UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO DI INGEGNERIA

CORSO DI LAUREA TRIENNALE IN

INGEGNERIA INFORMATICA

System Design Document

Sportify™

INGEGNERIA DEL SOFTWARE A.A. 2023/2024

A cura di:

Francesco Fausto Franchina

Marco Marino

Giovanni Dolcemascolo

Diego Lo Voi

SOMMARIO

1. Obiettivo del sistema

Il sistema progettato è destinato al supporto di una catena di campi sportivi avente più sedi in una sola città.

Il software consente a più tipologie di utenti di gestire le prenotazioni dei campi da gioco messi a disposizione dalle sedi della catena, mirando a massimizzare il numero di partecipanti di ogni prenotazione. A tal fine il software prevede dei sistemi integrati di notifiche e di inviti che incentivano l'occupazione dei campi e agevolano il riempimento delle prenotazioni già esistenti.

Il sistema punta a semplificare l'amministrazione dei campi da gioco per i gestori delle sedi della società sportiva, fornendo loro un'ampia gamma di operazioni effettuabili, fra cui anche l'apertura di tornei per specifiche discipline sportive e la creazione di promozioni volte a fidelizzare le differenti fasce di giocatori (abbonati e non).

Per limitare i danni arrecati dai giocatori che tendono ad abbandonare con frequenza le prenotazioni aperte, il sistema implementa un sistema di blacklist atto a tracciare i comportamenti scorretti dei giocatori in modo da filtrare il bacino di destinatari delle notifiche volte a segnalare l'apertura di prenotazioni o la presenza di posti in prenotazioni già aperte.

Infine, al fine di facilitare le comunicazioni tra gli utenti, il software offre un apposito sistema di messaggistica e consente ai gestori di inviare, tramite notifiche, avvisi generali destinati a tutti gli utenti del sistema.

2. Architettura software corrente

La catena di campi sportivi non dispone attualmente di alcun sistema informatizzato. Il sistema proposto mira quindi a semplificare e ad automatizzare, ove possibile, le operazioni svolte manualmente dallo staff della società sportiva, e a rendere più accessibile per i clienti sia la prenotazione dei campi che l'adesione ai tornei.

3. Obiettivi di progettazione

3.1. Sicurezza

I dati sensibili vanno gestiti in modo da garantire la loro sicurezza. A tal fine è necessario prevedere meccanismi che consentano un accesso regolamentato ai dati, impedendo un accesso diretto al DBMS da parte degli utenti. Oltretutto è necessario effettuare una divisione delle funzioni del software in modo da evitare che utenti malintenzionati possano accedere alla logica di business del sistema.

3.2. Scalabilità

Il sistema deve essere scalabile in modo da poter essere opportunamente ampliato in funzione delle necessità del cliente. Infatti, aumento dell'utenza del sistema potrebbe generare un carico sempre maggiore e quindi difficile da gestire.

3.3. Prestazioni

Il sistema deve poter essere utilizzato da utenti che utilizzano configurazioni hardware anche molto diverse tra loro. Per tale ragione, il software deve essere realizzato mirando ad una implementazione efficiente, dal momento che bisogna tener conto degli utenti che utilizzano dispositivi poco performanti.

4. Architettura software proposta

4.1. Panoramica

Per la realizzazione del software è stata scelta un'architettura di tipo client-server. I principali punti di forza di tale architettura sono:

- Sicurezza fornita: la gestione dei dati sensibili è affidata al server, che opera da unità centrale che regola l'accesso ai dati. Al fine di irrobustire il sistema e ridurre le vulnerabilità a eventuali attacchi informatici, è necessario inoltre adottare opportune misure di sicurezza.

- Prestazioni: il carico generato sul sistema è distribuito tra i client e il server, riducendo complessivamente il quantitativo di operazioni che deve effettuare ciascun calcolatore. Le operazioni più complesse vengono gestite dal server, che dispone di hardware dedicato in grado di garantire prestazioni elevate.

- Espandibilità: la divisione delle funzioni da eseguire tra quelle dei client e quelle del server, consente un'agevole implementazione di ulteriori funzioni del software non originariamente previste.

- Scalabilità e modularità: l'architettura si presta all'integrazione di hardware aggiuntivo (server), consentendo quindi di ampliare le risorse e le capacità del sistema. Ciò, abbinato all'adozione di tecniche di bilanciamento del carico atte a distribuire tra i server le richieste ricevute dai client, garantisce il miglioramento dell'efficienza del sistema e l'erogazione continuativa dei servizi offerti anche in caso di guasti e problematiche legate ad un singolo server. L'utilizzo di più server consentirebbe inoltre di operare, qualora sia necessario effettuare lavori di manutenzione, su una parte ristretta di essi senza compromettere il funzionamento dell'intero sistema.

- Consistenza dei dati salvati: con l'adozione di tecniche di controllo della concorrenza e di meccanismi di backup e ripristino, un server centralizzato è in grado di garantire la coerenza e l'integrità dei dati gestiti.

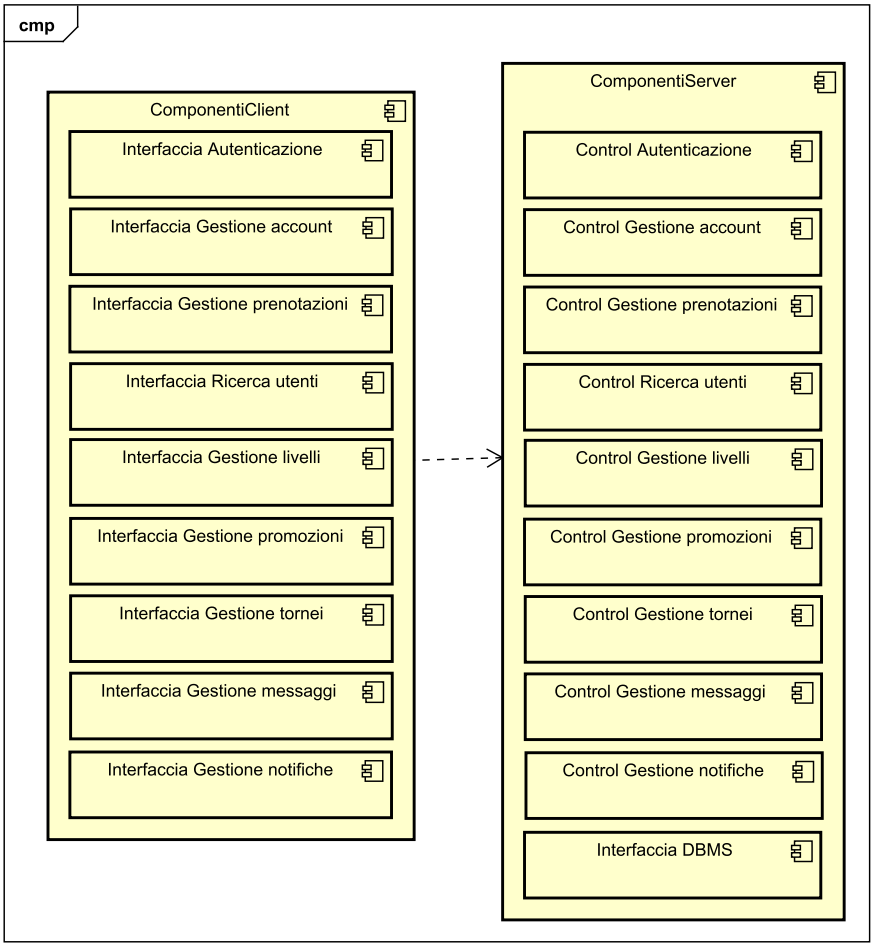
È quindi evidente che le soluzioni di tipo client-server siano tra le più adatte per lo sviluppo di applicazioni di rete. Infine, con l'impiego di connessioni TCP fra i client e il server, è possibile garantire un ulteriore livello di sicurezza per i dati scambiati tra gli host; infatti, è possibile adottare il TLS, ossia un protocollo crittografico che cripta i dati in luogo del loro invio nella rete e li decripta una volta giunti a destinazione.

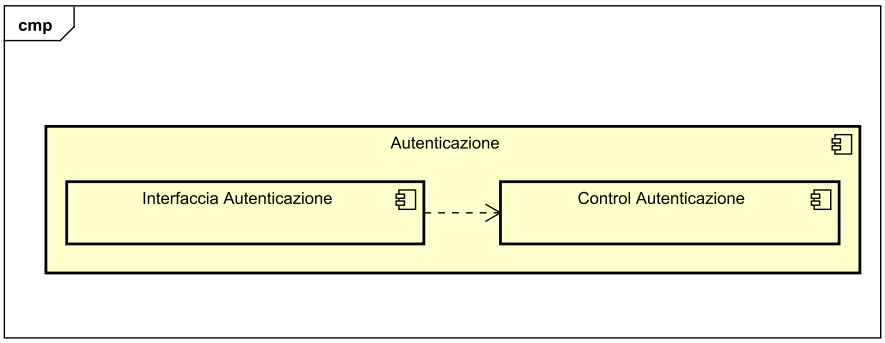
Infine, si aggiunge che, con l'architettura scelta, le interfacce utente risiedono sullo stesso nodo utente, mentre la logica di business e l'interfaccia al DBMS risiede su un nodo server separato in modo da impedire un accesso diretto al database da parte degli utenti.

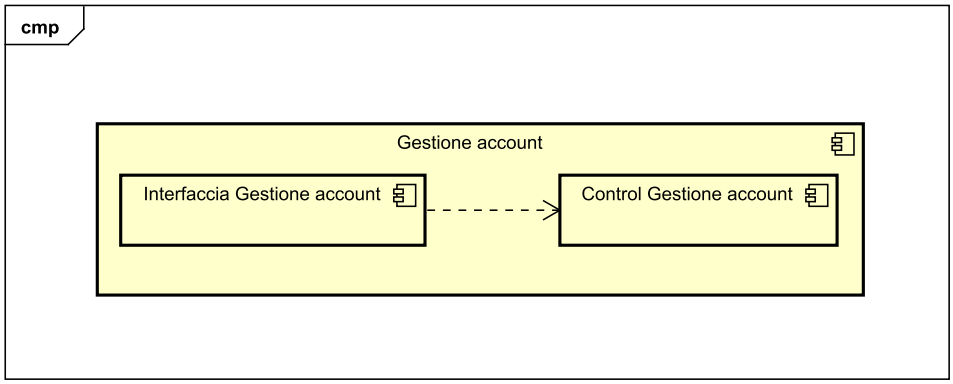
4.2. Requisiti minimi per l'utilizzo del software

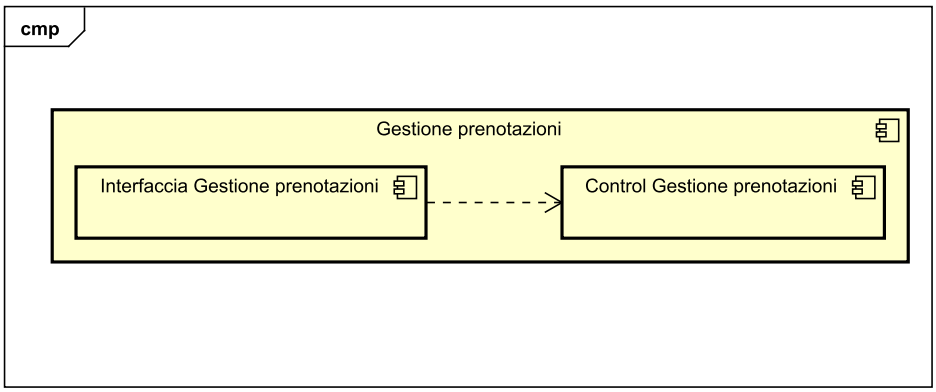
Per operare correttamente, il sistema ha bisogno di una connessione ad internet stabile, in modo da consentire ai client e al server di comunicare tra loro e per poter inviare e-mail agli utenti.

