UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA LUIGI VANVITELLI

Ingegneria del Software

Trading Card Selling System

Giovanni D'Ambrosio
A13002622

Indice

1 Introduzione

Si vuole realizzare un sistema di vendita di carte da gioco che permette l'interfacciamento tra compratori e venditori. Il sistema permetterà a entrambi di fare acquisti, aggiungendo al proprio carrello le carte desiderate. I venditori potranno mettere in vendita i prodotti desiderati per renderli visibili agli altri utenti.
Per realizzare il sistema di vendita, è stato progettato un programma software in esecuzione sull'elaboratore dell'utente, che si metterà un contatto con un server per avere accesso al database delle informazioni personali e degli altri utenti. Per realizzare questa architettura, è stato scelto di utilizzare i pattern architetturali *Client Server* e *Model-View-Controller*, che saranno analizzati successivamente.
Questo documento è il risultato della fase di analisi e design elaborate secondo un modello di processo lineare a cascata.

2 Attori e Requisiti

2.1. Attori

- User
- Seller
- Tempo
- Gestore Pagamenti

2.2. Requisiti funzionali

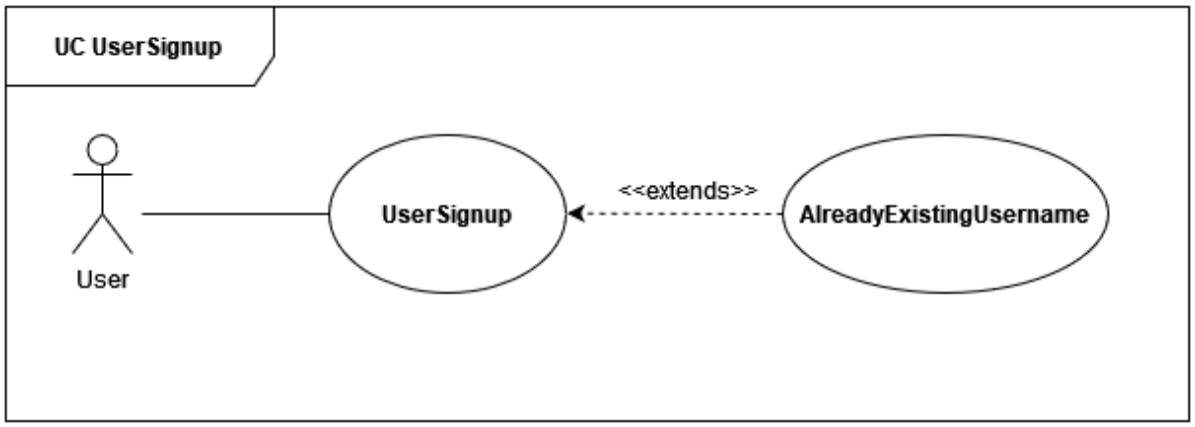
- Il sistema consentirà all'utente di registrarsi come compratore o venditore.
- Il sistema consentirà all'utente l'accesso alla sua area personale e alla piattaforma di ricerca carte ed espansioni.
- Il sistema consentirà all'utente di cercare per nome una carta da gioco o un'espansione di carte e di visualizzare i venditori che ne mettono a disposizione una o più copie.
- Il sistema consentirà all'utente di filtrare le copie di una carta da gioco messa in vendita per rarità e/o costo.
- Il sistema consentirà all'utente di filtrare le copie di un'espansione messa in vendita per costo.
- L'utente potrà aggiungere una o più copie di una carta da gioco o di un'espansione al carrello.

- L’utente potrà visualizzarne il contenuto del proprio carrello e modificare le quantità di ciascun oggetto.
- L’utente potrà visualizzare lo storico ordini o i dettagli dell’ordine appena inoltrato.
- L’utente potrà accedere al carrello per avviare la procedura di acquisto e scegliere il metodo di pagamento preferito.
- Il sistema permetterà al venditore di mettere in vendita una o più copie di una carta da gioco o espansione dalla sua area personale.
- Il venditore potrà visualizzare gli ordini ricevuti dalla sua area personale.
- Il sistema permetterà al venditore di rimuovere dalla vendita una o più delle carte da gioco o espansioni da lui pubblicate dalla sua area personale.
- Il venditore potrà modificare il proprio conto IBAN o numero di carta per la ricezione dei pagamenti dalla sua area personale.

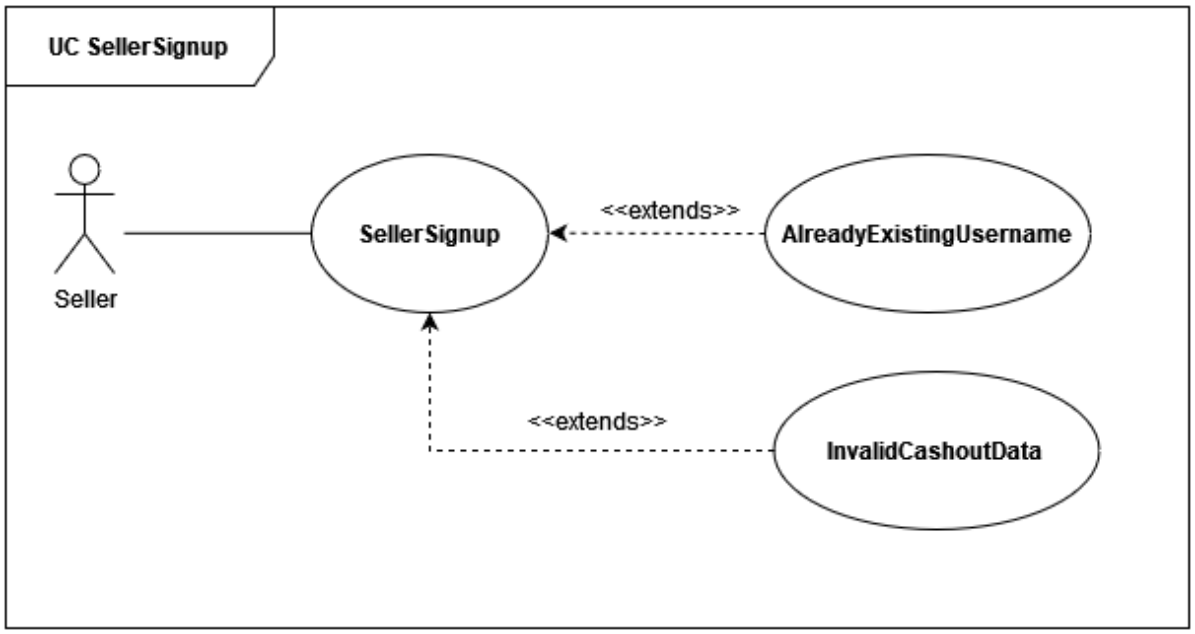
2.3. Requisiti non funzionali

- Il sistema utilizzerà una funzione di hash per crittografare la password dell’utente.
- Il sistema comunicherà con il server attraverso esclusivamente connessioni di tipo HTTPS.
- L’interfaccia grafica sarà responsiva e fluida all’utilizzo.
- Il sistema non sprecherà risorse attraverso una gestione efficiente dei dati.
- Il sistema bloccherà eventuali richieste di interazione con il server in rapida successione.

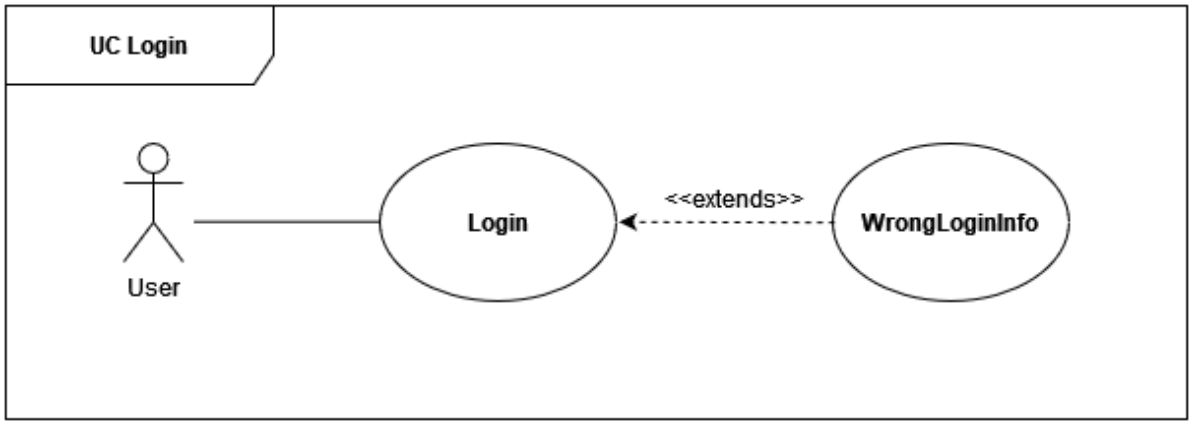
3 Diagrammi dei casi d'uso



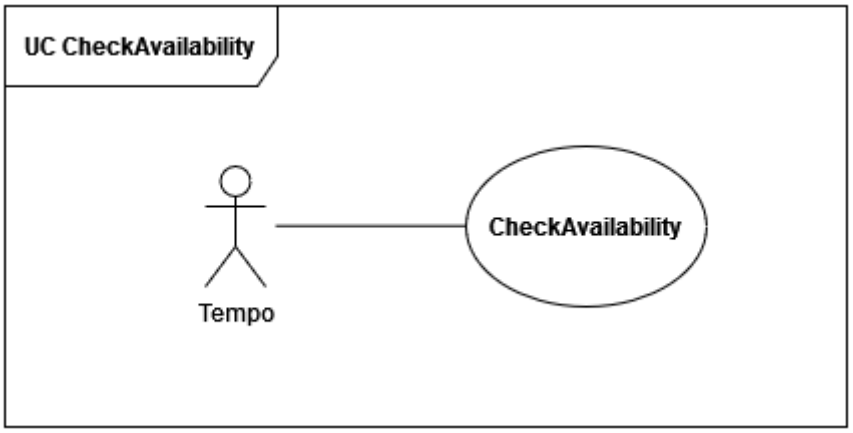
Caso d'uso 0: UserSignup



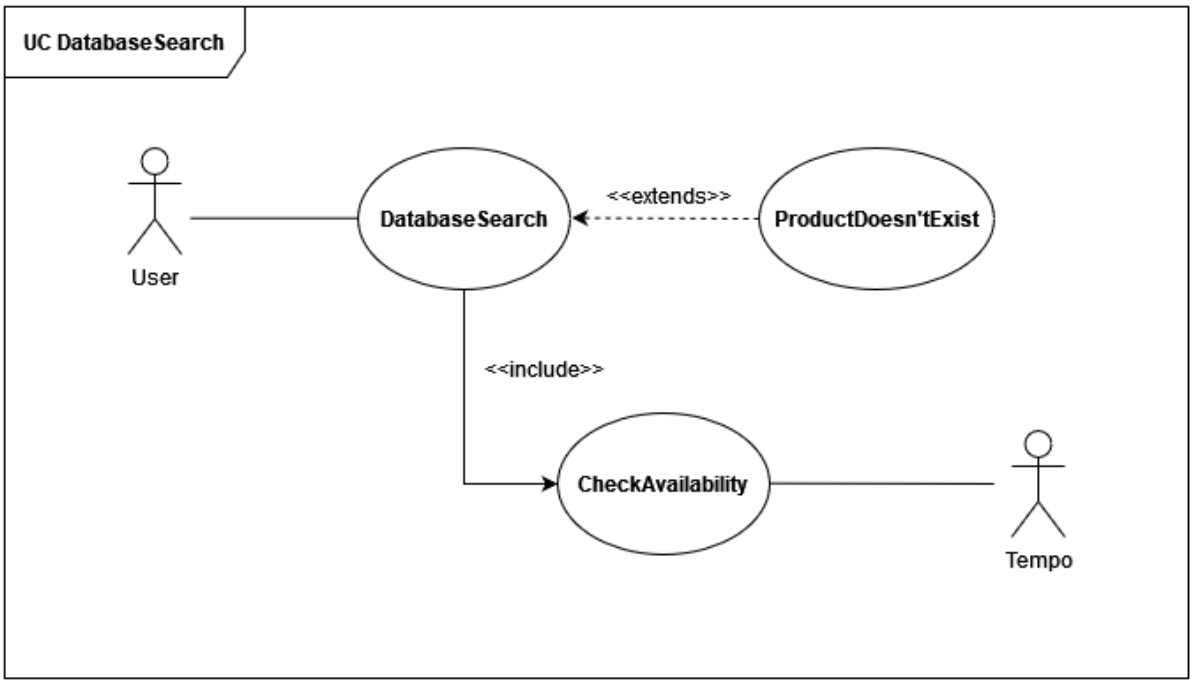
Caso d'uso 1: SellerSignup



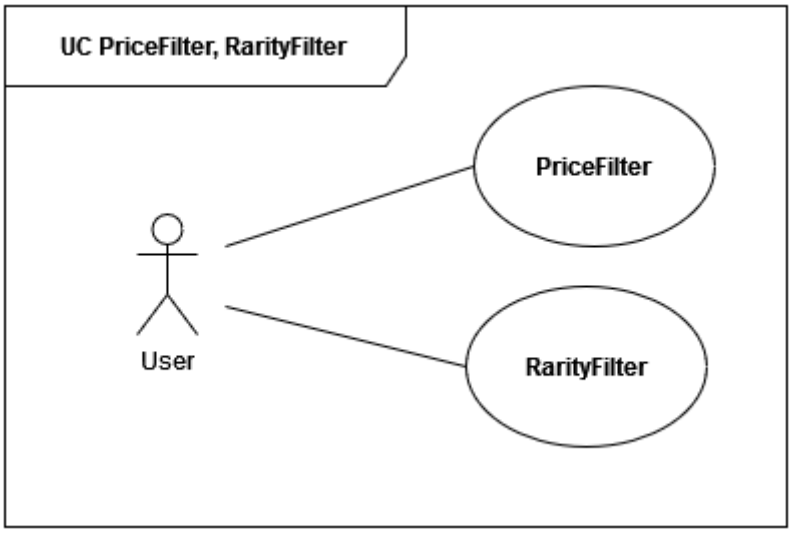
Caso d’uso 2: Login



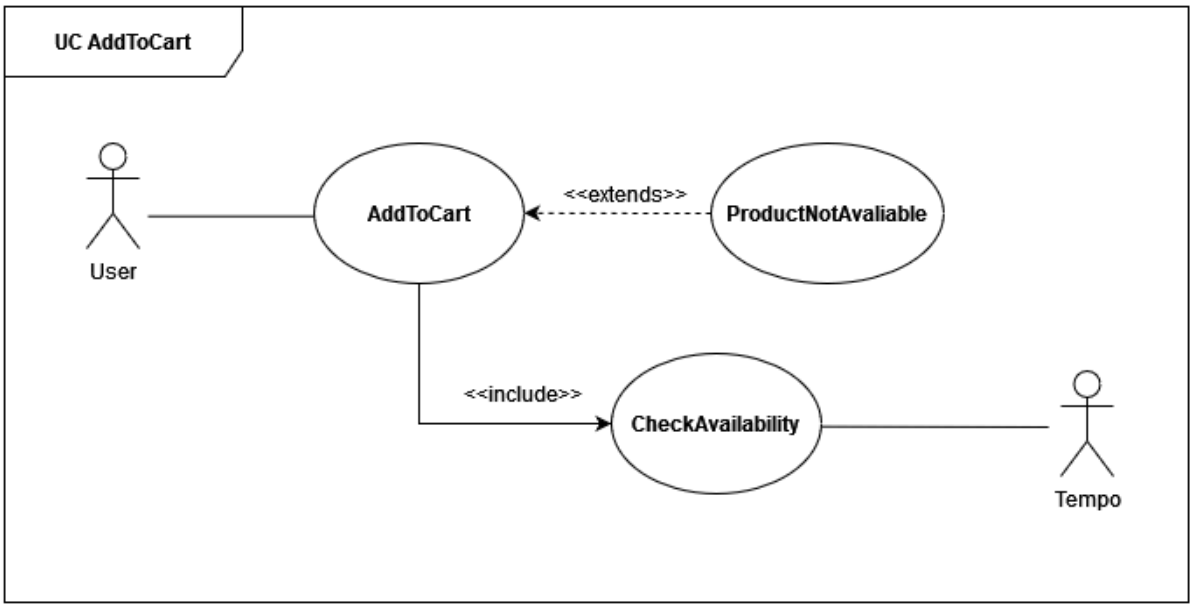
Caso d’uso 3: CheckAvailability



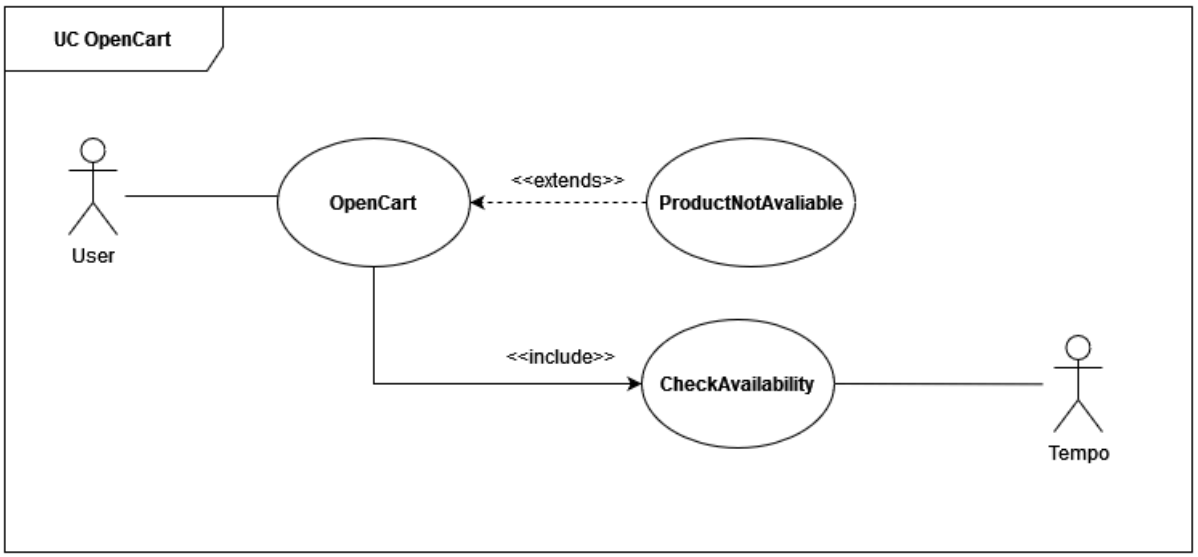
Caso d’uso 4: DatabaseSearch



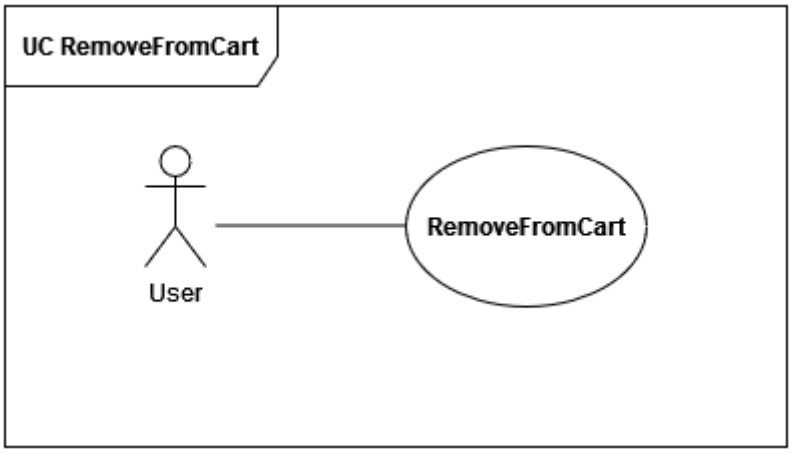
Casi d’uso 5-6: PriceFilter, RarityFilter



