Corso di Laurea Magistrale in Ingegneria Informatica – A.A. 2019-2020

Ingegneria del Software

Pizzeria Panucci

Studenti:

Andrea Calabretta
Antonino Romeo

Prefazione

Il seguente documento presenta una descrizione relativa allo sviluppo dell'applicativo Pizzeria Panucci implementato durante il corso di Ingegneria del Software. L'applicativo è stato implementato con il linguaggio Java sfruttando l'ambiente di sviluppo IntelliJ, invece la progettazione in UML è stata utilizzata con il software Astah Professional.
Vengono qui presentate esclusivamente le versioni definitive di ciascun elaborato ottenute al termine di tutte le fasi di progettazione. Le versioni intermedie sono visionabili nelle corrispondenti sottocartelle della documentazione.
Per concludere vengono descritte le fasi conclusive di testing e di persistenza del programma.

Indice

1. Ideazione e analisi dei requisiti

1.1. Introduzione

L'obiettivo della fase di ideazione è quello di effettuare un'indagine per avere un'idea generale sul progetto da realizzare, sulla sua fattibilità tenendo in considerazione del tempo e delle risorse necessarie.
La presente fase di ideazione si compone dei seguenti documenti: Modello dei Casi d'Uso, Documento di Visione, Regole di Business e Glossario.

1.2. Requisiti

Il titolare della pizzeria "Panucci Pizza" sita in Belpasso richiede la realizzazione di un software per la gestione della sua attività commerciale ed in particolare il software deve gestire i clienti e le loro ordinazioni, permettendo agli stessi di modificare eventualmente le pizze selezionate con gli ingredienti disponibili nel menù.
L'applicativo deve rappresentare uno strumento utile al titolare per la gestione della sua attività e permettere al cliente di effettuare ordinazioni in maniera flessibile.
In particolare:

- ✓ il titolare deve poter gestire le informazioni dei clienti e tenere traccia delle ordinazioni da loro effettuate.
- ✓ Il titolare deve poter gestire e consultare l'archivio delle pizze, degli ingredienti e delle vendite effettuate presso la sua pizzeria.
- ✓ Il titolare deve poter gestire politiche di sconto.
- ✓ Il cliente, una volta registrato, può eseguire le seguenti:
    a. effettuare ordinazioni. Le informazioni riguardanti l'acquisto saranno riportare su una cartella digitale.
    b. comporre la propria pizza scegliendo tra gli ingredienti disponibili.
    c. pagare con carta di Credito, di Debito o Paypal.

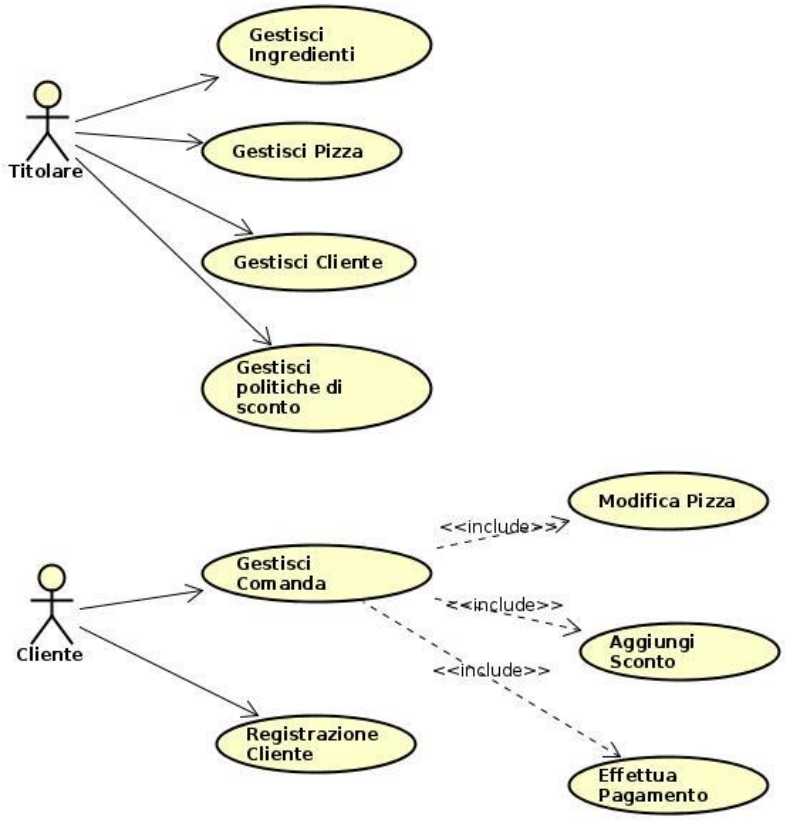
- ✓ La cartella digitale che registra l'acquisto effettuato deve tenere traccia di:
    a. Dati anagrafici del cliente
    b. Descrizione dell'ordine
- ✓ In qualsiasi momento il cliente può:
    a. Eliminare la registrazione dal sistema.

1.3. Obiettivi e casi d'uso

Analizzando i requisiti riportati nel paragrafo precedente, abbiamo individuato gli attori che utilizzano il sistema e i loro rispettivi obiettivi che descriveremo nei casi d'uso principali.

Attore	Obiettivo	Caso d'uso
Cliente	Effettuare comande e relativi pagamenti	UC1: Gestisci comanda
Cliente	Modificare gli ingredienti della pizza	UC2: Modifica Pizza
Titolare	Gestire l'inserimento di un ingrediente	UC3: Gestisci Ingrediente
Titolare	Gestire l'inserimento di una pizza nel sistema	UC4: Gestisci pizza
Cliente	Registrazione nel sistema usando credenziali	UC5: Registrazione cliente
Titolare	Inserire, ricercare, modificare i dati relativi ad un cliente	UC6: Gestisci Cliente (CRUD, Ricerca Cliente)
Cliente	Applicare uno sconto sul prezzo della comanda	UC7: Aggiungi sconto
Titolare	Selezionare una o più politiche di sconto applicabili e regolarne i parametri	UC8: Gestisci politiche di sconto (CRUD, Crea Promozione)
Cliente	Risoluzione pagamento attraverso i vari metodi disponibili	UC9: Effettua pagamento

I casi d'uso appena descritti si traducono graficamente nei seguenti diagrammi UML:



1.4. Modelli dei casi d'uso

Tra tutti i casi d'uso individuati, si è scelto di fornire una descrizione dettagliata per i casi d'uso Gestisci Comanda e Gestisci Ingrediente.

UC1: Gestisci comanda

Nome del caso d'uso	UC1: Gestisci comanda
Portata	Pizzeria Panucci
Livello	Obiettivo utente
Attore primario	Cliente
Parti interessate e Interessi	- Titolare: vuole che il cliente possa effettuare una comanda in modo corretto ed efficace. - Cliente: vuole effettuare una comanda.
Pre-condizioni	Il Cliente si è autenticato
Garanzia di successo	La comanda viene ordinata correttamente
Scenario principale di successo	1. Il Cliente desidera fare una comanda. 2. Il Sistema mostra le pizze disponibili. 3. Il Cliente seleziona una pizza da aggiungere nella comanda. 4. Il Sistema mostra una scheda di riepilogo per la pizza selezionata. 5. Il Cliente modifica eventualmente la pizza [vedi caso d'uso Modifica Pizza] 6. il cliente conferma la pizza. 7. I passi 4, 5 e 6 sono ripetuti finché il cliente ha ancora pizze da selezionare. 8. Il Cliente conferma la comanda. 9. Il software calcola eventuali sconti da applicare [vedi caso d'uso Aggiungi Sconto]. 10. Il software mostra una scheda di riepilogo contenente le pizze selezionate, il relativo prezzo e il prezzo totale. 11. Il cliente conferma la comanda ed effettua il pagamento.
Estensioni	*a. In qualsiasi momento il sistema fallisce 1. Il Cliente riavvia il software e ripristina lo stato. 2. Il sistema ripristina lo stato precedente. 4a. Il Cliente non accetta la scheda di riepilogo della pizza selezionata 1. Il Cliente annulla l'inserimento della pizza nella comanda 2. Il sistema non prosegue con l'inserimento della pizza nella comanda 5a. il Cliente non accetta la scheda di riepilogo della pizza [vedi caso d'uso Modifica Pizza] 10a. il Cliente non accetta la scheda di riepilogo della comanda 1. Il Cliente annulla la comanda 2. Il sistema non prosegue con l'effettuazione della comanda 10b. Il Cliente non è soddisfatto di un prodotto 1. Il Cliente rimuove il prodotto dalla comanda 2. Il Sistema calcola il totale corrispondente

	11a. Il cliente vuole effettuare il pagamento mediante pagamento elettronico [vedi caso d'uso Effettua pagamento]

UC2: Modifica Pizza

Nome del caso d'uso	UC2: Modifica Pizza
Portata	Pizzeria Panucci
Livello	Obiettivo utente
Attore primario	Titolare
Parti interessate e Interessi	- Cliente: vuole modificare una pizza secondo il suo gradimento scegliendo tra le opzioni disponibili offerte dalla pizzeria.
Pre-condizioni	Il cliente sta effettuando una comanda
Garanzia di successo	La modifica della pizza si conclude correttamente
Scenario principale di successo	1. Il Cliente seleziona l'attività Modifica pizza 1. Il Sistema ritorna l'elenco degli ingredienti disponibili 2. Il Cliente seleziona l'ingrediente desiderato 3. Il passo 3 viene ripetuto fino alla conferma del cliente di aver finito 4. Il Sistema mostra una scheda di riepilogo 5. Il Cliente conferma la pizza
Estensioni	*a. In qualsiasi momento il sistema fallisce 1. Il sistema ripristina lo stato precedente 5a. il Cliente non accetta la scheda di riepilogo 1. Il Cliente annulla la modifica effettuata 2. Il sistema non prosegue con l'inserimento nella comanda
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Legata all'affluenza dei clienti
Varie	

UC3: Gestisci Ingrediente

Nome del caso d'uso	UC3: Gestisci Ingrediente
Portata	Pizzeria Panucci
Livello	Obiettivo amministratore
Attore primario	Titolare
Parti interessate e Interessi	- Titolare: vuole gestire l'intero archivio degli ingredienti disponibili per comporre le pizze e per eventuali modifiche da parte del cliente.
Pre-condizioni	L'amministratore si è autenticato
Garanzia di successo	L'inserimento del nuovo ingrediente si conclude correttamente
Scenario principale di successo	1. Il Titolare richiede al sistema l'inserimento di un nuovo ingrediente. 2. Il Titolare inserisce i dati di un nuovo ingrediente. 3. Il Sistema mostra una scheda di riepilogo.

	4. Il Titolare conferma l'inserimento dell'ingrediente.
Estensioni	*a. In qualsiasi momento il sistema fallisce 1. Il Titolare riavvia il software e ripristina lo stato. 2. Il sistema ripristina lo stato precedente. 3a. il Titolare non accetta la scheda di riepilogo 1. Il Titolare annulla l'inserimento. 2. Il sistema non prosegue con l'inserimento.
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	giornaliera
Varie	

UC4: Gestisci Pizza

Nome del caso d'uso	UC4: Gestisci Pizza
Portata	Pizzeria Panucci
Livello	Obiettivo amministratore
Attore primario	Titolare
Parti interessate e Interessi	- Titolare: vuole gestire l'intero archivio di pizze disponibili per il cliente secondo le proprie preferenze
Pre-condizioni	L'amministratore si è autenticato
Garanzia di successo	L'inserimento della nuova pizza si conclude correttamente
Scenario principale di successo	1. Il Titolare seleziona l'attività Gestisci pizza. 2. Il Sistema mostra la lista degli ingredienti disponibili. 3. Il Titolare seleziona l'ingrediente da aggiungere ad una nuova pizza. 4. Il passo 3 si ripete fino a che ci sono ingredienti da inserire. 5. Il Sistema mostra una scheda di riepilogo della pizza. 6. Il Titolare conferma l'inserimento assegnando un nome alla pizza
Estensioni	*a. In qualsiasi momento il sistema fallisce 1. Il Titolare riavvia il software e ripristina lo stato 2. Il sistema ripristina lo stato precedente 5a. il Titolare non accetta la scheda di riepilogo 1. Il Titolare annulla l'inserimento 2. Il sistema non prosegue con l'inserimento della pizza
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	mensile
Varie	

UC5: Registrazione Cliente

Nome del caso d'uso	UC5: Registrazione Cliente
Portata	Pizzeria Panucci
Livello	Obiettivo utente
Attore primario	Cliente
Parti interessate e Interessi	- Cliente: vuole registrarsi nel sistema affinché possa effettuare comande e relativi pagamenti.
Pre-condizioni	
Garanzia di successo	Consistenza dei dati in output
Scenario principale di successo	1. Il Cliente seleziona Registrazione 2. Il Sistema richiede le informazioni relative 3. Il Cliente inserisce i suoi dati
Estensioni	1a. Il Cliente richiede al sistema la modifica delle informazioni 1. Il sistema richiede al Cliente i dati da aggiornare. 2. Il Cliente aggiorna i campi di interesse e conferma 1c. Il Cliente richiede al sistema di eliminare la sua registrazione 1. Il Sistema chiede conferma dell'operazione di eliminazione 2. Il Cliente conferma l'eliminazione 3. il Cliente viene eliminato
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Non stimabile, non costante, legata all'affluenza di nuovi clienti e dei vecchi che aggiornano i loro dati
Varie	

UC6: Gestisci Cliente, CRUD

Nome del caso d'uso	UC6: Gestisci Cliente
Portata	Pizzeria Panucci
Livello	Obiettivo amministratore
Attore primario	Titolare
Parti interessate e interessi	- Titolare: vuole tenere traccia dei suoi clienti
Pre-condizioni	
Garanzia di successo	I dati in uscita sono consistenti
Scenario principale di successo	1. Il titolare richiede al sistema di inserire un nuovo cliente 2. Il sistema richiede le informazioni relative al cliente 3. Il titolare inserisce i dati del cliente
Estensioni	1a. il titolare richiede al sistema la modifica di un cliente 1. Il sistema richiede al titolare i dati da aggiornare 2. Il titolare aggiorna i campi di interesse e conferma 1b. il titolare richiede al sistema di ricercare un cliente 1. Il sistema richiede di inserire i dati del cliente nei campi di ricerca 2. Il titolare inserisce le informazioni 3. Il sistema mostra il cliente che soddisfa la ricerca 1c. Il titolare richiede al sistema di eliminare un cliente 1. Il sistema chiede quale utente eliminare 2. Il titolare seleziona l'utente

	3. L’utente viene eliminato
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Non stimabile, legata all’affluenza di nuovi clienti e dei vecchi clienti che modificano i propri dati.
Varie	

UC7: Aggiungi Sconto

Nome del caso d’uso	UC7: Aggiungi sconto
Portata	Pizzeria Panucci
Livello	Sottofunzione
Attore primario	Cliente
Parti interessate e interessi	- Cliente: vuole verificare in modo semplice se è possibile applicare uno sconto sulla sua comanda - Titolare: vuole che gli sconti siano applicati in modo corretto secondo le politiche interne della Pizzeria
Pre-condizioni	C’è una comanda in corso
Garanzia di successo	
Scenario principale di successo	1. Il Cliente conferma di aver finito con l’inserimento di pizze nella comanda 2. Il Sistema verifica la presenza di uno sconto applicabile 3. Il Sistema calcola lo sconto e lo applica sull’importo della comanda
Estensioni	2a. Il sistema non trova sconti applicabili 1. Il sistema non applica sconti
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Ogni volta che viene eseguita una comanda
Varie	

UC8: Gestisci Politiche di sconto, CRUD

Nome del caso d'uso	UC8: Gestisci politiche di sconto
Portata	Pizzeria Panucci
Livello	Obiettivo amministratore
Attore primario	Titolare
Parti interessate e interessi	- Titolare: vuole gestire in maniera semplice la creazione, la modifica, la cancellazione e la ricerca di uno sconto da associare ad una comanda
Pre-condizioni	
Garanzia di successo	I dati in uscita sono consistenti
Scenario principale di successo	1. Il titolare chiede al sistema l'inserimento di uno sconto 2. Il sistema richiede la condizione per applicare lo sconto 3. Il titolare inserisce la condizione 4. Il sistema richiede la percentuale di sconto da applicare 5. Il titolare inserisce la percentuale di sconto
Estensioni	1a. Il titolare richiede di eliminare uno sconto 1. Il sistema chiede al titolare lo sconto da eliminare 2. Il titolare inserisce lo sconto da eliminare 3. Il sistema elimina lo sconto 2a. Il titolare richiede di modificare uno sconto 1. Il sistema richiede al titolare i dati da aggiornare 2. Il titolare aggiorna i dati di interesse
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Bassa
Varie	

UC9: Effettua pagamento

Nome del caso d'uso	UC9: Effettua pagamento
Portata	Pizzeria Panucci
Livello	Sottofunzione
Attore primario	Cliente
Parti interessate e interessi	- Cliente: vuole pagare per via elettronica - Titolare: vuole ricevere i pagamenti per via elettronica
Pre-condizioni	C'è una comanda in corso
Garanzia di successo	Il pagamento è andato a buon fine
Scenario principale di successo	1. Il caso d'uso inizia quando il Cliente conferma la comanda 2. Il Sistema chiede la tipologia di pagamento 3. Il Cliente seleziona la tipologia di pagamento "con carta" 4. Il Cliente inserisce i dati della propria carta 5. Il Sistema verifica i dati immessi dal cliente ed effettua il pagamento
Estensioni	3a. La tipologia di pagamento selezionata è "paypal" 1. Il Cliente inserisce i suoi dati 2. Il sistema verifica i dati ed effettua il pagamento
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Ogni volta che viene eseguita una comanda
Varie	

1.5. Documento di visione

Contemporaneamente alla stesura di questo capitolo è stata realizzata una prima bozza del documento di Visione (vedi Appendice A). Si prevede che nelle successive iterazioni dello sviluppo dell'applicativo, tali informazioni verranno dettagliate ulteriormente.

1.6. Regole di Business

ID	Regola	Modificabilità	Sorgente
R1	Per i clienti che acquistano una comanda di ordine superiore a 40 euro viene applicato uno sconto del 5 sul totale	Media, Il titolare potrebbe voler modificare la percentuale ma non più di due volte all'anno	Politica interna della pizzeria
R2	Se l'ordine è di più di 90 euro, allora si applica uno sconto del 20%	Media, il titolare potrebbe voler modificare la quantità di pizze necessarie per averne una in omaggio	Politica interna della pizzeria

1.7. Specifiche Supplementari

Vincoli hardware e software

- Per eseguire l'applicativo non ci sono particolari requisiti per il sistema operativo purché sia presente la JVM (Java Virtual Machine).
- Per eseguire il pagamento elettronico è necessaria una connessione a Internet.

1.8. Glossario

Vengo riportati i termini più significativi e le loro definizioni:

1. **Sconto:** sconto applicato dal sistema sul prezzo della comanda.
2. **Comanda**: termine riferito all'ordine effettuato dal cliente. Questo implica che il cliente ha già pagato l'ammontare complessivo.
3. **Ingrediente**: termine che si riferisce ai componenti della pizza.
4. **Paypal**: metodo di pagamento elettronico che richiede Username e Password.
5. **Carta di Credito**: metodo di pagamento elettronico che richiede Nome del titolare, numero della carta, cvv e scadenza della carta.
6. **Carta di Debito**: metodo di pagamento elettronico che richiede Nome del titolare, numero della carta, cvv e scadenza della carta.

