Object Oriented Software Design

Anno Accademico 2016-2017

Progetto
realizzato da

Formichetti Leonardo
242150

Ventimiglia Vittorio
242511

Petrucci Samuele
242385

Specifica Cliente

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

Una piattaforma online di gaming è quindi costituita da utenti, giochi e servizi. Lo scopo di questo progetto è quello di realizzare tale piattaforma in modo da garantire un'esperienza di gioco completa relativa ad un singolo gioco da parte di un utente.
Tale esperienza è comprensiva di vari ruoli e di vari servizi relativi al giocatore.

Il processo di gaming può suddividersi in diversi flow-of-events a seconda del servizio scelto dall'utente:

- Registrazione da parte dell'utente al portale
  L'utente avrà la possibilità di effettuare la registrazione al portale di gaming in modo da effettuare in un secondo momento la login e dare il via quindi alla propria esperienza di gioco.
- Visualizzazione del profilo e delle pagine dei relativi giochi. Sarà possibile visualizzare il proprio profilo con le relative informazioni anagrafiche e con i vari traguardi conquistati durante le sessioni di gioco. Sarà quindi possibile visualizzare la timeline con i livelli conquistati o il bilancio dei propri punti esperienza, che incrementando porteranno all'acquisizione di un nuovo livello.
- Possibilità di votare un gioco o esprimere un giudizio sullo stesso tramite recensione. Nella pagina relativa al gioco specifico sarà possibile,

oltre che giocare allo stesso, votare quest'ultimo con un punteggio compreso in un range variabile o effettuare una recensione che consentirà a nuovi utenti di sapere l'esperienza di gioco che li aspetta. Tali recensioni tuttavia saranno soggette a moderazione e solo dopo tale passo saranno rese pubbliche.

- Effettuare una sessione di gioco e collezionare punti esperienza ad ogni sessione. Ad ogni sessione di gioco l'utente vedrà incrementare la propria "balance" relativa ai propri punti esperienza. Tale attributo servirà allo stesso per conseguire nuovi livelli e arricchire il proprio profilo di nuovi traguardi.
- Collezionare trofei relativamente al livello conquistato. Ogni livello (eccetto quello iniziale) avrà associato il relativo "award". Tale trofeo, associato 1:1 con il relativo livello, sarà assegnato insieme a quest'ultimo al relativo profilo utente.

Analisi dei Requisiti

Attori del sistema:

- Admin
- Utente base
- Moderatore
- Database

Requisiti funzionali:

1. Funzionali all’amministratore (Admin):
    - Gestire richieste moderatore;
    - Accedere al database esterno;
    - Eliminare utenti base e/o moderatori dal sistema;
    - Accettare la prima richiesta di moderatore;
2. Funzionali all’utente:
    - Registrarsi al sistema;
    - Effettuare login nel sistema;
    - Visualizzare profilo;
    - Esperienza di gioco (tra una selezione di giochi disponibili);
    - Recensire giochi;
    - Acquisire punti esperienza e livelli;
    - Conquistare trofei in base alle vittorie;
    - Richiedere upgrade a Moderatore;

3. Funzionali al moderatore:
    - Promuovere utenti base a moderatori;
    - Retrocedere moderatori a utenti base;
    - Approvare o eliminare recensioni;

Requisiti non funzionali:

1. Requisiti di qualità:
    - Il sistema non costituisce un’applicazione web e pertanto non deve necessariamente essere dotato di sistemi di sicurezza raffinati;
    - L’affidabilità del sistema è garantita tramite un’efficiente gestione delle eccezioni, ovvero problemi che possono essere causati dal sistema stesso e/o dal database esterno;
    - Il sistema deve garantire usabilità verso l’utente base, pertanto facilità nell’uso delle funzionalità a lui destinate;

a questo proposito sarà dedicata attenzione alla semplicità delle interfacce grafiche e l'applicazione sarà dotata di un sistema di notifiche per aiutare l'utilizzatore.

