 **Università degli studi di Salerno**

**Corso di Ingegneria del Software**

****

*Partecipanti:*

|  |  |
| --- | --- |
| **Nome Cognome** | **Matricola** |
| **David Grieco** | **0512105112** |
| **Ciro Perfetto** | **0512105520** |
| **Silvio Esposito** | **0512105100** |
| **Fabrizio Trapani** | **0512105298** |

**Object Design Document (ODD)**

**1.Introduction**

* **1.1 OBJECT DESIGN TRADE-OFFS**

**Comprensibilità vs Tempo:** Il codice del sistema deve essere comprensibile, in modo da facilitare la fase di testing ed eventuali future modifiche da apportare. Al fine di rispettare queste linee guida il codice sarà integrato da commenti volti a migliorarne la leggibilità; tuttavia questo richiederà una maggiore quantità di tempo necessario per lo sviluppo del nostro progetto.

**Prestazioni vs Costi:** Dal momento che il budget allocato è spendibile principalmente in risorse umane e non consente l’acquisto di tecnologie proprietarie specifiche verranno utilizzati template open source e componenti hardware di nostra proprietà.

**Interfaccia vs Usabilità:** Verrà realizzata un’interfaccia grafica chiara e concisa, usando form e pulsanti predefiniti che hanno lo scopo di rendere semplice l’utilizzo del sistema da parte dell’utente finale.

**Sicurezza vs Efficienza:** La sicurezza rappresenta uno degli aspetti principali del sistema. Tuttavia, a causa di tempi di sviluppo molto limitati, ci limiteremo ad implementare sistemi di sicurezza basati su e-mail e password.

* **1.2 Linee guida per la documentazione delle interfacce**

Gli sviluppatori dovranno seguire precise linee guida per la stesura del codice:

**Naming Convention:**

Per la documentazione delle interfacce bisognerà utilizzare nomi:

* Descrittivi
* Pronunciabili;
* Di lunghezza medio-corta;
* Utilizzando solo caratteri consentiti (a-z, A-Z, 0-9)

**Variabili:**

* I nomi delle variabili dovranno iniziare con la lettera minuscola, e le parole successive con la lettera maiuscola.
* In ogni riga dovrà esserci un’unica variabile dichiarata, eventualmente allineata con quelle del blocco dichiarativo.
* In determinati casi, è possibile utilizzare il carattere underscore “\_”: il caso principale previsto è quello relativo alla dichiarazione di costanti oppure di proprietà statiche.

**Metodi:**

* I nomi dei metodi dovranno iniziare con la lettera minuscola, e le parole successive con la lettera maiuscola, secondo la “*Camel Notation*”.
* Il nome del metodo sarà costituito da un verbo che ne identifica l’azione seguito da un sostantivo, eventualmente aggettivato.
* Il nome dei metodi accessori e modificatori seguirà, rispettivamente, i pattern getNomeVariabile e setNomeVariabile.
* Ai metodi verrà aggiunto un commento JavaDoc, il quale deve essere posizionata prima della dichiarazione del metodo, e deve descriverne lo scopo.

**Classi Java e pagine JSP:**

* I nomi delle classi e delle pagine dovrà iniziare con la lettera maiuscola, così come le parole successive all’interno del nome.
* I nomi delle classi e delle pagine dovrà corrispondere alle informazioni e le funzioni fornite da quest’ultime.

Le classi saranno strutturate prevedendo rispettivamente:

1. Dichiarazione della classe pubblica;

2. Dichiarazione di costanti;

3. Dichiarazioni di variabili di classe;

4. Dichiarazioni di variabili di istanza;

5. Costruttore;

6. Metodi;

**Packages:**

* I nomi dei packages dovranno essere scritti in minuscolo concatenando insieme diversi sostantivi o sigle, separate dal carattere “.” .
* Non saranno ammessi caratteri speciali.

**2 Packages**

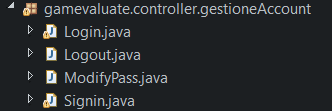
Il sistema è in diviso orizzontalmente in package come segue:

* *controller*
* *model*
* *bean*
* *connectionPool*
* *presentation*
* *testing*
* *utilities*
* *filters*
* **2.1 Controller**

Il package controller è suddiviso verticalmente in package come segue

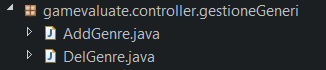
* gestioneAccount
* gestioneGeneri
* gestioneGiochi
* gestionePiattaforme
* gestioneRecensioni
* gestioneUtenti
* gestioneRicerca

