

Relazione Progetto E-Shop

Stefano Nicolis - Alberto Molinari
A.A. 2019-2020

25 giugno 2020



Corso di
Ingegneria del Software

Indice

1	Introduzione	2
2	Sviluppo e attività di test	3
3	Descrizione Software e Scelte Progettuali	4
3.1	Linguaggio di scelta	4
3.2	Database dei dati	4
3.3	Design Pattern e applicazione	4
3.4	Tipi di utenza	4
3.5	Istanze multiple	5
3.6	Note utili	5
4	Viste principali	6
4.1	Login.fxml	6
4.2	Registration.fxml	7
4.3	Shop.fxml	8
4.4	ShoppingCart.fxml	9
4.5	Customer.fxml	10
4.6	Payment.fxml	11
4.7	Employee.fxml	12
5	Casi d'uso	13
5.1	Casi d'uso relativi al Customer	13
5.2	Casi d'uso relativi all'Employee	16
6	Sequence diagram per i principali Use Case	17
7	Application Diagrams	21
7.1	Class Diagram	21

1 Introduzione

Viene richiesta l'implementazione di un sistema informatico per la gestione del servizio di spesa on-line di un supermercato. Tale applicazione costituisce un modo per mettere in pratica le conoscenze di Ingegneria del Software acquisite nella seconda parte del corso di Programmazione II, tra le quali metodologie di sviluppo, documentazione e test del software. Questa applicazione prende il nome di **E-Shop**.

2 Sviluppo e attività di test

L'applicazione è stata sviluppata con metodologia incrementale.

Il codice è stato scritto sia in modo individuale che durante sessioni di pair-programming tramite Skype, svoltesi ogni circa due giorni durante un lasso di tempo di circa tre settimane.

Ogni sessione di pair programming è stata così strutturata:

- A turno ognuno spiegava all'altro le parti di applicazione sviluppate indipendentemente concordate nella sessione precedente, esponendo i progressi, eventuali idee avute e problemi incontrati
- Le due parti di codice venivano integrate in un progetto unico, condiviso con l'altro e testato subito su Windows e Ubuntu
- Le nuove parti di codice da implementare venivano determinate, si iniziava a svilupparle assieme nell'ultima parte della sessione per poi terminare in maniera autonoma entro la sessione successiva

La durata media di una sessione era di due ore.

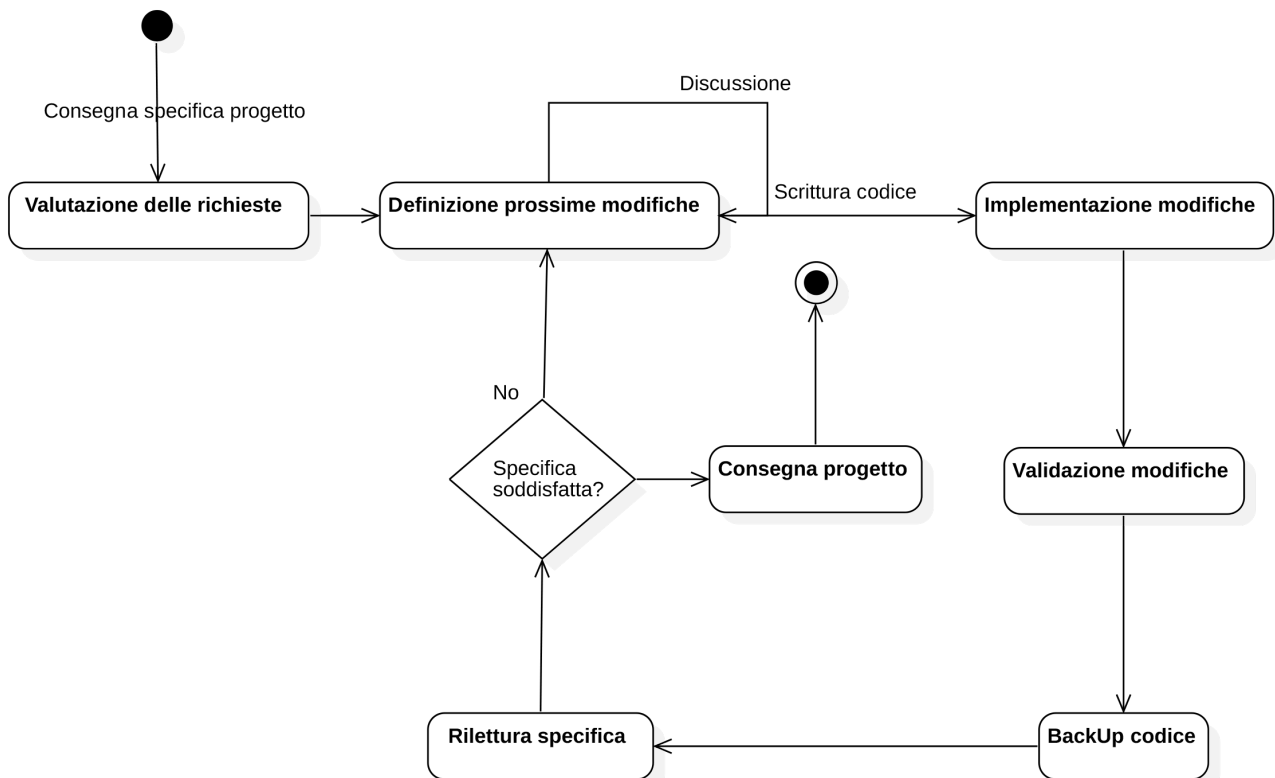


Figura 1: Activity Diagram delle attività di sviluppo/test dell'applicazione

3 Descrizione Software e Scelte Progettuali

Di seguito vengono elencate le scelte progettuali più significative.

3.1 Linguaggio di scelta

Per la realizzazione del progetto è stato scelto il linguaggio **Java** con l'ausilio di **JavaFx** per l'implementazione della parte grafica. La scelta è ricaduta su tali tecnologie perchè, oltre che essere oggetto del corso di studi, tutti i requisiti del progetto erano realizzabili senza problemi con esse, e quindi non abbiamo sentito il bisogno di andare a valutare altre opzioni.

3.2 Database dei dati

I vari database dell'applicazione, ovvero la lista degli **utenti**, dei **prodotti** e delle **spese effettuate**, sono realizzati tramite file **.txt** sui quali vengono scritte le strutture dati utilizzate nel codice, nel nostro caso **HasMap** contenenti oggetti di tipo **Product**, **User** e **ShoppingCart**. Ciò è permesso dall'implementazione dell'interfaccia **Serializable** da parte degli oggetti presenti nei database. Abbiamo scelto questo approccio perchè, anche se entrambi avevamo un minimo di esperienza con database relazionali e file xml/json, volevamo fare tutto tramite gli strumenti Java, oltre che sperimentare questo nuovo approccio per la gestione dati che nessuno dei due aveva mai utilizzato prima.

3.3 Design Pattern e applicazione

Il design pattern di scelta per eShop è **Model-View-Control** (MVC): oltre che essere fortemente indicato per l'architettura Vista-Controller di JavaFx, non abbiamo trovato motivo di scegliere altro in quanto esso, oltre ad essere molto facile da usare, ci ha permesso di implementare tutto ciò richiesto dalla specifica con semplicità.

Nel nostro caso, la parte Model è costituita dalle classi del package *models*, ovvero le classi che descrivono gli oggetti che vengono utilizzati nell'applicazione (ad esempio ShoppingCart.java e le enumerazioni Ward.java, PaymentMethod.java), la parte View è costituita dalle viste di JavaFx, ovvero i vari file **.fxml** del package *views*, e Control sono i rispettivi controllori delle viste contenuti nel package *controllers*.

3.4 Tipi di utenza

Gli utenti sono suddivisi in clienti (Customer) e dipendenti (Employee).

- **Customer:** può effettuare una spesa all'interno dello shop, controllare il proprio record spese e verificare lo stato degli ordini
- **Employee:** può controllare le spese effettuate dai Customer, aggiungere e rimuovere prodotti dallo shop e modificare alcune informazioni come il prezzo e la quantità disponibile dei prodotti già presenti nel database.

Entrambe le categorie hanno a disposizione delle credenziali con cui possono effettuare l'autenticazione: i clienti per accedere al sistema usano le credenziali create in fase di registrazione, mentre i dipendenti usano le credenziali fornite dagli amministratori del sistema. Nel caso in cui l'autenticazione vada a buon fine, ogni categoria di utenti viene reindirizzata alla corrispondente interfaccia.

3.5 Istanze multiple

L'integrità dei database in presenza di istanze multiple dell'applicazione viene garantita nei seguenti modi:

- **Utenti loggati:** all'interno della classe User è presente una flag che viene messa a true quando un utente fa il login e false quando fa il logout, in questo modo non è possibile loggarsi con la stessa utenza in due istanze diverse del programma contemporaneamente.
- **Prodotti disponibili:** quando un Customer si logga, viene letto il database dei prodotti da file e caricati i prodotti nella GUI dello shop, a momento di pagamento viene letta una nuova istanza del database dei prodotti per assicurarsi che i prodotti scelti dal Customer corrente siano effettivamente ancora disponibili, ovvero verificare che un'altro utente non abbia nel frattempo effettuato una spesa che renda parzialmente o completamente non soddisfabile la quantità dei prodotti richiesti dall'utente corrente.

3.6 Note utili

- Tutti i file utilizzati dall'applicazione, .txt e .png, vengono caricati come risorsa all'interno dell'applicazione e non come oggetti File.
- Le icone utilizzate nell'applicazione sono state prese da un sito di immagini stock.

4 Viste principali

Di seguito vengono elencate le principali viste utilizzate.

4.1 Login.fxml

Questa vista rappresenta un'interfaccia comune a Customer ed Employee per il login. Un utente non registrato può accedere alla vista Registration.fxml tramite l'apposito bottone per registrare una nuova utenza di tipo Customer. Le utenze di tipo Employee si assume vengano inserite dagli sviluppatori, pertanto non è possibile registrare utenze di tipo Employee da questa schermata. Per questioni di test viene fornita una sola utenza Employee con email e password = (8,8)

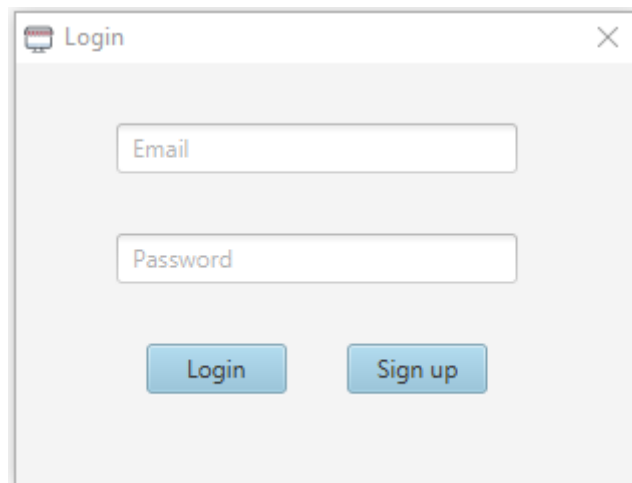
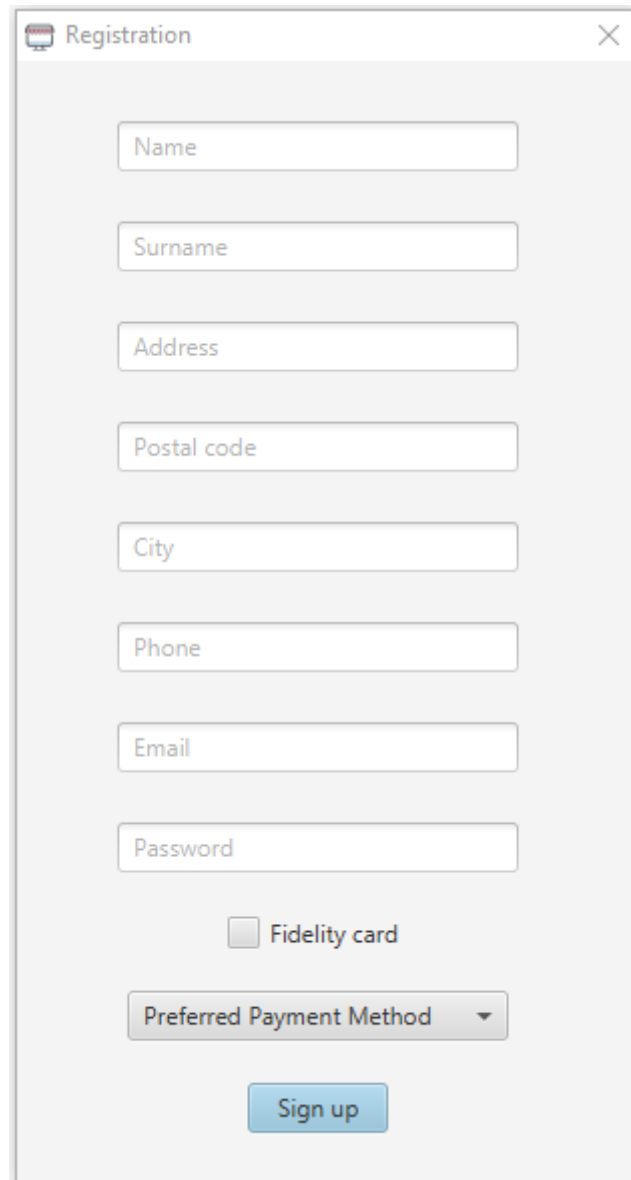


Figura 2: Vista Login.fxml

4.2 Registration.fxml

Tramite questa vista un potenziale utente Customer può registrarsi sulla piattaforma eShop. L'utente può scegliere se associare al proprio account una Fidelity Card e un metodo di pagamento preferito.

Si ricorda che **non** è possibile creare utenze di tipo Employee tramite questa schermata.



The image shows a JavaFX window titled "Registration" with a close button (X) in the top right corner. The window contains a registration form with the following elements:

- Text input field for "Name"
- Text input field for "Surname"
- Text input field for "Address"
- Text input field for "Postal code"
- Text input field for "City"
- Text input field for "Phone"
- Text input field for "Email"
- Text input field for "Password"
- A checkbox labeled "Fidelity card" (currently unchecked)
- A dropdown menu labeled "Preferred Payment Method" with a downward arrow
- A blue "Sign up" button at the bottom

Figura 3: Vista Registration.fxml

4.3 Shop.fxml

Questa vista rappresenta l'interfaccia grafica per l'utente di tipo Customer. La schermata è divisa in tre riquadri principali:

- Sinistra: lista dei prodotti messi nel carrello
- Centro: vengono visualizzati i prodotti dello shop
- Destra: informazioni riguardo il prodotto correntemente selezionato

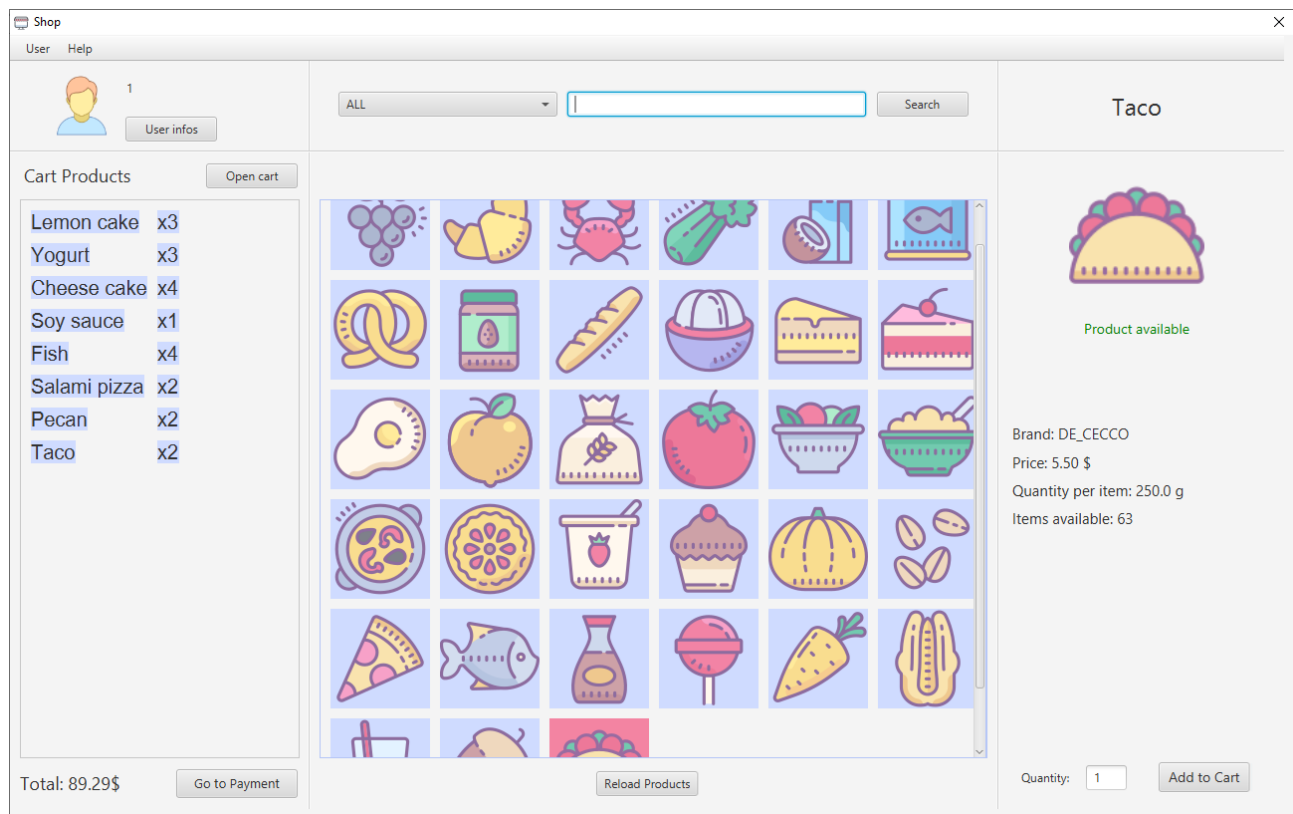


Figura 4: Vista Shop.fxml

Per poter aggiungere un prodotto al carrello l'utente deve spostare il cursore sopra il prodotto desiderato, selezionarlo con un click singolo e premere Add to Cart in basso a destra nel pannello del prodotto.

Il tasto *Reload products* serve per aggiornare le quantità visualizzate con le effettive quantità presenti nel database. Se vengono individuati prodotti nel carrello che nel frattempo sono stati comprati, e quindi non più disponibili, essi vengono tolti e un avviso con dettagli viene mostrato all'utente. Lo stesso tipo di controllo viene effettuato al momento del pagamento, vedi sezione Payment.fxml.

4.4 ShoppingCart.fxml

Dalla vista *Shop.fxml* il Customer può visualizzare in dettaglio i prodotti che ha messo nel carrello cliccando sul tasto *Open Cart*, questa interfaccia permette di modificare la quantità dei prodotti, rimuoverne uno o più e filtrarli secondo i parametri forniti nella parte alta della vista.

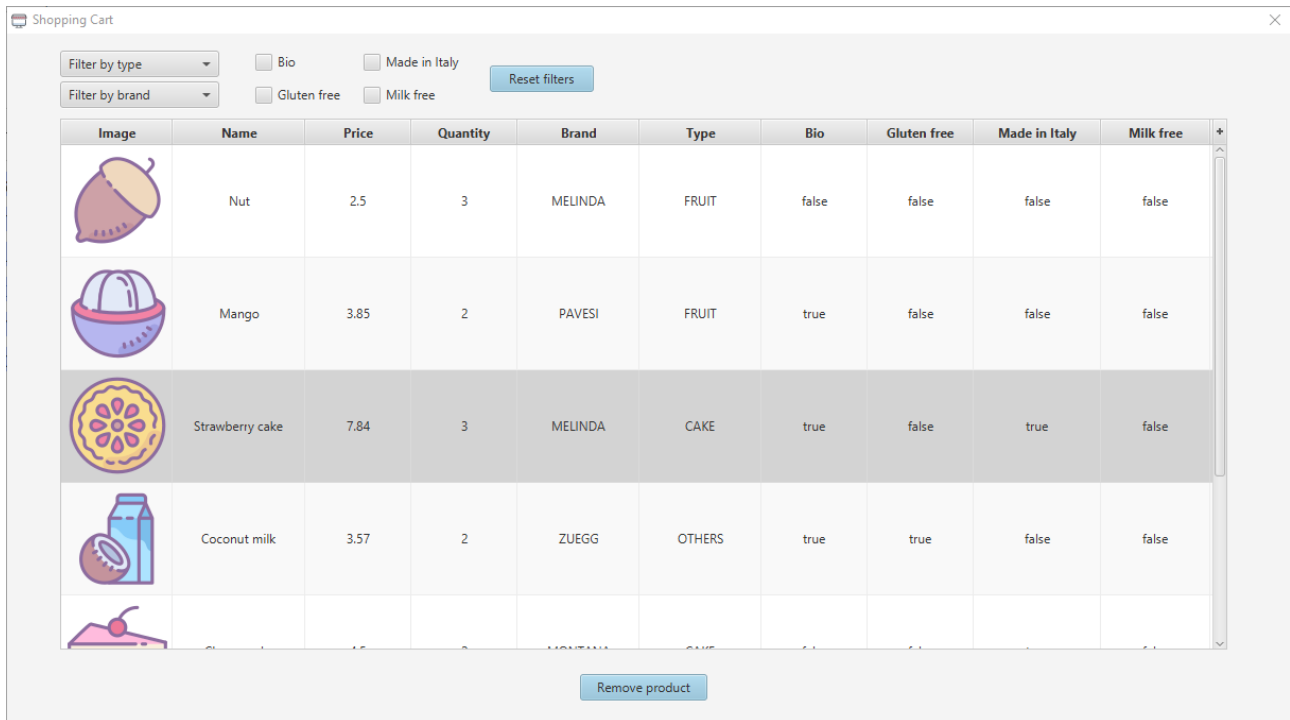


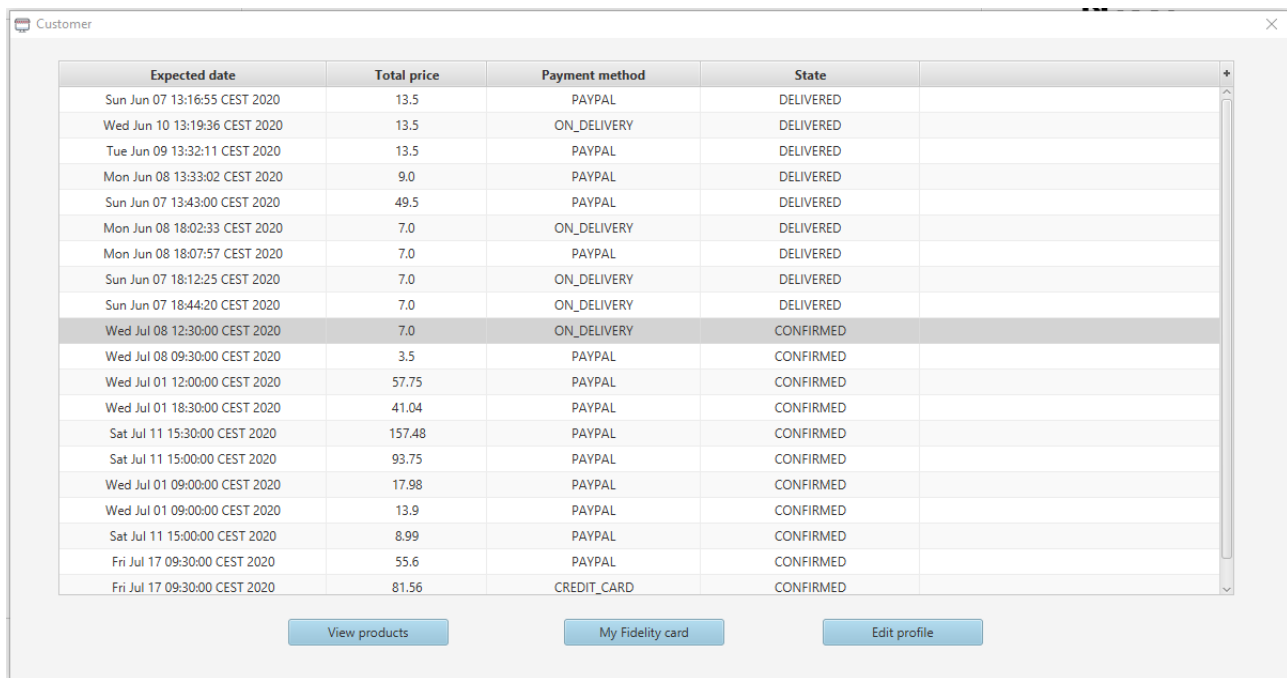
Figura 5: Vista ShoppingCart.fxml

4.5 Customer.fxml

Tramite questa vista il Customer può visualizzare la lista di spese effettuate, i prodotti associati a tali spese, e accedere alle viste *FidelityCard.fxml* per visualizzare la propria FidelityCard (se ne esiste una associata al Customer) e *EditProfile.fxml* per modificare le informazioni dell'utente (email, password, etc).

Se in fase di registrazione il customer non ha sottoscritto la carta fedeltà, il sistema disabilita il bottone corrispondente.

È possibile sottoscrivere la carta fedeltà anche in un momento successivo dalla vista *EditProfile.fxml*



Expected date	Total price	Payment method	State
Sun Jun 07 13:16:55 CEST 2020	13.5	PAYPAL	DELIVERED
Wed Jun 10 13:19:36 CEST 2020	13.5	ON_DELIVERY	DELIVERED
Tue Jun 09 13:32:11 CEST 2020	13.5	PAYPAL	DELIVERED
Mon Jun 08 13:33:02 CEST 2020	9.0	PAYPAL	DELIVERED
Sun Jun 07 13:43:00 CEST 2020	49.5	PAYPAL	DELIVERED
Mon Jun 08 18:02:33 CEST 2020	7.0	ON_DELIVERY	DELIVERED
Mon Jun 08 18:07:57 CEST 2020	7.0	PAYPAL	DELIVERED
Sun Jun 07 18:12:25 CEST 2020	7.0	ON_DELIVERY	DELIVERED
Sun Jun 07 18:44:20 CEST 2020	7.0	ON_DELIVERY	DELIVERED
Wed Jul 08 12:30:00 CEST 2020	7.0	ON_DELIVERY	CONFIRMED
Wed Jul 08 09:30:00 CEST 2020	3.5	PAYPAL	CONFIRMED
Wed Jul 01 12:00:00 CEST 2020	57.75	PAYPAL	CONFIRMED
Wed Jul 01 18:30:00 CEST 2020	41.04	PAYPAL	CONFIRMED
Sat Jul 11 15:30:00 CEST 2020	157.48	PAYPAL	CONFIRMED
Sat Jul 11 15:00:00 CEST 2020	93.75	PAYPAL	CONFIRMED
Wed Jul 01 09:00:00 CEST 2020	17.98	PAYPAL	CONFIRMED
Wed Jul 01 09:00:00 CEST 2020	13.9	PAYPAL	CONFIRMED
Sat Jul 11 15:00:00 CEST 2020	8.99	PAYPAL	CONFIRMED
Fri Jul 17 09:30:00 CEST 2020	55.6	PAYPAL	CONFIRMED
Fri Jul 17 09:30:00 CEST 2020	81.56	CREDIT_CARD	CONFIRMED

View products My Fidelity card Edit profile

Figura 6: Vista Customer.fxml

4.6 Payment.fxml

Questa vista permette ai Customer di selezionare una data per la consegna e modalità di pagamento. Per semplicità, le date generate sono random a 1-3 giorni di distanza dalla data di pagamento.

Alla pressione del tasto "Pay" viene inizializzata la sequenza di pagamento:

- Viene controllata la disponibilità dei prodotti del carrello (nel mentre un'altro utente avrebbe potuto esaurire le scorte che a inizio spesa erano disponibili) e i prodotti non più disponibili vengono tolti dal carrello.
- Vengono comprati i prodotti (aggiornamento products.txt).

Al termine della sequenza di pagamento viene visualizzato un messaggio per informare l'utente che la transazione è andata a buon fine, il record della spesa sarà ora disponibile nel file shoppingCarts.txt, visualizzabile dal relativo Customers e i vari Employee.

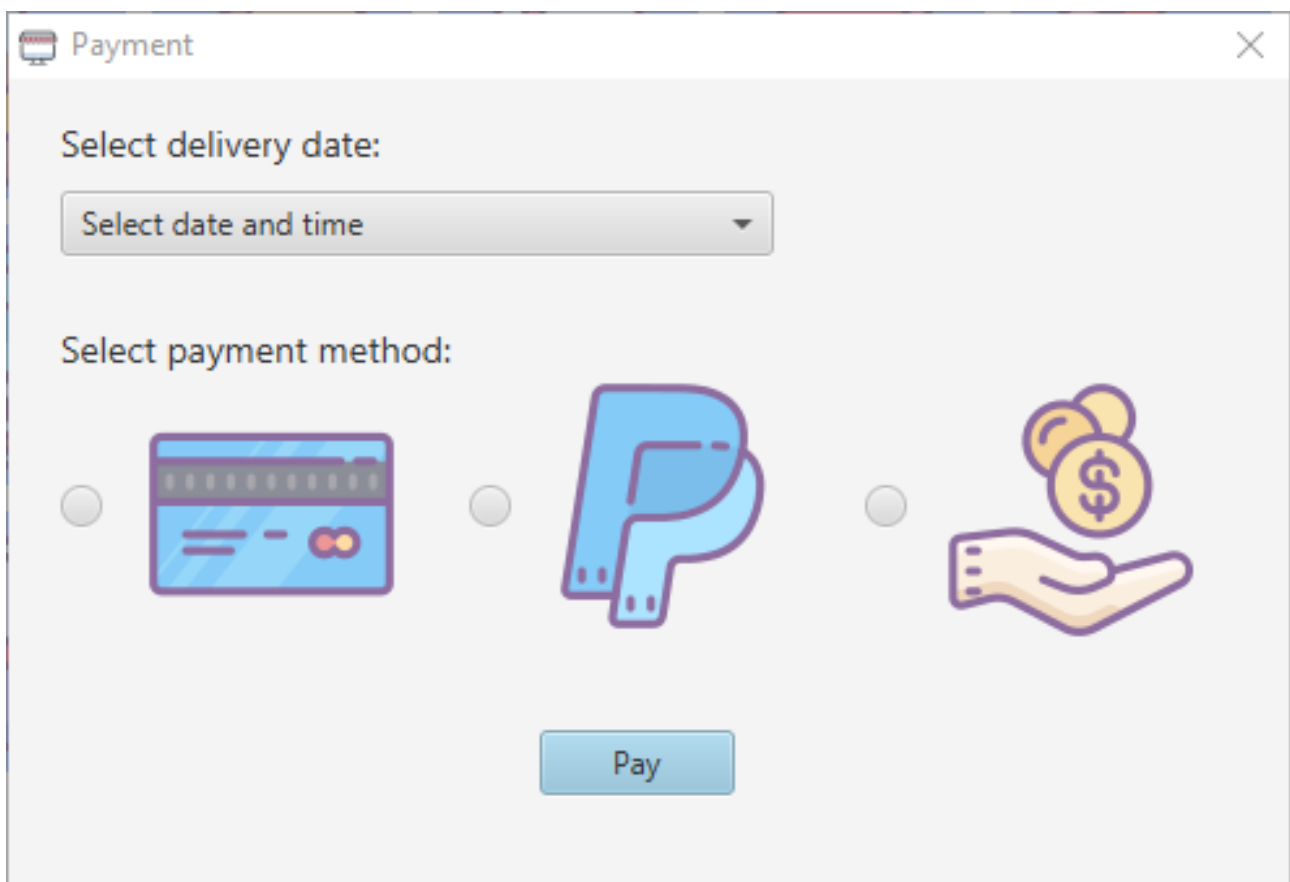


Figura 7: Vista Payment.fxml

4.7 Employee.fxml

Questa vista fornisce agli Employees la possibilità di:

- Visualizzare e modificare le informazioni riguardo i prodotti dello shop
- Aggiungere un nuovo prodotto
- Visualizzare il database delle spese dei Customer

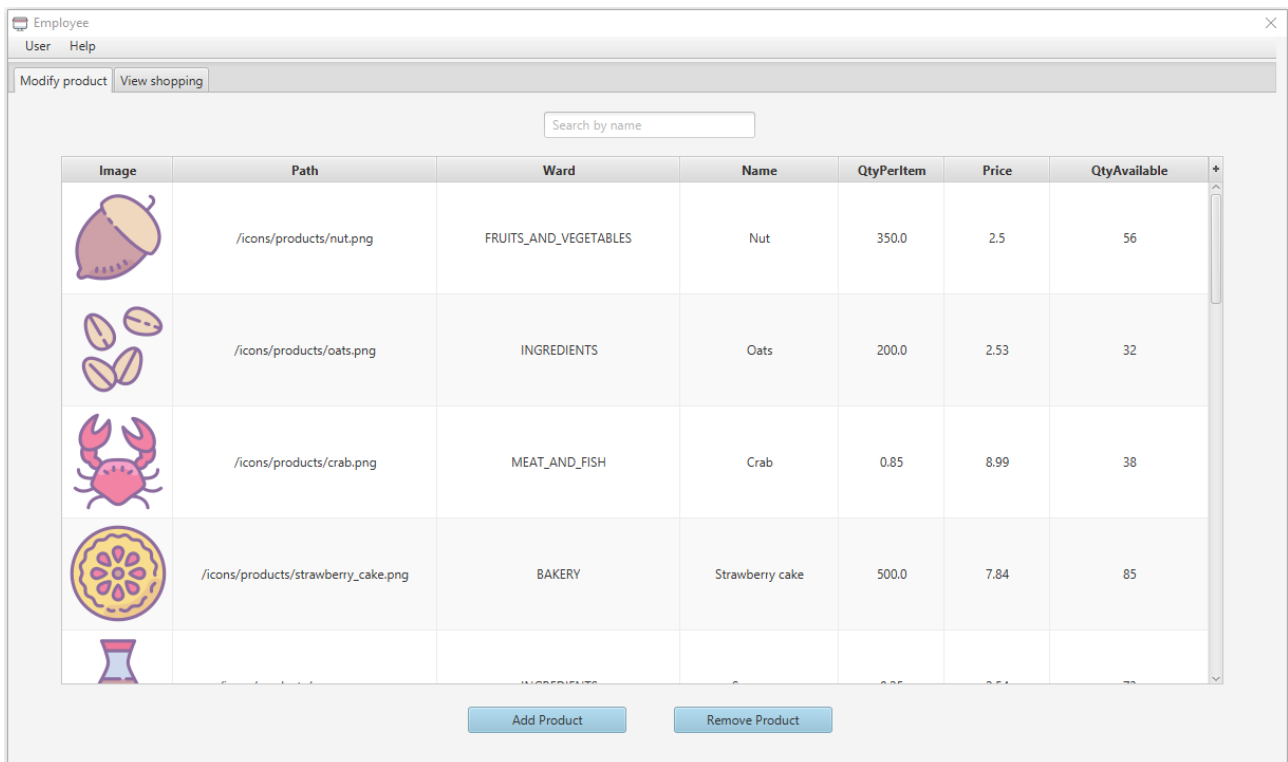


Figura 8: Vista Employee.fxml

Si assume che quando l'employee modifichi il database dei prodotti nessun Customer possa fare la spesa, in quanto ciò creerebbe incongruenze nel database.

5 Casi d'uso

Di seguito vengono illustrati i casi d'uso del Customer sia in forma di Diagramma che in forma testuale.

5.1 Casi d'uso relativi al Customer

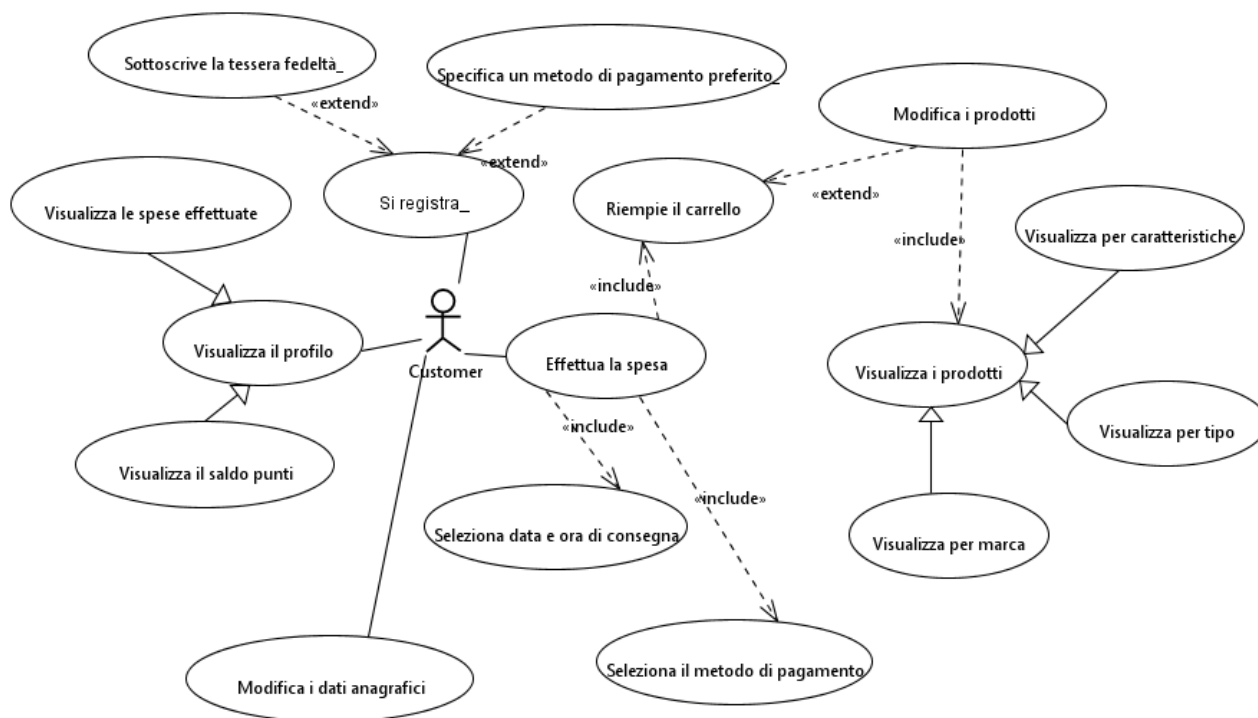


Figura 9: Use case del Customer

Visualizza profilo

Precondizioni: /

1. Il customer deve essere autenticato

Sequenza degli eventi:

1. Il caso d'uso inizia quando il customer apre la finestra relativa all'account
2. Se il customer seleziona una spesa effettuata e preme il bottone 'View products'
- 2.1. Il sistema apre una nuova finestra visualizzando i prodotti presenti in quella spesa
3. Se il customer preme il bottone 'Fidelity card'
- 3.1. Il sistema apre una nuova finestra visualizzando i dettagli della carta fedeltà

Postcondizioni: /

Registrazione

Precondizioni: /

Sequenza degli eventi:

1. Il caso d'uso inizia quando il customer apre la finestra di registrazione
2. Il customer inserisce le informazioni richieste dal form di registrazione
3. Se il customer sottoscrive la carta fedeltà
- 3.1 Il sistema associa una nuova carta fedeltà al customer
4. Se il customer seleziona un metodo di pagamento preferito
- 4.1 Il sistema associa il metodo selezionato al customer
5. Il caso d'uso finisce quando il customer preme il bottone 'Sign up'

Postcondizioni: /

Modifica i prodotti nel carrello

Precondizioni: Il customer deve aver inserito qualche prodotto nel carrello

Sequenza degli eventi:

1. Il caso d'uso inizia quando il customer preme il bottone 'Open cart'
2. Il customer viene introdotto alla finestra del carrello
3. Se il customer seleziona uno o più filtri presenti
- 3.1 Il sistema visualizza solo i prodotti che corrispondono ai filtri selezionati
4. Se il customer seleziona un prodotto e digita una nuova quantità
- 4.1 Il sistema aggiorna la quantità del prodotto presente nel carrello
5. Se il customer seleziona un prodotto e preme il bottone 'Remove Product'
- 5.1 Il sistema rimuove il prodotto selezionato dal carrello
6. Il caso d'uso finisce quando il customer chiude la finestra

Postcondizioni: Il contenuto del carrello viene aggiornato

Riempie il carrello

Precondizioni: Il customer deve essere autenticato

Sequenza degli eventi:

1. Il caso d'uso inizia quando il customer logga nel sistema
2. Il customer viene introdotto all'interfaccia principale
3. Il customer seleziona i prodotti che desidera acquistare e li aggiunge al carrello

Postcondizioni: Il sistema riempie il carrello con i prodotti selezionati

Effettuare un'ordine

Precondizioni: Il customer deve aver aggiunto dei prodotti nel carrello

Sequenza degli eventi:

1. Il caso d'uso inizia quando il customer preme il bottone 'Go to Payment'
2. Il customer viene introdotto all'interfaccia per il pagamento
3. Il customer seleziona la data e ora di consegna
4. Il customer seleziona una modalità di pagamento
5. Il customer finalizza l'acquisto premendo il bottone 'Pay'

Postcondizioni: Il sistema registra l'ordine aggiornando i database di prodotti, utenti e spese

5.2 Casi d'uso relativi all'Employee

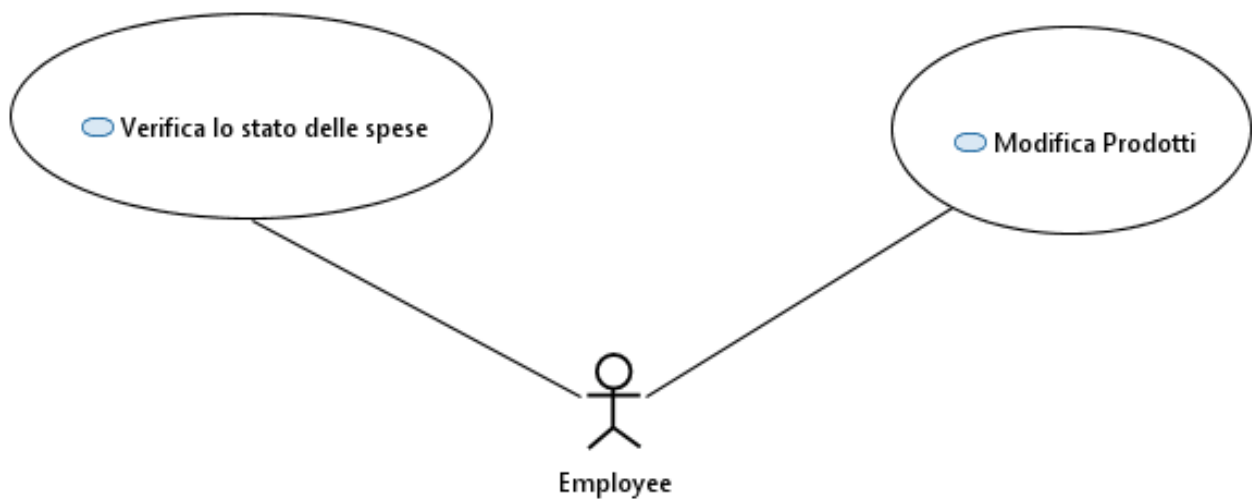


Figura 10: Use case dell'Employee

Modifica prodotti

Precondizioni: L' employee deve essere autenticato

Sequenza degli eventi:

1. Se l' employee preme il bottone "Add Product"
2. Se il customer seleziona uno o più filtri presenti
 - 2.1 Il sistema visualizza solo i prodotti che corrispondono ai filtri selezionati
3. Se l' employee digita una nuova quantità
 - 3.1 Il sistema aggiorna la quantità del prodotto presente nel carrello
4. Se l' Employee seleziona un prodotto e preme il bottone "Remove"