2. Pseudo-requisiti:
    - Nell'implementazione del sistema (che verrà realizzata mediante l'uso del linguaggio Java) deve essere garantito l'utilizzo di:
        - Classi, package e modificatori di accesso per una corretta strutturazione del sistema
        - Interfacce
        - Ereditarietà
        - Polimorfismo e overriding/overloading di metodi con almeno un esempio di dynamic binding
        - Uso di final, static
        - Eccezioni
        - Collection/Map
        - JDBC
    - Il sistema deve fornire (come già specificato) un'interfaccia di facile comprensione per l'utente base che, in particolare, verrà realizzata tramite la libreria Swing di Java;

Scenarios

Scenario: Registrazione e Login

Nome Scenario	*Registrazione e Login*
Istanze Attori Coinvolti	*Mario Rossi*
Corso degli Eventi	○ *Mario si trova nella pagina di benvenuto.* ○ *Preme sul pulsante Registrati ed è condotto in un'apposita pagina dove può compilare un form per iscriversi.* ○ *Inserisce il suo Username, la sua Password, il suo Nome e Cognome e preme invio.* ○ *Una volta completata l'operazione, se tutto è andato a buon fine, viene riportato nella pagina Home.* ○ *Mario ora preme sul pulsante Login e compilando un form con le sue informazioni di registrazione accede al sistema.* ○ *Mario è finalmente condotto nella pagina del suo profilo appena creato.*

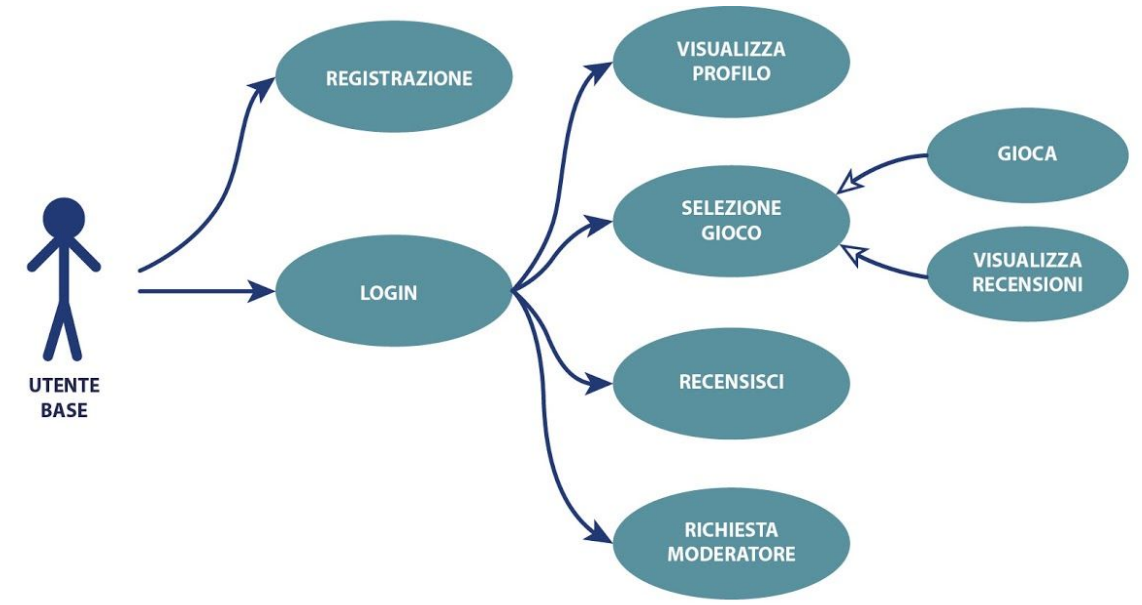
1. *Rappresentazione Tabellare dello scenario Registrazione e Login*

