INVENTORY MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

TAMILDEEPAA A(2303811724322116)

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–621112

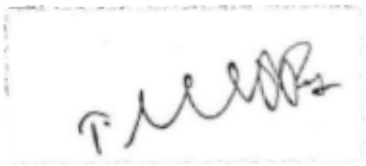
DECEMBER,2024

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY(AUTONOMOUS)

SAMAYAPURAM–621112

BONAFIDE CERTIFICATE

Certified that this project report on **" INVENTORY MANAGEMENT SYSTEM"** is the bonafide work of **TAMILDEEPAA A(2303811724322116)**who carriedout 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

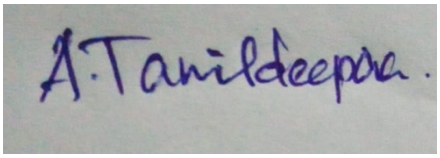
INTERNALEXAMINER

EXTERNALEXAMINER

DECLARATION

I declare that the project report on "**INVENTORY 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



TAMILDEEPAA A

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 acknowledgment 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 the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE** ,for providing his 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.

Mission1:To educate future engineers with solid fundamentals,continually improving teaching methods using modern tools.

Mission2:To collaborate with industry and offer top-notch facilities in a conductive learning environment.

Mission3:To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission4: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.

PEO3: Hone their professional skills through research and life long 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 ofdata, and synthesis of the information to provide valid conclusions.
5. **Moderntoolusage:**Create,select,andapplyappropriatetechniques,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 oftechnological 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

An Inventory Management System (IMS) is a crucial tool for efficiently managing and tracking goods, materials, and inventory in businesses. This project utilizesJava's Abstract Window Toolkit (AWT) to develop a graphical user interface (GUI) that simplifies essential inventory operations, such as adding, updating, deleting, and searching for items. By leveraging AWT's cross-platform compatibility and lightweight interface, this system demonstrates a practical solution for small and medium-sized enterprises to improve inventory management processes. The IMS aims to enhance accuracy, streamline workflows, and reduce manual errors in inventory control.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Inventory management is an essential aspect of business operations, ensuring that goods and resources are readily available while minimizing overstock and shortages. The proposed Inventory Management System, built using Java's AWT, provides a straightforwardGUI-basedapplicationtofacilitateeffectiveinventory control. AWT's comprehensive API offers tools to design interactive and platform- independent interfaces, making it an ideal choice for this project. The system enables users to perform core inventory functions, such as adding new items, updating stock information, deleting records, and searching for specific products, all through an intuitive interface. By automating and organizing inventory tasks, the IMS significantly reduces manual effort, enhances productivity, and improves overall inventory accuracy**.**

