Project Group Number/Name: Group Number 8
Group Members:

- Christ Rodrigues
- Yash Zaveri
- Viral Suchak
- Kevin Rodrigues
- Omkar Vaity

Project Topic/Name: Google Product Inventory Management System

Tech Stack:

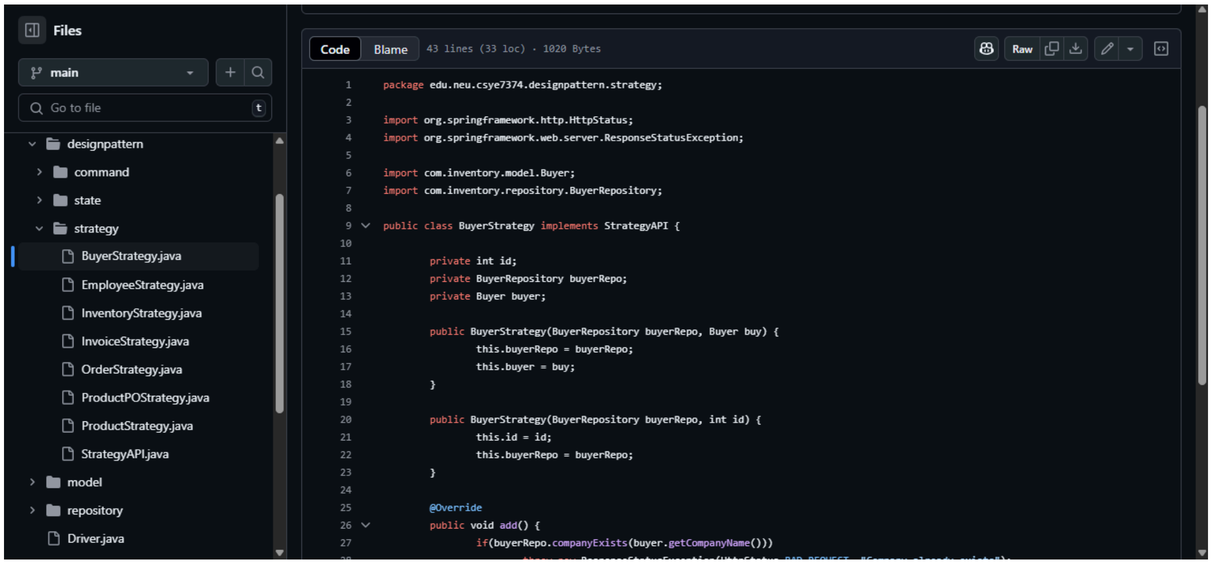
- **Frontend:** React and Bootstrap
- **Backend:** Java EE, Spring, Apache Tomcat
- **Database:** MySQL
- **Database Connectivity:** Hibernate

Functionalities Implemented:

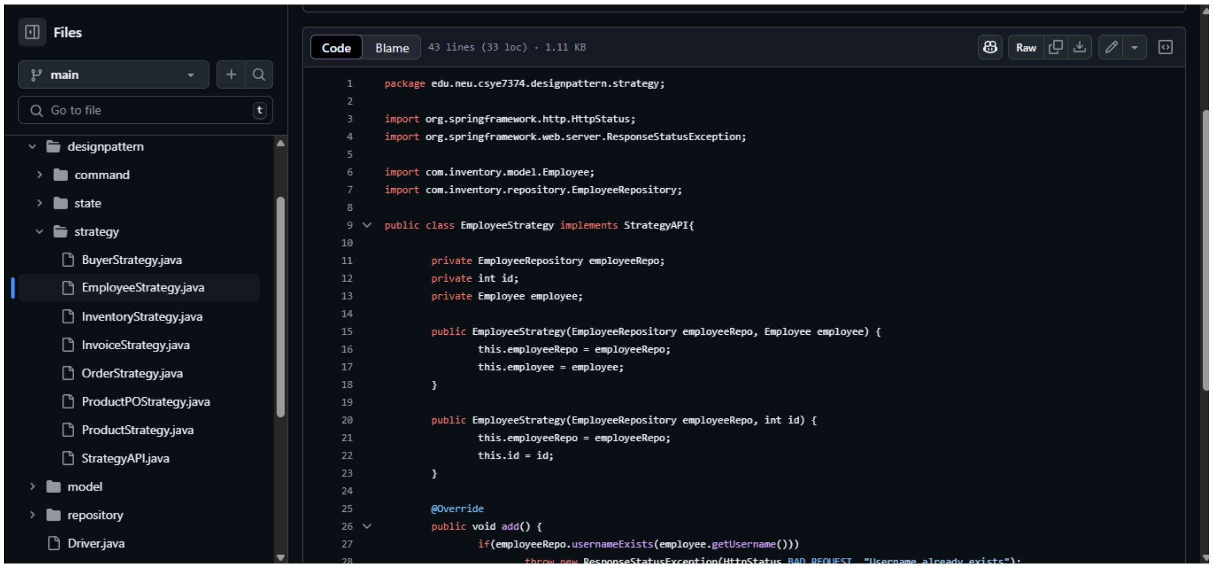
1. **Inventory Management Core Features**:
    - Product management (CRUD operations for products)
    - Employee management with different access levels based on designation
    - Buyer/Customer management
    - Purchase order processing
    - Invoice generation and management
2. **Technical Architecture**:
    - Spring Boot backend with REST controllers
    - Hibernate for database interactions using SessionFactory
    - React frontend for user interface
    - Authentication system for employee login
3. **Document Generation**:
    - PDF generation for invoices using iText library
    - The system can create and format invoice documents for purchase orders
4. **Business Logic**:
    - Inventory tracking and stock management
    - Purchase order processing including payment tracking
    - Employee role-based access control (Manager vs regular employees)

Design Patterns implemented:

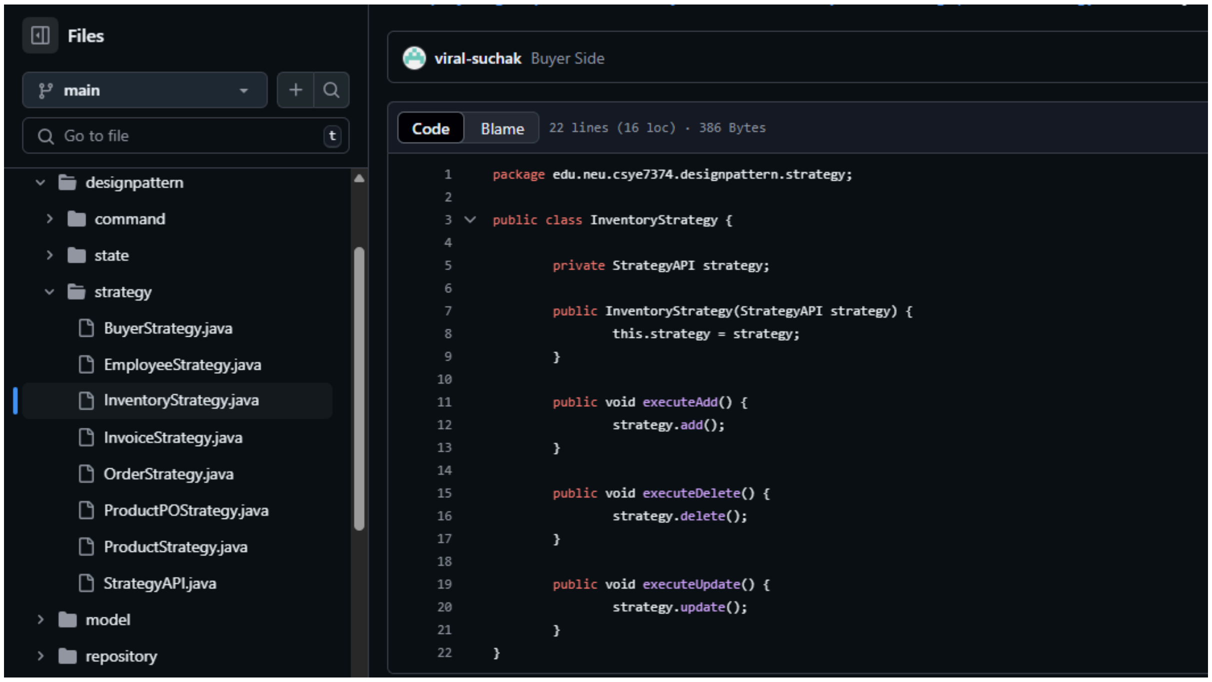
1) Strategy Design Pattern



```java
package edu.neu.csye7374.designpattern.strategy;

import org.springframework.http.HttpStatus;
import org.springframework.web.server.ResponseStatusException;

import com.inventory.model.Buyer;
import com.inventory.repository.BuyerRepository;

public class BuyerStrategy implements StrategyAPI {

        private int id;
        private BuyerRepository buyerRepo;
        private Buyer buyer;

        public BuyerStrategy(BuyerRepository buyerRepo, Buyer buy) {
                this.buyerRepo = buyerRepo;
                this.buyer = buy;
        }

        public BuyerStrategy(BuyerRepository buyerRepo, int id) {
                this.id = id;
                this.buyerRepo = buyerRepo;
        }

        @Override
        public void add() {
                if(buyerRepo.companyExists(buyer.getCompanyName()))
```



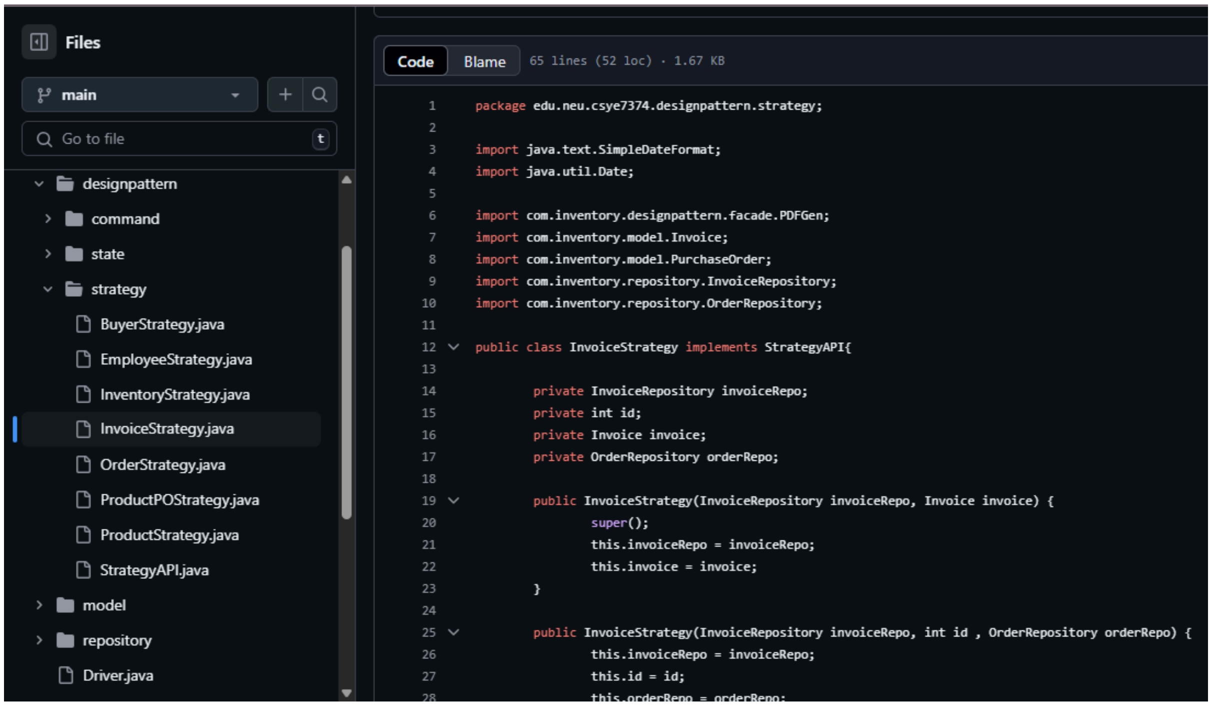
```java
package edu.neu.csye7374.designpattern.strategy;

import org.springframework.http.HttpStatus;
import org.springframework.web.server.ResponseStatusException;

import com.inventory.model.Employee;
import com.inventory.repository.EmployeeRepository;

public class EmployeeStrategy implements StrategyAPI{

        private EmployeeRepository employeeRepo;
        private int id;
        private Employee employee;

        public EmployeeStrategy(EmployeeRepository employeeRepo, Employee employee) {
                this.employeeRepo = employeeRepo;
                this.employee = employee;
        }

        public EmployeeStrategy(EmployeeRepository employeeRepo, int id) {
                this.employeeRepo = employeeRepo;
                this.id = id;
        }

        @Override
        public void add() {
                if(employeeRepo.usernameExists(employee.getUsername()))
```



viral-suchak Buyer Side

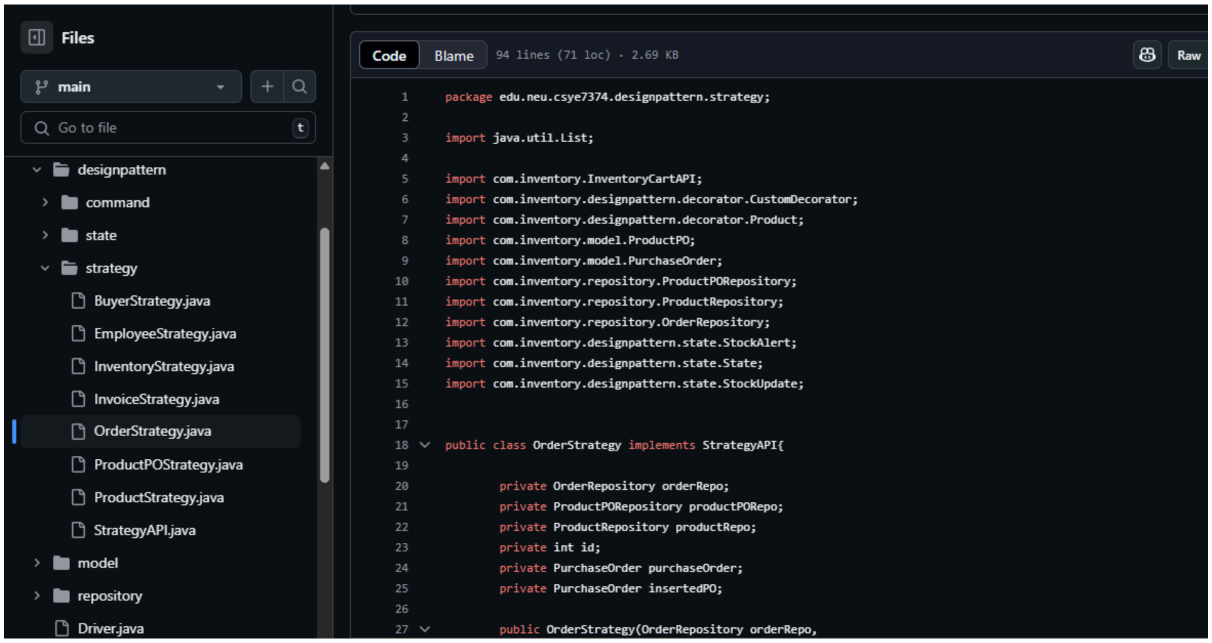
22 lines (16 loc) · 386 Bytes

```java
package edu.neu.csye7374.designpattern.strategy;

public class InventoryStrategy {

        private StrategyAPI strategy;

        public InventoryStrategy(StrategyAPI strategy) {
                this.strategy = strategy;
        }

        public void executeAdd() {
                strategy.add();
        }

        public void executeDelete() {
                strategy.delete();
        }

        public void executeUpdate() {
                strategy.update();
        }
}
```



65 lines (52 loc) · 1.67 KB

```java
package edu.neu.csye7374.designpattern.strategy;

import java.text.SimpleDateFormat;
import java.util.Date;

import com.inventory.designpattern.facade.PDFGen;
import com.inventory.model.Invoice;
import com.inventory.model.PurchaseOrder;
import com.inventory.repository.InvoiceRepository;
import com.inventory.repository.OrderRepository;

public class InvoiceStrategy implements StrategyAPI{

        private InvoiceRepository invoiceRepo;
        private int id;
        private Invoice invoice;
        private OrderRepository orderRepo;

        public InvoiceStrategy(InvoiceRepository invoiceRepo, Invoice invoice) {
                super();
                this.invoiceRepo = invoiceRepo;
                this.invoice = invoice;
        }

        public InvoiceStrategy(InvoiceRepository invoiceRepo, int id , OrderRepository orderRepo) {
                this.invoiceRepo = invoiceRepo;
                this.id = id;
                this.orderRepo = orderRepo;
```