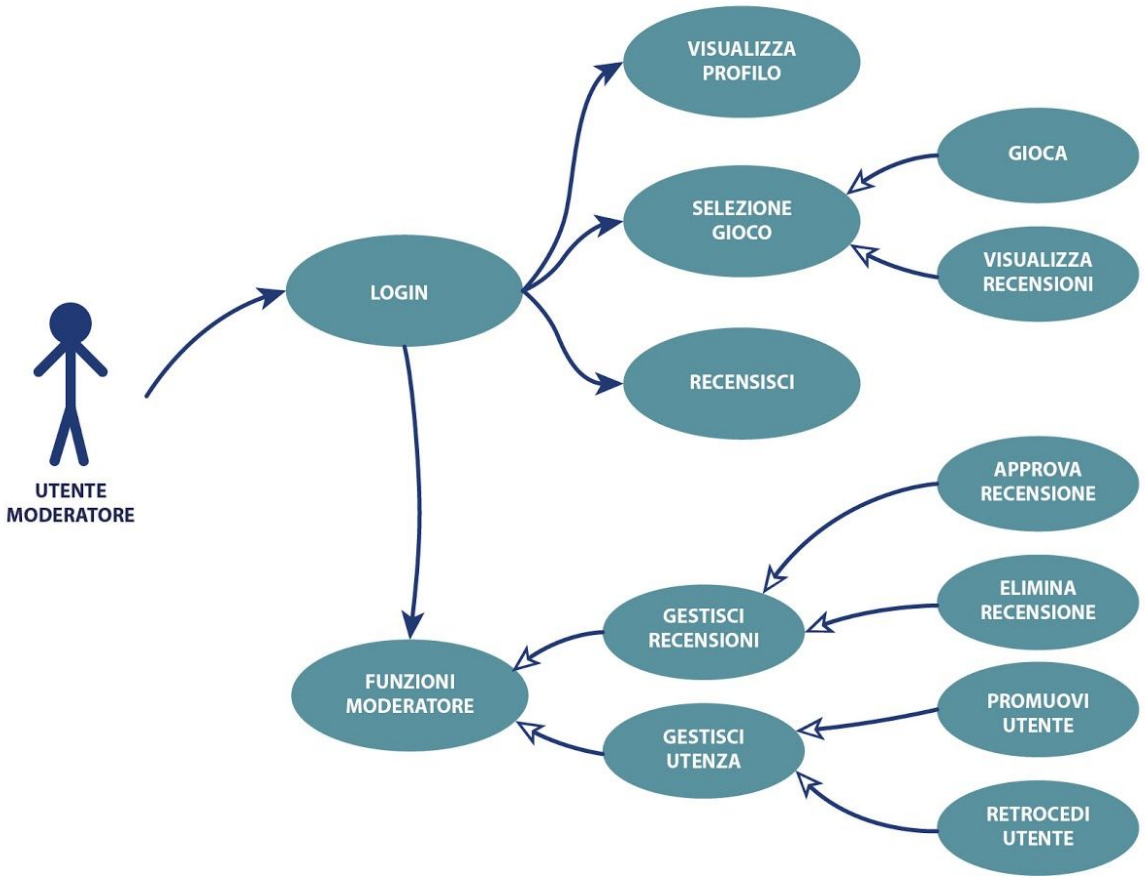
Scenario: Sessione di Gioco e Controllo Progressi

Nome Scenario	*Sessione di gioco e Controllo progressi*
Istanze Attori Coinvolti	*Mario Rossi*
Corso degli Eventi	○ *Mario si trova nella User Homepage e quindi ha effettuato il Login.* ○ *Preme sul pulsante Seleziona Gioco ed è condotto in un'apposita pagina dove compare una lista dei giochi disponibili.* ○ *Seleziona un gioco che ha attirato la sua attenzione e automaticamente si apre la pagina del gioco.* ○ *Mario preme sul pulsante Gioca e in primo piano compare la schermata di gioco, con la quale può cominciare ad interagire.* ○ *Finito di giocare Mario preme su Esci e viene riportato nella pagina del Gioco.* ○ *Mario è soddisfatto e vuole ora controllare se ha fatto progressi con l'esperienza e se ha conquistato trofei.* ○ *Mario torna indietro nella User Homepage e preme sul pulsante Visualizza Profilo.* ○ *E' condotto nella sua pagina personale dalla quale ha accesso a tutte le informazioni che voleva controllare.*

2. *Rappresentazione Tabellare dello scenario Sessione di gioco e Controllo progressi*

