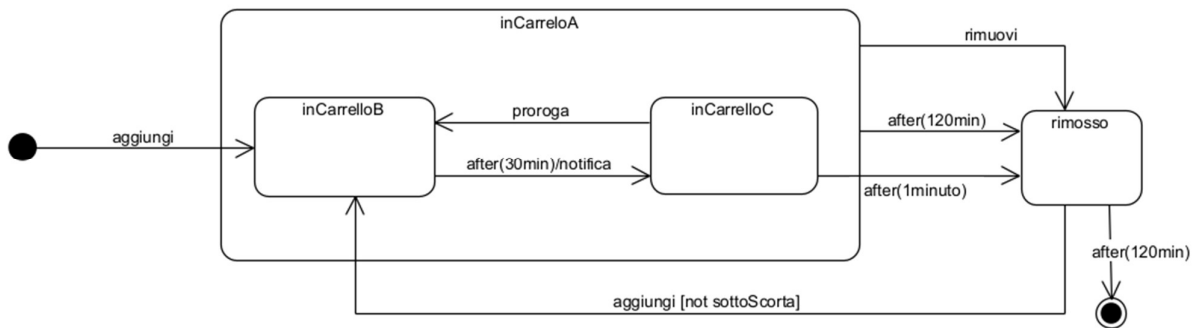


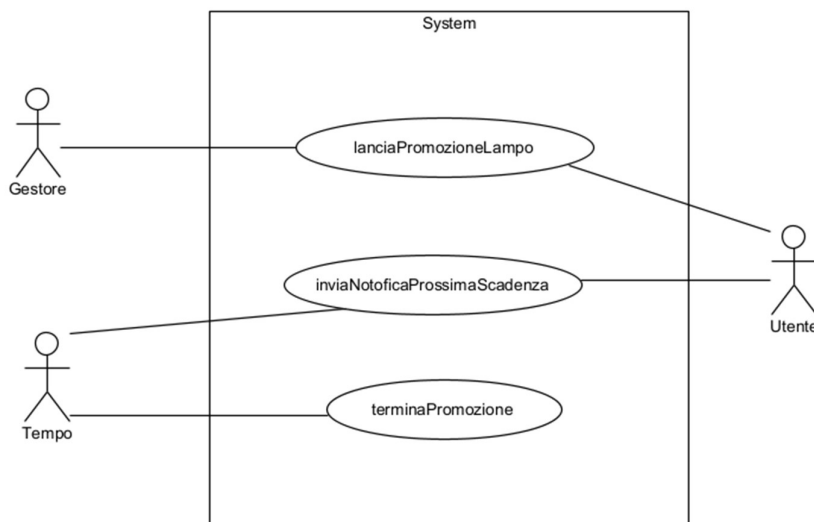
Gli occhiali d'oro

Domanda 1 (7 punti) Usando un diagramma UML opportuno, modellare i requisiti sul carrello virtuale indicati nel penultimo paragrafo del testo, riscritto di seguito in modo meno ambiguo: Il sistema deve permettere agli utenti di conservare ogni articolo nel carrello virtuale per un tempo massimo di 120 minuti dall'aggiunta. Se un articolo non viene acquistato entro 30 minuti, il sistema deve notificare l'utente offrendogli l'opzione di prorogare la permanenza nel carrello per ulteriori 30 minuti. Se l'utente non proroga la permanenza nel carrello entro 1 minuto dalla notifica, allora l'articolo viene rimosso dal carrello. La proroga può essere esercitata fino a raggiungere il limite di 2 ore. Allo scadere di questo termine, l'articolo sarà rimosso automaticamente dal carrello e reso disponibile ad altri clienti. Articoli rimossi che risultino sotto il livello minimo di scorta non potranno essere reinseriti nel carrello dallo stesso utente per un periodo di 2 ore.



Si consideri ora il seguente nuovo requisito: I gestori del negozio virtuale possono lanciare promozioni lampo con sconto del 30% che durano 7 minuti. Il sistema deve: aggiornare il prezzo sul catalogo; inviare una notifica agli utenti nel cui carrello virtuale c'è un articolo in promozione lampo; inviare agli stessi una seconda notifica a 2 minuti dallo scadere della promozione.

Domanda 2 (5 punti) Fornire il diagramma dei casi d'uso con la relativa narrativa ESCLUSIVAMENTE in riferimento al nuovo requisito.



Caso d'uso: lanciaPromozioneLampo

Breve descrizione: il gestore lancia una promozione al 30% su un articolo e tutti gli utenti con quell'articolo nel carrello sono notificati

Attore principale: Gestore

Attori secondari: Utente

Pre-condizione: articolo non in promozione lampo

SPE:

1. Il Gestore indica l'articolo su cui fare la promozione
2. Il Sistema aggiorna il catalogo
3. Per (ogni utente utenti con quell'articolo nel carrello)
 - a. Il Sistema manda una notifica

Post:condizione: articolo in promozione lampo, catalogo aggiornato, utenti notificati

SAE: nessuna

Caso d'uso: invioNotificaProssimaScadenza

Breve descrizione: a 2 minuti dallo scadere della promozione lampo tutti gli utenti con quell'articolo nel carrello sono notificati

Attore principale: Tempo

Attori secondari: Utente

Pre-condizione: articolo in promozione lampo da 5 minuti

SPE:

1. Per (ogni utente utenti con quell'articolo nel carrello)
 - a. Il Sistema manda una notifica

Post:condizione: articolo in promozione lampo, utenti notificati per la seconda volta

SAE: nessuna

Caso d'uso: terminaPromozione

Breve descrizione: al termine dei 7 minuti di promozione lampo il prezzo in catalogo ritorna al valore originale

Attore principale: Tempo

Attori secondari: nessuno

Pre-condizione: : articolo in promozione lampo da 7 minuti

SPE:

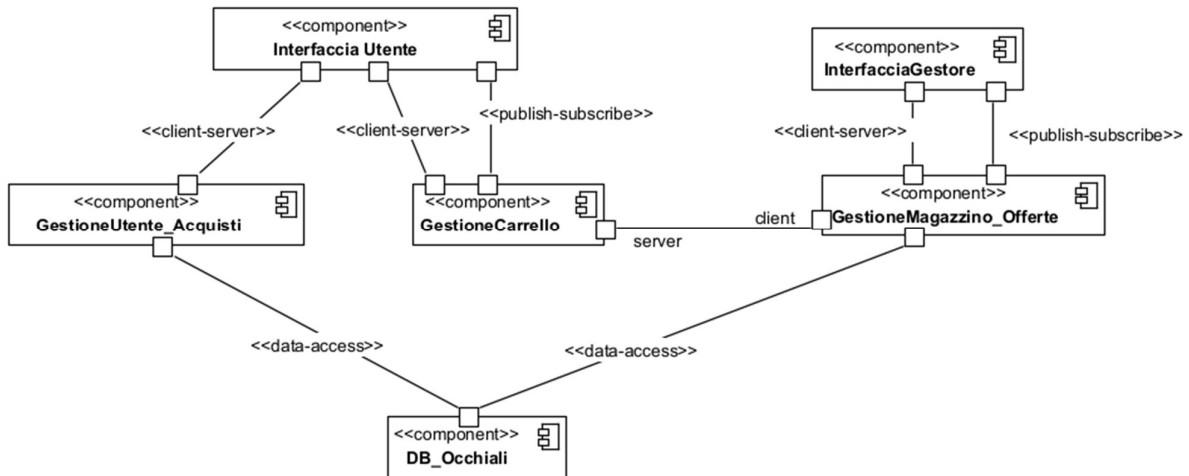
1. Il Sistema aggiorna il catalogo

Post:condizione: articolo non in promozione lampo, catalogo aggiornato

SAE: nessuna

(andava ragionevolmente bene anche con solo un caso d'uso e le notifiche e i ripristino del prezzo precedente nella sequenza principale degli eventi)

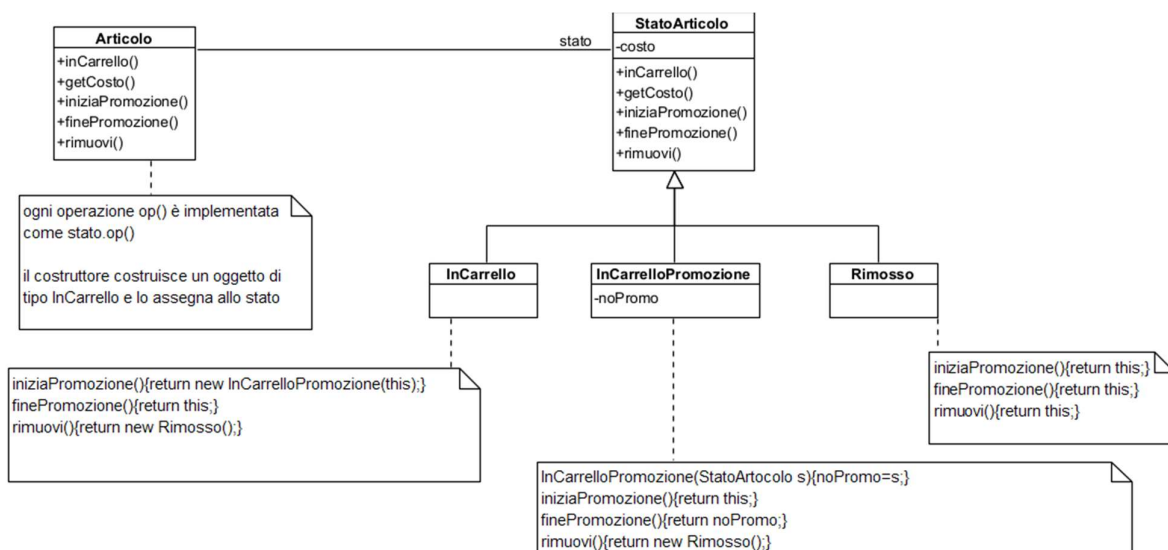
Domanda 3 (6 punti) Dare un diagramma componenti e connettori che descriva l'architettura del sistema. Considerare ANCHE il nuovo requisito.



Domanda 4 (7 punti) Per semplicità in questo esercizio si dimentichino le proroghe e si assuma che ogni articolo rimanga nel carrello virtuale per un tempo massimo di 120 minuti. Un oggetto di tipo articolo viene creato quando viene messo nel carrello e ha 3 stati (inCarrello, inCarrello_promozione, rimosso) e i seguenti metodi la cui implementazione dipende dallo stato dell'articolo:

- `int inCarrello()` che dice da quanto tempo è nel carrello e restituisce un valore negativo se l'articolo non è nel carrello
- `int getCosto()` che restituisce il costo dell'articolo, e restituisce un valore negativo se l'articolo non è nel carrello
- `StatoArticolo iniziaPromozione()`
- `StatoArticolo finePromozione()`
- `StatoArticolo rimuovi()`

Dare un diagramma delle classi che mostra come usare il pattern State per modellare il comportamento di un articolo. Dare lo pseudocodice dell'implementazione di un metodo non banale, nella classe Articolo e nei vari stati.



Domanda 5 (5 punti) Nel definire lo scaffolding per testare il corretto monitoraggio dell'inventario si deve costruire un driver di test. Una funzionalità del driver è per esempio simulare una vendita e verificare che l'inventario venga aggiornato di conseguenza. Indicare altre 2 funzionalità che devono essere realizzate dal driver sulla base del testo del progetto.

Soluzione

Simulare un'impennata di vendite e verificare che i gestori siano notificati

Simulare una vendita di un numero di prodotti maggiore della scorta e verificare che fallisca