

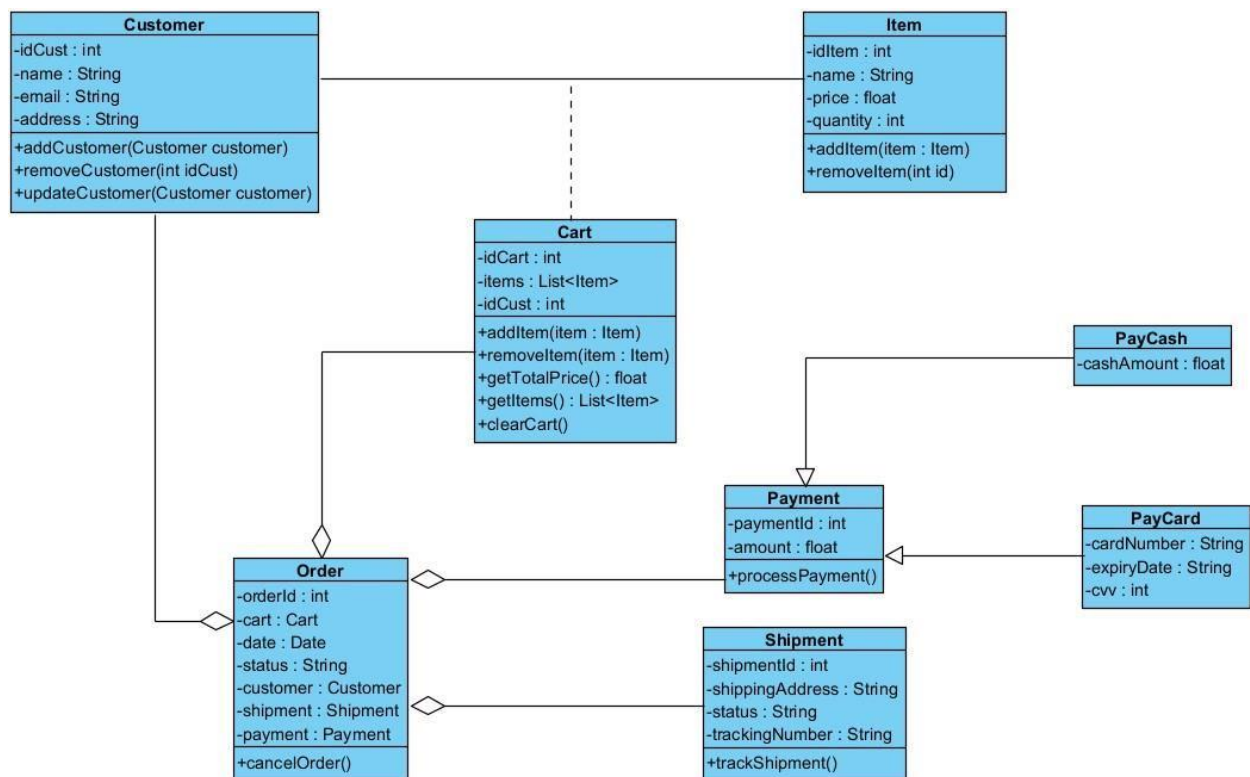
Họ và tên: Đinh Hoàng Anh

Mã sinh viên: B21DCCN143

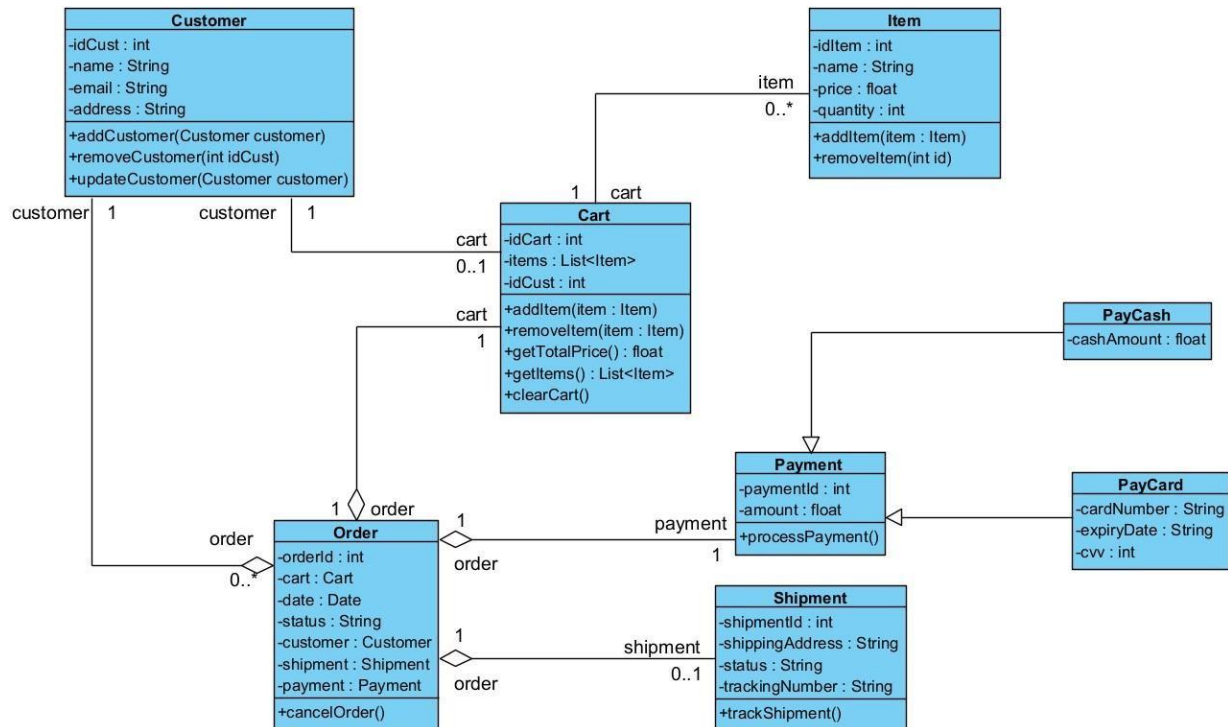
