

Iterazione 3

1.0 Introduzione

La terza iterazione del progetto *TeamManager* arricchisce l'applicazione con funzionalità di analisi aggregate e supporto decisionale avanzato, consolidando il ruolo del software come strumento strategico per l'allenatore. Dopo aver implementato la gestione di base dei giocatori, della rosa e degli eventi con le annesse disponibilità (Iterazione 1) e le funzionalità di registrazione delle statistiche individuali strumento per le creazioni di sessioni mirate per colmare specifiche lacune e per confrontare diversi giocatori (Iterazione 2), questa fase si concentra sull'elaborazione di dati a livello di Rosa e sulla generazione di suggerimenti tattici automatizzati. Nello specifico, l'Iterazione 3 prevede la progettazione e l'implementazione degli scenari principali per i seguenti casi d'uso:

UC4: Andamento Rosa: per fornire all'allenatore una visione d'insieme delle performance della Rosa attraverso report aggregati basati sui dati raccolti nelle amichevoli.

UC8: Suggerisci formazione: per elaborare automaticamente una formazione consigliata basandosi su un'analisi combinata dello stato di forma, delle performance recenti e della disponibilità dei giocatori.

Tutte le nuove funzionalità manterranno la struttura e la coerenza dell'architettura definita nella seconda iterazione, facendo uso di file persistenti per la memorizzazione dei dati e del pattern **Facade Controller** tramite la classe *TeamManager*. L'obiettivo di questa iterazione è trasformare i dati individuali raccolti finora in informazioni strategiche a livello di squadra, offrendo all'allenatore strumenti potenti per valutare l'efficacia collettiva e ottimizzare le scelte tattiche.

1.1 Affinamento regole di dominio

Durante questa fase, le regole di dominio del sistema sono state oggetto di una revisione approfondita per migliorarne la coerenza e la robustezza. In particolare, sono state

introdotte nuove regole focalizzate sull'analisi aggregata delle performance di squadra e sulla generazione di suggerimenti tattici, in linea con i nuovi casi d'uso introdotti.

ID	Regola	Modificabilità	Sorgente
R1	Ogni giocatore registrato non può essere rimosso dall'allenatore senza una motivazione valida (es. infortunio prolungato, uscita dalla squadra, comportamento inadeguato, etc.).	Media, potrebbe essere necessario prevedere la gestione di ex-giocatori.	Regola interna della squadra.
R2	Il giocatore deve avere un identificativo univoco, non sono consentiti duplicati e non può essere inserito più volte all'interno della stessa rosa.	Bassa, strettamente legata alla gestione della rosa.	Regola interna del sistema.
R3	La rosa può contenere un massimo di 22 giocatori , qualora la rosa fosse piena nessun altro giocatore potrà essere inserito al suo interno	Bassa, strettamente legata alla gestione della rosa.	Regola interna della squadra.
R4	Un giocatore può avere uno e un solo stato tra: Disponibile, Infortunato o Sospeso .	Media, nuove categorie potrebbero essere aggiunte in futuro.	Regola interna del sistema.
R5	Non ci possono essere due giocatori con lo stesso numero di maglia all'interno della rosa	Bassa, strettamente legata alla gestione della rosa.	Regola interna della squadra.
R6	Ogni giocatore inserito nella rosa deve avere un ruolo assegnato dall'enumerazione predefinita (PORTIERE, DIFENSORE, ecc.).	Media, nuove categorie potrebbero essere aggiunte in futuro.	Regola interna del sistema.
R7	Più eventi possono essere pianificati nello stesso giorno, ma non possono sovrapporsi .	Bassa, modificabile solo con una revisione del modello degli eventi.	Regola interna del sistema per evitare conflitti.
R8	L'analisi delle performance può essere effettuata solo su giocatori esistenti in rosa .	Bassa, strettamente legata alla gestione della rosa.	Regola tecnica interna.
R9	Le statistiche di performance relative a un evento possono essere registrate solo per i giocatori che hanno confermato la loro presenza e vi hanno effettivamente partecipato.	bassa, strettamente legata alla partecipazione dei giocatori al suddetto evento	Regola tecnica interna.

R10	Il sistema può suggerire una formazione solo se la Rosa risulta completa.	Bassa, strettamente legata alla generazione dei dati	Regola interna del sistema.
R11	Il sistema dispone all'allenatore un set di combinazioni di formazioni disponibili (4-3-3, 3-4-3, 3-5-2), al fine di suggerire una formazione.	Media, nuove formazioni potrebbero essere aggiunte in futuro.	Regola interna del sistema.

Di seguito è presentata la cronologia delle varie versioni delle regole di dominio del sistema, con un focus sulle eventuali modifiche effettuate a cui il sistema dovrà eventualmente adeguarsi

Versione	data	Descrizione	Autori
1.0	02/06/2025	Prima bozza presente all'interno del file "TeamManager"	Antonio Nicolò Scarvaglieri, Giuseppe Ravesi, Vincenzo Venezia
2.0	20/06/2025	Aggiunta R2,R3,R5.R6 e eliminazione regole superflue	Antonio Nicolò Scarvaglieri, Giuseppe Ravesi, Vincenzo Venezia
3.0	02/07/2025	Aggiunta R10, R11	Antonio Nicolò Scarvaglieri, Giuseppe Ravesi, Vincenzo Venezia

2.0 Fase di Analisi

2.1 Implementazione regole di dominio

Rimanendo in linea alle regole di dominio esplicitate nel paragrafo *1.1 – Affinamento delle Regole di Dominio*, sono state implementate le seguenti:

- **Verifica della completezza della rosa (R10)**, in quanto il suggerimento della formazione è possibile solamente con una rosa costituita da 22 giocatori.
- L'allenatore potrà **selezionare solo un set limitato di moduli (R11)**, tra cui 4-3-3, 3-5-2 e 3-4-3, in modo tale da ottenere la formazione con un numero, rispettivamente, di difensori, centrocampisti e attaccanti desiderato.

