

INDICE

INTRODUZIONE	1
1. Analisi UC4	6
1.1. Elaborazione.....	6
1.1.1. Modello di dominio	6
1.1.2. Diagramma di sequenza di sistema	7
1.1.3. Contratti delle operazioni	7
1.2. Progettazione.....	9
1.2.1. Diagrammi di sequenza	9
2. Analisi UC5	10
2.1. Elaborazione.....	10
2.1.1. Modello di Dominio	10
2.1.2. Diagramma di sequenza di sistema	11
2.1.3. Contratti delle operazioni	11
2.2. Progettazione.....	14
2.2.1. Diagrammi di sequenza	14
3. Modello delle classi	16
4. Refactoring.....	17
4.1. Progettazione Iterazione 1	18
4.1.1. UC1: Inserisci Sala	18
4.1.2. UC3: Inserisci Proiezione.....	19
4.2. Progettazione Iterazione 2	20
4.2.1. UC5: Vendita Biglietto.....	20

4.3.	Caso d'uso d'avviamento	21
4.4.	Diagramma delle classi	22

INTRODUZIONE

Per l'iterazione 2 sono stati scelti i seguenti requisiti:

- Lo scenario principale di successo del caso d'uso **UC4: Creazione tessera**
- Lo scenario principale di successo del caso d'uso **UC5: Vendita biglietto**, applicando le regole di dominio R1, R2 e R6

Di seguito vengono riportati i casi d'uso scelti:

UC4: Creazione tessera

1. L'Addetto interagisce col Sistema al fine di inserire i dati del cliente
2. Il Sistema chiede i dati del cliente
3. L'Addetto inserisce, in ordine:
 - a. Nome
 - b. Cognome
 - c. Data di nascita
 - d. Codice Fiscale
4. Il sistema mostra all'Addetto un riepilogo dei dati inseriti e chiede conferma per concludere l'operazione
5. Alla conferma da parte dell'Addetto, il Sistema completa la creazione della tessera generando per questa un ID univoco e inizializzando a 0 il numero di punti
6. Il Sistema manda in stampa la carta con le relative informazioni inserite

Scenari alternativi

- 1a. Il cliente è già registrato

1. L'Addetto inserisce le informazioni del cliente
2. Il Sistema notifica che il codice fiscale inserito è già in uso e restituisce il nominativo

1b. Il cliente è minorenni

1. L'Addetto inserisce le informazioni anagrafiche del cliente
 2. Il Sistema impedisce all'Addetto di proseguire in quanto risulta che il cliente è minorenni
1. L'addetto viene riportato nel menu principale

UC5: Vendita biglietto

Nome caso d'uso	UC4: Vendita biglietto
Portata	Applicazione ParadoxEntertainment
Livello	Obiettivo utente
Attore primario	Addetto alla cassa
Parti interessate	<ul style="list-style-type: none"> • Addetto, che procede alla vendita del biglietto e al pagamento dello stesso • Cliente, che ottiene il biglietto per vedere lo spettacolo
Pre-condizioni	<ul style="list-style-type: none"> • L'Addetto deve essere loggato al Sistema

Garanzia di successo	<ul style="list-style-type: none"> • Il biglietto viene erogato e consegnato al cliente • Se in possesso della tessera fedeltà, il cliente ottiene un accumulo punti pari a 10 volte il prezzo del biglietto • Lo spettacolo conterà un posto in meno, standard o VIP
Scenario principale	<ol style="list-style-type: none"> 1. L'Addetto comunica al sistema di inizializzare una nuova transazione 2. Il sistema mostra all'Addetto l'elenco dei film per i quali ci sono in programma delle proiezioni 3. L'Addetto comunica la pellicola da visionare 4. Il Sistema restituisce tutte le proiezioni di quel film nelle diverse sale e fasce orarie, con il numero dei posti liberi rimanenti 5. L'Addetto sceglie una specifica proiezione 6. L'Addetto specifica se la poltrona richiesta è standard o VIP, se il cliente ha diritto ad un biglietto ridotto (bambini, studenti, over-65) 7. Il sistema calcola il costo totale del biglietto e lo comunica all'Addetto 8. Il Sistema chiede all'Addetto se il cliente esibisce una tessera fedeltà 9. Se il cliente è in possesso di una tessera fedeltà, l'addetto può: <ol style="list-style-type: none"> a. Effettuarne la lettura tramite lettore di codice a barre

	<p>b. Inserire manualmente il codice univoco della carta</p> <p>10. L'Addetto conferma il pagamento</p> <p>11. Il Sistema emette il biglietto, se esibita una tessera carica i punti della vendita e conferma l'operazione</p>
Estensioni	<p>a. Il sistema non ha trovato nel database l'ID della tessera inserita</p> <ol style="list-style-type: none"> 1. L'addetto inserisce manualmente l'ID della tessera fedeltà 2. Il sistema non trova una corrispondenza tra l'ID inserito dall'addetto e gli ID presenti nel database 3. Il sistema chiede se si vuole ritentare l'inserimento manuale dell'ID o se saltare il passaggio <p>b. Non risultano posti disponibili per quel determinato spettacolo</p> <ol style="list-style-type: none"> 1. L'addetto seleziona lo spettacolo, per il quale non ci sono più posti disponibili 2. Il Sistema comunica all'addetto che non ci sono abbastanza posti disponibili e lo riporta al menu principale <p>c. Viene selezionata una pellicola per la quale non ci sono proiezioni in programma</p> <ol style="list-style-type: none"> 1. Il Sistema notifica che non ci sono spettacoli disponibili per quel film e annulla la vendita

	<p>d. Il cliente riscatta il biglietto gratuitamente pagandolo con i punti accumulati sulla tessera</p> <ol style="list-style-type: none"> 1. Il cliente esibisce la tessera fedeltà 2. L'addetto effettua la lettura della tessera fedeltà 3. Il sistema verifica che è stato accumulato un numero sufficiente di punti per poter riscattare il biglietto gratuitamente, e in tal caso, viene comunicato all'addetto <p>e. Il Sistema si interrompe per un malfunzionamento</p> <ol style="list-style-type: none"> 1. Durante la procedura, si subisce una brusca interruzione 2. Il Sistema perde i dati finora immessi e si riavvia
--	---

