```java
import javax.swing.*;
import java.awt.event.ActionEvent;

import java.awt.*
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.List;

public class Dashboard extends JFrame {

    private List<Product> products;
    private List<StockItem> stockItems;
    private List<Sale> sales;

    public Dashboard() {
        initData();
        initComponents();
    }

    private void initData() {
        products = new ArrayList<>();
        products.add(new Product(1, "Apples", 1.99));
        products.add(new Product(2, "Tomatoes", 2.49));

        stockItems = new ArrayList<>();
        stockItems.add(new StockItem(1, "Apples", 50));
        stockItems.add(new StockItem(2, "Tomatoes", 100));

        sales = new ArrayList<>();
        sales.add(new Sale(1, "Apples", 10));
        sales.add(new Sale(2, "Tomatoes", 20));
    }

    private void initComponents() {
        JButton productButton = new JButton("Product");
        JButton stockButton = new JButton("Current Stock");
        JButton salesButton = new JButton("Sales");
        JButton searchButton = new JButton("Search");
        JButton addButton = new JButton("Add");

        productButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                showAllProducts();
            }
        });

        stockButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                showAllStockItems();
            }
```

```java
    });
    salesButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            showAllSales();
        }
    });

    searchButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            showSearchBar();
        }
    });

    addButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            showAddDataWindow();
        }
    });

    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout(5, 1));
    panel.add(productButton);
    panel.add(stockButton);
    panel.add(salesButton);
    panel.add(searchButton);
    panel.add(addButton);

    add(panel);

    setTitle("Warehouse Accounting System");
    setSize(300, 250);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
    setVisible(true);
}

private void showAllProducts() {
    showProductData(products);
}

private void showAllStockItems() {
    showStockData(stockItems);
}

private void showAllSales() {
    showSalesData(sales);
}

private void showSearchBar() {
```

```java
            SearchBar searchBar = new SearchBar(products, stockItems, sales);
            searchBar.setVisible(true);
        }

    private void showAddDataWindow() {
            AddDataWindow addDataWindow = new AddDataWindow(products, stockItems, sales);
            addDataWindow.setVisible(true);
        }

    private void showProductData(List<Product> productList) {
            showData("Product Data", productList);
        }

    private void showStockData(List<StockItem> stockItemList) {
            showData("Current Stock Data", stockItemList);
        }

    private void showSalesData(List<Sale> saleList) {
            showData("Sales Data", saleList);
        }

    private void showData(String title, List<?> dataList) {
            JFrame dataFrame = new JFrame(title);
            JTextArea textArea = new JTextArea();
            for (Object data : dataList) {
                textArea.append(data.toString() + "\n");
            }
            dataFrame.add(new JScrollPane(textArea));
            dataFrame.setSize(300, 200);
            dataFrame.setLocationRelativeTo(this);
            dataFrame.setVisible(true);
        }

    public static void main(String[] args) {
            SwingUtilities.invokeLater(() -> new Dashboard());
        }
}

class SearchBar extends JFrame {

    private List<Product> products;
    private List<StockItem> stockItems;
    private List<Sale> sales;

    private JTextField idField;
    private JButton searchButton;

    public SearchBar(List<Product> products, List<StockItem> stockItems, List<Sale>
sales) {
        this.products = products;
        this.stockItems = stockItems;
```

```java
        this.sales = sales;

        initComponents();
    }

    private void initComponents() {
        idField = new JTextField(10);
        searchButton = new JButton("Search");

        searchButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                searchById();
            }
        });

        JPanel panel = new JPanel();
        panel.add(new JLabel("Enter ID:"));
        panel.add(idField);
        panel.add(searchButton);

        add(panel);

        setTitle("Search Bar");
        setSize(300, 100);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
    }

    private void searchById() {
        int id;
        try {
            id = Integer.parseInt(idField.getText());
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this, "Invalid ID. Please enter a valid
integer.");
            return;
        }

        List<Product> matchingProducts = getMatchingProducts(id);
        List<StockItem> matchingStockItems = getMatchingStockItems(id);
        List<Sale> matchingSales = getMatchingSales(id);

        showData("Search Results", matchingProducts, matchingStockItems, matchingSales);
    }

    private List<Product> getMatchingProducts(int id) {
        List<Product> matchingProducts = new ArrayList<>();
        for (Product product : products) {
            if (product.getId() == id) {
                matchingProducts.add(product);
```

```java
            }
        }
        return matchingProducts;
    }

    private List<StockItem> getMatchingStockItems(int id) {
        List<StockItem> matchingStockItems = new ArrayList<>();
        for (StockItem stockItem : stockItems) {
            if (stockItem.getId() == id) {
                matchingStockItems.add(stockItem);
            }
        }
        return matchingStockItems;
    }

    private List<Sale> getMatchingSales(int id) {
        List<Sale> matchingSales = new ArrayList<>();
        for (Sale sale : sales) {
            if (sale.getId() == id) {
                matchingSales.add(sale);
            }
        }
        return matchingSales;
    }

    private void showData(String title, List<Product> products, List<StockItem>
stockItems, List<Sale> sales) {
        JFrame dataFrame = new JFrame(title);
        JTextArea textArea = new JTextArea();

        textArea.append("Product Data:\n");
        for (Product product : products) {
            textArea.append(product.toString() + "\n");
        }

        textArea.append("\nCurrent Stock Data:\n");
        for (StockItem stockItem : stockItems) {
            textArea.append(stockItem.toString() + "\n");
        }

        textArea.append("\nSales Data:\n");