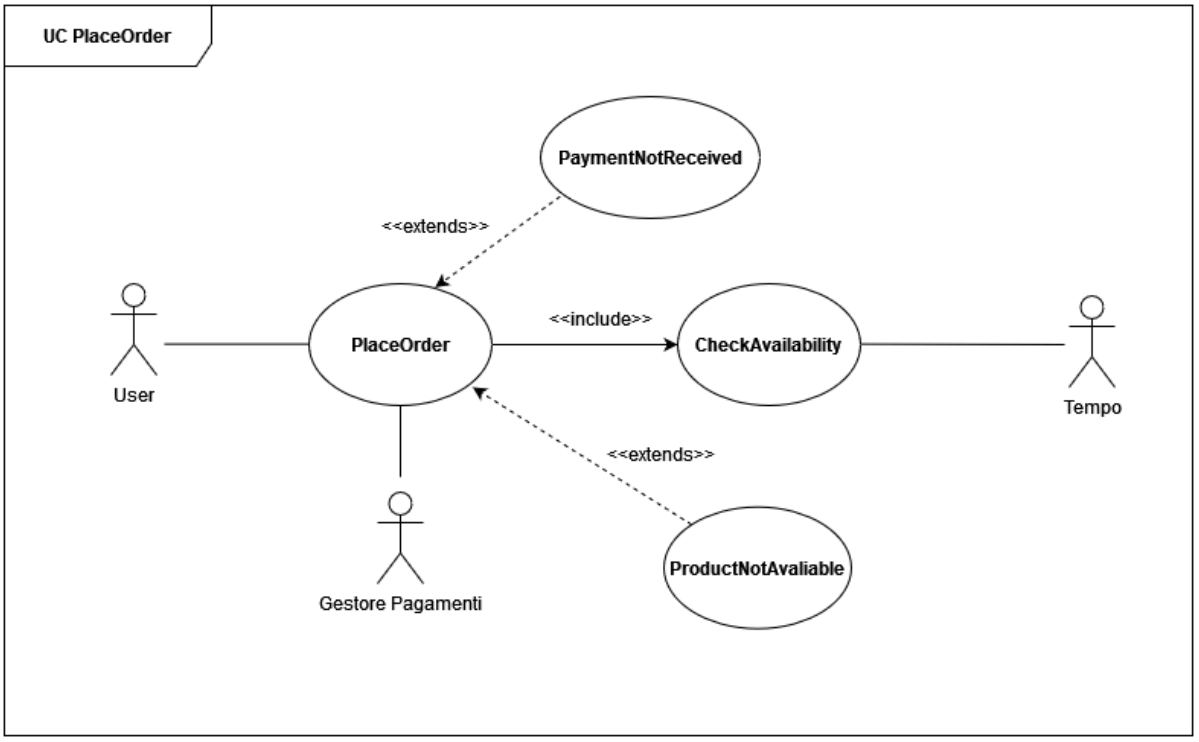
Caso d’uso 7: AddToCart



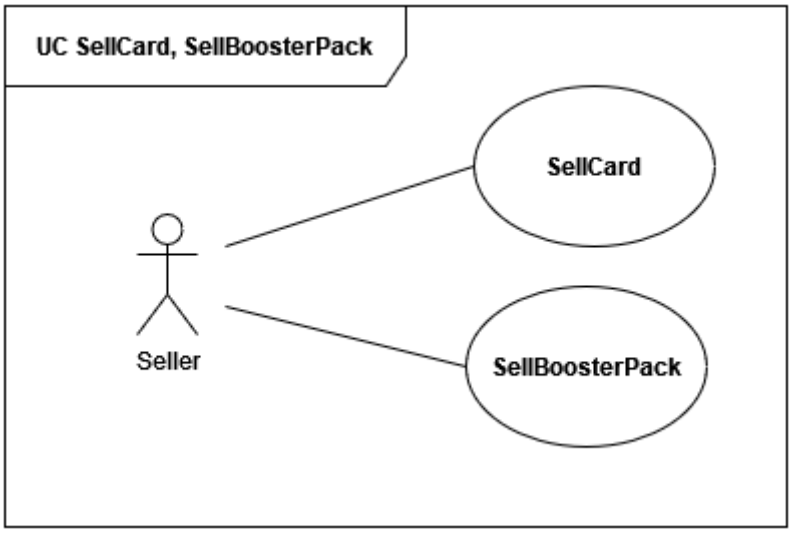
Caso d’uso 8: OpenCart



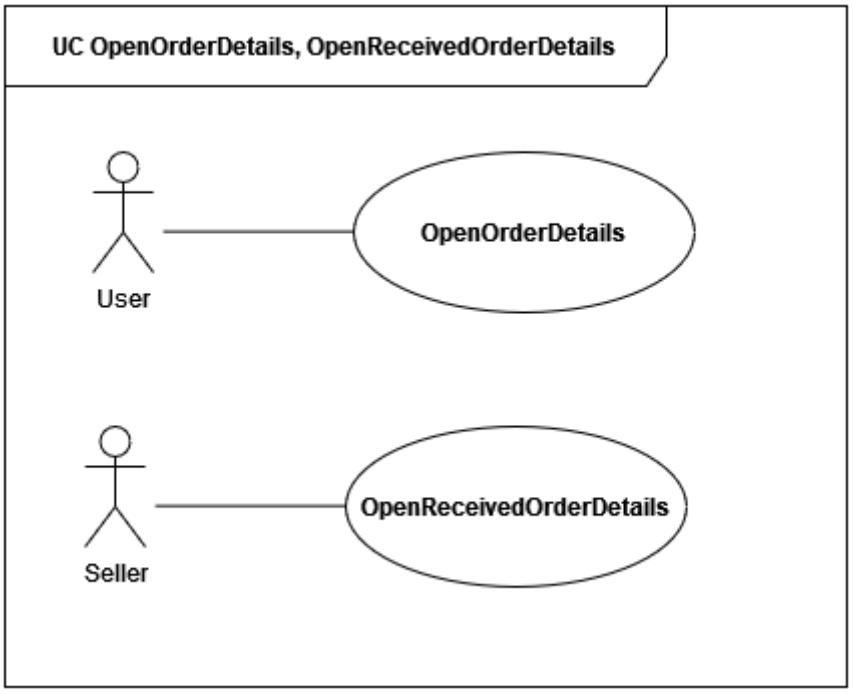
Caso d’uso 9: RemoveFromCart



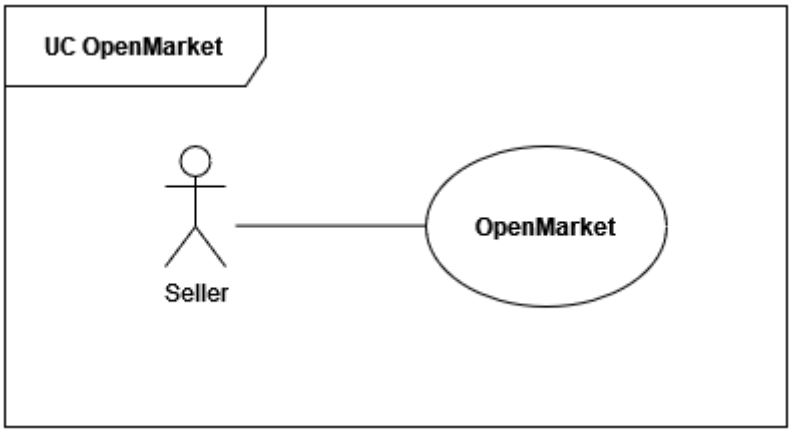
Caso d’uso 10: PlaceOrder



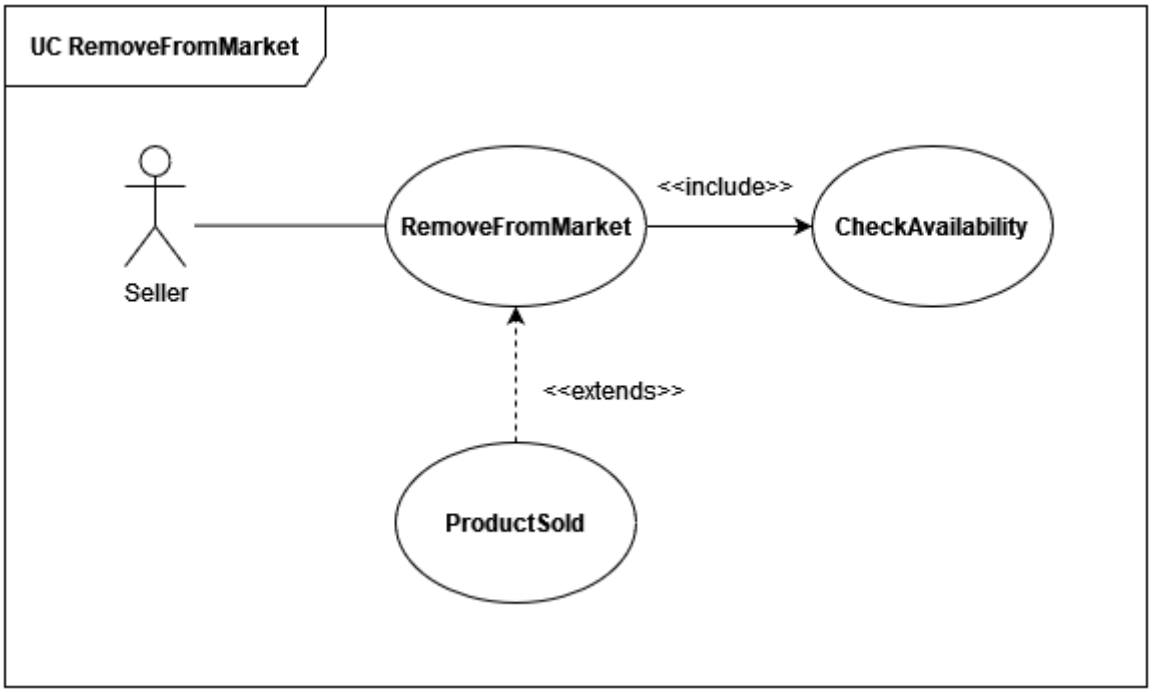
Casi d'uso 11-12: SellCard, SellBoosterPack



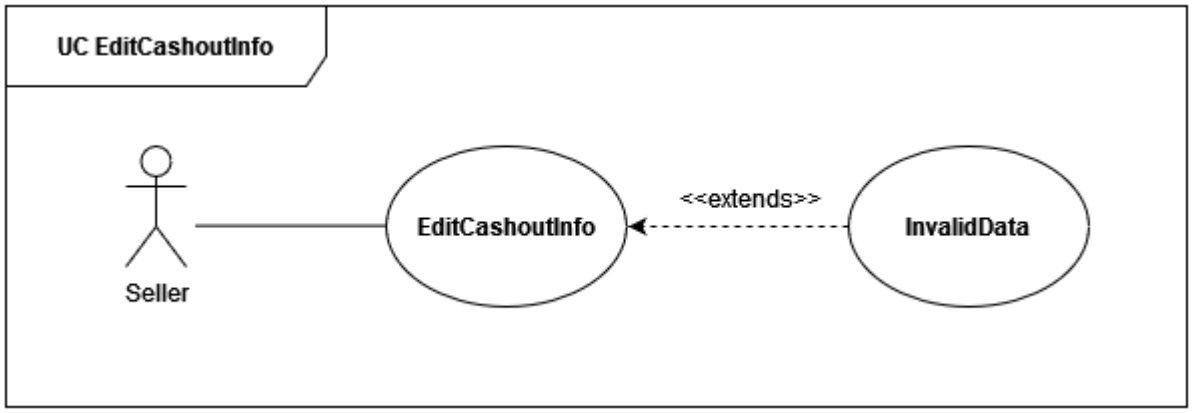
Casi d'uso 13-14: OpenOrderDetails, OpenReceivedOrderDetails



Caso d’uso 15: OpenMarket



Caso d’uso 16: RemoveFromMarket



Caso d’uso 17: EditCashoutInfo

4 Scenari dei casi d'uso

4.0 UserSignup

Use Case ID:	#0000
Use Case Name:	UserSignup
Actors:	User
Description:	L'utente crea il suo account.
Preconditions:	
Postconditions:	Il sistema garantisce all'utente l'accesso alla sua area personale.
Normal Flow:	1. L'utente inserisce le sue informazioni di registrazione (username e password). 2. Il sistema invia una query al database dei dati per verificare l'unicità dell'username. 3. Il database conferma l'unicità dell'username. 4. Il sistema aggiunge al database dei dati le informazioni dell'utente appena registrato. 5. Il sistema porta in visualizzazione l'area personale dell'utente.
Alternative Flow:	
Exceptions:	L'username esiste già in database: 1. Il sistema porta in visualizzazione un messaggio di errore.
Includes:	

Scenario 0: UserSignup

4.1 SellerSignup

Use Case ID:	#0001
Use Case Name:	SellerSignup
Actors:	Seller
Description:	Il venditore crea il suo account.
Preconditions:	
Postconditions:	Il sistema garantisce al venditore l'accesso alla sua area personale.
Normal Flow:	1. Il venditore inserisce le sue informazioni di registrazione (username, password e informazioni di cashout). 2. Il sistema verifica la validità delle informazioni di cashout. 3. Il sistema invia una query al database dei dati per verificare l'unicità dell'username. 4. Il database conferma l'unicità dell'username. 5. Il sistema aggiunge al database dei dati le informazioni dell'utente appena registrato. 6. Il sistema porta in visualizzazione l'area personale dell'utente.
Alternative Flow:	
Exceptions:	Le informazioni di cashout non sono valide: 1. II sistema porta in visualizzazione un messaggio di errore. L'username esiste già in database: 1. Il sistema porta in visualizzazione un messaggio di errore.
Includes:	

Scenario 1: SellerSignup

4.2 Login

Use Case ID:	#0002
Use Case Name:	Login
Actors:	User
Description:	L'utente accede al suo account.
Preconditions:	
Postconditions:	Il sistema garantisce all'utente l'accesso alla sua area personale.
Normal Flow:	1. L'utente inserisce le sue informazioni di login (username e password). 2. Il sistema invia una query al database dei dati per la verifica dei dati. 3. Il database conferma la correttezza delle informazioni di login. 4. Il sistema visualizza l'area personale dell'utente.
Alternative Flow:	
Exceptions:	Le informazioni di login inserite dall'utente non sono corrette: 1. Il sistema porta in visualizzazione un messaggio di errore.
Includes:	

Scenario 2: Login

4.3 CheckAvailability

Use Case ID:	#0003
Use Case Name:	CheckAvailability
Actors:	Tempo
Description:	Il sistema verifica la disponibilità di uno o più prodotti.
Preconditions:	
Postconditions:	Il sistema fornisce le informazioni più aggiornate circa la disponibilità di uno o più prodotti.
Normal Flow:	1. Il sistema invia una query al database dei prodotti per verificare la disponibilità di uno o più prodotti richiesti. 2. Il database dei prodotti risponde con le informazioni aggiornate sui prodotti richiesti.
Alternative Flow:	
Exceptions:	
Includes:	

Scenario 3: CheckAvailability

4.4 DatabaseSearch

Use Case ID:	#0004
Use Case Name:	DatabaseSearch
Actors:	User, Tempo
Description:	L'utente ricerca un prodotto in vendita sul sistema.
Preconditions:	
Postconditions:	Il sistema porta in visualizzazione tutte le istanze di vendita disponibili per il prodotto cercato.
Normal Flow:	1. L'utente inserisce il nome del prodotto richiesto. 2. Il sistema verifica la disponibilità del prodotto richiesto. (Use Case #0003) 3. Il sistema porta in visualizzazione all'utente i risultati della ricerca.
Alternative Flow:	
Exceptions:	Il prodotto cercato non esiste all'interno del database: 1. Il sistema porta in visualizzazione all'utente un messaggio di errore circa l'inesistenza del prodotto richiesto.
Includes:	Check Availability (Use Case #0003)

Scenario 4: DatabaseSearch

4.5 PriceFilter

Use Case ID:	#0005
Use Case Name:	PriceFilter
Actors:	User
Description:	Il sistema ordina i risultati in base al prezzo.
Preconditions:	L'utente deve aver effettuato una ricerca di un prodotto oppure il venditore deve aver accesso ai prodotti da lui messi in vendita.
Postconditions:	Il sistema porta in visualizzazione tutte le istanze di vendita disponibili per il prodotto cercato ordinate secondo il prezzo nel modo specificato dall'utente.
Normal Flow:	1. L'utente seleziona una tipologia di ordinamento di prezzo a scelta tra crescente o decrescente. 2. Il sistema ordina i risultati nel modo richiesto. 3. Il sistema porta in visualizzazione i risultati ordinati.
Alternative Flow:	1.1. Il venditore accede ai prodotti da lui messi in vendita e seleziona una tipologia di ordinamento di prezzo a scelta tra crescente o decrescente per le carte da gioco o le espansioni da lui messe in vendita.
Exceptions:	
Includes:	

Scenario 5: PriceFilter

4.6 RarityFilter

Use Case ID:	#0006
Use Case Name:	RarityFilter
Actors:	User
Description:	Il sistema ordina i risultati in base alla rarità.
Preconditions:	L'utente deve aver effettuato una ricerca di una carta da gioco oppure il venditore deve aver accesso alle carte da gioco da lui messe in vendita.
Postconditions:	Il sistema porta in visualizzazione tutte le istanze di vendita disponibili per il prodotto cercato ordinate secondo rarità nel modo specificato dall'utente.
Normal Flow:	1. L'utente seleziona una tipologia di ordinamento di rarità a scelta tra crescente o decrescente. 2. Il sistema ordina i risultati nel modo richiesto. 3. Il sistema porta in visualizzazione i risultati ordinati.
Alternative Flow:	1.1. Il venditore accede alle carte da gioco da lui messe in vendita e seleziona una tipologia di ordinamento di rarità a scelta tra crescente e decrescente per le carte da gioco da lui messe in vendita.
Exceptions:	
Includes:	

Scenario 6: RarityFilter

4.7 AddToCart

Use Case ID:	#0007
Use Case Name:	AddToCart
Actors:	User, Tempo
Description:	L'utente aggiunge al carrello il prodotto desiderato.
Preconditions:	L'utente deve aver effettuato una ricerca di un prodotto.
Postconditions:	Il prodotto selezionato è salvato nel carrello dell'utente.
Normal Flow:	1. L'utente seleziona uno dei risultati da aggiungere al carrello. 2. Il sistema verifica la disponibilità dei prodotti da aggiungere al carrello (Use Case #0003). 3. Il sistema aggiorna il carrello dell'utente, aggiungendo i prodotti desiderati. 4. Il sistema porta in visualizzazione un messaggio di buona riuscita dell'operazione.
Alternative Flow:	
Exceptions:	Il prodotto selezionato non è più disponibile: 1. il sistema porta in visualizzazione un messaggio di errore.
Includes:	CheckAvailability (Use Case #0003)

Scenario 7: AddToCart

4.8 OpenCart

Use Case ID:	#0008
Use Case Name:	OpenCart
Actors:	User, Tempo
Description:	L'utente apre il proprio carrello.
Preconditions:	
Postconditions:	Il sistema porta in visualizzazione tutti i prodotti disponibili aggiunti al carrello dall'utente.
Normal Flow:	1. L'utente accede alla sua area personale e seleziona di visualizzare il proprio carrello. 2. Il sistema invia una query al database dei dati per ottenere una lista dei prodotti aggiunti al carrello. 3. Il database risponde con la lista dei prodotti aggiunti al carrello dall'utente aggiornata all'ultima modifica. 4. Il sistema verifica la disponibilità dei prodotti presenti nel carrello (Use Case #0003). 5. Il sistema porta in visualizzazione la lista dei prodotti aggiunti al carrello dall'utente.
Alternative Flow:	4.1. In caso di uno o più prodotti non disponibili, il sistema rimuove dal carrello quei prodotti. 5.1. Il sistema porta in visualizzazione la lista dei prodotti disponibili presenti nel carrello, insieme ad un messaggio di notifica di non disponibilità per i prodotti non più disponibili
Exceptions:	
Includes:	CheckAvailability (Use Case #0003)