2.1.1 gestioneAccount



|  |  |
| --- | --- |
| Classe | Descrizione |
| Login | Servlet che permette di effettuare il login |
| Logout | Servlet che permette di effettuare il logout |
| ModifyPass | Servlet che permette di modificare la password dell’account |
| Signin | Servlet che gestisce la registrazione di un nuovo account |

2.1.2 gestioneGeneri



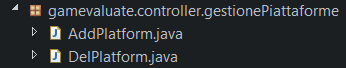
|  |  |
| --- | --- |
| Classe | Descrizione |
| AddGenre | Servlet che permette di aggiungere un genere al db |
| DelGenre | Servlet che permette di eliminare un genere dal db |

2.1.3 gestioneGiochi



|  |  |
| --- | --- |
| Classe | Descrizione |
| AddGame | Servlet che permette di aggiungere un nuovo gioco al db |
| DelGame | Servlet che permette di eliminare un gioco dal db |
| FillModifyGame | Servlet che permette di riempire la pagina di modifica di un gioco |
| ModifyGame | Servlet che permette di modificare le informazioni di un gioco |
| VoteGame | Servlet che permette di inserire una votazione inerente a un gioco |

2.1.4 gestionePiattaforme



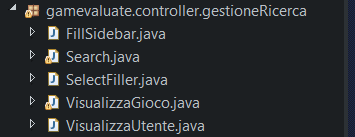
|  |  |
| --- | --- |
| Classe | Descrizione |
| AddPlatform | Servlet che permette di aggiungere una piattaforma al db |
| DelPlatform | Servlet che permette di eliminare una piattaforma dal db |

2.1.5 gestioneRecensioni



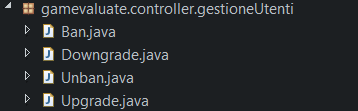
|  |  |
| --- | --- |
| Classe | Descrizione |
| AddReview | Servlet che permette di aggiungere una recensione testuale ad un gioco |
| DelReview | Servlet che permette di eliminare una recensione fatta ad un gioco |

2.1.6 gestioneRicerca



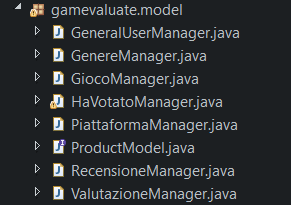
|  |  |
| --- | --- |
| Classe | Descrizione |
| FillSidebar | Servlet che permette di riempire la barra a scorrimento in base al ruolo dell’utente |
| Search | Servlet che gestisce la ricerca di un utente o di un gioco |
| SelectFiller | Servlet che permette di riempire le barre per selezionare i generi e le piattaforme presenti nel db |
| VisualizzaGioco | Servlet che permette di visualizzare la pagina informativa di un gioco |
| VisualizzaUtente | Servlet che permette di visualizzare la pagina informativa di un utente |

2.1.7 gestioneUtenti



|  |  |
| --- | --- |
| Classe | Descrizione |
| Ban | Servlet che permette di bannare di un utente |
| Downgrade | Servlet che permette di declassare un moderatore a utente |
| Unban | Servlet che permette di togliere il ban ad un utente |
| Upgrade | Servlet che permette di rendere moderatore un utente |

* **2.2 Model**

****

|  |  |
| --- | --- |
| Classe | Descrizione |
| GeneralUserManager | Classe che implementa le operazioni riguardanti l’account |
| GenereManager | Classe che implementa le operazioni riguardanti i generi |
| GiocoManager | Classe che implementa le operazioni riguardanti i giochi |
| HaVotatoManager | Classe che implementa le operazioni riguardanti le votazioni effettuati dagli utenti ai giochi |
| PiattaformaManager | Classe che implementa le operazioni riguardanti le piattaforme |
| ProductModel | Interfaccia che viene implementata da alcune classi dello stesso pacchetto |
| RecensioneManager | Classe che implementa le operazioni riguardanti le recensioni |
| ValutazioneManager | Classe che implementa le operazioni riguardanti le valutazioni |

* **2.3 Bean**



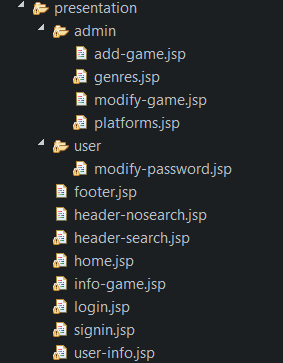
|  |  |
| --- | --- |
| Classe | Descrizione |
| GeneralUser | Classe che rappresenta le informazioni relative ad un utente |
| Genere | Classe che rappresenta le informazioni relative ad un genere |
| Gioco | Classe che rappresenta le informazioni relative ad un gioco |
| Piattaforma | Classe che rappresenta le informazioni relative ad una piattaforma |
| Recensione | Classe che rappresenta le informazioni relative ad una recensione |
| Valutazione | Classe che rappresenta le informazioni relative ad una valutazione |

* **2.4 ConnectionPool**

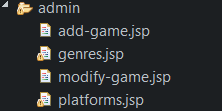


|  |  |
| --- | --- |
| Classe | Descrizione |
| DriverManagerConnectionPool | Classe che implementa l’Object pool pattern, responsabile di fornire le connessioni al database. |

* **2.5 Presentation**



* **2.5.1 presentetion admin**



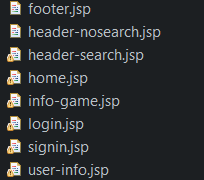
|  |  |
| --- | --- |
| Classe | Descrizione |
| add-game | Sezione in cui l’amministratore può aggiungere i giochi |
| genres | Sezione in cui l’amministratore può gestire i generi, aggiungendoli o eliminandoli |
| modify-game | Sezione in cui l’amministratore può modificare le informazioni di un gioco |
| platforms | Sezione in cui l’amministratore può gestire le piattaforme, aggiungendole o eliminandole |

* **2.5.2 presentation user**



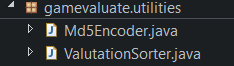
|  |  |
| --- | --- |
| Classe | Descrizione |
| Modify-password | Sezione in cui un utente registrato può modificare la propria password |

* **2.5.3 presentation**



|  |  |
| --- | --- |
| Classe | Descrizione |
| home | Sezione in cui l’utente/visitatore può :   * Cercare giochi * Cercare utenti * Filtrare i giochi   In questa sezione inoltre soltanto l’amministratore ha il diritto di eliminare un gioco |
| Info-game | Sezione in cui il visitatore può visualizzare le informazioni di un gioco e la lista delle recensioni  L’Utente in questa sezione può inserire una votazione e inoltre può aggiungere o eliminare una recensione  Il Moderatore e Amministratore possono elimiare qualsiasi recensione relativa al gioco |
| login | Sezione in cui si possono inserire le credenziali per accedere al sito |
| Sign-in | Sezione in cui si possono inserire le credenziali per registrarsi al sito |
| User-info | Sezione in cui i visitatori possono visualizzare le informazioni di un utente  L’utente può eliminare le proprie recensioni fatte a qualsiasi gioco  Il moderatore può eliminare una recensione o bannare un altro utente  L’ammistratore, oltre alle funzionalità elencate prima, può effetuare il dowgrade/upgrade di un utente |

* **2.6 Utilities**



|  |  |
| --- | --- |
| Classe | Descrizione |
| Md5Encoder | Classe che permette di convertire le password da salvare o modificare nel formato md5 |
| ValutationSorter | Classe che permette di ordinare le valutazioni |

* **2.7 Filters**



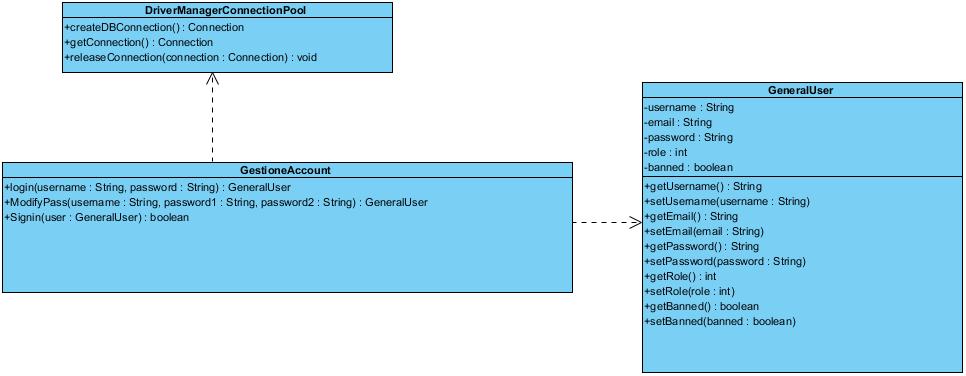
|  |  |
| --- | --- |
| Classe | Descrizione |
| AdminFilter | Classe che permette soltanto agli amministratori di accedere alle pagine a loro dedicate e di effettuare alcune operazioni riservate |
| FiltroGuest | Classe che gestisce a quali pagine possono accedere i visitatori e quali operazioni possono fare |
| ModeratorFilter | Classe che permette soltanto ai moderatori di compiere alcune operazioni a loro permesse |