Nhóm lớp: 10

Nhóm bài tập: 10

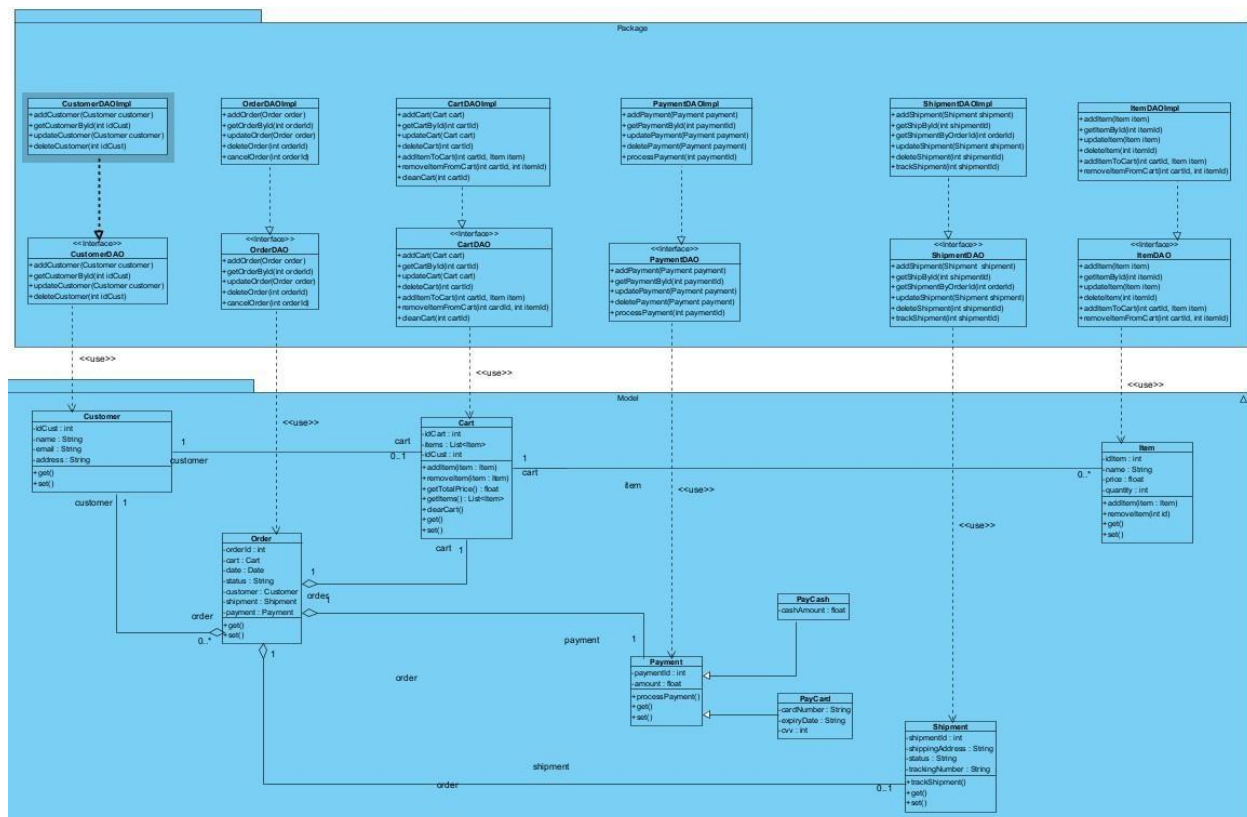
Bài 1. Biểu đồ lớp, phân tích nhóm.



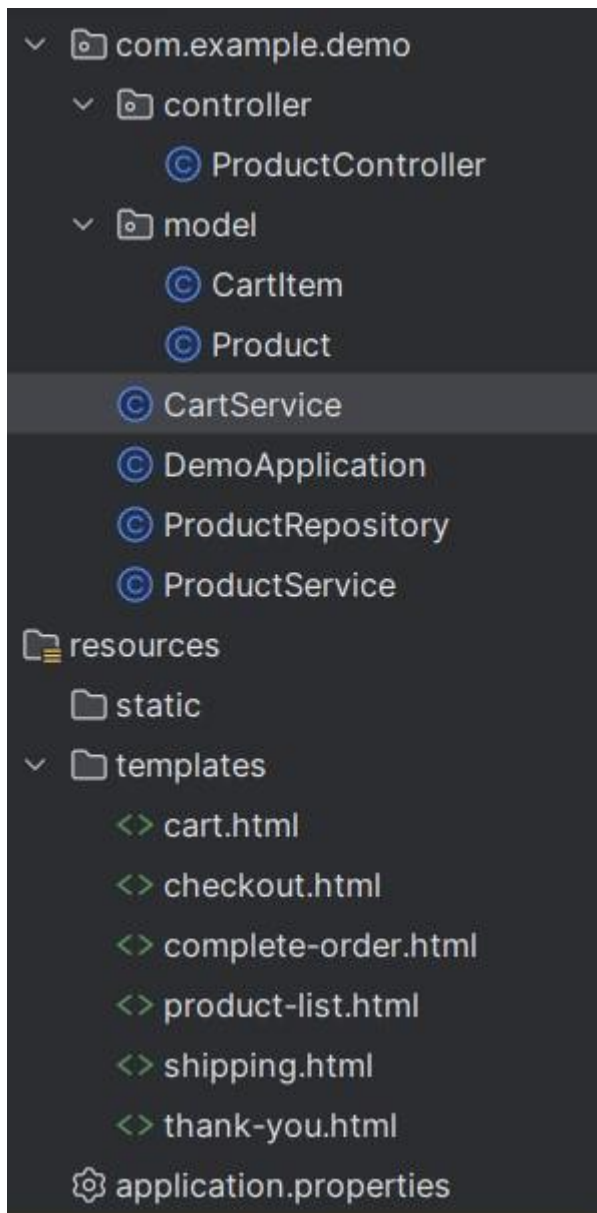
Bài 2. Biểu đồ lớp thiết kế



Bài 3. Package – Layer



Bài 4. Sinh code từ thiết kế



```

@Controller
@RequestMapping("/products")
public class ProductController {

    @Autowired
    private ProductService productService;
    @Autowired
    private CartService cartService;

    @GetMapping
    public String showProducts(Model model) {
        List<Product> products = productService.getAllProducts();
        model.addAttribute(attributeName: "products", products);
        return "product-list";
    }

    @GetMapping("/add-to-cart")
    public String addToCart(@RequestParam("id") Long productId, @RequestParam("quantity") int quantity, HttpSession session) {
        // Logic to add product to cart stored in session
        Product product = productService.getProductById(productId);
        if (product != null) {
            cartService.addToCart(product, quantity, session);
            System.out.println("Sản phẩm đã được thêm vào giỏ hàng: " + product.name);
        } else {
            System.out.println("Sản phẩm không tồn tại với ID: " + productId);
        }
        return "redirect:/products"; // Sau khi thêm vào giỏ hàng, chuyển hướng về trang sản phẩm
    }
}

```

```

@GetMapping("/checkout")
public String checkout() {
    // Display payment options
    return "checkout";
}

@PostMapping("/select-payment")
public String selectPaymentMethod(@RequestParam("paymentMethod") String paymentMethod, HttpSession session) {
    // Store selected payment method
    session.setAttribute(s: "paymentMethod", paymentMethod);
    return "redirect:/products/shipping";
}

@GetMapping("/shipping")
public String selectShipping() { return "shipping"; }

@PostMapping("/complete-order")
public String completeOrder() { return "complete-order"; }

@GetMapping("/search")
public String showProductByName(@RequestParam("keyword") String keyword, Model model){
    List<Product> products = productService.searchProductsByName(keyword);
    model.addAttribute(attributeName: "products", products);
    return "product-list";
}

```

```

@GetMapping("/cart")
public String showCart(HttpSession session, Model model) {
    model.addAttribute( attributeName: "cartItems", cartService.getCart(session));
    List<CartItem> cartItems = cartService.getCart(session);
    double total = 0;
    for (int i=0; i<cartItems.size(); i++)total += cartItems.get(i).getTotal();
    model.addAttribute( attributeName: "total", total);
    return "cart"; // Hiển thị trang giỏ hàng
}

@PostMapping("/feedback")
public String handleFeedback(@RequestParam("name") String name,
                             @RequestParam("email") String email,
                             @RequestParam("message") String message) {
    // Xử lý feedback (lưu vào database, gửi email, etc.)
    System.out.println("Feedback received from: " + name);
    System.out.println("Email: " + email);
    System.out.println("Message: " + message);

    // Có thể redirect đến một trang khác sau khi gửi feedback
    return "redirect:/products/thank-you"; // Trang cảm ơn hoặc quay về trang chính
}

@GetMapping("/thank-you")
public String showThankYou() { return "thank-you"; }

```

```

@GetMapping("/thank-you")
public String showThankYou() { return "thank-you"; }

@GetMapping("/products/clear")
public String showPro(Model model){
    List<Product> products = productService.getAllProducts();
    model.addAttribute( attributeName: "products", products);
    return "redirect:/products";
}

```

Figure 9: ProductController

```

public class CartItem {
    private Product product;
    private int quantity;
    private double total;

    public CartItem(Product product, int quantity) {
        this.product = product;
        this.quantity = quantity;
        this.total = product.price * quantity;
    }

    public Product getProduct() { return product; }

    public void setProduct(Product product) { this.product = product; }

    public int getQuantity() { return quantity; }

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

    public double getTotal() { return total; }
}

```

Figure 10: Model CartItem

```

public class Product {

    public Long id;
    public String name;
    public String description;
    public Double price;

    // Constructors
    public Product() {}

    public Product(Long id, String name, String description, Double price) {
        this.id = id;
        this.name = name;
        this.description = description;
        this.price = price;
    }
}

```

Figure 11: Model Product

2 lớp trên lưu thông tin của sản phẩm và các sản phẩm trong giỏ hàng


```

@Service
public class CartService {

    private static final String CART_SESSION_KEY = "cart";

    public List<CartItem> getCart(HttpSession session) {
        List<CartItem> cart = (List<CartItem>) session.getAttribute(CART_SESSION_KEY);
        if (cart == null) {
            cart = new ArrayList<>();
            session.setAttribute(CART_SESSION_KEY, cart);
        }
        return cart;
    }

    public void addToCart(Product product, int quantity, HttpSession session) {
        List<CartItem> cart = getCart(session);
        boolean exists = false;

        for (CartItem item : cart) {
            if (item.getProduct().id.equals(product.id)) {
                item.setQuantity(item.getQuantity() + quantity); // Tăng số lượng nếu sản phẩm đã có trong giỏ hàng
                exists = true;
                break;
            }
        }

        if (!exists) {
            cart.add(new CartItem(product, quantity)); // Thêm sản phẩm mới nếu chưa có trong giỏ hàng
        }

        session.setAttribute(CART_SESSION_KEY, cart); // Cập nhật giỏ hàng trong session
    }

    public void removeFromCart(Long productId, HttpSession session) {
        List<CartItem> cart = getCart(session);
        cart.removeIf(item -> item.getProduct().id.equals(productId));
        session.setAttribute(CART_SESSION_KEY, cart);
    }

    public void clearCart(HttpSession session) { session.removeAttribute(CART_SESSION_KEY); }
}

```

Figure 12: CartService

Xử lý các phương thức của giỏ hàng CRUD

```

@Repository
public class ProductRepository {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    // RowMapper để ánh xạ kết quả từ ResultSet vào đối tượng Product
    private RowMapper<Product> productRowMapper = new RowMapper<Product>() {
        @Override
        public Product mapRow(ResultSet rs, int rowNum) throws SQLException {
            return new Product(
                rs.getLong( columnLabel: "id"),
                rs.getString( columnLabel: "name"),
                rs.getString( columnLabel: "description"),
                rs.getDouble( columnLabel: "price")
            );
        }
    };

    // Lấy danh sách tất cả sản phẩm
    public List<Product> findAll() {
        String sql = "SELECT * FROM product";
        return jdbcTemplate.query(sql, productRowMapper);
    }
}

```

```

// Tìm sản phẩm theo ID
public Product findById(Long id) {
    String sql = "SELECT * FROM product WHERE id = ?";
    return jdbcTemplate.queryForObject(sql, new Object[]{id}, productRowMapper);
}

public List<Product> findByName(String name) {
    String sql = "SELECT * FROM product WHERE name LIKE ?";
    return jdbcTemplate.query(sql, new Object[]{"%" + name + "%"}, productRowMapper);
}
}

```

Figure 13: ProductRepository

CRUD của Product


```

package com.example.demo;

import com.example.demo.model.Product;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class ProductService {

    @Autowired
    private ProductRepository productRepository;

    public List<Product> getAllProducts() {
        return productRepository.findAll();
    }

    public Product getProductById(Long id) {
        return productRepository.findById(id);
    }

    public List<Product> searchProductsByName(String name) {
        return productRepository.findByName(name);
    }
}

```

Figure 14: ProductService

Gọi các phương thức trong ProductRepository

```

<> cart.html
<> checkout.html
<> complete-order.html
<> product-list.html
<> shipping.html
<> thank-you.html

```

Figure 15: Giao diện

5. GIAO DIỆN THIẾT KẾ THEO CODE