6 Sequence diagram per i principali Use Case

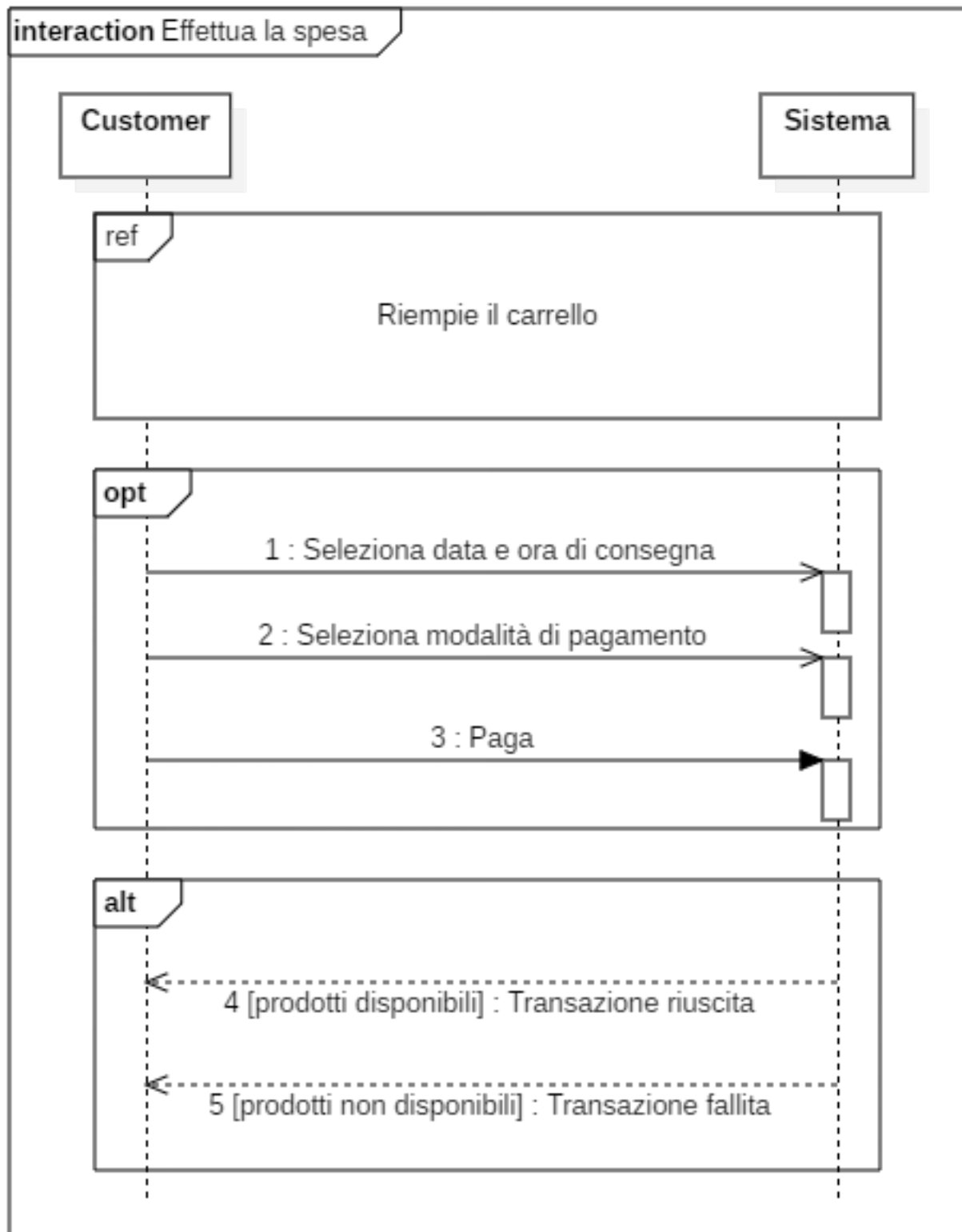


Figura 11: Sequence Diagram delle attività di spesa del Customer

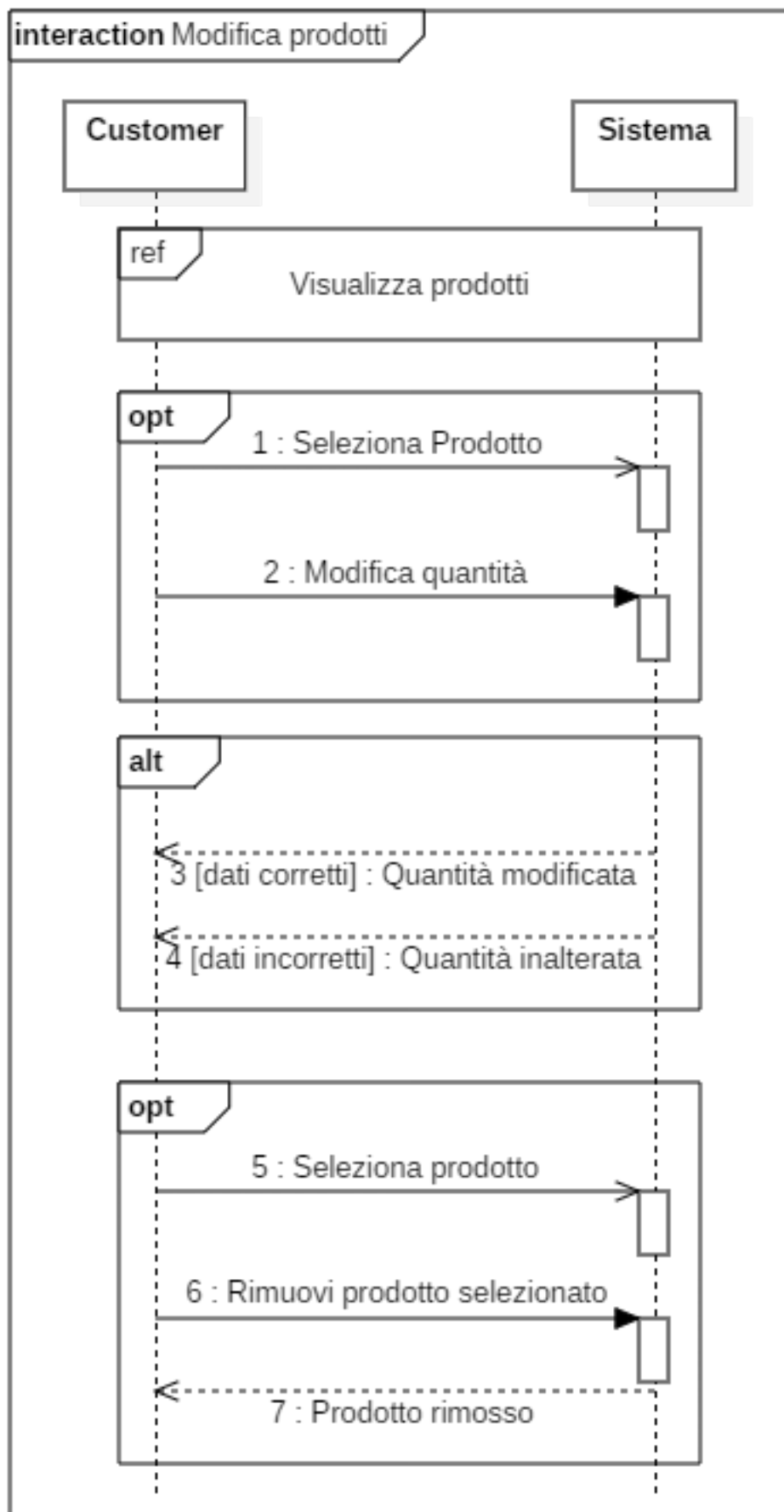
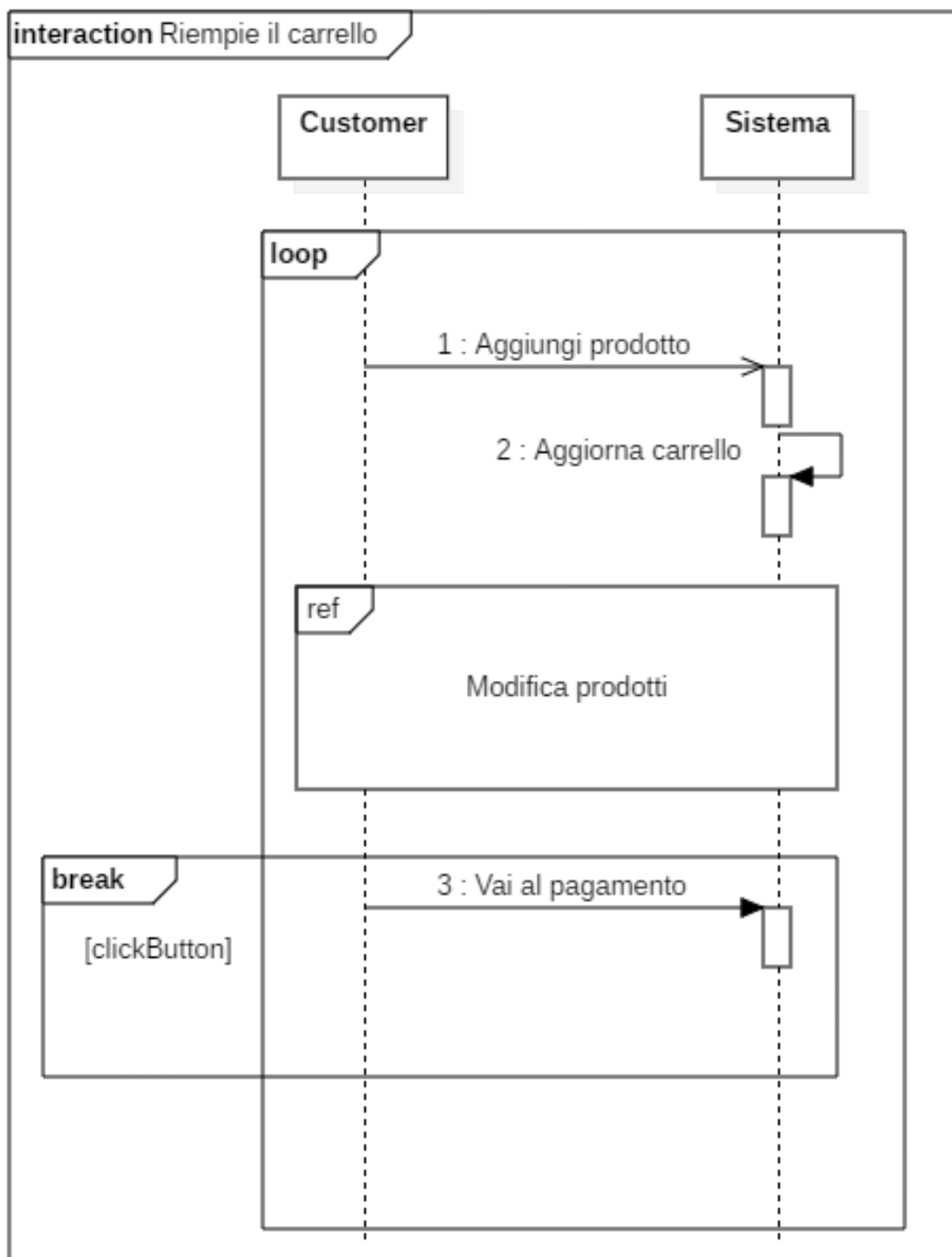
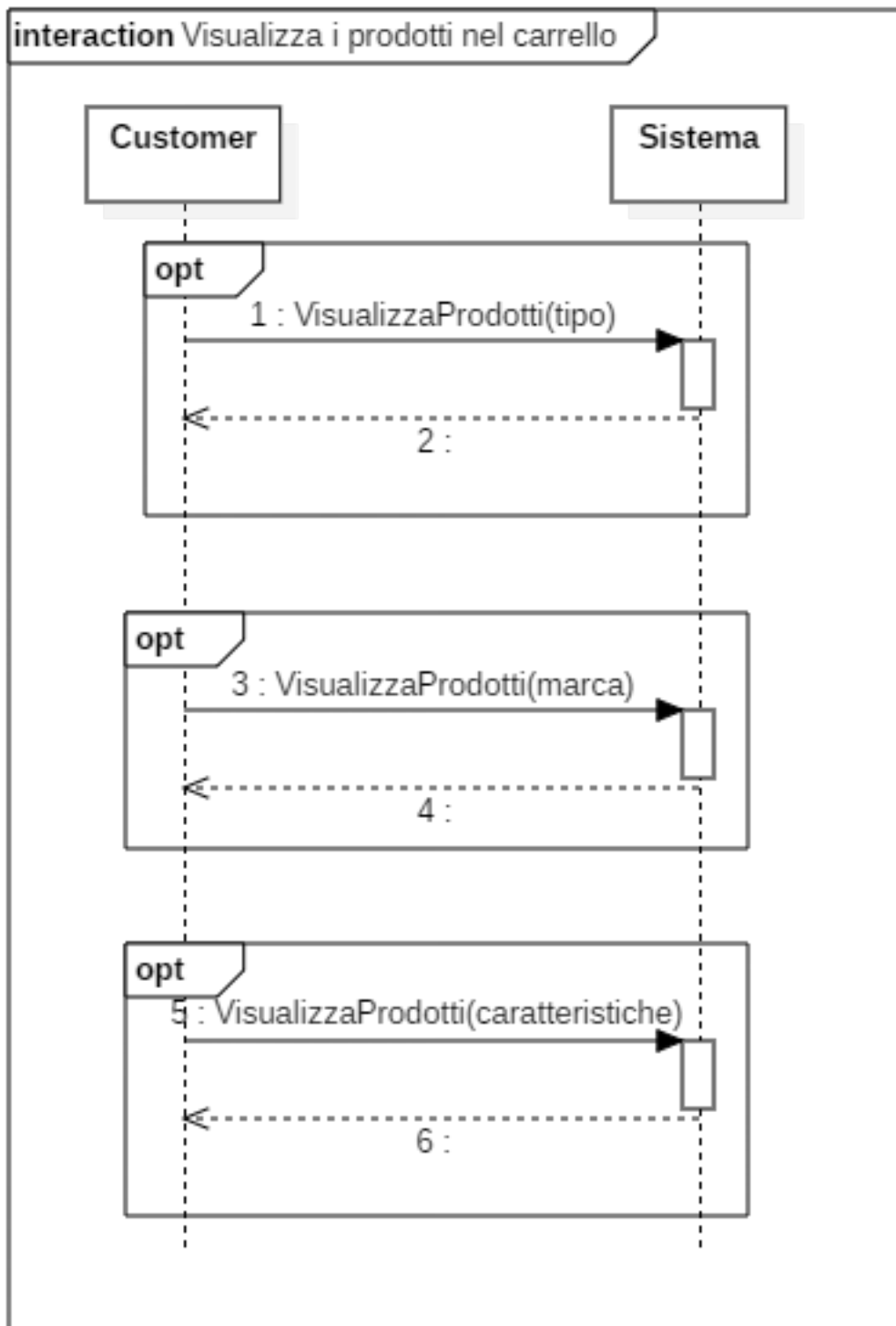


