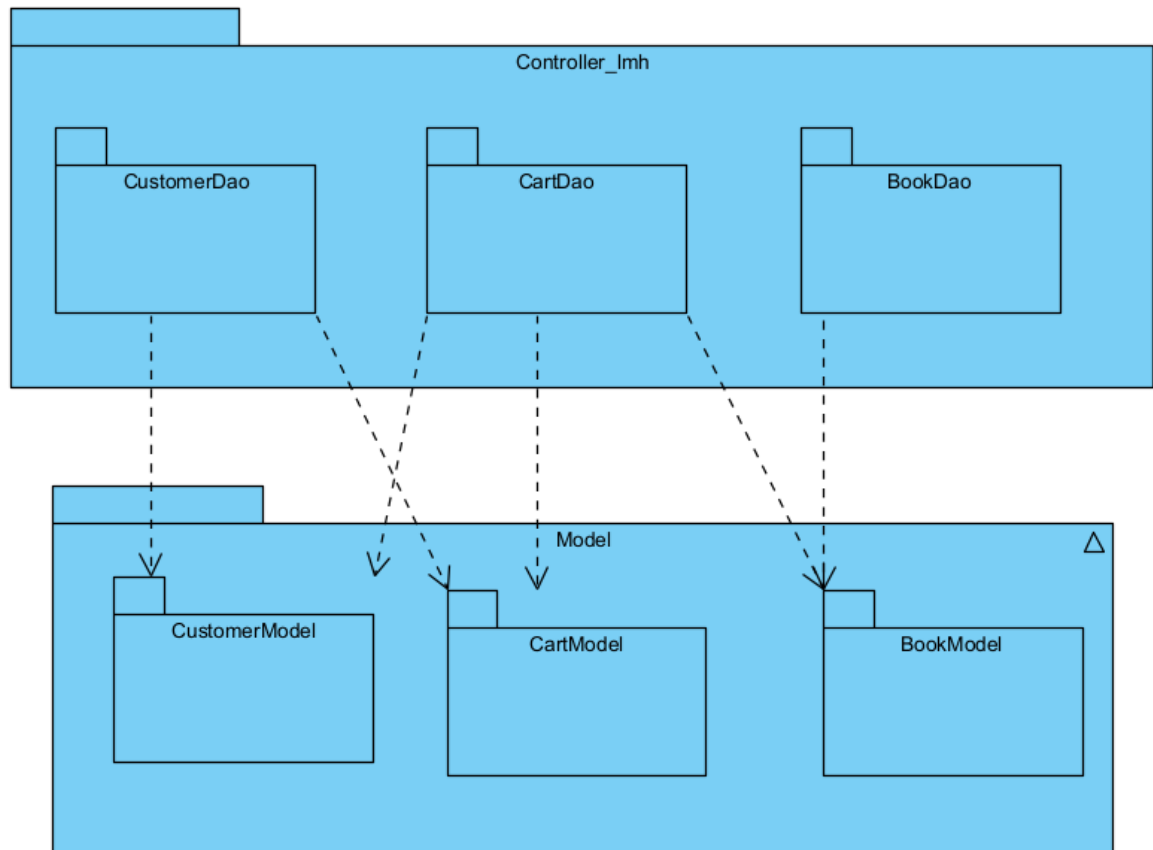


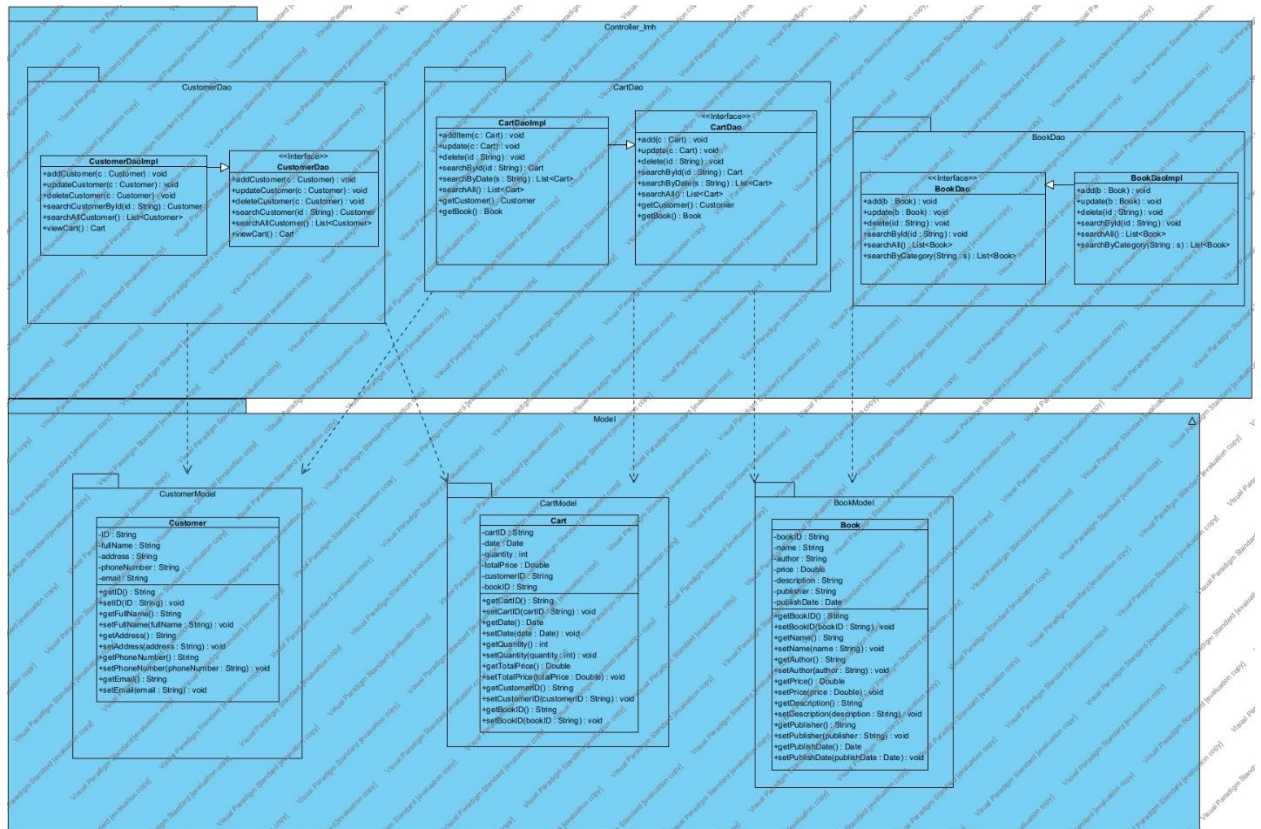
