

Relazione Progetto E-Shop

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

10 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	Note utili	5
4	Viste principali e relativi Use Case	6
4.1	Login.fxml	6
4.2	Registration.fxml	6
4.3	Shop.fxml	6
4.4	Employee.fxml	6
4.5	ShoppingCart.fxml	6
4.6	Payment.fxml	6
4.7	Customer.fxml	7
5	Sequence diagram per i principali Use Case	8
6	Application Diagrams	8
6.1	Activity Diagram	8
6.2	Class Diagram	8
6.3	Sequence diagram	9

1 Introduzione

Viene richiesta l'implementazione di un sistema informatico per gestire il servizio di spesa online 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 del software, design patter, best practice, documentazione del software.

2 Sviluppo e attività di test

L'applicazione è stata sviluppata con metodologia Il codice è stato scritto sia in modo individuale che durante sessioni di pair-programming, tali sessioni si sono svolte ogni circa due giorni durante un lasso di tempo di circa tre settimane. Ogni sessione di pair programming è stato così strutturata:

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

La durata media di una sessione era di due ore.

Di seguito viene riportato il diagramma di flusso dell'attività di sviluppo/test dell'applicazione.

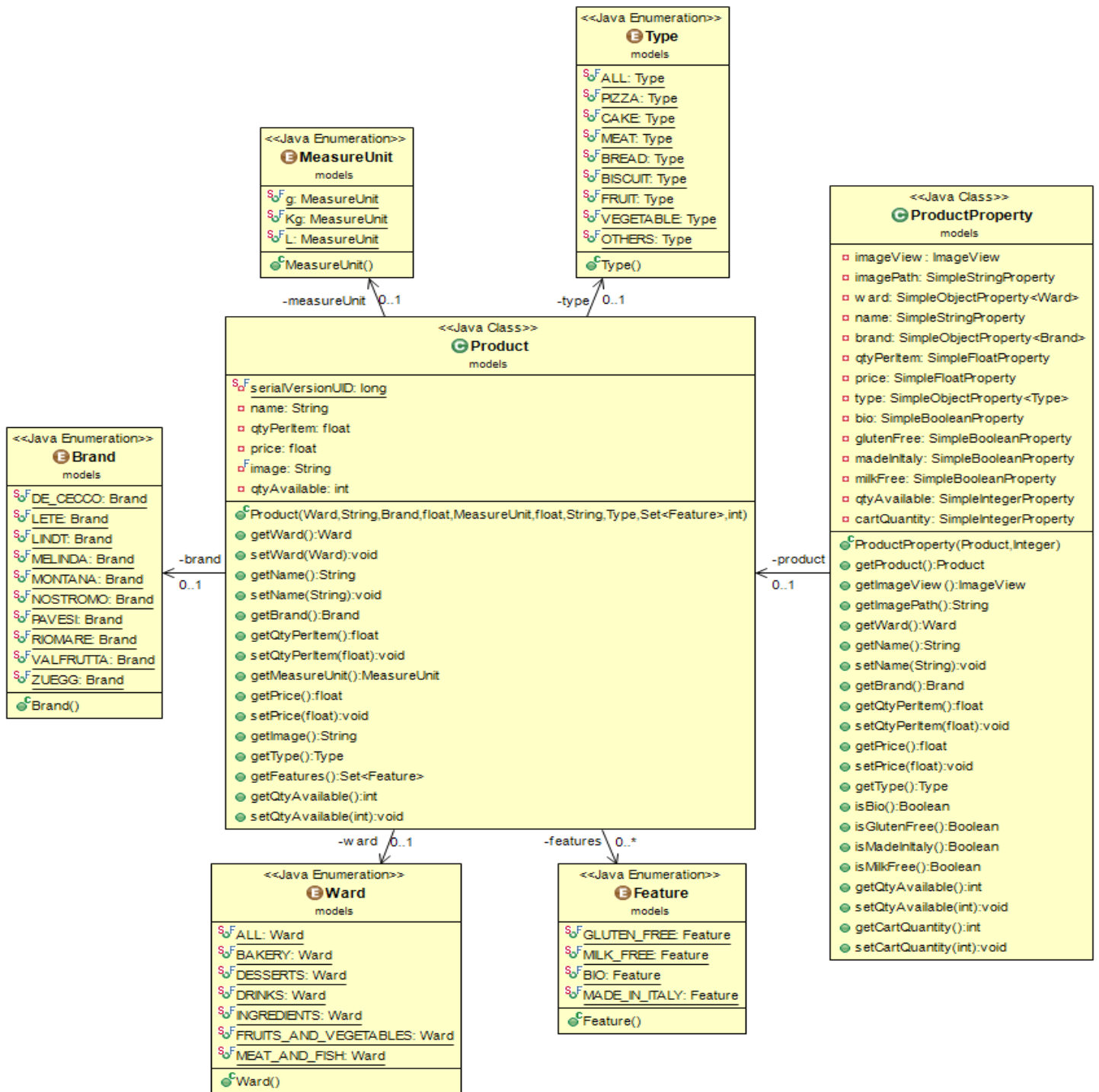


Figura 1: Class diagram di

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 Java con l'ausilio di JavaFx per l'implementazione della parte grafica. La scelta è ricaduta su tali tecnologie perchè i requisiti del progetto erano realizzabili con esse.

3.2 Database dei dati

I vari database dell'applicazione, ovvero la lista degli utenti, dei prodotti prodotti e delle spese effettuate, sono realizzati tramite file .txt sui quali vengono scritte le strutture dati utilizzate nel codice, nel nostro caso HashMap contenenti oggetti di tipo Product, User e ShoppingCart. Ciò è possibile perchè tali strutture e oggetti implementano l'interfaccia Serializable. Abbiamo scelto questo approccio perchè, anche se entrambi abbiamo avuto 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.

Nel nostro caso, la parte Model è costituita dalle classi Models, ovvero le classi che descrivono gli oggetti che vengono utilizzati nell'applicazione, come ShoppingCart.java, le varie enumerazioni Ward.java, PaymentMethod.java etc