**3 Class Interfaces**

* **3.1 Interfacce Bean**



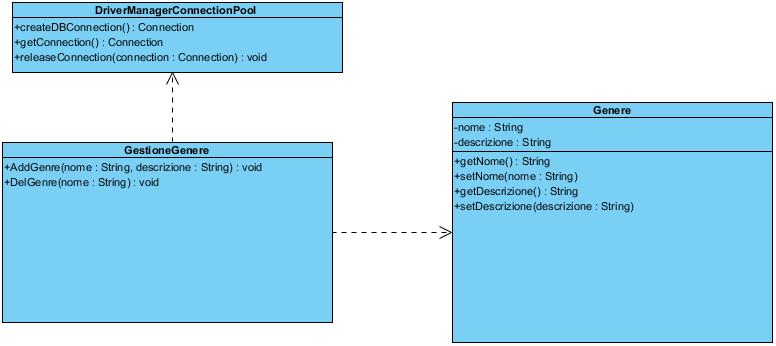
* **3.2 Gestione Account**

****

|  |  |
| --- | --- |
| Nome Classe | Gestione Account |
| Pre-condizione | **context** GestioneAccount::login(username: String, password : String) : GeneralUser **pre** username != null && password != null;  **context** GestioneAccount::modifyPassword(username: String, password1: String, password2: String) : GeneralUser **pre** username != null && password1 != null && password2 != null && password1 == password2;  **context** GestioneAccount::signin(user: GeneralUser) : boolean **pre** user != null; |
| Post-condizione | **context** GestioneAccount::modifyPassword(username: String, password1: String, password2: String) : boolean **post** GeneralUser.password = password1 |
| Invarianti |  |

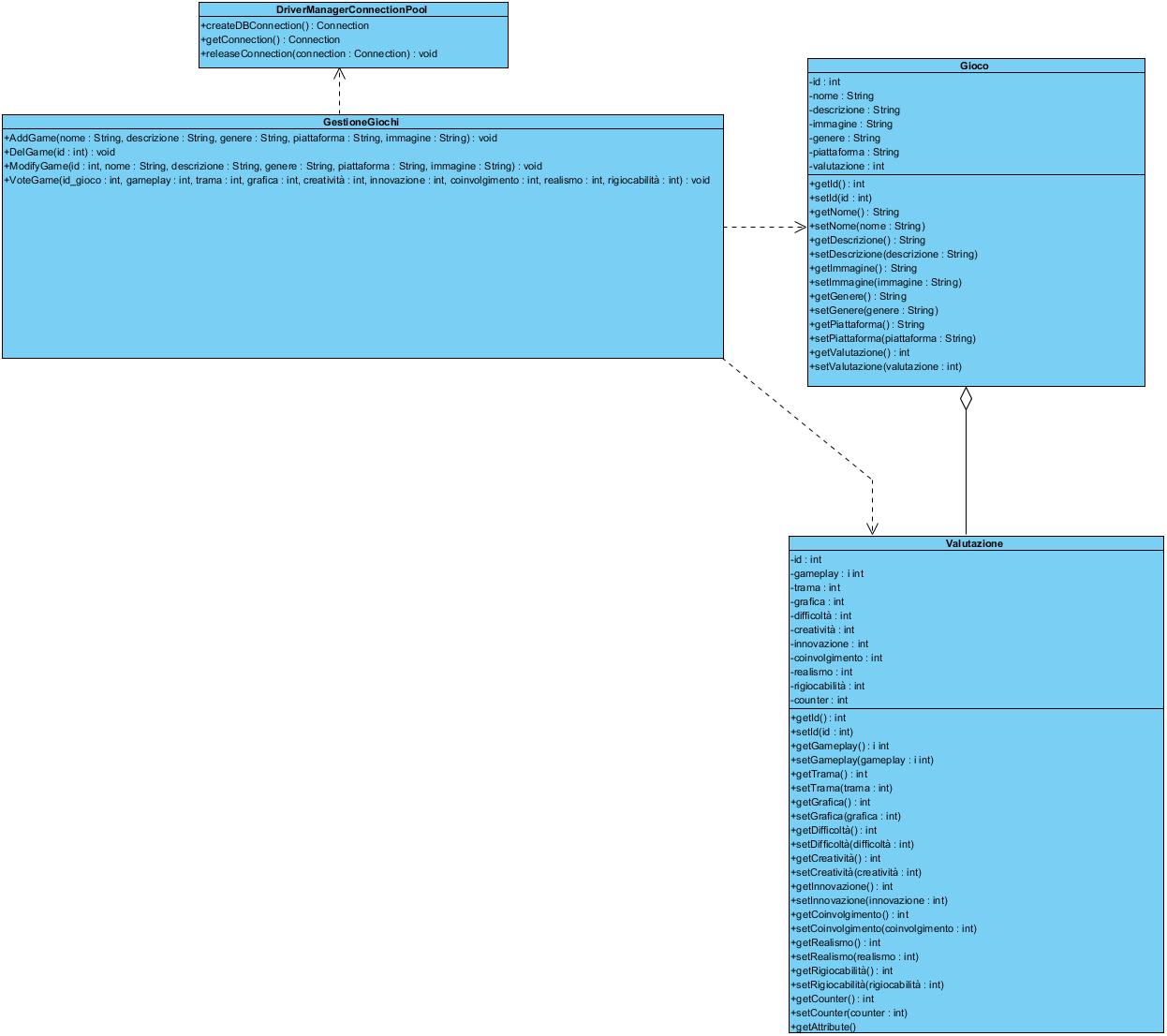
\*password1 e password2 vengono usati per indicare “password” e “conferma password”