1.2 OBJECTIVE

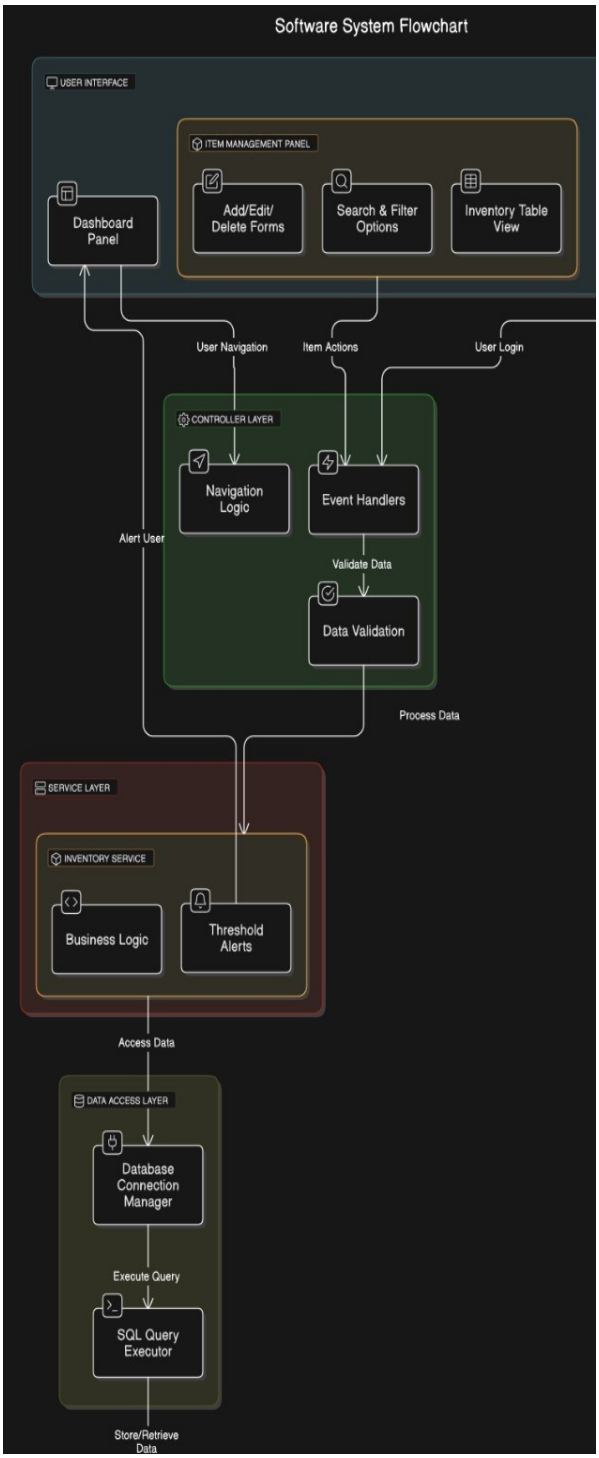
The primary objective of the Inventory Management System is to provide a user-friendly, efficient, and reliable solution for managing inventory, specifically designed for small and medium-sized businesses. It aims to streamline inventory processes by offering an easy-to-use GUI built with AWT components such as buttons,textfields,andtables.Thesystemfocusesonautomatingessential inventory operations to reduce human errors and improve accuracy. Additionally, it seeks to maintain organized and centralized data for better tracking of inventory levels, pricing, and details.

CHAPTER 2
PROJECT METHODOLOGY

2.1 PROPOSEDWORK

The proposed work involves developing an Inventory Management System using Java to efficiently manage products in a simplified and user-friendly manner. The system is designed to leverage the Abstract Window Toolkit (AWT) for creating an interactive graphical user interface, enabling users to perform essential operations such as adding, updating, removing, and displaying products. Each product is represented by attributes like ID, name, quantity, and price, encapsulated in a Product class, while an Inventory class manages the product data using a HashMap for fast and efficient access. The application incorporates event-driven programming to handle user actions through buttons and dialog boxes, providing real-time feedback and error handling for invalid inputs. The system emphasizes modularity and scalability, ensuring that it can be extended to include advanced features such as database integration or reporting tools. This project aims to combine the principles of object-oriented programming, GUI design, and data management to create a reliable and practical desktop application for inventory management.

2.2 BLOCK DIAGRAM



CHAPTER 3

JAVA PROGRAMMING CONCEPTS

3.1 OBJECT-ORIENTED PROGRAMMING(OOP)CONCEPT

- **Classes and Objects**:
  - The program defines multiple classes: Product, Inventory, and InventoryManagementSystemAWT (the main GUI class).Each class is a blueprint for creating objects, which represent real-world entities (products, inventory).
- **Encapsulation**:
  - The Product class encapsulates data such as id, name, quantity, and price with private fields. Public getter and setter methods (e.g., getId(), setQuantity()) are used to access and modify these fields.
- **Abstraction**:
  - The details of product management (adding, updating, removing products) are abstracted within the Inventory class. The user does not need to know how the products are stored (in a HashMap) or how the logic works internally. They only interact with public methods like addProduct(), removeProduct(), and displayProducts().
- **Inheritance**:
  - The main class InventoryManagementSystemAWT extends the Frame class, inheriting properties and behavior from the AWT Frame (awindow or frame in a GUI). This enables it to be a window that can display content and handle user interactions.

2. DATA STRUCTURE

- **HashMap**:
    - The Inventory class uses a HashMap<Integer, Product> to storeproducts. The Integer is the product ID, and the Product is the value associated with that ID.