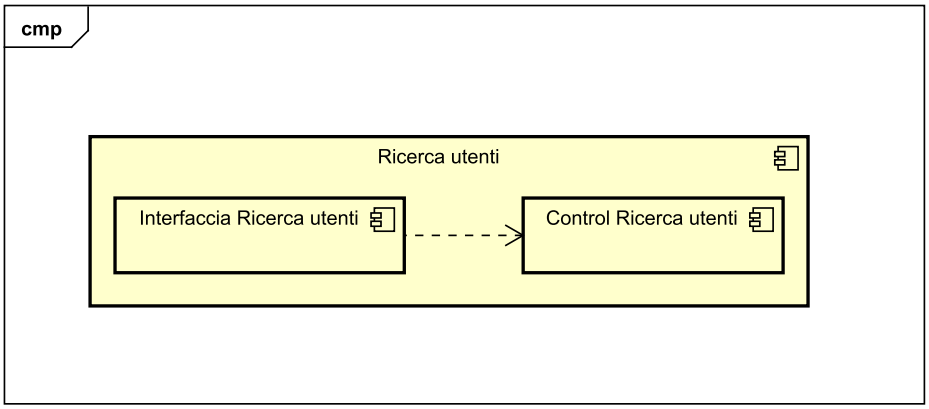
4.3. Scomposizione in sottosistemi

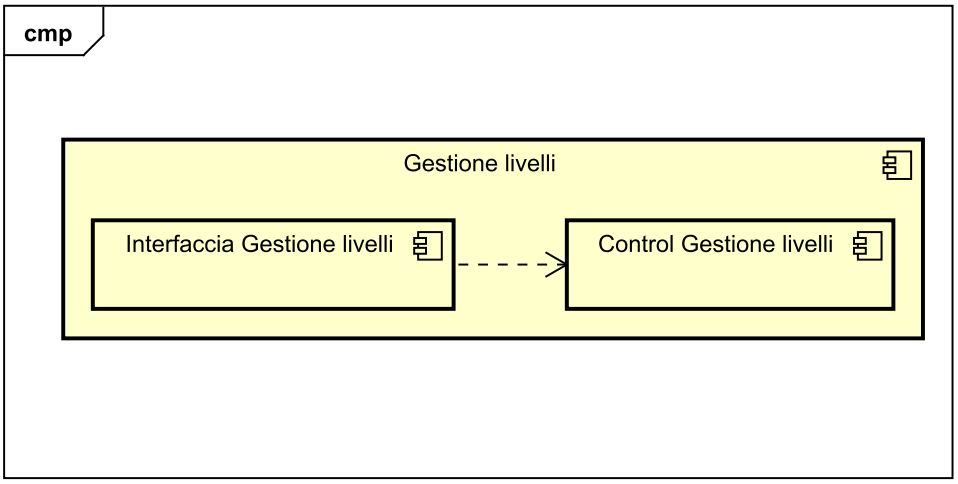


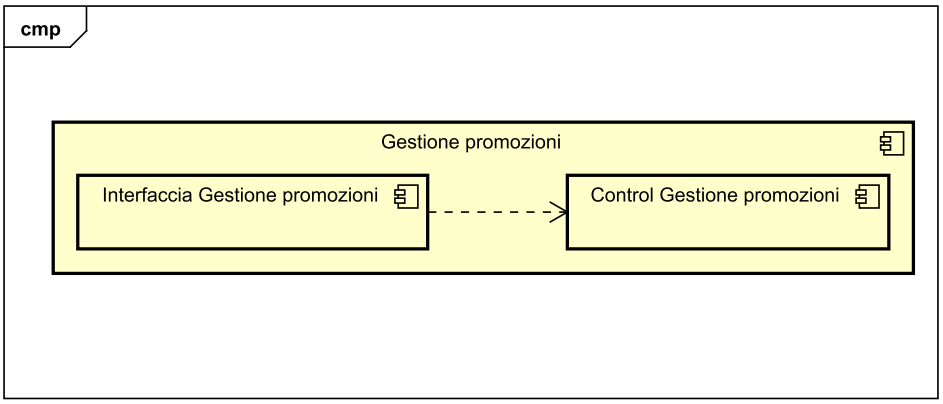


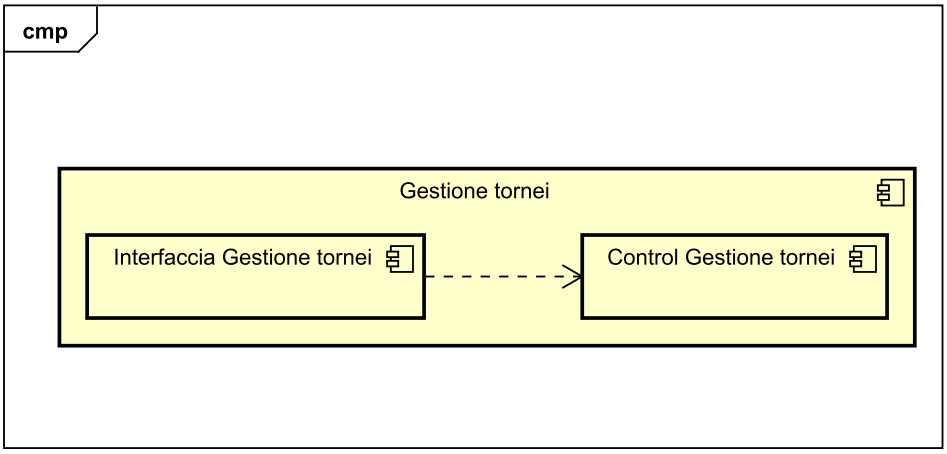

cmp
Gestione account
Interfaccia Gestione account
Control Gestione account

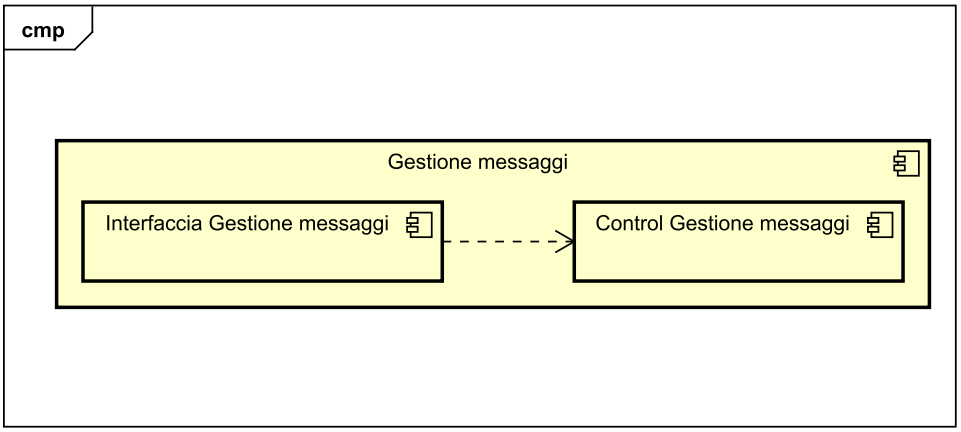

cmp
Gestione prenotazioni
Interfaccia Gestione prenotazioni
Control Gestione prenotazioni

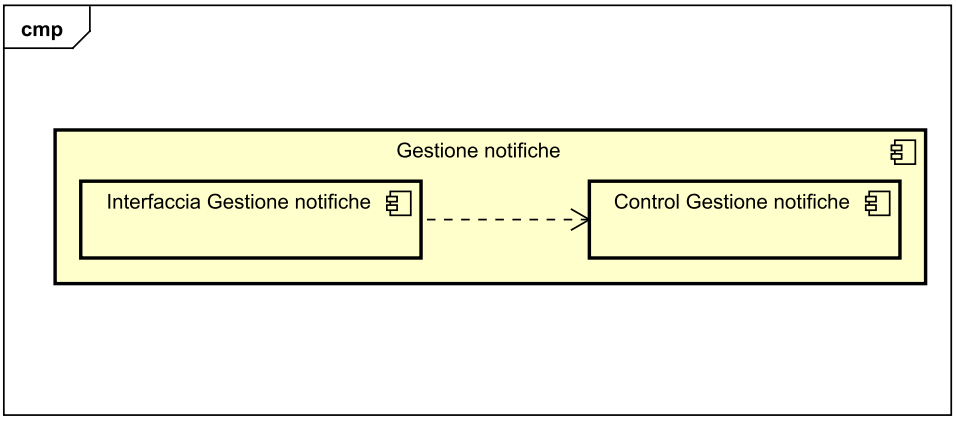

cmp
Ricerca utenti
Interfaccia Ricerca utenti
Control Ricerca utenti


cmp
Gestione livelli
Interfaccia Gestione livelli
Control Gestione livelli


cmp
Gestione promozioni
Interfaccia Gestione promozioni
Control Gestione promozioni


cmp
Gestione tornei
Interfaccia Gestione tornei
Control Gestione tornei





Interfaccia Autenticazione: gestisce le interfacce grafiche necessarie per l'accesso degli utenti al sistema. Fornisce dunque le schermate necessarie per effettuare il login, la registrazione e il recupero dell'account.

Control Autenticazione: gestisce la logica applicativa relativa all'accesso degli utenti. Implementa le funzionalità per la registrazione di un nuovo utente, per il login e per il recupero dell'account.

Interfaccia Gestione account: gestisce le interfacce grafiche necessarie per consentire agli utenti di gestire i propri dati personali. Fornisce dunque le schermate per la visualizzazione dei dati personali, la modifica di e-mail e password, e le schermate per la disconnessione dell'account e la cancellazione dell'account.

