

Progetto SWE

Beatrice Vangi e Caterina Giardi

14 aprile 2022

1 Statement

Lo scopo del nostro progetto è quello di simulare la gestione di un negozio di fiori. Un cliente, dopo aver fatto il login, effettua un ordine online, il quale può essere un fiore, un bouquet, una decorazione o una composizione di questi. In particolare i bouquet, composti da fiori e da un'eventuale decorazione, possono essere scelti dal catalogo.

Una volta che giunge un ordine effettuato da un cliente, si prende in carico e si comunica al cliente, tramite mail, che l'ordine è in fase di elaborazione. Tutto ciò viene effettuato sfruttando il pattern Observer, il quale considera come Subject l'OrderList. Quest'ultimo contiene una lista di ordini e, non appena viene aggiunto un ordine alla lista, la classe notifica il CustomerNotifier che procede all'invio della mail al cliente. Inoltre quest'ultimo è in grado di controllare lo stato degli ordini (in preparazione o spedito) e il loro contenuto tramite la funzione myOrder().

L'OrderManager, incaricato dalla classe Program (che è un Singleton), si occupa di creare l'ordine, di inserirlo e di rimuoverlo dalla lista degli ordini.

Una volta che l'ordine è stato preso in carico, si procede all'elaborazione. In questa fase il Florist preleva l'ordine dalla lista e si occupa della sua preparazione. Al fine di poter considerare l'ordine come un insieme di prodotti, sfruttiamo il pattern Composite. In questo modo possiamo considerare sia il bouquet che il fiore che l'eventuale decorazione come un Product, senza fare distinzioni. Ogni fiore ha un costo, di conseguenza il prezzo dei bouquet è dato dalla somma di tutti i fiori che lo compongono e dal prezzo delle decorazioni eventuali. Per la creazione dello stesso, si consulta l'ordine che implicitamente specifica come è composto. Dal momento che si fa in modo che in magazzino ci siano sempre le risorse necessarie alla creazione dei prodotti presenti sul catalogo, il cliente riceverà sempre il prodotto richiesto.

La classe Supplier rappresenta i fornitori, i quali hanno un loro catalogo e il cui ruolo è quello di istanziare i fiori e inviarli allo storage così che possano essere immagazzinati. Tutto ciò avviene tramite un pattern Observer nel quale i Supplier svolgono in ruolo di

ConcreteObserver, infatti quest'ultimi invieranno i fiori una volta notificati dallo Storage, che è un ConcreteSubject. La notifica viene inviata nel momento in cui il fioraio constata che le quantità di fiori presenti in magazzino sono inferiori a una determinata soglia.

Al completamento dell'ordine, corrisponde una notifica inviata al cliente, anche questa procedura è implementata seguendo lo scheletro del pattern Observer. La notifica è inviata dal CustomerNotifier (ConcreteObserver) e consiste nell'invio di una mail al cliente, in cui si comunica che l'ordine è stato evaso. Dopodiché il fioraio pianifica il ritiro dell'ordine da parte della compagnia di spedizione (ShippingCompany).

2 Use Case Diagram

La prima fase di progettazione si è incentrata sulla stesura dello Use Case Diagram (Figura 1) , che illustra come si strutturano le relazioni tra attori e casi d'uso.