Files

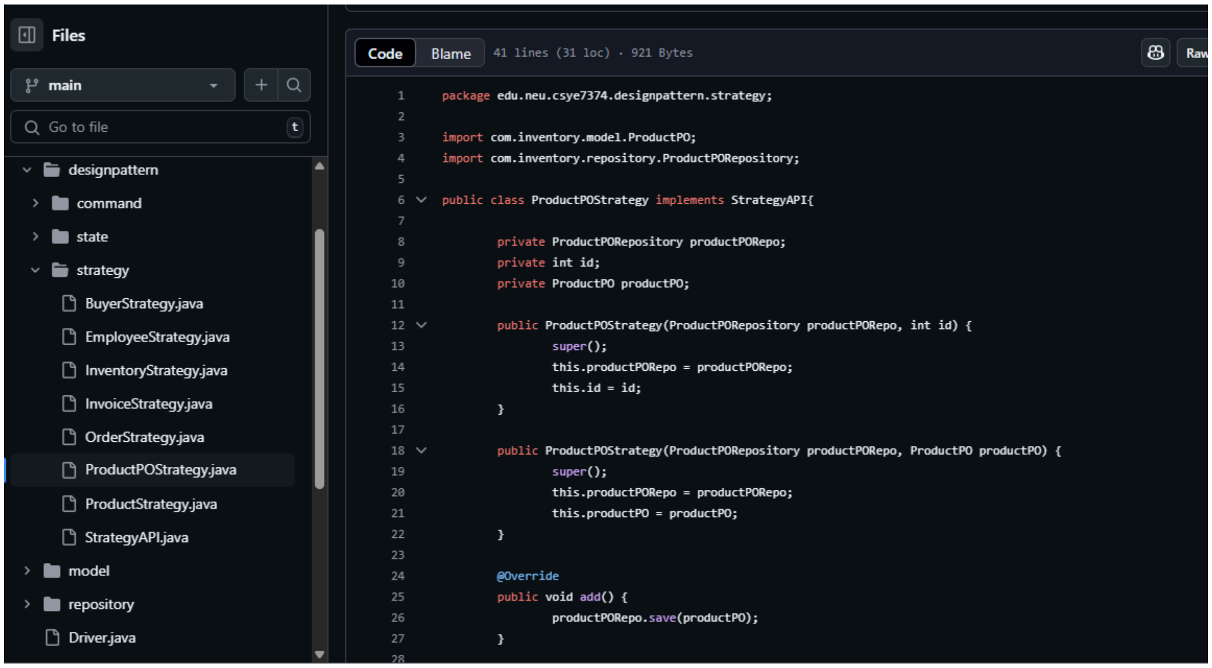
main

Go to file

- designpattern
  - command
  - state
  - strategy
    - BuyerStrategy.java
    - EmployeeStrategy.java
    - InventoryStrategy.java
    - InvoiceStrategy.java
    - OrderStrategy.java
    - ProductPOStrategy.java
    - ProductStrategy.java
    - StrategyAPI.java
  - model
  - repository
  - Driver.java

Code Blame 94 lines (71 loc) · 2.69 KB Raw

```
package edu.neu.csye7374.designpattern.strategy;

import java.util.List;

import com.inventory.InventoryCartAPI;
import com.inventory.designpattern.decorator.CustomDecorator;
import com.inventory.designpattern.decorator.Product;
import com.inventory.model.ProductPO;
import com.inventory.model.PurchaseOrder;
import com.inventory.repository.ProductPORepository;
import com.inventory.repository.ProductRepository;
import com.inventory.repository.OrderRepository;
import com.inventory.designpattern.state.StockAlert;
import com.inventory.designpattern.state.State;
import com.inventory.designpattern.state.StockUpdate;


public class OrderStrategy implements StrategyAPI{

        private OrderRepository orderRepo;
        private ProductPORepository productPORepo;
        private ProductRepository productRepo;
        private int id;
        private PurchaseOrder purchaseOrder;
        private PurchaseOrder insertedPO;

        public OrderStrategy(OrderRepository orderRepo,
```



Files

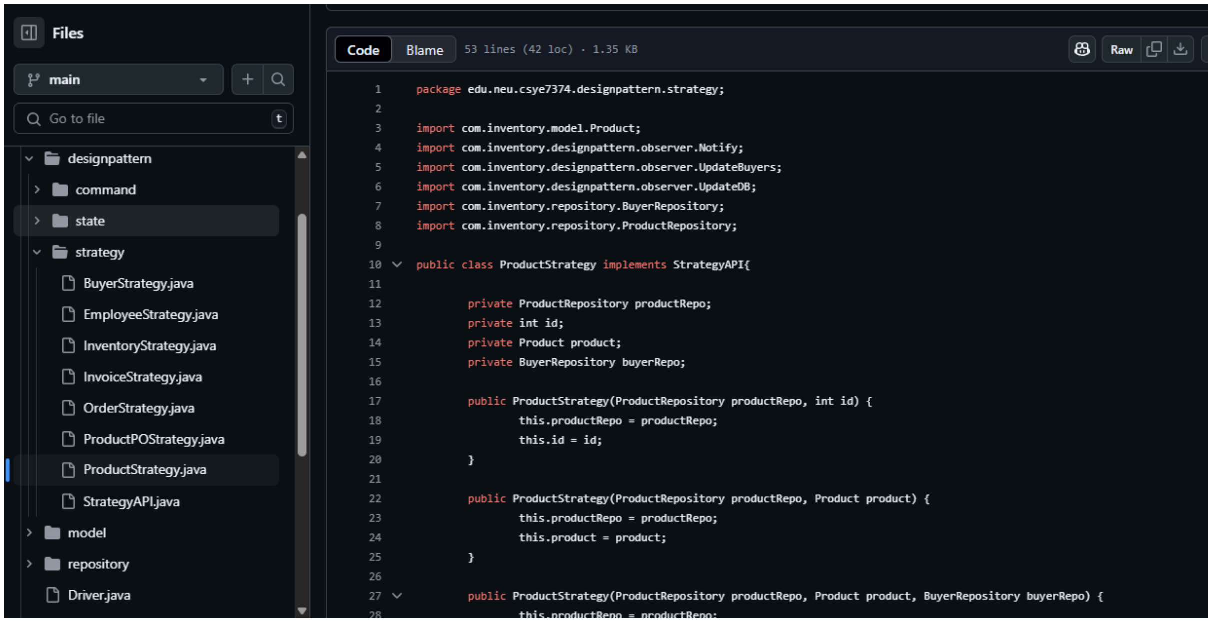
main

Go to file

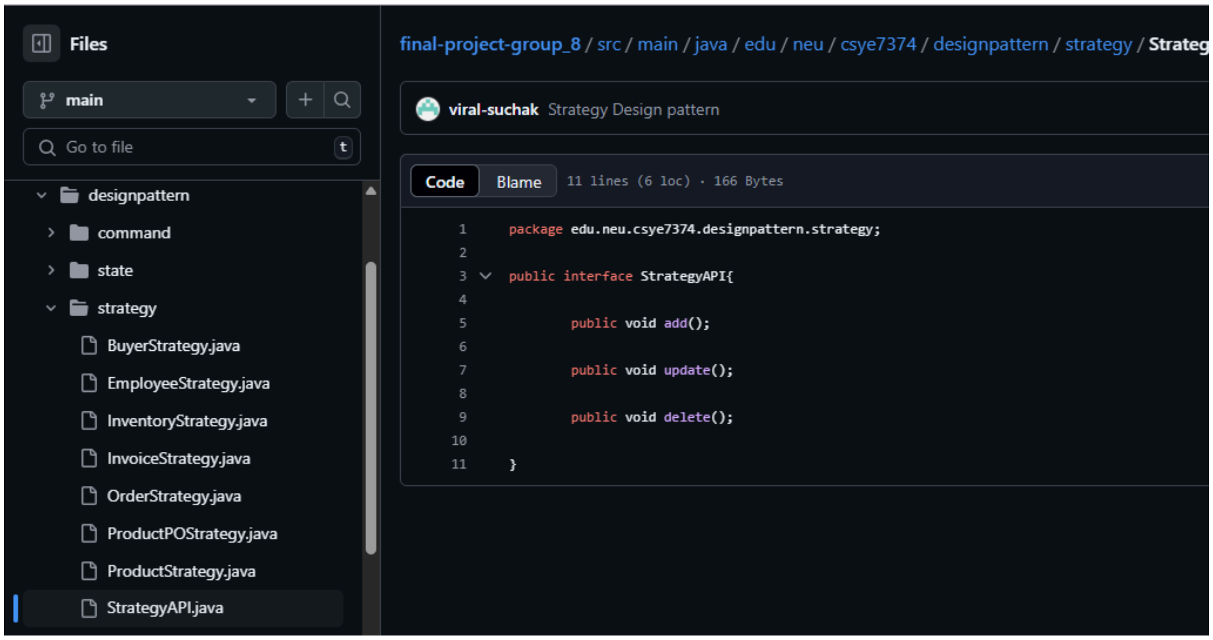
- designpattern
  - command
  - state
  - strategy
    - BuyerStrategy.java
    - EmployeeStrategy.java
    - InventoryStrategy.java
    - InvoiceStrategy.java
    - OrderStrategy.java
    - ProductPOStrategy.java
    - ProductStrategy.java
    - StrategyAPI.java
  - model
  - repository
  - Driver.java

Code Blame 41 lines (31 loc) · 921 Bytes Raw

```
package edu.neu.csye7374.designpattern.strategy;

import com.inventory.model.ProductPO;
import com.inventory.repository.ProductPORepository;

public class ProductPOStrategy implements StrategyAPI{

        private ProductPORepository productPORepo;
        private int id;
        private ProductPO productPO;

        public ProductPOStrategy(ProductPORepository productPORepo, int id) {
                super();
                this.productPORepo = productPORepo;
                this.id = id;
        }

        public ProductPOStrategy(ProductPORepository productPORepo, ProductPO productPO) {
                super();
                this.productPORepo = productPORepo;
                this.productPO = productPO;
        }

        @Override
        public void add() {
                productPORepo.save(productPO);
        }

```

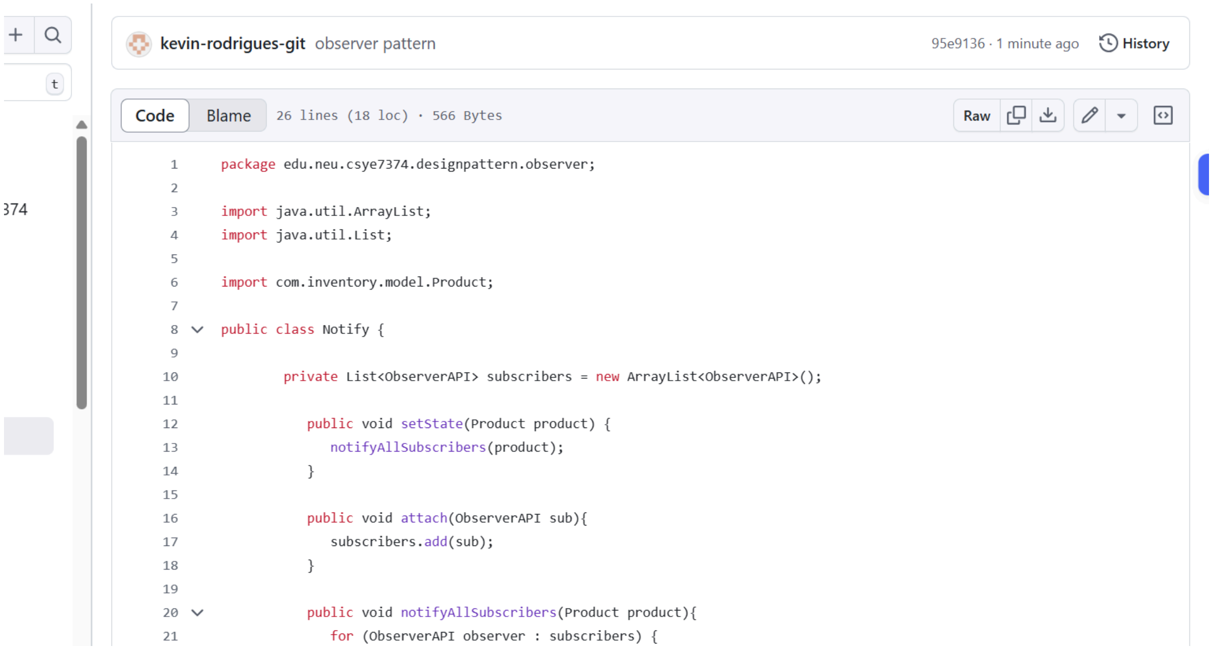


```java
package edu.neu.csye7374.designpattern.strategy;

import com.inventory.model.Product;
import com.inventory.designpattern.observer.Notify;
import com.inventory.designpattern.observer.UpdateBuyers;
import com.inventory.designpattern.observer.UpdateDB;
import com.inventory.repository.BuyerRepository;
import com.inventory.repository.ProductRepository;

public class ProductStrategy implements StrategyAPI{

        private ProductRepository productRepo;
        private int id;
        private Product product;
        private BuyerRepository buyerRepo;

        public ProductStrategy(ProductRepository productRepo, int id) {
                this.productRepo = productRepo;
                this.id = id;
        }

        public ProductStrategy(ProductRepository productRepo, Product product) {
                this.productRepo = productRepo;
                this.product = product;
        }

        public ProductStrategy(ProductRepository productRepo, Product product, BuyerRepository buyerRepo) {
                this.productRepo = productRepo;
```



```java
package edu.neu.csye7374.designpattern.strategy;

public interface StrategyAPI{

        public void add();

        public void update();

        public void delete();

}
```

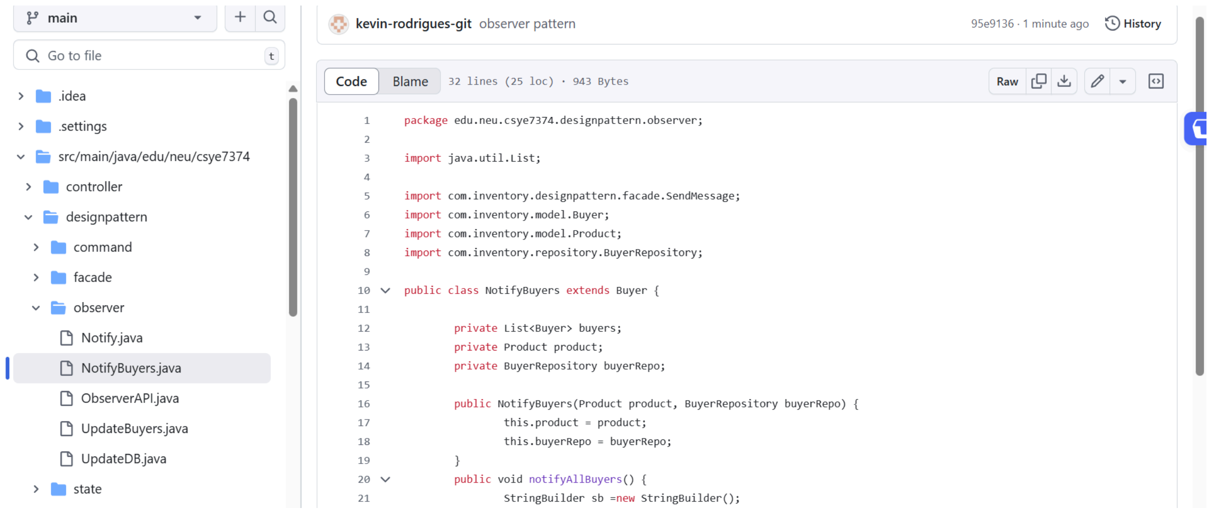
2) Observer Design Pattern



kevin-rodrigues-git observer pattern 95e9136 · 1 minute ago History

Code Blame 26 lines (18 loc) · 566 Bytes

```java
package edu.neu.csye7374.designpattern.observer;

import java.util.ArrayList;
import java.util.List;

import com.inventory.model.Product;

public class Notify {

        private List<ObserverAPI> subscribers = new ArrayList<ObserverAPI>();

          public void setState(Product product) {
             notifyAllSubscribers(product);
          }

          public void attach(ObserverAPI sub){
             subscribers.add(sub);
          }

          public void notifyAllSubscribers(Product product){
             for (ObserverAPI observer : subscribers) {
```