1. Analisi UC4

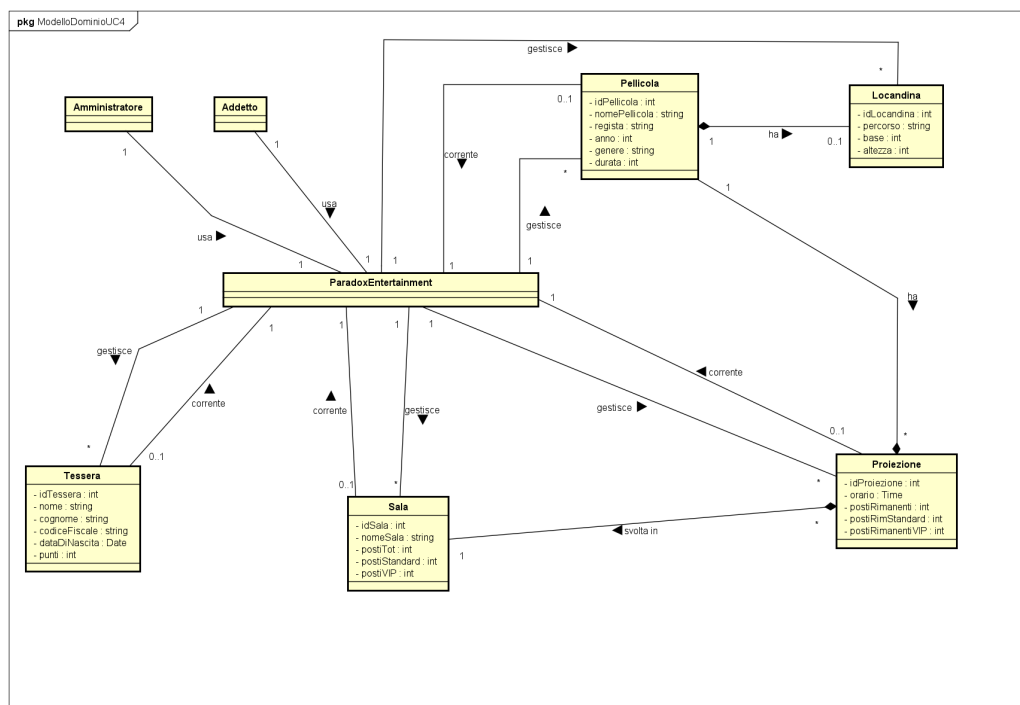
Dallo scenario principale di successo del caso d'uso 4 (Creazione tessera) è possibile identificare le seguenti classi concettuali:

- **Addetto**, l'attore principale che interagisce col sistema
- **Paradox Entertainment**, che rappresenta il sistema
- **Tessera**, che è l'elemento fisico rilasciato poi all'utente richiedente

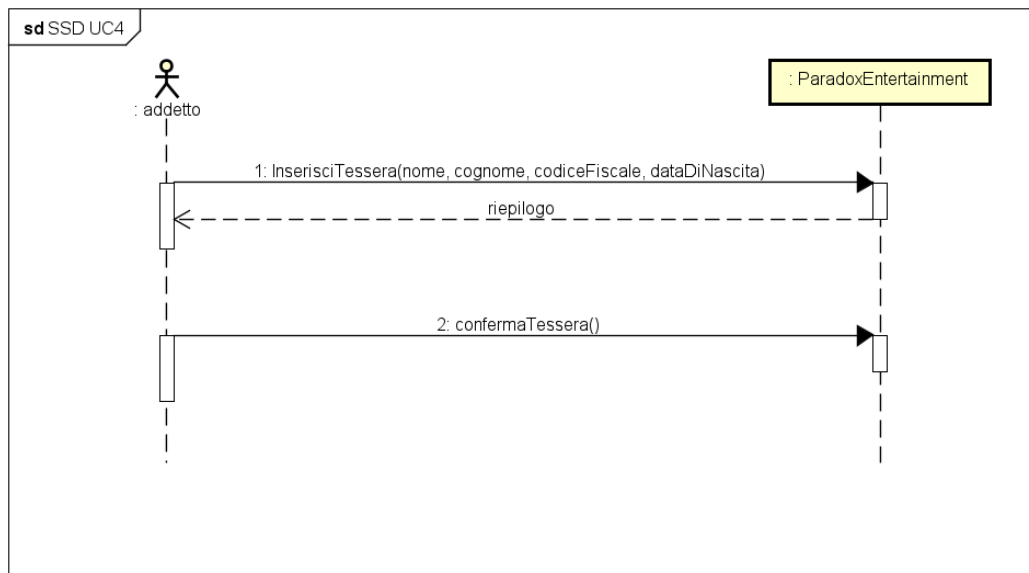
1.1. Elaborazione

1.1.1. Modello di dominio

Il modello di dominio aggiornato con questo caso d'uso dopo l'iterazione 1 è il seguente:



1.1.2. Diagramma di sequenza di sistema



1.1.3. Contratti delle operazioni

Operazione	InserisciTessera (nome: string, cognome: string, codiceFiscale: string, dataDiNascista: Date)
Riferimenti	Caso d'uso UC4: Crea Tessera
Pre-condizioni	<ul style="list-style-type: none">L'Addetto è già loggato nel sistema
Post-condizioni	<ul style="list-style-type: none">L'istanza della classe Tessera: T è stata creata

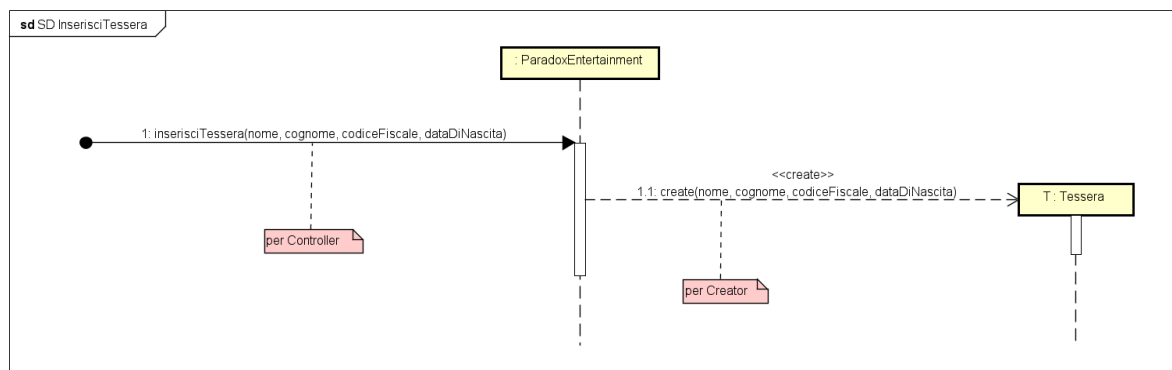
	<ul style="list-style-type: none"> • Gli attributi di T sono stati definiti • Viene associata l'istanza di T al Sistema mediante "corrente"
--	---

Operazione	ConfermaTessera ()
Riferimenti	Caso d'uso UC4: Crea tessera fedeltà
Pre-condizioni	<ul style="list-style-type: none"> • È stato creato un oggetto T
Post-condizioni	<ul style="list-style-type: none"> • Viene generato l'ID associato all'oggetto T • L'oggetto T creato viene aggiunto alla lista delle tessere esistenti