Lương Mạnh Hòa – B21DCCN378

Nhóm 1

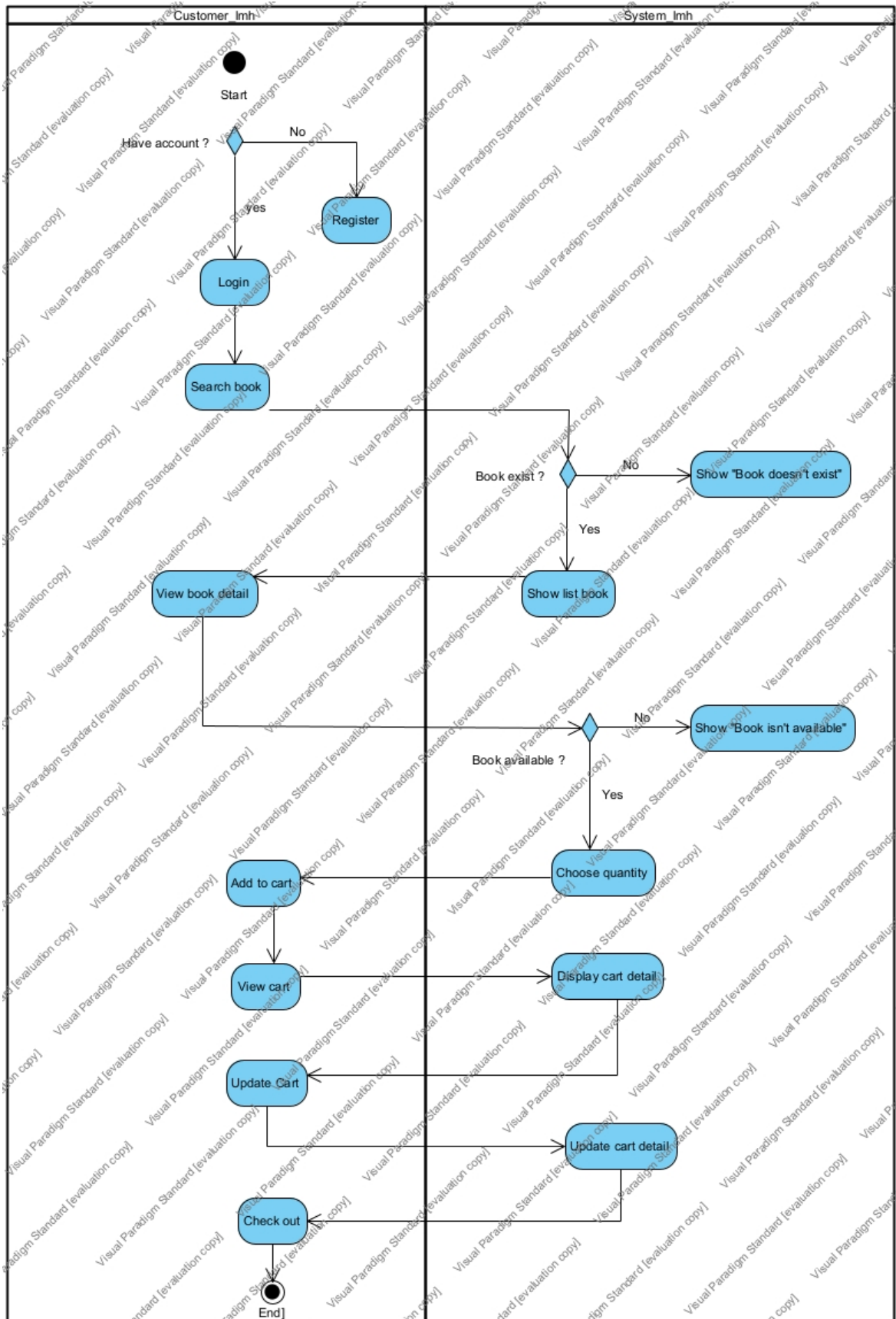
1. Overall