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.

3.4 Tipi di utenza

Gli utenti dell'applicazione possono loggarsi come Customer o come 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,...

Istanze multiple dell'applicazione non costituiscono un problema per l'integrità dei database in quanto prima. È possibile aprire istanza multiple dell'applicazione in quanto all'interno del file users.txt viene tenuta traccia di quali utenti siano loggati, ciò è realizzato tramite una flag all'interno della classe User, la quale se posta a true segnala all'applicazione quando tenta di loggare un utente, se quell'utente è già loggato o meno. Ogni volta che un utente fa log in-out viene aggiornato il database users.txt.

3.5 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 immagini dei prodotti sono state prese da un sito internet etc etc...

4 Viste principali e relativi Use Case

Di seguito vengono elencate le principali viste utilizzate e i possibili use case per ognuna di esse.

4.1 Login.fxml

Questa vista rappresenta un'interfaccia per il login comune a Customers ed Employees. Un utente non registrato può accedere alla vista Registration.fxml tramite l'apposito bottone.

4.2 Registration.fxml

Tramite questa vista un utente Customer può registrarsi sulla piattaforma eShop. Non è possibile creare utenze di tipo Employee tramite questa schermata, tali utenze si assume che vengano inserite tramite codice dai programmatori dell'applicazione.

4.3 Shop.fxml

Questa vista rappresenta l'interfaccia grafica per l'utente di tipo Customer. METTERE FOTO QUA

4.4 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

4.5 ShoppingCart.fxml

Questa vista permette ai Customer di visualizzare i prodotti presenti nel proprio carrello in maniera dettagliata, modificarne la quantità, rimuoverne uno o più e visualizzarli in ordine per TODO

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.
- sdad
- Vengono comprati i prodotti, ovvero vengono aggiornate le quantità disponibili nel file 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 da Customers ed Employee.

4.7 Customer.fxml

Questa vista permette ai Customers di visualizzare il proprio record di spese, modificare i propri dati anagrafici, le spese effettuate e visualizzare la propria carta di fedeltà (se associata al Customer durante la registrazione dell'utenza).