main

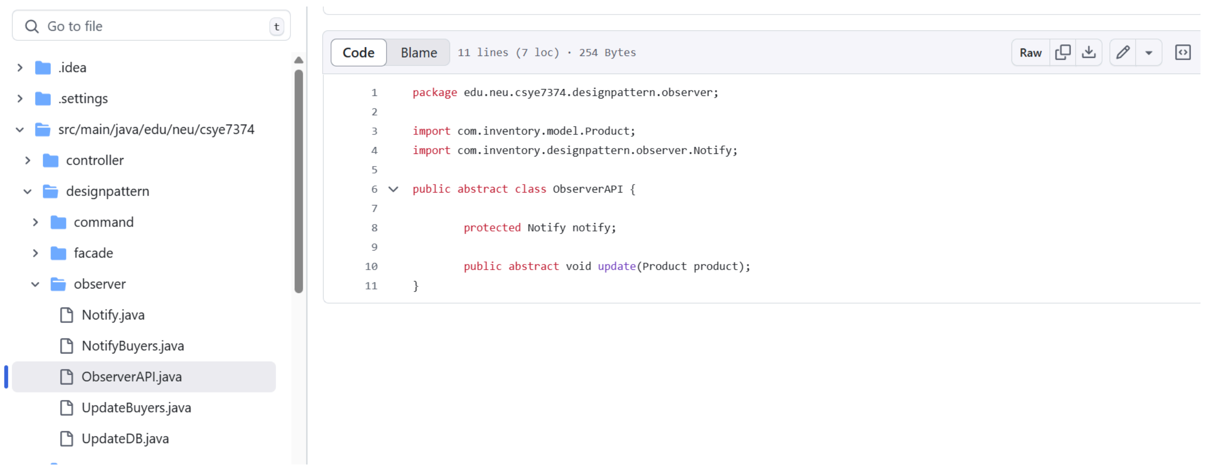
Go to file

- .idea
- .settings
- src/main/java/edu/neu/csye7374
  - controller
  - designpattern
    - command
    - facade
    - observer
      - Notify.java
      - NotifyBuyers.java
      - ObserverAPI.java
      - UpdateBuyers.java
      - UpdateDB.java
    - state

kevin-rodrigues-git observer pattern 95e9136 · 1 minute ago History

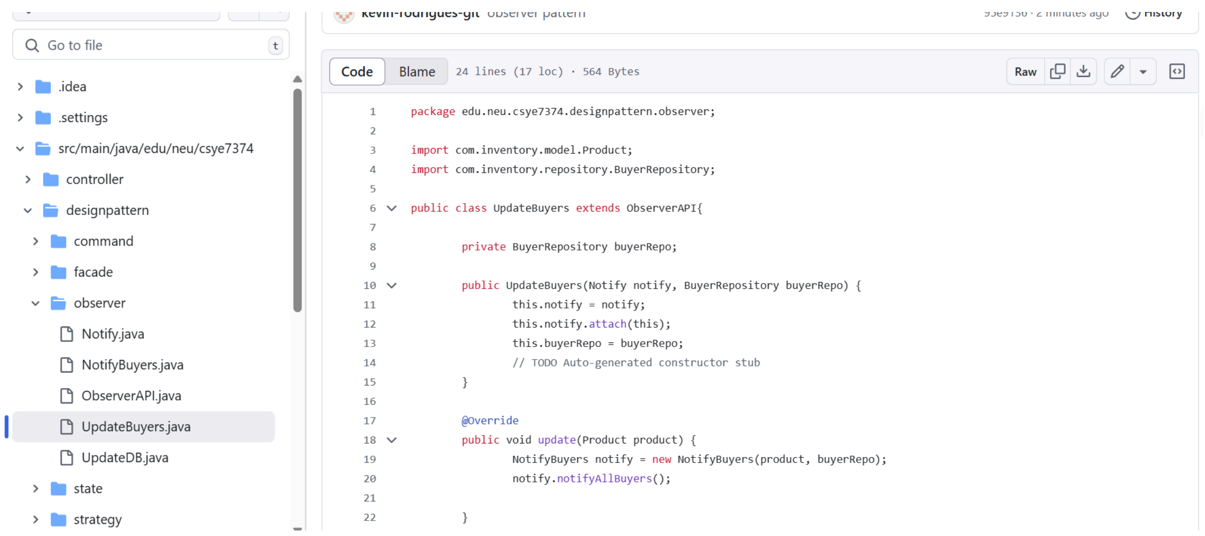
Code Blame 32 lines (25 loc) · 943 Bytes

```java
package edu.neu.csye7374.designpattern.observer;

import java.util.List;

import com.inventory.designpattern.facade.SendMessage;
import com.inventory.model.Buyer;
import com.inventory.model.Product;
import com.inventory.repository.BuyerRepository;

public class NotifyBuyers extends Buyer {

        private List<Buyer> buyers;
        private Product product;
        private BuyerRepository buyerRepo;

        public NotifyBuyers(Product product, BuyerRepository buyerRepo) {
                this.product = product;
                this.buyerRepo = buyerRepo;
        }
        public void notifyAllBuyers() {
                StringBuilder sb =new StringBuilder();
```



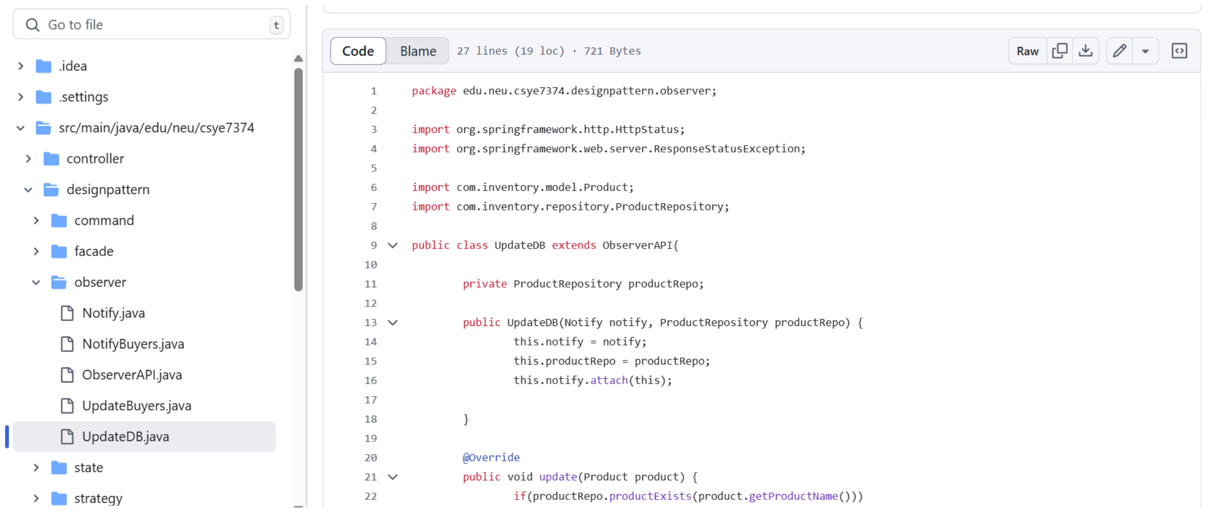
11 lines (7 loc) · 254 Bytes

```java
package edu.neu.csye7374.designpattern.observer;

import com.inventory.model.Product;
import com.inventory.designpattern.observer.Notify;

public abstract class ObserverAPI {

	protected Notify notify;

	public abstract void update(Product product);
}
```



24 lines (17 loc) · 564 Bytes

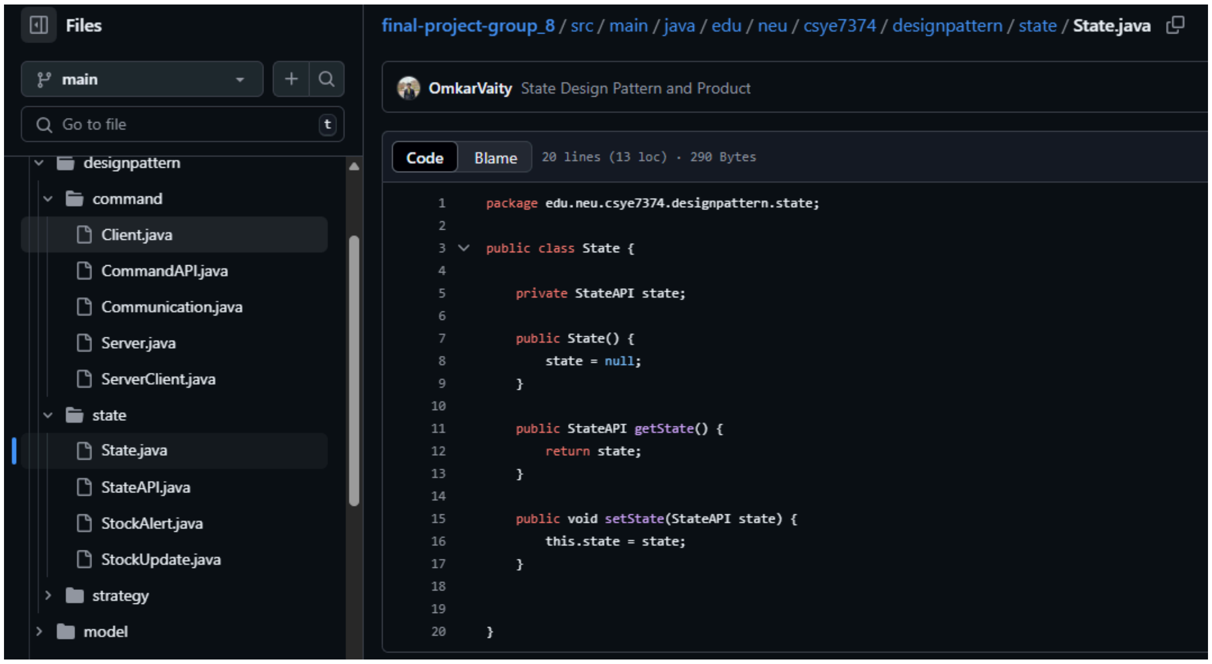
```java
package edu.neu.csye7374.designpattern.observer;

import com.inventory.model.Product;
import com.inventory.repository.BuyerRepository;

public class UpdateBuyers extends ObserverAPI{

	private BuyerRepository buyerRepo;

	public UpdateBuyers(Notify notify, BuyerRepository buyerRepo) {
		this.notify = notify;
		this.notify.attach(this);
		this.buyerRepo = buyerRepo;
		// TODO Auto-generated constructor stub
	}

	@Override
	public void update(Product product) {
		NotifyBuyers notify = new NotifyBuyers(product, buyerRepo);
		notify.notifyAllBuyers();

	}
```



27 lines (19 loc) · 721 Bytes

```java
package edu.neu.csye7374.designpattern.observer;

import org.springframework.http.HttpStatus;
import org.springframework.web.server.ResponseStatusException;

import com.inventory.model.Product;
import com.inventory.repository.ProductRepository;

public class UpdateDB extends ObserverAPI{

	private ProductRepository productRepo;

	public UpdateDB(Notify notify, ProductRepository productRepo) {
		this.notify = notify;
		this.productRepo = productRepo;
		this.notify.attach(this);

	}

	@Override
	public void update(Product product) {
		if(productRepo.productExists(product.getProductName()))
```

3) State Design Pattern

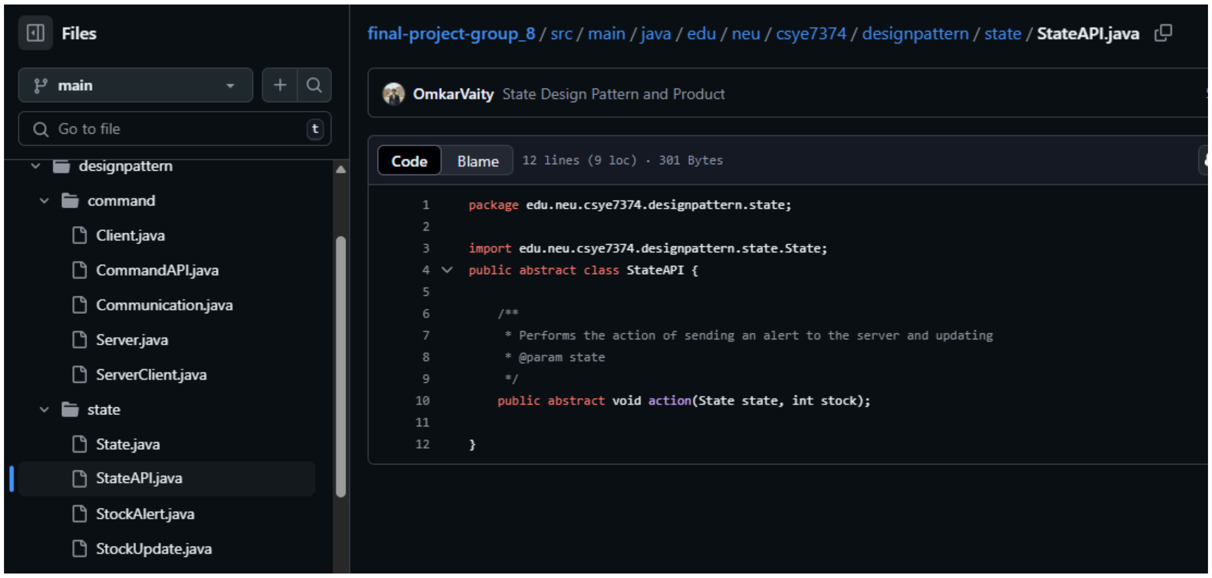


final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / state / State.java

OmkarVaity State Design Pattern and Product

20 lines (13 loc) · 290 Bytes

```java
package edu.neu.csye7374.designpattern.state;

public class State {

    private StateAPI state;

    public State() {
        state = null;
    }

    public StateAPI getState() {
        return state;
    }

    public void setState(StateAPI state) {
        this.state = state;
    }


}
```

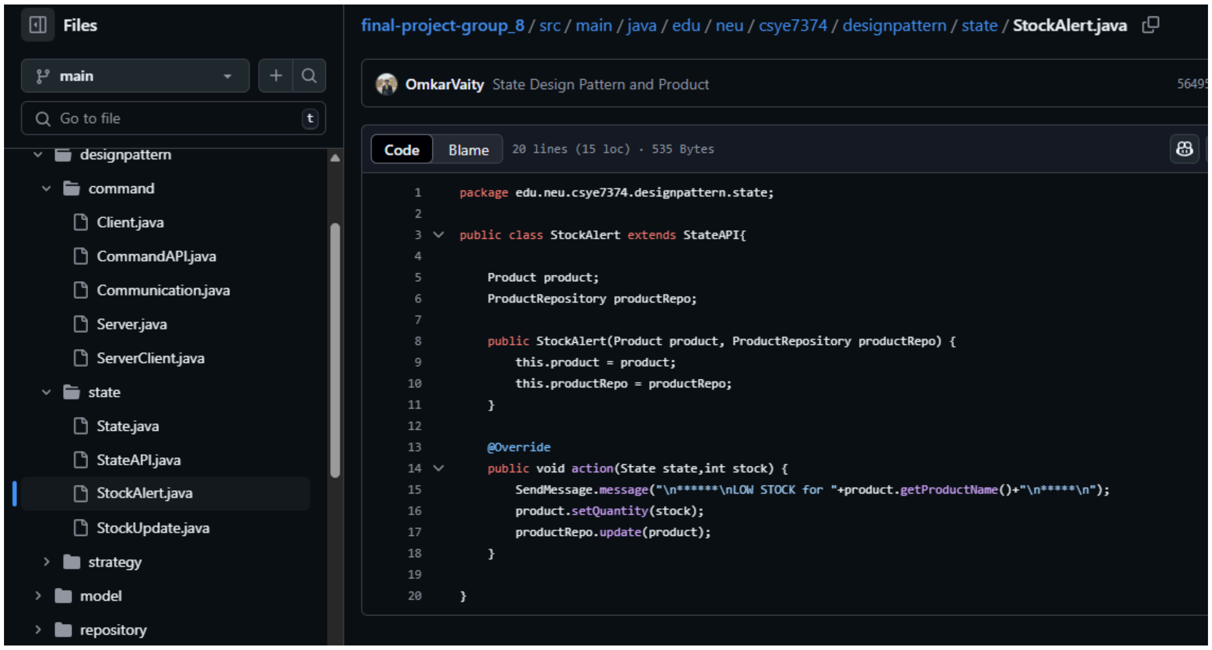


final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / state / StateAPI.java

OmkarVaity State Design Pattern and Product

12 lines (9 loc) · 301 Bytes

```java
package edu.neu.csye7374.designpattern.state;

import edu.neu.csye7374.designpattern.state.State;
public abstract class StateAPI {

    /**
     * Performs the action of sending an alert to the server and updating
     * @param state
     */
    public abstract void action(State state, int stock);

}
```