2. Analisi Orientata agli Oggetti

2.1. Introduzione

Seguendo l'approccio iterativo evolutivo consigliato da UP, la realizzazione dell'applicazione è stata articolata su 4 iterazioni. In questo modo è stato possibile implementare in maniera iterativo il cuore del software applicativo, sono state risolte le problematiche relative ai rischi maggiori ed è stata affrontata un'analisi dei requisiti graduale in modo da limitare al minimo il danno causato da eventuali errori di progettazione e implementazione.
Di seguito vengono mostrate le problematiche prese in considerazione nelle varie iterazioni:

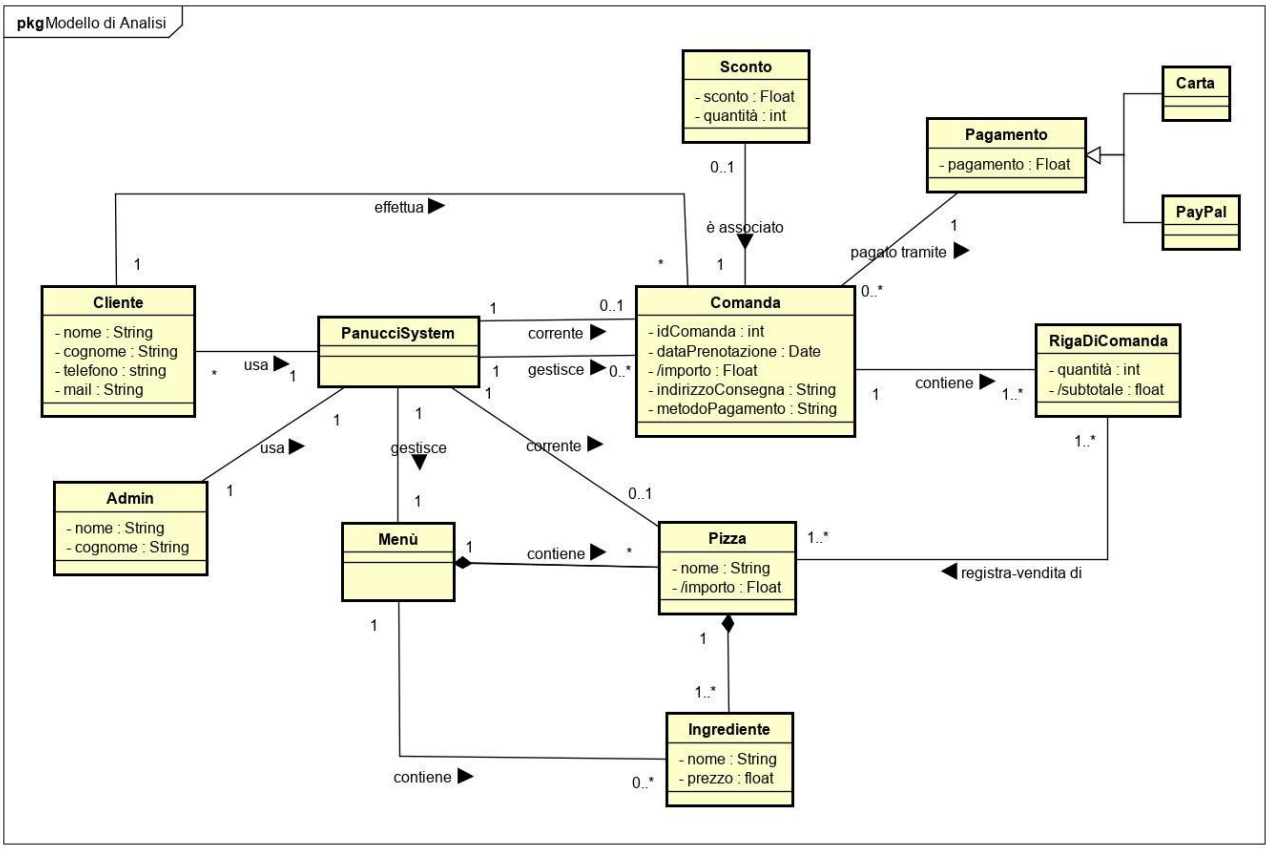
- **Iterazione 1**
  - Implementazione scenario principale UC1: Gestisci Comanda in cui il cliente ordina le pizze desiderate. Essendo necessario per il corretto funzionamento del caso d'uso il database contenente i dati del cliente, degli ingredienti e delle pizze disponibili, sono stati simulati momentaneamente tenendo una lista in memoria caricata all'avvio dell'applicazione. Poiché l'aggiunta di uno sconto e il pagamento rientrano in altri casi d'uso, sono stati tralasciati in questa fase.

- **Iterazione 2**
  - Implementazione scenario principale UC3: Gestisci Ingrediente in cui si prevede che il titolare inserisca un ingrediente per comporre le pizze.
  - Implementazione scenario principale UC4: Gestisci Pizza in cui si prevede che il titolare inserisca una nuova pizza nel menù componendola usando gli ingredienti precedentemente inseriti.
- **Iterazione 3**
  - Implementazione scenario principale di successo UC2: Modifica Pizza, sottofunzione di UC1 in cui si prevede che il cliente possa modificare la pizza selezionata con gli ingredienti disponibili nel menù.
  - Implementazione scenario alternativo UC2 dove il cliente rimuove un ingrediente dalla pizza selezionata.
  - Implementazione scenario principale di successo UC5: Registrazione Cliente, in cui il cliente si registra nel sistema.
- **Iterazione 4**
  - Implementazione scenario principale di successo UC7: Aggiungi sconto, sottofunzione di UC1 in cui viene aggiunto uno sconto nella comanda.
  - Implementazione scenario principale di successo UC9: Effettua Pagamento, sottofunzione di UC1 in cui viene effettuato il pagamento usando servizi di validazione esterni al sistema. Il pagamento effettivo è stato simulato con metodi stub, i quali simulano che il pagamento sia andato a buon fine.

Il passo iniziale di ogni iterazione è stato quello di effettuare un'attenta analisi dei requisiti Orientata agli Oggetti, ovvero un'analisi basata sulla descrizione del dominio da un punto di vista ad oggetti. Per fornire tale descrizione sono stati utilizzati diversi strumenti: Modello di Dominio, Diagramma degli Oggetti, SSD (Squence System Diagram) e Contratti delle operazioni.

2.2. Modello di dominio

La disciplina che in termini di UP si occupa di fornire dettagli sul dominio è la Modellazione del Business, in particolare essa comprende la stesura del Modello di Dominio dove vengono identificati i concetti, gli attributi e le associazioni considerate significative.
Prendendo in considerazione del contributo fornito da ciascuna iterazione, il modello di dominio è il seguente:



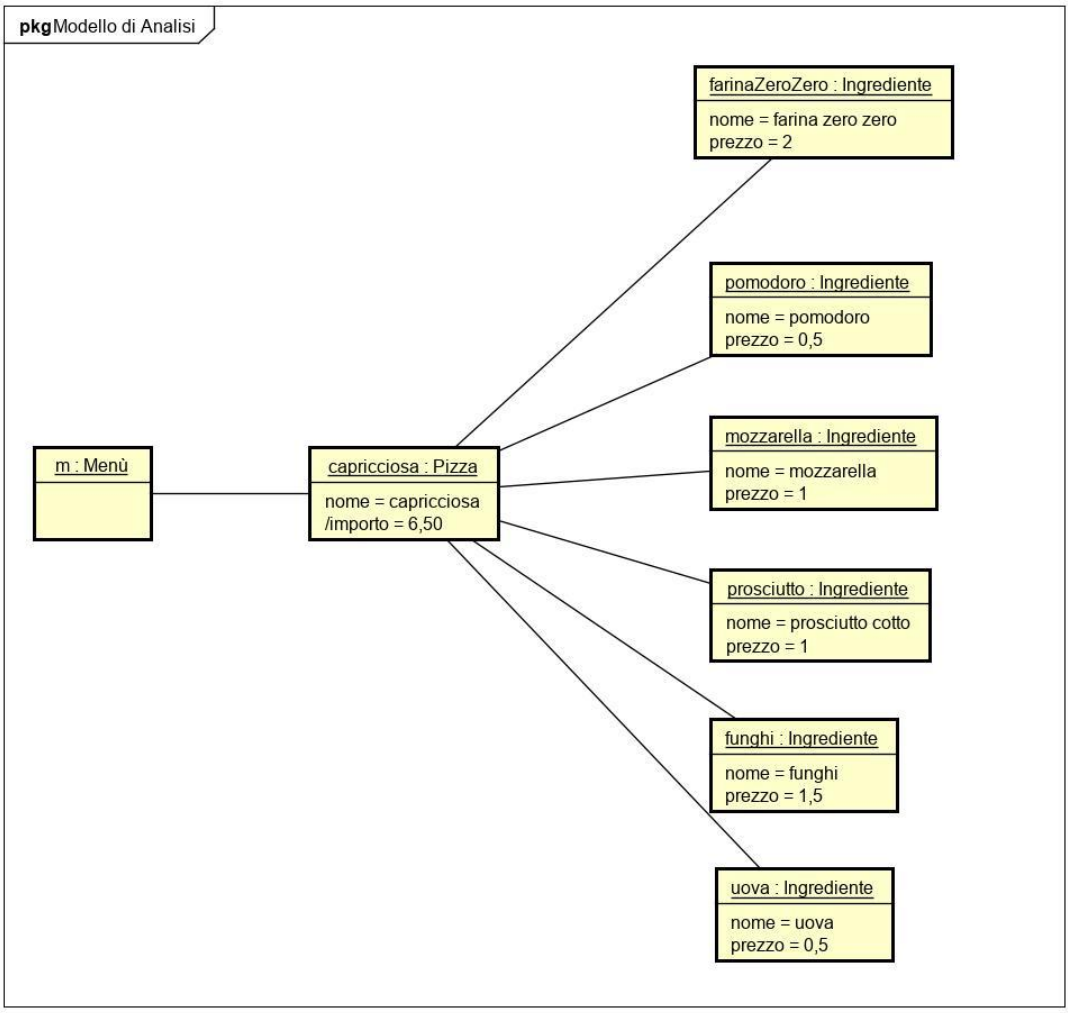
Per un maggiore dettaglio grafico si consulti il file astah allegato.
Come si può vedere, sono state identificate le seguenti classi concettuali:

- **Cliente:** attore primario che interagisce direttamente con il sistema, in particolare rappresenta il cliente della pizzeria che vuole ordinare la pizza e di cui si devono registrare le informazioni.
- **Admin:** attore primario che interagisce direttamente con il sistema, in particolare rappresenta il titolare dell'attività.
- **Pizza**: indica il tipo di pizza presente nel Menù.
- **Menù**: contiene le pizze e gli ingredienti disponibili.
- **Ingrediente:** indica uno degli elementi base di cui è composta una pizza.
- **Comanda:** indica l'ordinazione effettuata dal Cliente.
- **RigaDiComanda**: contiene i dettagli relativi all'acquisto di una particolare pizza e alla relativa quantità acquistata.
- **PanucciSystem:** rappresenta il sistema che dovrà gestire le operazioni.
- **Sconto:** rappresenta lo sconto da applicare alla comanda.
- **Pagamento:** rappresenta il metodo di pagamento utilizzato, specializzato dalle sottoclassi Carta e PayPal.
- **Paypal**: sottoclasse concettuale di pagamento che rappresenta il pagamento attraverso il sistema PayPal.
- **Carta:** sottoclasse concettuale di pagamento che rappresenta il pagamento con carta.

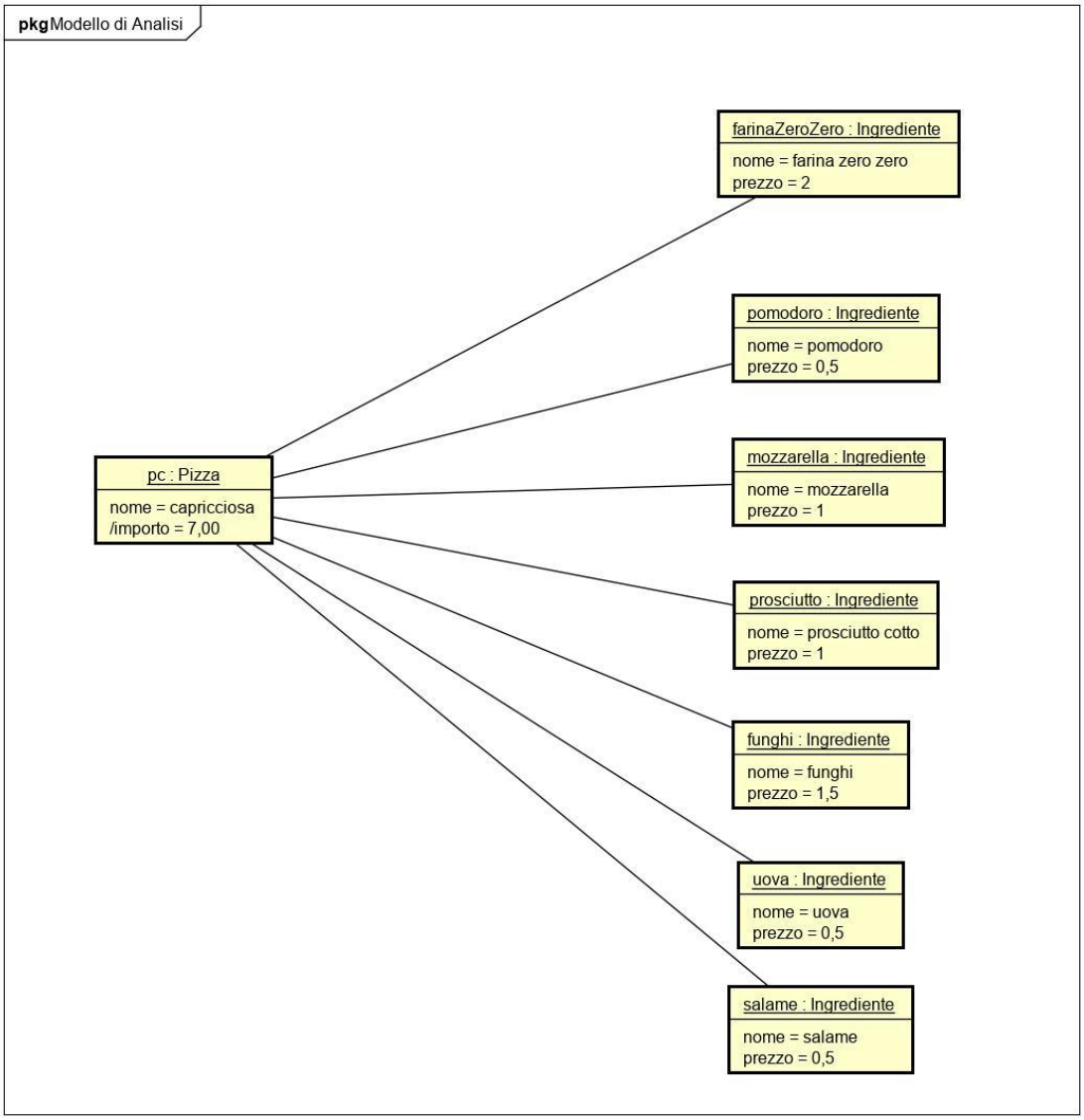
2.3. Diagramma degli Oggetti

Nel corso delle iterazioni sono stati costruiti i diagrammi degli oggetti e vengono ora ripresi. Le situazioni descritte con i relativi diagrammi vengono ora riprese.

- La pizza capricciosa nel menù è composta dagli ingredienti pomodoro, mozzarella, prosciutto cotto, funghi, uova.



- La pizza selezionata dall'utente, in particolare la pizza capricciosa viene modificata con l'aggiunta dell'ingrediente salame con un costo supplementare di 0,5 euro.

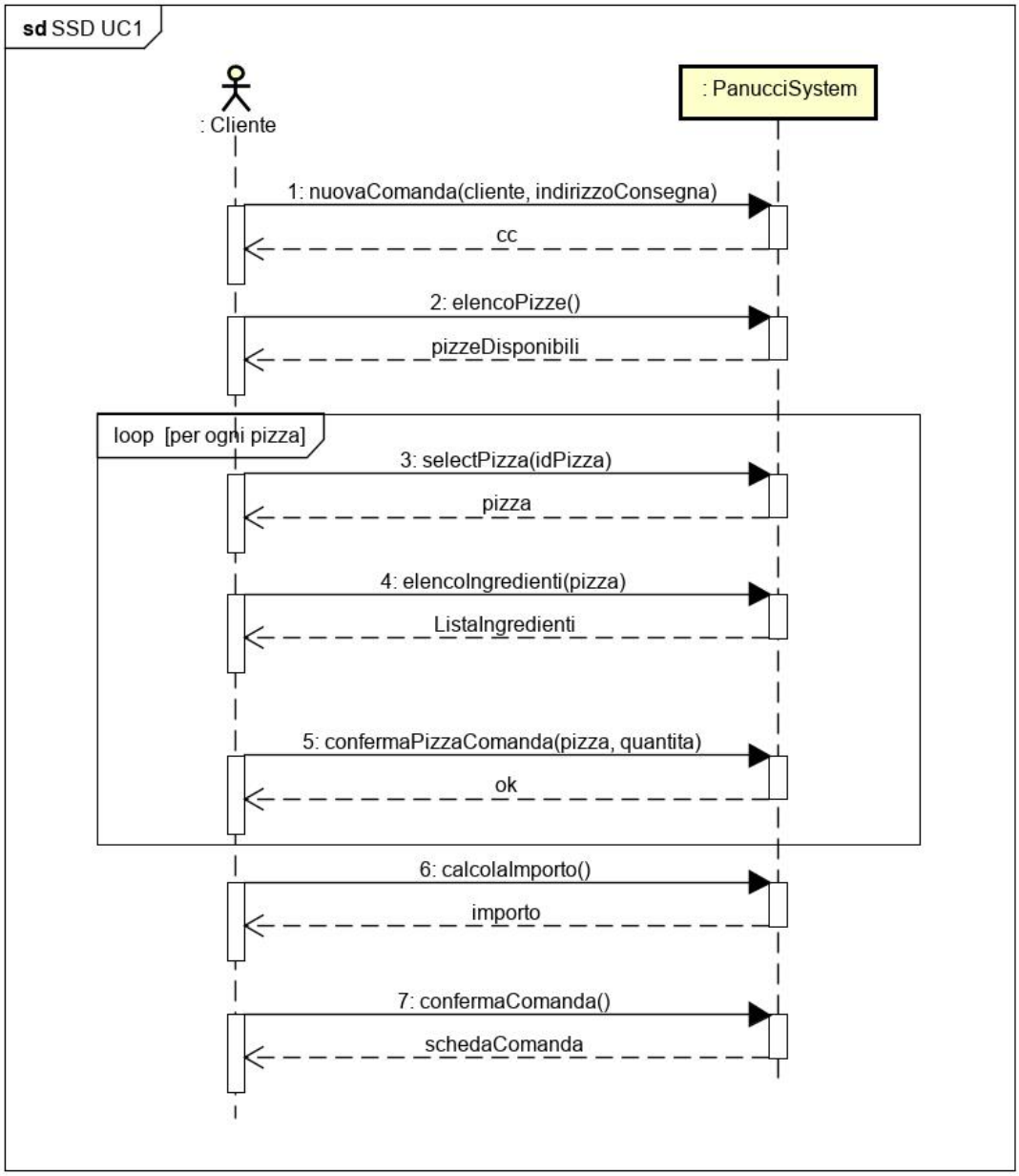


2.4. SSD e Contratti

Proseguendo con l'analisi Orientata agli Oggetti, il passo che segue è la creazione dei Diagrammi di Sequenza di Sistema (SSD) per mostrare il corso degli eventi di input e output per i vari casi d'uso esaminati in ogni iterazione. Inoltre, le principali operazioni di sistema individuate negli SSD saranno descritte attraverso i Contratti, quindi avremo:

- **Iterazione 1**

Scenario principale del caso d'uso UC1:



Si noti che viste le pre-condizioni del caso d'uso UC1, il cliente ha già effettuato l'accesso, pertanto nella prima operazione di sistema, associamo direttamente cliente alla comanda e non utilizziamo una operazione aggiuntiva associaCliente.