Control Gestione account: gestisce la logica applicativa relativa alla gestione dei dati personali degli utenti. Implementa quindi le funzionalità necessarie per la visualizzazione dei propri dati, per la modifica di e-mail e password, per la disconnessione dell'account e per la cancellazione dell'account.

Interfaccia Gestione prenotazioni: gestisce le interfacce grafiche per consentire agli utenti di prenotare campi da gioco ed effettuare operazioni sulle prenotazioni già esistenti. Fornisce

dunque le schermate per visualizzare prenotazioni aperte o chiuse, per la gestione delle prenotazioni aperte dell'utente e per la gestione degli inviti ricevuti dai giocatori.

Control Gestione prenotazioni: gestisce la logica applicativa per l'amministrazione delle prenotazioni degli utenti. Implementa quindi le funzionalità per la creazione e la gestione delle prenotazioni, per la ricerca e la visualizzazione di prenotazioni aperte o chiuse e per la gestione degli inviti ricevuti.

Interfaccia Gestione tornei: gestisce le interfacce grafiche per consentire agli utenti di creare tornei, iscriversi ai tornei aperti, ed effettuare operazioni sui tornei esistenti. Fornisce dunque le schermate per visualizzare i tornei attivi e passati e per gestire i tornei attivi.

Control Gestione tornei: gestisce la logica applicativa per l'amministrazione dei tornei. Implementa quindi le funzionalità per la creazione e la gestione dei tornei, per la visualizzazione dei tornei attivi e passati, per iscriversi ai tornei attivi.

Interfaccia Gestione promozioni: gestisce le interfacce grafiche per consentire la gestione delle promozioni. Fornisce dunque le schermate che permettono ai gestori di creare, visualizzare e applicare promozioni, e che permettono ai giocatori di visualizzare le promozioni valide.

Control Gestione promozioni: gestisce la logica applicativa necessaria per la gestione delle promozioni. Implementa le funzionalità per permettere agli utenti di creare, visualizzare e applicare promozioni.

Interfaccia Gestione notifiche: gestisce le interfacce grafiche per consentire la gestione delle notifiche. Fornisce dunque una schermata per la visualizzazione delle notifiche ricevute, nonché dei pop-up di supporto all'interfaccia.

Control Gestione notifiche: gestisce la logica applicativa necessaria per la gestione delle notifiche. Implementa quindi le funzionalità per la visualizzazione e la gestione delle notifiche ricevute.

Interfaccia Ricerca utenti: gestisce le interfacce grafiche per la ricerca degli utenti. Fornisce le schermate per la ricerca degli utenti, nonché dei pop-up di supporto all'interfaccia.

Control Ricerca utenti: gestisce la logica applicativa necessaria per la ricerca degli utenti. Implementa quindi le funzionalità per la ricerca degli utenti e la cancellazione degli account dei giocatori.

Interfaccia Gestione messaggi: gestisce le interfacce grafiche per la gestione dei messaggi. Fornisce dunque le schermate per la ricerca e la visualizzazione delle chat con gli altri utenti.

Control Gestione messaggi: gestisce la logica applicativa necessaria per la gestione dei messaggi. Implementa quindi le funzionalità per la ricerca e la visualizzazione delle chat con gli altri utenti.

Interfaccia Gestione livelli: gestisce le interfacce grafiche per la gestione dei livelli dei giocatori. Fornisce dunque le schermate per l'assegnazione dei livelli ai giocatori per gli sport consentiti.

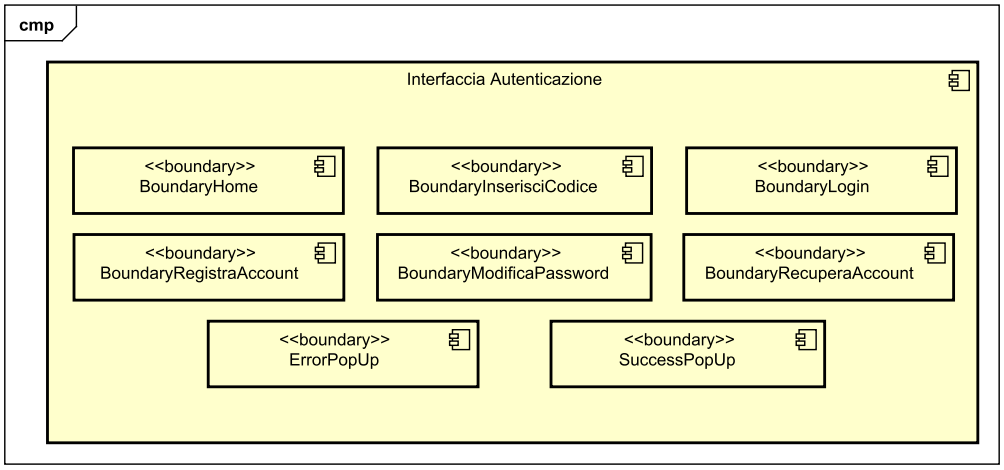
Control Gestione livelli: gestisce la logica applicativa necessaria per la gestione dei livelli dei giocatori. Implementa quindi le funzionalità per l'assegnazione dei livelli ai giocatori per gli sport consentiti.

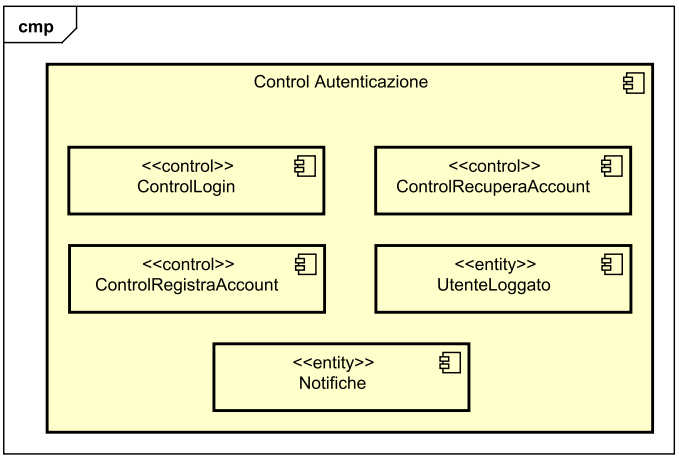
Interfaccia DBMS: gestisce la comunicazione tra il sottosistema server e il DBMS. Fornisce tutte le funzionalità necessarie per l'interazione con il DBMS mediante operazioni di inserimento, aggiornamento e cancellazione di dati memorizzati.

4.3.1. Mappatura degli oggetti nei sottosistemi

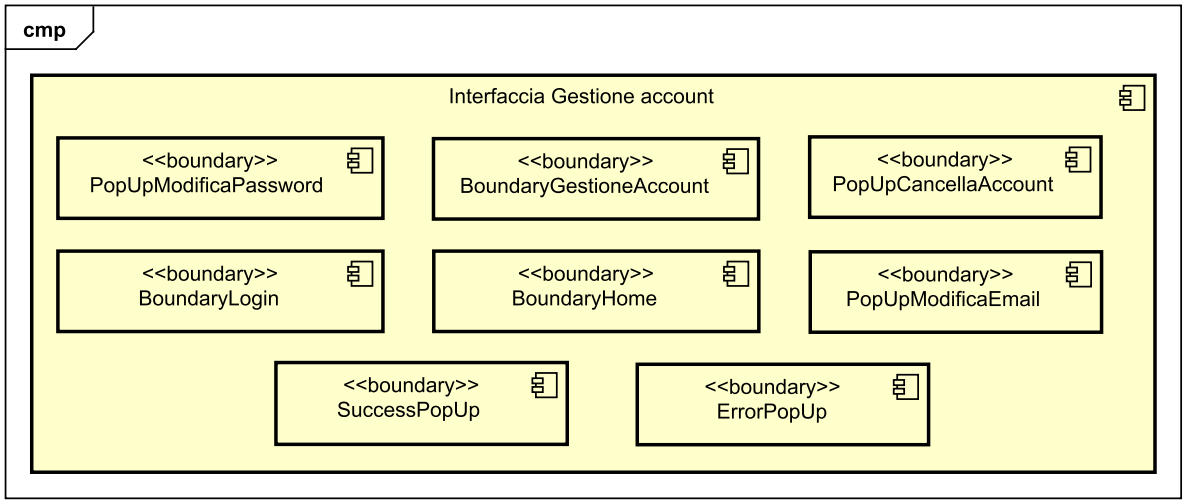
Di seguito sono riportate le componenti dei sottosistemi.