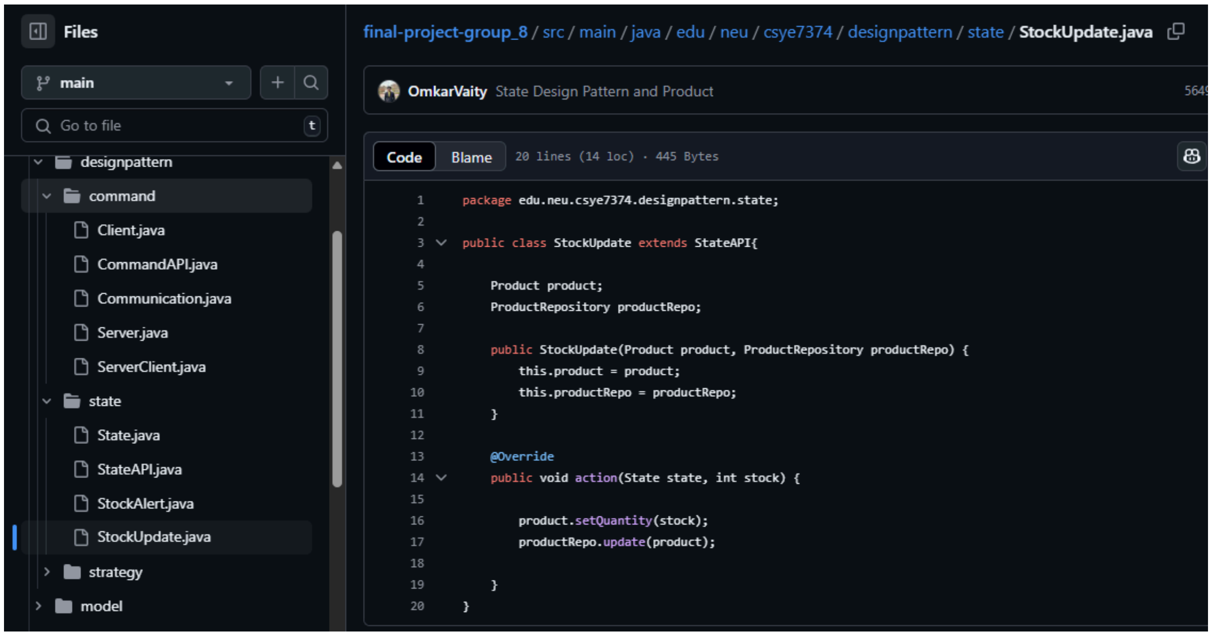


final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / state / StockAlert.java

OmkarVaity State Design Pattern and Product

20 lines (15 loc) · 535 Bytes

```java
package edu.neu.csye7374.designpattern.state;

public class StockAlert extends StateAPI{

    Product product;
    ProductRepository productRepo;

    public StockAlert(Product product, ProductRepository productRepo) {
        this.product = product;
        this.productRepo = productRepo;
    }

    @Override
    public void action(State state,int stock) {
        SendMessage.message("\n******\nLOW STOCK for "+product.getProductName()+"\n*****\n");
        product.setQuantity(stock);
        productRepo.update(product);
    }

}
```



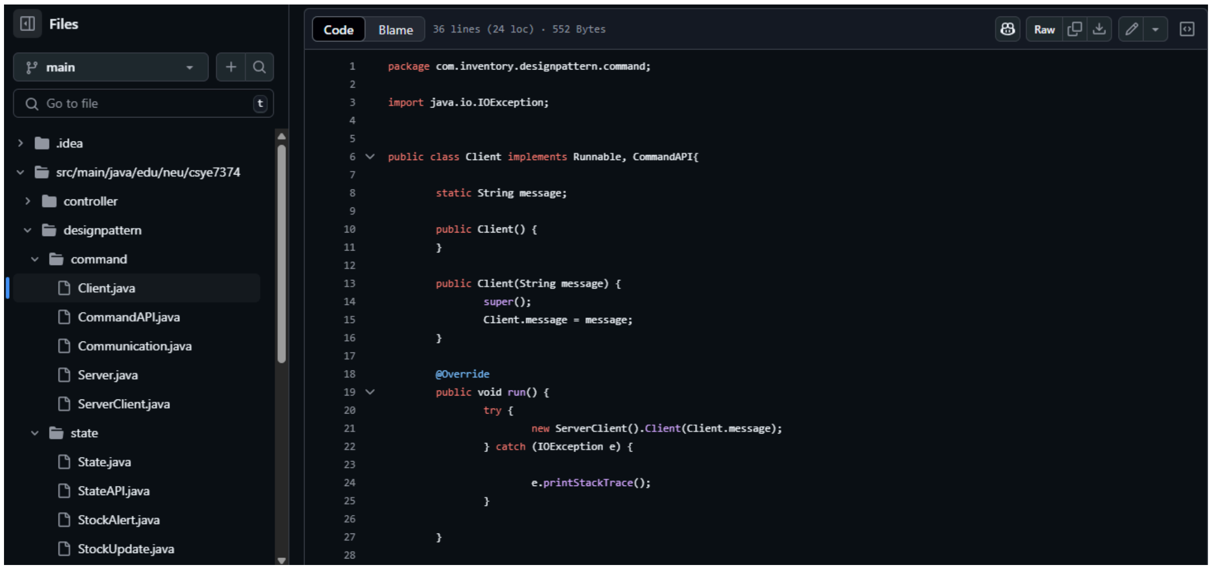
final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / state / StockUpdate.java

OmkarVaity State Design Pattern and Product

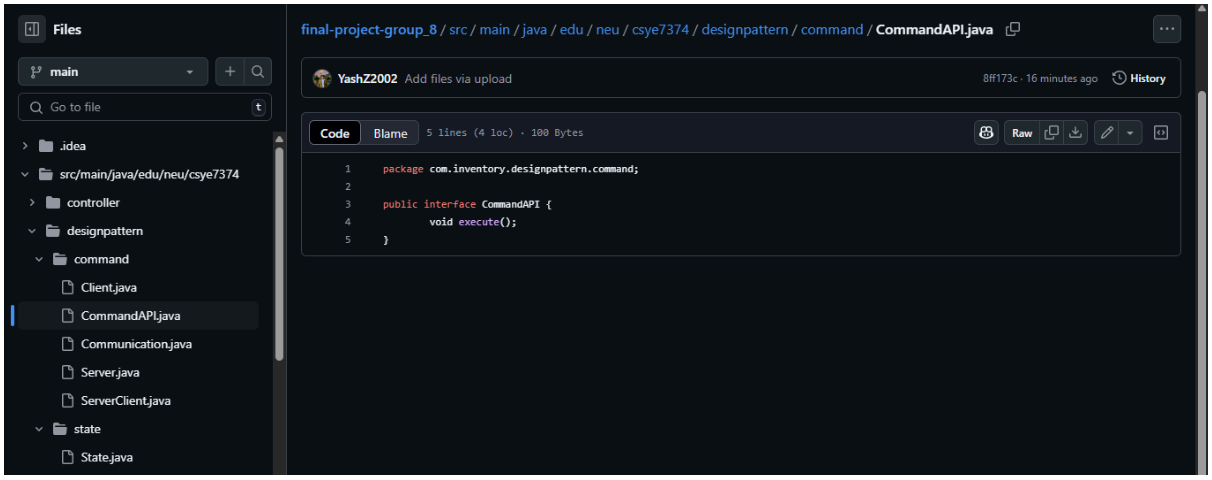
20 lines (14 loc) · 445 Bytes

```java
package edu.neu.csye7374.designpattern.state;

public class StockUpdate extends StateAPI{

    Product product;
    ProductRepository productRepo;

    public StockUpdate(Product product, ProductRepository productRepo) {
        this.product = product;
        this.productRepo = productRepo;
    }

    @Override
    public void action(State state, int stock) {

        product.setQuantity(stock);
        productRepo.update(product);

    }
}
```

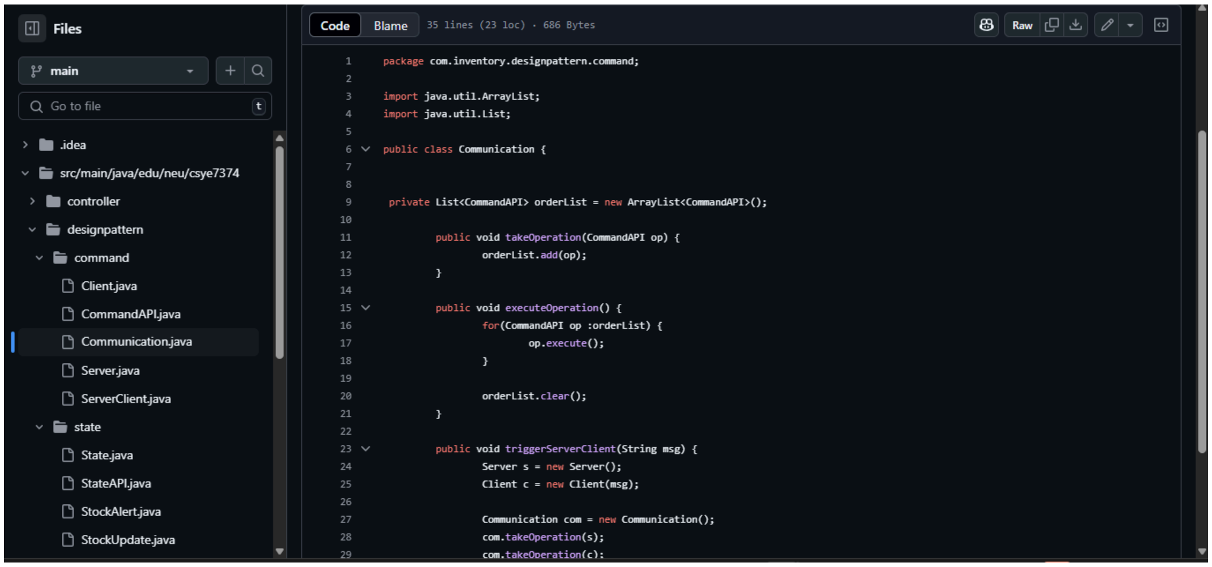
4) Command Design Pattern



```java
package com.inventory.designpattern.command;

import java.io.IOException;


public class Client implements Runnable, CommandAPI{

        static String message;

        public Client() {
        }

        public Client(String message) {
                super();
                Client.message = message;
        }

        @Override
        public void run() {
                try {
                        new ServerClient().Client(Client.message);
                } catch (IOException e) {

                        e.printStackTrace();
                }

        }

```

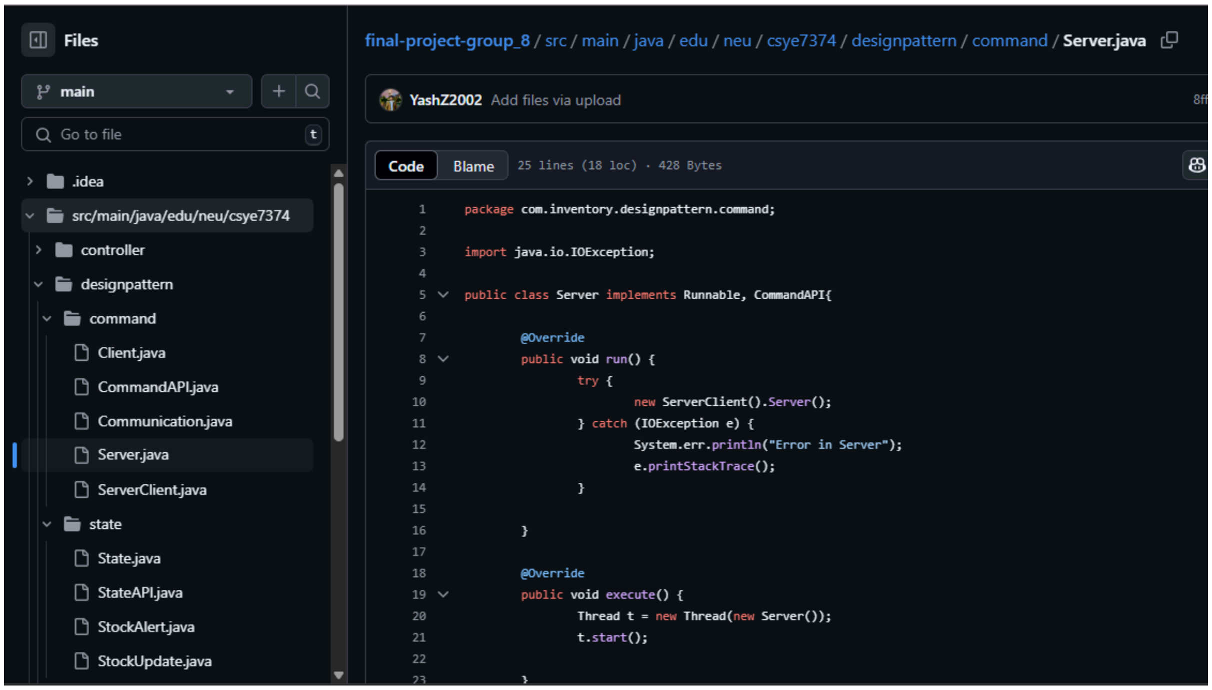

Files
main
Go to file
.idea
src/main/java/edu/neu/csye7374
controller
designpattern
command
Client.java
CommandAPI.java
Communication.java
Server.java
ServerClient.java
state
State.java
final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / command / CommandAPI.java
YashZ2002 Add files via upload
8ff173c · 16 minutes ago
History
Code
Blame
5 lines (4 loc) · 100 Bytes
Raw

```java
package com.inventory.designpattern.command;

public interface CommandAPI {
        void execute();
}
```



35 lines (23 loc) · 686 Bytes

```java
package com.inventory.designpattern.command;

import java.util.ArrayList;
import java.util.List;

public class Communication {


 private List<CommandAPI> orderList = new ArrayList<CommandAPI>();

	public void takeOperation(CommandAPI op) {
		orderList.add(op);
	}

	public void executeOperation() {
		for(CommandAPI op :orderList) {
			op.execute();
		}

		orderList.clear();
	}

	public void triggerServerClient(String msg) {
		Server s = new Server();
		Client c = new Client(msg);

		Communication com = new Communication();
		com.takeOperation(s);
		com.takeOperation(c);
```

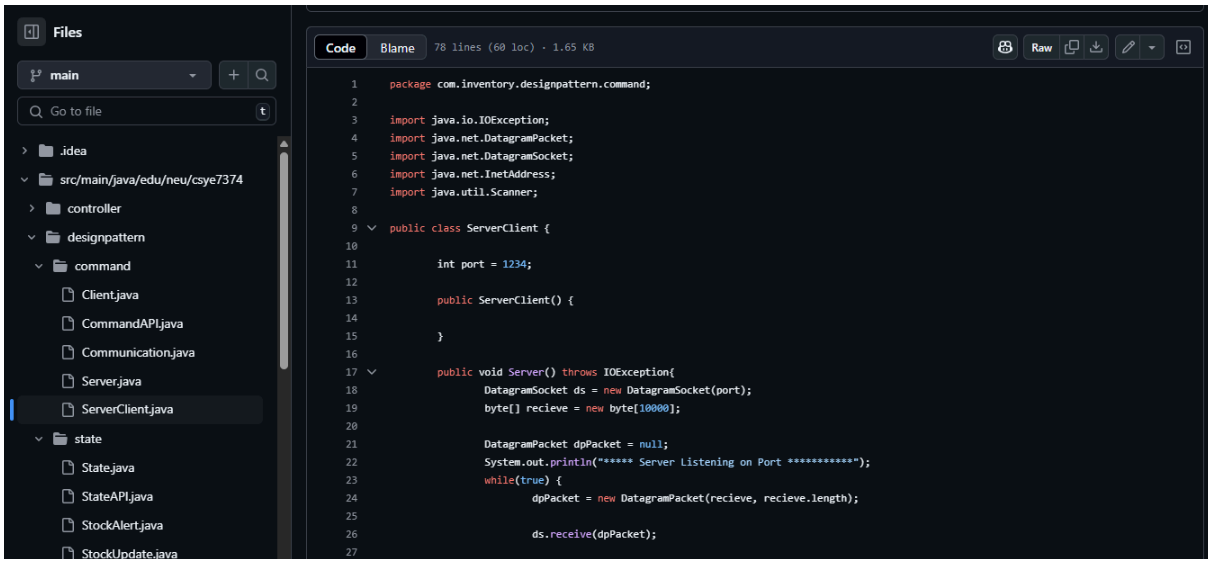


final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / command / Server.java