5 Sequence diagram per i principali Use Case

6 Application Diagrams

6.1 Activity Diagram

6.2 Class Diagram

Di seguito vengono riportati i Class Diagram dell'applicazione

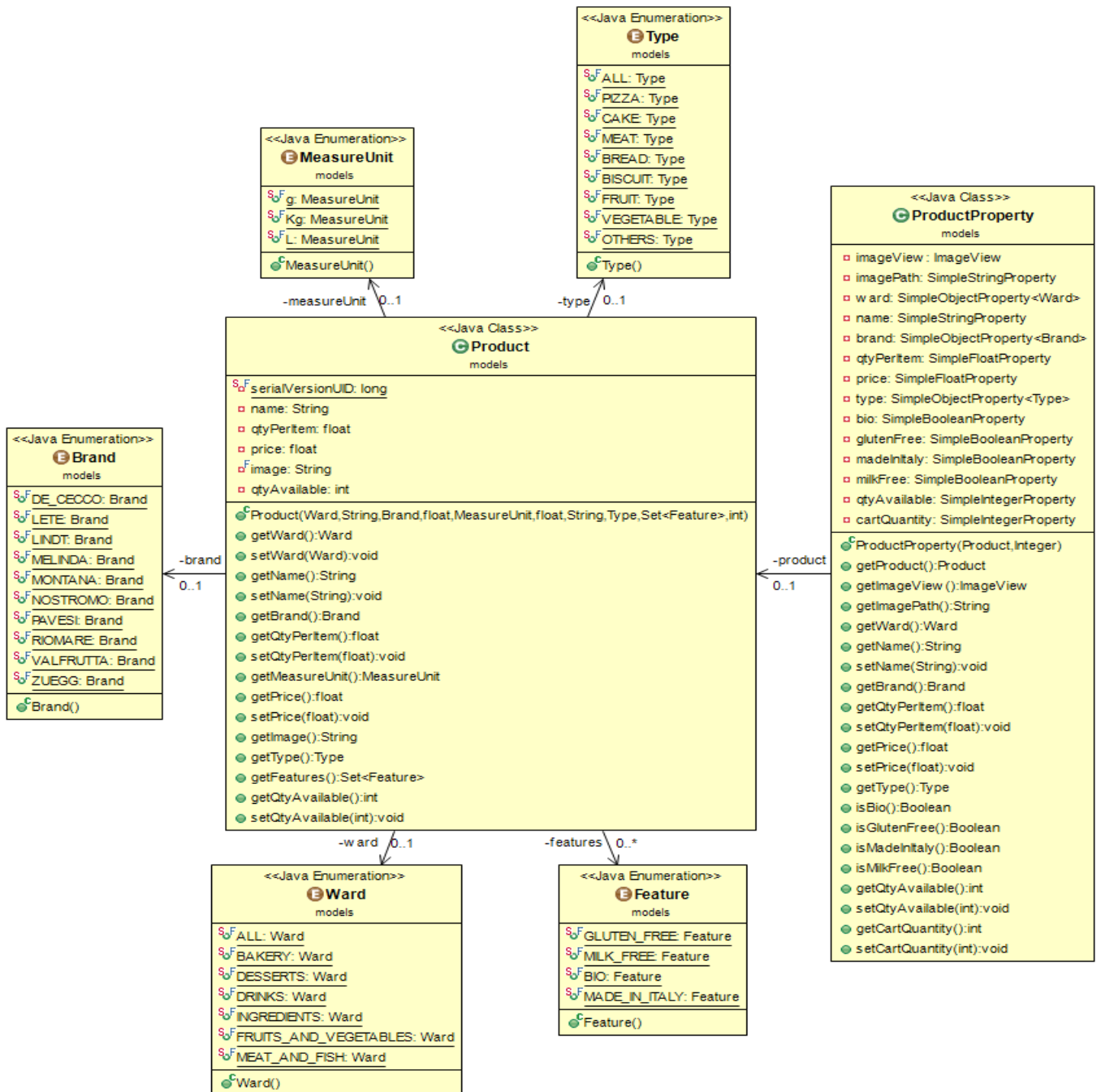


Figura 2: Class diagram di

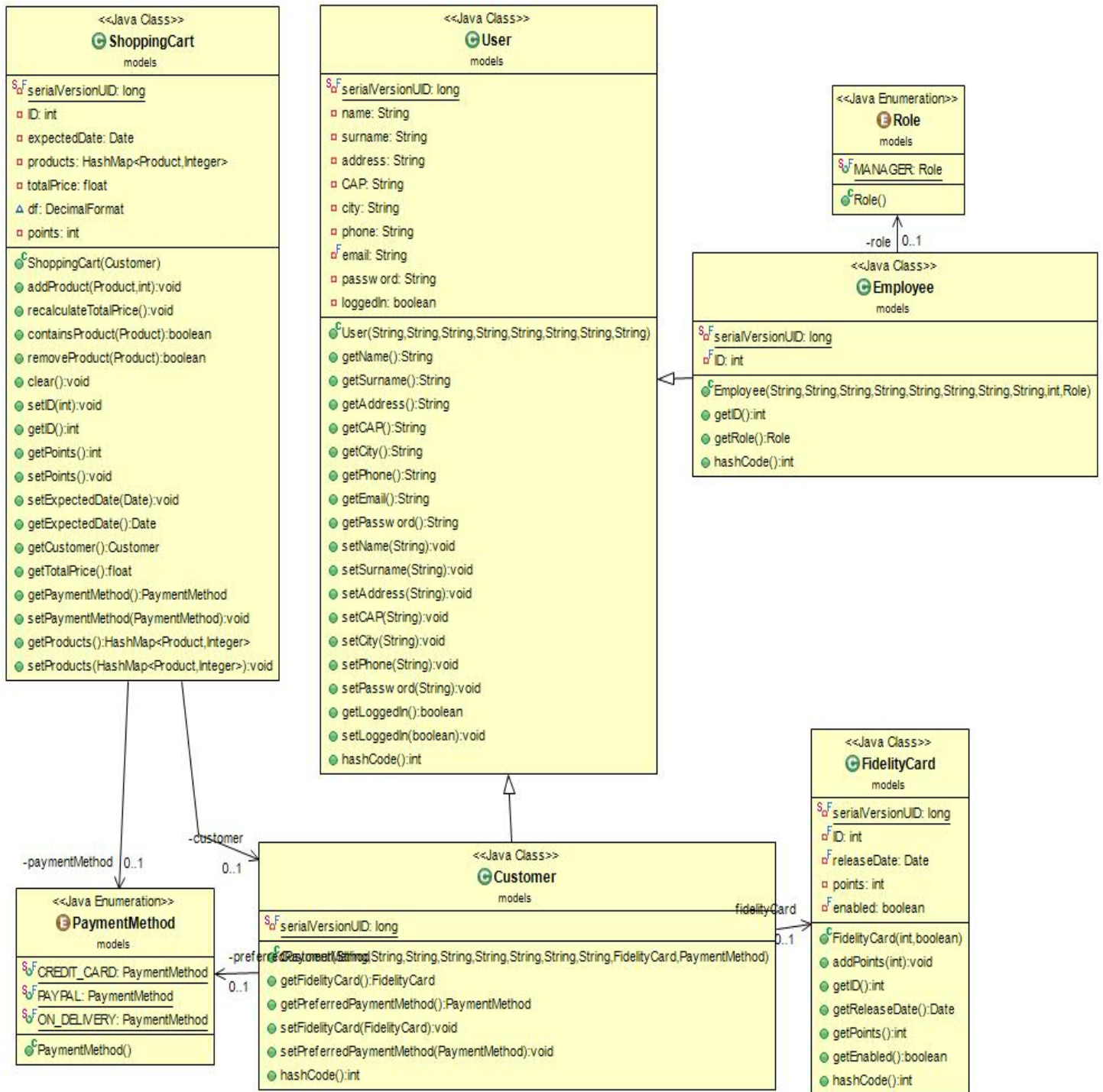


Figura 3: Class diagram delle utenze

6.3 Sequence diagram