Figura 12: Sequence Diagram della modifica dei prodotti del carrello del Customer





7 Application Diagrams

7.1 Class Diagram

Di seguito vengono riportati i Class Diagram di maggior rilevanza

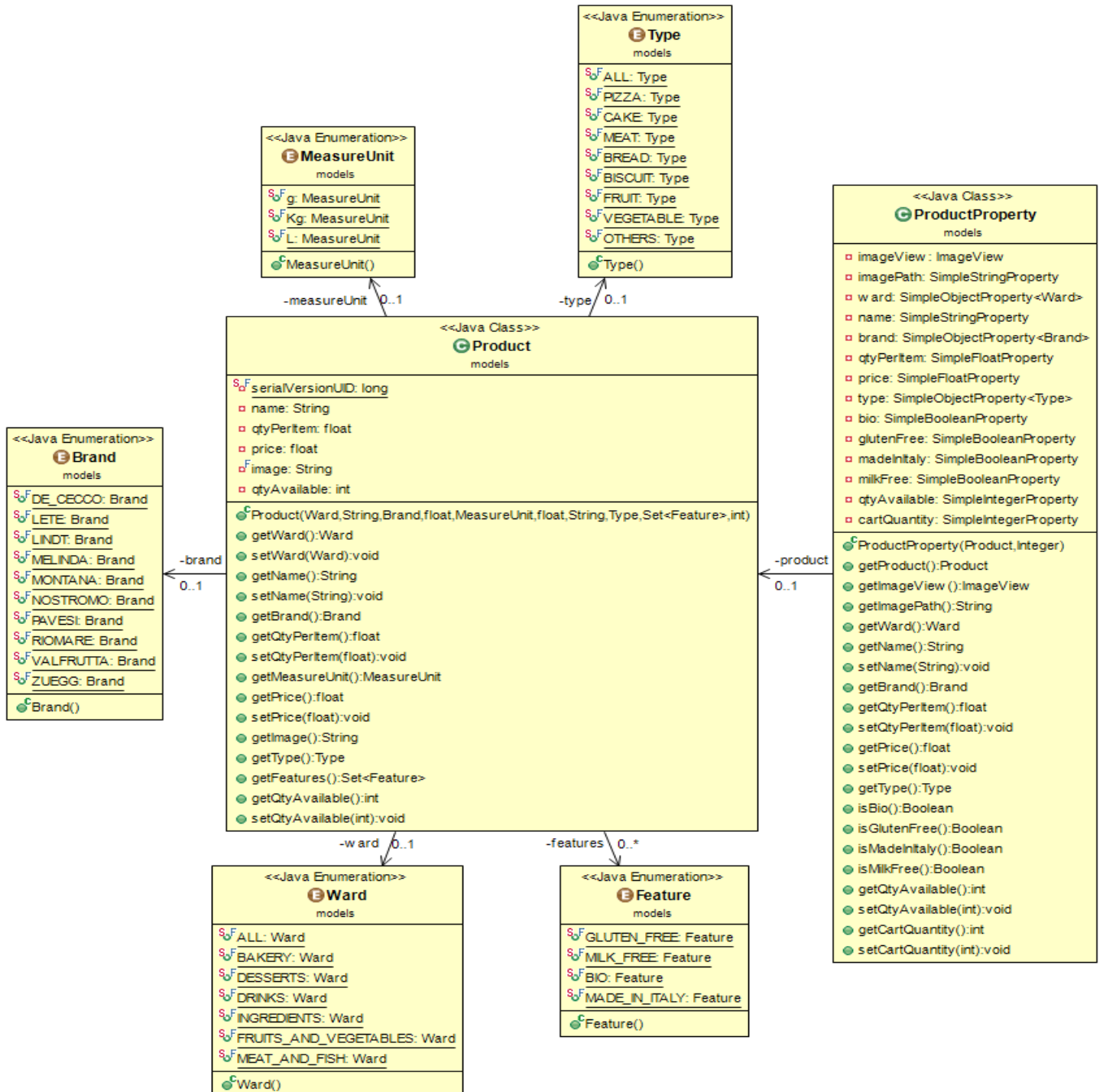


Figura 13: Class diagram della logica dei prodotti

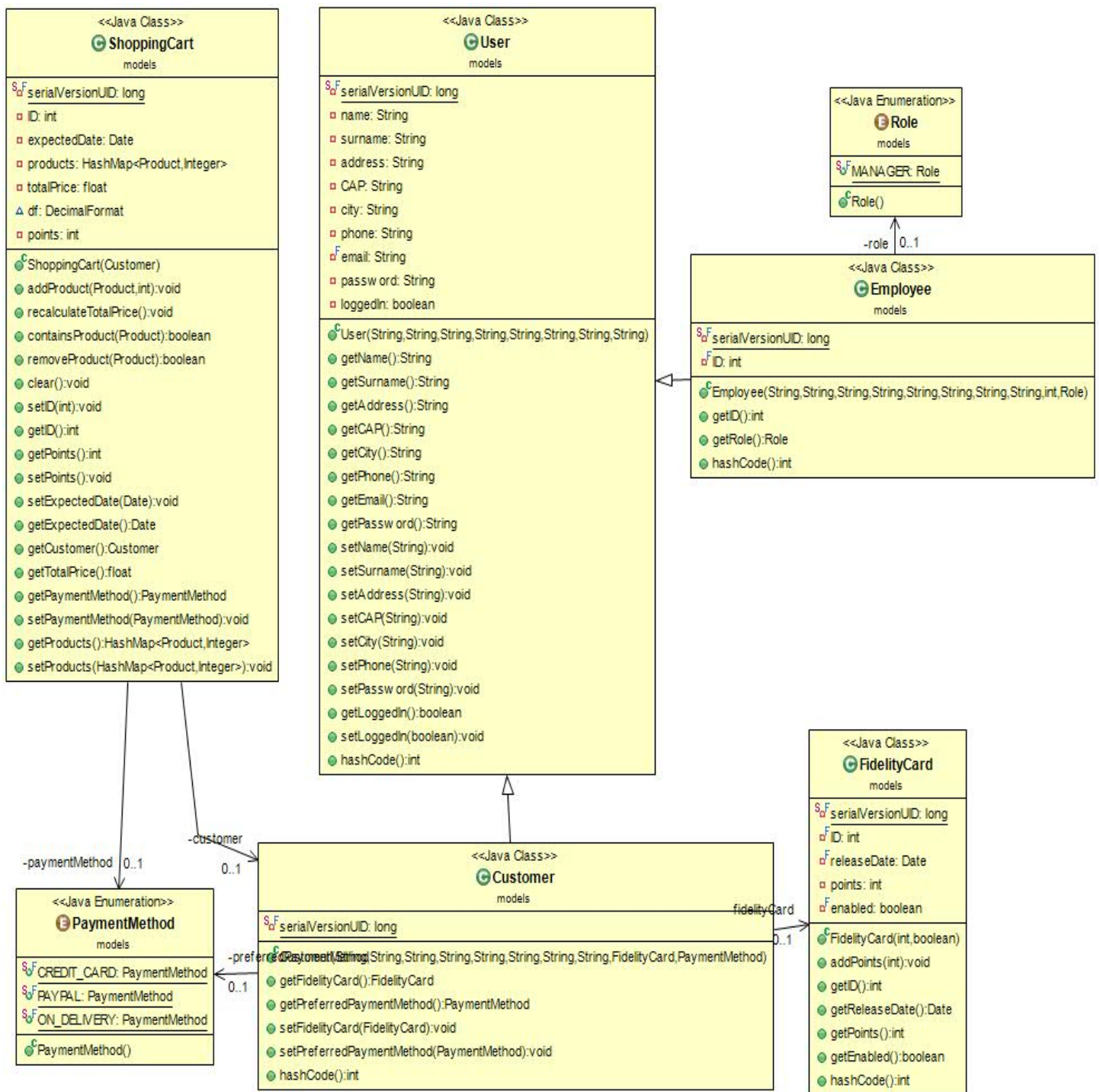


Figura 14: Class diagram della logica delle utenze