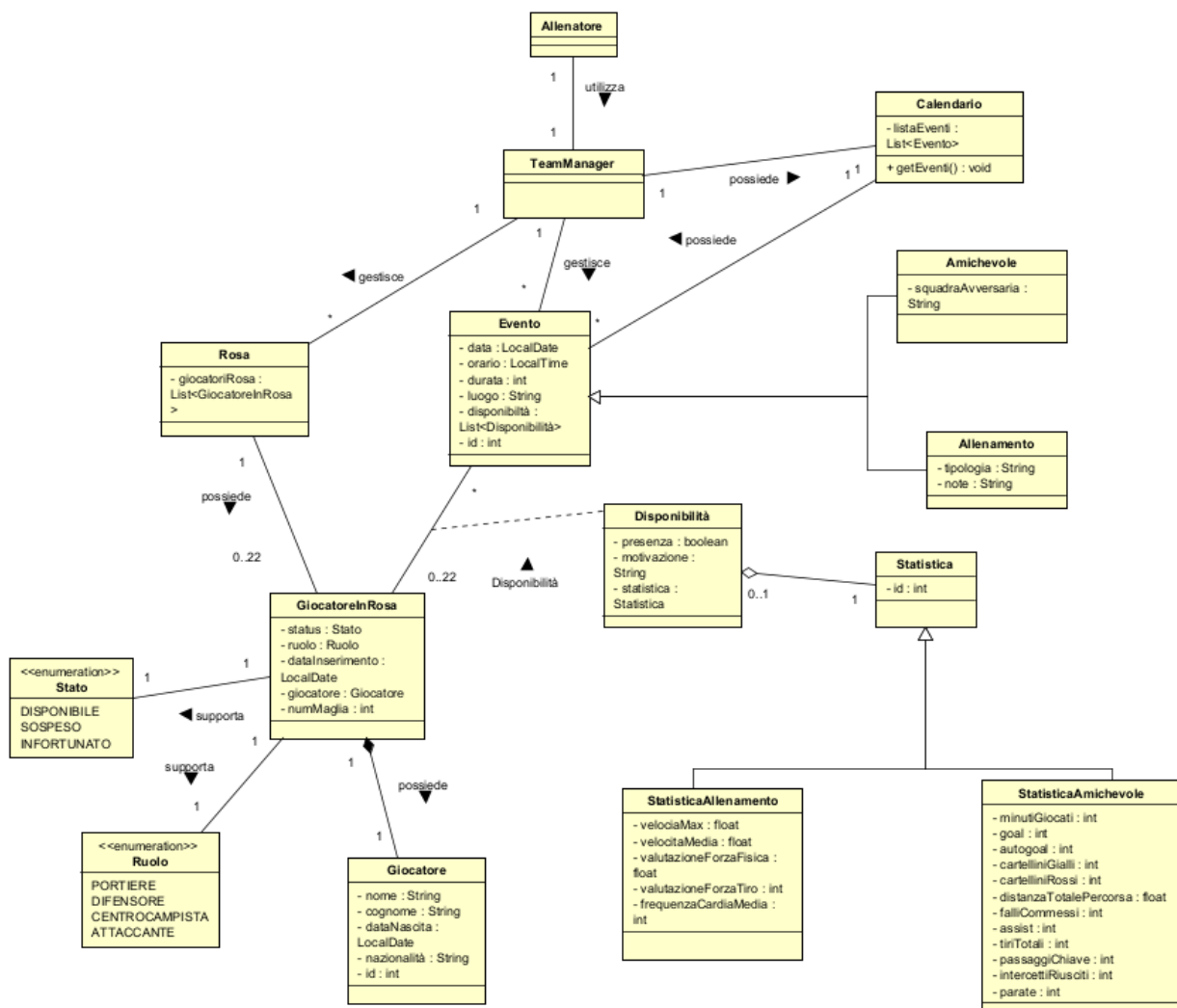
2.2 Classi concettuali derivate dai casi d'uso e modello di dominio

Come nella precedente iterazione anche in questa fase individuiamo le principali **classi concettuali** coinvolte nei casi d'uso previsti per questa seconda iterazione che saranno successivamente trasformate in componenti software all'interno del modello di dominio. Di seguito sono elencate le classi concettuali emerse dall'analisi dei casi d'uso **UC4** e **UC8**, accompagnate da una breve descrizione:

Classe	Descrizione
Giocatore	Rappresenta un individuo iscrivibile alla squadra. Contiene informazioni anagrafiche (nome, cognome, e-mail), tecniche (ruolo).
GiocatoreInRosa	Rappresenta il giocatore pronto ad essere inserito all'interno della rosa a cui vengono associati parametri esclusivi.
Rosa	Contiene i giocatori attualmente attivi nella squadra. È gestita dall'allenatore ed è derivata dall'archivio generale.
Evento	Rappresenta una sessione di squadra (allenamento o partita amichevole), con informazioni su data, orario, luogo e obiettivo.
Disponibilità	Associa un giocatore a un evento specifico, indicandone la disponibilità (presente/assente) e una possibile motivazione.

