

Documento di progetto - TifoTicket

Sommario

Requisiti	4
1.1 Introduzione	4
1.2 Casi d'uso	4
Caso d'uso UC1: Inserimento nuova partita.....	4
Caso d'uso UC2: Gestisci acquisto biglietto	5
Caso d'uso UC3: Gestisci sottoscrizione abbonamento	6
Caso d'uso UC4: Gestisci iscrizione fanclub.....	7
Caso d'uso UC5: Gestisci sostituzione nominativo	7
Caso d'uso UC6: Visualizza informazioni vendite partita	7
Caso d'uso UC7: Visualizza informazioni vendite complessive	7
1.3 Regole di dominio	7
Iterazione 1: Analisi	8
2.1 Introduzione	8
2.2 Modello di dominio	9
2.3 Caso d'uso UC1: Diagramma di Sequenza di Sistema	11
2.4 Caso d'uso UC1: Contratti delle operazioni	12
2.4.1: inserimentoNuovaPartita	12
2.4.2: inserimentoPrezzoBiglietti	12
2.4.3: confermaInserimento	12
Iterazione 1: Progettazione	12
3.1 Introduzione	12
3.2 Diagrammi di interazione	13
3.3 Creazione classi di progetto	14
3.4 Diagramma delle classi	15
Iterazione 2: Analisi	15
4.1 Introduzione	15
4.2 Modello di dominio	15
4.3 Diagramma di Sequenza di Sistema	17
3.4 Caso d'uso UC2: Contratti delle operazioni	17
4.4.1 cercaPartita	17

4.4.2 sceltaSettore	18
4.4.3 sceltaPosto	18
4.4.4 datiCliente	18
4.4.5 confermaAcquisto	18
Iterazione 2: Progettazione	19
5.1 Caso d'uso UC2: Diagrammi di interazione	19
5.2 Creazione classi di progetto	23
5.3 Iterazione 2: Diagramma delle Classi.....	23
Iterazione 3: Analisi	24
6.1 Introduzione	24
6.2 Modello di dominio	24
6.3 Diagramma di Sequenza di Sistema	26
6.4.1 settoreAbbonamento	26
6.4.2 datiClienteAbbonamento	27
6.4.3 postoAbbonamento.....	27
6.4.3 confermaAbbonamento	27
Iterazione 3: Progettazione	27
7.1 Diagrammi di interazione	27
7.2 Diagramma delle classi di Progetto	31
7.3 Re-factoring del codice	32
Iterazione 4: Analisi	33
8.1 Introduzione	33
8.2 Modello di dominio	34
8.3 Diagrammi di Sequenza di Sistema	35
8.3.1 SSD del caso d'uso UC6	35
8.3.1 SSD del caso d'uso UC7	36
Iterazione 4: Progettazione	37
9.1 diagrammi di interazione	37
9.2 Diagramma delle classi di Progetto	38
Iterazione 5: Analisi	38
10.1 Introduzione	38
10.2 Modello di dominio	38
10.3 Diagramma di sequenza di sistema.....	39
Iterazione 5: Progettazione	40

11.1 Diagrammi di Interazione.....	40
11.2 Diagramma delle Classi di Progetto	41
Testing.....	42
12.1 Introduzione	42
12.2 Le classi di Test.....	42
Indice delle Figure.....	43

Requisiti

1.1 Introduzione

TifoTicket è un software per la vendita di biglietti ed abbonamenti ai tifosi di una squadra di calcio. È sviluppato per essere usato nei botteghini degli stadi e dai rivenditori autorizzati delle varie squadre, ad esempio nei negozi ufficiali.

Il software permette l'inserimento in memoria delle partite giocate in casa da una determinata squadra, in più consente di registrare vendite ai tifosi quali: la vendita di un biglietto nominale per una partita o la sottoscrizione di un abbonamento che permette al possessore di vedere tutte le partite di Campionato giocate in casa, oltre ad altri servizi come: la sostituzione del nominativo presente sul biglietto e l'iscrizione ad un 'club del tifoso' che porta benefit presso gli sponsor della squadra.

Ogni settore dello stadio avrà una sua capienza, quindi raggiunto il valore limite non sarà possibile acquistare altri biglietti, in questa capienza effettiva vanno considerati anche gli abbonamenti ai diversi settori per le partite di Campionato, mentre per le partite disputate in coppa gli abbonamenti non sono validi e quindi se un abbonato ad un settore volesse vedere la partita dovrà comunque comprare un biglietto.

Per quanto riguarda i prezzi dei biglietti sarà l'amministratore del sistema ad inserirli, settore per settore, per ogni partita, in base ai prezzi stabiliti dalla società. Ovviamente la società punterà a massimizzare i profitti quando si giocherà contro squadre più blasonate, mentre quando si giocherà con squadre che hanno meno appeal sui tifosi si punterà ad avere un numero maggiore di essi allo stadio piuttosto che sul guadagno.

1.2 Casi d'uso

Si considerino i seguenti casi d'uso, dei quali solo 3 sono i casi presi in esame e di cui verranno riportati lo scenario Principale di successo e le estensioni. I restanti casi d'uso saranno espressi in forma breve.

Caso d'uso UC1: Inserimento nuova partita

Attore primario: un Amministratore del sistema.

Scenario principale di successo:

1. L'amministratore vuole inserire una nuova partita con i relativi prezzi dei biglietti.
2. L'amministratore sceglie l'attività "Inserimento nuova partita".
3. L'amministratore inserisce un codice univoco, data ed ora della partita, squadra avversaria e tipologia di partita. Il sistema registra le informazioni.

Il passo 4 sarà ripetuto finché serve.

4. L'amministratore inserisce il prezzo base dei biglietti per la determinata partita. Il sistema registra le informazioni.
5. L'amministratore indica di aver finito.

Estensioni:

***a.** In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente:

1. L'Amministratore riavvia il software e ripristina lo stato precedente del Sistema.
2. Il Sistema ripristina lo stato.

3a. L'Amministratore inserisce un codice univoco non valido perché già presente nel sistema:

1. Il Sistema genera un messaggio di errore.
2. L'Amministratore ripete l'inserimento digitando un nuovo codice.

3b. L'Amministratore inserisce una partita in una data e un'ora in cui è già prevista un'altra partita:

1. Il Sistema genera un messaggio di errore.
2. L'Amministratore ripete l'inserimento cambiando data e ora per la partita.

Caso d'uso UC2: Gestisci acquisto biglietto

Attore primario: un Cassiere.

Scenario principale di successo:

1. Il cliente, munito di documento di identità, si reca alla biglietteria per l'acquisto di un biglietto.
2. Il cliente indica al cassiere la partita per cui vuole acquistare il biglietto e porge il documento di identità.
3. Il cassiere sceglie l'attività "Acquisto biglietto" e inserisce il nome della squadra in trasferta.
4. Il Sistema mostra le partite disponibili contro la squadra selezionata.
5. Il cassiere sceglie la partita opportuna inserendone il codice identificativo.
6. Il cliente comunica al cassiere il settore da cui desidera vedere la partita.
7. Il cassiere inserisce nel sistema il settore scelto dal cliente.
8. Il cassiere inserisce i dati anagrafici del Cliente.
9. Il sistema calcola il prezzo.
10. Il sistema genera un codice univoco per il biglietto e lo stampa.
11. Il cliente paga e va via.

Estensioni:

***a.** In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.

1. L'Amministratore riavvia il software e ripristina lo stato precedente del Sistema.
2. Il Sistema ripristina lo stato.

- 2a.** Il cliente indica al cassiere una partita non presente nel Sistema.
 - 1. Il cassiere comunica al cliente che la partita richiesta non è presente nel sistema.
 - 1.a. Il cliente sceglie un'altra partita.
 - 1.b Il cliente va via.
- 3a.** Il cassiere inserisce nel sistema il nome di una squadra in modo errato
 - 1. Il sistema comunica che non esistono partite in programma con quella squadra.
 - 2. Il cassiere inserisce di nuovo il nome della squadra ospite ripetendo il passo 3.
- 5a.** L'Amministratore inserisce un codice errato:
 - 1. Il Sistema genera un messaggio di errore.
 - 2. Il cassiere ripete la ricerca inserendo un nuovo codice
- 7a.** Il settore scelto dal cliente ha i posti assegnati.
 - 1. Il cliente comunica il posto da cui vuole vedere la partita.
 - 2. Il cassiere inserisce il posto richiesto.
 - a. Il posto è occupato.
 - a. 1. Il cliente comunica un nuovo posto.
- 7b.** Il cassiere inserisce il nome del settore in modo errato.
 - 1. Il sistema comunica che il settore scelto non esiste.
 - 2. Il cassiere effettua nuovamente la ricerca inserendo il nome del settore corretto.
- 7c.** Il sistema mostra che non ci sono più posti disponibili in quel settore.
 - 1. Il cliente sceglie un nuovo settore.
 - 2. Il cassiere inserisce il nuovo settore scelto ripetendo la ricerca.

Caso d'uso UC3: Gestisci sottoscrizione abbonamento

Attore primario: un Cassiere.

Scenario principale di successo:

- 1. Il cliente, munito di documento di identità, si reca alla biglietteria per la sottoscrizione di un abbonamento.
- 2. Il cliente indica al cassiere di voler acquistare un abbonamento e porge il documento di identità.
- 3. Il cassiere sceglie l'attività "Acquisto abbonamento".
- 4. Il cassiere inserisce le generalità del Cliente.
- 5. Il cliente comunica il settore da cui desidera vedere le partite.
- 6. Il cassiere inserisce il nome del settore ed ottiene i posti disponibili.
- 7. Il sistema calcola il prezzo.
- 8. Il sistema genera un codice univoco e stampa la tessera di abbonamento.
- 9. Il cliente paga e va via.

Estensioni:

- *a.** In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.
 - 1. L'Amministratore riavvia il software e ripristina lo stato precedente del Sistema

- 2. Il Sistema ripristina lo stato
- 4a.** Il codice fiscale del cliente risulta già assegnato ad un abbonamento:
 - 1. Il sistema mostra un messaggio di errore.
- 6a.** Il cassiere inserisce il nome del settore in modo errato:
 - 1. Il sistema comunica che il settore scelto non esiste.
 - 2. Il cassiere inserisce nuovamente il nome del settore ripetendo l'operazione.
- 6b.** Il settore scelto dal cliente ha i posti assegnati.
 - 1. Il cliente comunica il posto desiderato.
 - 2. Il cassiere inserisce il posto richiesto.
- 2a.** Il posto è occupato.
 - 2a. 1.** Il cliente torna al passo 4 e comunica un nuovo posto.

Caso d'uso UC4: Gestisci iscrizione fanclub

Un cliente desidera iscriversi al fan club ufficiale della squadra. Il cassiere prende le generalità del cliente e, se il cliente non risulta già iscritto, il sistema genera un codice univoco e stampa la tessera. Il cliente paga, prende la tessera e va via.

Caso d'uso UC5: Gestisci sostituzione nominativo

Il cliente desidera sostituire il nominativo presente sul biglietto e si reca al botteghino. Il cassiere inserisce nel sistema le generalità della persona a cui verrà intestato il nuovo biglietto. Il cliente paga la sostituzione, prende il nuovo biglietto e va via.

Caso d'uso UC6: Visualizza informazioni vendite partita

Un amministratore vuole visualizzare le informazioni sui biglietti venduti per ogni settore in una determinata partita. L'amministratore inserisce la partita scelta e il sistema mostra per ogni settore il numero di biglietti venduti.

Caso d'uso UC7: Visualizza informazioni vendite complessive

Un amministratore desidera visualizzare le informazioni sui biglietti complessivi venduti per la stagione in corso per motivi statistici. Seleziona la voce opportuna dal menu ed il sistema mostra le informazioni richieste.

1.3 Regole di dominio

Di seguito sono illustrate le regole di dominio stabilite dalla Società per la stagione attuale. Esse riguardano i prezzi dei biglietti e degli abbonamenti per le diverse tipologie di partite e categorie di età.

- A. I prezzi base dei biglietti per ciascun settore differiscono di partita in partita, in base all'importanza di essa e della squadra avversaria.

- B. è stato stabilito dalla Società che i prezzi dei biglietti per qualsiasi settore siano scontati del 20% per i tifosi con età inferiore ai 14 anni per le partite di Campionato. Ciò con l'obiettivo di incentivare le famiglie a venire allo stadio e coinvolgere anche i più piccoli.
- C. è stato stabilito dalla Società che i prezzi dei biglietti per qualsiasi settore siano scontati del 30% per i tifosi con età superiore ai 65 anni per le partite di Campionato. Ciò con l'obiettivo di favorire la presenza di quei tifosi che ci hanno supportato per tanto tempo.
- D. È stato stabilito dalla Società che i prezzi dei biglietti per qualsiasi settore siano scontati del 10% per tutte le partite di Coppa per i tifosi Under 14, mentre per i tifosi Over 65 è stato stabilito uno sconto del 15%. Ciò con l'obiettivo di aumentare il numero di tifosi presenti, dato che dalle statistiche è risultato che queste partite siano quelle con la minor affluenza dell'anno.
- E. È stato stabilito dalla Società che i prezzi dei biglietti per qualsiasi settore siano scontati del 10% per tutte le partite di Coppa Europea per i tifosi Under 14, mentre per i tifosi Over 65 è stato stabilito uno sconto del 15% Ciò per soddisfare gli stessi obiettivi di cui sopra.
- F. È stato stabilito dalla società che il sovrapprezzo da pagare per la sostituzione del nominativo presente su un biglietto è di 5 euro. Il cambio di nominativo è consentito solo tra tifosi che rientrano nella stessa fascia di età. (Es. il nominativo su un biglietto con tariffa Under 14 può essere sostituito solo con un altro nominativo di un Tifoso Under 14).
- G. Dalle discussioni interne alla Società è emerso che gli abbonamenti per questa stagione avranno prezzi fissi in base alle fasce d'età.
- H. Prezzo Abbonamento per Under 14:
 - **Tribuna Est:** 200 euro, **Tribuna Ovest:** 220 euro, **Curva Sud:** 100 euro, **Curva Nord:** 85 euro.
- I. Prezzo Abbonamento per Over 65:
 - **Tribuna Est:** 230 euro, **Tribuna Ovest:** 270 euro, **Curva Sud:** 120 euro, **Curva Nord:** 100 euro.
- J. Prezzo Abbonamento per coloro che non rientrano nelle fasce descritte:
 - **Tribuna Est:** 270 euro, **Tribuna Ovest:** 320 euro, **Curva Sud:** 170 euro, **Curva Nord:** 150 euro.

Iterazione 1: Analisi

2.1 Introduzione

Per l'iterazione 1 è stato scelto di utilizzare i seguenti requisiti:

- Scenario principale di successo del caso d'uso UC1: Inserimento partita.
- caso d'uso di avviamento
- dati solo in memoria principale

Questo capitolo descrive l'analisi svolta nell'iterazione 1, mentre il capitolo successivo descrive l'attività di progettazione.

2.2 Modello di dominio

In questa iterazione, del caso d'uso UC1 è di interesse lo scenario principale di successo, nella sua interezza, riportato nel Paragrafo 1.2.1. Da esso è possibile identificare le seguenti classi concettuali:

- *Amministratore*: Attore primario, colui che interagisce con il software.
- *TifoTicket*: rappresenta il sistema TifoTicket in sé.
- *Partita*: Una partita da inserire nel sistema.
- *Stadio*: Lo stadio in cui verrà giocata la partita.
- *Settore*: I diversi settori da cui è composto lo Stadio.