Contratti delle operazioni

selectPizza

L'operazione di sistema selectPizza seleziona la pizza di cui mostrare gli ingredienti.
Operazione selectPizza(idPizza)
Riferimenti Caso d'uso: UC1 Gestisci Comanda
Pre-condizioni - esiste un'istanza m di Menù
Post-condizioni - è stata recuperata l'istanza pizza di Pizza.

confermaPizzaComanda

L'operazione di sistema confermaPizzaComanda permette al sistema di aggiungere la pizza selezionata alla comanda.
Operazione confermaPizzaComanda(pizza)
Riferimenti Caso d'uso: UC1 Gestisci Comanda
Pre-condizioni - esiste un'istanza cc di Comanda
Post-condizioni - è stata creata un'istanza rdc di RigaDiComanda
- rdc è stata associata ad cc
- l'attributo quantità di rdc è stato aggiornato con quantità.

calcolaImporto

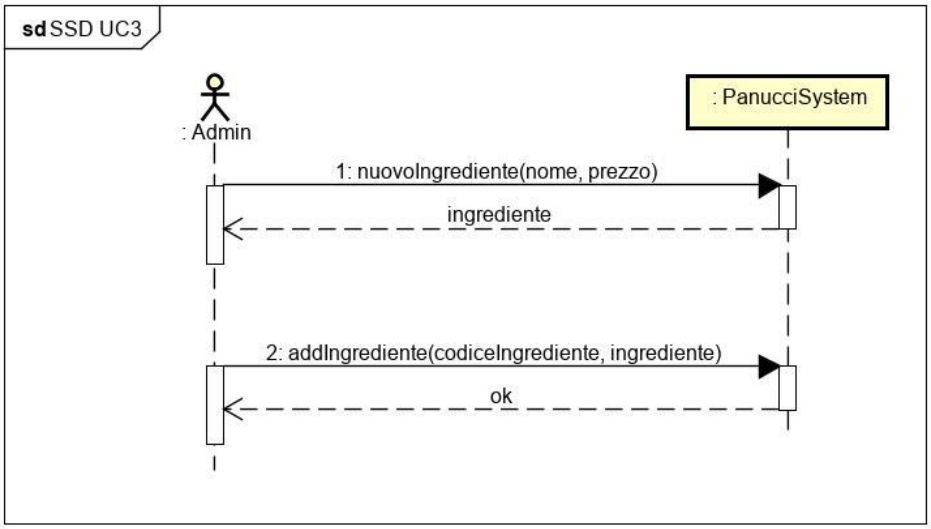
L'operazione di sistema confermaComanda permette al sistema di confermare l'acquisto.
Operazione calcolaImporto()
Riferimenti Caso d'uso: UC1 Gestisci Comanda
Pre-condizioni - esiste un'istanza cc di Comanda
Post-condizioni - l'attributo importo di cc è stato aggiornato.

confermaComanda

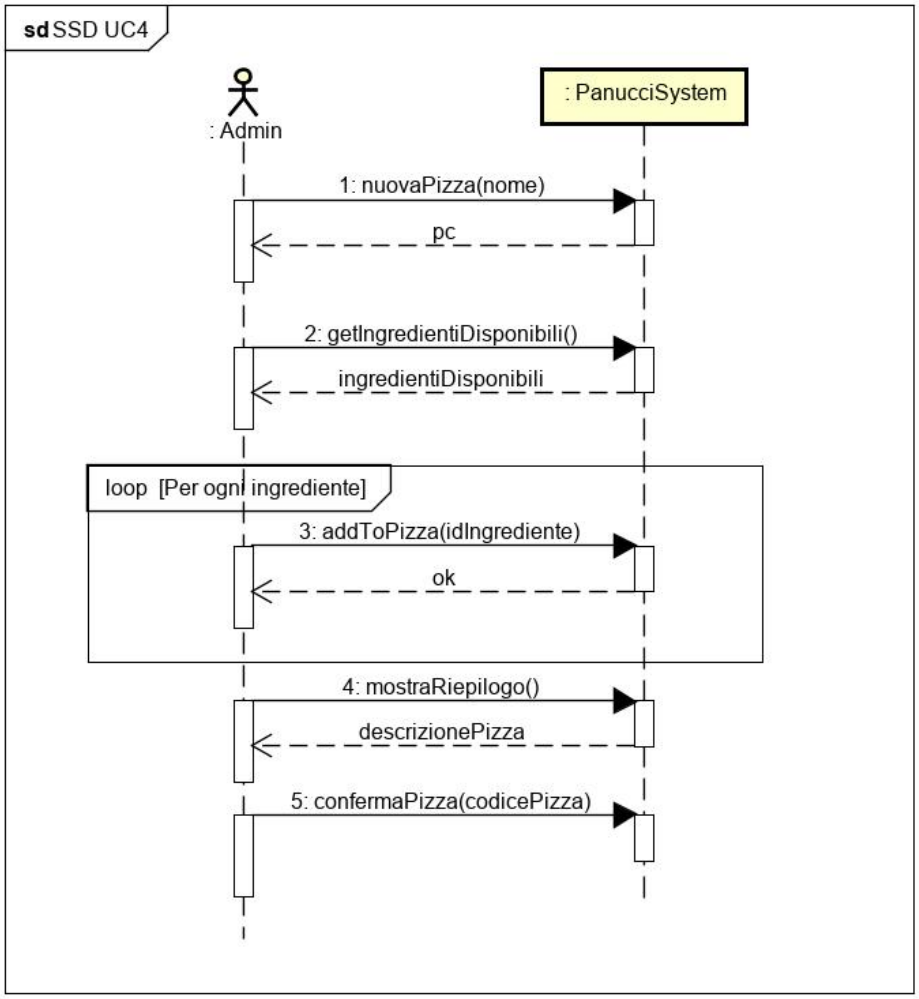
L'operazione di sistema confermaComanda permette al sistema di confermare l'acquisto.
Operazione confermaComanda()
Riferimenti Caso d'uso: UC1 Gestisci Comanda
Pre-condizioni - esiste un'istanza cc di Comanda
Post-condizioni - è stata aggiunta l'istanza cc di Comanda ad una lista di comande.

- **Iterazione 2**

Per lo scenario principale del caso UC3, il sistema dovrà permettere all'amministratore di inserire un nuovo ingrediente nel menù, quindi segue il Diagramma di Sequenza di Sistema:



Lo scenario principale del caso UC4 prevede l'inserimento di una pizza nel menù componendola con gli ingredienti disponibili nell'inventario, il Diagramma di Sequenza di Sistema è il seguente:



Contratti delle operazioni

addIngrediente

L'operazione di sistema aggiunge l'ingrediente nella listIngredientiDisponibili di Menù.
Operazione addiIngrediente (codiceIngrediente, ingrediente)
Riferimenti Caso d'uso: UC3 Gestisci Ingrediente
Pre-condizioni - è stato creato un ingrediente di tipo Ingrediente.
Post-condizioni - è stato aggiunto ingrediente nella listIngredientiDisponibili di Menù.

addToPizza

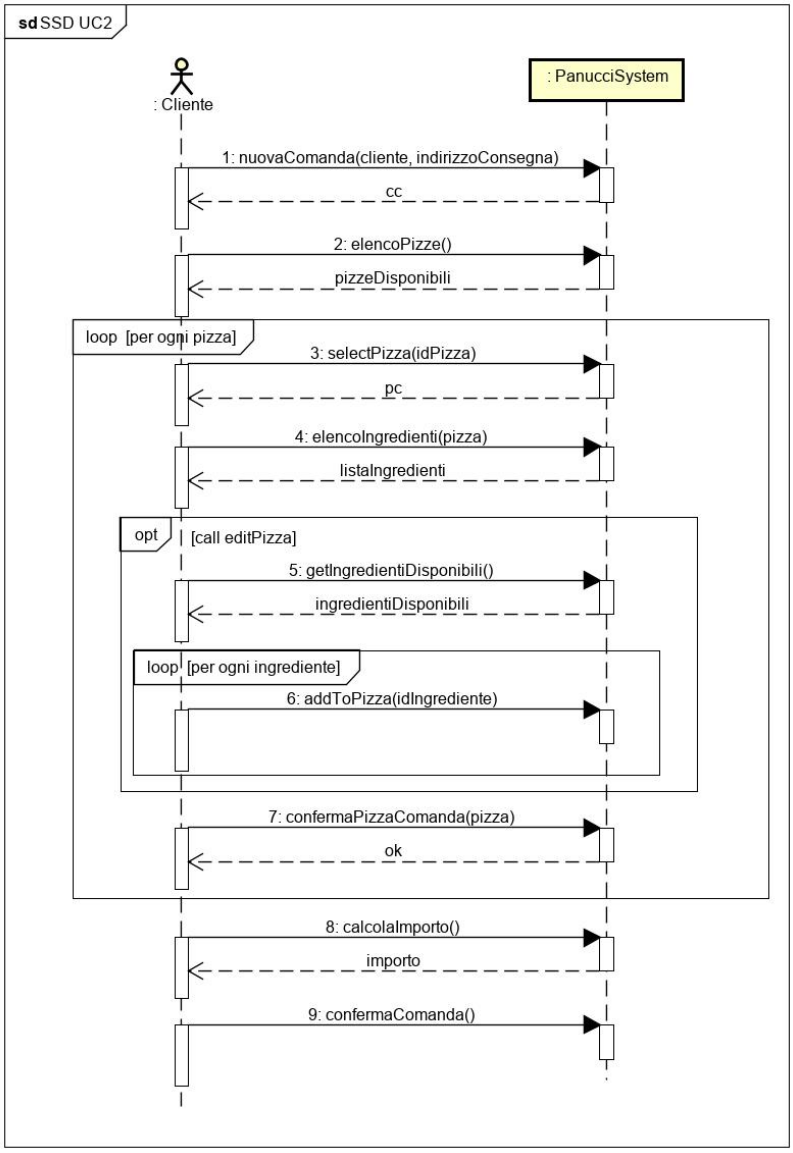
L'operazione di sistema aggiunge l'ingrediente nella pizza corrispondente all'identificativo idIngrediente.
Operazione addToPizza (idIngrediente)
Riferimenti Caso d'uso: UC4 Gestisci Pizza
Pre-condizioni - è stata creata un'istanza pc di Pizza.
Post-condizioni - è stato aggiunto ingrediente identificato da idIngrediente nella Pizza pc.

confermaPizza

L'operazione di sistema conferma la pizza pc e l'aggiunge nell'elenco delle pizze disponibili di Menù.
Operazione confermaPizza(codicePizza)
Riferimenti Caso d'uso: UC4 Gestisci Pizza
Pre-condizioni - è stata creata un'istanza pc di Pizza.
Post-condizioni - è stato aggiunto pc di tipo Pizza nella listPizza di Menù con codice codicePizza.

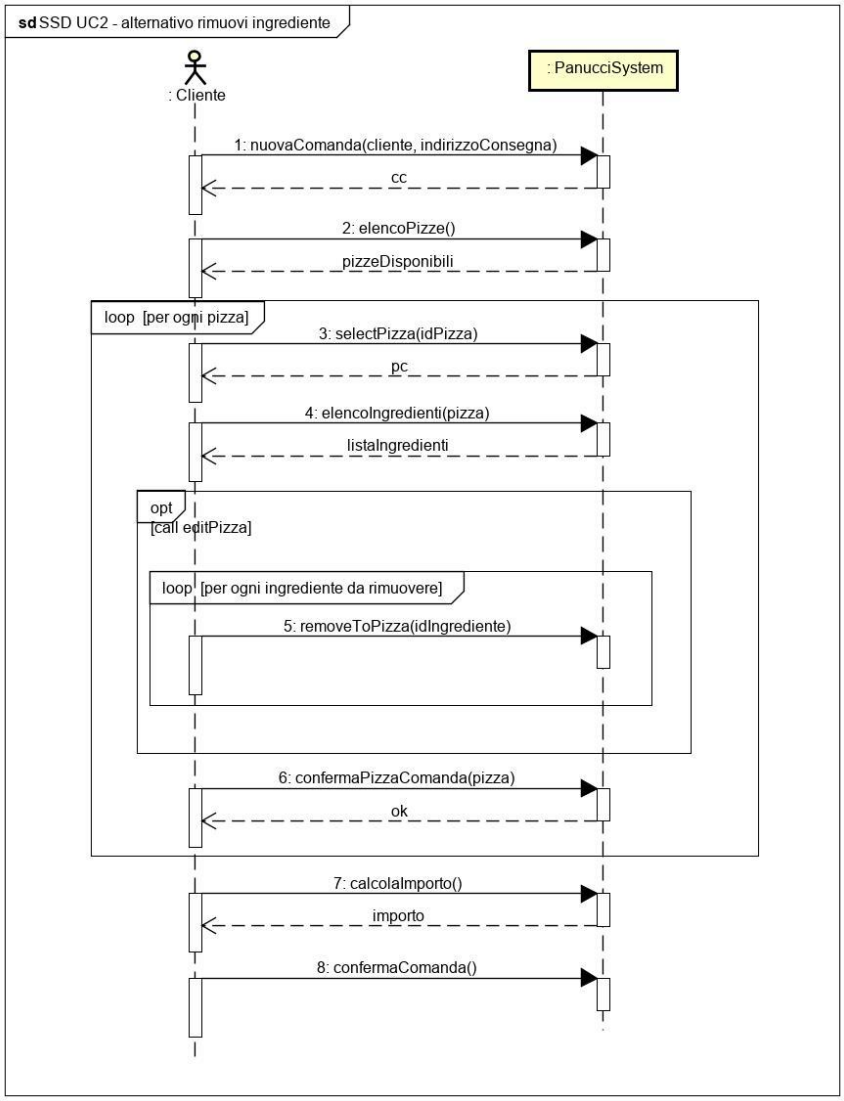
- **Iterazione 3**

Lo scenario principale del caso UC2 prevede la modifica degli ingredienti di una pizza precedentemente selezionata dal menù, aggiungendo gli ingredienti disponibili nell'inventario.
Essendo essa uno scenario alternativo del caso d'uso UC1, viene riportato il Diagramma di Sequenza di Sistema per intero per una maggiore chiarezza.

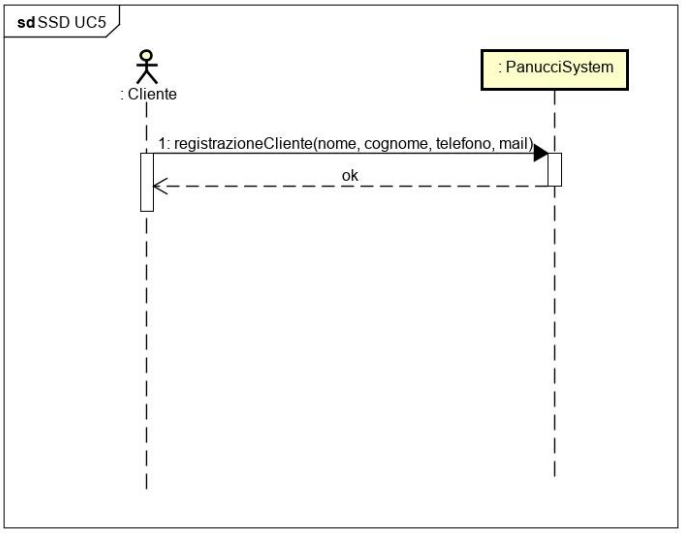


Per maggiore dettaglio grafico si consultino i file astah relativi.

Lo scenario alternativo 1a del caso d'uso UC2 prevede il seguente diagramma:



Lo scenario principale del caso UC5 prevede la registrazione dei dati del cliente nel sistema, il Diagramma di Sequenza di Sistema è il seguente:



Contratti delle operazioni

registrazioneCliente

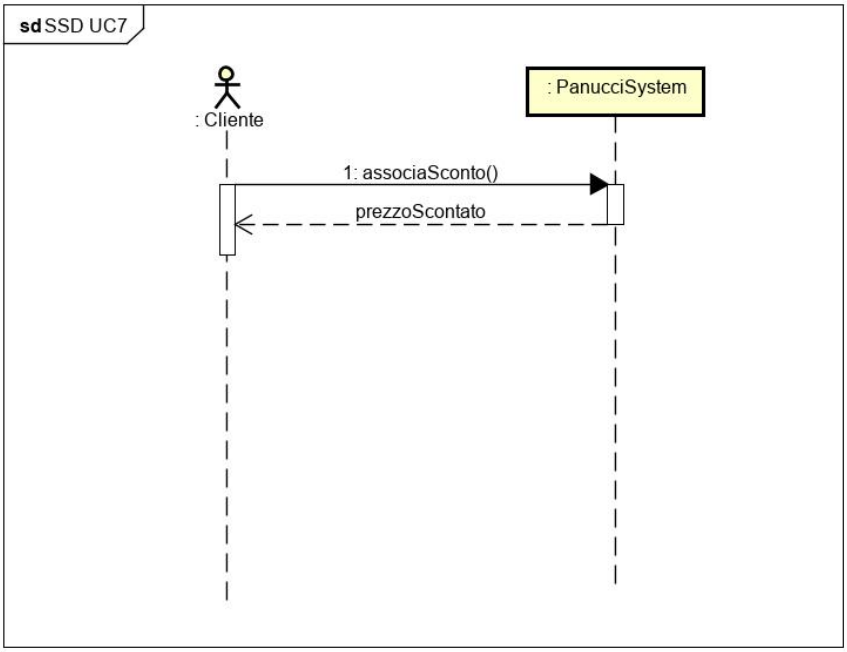
Il cliente si registra nel sistema tramite l'operazione di sistema registrazione.

