

**Giulia Trespiolli**  
**Vincenzo Del Grosso**  
**Francesco Borderi**  
**Aurora Rosciano**



# **Smart Warehouse**

**Progetto J24 Programmazione ad Oggetti e Ingegneria del Software**  
**JAVA**

**Università degli studi di Pavia**  
**A.A. 2023/2024**



# Introduzione





# Introduzione

**Smart Warehouse è un software di gestione adatto alle esigenze di warehousing aziendale.**

**Servizi:**

- **Monitoraggio dell'inventario e riordino degli articoli in esaurimento**
- **Gestione degli ordini dei clienti**
- **Procedure di rimborso semplificate**
- **Acquisti diretti tramite lo shop**



# Unified Process

## Abbiamo scelto UP perché:

- **Permette un approccio iterativo e incrementale**
- **Permette di gestire in modo flessibile cambiamenti nei requisiti**
- **Si basa su casi d'uso e fattori di rischio**
- **Fa ampio uso di UML (Unified Modeling Language)**

# Fasi di UP



**Ideazione**

**1**

**Elaborazione**

**2**

**Costruzione**

**3**

**Transizione**

**4**



# Flussi di Progetto





# Requisiti

- **Più tipi di utilizzatori del software: Operatori e Clienti**
- **Visualizzazione dell'inventario**
- **Aggiunta di articoli all'inventario**
- **Aggiornamenti della quantità su acquisti e rifornimenti**
- **Creazione di ordini**
- **Operazioni di reso e rimborsi**
- **Operazioni di pagamento**
- **Strategie di acquisto**
- **Creazione dei pacchi in uscita dal magazzino, semplici veloci e tracciabili**