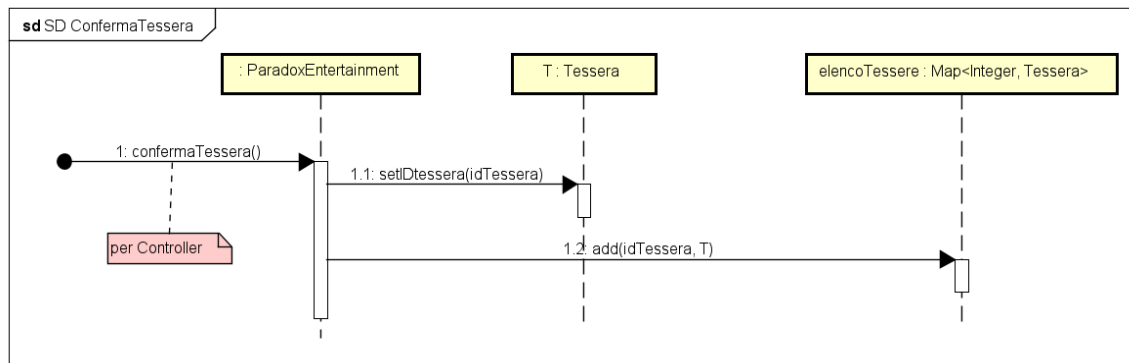
1.2. Progettazione

1.2.1. Diagrammi di sequenza

- inserisciTessera



- confermaTessera



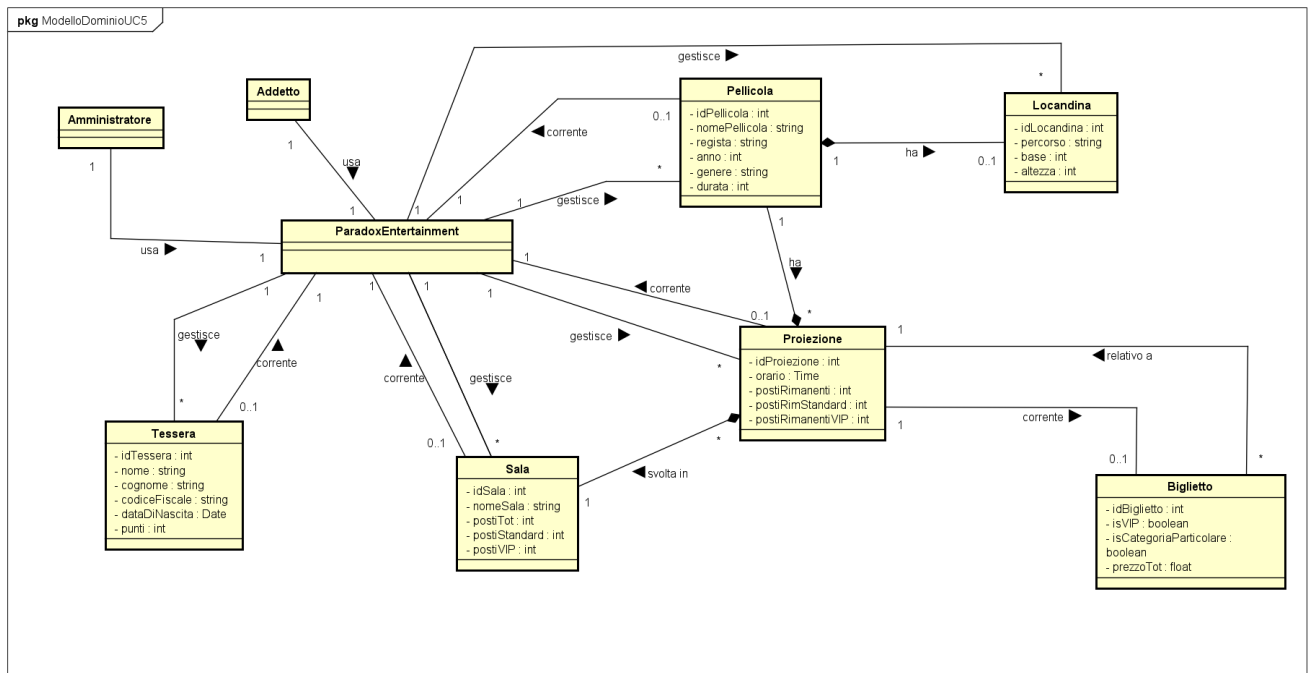
2. Analisi UC5

Dallo scenario principale di successo del caso d'uso 5 (Vendita Biglietto) è possibile identificare le seguenti classi concettuali:

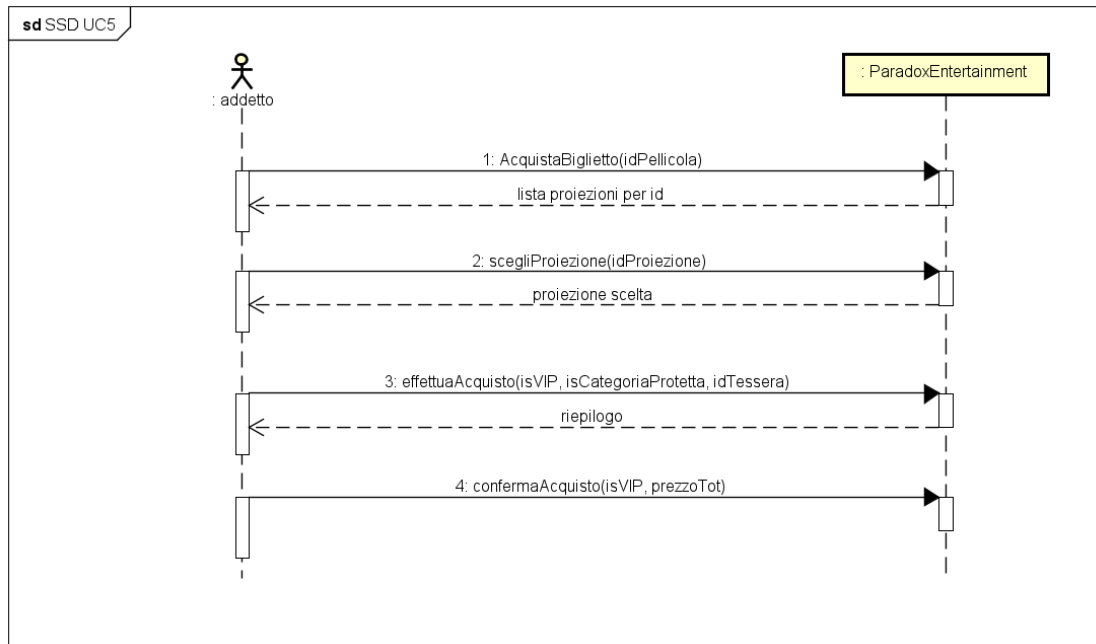
- **Addetto**, l'attore principale che interagisce col sistema
- **Paradox Entertainment**, che rappresenta il sistema
- **Biglietto**, che rappresenta il mezzo fisico rilasciato al cliente dopo il pagamento, che consente di accedere ad uno spettacolo