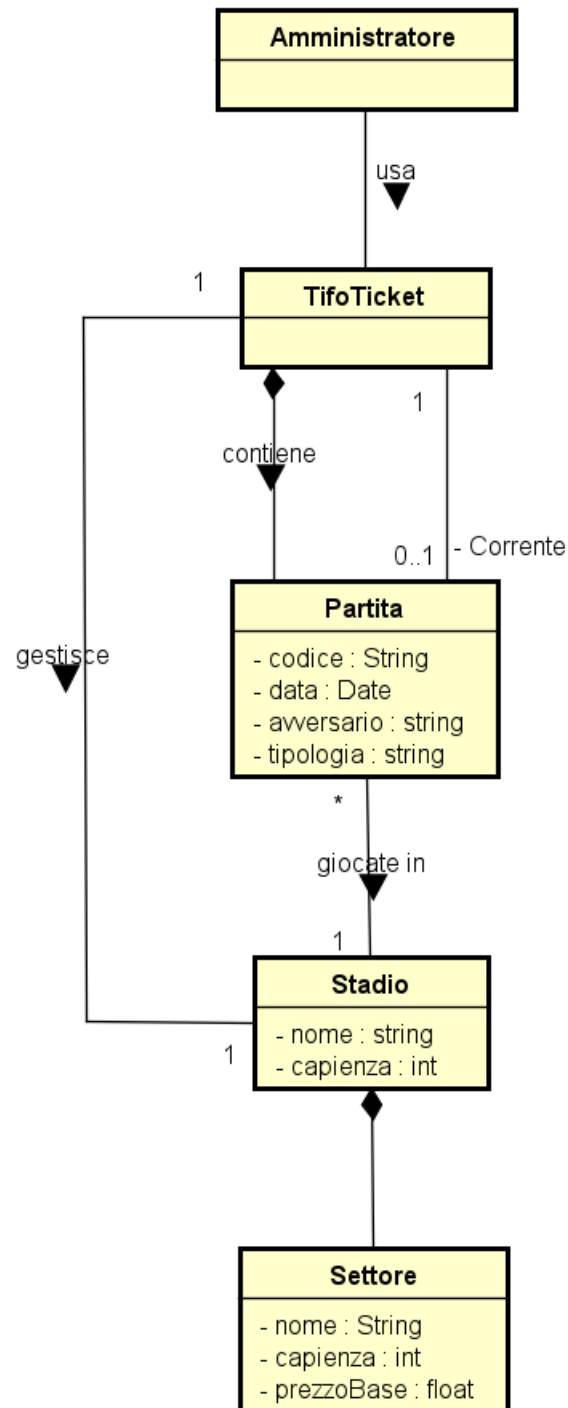


Figure 1: modello di dominio Iterazione 1

2.3 Caso d'uso UC1: Diagramma di Sequenza di Sistema

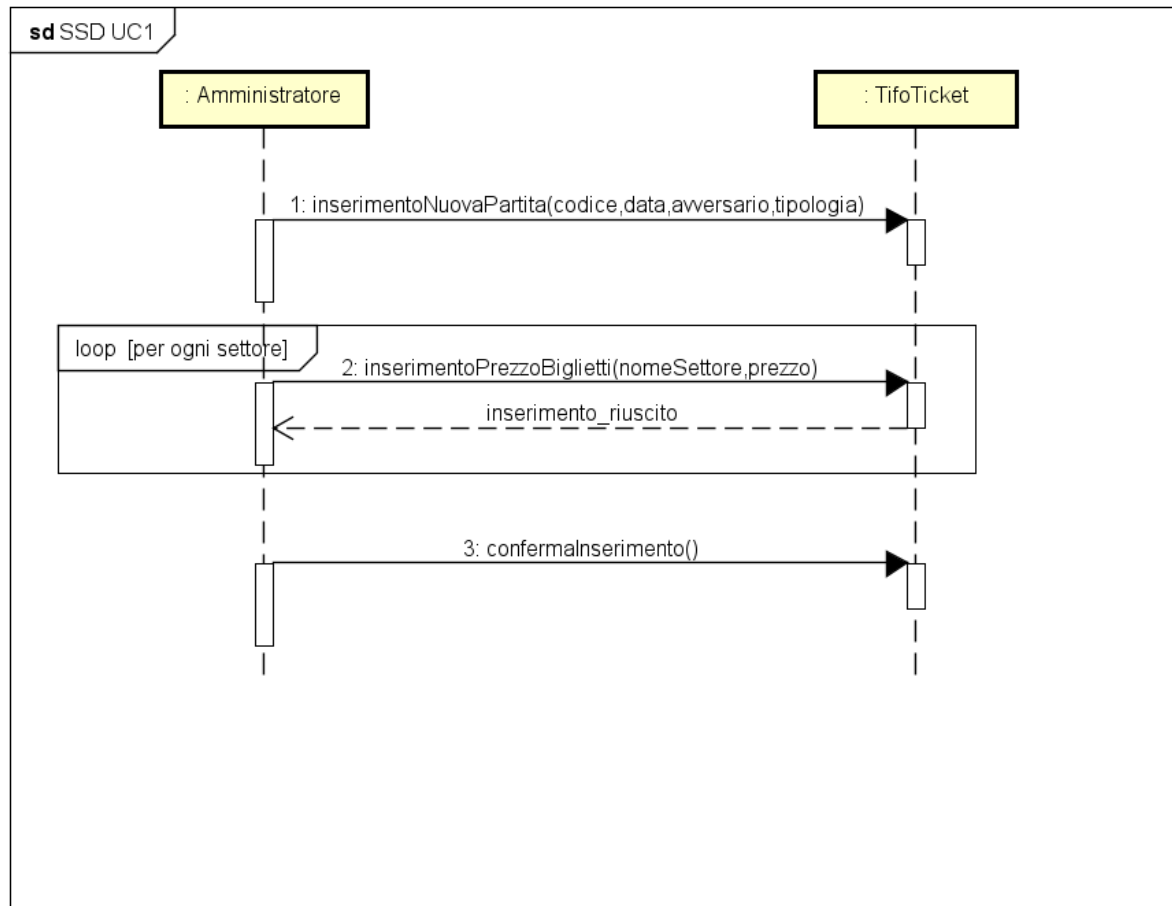


Figure 2: SSD UC1

Il diagramma di sequenza di sistema per il caso d'uso riportato sopra individua 3 operazioni:

1. *InserimentoNuovaPartita*: con cui viene iniziato l'inserimento dei dati di una partita nel Sistema.
2. *InserimentoPrezzoBiglietti*: con cui per ogni settore dello stadio viene decretato il prezzo base di un biglietto.
3. *ConfermaInserimento*: la quale si verifica quando l'inserimento di una nuova partita è terminato e questa va confermata e registrata.

2.4 Caso d'uso UC1: Contratti delle operazioni

2.4.1: inserimentoNuovaPartita

Operazione: *inserimentoNuovaPartita (String codice, LocalDateTime data, String avversario, String tipologia)*

Riferimenti: Caso d'uso Gestisci inserimento partita

Pre-condizioni: - è stata creata un'istanza st di Stadio

Post-condizioni: - è stata creata un'istanza pa di Partita

- i campi codice, data, avversario e tipologia di pa sono stati riempiti con i valori passati in input.

- è stata formata l'associazione partitaCorrente con TifoTicket.

- è stata formata l'associazione con lo Stadio st.

2.4.2: inserimentoPrezzoBiglietti

Operazione: *inserimentoPrezzoBiglietti (String nomeSettore, float prezzo)*

Riferimenti: Caso d'uso Gestisci inserimento partita

Pre-condizioni: - è in corso l'inserimento di una Partita pa

Post-condizioni: - sono stati riempiti i campi prezzoBase dei Settori dello Stadio st.

2.4.3: confermaInserimento

Operazione: *confermaInserimento ()*

Riferimenti: Caso d'uso Gestisci inserimento partita

Pre-condizioni: - è in corso l'inserimento di una Partita pa

Post-condizioni: -

Iterazione 1: Progettazione

3.1 Introduzione

La prima decisione da prendere è scegliere quali tra le classi software debba essere la prima a ricevere e coordinare una operazione di sistema. Applicando il pattern **Controller** (GRASP), La scelta ricade sull'utilizzo della classe TifoTicket come **Facade Controller**, la quale permetterà, attraverso un'interfaccia più semplice, l'accesso a sottoclassi che espongono interfacce complesse e molto diverse tra loro. Sarà quindi creata una classe TifoTicket che farà da Controller per i casi d'uso e quest'ultima sarà implementata come classe **Singleton**, in modo tale da garantire che venga creata *una e una sola* istanza della classe, e di fornire un punto di accesso globale a tale istanza.

3.2 Diagrammi di interazione

Diagramma di sequenza dell'operazione inserimentoNuovaPartita:

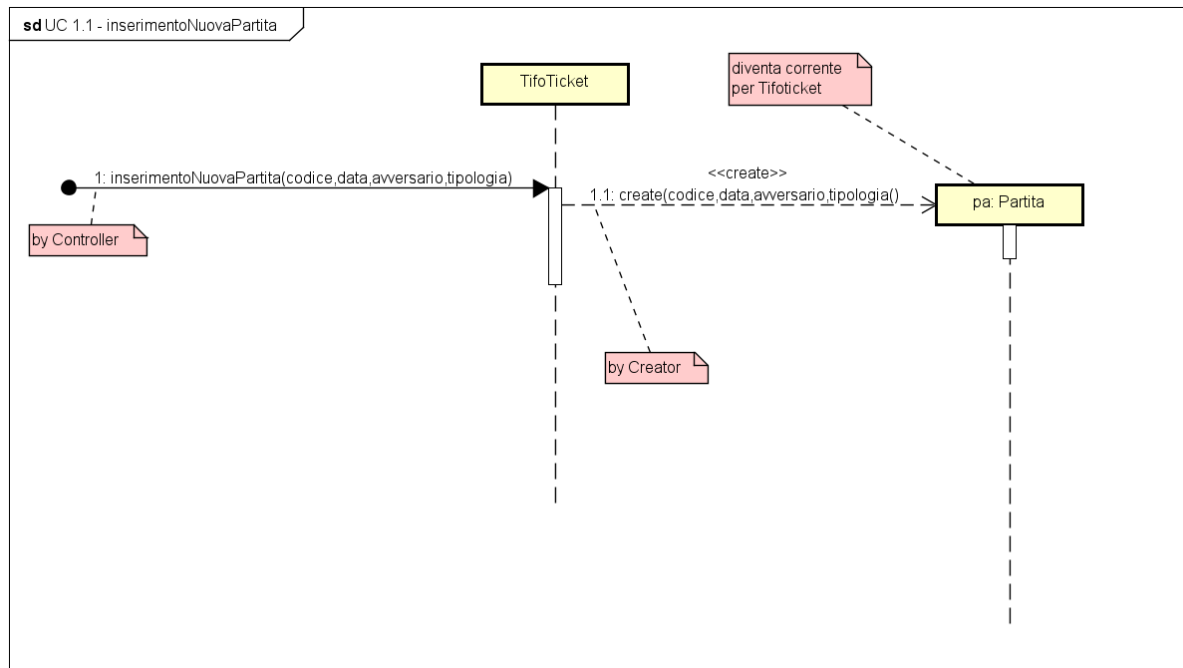


Figure 3: SD inserimentoNuovaPartita

Diagramma di interazione dell'operazione inserimentoPrezzoBiglietto:

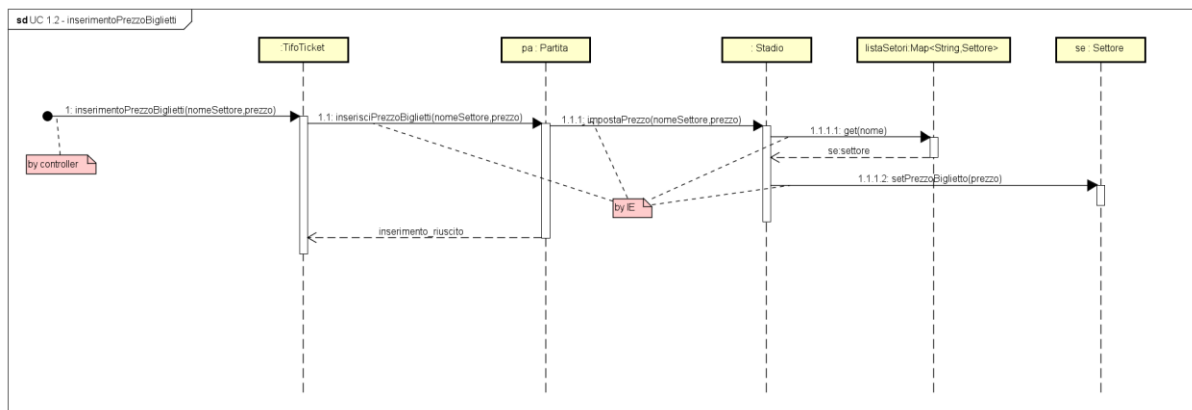


Figure 4: SD inserimentoPrezzoBiglietto

Diagramma di interazione dell'operazione confermaInserimento:

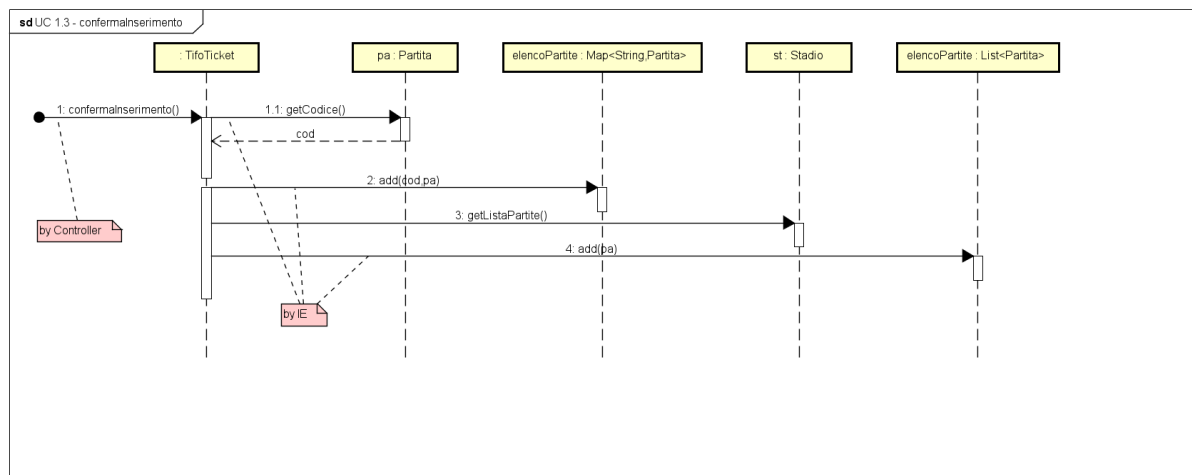


Figure 5: SD confermaInserimento

3.3 Creazione classi di progetto

Come risultato delle operazioni descritte dai diagrammi di sequenza rappresentati sopra sono state create le prime classi software del progetto:

- *TifoTicket*: classe che rappresenta il sistema stesso e funge da Facade Controller.
- *Stadio*: classe che rappresenta lo stadio della società.
- *Settore*: classe che rappresenta i vari settori in cui è diviso lo Stadio
- *Tribuna* e *Curva*: sono due specializzazioni della classe *Settore*, l'unica differenza tra di loro è che Tribuna possiede un elenco di posti, in quanto i posti nelle tribune sono numerati, a differenza delle curve.
- *Posto*: come spiegato sopra, serve per identificare i posti presenti nelle tribune, questa classe possiede come attributi la fila ed il numero del posto.
- *Partita*: classe che rappresenta la Partita che verrà inserita in memoria

3.4 Diagramma delle classi

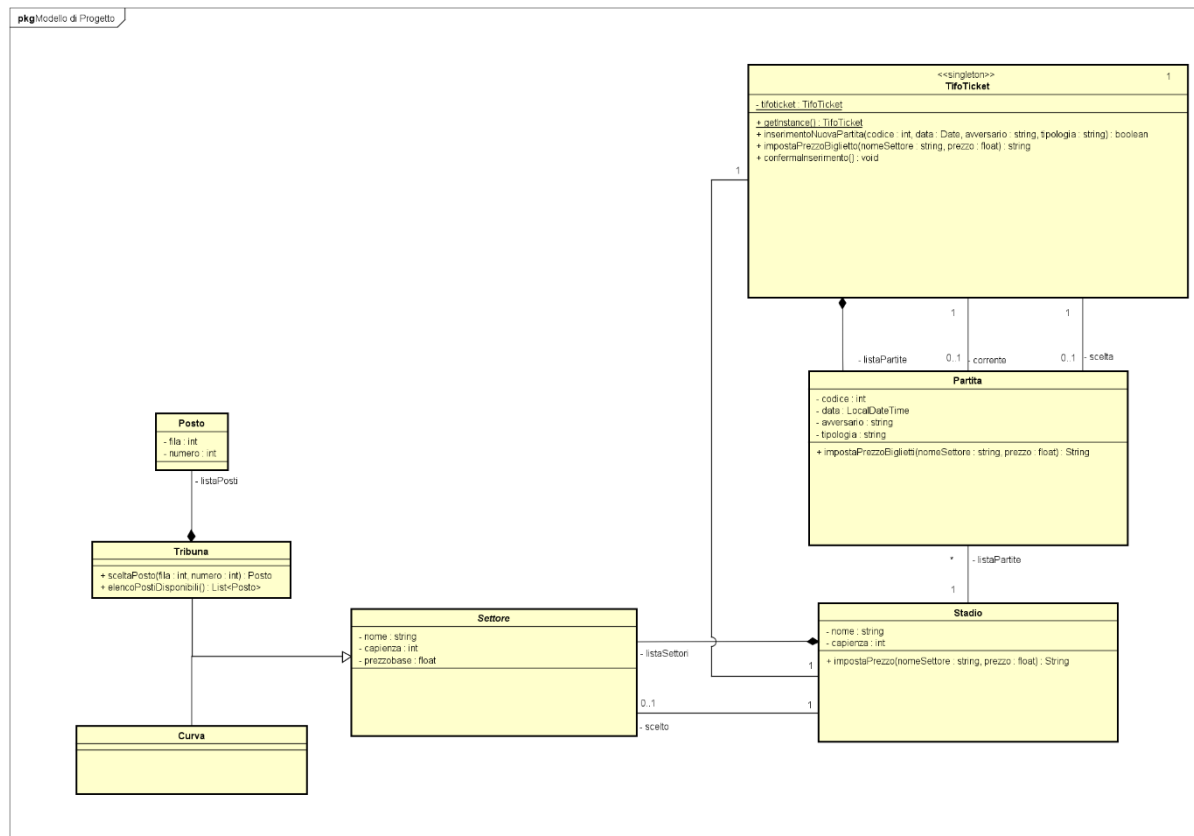


Figure 6: DCD iterazione 1

Iterazione 2: Analisi

4.1 Introduzione

Per l'iterazione 2 è stato scelto di utilizzare i seguenti requisiti:

- Estensioni del caso d'uso UC1: Inserimento partita.
- caso d'uso UC2: Gestisci Acquisto biglietto.
- Dati solo in memoria principale
- Il calcolo del prezzo di un biglietto è stabilito dalle regole di dominio B, C, D ed E.

