

# PIATTAFORMA GAMING

## PROGETTO OBJECT-ORIENTED SOFTWARE DESIGN

---

### STUDENTI:

D'Ascenzo Andrea	243358
Icaro Iari	243742
Martella Stefano	243848
Morelli Davide	243745
Tramontozzi Paolo	242900

## SCOPO

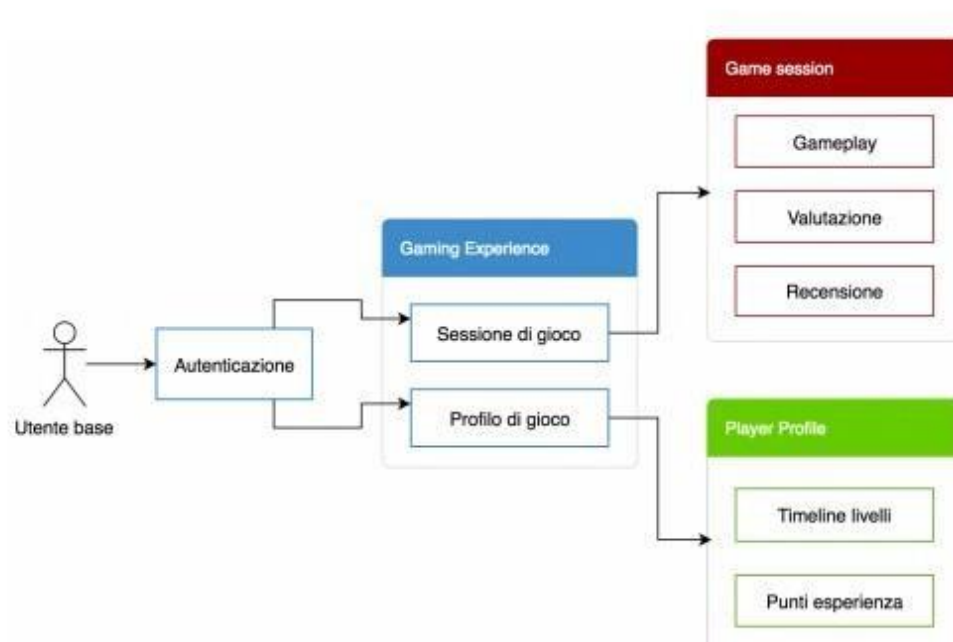
Il progetto si propone di realizzare una piattaforma gaming che coinvolga gli utenti a giocare al fine di guadagnare punti esperienza e nuovi livelli di gioco.

## ATTORI DEL SISTEMA

### UTENTE

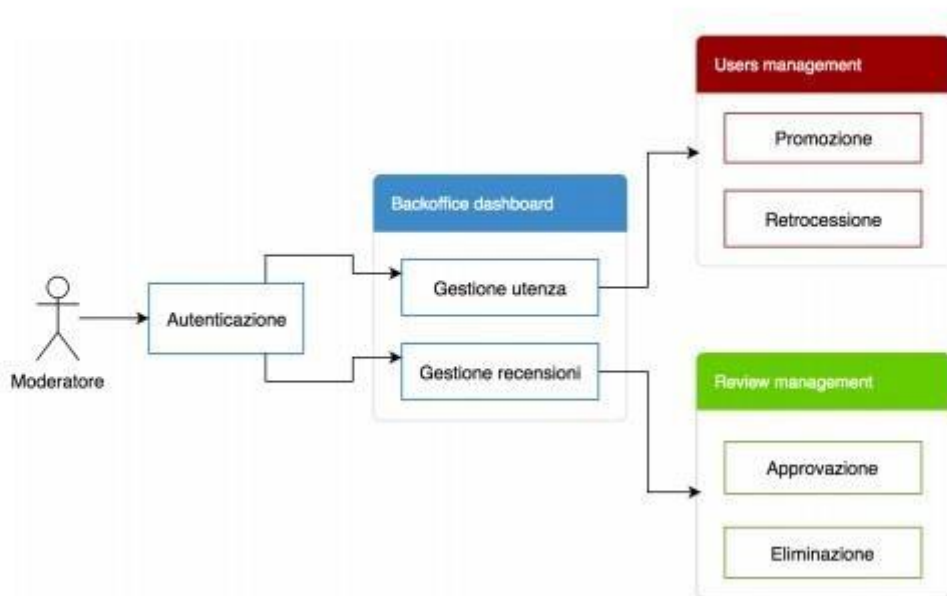
Attore presente nel sistema in seguito a registrazione. Potrà accedere tramite login e visualizzare la sua home. Da qui sarà capace di controllare e, volendo, modificare il suo profilo anagrafico oppure avrà la possibilità di entrare nel profilo gaming, il quale conterrà informazioni sulla sua esperienza di gioco attuale, descritta da un livello esperienza e da una serie di trofei ottenuti aumentando il proprio livello.

Potrà inoltre accedere alla lista di giochi e prendervi parte a suo piacimento, e successivamente valutarli con recensioni o voti.



## MODERATORE

Il moderatore è un attore con dei privilegi: potrà eseguire le stesse attività dell'utente, ma avrà anche il compito di moderare la parte di piattaforma relativa all'esperienza di gioco; in particolare i suoi compiti sono quelli di valutare le recensioni proposte dagli utenti, quindi accettarle o declinarle, e promuovere utenti base a nuovi moderatori o retrocedere quest'ultimi in utenti base.



## AMMINISTRATORE DI SISTEMA

Attore che gestisce l'intera piattaforma gaming. Saranno suoi doveri controllare la fruibilità e l'efficienza del sistema, interagire con la base di dati, promuovere a moderatore utenti già registrati nella piattaforma.

A differenza degli altri attori, l'amministratore non potrà prendere parte ai giochi, di conseguenza non possiederà livelli o trofei.

## REQUISITI

### ELENCO DEI REQUISITI

- *Registrazione*: feature che consente la registrazione al sistema da parte di un utente
- *Autenticazione*: feature che permette di effettuare il login da parte di un utente precedentemente registrato
- *Consultazione profilo*: azione che consente all'utente autenticato la visualizzazione del profilo, diviso in sezione gaming e sezione informazioni personali
- *Elenco giochi*: lista di giochi che l'utente può consultare ed infine scegliere per giocare
- *Sessione di gioco*: insieme di funzionalità connesse al gioco; la suddivisione in macro aree è la seguente:
  - Partecipazione al gioco: feature che consente all'utente di selezionare il gioco al quale vuole partecipare
  - Attribuzione XP: assegnazione punti XP per ogni bet effettuata da un utente per un determinato gioco
  - Evoluzione livello: funzione che aumenta il livello di un utente, raggiunti i punti XP necessari
  - Ottenimento trofei: feature consecutiva al raggiungimento di un livello – l'utente ottiene un trofeo come premio per essere avanzato di livello
- *Area recensione*: insieme di funzionalità che consentono ad un utente di valutare un'esperienza di gioco; la valutazione può essere espressa sotto forma di:
  - Voto: valutazione numerica dell'esperienza di gioco
  - Recensione: giudizio espresso tramite un commento dall'utente

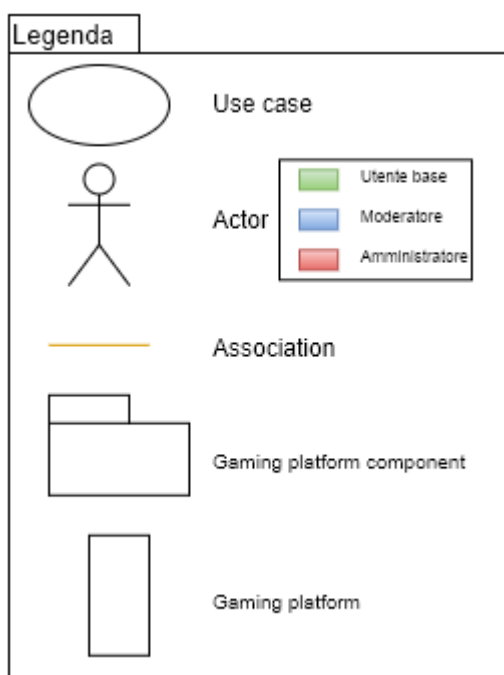
Relativamente ai moderatori, l'area recensione è estesa a

- Validazione: attività di valutazione della recensione: un moderatore sceglierà se pubblicare o rifiutare la recensione di un utente

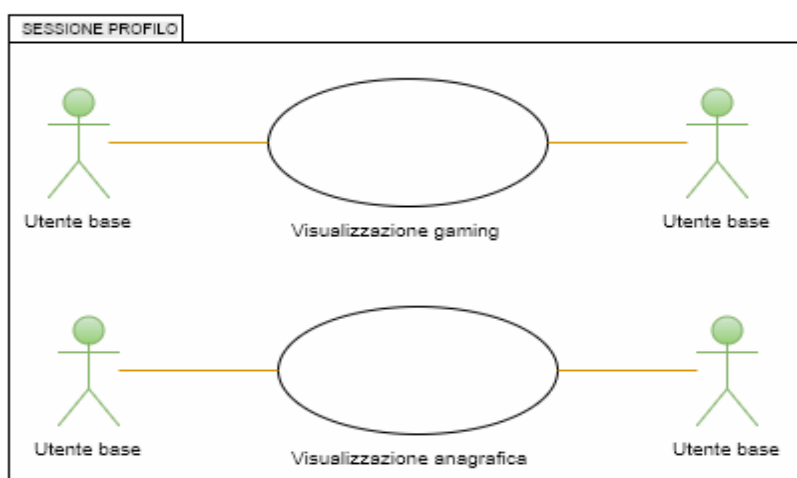
## USE CASES

Tecnica che valuta ogni requisito focalizzandosi sugli attori che interagiscono col sistema. Sono rappresentati in modello UML tramite gli *Use Cases Diagrams*

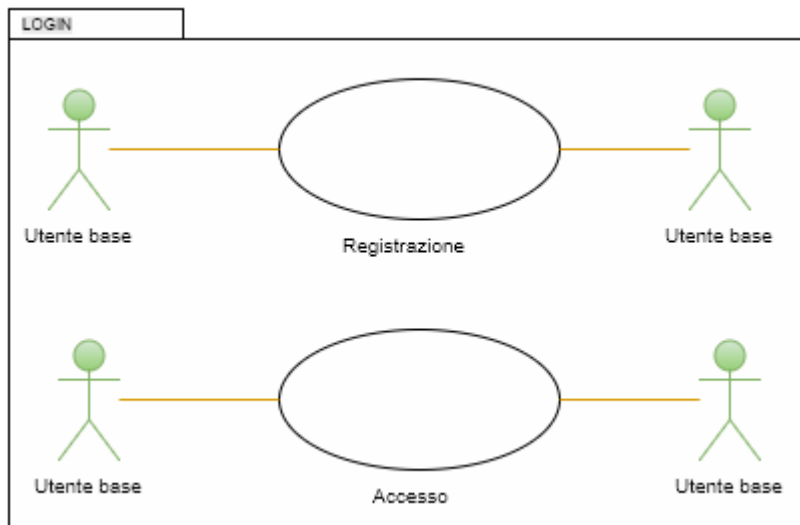
### LEGENDA



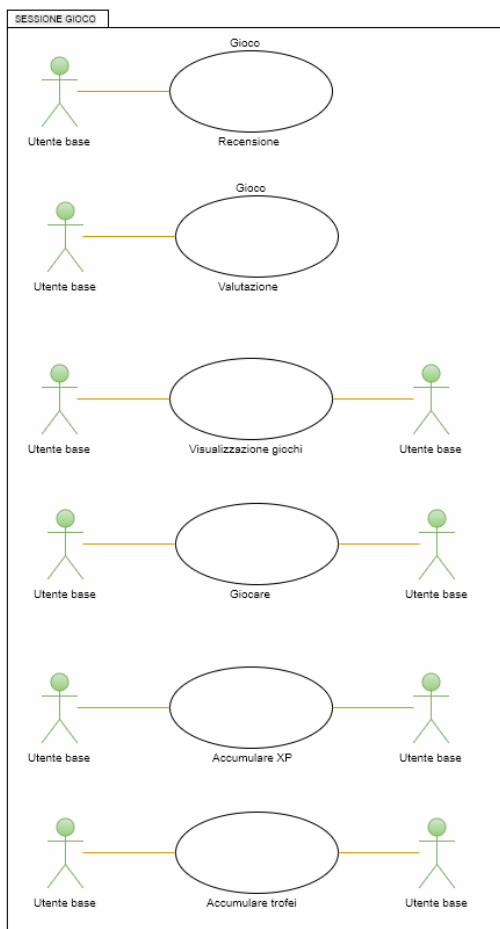
### SESSIONE PROFILO



## LOGIN

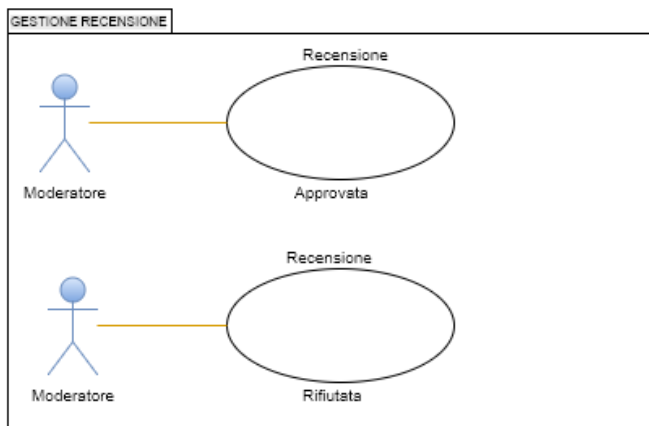
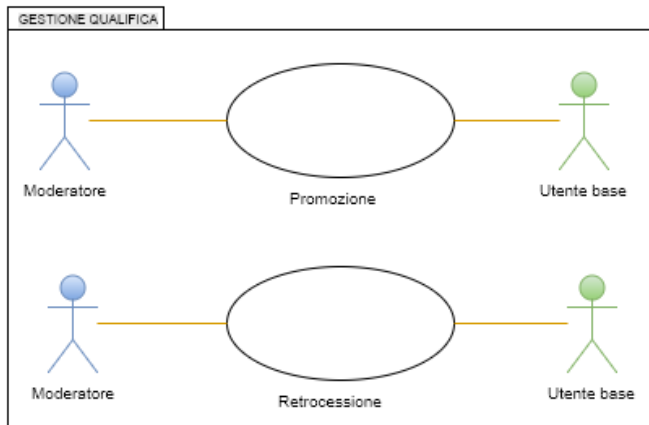


## SESSIONE GIOCO

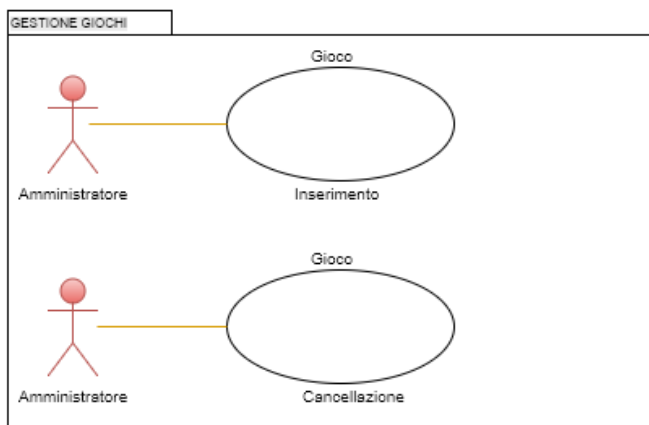
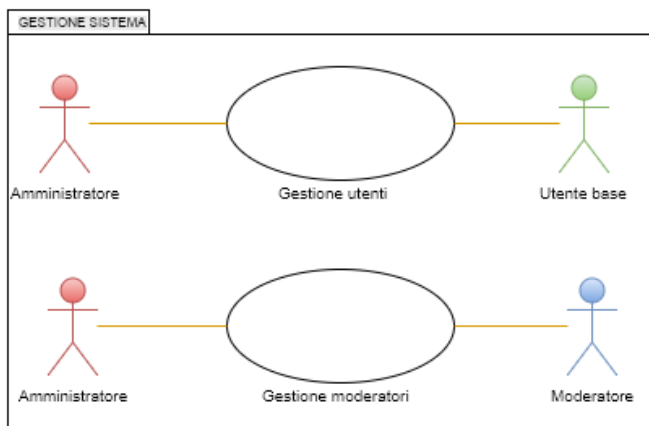


## MODERATORE

Il moderatore, oltre a possedere i cases dell'Utente base, è esteso con i seguenti:

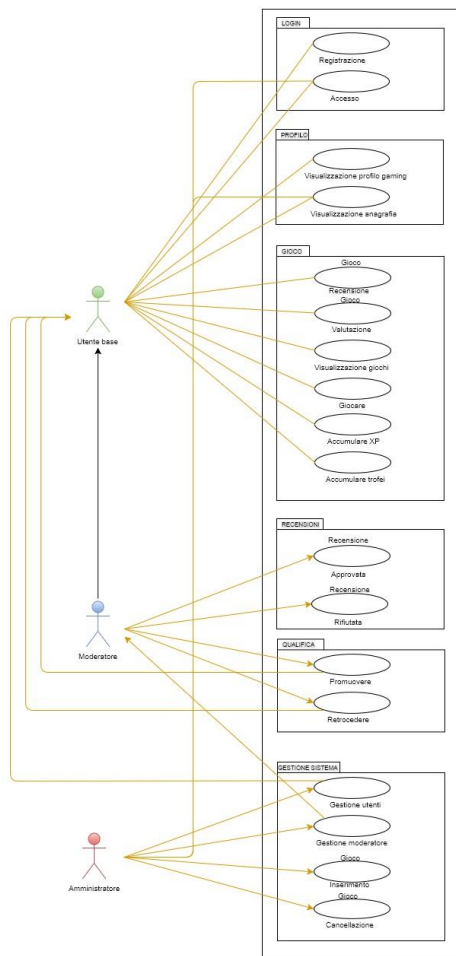


## AMMINISTRATORE





## GAMING PLAFORM



### SCENARI:

**REGISTRAZIONE:** un utente che vuole entrare a far parte della piattaforma ha la possibilità di registrarsi, inserendo dati anagrafici e username con la quale sarà riconosciuto in gioco

**LOGIN:** per accedere alla piattaforma, un qualsiasi attore dovrà inserire username e password compilati nella registrazione

**PROFILO PERSONALE:** ogni attore può visualizzare il proprio profilo personale, che registra i dati anagrafici dello stesso; in questa sezione si potranno modificare i dati inseriti

**PROFILO GAMING:** ad ogni moderatore ed utente base è associato un profilo gaming, il quale registra i progressi svolti nella piattaforma; in particolare saranno riportati il livello, i punti exp, la lista dei trofei ottenuti e la timeline

**GAMING:** i moderatori e gli utenti base possono accedere alla lista giochi, dalla quale selezionarne uno per accedere alla pagina relativa. Da qui il giocatore potrà clickare sul tasto *Gioca* per effettuare una partita oppure potrà recensire o votare il gioco; inoltre sono riportate la media dei voti e le recensioni per tale gioco

**LIVELLO:** Il livello di un moderatore o di un utente base potrà solamente aumentare, di quantità pari all'esperienza fornita da ogni gioco al quale si partecipa

**RECENSIONE:** Il moderatore ha il compito di approvare le recensioni scritte dai giocatori, quindi una recensione non può essere pubblicata prima di essere approvata dal moderatore. Le recensioni scritte dai moderatori sono pubblicate in automatico

**SISTEMA:** L'amministratore ha il compito di garantire il buon funzionamento della piattaforma. La piattaforma deve funzionare correttamente sia per la sessione gaming sia per le attività di moderazione.

## SYSTEM DESIGN

---

### OVERVIEW

Il sistema sarà suddiviso nella parte server-side, composta dal database, e nella parte client-side, rappresentante l'applicazione che sarà utilizzata dagli attori.

---

### CLIENT-SIDE

In questa sezione il team si occupa della implementazione degli attori, della componente grafica dell'applicazione e della relazione client-server; si può perciò ricorrere al pattern MVC per implementare tale sistema.

Il Model View Controller è un pattern architetturale formato da, come suggerisce il nome, tre componenti:

- *Model*: contiene lo stato dell'applicazione, definisce la logica implementativa dei dati e delle operazioni che possono essere svolte su di essi; in Java, seguendo la politica della programmazione ad oggetti, incapsula le classi che definiscono le parti principali della piattaforma
- *View*: gestisce l'output dei dati, quindi tutto ciò che concerne la parte grafica dell'applicazione. In Java, sarà principalmente sfruttato il framework Swing. Data la scelta della strategia *pull model*, la *View* richiede gli aggiornamenti quando lo ritiene necessario ed inoltre delega al *Controller* l'esecuzione dei processi richiesti dall'utente dopo averne catturato gli input e la scelta delle eventuali schermate da presentare
- *Controller*: componente che gestisce le azioni dell'utente che attua sulla *View*, e le comunica al *Model*.

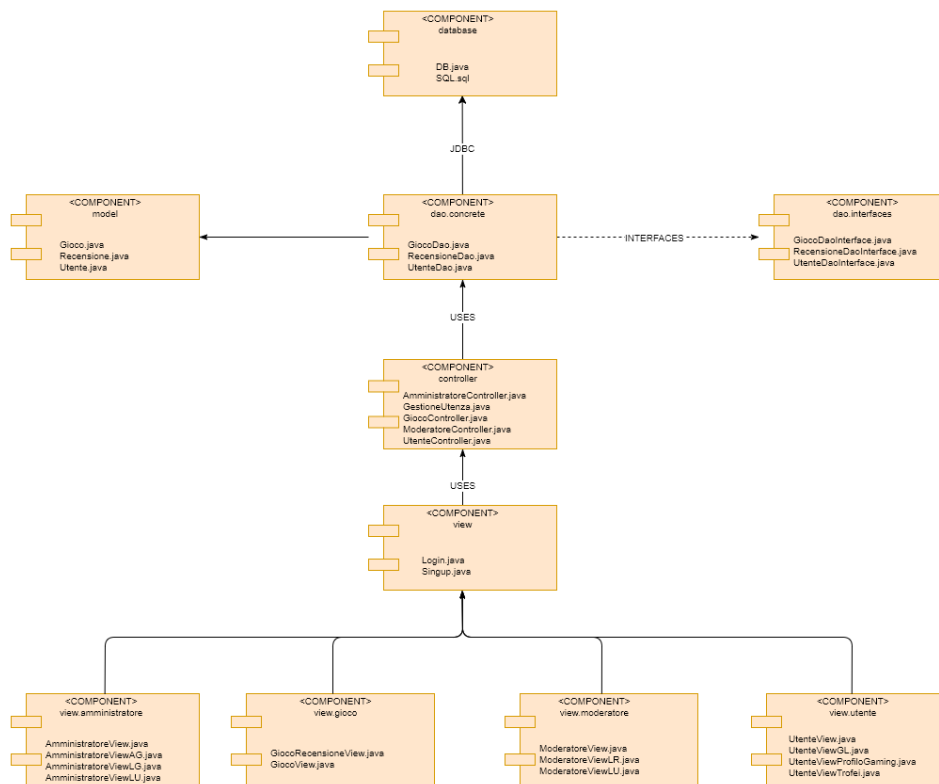
Nella piattaforma, ha inoltre il compito di richiamare i metodi descritti nelle classi DAO, quest'ultime implementate per interagire con il database

Presentiamo quindi un altro pattern, il Data Access Object, per la gestione della persistenza dei dati. In Java è implementato come un insieme di interfacce che specificano i metodi di cui le classi concrete effettueranno l'overriding, composti da queries che verranno eseguite dal DBMS.

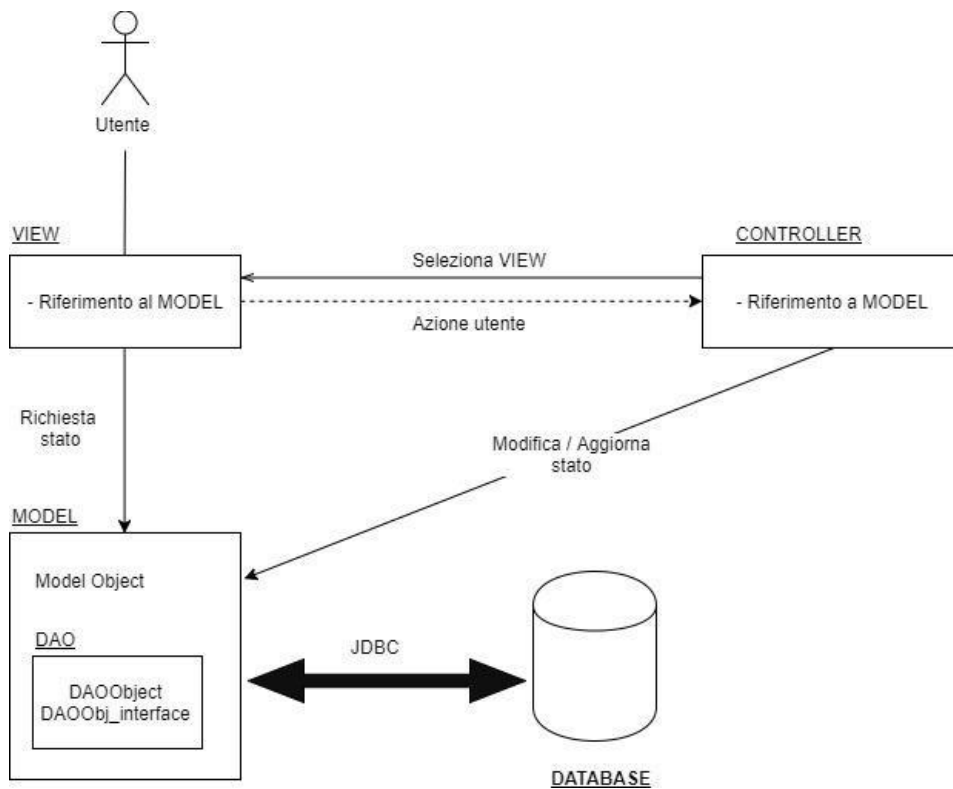
Per permettere il collegamento fra classi DAO e DBMS è stato impiegato un *Java DataBase Connector*, un driver che appunto consente l'accesso e la gestione dei dati su un DB dal programma Java che lo utilizza, indipendentemente dal tipo di DBMS impiegato.

L'implementazione dei patterns descritti garantisce un'ottima manutenibilità del sistema e raggiungere una maggiore astrazione dei componenti; di conseguenza, l'architettura del sistema è la seguente

## COMPONENT DIAGRAM

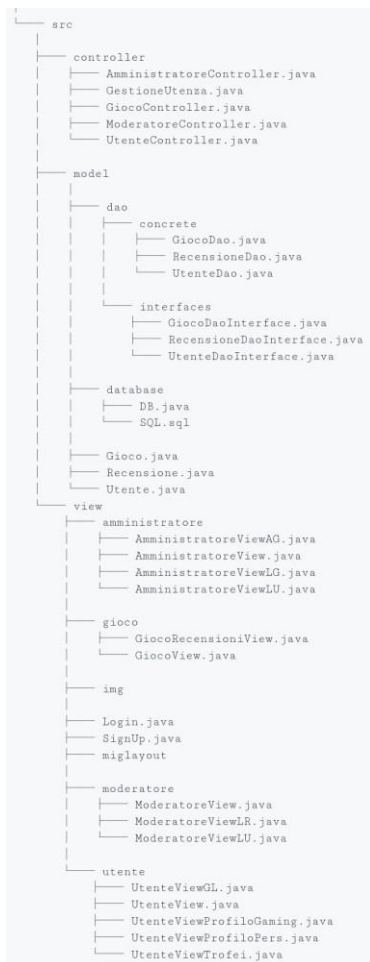


## GRAFICO DELLE INTERAZIONI FRA I DESIGN PATTERNS



## SOFTWARE DESIGN

L'implementazione della piattaforma è stata ideata in una suddivisione in moduli, per ampliare la manutenibilità del sistema e separare logicamente i componenti; in Java sono stati quindi impiegati diversi packages che incapsulano le relative classi.

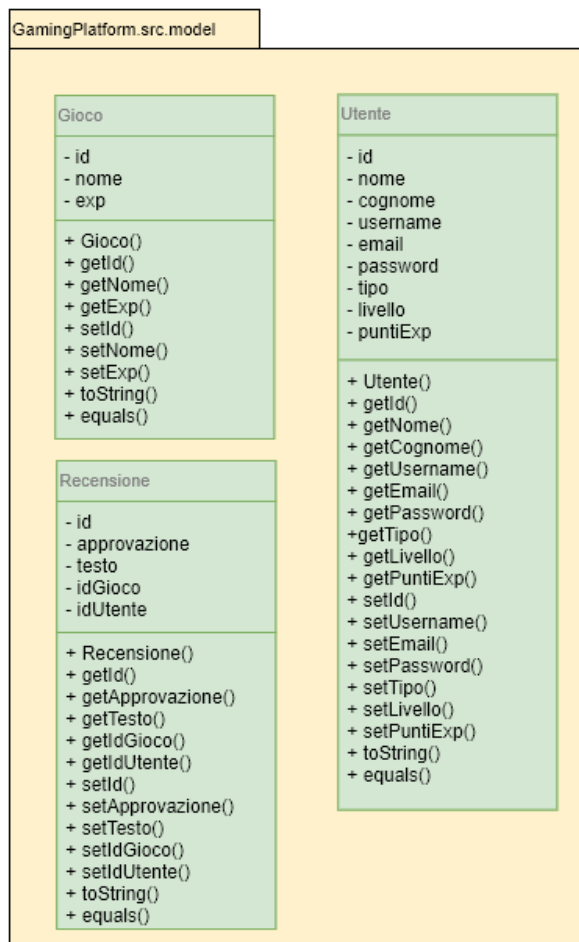


---

## CLASS DIAGRAMS

Sono un tipo di diagrammi di modello UML; li utilizziamo per rappresentare logicamente i package.

## MODEL'S CLASS DIAGRAM



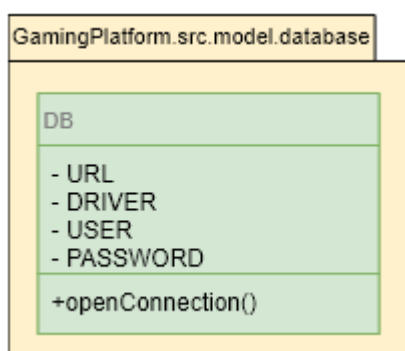
Come da preambolo, il precedente package incapsula il componente *Model* del pattern MVC: esso include le tre classi principali della piattaforma

- *Gioco*: ogni istanza determina appunto un gioco presente nella piattaforma caratterizzato dall'ID, dal nome e dai punti exp che fornisce. Sono a disposizione i metodi del costruttore, i getters ed i setters, nonché il metodo *toString()* che ritorna informazioni relative al gioco
- *Utente*: anche per questa classe, ogni istanza rappresenta un utente registrato nel database; è composto dai dati anagrafici e dai dati relativi alla piattaforma gaming, particolare attenzione all'attributo *Tipo* che specifica la natura dell'utente, quindi se amministratore, moderatore o utente base. Come nella classe precedente, sono disponibili i metodi del costruttore, i

getters e i setters, il metodo *toString()* ed *equals()* che controlla l'uguaglianza fra due Utenti

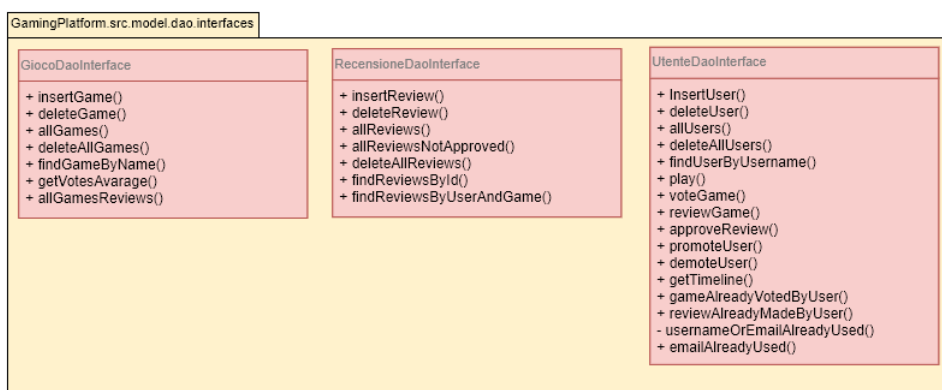
- *Recensione*: classe che modella il concetto logico di recensione; ha come attributi il proprio ID, un valore utilizzato dai moderatori per controllare se è stata approvata o meno, il testo e gli ID del giocatore che l'ha composta e il gioco che giudica. Anche qui notiamo il costruttore, i getters ed i setters, e il metodo informativo

## DB'S CLASS DIAGRAM



Classe adibita al DataBase, contiene l'URL, un riferimento al JDBC impiegato, i dati di autenticazione al DB e, di conseguenza, il metodo che stabilisce la connessione

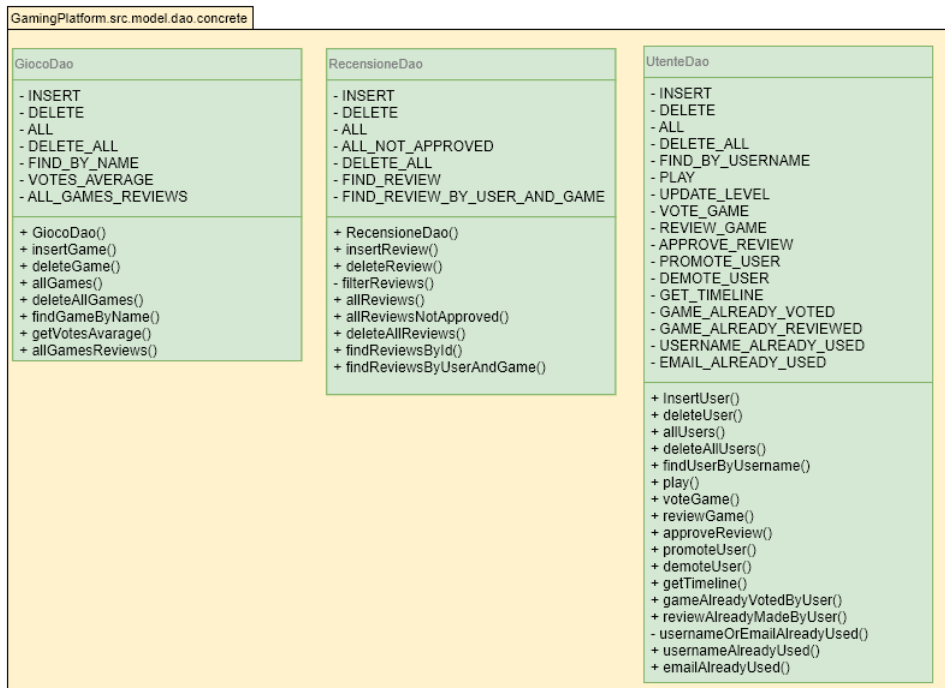
## DAO'S CLASS DIAGRAM – INTERFACES



Package che contiene le interfacce delle classi implementate per il pattern DAO; ognuna di esse specifica i metodi sfruttabili dalle classi concrete che le implementeranno per effettuare queries al DB

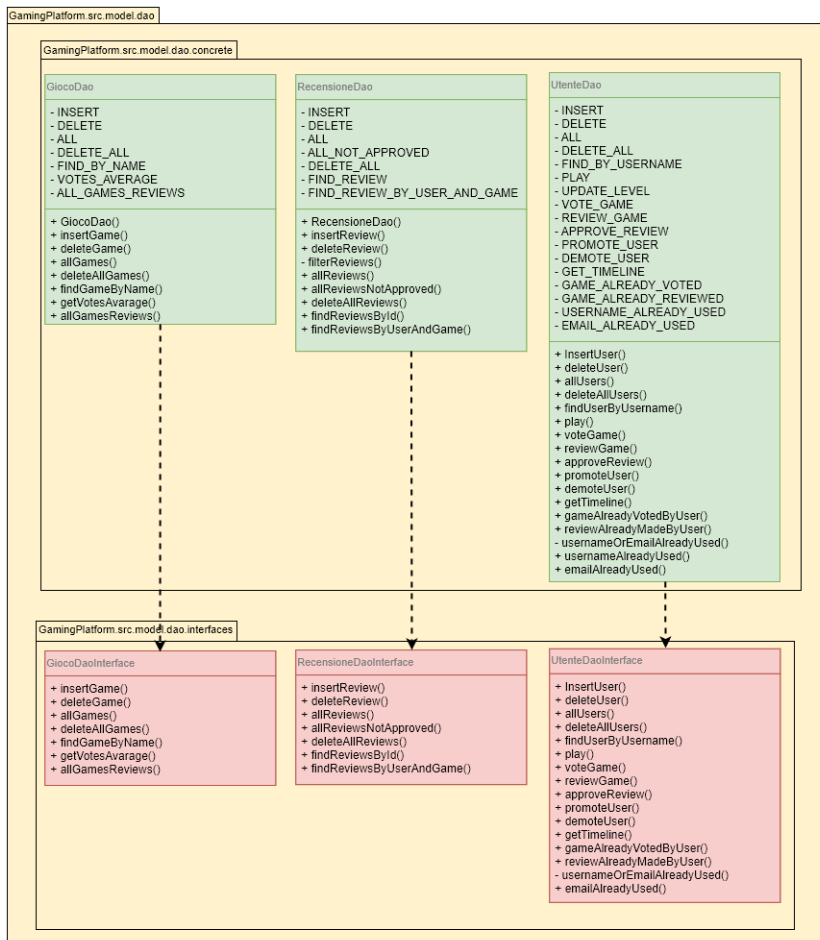


## DAO'S CLASS DIAGRAM – CONCRETE



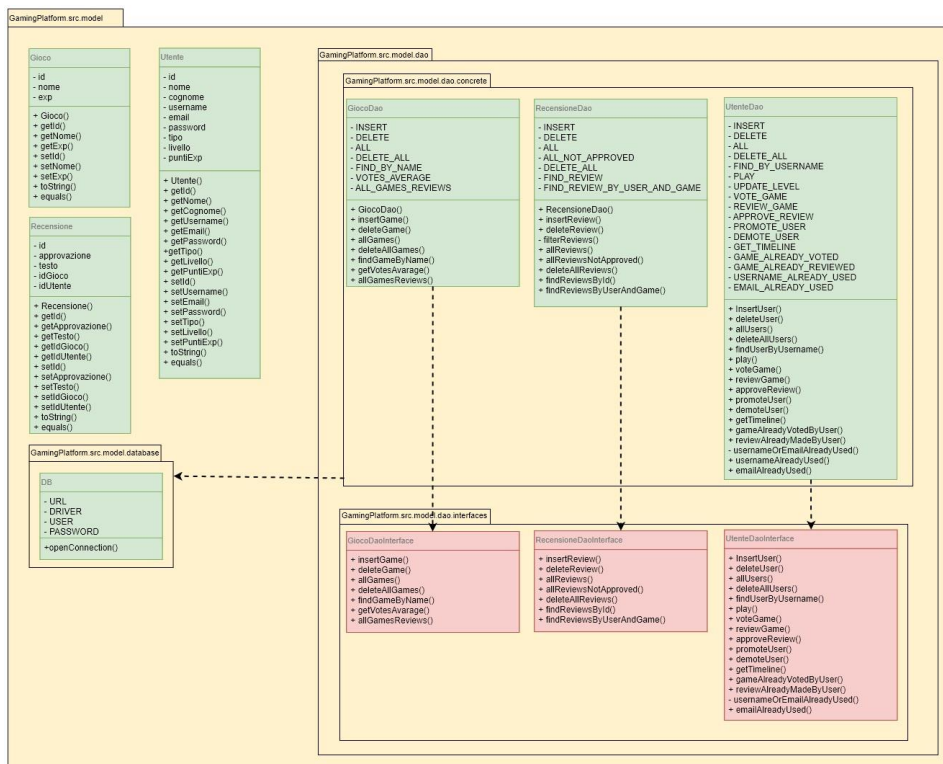
Come descritto nelle interfacce, queste classi implementano le DAO'S interfaces: negli attributi sono contenute le queries utili alla gestione del sistema, che saranno richiamate dai metodi specifici definiti nelle classi stesse

## DAO'S CLASS DIAGRAM



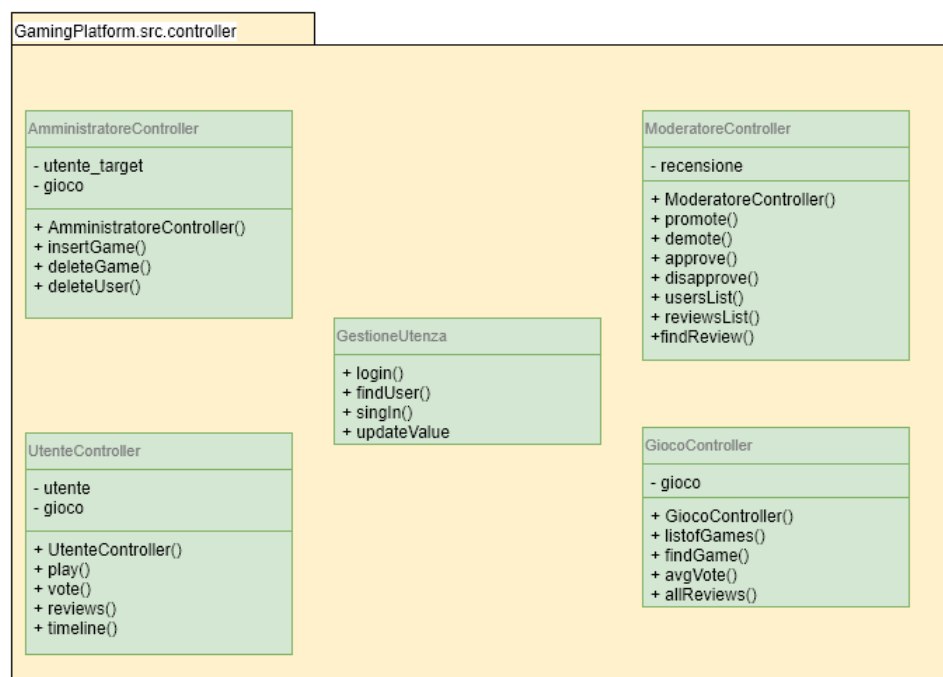
Class diagram che sottolinea il collegamento fra interfacce e classi concrete

## INTERAZIONI DAO-DB



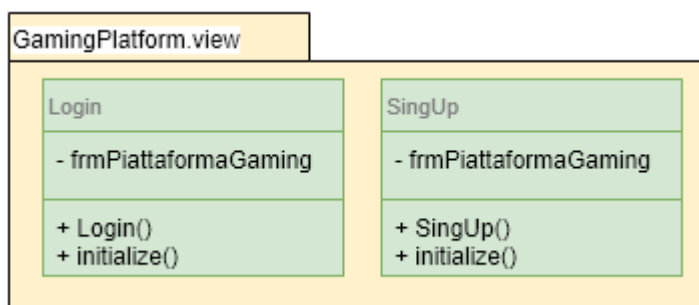
Link tra componente DAO e DB: chiaramente, per eseguire le suddette queries è necessario passare per il canale definito dalla classe *database*

## CONTROLLER'S CLASS DIAGRAM



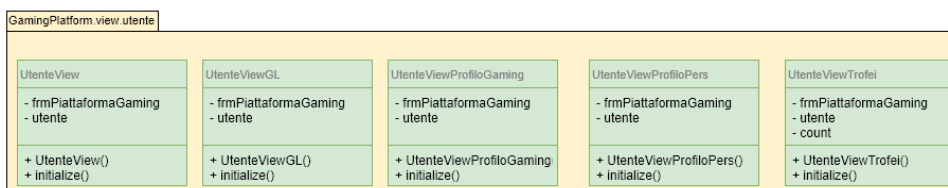
Il sopra illustrato package implementa il componente *Controller* del pattern MVC; qui viene sviluppata la logica di controllo, dunque la divisione in sottoelementi facilita l'implementazione. Ogni coppia <*Componente*>*Controller* incapsula le operazioni correlate a tale componente – i.e. *UtenteController* mette a disposizione i metodi per giocare, votare, recensire un gioco; *GiocoController* riguarda le azioni svolte su un gioco, come listare l'insieme dei giochi, ottenere una media dei voti relativi ad esso o la lista delle recensioni pubblicate -

## VIEW'S CLASS DIAGRAM



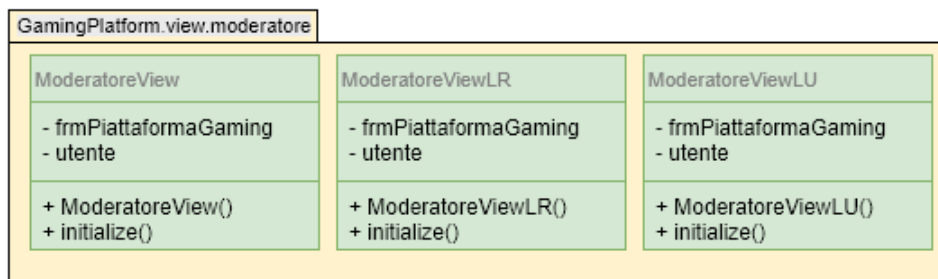
Seguendo la gerarchia dei package, abbiamo in un primo livello di view la GUI relativa alla registrazione e al login – in seguito sarà riportato un CD completo del componente *View*

