

Class Diagram

View



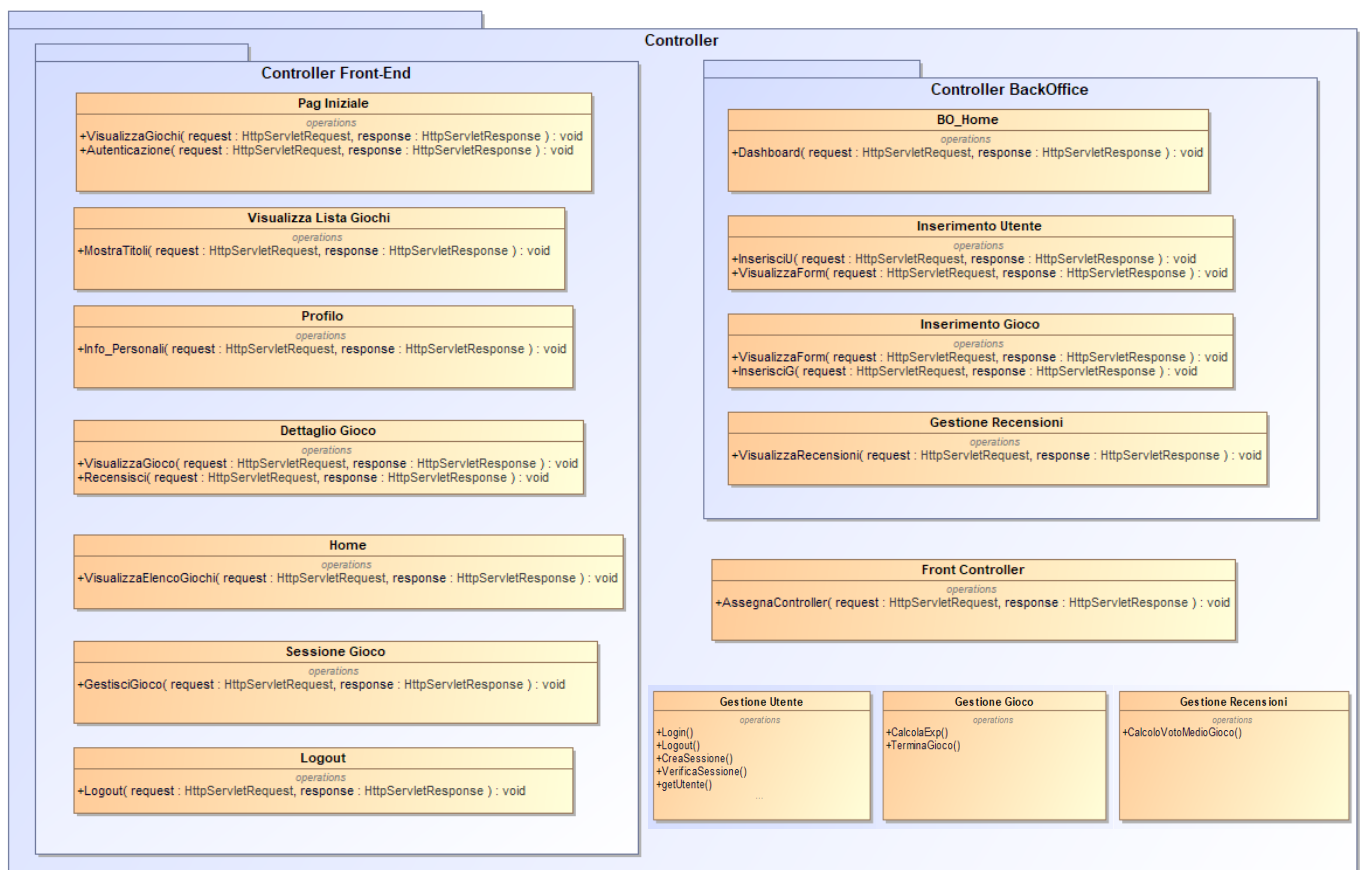
Questo Package corrisponde alla nostra componente View.

Rispettando il modello MVC adottato questa sezione risulta completamente separata dalla parte applicativa.

Per gestire l'interazione tra gli utenti e la piattaforma abbiamo deciso di utilizzare FreeMarker, un template engine basato su Java utilizzabile in un ambiente web.

All'interno della classe FreeMarker è presente un metodo chiamato MostraView al cui interno saranno presenti le opportune operazioni per recuperare il path della pagina da visualizzare e i dati da mandare; infine verrà richiamato il metodo process della libreria freemarker per mandare tutto a video.