2.1. Elaborazione

2.1.1. Modello di Dominio



2.1.2. Diagramma di sequenza di sistema



2.1.3. Contratti delle operazioni

Operazione	AcquistaBiglietto (idPellicola: int)
Riferimenti	UC5: AcquistaBiglietto
Pre-condizioni	<ul style="list-style-type: none"> • L'addetto è loggato nel sistema • Esiste almeno una pellicola P • Esiste almeno una sala S • Esiste almeno una Proiezione Pr relativa alla pellicola P tenuta nella sala S

Post-condizioni	<ul style="list-style-type: none"> • Vengono stampati le proiezioni associate all'ID della pellicola
-----------------	---

Operazione	ScegliProiezione(idProiezione: int)
Riferimenti	UC5: AcquistaBiglietto
Pre-condizioni	<ul style="list-style-type: none"> • L'addetto è loggato nel sistema • È stata stampata una lista di Proiezioni associata ad una Pellicola
Post-condizioni	<ul style="list-style-type: none"> • Viene selezionata una Proiezione Pr

Operazione	EffettuaAcquisto(idTessera: int, isVIP: boolean, isCategoriaProtetta: boolean)
Riferimenti	UC5: AcquistaBiglietto
Pre-condizioni	<ul style="list-style-type: none"> • L'addetto è loggato nel sistema • È stata selezionata una proiezione Pr • È in corso la creazione di un Biglietto B
Post-condizioni	<ul style="list-style-type: none"> • Viene creato un Biglietto B • Gli attributi di B vengono inizializzati

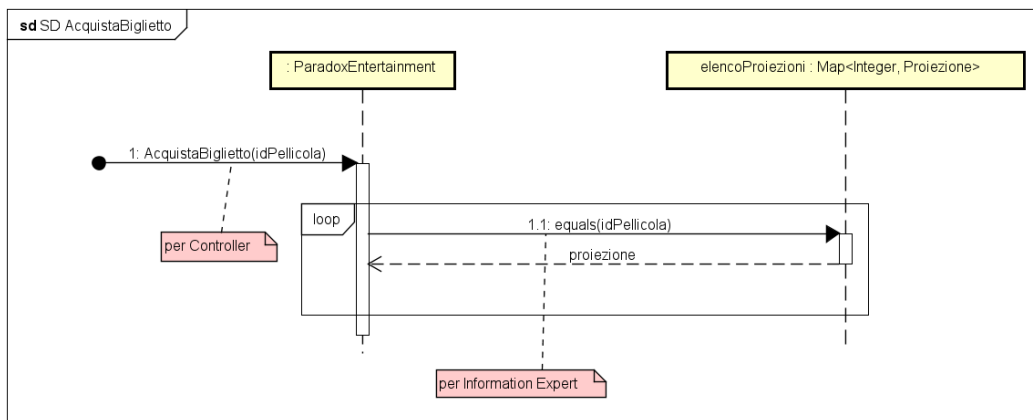
	<ul style="list-style-type: none"> Il biglietto B viene collegato a Proiezione mediante le associazioni “relativo a” e “corrente”
--	--

Operazione	ConfermaAcquisto(isVIP: Boolean, prezzoTot: float)
Riferimenti	UC5: AcquistaBiglietto
Pre-condizioni	<ul style="list-style-type: none"> L’addetto è loggato nel sistema È stato creato un biglietto B per la proiezione Pr
Post-condizioni	<ul style="list-style-type: none"> Il Biglietto B creato viene aggiunto alla lista dei biglietti venduti Viene scalato il posto dalla Proiezione associata al biglietto venduto Se il cliente è in possesso di una tessera T, vengono caricati punti sulla stessa