## UTENTE'S VIEW'S CLASS DIAGRAM - VIEW' SUBPACKAGE -



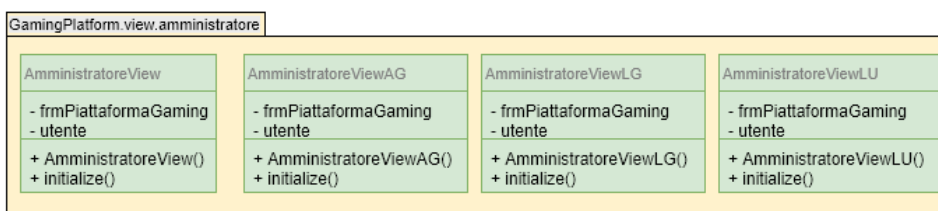
Questo package rappresenta l'ambito del *View* riguardante gli utenti base; in particolare *UtenteViewGL* implementa la grafica per lista giochi ai quali può partecipare; *UtenteViewProfiloGaming* la GUI per il profilo con i progressi svolti nella piattaforma; *UtenteViewProfiloPers* mostra i dati personali dell'utente, dalla quale possono anche essere modificati – richiamando metodi del controller -; *UtenteViewTrofei* è la bacheca dei premi ottenuti

## MODERATORE'S VIEW'S CLASS DIAGRAM - VIEW' SUBPACKAGE -



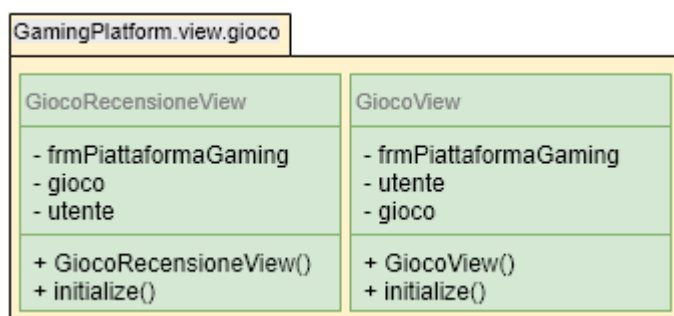
Come per l'utente base, è stata implementata la *View* per i moderatori, in cui *ModeratoreViewLR* mostra la lista delle recensioni da approvare, mentre *ModeratoreViewLU* la lista degli utenti da gestire per la promozione o retrocessione

## AMMINISTRATORE'S VIEW'S CLASS DIAGRAM - VIEW' SUBPACKAGE -



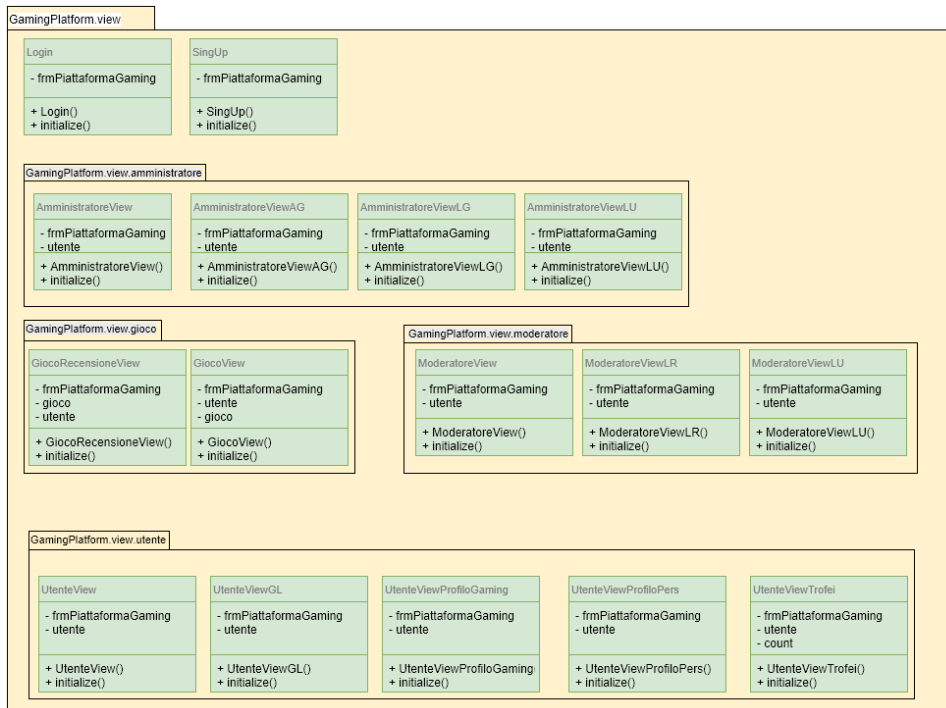
Implementazione della *View* per l'amministratore: *AmministratoreViewAG* rappresenta la GUI per l'aggiunta di un gioco dalla piattaforma; *AmministratoreViewLG* mostra la lista dei giochi, dalla quale si può effettuare la rimozione; *AmministratoreViewLU* è l'interfaccia che permette la visualizzazione degli utenti registrati alla piattaforma

## GIOCO'S VIEW'S CLASS DIAGRAM - VIEW' SUBPACKAGE -



GUI per la visualizzazione della pagina del gioco (*GiocoView*) e per la lista delle recensioni pubblicate (*GiocoRecensioneView*)

## FULL VIEW



Come anticipato, il CD intero per il componente *View* del pattern MVC

The diagram illustrates the architecture of the GamingPlatform application, organized into several interconnected components:

- GamingPlatform.src.controller**: Contains the core logic for handling requests.
  - Administrative\_Controller**: Manages user accounts (login, register, delete) and game reviews (create, update, delete).
  - Game\_Controller**: Manages game listings (create, update, delete) and game details (view, review).
  - Moderator\_Controller**: Manages game reviews (create, update, delete) and game details (view, review).
  - User\_Controller**: Manages user profiles (view, update, delete) and game reviews (create, update, delete).
- GamingPlatform.src.model**: Defines the data structures and relationships.
  - Game**: Attributes include title, genre, platform, price, rating, and review.
  - Game\_Review**: Attributes include game\_id, user\_id, rating, and review\_text.
  - User**: Attributes include username, password, email, and profile\_picture.
  - Game\_Review\_Reply**: Attributes include review\_id, user\_id, and reply\_text.
- GamingPlatform.src.model.dao**: Contains the Data Access Objects (DAOs) for interacting with the database.
  - Game\_DAO**: Methods for creating, updating, deleting, and retrieving game information.
  - Game\_Review\_DAO**: Methods for creating, updating, deleting, and retrieving game reviews.
  - User\_DAO**: Methods for creating, updating, deleting, and retrieving user information.
  - Game\_Review\_Reply\_DAO**: Methods for creating, updating, deleting, and retrieving game review replies.
- GamingPlatform.view**: Contains the user interface components.
  - Game\_View**: Displays game listings and details.
  - Game\_Review\_View**: Displays game reviews and replies.
  - User\_View**: Displays user profiles and game reviews.
- GamingPlatform.view.administration**: Contains the administrative interface components.
  - Game\_Administration\_View**: Displays game listings for administrators.
  - Game\_Review\_Administration\_View**: Displays game reviews for administrators.
  - User\_Administration\_View**: Displays user profiles for administrators.
- GamingPlatform.view.player**: Contains the player interface components.
  - Game\_Player\_View**: Displays game listings for players.
  - Game\_Review\_Player\_View**: Displays game reviews for players.
  - User\_Player\_View**: Displays user profiles for players.
- GamingPlatform.view.moderator**: Contains the moderator interface components.
  - Game\_Moderator\_View**: Displays game listings for moderators.
  - Game\_Review\_Moderator\_View**: Displays game reviews for moderators.
  - User\_Moderator\_View**: Displays user profiles for moderators.
- GamingPlatform.view.user**: Contains the user interface components.
  - Game\_User\_View**: Displays game listings for users.
  - Game\_Review\_User\_View**: Displays game reviews for users.
  - User\_User\_View**: Displays user profiles for users.
- GamingPlatform.src.model.database**: Contains the database schema and queries.
  - Game**: Table with columns for title, genre, platform, price, rating, and review.
  - Game\_Review**: Table with columns for game\_id, user\_id, rating, and review\_text.
  - User**: Table with columns for username, password, email, and profile\_picture.
  - Game\_Review\_Reply**: Table with columns for review\_id, user\_id, and reply\_text.
- GamingPlatform.src.model.dao.interface**: Defines the interfaces for the DAOs.
  - Game\_DAO\_Interface**: Methods for creating, updating, deleting, and retrieving game information.
  - Game\_Review\_DAO\_Interface**: Methods for creating, updating, deleting, and retrieving game reviews.
  - User\_DAO\_Interface**: Methods for creating, updating, deleting, and retrieving user information.
  - Game\_Review\_Reply\_DAO\_Interface**: Methods for creating, updating, deleting, and retrieving game review replies.

## DATABASE

È implementato come un database relazionale, perciò è composto da tabelle che modellano i concetti principali; la sua composizione è la seguente:



## LA TABELLA UTENTE:

id	nome	cognome	username	email	password	tipo	livello	puntiExp
1	Peter	Parker	Spiderman	uomoragno@avengers.com	ragnetto	moderatore	5	525
2	Tony	Stark	IronMan	starkcorporation@avengers.com	avengers	amministratore	3	100
3	Bruce	Banner	Hulk	incredibile@avengers.com	uomoverde	utente	2	30
7	p	p	p	p@p.it	p	utente	4	400
8	Oppa	Tze	oppa	oppa@tze.huppy	eheh	utente	0	0

## LA TABELLA GIOCO

id	nome	exp
1	Poker	30
2	Briscola	5
3	Asso piglia tutto	10
4	AcchiappaLaTalpa	20
7	Morra Cinese	70
8	LoL	100
10	Fifa	15
12	Sudoku	30
13	Scacchi	80
15	Dama	20

## LA TABELLA RECENSIONE

id	approvazione	testo	gioco	utente
9	1	Recensione di Spiderman:		
Pazzesci!		2	1	
10	1	Recensione di Spiderman:		
Ganzo		10	1	
11	1	Recensione di p:		
Gioco dell'anno		8	7	
12	1	Recensione di Spiderman:		
Dama è anche una donna		15	1	
13	1	Recensione di Spiderman:		
Hahhaha divertente		3	1	
14	1	Recensione di p:		
Noiosissimo		3	7	

## LA TABELLA VOTO

votazione	utente	gioco
3	1	1
4	2	2
1	2	4
5	7	8



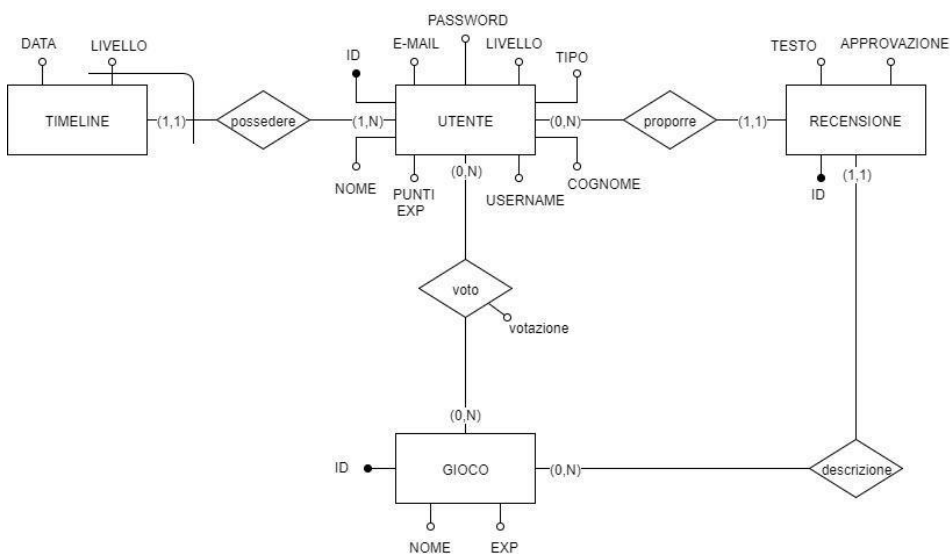
## LA TABELLA TIMELINE

data	livello	utente
2017-08-11	0	1
2017-08-14	1	1
2017-08-14	2	1
2017-08-14	3	1
2017-08-14	4	1
2017-08-14	5	1
1999-03-02	0	2
2003-06-06	1	2
2016-02-03	2	2
2017-08-11	3	2
2000-12-12	0	3
2017-03-28	1	3
2017-08-11	2	3
2017-08-12	0	7
2017-08-12	1	7
2017-08-12	2	7
2017-08-12	3	7
2017-08-14	4	7
2017-08-14	0	8

- Il codice sql è disponibile nel file SQL.sql, che contiene inoltre i triggers per gli aggiornamenti della timeline per la registrazione, la cancellazione e il raggiungimento di un nuovo livello -

Il database inoltre è lo strumento col quale l'amministratore ha completa gestione dei dati della piattaforma: da qui può svolgere altre attività, oltre quelle disponibili nell'applicazione tramite interfaccia grafica, come la query per promuovere un utente base a moderatore e viceversa.

## MODELLO E-R



## MODELLO RELAZIONALE

-Gli attributi sottolineati rappresentano le chiavi primarie dell'entità-

Utente (ID, Nome, Cognome, Username, Email, Password, Tipo, Livello, PuntiExp)

Gioco (ID, Nome, Exp)

Recensione (ID, Approvazione, Testo, *Gioco*, *Utente*)

Gioco e Utente chiavi esterne delle relative entità

Voto (Votazione, Utente, Gioco)

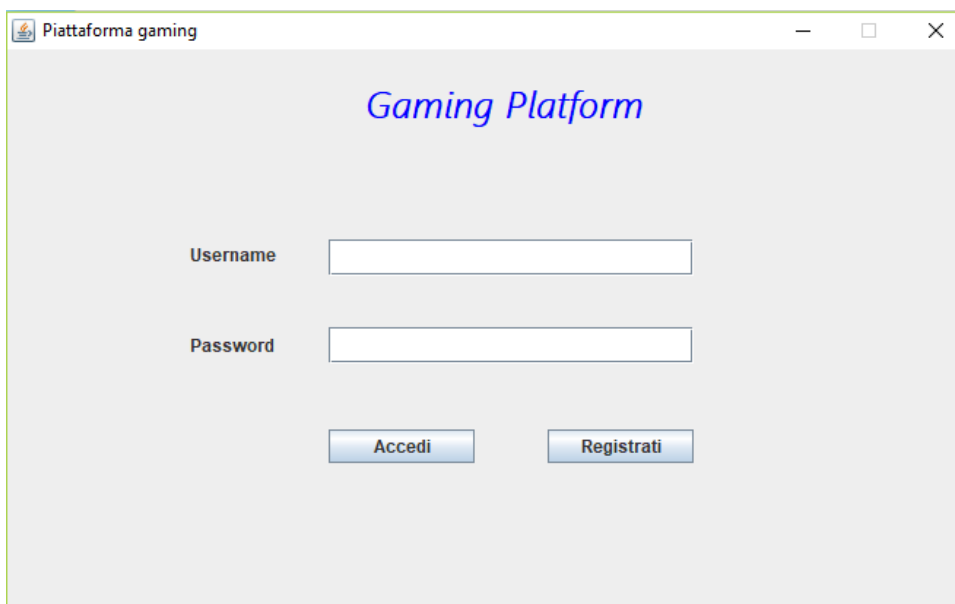
Utente e Gioco chiavi esterne delle relative entità

Timeline (Data, Livello, Utente)

Livello e Utente chiavi esterne delle relative entità

## IMMAGINI ESPLICATIVE DELLA PIATTAFORMA GAMING

### LOGIN

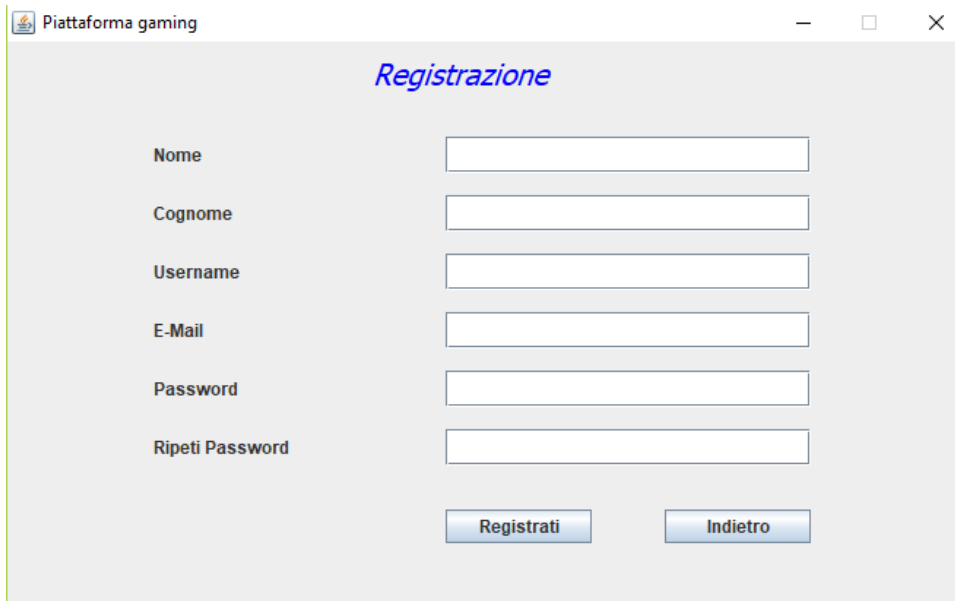


The screenshot shows a web browser window titled "Piattaforma gaming". The page has a light gray background and features the title "Gaming Platform" in a blue, italicized font at the top center. Below the title, there are two input fields: "Username" and "Password", each with a corresponding label to its left. At the bottom of the form, there are two buttons: "Accedi" (Login) and "Registrati" (Register), both with a blue gradient and white text. The window includes standard OS controls (minimize, maximize, close) in the top right corner.

Pagina di apertura dalla quale si può eseguire l'autenticazione se si è già registrati, altrimenti dal bottone *Registrati* si può accedere alla pagina di registrazione. I campi sono case-sensitive; in generale, se le stringhe non combaciano ad una

coppia username/password presente sul database, verrà mostrato un messaggio di errore nell'autenticazione

## REGISTRAZIONE

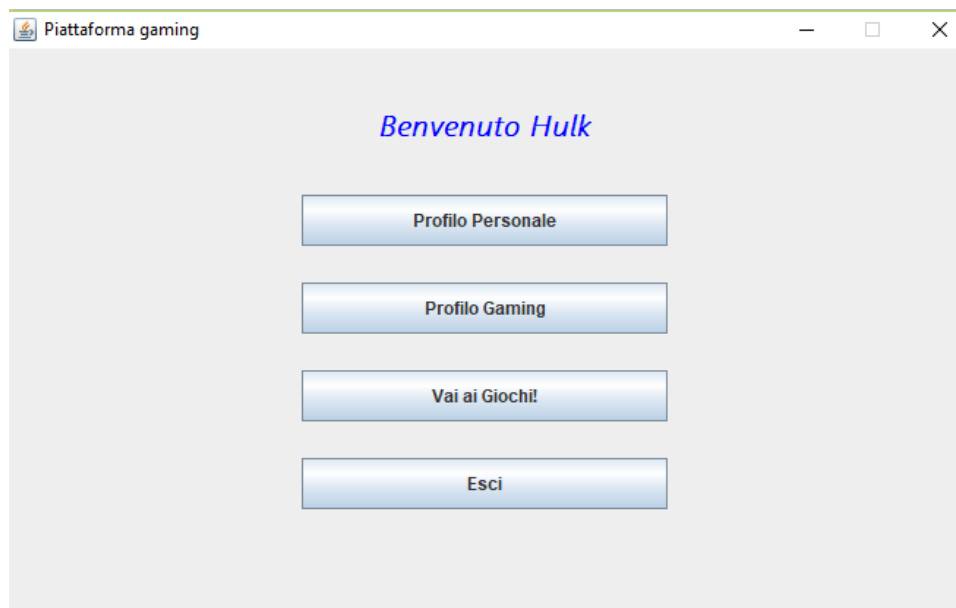


The screenshot shows a web application window titled "Piattaforma gaming". Inside, there is a section titled "Registrazione" in blue italicized text. Below the title, there are six input fields arranged vertically, each with a label to its left: "Nome", "Cognome", "Username", "E-Mail", "Password", and "Ripeti Password". At the bottom of the form, there are two buttons: "Registrati" and "Indietro".

Da questa schermata è possibile eseguire la Registrazione alla piattaforma; username ed E-Mail dovranno essere univoci, quest'ultima inoltre deve rispettare i caratteri di una mail modello (stringa@dominio.estensione)

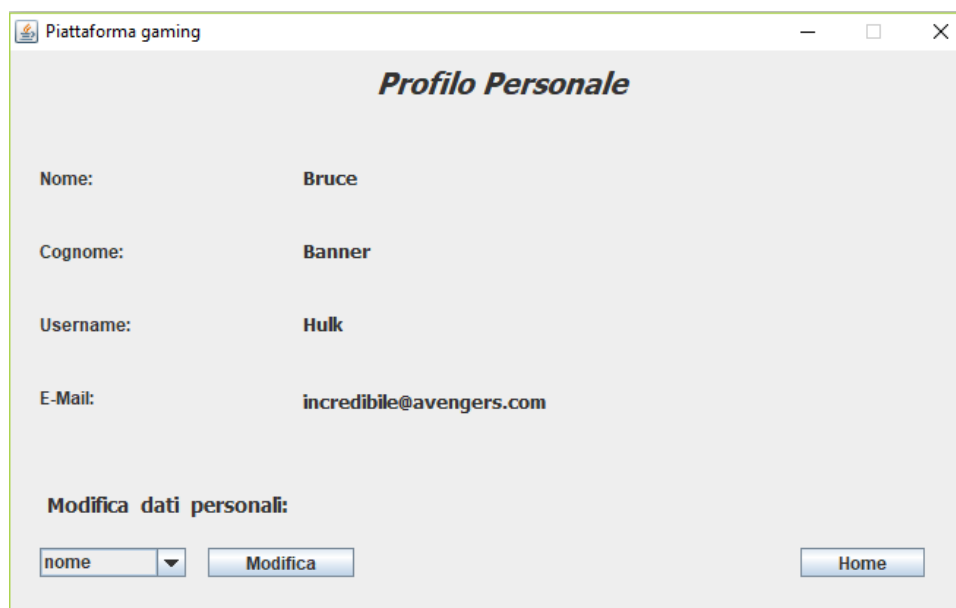
Ogni campo è soggetto a controllo sulla lunghezza per vincoli sul database; un messaggio di errore verrà visualizzato in caso di errore

## HOME UTENTE BASE



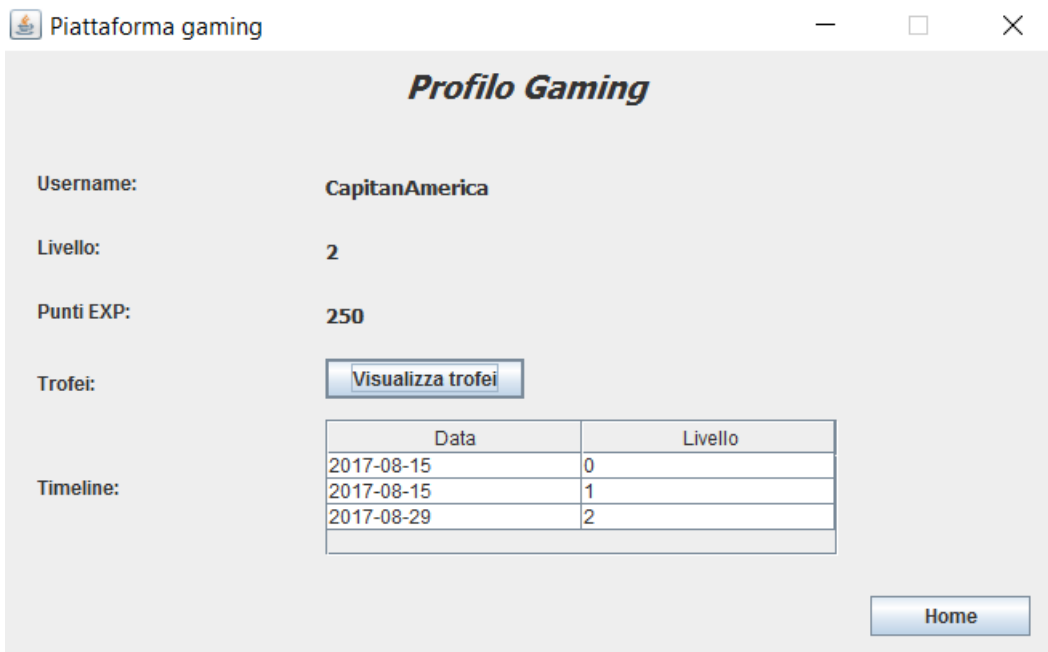
Vista della homepage Utente: come da nome, i bottoni permettono di accedere al Profilo personale, Profilo gaming, scelta del gioco, eseguire il logout

## PROFILO PERSONALE



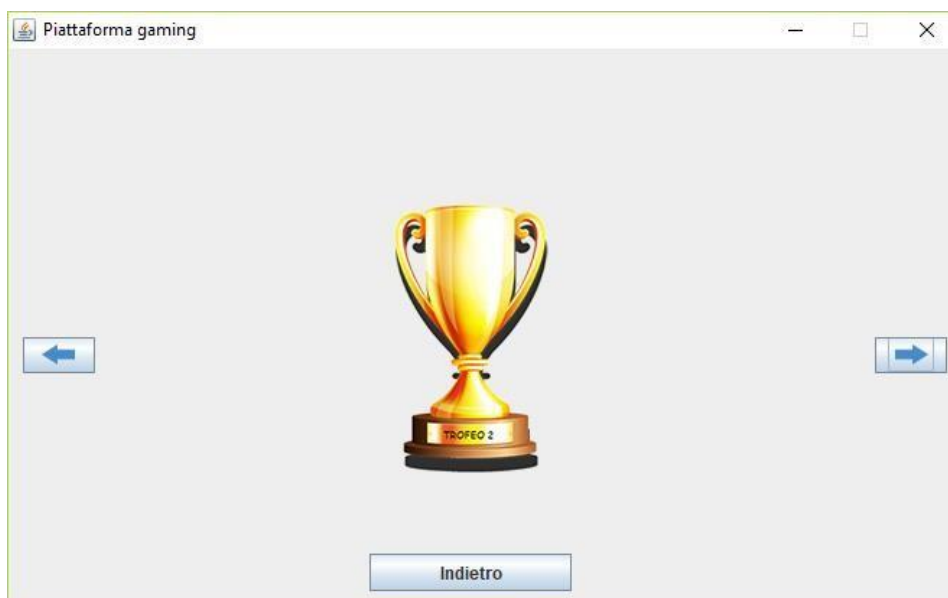
Pagina relativa al profilo personale: è uguale per tutti e tre gli attori. Sono mostrate le informazioni personali inserite al momento della registrazione; eventualmente possono essere modificate dal menù a tendina. Può essere anche cambiata la username, che è mostrata nel profilo gaming

## PROFILO GAMING UTENTE



Profilo che mostra i progressi nel gioco; anch'esso uguale per i tre attori

## VISUALIZZAZIONE TROFEI



Pagina per la visualizzazione dei trofei: se le frecce compaiono, vuol dire che ci sono ancora trofei da scorrere. Se non si è guadagnato nessun trofeo, non si potrà accedere alla pagina

## LISTA GIOCHI UTENTE



Lista giochi a disposizione dell'utente base e del moderatore – si ricorda che l'amministratore non prende parte ai giochi -; clickando su uno dei tasti nella lista si accede alla pagina relativa al gioco

## PAGINA GIOCO

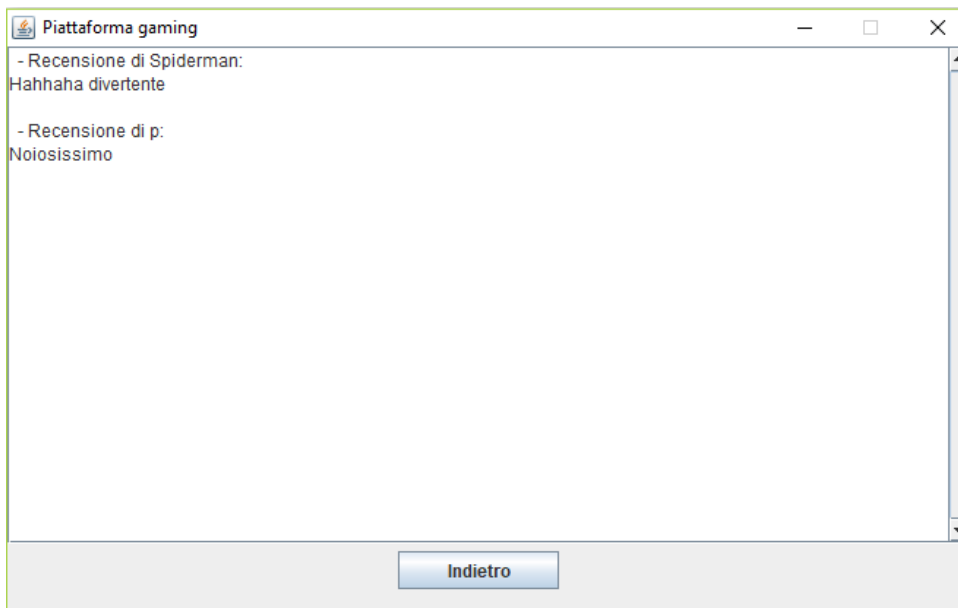


Ogni gioco ha la sua pagina esplicativa: contiene il nome, l'esperienza fornita ad ogni giocata, la media dei voti assegnati, i bottoni per accedere alla lista delle recensioni approvate, per assegnare un voto – scelto dall'opzione che può essere

incrementata tramite frecce in un range 1-5 -, per inviare la recensione composta nel campo testo, per giocare al gioco prescelto

---

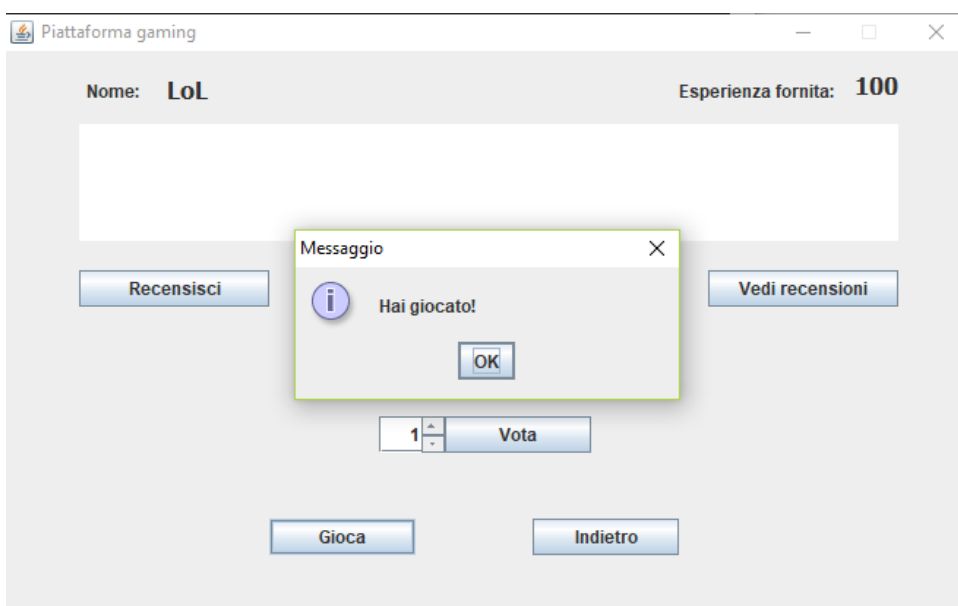
## LISTA RECENSIONI APPROVATE PER UN GIOCO



Il bottone *Vedi recensioni* della pagina gioco conduce a questa view, che contiene informazioni sullo scrittore ed il testo della recensione