Allenatore	Figura responsabile della gestione della rosa, della creazione degli eventi e della consultazione delle disponibilità.
Statistica	Raccoglie i parametri individuali del giocatore in seguito a una sua partecipazione ad un evento.
TeamManager	Rappresenta il sistema TeamManager

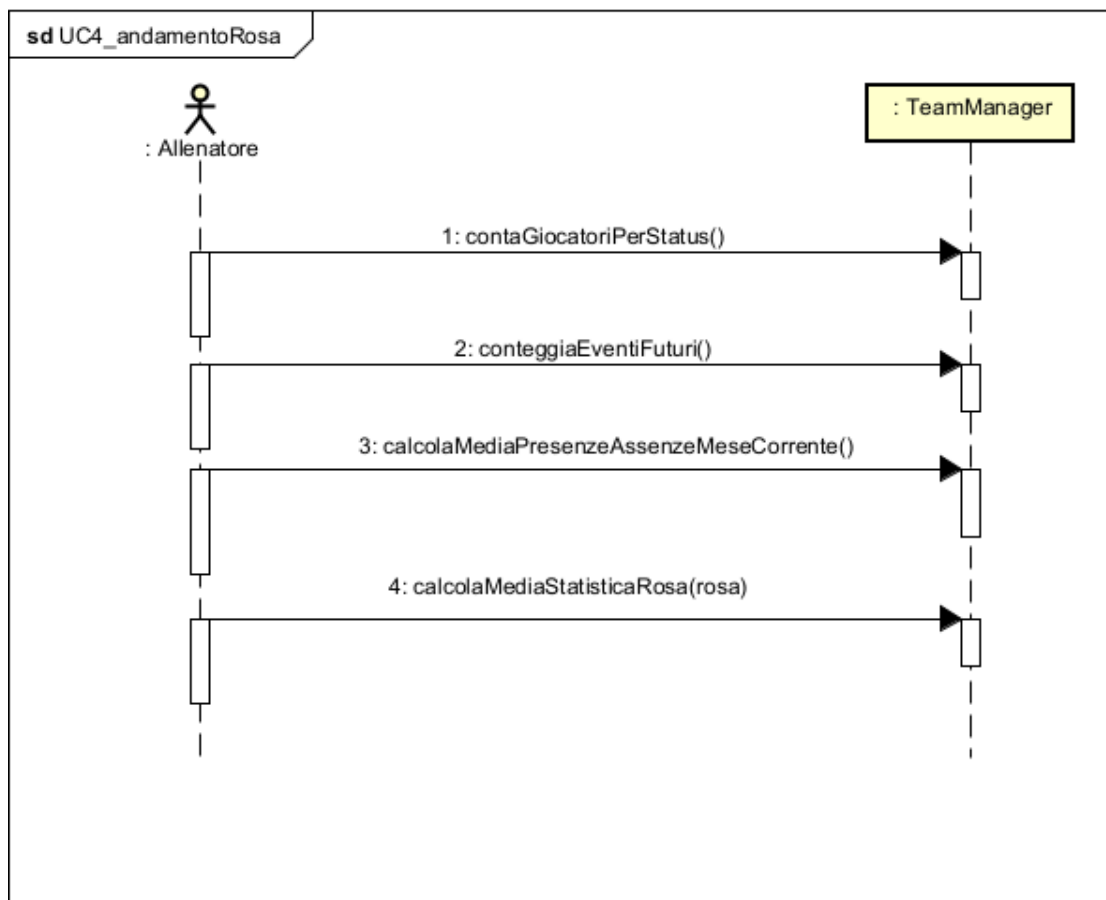
Da cui tenendo conto dei vari attributi e associazioni di ogni entità, si è generato il seguente modello di dominio che rimane pressoché invariato rispetto alla seconda iterazione.



2.3 Diagrammi di Sequenza di Sistema e Contratti delle Operazioni

UC4 – Andamento Rosa

I seguenti Diagrammi di Sequenza di Sistema illustrano le interazioni I/O dell'utente con il sistema TeamManager relative al Caso d'Uso 4 con i vari scenari di successo:



Contratti delle operazioni UC4

I contratti delle operazioni descrivono i principali metodi identificati nei SSD di UC4:

CONTRATTO 1 UC4: Andamento Rosa

- **Operazione:** contaGiocatoriPerStatus()
- **Riferimento:** UC4: Andamento Rosa
- **Pre-condizione:**
 1. L'allenatore accede al sistema.

2. Dal menu principale l'allenatore ha selezionato l'opzione rosa.
3. L'allenatore visualizza la rosa attuale.
4. L'allenatore seleziona l'opzione di Andamento rosa.
5. Il file contenente i vari giocatori presenti in rosa è accessibile in lettura

- **Post-condizione:**

1. L'allenatore visualizza a schermo un pannello in cui è riportata una sezione dove verranno mostrate le percentuali di giocatori disponibili, infortunati e sospesi.
2. Nessuna modifica viene apportata ai dati.

CONTRATTO 2 UC4: Andamento Rosa

- **Operazione:** conteggiaEventiFuturi()

- **Riferimento:** UC4: Andamento Rosa

- **Pre-condizione:**

1. L'allenatore accede al sistema.
2. Dal menu principale l'allenatore ha selezionato l'opzione rosa.
3. L'allenatore visualizza la rosa attuale.
4. L'allenatore seleziona l'opzione di Andamento rosa.
5. Il file contenente i vari giocatori presenti in rosa è accessibile in lettura

- **Post-condizione:**

1. L'allenatore visualizza a schermo un pannello in cui è riportata una sezione dove verranno mostrati il totale di eventi suddivisi in amichevoli e allenamenti.
2. Nessuna modifica viene apportata ai dati.