Diagrammi Use Case





Use Case

1. Registrazione utente

Nome Use-Case	*Registrazione Utente*
Attori Coinvolti	*Utente*
Corso degli Eventi	○ *L'utente inizia questa operazione riempiendo i campi richiesti nell'apposita schermata di registrazione utente, alla quale si potrà accedere direttamente dalla Homepage del sistema.* ○ *L'utente conclude l'operazione di registrazione cliccando sul pulsante Registrati in fondo alla schermata.* ○ *Le informazioni fornite dall'utente sono trasformate dal sistema in oggetto utente e inviate al database.*
Condizioni iniziali	*L'utente ha avviato il sistema e deve ancora registrarsi.*
Condizioni finali	*L'utente ha creato con successo il suo profilo sul sistema* *OR* *L'utente riceve una notifica che segnala un errore nel funzionamento del sistema o nella comunicazione con il database e che pertanto la registrazione non è andata a buon fine.*

2. Login utente

Nome Use-Case	*Login Utente*
Attori Coinvolti	*Utente, Moderatore, Admin*
Corso degli Eventi	○ *L'utente inizia questa operazione riempiendo i campi richiesti nell'apposita schermata di login, alla quale si potrà accedere direttamente dalla Homepage del sistema.* ○ *L'utente conclude l'operazione cliccando sul pulsante Login in fondo alla schermata.* ○ *Il sistema effettua un controllo sul database tramite le informazioni fornite dall'utente.* ○ *Se il check va a buon fine le informazioni sono caricate sul sistema, altrimenti si notifica che l'operazione non è riuscita.* ○ *L'utente è condotto nella schermata Home Utente.*
Condizioni iniziali	*L'utente ha avviato il sistema e vuole effettuare il login.*
Condizioni finali	*L'utente accede con successo al suo profilo* *OR* *L'utente riceve una notifica che segnala un errore nel funzionamento del sistema, un errore nella comunicazione con il database oppure che i dati inseriti sono errati o non esistenti e che pertanto il login non è andato a buon fine.*

3. Gioca

L'operazione "Gioca" comprende l'operazione di Selezione giochi e Visualizzazione pagine giochi.

Nome Use-Case	*Gioca*
Attori Coinvolti	*Utente, Moderatore*
Corso degli Eventi	○ *L'utente si sposta sulla schermata di selezione dei giochi;* ○ *L'utente seleziona il gioco a cui vuole giocare e clicca sul pulsante Gioca;* ○ *Il sistema comunica con il database e recupera le informazioni del gioco selezionato;* ○ *L'utente effettua la sua sessione di gioco. Una volta terminata viene ricondotto sulla schermata del gioco selezionato;*
Condizioni iniziali	*L'utente accede al sistema e ha intenzione di effettuare una sessione di gioco.*
Condizioni finali	*L'utente gioca a uno o più giochi tra i disponibili* *OR* *L'utente riceve una notifica che segnala un errore nel funzionamento del sistema o nella comunicazione con il database e non può effettuare la sessione di gioco.*

4. Visualizza profilo utente

L'operazione "Visualizza profilo utente" riassume le operazioni di visualizzazione profilo, visualizzazione esperienza, visualizzazione livello e visualizzazione trofei.

Nome Use-Case	*Visualizza Profilo Utente*
Attori Coinvolti	*Utente, Moderatore*
Corso degli Eventi	○ *L'utente clicca sul pulsante Visualizza Profilo sulla schermata Home Utente;* ○ *Il sistema comunica con il database per recuperare le informazioni dell'utente e dei trofei conquistati dal suddetto;* ○ *L'utente è condotto nella schermata del suo profilo e può controllare informazioni di autenticazione, esperienza acquisita, livello e trofei conquistati nel tempo;*
Condizioni iniziali	*L'utente accede al sistema e vuole visualizzare il suo profilo.*
Condizioni finali	*L'utente riesce a visualizzare correttamente il profilo.*

5. Recensisci gioco

Nome Use-Case	*Recensisci Gioco*
Attori Coinvolti	*Utente, Moderatore*
Corso degli Eventi	○ *L'utente si sposta sulla schermata di selezione dei giochi;* ○ *L'utente seleziona il gioco che vuole recensire;* ○ *Dalla schermate del gioco selezionato l'utente clicca sul pulsante Recensisci;* ○ *Una schermata apposita permette di lasciare una votazione da 1 a 5 e un giudizio scritto sul gioco recensito;*
Condizioni iniziali	*L'utente (che ha effettuato il login) vuole lasciare un giudizio personale su uno o più giochi tra quelli disponibili.*
Condizioni finali	*L'utente completa la sua recensione e viene notificato della riuscita dell'operazione, ovvero che la recensione è stata inviata ai moderatori per ricevere approvazione* *OR* *L'utente riceve una notifica che segnala un errore nel funzionamento del sistema o nella comunicazione con il database e che pertanto l'operazione di recensione non è andata a buon fine.*
Requisiti aggiuntivi	*Un moderatore può recensire giochi come ogni altro utente del sistema MA l'approvazione delle sue recensioni spetta ad altri moderatori.*