Scenario 8: OpenCart

4.9 RemoveFromCart

Use Case ID:	#0009
Use Case Name:	RemoveFromCart
Actors:	User
Description:	L'utente rimuove un prodotto dal proprio carrello.
Preconditions:	L'utente deve aver aperto il proprio carrello.
Postconditions:	Il sistema aggiorna il carrello dell'utente, rimuovendo il prodotto selezionato.
Normal Flow:	1. L'utente seleziona un prodotto da rimuovere dal carrello. 2. Il sistema aggiorna il database dei dati modificando le informazioni sul carrello dell'utente. 3. Il sistema porta in visualizzazione un messaggio di conferma di rimozione.
Alternative Flow:	
Exceptions:	
Includes:	

Scenario 9: RemoveFromCart

4.10 PlaceOrder

Use Case ID:	#0010
Use Case Name:	PlaceOrder
Actors:	User, Tempo, Gestore Pagamenti
Description:	L'utente acquista i prodotti disponibili presenti nel suo carrello.
Preconditions:	L'utente deve aver aperto il proprio carrello, che contiene almeno un prodotto disponibile al momento dell'acquisto.
Postconditions:	L'utente completa l'ordine dei prodotti disponibili presenti nel suo carrello.
Normal Flow:	1. L'utente avvia la procedura di inoltro ordine dopo aver aperto il carrello. 2. Il sistema verifica la disponibilità dei prodotti che l'utente ha nel carrello al momento dell'acquisto (Use Case #0003). 3. Il sistema reindirizza l'utente al sistema di gestione pagamenti. 4. Il sistema di gestione pagamenti conferma l'avvenuto pagamento. 5. Il sistema rimuove dal database dei prodotti le informazioni sui prodotti appena acquistati dall'utente. 6. Il sistema aggiorna il carrello e la lista ordini effettuati dell'utente. 7. Il sistema porta in visualizzazione un messaggio di conferma ordine all'utente.
Alternative Flow:	
Exceptions:	Il sistema di gestione pagamenti non conferma il pagamento: 1. Il sistema mostra un messaggio di non completamento dell'ordine. Uno o più prodotti non sono disponibili: 1. Il sistema porta in visualizzazione un messaggio di errore e annulla l'esecuzione dell'ordine.
Includes:	CheckAvailability (Use Case #0003)

Scenario 10: PlaceOrder

4.11 SellCard

Use Case ID:	#0011
Use Case Name:	SellCard
Actors:	Seller
Description:	Il venditore mette in vendita una carta da gioco tra quelle listate dal sistema.
Preconditions:	
Postconditions:	Il sistema aggiorna il database dei prodotti circa la disponibilità della carta da gioco messa in vendita dal venditore, aggiungendo la copia in suo possesso.
Normal Flow:	1. Il seller seleziona la carta da gioco di cui dispone una copia da voler mettere in vendita. 2. Il sistema invia una query al database dei prodotti circa le rarità rilasciate della carta da gioco selezionata. 3. Il database dei prodotti risponde con una lista di tutte le rarità rilasciate della carta da gioco selezionata. 4. Il sistema porta in visualizzazione la lista delle rarità disponibili. 5. Il seller seleziona la rarità corrispondente alla copia da mettere in vendita. 6. Il seller inserisce il prezzo di vendita della copia della carta da gioco da lui in possesso. 7. Il sistema aggiorna il database dei prodotti inserendo le informazioni sulla copia appena messa in vendita. 8. Il sistema porta in visualizzazione un messaggio di conferma di messa in vendita della carta da gioco selezionata dal seller.
Alternative Flow:	
Exceptions:	
Includes:	

Scenario 11: SellCard

4.12 SellBoosterPack

Use Case ID:	#0012
Use Case Name:	SellBoosterPack
Actors:	Seller
Description:	Il venditore mette in vendita un Booster Pack tra quelli listati dal sistema.
Preconditions:	
Postconditions:	Il sistema aggiorna il database dei prodotti circa la disponibilità del Booster Pack messo in vendita dal venditore, aggiungendo la copia in suo possesso.
Normal Flow:	1. Il seller seleziona il Booster Pack di cui dispone una copia da voler mettere in vendita. 2. Il seller inserisce il prezzo di vendita del Booster Pack da lui in possesso. 3. Il sistema aggiorna il database dei prodotti inserendo le informazioni sul Booster Pack appena messo in vendita. 4. Il sistema porta in visualizzazione un messaggio di conferma di messa in vendita del Booster Pack selezionato dal seller.
Alternative Flow:	
Exceptions:	
Includes:	

Scenario 12: SellBoosterPack

4.13 OpenOrderDetails

Use Case ID:	#0013
Use Case Name:	OpenOrderDetails
Actors:	User
Description:	L'utente visualizza una lista degli ordini effettuati.
Preconditions:	
Postconditions:	Il sistema porta in visualizzazione una lista degli ordini effettuati con i rispettivi dettagli d'ordine.
Normal Flow:	1. L'utente richiede di visualizzare gli ordini inoltrati dalla sua area personale. 2. Il sistema invia una query al database dei dati circa le informazioni sugli ordini inoltrati dell'utente. 3. Il database risponde con una lista degli ordini effettuati. 4. Il sistema porta in visualizzazione le informazioni ricevute dal database.
Alternative Flow:	1.1. L'utente seleziona di visualizzare i dettagli di un ordine appena completato. 2.1. Il sistema porta in visualizzazione i dettagli sull'ordine appena completato.
Exceptions:	
Includes:	

Scenario 13: OpenOrderDetails

4.14 OpenReceivedOrderDetails

Use Case ID:	#0014
Use Case Name:	OpenReceivedOrderDetails
Actors:	Seller
Description:	Il venditore visualizza una lista degli ordini ricevuti.
Preconditions:	
Postconditions:	Il sistema porta in visualizzazione al venditore una lista degli ordini ricevuti con i rispettivi dettagli d'ordine.
Normal Flow:	1. Il venditore richiede di visualizzare gli ordini ricevuti dalla sua area personale. 2. Il sistema invia una query al database dei dati circa le informazioni sugli ordini ricevuti dal venditore. 3. Il database risponde con una lista degli ordini ricevuti. 4. Il sistema porta in visualizzazione le informazioni ricevute dal database.
Alternative Flow:	
Exceptions:	
Includes:	

Scenario 14: OpenReceivedOrderDetails

4.15 OpenMarket

Use Case ID:	#0015
Use Case Name:	OpenMarket
Actors:	Seller
Description:	Il venditore visualizza una lista dei prodotti da lui messi in vendita.
Preconditions:	
Postconditions:	Il sistema porta in visualizzazione al venditore una lista dei prodotti messi in vendita dal venditore.
Normal Flow:	1. Il venditore richiede di visualizzare tutti i prodotti da lui messi in vendita dalla sua area personale. 2. Il sistema invia una query al database dei dati circa le informazioni sui prodotti messi in vendita dal venditore. 3. Il database risponde con una lista dei prodotti messi in vendita dal venditore. 4. Il sistema porta in visualizzazione le informazioni ricevute dal database.
Alternative Flow:	
Exceptions:	
Includes:	

Scenario 15: OpenMarket

4.16 RemoveFromMarket

Use Case ID:	#0016
Use Case Name:	RemoveFromMarket
Actors:	Seller
Description:	Il venditore rimuove un prodotto tra quelli da lui messi in vendita.
Preconditions:	Il venditore deve aver accesso alla pagina dei prodotti da lui messi in vendita.
Postconditions:	Il sistema rimuove dal database dei prodotti le informazioni sul prodotto selezionato dal venditore.
Normal Flow:	1. Il venditore seleziona un prodotto da rimuovere tra quelli da lui messi in vendita. 2. Il sistema verifica la disponibilità del prodotto richiesto. (Use Case #0003) 3. Il sistema rimuove le informazioni sul prodotto selezionato dal database dei prodotti. 4. Il sistema porta in visualizzazione un messaggio di conferma al venditore.
Alternative Flow:	/
Exceptions:	Il prodotto messo in vendita è già stato acquistato: 1. Il sistema porta in visualizzazione un messaggio di errore.
Includes:	CheckAvailability (Use Case #0003)

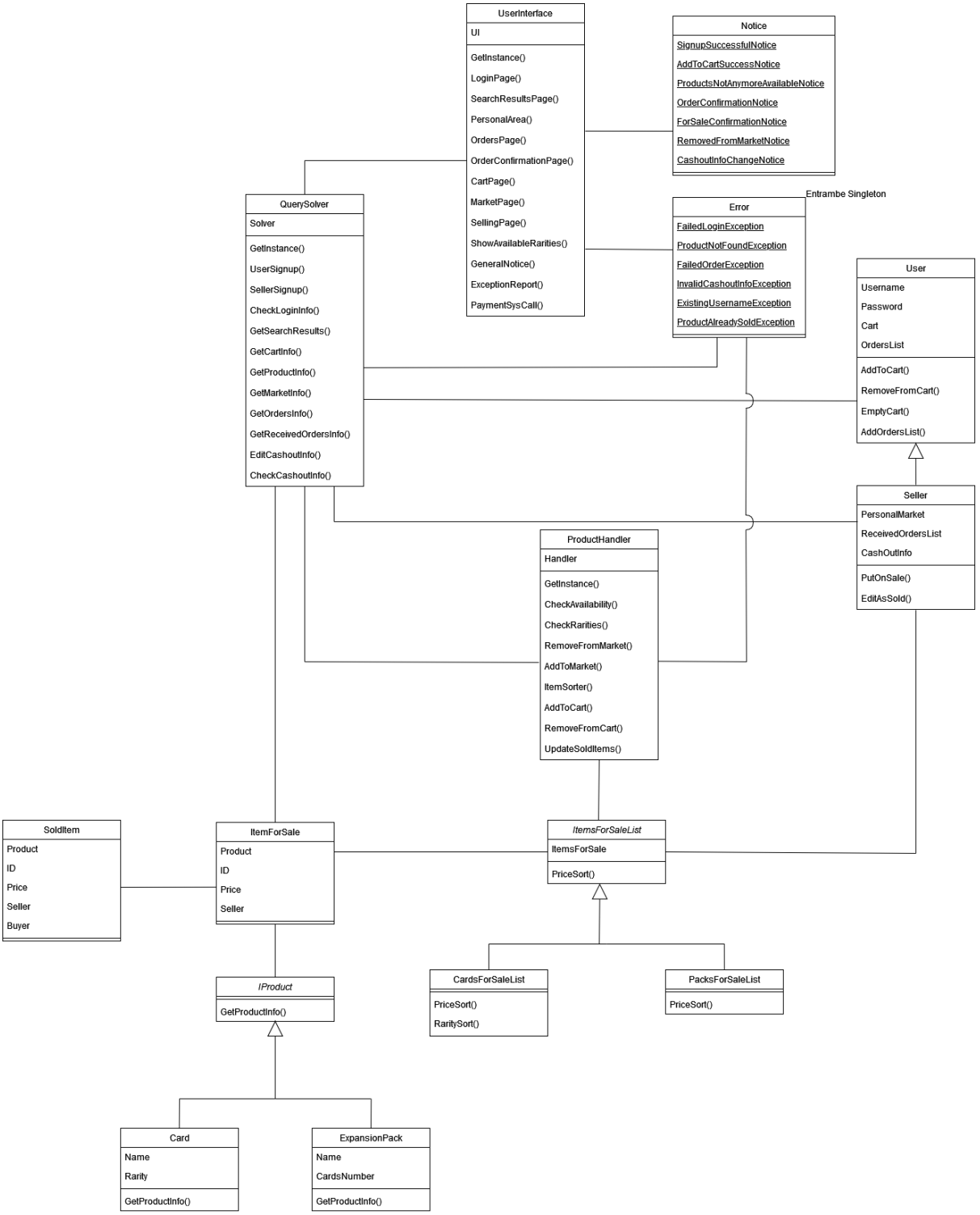
Scenario 16: RemoveFromMarket

4.17 EditCashoutInfo

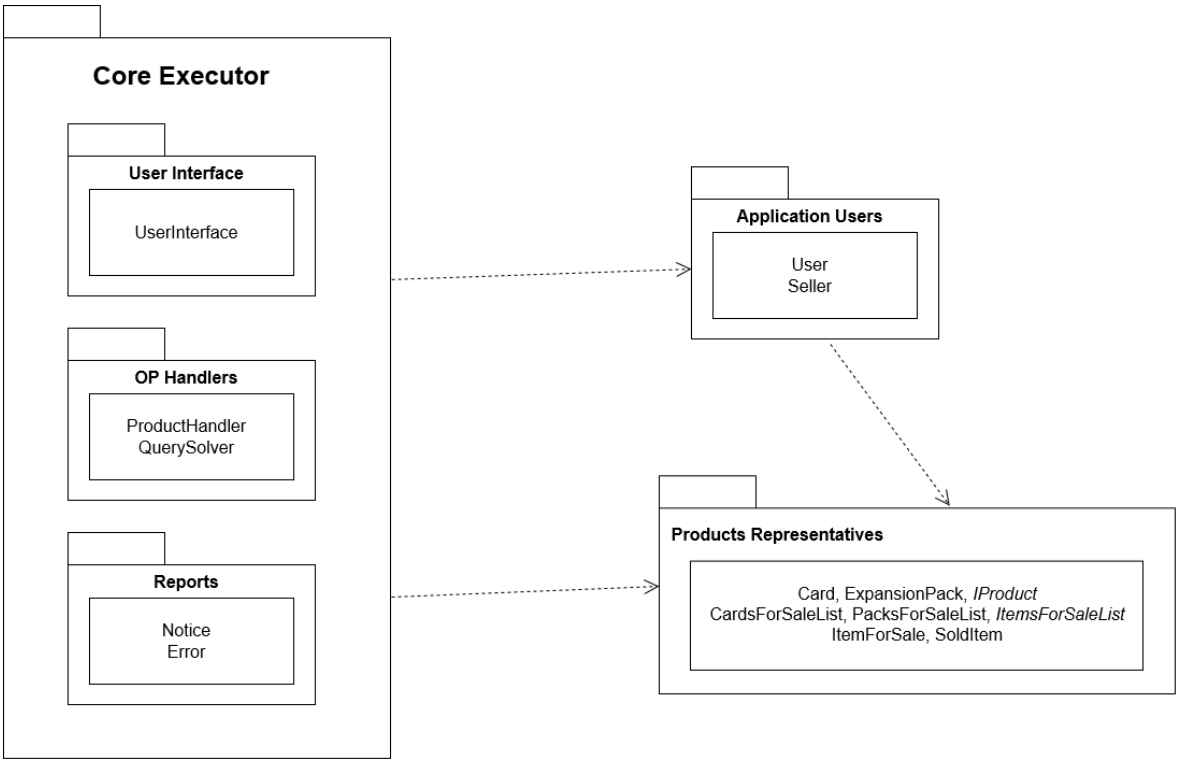
Use Case ID:	#0017
Use Case Name:	EditCashoutInfo
Actors:	Seller
Description:	Il venditore modifica il proprio metodo di pagamento.
Preconditions:	
Postconditions:	Il sistema aggiorna il metodo di pagamento del venditore nel database dei dati.
Normal Flow:	1. Il venditore inserisce le informazioni sul suo metodo di pagamento dalla sua area personale. 2. Il sistema verifica la validità delle informazioni inserite. 3. Il sistema aggiorna il database dei dati con le informazioni sottomesse dal venditore. 4. Il sistema porta in visualizzazione un messaggio di conferma di modifica.
Alternative Flow:	
Exceptions:	I dati inseriti non sono validi: 1. Il sistema porta in visualizzazione un messaggio di errore.
Includes:	

Scenario 17: EditCashoutInfo

5 Diagramma delle classi di analisi



6 Diagramma dei package di analisi



7 Diagrammi delle sequenze di analisi

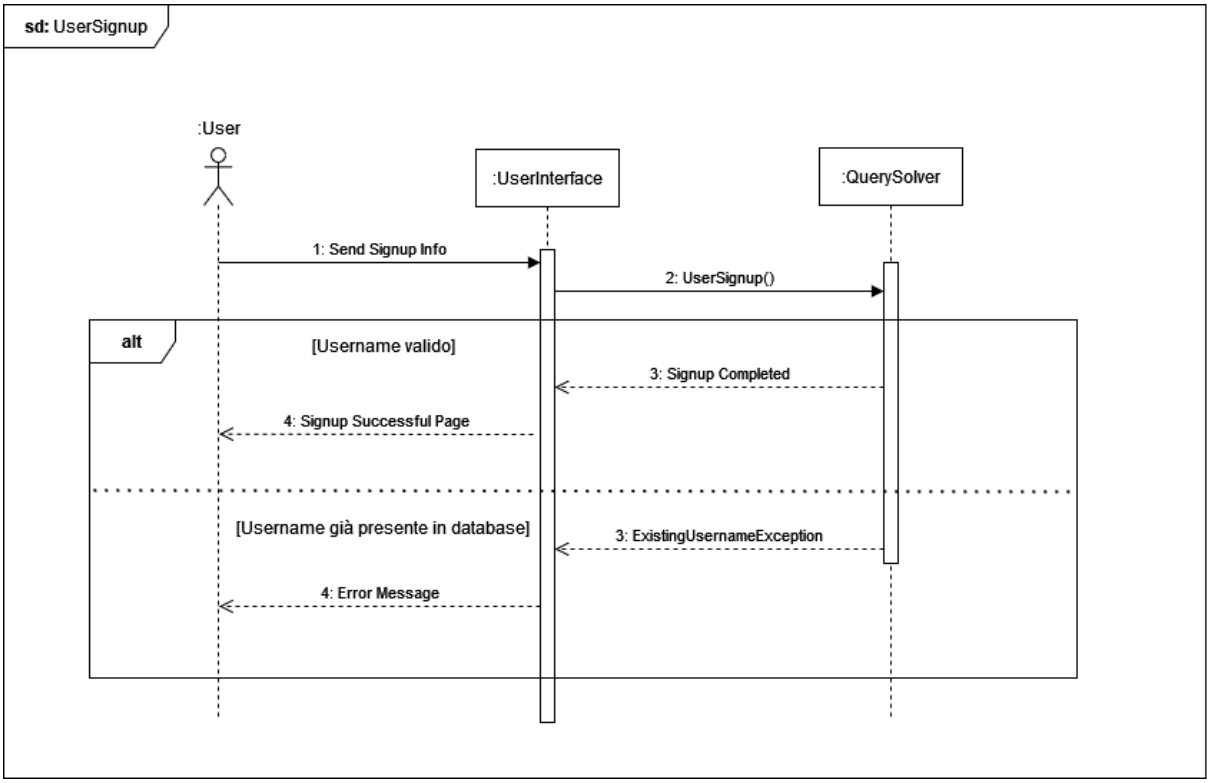


Diagramma sequenze 0: UserSignup

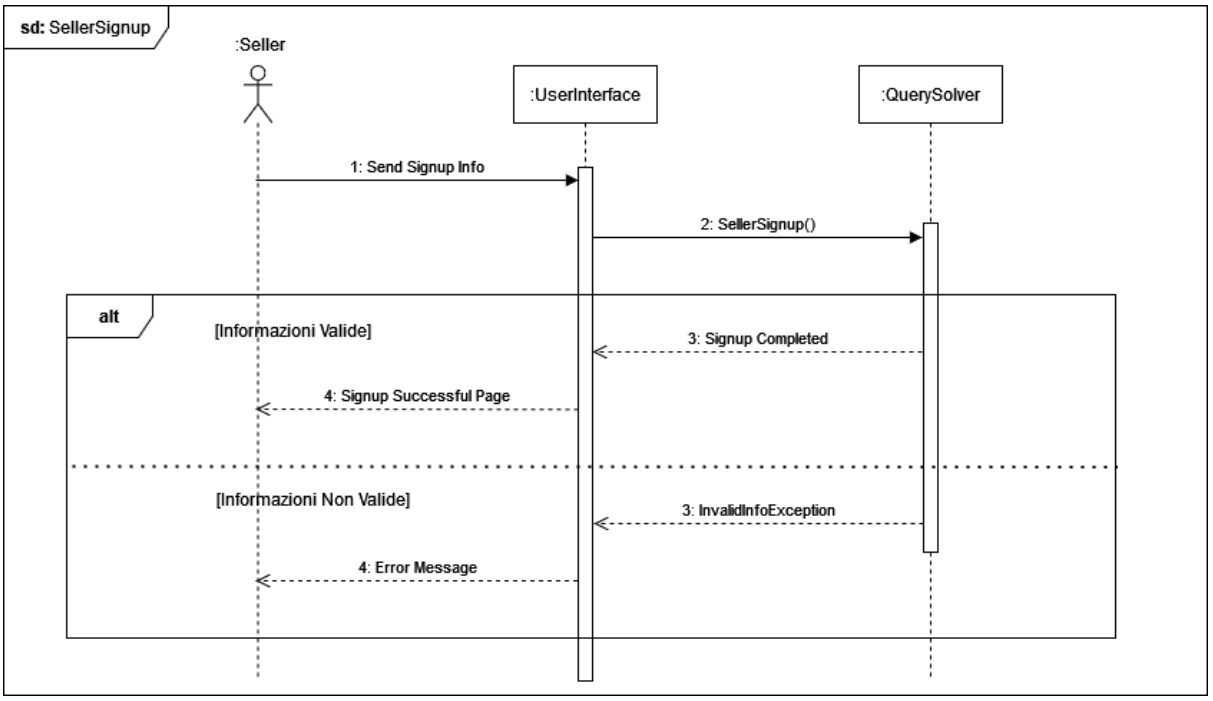


Diagramma sequenze 1: SellerSignup

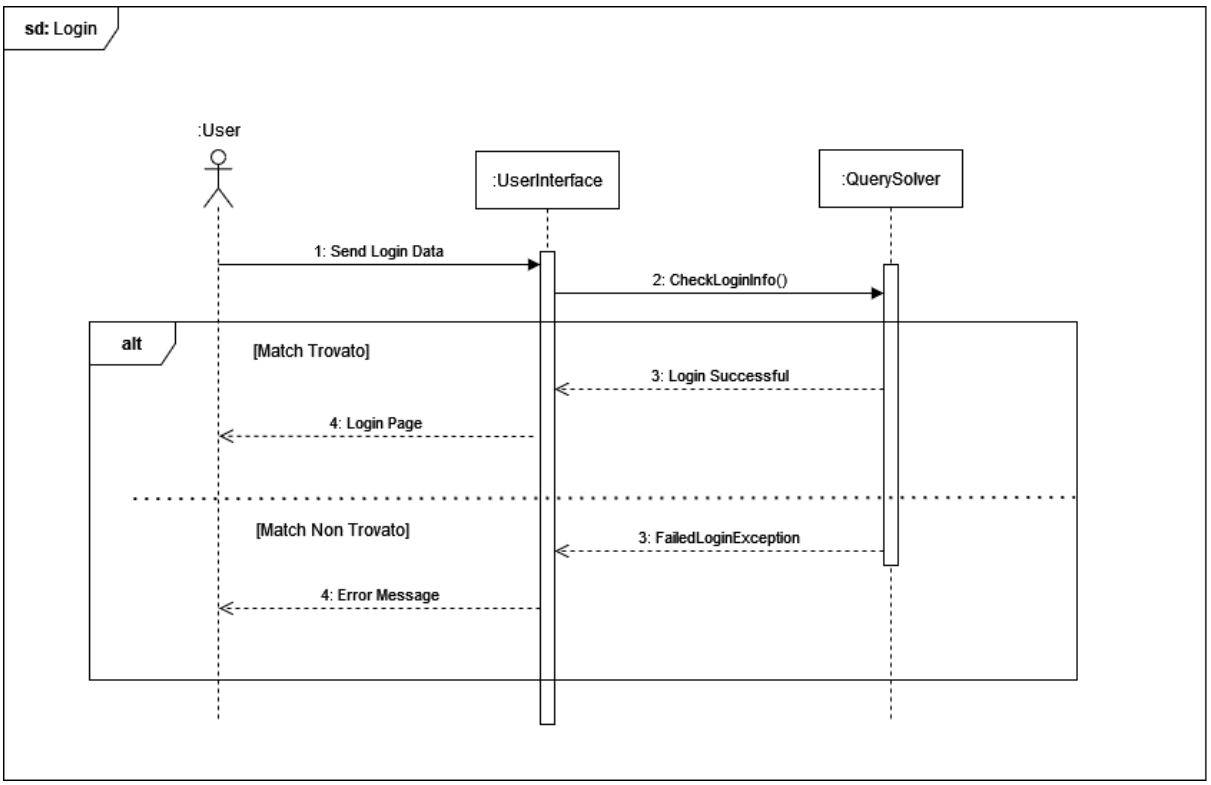


Diagramma sequenze 2: Login

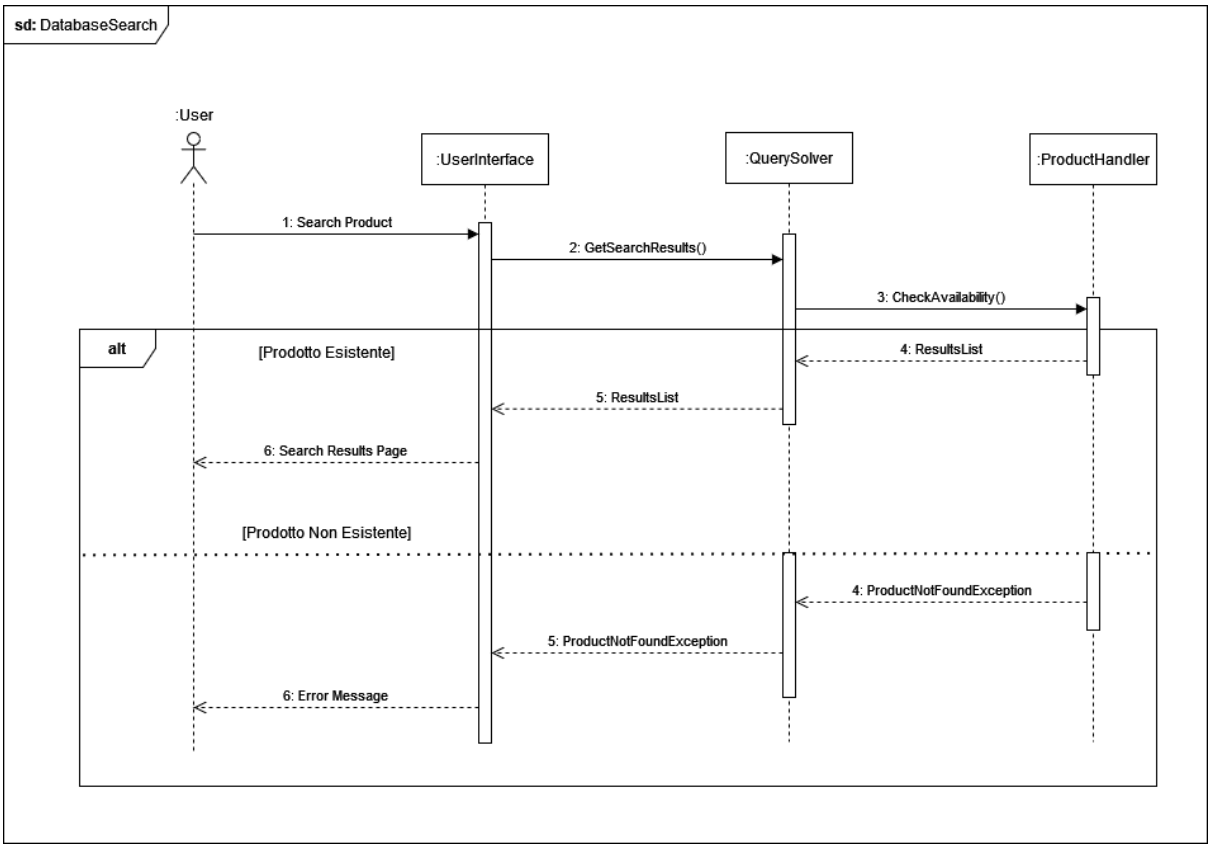


Diagramma sequenze 3: DatabaseSearch

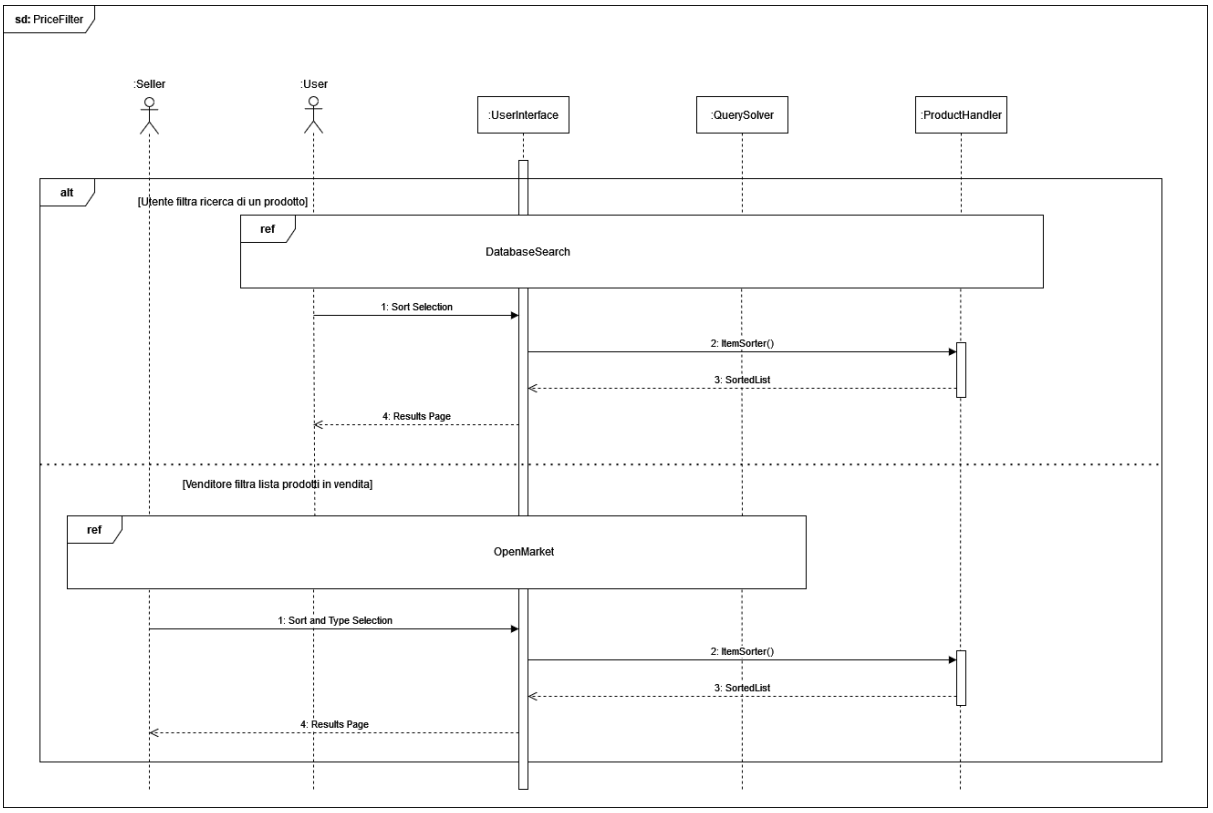


Diagramma sequenze 4: PriceFilter

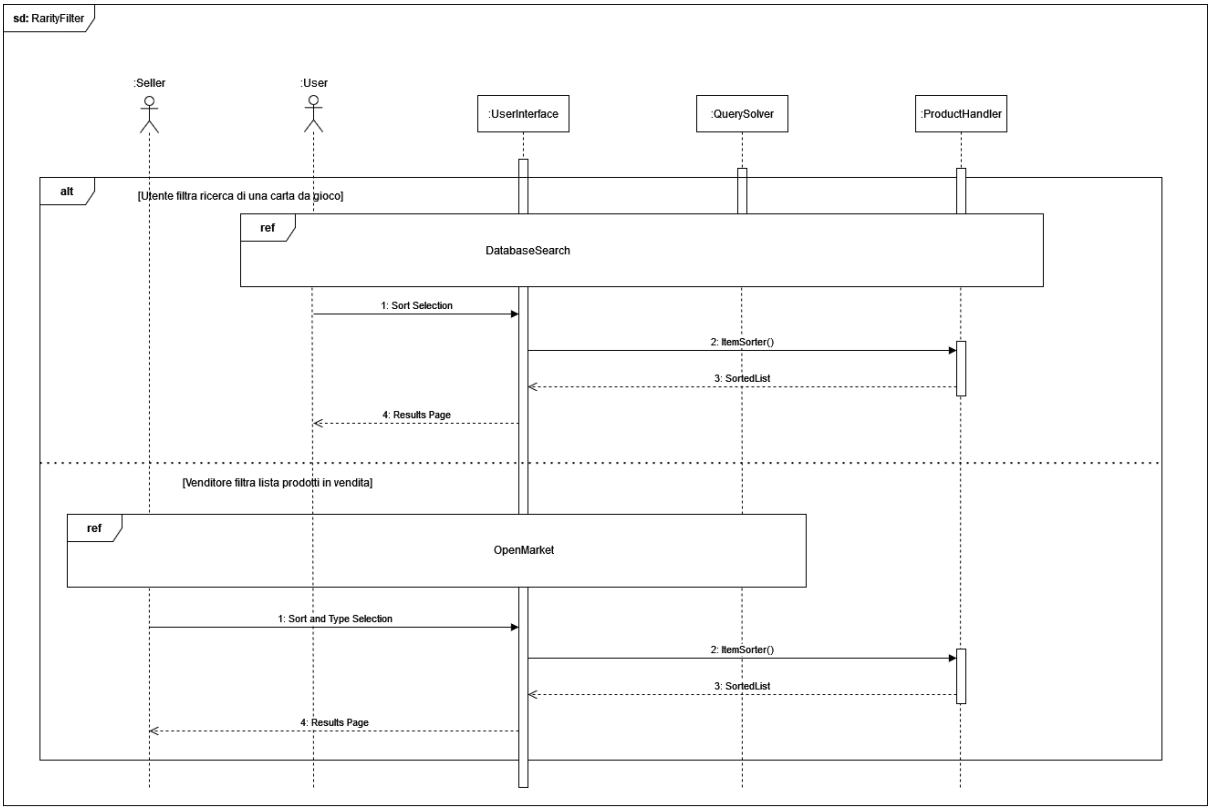


Diagramma sequenze 5: RarityFilter

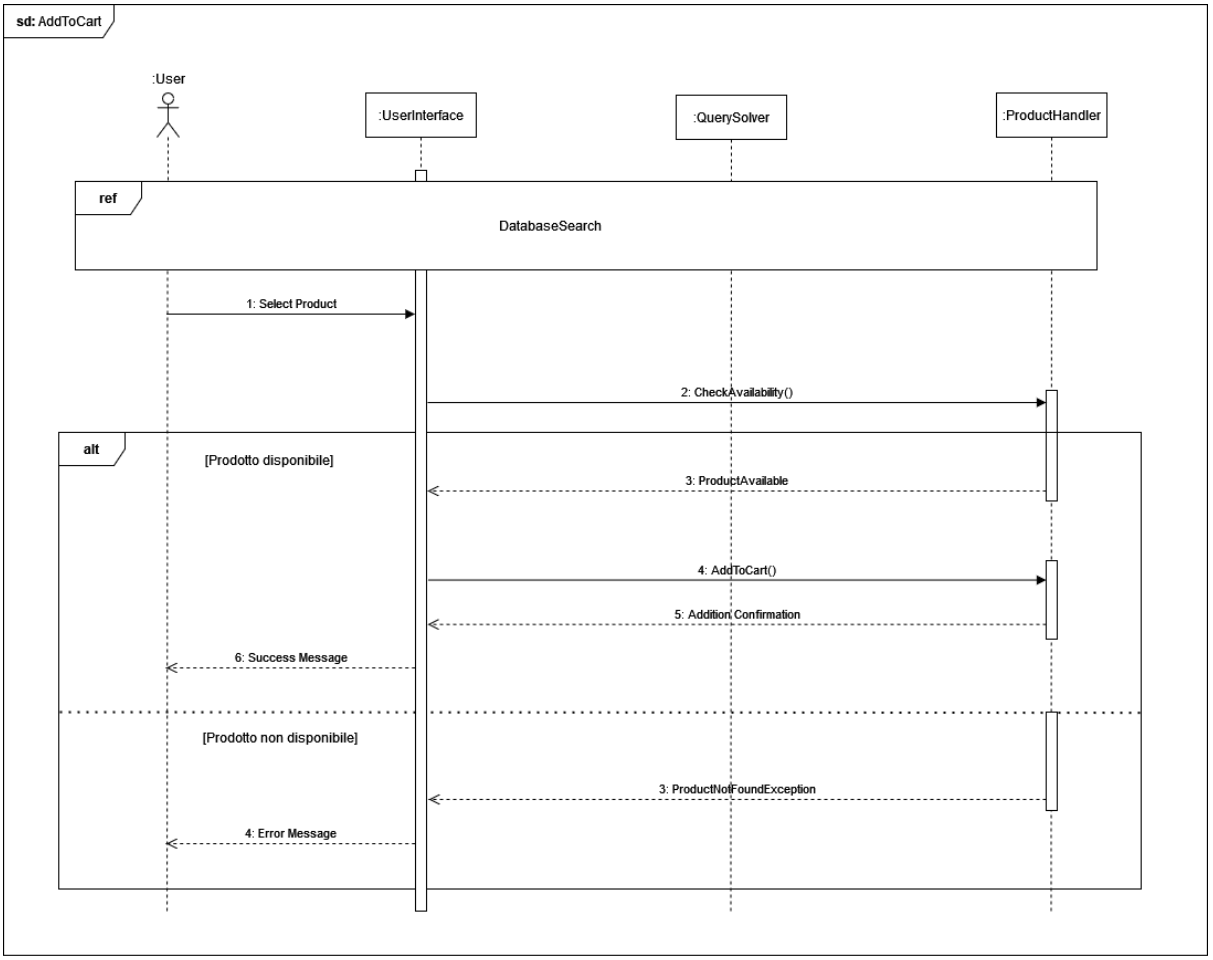


Diagramma sequenze 6: AddToCart

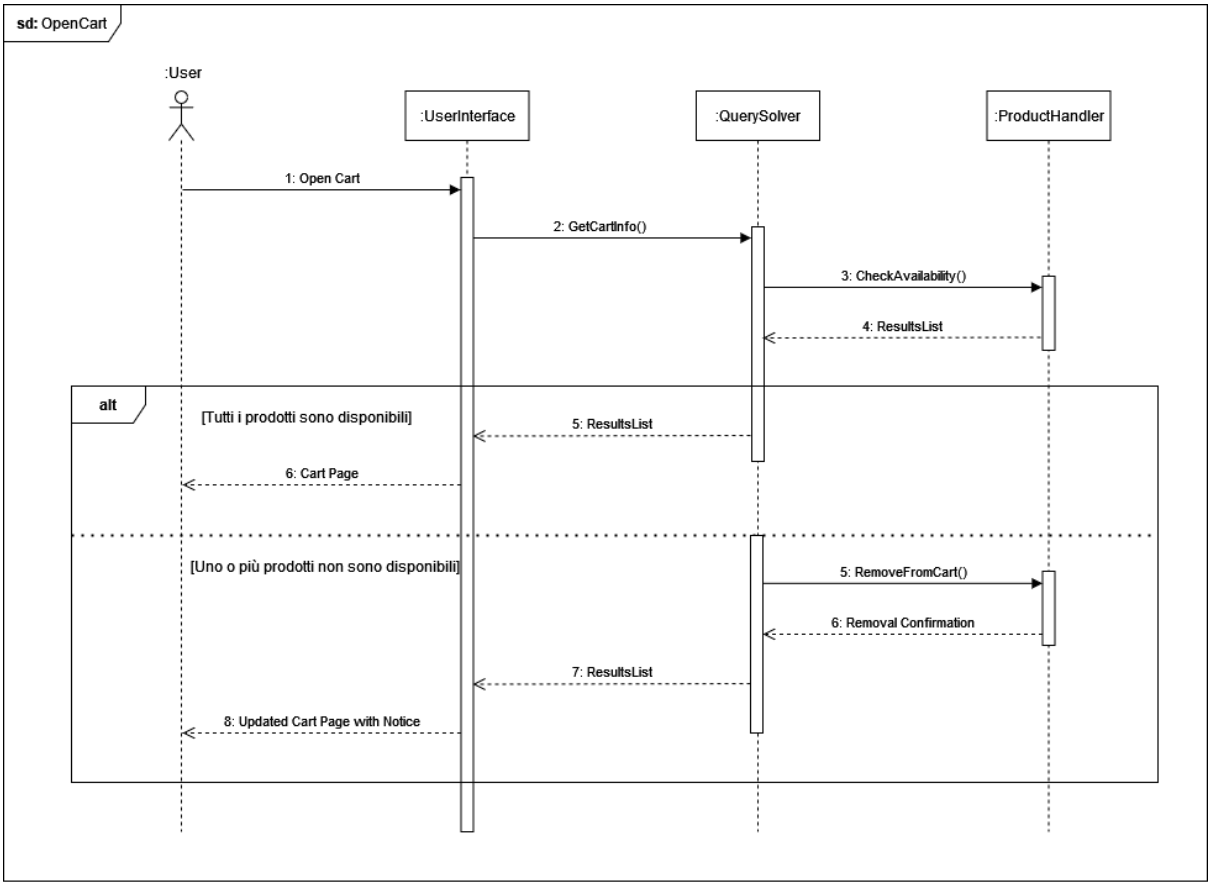


Diagramma sequenze 7: OpenCart

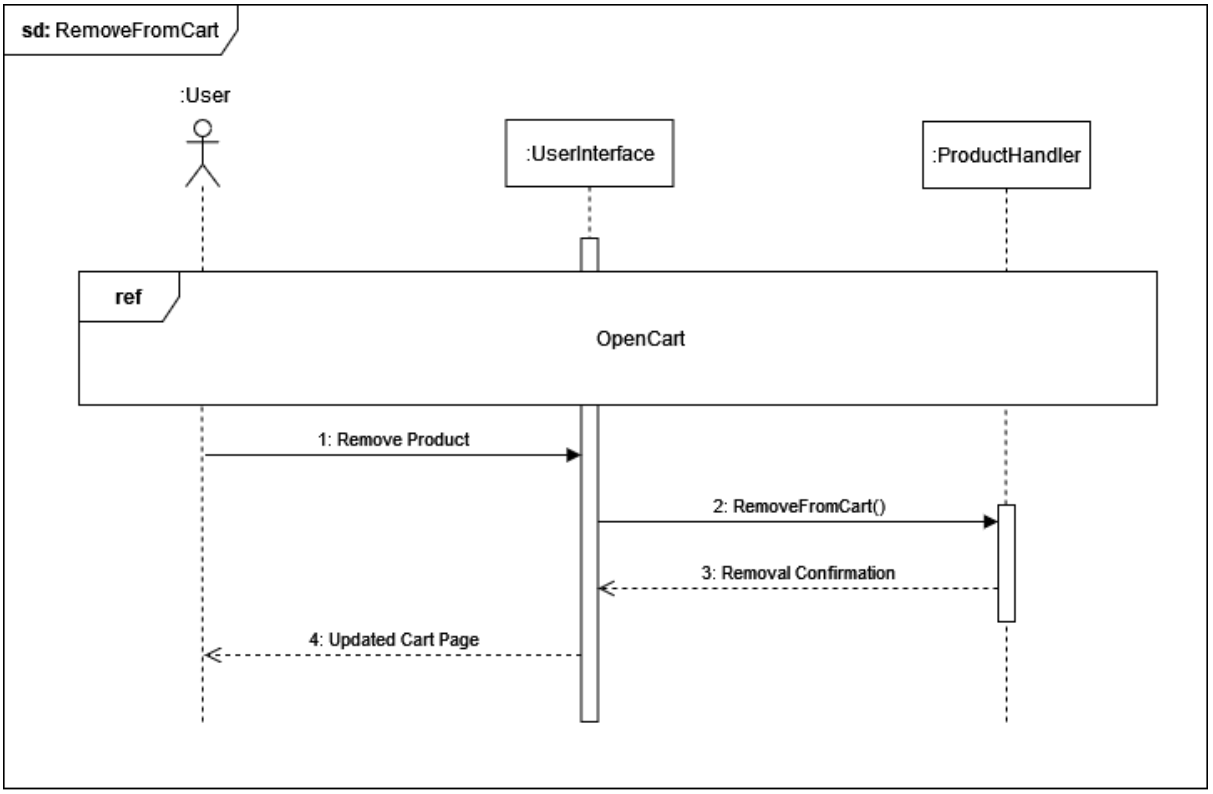


Diagramma sequenze 8: RemoveFromCart

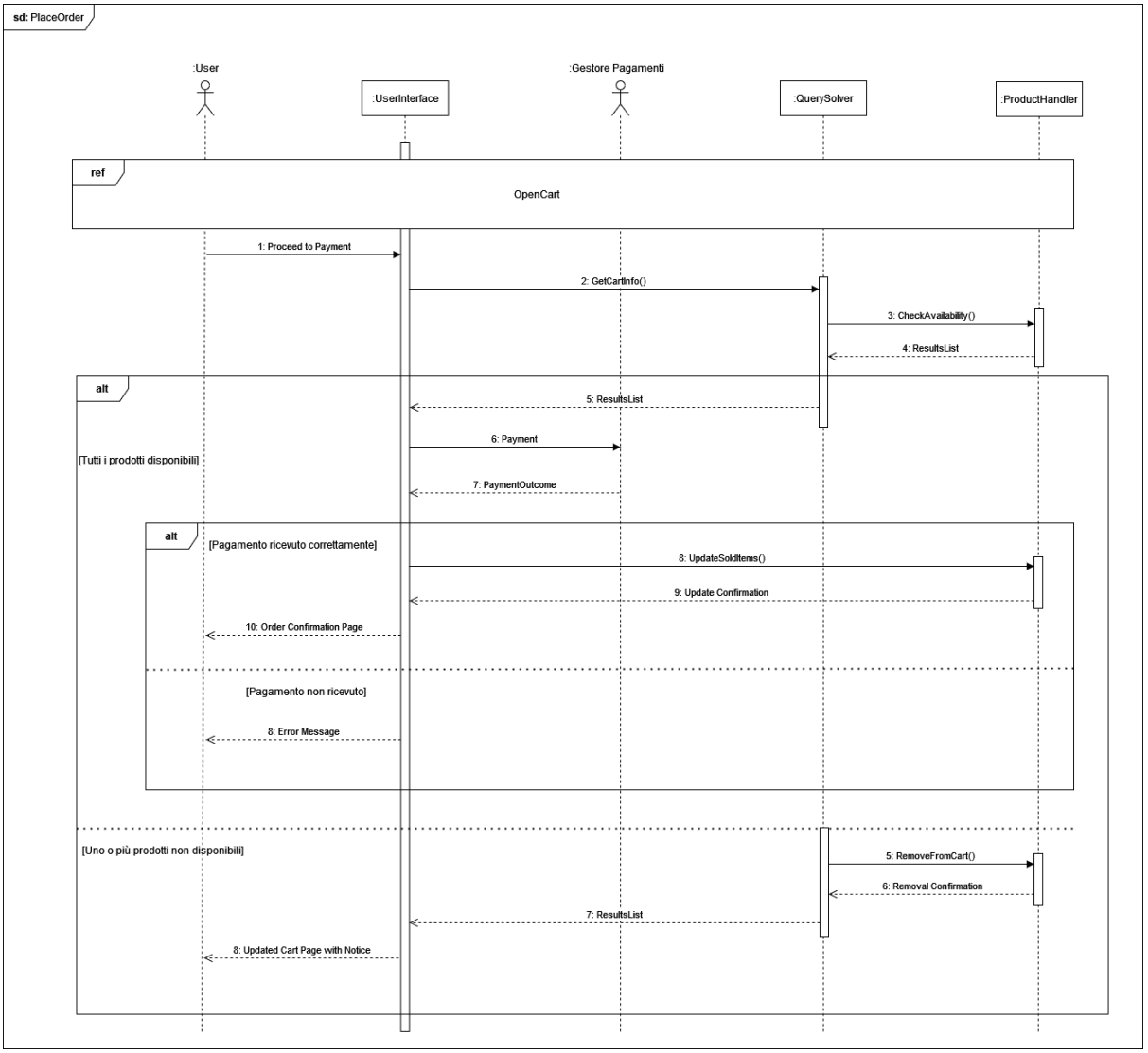


Diagramma sequenze 9: PlaceOrder

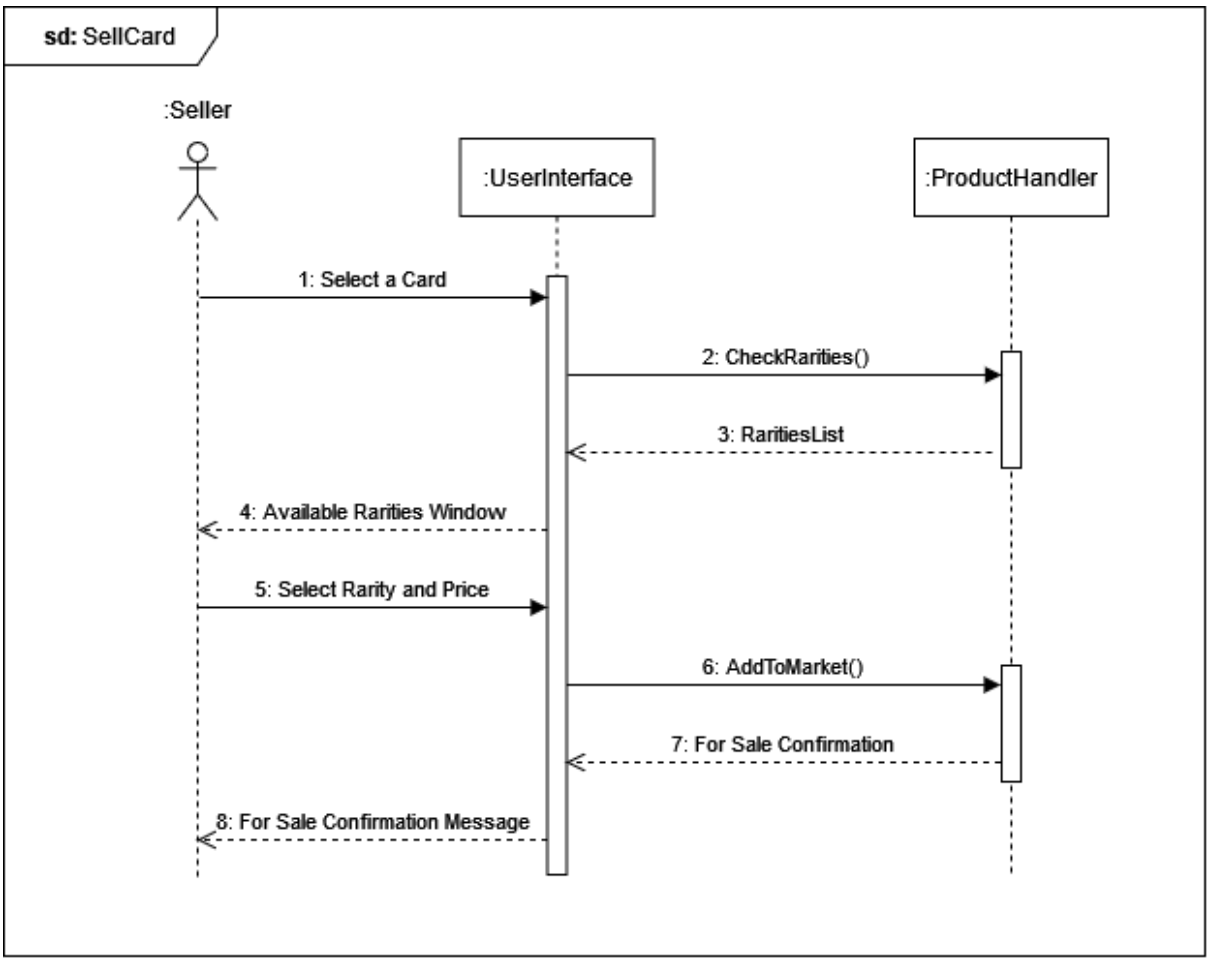


Diagramma sequenze 10: SellCard

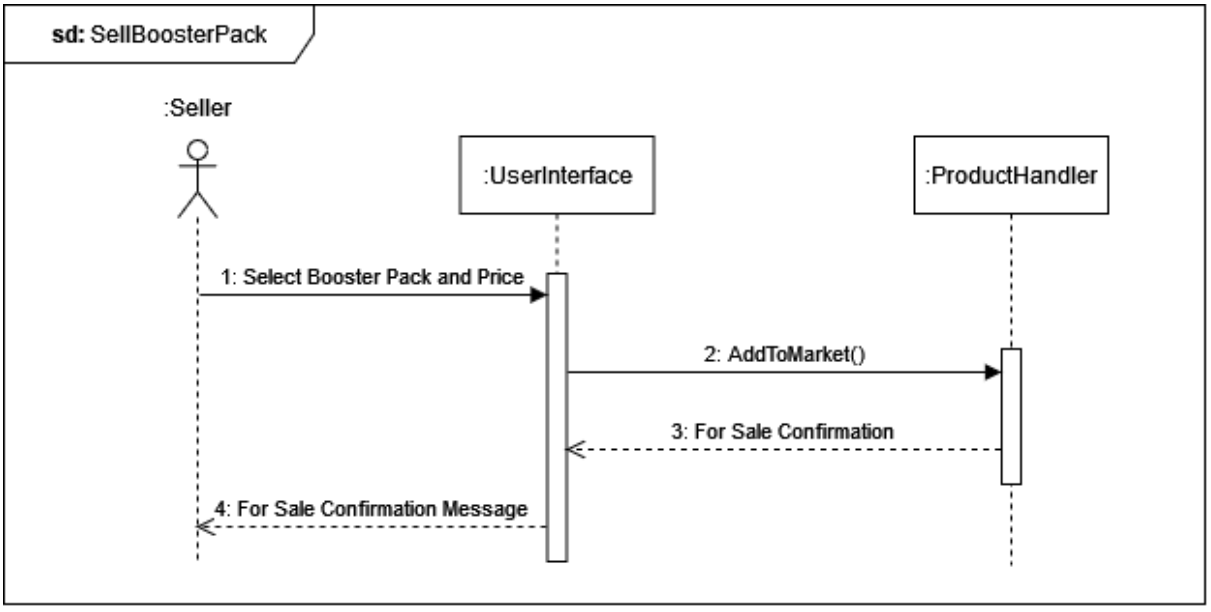


Diagramma sequenze 11: SellBoosterPack

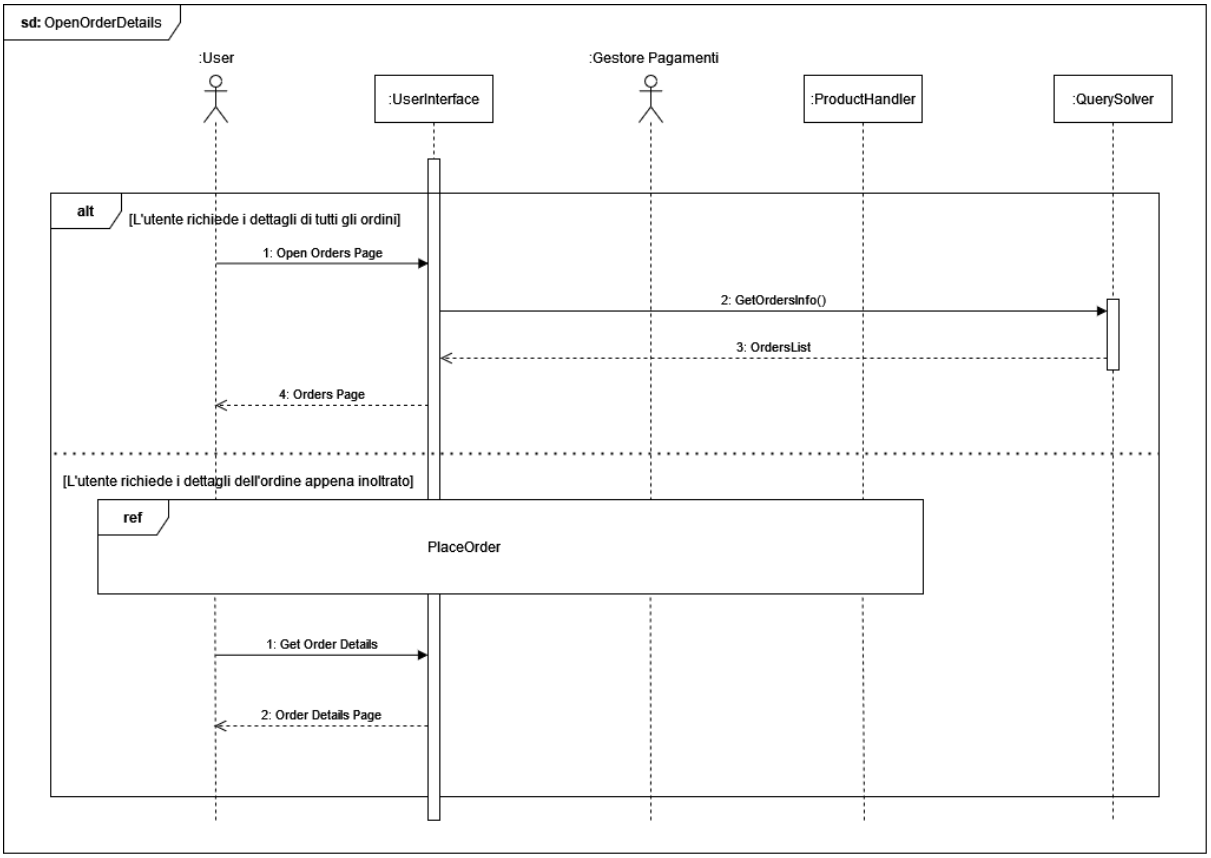


Diagramma sequenze 12: OpenOrderDetails

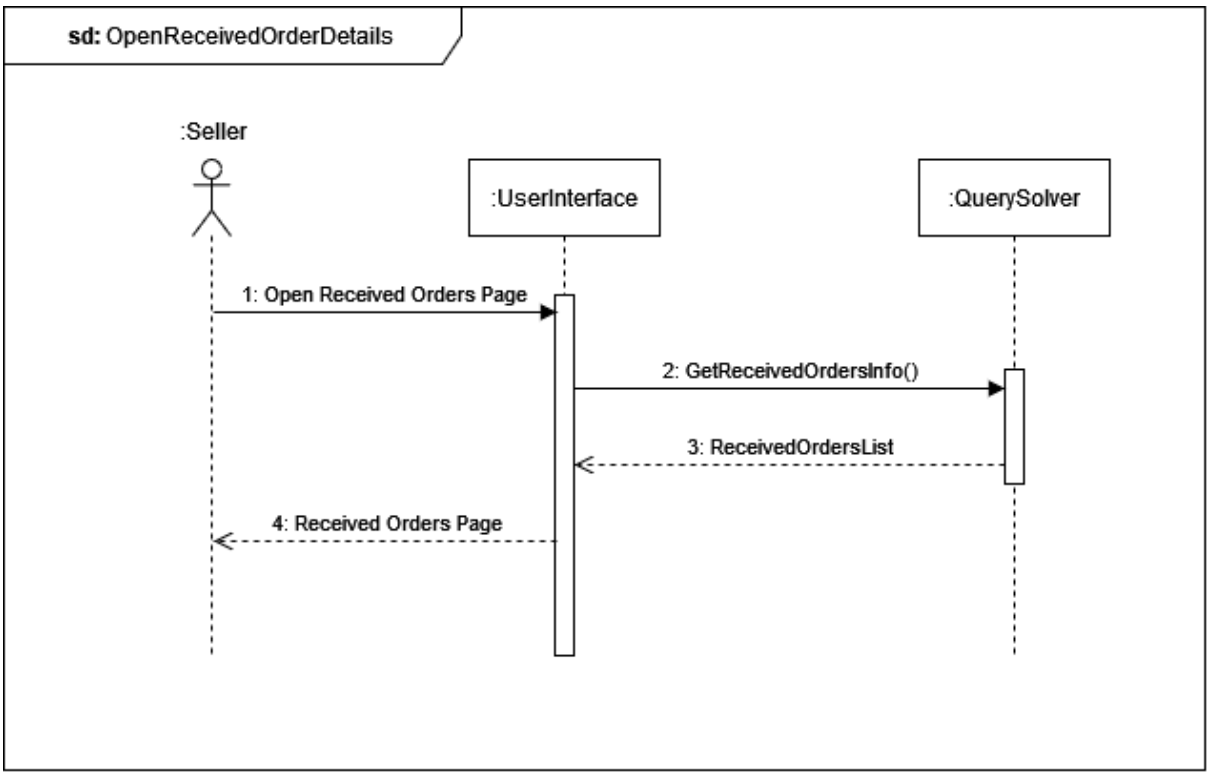


Diagramma sequenze 13: OpenReceivedOrderDetails

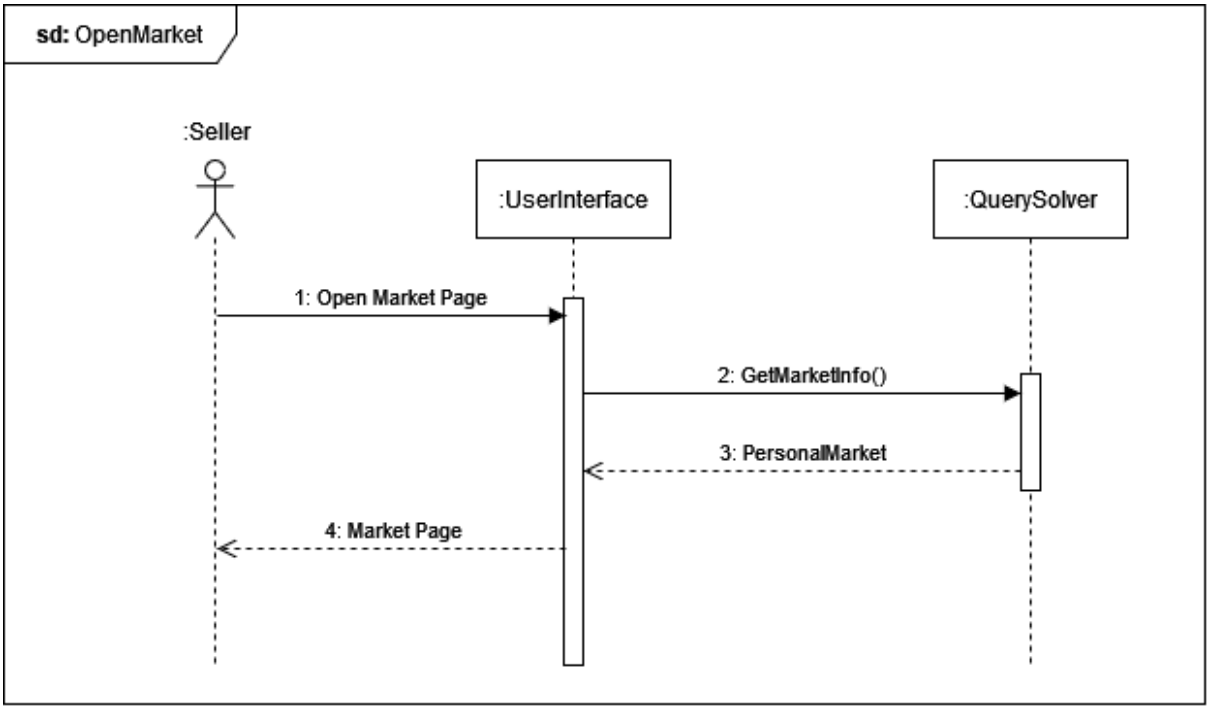


Diagramma sequenze 14: OpenMarket

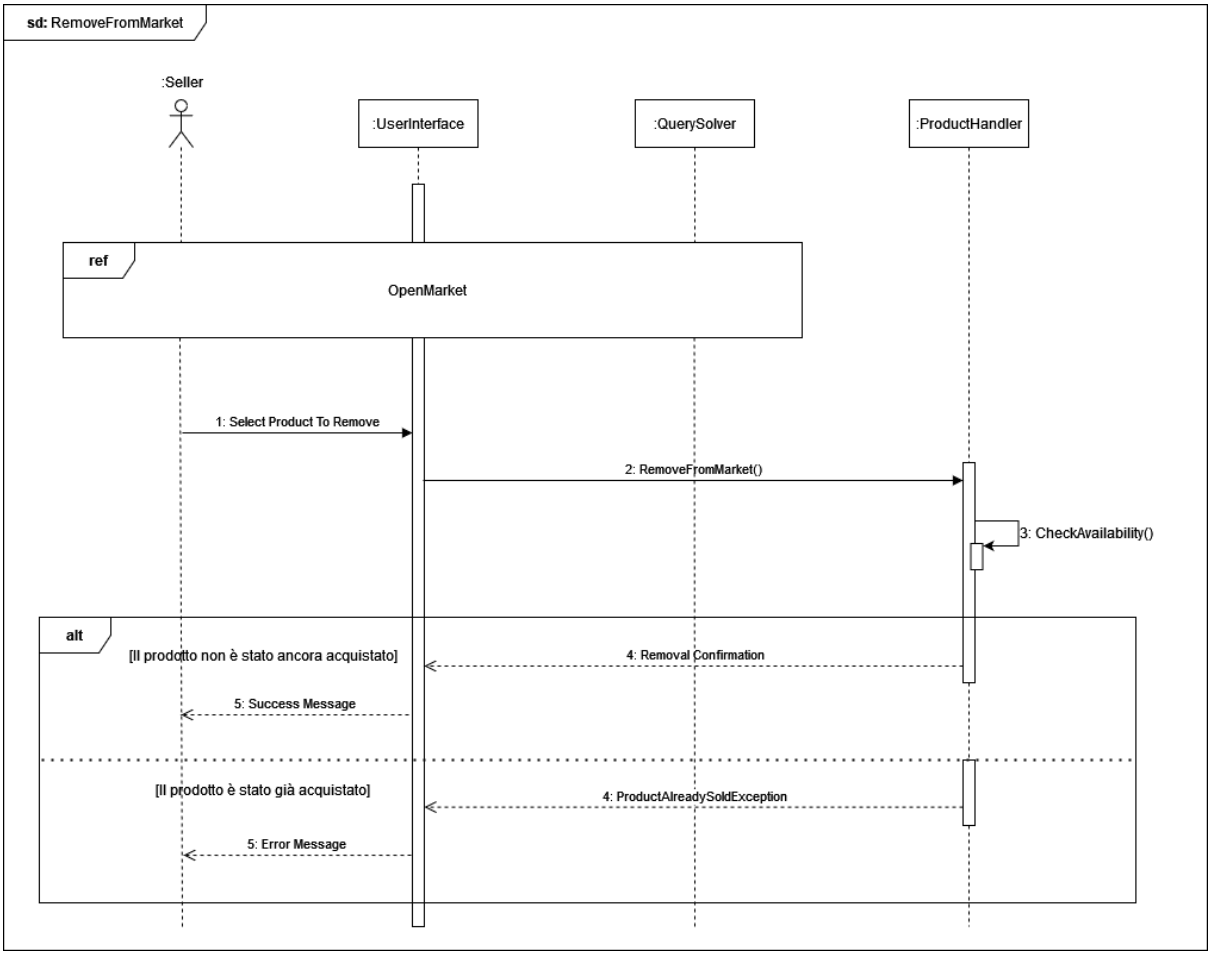


Diagramma sequenze 15: RemoveFromMarket

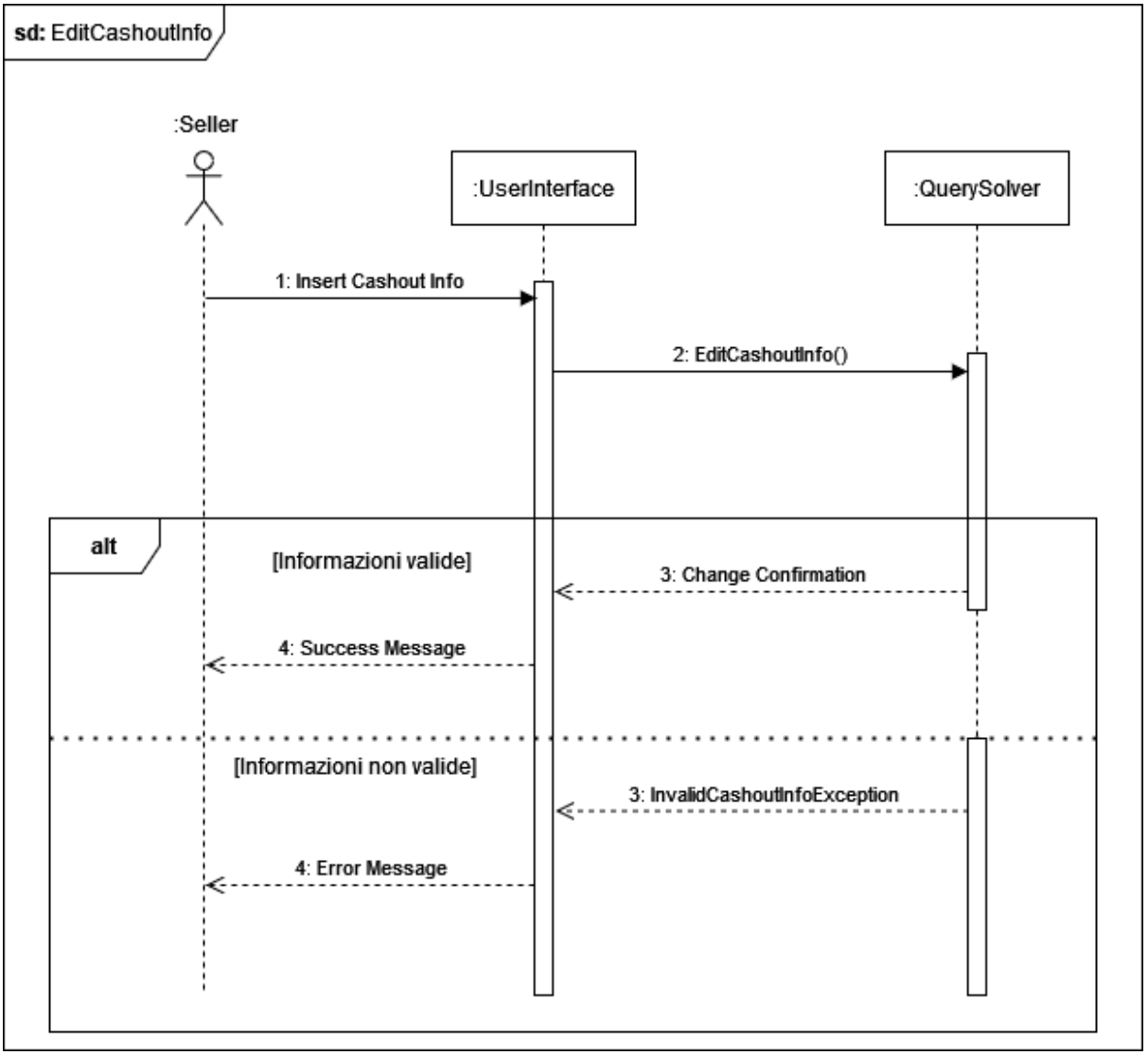


Diagramma sequenze 16: EditCashoutInfo

8 Diagrammi delle attività