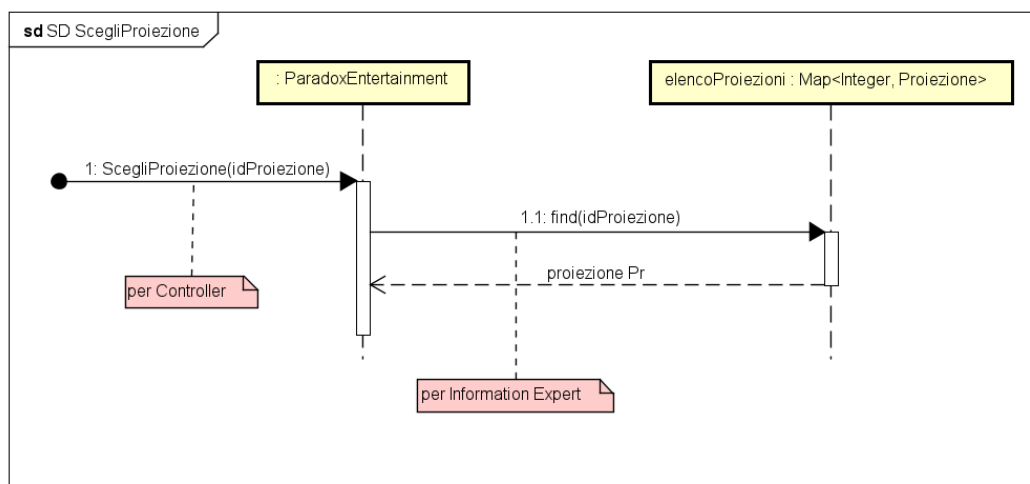
2.2. Progettazione

2.2.1. Diagrammi di sequenza

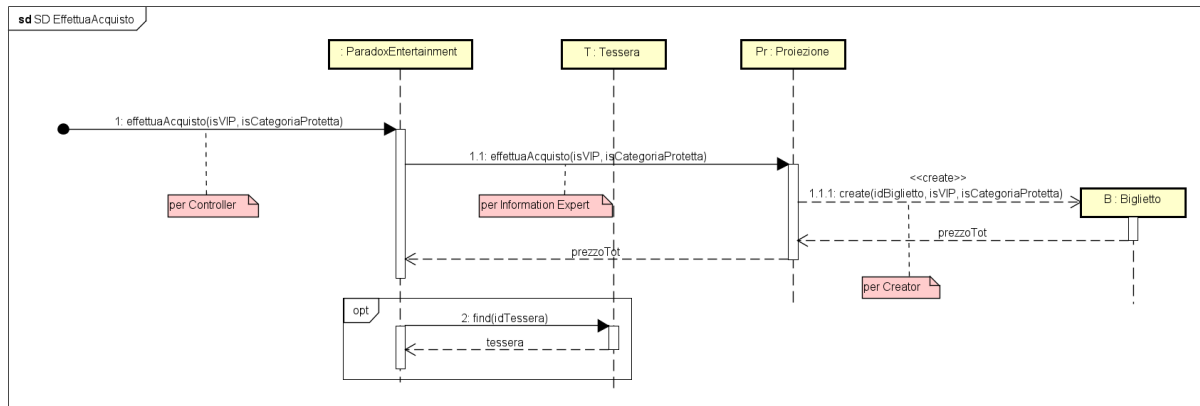
- AcquistaBiglietto



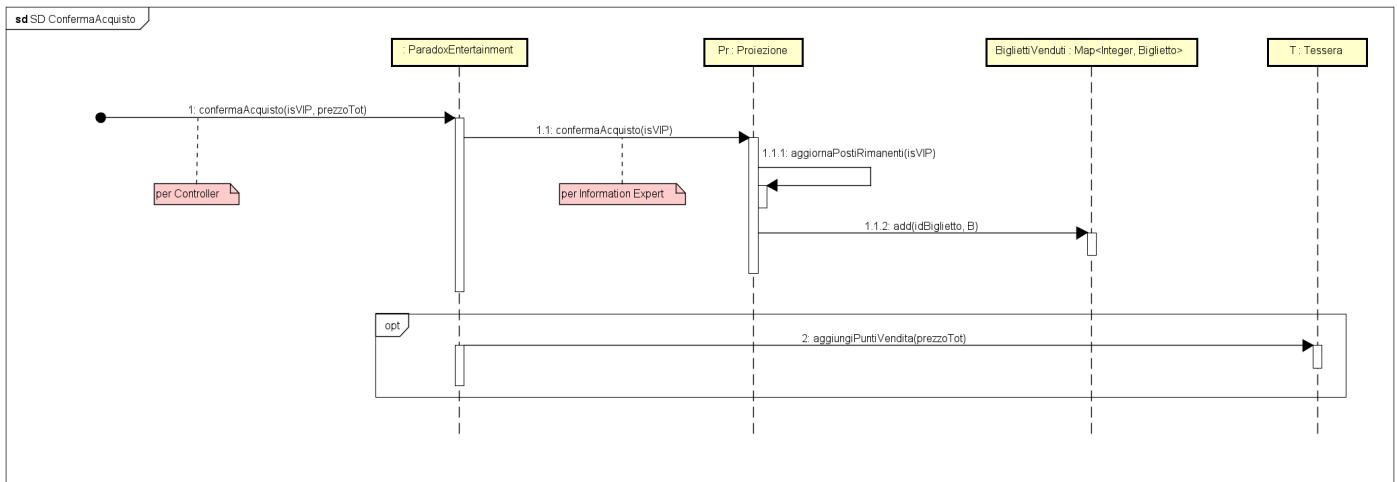
- ScegliSpettacolo



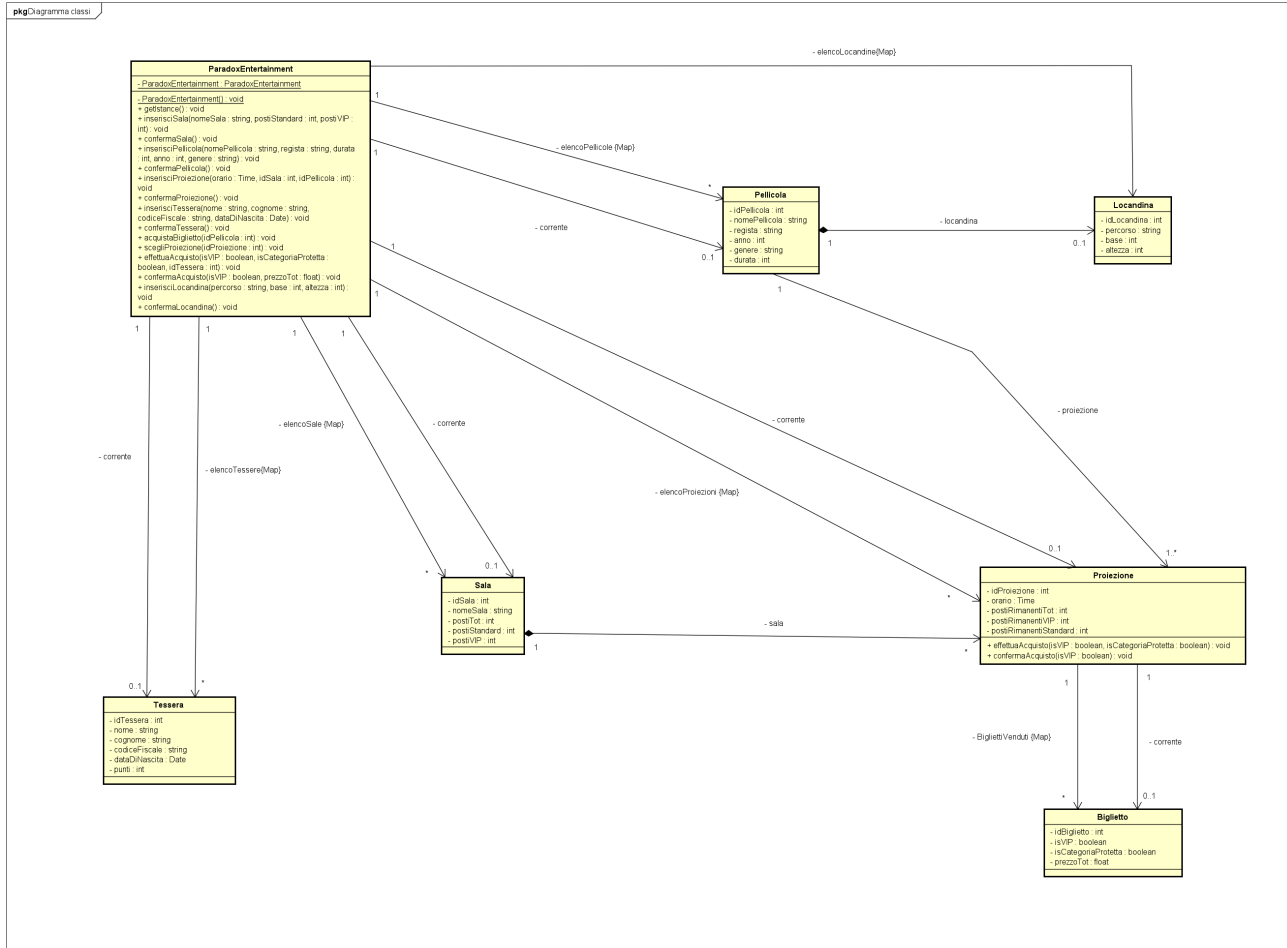
- **effettuaAcquisto**



- **confermaAcquisto**



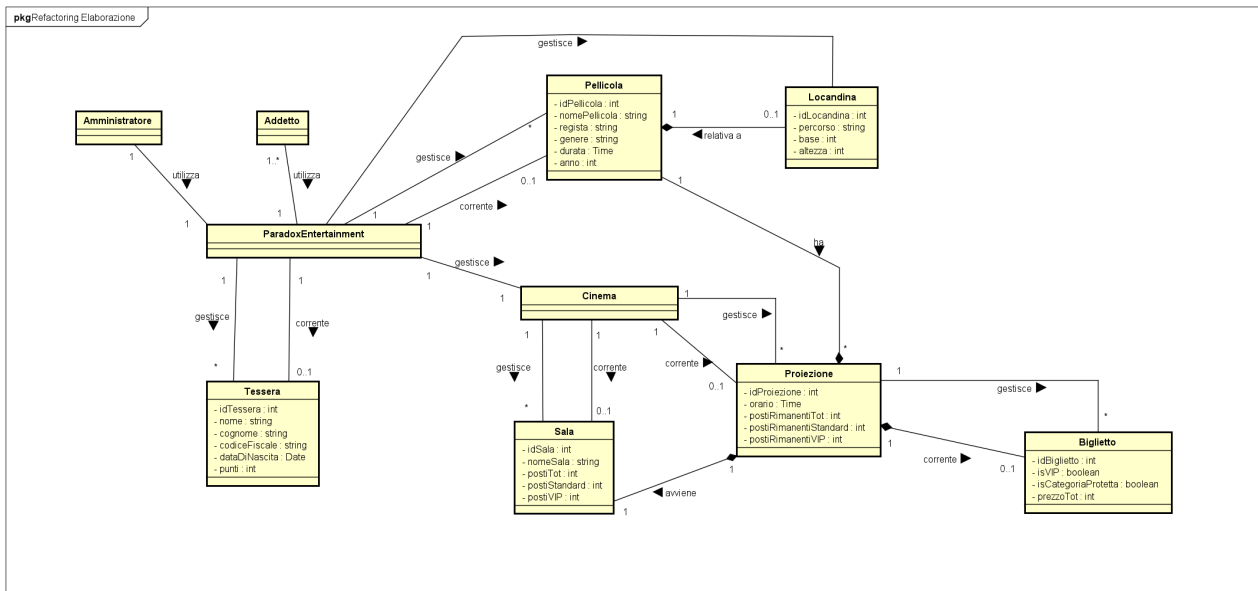
3. Modello delle classi



4. Refactoring

Dall'analisi del codice del progetto, si evince che la classe controller ParadoxEntertainment ha responsabilità eccessive, che possono essere risolte introducendo