3.2 EVENT–DRIVEN PROGRAMMING(EVENT HANDLING)

- **ActionListenerInterface**:

Event-driven programming is used to respond to user actions (e.g.,button clicks). The program implements ActionListener interfaces for various buttons (e.g., addButton.addActionListener), which listen for events (button clicks) and trigger specific actions in response.

CHAPTER 4

MODULE DESCRIPTION

4.1 ADD MODULE

The Add Module is responsible for enabling users to add new products to the inventory. It captures essential details such as product ID, name, quantity, and price through an input dialog box. After validating the provided data, the module interacts with the inventory by calling the addProduct() method, which stores the new product in the system.

4.2 UPDATE MODULE

The Update Module facilitates updating the quantity of an existing product. Users are prompted to enter the product ID and the new quantity through a dialog box. The module validates the input to ensure the product exists in the inventory and that the quantity is valid.

4.3 DISPLAY MODULE

The Display Module is designed to provide a comprehensive view of all products currently stored in the inventory. It retrieves the product list using the `displayProducts()` method and formats the details into a user-friendly display within the graphical interface.

4.4 REMOVE MODULE

The Remove Module allows users to delete a specific product from the inventory. It prompts the user to input the product ID through a dialog box and validates the input to ensure the product exists. A confirmation message is displayed.

CHAPTER 5
CONCLUSION

The **Inventory Management System** demonstrates a practical implementation ofJava programming concepts, combining object-oriented principles, GUI design, and efficient data management. By leveraging AWT for the user interface, the application provides an intuitive and interactive experience, while event-driven programming ensures responsive handling of user actions. The use of a HashMap from the Java Collections Framework enables fast and effective storage and retrieval of products, showcasing the program's efficiency in managing inventory data. Error handling through exceptions like NumberFormatException enhances the system's robustness, ensuring reliability even with invalid user inputs. Furthermore, the modular architecture makes the system scalable and extendable for additional features in the future. Overall, this project highlights the strength of Java as a language for building functional and user-friendly desktop applications.

REFERENCES:

- **Java Official Documentation**: Comprehensive resource for Java programming concepts, including AWT, event handling, and collections. Available at: https://docs.oracle.com/javase/

- **Java Tutorials by Oracle**: A beginner-friendly guide for learning Java concepts, including GUI programming and core libraries. Available at: https://docs.oracle.com/javase/tutorial/

- **Effective Java by Joshua Bloch**: A highly recommended book for understanding best practices and advanced concepts in Java.

- **Head First Java by Kathy Sierra and Bert Bates**: A user-friendly book that introduces Java programming, including object-oriented design and GUI basics.

APPENDICES

APPENDIX A–SOURCECODE

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

// Product class to represent individual product items
class Product {
    private int id;private
    String name; private
    int quantity;
    privatedoubleprice;

    public Product(int id, String name, int quantity, double price) {
        this.id = id;
        this.name = name;
        this.quantity = quantity;
        this.price = price;
    }


    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public int getQuantity() {
        return quantity;
    }
```

```
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

    public double getPrice() {
        return price;
    }

    @Override
    publicStringtoString(){
        return"ID:"+id+",Name:"+name+",Quantity:"+quantity+",Price:"+price;
    }
}

// Inventory class to manage products
class Inventory {
    privateMap<Integer,Product>products=newHashMap<>();

    public void addProduct(Product product) {
        products.put(product.getId(), product);
    }

    public void updateProductQuantity(int productId, int quantity) {
        if (products.containsKey(productId)) {
            Product product = products.get(productId);
            product.setQuantity(quantity);
        }
    }

    public String displayProducts() {
        if (products.isEmpty()) {
            return"Noproductsininventory.";
        }else{
```

```
        StringBuilder sb = new StringBuilder();for
        (Product product : products.values()) {
            sb.append(product).append("\n");
        }
        returnsb.toString();
    }
  }

  public void removeProduct(int productId) {
    products.remove(productId);
  }
}

//MainclasswithAWTGUI
public class InventoryManagementSystemAWT extends Frame {
  private Inventory inventory = new Inventory();

  publicInventoryManagementSystemAWT(){
    //Framesetup
    setTitle("Inventory Management System");
    setSize(500, 400);
    setLayout(new FlowLayout());
    setResizable(false);

    //Components
    Label label = new Label("Inventory Management System", Label.CENTER);
    label.setFont(new Font("Arial", Font.BOLD, 20));
    add(label);

    ButtonaddButton=newButton("AddProduct");
    ButtonupdateButton=newButton("UpdateQuantity");
    Button displayButton = new Button("Display Products");
    Button removeButton = new Button("Remove Product");
    Button exitButton = new Button("Exit");
```

```
add(addButton);
add(updateButton);
add(displayButton);
add(removeButton);
add(exitButton);

TextArea outputArea = new TextArea(15, 40);
outputArea.setEditable(false);add(outputArea);

// Event listeners
addButton.addActionListener(e ->{
    Dialog dialog = new Dialog(this, "Add Product", true);
    dialog.setLayout(new FlowLayout());
    dialog.setSize(300, 200);

    Label idLabel = new Label("ID:");
    TextField idField = new TextField(10);
    Label nameLabel = new Label("Name:");
    TextField nameField = new TextField(20);
    Label quantityLabel = new Label("Quantity:");
    TextField quantityField = new TextField(10);
    Label priceLabel = new Label("Price:");
    TextField priceField = new TextField(10);

    Button submitButton = new Button("Submit");
    submitButton.addActionListener(ae -> {
       try{
          int id = Integer.parseInt(idField.getText());
          String name = nameField.getText();
          int quantity = Integer.parseInt(quantityField.getText());
          double price = Double.parseDouble(priceField.getText());

          inventory.addProduct(newProduct(id,name,quantity,price));
```

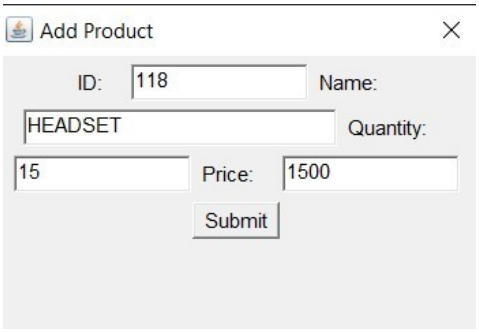
```
            outputArea.setText("Product added successfully!\n" + inventory.displayProducts());
            dialog.dispose();
        } catch (NumberFormatException ex) {
            outputArea.setText("Invalid input. Please try again.");
        }
    });

    dialog.add(idLabel);
    dialog.add(idField);
    dialog.add(nameLabel);
    dialog.add(nameField);
    dialog.add(quantityLabel);
    dialog.add(quantityField);
    dialog.add(priceLabel);
    dialog.add(priceField);
    dialog.add(submitButton);
    dialog.setVisible(true);
});

updateButton.addActionListener(e->{
    Dialog dialog = new Dialog(this, "Update Quantity", true);
    dialog.setLayout(new FlowLayout());
    dialog.setSize(300,150);

    Label idLabel = new Label("Product ID:");
    TextField idField = new TextField(10);
    Label quantityLabel = new Label("New Quantity:");
    TextField quantityField = new TextField(10);

    Button submitButton = new Button("Submit");
    submitButton.addActionListener(ae -> {
        try{
            intid=Integer.parseInt(idField.getText());
            intquantity=Integer.parseInt(quantityField.getText());
```

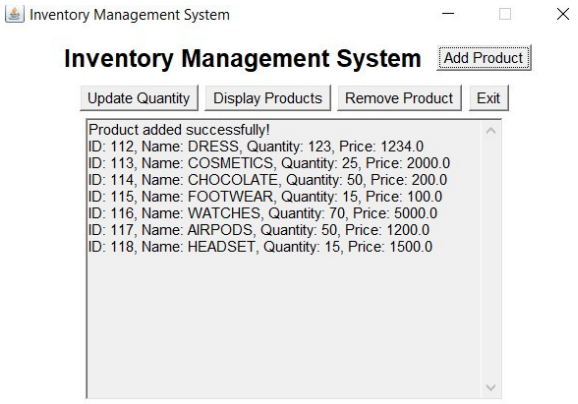
```
            inventory.updateProductQuantity(id,quantity);
            outputArea.setText("Quantity updated successfully!\n" + inventory.displayProducts());
            dialog.dispose();
        } catch (NumberFormatException ex) {
            outputArea.setText("Invalid input. Please try again.");
        }
    });

    dialog.add(idLabel);
    dialog.add(idField);
    dialog.add(quantityLabel);
    dialog.add(quantityField);
    dialog.add(submitButton);
    dialog.setVisible(true);
});

displayButton.addActionListener(e->outputArea.setText(inventory.displayProducts()));

removeButton.addActionListener(e->{
    Dialog dialog = new Dialog(this, "Remove Product", true);
    dialog.setLayout(new FlowLayout());
    dialog.setSize(300,150);

    Label idLabel = new Label("Product ID:");
    TextField idField = new TextField(10);

    Button submitButton = new Button("Submit");
    submitButton.addActionListener(ae -> {
        try{
            int id = Integer.parseInt(idField.getText());
            inventory.removeProduct(id);
            outputArea.setText("Product removed successfully!\n" + inventory.displayProducts());
            dialog.dispose();
        }catch(NumberFormatExceptionex){
```

```
                    outputArea.setText("Invalidinput.Pleasetryagain.");
}
});

                dialog.add(idLabel);
                dialog.add(idField);
                dialog.add(submitButton);
                dialog.setVisible(true);
            });

            exitButton.addActionListener(e->System.exit(0));

            // Window close handler
            addWindowListener(new WindowAdapter() {
                public void windowClosing(WindowEvent e) {
                    System.exit(0);
}
});
        }

        public static void main(String[] args) {
            EventQueue.invokeLater(() -> {
                InventoryManagementSystemAWT ims = new InventoryManagementSystemAWT();
                ims.setVisible(true);
            });
        }
    }
```

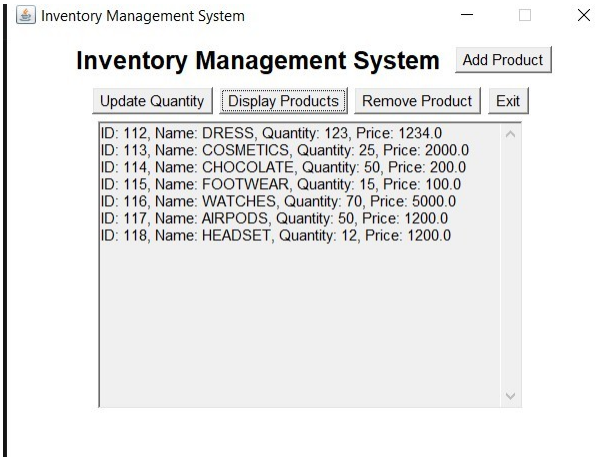
APPENDIX B-SCREENSHOTS

ADDING A PRODUCT

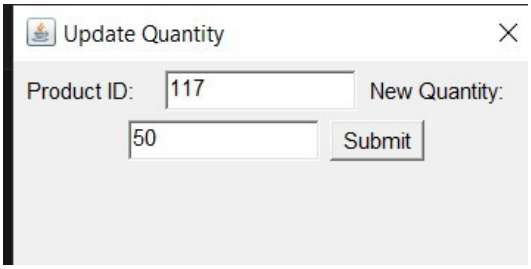


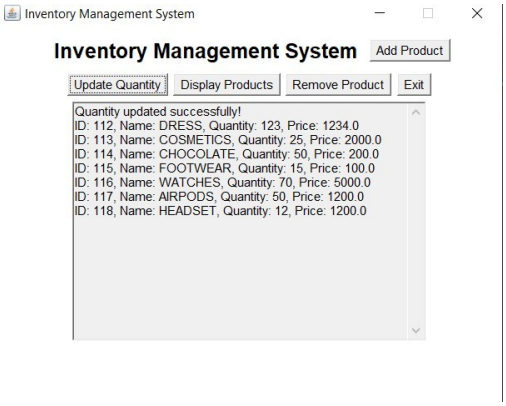


DISPLAY PRODUCTS



UPDATE PRODUCT QUANTITY





REMOVE PRODUCT