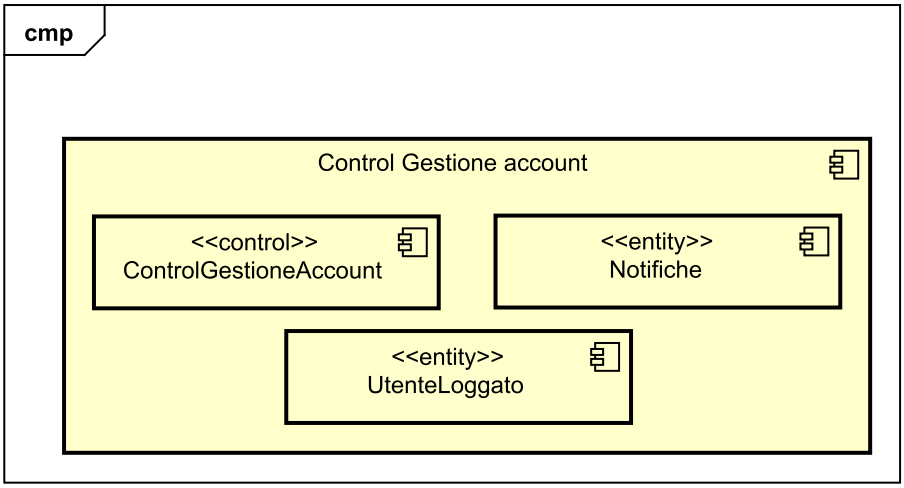
4.3.1.1. Autenticazione



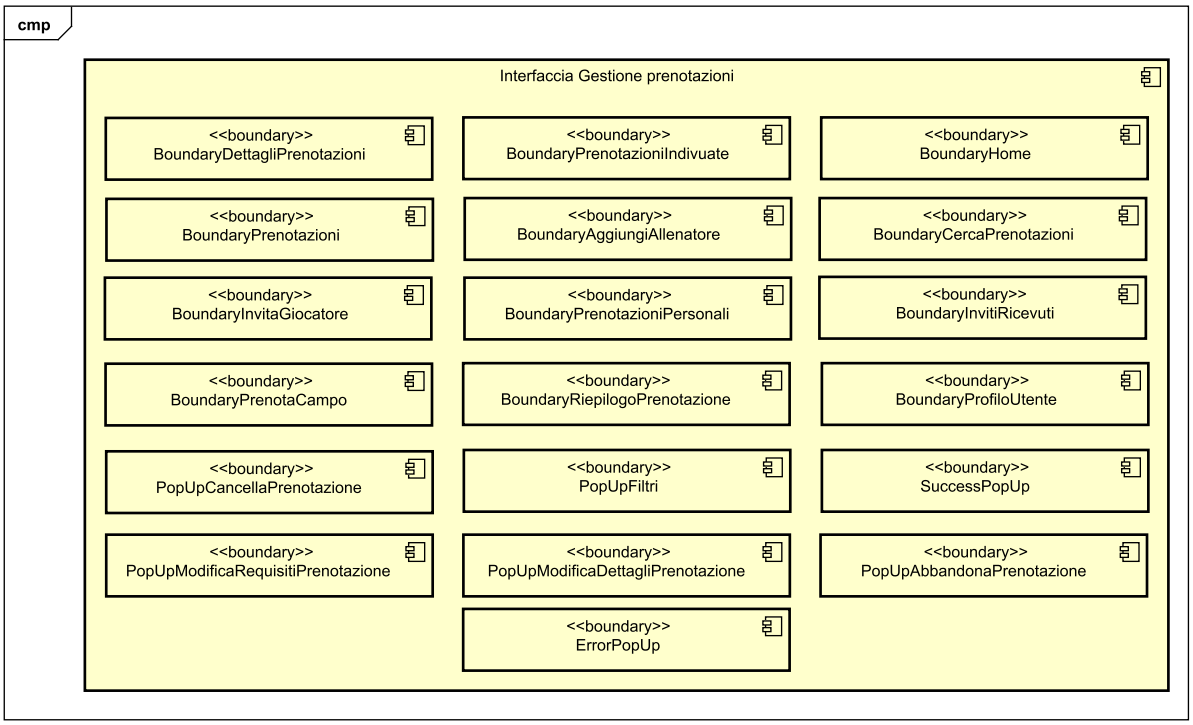


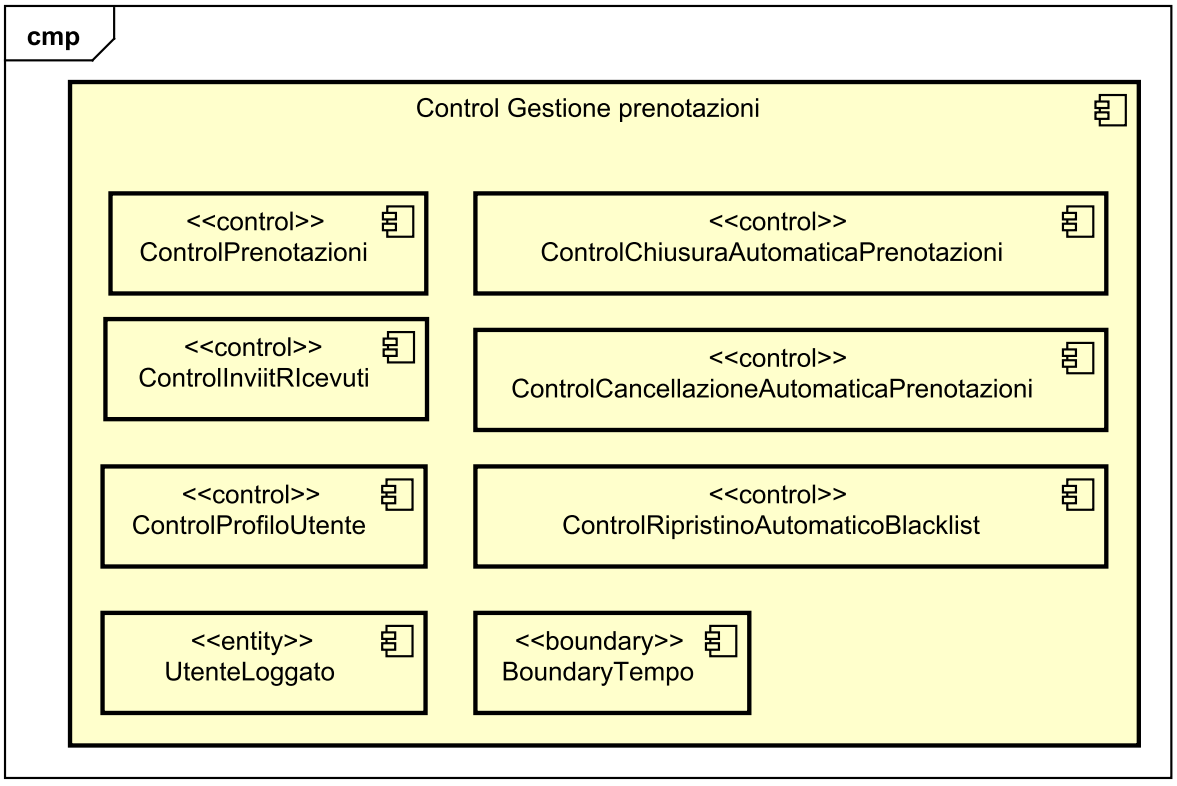
4.3.1.2. Gestione account



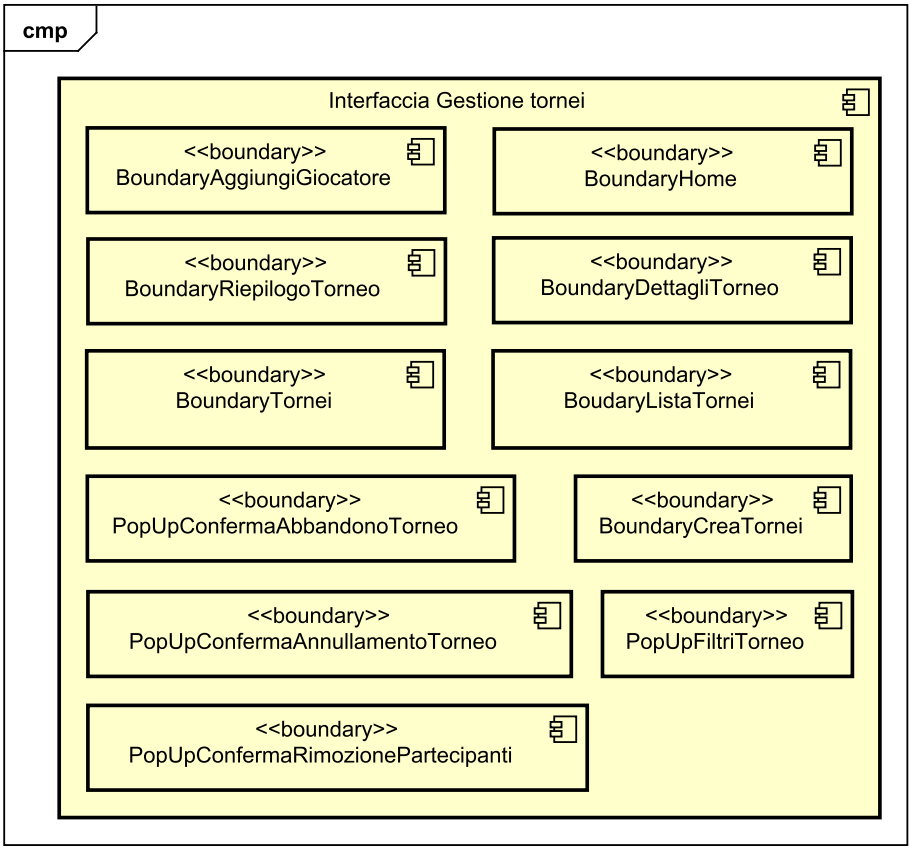


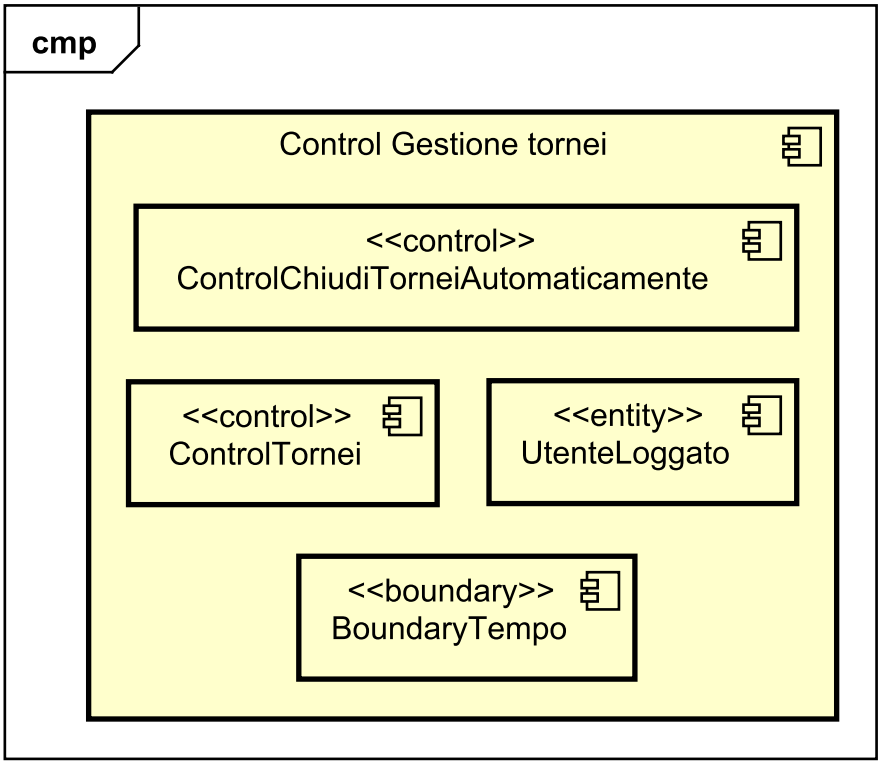
4.3.1.3. Gestione prenotazioni



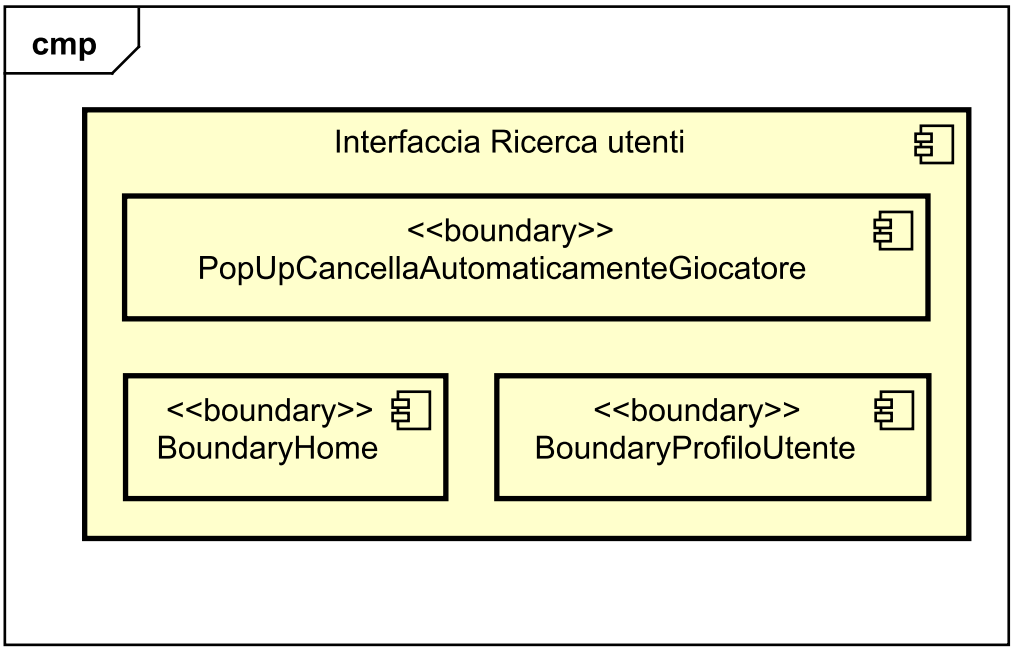


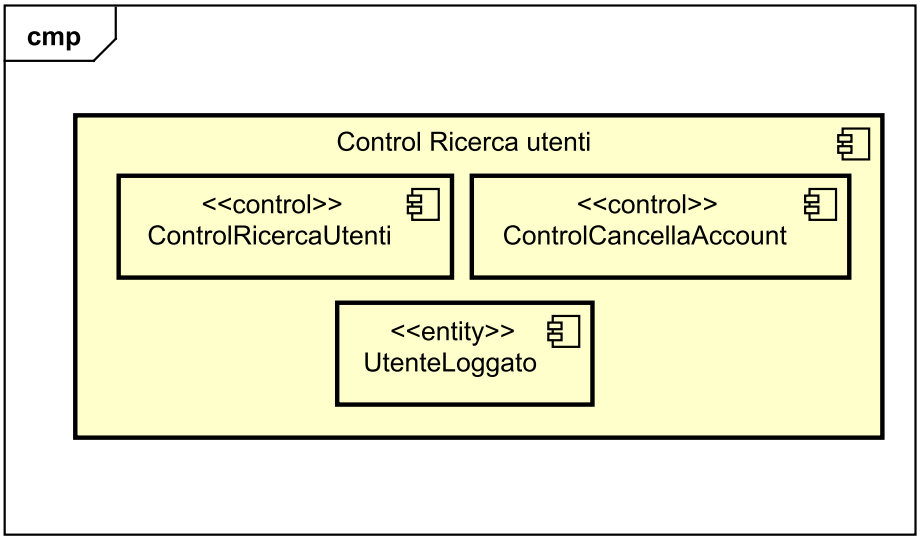
4.3.1.4. Gestione tornei



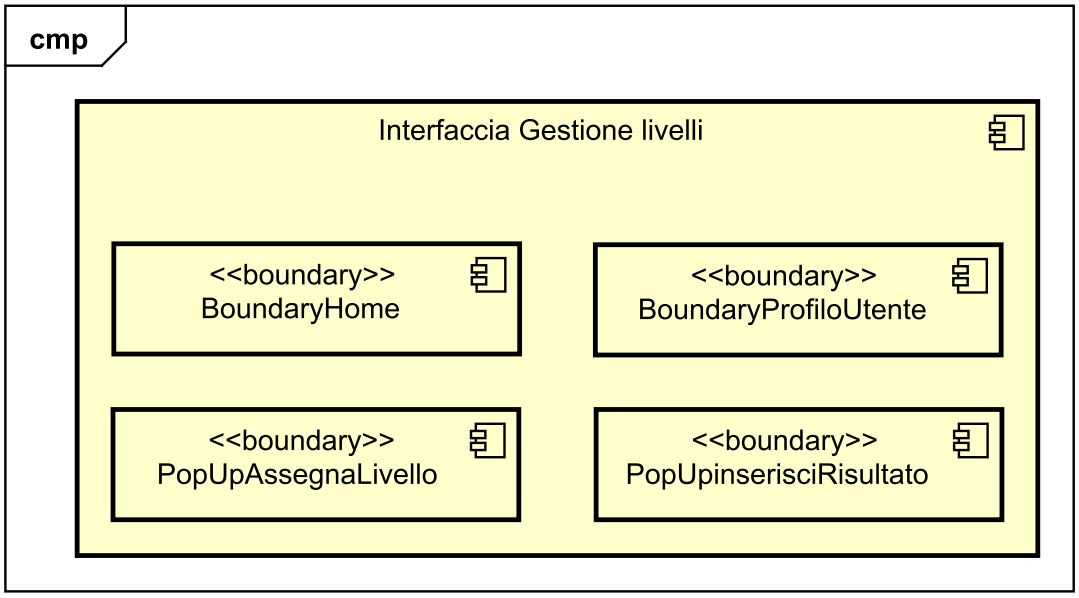


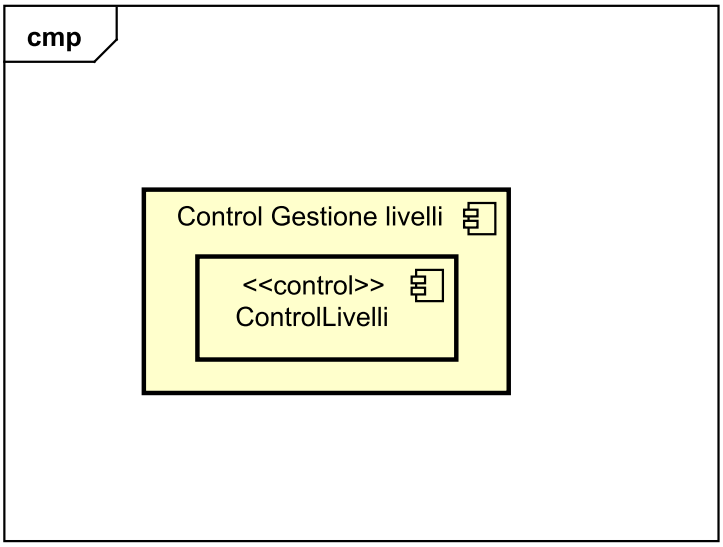
4.3.1.5. Ricerca utenti



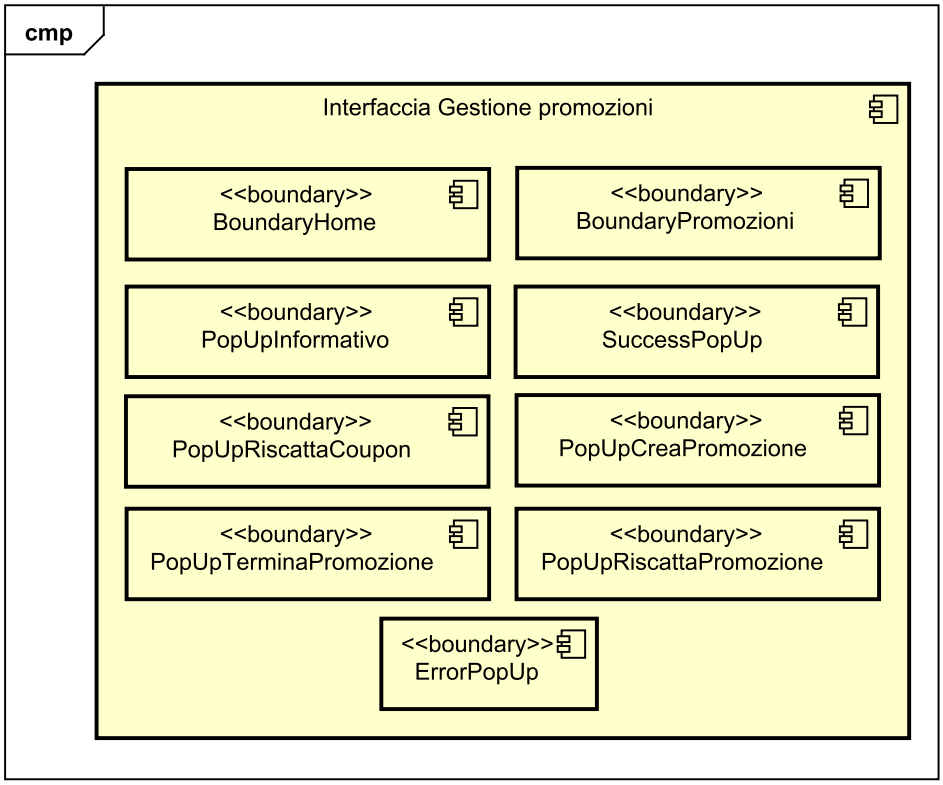


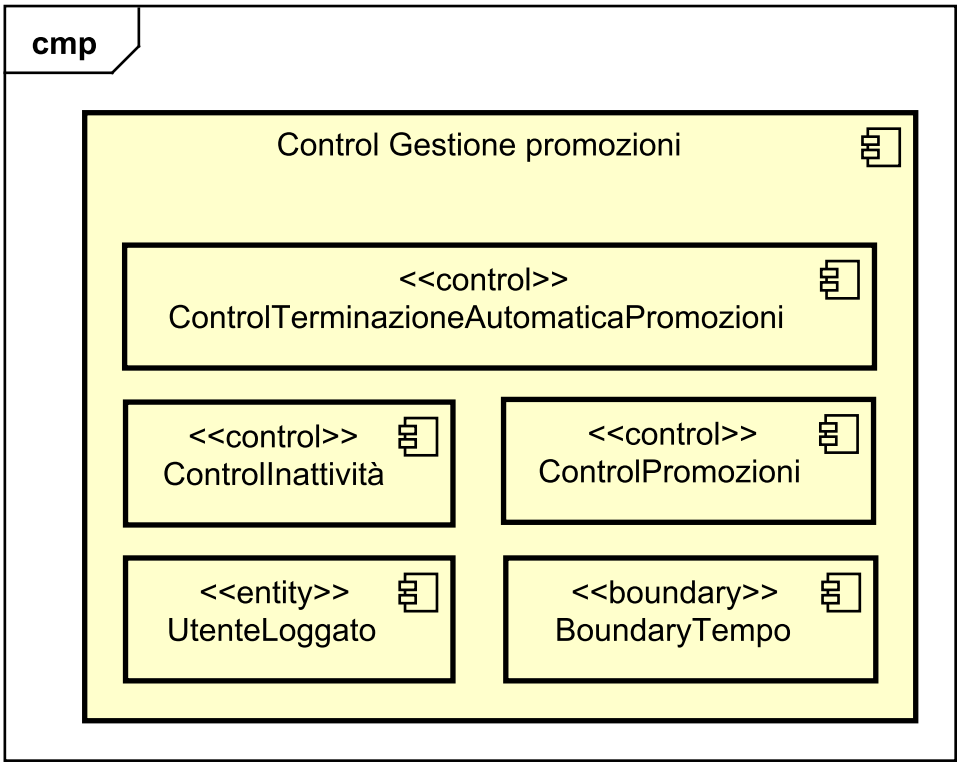
4.3.1.6. Gestione livelli



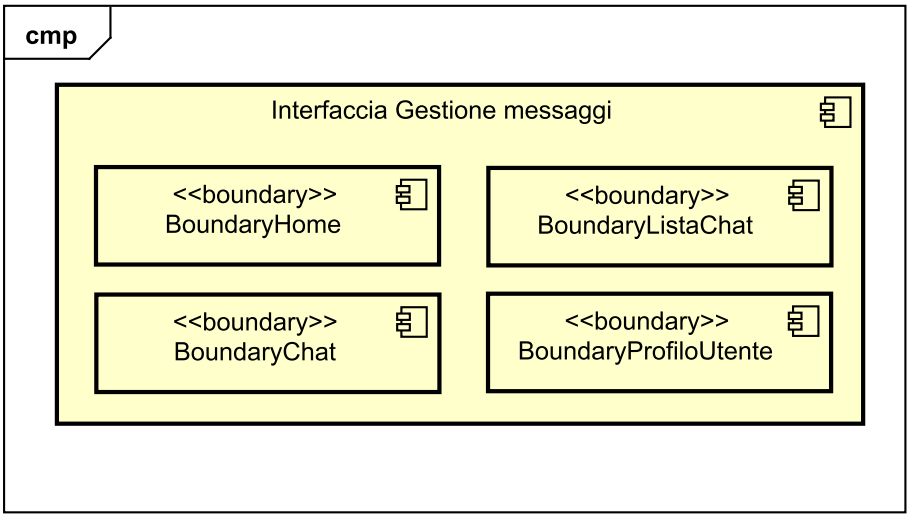


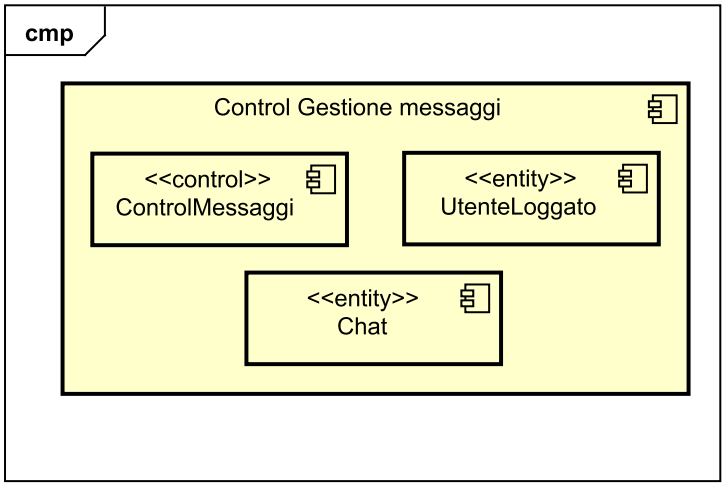
4.3.1.7. Gestione Promozioni



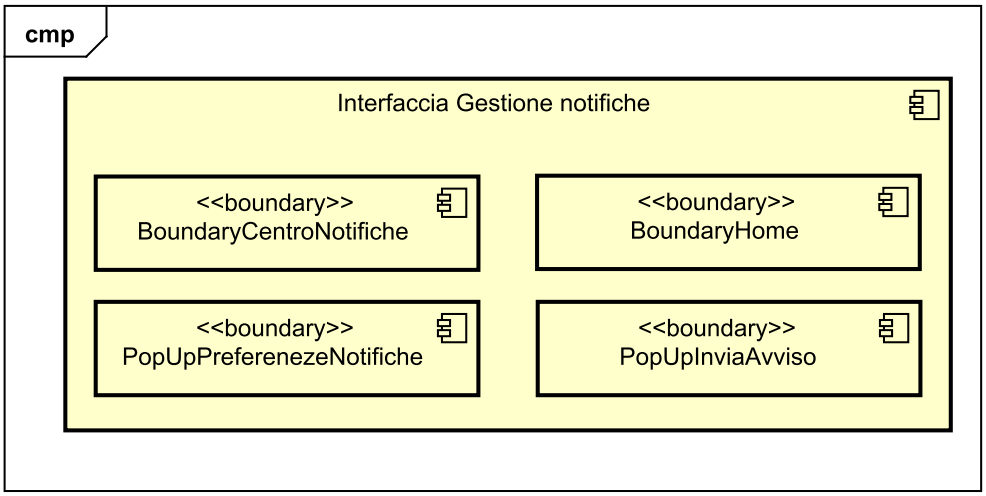


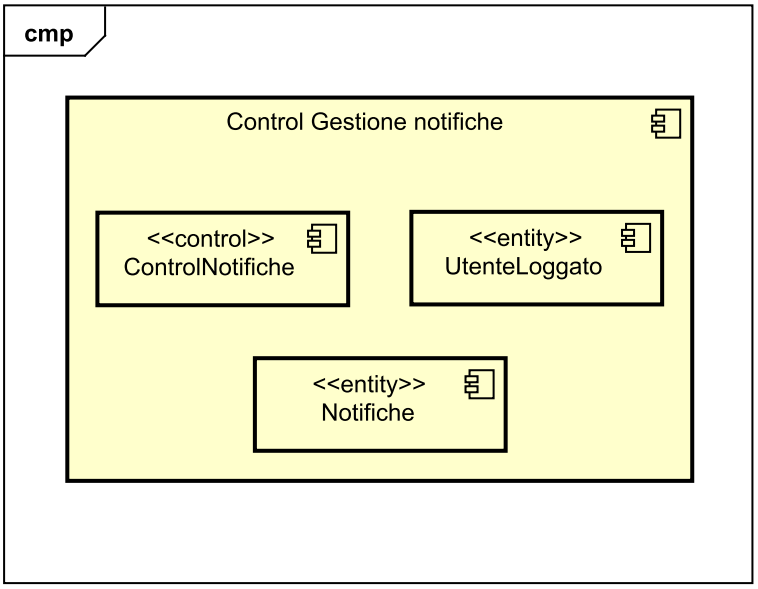
4.3.1.8. Gestione messaggi



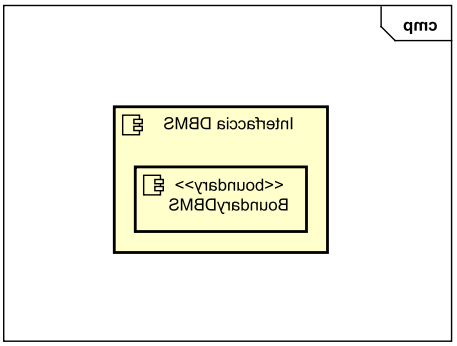


4.3.1.9. Gestione notifiche





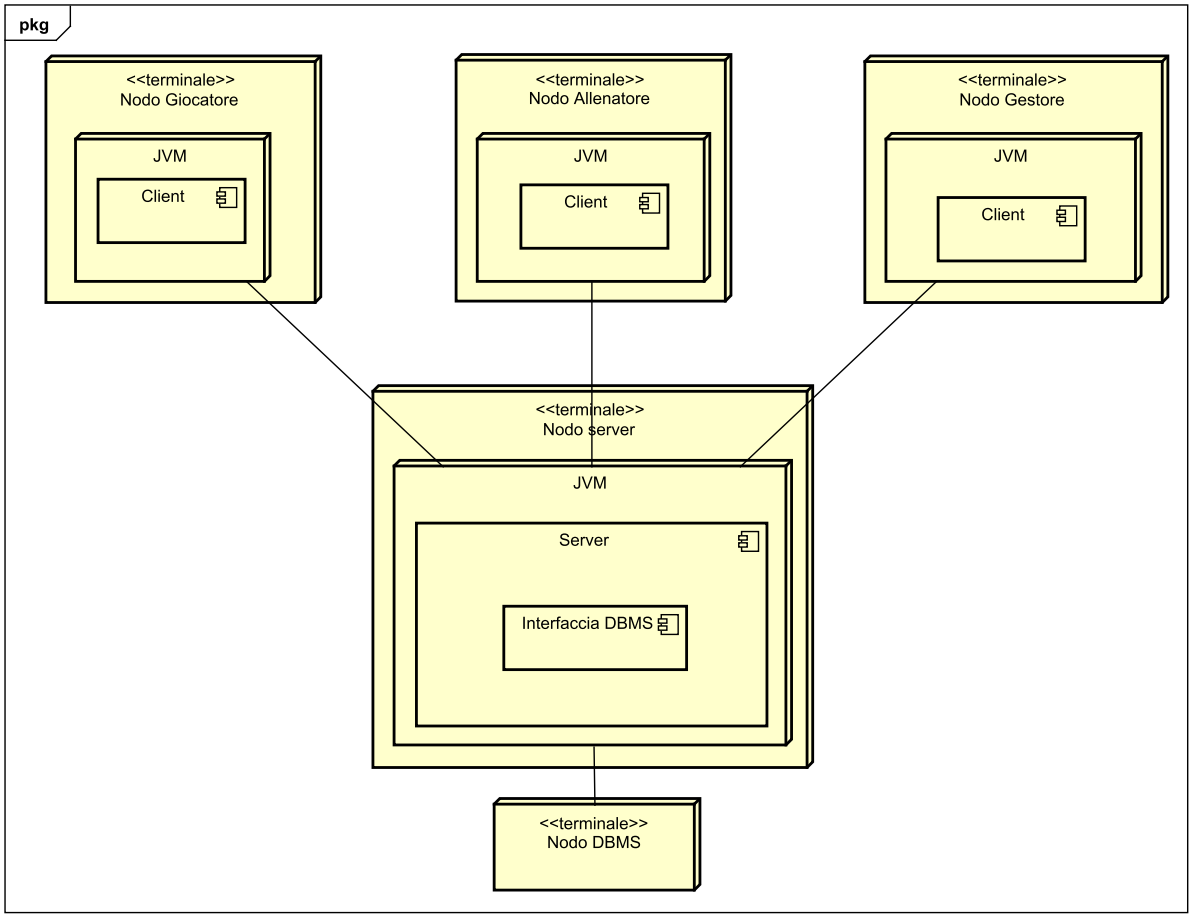
4.3.1.10. Interfaccia DBMS



4.4. Mappatura Hardware/Software

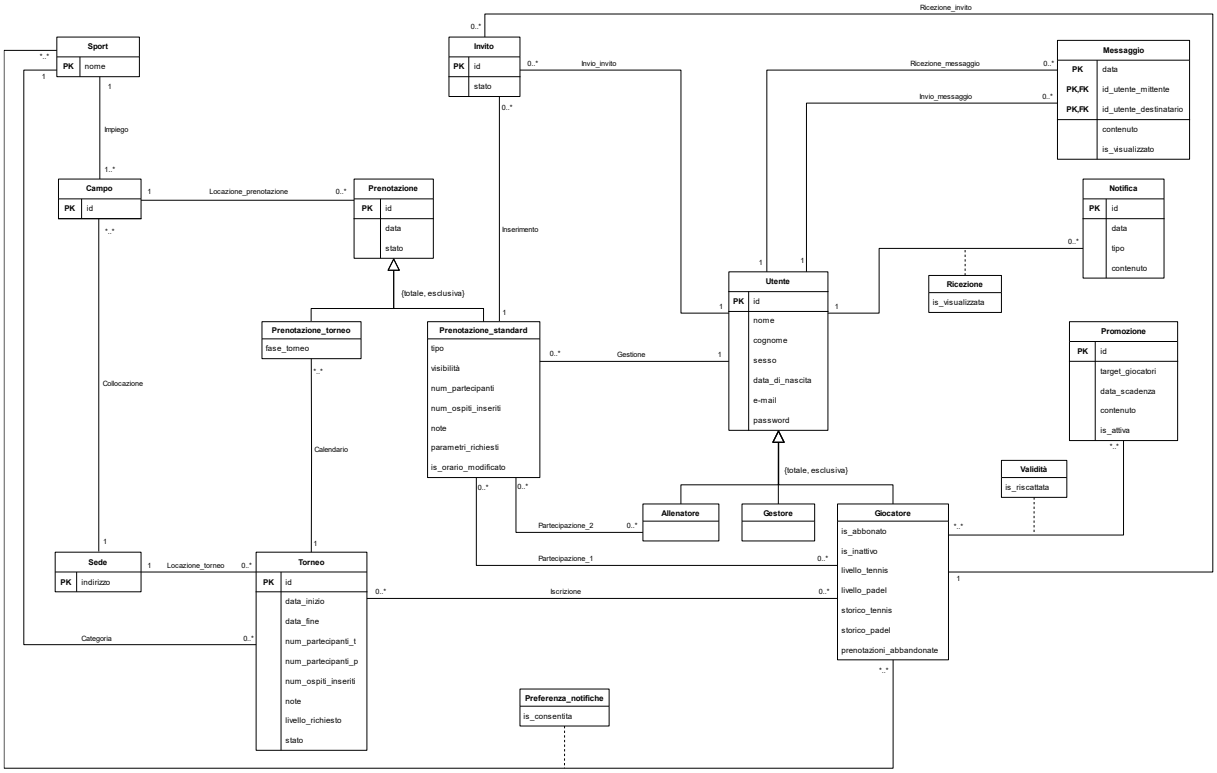
La mappatura è stata effettuata su un'architettura client-server. Il numero di nodi utente nell'immagine è puramente a scopo illustrativo; infatti, l'architettura permette un'implementazione con N nodi utente (per ciascuna classe di utente) mantenendo un solo server.