Questo capitolo descrive l'analisi svolta nell'iterazione 2, mentre il capitolo successivo descrive l'attività di progettazione.

4.2 Modello di dominio

In questa iterazione, viene attenzionato il caso d'uso UC2 nella sua interezza, comprendendo quindi anche le estensioni, come riportato nel Paragrafo 1.2.2. Da esso è possibile identificare le seguenti classi concettuali:

- *Cassiere*: attore principale UC2.
- *Documento identità*: Documento posseduto dal cliente e fornito al cassiere.
- *Cliente*: colui che si reca allo sportello per acquistare un biglietto.
- *Biglietto*: Il biglietto in sé, stampato dal sistema ed acquistato dal cliente.

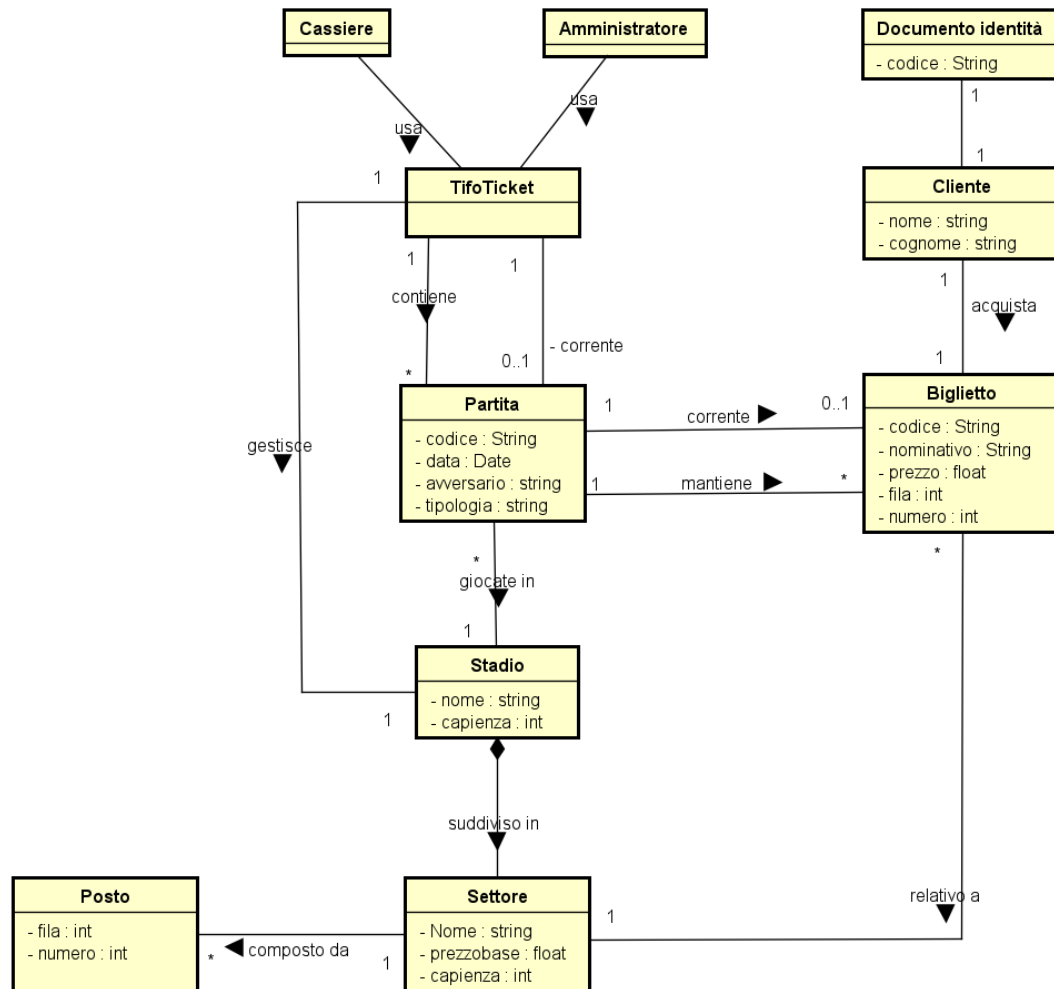


Figure 7: Modello di dominio Iterazione 2

4.3 Diagramma di Sequenza di Sistema

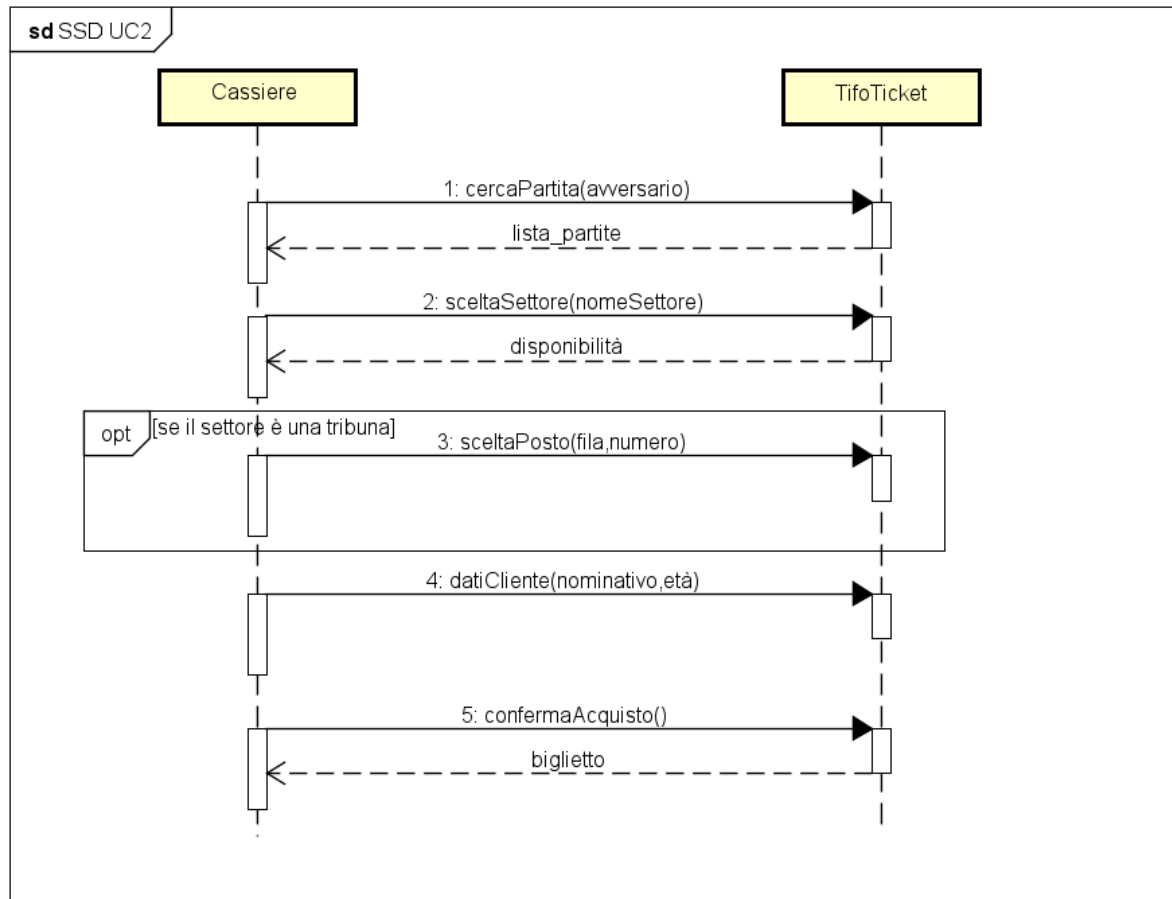


Figure 8: SSD UC2

Il diagramma di Sequenza di sistema riportato sopra individua 5 operazioni:

1. *cercaPartita*: con cui vengono cercate le partite presenti nel sistema contro una determinata squadra avversaria.
2. *sceltaSettore*: con cui viene verificata la disponibilità di posti nel settore richiesto dal cliente.
3. *sceltaPosto*: questa operazione si verifica solo se il settore scelto dal cliente è una tribuna e in quanto tale ha i posti assegnati; quindi, permette di selezionare il posto richiesto dal cliente.
4. *inserimentoDatiCliente*: con cui vengono immessi i dati del cliente nel sistema che saranno utilizzati per calcolare il prezzo del biglietto in base alle regole di dominio e dovranno poi essere stampati sul biglietto.
5. *confermaAcquisto*: si verifica quando la vendita del biglietto è terminata e questa va confermata e registrata.

3.4 Caso d'uso UC2: Contratti delle operazioni

4.4.1 cercaPartita

Operazione: *cercaPartita* (*String avversario*)

Riferimenti: Caso d'uso Gestisci acquisto biglietto.

Pre-condizioni: - è stata inserita almeno una partita.

Post-condizioni: - è stata creata la relazione partitaScelta tra un'istanza di Partita e TifoTicket.

4.4.2 sceltaSettore

Operazione: *sceltaSettore (String nomeSettore)*

Riferimenti: Caso d'uso Gestisci acquisto biglietto.

Pre-condizioni: - è stata inserita almeno una partita.

Post-condizioni: - è stata formata l'associazione partitaScelta tra una partita e TifoTicket

- è stata formata l'associazione settoreScelto tra un'istanza di Settore ed st.
- è stata creata un'istanza bi di Biglietto
- è stata formata l'associazione bigliettoCorrente tra bi e Partita
- sono stati riempiti i campi codice, settore e partita di bi.

4.4.3 sceltaPosto

Operazione: *sceltaPosto (int fila, int numero)*

Riferimenti: Caso d'uso Gestisci acquisto biglietto.

Pre-condizioni: - è stato selezionato un settore di tipo Tribuna.

Post-condizioni: - è stato riempito il campo posto di bi.

4.4.4 datiCliente

Operazione: *datiCliente (String nominativo, int eta)*

Riferimenti: Caso d'uso Gestisci acquisto biglietto.

Pre-condizioni: - è in corso l'inserimento di un Biglietto bi.

Post-condizioni: - sono stati riempiti i campi nominativo ed eta di bi.

4.4.5 confermaAcquisto

Operazione: *confermaAcquisto ()*

Riferimenti: Caso d'uso Gestisci acquisto biglietto.

Pre-condizioni: - è in corso l'inserimento di un Biglietto bi.

Post-condizioni: -

Iterazione 2: Progettazione

5.1 Caso d'uso UC2: Diagrammi di interazione

Diagramma di sequenza dell'operazione cercaPartita:

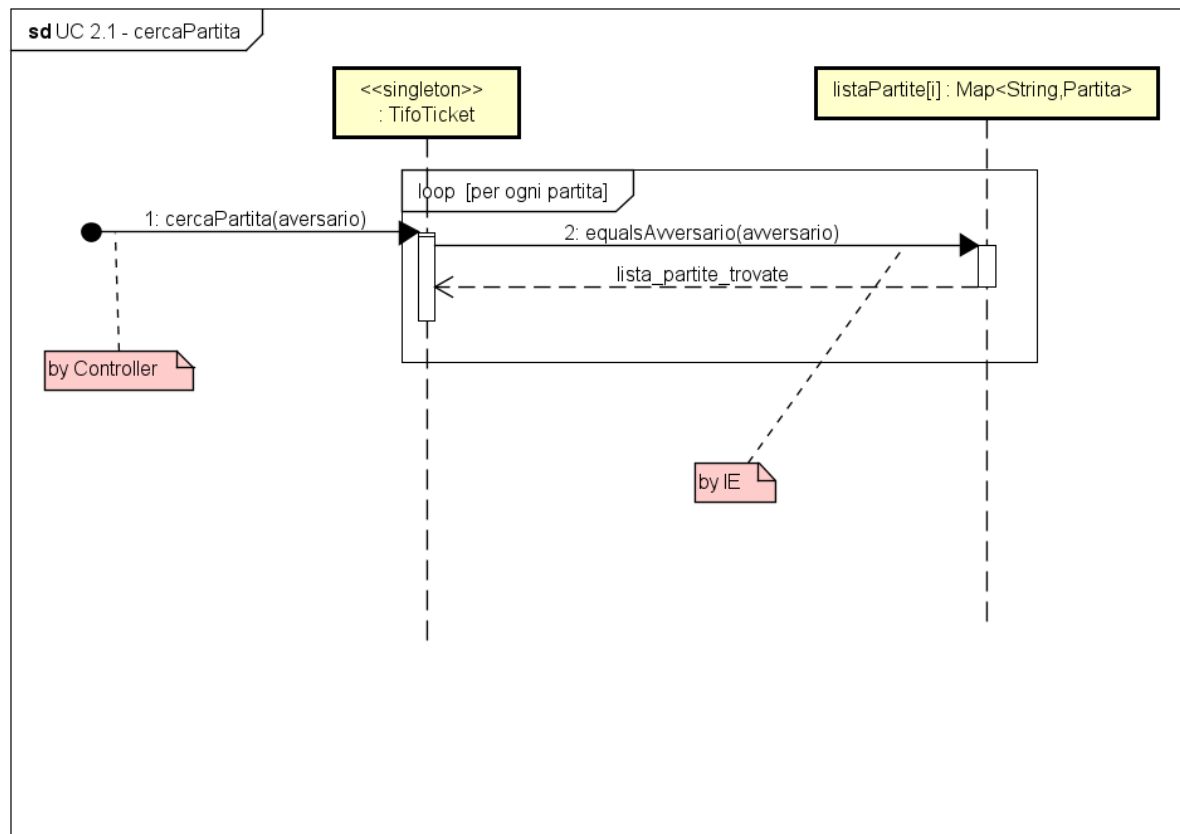


Figure 9: SD *cercaPartita*

Diagramma di sequenza dell'operazione sceltaSettore:

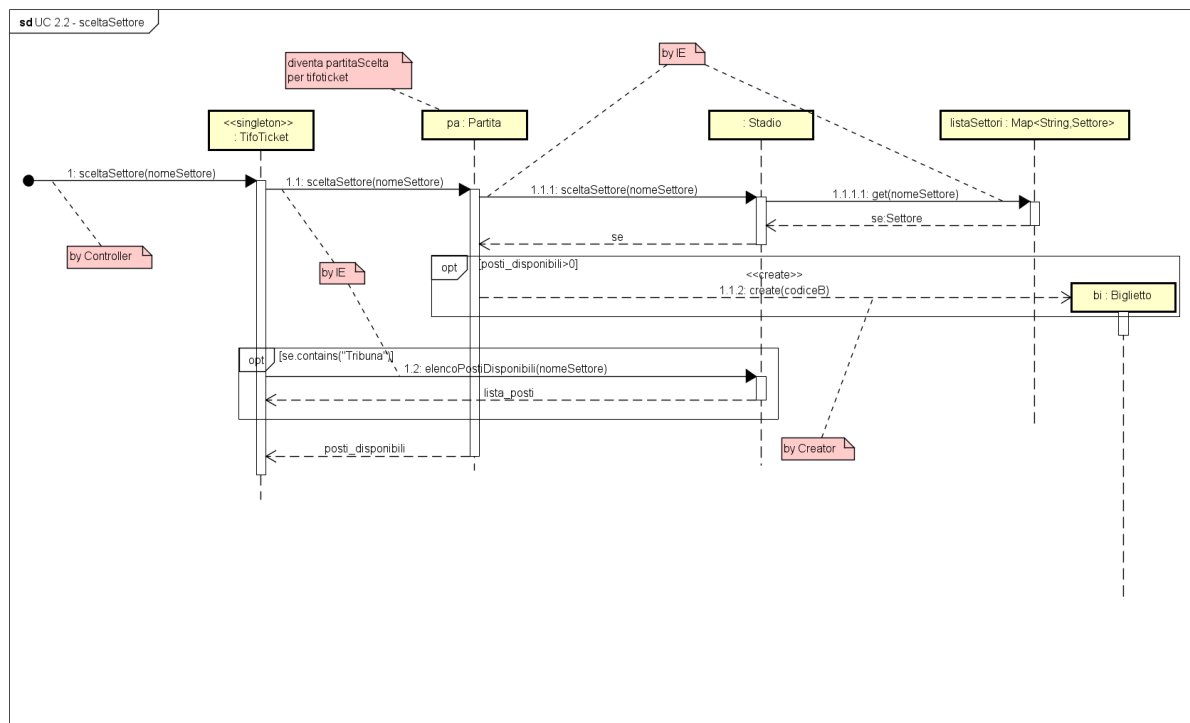


Figure 10: SD sceltaSettore

Diagramma di sequenza dell'operazione sceltaPosto:

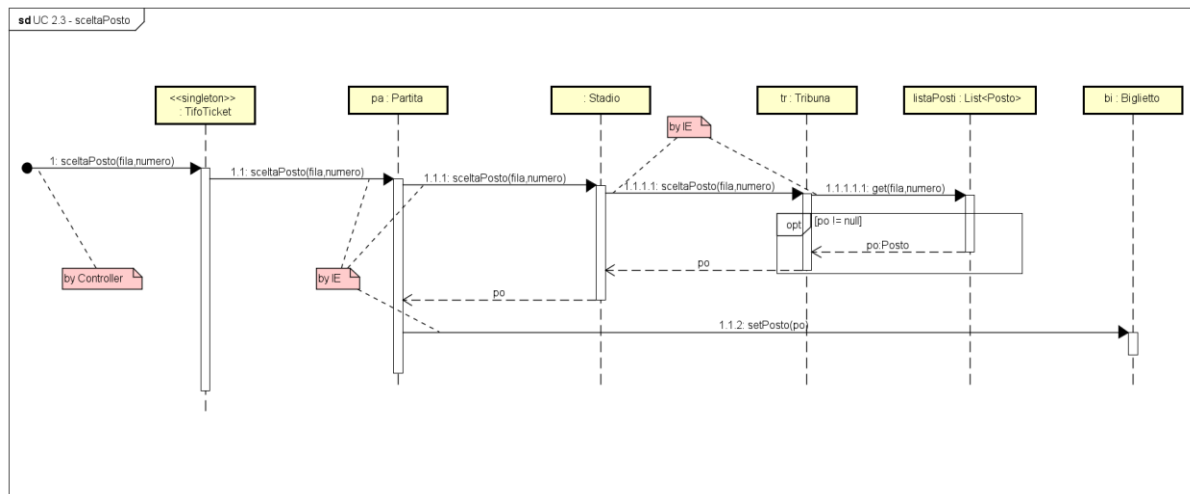


Figure 11: SD sceltaPosto

Diagramma di sequenza dell'operazione datiCliente:

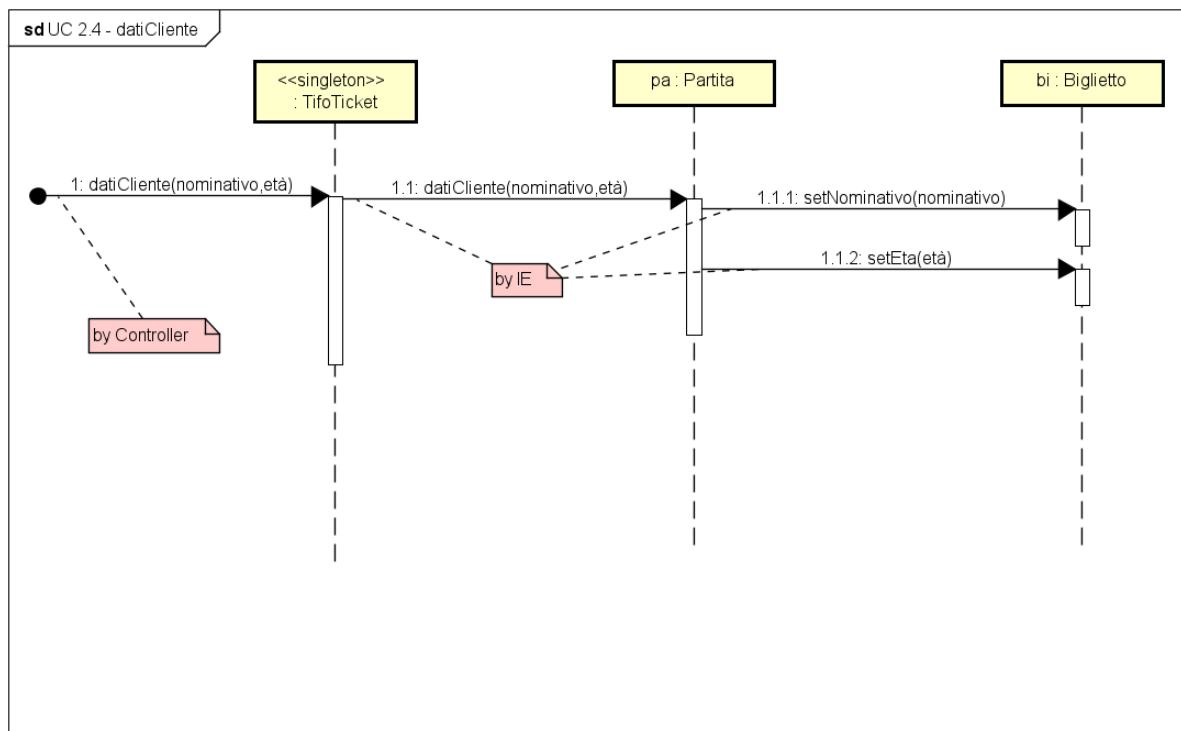


Figure 12: SD datiCliente

Diagramma di sequenza dell'operazione confermaAcquisto:

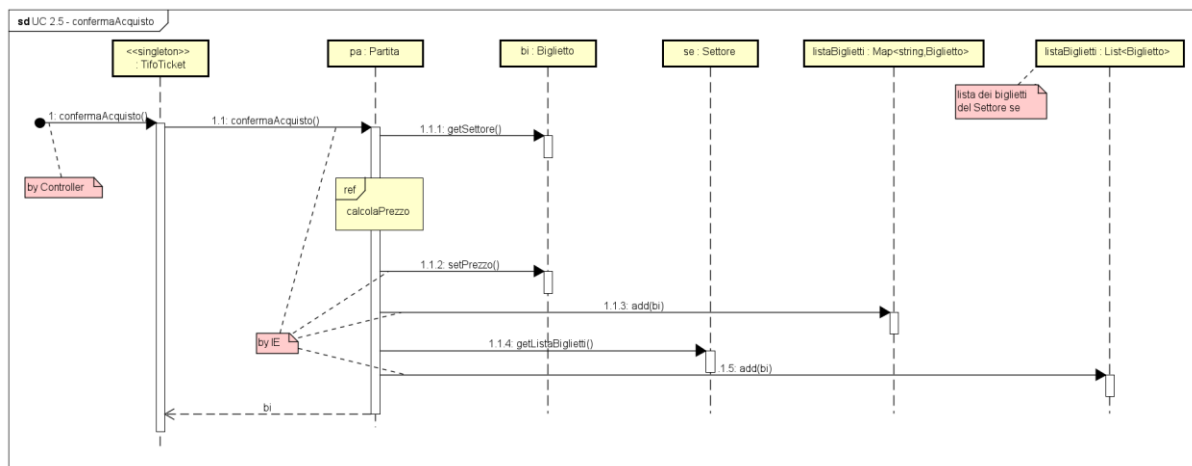


Figure 13: SD confermaAcquisto

Diagramma di sequenza dell'operazione calcolaPrezzo:

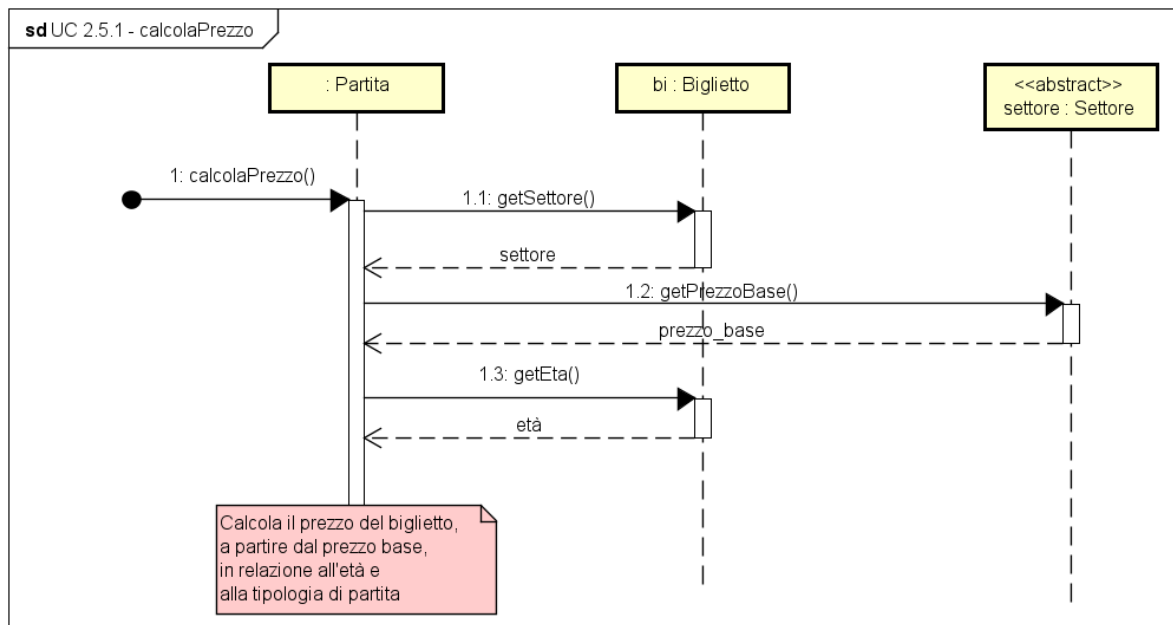


Figure 14: SD calcolaPrezzo

5.2 Creazione classi di progetto

Come risultato delle operazioni descritte dai diagrammi di sequenza rappresentati sopra è stata creata un'unica classe software:

- **Biglietto**: classe che rappresenta i biglietti creati dal sistema e possiede come attributi: un codice univoco, il nominativo del cliente, l'età del cliente, il prezzo del biglietto, il Settore per cui è stato acquistato il biglietto ed un attributo di tipo Posto.

5.3 Iterazione 2: Diagramma delle Classi

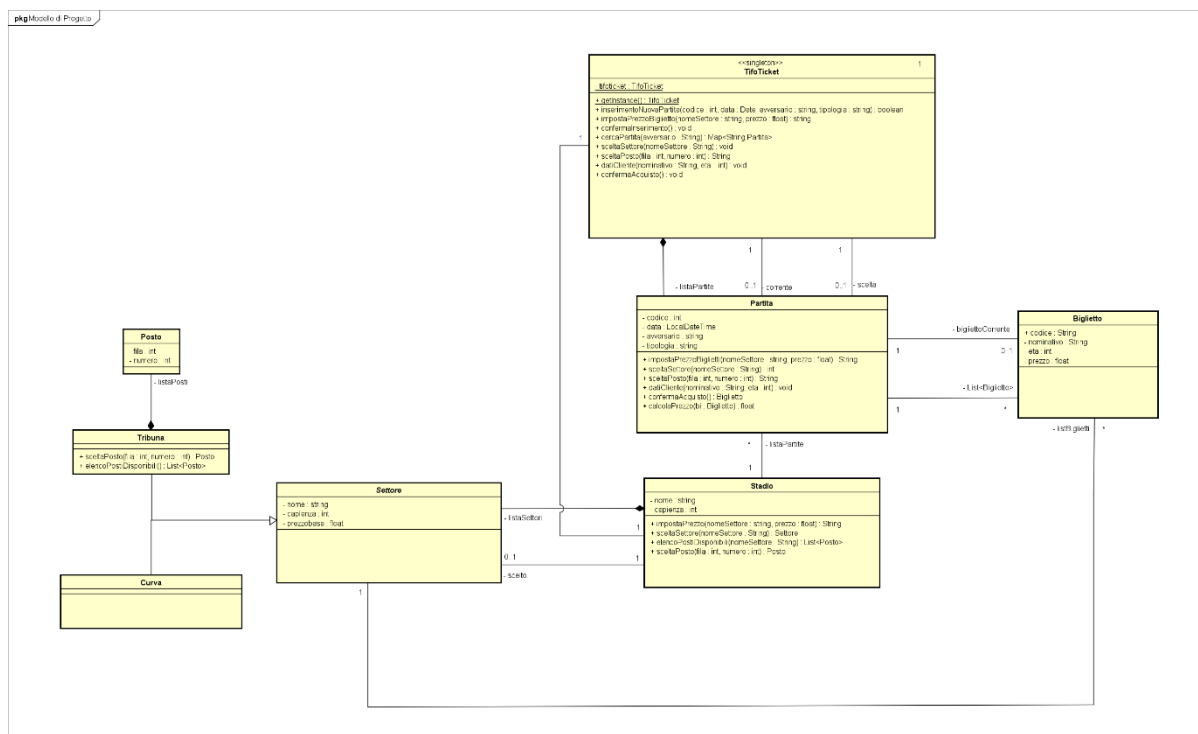


Figure 15: DCD iterazione 2

Iterazione 3: Analisi

6.1 Introduzione

Per l'iterazione 3 è stato scelto di utilizzare i seguenti requisiti:

- Scenario principale di successo del caso d'uso UC3: Gestisci sottoscrizione abbonamento.
- Dati solo in memoria principale
- Il calcolo del prezzo di un abbonamento è stabilito dalle regole di dominio H, I e J.

Questo capitolo descrive l'analisi svolta nell'iterazione 3, mentre il capitolo successivo descrive l'attività di progettazione.

6.2 Modello di dominio

In questa iterazione, del caso d'uso UC3 sono di interesse lo scenario principale di successo, nella sua interezza, e le estensioni, il tutto riportato nel Paragrafo 1.2.3. Da esso è possibile identificare un'unica classe concettuale:

- *Abbonamento*: Il tagliando stampato dal sistema, valido per tutte le partite di Campionato giocate in casa ed acquistato dal Cliente.