6. Richiesta upgrade a moderatore

Nome Use-Case	*Richiesta Upgrade a Moderatore*
Attori Coinvolti	*Utente*
Corso degli Eventi	○ *L'utente, dalla schermata Home Utente, clicca sul pulsante Richiesta Moderatore;* ○ *Una schermata apposita richiede la stesura di una breve presentazione;* ○ *La richiesta viene inviata al database e segnalata all'admin e ai moderatori per ricevere l'approvazione;*
Condizioni iniziali	*L'utente vuole diventare moderatore del sistema.*
Condizioni finali	*L'utente invia con successo la sua richiesta e viene notificato della riuscita dell'operazione e che la richiesta è in attesa di approvazione* *OR* *L'utente riceve una notifica che segnala un errore nel funzionamento del sistema o nella comunicazione con il database e che pertanto la richiesta non è stata inviata.*

7. Gestisci utenza

L'operazione "Gestisci utenza" riassume le operazioni "Promuovi utente a moderatore" e "Retrocedi moderatore a utente base".

Nome Use-Case	*Gestisci Utenza*
Attori Coinvolti	*Moderatore*
Corso degli Eventi	○ *Il moderatore, dalla pagina Home Utente, clicca sul pulsante Funzioni Moderatore;* ○ *Il moderatore clicca sul pulsante Gestisci Utenza nella schermata delle funzioni moderatore;* ○ *Viene visualizzata una lista degli utenti registrati nel sistema;* ○ *Il moderatore può scegliere se promuovere a moderatore gli utenti che ne hanno fatto richiesta* ○ *o retrocedere altri moderatori a utente base;*
Condizioni iniziali	*Il moderatore deve gestire l'utenza del sistema.*
Condizioni finali	*Il moderatore riesce a svolgere con successo le operazioni di promozione e retrocessione* *OR* *L'utente riceve una notifica che segnala un errore nel funzionamento del sistema o nella comunicazione con il database e che pertanto le operazioni effettuate non sono andate a buon fine.*

8. Gestisci recensioni

L'operazione "Gestisci recensioni" riassume le operazioni di Approvazione recensioni e Eliminazione recensioni.

Nome Use-Case	*Gestisci Recensioni*
Attori Coinvolti	*Moderatore*
Corso degli Eventi	○ *Il moderatore, dalla pagina Home Utente, clicca sul pulsante Funzioni Moderatore;* ○ *Il moderatore clicca sul pulsante Gestisci Recensioni nella schermata delle funzioni moderatore;* ○ *Viene visualizzata una lista delle recensioni da approvare o eliminare;* ○ *Il moderatore può scegliere se accettare una recensione oppure eliminarne una non idonea alla pubblicazione;*
Condizioni iniziali	*Il moderatore deve gestire le recensioni dei giochi non ancora approvate.*
Condizioni finali	*Il moderatore riesce a svolgere con successo le operazioni di approvazione o eliminazione delle recensioni* *OR* *L'utente riceve una notifica che segnala un errore nel funzionamento del sistema o nella comunicazione con il database e che pertanto le operazioni effettuate non sono andate a buon fine.*

System Design

Architettura Software :

L'architettura software scelta per la realizzazione del progetto è un pattern il cui utilizzo è sempre più diffuso per via della semplicità di utilizzo offerta e al contempo la chiarezza nella comprensione. Si tratta del pattern architetturale MVC: Model - View - Controller.

Descrizione :

Il pattern MVC suddivide funzionalmente il codice sorgente dell'applicazione in tre sottogruppi. L'applicazione viene infatti realizzata mediante la comunicazione costante tra Model, View e Controller.

Suddivisione in packages :

1. *Model*

Business.model racchiude la parte "concreta" dell'applicazione, ovvero è qui che vengono definiti gli attori partecipanti al sistema. Esso raccoglie per esempio le classi:

- Utente
- Gioco
- Recensione

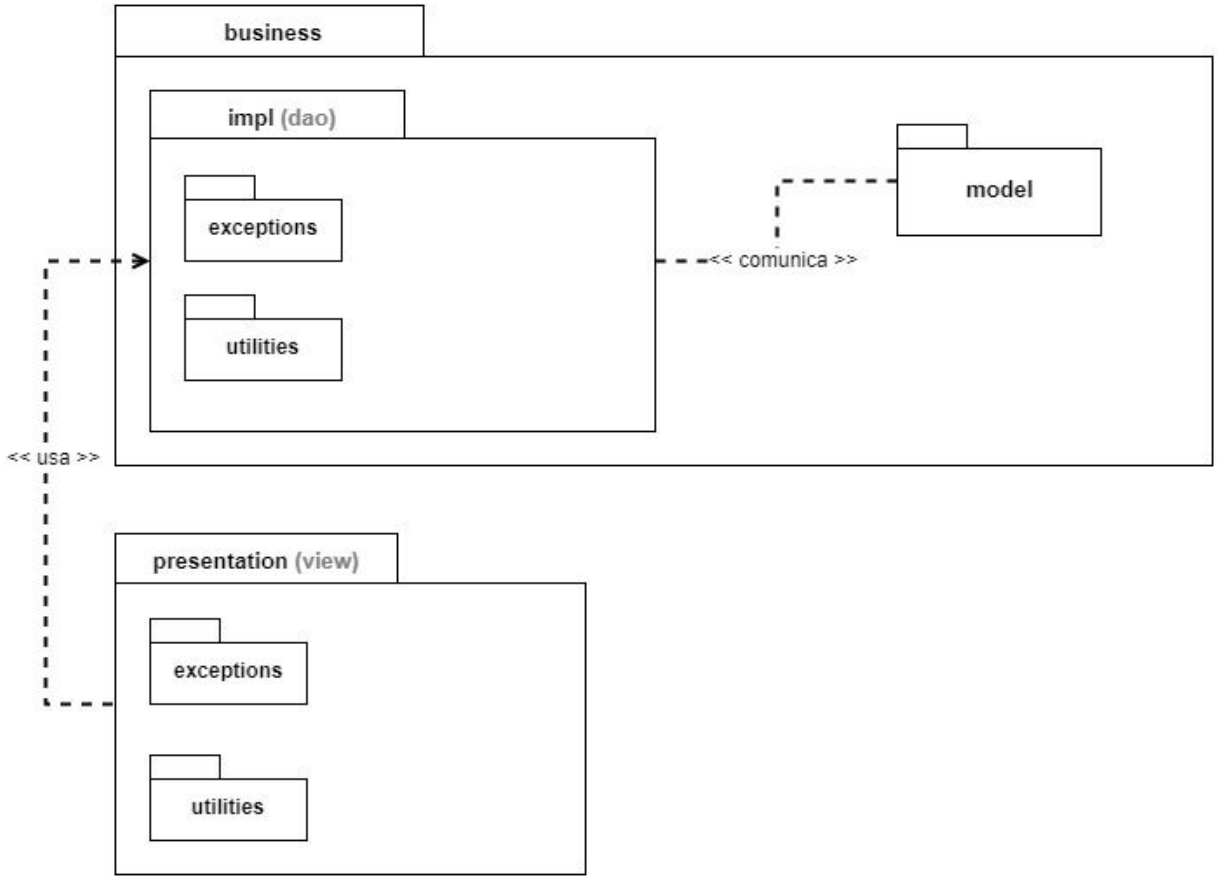
Tramite esse si creano gli oggetti che prendono parte al funzionamento del sistema e che permettono la gestione dei dati in arrivo dal controller e dalla view a livello di RAM e quindi non ancora inviati al database. Ogni cambiamento verificatosi sulla view ***(presentation)*** deve essere notificato tramite il controller ***(impl)*** al model, il quale ha il compito di immagazzinare le informazioni riguardo ogni evento accaduto e, sempre tramite controller, comunicare alla view che cosa visualizzare.

2. