

Object Oriented Programming

Piattaforma gaming

Claudio Morè – 228611

Simone Sfredda - 228610

REQUISITI FUNZIONALI

- **Login:** tramite un form è possibile effettuare il login all'applicazione per visualizzare la homepage
- **Registrazione:** tramite un form è possibile registrarsi all'applicazione in modo da poter usufruire di tutti i servizi
- **Dettagli gioco:** è possibile visualizzare i dettagli del gioco, compresa di valutazione e descrizione
- **Recensione:** sotto ad ogni gioco è possibile lasciare un commento e un voto da 1 a 5 per valutare l'esperienza di gioco
- **Profilo:** è possibile visualizzare il profilo personale dove sono presenti gli achievements conseguiti giocando e il progresso per il livello successivo
- **Giocare:** all'interno di ogni gioco c'è un pulsante che ci permette di aprire una finestra ed effettuare una sessione di gioco
- **Approva Commento:** dal back end è possibile, solo da parte dell'amministratore o del moderatore, approvare o meno un commento
- **Promozione/Retrocessione Utente:** nel back end è possibile promuovere un utente base a moderatore o amministratore. Inoltre è anche possibile retrocede un utente da amministratore a moderatore o utente base
- **Registrazione moderatore:** nel back end è possibile registrare un utente che sarà direttamente moderatore

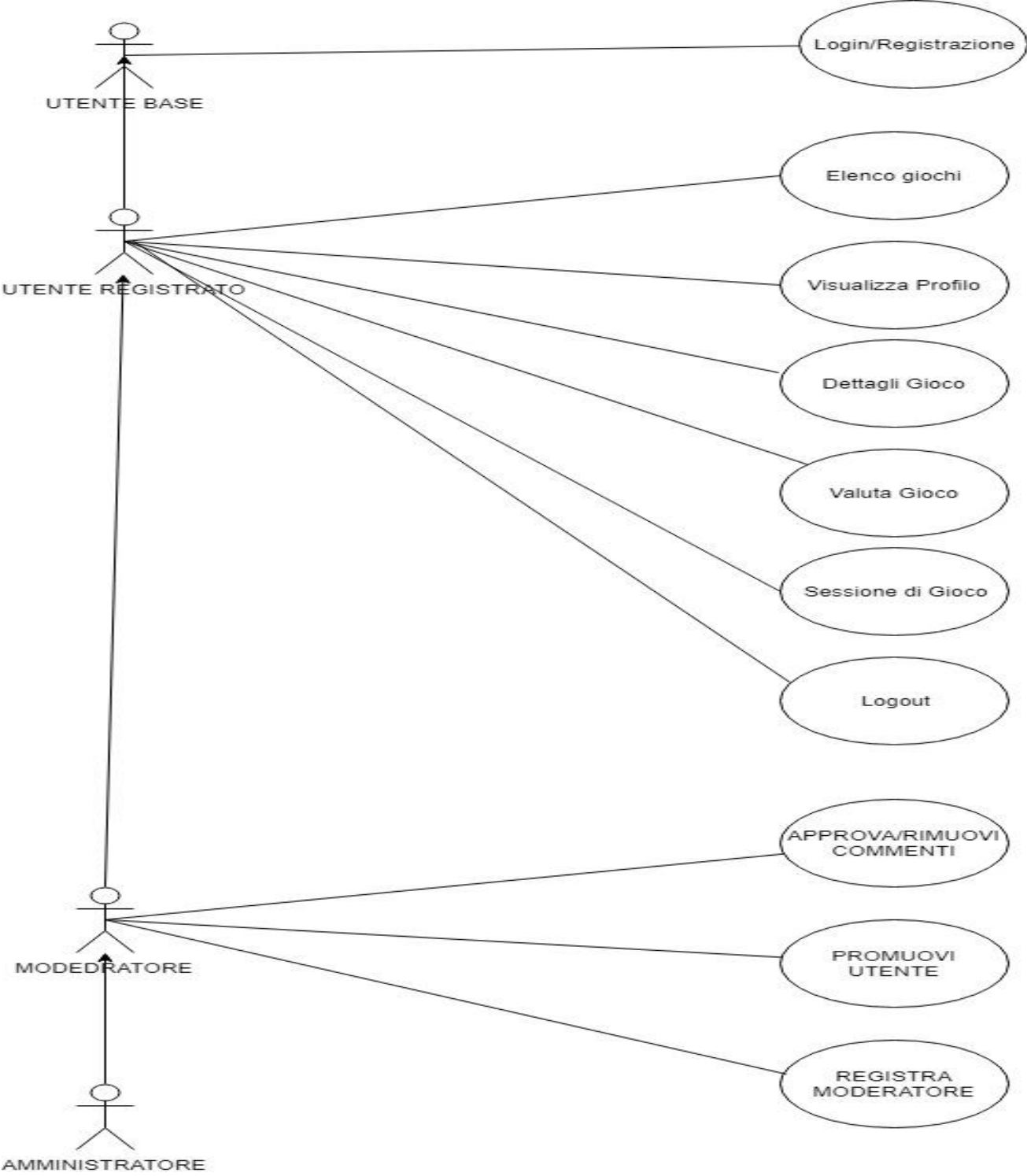
REQUISITI NON FUNZIONALI

- **AFFIDABILITA':** il sistema dovrà garantire il pieno funzionamento dell'applicazione
- **USER-FRIENDLY:** l'applicazione dovrà essere facile da usare per qualunque tipo di utente
- **DISPONIBILITA':** Le funzioni offerte dall'applicazione dovranno essere raggiungibili in qualunque momento

ATTORI DEL SISTEMA

- **UTENTE BASE:** può solo effettuare il login o la registrazione
- **UTENTE AVANZATTO:** dopo aver effettuato il login, può visualizzare l'elenco dei giochi, i dettagli di ogni singolo gioco, effettuare un'esperienza di gioco, recensire un gioco, visualizzare il proprio profilo e infine fare logout
- **MODERATORE:** ha tutte le funzioni dell'utente avanzato e in più può approvare i commenti, promuovere un utente avanzato a moderatore e aggiungere un nuovo moderatore
- **AMMINISTRATORE:** ha tutte le funzioni del moderatore e in più può promuovere un moderatore ad amministratore

USE CASE DIAGRAM



DESCRIZIONE DEGLI USE CASES

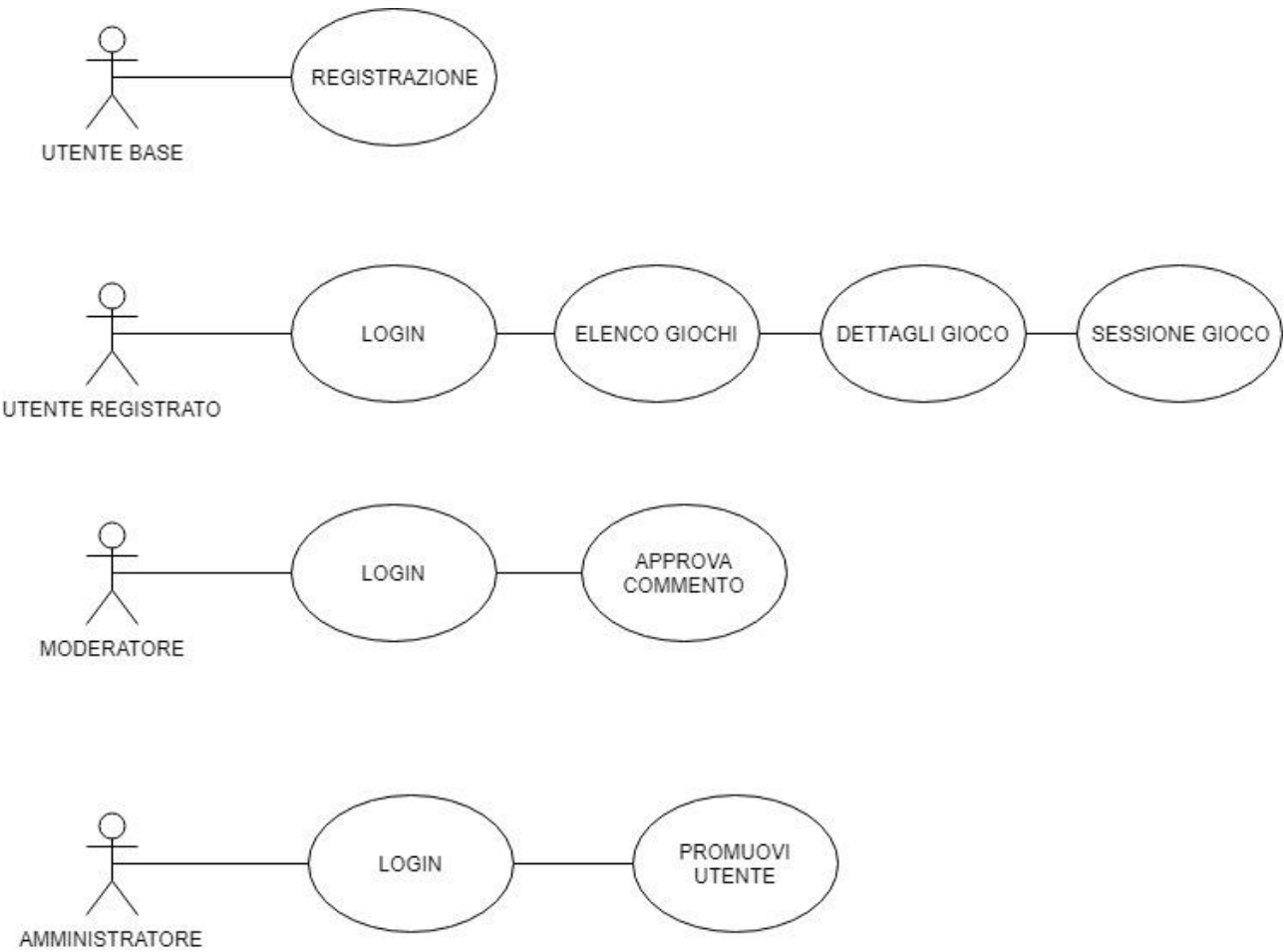
Use-Case	Registrazione
Attori Presenti	Utente base, Utente Registrato, Moderatore, Amministratore
Evento	Click sul pulsante registrazione
Descrizione	Permette a tutti gli attori del sistema di registrarsi all'interno dell'applicazione web, in modo da usufruire dei servizi messi a disposizione per loro.

Use-Case	Dettagli Gioco
Attori Presenti	Utente Registrato, Moderatore, Amministratore
Evento	Click sull'icona del gioco
Descrizione	Permette a tutti gli utenti ad esclusione dell'utente base di visualizzare la descrizione del gioco, la valutazione e le sue relative recensioni lasciate dagli utenti.

Use-Case	Approva Recensione
Attori Presenti	Moderatore, Amministratore
Evento	Click sul tasto approva
Descrizione	Permette al Moderatore e all'Amministratore di controllare una recensione prima che essa venga caricata all'interno dell'applicazione web. Il loro compito sarà quindi di decidere se approvarla oppure rifiutarla.