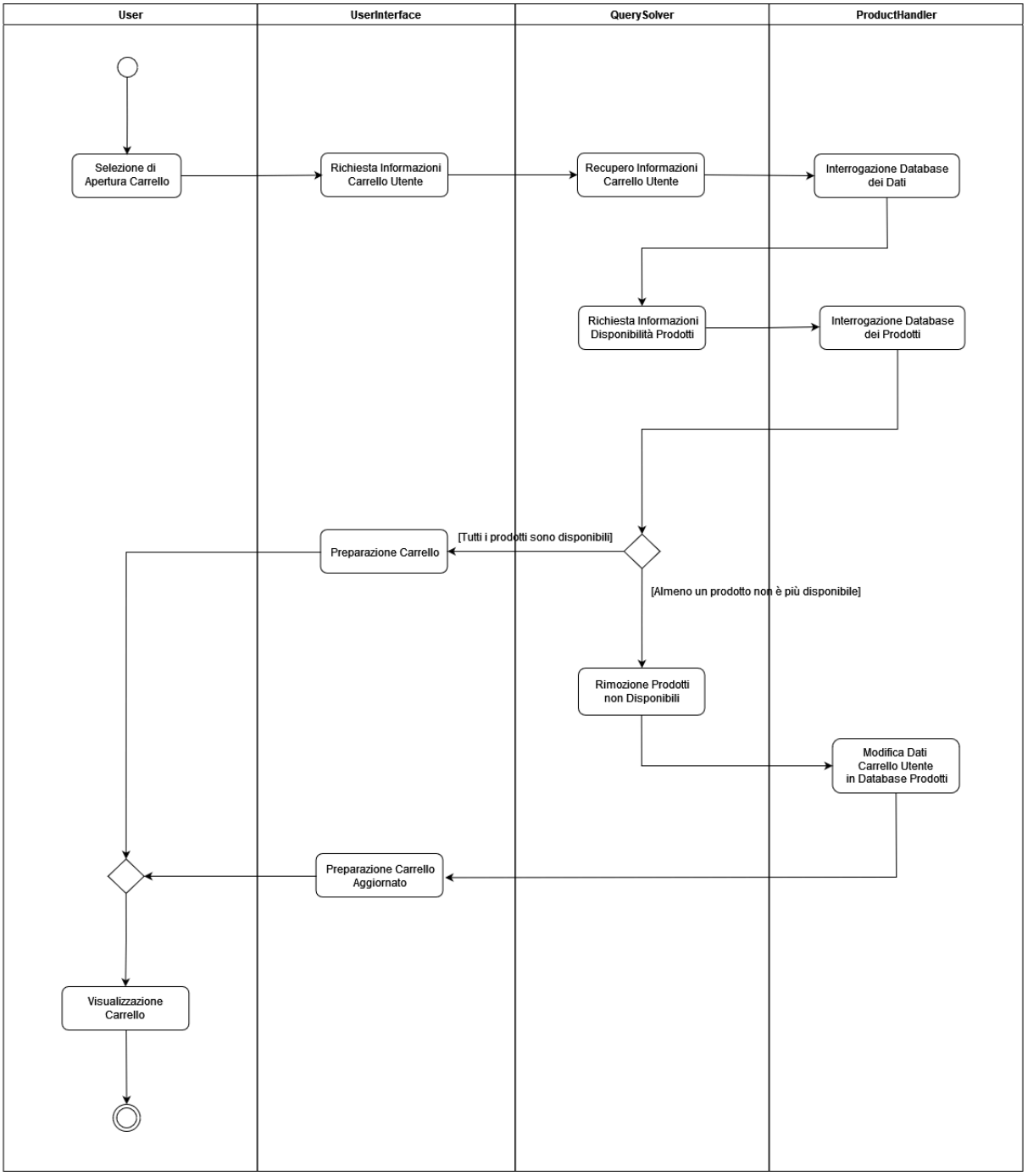


Diagramma delle attività 1: Apertura Carrello

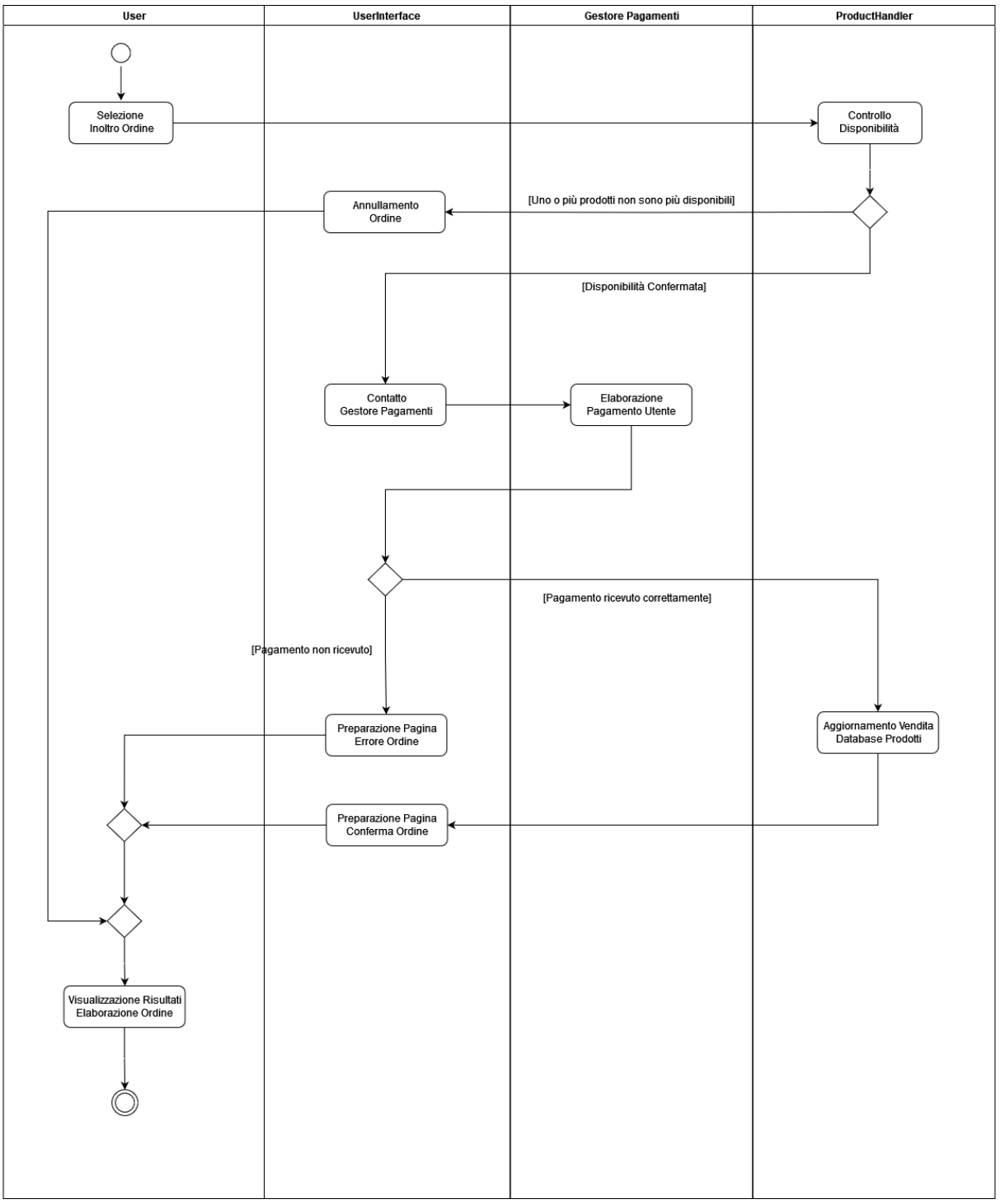


Diagramma delle attività 2: Inoltro Ordine

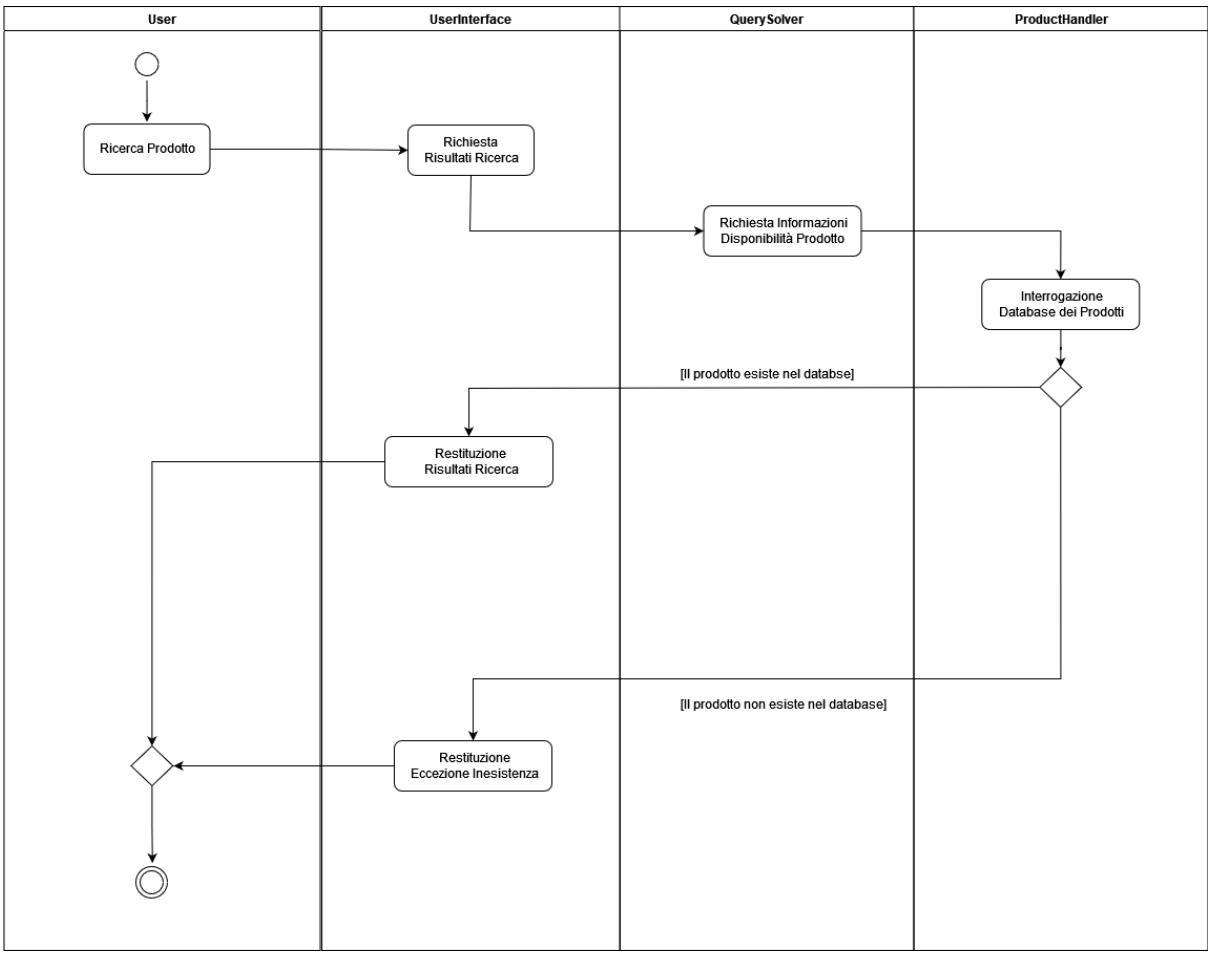
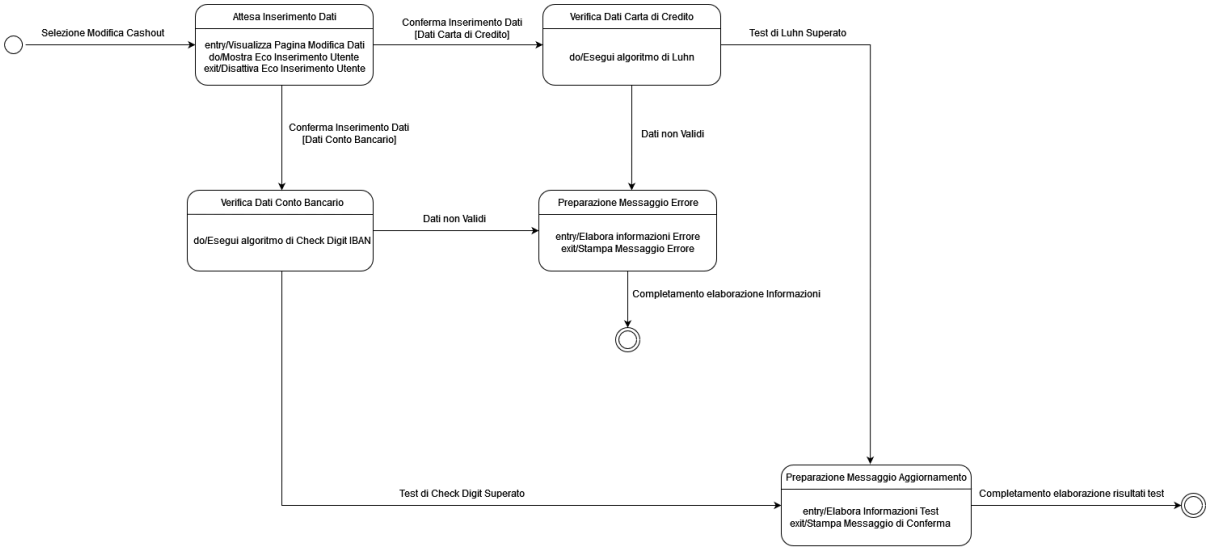


Diagramma delle attività 3: Ricerca Prodotto

9 Diagramma delle transizioni di stato



Il diagramma descrive le transizioni di stato del sistema durante la fase di modifica di informazioni di Cashout

10 Gherkin Test Cases

```
Feature: AddToCart

  The user chooses a product to add to the cart.
  The product should not be added to the cart if it is not available.
  The user shall be notified in case the product is not available.

Scenario: User wants to add a product to their cart
Given user can select a product to add to the cart
  And the product is available
When user selects the intended product
Then the product should be added to the cart
```

Test Case 1: AddToCart Test Case

```
Feature: RemoveFromCart

  The user chooses a product to remove from their cart.
  The product should be removed even if it is not available anymore.

Scenario: User wants to remove a product from their cart
Given user has opened their cart
When user selects the intended product to remove
Then the product should be removed from the user's cart
```

Test Case 2: RemoveFromCart Test Case

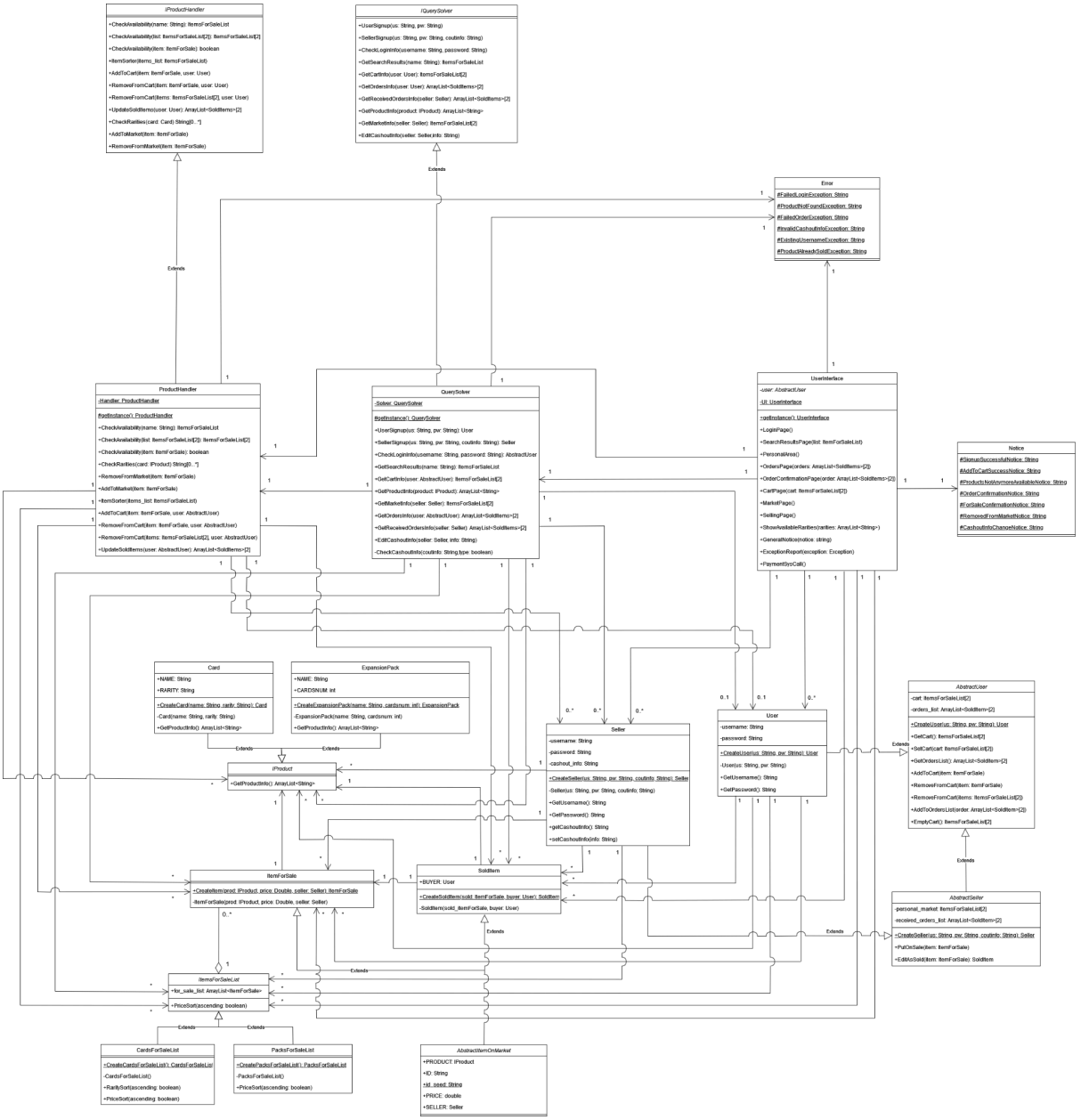
```
Feature: RemoveNotAvailableProducts

  The system selects the products which are not available anymore.
  The system will automatically remove all the unavailable products and notify the user.