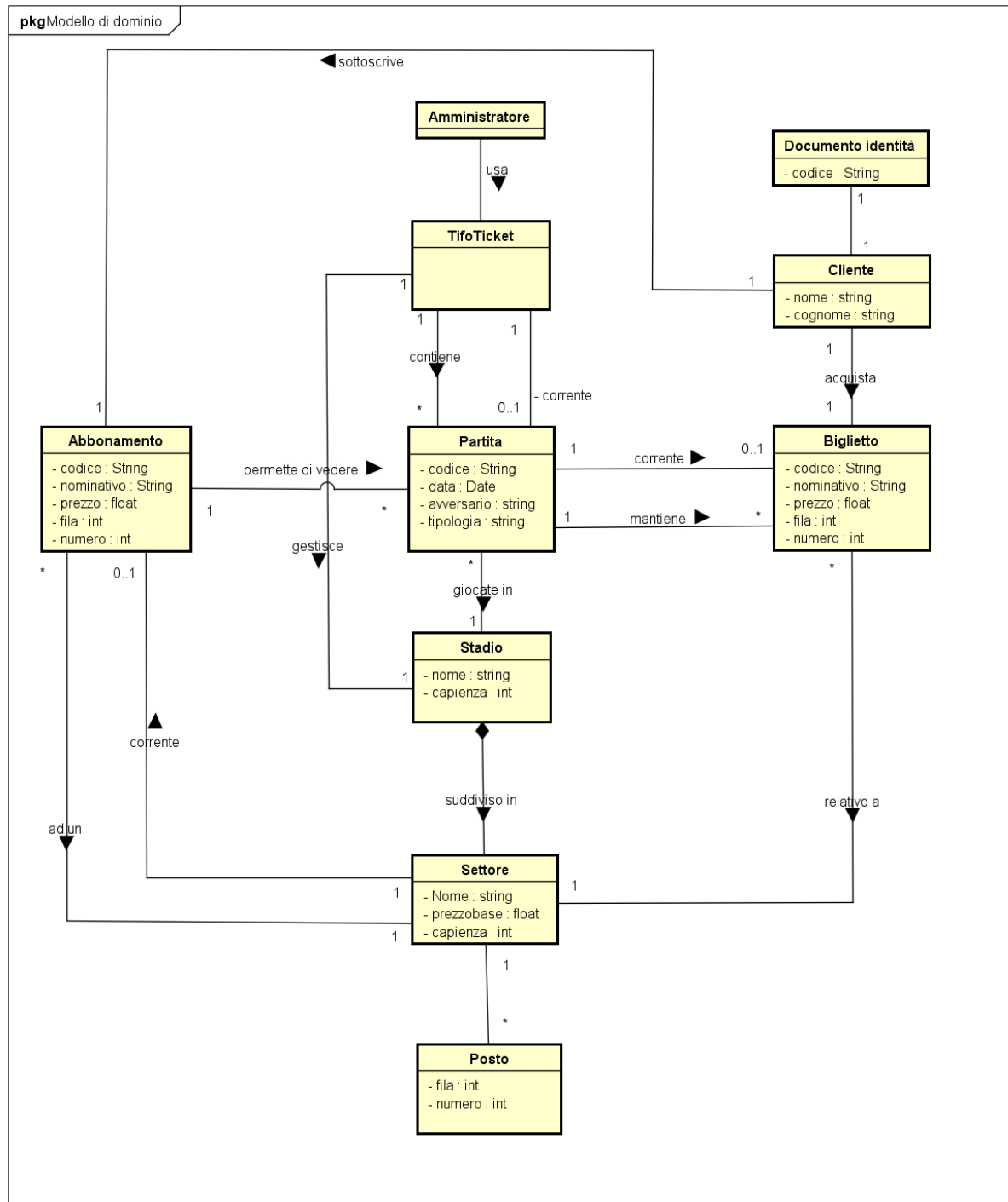


Figure 16: Modello di dominio UC3

6.3 Diagramma di Sequenza di Sistema

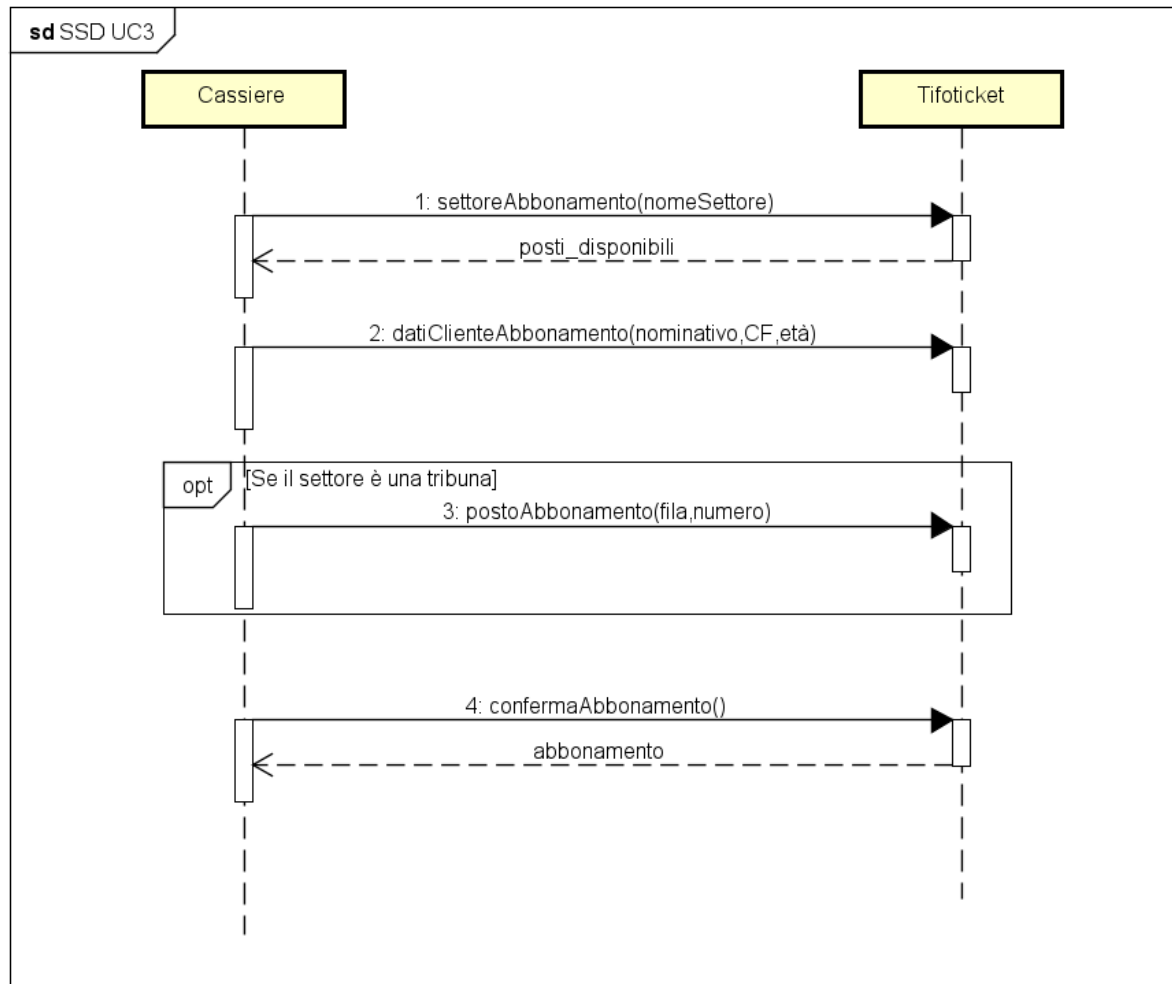


Figure 17: SSD UC3

Il diagramma di Sequenza di sistema riportato sopra individua 4 operazioni:

1. *settoreAbbonamento*: con cui viene verificata la disponibilità di posti nel settore richiesto dal cliente.
2. *datiClienteAbbonamento*: con cui vengono immessi i dati del cliente nel sistema che saranno utilizzati per calcolare il prezzo dell'abbonamento in base alle regole di dominio e dovranno poi essere stampati sull'abbonamento.
3. *postoAbbonamento*: questa operazione si verifica solo se il settore scelto dal cliente è una tribuna e in quanto tale ha i posti assegnati; quindi, permette di selezionare il posto richiesto dal cliente.
4. *confermaAbbonamento*: si verifica quando la vendita dell'abbonamento è terminata e questa va confermata e registrata.

6.4.1 settoreAbbonamento

Operazione: *settoreAbbonamento* (*String nomeSettore*)

Riferimenti: Caso d'uso Gestisci sottoscrizione abbonamento.

Pre-condizioni: - è stata creata un'istanza st di Stadio.

Post-condizioni: - è stata creata l'associazione settoreScelto tra un'istanza di Settore ed st.

6.4.2 datiClienteAbbonamento

Operazione: *datiClienteAbbonamento (String nominativo, String CF, int eta)*

Riferimenti: Caso d'uso Gestisci sottoscrizione abbonamento.

Pre-condizioni: -

Post-condizioni: - viene creata un'istanza abb di Abbonamento.

- i campi nominativo, CF ed eta di abb sono stati riempiti.
- Viene creata l'associazione abbonamentoCorrente tra abb e un'istanza di Settore.

6.4.3 postoAbbonamento

Operazione: *postoAbbonamento (int fila, int numero)*

Riferimenti: Caso d'uso Gestisci sottoscrizione abbonamento.

Pre-condizioni: - è in corso l'inserimento di un Abbonamento abb

Post-condizioni: - il campo posto di abb è stato riempito.

6.4.3 confermaAbbonamento

Operazione: *confermaAbbonamento ()*

Riferimenti: Caso d'uso Gestisci sottoscrizione abbonamento.

Pre-condizioni: - è in corso l'inserimento di un Abbonamento abb

Post-condizioni: -

Iterazione 3: Progettazione

7.1 Diagrammi di interazione

Diagramma di sequenza dell'operazione settoreAbbonamento:

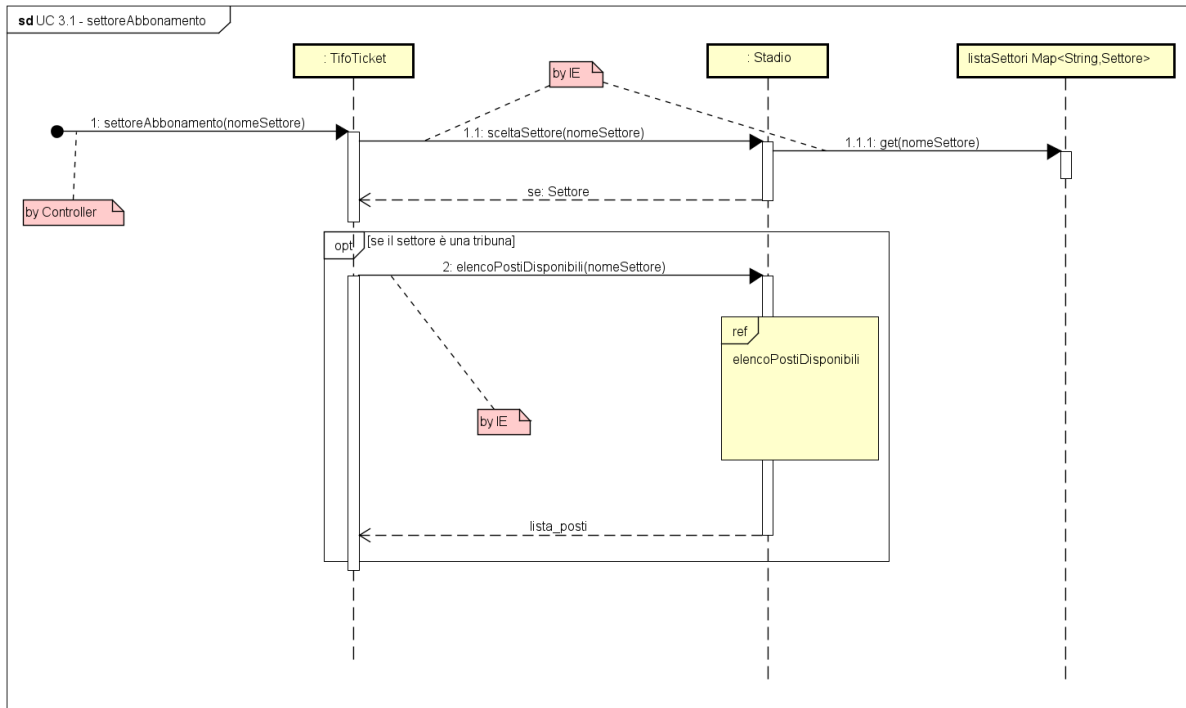


Figure 18: SD settoreAbbonamento

Diagramma di sequenza dell'operazione elencoPostiDisponibili:

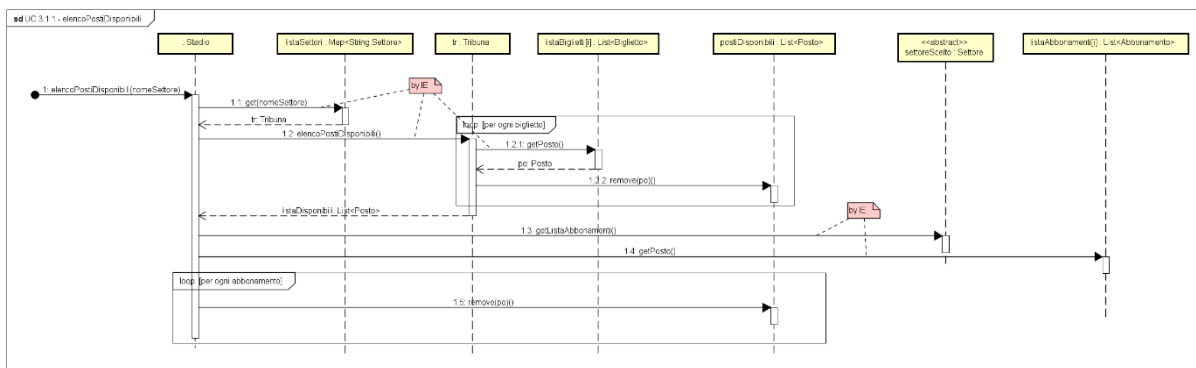


Figure 19: SD elencoPostiDisponibili

Diagramma di sequenza dell'operazione datiClienteAbbonamento:

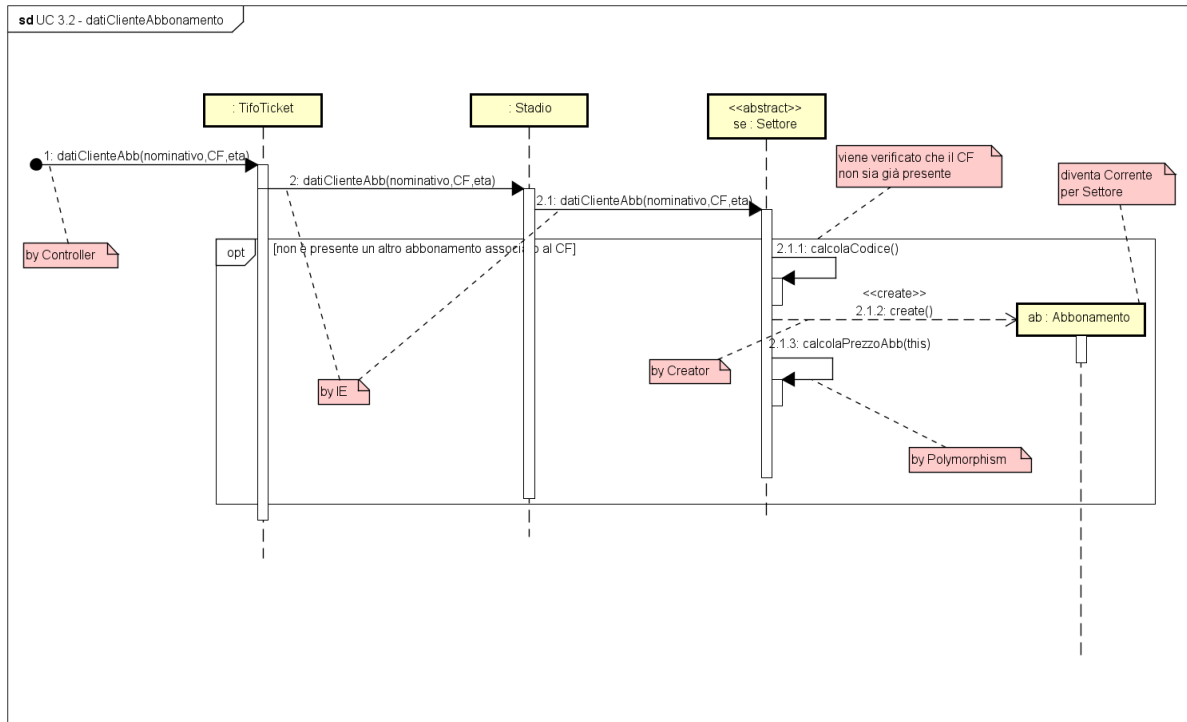


Figure 20: SD datiClienteAbbonamento

Di seguito i 2 diagrammi dell'operazione polimorfica *calcolaPrezzoAbb*:

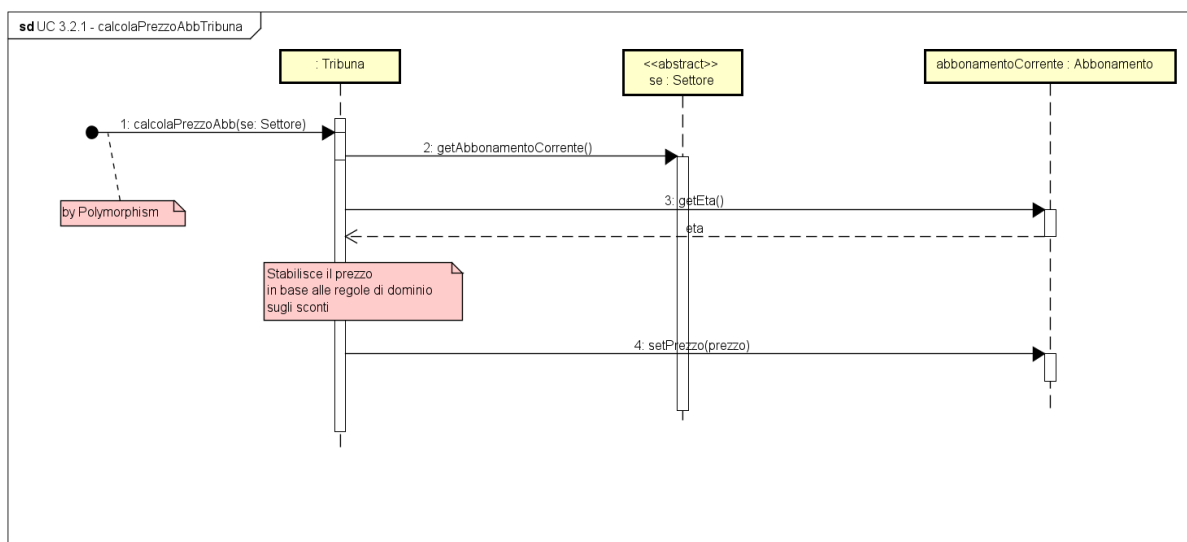


Figure 21: SD di calcolaPrezzoAbb variante Tribuna

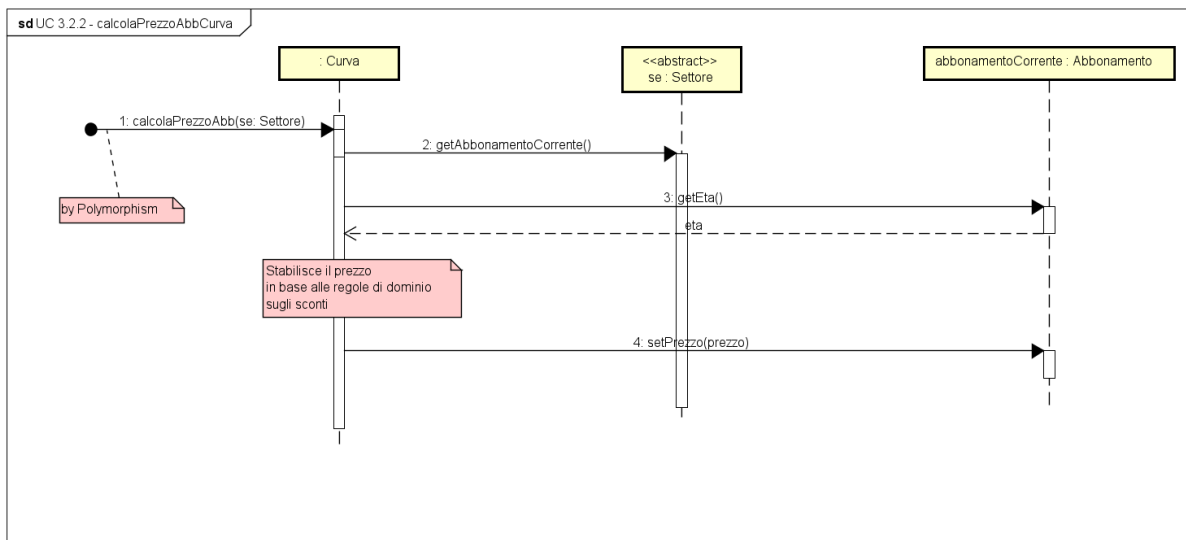


Figure 22: SD calcolaPrezzoAbb variante Curva

Diagramma di sequenza dell'operazione postoAbbonamento:

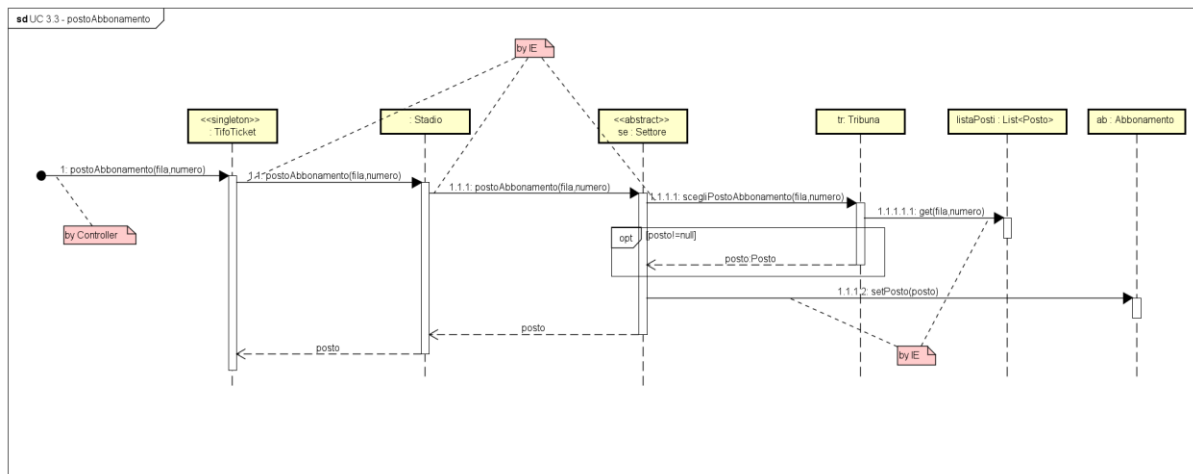


Figure 23: SD postoAbbonamento

Diagramma di sequenza dell'operazione confermaAbbonamento:

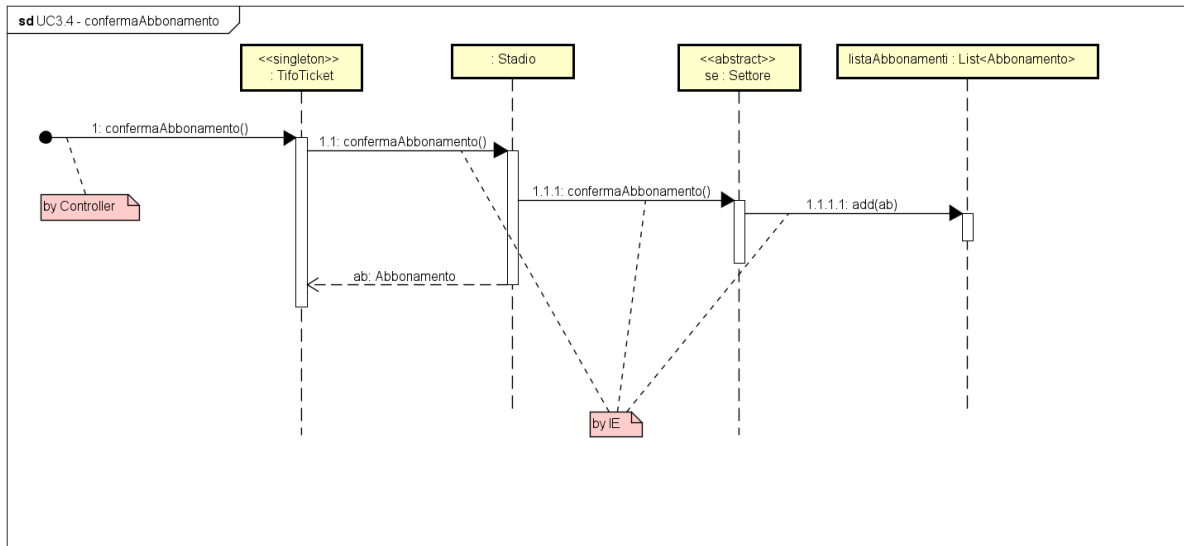


Figure 24: SD confermaAbbonamento

7.2 Diagramma delle classi di Progetto

Al termine delle prime 3 iterazioni di cui sono stati oggetto i primi tre casi d'uso UC1, UC2 e UC3 il diagramma delle classi di progetto risulta il seguente.

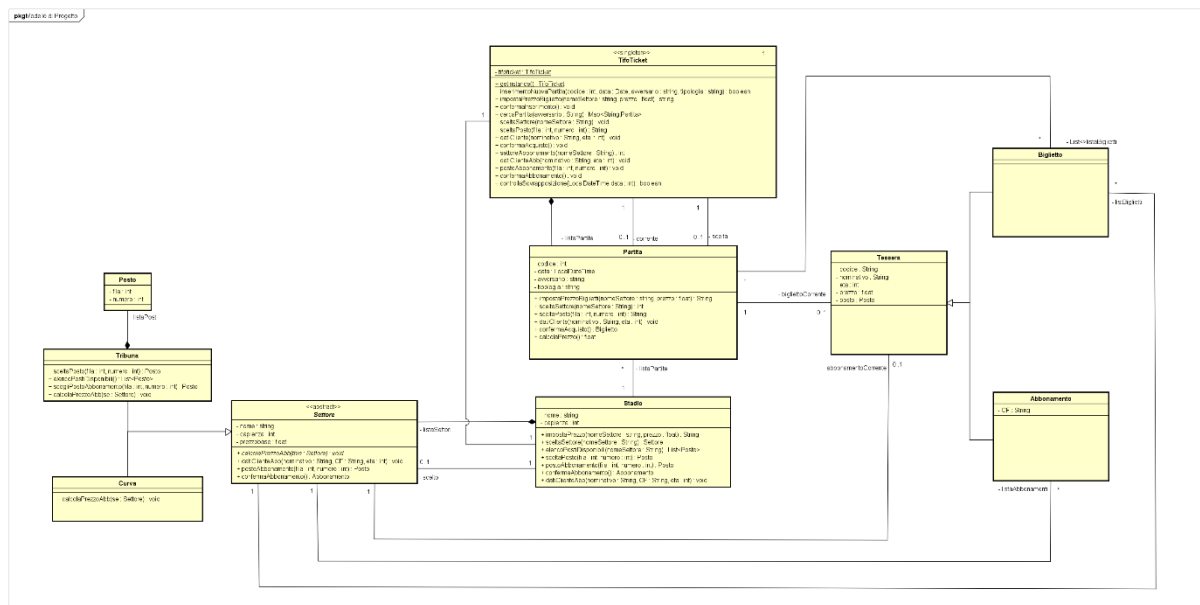


Figure 25: DCD iterazione 3

Come si può notare dal DCD e dai diagrammi di Sequenza del paragrafo precedente, è stato applicato il pattern **polymorphism** (GoF) alla classe *Settore* per far sì che i comportamenti correlati varino con il tipo di oggetti in questione. In questo caso viene utilizzato per far variare la modalità con cui viene stabilito il prezzo di un abbonamento a seconda che questo sia riferito ad un settore di tipo *Tribuna* o *Curva*. Questa operazione tiene conto del fatto che

sono state decise regole di dominio diverse che stabiliscono i prezzi di Abbonamenti in Curva e in Tribuna.

7.3 Re-factoring del codice

Al termine di questa terza iterazione si è optato per un re-factoring del codice ottenuto, il quale permette di cambiare l'organizzazione del codice senza cambiarne il comportamento complessivo. Nello svolgere questa attività, viene scelto di usare due design pattern GoF: **Factory** e **Strategy**, i quali mi permettono di selezionare a Runtime la regola corretta per il calcolo dei prezzi dei biglietti, a seconda della tipologia della partita e dell'età del tifoso che acquista il biglietto.

Dal codice così ottenuto, tramite reverse engineering, sono state apportate delle modifiche ad alcuni dei diagrammi mostrati in precedenza, i quali mostrano il progetto effettivo allo stato attuale. Questo capitolo ha lo scopo di illustrare questi diagrammi.

Un primo cambiamento dovuto al re-factoring si nota nel diagramma di sequenza **UC 2.5.1** dell'operazione **calcolaPrezzo**, derivata dall'analisi del caso d'uso UC2:

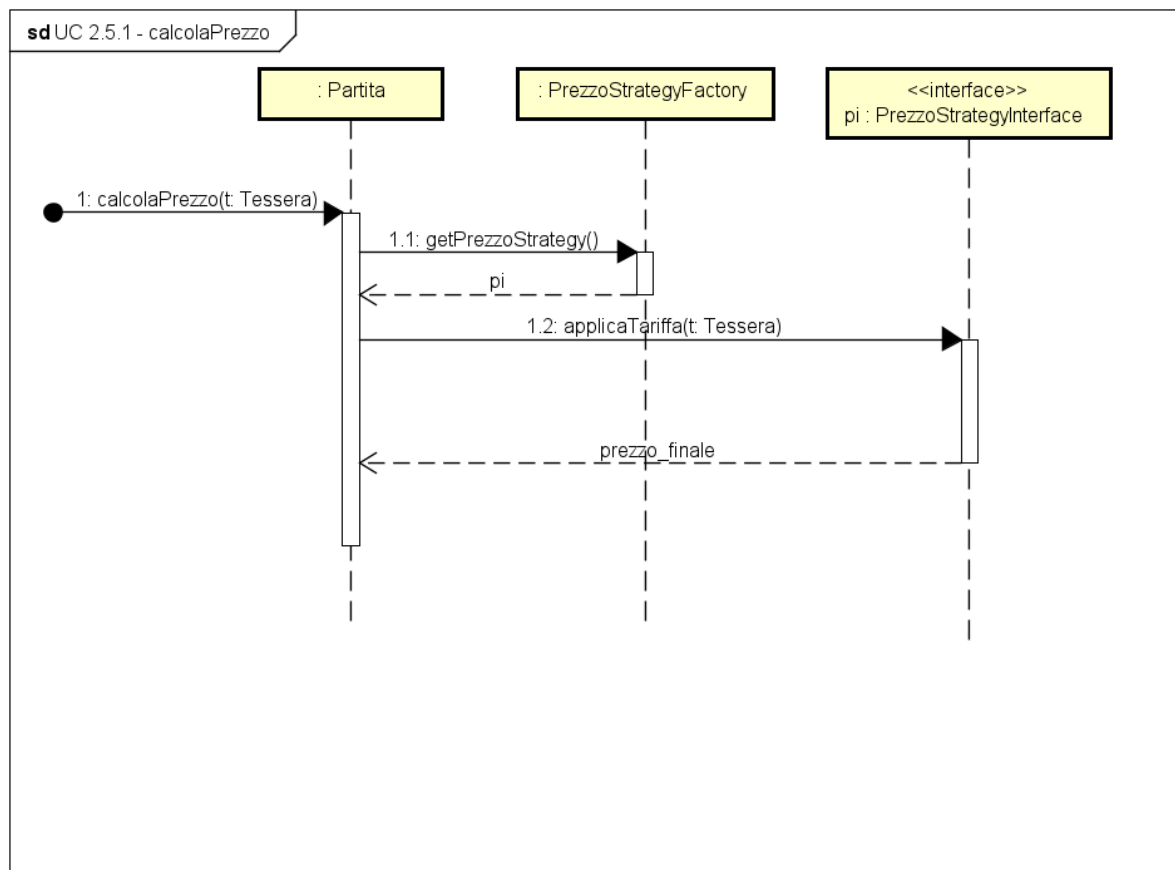


Figure 26: SD aggiornato operazione calcolaPrezzo

La principale differenza con quello di Figure 13: SD confermaAcquisto, è il fatto che il prezzo non viene più stabilito all'interno della classe *Partita* stessa, ma grazie al pattern **Strategy**

sarà possibile definire molteplici implementazioni diverse per l'operazione *applicaTariffa*. Tutto ciò porta ad una maggiore pulizia sia nei diagrammi che nel codice stesso.

L'altro cambiamento è riportato all'interno del DCD poiché la scelta di applicare nuovi pattern ha portato alla creazione di ulteriori 3 classi software:

- *PrezzoStrategyFactory*: La factory che si occupa di fornire l'interfaccia per la creazione di un oggetto, implementata come classe Singleton.
- *PrezzoStrategyInterface*: L'interfaccia che fornisce il metodo astratto per il calcolo del prezzo di un biglietto.
- *PrezzoStrategy*: La classe che implementa l'interfaccia descritta sopra e quindi fornisce l'implementazione concreta della logica del calcolo del prezzo di un biglietto.

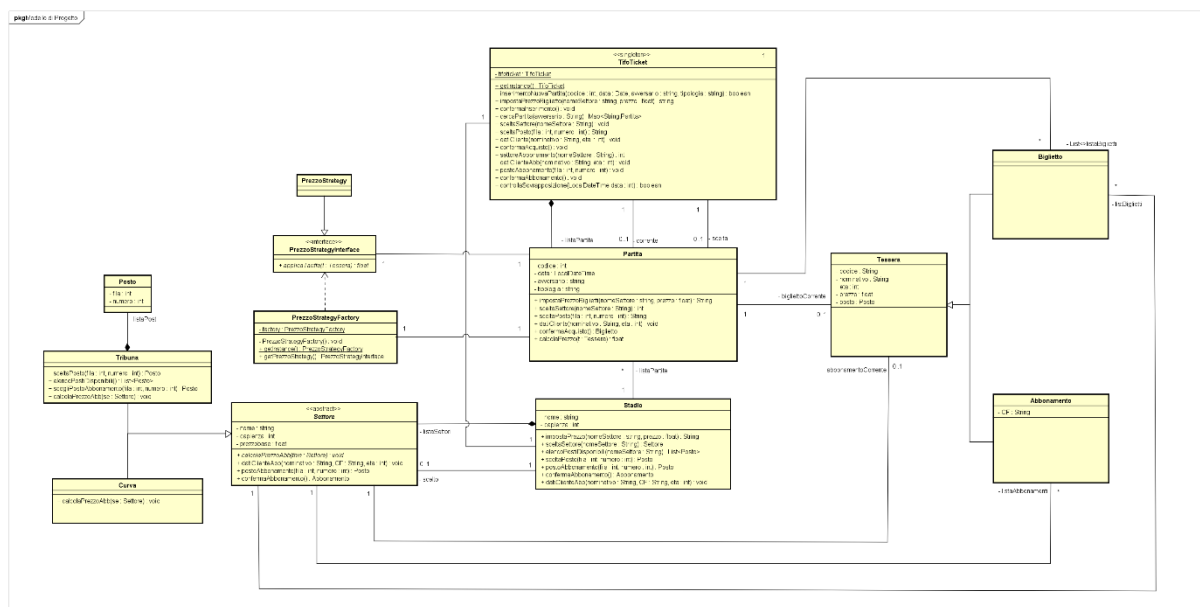


Figure 27: DCD dopo il re-factoring

Iterazione 4: Analisi

8.1 Introduzione

Per l'iterazione 4 è stato scelto di utilizzare i seguenti requisiti:

- Scenario principale di successo del caso d'uso UC6: Visualizza informazioni vendite partita.
- Scenario principale di successo del caso d'uso UC7: Visualizza informazioni vendite complessive.
- Dati solo in memoria principale