Operazione	registrazioneCliente(nome, cognome, telefono, mail)
Riferimenti	Caso d'uso: UC2 Modifica Pizza
Pre-condizioni	- il cliente ha selezionato l'operazione registrazione
Post-condizioni	- il cliente viene aggiunto alla lista elencoClienti di PanucciSystem che comprende l'elenco degli utenti registrati.

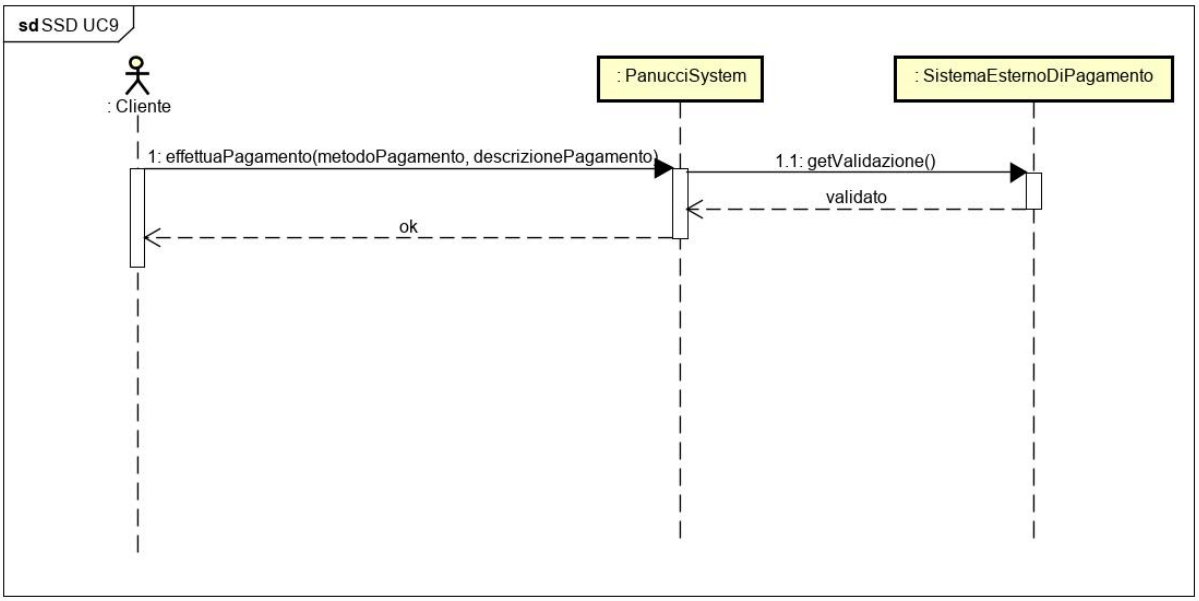
Si noti che le operazioni di sistema getIngredientiDisponibili e addToPizza non vengono specificate attraverso i contratti delle operazioni di sistema e non saranno prese in considerazione neanche i loro diagrammi di sequenza in quanto queste operazioni sono le stesse usate dall'amministratore in UC1, la differenza sta nel fatto che la pizza corrente pc in questo caso sarà restituita dall'operazione di sistema findPizza.
In questo modo non abbiamo necessità di aggiungere metodi per fare le stesse operazioni elementari e l'utente potrà aggiungere l'ingrediente solo nella pizza corrente selezionata da lui precedentemente.

- **Iterazione 4**

Lo scenario principale di successo del caso d'uso UC7 prevede il calcolo di uno sconto su una comanda, il Diagramma di Sequenza di Sistema è il seguente:



Lo scenario principale di successo del caso d'uso UC9 prevede il seguente diagramma:



Contratti delle operazioni

verificaSconto

L'operazione di sistema verificaSconto avrà il compito di cercare uno sconto applicabile alla comanda in corso.

Operazione	verificaSconto()
Riferimenti	Caso d'uso: UC7 Aggiungi Sconto
Pre-condizioni	- c'è una comanda in corso - è stato trovato uno sconto applicabile (numero righe comande superiore soglia)
Post-condizioni	- è stato ritornato uno sconto di Tipo Sconto.

associaSconto

L'operazione di sistema associaSconto avrà il compito di associare uno sconto alla comanda in corso ed aggiornare il prezzo relativo.

Operazione	associaSconto(sconto)
Riferimenti	Caso d'uso: UC7 Aggiungi Sconto
Pre-condizioni	- c'è una comanda in corso - è stato trovato uno sconto applicabile (numero righe comande superiore soglia)
Post-condizioni	- è stato trovato uno sconto applicabile - l'attributo importo della comanda in corso è stato aggiornato all'importo finale.

effettuaPagamento

L'operazione di sistema effettuaPagamento si occuperà di effettuare il pagamento.

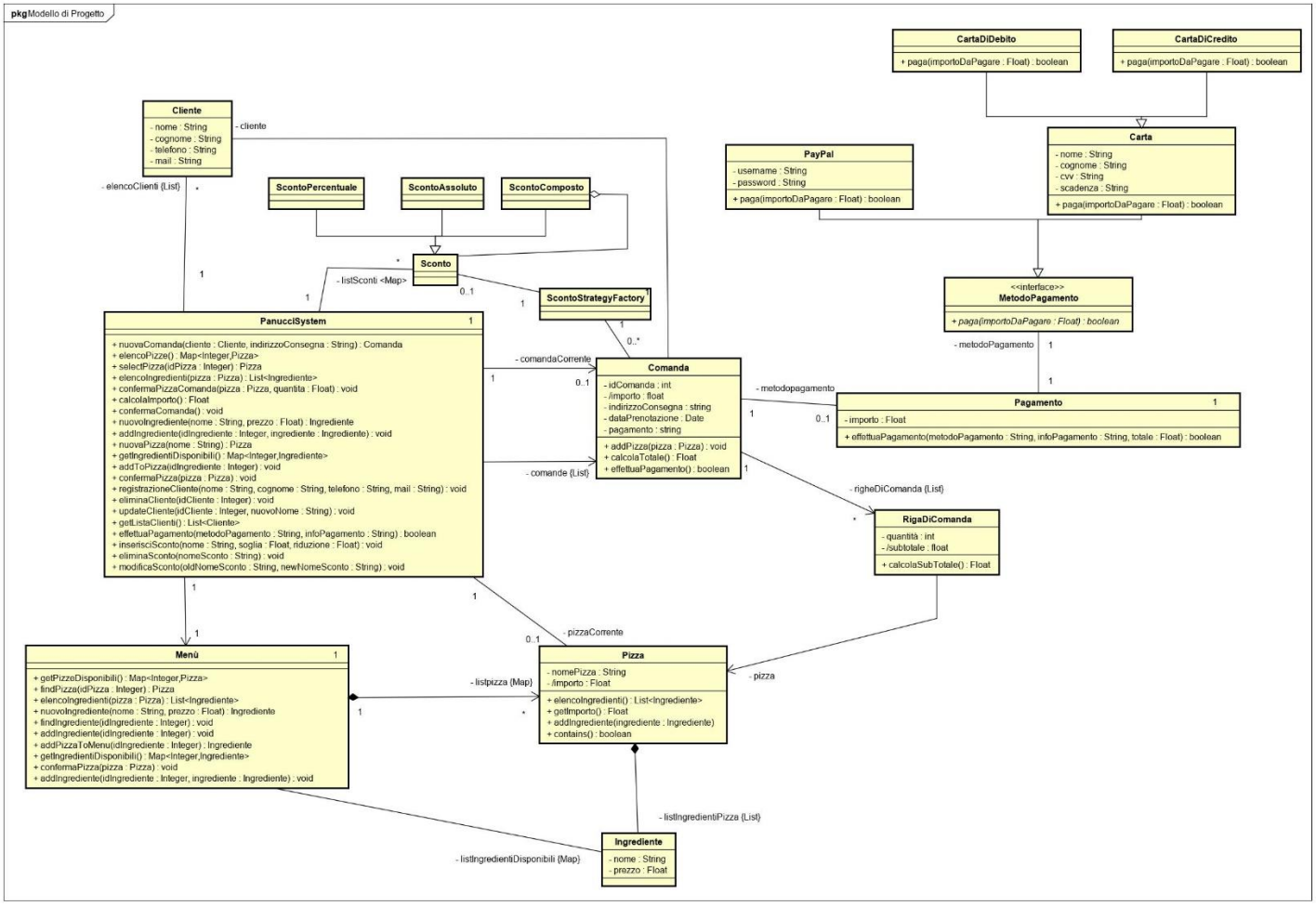
Operazione	effettuaPagamento(metodoPagamento, descrizionePagamento)
Riferimenti	Caso d'uso: UC9 Effettua pagamento
Pre-condizioni	- c'è una comanda in corso
Post-condizioni	- è stata creata un'istanza della classe concreta che implementa l'interfaccia Pagamento - è stato chiamato il metodo paga() che inoltra la chiamata al sistema esterno che si occupa di gestire il pagamento - è stata restituita la conferma che il pagamento si è concluso a buon fine.

3. Progettazione

La fase di progettazione è interessata alla definizione degli oggetti software, delle loro responsabilità e a come questi collaborano per soddisfare i requisiti individuati nei passi precedenti. L'elaborato principale di questa fase considerato è il Modello di Progetto ovvero l'insieme dei diagrammi che descrivono la logica sia da un punto di vista dinamico (Diagrammi di Interazione) che statico (Diagramma delle Classi).

3.1. Diagramma delle Classi

Una volta conclusa l'ultima iterazione, il diagramma delle classi è il seguente:



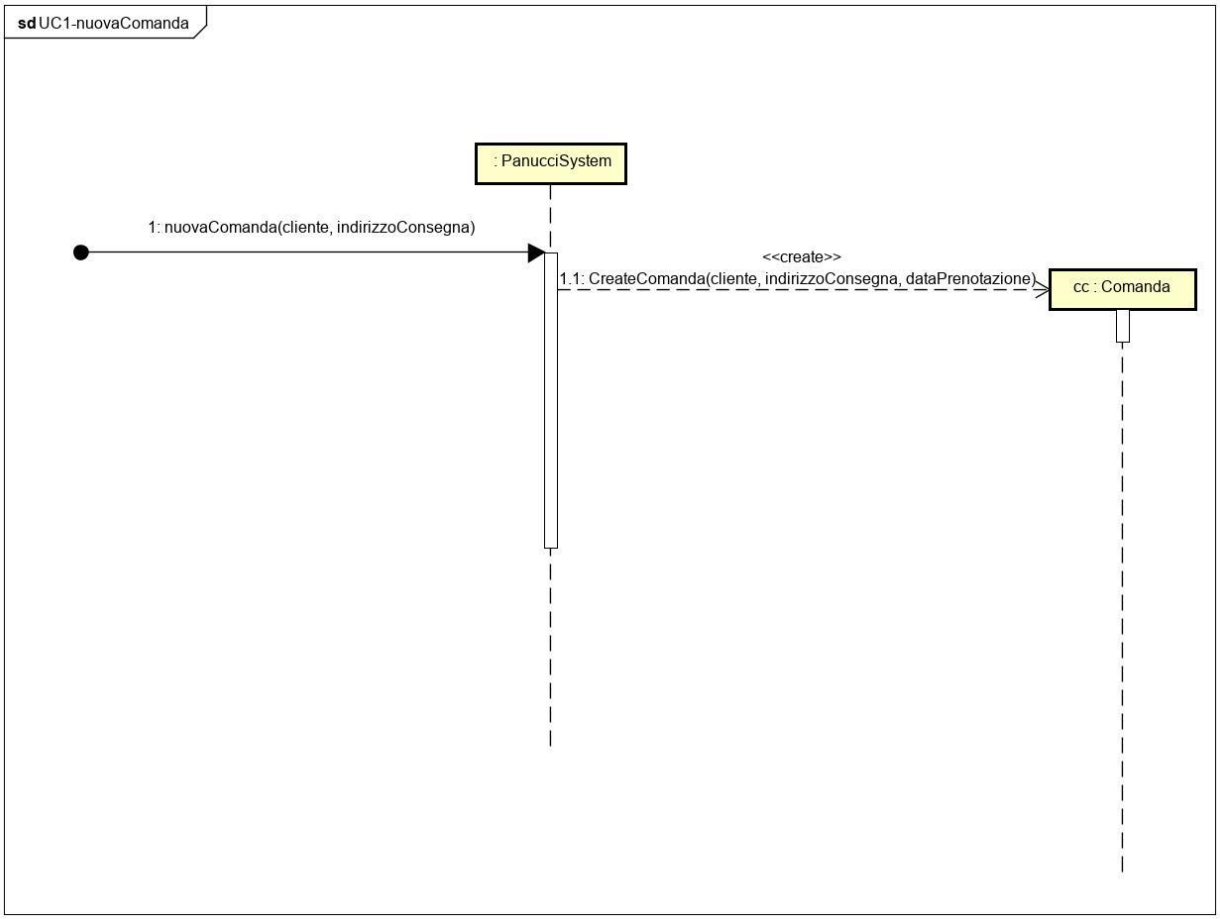
Per una maggiore dettaglio grafico si consulti il file astah allegato.

3.2. Diagrammi di Sequenza

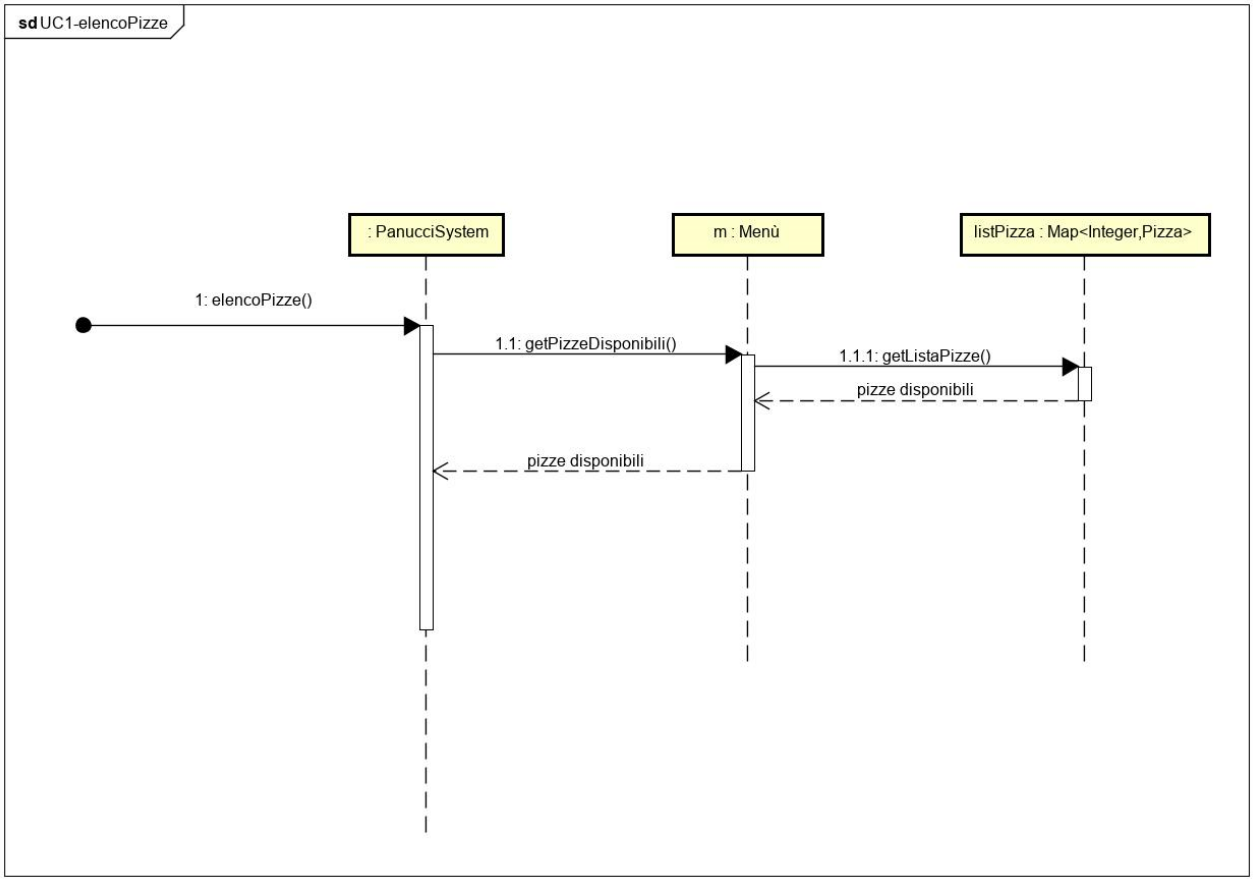
I diagrammi di sequenza ricavati nelle varie iterazioni sono mostrati di seguito.