Controller



Il controller fa da mediatore tra la View ed il Model; esso riceve i comandi dell'utente attraverso la View ed esegue le opportune operazioni che possono interessare il Model.

Nel nostro specifico caso, basandoci sul Front Controller Pattern, tutte le richieste da parte dell'utente arrivano al Front Controller.

Fornisce un punto di accesso centralizzato di tutte le request e le inoltra opportunamente alla servlet specifica di ogni pagina;

Abbiamo deciso di creare 2 packages per le Servlet dividendole in quelle che interessano il FrontEnd e quelle che invece operano nel BackOffice.

Le Servlet individuate nel FrontEnd sono:

- **Pag_Iniziale**
 - *VisualizzaGiochi* : Chiama la pagina relativa alla visualizzazione dei titoli.
 - *Autenticazione* : Gestisce le funzionalità di Login e Registrazione.
- **Visualizza_Lista_Giochi**
 - *Mostra_Titoli* : Metodo che permette di visualizzare la lista dei titoli dei giochi nella piattaforma.
- **Profilo**
 - *Info_Personali* : Funzione che permette di recuperare tutti i dati relativi all'utente in sessione quali: anagrafica, pti esperienza, livello, trofei.
- **Dettaglio_Gioco**
 - *Visualizza_Gioco* : Recupera i dati relativi al gioco selezionato: nome, descrizione, media dei voti del gioco calcolata con l'apposita funzione.
 - *Recensisci* : Funzione che permette di votare e inserire una recensione relativa al gioco.
- **Home**
 - *Visualizza_Elenco_Giochi* : Permette di recuperare l'elenco dei giochi e renderli visibili agli utenti.
- **Sessione_Gioco**
 - *Gestisci_Gioco* : Metodo relativo ad una sessione di gioco; calcola i pti esperienza ottenuti e termina la sessione aggiornando l'esperienza, il livello e i trofei con le relative funzioni.
- **Logout**
 - *Logout* : Termina la sessione di un utente.

Le Servlet individuate nel BackOffice sono:

- **BO_Home**
 - *Dashboard* : Recupera la lista degli utenti gestendone promozione e retrocessione; inoltre mostra la lista dei giochi con la funzionalità di eliminazione.
- **Inserimento_Utente**
 - *Visualizza_Form* : Mostra a video il form html da compilare per l'inserimento di un utente.
 - *Inserisci_U* : Metodo di inserimento di un nuovo utente.
- **Inserimento_Gioco**
 - *Visualizza_Form* : Mostra a video il form html da compilare per l'inserimento di un gioco.
 - *Inserisci_G* : Metodo di inserimento di un nuovo gioco.
- **Recensioni**
 - *Visualizza_Recensioni_Non_App* : Metodo che permette di recuperare l'insieme delle recensioni non approvate e mostrarle a video; gestisce inoltre l'approvazione o l'eliminazione di essa con i metodi appropriati.

All'interno del Controller sarà gestito anche il Front Controller attraverso l'omonima Servlet; il front controller si farà carico di tutte le richieste e le farà elaborare al Controller giusto.

- **Front_Controller**
 - *AssegnaController* : La funzione all'interno di questa Servlet andrà a ricavare l'URI della richiesta e richiamare il Controller specifico; in seguito caricherà l'implementazione di esso ed eseguirà una action per eseguirlo.

All'interno del package Controller sono inoltre presenti 3 classi contenenti tutti i metodi che possono risultare utili all'interno della piattaforma:

- **Gestione Utente**
 - *Login* : Controlla le credenziali inserite da un utente con i dati nel DB e torna un booleano.
 - *Crea_Sessione* : Metodo che permette di instanziare una sessione.
 - *Logout* : Funzione che chiude una sessione di un utente.
 - *Verifica_Sessione* : Metodo che verifica se la sessione è attiva o meno.
 - *Get_Utente* : Funzionalità che permette di restituire il tipo di dato utente in sessione.
- **Gestione Gioco**
 - *Calcola_Exp* : Funzione che calcola l'esperienza durante una sessione di gioco.
 - *Termina_Gioco* : Termina una sessione di gioco richiamando i metodi di aggiornamento esperienza, livelli e trofei.
- **Gestione Recensioni**
 - *CalcolaVotoMedioGioco* : Permette di calcolare una media di tutti i voti assegnati dagli utenti a quel gioco.

Model



Il Model è formato dagli oggetti principali della nostra applicazione; insieme alla componente DAO, che mostreremo di seguito, si occupa di recuperare ed archiviare informazioni da o verso il database.