        for (Sale sale : sales) {
            textArea.append(sale.toString() + "\n");
        }

        dataFrame.add(new JScrollPane(textArea));
        dataFrame.setSize(300, 200);
        dataFrame.setLocationRelativeTo(this);
        dataFrame.setVisible(true);
    }
}
```

```java
class AddDataWindow extends JFrame {

    private List<Product> products;
    private List<StockItem> stockItems;
    private List<Sale> sales;

    private JTextField idField;
    private JTextField nameField;
    private JTextField priceField;
    private JTextField quantityField;
    private JTextField salesQuantityField;
    private JButton addButton;

    public AddDataWindow(List<Product> products, List<StockItem> stockItems, List<Sale>
sales) {
        this.products = products;
        this.stockItems = stockItems;
        this.sales = sales;

        initComponents();
    }

    private void initComponents() {
        idField = new JTextField(10);
        nameField = new JTextField(10);
        priceField = new JTextField(10);
        quantityField = new JTextField(10);
        salesQuantityField = new JTextField(10);
        addButton = new JButton("Add");

        addButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                addData();
            }
        });

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(6, 2));
        panel.add(new JLabel("ID:"));
        panel.add(idField);
        panel.add(new JLabel("Name:"));
        panel.add(nameField);
        panel.add(new JLabel("Price:"));
        panel.add(priceField);
        panel.add(new JLabel("Quantity:"));
        panel.add(quantityField);
        panel.add(new JLabel("Sales Quantity:"));
        panel.add(salesQuantityField);
        panel.add(new JLabel(""));
        panel.add(addButton);
```

```java
        add(panel);

        setTitle("Add Data");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
    }

    private void addData() {
        int id;
        String name;
        double price;
        int quantity;
        int salesQuantity;

        try {
            id = Integer.parseInt(idField.getText());
            name = nameField.getText();
            price = Double.parseDouble(priceField.getText());
            quantity = Integer.parseInt(quantityField.getText());
            salesQuantity = Integer.parseInt(salesQuantityField.getText());
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this, "Invalid input. Please enter valid
numbers.");
            return;
        }

        // Check if the ID already exists
        Product existingProduct = Product.getProductById(products, id);
        StockItem existingStockItem = StockItem.getStockItemById(stockItems, id);
        Sale existingSale = Sale.getSaleById(sales, id);

        if (existingProduct != null) {
            // If ID exists in products, update product data
            existingProduct.setName(name);
            existingProduct.setPrice(price);
        } else {
            // If ID doesn't exist in products, add a new product
            products.add(new Product(id, name, price));
        }

        if (existingStockItem != null) {
            // If ID exists in stockItems, update stockItem data
            existingStockItem.setName(name);
            existingStockItem.setQuantity(quantity);
        } else {
            // If ID doesn't exist in stockItems, add a new stockItem
            stockItems.add(new StockItem(id, name, quantity));
        }

        if (existingSale != null) {
            // If ID exists in sales, update sale data
```

```java
            existingSale.setProductName(name);
            existingSale.setQuantitySold(salesQuantity);
        } else {
            // If ID doesn't exist in sales, add a new sale
            sales.add(new Sale(id, name, salesQuantity));
        }

        JOptionPane.showMessageDialog(this, "Data added/updated successfully!");

        // Clear fields after adding/updating data
        idField.setText("");
        nameField.setText("");
        priceField.setText("");
        quantityField.setText("");
        salesQuantityField.setText("");
    }
}

class Product {
    private int id;
    private String name;
    private double price;

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

    public int getId() {
        return id;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    @Override
    public String toString() {
        return "Product{id=" + id + ", name='" + name + "', price=" + price + '}';
    }
    public static Product getProductById(List<Product> products, int id) {
        for (Product product : products) {
            if (product.getId() == id) {
                return product;
            }
        }
        return null;
```

```java
        }
    }
    class StockItem {
        private int id;
        private String name;
        private int quantity;

        public StockItem(int id, String name, int quantity) {
            this.id = id;
            this.name = name;
            this.quantity = quantity;
        }

        public int getId() {
            return id;
        }

        public void setName(String name) {
            this.name = name;
        }

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

        @Override
        public String toString() {
            return "StockItem{id=" + id + ", name='" + name + "', quantity=" + quantity +
    '}';
        }
        public static StockItem getStockItemById(List<StockItem> stockItems, int id) {
            for (StockItem stockItem : stockItems) {
                if (stockItem.getId() == id) {
                    return stockItem;
                }
            }
            return null;
        }
    }
    class Sale {
        private int id;
        private String productName;
        private int quantitySold;

        public Sale(int id, String productName, int quantitySold) {
            this.id = id;
            this.productName = productName;
            this.quantitySold = quantitySold;
        }

        public int getId() {
            return id;
```

```java
    }

    public void setProductName(String productName) {
        this.productName = productName;
    }

    public void setQuantitySold(int quantitySold) {
        this.quantitySold = quantitySold;
    }

    @Override
    public String toString() {
        return "Sale{id=" + id + ", productName='" + productName + "', quantitySold=" +
quantitySold + '}';
    }
    public static Sale getSaleById(List<Sale> sales, int id) {
        for (Sale sale : sales) {
            if (sale.getId() == id) {
                return sale;
            }
        }
        return null;
    }
}
```