**I requisiti sono definiti con dettaglio nella documentazione del progetto.**





# Casi d'uso (divisione interna)

**Shop**  
frazz02



**Supply**  
giul02



**Return Service**  
cienzman



**Picking**  
rosci01





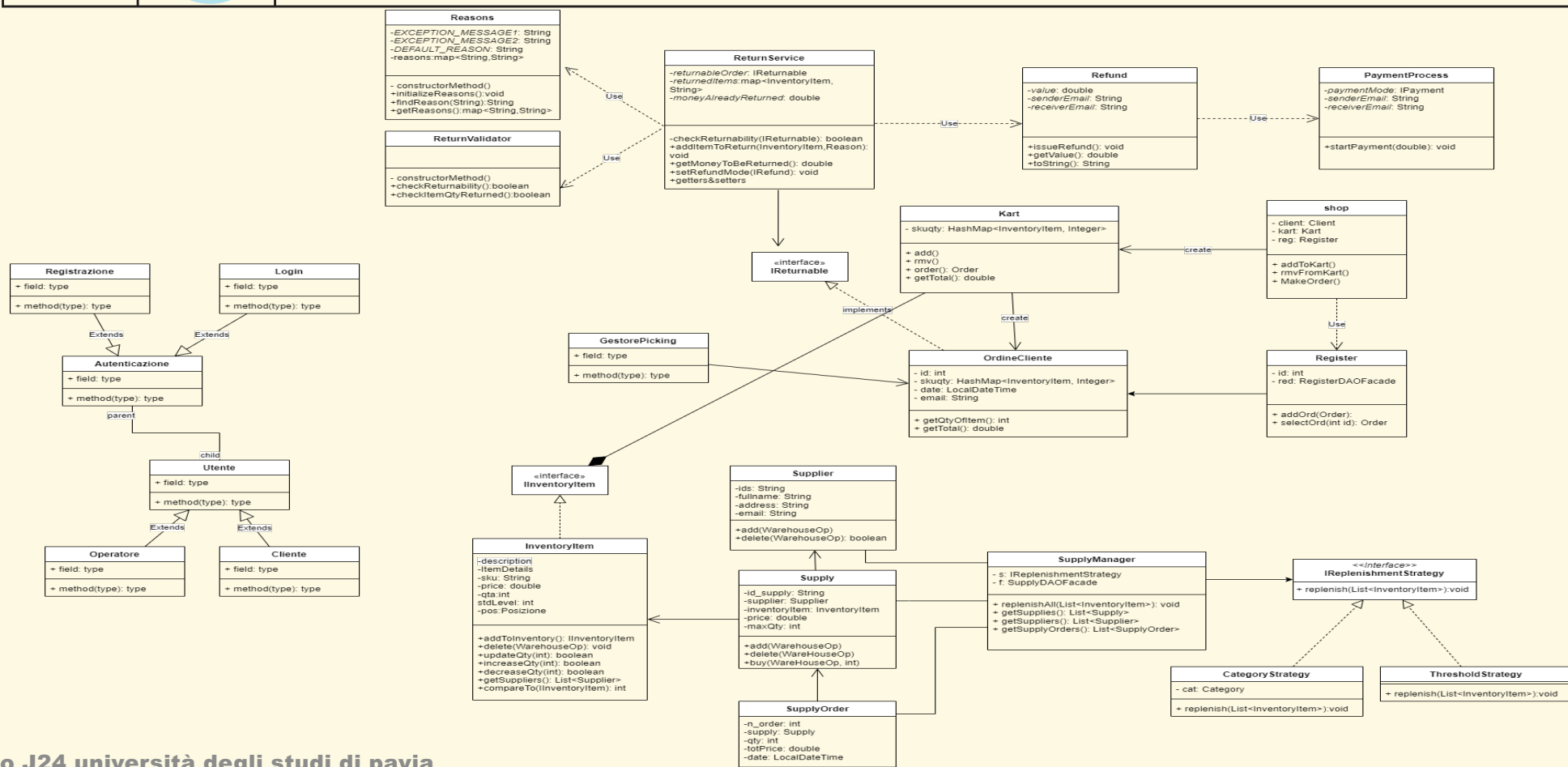


## Casi d'uso

- **UC1: Login**
- **UC2: Acquisti Shop**
- **UC3: Creazione Ordini**
- **UC4: Operazione di Reso**
- **UC5: Operazioni di Pagamento**
- **UC6: Operazione di Picking**



# Diagramma Classi2.0



# Progettazione





# Architettura MVC

**Abbiamo scelto MVC (Model View Controller) perchè la separazione tra i dati, la loro rappresentazione e la gestione degli eventi minimizza l'accoppiamento facilitando la gestione e la riusabilità del codice.**

**Risultato: codice organizzato, riutilizzabile, testabile e facilmente modificabile.**

An abstract graphic on a light yellow background. It features a red line in the top right corner, a green line that starts on the left, curves down, and then runs horizontally across the bottom, and a blue line that runs horizontally above the green line. There is a large orange circle on the left and two small black dots: one on the green line and one on the blue line.

**Funzionalità**



# Inventario

Inventory

Sku:

Filter by Position

Filter by UnderLevel

X

New Item

order

Sku	Description	Price	Qty	StdLevel	Line	Pod	Bin	Fragility	Dimension	Category
EXMCL	gonna rossa	40.9	40	30	B	B	F	2	2	Clothing
I22TQ	iphone nero ...	23.0	0	20	G	A	B	9	2	Electronics
C2V02	jeans	32.0	300	30	A	A	A	1	6	Clothing
53CQE	maglia viola	12.0	55	20	A	B	C	1	2	Clothing
SLKPG	mela	0.5	21	100	C	A	A	1	1	Groceries
LKJZC	monopoli	20.9	30	40	G	F	F	4	8	Toys
Q7SP2	peluche	10.99	55	20	D	A	B	1	3	Toys

all Suppliers

all Supplies

all Supply Orders

new Supplier

new Supply

Category Strategy

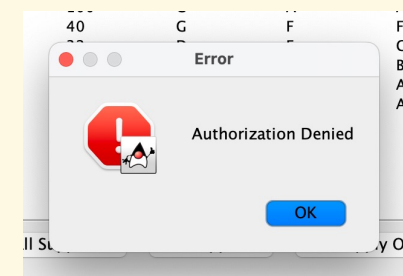
replenish

- **Elenco di tutti gli articoli**
- **Informazioni dettagliate per ciascun articolo**
- **Ricerca rapida**
- **Tracciamento delle posizioni**
- **Visualizzazione delle quantità attualmente disponibili**
- **Ordinamento**