Su ogni nodo utente è installato un pacchetto contente tutte le componenti client per il funzionamento del sistema, nel nodo server invece vengono salvate tutte le componenti server, tra cui l'interfaccia al DBMS, che permette di comunicare con il nodo DBMS e ottenere quindi informazioni da esso.



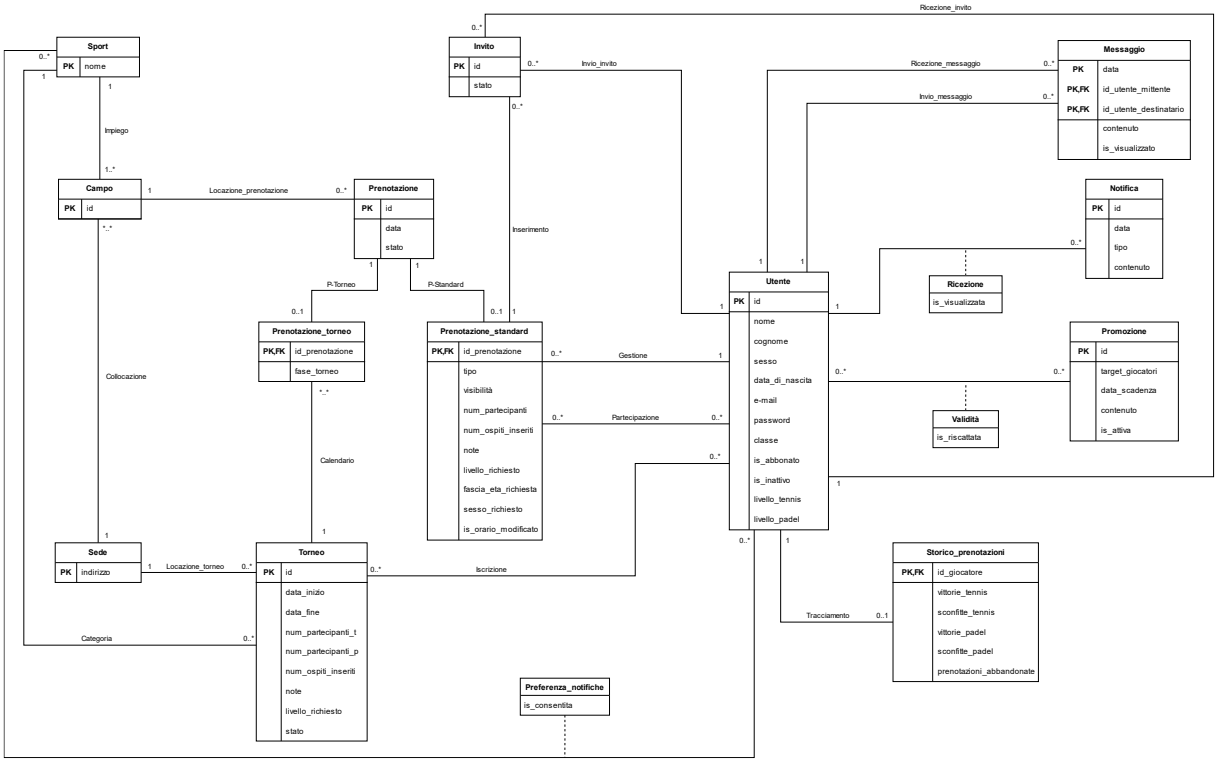
5. Gestione dei dati persistenti

5.1. Schema E-R



Regole di vincolo
(RV1) Un campo non può essere assegnato a più prenotazioni aperte previste per la stessa data e ora. (RV2) Il numero di partecipanti di una prenotazione non può essere superiore al massimo stabilito. (RV3) Il numero di partecipanti di un torneo non può essere superiore al massimo stabilito. (RV4) Il numero massimo di partecipanti di un torneo deve essere un multiplo del numero di partecipanti di ciascuna partita del torneo.
Regole di derivazione
(RD1) Il numero di partecipanti di una prenotazione si ottiene contando il numero di giocatori e allenatori che vi partecipano, e sommando a tale valore il numero di ospiti inseriti nella prenotazione. (RD2) Il numero di partecipanti di un torneo si ottiene contando il numero di giocatori che vi partecipano, e sommando a tale valore il numero di ospiti inseriti nel torneo.

5.2. Schema E-R ristrutturato



5.3. Modello relazionale