- Iterazione 1

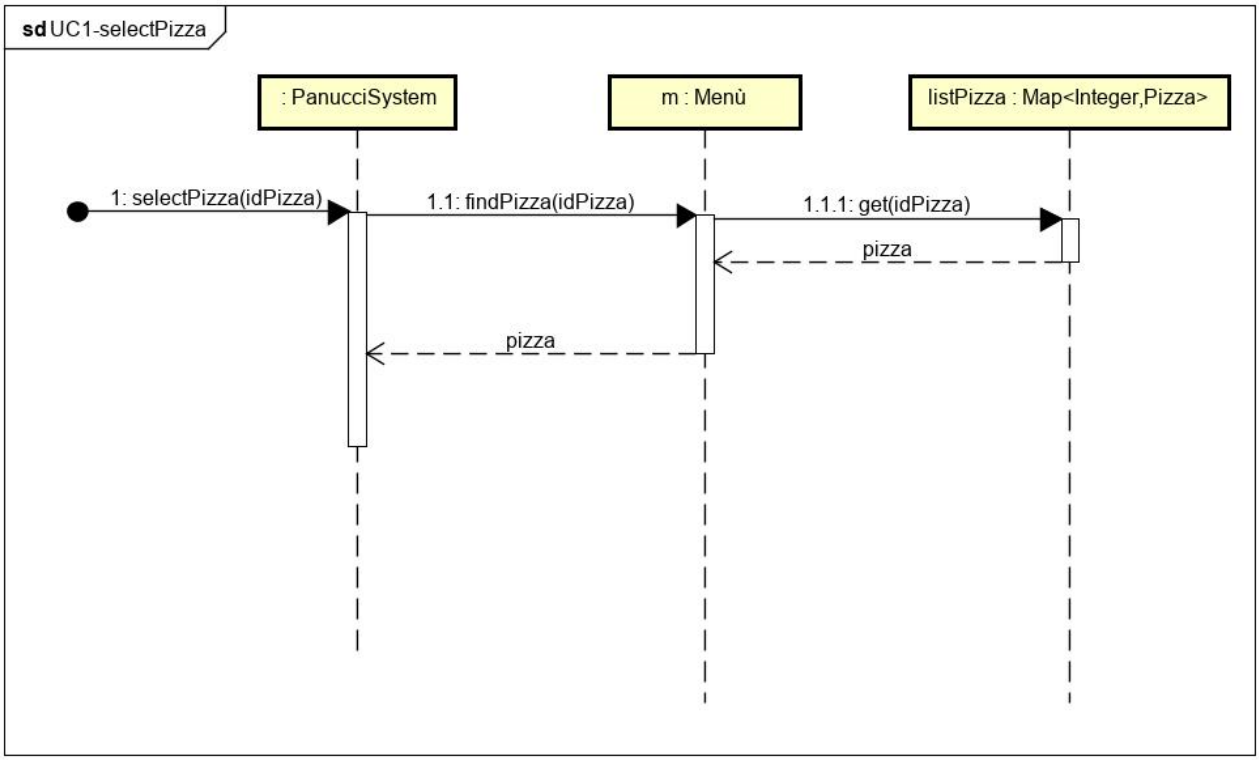
nuovaComanda (cliente: Cliente, indirizzoConsegna: String)



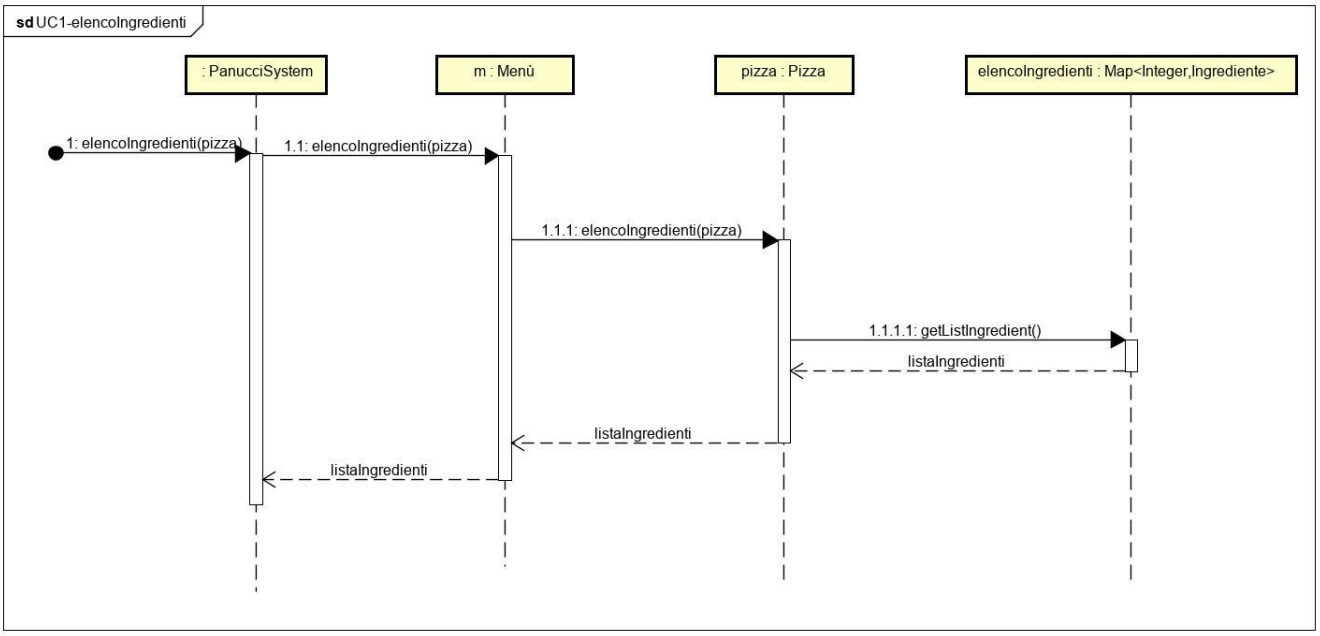
elencoPizze ()



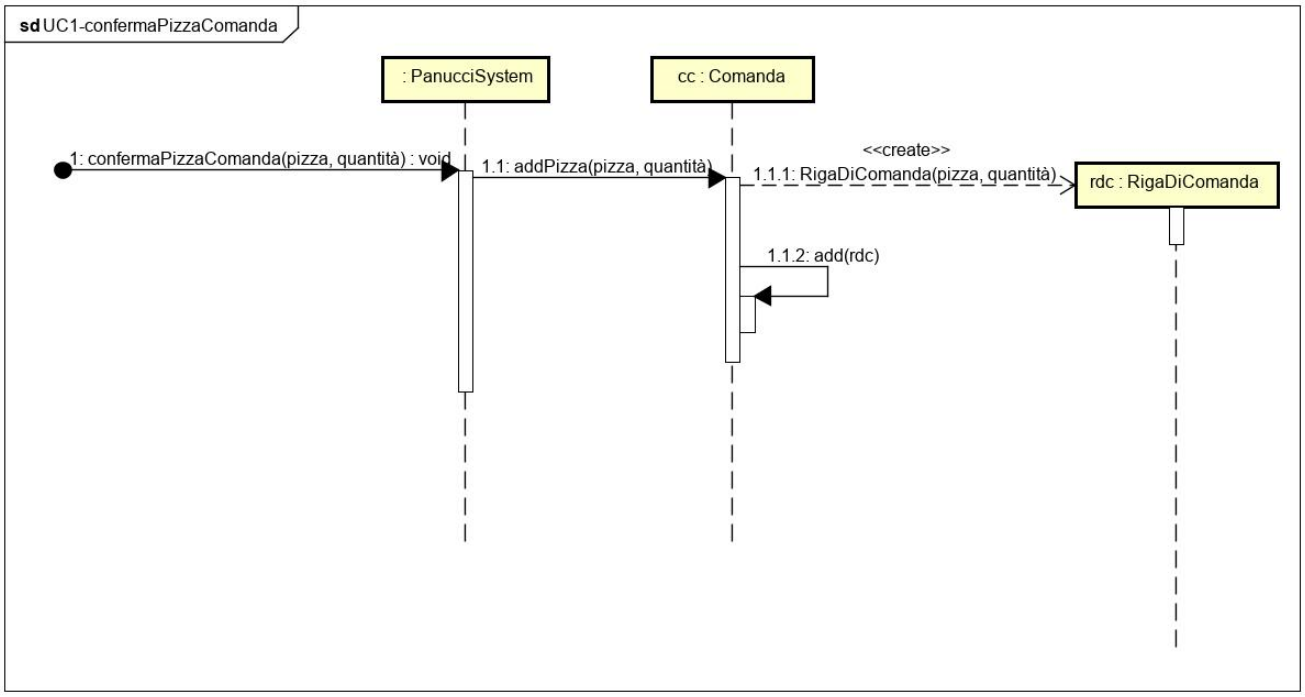
selectPizza(idPizza: Integer)



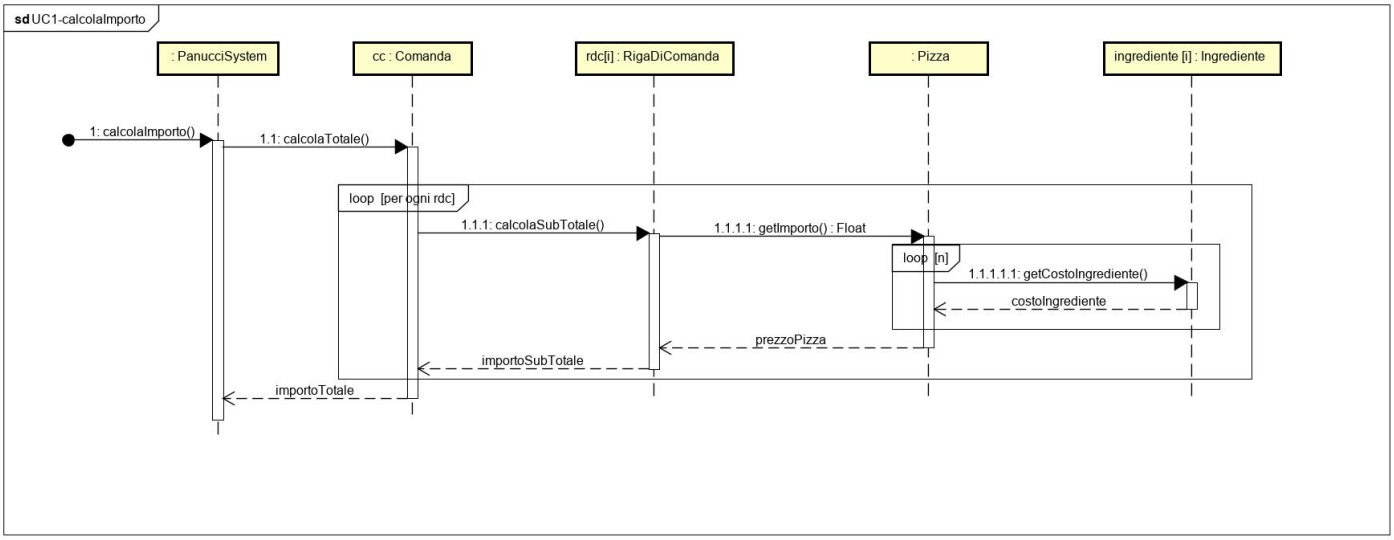
elencoIngredienti(pizza: Pizza)



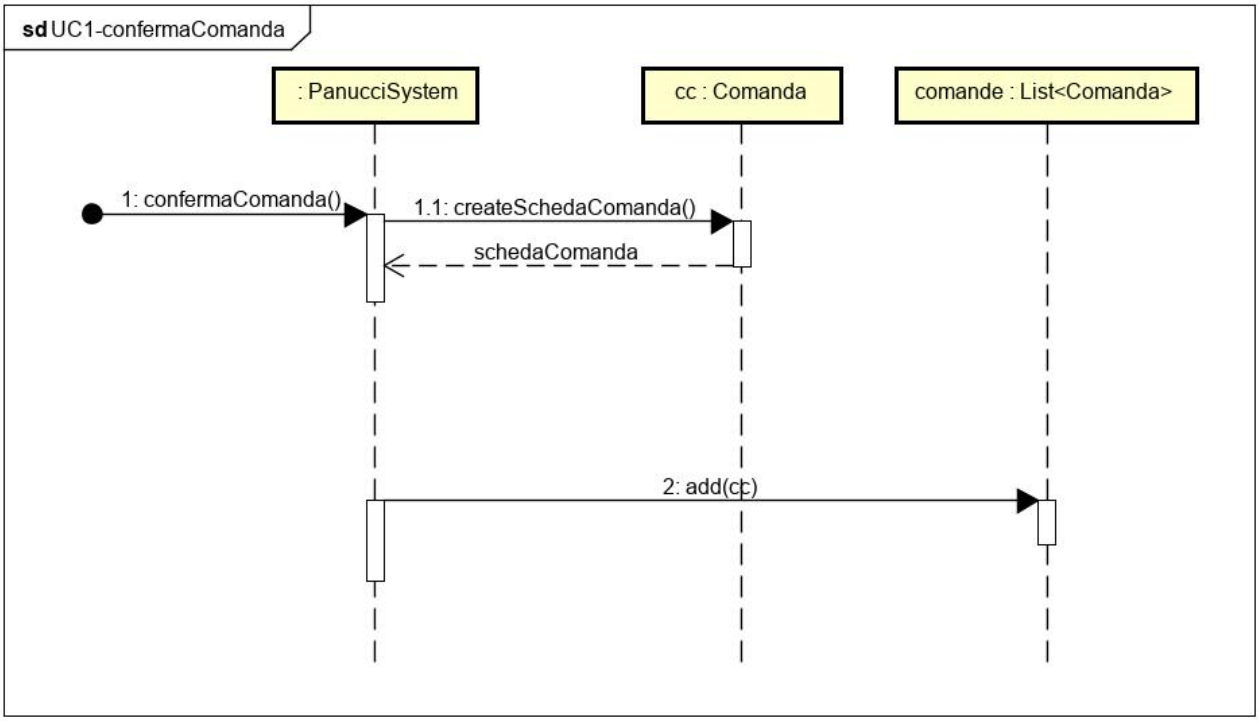
confermaPizzaComanda(pizza: Pizza, quantità: int)



calcolaImporto()

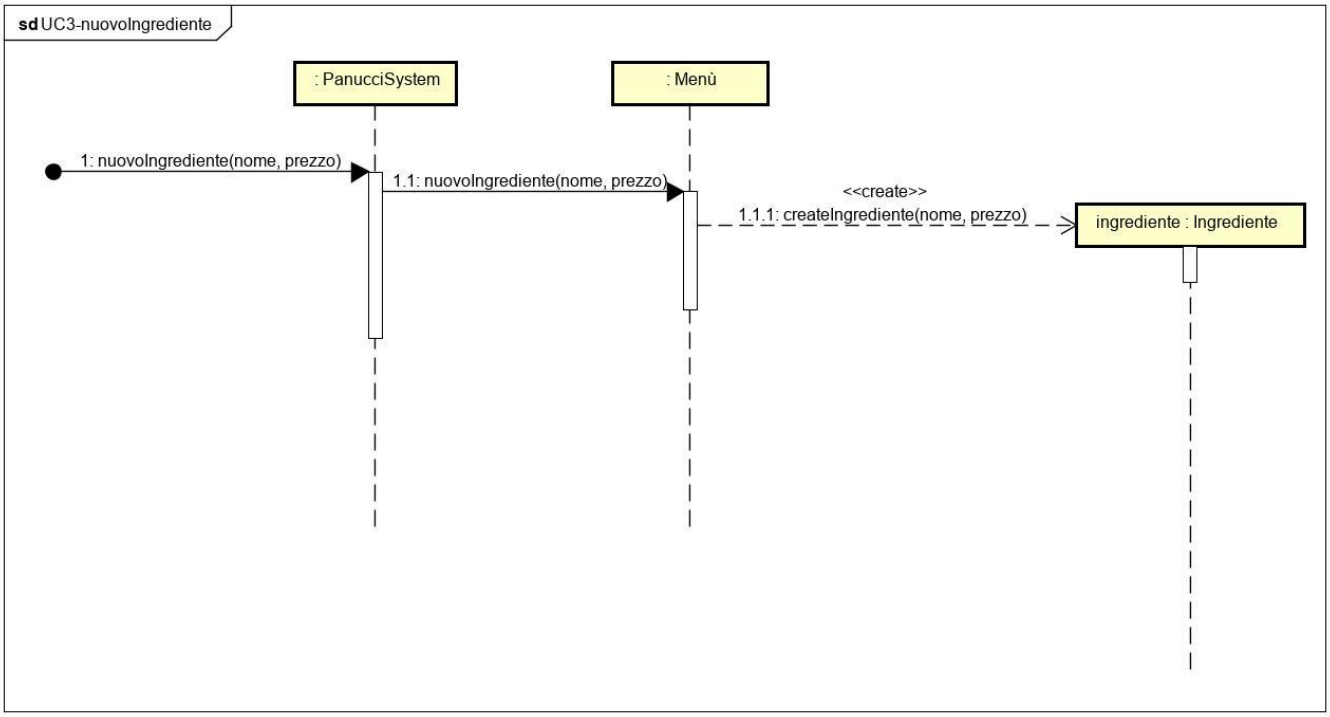


confermaComanda()

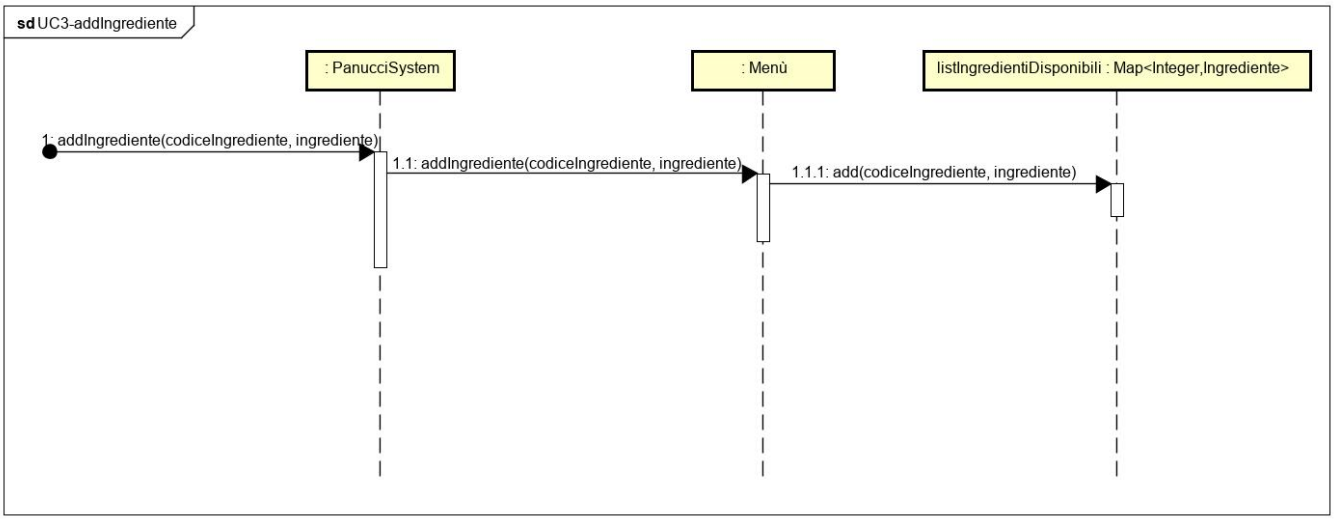


- Iterazione 2

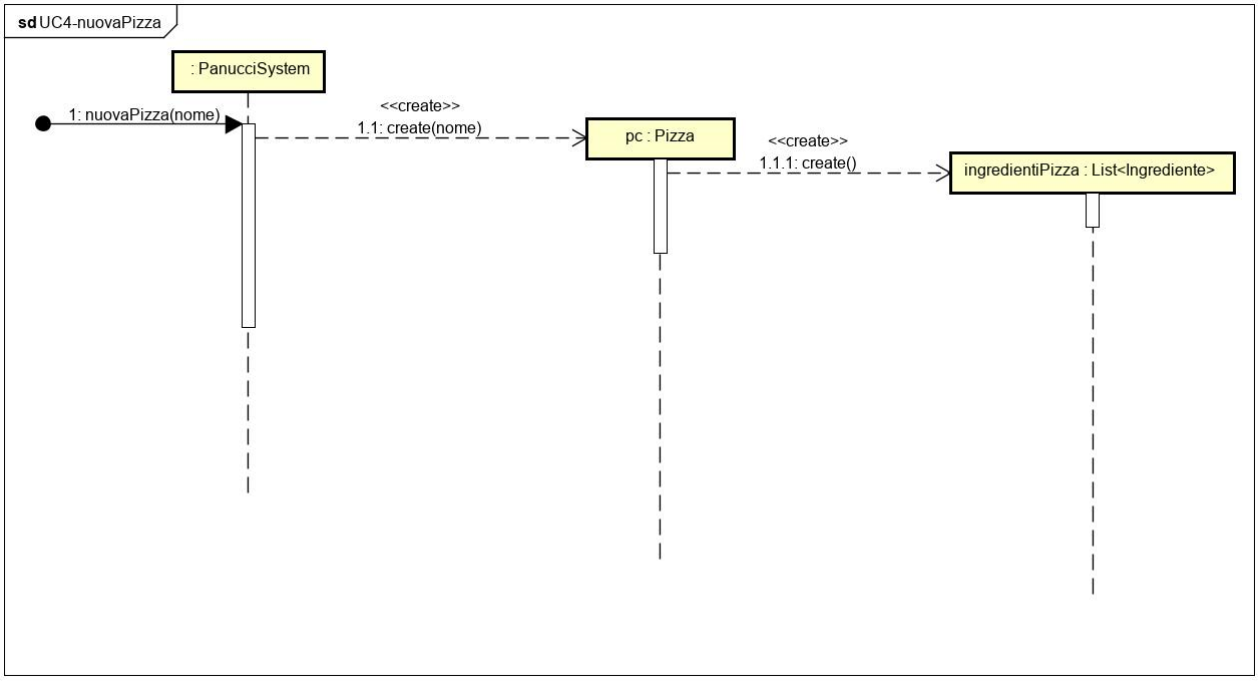
nuovoIngrediente (nome: String, prezzo :Float)