Scenario: The availability check has failed for some products
Given user has requested to open their cart or has selected to place an order
When one or more products are not available anymore
Then the system should remove those products from the user's cart
  And the system should notify the user of the changes
```

Test Case 3: RemoveNotAvailableProducts Test Case

11 Diagramma delle classi di design



11.1. Architectural Pattern

11.1.1 Client-Server

•Un'architettura di tipo client-server si è rivelata essenziale per la gestione di dati che dovranno essere disponibili contemporaneamente a tutti gli utilizzatori del prodotto. Il client performerà tutte le operazioni di preparazione e gestione dei risultati di una query, mentre il server risponderà alle richieste client interrogando i database dei dati e dei prodotti. La comunicazione avverrà attraverso connessione HTTPS.

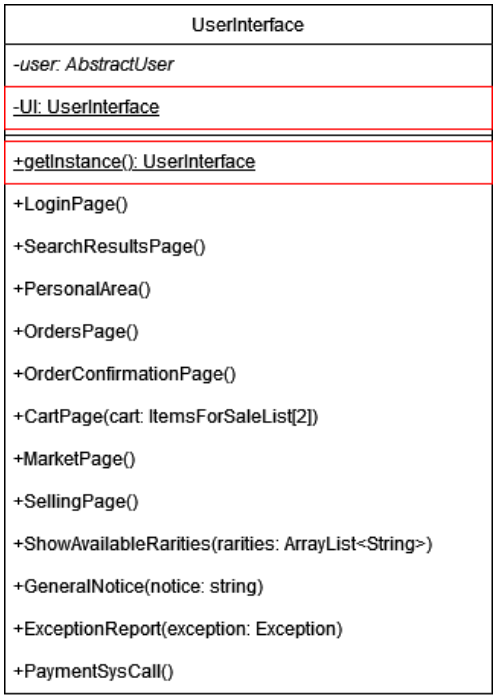
11.1.2 Model-View-Controller

•Per realizzare l'architettura di base del sistema, si è scelto di utilizzare il pattern architetturale Model-View-Controller. La scelta consentirà la facile suddivisione del software in una logica esecutiva, presentativa e di business.
Il cuore esecutivo del programma sarà composto dalle classi *QuerySolver* e *ProductHandler*, che gestiranno tutte le query da inviare al server, rispettivamente inerenti ai dati sugli utenti e sui prodotti. Il server interagirà con i database dei dati e dei prodotti, che gestiranno la logica di business e di controllo dati.