Utente(id_utente, nome, cognome, sesso, data_di_nascita, e-mail, password, classe, is_abbonato, is_inattivo, livello_tennis, livello_padel)

PK: id_utente

Prenotazione(id_prenotazione, ref_campo, data, stato)

PK: id_prenotazione

FK: ref_campo references Campo(id_campo)

Prenotazione_standard(ref_prenotazione, ref_utente_host, tipo, visibilità, num_partecipanti, num_ospiti_inseriti, note, livello_richiesto, fascia_eta_richiesta, sesso_richiesto, is_orario_modificato)

PK: ref_prenotazione

FK: ref_prenotazione references Prenotazione(id_prenotazione)

FK: ref_utente_host references Utente(id_utente)

Prenotazione_torneo(ref_prenotazione, ref_torneo, fase_torneo)

PK: ref_prenotazione

FK: ref_prenotazione references Prenotazione(id_prenotazione)

FK: ref_torneo references Torneo(id_torneo)

Torneo(id_torneo, ref_sede, ref_sport, data_inizio, data_fine, num_partecipanti_t, num_partecipanti_p, num_ospiti_inseriti, note, livello_richiesto, stato)

PK: id_torneo

FK: ref_sede references Sede(indirizzo_sede)

FK: ref_sport references Sport(nome_sport)

Sport(nome_sport)

PK: nome_sport

Campo(id_campo, ref_sede, ref_sport)

PK: id_campo

FK: ref_sede references Sede(indirizzo_sede)

FK: ref_sport references Sport(nome_sport)

Sede(indirizzo_sede)

PK: indirizzo_sede

Partecipazione(ref_prenotazione_standard, ref_utente_partecipante)

PK: ref_prenotazione_standard, ref_utente_partecipante

