(new Label("Quantity:"));

```

```

add(quantityField);
add(new Label("Product Name:"));
add(nameField);
add(new Label("Price:"));
add(priceField);
add(addButton);
add(orderButton);
add(replenishButton);
add(displayButton);
add(reportButton);
add(displayArea);

// Add Action Listeners
addButton.addActionListener(this);
displayButton.addActionListener(this);
orderButton.addActionListener(this);
replenishButton.addActionListener(this);
reportButton.addActionListener(this);

// Sample Data
addProduct("Laptop", 101, 50, 799.99);
addProduct("Smartphone", 102, 150, 499.99);
addProduct("Tablet", 103, 80, 299.99);

setSize(600, 500);
setVisible(true);

// Close button functionality
addWindowListener(new WindowAdapter() {

```

```

        public void windowClosing(WindowEvent we) {
            System.exit(0);
        }
    });
}

```

```

public void addProduct(String name, int id, int quantity, double price) {
    Product product = new Product(name, id, quantity, price);
    productCatalog.put(id, product);
}

```

```

public void displayProducts() {
    displayArea.setText("");
    for (Product product : productCatalog.values()) {
        displayArea.append(product + "\n");
    }
}

```

```

public void placeOrder(int productId, int quantity) {
    Product product = productCatalog.get(productId);
    if (product != null) {
        if (product.isAvailable(quantity)) {
            product.updateStock(-quantity);
            displayArea.setText("Order placed successfully for " + quantity + " of "
+ product.productName);
        } else {
            displayArea.setText("Insufficient stock for the order.");
        }
    } else {

```

```

        displayArea.setText("Product not found.");
    }
}

```

```

public void replenishStock(int productId, int replenishmentQuantity) {
    Product product = productCatalog.get(productId);
    if (product != null) {
        product.updateStock(replenishmentQuantity);
        displayArea.setText("Stock replenished for " + product.productName + "
by " + replenishmentQuantity + " units.");
    } else {
        displayArea.setText("Product not found.");
    }
}

```

```

public void generateReport() {
    displayArea.setText("Stock Report:\n");
    for (Product product : productCatalog.values()) {
        displayArea.append(product + "\n");
    }
}

```

@Override

```

public void actionPerformed(ActionEvent e) {
    try {
        if (e.getSource() == displayButton) {
            displayProducts();
        } else if (e.getSource() == orderButton) {
            int id = Integer.parseInt(productIdField.getText());

```

```

        int qty = Integer.parseInt(quantityField.getText());
        placeOrder(id, qty);
    } else if (e.getSource() == replenishButton) {
        int id = Integer.parseInt(productIdField.getText());
        int qty = Integer.parseInt(quantityField.getText());
        replenishStock(id, qty);
    } else if (e.getSource() == reportButton) {
        generateReport();
    } else if (e.getSource() == addButton) {
        String name = nameField.getText();
        int id = Integer.parseInt(productIdField.getText());
        int qty = Integer.parseInt(quantityField.getText());
        double price = Double.parseDouble(priceField.getText());
        addProduct(name, id, qty, price);
        displayArea.setText("Product added successfully.");
    }
} catch (Exception ex) {
    displayArea.setText("Error: " + ex.getMessage());
}
}
}

```

```

public class StockManagementApp {
    public static void main(String[] args) {
        new StockManagementSystem();
    }
}

```


APPENDIX B - SCREENSHOTS

1.DISPLAY PRODUCTS

Product ID: Quantity: Product Name:

Price:

Product: Laptop | ID: 101 | Stock: 50 | Price: \$799.99
Product: Smartphone | ID: 102 | Stock: 150 | Price: \$499.99
Product: Tablet | ID: 103 | Stock: 80 | Price: \$299.99

Product ID: Quantity: Product Name:

Price:

Stock Report:
Product: Laptop | ID: 101 | Stock: 50 | Price: \$799.99
Product: Smartphone | ID: 102 | Stock: 150 | Price: \$499.99
Product: Tablet | ID: 103 | Stock: 80 | Price: \$299.99

3.ADD PRODUCT

Product ID:	4	Quantity:	60	Product Name:		
earphone	Price:	499	Add Product	Place Order	Replenish Stock	
			Display Products	Generate Report		
<div>Product: earphone ID: 4 Stock: 60 Price: \$499.0 Product: Laptop ID: 101 Stock: 50 Price: \$799.99 Product: Smartphone ID: 102 Stock: 150 Price: \$499.99 Product: Tablet ID: 103 Stock: 80 Price: \$299.99</div>						

4.PLACE ORDER

Product ID:	102	Quantity:	60	Product Name:		
Smartphone	Price:	499	Add Product	Place Order	Replenish Stock	
			Display Products	Generate Report		
<div>Order placed successfully for 60 of Smartphone</div>						

5.REPLINSH STOCK

Product ID:	103	Quantity:	90	Product Name:		
	Price:	499	Add Product	Place Order	Replenish Stock	
			Display Products	Generate Report		
<div>Stock replenished for Tablet by 90 units.</div>						