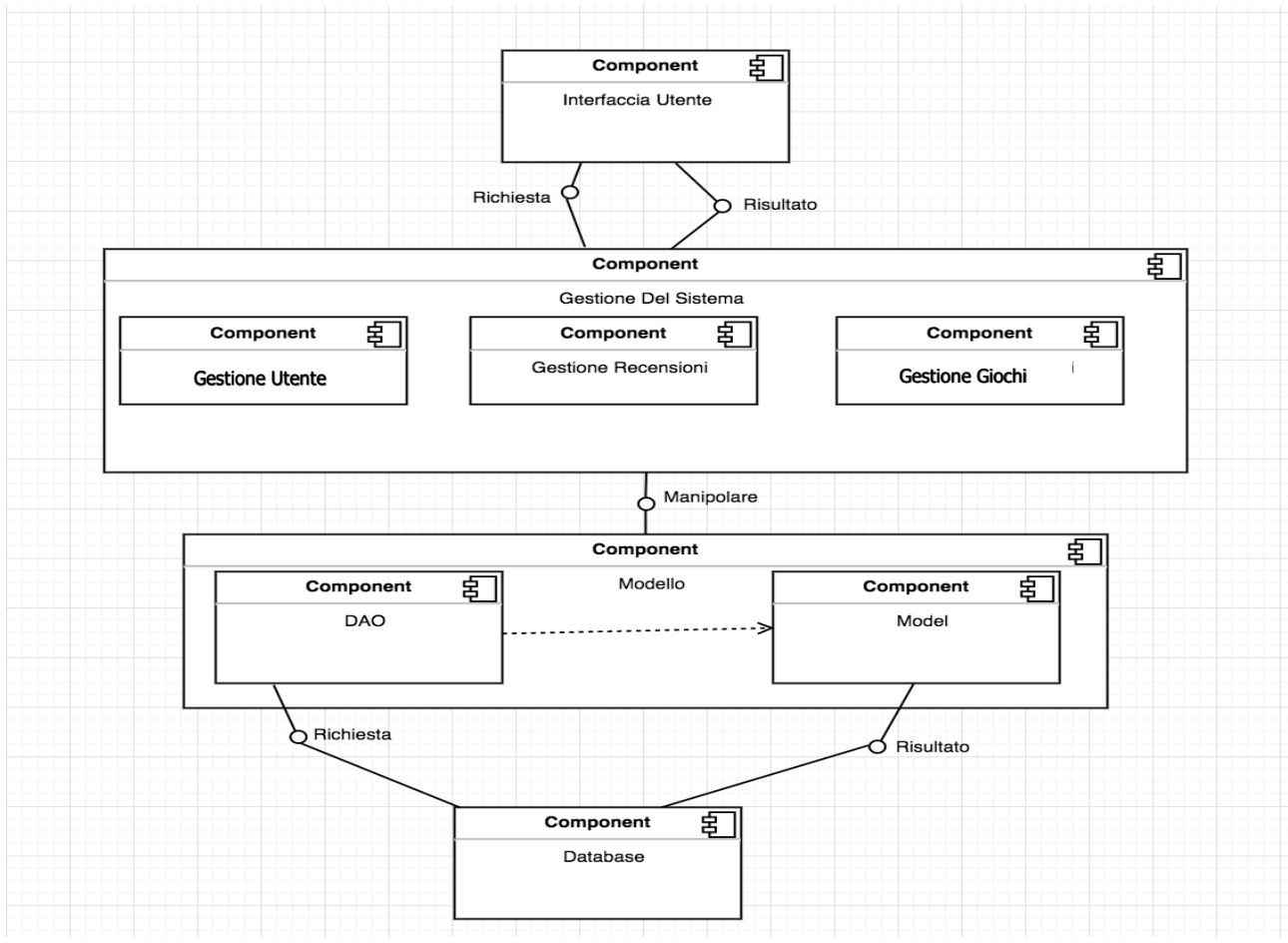
Use-Case	Inserimento Gioco
Attori Presenti	Amministratore
Evento	Click sul pulsante carica gioco
Descrizione	Permette all'Amministratore di caricare un nuovo gioco all'interno dell'applicazione web

SCENARI



L'immagine rappresenta esempi di azioni che, attori esterni al sistema, possono compiere

ARCHITETURA DEL SISTEMA



Ho sviluppato la piattaforma gaming sotto forma di applicazione web perché ho ritenuto questa scelta più appropriata visto l'argomento della piattaforma.

Le componenti principali del mio sistema sono:

- **L'interfaccia utente:** utilizzata dagli utenti per navigare nell'applicazione. Per far visualizzare i dati nelle mie pagine html ho utilizzato FreeMarker, utile componente che mi ha permesso di facilitare il passaggio dei dati elaborati dal sistema, dalle servlet alle pagine html.
- **Gestione del sistema:** la gestione dei dati è stata affidata alle Servlet scritte in linguaggio JAVA
- **Modello Dati:** ogni servlet fa riferimento a dei modelli di dati che implemento delle interfacce e che rendono più pulito il codice.
- **Database JDBC:** tutti i dati presenti nel sistema sono salvati in un database

SCELTE E STRATEGIE ADOTATE

Ho scelto di utilizzare **FreeMarker** semplicemente perché ho sviluppato diverse applicazioni web con questa tecnologia e quindi mi ha reso questo compito molto più semplice rispetto ad utilizzare le classiche pagine jsp come pagine di navigazione.