FK: ref_prenotazione_standard references Prenotazione_standard(ref_prenotazione)

FK: ref_utente_partecipante references Utente(id_utente)

Iscrizione(ref_torneo, ref_giocatore_partecipante)

PK: ref_torneo, ref_giocatore_partecipante

FK: ref_torneo references Torneo(id_torneo)

FK: ref_giocatore_partecipante references Utente(id_utente)

Invito(id_invito, ref_utente_mittente, ref_giocatore_destinatario, ref_prenotazione_standard, stato)

PK: id_invito

FK: ref_utente_mittente references Utente(id_utente)

FK: ref_giocatore_destinatario Utente(id_utente)

FK: ref_prenotazione_standard Prenotazione_standard(ref_prenotazione)

Messaggio(data, ref_utente_mittente, ref_utente_destinatario, contenuto, is_visualizzato)

PK: data, ref_utente_mittente, ref_utente_destinatario

FK: ref_utente_mittente references Utente(id_utente)

FK: ref_utente_destinatario references Utente(id_utente)

Storico_prenotazioni(ref_giocatore, vittorie_tennis, sconfitte_tennis, vittorie_padel, sconfitte_padel, prenotazioni_abbandonate)

PK: ref_giocatore

FK: ref_giocatore references Utente(id_utente)

Notifica(id_notifica, ref_utente_destinatario, data, tipo, contenuto, is_visualizzata)

PK: id_notifica

FK: ref_utente references Utente(id_utente)

Preferenza_notifiche(ref_giocatore, ref_sport, is_consentita)

PK: ref_giocatore, ref_sport

FK: ref_giocatore references Utente(id_utente)

