```
import java.awt.*;
import java.awt.event.*;
import java.util.HashMap;
import javax.swing.*;

class PharmacyManagementSystem extends Frame {
    private HashMap<String, Medicine> inventory = new HashMap<>();
    private TextArea outputArea;
    private TextField nameField, priceField, quantityField, purchaseField, purchaseQtyField;

    public PharmacyManagementSystem() {
        setLayout(new FlowLayout());
        setTitle("Pharmacy Management System");
        setSize(600, 500);

        // Title Label
        Label title = new Label("Pharmacy Management System");
        title.setFont(new Font("Arial", Font.BOLD, 20));
        add(title);

        // Add Medicine Section
        Label addSection = new Label("Add Medicine");
        add(addSection);

        nameField = new TextField("Medicine Name", 15);
        priceField = new TextField("Price", 10);
        quantityField = new TextField("Quantity", 10);
        Button addButton = new Button("Add Medicine");

        add(nameField);
        add(priceField);
        add(quantityField);
        add(addButton);

        // Purchase Section
        Label purchaseSection = new Label("Purchase Medicine");
        add(purchaseSection);

        purchaseField = new TextField("Medicine Name", 15);
        purchaseQtyField = new TextField("Quantity", 10);
        Button purchaseButton = new Button("Purchase");

        add(purchaseField);
        add(purchaseQtyField);
```

```
        add(purchaseButton);

        // Output Area
        outputArea = new TextArea(15, 50);
        add(outputArea);

        // Add Stock Button
        Button updateButton = new Button("Update Stock");
        add(updateButton);

        // Event Listeners
        addButton.addActionListener(e -> addMedicine());
        purchaseButton.addActionListener(e -> purchaseMedicine());
        updateButton.addActionListener(e -> updateStock());

        // Window Closing Event
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                dispose();
            }
        });
    }

    // Add Medicine
    private void addMedicine() {
        try {
            String name = nameField.getText().trim();
            double price = Double.parseDouble(priceField.getText().trim());
            int quantity = Integer.parseInt(quantityField.getText().trim());

            if (inventory.containsKey(name)) {
                inventory.get(name).quantity += quantity;
            } else {
                inventory.put(name, new Medicine(name, price, quantity));
            }

            outputArea.append("Medicine added: " + name + " | Price: $" + price + " | Quantity: " +
quantity + "\n");
        } catch (NumberFormatException ex) {
            outputArea.append("Error: Invalid input format.\n");
        }
    }

    // Purchase Medicine
```

```
    private void purchaseMedicine() {
        try {
            String name = purchaseField.getText().trim();
            int quantity = Integer.parseInt(purchaseQtyField.getText().trim());

            if (!inventory.containsKey(name)) {
                outputArea.append("Error: Medicine not found in inventory.\n");
                return;
            }

            Medicine med = inventory.get(name);
            if (med.quantity < quantity) {
                outputArea.append("Error: Insufficient stock for " + name + ".\n");
                return;
            }

            med.quantity -= quantity;
            double cost = quantity * med.price;
            outputArea.append("Purchased: " + quantity + " x " + name + " | Total Cost: $" + cost +
"\n");
        } catch (NumberFormatException ex) {
            outputArea.append("Error: Invalid input format.\n");
        }
    }

    // Update Stock
    private void updateStock() {
        StringBuilder stockDetails = new StringBuilder("Current Stock:\n");
        for (Medicine med : inventory.values()) {
            stockDetails.append(med.name).append(" | Price: $").append(med.price).append(" |
Quantity: ").append(med.quantity).append("\n");
        }
        outputArea.setText(stockDetails.toString());
    }

    // Main Class
    public static void main(String[] args) {
        PharmacyManagementSystem app = new PharmacyManagementSystem();
        app.setVisible(true);
    }

    // Inner Medicine Class
    class Medicine {
        String name;
```

```
        double price;
        int quantity;

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