1. **Name:** Yuchen Zhang
2. **Title: E-Commerce Service**
3. **Project Summary: An e-commerce service website that can allow customer to search, purchase and return products. What's more, it allows administrator to add products and refund customer. Also, Storage administrator can check product storage and package orders. Deliveryman can deliver orders**
4. **Project Requirements:**

UC-1 purchase products			
Requirement ID	Actor	Actor action	System response
UR-01	customer	Search product	Display search result
UR-02	customer	Add product to cart	Display the result of addition(success) and the number of products in cart
UR-03	customer	Delete product from cart	Display the result of deletion(success) and the number of products remained in cart
UC-2 order processing			
Requirement ID	Actor	Actor action	System response
UR-04	customer	check out	display order detail and total cost
UR-05	admin	set discount event	display discount detail when check out
UR-06	storage admin	check product storage	if products are out of stock, remind customer and reject checking out
UR-07	customer	choose payment method	display payment method
UR-08	customer	input account info	display payment result
UR-09	storage admin	package orders	update order statement
UR-10	deliveryman	deliver orders	update order statement
UC-3 after-sale service			
Requirement ID	Actor	Actor action	System response
UR-11	customer	return products	display result
UR-12	admin	accept returned products	display result
UR-13	admin	refund customer	display result

Requirement ID	Actor	Actor action	System response
UR-14	storage admin	update storage for the returned products	N/A
UC-4 log in and log out requirements			
Requirement ID	Actor	Actor action	System response
UR-15	customer	log in	display result
UR-16	admin	log in	display result
UR-17	storage admin	log in	display result
UR-18	customer	Log out	display result
UR-19	admin	Log out	display result
UR-20	Storage admin	Log out	display result

5. **UI Mockups:**

I want to use cmd window to act as a UI, firstly. If I can, I would use Javascript to do it instead of.

```
●  ●  ●                    📁 Homework-3 — -bash — 80×24                      ▤
please choose role: 1 customer 2 admin 3 storage admin
1
please choose action:
1: search product
2: view cart
3: view orders
1
please input your search key word in the format:
productname: xxx, brand: xxx
brand: Sunny
1. milk
2. bread
3. yogurt
please choose product number
1
please choose your action:
1. display product detail
2. add to cart
2
please choose your action:
2
please choose your action:
1. view cart
2. checkout
```

It is an interaction example above. It doesn't display all the actions. It is the basic goal. The optional for UI is to write a website.

6. **Class Diagram:**

The class diagram is drew by StarUML, I didn't find how to represent static attribute and static method. So I write the "static " key word directly. And the get set methods are not displayed in it.

ProductSearchFilter
- -brand: String
- -category: String

Address
- -addressLine: String
- -city: String
- -state: String
- +toString(): String

Cart
- -itemNum: int
- -items: List<Product>
- +getTotalPrice(): double

Customer
- -userName: String
- -address: Address
- -phone: String
- -cart: Cart
- -customerID: String
- +searchProduct(searchFilter: ProductSearchFilter): List<Product>
- +addProduct2Cart(products: List<Product>): boolean
- +delProductFromCart(products: List<Product>): boolean
- +clickCheckOut(): boolean
- +payment(pay: PaymentInfo): boolean
- +returnProducts(order: Order): boolean
- +login(keyword: String): boolean
- +logout(): boolean

PaymentInfo
- -payMethod: String
- -accountInfo: String
- +toString(): String

Product
- -productName: String
- -category: String
- -price: double
- -brand: String

Admin
- -adminId: String
- -static discountStrategy: DiscountStrategy
- +addProduct(products: List<Product>): boolean
- +refundCustomer(customer: Customer, order: Order): boolean
- +login(keyword: String): boolean
- +logout(): boolean

Order
- -productList: List<Product>
- -customerID: String
- -state: String
- -orderId: String
- +getTotalCost(discount: DiscountStrategy)

StorageAdmin
- -storageAdminId: String
- +checkStorage(product: Product): boolean
- +packageOrders(order: Order): void
- +updateProductStorage(products: List<Product>): boolean
- +login(keyword: String): boolean
- +logout(): boolean

DiscountStrategy
- -strategyType: Enum
- -strategyParam: List<String>
- +computeDiscount(products: List<Product>): double

StrategyFactory
- -strategyMap: Map<String, DiscountStrategy>
- +registerStrategy(strategy: DiscountStrategy): void
- +getStrategy(strategyType: Enum): DiscountStrategy

DollarOffStrategy
- +computeDiscount(products: List<Product>): double

PercentOffStrategy
- +computeDiscount(products: List<Product>): double