FK: ref_sport references Sport(nome_sport)

Validità(ref_giocatore, ref_promozione, is_riscattata)

PK: ref_giocatore, ref_promozione

FK: ref_giocatore references Utente(id_utente)

FK: ref_promozione references Promozione(id_promozione)

Promozione(id_promozione, target_giocatori, data_scadenza, contenuto, is_attiva)

PK: id_promozione

5.4. Tabelle del modello relazionale

5.4.1. Utente

Nome attributo	Tipo	Vincoli	Descrizione
id_utente	varchar(25)	PK	Identificativo alfanumerico che identifica l'utente
nome	varchar(60)	Not null	Nome dell'utente
cognome	varchar(60)	Not null	Cognome dell'utente
data_di_nascita	date	Not null	Data di nascita dell'utente
sesso	char(1)	Not null	Sesso dell'utente
e-mail	varchar(255)	Not null, unique	E-mail dell'utente
password	varchar(32)	Not null	Password di accesso dell'utente
classe	varchar(10)	Not null, default 'Giocatore'	Classe dell'utente ("Giocatore", "Gestore", "Allenatore")
is_abbonato	boolean	Default false	Attributo per identificare gli abbonati
is_inattivo	boolean	Default false	Attributo per identificare i giocatori inattivi

livello_tennis	varchar(20)		Livello del giocatore per lo sport “Tennis”
livello_padel	varchar(20)		Livello del giocatore per lo sport “Padel”

5.4.2. Prenotazione

Nome attributo	Tipo	Vincoli	Descrizione
id_prenotazione	integer unsigned	PK, autoincrement	Codice identificativo della prenotazione
ref_campo	integer unsigned	FK, references Campo(id_campo), not null	Riferimento al campo assegnato alla prenotazione
data	datetime	Not null	Data e ora della prenotazione
stato	varchar(10)	Not null, default ‘Aperta’	Stato attuale della prenotazione (“Aperta”, “Chiusa”, “Cancellata”)

5.4.3. Prenotazione_standard

Nome attributo	Tipo	Vincoli	Descrizione
ref_prenotazione	integer unsigned	PK, FK, references Prenotazione (id_prenotazione)	Riferimento all’id della prenotazione
ref_utente_host	varchar(20)	FK, references Utente(id_utente), not null	Riferimento all’utente host della prenotazione
tipo	varchar(18)	Not null	Tipo di prenotazione (“Partita amichevole”, “Partita a livello”, “Allenamento”)
visibilità	varchar(14)	Not null	Visibilità della prenotazione (“Pubblica”, “Solo su invito”)
num_partecipanti	tinyint unsigned	Not null	Numero massimo di partecipanti (utenti e ospiti) della prenotazione
num_ospiti_inseriti	tinyint unsigned	Not null	Numero di ospiti inseriti nella prenotazione
note	varchar(250)		Note addizionali

livello_richiesto	varchar(20)		Livello richiesto per partecipare alla prenotazione
fascia_eta_richiesta	varchar(5)		Fascia d'età richiesta per partecipare alla prenotazione
sesso_richiesto	char(1)		Sesso richiesto per partecipare alla prenotazione
is_orario_modificato	boolean	Not null, default false	Attributo che indica se l'orario della prenotazione sia stato modificato o meno

5.4.4. Prenotazione_torneo

Nome attributo	Tipo	Vincoli	Descrizione
ref_prenotazione	integer unsigned	PK, FK, references Prenotazione (id_prenotazione)	Riferimento all'id della prenotazione
ref_torneo	integer unsigned	FK, references Torneo(id_torneo), not null	Riferimento all'id del torneo di cui la prenotazione fa parte
fase_torneo	varchar(12)	Not null	Fase del torneo a cui fa riferimento la prenotazione

5.4.5. Torneo

Nome attributo	Tipo	Vincoli	Descrizione
id_torneo	integer unsigned	PK, autoincrement	Codice identificativo del torneo
ref_sede	varchar(120)	FK, references Sede(indirizzo_sede), not null	Riferimento alla sede in cui si tiene il torneo
ref_sport	varchar(20)	FK, references Sport(nome_sport), not null	Riferimento allo sport del torneo
data_inizio	date	Not null	Data di inizio del torneo
data_fine	date	Not null	Data di fine del torneo
num_partecipanti_t	tinyint unsigned	Not null	Numero massimo di partecipanti (utenti e ospiti) del torneo

num_partecipanti_p	tinyint unsigned	Not null	Numero massimo di partecipanti (utenti e ospiti) di ogni partita del torneo
num_ospiti_inseriti	tinyint unsigned	Not null, default 0	Numero di ospiti inseriti nel torneo
note	varchar(250)		Note addizionali
livello_richiesto	varchar(20)	Not null	Livello richiesto per partecipare al torneo
stato	varchar(10)	Not null, default 'Aperto'	Stato attuale del torneo ("Aperto", "Chiuso", "Cancellato")

5.4.6. Sport

Nome attributo	Tipo	Vincoli	Descrizione
nome_sport	varchar(20)	PK	Nome dello sport

5.4.7. Campo

Nome attributo	Tipo	Vincoli	Descrizione
id_campo	integer unsigned	PK, autoincrement	Codice identificativo del campo
ref_sede	varchar(120)	FK, references Sede(indirizzo_sede), not null	Riferimento alla sede in cui è situato il campo
ref_sport	varchar(20)	FK, references Sport(nome_sport), not null	Riferimento allo sport per il quale viene impiegato il campo

5.4.8. Sede

Nome attributo	Tipo	Vincoli	Descrizione
indirizzo_sede	varchar(120)	PK	Indirizzo della sede

5.4.9. Partecipazione

Nome attributo	Tipo	Vincoli	Descrizione
ref_prenotazione_standard	integer unsigned	PK, FK, references Prenotazione_standard (ref_prenotazione)	Riferimento all'id della prenotazione
ref_utente_partecipante	varchar(20)	PK, FK, references Utente(id_utente)	Riferimento all'utente partecipante

5.4.10. Iscrizione

Nome attributo	Tipo	Vincoli	Descrizione
ref_torneo	integer unsigned	PK, FK, references Torneo(id_torneo)	Riferimento al torneo cui fa riferimento l'iscrizione
ref_giocatore_partecipante	varchar(20)	PK, FK, references Utente(id_utente)	Riferimento al giocatore iscritto

5.4.11. Invito

Nome attributo	Tipo	Vincoli	Descrizione
id_invito	integer	PK, autoincrement	Codice identificativo dell'invito
ref_utente_mittente	varchar(20)	FK, references Utente(id_utente), not null	Riferimento all'utente mittente dell'invito
ref_giocatore_destinatario	varchar(20)	FK, references Utente(id_utente), not null	Riferimento al giocatore destinatario dell'invito
ref_prenotazione_standard	integer unsigned	FK, references Prenotazione_standard (ref_prenotazione), not null	Riferimento all'id della prenotazione
stato	varchar(10)	Not null, default 'Pendente'	Stato dell'invito ("Pendente", "Accettato", "Rifiutato", "Non valido")

5.4.12. Messaggio

Nome attributo	Tipo	Vincoli	Descrizione
data	datetime	PK	Data e ora di invio del messaggio
ref_utente_mittente	varchar(20)	PK, FK, references Utente(id_utente)	Riferimento all'utente mittente del messaggio
ref_utente_destinatario	varchar(20)	PK, FK, references Utente(id_utente)	Riferimento all'utente destinatario del messaggio
contenuto	varchar(2000)	Not null	Contenuto testuale del messaggio
is_visualizzato	boolean	Not null, default false	Attributo che indica se il messaggio sia stato visualizzato o meno

5.4.13. Storico_prenotazioni

Nome attributo	Tipo	Vincoli	Descrizione
ref_giocatore	varchar(20)	PK, FK, references Utente(id_utente)	Riferimento al giocatore a cui appartiene lo storico
vittorie_tennis	tinyint unsigned	Not null, default 0	Numero delle ultime "Partite a livello" dello sport "Tennis" vinte dal giocatore
sconfitte_tennis	tinyint unsigned	Not null, default 0	Numero delle ultime "Partite a livello" dello sport "Tennis" perse dal giocatore
vittorie_padel	tinyint unsigned	Not null, default 0	Numero delle ultime "Partite a livello" dello sport "Padel" vinte dal giocatore
sconfitte_padel	tinyint unsigned	Not null, default 0	Numero delle ultime "Partite a livello" dello sport "Padel" perse dal giocatore

prenotazioni_abbandonate	tinyint unsigned	Not null, default 0	Numero delle prenotazioni abbandonate dal giocatore nell'ultimo mese

5.4.14. Notifica

Nome attributo	Tipo	Vincoli	Descrizione
id_notifica	integer	PK, autoincrement	Codice identificativo della notifica
ref_utente_destinatario	varchar(20)	FK, references Utente(id_utente), not null	Riferimento all'utente destinatario della notifica
data	datetime	Not null	Data e ora in cui la notifica è stata inviata
tipo	varchar(12)	Not null	Contesto cui fa riferimento la notifica ("Prenotazione", "Torneo", "Invito", "Messaggio", "Avviso", "Promozione")
contenuto	varchar(1000)	Not null	Contenuto testuale della notifica
is_visualizzata	boolean	Not null, default false	Attributo che indica se la notifica sia stata visualizzata o meno

5.4.15. Preferenza_notifiche

Nome attributo	Tipo	Vincoli	Descrizione
ref_giocatore	varchar(20)	PK, FK, references Utente(id_utente)	Riferimento al giocatore
ref_sport	varchar(6)	PK, FK, references Sport(nome_sport)	Riferimento sport per cui viene abilitata o meno la ricezione di notifiche
is_consentita	boolean	Not null, default true	Attributo che indica se il giocatore abbia abilitato o meno la ricezione di notifiche per un particolare sport

5.4.16. Validità

Nome attributo	Tipo	Vincoli	Descrizione
ref_giocatore	varchar(20)	PK, FK, references Utente(id_utente)	Riferimento al giocatore
ref_promozione	integer	PK, FK references Promozione (id_promozione)	Riferimento alla promozione
is_riscattata	boolean	Not null, default false	Attributo che indica se il giocatore abbia riscattato o meno la promozione

5.4.17. Promozione

Nome attributo	Tipo	Vincoli	Descrizione
id_promozione	integer	PK, autoincrement	Codice identificativo della promozione
target_giocatori	varchar(12)	Not null	Target di giocatori destinatari della promozione (“Abbonati”, “Non abbonati”, “Tutti”)
data_scadenza	date	Not null	Data di scadenza della promozione
contenuto	varchar(500)	Not null	Contenuto testuale della promozione
is_attiva	boolean	Not null, default true	Attributo che indica se la promozione sia attiva o meno