L'interazione con il controller, infine, avverrà attraverso l'interfaccia grafica gestita dalla classe *UserInterface*, che lavorerà sulla logica presentativa.
In conclusione: il server e le classi di gestione query rappresentano il controller, i database dei dati e dei prodotti rappresentano il model, mentre l'interfaccia grafica rappresenta la view.

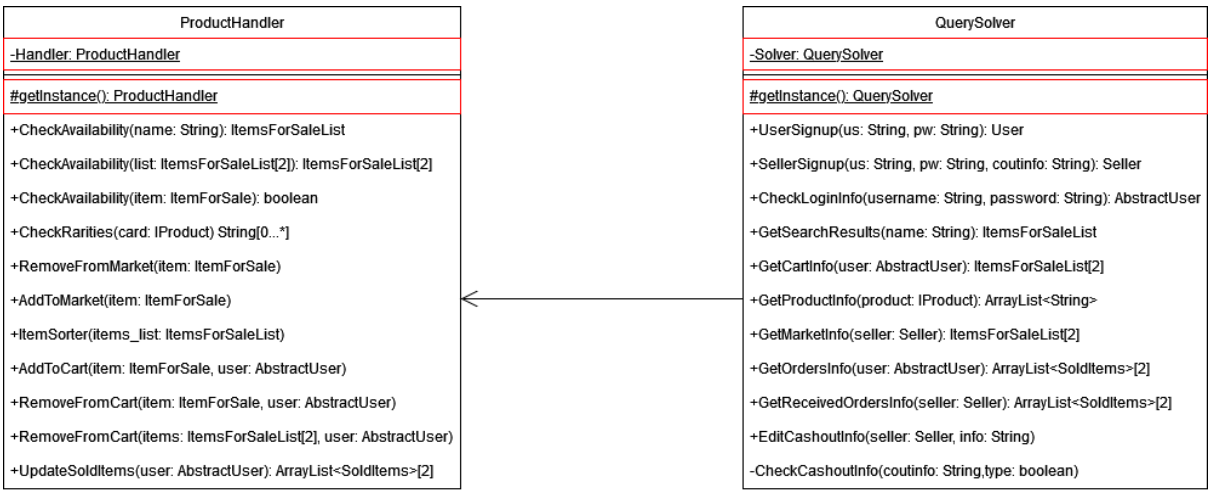
11.2. Design Pattern

11.2.1 Singleton

•Per la realizzazione del sistema di vendita, è stato scelto di basare la parte *View-Controller* del sistema sul pattern *Singleton.* I motivi sono molteplici: in primis, si vuole evitare uno spreco di risorse che richiederebbe di istanziare un oggetto di tipo *QuerySolver* o *ProductHandler* ogni volta che viene ricevuta una richiesta dall'utente, favorendo così un uso ottimizzato della memoria dal lato client. Per lo stesso motivo, è stato deciso di rendere possibile l'accesso ad un'unica istanza anche per la classe *UserInterface*, che gestirà la preparazione e l'aggiornamento dell'interfaccia sulla base delle informazioni fornite dal *Controller*.



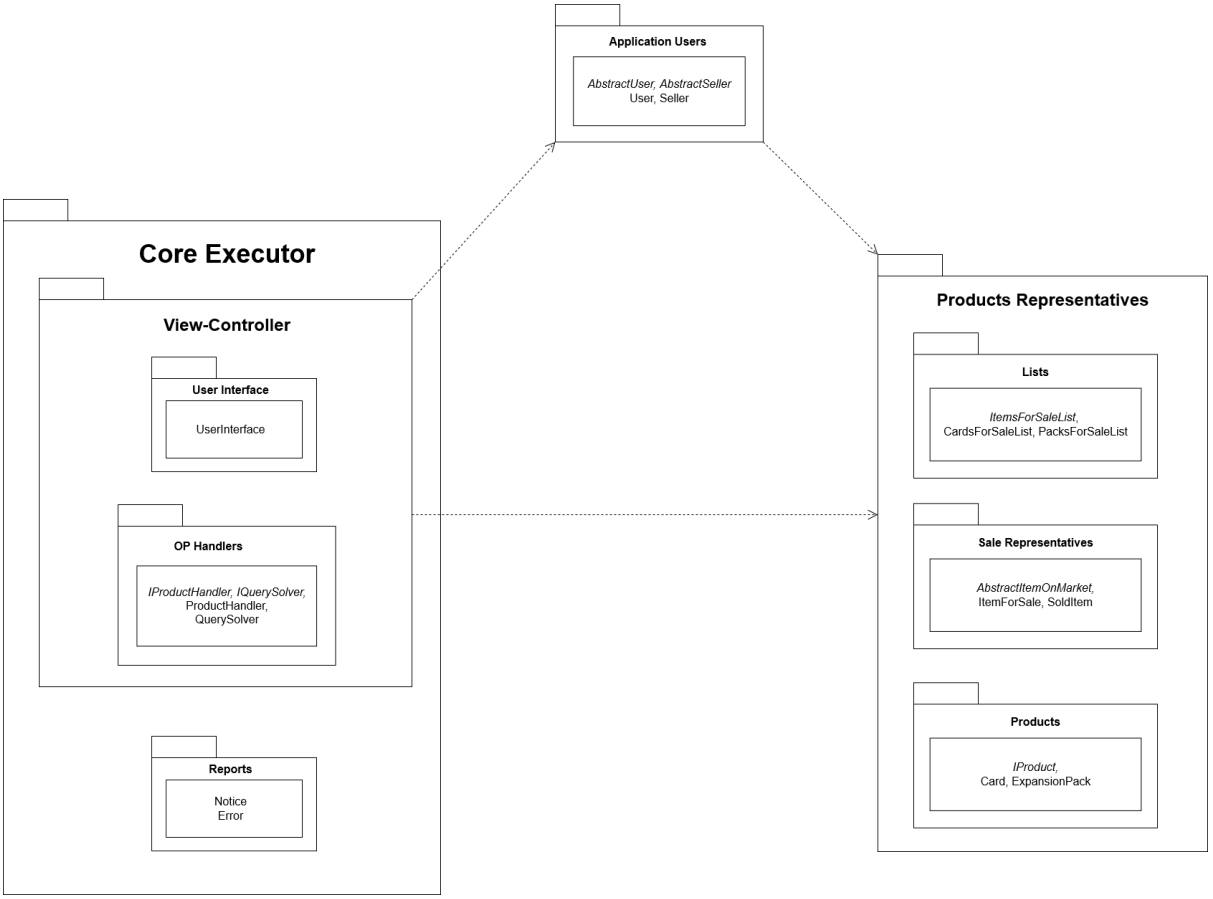
Pattern 1: Singleton - UserInterface



Pattern 1: Singleton - QuerySolver / ProductHandler

·Si noti che per garantire un accesso protetto alle istanze, il metodo GetInstance ha visibilità *Protected*: solo le classi che appartengono al blocco *View-Controller* (*UserInterface*, *QuerySolver*, *ProductHandler*) possono accedere alle istanze di ciascuna delle altre.