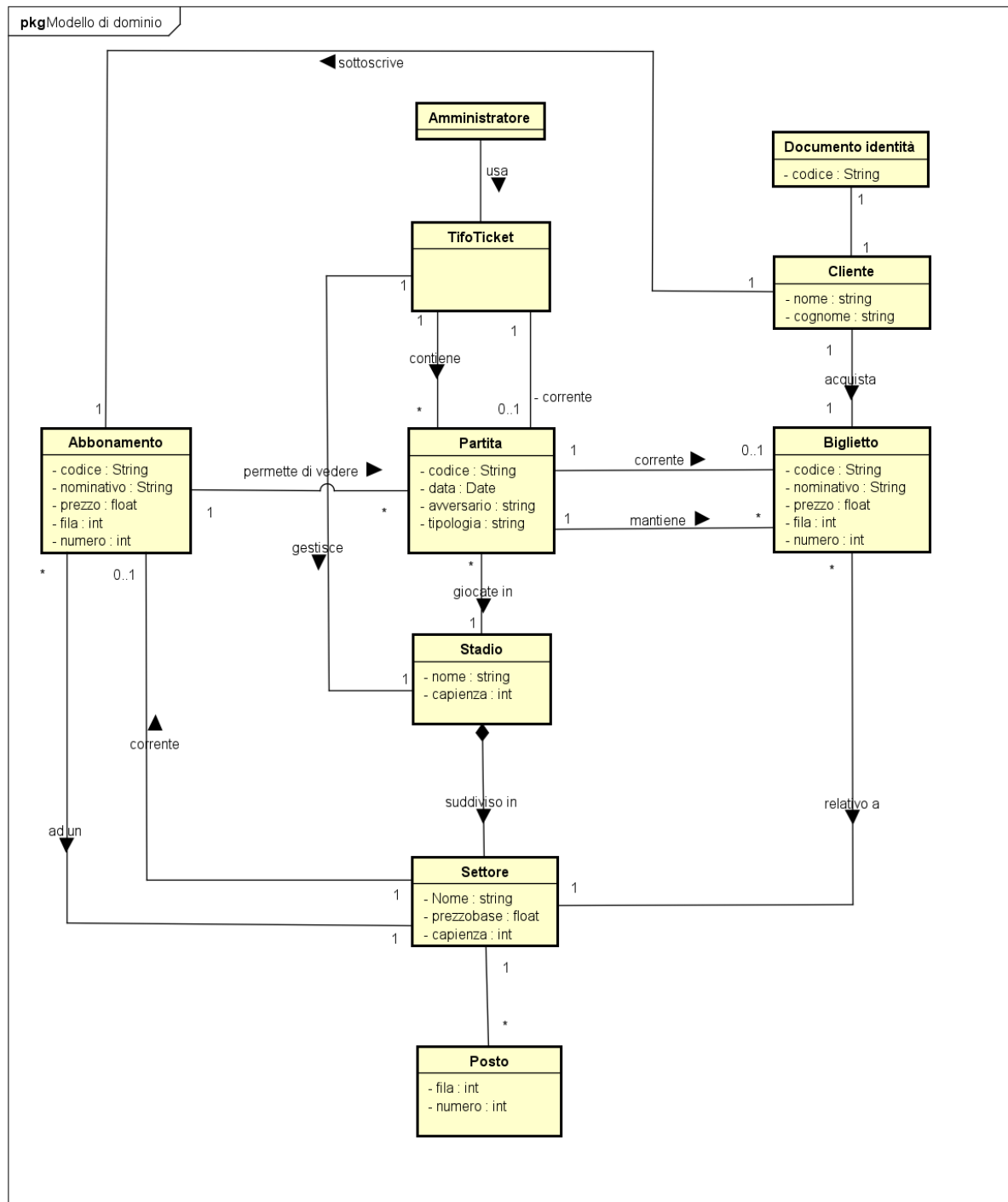
Per questa iterazione sono stati attenzionati due casi d'uso simili, entrambi permettono all'amministratore del sistema di ottenere informazioni circa le vendite dei biglietti. Il caso

d'uso UC7 fornisce una visione più generale, mentre UC6 permette di avere informazioni precise sui biglietti venduti in una specifica partita.

8.2 Modello di dominio

Data la natura dei casi d'uso scelti per lo svolgimento di questa iterazione, non ci sono cambiamenti effettivi sul modello di dominio rispetto allo schema mostrato in Figure 16: Modello di dominio UC3.

Pertanto, verrà solo riportato per comodità:



8.3 Diagrammi di Sequenza di Sistema

8.3.1 SSD del caso d'uso UC6

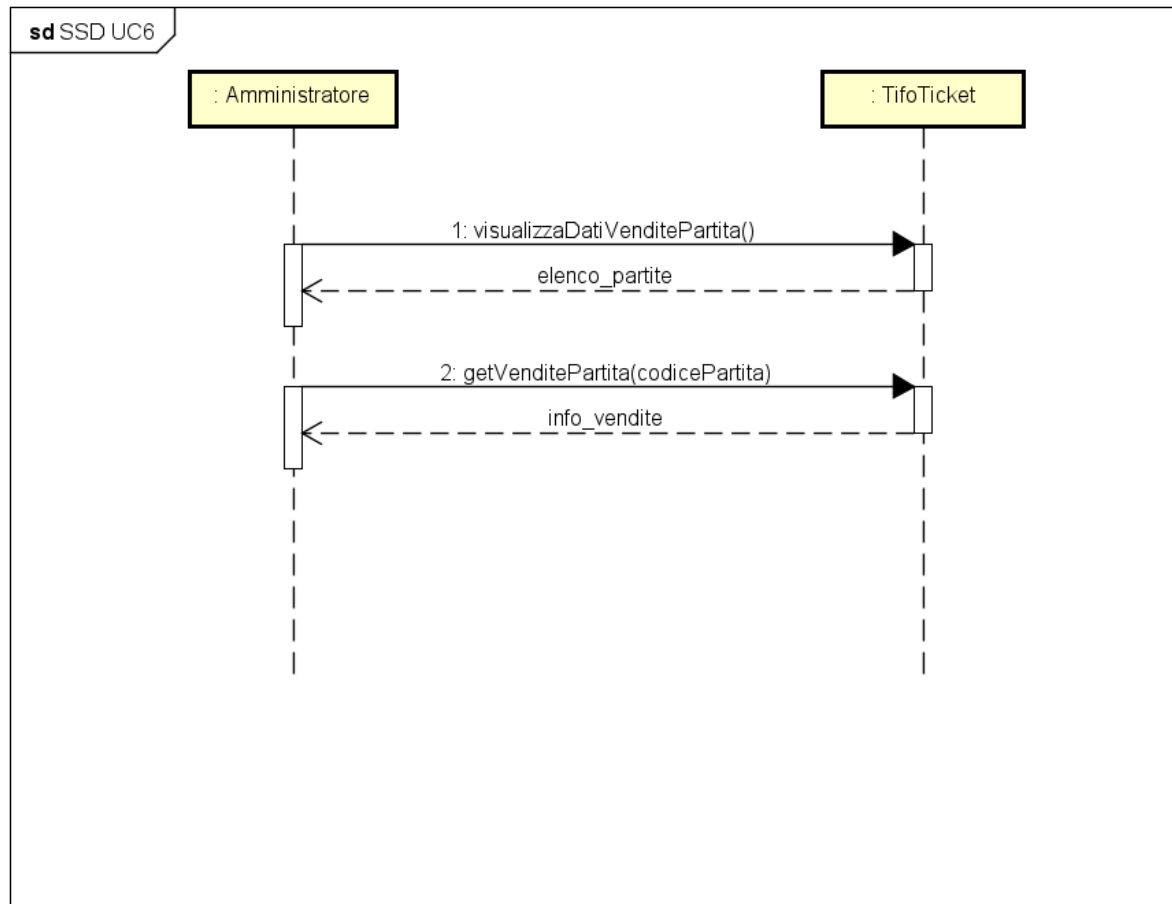


Figure 28: SSD UC6

Il diagramma di Sequenza di sistema riportato sopra individua 2 operazioni:

1. *visualizzaDatiVenditePartita*: con cui viene chiesto al sistema di mostrare le partite presenti in memoria.
2. *getVenditePartita*: con cui vengono visualizzate le informazioni sui biglietti venduti per ciascun settore dello stadio relativi ad una determinata Partita.

8.3.1 SSD del caso d'uso UC7

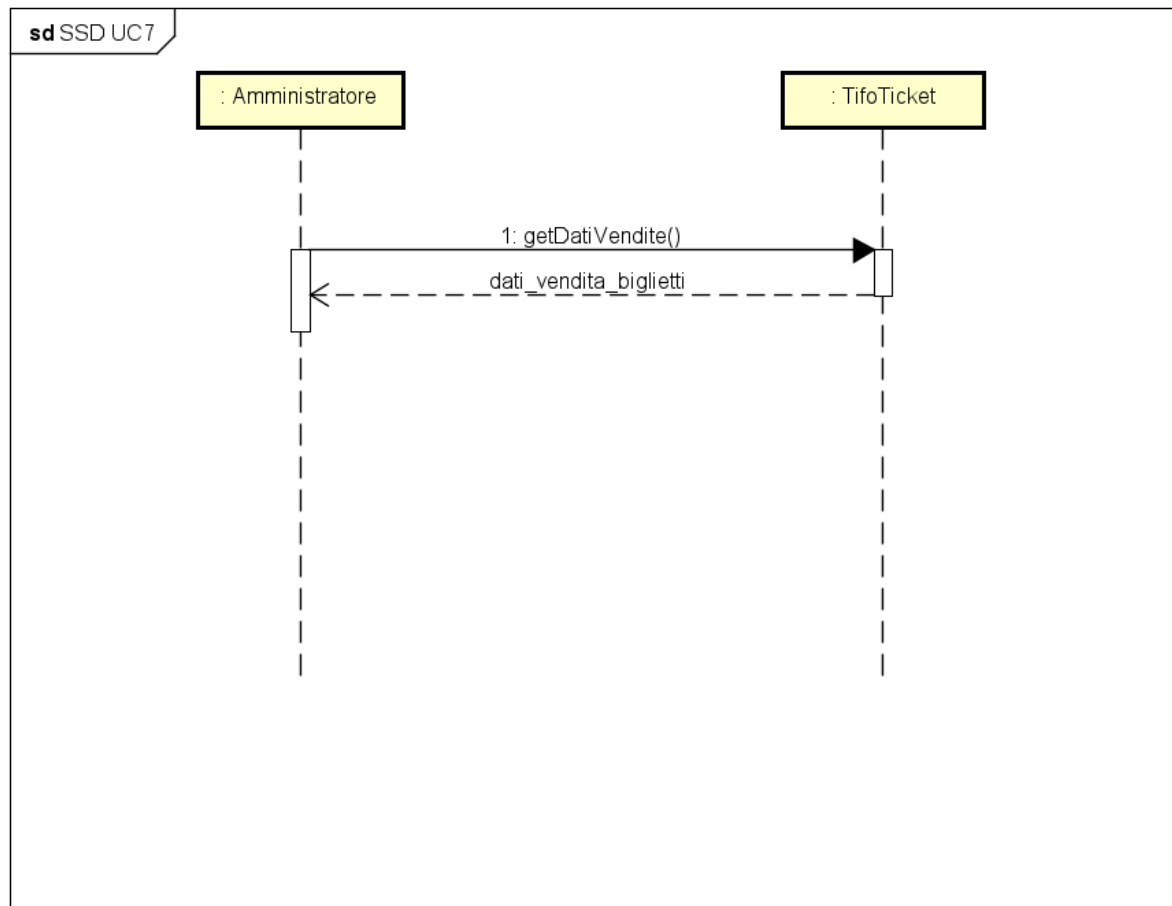


Figure 29: SSD UC7

Il diagramma di Sequenza di sistema riportato sopra individua una singola operazione:

1. *getDatiVenditeTotali*: con cui viene chiesto al sistema di mostrare i dati relativi al numero di biglietti venduti per ogni partita presente in memoria.

Iterazione 4: Progettazione

9.1 diagrammi di interazione

Diagramma di sequenza dell'operazione getVenditePartita:

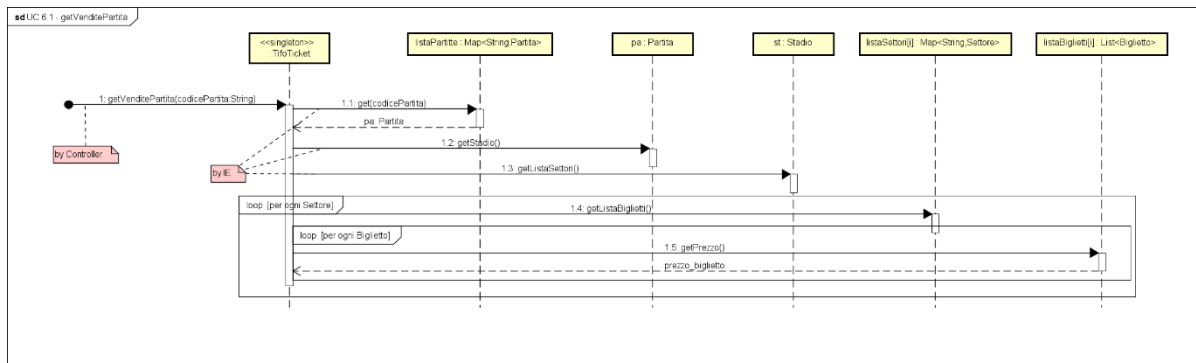
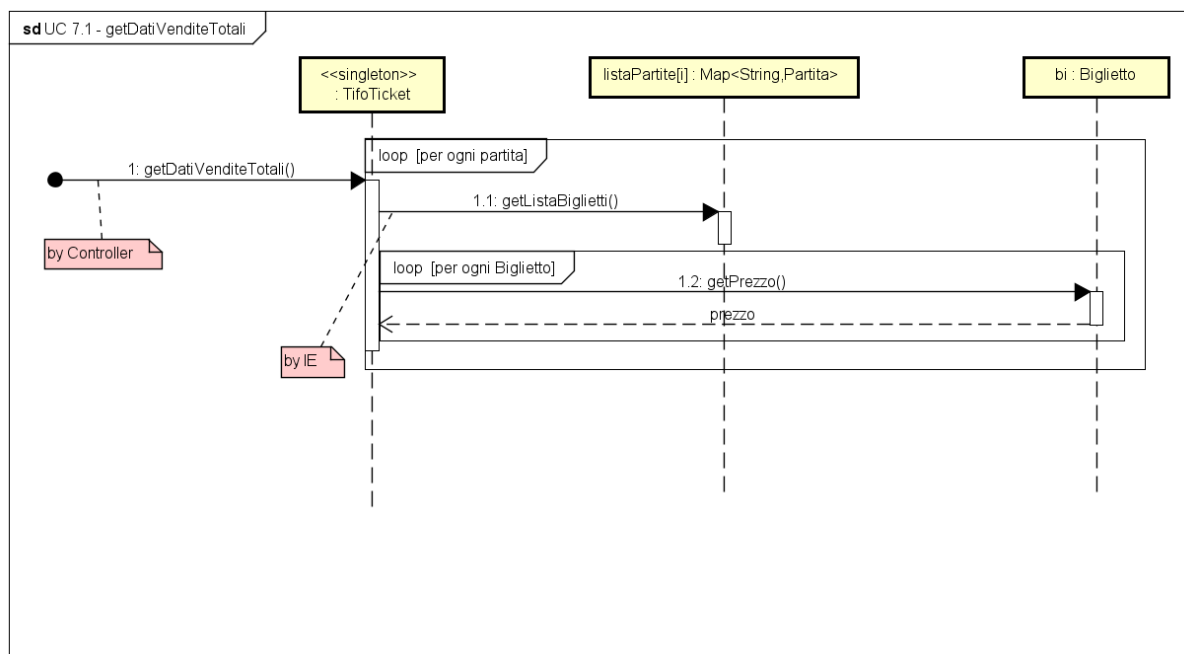


Diagramma di sequenza dell'operazione getDataVenditeTotali:



9.2 Diagramma delle classi di Progetto

Al termine dell'iterazione di cui sono stati oggetto i casi d'uso UC6 e UC7 il numero delle classi è rimasto invariato, poiché non sono state introdotte nuove classi software, ma nella classe TifoTicket sono state introdotte 2 nuove operazioni, quindi il DCD risulta il seguente:

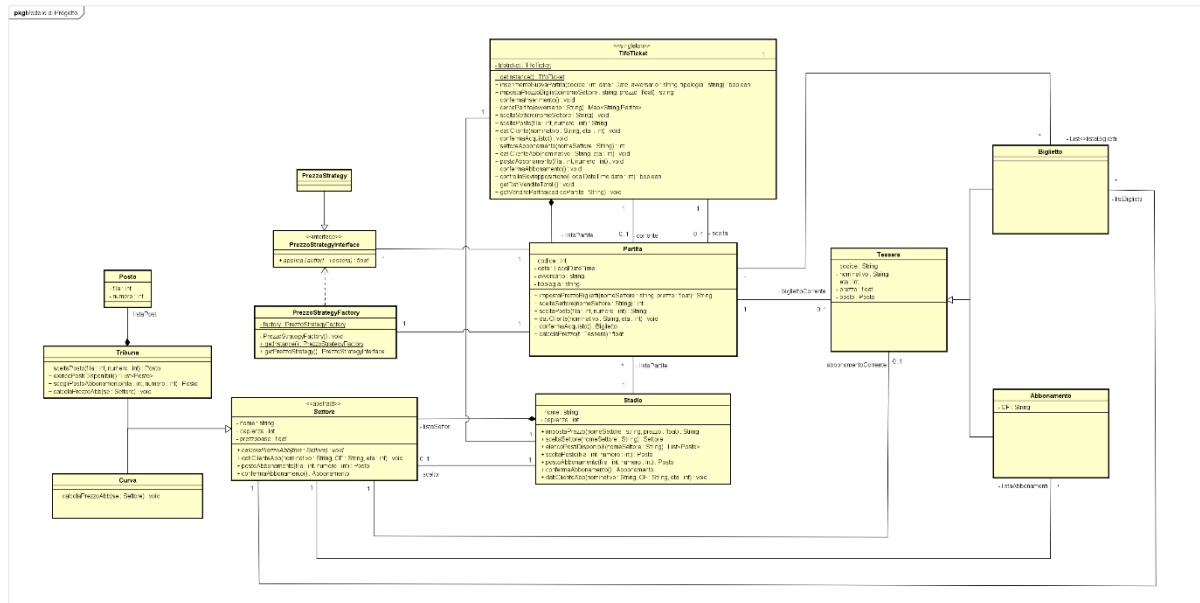


Figure 30:DCD iterazione 4

Iterazione 5: Analisi

10.1 Introduzione

Per l'iterazione 5 è si è scelto di utilizzare i seguenti requisiti:

- Scenario principale di successo del caso d'uso UC5: Gestisci sostituzione nominativo.
- Dati solo in memoria principale
- Il calcolo del prezzo di un biglietto è stabilito dalle regole di dominio B, C, D ed E.
- La politica di gestione delle sostituzioni di nominativo è definita dalla regola di dominio F.