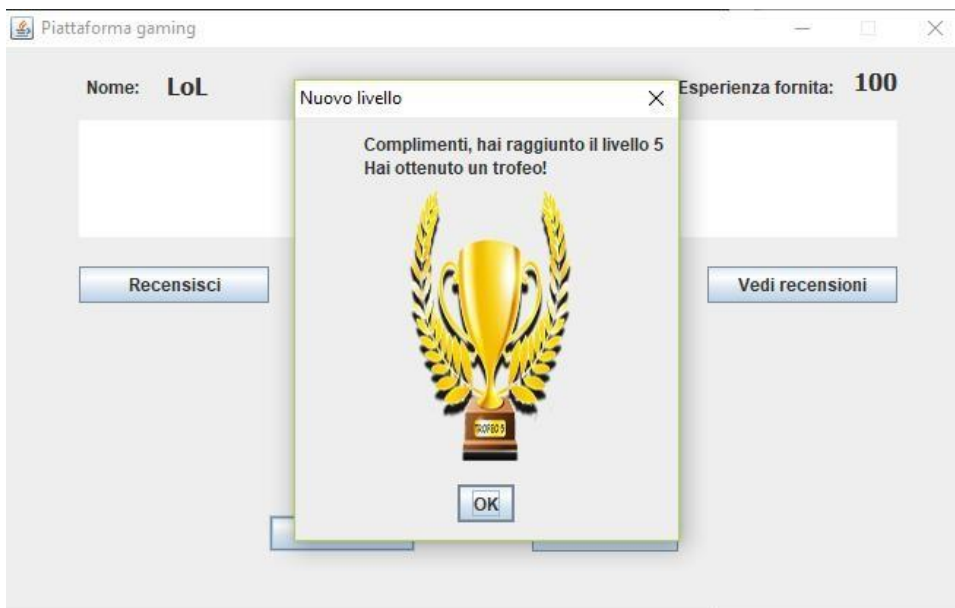
---

## GIOCARE AD UN GIOCO



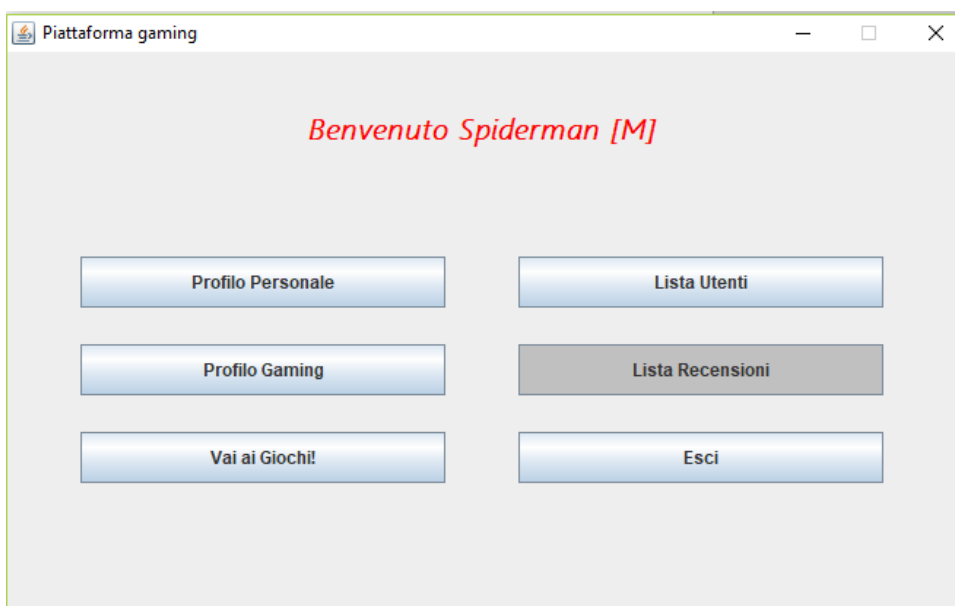
Il click sul bottone *Gioca* produce una bet, e il messaggio di avvenuta giocata avvisa l'aumento di exp dell'utente

## AVVISO AUMENTO DI LIVELLO



Se la giocata comporta l'aumento di livello, viene notificato tramite un messaggio e un'anteprima del trofeo guadagnato

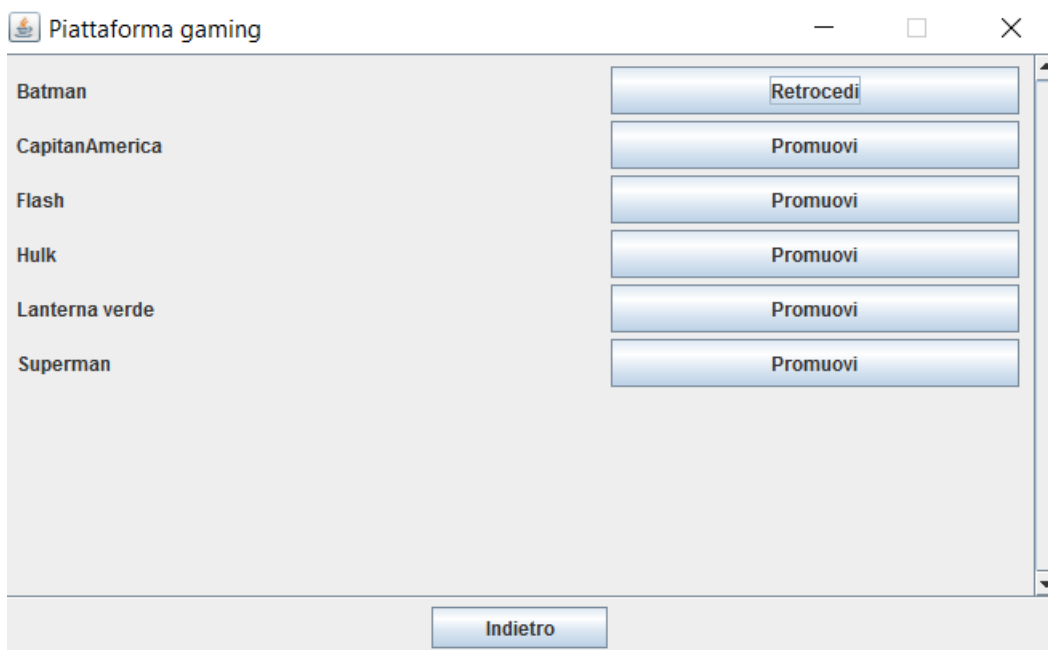
## HOME MODERATORE



Pagina di benvenuto del moderatore. Oltre alle features presenti anche nell'Utente base, esso ha a disposizione altri due bottoni: la lista utenti e la lista recensioni da approvare– nell'immagine non in sovrapposizione dal momento che non sono presenti recensioni da approvare



## LISTA UTENTI MODERATORE



La lista utenti visibile dal moderatore permette di promuovere a moderatore un utente base o, viceversa, retrocedere un moderatore ad utente base

## LISTA RECENSIONI DA APPROVARE



Se è presente una recensione da approvare, il tasto *Lista recensioni* presente nella homepage del moderatore potrà essere clickabile e porterà alla pagina con la lista delle recensioni. In quest'ultima, clickando su *Leggi*, si potrà controllare la recensione e scegliere se approvarla o negarla, come da immagine

## HOME AMMINISTRATORE



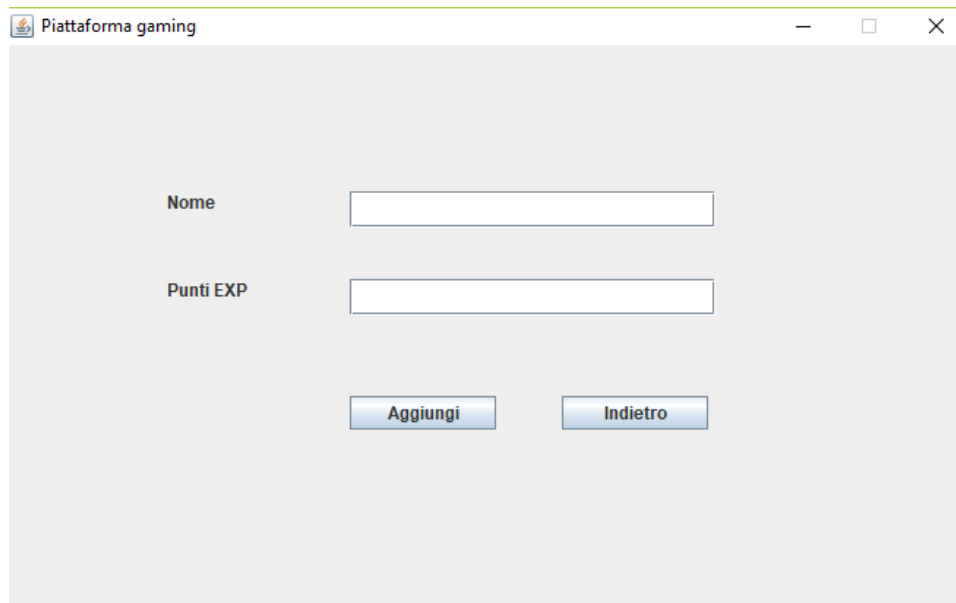
Homepage dell'amministratore

## LISTA GIOCHI AMMINISTRATORE



Il click sul bottone *Lista Giochi* nella homeview dell'amministratore produrrà la seguente schermata, composta dalla lista dei giochi presenti nella piattaforma, seguiti dal bottone *Elimina* per un'eventuale rimozione del gioco. Inoltre è presente il tasto *Inserisci* per poter effettuare l'inserimento di un nuovo gioco

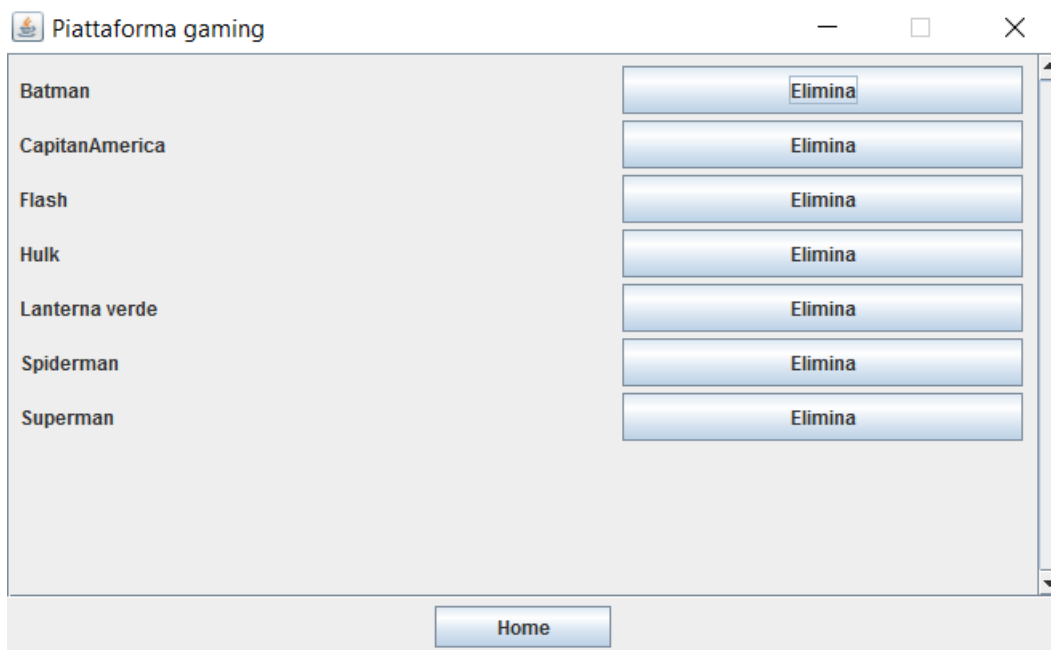
## INSERIMENTO NUOVO GIOCO



The screenshot shows a web application window titled "Piattaforma gaming". Inside, there is a form for adding a new game. It consists of two text input fields: the first is labeled "Nome" and the second is labeled "Punti EXP". Below these fields are two buttons: "Aggiungi" (Add) and "Indietro" (Back).

Il click infatti riporterà a questa schermata, nella quale si inseriranno il nome ed i punti exp del nuovo gioco; se il gioco è già presente, verrà segnalato un messaggio di errore

## LISTA UTENTI AMMINISTRATORE



The screenshot shows a web application window titled "Piattaforma gaming". Inside, there is a list of users with an "Elimina" (Delete) button next to each name. The users listed are Batman, CapitanAmerica, Flash, Hulk, Lanterna verde, Spiderman, and Superman. At the bottom of the window, there is a "Home" button.

Nome	Elimina
Batman	Elimina
CapitanAmerica	Elimina
Flash	Elimina
Hulk	Elimina
Lanterna verde	Elimina
Spiderman	Elimina
Superman	Elimina

Da questa lista l'amministratore potrà eliminare un utente dalla piattaforma