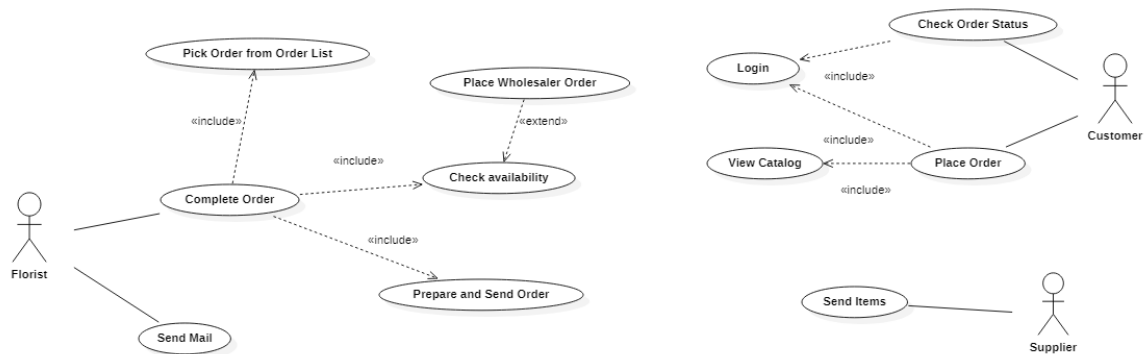


Figura 1: Use Case Diagram

3 UML

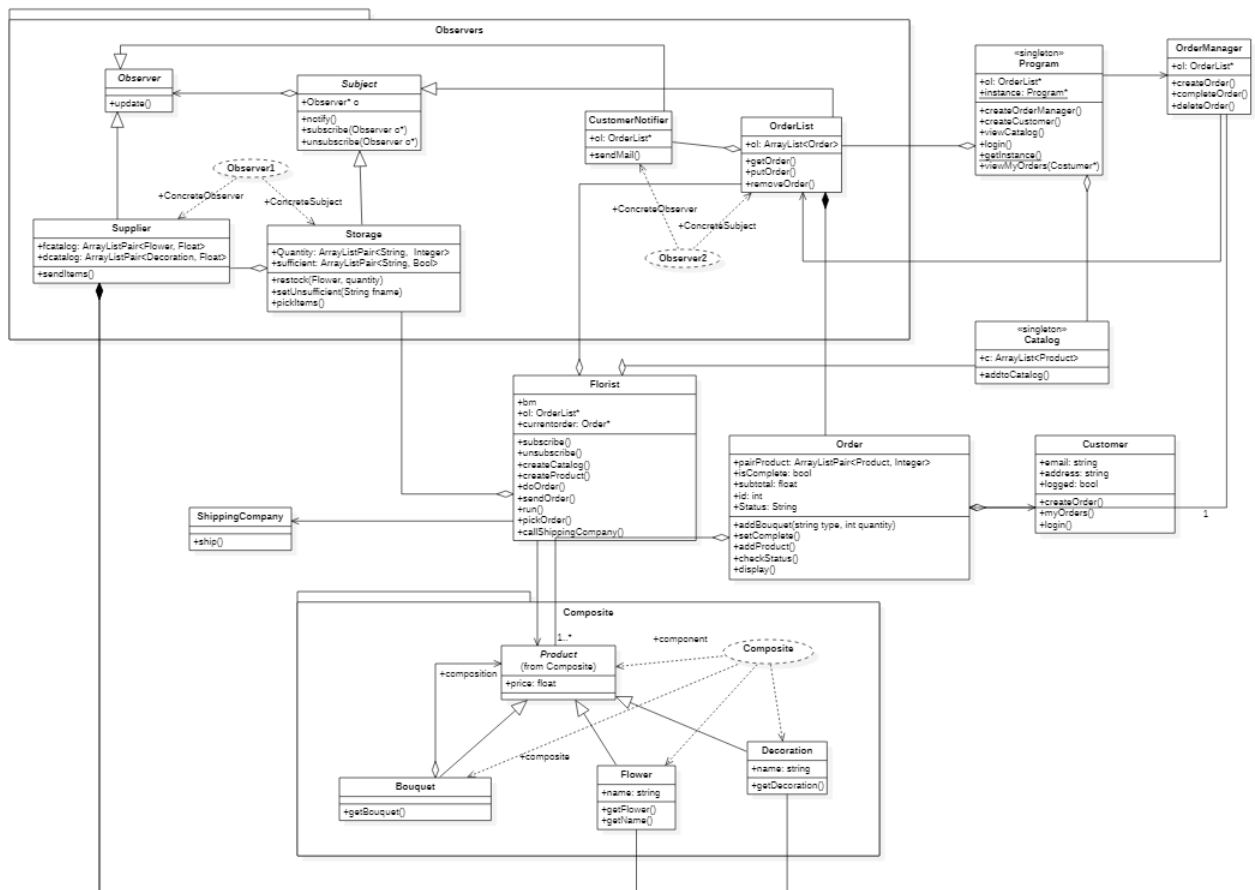


Figura 2: UML

4 Template

UC	Place Order
Level	User Goal
Actor	Customer
Basic Course	<ol style="list-style-type: none">1. L'utente esegue il login2. L'utente clicca su View Catalog3. Il sistema mostra la finestra Catalog4. L'utente sceglie gli articoli da comprare cliccando sull'icona del carrello dell'articolo corrispondente5. L'utente visiona gli articoli nel carrello cliccando su Cart6. Il sistema mostra la finestra Cart7. L'utente conferma l'ordine8. L'utente riceve la mail di presa in carico dell'ordine
Alternative Course	<ol style="list-style-type: none">7a. L'ordine non viene confermato perché il carrello è vuoto<ol style="list-style-type: none">7a.1. Il sistema mostra un dialog di fallimento

UC	Prepare Order
Level	User Goal
Actor	Florist
Basic Course	<ol style="list-style-type: none">1. Il fioraio apre la lista degli ordini.2. Il sistema mostra la finestra Order List3. Il fioraio preleva il primo ordine4. Il fioraio clicca su Storage5. Il sistema mostra la finestra Storage6. Il fioraio controlla la disponibilità degli articoli ordinati7. Il fioraio preleva gli articoli dallo storage8. Il fioraio completa l'ordine9. Il fioraio chiama la ditta di spedizioni10. Il fioraio spedisce l'ordine11. Il fioraio manda la mail al cliente
Alternative Course	<ol style="list-style-type: none">7a. Gli articoli rimasti nello storage sono sotto la soglia minima<ol style="list-style-type: none">7a.1. Il fioraio clicca su "Go to Supplier"7a.2. Il fioraio ordina l'articolo mancante

I template sopra esposti definiscono le funzionalità che stanno alla base del software.