2. Detail Design

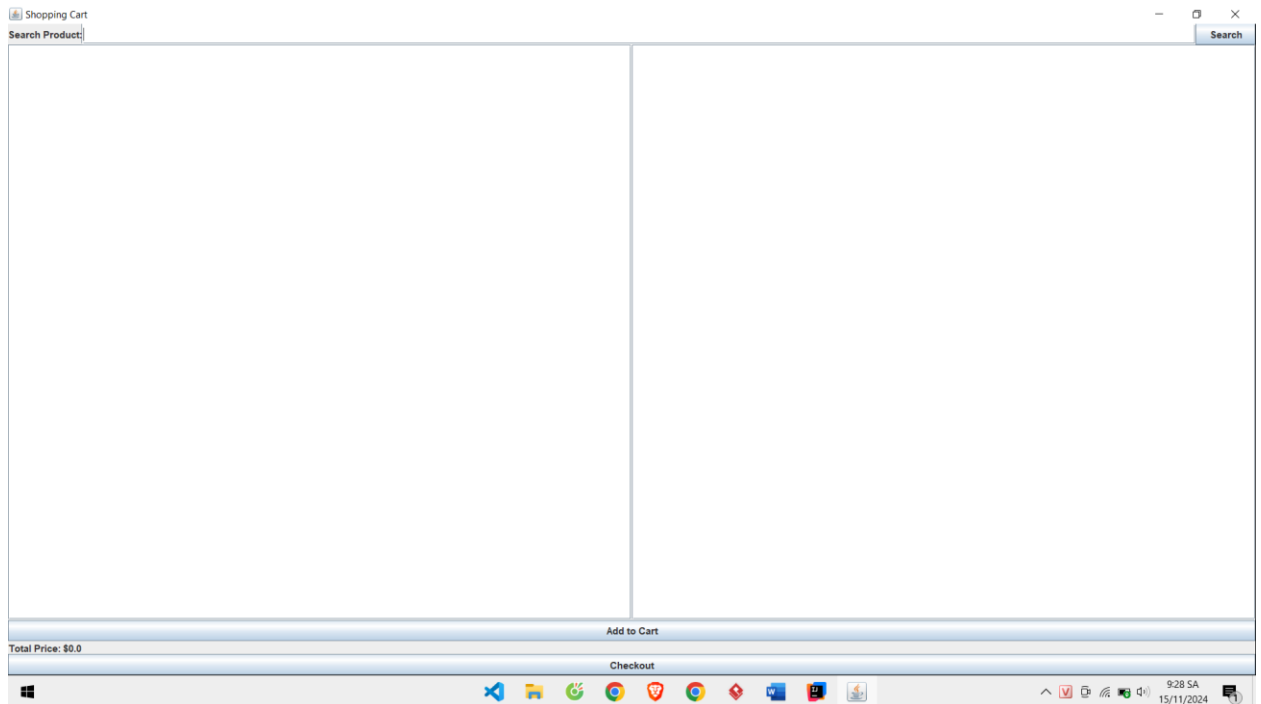


### 3. Activity Diagram