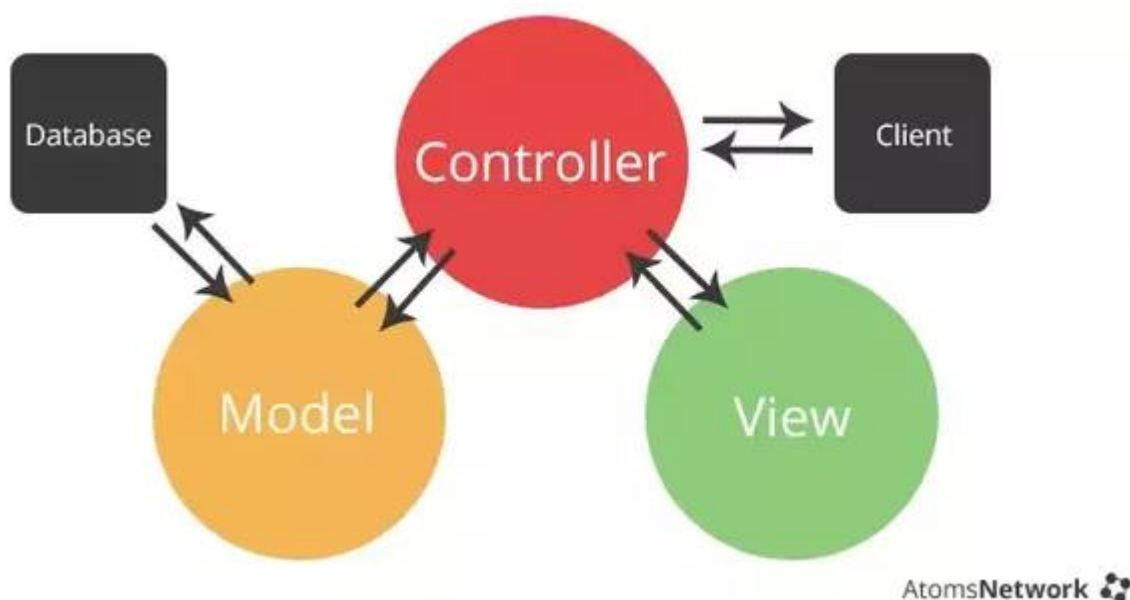
Ho deciso di non permettere l'inserimento di altri giochi visto che ho trovato difficoltà nell'inserire, non tanto la pagina che visualizza il gioco, ma quanto la pagina JAVA che contiene le funzioni del gioco.

Un utente base per vedere la lista dei giochi deve per forza effettuare il login. Ho fatto questa scelta prendendo spunto dalle piattaforme gaming più importanti.

DESIGN PATTERN

Ho utilizzato il design pattern MVC (Model view control).

- **MODEL:** qui sono contenuti tutti i metodi per manipolare i dati
- **VIEW:** questa componente rappresenta tutte le pagine che visualizzano i dati manipolati nel model
- **CONTROLLER:** riceve i dati dall'utente tramite le view e poi appoggiandosi al model elabora i dati

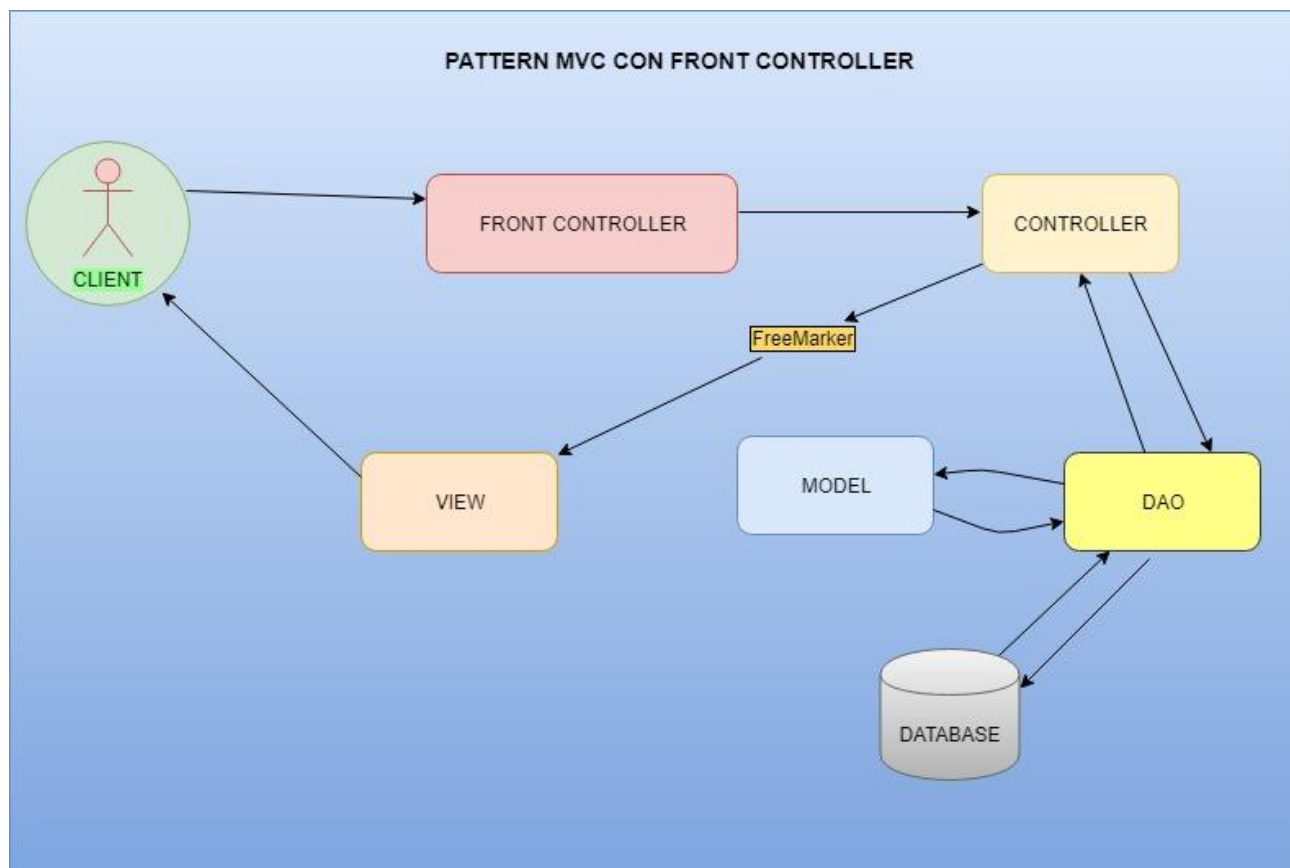


Per rendere questo pattern ancora più efficace ho cercato di tenere queste tre componenti il più separate possibile in modo che se si dovessero attuare modifiche a una delle tre componenti non andiamo nella maggior parte dei casi a toccare le altre

Un altro pattern utilizzato è stato il DAO. Le classi DAO hanno il compito di comunicare con il DB svolgendo le opportune query.

Ho anche utilizzato il pattern Front Controller che completa quello dell'MVC. Questo pattern permette di intercettare tutti i dati, diretti alle rispettive servlet, e convogliarli presso il front controller che a sua volta instraderà i dati verso la servlet apposita che elaborerà i dati e restituirà un risultato. Generalmente i risultati delle servlet vengono rimandati nuovamente al front controller che a sua volta manderà i dati alla view corretta ma nel mio caso, utilizzando FreeMarker i risultati delle servlet vengono rimandati direttamente alla view corretta senza ripassare nel front controller.

PATTERN MVC CON FRONT CONTROLLER



In questa immagine è possibile vedere il funzionamento del sistema mostrando i design pattern utilizzati. I dati vengono inviati dal client verso il front controller che avrà il compito di smistarli al giusto controller/servlet. All'interno della servlet le richieste verranno elaborate interrogando il DAO che si occuperà di effettuare delle query e impacchettare i risultati da rinviare alla servlet. Nel fare questo il DAO si appoggia al model che contiene le classi JAVA che rappresentano le entità del Database con i vari metodi get e set. Una volta che i dati sono stati elaborati vengono inviati alla view corretta tramite l'utilizzo del FreeMarker, senza ripassare nel front controller (come la normale implementazione di questo pattern richiede).

SCELTE IMPLEMENTATIVE

- **PROMOZIONE E RETROCESSIONE DEGLI UTENTI:** ho scelto di implementare questa funzione considerando che i moderatori possono promuovere solo gli utenti registrati e retrocedere altri moderatori ma non possono arrivare al livello dell'amministratore, invece l'amministratore può promuovere o retrocedere chiunque.
- **INSERIMENTO DI UN NUOVO GIOCO:** ho riscontrato diversi problemi nell'inserimento del nuovo gioco quindi ho evitato di implementare questa funzione. Per inserire un nuovo gioco si dovrà entrare nel codice e modificare da lì
- **RECENSIONE E VOTAZIONE DI UN GIOCO:** queste due funzioni ho pensato di unirle in un'unica schermata. Queste due funzioni devono essere fatte insieme.

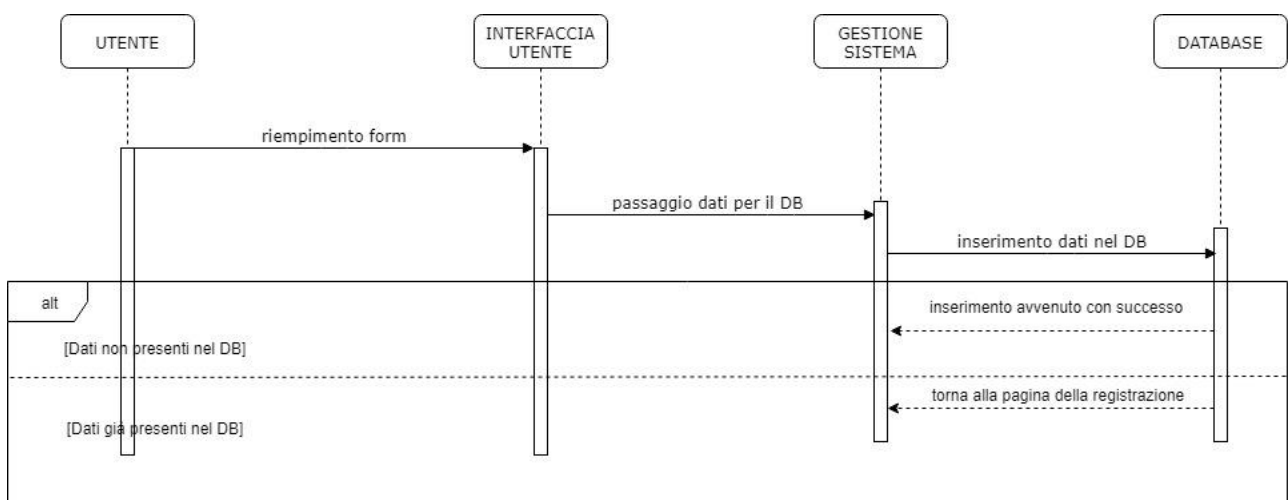
LIBRERIE UTILIZZATE

- **APACHE TOMCAT:** web server open source sviluppato dalla Apache Software Foundation. Implementa le specifiche JavaServer Pages e servlet, fornendo quindi una piattaforma software per l'esecuzione di applicazioni Web sviluppate in linguaggio Java.
- **FREEMARKER:** template engine, originariamente incentrato sulla generazione di pagine web dinamiche con architettura software MVC. Si tratta di un template engine senza dipendenze da servlet o HTTP o HTML.
- **MYSQL CONNECTOR:** libreria che permette la comunicazione con il database MySQL.
- **BOOTSTRAP:** utilizzato per facilitare il lavoro nella creazione delle pagine html grazie ai tantissimi widget messi a disposizione.

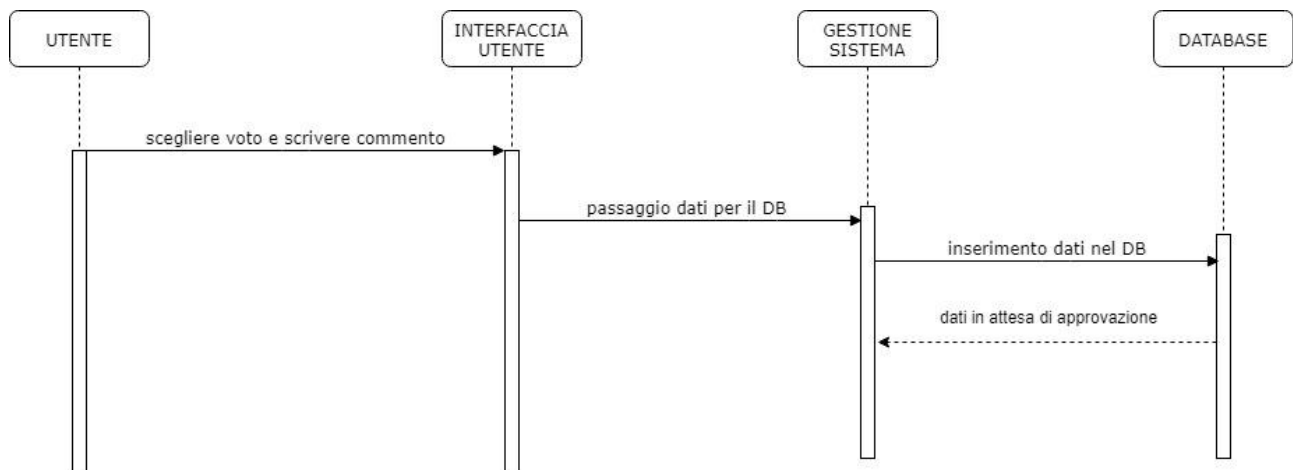
SEQUENCE DIAGRAM

Il sequence diagram ci aiuta a rappresentare un possibile scenario in cui gli attori effettuano determinate azioni all'interno del sistema.