5 Mock Up

I Mock Up riportati (Figure 3 e 4) designano l'interfaccia grafica destinata agli utenti.

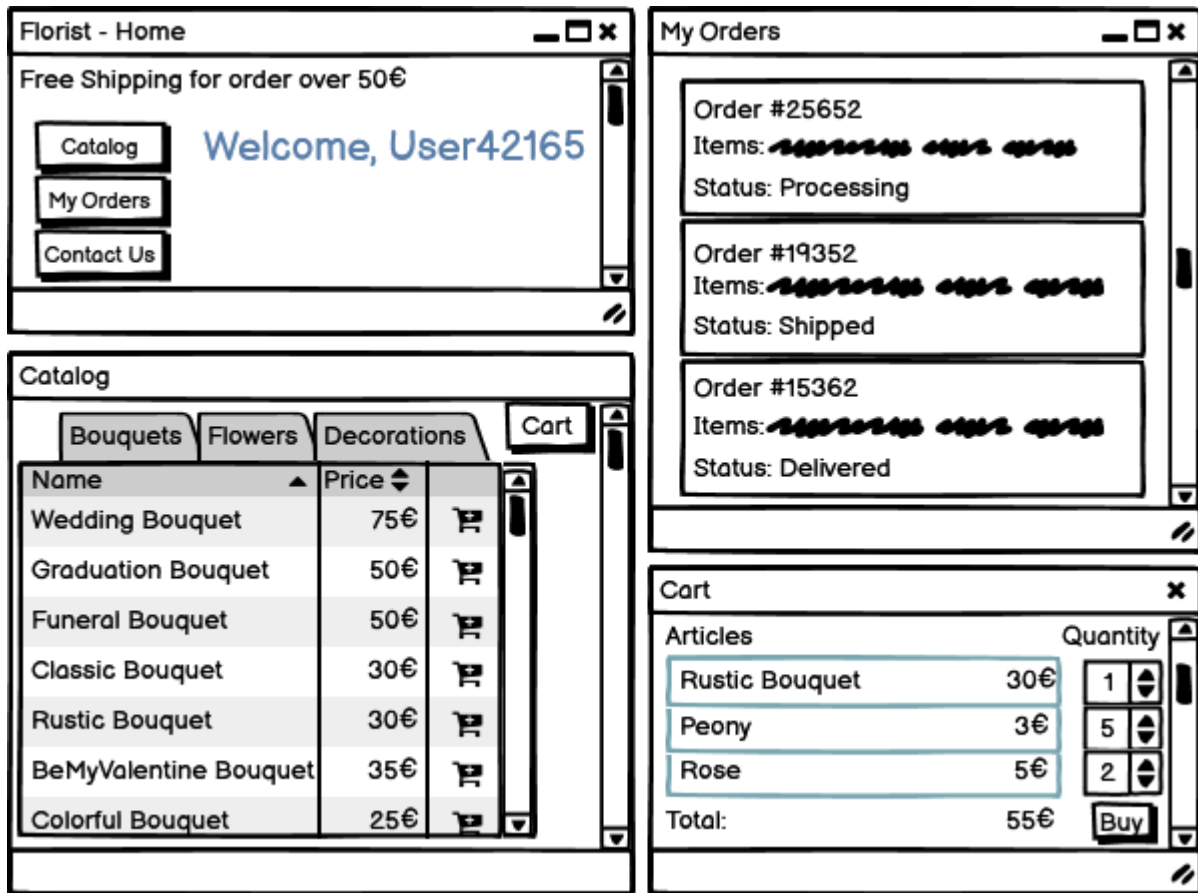


Figura 3: Costumer POV Mock Up

Order List

☐ Order 1: 5 Wedding Bouquets, 10 Peonies, 2 Paper Wraps
☐ Order 2: 2 Graduation Bouquets, 1 Classic Bouquet
☐ Order 3: 5 Tulips, 1 Ribbon Bow, 1 Tulle, 1 Paper Wrap

☐ to do
☒ done
☐ processing

Call Shipping Company

Storage

Supplier

Flow&Dec Wholesaler

Order Flower or Decoration

View Catalog

New items available

100 pieces minimum!

Storage

Name ▲	Quantity ◆	Minimum amount
Rose	100	35
Tulip	92	30
Gardenia	65	20
Peony	81	20
Laurel	73	15
Ribbon Bow	46	10
Paper Wrap	59	20
Tulle	38	15

Supplier

Figura 4: Florist POV Mock Up