YashZ2002 Add files via upload

25 lines (18 loc) · 428 Bytes

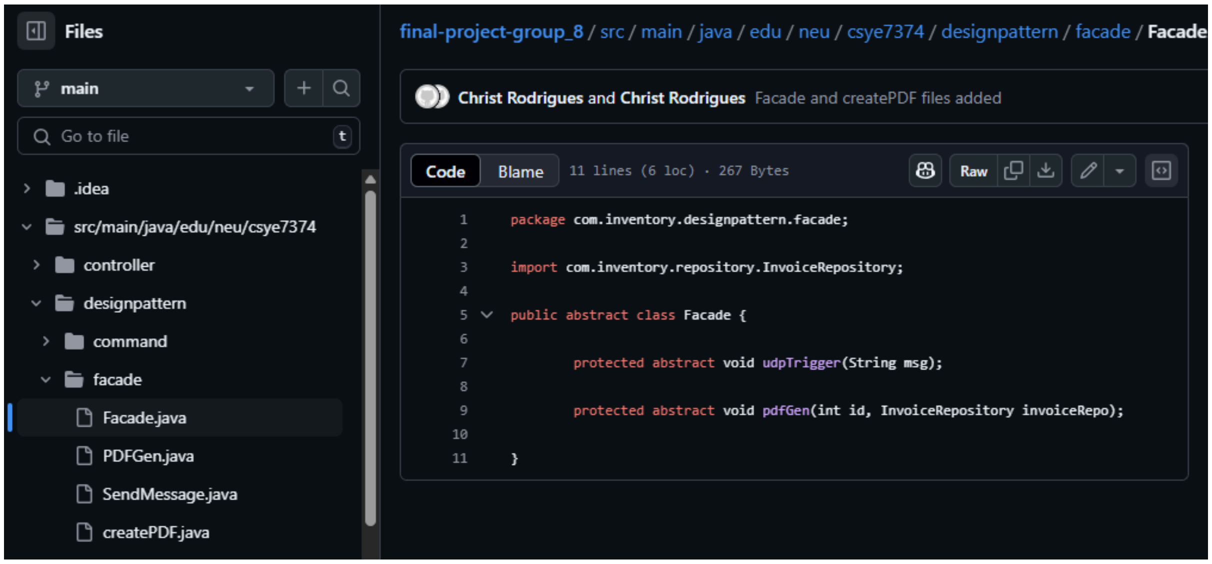
```java
package com.inventory.designpattern.command;

import java.io.IOException;

public class Server implements Runnable, CommandAPI{

	@Override
	public void run() {
		try {
			new ServerClient().Server();
		} catch (IOException e) {
			System.err.println("Error in Server");
			e.printStackTrace();
		}

	}

	@Override
	public void execute() {
		Thread t = new Thread(new Server());
		t.start();

	}
```



78 lines (60 loc) · 1.65 KB

```java
package com.inventory.designpattern.command;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class ServerClient {

        int port = 1234;

        public ServerClient() {

        }

        public void Server() throws IOException{
                DatagramSocket ds = new DatagramSocket(port);
                byte[] recieve = new byte[10000];

                DatagramPacket dpPacket = null;
                System.out.println("***** Server Listening on Port ***********");
                while(true) {
                        dpPacket = new DatagramPacket(recieve, recieve.length);

                        ds.receive(dpPacket);

```

5) Facade Design Pattern

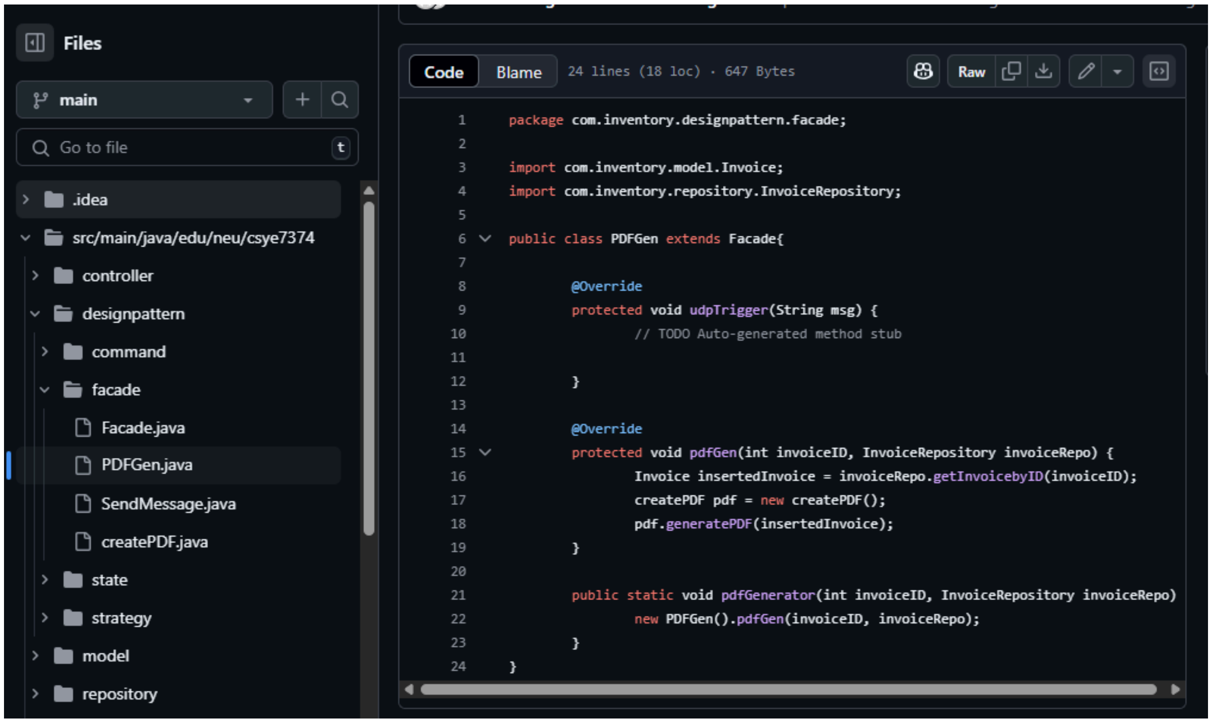


final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / facade / Facade

Christ Rodrigues and Christ Rodrigues Facade and createPDF files added

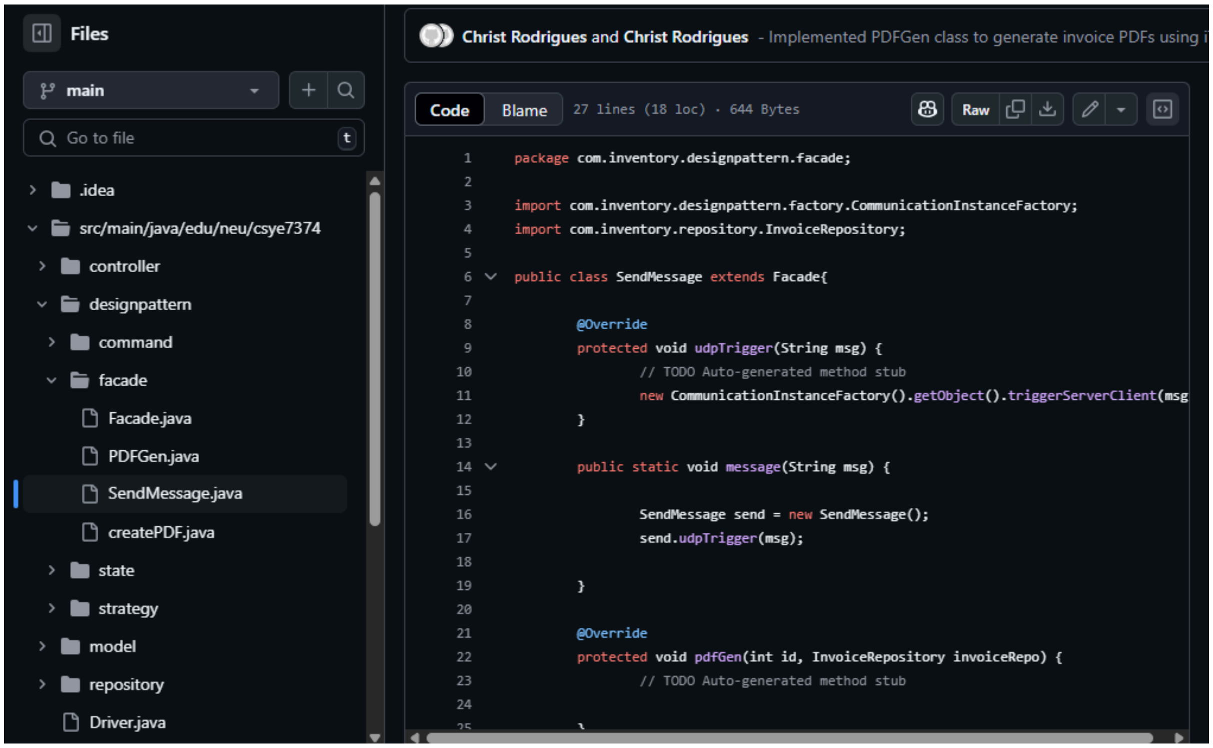
11 lines (6 loc) · 267 Bytes

```java
package com.inventory.designpattern.facade;

import com.inventory.repository.InvoiceRepository;

public abstract class Facade {

        protected abstract void udpTrigger(String msg);

        protected abstract void pdfGen(int id, InvoiceRepository invoiceRepo);

}
```



24 lines (18 loc) · 647 Bytes

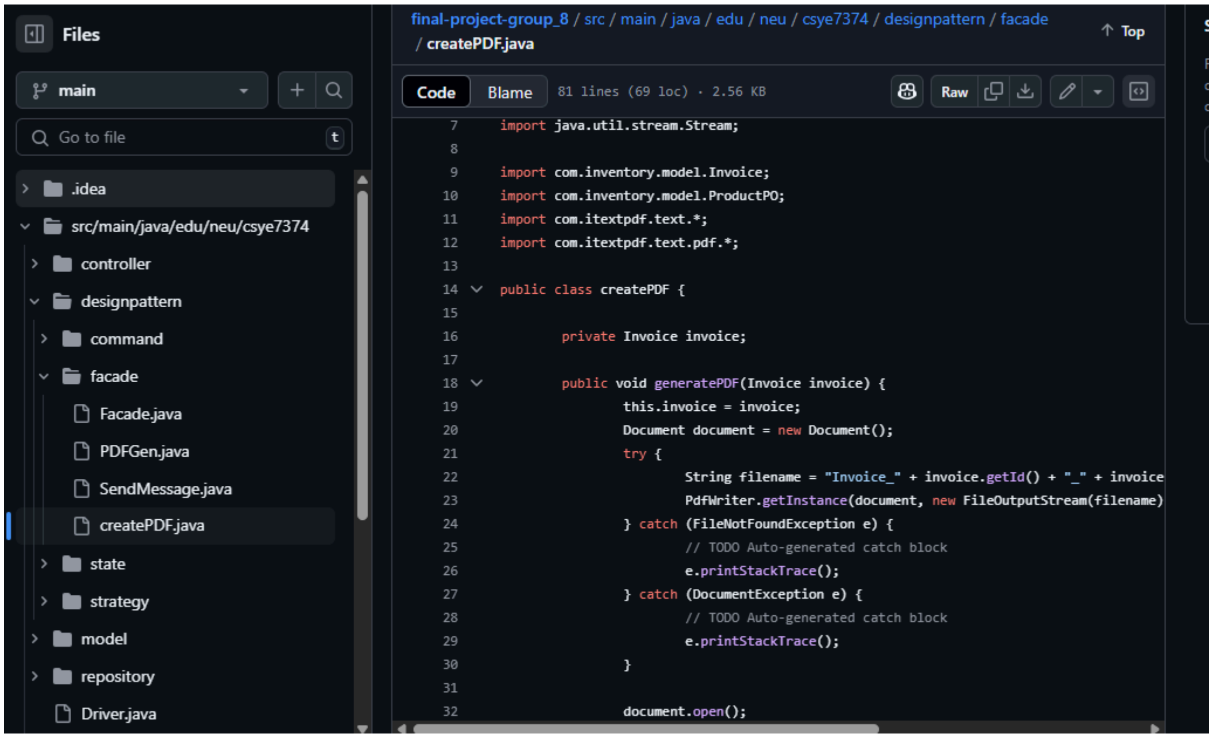
```java
package com.inventory.designpattern.facade;

import com.inventory.model.Invoice;
import com.inventory.repository.InvoiceRepository;

public class PDFGen extends Facade{

        @Override
        protected void udpTrigger(String msg) {
                // TODO Auto-generated method stub

        }

        @Override
        protected void pdfGen(int invoiceID, InvoiceRepository invoiceRepo) {
                Invoice insertedInvoice = invoiceRepo.getInvoicebyID(invoiceID);
                createPDF pdf = new createPDF();
                pdf.generatePDF(insertedInvoice);
        }

        public static void pdfGenerator(int invoiceID, InvoiceRepository invoiceRepo)
                new PDFGen().pdfGen(invoiceID, invoiceRepo);
        }
}
```



Christ Rodrigues and Christ Rodrigues - Implemented PDFGen class to generate invoice PDFs using i

27 lines (18 loc) · 644 Bytes

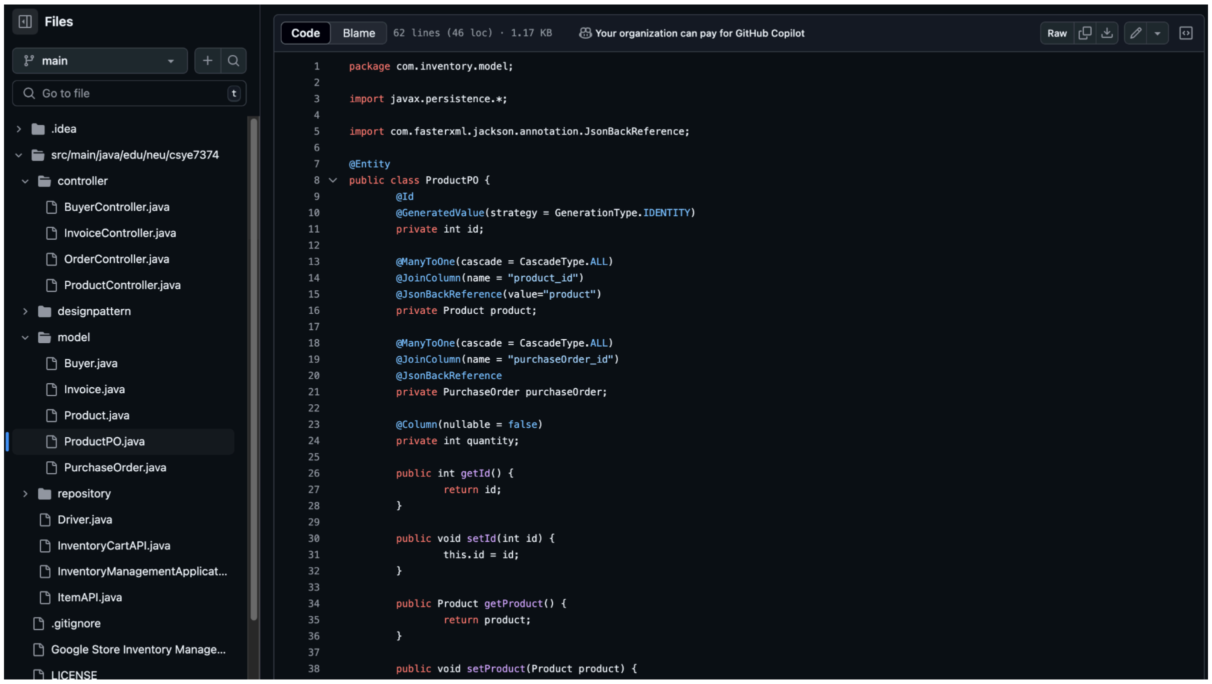
```java
package com.inventory.designpattern.facade;

import com.inventory.designpattern.factory.CommunicationInstanceFactory;
import com.inventory.repository.InvoiceRepository;

public class SendMessage extends Facade{

        @Override
        protected void udpTrigger(String msg) {
                // TODO Auto-generated method stub
                new CommunicationInstanceFactory().getObject().triggerServerClient(msg
        }

        public static void message(String msg) {

                SendMessage send = new SendMessage();
                send.udpTrigger(msg);

        }

        @Override
        protected void pdfGen(int id, InvoiceRepository invoiceRepo) {
                // TODO Auto-generated method stub

        }
```



final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / facade / createPDF.java

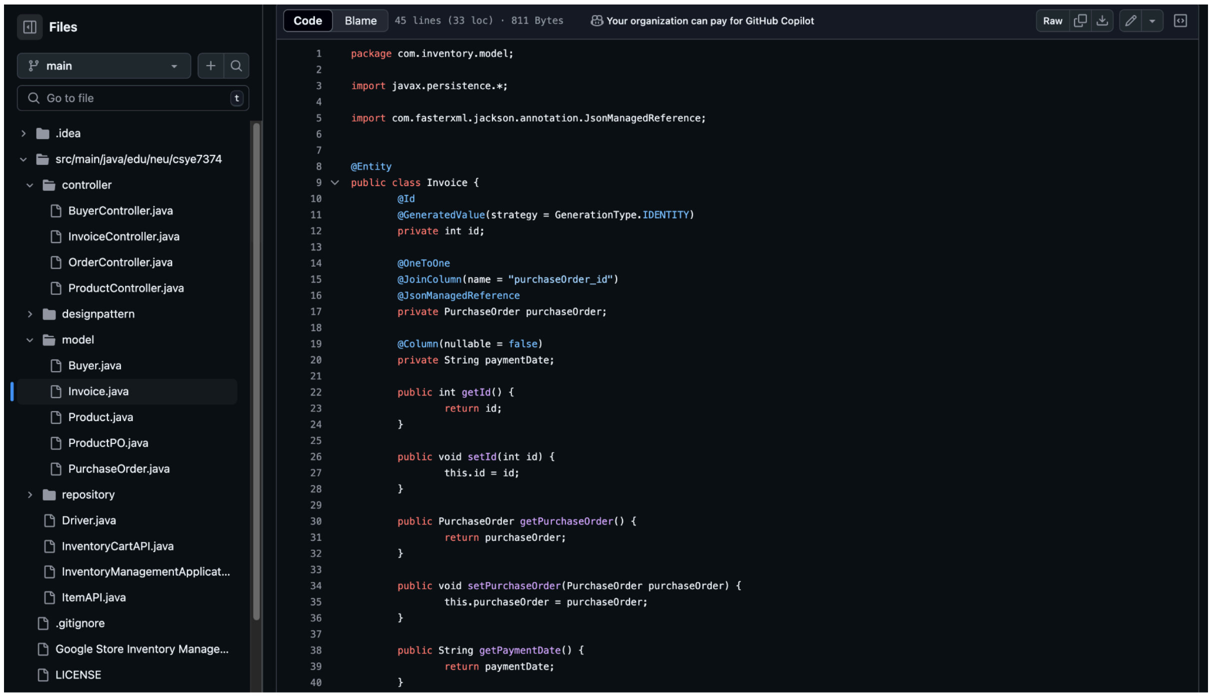
81 lines (69 loc) · 2.56 KB

```java
import java.util.stream.Stream;

import com.inventory.model.Invoice;
import com.inventory.model.ProductPO;
import com.itextpdf.text.*;
import com.itextpdf.text.pdf.*;

public class createPDF {

        private Invoice invoice;

        public void generatePDF(Invoice invoice) {
                this.invoice = invoice;
                Document document = new Document();
                try {
                        String filename = "Invoice_" + invoice.getId() + "_" + invoice
                        PdfWriter.getInstance(document, new FileOutputStream(filename)
                } catch (FileNotFoundException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                } catch (DocumentException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

                document.open();
```



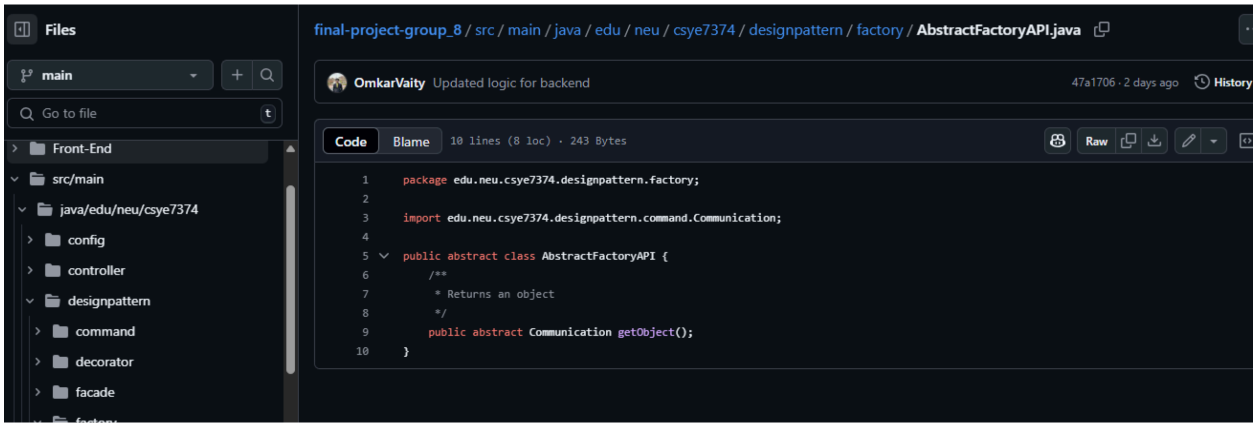
62 lines (46 loc) · 1.17 KB

```java
package com.inventory.model;

import javax.persistence.*;

import com.fasterxml.jackson.annotation.JsonBackReference;

@Entity
public class ProductPO {
        @Id
        @GeneratedValue(strategy = GenerationType.IDENTITY)
        private int id;

        @ManyToOne(cascade = CascadeType.ALL)
        @JoinColumn(name = "product_id")
        @JsonBackReference(value="product")
        private Product product;

        @ManyToOne(cascade = CascadeType.ALL)
        @JoinColumn(name = "purchaseOrder_id")
        @JsonBackReference
        private PurchaseOrder purchaseOrder;

        @Column(nullable = false)
        private int quantity;

        public int getId() {
                return id;
        }

        public void setId(int id) {
                this.id = id;
        }

        public Product getProduct() {
                return product;
        }

        public void setProduct(Product product) {
```



```java
package com.inventory.model;

import javax.persistence.*;

import com.fasterxml.jackson.annotation.JsonManagedReference;


@Entity
public class Invoice {
        @Id
        @GeneratedValue(strategy = GenerationType.IDENTITY)
        private int id;

        @OneToOne
        @JoinColumn(name = "purchaseOrder_id")
        @JsonManagedReference
        private PurchaseOrder purchaseOrder;

        @Column(nullable = false)
        private String paymentDate;

        public int getId() {
                return id;
        }

        public void setId(int id) {
                this.id = id;
        }

        public PurchaseOrder getPurchaseOrder() {
                return purchaseOrder;
        }

        public void setPurchaseOrder(PurchaseOrder purchaseOrder) {
                this.purchaseOrder = purchaseOrder;
        }

        public String getPaymentDate() {
                return paymentDate;
        }
```

6) Factory Design Pattern:

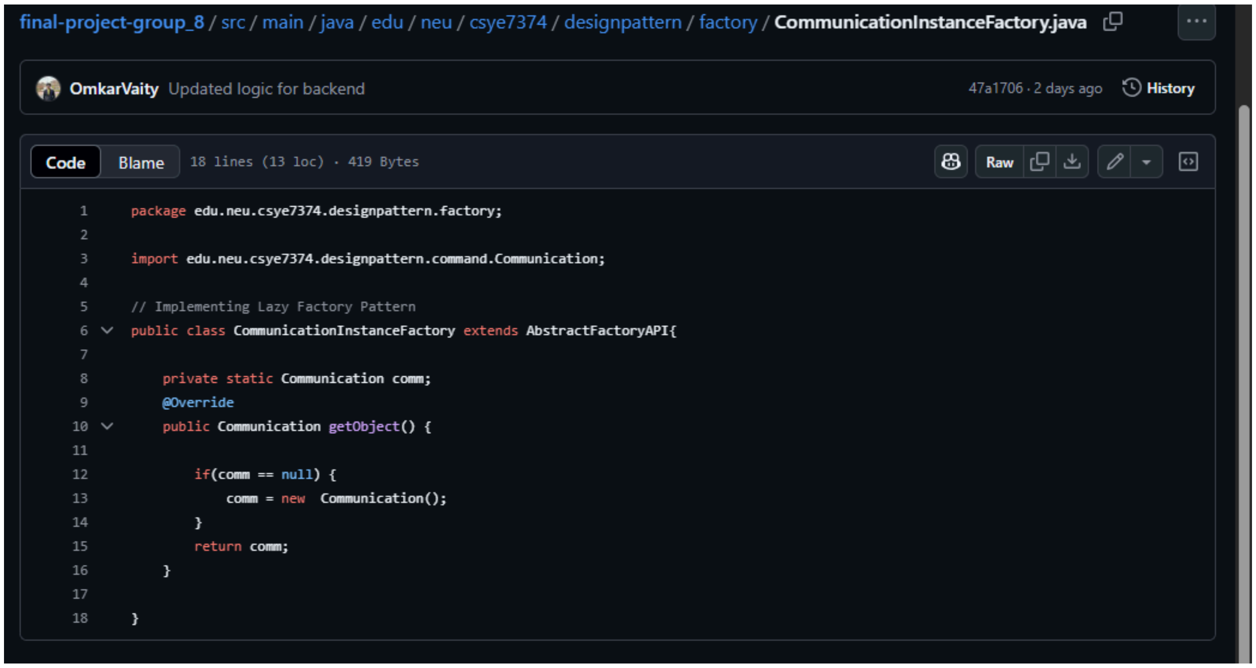


final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / factory / AbstractFactoryAPI.java

OmkarVaity Updated logic for backend 47a1706 · 2 days ago History

Code Blame 10 lines (8 loc) · 243 Bytes

```java
package edu.neu.csye7374.designpattern.factory;

import edu.neu.csye7374.designpattern.command.Communication;

public abstract class AbstractFactoryAPI {
    /**
     * Returns an object
     */
    public abstract Communication getObject();
}
```



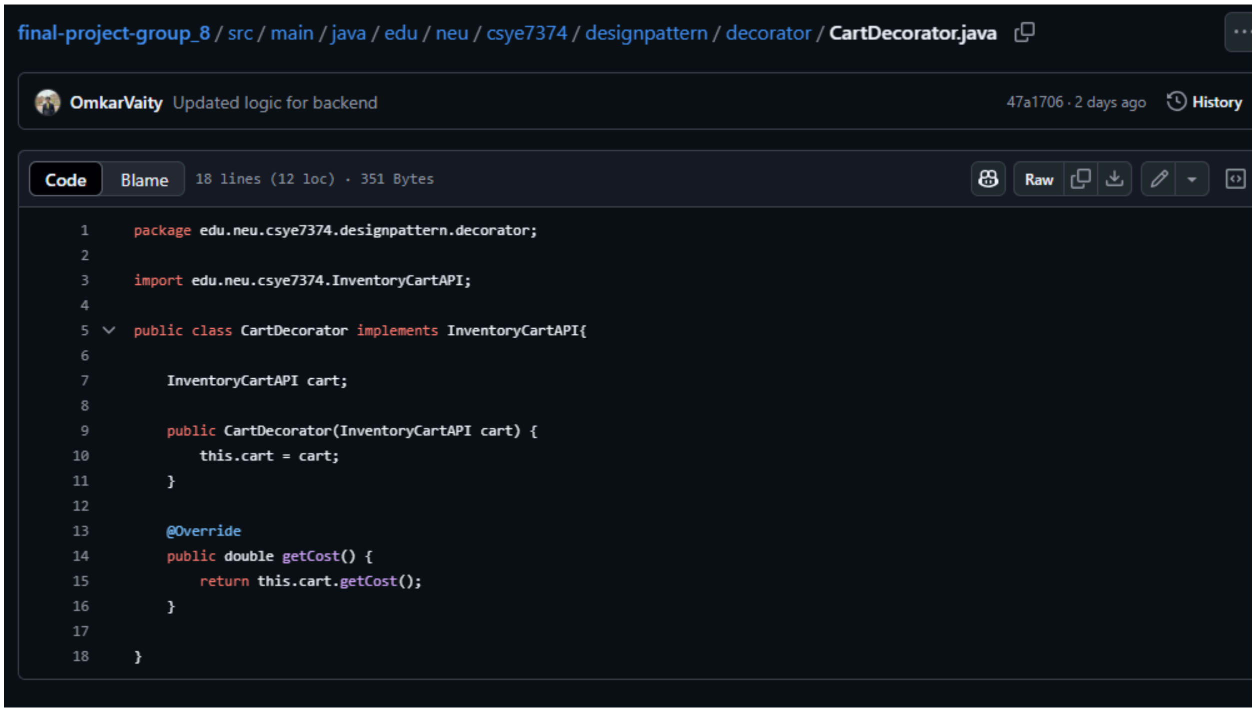
final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / factory / CommunicationInstanceFactory.java

OmkarVaity Updated logic for backend 47a1706 · 2 days ago History

Code Blame 18 lines (13 loc) · 419 Bytes

```java
package edu.neu.csye7374.designpattern.factory;

import edu.neu.csye7374.designpattern.command.Communication;

// Implementing Lazy Factory Pattern
public class CommunicationInstanceFactory extends AbstractFactoryAPI{

    private static Communication comm;
    @Override
    public Communication getObject() {

        if(comm == null) {
            comm = new  Communication();
        }
        return comm;
    }

}
```

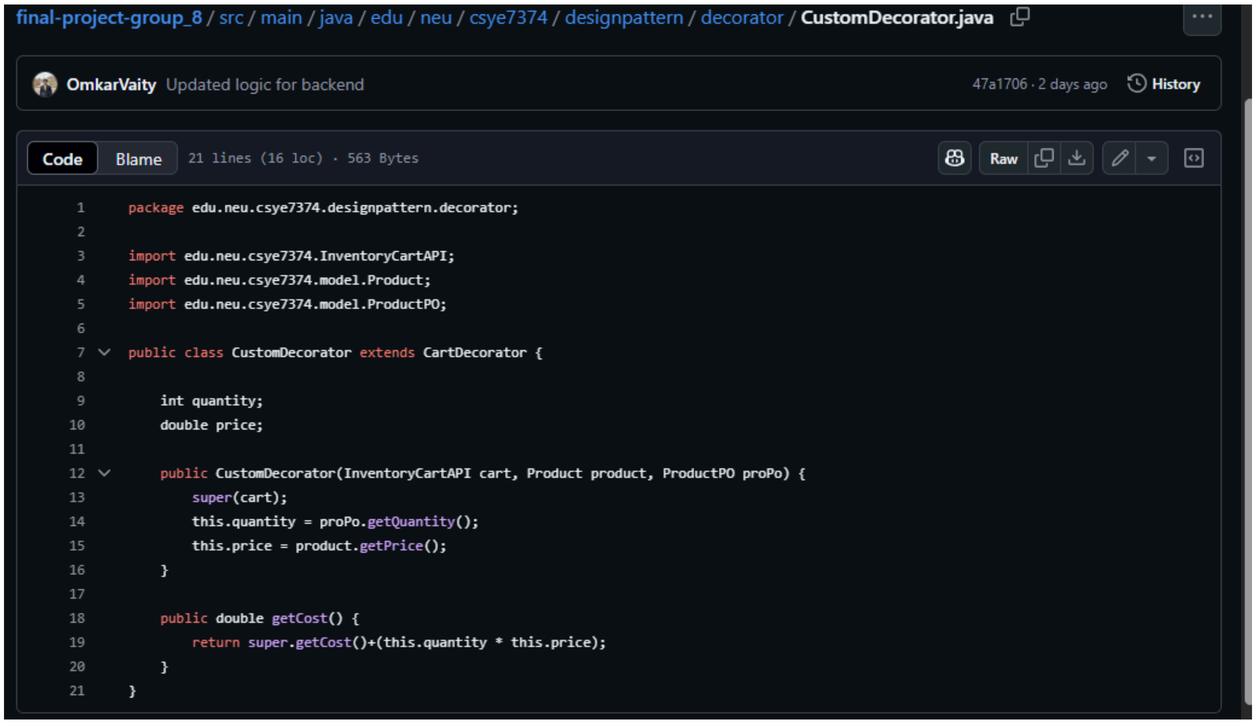
7) Decorator Design Pattern:


final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / decorator / CartDecorator.java