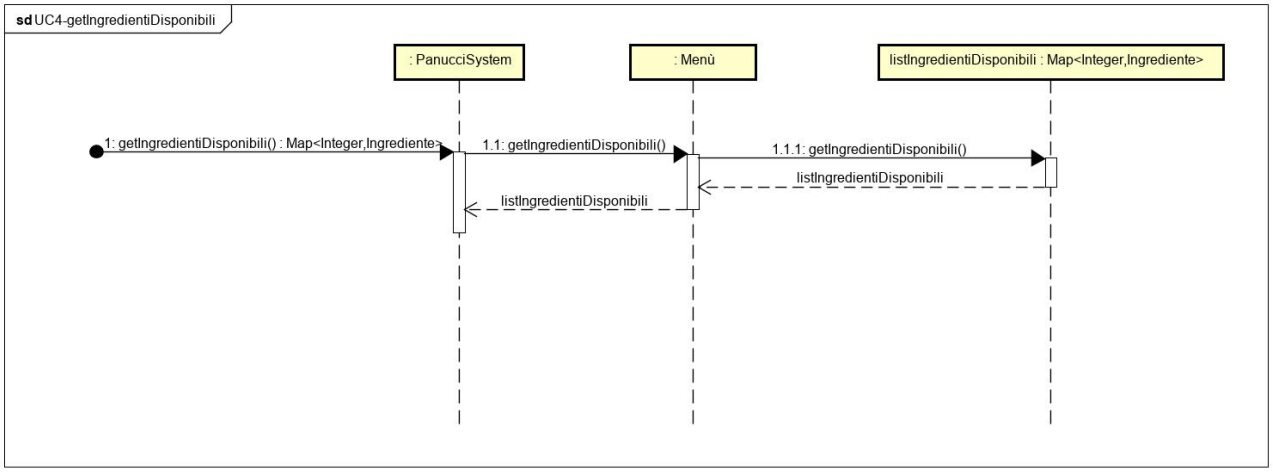
addIngrediente (codiceIngrediente:Integer, ingrediente:Ingrediente)



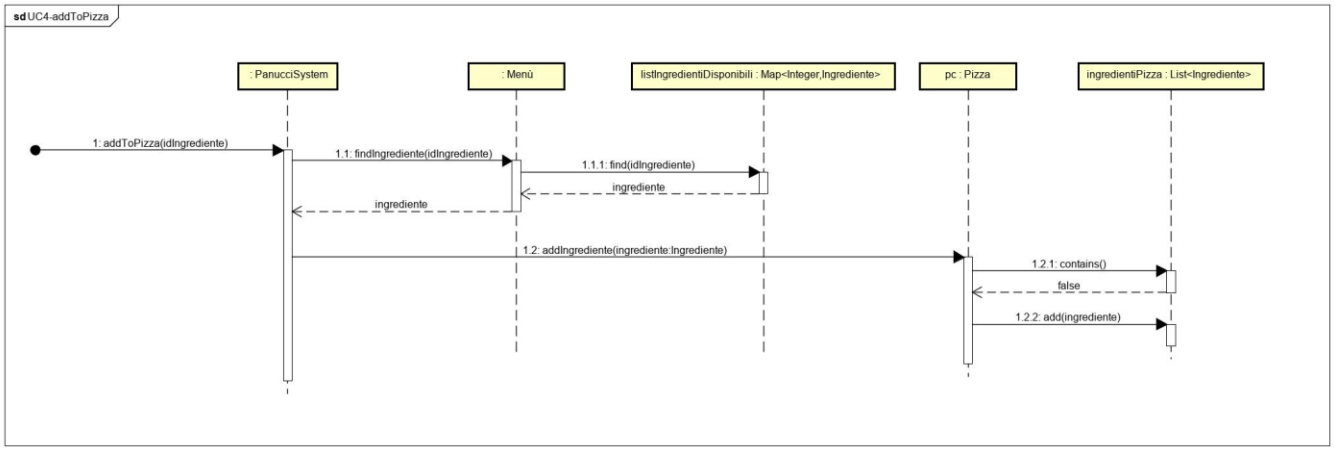
nuovaPizza(nome:String)



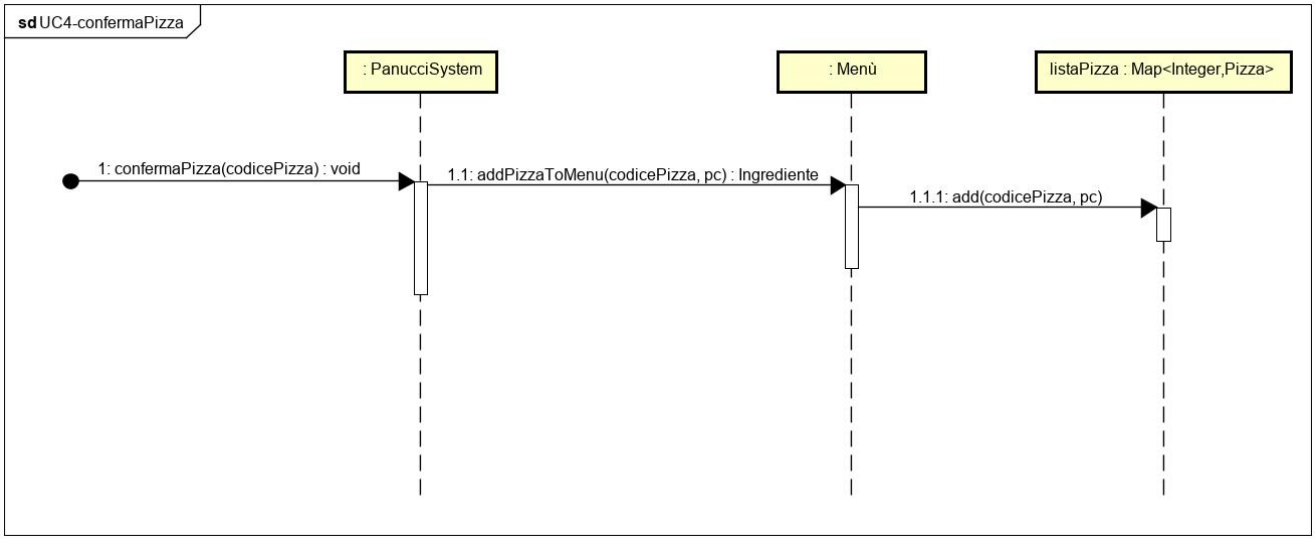
getIngredientiDisponibili ()



addToPizza (idIngrediente:Integer)

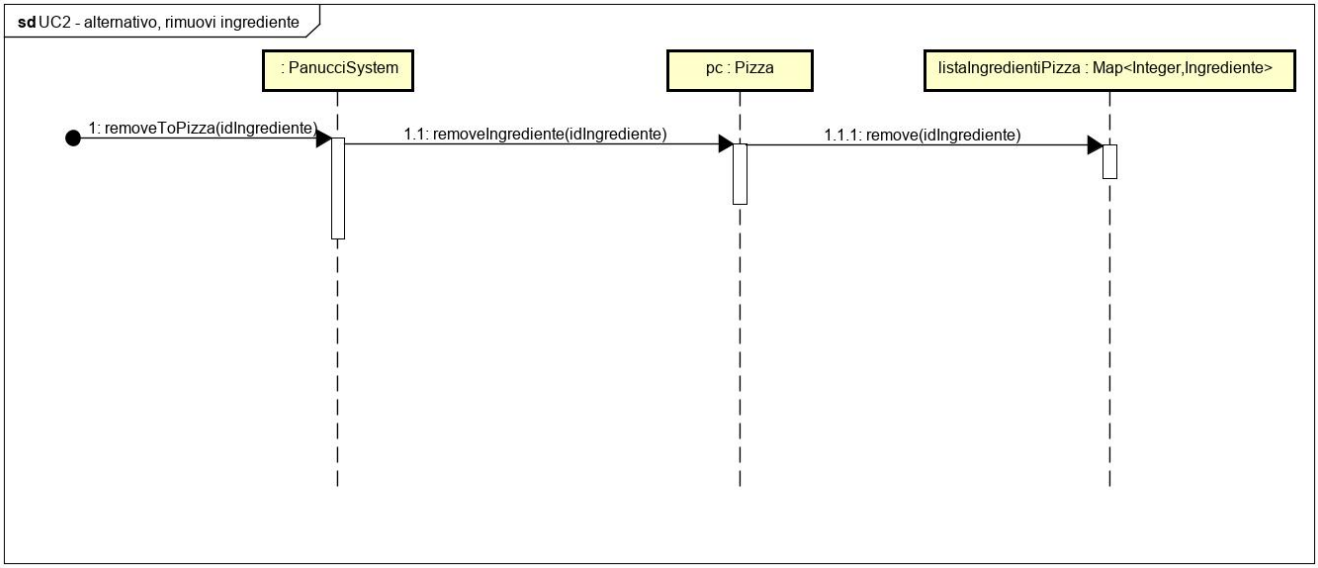


confermaPizza(codicePizza :Integer)

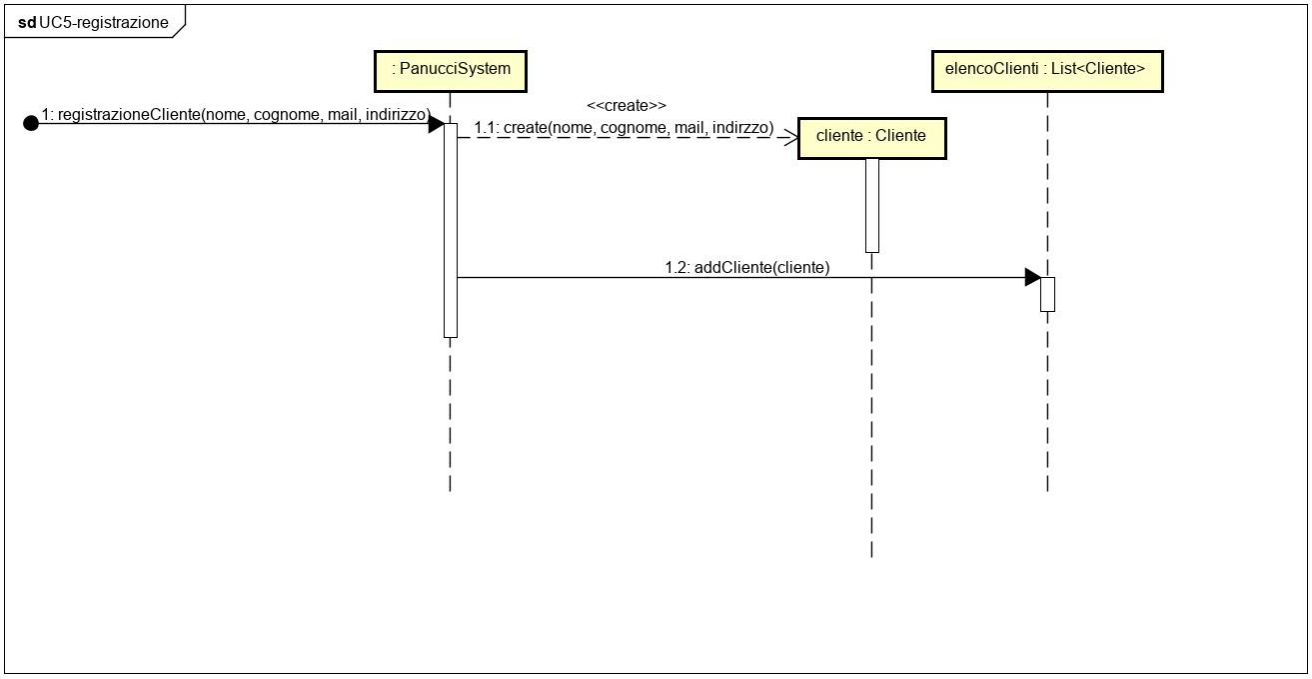


- Iterazione 3

removeToPizza(idIngrediente:int)

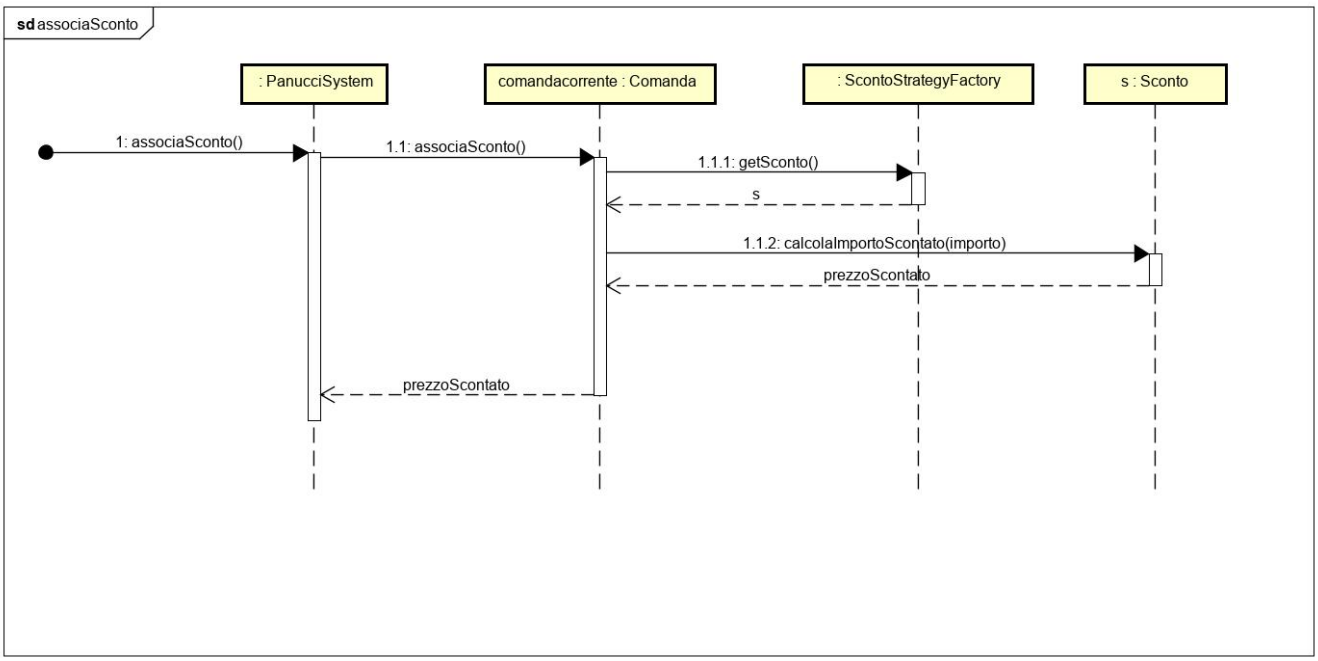


registrazioneCliente(nome:String, cognome:String, mail:String, indirizzo:String)

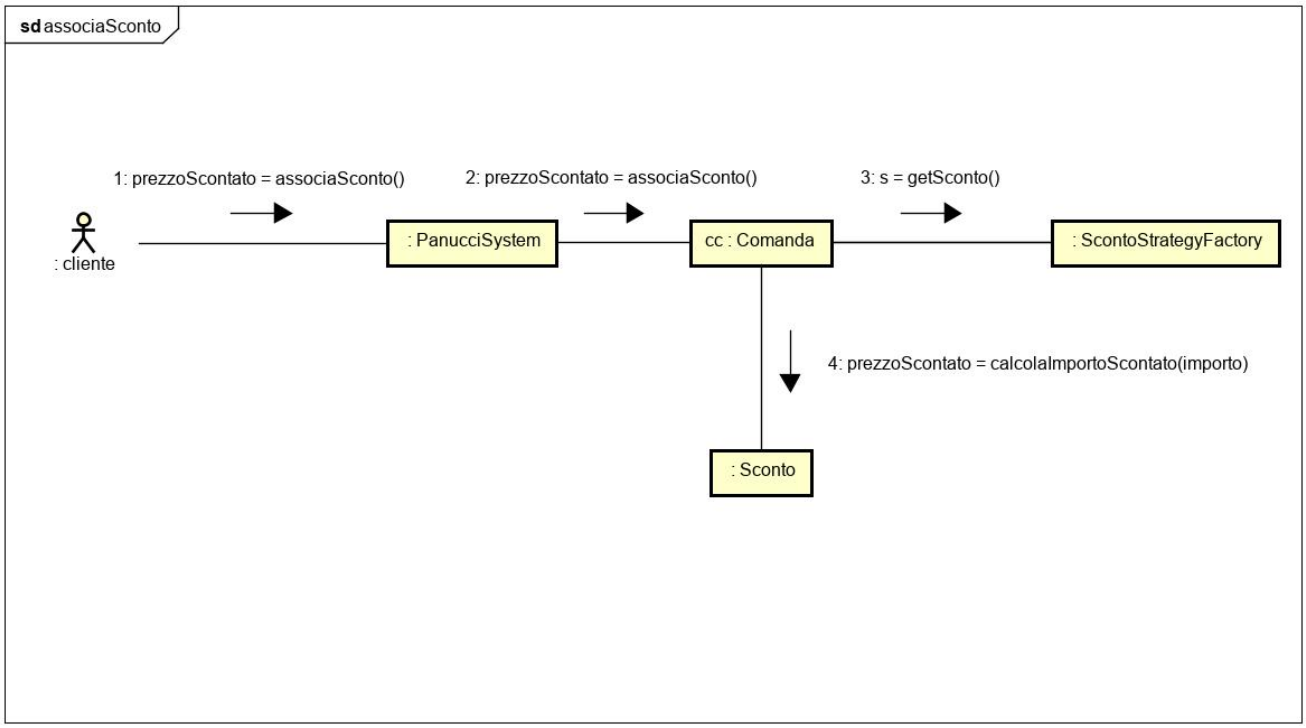


- Iterazione 4

associaSconto()

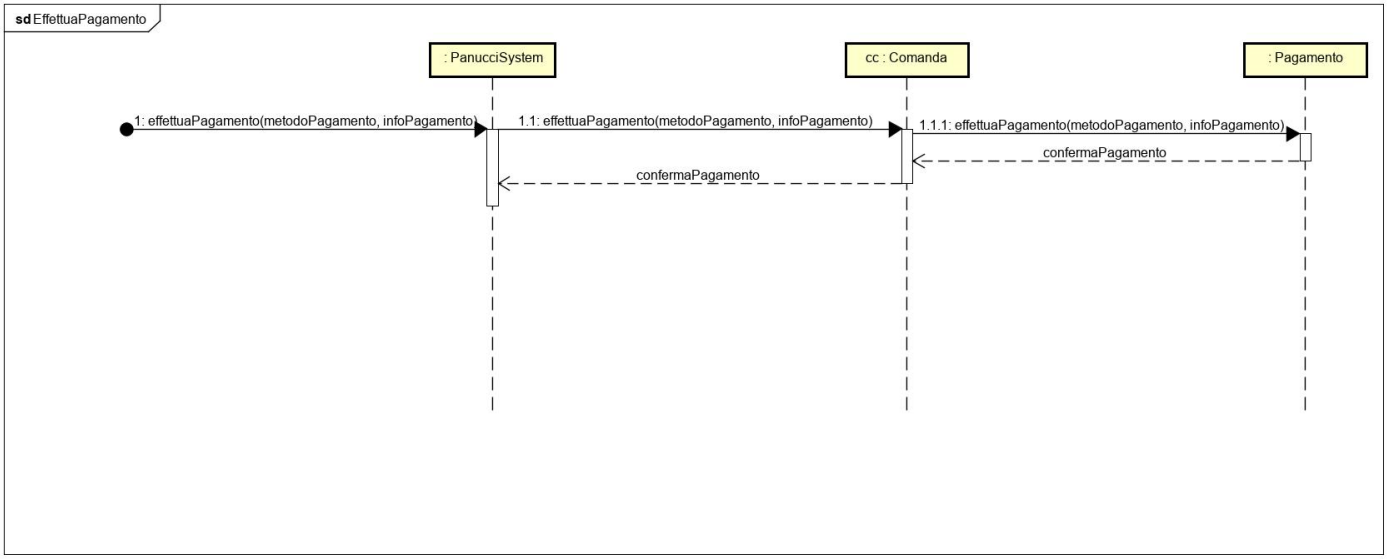


In questa occasione viene riportato anche il diagramma di comunicazione.

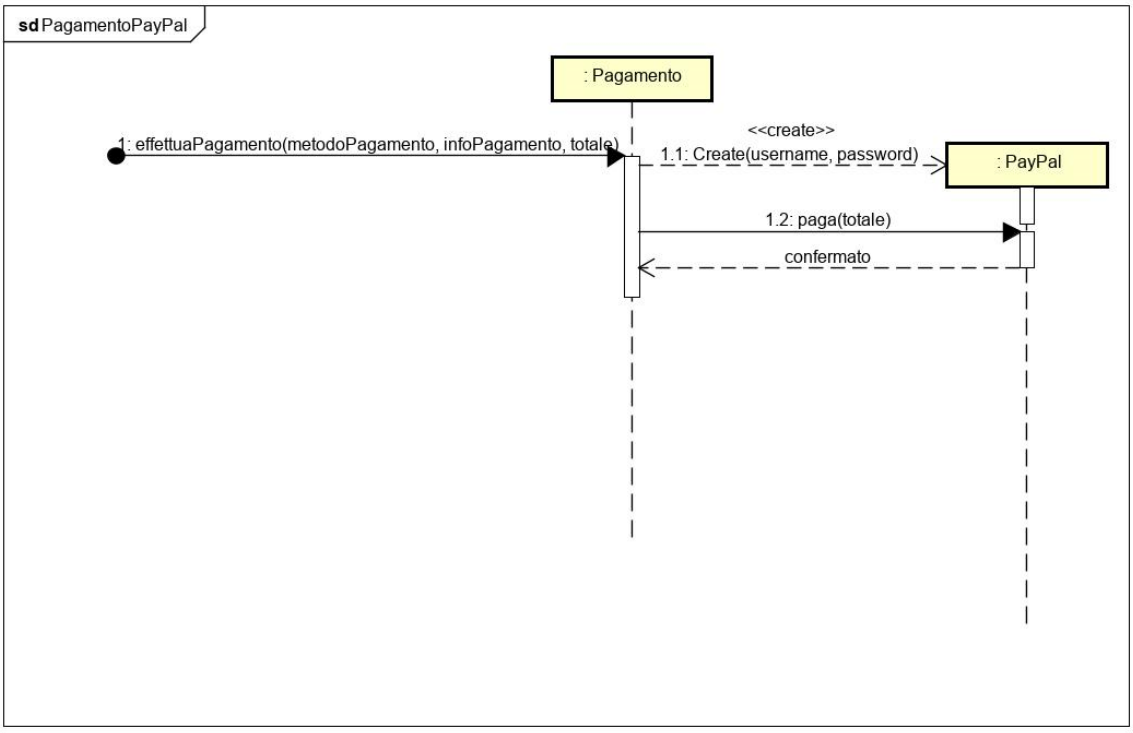


effettuaPagamento(metodoPagamento, infoPagamento)

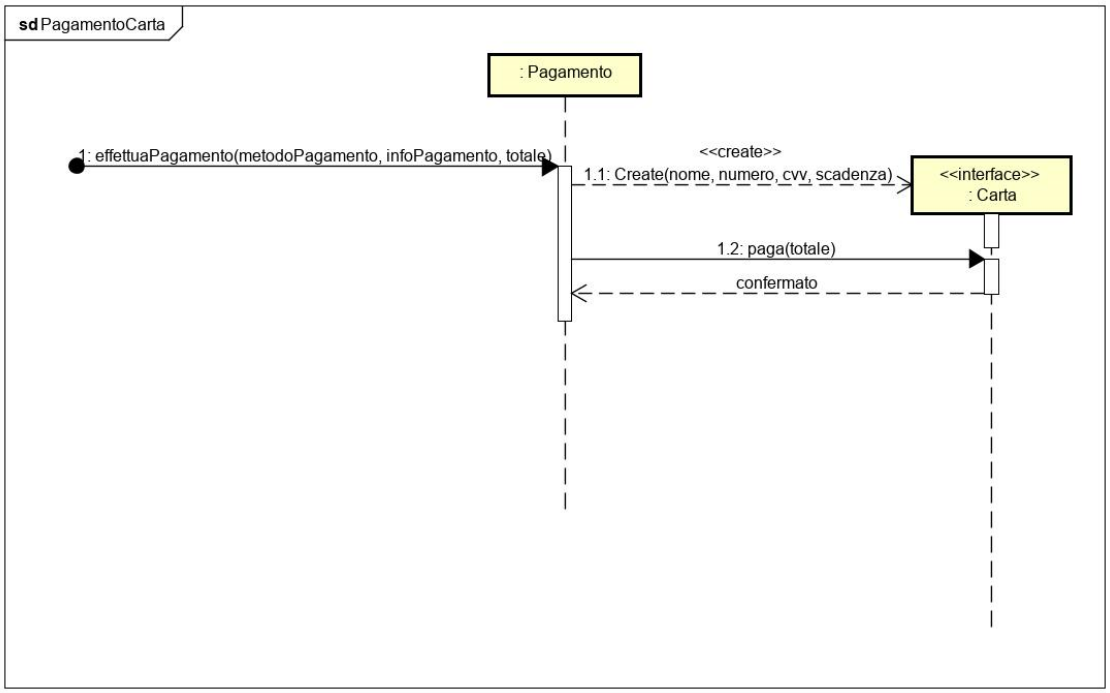
Visto che gli scenari alternativi prevedono l'uso di diversi metodi di pagamenti, i quali vengono realizzati attraverso l'ausilio del pattern factory method e strategy per la gestione dell'algoritmo da usare per il pagamento, vengono inseriti di seguito i diagrammi per ogni percorso alternativo.



- Caso PayPal

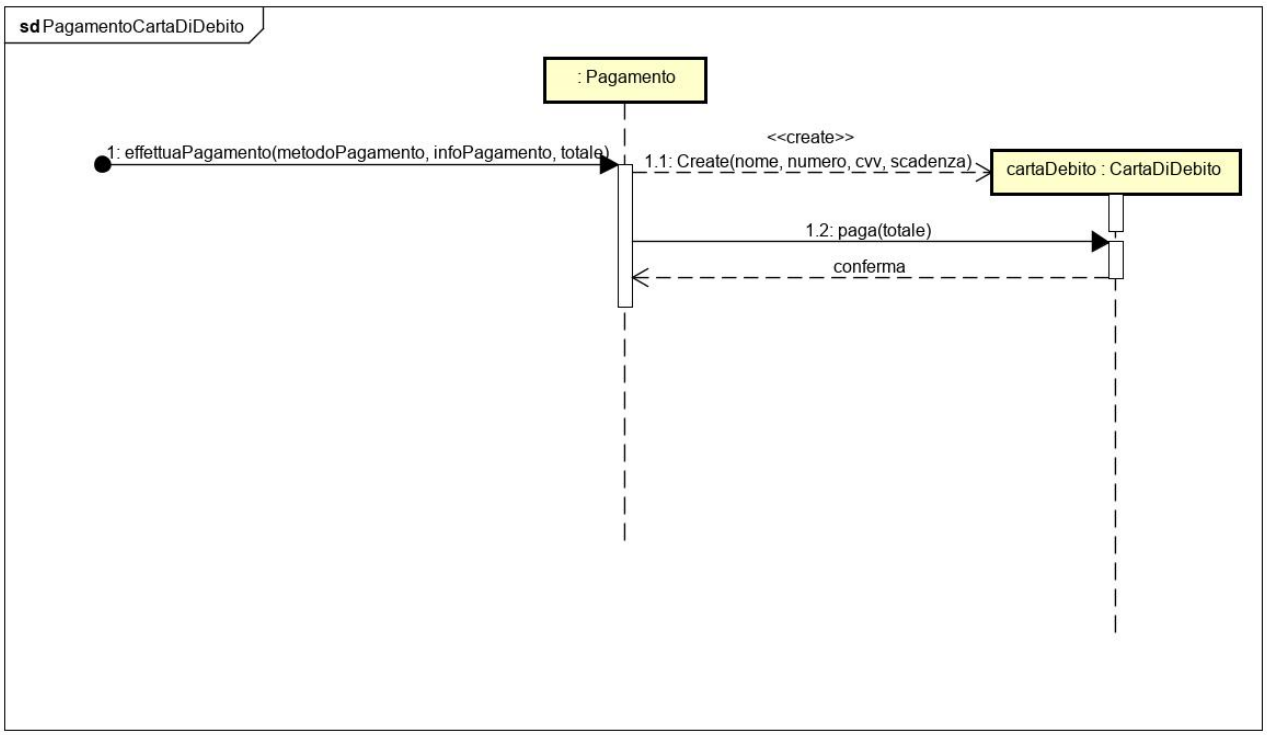


- Caso Carta

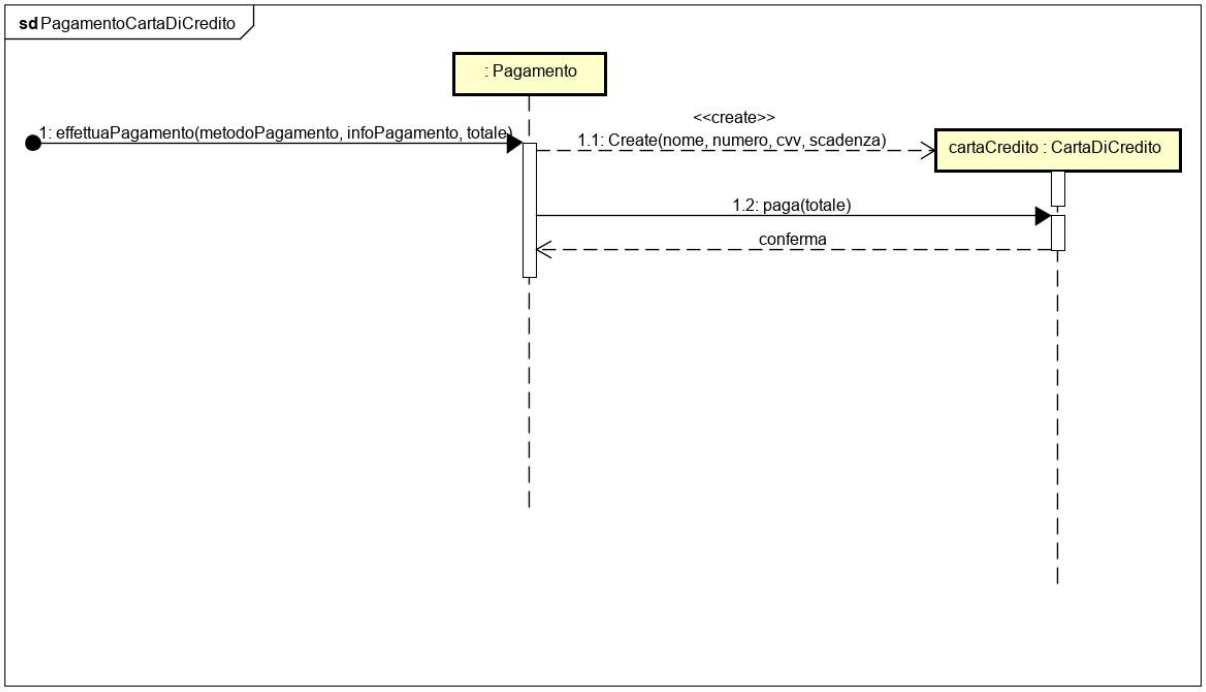


Visto l'uso del polimorfismo nel caso di Carta, essendo Carta una classe astratta che ha le sottoclassi concrete CartaDiCredito e CartaDiDebito, vengono realizzati i diagrammi per ogni caso polimorfico.

  - CartaDiDebito

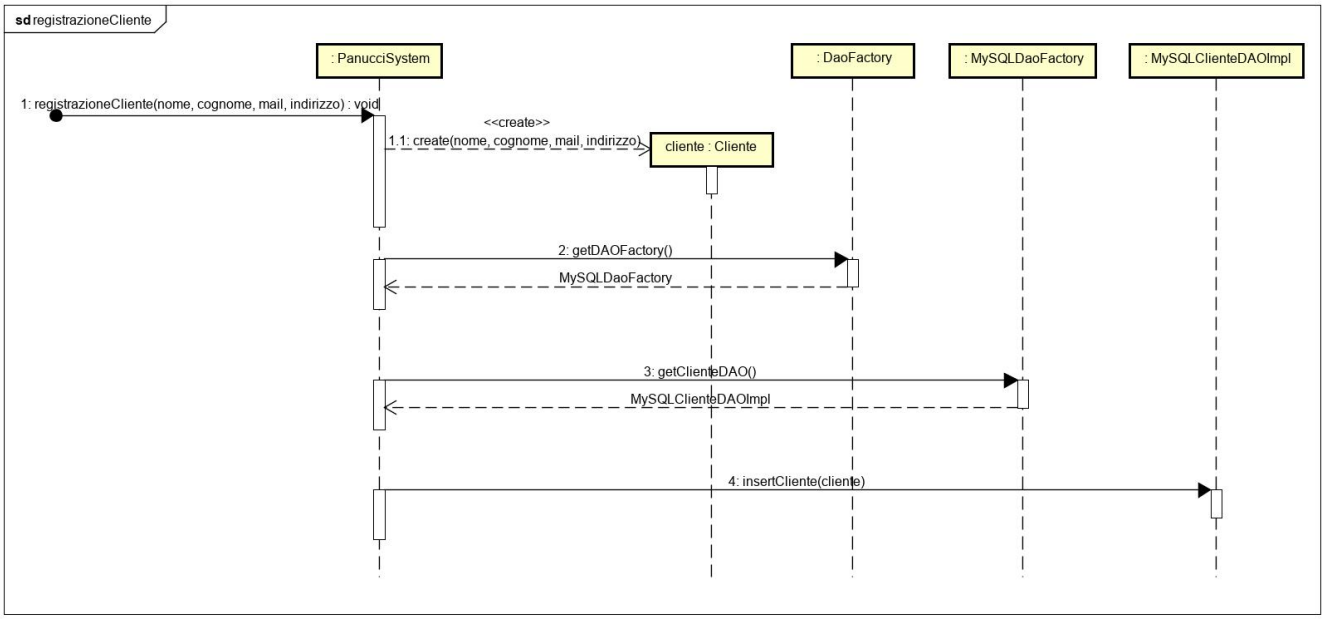


- CartaDiCredito



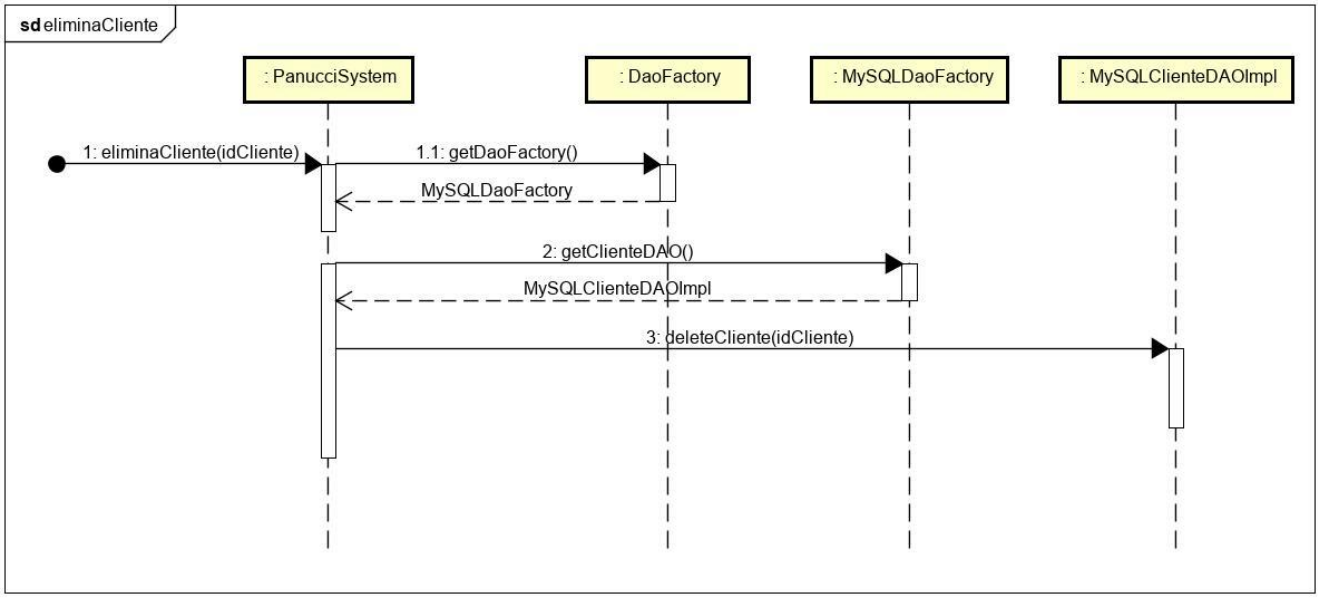
- Aggiunta dei casi d'uso CRUD UC6 e UC8

registrazioneCliente(nome:String, cognome: String, mail: String, indirizzo:String)

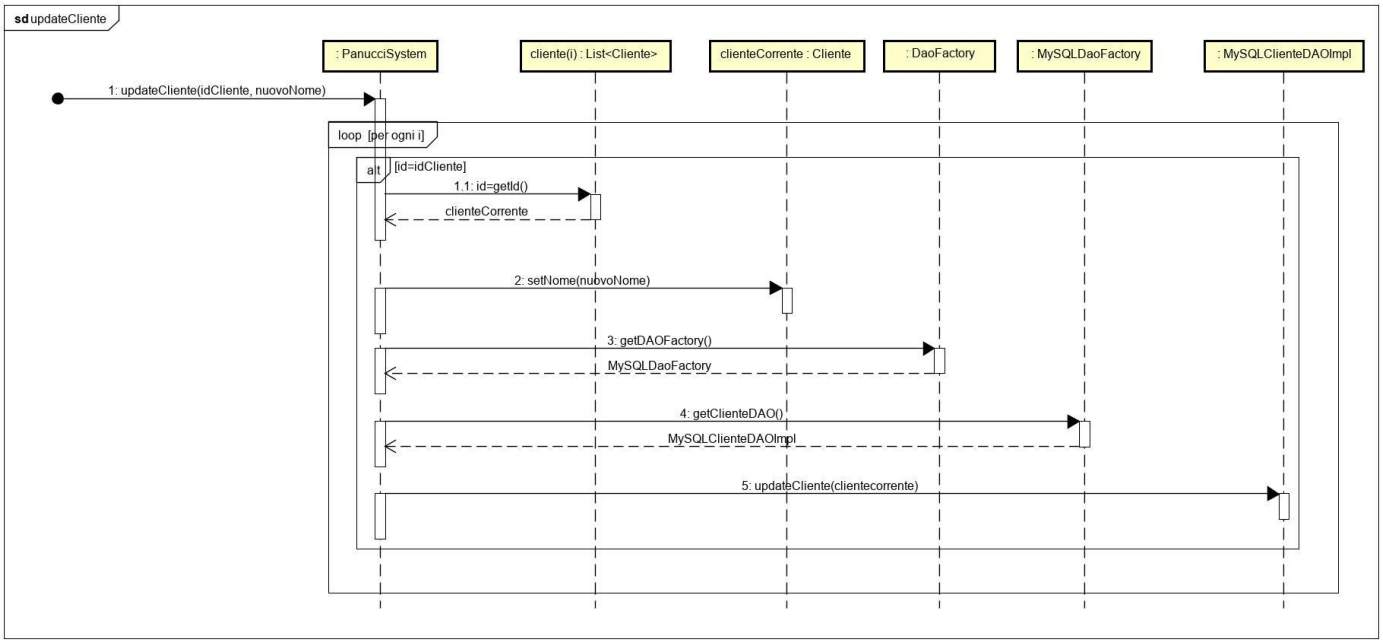


L’operazione di sistema registrazioneCliente è stata rivista in seguito all’introduzione della gestione della persistenza con DAO.