12 Diagramma dei package di design



13 Diagrammi delle sequenze di design

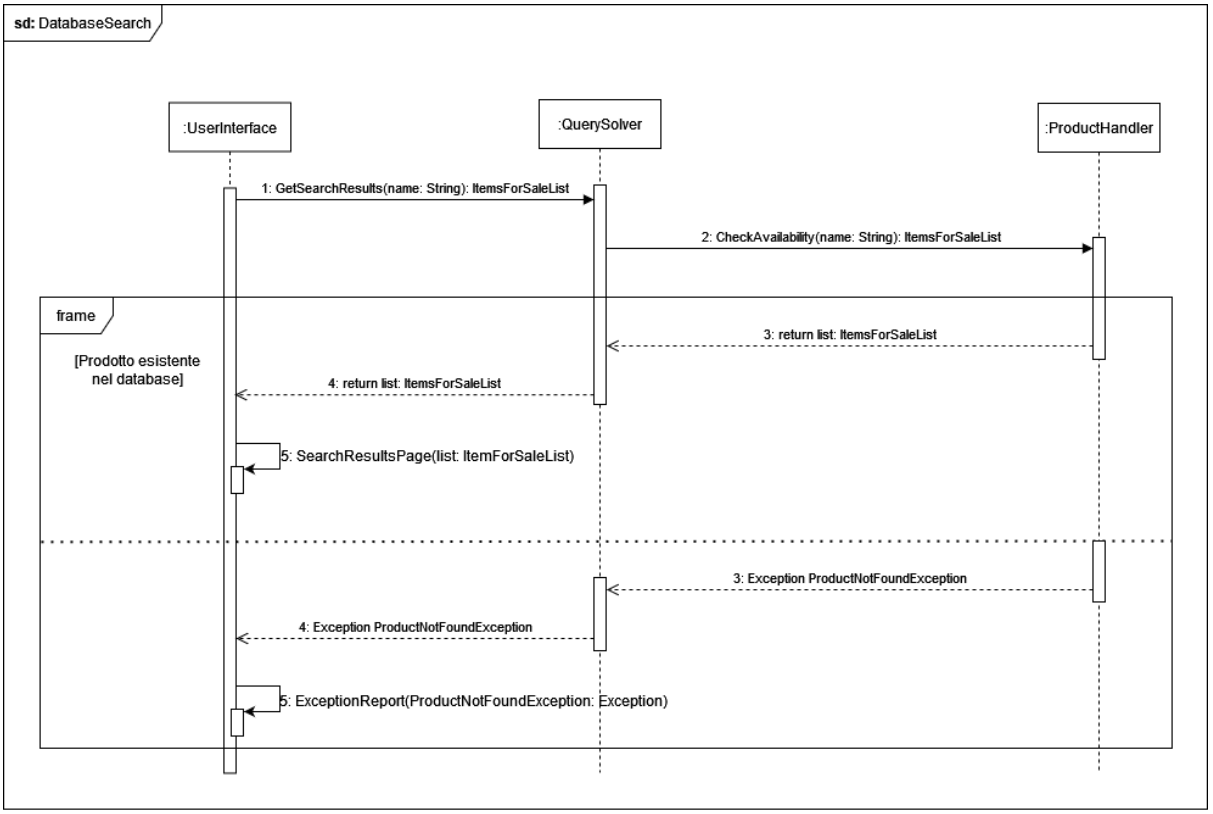


Diagramma sequenze 1: DatabaseSearch

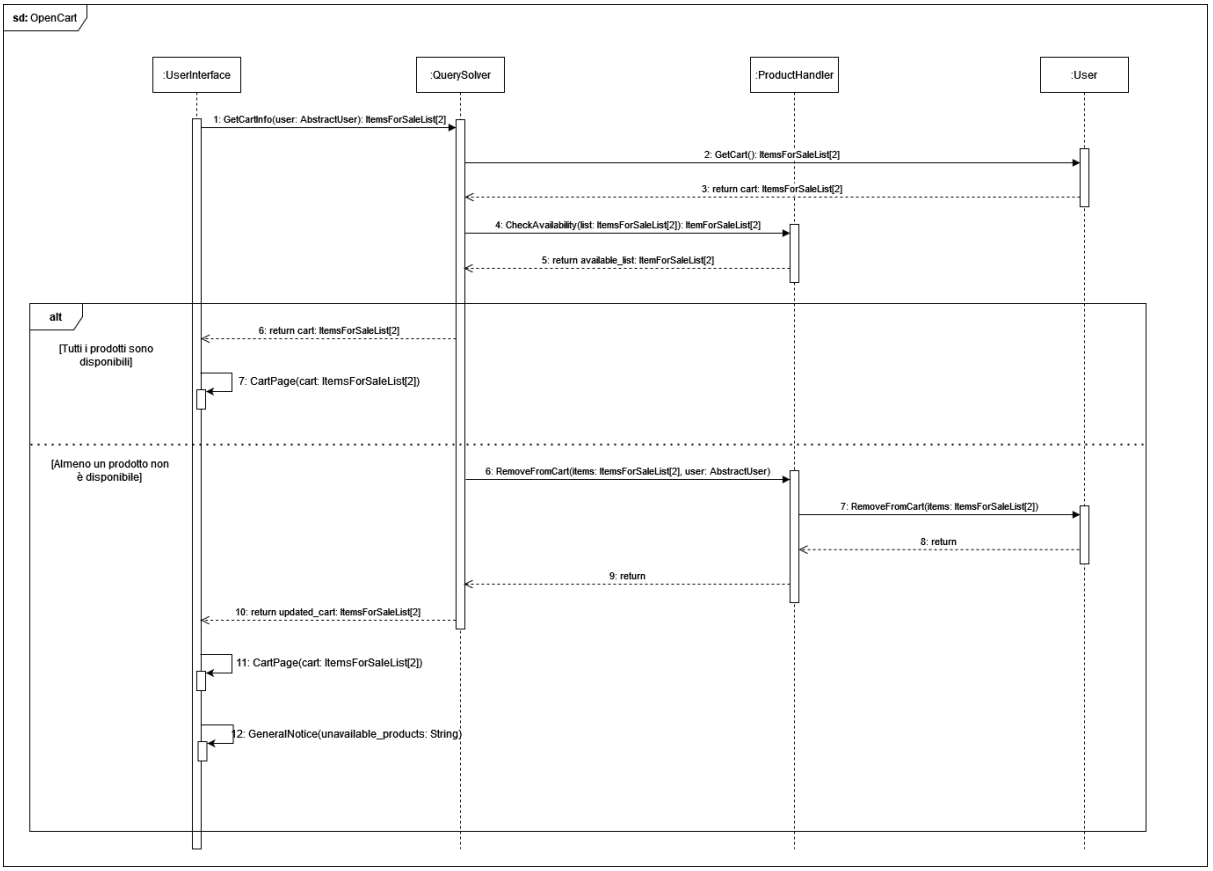


Diagramma sequenze 2: OpenCart

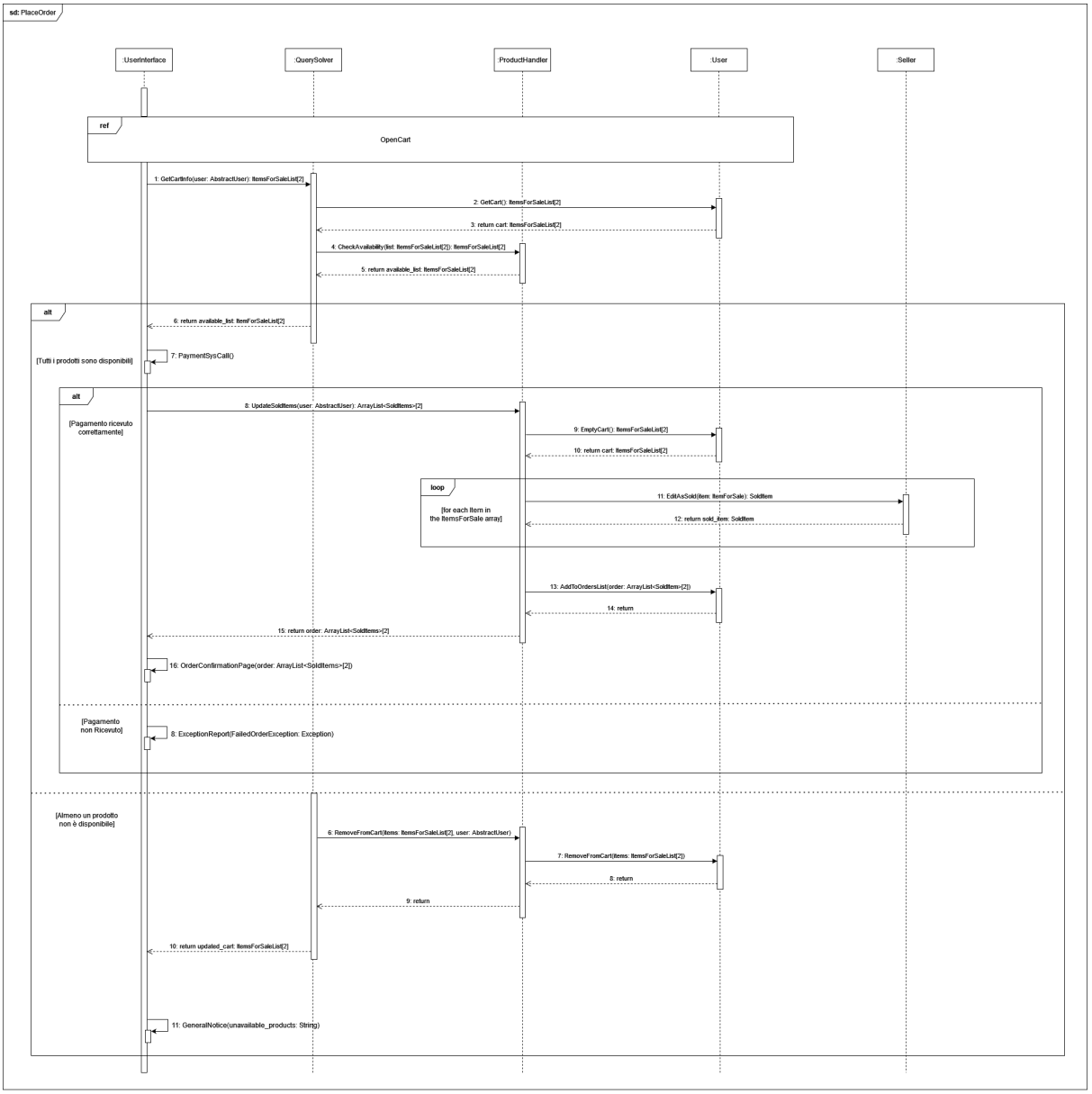
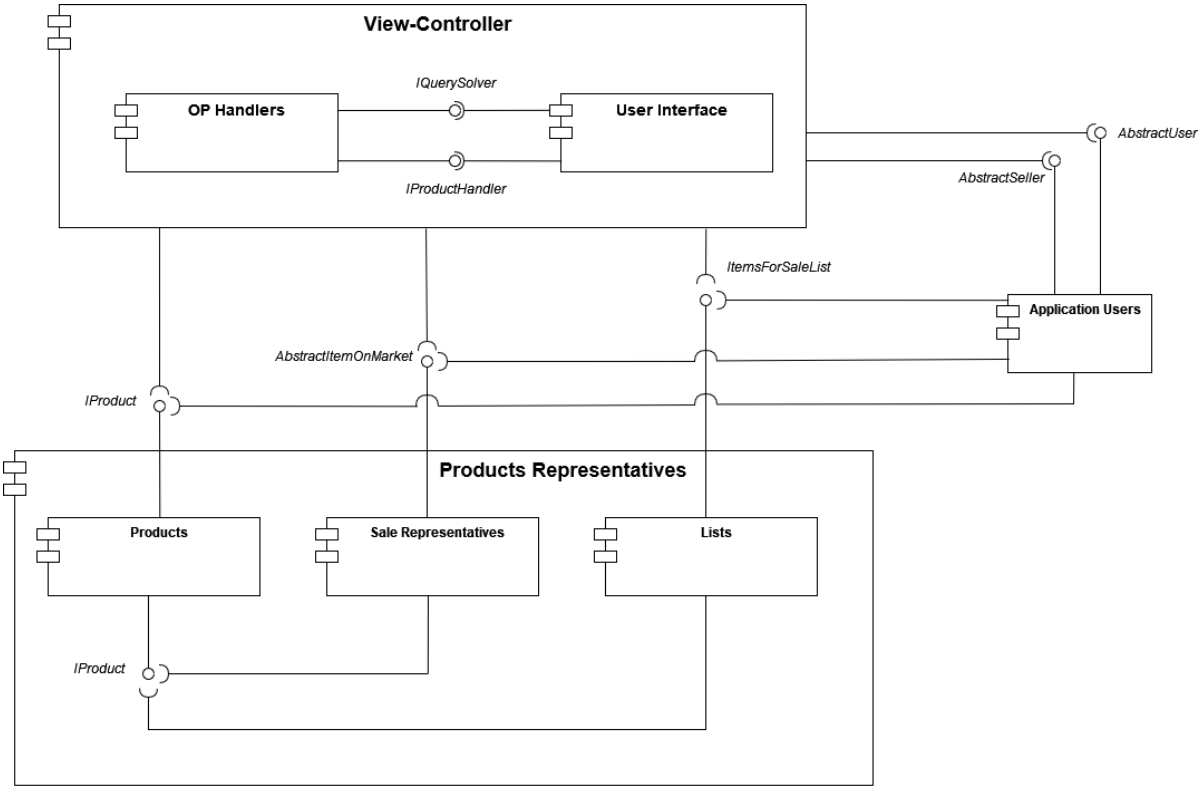
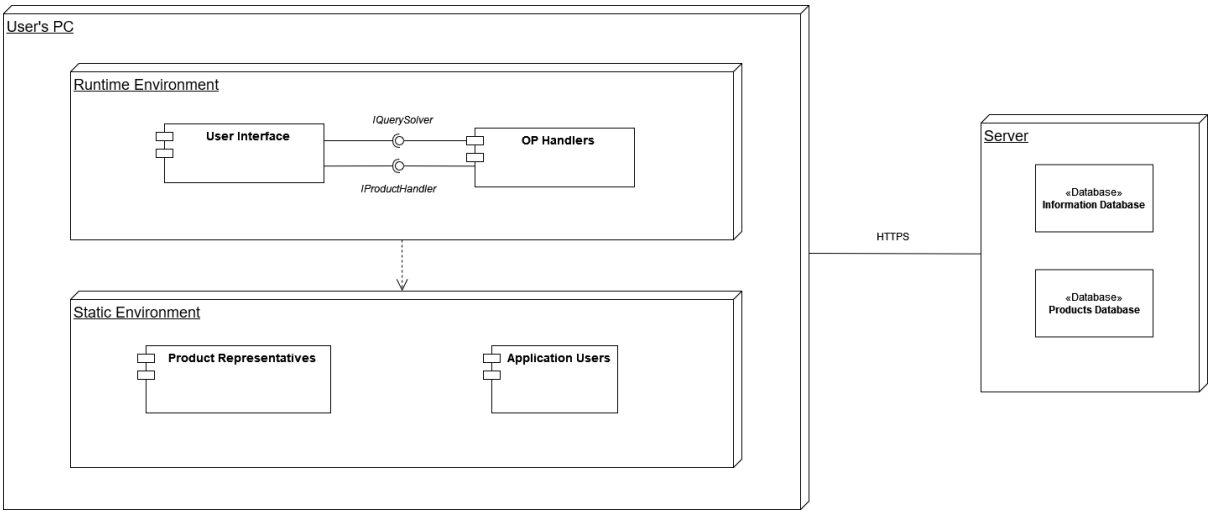


Diagramma sequenze 3: PlaceOrder

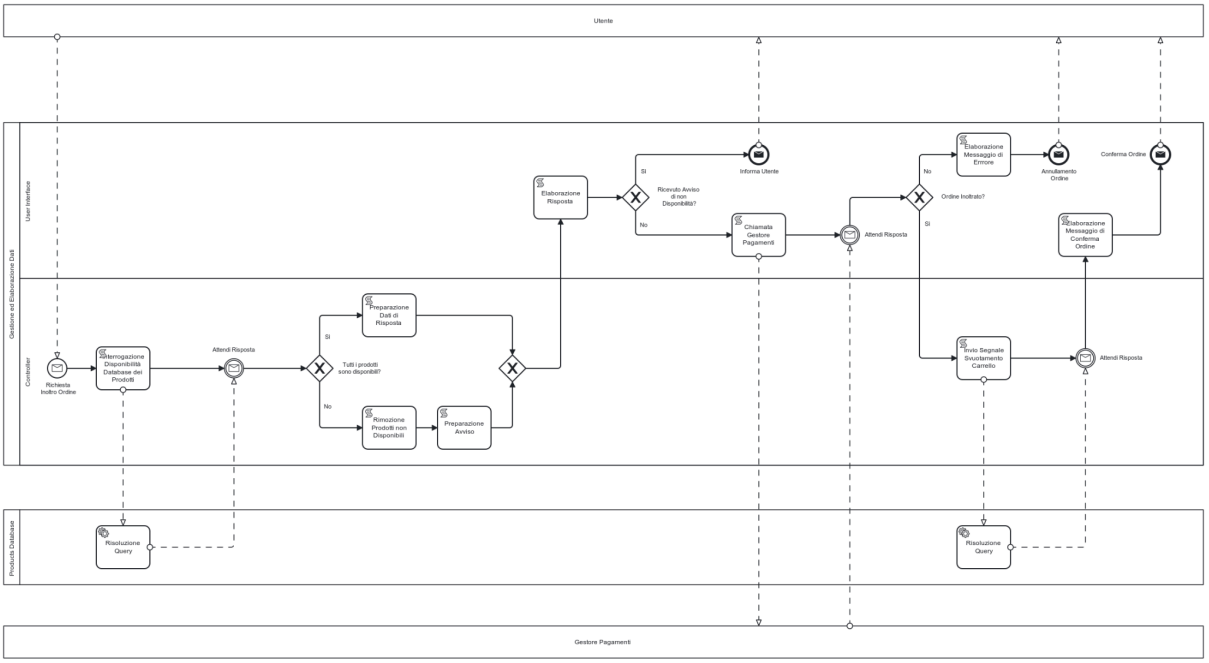
14 Diagramma delle componenti



15 Diagramma di deployment



16 Diagramma BPMN



Rappresentazione delle interazioni con il sistema

durante l’operazione di inoltro ordine

Glossario

Nome:	**Definizione:**
Prodotto	Elemento generico che può essere messo in vendita o acquistato.
Carta da gioco	Una delle due tipologie di prodotto. Caratterizzata da nome e rarità.
Expansion Pack / Booster Pack	Una delle due tipologie di prodotto. Caratterizzata da nome e numero di carte contenute.
Market	Area a disposizione personale del venditore. Contiene tutti i prodotti da lui messi in vendita.