* **3.2 Gestione Generi**



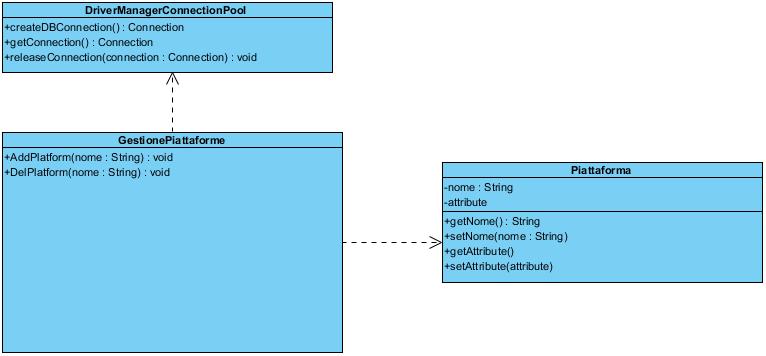
|  |  |
| --- | --- |
| Nome Classe | Gestione Generi |
| Pre-condizione | **context** GestioneAccount::AddGenre(nome: String, descrizione: String) : void **pre**: nome != null && descrizione != null;  **context** GestioneAccount::DelGenre(nome: String) : void **pre** nome != null; |
| Post-condizione | **context** GestioneAccount::AddGenre(nome: String, descrizione: String) : void **post**: generi.size() = @pre.generi.size() + 1  **context** GestioneAccount::DelGenre(nome: String) : void **post**: generi.size() = @pre.generi.size() - 1 |
| Invarianti |  |