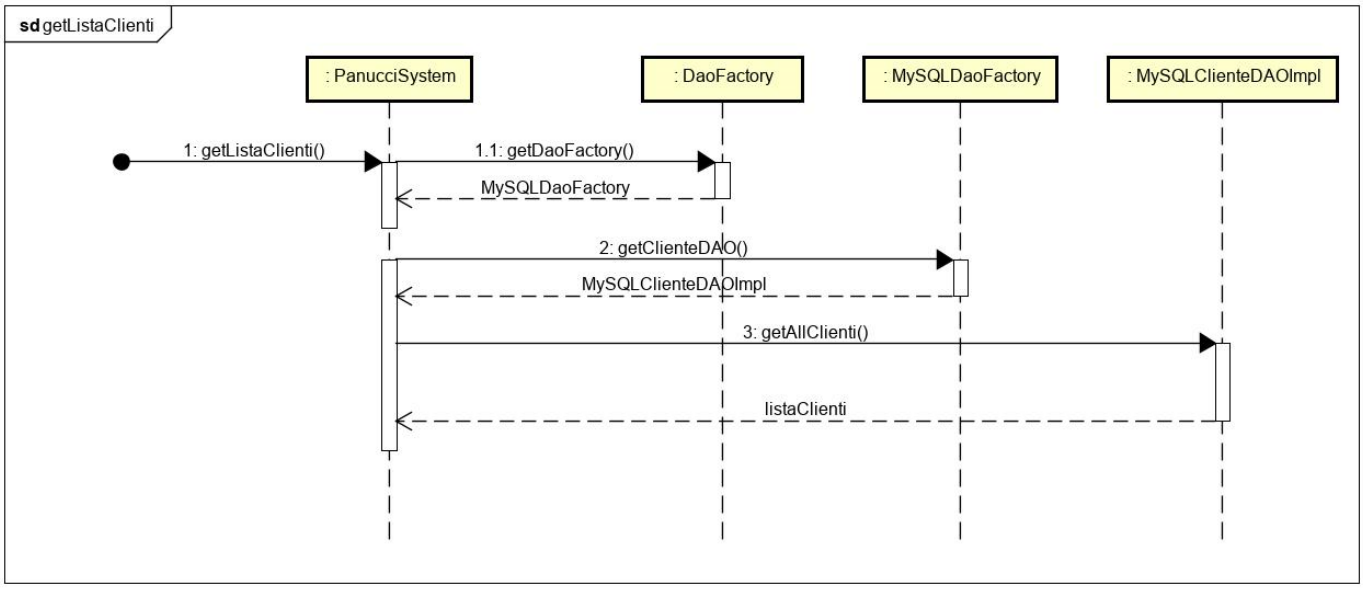
eliminaCliente(idCliente:Integer)



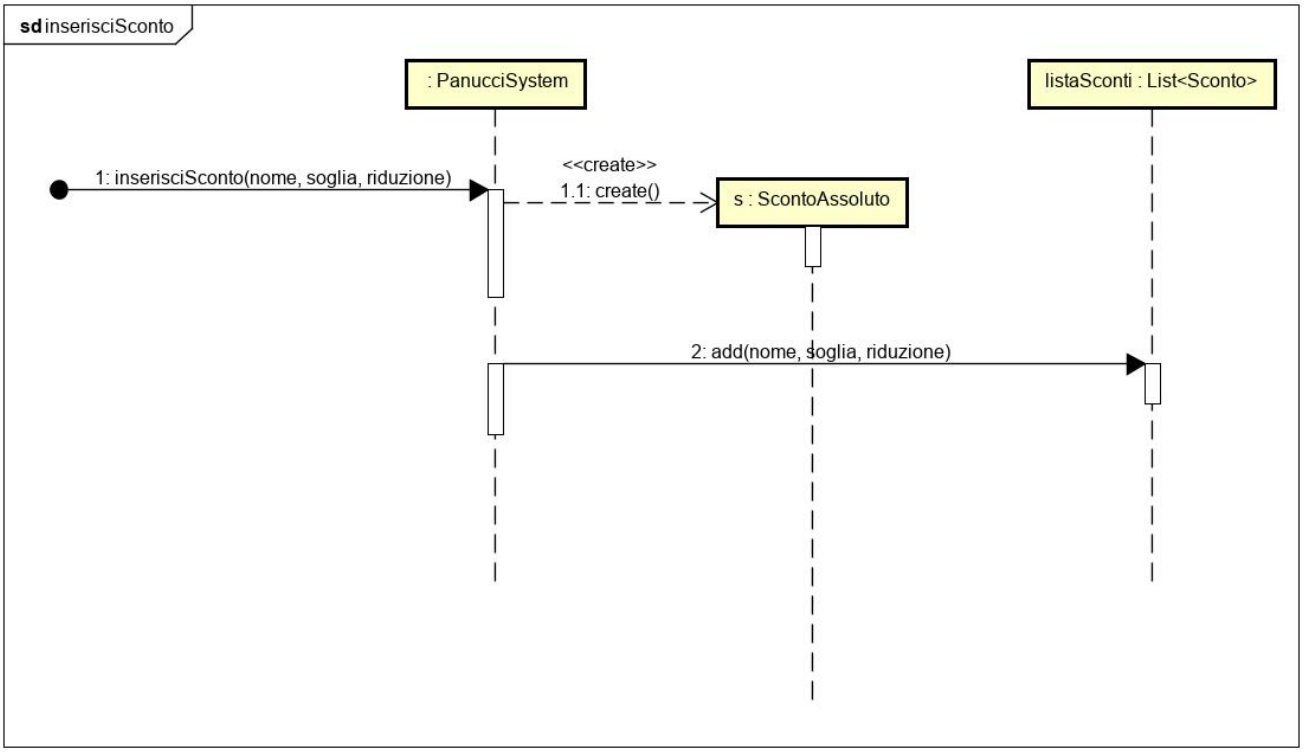
modificaCliente(idCliente:Integer, nuovoNome:String)



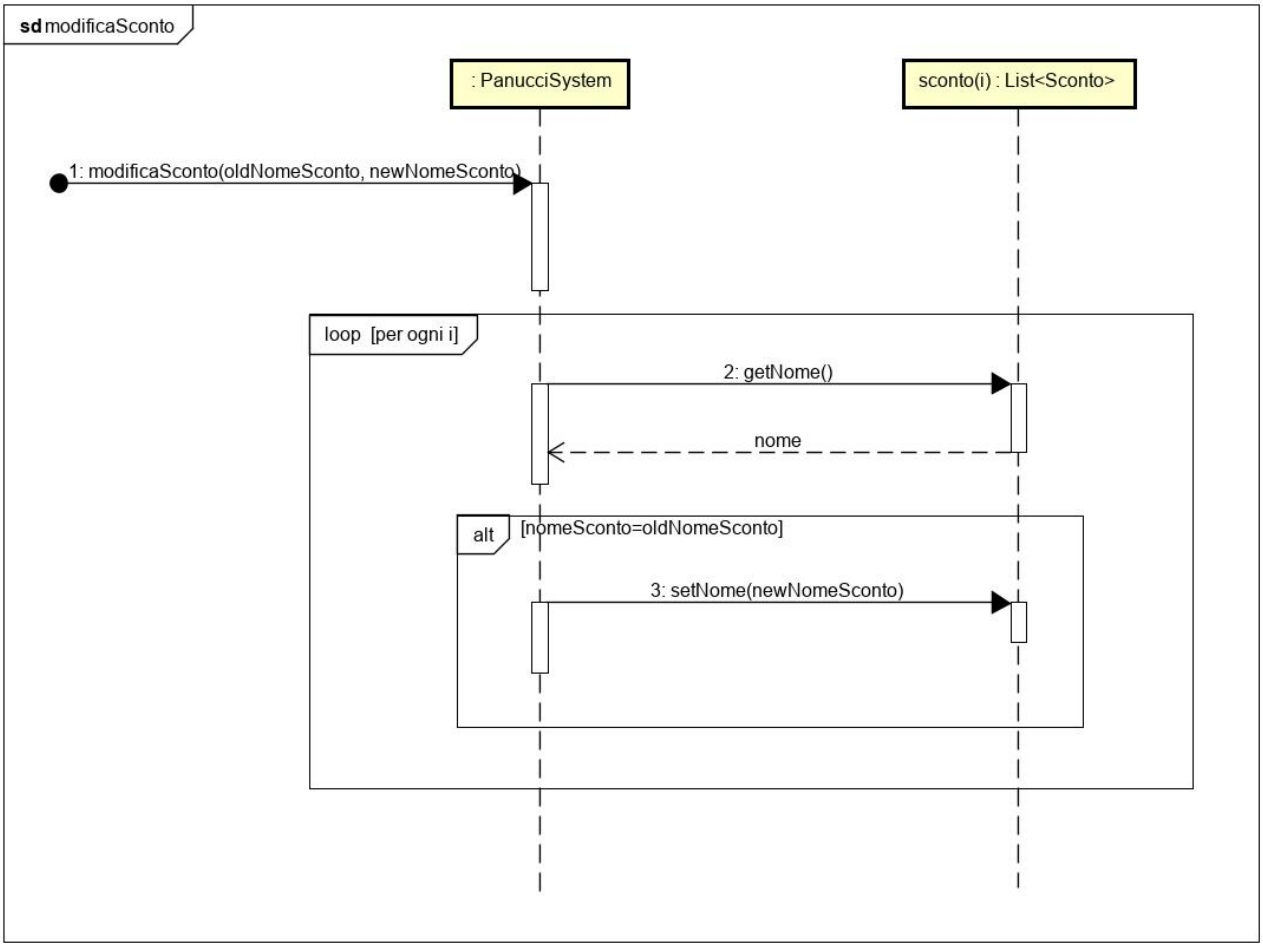
getListaClienti()



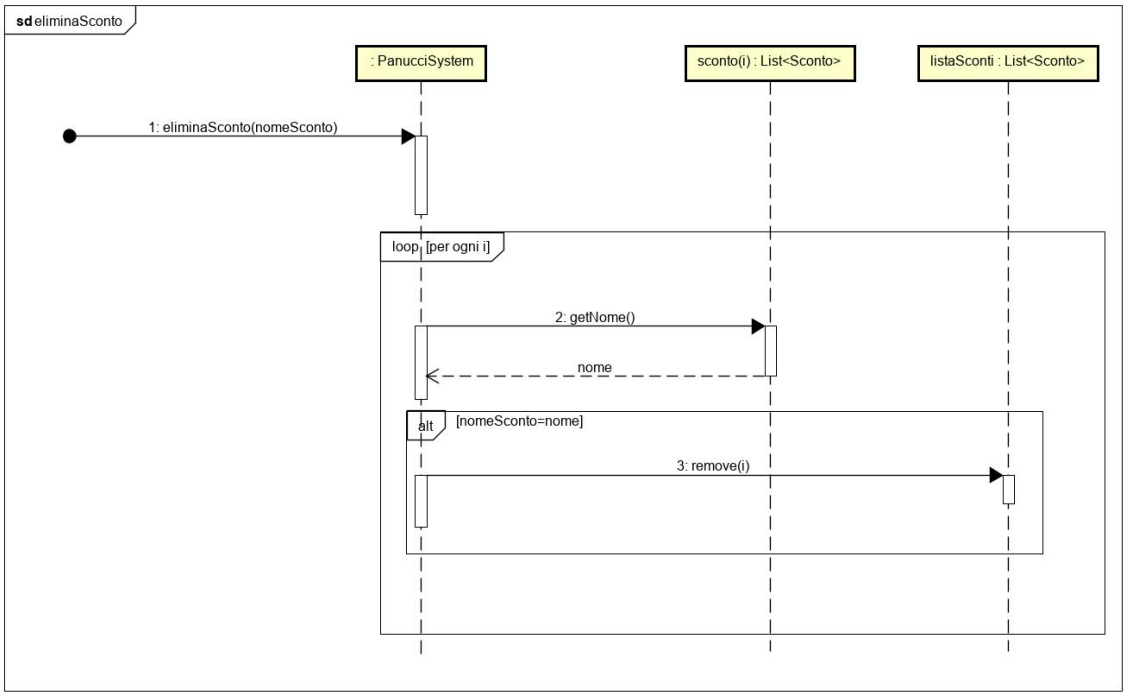
inserisciSconto(nome:String, soglia:Float, riduzione:Float)



modificaSconto(oldNomeSconto:String, newNomeSconto:String)



eliminaSconto(nomeSconto:String)



4. Testing

4.1. Introduzione

La fase di testing ha il compito di testare la componente software per scoprire eventuali errori presenti. Quindi il fine del test è quello di scovare gli errori e non quello di dimostrare l'assenza di errori nel codice. Il test si realizza usando una serie di possibili ingressi e confrontandoli con un risultato atteso, detto ciò per un test che sia esaustivo sarebbe necessario eseguire tutti i possibili ingressi e questo non è ovviamente attuabile quando vi sono molti possibili valori in ingresso. I test si dividono essenzialmente in tre categorie:

- Test che non usano il calcolatore: sono dei test che non usano il calcolatore bensì gruppi di persone e simulano il comportamento del programma dopo averlo visionato.
- Test a scatola bianca (strutturale): usa la struttura del controllo dei programmi per ricavare i casi di prova, rende perciò fattibile il test di diverse parti del programma e può trovare errori presenti in casi particolari.
- Test a scatola nera (funzionale): sfrutta la definizione del comportamento input-output dei programmi per ricavare i casi di prova, non è necessario conoscere la struttura interna del programma.

I test a scatola nera sono complementari ai test a scatola bianca e non sostitutivi in quanto i test a scatola bianca si occupano di testare i singoli componenti e non l'interezza del programma.
Tra le moltitudini di test abbiamo i test unitari che vengono eseguiti per ogni modulo o blocco di codice del programma mentre viene sviluppato, questo è un test che viene svolto principalmente dal programmatore. Esistono molti tool per i diversi linguaggi di programmazione riguardanti i test unitari, in Java il più famoso è JUnit.

Nel progetto il testing è stato realizzato attraverso l'uso dei test unitari eseguiti con JUnit seguendo un approccio di tipo Bottom-Up.

4.2. Individuazione dei casi di test e Testing Unitario

Nel nostro caso abbiamo testato solo una parte dei metodi presenti nel progetto e li elencheremo di seguito. In modo particolare ci siamo focalizzati sul test dei metodi costruttori delle classi e i relativi metodi ponderanti.

Li elenchiamo qui di seguito:

-Ingrediente

- getNome
- getCostoIngrediente
- getIngrediente

-Cliente

- getNome
- getCognome
- getEmail
- getIndirizzo

-Comanda

- getDataPrenotazione
- getIndirizzoConsegna
- getCliente
- calcolaTotale
- effettuaPagamento
- associaSconto

-Sconto Assoluto

- calcolaImportoScontato
  - questo metodo decrementa l'importo da pagare di una quantità fissata al verificarsi del superamento della soglia stabilita.

-Sconto Percentuale

- calcolaImportoScontato
  - questo metodo decrementa l'importo da pagare di una quantità percentuale al verificarsi del superamento della soglia stabilita.

Dopo il test sui costruttori abbiamo testato i metodi relativi allo sconto Assoluto e Percentuale considerando i tre casi possibili (importo sopra la soglia, importo sotto soglia, importo uguale alla soglia). Grazie alla fase di test ci siamo resi conto di un errore relativo ad un nominativo sbagliato di un parametro del costruttore.

4.3. Test di Sistema

Successivamente ai test effettuati con JUnit sono stati effettuati dei test manuali completi portando al termine tutti i casi d'uso secondo l'approccio a scatola nera. A seguito di ciò non sono stati evidenziati errori nel funzionamento o eventuali mancanze.

5. Refactoring e Conclusioni

5.1. Database e Refactoring

In questa fase è stato utilizzato il DAO (Data Access Object) per avere una integrazione con il database. È stato utilizzato MySQL lato database e il connettore JDBC.
La persistenza è stata implementata parzialmente riguardo solo alla listaClienti; il resto dei dati viene gestito in memoria.
Le classi introdotte in questa fase sono:

- DAOFactory: è una classe astratta che ha la responsabilità di creare l'opportuna classe concreta.
- MySQLDAOFactory: estende la classe DAO Factory e ha la responsabilità di effettuare la connessione al DB MySQL. La Factory MySQLDAOFactory avrà anche il compito di istanziare la classe MySQLClienteDAOImpl.
- Cliente DAO: è l'interfaccia che contiene tutte le signature dei metodi che si useranno per le query sulla classe Cliente.
- MySQLClienteDAOImpl: contiene il codice per effettuare le query al DB.

5.2. Test di Accettazione

Per concludere è stato eseguito un test globale di accettazione sia simulando di essere un amministratore che un Cliente portando quindi a termine le relative funzionalità implementate.