una classe Cinema, assegnando a quest'ultima la responsabilità di conoscere e gestire le Sale e le Proiezioni associate e, di conseguenza, i biglietti relative ad ogni proiezione. L'obiettivo è quello di cambiare l'organizzazione del codice senza alterarne il comportamento complessivo.

Il modello di dominio, a seguito di questa scelta, è il seguente:

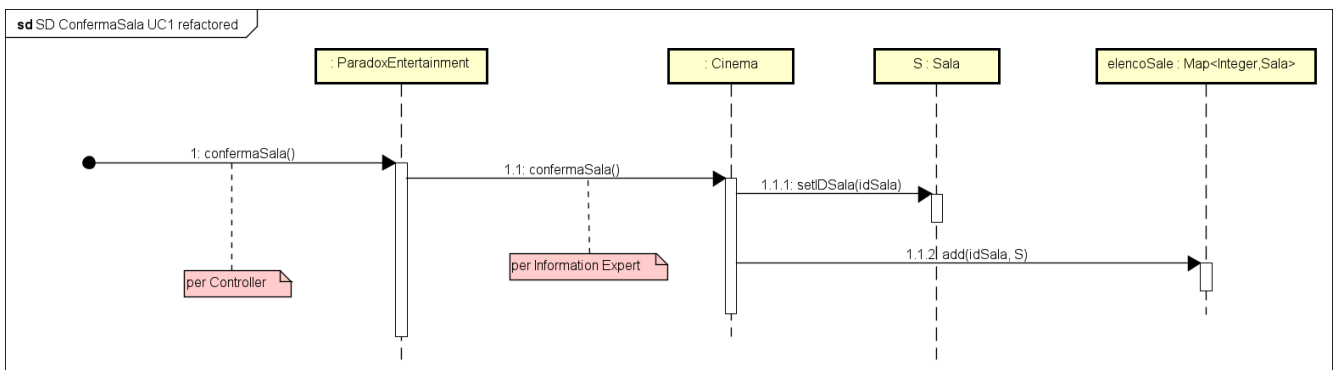
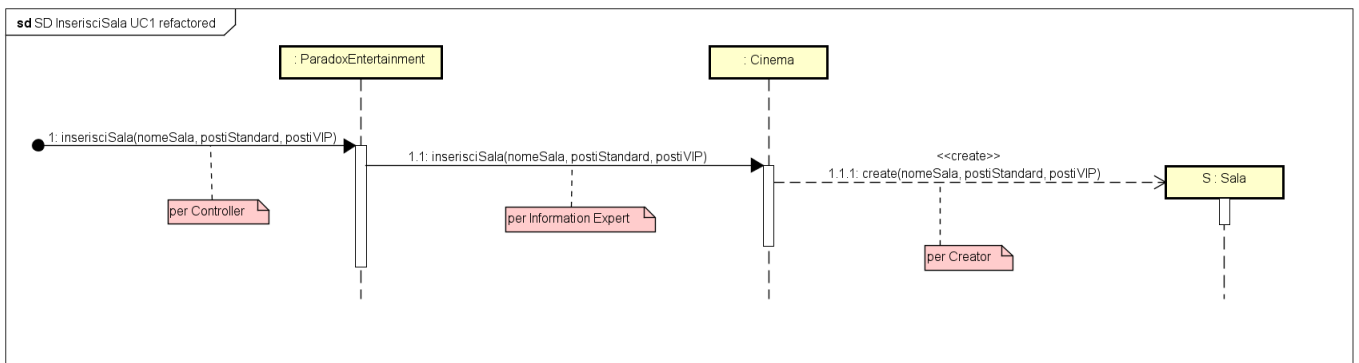


Di conseguenza, vediamo come questa modifica si ripercuote su elementi dell'Iterazione 1 e di questa qua, la 2.

4.1. Progettazione Iterazione 1

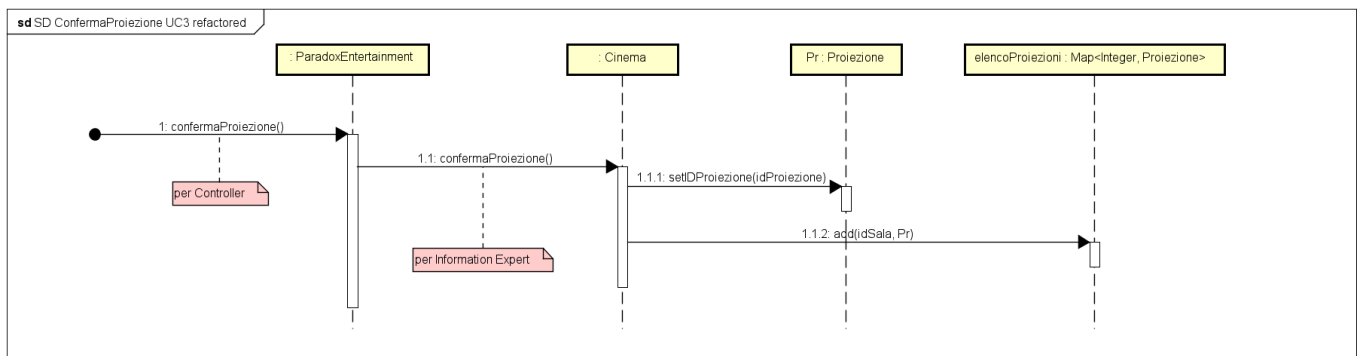
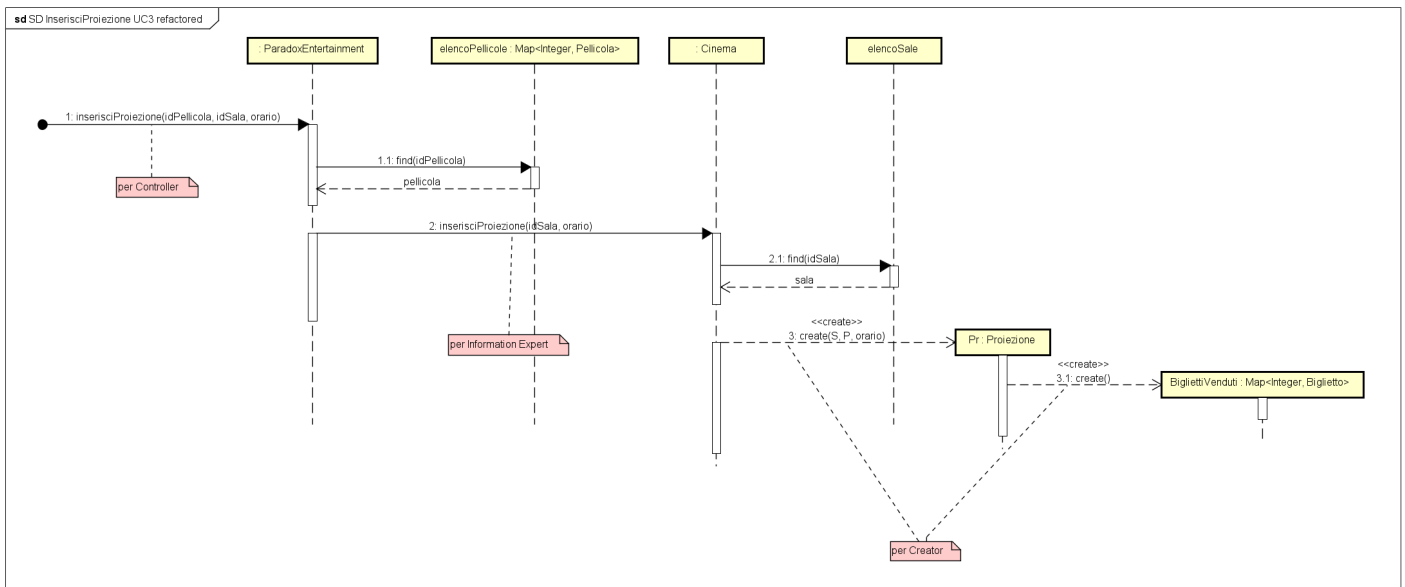
4.1.1. UC1: Inserisci Sala

Casistica relativa all'inserimento di una sala nel Sistema, ecco come viene modificata:



4.1.2. UC3: Inserisci Proiezione

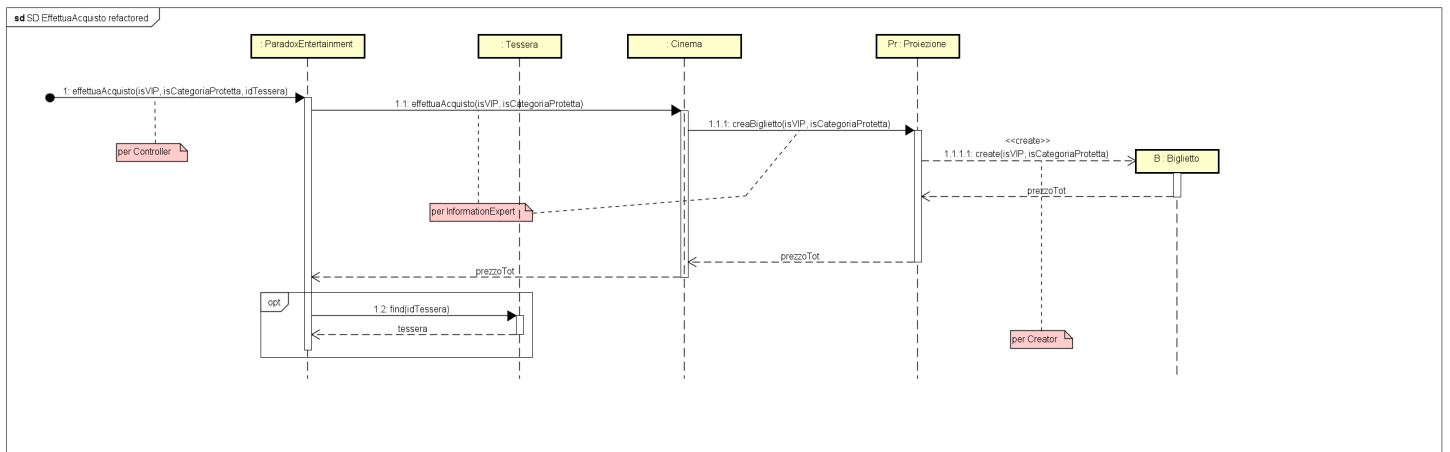
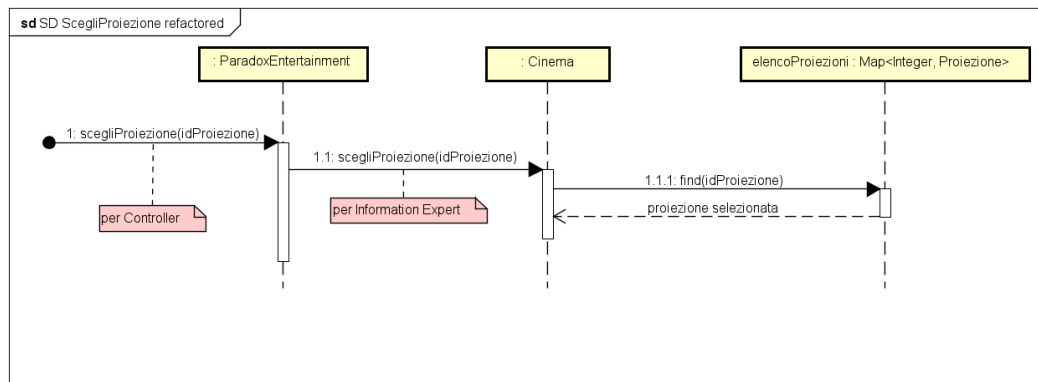
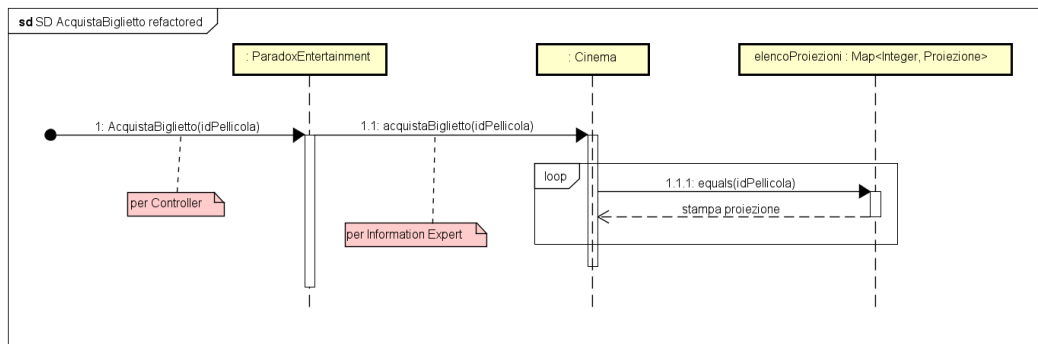
Casistica relativa all'inserimento di una Proiezione nel Sistema, che viene modificata nel seguente modo:

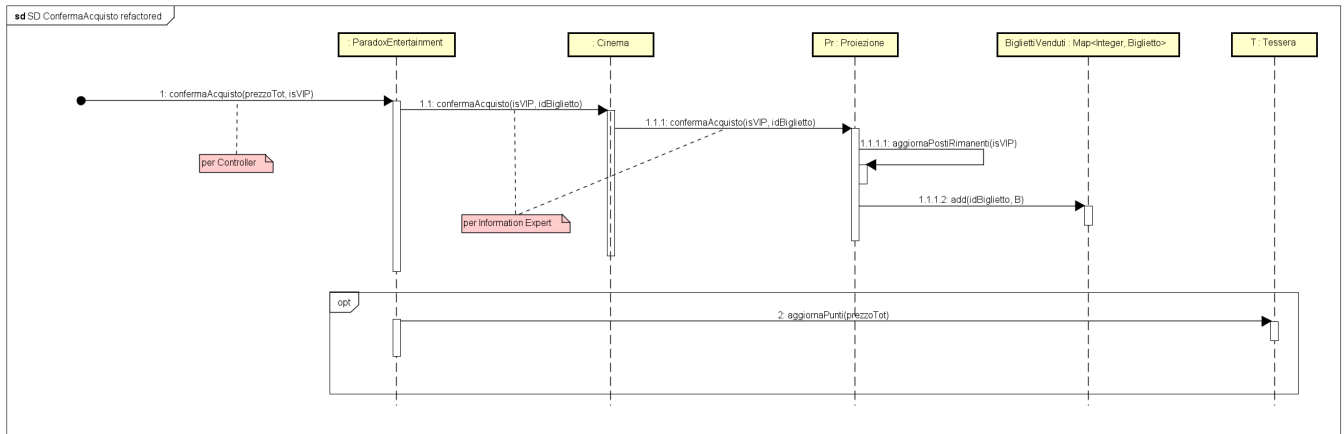


4.2. Progettazione Iterazione 2

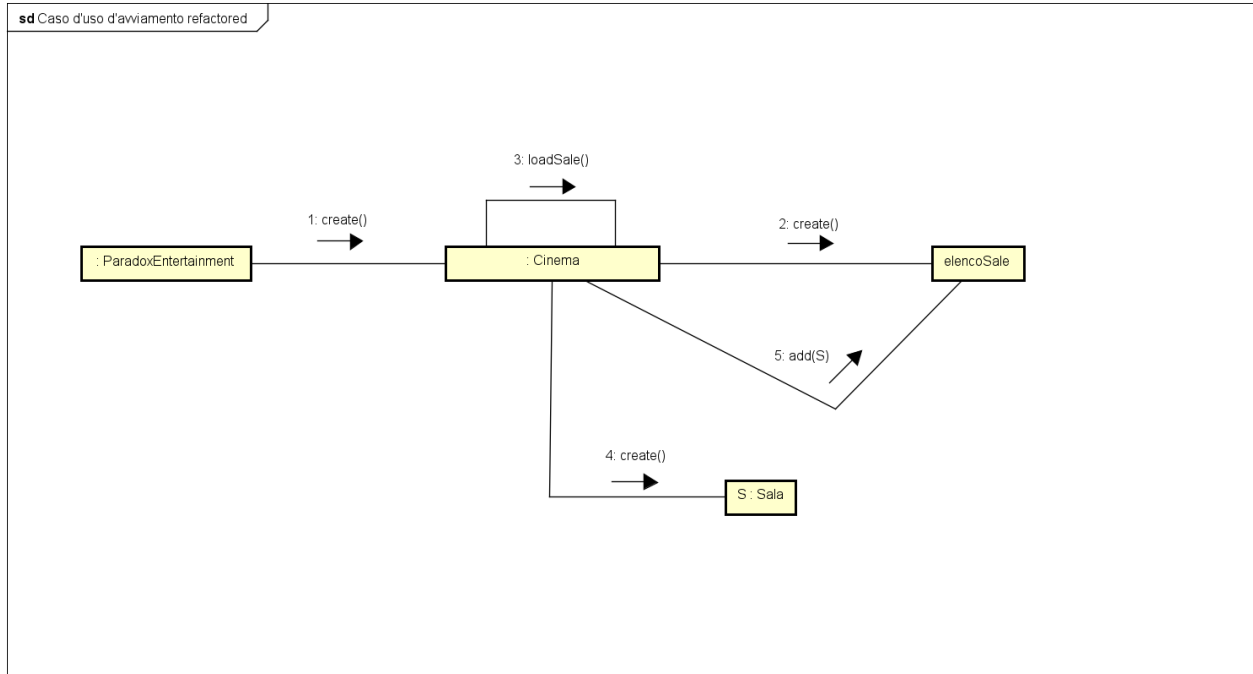
4.2.1. UC5: Vendita Biglietto

Casistica relativa alla vendita del Biglietto di una Proiezione, che viene modificata





4.3. Caso d'uso d'avviamento



4.4. Diagramma delle classi

Anche Cinema viene istanziato come **Singleton**, mentre per la creazione del biglietto si è scelto di implementare il *pattern creazionale Factory Method*. Viene chiamata dalla classe Proiezione il metodo *creaBiglietto* presente su Ticket Factory. Successivamente, in base al giorno della settimana, per rispettare la regola di dominio R6, viene istanziato un Biglietto Feriale o uno Weekend, entrambi con prezzo di base differente. Precedentemente, si era scelto semplicemente di implementare un *if*, verificare il giorno e calcolare direttamente il prezzo totale.