# Rifornimento

- **Controllo dell'autorizzazione (SupplyOperator)**
- **Monitoraggio di fornitori e forniture**
- **Possibilità di visualizzare i fornitori associati a ciascun articolo in ordine di prezzo**



Sku	Description	Price	Qty	StdLevel	Line	Pod	Bin	Fragility	Dimension	Category
OKIYK	acqua	0.5	0	200	A	F	B	1	6	Groceries
XZAZV	asus 12.2	699.0	0	20	E	B	B	9	5	Electronics
EXMCL	gonna rossa	40.9	40	30	B	B	F	2	2	Clothing
DUEOV	lpad 10	399.0	3	60	F	G	G	9	2	Electronics
I22TQ	iphone nero 11	23.0	0	20	G	A	B	9	2	Electronics
C2V02	jeans	32.0							6	Clothing
53CQE	maglia viola	12.0							2	Clothing
SLKPG	mela	0.5							1	Groceries
LKJZC	monopoli	20.9							8	Toys
OEHKZ	pantalone blu	90.9							6	Clothing
Q7SP2	peluche	10.99							3	Toys
JS946	pera	0.8							1	Groceries
SUK2N	risico	30.0							9	Toys

OKIYK

+

-

Qty:

Delete Item

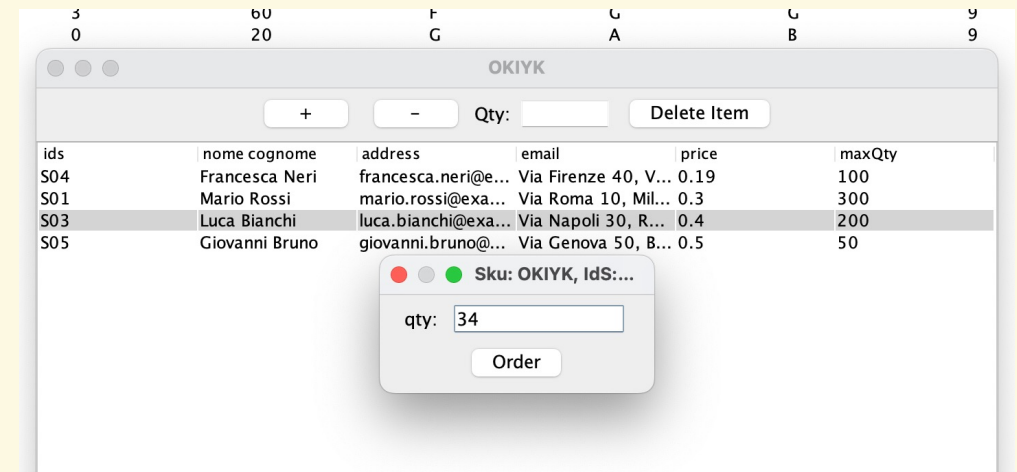
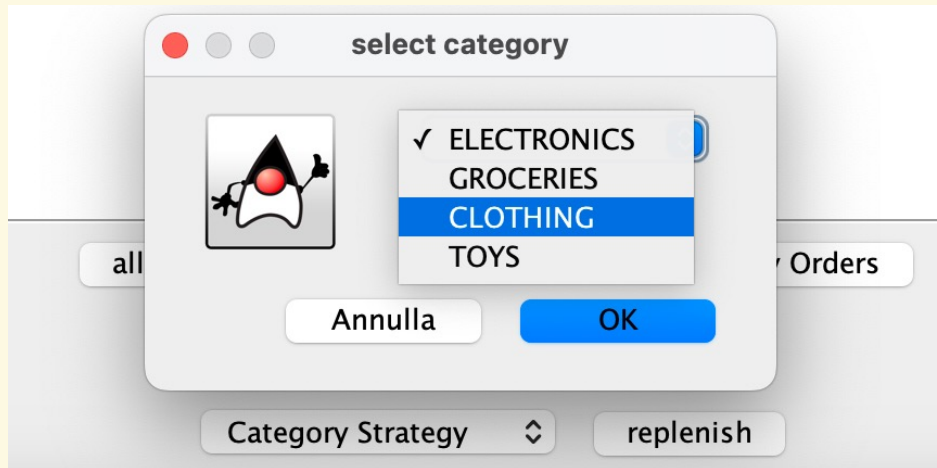
ids	nome cognome	address	email	price	maxQty
S04	Francesca Neri	francesca.neri@e...	Via Firenze 40, V...	0.19	100
S01	Mario Rossi	mario.rossi@exa...	Via Roma 10, Mil...	0.3	300
S03	Luca Bianchi	luca.bianchi@exa...	Via Napoli 30, R...	0.4	200
S05	Giovanni Bruno	giovanni.bruno@...	Via Genova 50, B...	0.5	50