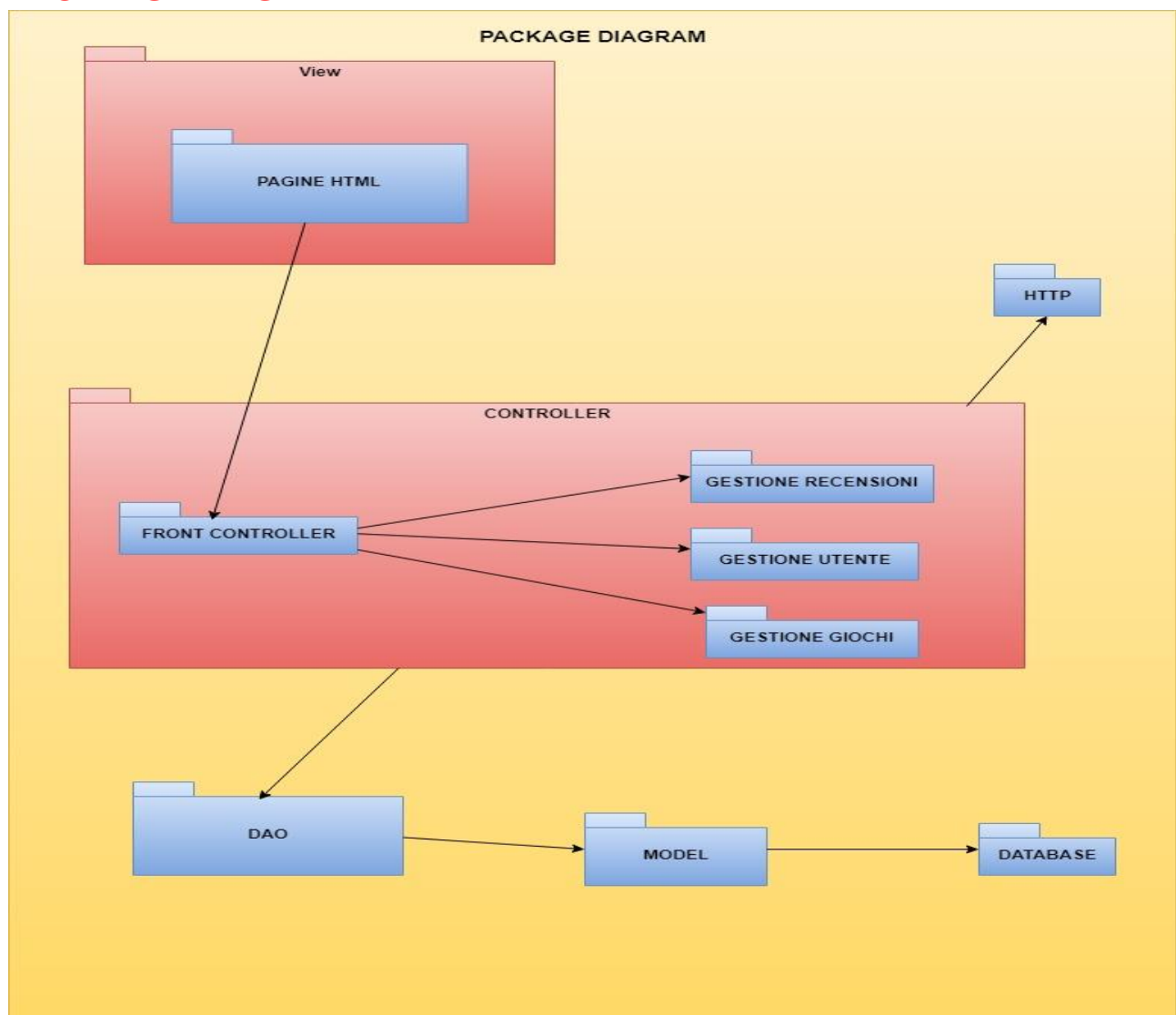
Registrazione



Commenti

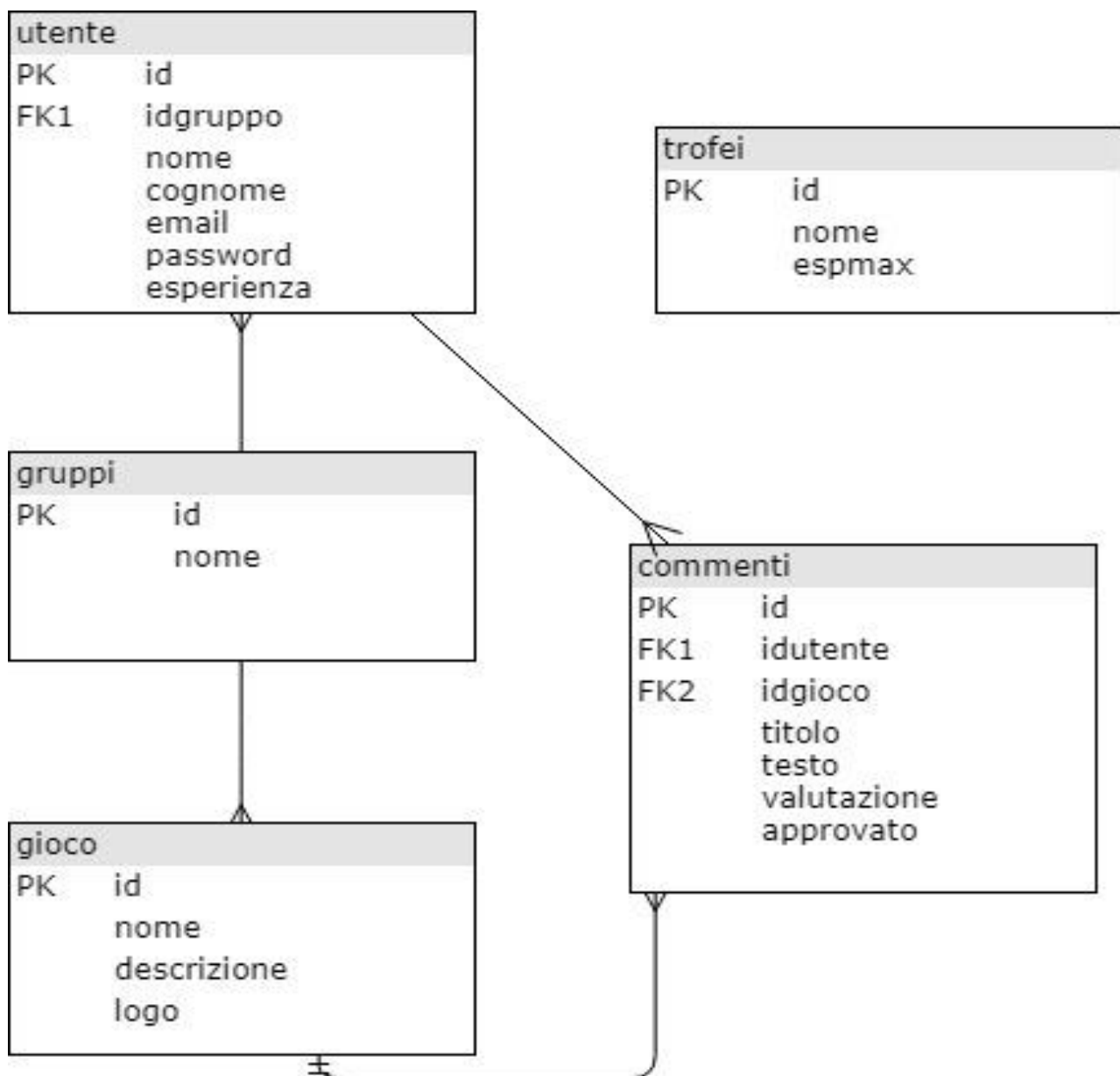


PACAKAGE DIAGRAM



Il package diagram nell'UML è usato per raggruppare elementi e fornire un namespace per gli elementi. Un package può contenere altri package, fornendo così un'organizzazione gerarchica dei package. I nostri package rappresentano i vari pattern utilizzati.

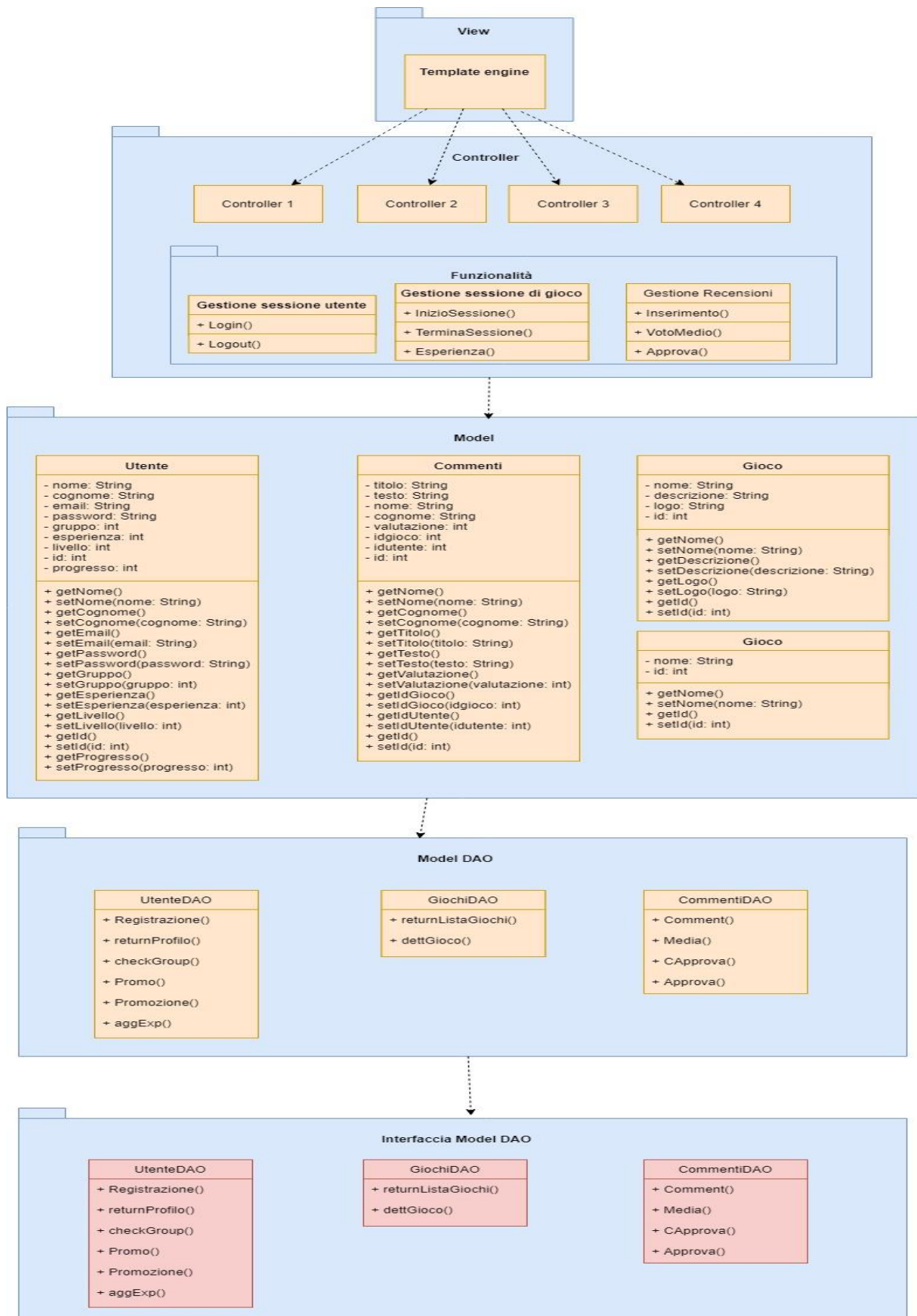
DESCRIZIONE DATABASE



Per sviluppare il database della mia applicazione ho utilizzato 5 entità:

- **UTENTE:** in questa entità vengono memorizzati tutti gli utenti registrati al sistema con i relativi dati anagrafici, le credenziali per l'accesso all'applicazione e la chiave esterna dell'id del gruppo a cui appartengono per distinguere quelli che sono utenti registrati, moderatori o amministratori
- **GRUPPI:** questa entità contiene i tre gruppi fondamentali del sistema che sono il gruppo degli utenti registrati, il gruppo dei moderatori e il gruppo degli amministratori
- **COMMENTI:** in questa entità sono contenuti tutte le recensioni fatte dagli utenti. I commenti sono collegati agli utenti e ai giochi con la chiave esterna. Con l'attributo approvato è possibile capire appunto se il commento è stato revisionato o meno
- **GIOCO:** qui sono contenute le informazioni riguardanti ogni singolo gioco
- **TROFEI:** qui troviamo i vari livelli raggiungibili da parte degli utenti incrementando la propria esperienza

CLASS DIAGRAM



VIEW APPLICAZIONE WEB

Login

PIATTAFORMA GAMING

INDIRIZZO E-MAIL

Inserisci qui la tua mail

PASSWORD

Inserisci qui la tua password

LOGIN

NON SEI REGISTRATO? FALLO ORA!!

Home

PIATTAFORMA GAMING

Home

Profilo

LOGOUT



su o giu?




roulette



roulette



roulette



roulette

Profilo

PIATTAFORMA GAMING

Home

Profilo

LOGOUT

DATI PROFILO

NOME: c

COGNOME: c

LIVELLO: 6

ESPERIENZA: 1.180

PROGRESSO VERSO IL PROSSIMO LIVELLO

ACHIEVEMENTS

2

3

4

5

6