* **3.3 Gestione Giochi**



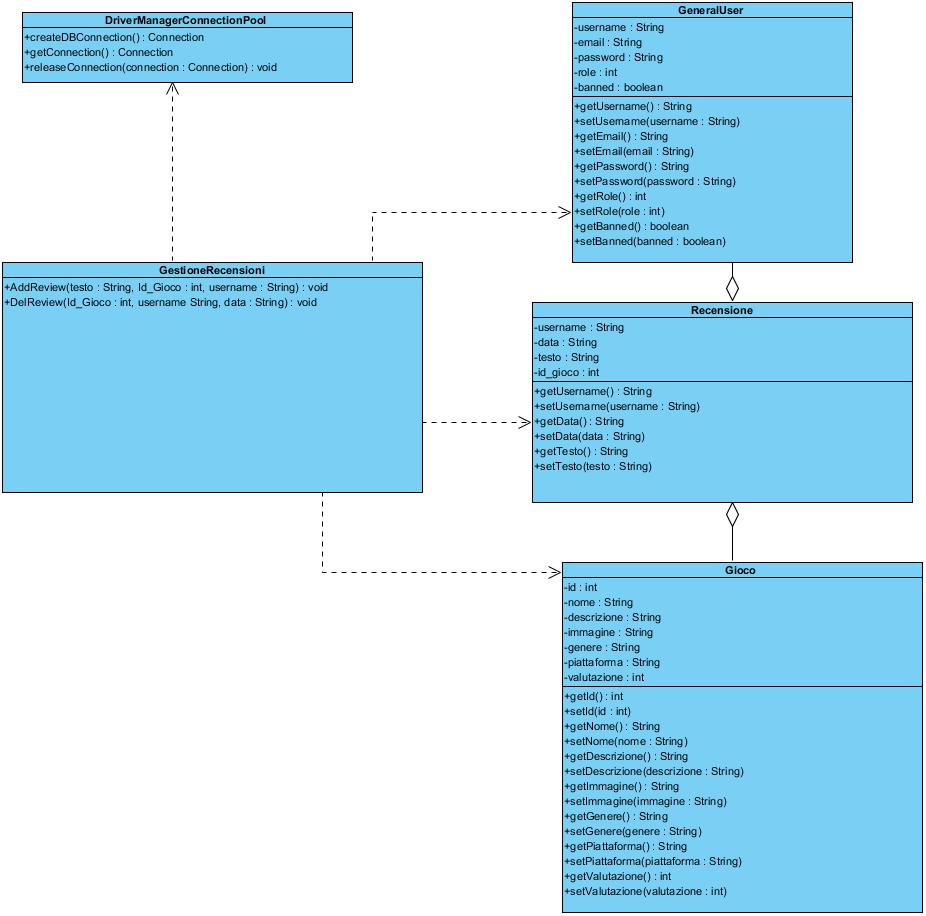
|  |  |
| --- | --- |
| Nome Classe | Gestione Giochi |
| Pre-condizione | **context** GestioneGiochi::AddGame(nome: String, descrizione: String, genere: String, piattaforma: String, immagine: String) : void **pre** : nome != null && descrizione != null && genere != null && piattaforma != null && immagine != null;  **context** GestioneGiochi::DelGame(id\_gioco: int) : void **pre**: id\_gioco!= null;  **context** GestioneGiochi::ModifyGame( nome: String, descrizione: String, genere: String, piattaforma: String, immagine: String) : void **pre** : nome != null && descrizione != null && genere != null && piattaforma != null && immagine != null;  **context** GestioneGiochi::VoteGame(id\_gioco : int, gameplay : int, trama : int, grafica : int, creatività : int, innovazione : int, coinvolgimento : int, realismo : int, rigiocabilità : int): void; |
| Post-condizione | **context** GestioneGiochi::AddGame(nome: String, descrizione: String, genere: String, piattaforma: String, immagine: String) : void **post**: giochi.size() = @pre.giochi.size() + 1;  **context** GestioneGiochi::DelGame(id\_gioco: int) : void **post**: giochi.size() = @pre.giochi.size() + -1;  **context** GestioneGiochi::ModifyGame(nome: String, descrizione: String, genere: String, piattaforma: String, immagine: String) : void **post**: gioco.nome = nome && gioco.descrizione = descrizione && gioco.genere = genere && gioco.piattaforma = piattaforma && gioco.immagine = immagine; |
| Invarianti |  |

* **3.4 Gestione Piattaforme**

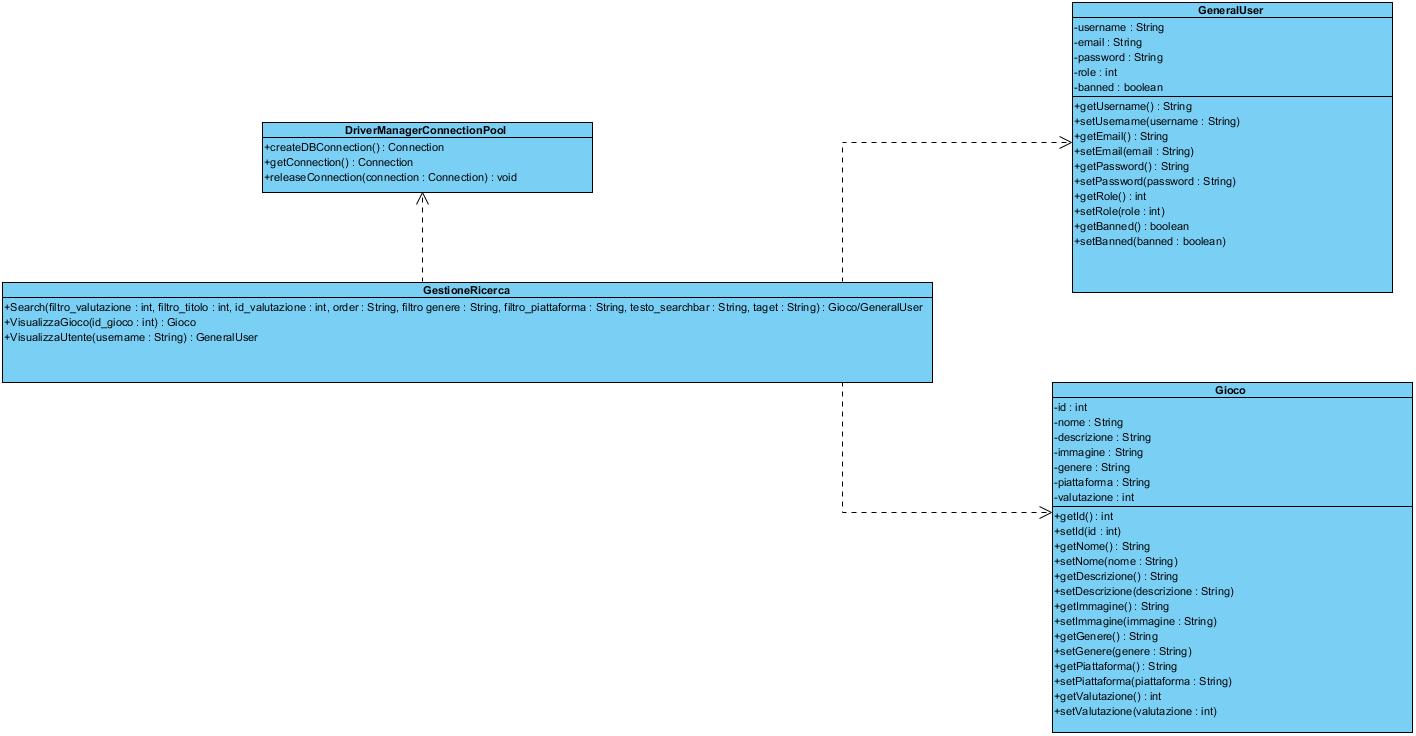
****

|  |  |
| --- | --- |
| Nome Classe | Gestione Piattaforme |
| Pre-condizione | **context** GestionePiattaforme::AddPlatform(nome: String) : void **pre** : nome != null;  **context** GestionePiattaforme::DelPlatform(nome: String) : void **pre** : nome != null; |
| Post-condizione | **context** GestionePiattaforme::AddPlatform(nome: String) : void **post**: piattaforme.size() = @pre.piattaforme.size() + 1;  **context** GestionePiattaforme::DelPlatform(nome: String) : void **post**: piattaforme.size() = @pre.piattaforme.size() – 1; |
| Invarianti |  |

* **3.5 Gestione Recensioni**

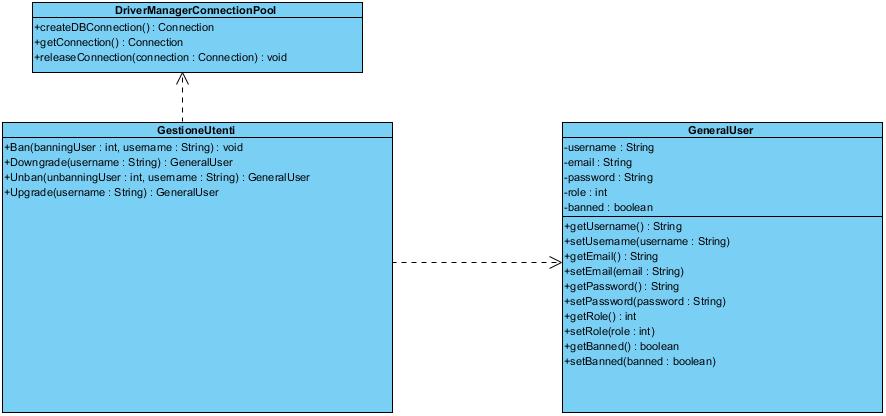


|  |  |
| --- | --- |
| Nome Classe | Gestione Recensioni |
| Pre-condizione | **context** GestioneRecensioni::AddReview(testo: String, id\_gioco : int, username : String) : void **pre** : nome != null && gioco != null && username != null;  **context** GestioneRecensioni::DelReview(data: String, id\_gioco : int, username : String) : void **pre** : data != null && gioco != null && username != null; |
| Post-condizione | **context** GestioneRecensioni::AddReview( testo: String, id\_gioco : int, username : String ) : void **post**: recensioni.size() = @pre.recensioni.size() + 1;  **context** GestioneRecensioni::DelReview(data: String, id\_gioco : int, username : String ) : void **post**: recensioni.size() = @pre.recensioni.size() - 1; |
| Invarianti |  |