# Rifornimento

**Gestione efficiente delle forniture  
con:**

- **Creazione manuale di nuovi ordini**
- **Strategie per il rifornimento**







# Pattern

## DAO

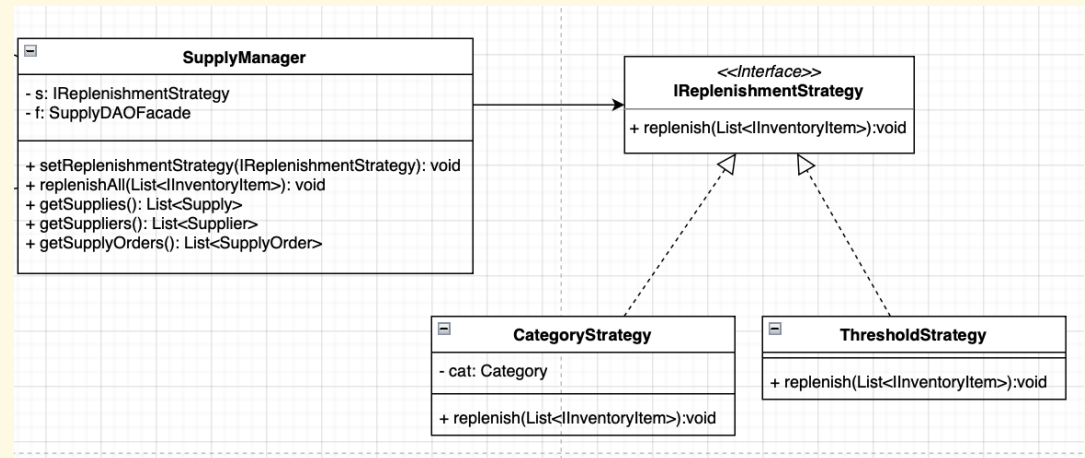
- per separare le operazioni sui dati persistenti dal resto dell'applicazione