CONTRATTO 3 UC4: Andamento Rosa

- **Operazione:** calcolaMediaStatisticaRosa()

- **Riferimento:** UC4: Andamento Rosa

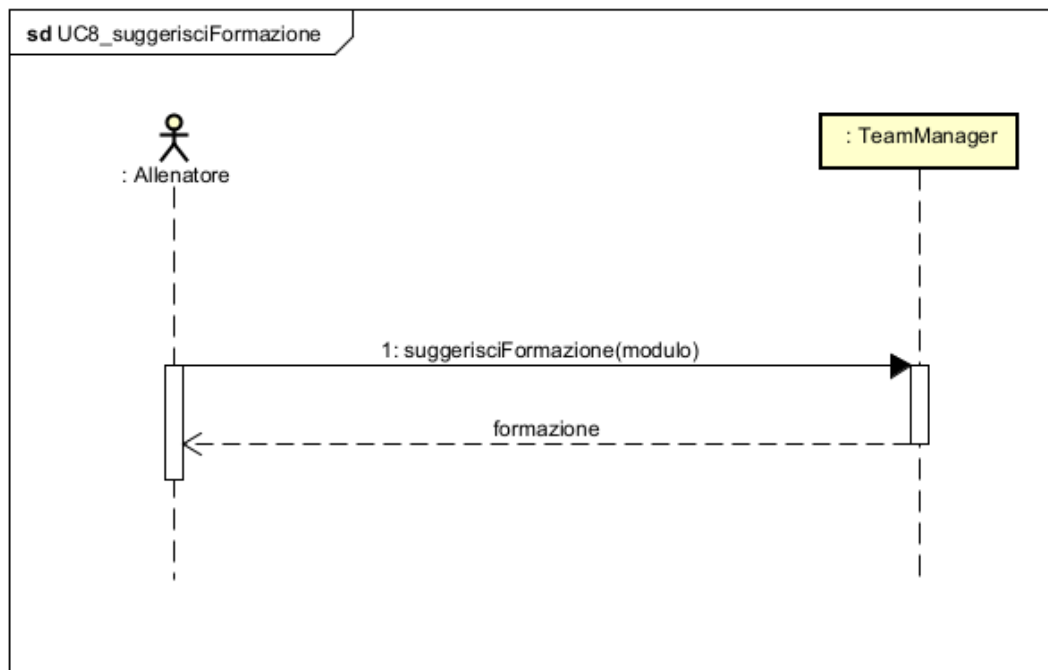
- **Pre-condizione:**

1. L'allenatore accede al sistema.

<ol style="list-style-type: none"> 2. Dal menu principale l'allenatore ha selezionato l'opzione rosa. 3. L'allenatore visualizza la rosa attuale. 4. L'allenatore seleziona l'opzione di Andamento rosa. 5. Il file contenente i vari giocatori presenti in rosa è accessibile in lettura <ul style="list-style-type: none"> • Post-condizione: <ol style="list-style-type: none"> 1. L'allenatore visualizza a schermo un pannello in cui è riportata una sezione dove verranno mostrate statistiche di rosa in percentuale. 2. Nessuna modifica viene apportata ai dati.
<p>CONTRATTO 4 UC4: Andamento Rosa</p> <ul style="list-style-type: none"> • Operazione: calcolaMediaStatisticaRosa() • Riferimento: UC4: Andamento Rosa • Pre-condizione: <ol style="list-style-type: none"> 1. L'allenatore accede al sistema. 2. Dal menu principale l'allenatore ha selezionato l'opzione rosa. 3. L'allenatore visualizza la rosa attuale. 4. L'allenatore seleziona l'opzione di Andamento rosa. 5. Il file contenente i vari giocatori presenti in rosa è accessibile in lettura • Post-condizione: <ol style="list-style-type: none"> 1. L'allenatore visualizza a schermo un pannello in cui è riportata una sezione dove verranno mostrate le percentuali di giocatori presenti e assenti agli eventi pianificati. 2. Nessuna modifica viene apportata ai dati.

UC8 – Suggerisci Formazione

I seguenti Diagrammi di Sequenza di Sistema illustrano le interazioni I/O dell'utente con il sistema TeamManager relative al Caso d'Uso 8 con i vari scenari di successo:



Contratti delle operazioni UC8

I contratti delle operazioni descrivono i principali metodi identificati nei SSD di UC8:

CONTRATTO 1 UC8: Suggerisci Formazione

- **Operazione:** suggerisciFormazione(idGiocatoreSelezionato)
- **Riferimento:** UC8: Suggerisci Formazione
- **Pre-condizione:**
 1. L'allenatore ha effettuato correttamente l'accesso al sistema.
 2. Dal menu principale l'allenatore ha selezionato l'opzione Suggerisci Formazione.
 3. L'allenatore visualizza la lista dei moduli di formazione selezionabili.
 4. Il file contenente i vari giocatori presenti in rosa è accessibile in lettura
- **Post-condizione:**
 1. L'allenatore visualizza correttamente a schermo la Lista della nuova formazione suggerita.
 2. Nessuna modifica viene apportata ai dati.

3.0 Fase di Progettazione

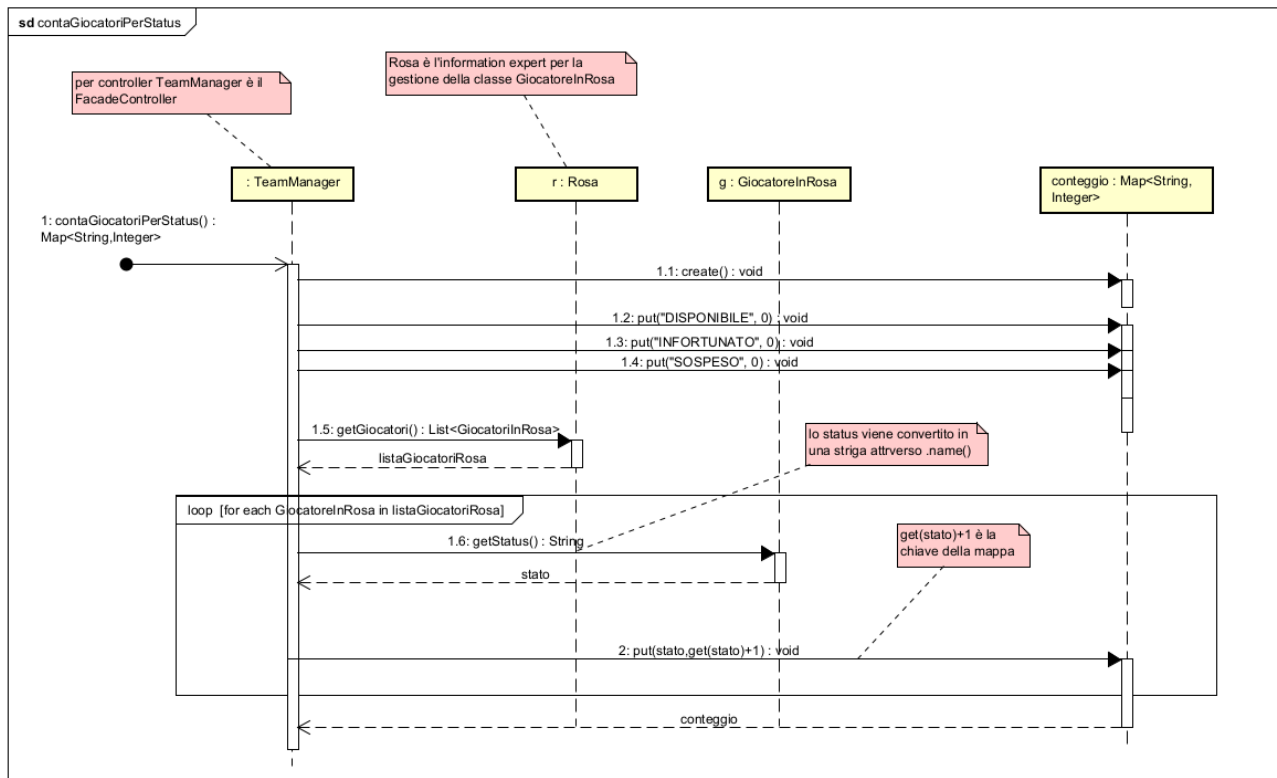
Gli elaborati principali della fase di progettazione comprendono i diagrammi di sequenza, utilizzati per descrivere il comportamento dinamico del sistema durante

l'esecuzione dei casi d'uso selezionati per la seconda iterazione. A supporto di questi, il diagramma delle classi offre una rappresentazione statica del sistema, evidenziando le principali entità coinvolte, i relativi attributi e metodi, nonché le associazioni tra di esse. Di seguito vengono riportati i diagrammi realizzati.

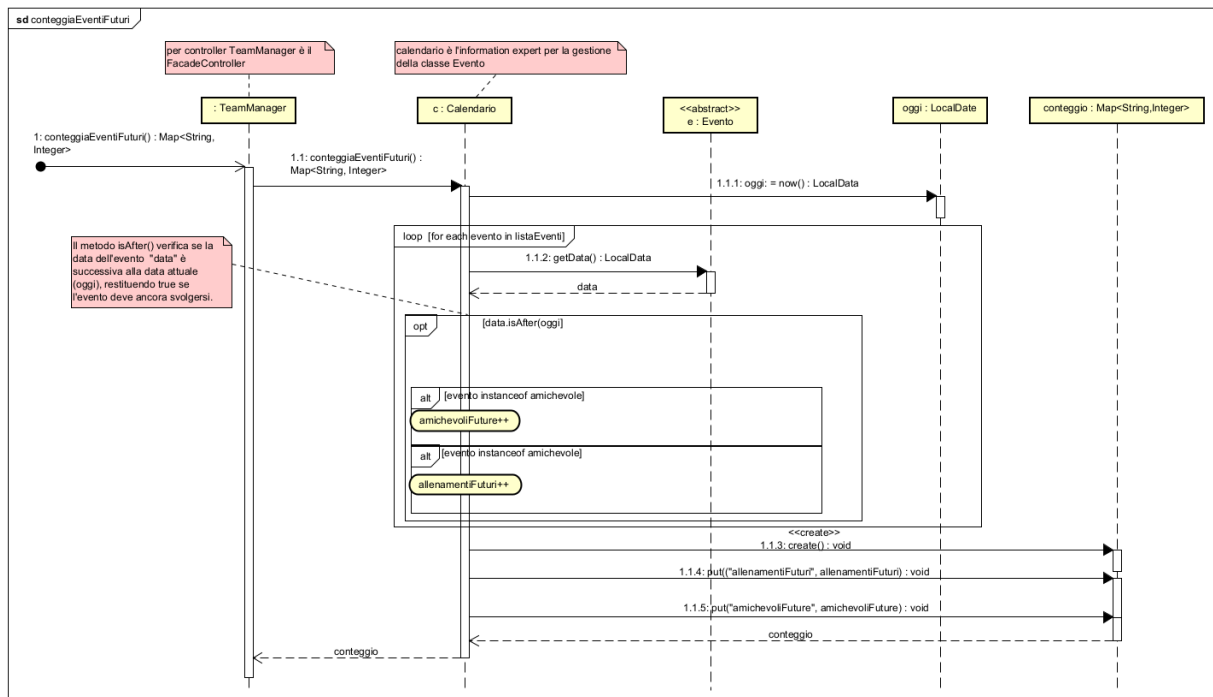
3.1 Diagrammi di Sequenza UC4

La classe TeamManager è stata progettata secondo il pattern GoF Façade Controller, fungendo da punto di accesso unificato alle funzionalità principali del sistema. Nel caso d'uso *UC4 – Andamento Rosa*, TeamManager coordina l'interazione tra l'interfaccia utente e le classi del dominio per fornire all'allenatore una panoramica aggregata delle prestazioni della rosa nel tempo. Il processo inizia con la ricezione della richiesta di visualizzazione dell'andamento. Successivamente, TeamManager si avvale della classe Calendario, che agisce come Information Expert, per accedere a tutti gli eventi (allenamenti e amichevoli) e le disponibilità associate. Per ogni giocatore della rosa, Calendario calcola delle medie a partire da statistiche pertinenti, conteggia il numero di giocatori disponibili, sospesi ecc. le presenze/assenze. tutti i dati vengono aggregati opportunamente. I dati aggregati e calcolati, che mostrano l'andamento collettivo della rosa, vengono restituiti al TeamManager sotto forma di mappe o liste di valori. La logica di visualizzazione è demandata all'interfaccia utente, che interpreta questi dati per mostrare le tendenze di rendimento. Questa funzionalità rappresenta uno strumento strategico di supporto alle decisioni tecniche, utile per monitorare lo stato di forma della squadra, identificare aree di miglioramento e valutare l'efficacia degli allenamenti.

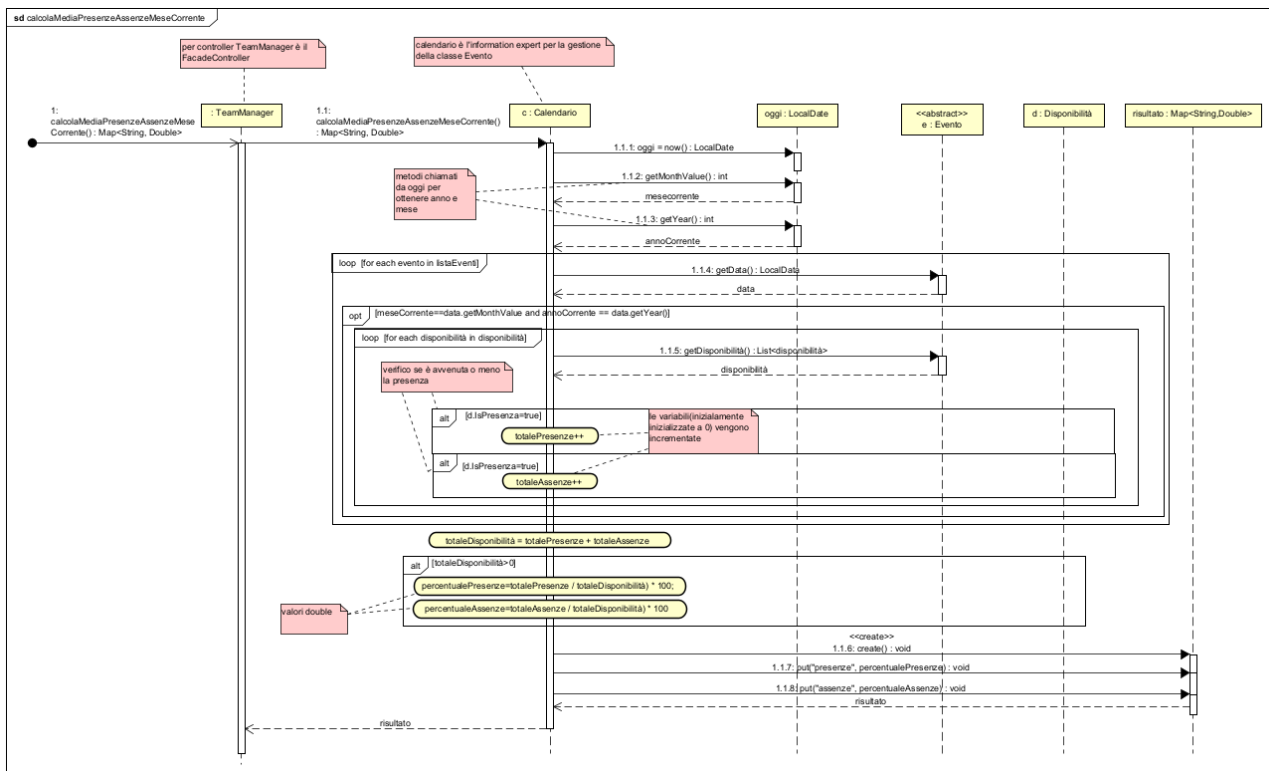
3.1.1 contaGiocatoriPerStatus()



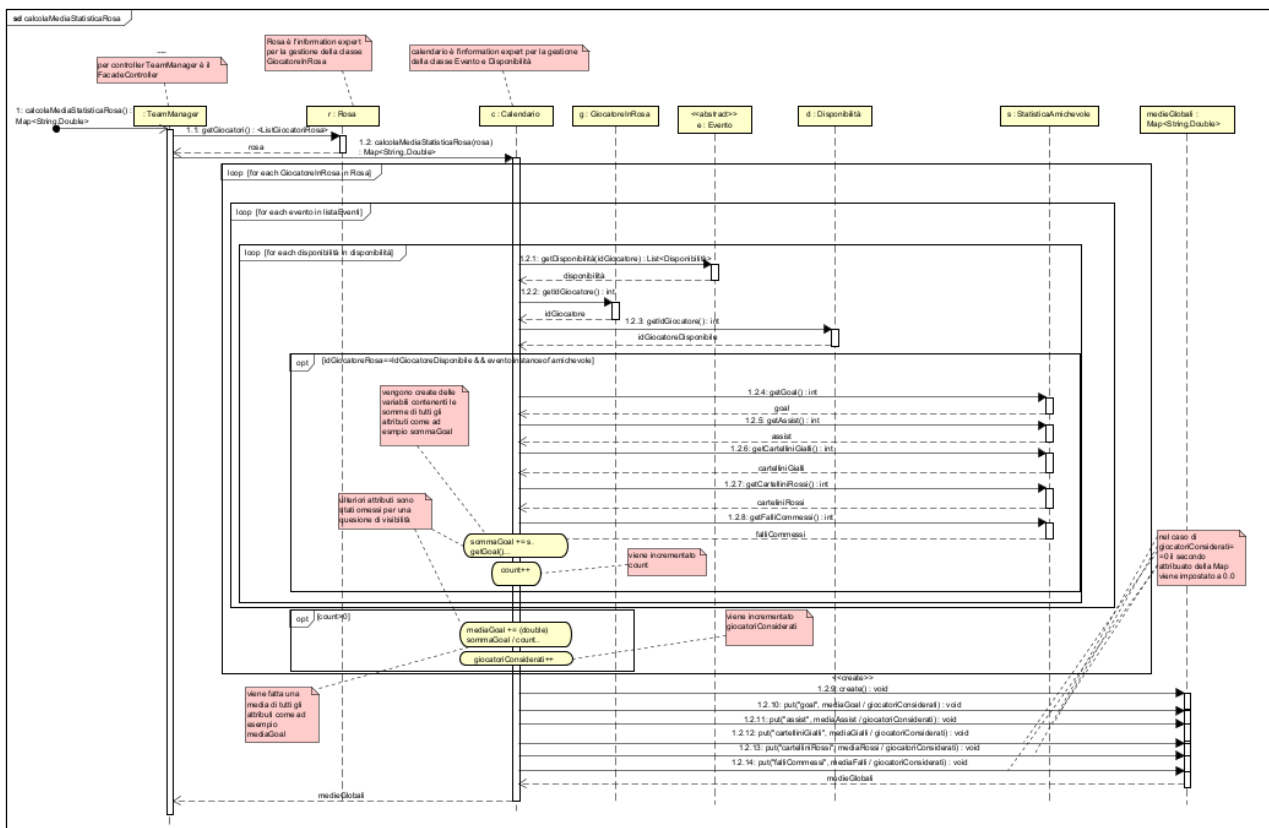
3.1.2 conteggiaEventiFuturi()



3.1.3 CalcolaMediaPresenzeAssenzeMeseCorrente()



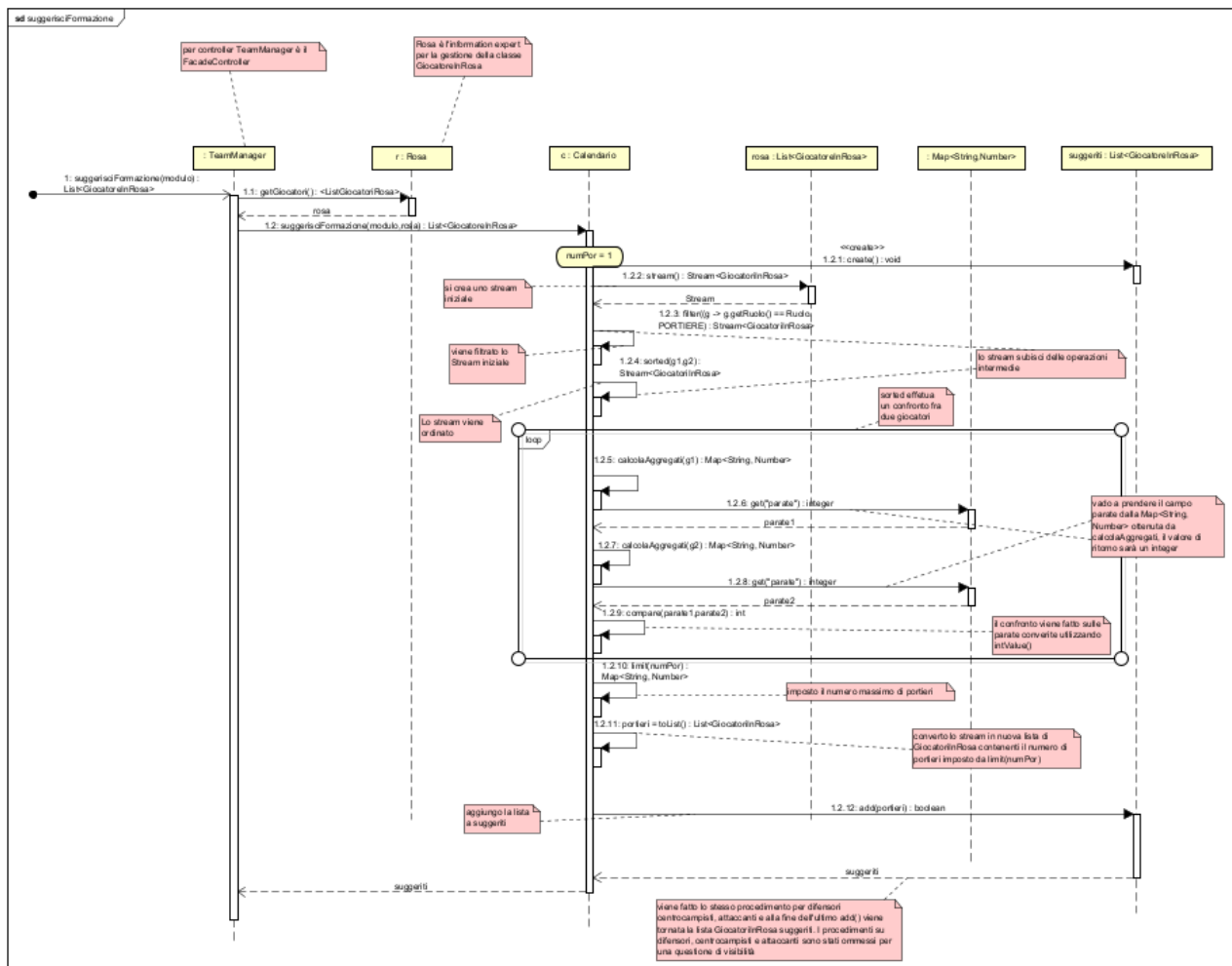
3.1.4 calcolaMediaStatisticaRosa(rosa)