Questo capitolo descrive l'analisi svolta nell'iterazione 5, mentre il capitolo successivo descrive l'attività di progettazione.

10.2 Modello di dominio

In questa iterazione è stato attenzionato il caso d'uso UC5 nella sua interezza, come descritto nel Paragrafo 1.2.5. Dall'analisi di esso non sono state create nuove classi

concettuali nel modello di dominio che quindi rimane invariato rispetto a quello descritto dalla Figure 16: Modello di dominio UC3.

10.3 Diagramma di sequenza di sistema

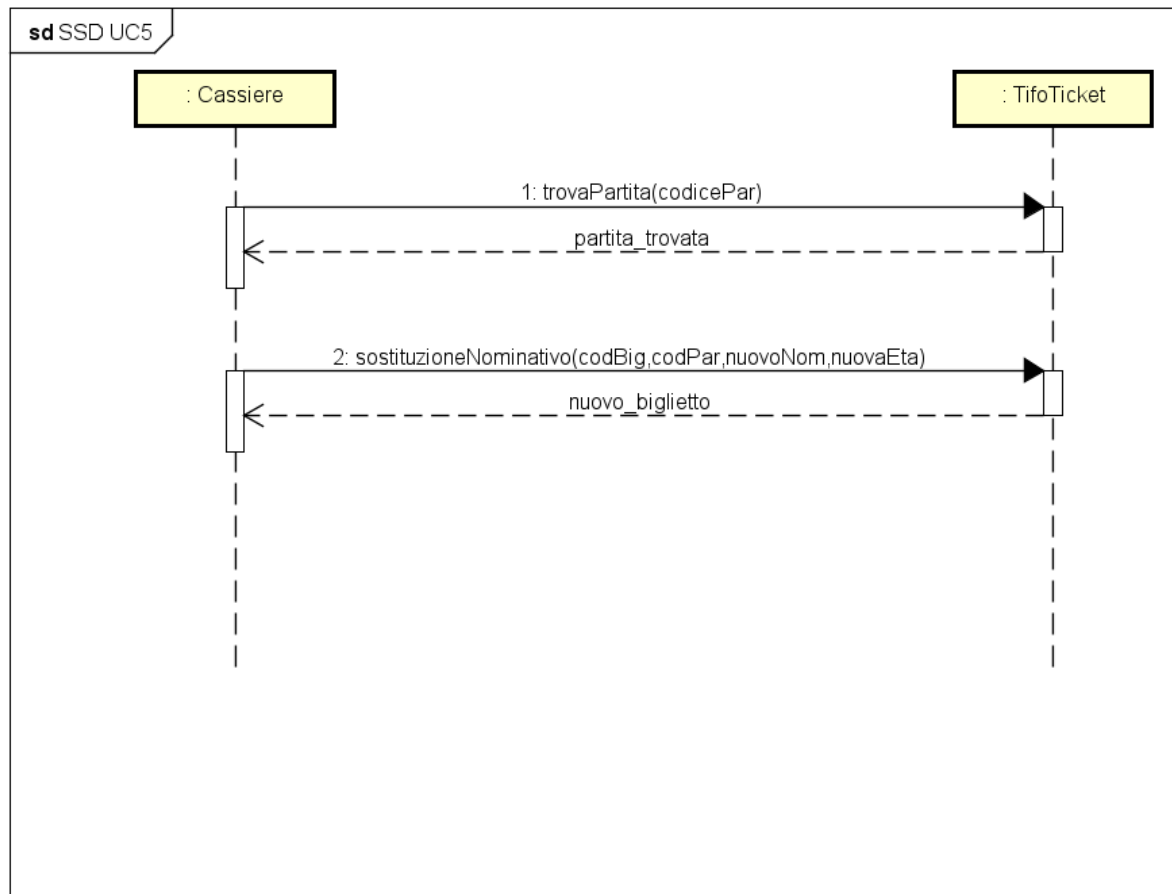


Figure 31:SSD UC5

Come si evince dal diagramma l'implementazione di questo caso d'uso ha portato all'individuazione di 2 operazioni: *trovaPartita* e *gestisciSostituzione*. Queste operazioni permettono, individuato il biglietto relativo ad una partita, di sostituire il nominativo e l'età presenti su di esso in base alle condizioni dettate dalla regola di dominio F.

Iterazione 5: Progettazione

11.1 Diagrammi di Interazione

Diagramma di sequenza dell'operazione trovaPartita:

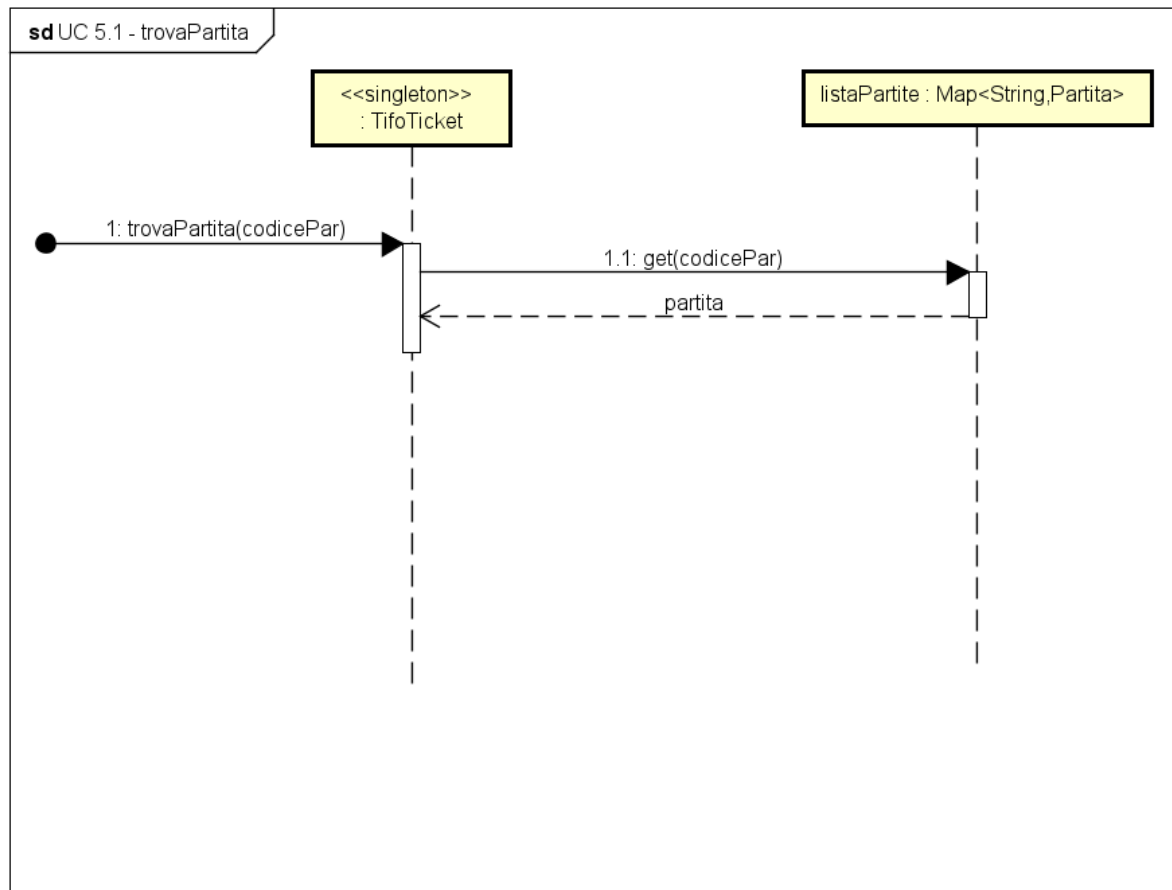


Figure 32: SD trovaPartita

Diagramma di sequenza dell'operazione sostituzioneNominativo:

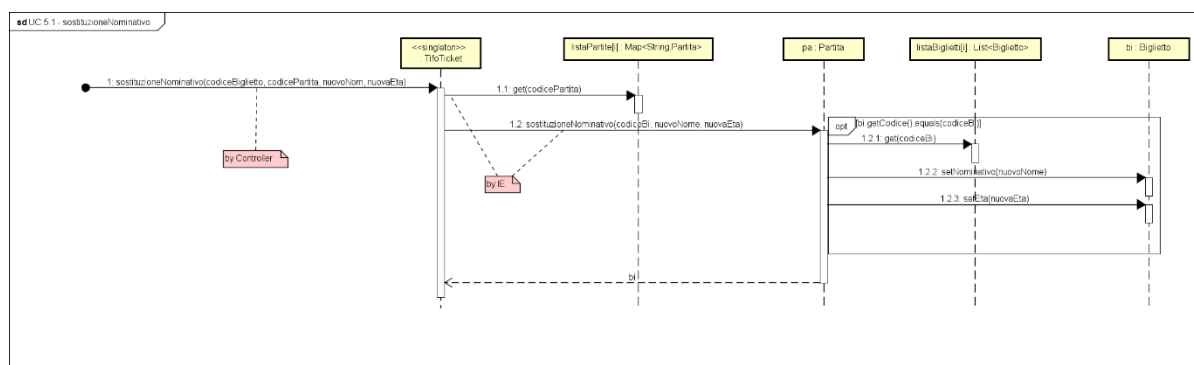


Figure 33:SD sostituzioneNominativo

11.2 Diagramma delle Classi di Progetto

Al termine dell'iterazione 5, di cui è stato oggetto i casi d'uso UC5 il numero delle classi è rimasto invariato, poiché non sono state introdotte nuove classi software, ma nelle classi TifoTicket e Partita è stata introdotta una nuova operazione, pertanto, il DCD risulta il seguente:

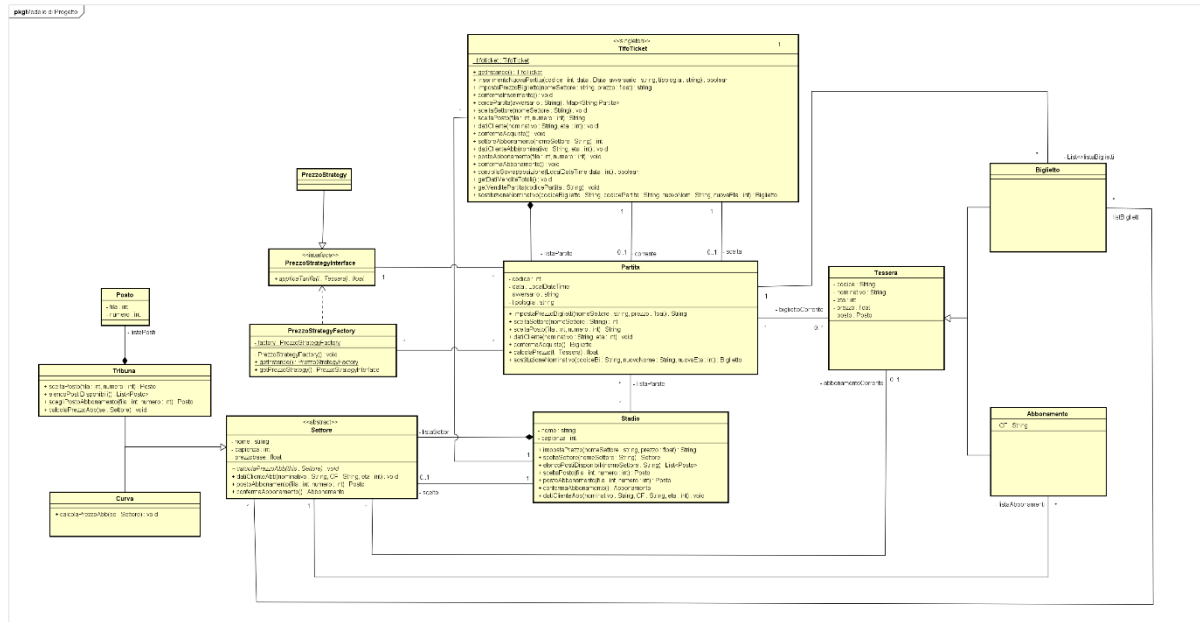


Figure 34:DCD iterazione 5

Testing

12.1 Introduzione

Al termine di ogni iterazione sono state effettuate operazioni di testing sul codice prodotto come risultato dell'iterazione stessa. Ciò per verificarne la correttezza e per trovare e risolvere eventuali bugs. In questo capitolo verrà descritto per sommi capi come sono state costruite le varie classi di Test ed i relativi test cases. Il testing è stato condotto prevalentemente in modalità **white-box**; dunque, sono stati effettuati test strutturali sul software. Avendo buona conoscenza del codice sorgente, si è provato a trovare un buon numero test cases per le classi software principali, in modo da ricoprire tutti i possibili cammini all'interno del codice.

I test svolti sono **unit tests**, pertanto servono a dimostrare il corretto funzionamento di ogni unità che compone il codice, in questo caso intesa come classe software. I test white-box rilevano potenzialmente molti difetti all'inizio della scrittura e aiuta a risolvere i difetti che si verificano successivamente dopo che il codice è stato integrato con il resto dell'applicazione e quindi riduce l'impatto degli errori più avanti nello sviluppo.

L'implementazione di questi test è stata automatizzata, utilizzando il framework predisposto **JUnit**, versione 5.

12.2 Le classi di Test

Per l'esecuzione dei test sono state create delle classi apposite. Come da convenzione il nome di queste classi segue il formato *Test + 'nome classe software'*, le classi di test sono le seguenti:

- *TestTifoTicket*
- *TestStadio*
- *TestPartita*
- *TestSettore*
- *TestTribuna*

In ciascuna di queste classi sono definiti i metodi di test per le funzioni più importanti ai fini dell'esecuzione del software.

Indice delle Figure

Figure 1: modello di dominio Iterazione 1	10
Figure 2: SSD UC1	11
Figure 3: SD inserimentoNuovaPartita	13
Figure 4: SD inserimentoPrezzoBiglietto	13
Figure 5: SD confermaInserimento.....	14
Figure 6: DCD iterazione 1	15
Figure 7: Modello di dominio Iterazione 2.....	16
Figure 8: SSD UC2.....	17
Figure 9: SD cercaPartita.....	19
Figure 10: SD sceltaSettore.....	20
Figure 11: SD sceltaPosto	20
Figure 12: SD datiCliente	21
Figure 13: SD confermaAcquisto.....	21
Figure 14: SD calcolaPrezzo	22
Figure 15: DCD iterazione 2.....	23
Figure 16: Modello di dominio UC3	25
Figure 17: SSD UC3.....	26
Figure 18: SD settoreAbbonamento	28
Figure 19: SD elencoPostiDisponibili	28
Figure 20: SD datiClienteAbbonamento	29
Figure 21: SD di calcolaPrezzoAbb variante Tribuna	29
Figure 22: SD calcolaPrezzoAbb variante Curva	30
Figure 23: SD postoAbbonamento	30
Figure 24: SD confermaAbbonamento.....	31
Figure 25: DCD iterazione 3.....	31
Figure 26: SD aggiornato operazione calcolaPrezzo	32
Figure 27: DCD dopo il re-factoring	33
Figure 28: SSD UC6.....	35
Figure 29: SSD UC7	36
Figure 30:DCD iterazione 4.....	38
Figure 31:SSD UC5.....	39
Figure 32: SD trovaPartita	40
Figure 33:SD sostituzioneNominativo.....	40
Figure 34:DCD iterazione 5.....	41