## Facade

- per unire i metodi delle classi DAO in un'unica classe
- oggetto di facciata

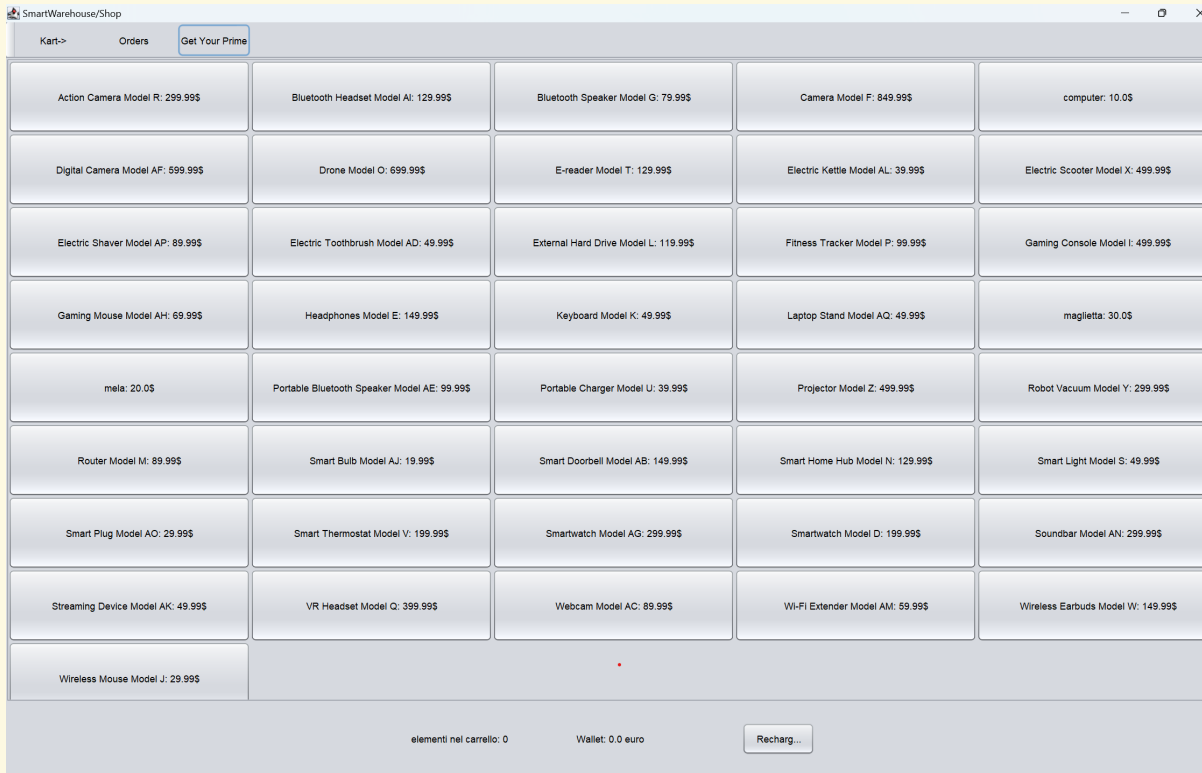
## Strategy

- per gestire diversi algoritmi di rifornimento sfruttando il polimorfismo (la versione del metodo invocata dipende dall'oggetto attualmente in memoria)





# Shop e funzionalità

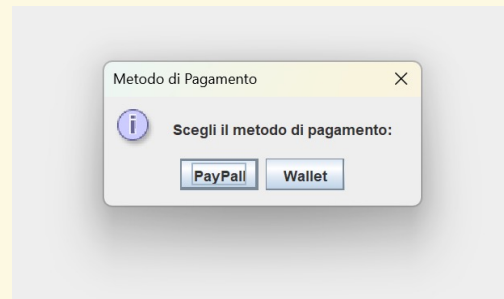
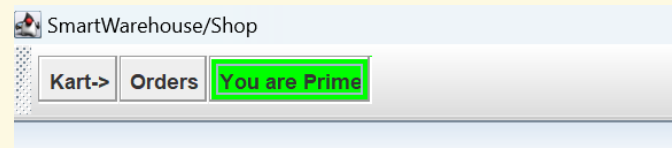
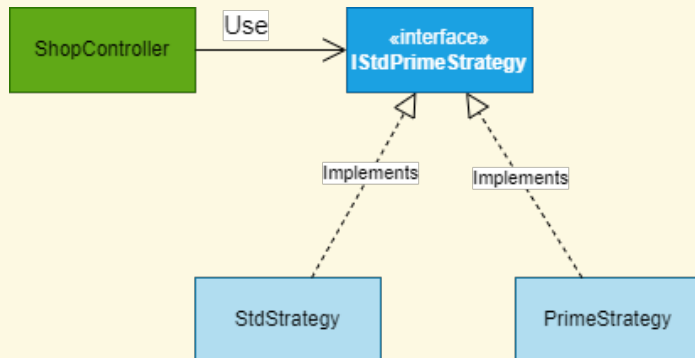


- Visualizzazione di tutti gli Item forniti
- Selezione di un Item
- Ricarica del Wallet
- Accesso all'area Ordini e Resi
- Navigazione tra Shop e Carrello
- Acquisto e creazione di Ordini
- Scelta del metodo di pagamento



# Shop pattern strategy

**Nel caso d'uso relativo alla gestione dello shop è stato inserito un pattern GOF (Gang Of Four) Strategy per gestire la possibilità di diverse strategie di pagamento nel caso di un cliente abbonato oppure no facili da modificare o estendere.**





# Picking e funzionalità

Order

Order IDs

Selected Order Details:

Package Summary:

Inserted package:

Inserted Items:

Small

Medium

Large

Item

Calculate Package

Packed

1

3

6

7

8

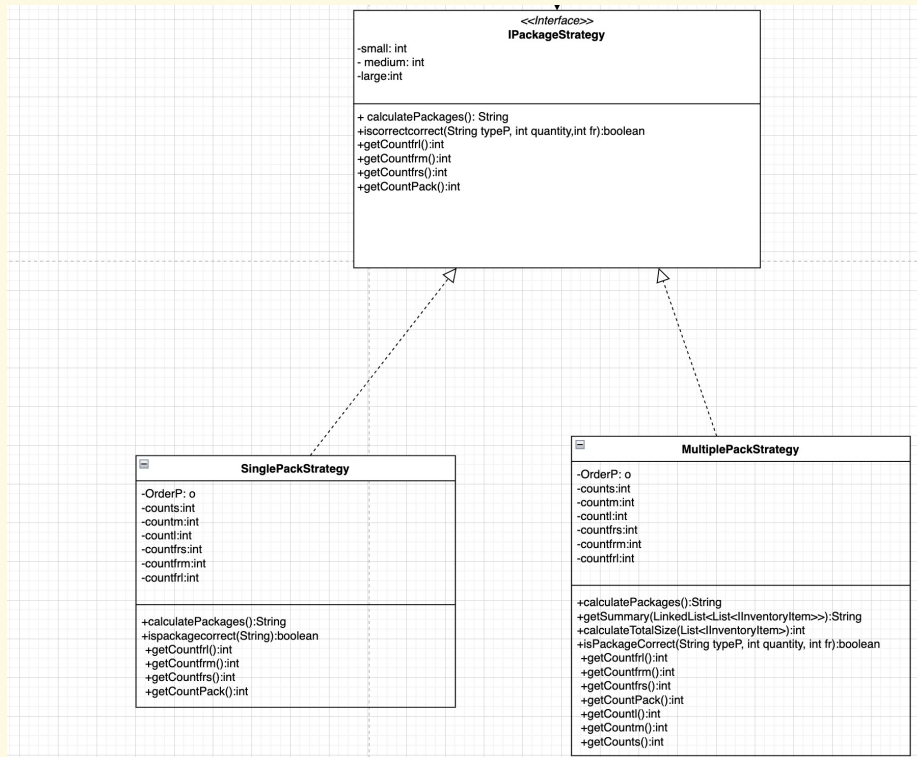
9

11

- **Visualizza tutti gli ordini non pickati**
- **Calcola: quanti pacchi mi servono, se sono fragili e la tipologia**
- **Facilità il picking a prendere gli item e i pacchi corretti**



# Strategy per picking



Per semplificare la scelta del pacco è stato implementato un GOF ossia lo strategy, semplificando così l'algoritmo di selezione della quantità di pacchi.

# RETURN SERVICE FUNZIONALITA'

Servizio di restituzione degli articoli acquistati.

☐ 2 T-shirt basic black L, 2024-06-05T13:13:03


☐ 1 Basic SweatShirt M, 2024-06-05T16:22:23

☐ 2 Basic SweatShirt M, 2024-06-05T18:46:20

☐ 5 T-shirt basic black S, 2024-06-05T17:24:56

☒ 2 T-shirt basic black M, 2023-06-05T17:45:10

Error

 You can no longer return this order: too many days have passed!

OK

Confirm >



## VISUALIZZAZIONE DI TUTTI GLI ORDINI EFFETTUATI.

Visualizzazione ordinata per data.



## SELEZIONE DI UN ORDINE

Selezione dell'ordine di cui si intende avviare la procedura di reso.



## CONDIZIONI DI RESTITUIBILITA'

Massimo 30 giorni per effettuare un reso



## NAVIGAZIONE

- Back
- Confirm
- Close\_Operation

# RETURN SERVICE FUNZIONALITA'

Prodotti da restituire, motivazioni,  
modalità di rimborso.

The screenshot displays a web interface for the return service. At the top, there are buttons: '< Back', 'Deselect All', 'Select All', and 'InfoPoint'. Below these, it says 'Selected order: 1' and 'Choose what to return'. A list of items is shown with checkboxes and dropdown menus for reasons:

- ☒ T-shirt basic black M: Choose a reason
- ☐ T-shirt basic black M: Choose a reason
- ☒ T-shirt basic black M: The product is not as described
- ☐ Basic SweatShirt M: Choose a reason
- ☐ Basic SweatShirt M: Choose a reason
- ☐ T-shirt basic black S: Choose a reason
- ☒ T-shirt basic black S: Other

Below the list, there is a section 'Describe the reason for the return' with a text area labeled 'PERSONALIZED REASON'. At the bottom, there are more items with reasons:

- ☒ T-shirt basic black S: I found the product at a lower price
- ☒ T-shirt basic black L: I changed my mind
- ☒ T-shirt basic black L: Defective item
- ☐ Basic SweatShirt M: Choose a reason

An error dialog box is open in the center, titled 'Error', with a red exclamation mark icon and the text 'Missing reason'. It has an 'OK' button.

At the bottom, there is a 'Refund method' section with a cyan background. It says 'Choose the way you prefer to be reimbursed:' and lists two options:

- ☒ VOUCHER (recommended): receive an instant refund on a non-expiring voucher that can be spent throughout the shop.
- ☐ Bank Transfer: receive a transfer to the card you used to make the purchase. Estimated time 14 working days.

A 'Next' button is at the bottom right.



## SCELTA DEI PRODOTTI DA RESTITUIRE

Visualizzazione e scelta di tutti gli articoli  
dell'ordine selezionato.



## MOTIVAZIONE DELLA RESTITUZIONE

Per restituire un articolo è obbligatorio  
indicare una motivazione valida;  
predefinita o personalizzata



## MODALITÀ DI RIMBORSO

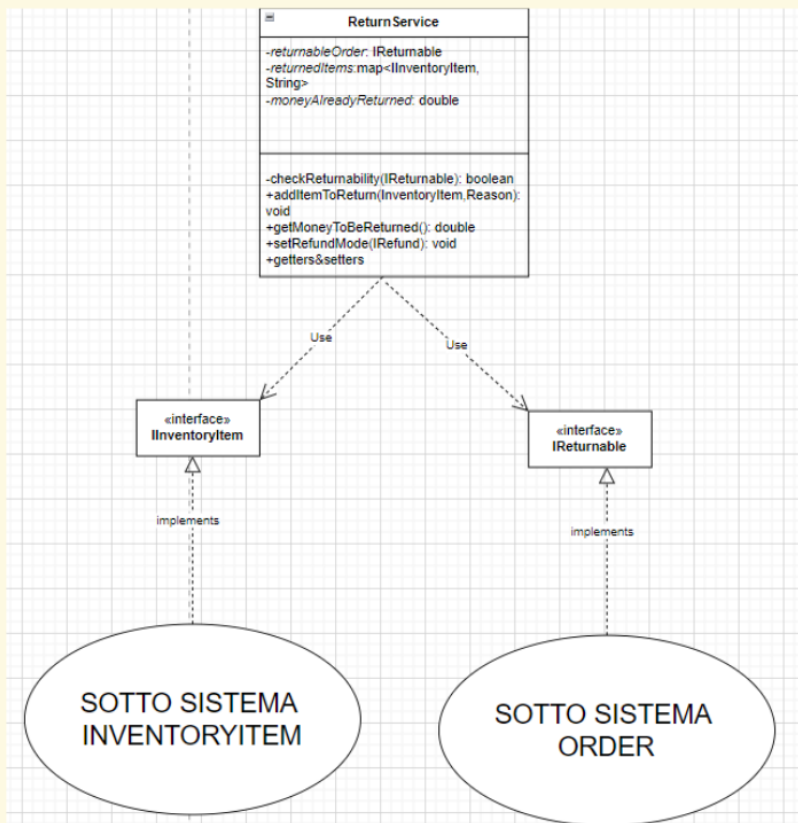
E' possibile scegliere la modalità di  
rimborso desiderata tra le possibili  
opzioni.



## SMARTWAREHOUSE INFO POINT

Info point panel con sezione help, e recap  
degli articoli restituiti.

# RETURN SERVICE. PROGETTAZIONE ED IMPLEMENTAZIONE



## INTERFACCIA IRETURNABLE

La classe Order si impegna a mettere a disposizione certe funzionalità.



## INTERFACCIA IINVENTORYITEM

La classe InventoryItem si impegna a mettere a disposizione certe funzionalità.



## INTERFACCIA COME CONTRATTO

Definisce un contratto in forma puramente astratta. Interfacciamo il sottosistema ReturnService con Order e Item senza legarci ad una specifica gerarchia di classi.



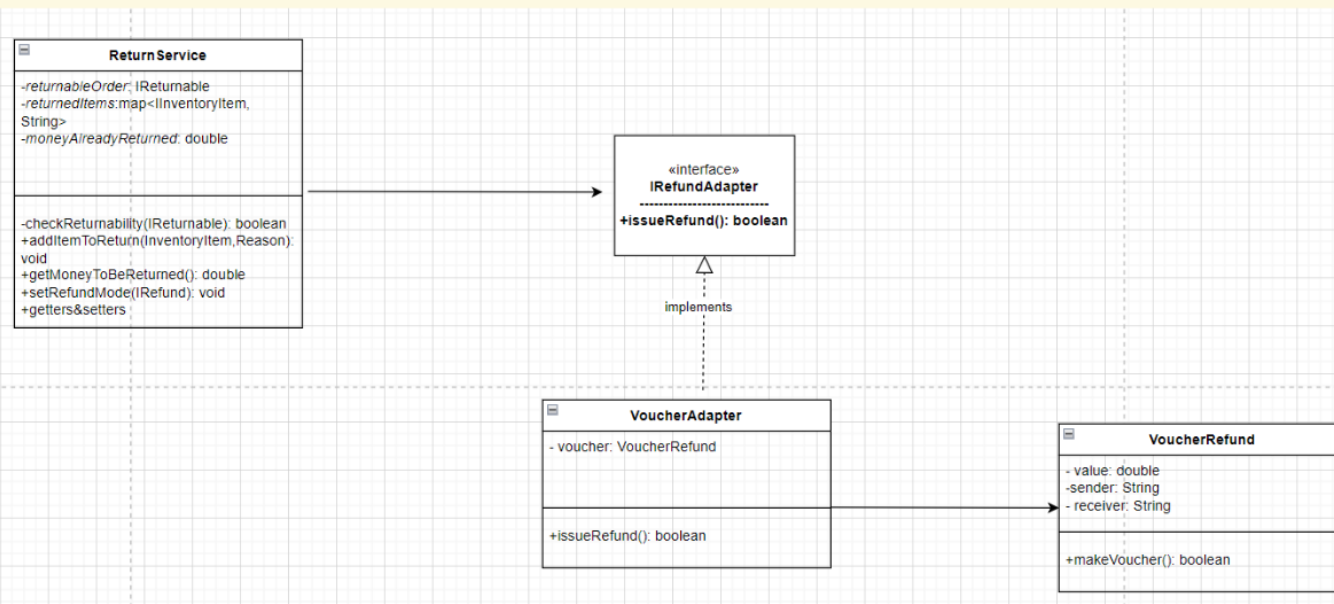
## DEPENDENCY

Accoppiamento nei confronti di elementi stabili e pervasivi



# RETURN SERVICE. PROGETTAZIONE ED IMPLEMENTAZIONE

```
public boolean issueRefund(IRefund rm) throws PaymentException {  
    return rm.issueRefund();  
}
```



## ADAPTER

Traduttore per sottosistemi che fanno la stessa cosa ma con lingue diverse. (e.g VoucherAdapter, BankTransferAdapter)



## ESTENDIBILITÀ

Possibilità di integrazione con sistemi di rimborso futuri non ancora esistenti.



## POLYMORPHISM

Evitare di usare la logica condizionale per eseguire alternative dipendenti dal tipo.



## CODICE COMPATTO E LEGGIBILE

```
public static boolean checkReturnability(ReturnService returnService){  
    LocalDate criticalDate=returnService.getCriticalDate();  
    LocalDate now=LocalDate.now();  
    return now.isAfter(criticalDate) ? false : true;  
}
```

**Progetto J24 università degli studi di  
pavia**



# **Grazie per l'attenzione**

**giul02  
cienzman  
frazz02  
rosci01**