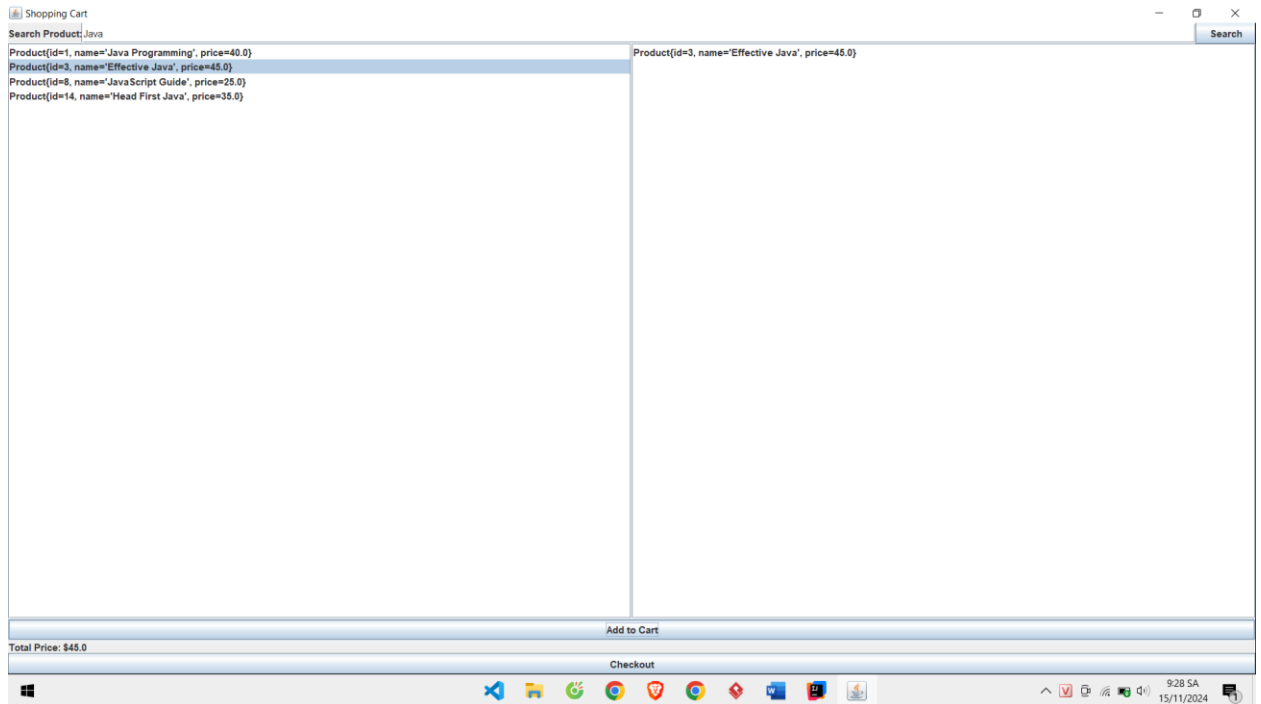


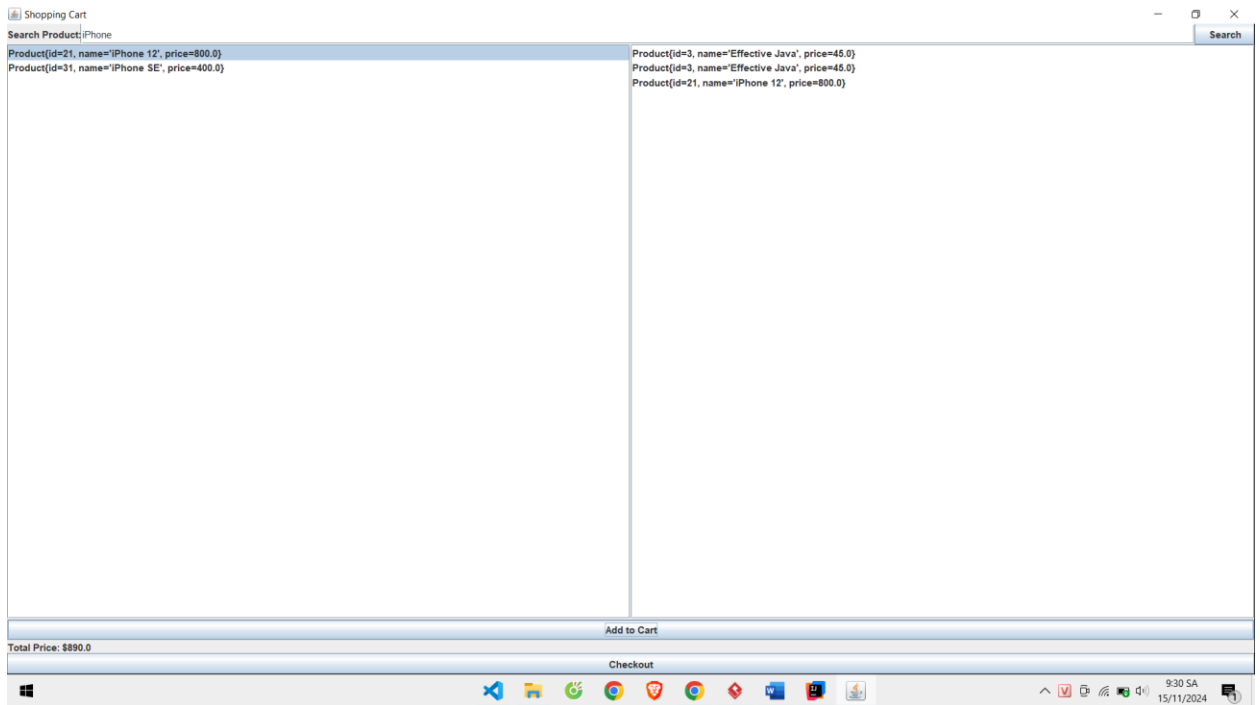
#### 4. Code:

Giao diện:

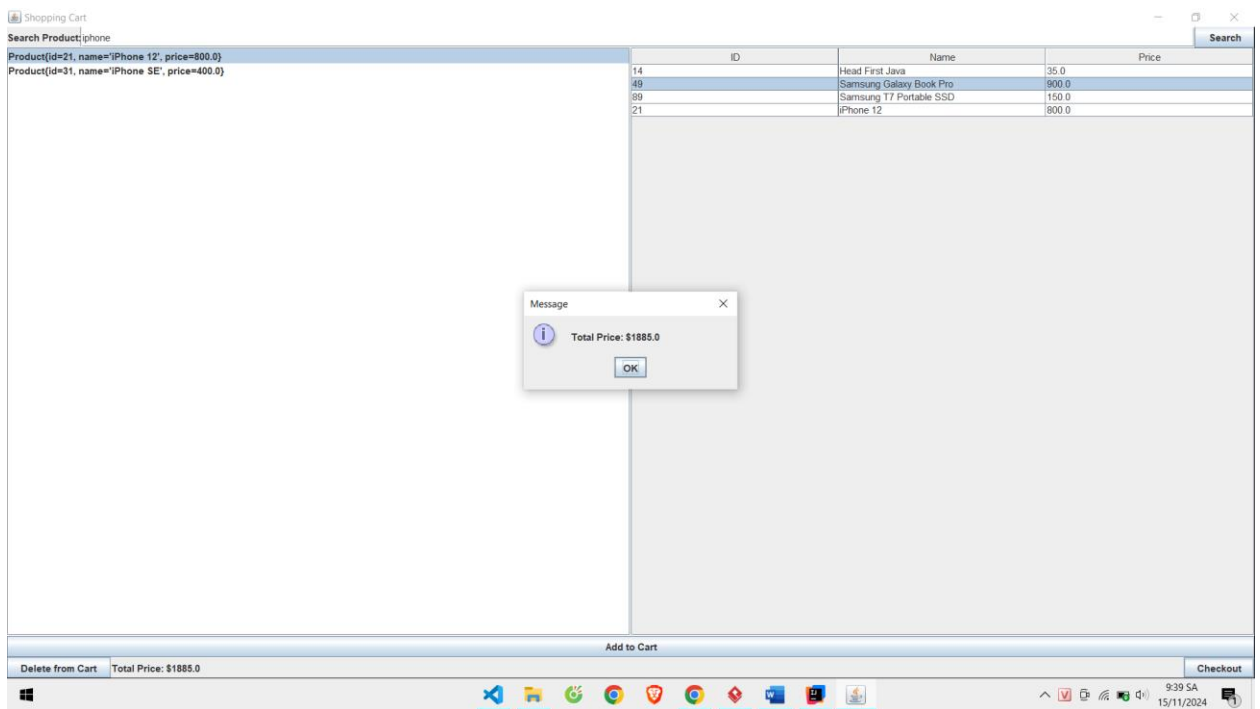


Thêm vào giỏ hàng