OmkarVaity Updated logic for backend 47a1706 · 2 days ago History

Code Blame 18 lines (12 loc) · 351 Bytes

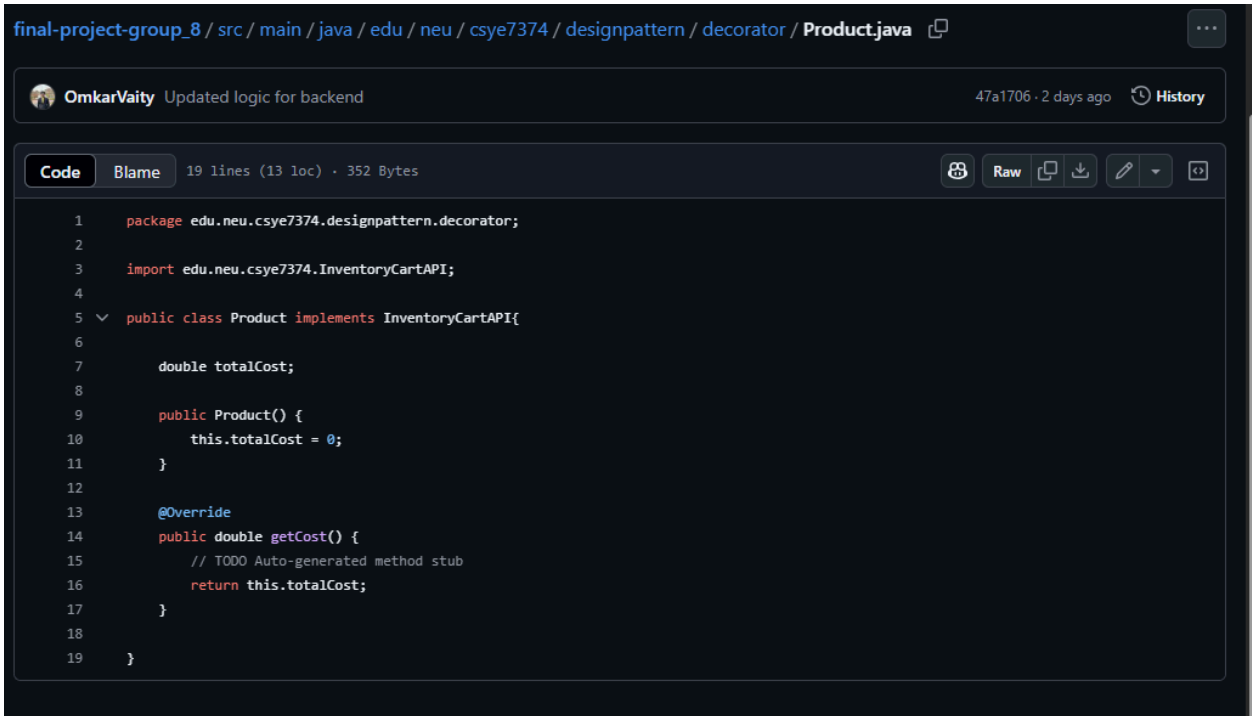
```java
package edu.neu.csye7374.designpattern.decorator;

import edu.neu.csye7374.InventoryCartAPI;

public class CartDecorator implements InventoryCartAPI{

    InventoryCartAPI cart;

    public CartDecorator(InventoryCartAPI cart) {
        this.cart = cart;
    }

    @Override
    public double getCost() {
        return this.cart.getCost();
    }

}
```


final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / decorator / CustomDecorator.java

OmkarVaity Updated logic for backend 47a1706 · 2 days ago History

Code Blame 21 lines (16 loc) · 563 Bytes

```java
package edu.neu.csye7374.designpattern.decorator;

import edu.neu.csye7374.InventoryCartAPI;
import edu.neu.csye7374.model.Product;
import edu.neu.csye7374.model.ProductPO;

public class CustomDecorator extends CartDecorator {

    int quantity;
    double price;

    public CustomDecorator(InventoryCartAPI cart, Product product, ProductPO proPo) {
        super(cart);
        this.quantity = proPo.getQuantity();
        this.price = product.getPrice();
    }

    public double getCost() {
        return super.getCost()+(this.quantity * this.price);
    }
}
```



final-project-group_8 / src / main / java / edu / neu / csye7374 / designpattern / decorator / Product.java

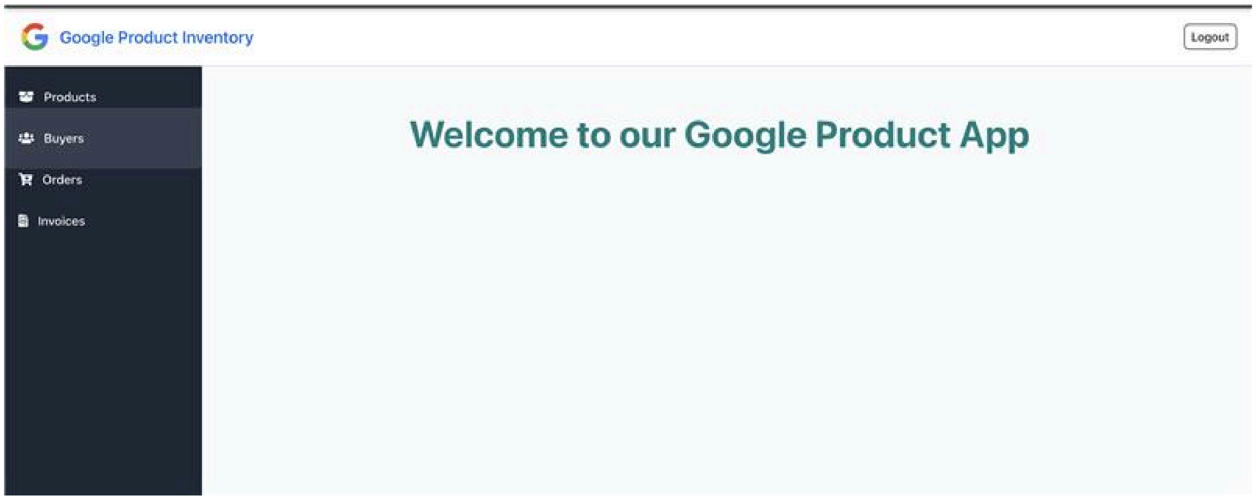
OmkarVaity Updated logic for backend 47a1706 · 2 days ago History

Code Blame 19 lines (13 loc) · 352 Bytes Raw

```java
package edu.neu.csye7374.designpattern.decorator;

import edu.neu.csye7374.InventoryCartAPI;

public class Product implements InventoryCartAPI{

    double totalCost;

    public Product() {
        this.totalCost = 0;
    }

    @Override
    public double getCost() {
        // TODO Auto-generated method stub
        return this.totalCost;
    }

}
```

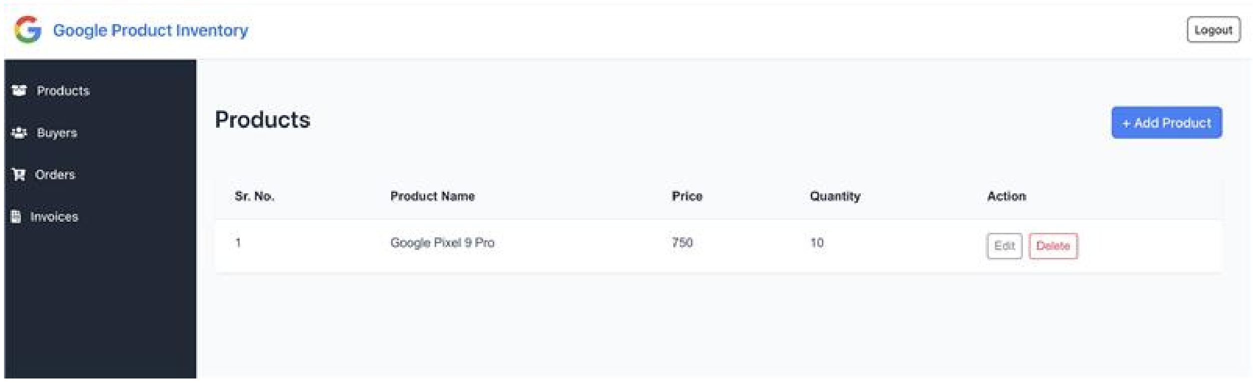
Screenshot of UI (feature-based) at every stage of the application:

1. **Welcome Page:**



Welcome to our Google Product App

2. **Product Page:**



Google Product Inventory

Logout

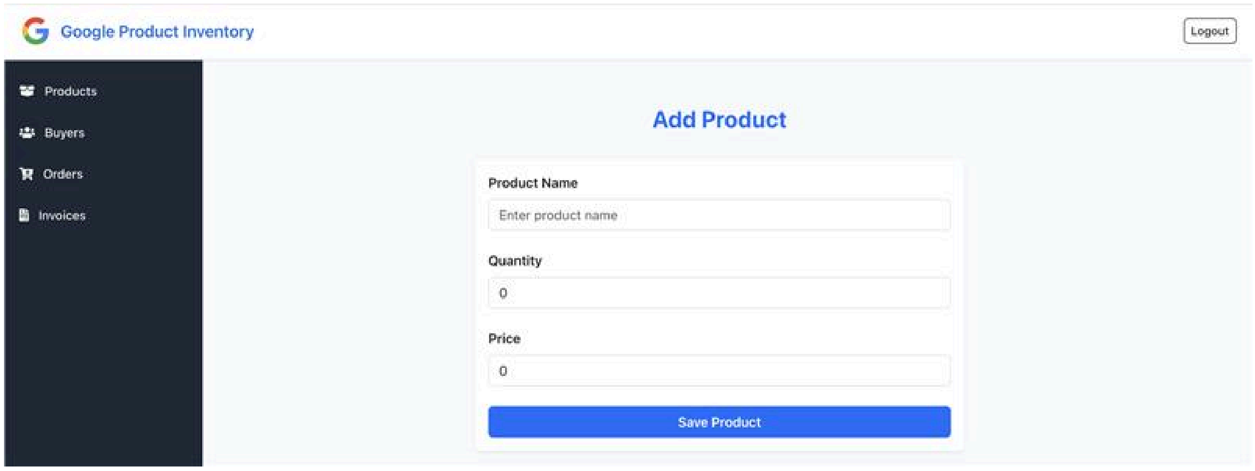
Products
Buyers
Orders
Invoices

Products

+ Add Product

Sr. No.	Product Name	Price	Quantity	Action
1	Google Pixel 9 Pro	750	10	Edit Delete

3. **Add Product:**



Google Product Inventory

Logout

Products
Buyers
Orders
Invoices

Add Product

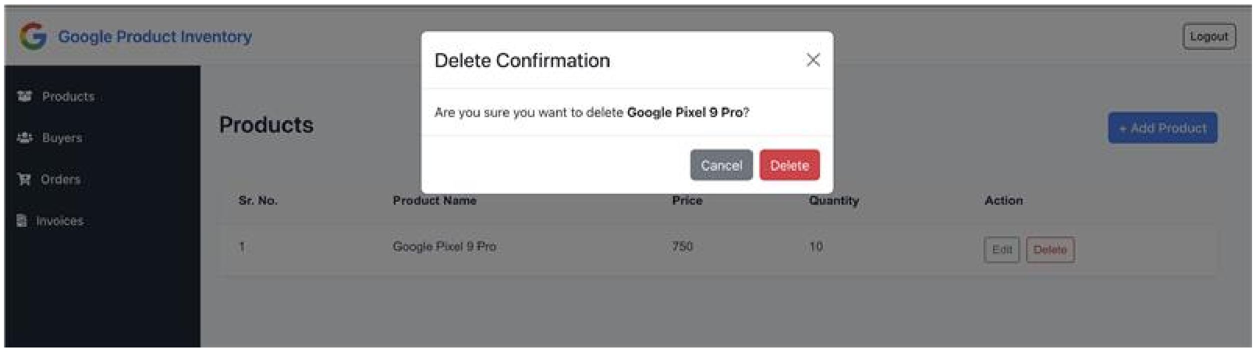
Product Name
Enter product name

Quantity
0

Price
0

Save Product

4. Delete Product:



Delete Confirmation

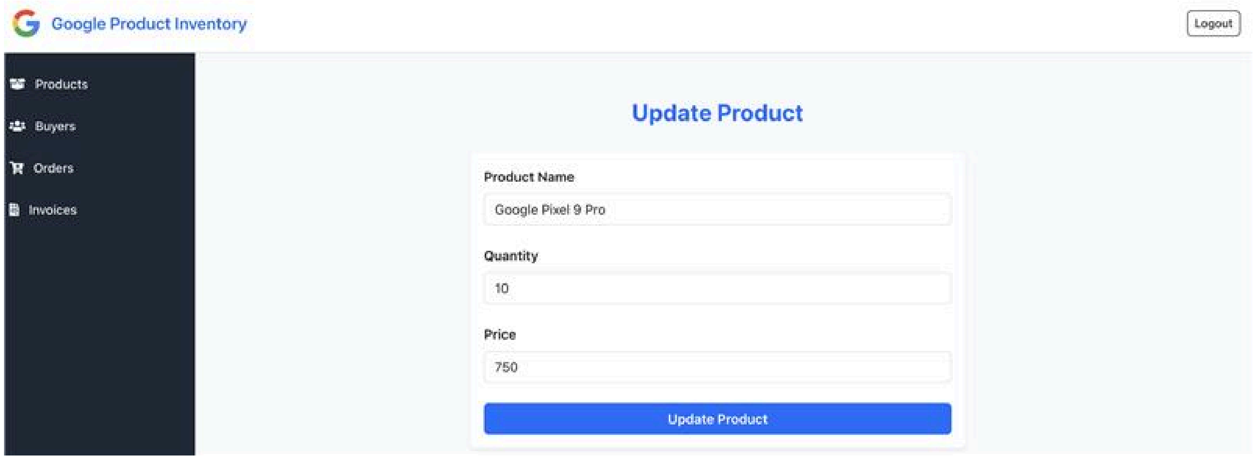
Are you sure you want to delete **Google Pixel 9 Pro**?

Cancel Delete

Products

Sr. No.	Product Name	Price	Quantity	Action
1	Google Pixel 9 Pro	750	10	Edit Delete

5. Edit Product:



Update Product

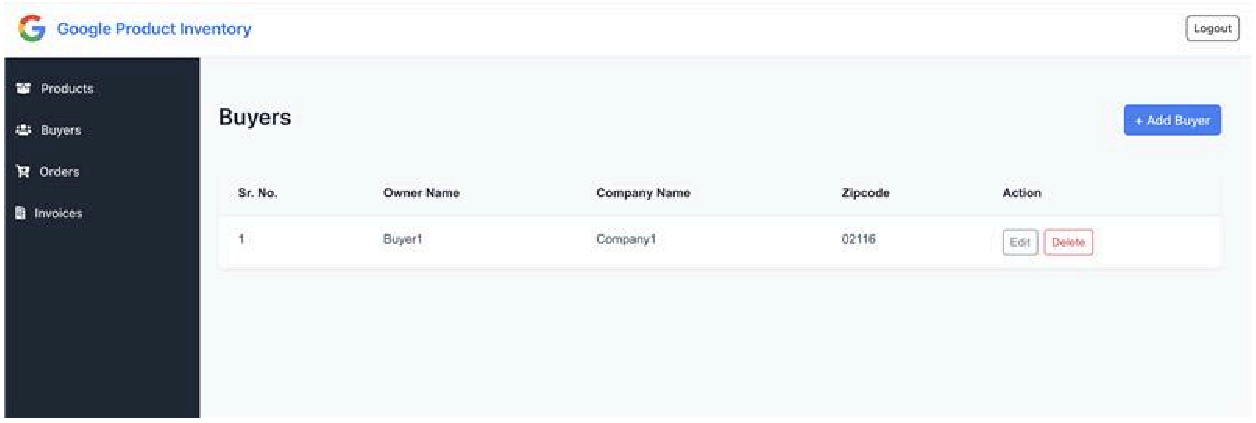
Product Name: Google Pixel 9 Pro

Quantity: 10

Price: 750

Update Product

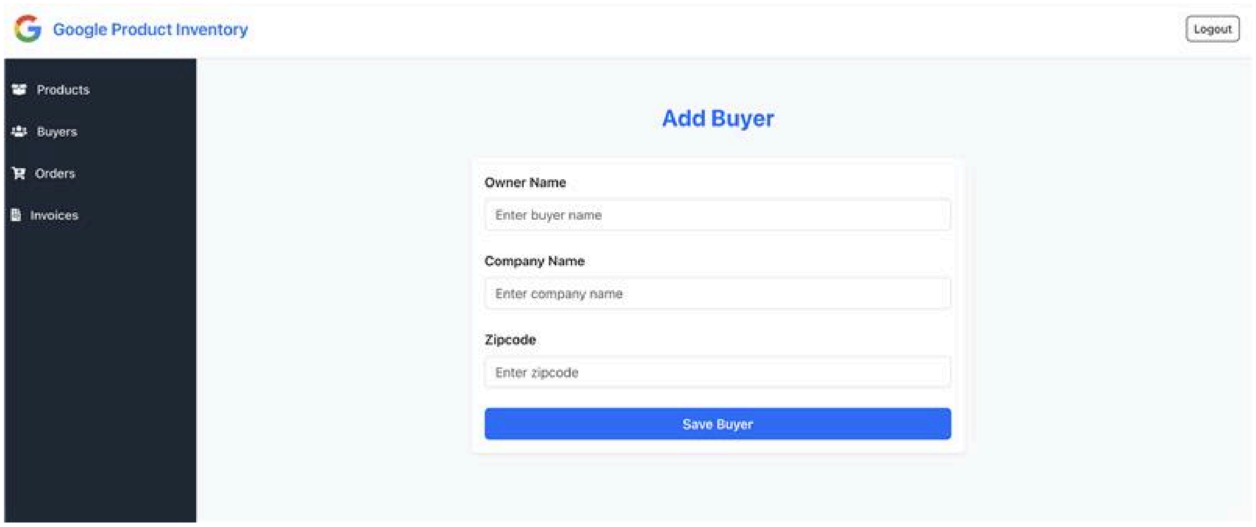
6. Buyer Page:



Buyers

Sr. No.	Owner Name	Company Name	Zipcode	Action
1	Buyer1	Company1	02116	Edit Delete

7. **Add Buyer:**



Add Buyer

Owner Name

Enter buyer name

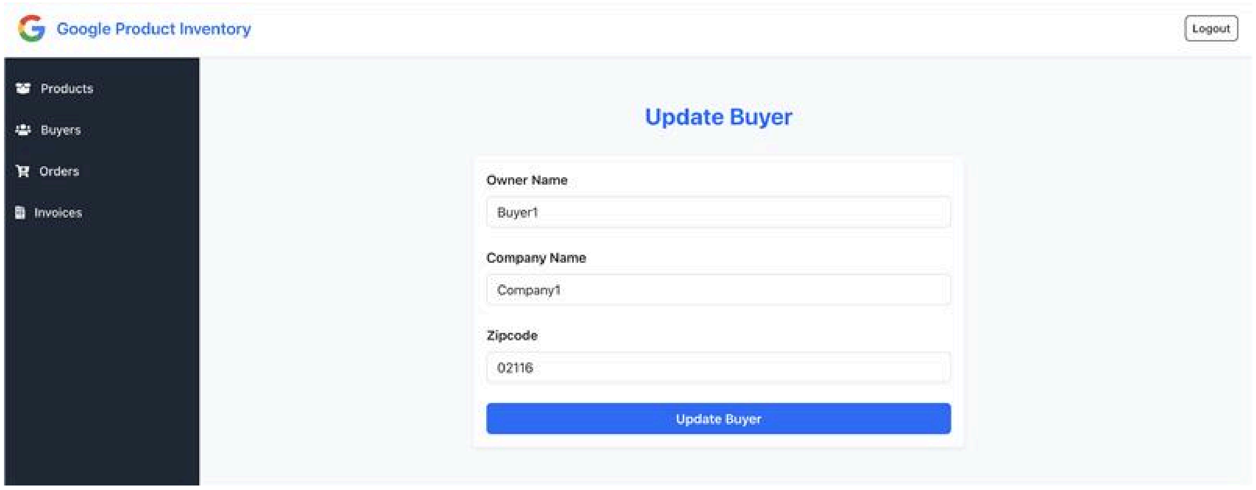
Company Name

Enter company name

Zipcode

Enter zipcode

8. **Update Buyer:**



Update Buyer

Owner Name

Buyer1

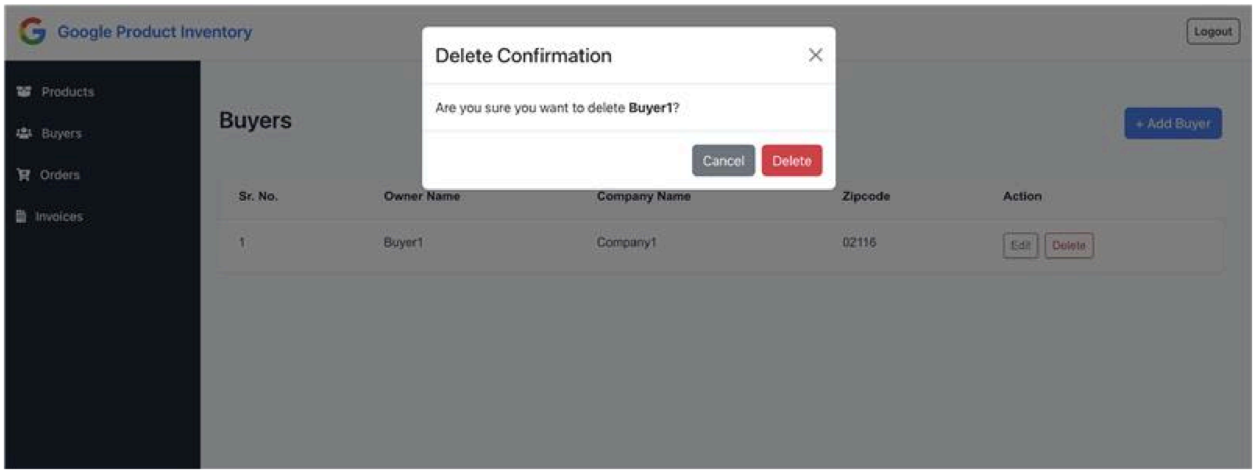
Company Name

Company1

Zipcode

02116

9. **Delete Buyer:**



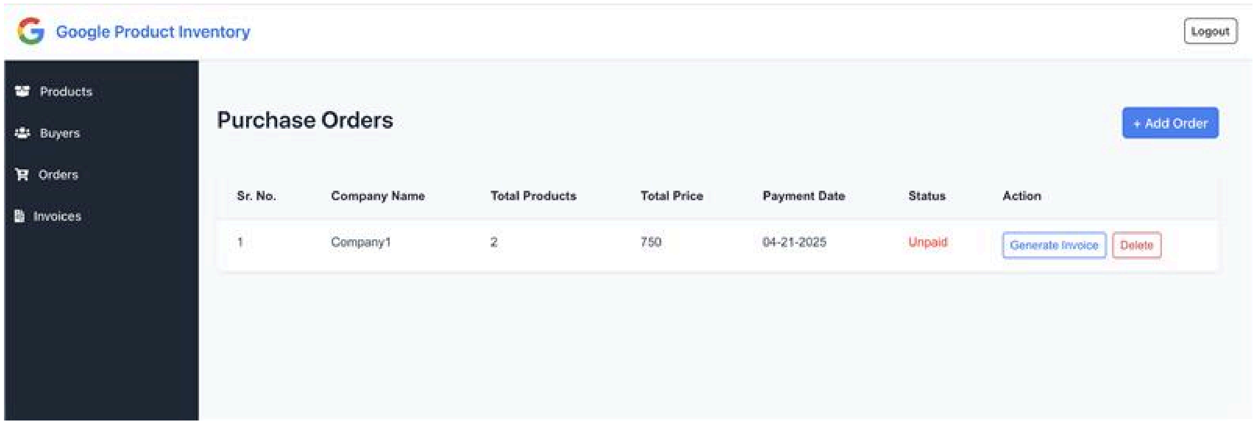
Delete Confirmation

Are you sure you want to delete **Buyer1**?

Buyers

Sr. No.	Owner Name	Company Name	Zipcode	Action
1	Buyer1	Company1	02116	Edit Delete

10. Order Page:



Google Product Inventory

Logout

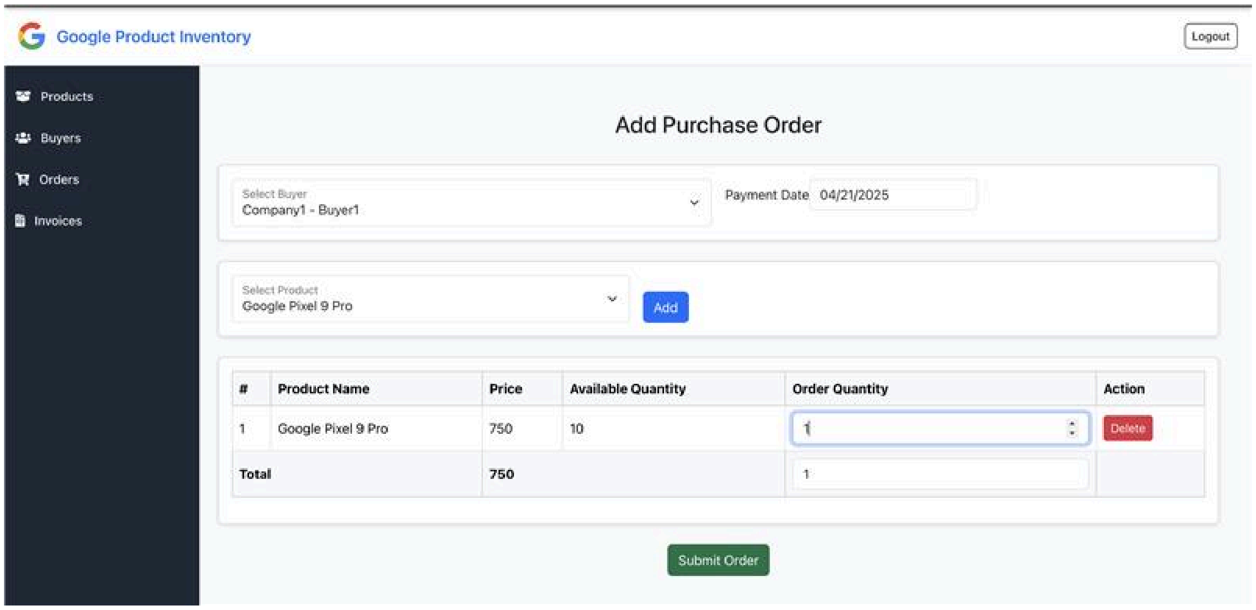
Products
Buyers
Orders
Invoices

Purchase Orders

+ Add Order

Sr. No.	Company Name	Total Products	Total Price	Payment Date	Status	Action
1	Company1	2	750	04-21-2025	Unpaid	Generate Invoice Delete

11. Add Order:



Google Product Inventory

Logout

Products
Buyers
Orders
Invoices

Add Purchase Order

Select Buyer: Company1 - Buyer1

Payment Date: 04/21/2025

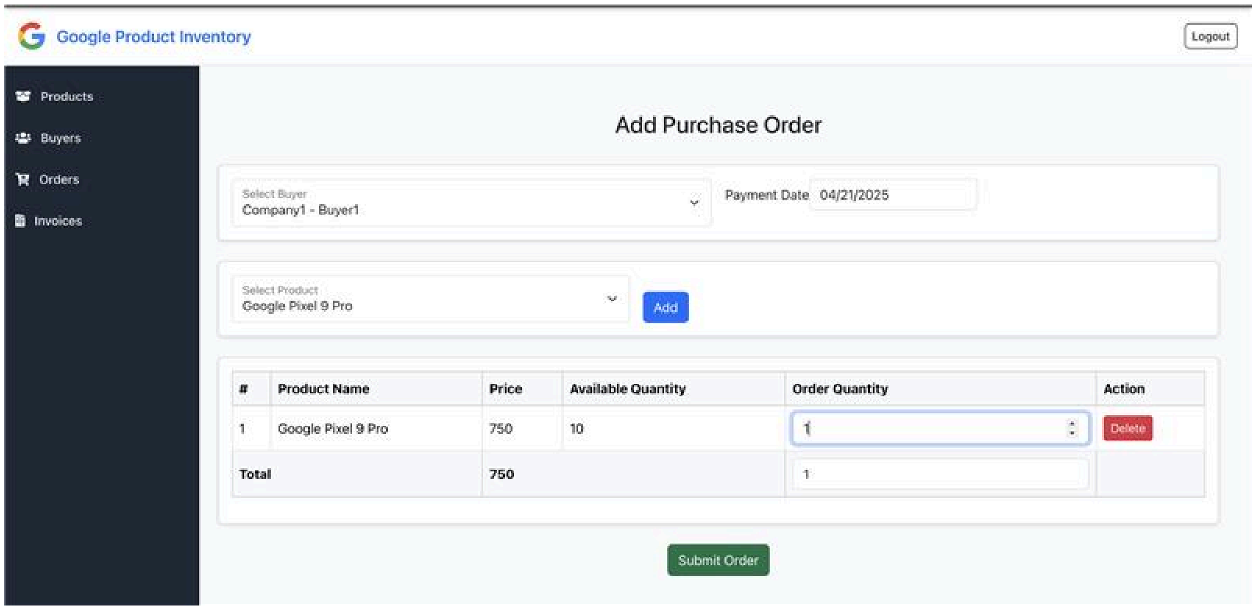
Select Product: Google Pixel 9 Pro

Add

#	Product Name	Price	Available Quantity	Order Quantity	Action
1	Google Pixel 9 Pro	750	10	1	Delete
Total		750		1	

Submit Order

12. Delete Order:



Add Purchase Order

Select Buyer: Company1 - Buyer1

Payment Date: 04/21/2025

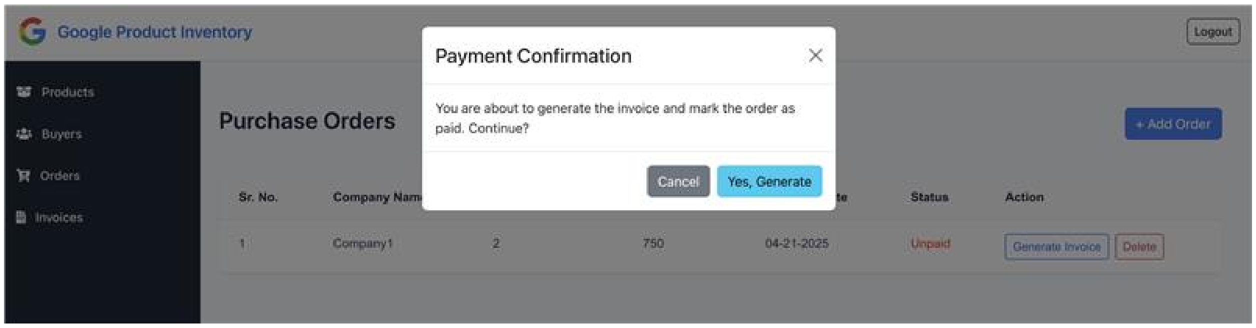
Select Product: Google Pixel 9 Pro

Add

#	Product Name	Price	Available Quantity	Order Quantity	Action
1	Google Pixel 9 Pro	750	10	1	Delete
Total		750		1	

Submit Order

13. Generate Invoice:



Purchase Orders

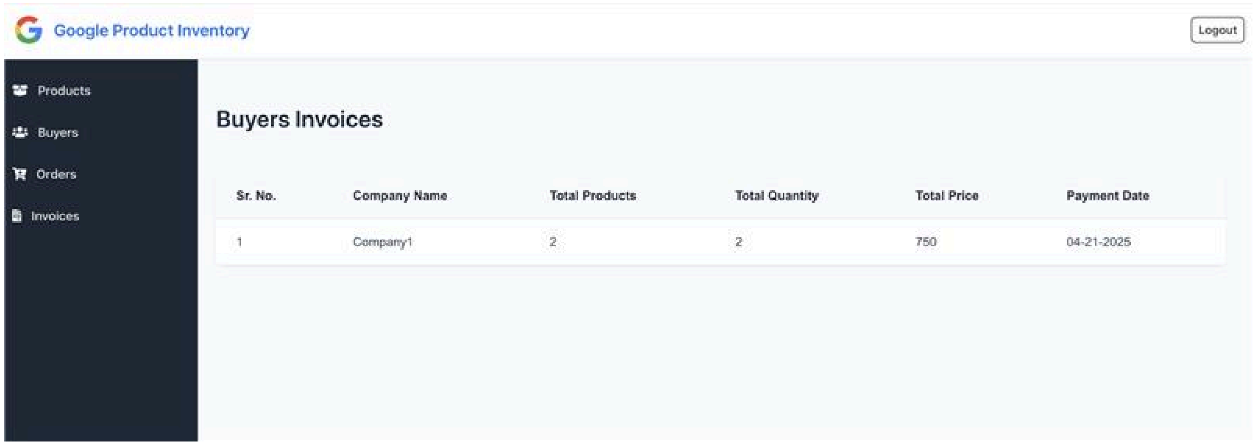
Payment Confirmation

You are about to generate the invoice and mark the order as paid. Continue?

Cancel Yes, Generate

Sr. No.	Company Nam			te	Status	Action
1	Company1	2	750	04-21-2025	Unpaid	Generate Invoice Delete

14. Invoice Page:



Buyers Invoices

Sr. No.	Company Name	Total Products	Total Quantity	Total Price	Payment Date
1	Company1	2	2	750	04-21-2025

Contributions:

- **Christ Rodrigues:** Facade Design Pattern, Frontend.
- **Yash Zaveri:** Command Design Pattern, Frontend.
- **Viral Suchak:** Strategy Design Pattern, Factory Design Pattern.
- **Kevin Rodrigues:** Observer Design Pattern, Frontend.
- **Omkar Vaity:** State Design Pattern, Singleton Design Pattern.