3.2 Diagrammi di Sequenza UC8

La classe TeamManager è stata progettata secondo il pattern GoF Façade Controller, fungendo da punto di accesso unificato alle funzionalità principali del sistema. Nel caso d'uso UC8 – Suggerisci Formazione, TeamManager coordina l'interazione tra l'interfaccia utente e le classi del dominio per proporre all'allenatore una formazione ottimale basata su criteri di performance. Il processo inizia con la selezione del modulo desiderato (es. "4-3-3") da parte dell'allenatore e il recupero della rosa completa di GiocatoreInRosa. Successivamente, TeamManager si avvale della classe Calendario per orchestrare la selezione dei giocatori. Calendario utilizza le Stream API sulla lista dei giocatori della rosa per filtrare i giocatori per ruolo, ordinandoli in base a statistiche chiave (es. "parate" per i portieri, "intercettiRiusciti" per i difensori) precedentemente aggregate tramite la logica di calcolaAggregati, e limitando il numero di giocatori per ruolo base al modulo. La lista dei giocatori per ciascun ruolo viene infine consolidata in un'unica lista suggeriti che rappresenta la formazione completa. Questa lista viene restituita al TeamManager. La logica di visualizzazione della formazione suggerita è demandata all'interfaccia utente, che presenta i giocatori scelti all'allenatore. Questa funzionalità rappresenta uno strumento strategico di supporto alle decisioni tecniche, utile per assistere l'allenatore nelle scelte pre-partita e ottimizzare le prestazioni della squadra.

3.2.1 suggerisciFormazione



4.0 Diagramma delle Classi

Il diagramma delle classi rappresenta la struttura statica del sistema TeamManager, evidenziando le principali entità coinvolte nella seconda iterazione e le relazioni tra esse. La classe TeamManager, progettata come Singleton secondo i pattern GoF, funge da Facade Controller, centralizzando il coordinamento delle operazioni tra l'interfaccia utente e il modello di dominio. Per semplicità di rappresentazione, nei diagrammi UML non sono riportati esplicitamente i metodi getter e setter, i quali saranno comunque implementati nel codice per garantire l'accesso controllato agli attributi delle classi.

5.0 Testing

Durante la terza iterazione sono stati implementati e verificati nuovi metodi di test per coprire le funzionalità aggiuntive previste dai casi d'uso di questa iterazione. Di seguito si riportano i nuovi test definiti:

- **testSuggerisciFormazione:** Questo test verifica in modo completo la funzionalità di suggerimento della formazione. Simula la creazione di una di serie di giocatoriInRosa, distribuiti in tutti i ruoli, e ne definisce i relativi attributi. Successivamente, crea degli eventi di tipo allenamento e di tipo amichevole, aggiungendoli a un calendario. Per ogni amichevole, genera statistiche fittizie e casuali (come minuti giocati, goal, ecc) per ciascun giocatore, associandole a oggetti di disponibilità che indicano la presenza. Infine, invoca il metodo suggerisciFormazione("4-3-3", rosa) e verifica che la formazione restituita contenga esattamente 11 giocatori, confermando che il sistema seleziona e filtra correttamente i giocatori secondo il modulo specificato.
- **testConteggiaEventiFuturi:** Questo test verifica la capacità del sistema di contare correttamente gli eventi futuri. Inizia svuotando la lista degli eventi del calendario per garantire un ambiente di test pulito. Successivamente, aggiunge al calendario due eventi (un allenamento e un'amichevole) con date passate e tre eventi (due allenamenti e un'amichevole) con date future. Invoca poi il metodo conteggiaEventiFuturi() e controlla che la mappa risultante indichi esattamente 2 "allenamentiFuturi" e 1 "amichevoliFuture", confermando l'accuratezza del conteggio degli eventi futuri per tipologia.
- **testCalcolaMediaPresenzeAssenzeMeseCorrente:** Questo test verifica la corretta implementazione del calcolo della media di presenze e assenze per il mese corrente. Il test svuota la lista degli eventi e crea un allenamento per la data odierna. Associa a questo evento due disponibilità: una che registra la presenza di un giocatore e una che registra l'assenza di un altro giocatore. Dopo aver aggiunto l'evento al calendario, invoca il metodo

calcolaMediaPresenzeAssenzeMeseCorrente() e verifica che le medie restituite siano entrambe del 50.0% (con una tolleranza di 0.01), a riprova che il calcolo delle percentuali di presenza e assenza nel mese corrente è esatto.

- **testCalcolaMediaStatisticaRosa:** Questo test verifica l'accuratezza del calcolo delle statistiche medie per l'intera rosa di giocatori. Il test pulisce gli eventi precedenti e inizializza una rosa con due giocatori (g1 e g2) e crea una lista fissa e immutabile (non puoi aggiungere o rimuovere elementi dopo la creazione) che contiene esattamente i due giocatori g1 e g2 definiti per il test. Vengono create due amichevoli e aggiunte al calendario. Vengono poi definite statistiche specifiche e non casuali per g1 (due istanze di statistiche per due amichevoli diverse) e per g2 (una istanza di statistiche per una amichevole), che vengono associate alle rispettive disponibilità e agli eventi. Infine, il test invoca il metodo calcolaMediaStatisticaRosa(rosa) e confronta le medie calcolate per "goal", "assist", "cartelliniGialli", "cartelliniRossi" e "falliCommessi" con valori predefiniti, assicurando la precisione delle aggregazioni statistiche sull'intera rosa sempre con una tolleranza di 0.01.