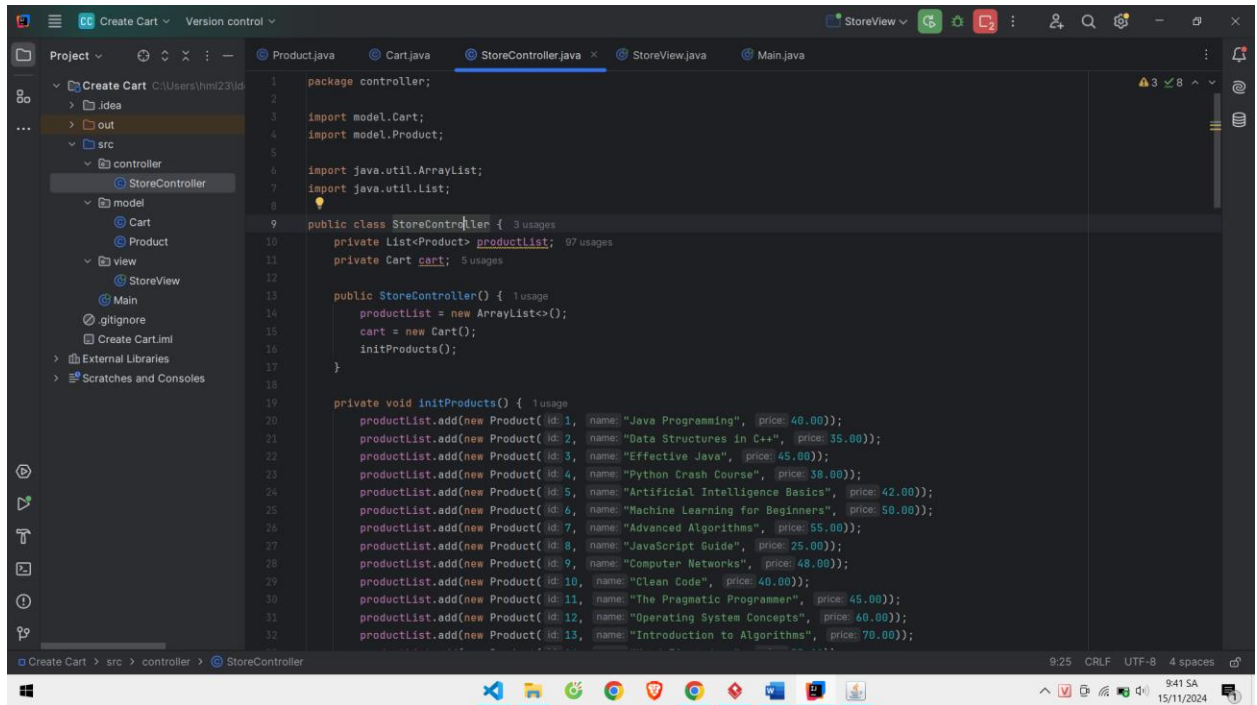


## Check out



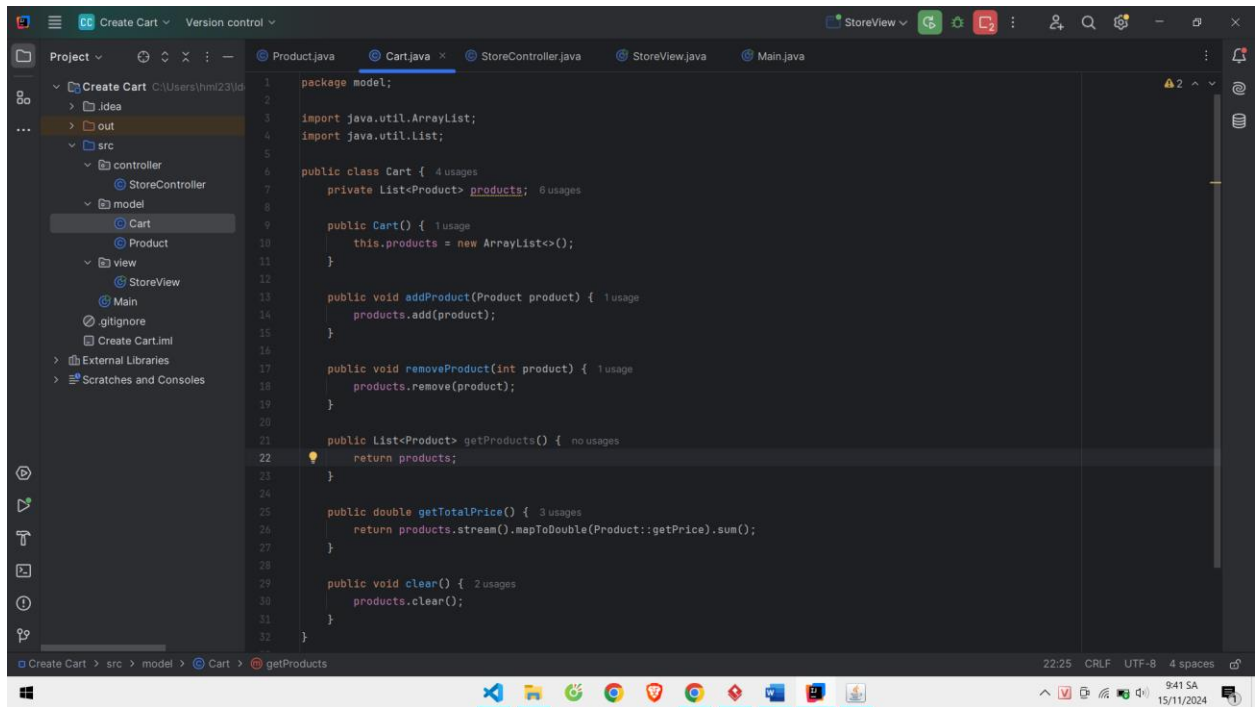
Code của bài:

a. StoreController



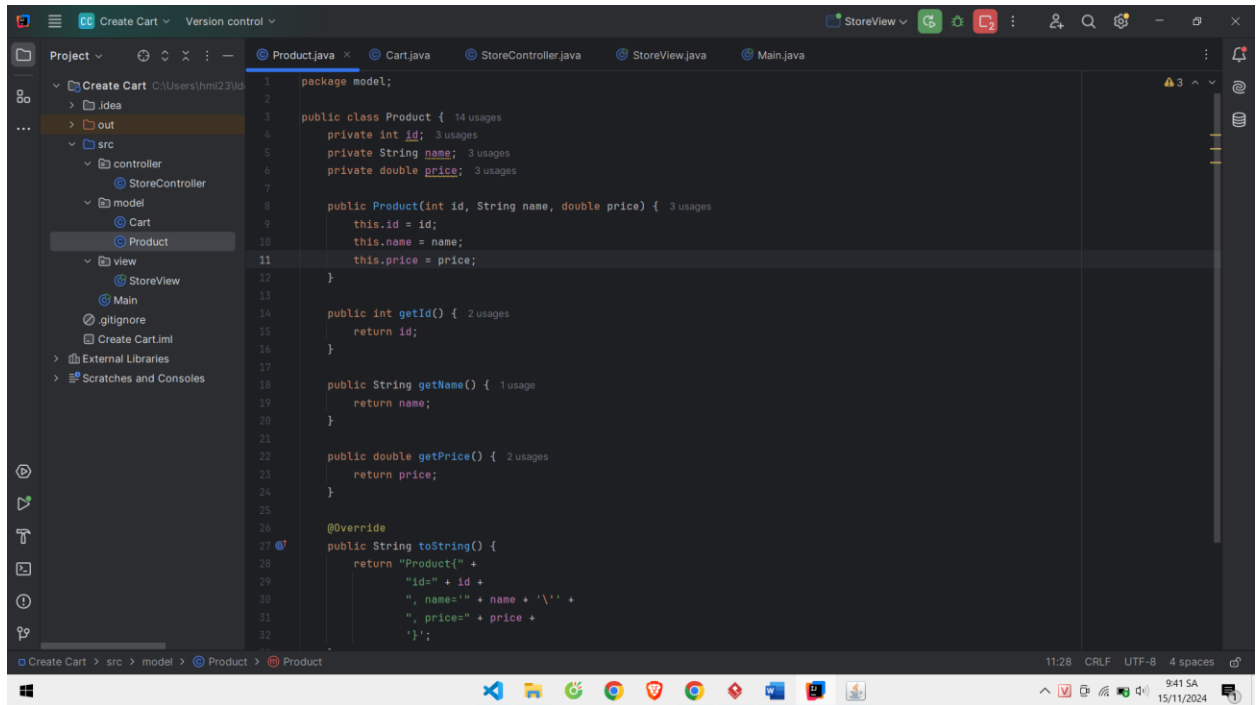
```
1 package controller;
2
3 import model.Cart;
4 import model.Product;
5
6 import java.util.ArrayList;
7 import java.util.List;
8
9 public class StoreController {
10     private List<Product> productList;
11     private Cart cart;
12
13     public StoreController() {
14         productList = new ArrayList<>();
15         cart = new Cart();
16         initProducts();
17     }
18
19     private void initProducts() {
20         productList.add(new Product(1, "Java Programming", 40.00));
21         productList.add(new Product(2, "Data Structures in C++", 35.00));
22         productList.add(new Product(3, "Effective Java", 45.00));
23         productList.add(new Product(4, "Python Crash Course", 38.00));
24         productList.add(new Product(5, "Artificial Intelligence Basics", 42.00));
25         productList.add(new Product(6, "Machine Learning for Beginners", 50.00));
26         productList.add(new Product(7, "Advanced Algorithms", 55.00));
27         productList.add(new Product(8, "JavaScript Guide", 25.00));
28         productList.add(new Product(9, "Computer Networks", 48.00));
29         productList.add(new Product(10, "Clean Code", 40.00));
30         productList.add(new Product(11, "The Pragmatic Programmer", 45.00));
31         productList.add(new Product(12, "Operating System Concepts", 60.00));
32         productList.add(new Product(13, "Introduction to Algorithms", 70.00));
33     }
34 }
```

## b. Cart



```
1 package model;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Cart {
7     private List<Product> products;
8
9     public Cart() {
10         this.products = new ArrayList<>();
11     }
12
13     public void addProduct(Product product) {
14         products.add(product);
15     }
16
17     public void removeProduct(int product) {
18         products.remove(product);
19     }
20
21     public List<Product> getProducts() {
22         return products;
23     }
24
25     public double getTotalPrice() {
26         return products.stream().mapToDouble(Product::getPrice).sum();
27     }
28
29     public void clear() {
30         products.clear();
31     }
32 }
```

## c. Product

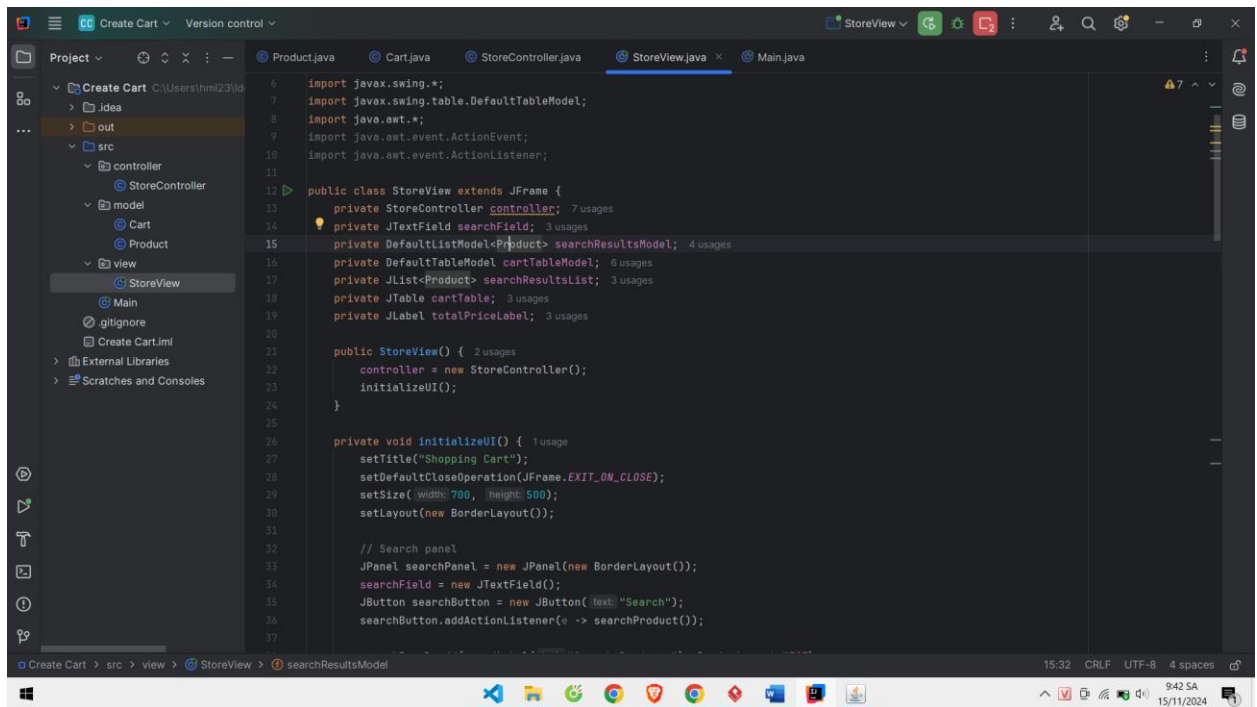


The screenshot shows the IntelliJ IDEA IDE with the 'Create Cart' project open. The 'Product.java' file is selected in the editor, located in the 'model' package. The code defines a 'Product' class with attributes 'id', 'name', and 'price', and methods for getting these values and a 'toString()' method. The project structure on the left shows the following hierarchy:

- src
  - controller
    - StoreController
  - model
    - Cart
    - Product
  - view
    - StoreView
    - Main

```
1 package model;
2
3 public class Product {
4     private int id;
5     private String name;
6     private double price;
7
8     public Product(int id, String name, double price) {
9         this.id = id;
10        this.name = name;
11        this.price = price;
12    }
13
14    public int getId() {
15        return id;
16    }
17
18    public String getName() {
19        return name;
20    }
21
22    public double getPrice() {
23        return price;
24    }
25
26    @Override
27    public String toString() {
28        return "Product{" +
29            "id=" + id +
30            ", name='" + name + '\'' +
31            ", price=" + price +
32            '}';
33    }
34 }
```

d. StoreView



The screenshot shows the IntelliJ IDEA IDE with the 'Create Cart' project open. The 'StoreView.java' file is selected in the editor, located in the 'view' package. The code defines a 'StoreView' class that extends 'JFrame', containing a search field, search results list, and a table to display products. The project structure on the left shows the following hierarchy:

- src
  - controller
    - StoreController
  - model
    - Cart
    - Product
  - view
    - StoreView
    - Main

```
1 import javax.swing.*;
2 import javax.swing.table.DefaultTableModel;
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6
7 public class StoreView extends JFrame {
8     private StoreController controller;
9     private JTextField searchField;
10    private DefaultListModel<Product> searchResultsModel;
11    private DefaultTableModel cartTableModel;
12    private JList<Product> searchResultsList;
13    private JTable cartTable;
14    private JLabel totalPriceLabel;
15
16    public StoreView() {
17        controller = new StoreController();
18        initializeUI();
19    }
20
21    private void initializeUI() {
22        setTitle("Shopping Cart");
23        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24        setSize(700, 500);
25        setLayout(new BorderLayout());
26
27        // Search panel
28        JPanel searchPanel = new JPanel(new BorderLayout());
29        searchField = new JTextField();
30        JButton searchButton = new JButton("Search");
31        searchButton.addActionListener(e -> searchProduct());
32    }
33 }
```