* **3.6 Gestione Ricerca**

|  |  |
| --- | --- |
| Nome Classe | Gestione Ricerca |
| Pre-condizione | **context** GestioneRicerca::Search(filtro\_valutazione : int, filtro\_titolo : int, id\_valutazione : int, order : String, filtro genere : String, filtro\_piattaforma : String, testo\_searchbar : String, taget : String) : Gioco/GeneralUser **pre**: filtro\_valutazione != null && filtro\_titolo != null && id\_valutazione != null && order != null && filtro genere != null && filtro\_piattaforma != null && testo\_searchbar != null && taget != null;  **context** GestioneRicerca::VisualizzaGioco(id\_gioco: int) : Gioco **pre:** id\_gioco != null;  **context** GestioneRicerca::VisualizzaUtente(username: String): GeneralUser **pre:** username != null; |
| Post-condizione |  |
| Invarianti |  |

**3.7 Gestione Utenti**

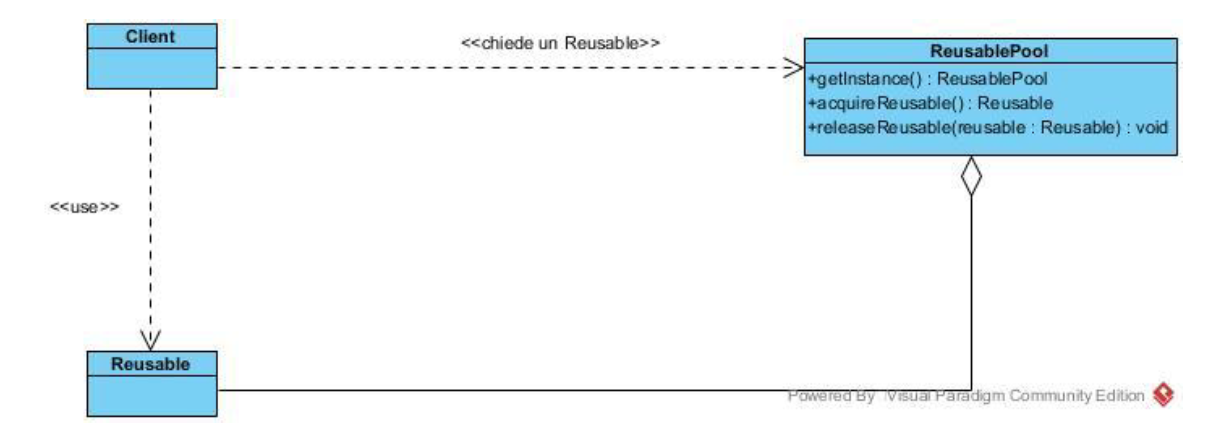


|  |  |
| --- | --- |
| Nome Classe | Gestione Utenti |
| Pre-condizione | **context** GestioneUtenti::Ban(banningUser: int, username : String): GeneralUser **pre:** banningUser != null && username != null;  **context** GestioneUtenti::Downgrade(username: String) GeneralUser **pre:** username != null;  **context** GestioneUtenti::Unban(unbanningUser: int, username : String): GeneralUser **pre:** unbanningUser != null && username != null;  **context** GestioneUtenti::Upgrade(username : String): GeneralUser **pre:** username != null; |
| Post-condizione | **context** GestioneUtenti::Ban(banningUser: int, username : String): GeneralUser **post:** GeneralUser.banned = true;  **context** GestioneUtenti::Downgrade(username : String): GeneralUser **post:** GeneralUser.role = User;  **context** GestioneUtenti::Unban(unbanningUser: int, username : String): GeneralUser **post:** GeneralUser.banned = false;  **context** GestioneUtenti::Downgrade(username : String): GeneralUser **post:** GeneralUser.role = Moderator; |
| Invarianti |  |

**4 Design Patterns**

L’Object pool pattern è un design pattern creazionale che usa un insieme di oggetti inizializzati pronti per l’uso mantenuti in una “pool”, che si occupa di allocarli e de-allocherà su richiesta. Il client della pool invierà richiesta a un oggetto nella pool ed eseguirà operazioni sull’oggetto ritornato. Quando il client ha finito, ritorna l’oggetto alla pool che lo de-allocherà.

*Object pool pattern*



Utilizzo: L’Object Pool Pattern sarà utilizzato per gestire le connessioni con il database. Più precisamente, un oggetto Model richiederà connessioni al DriverManagerPool che ritornerà un oggetto Connection, l’oggetto Model effettuerà operazioni con l’oggetto connection e successivamente richiederà al DriverManagerPool la de-allocazione.

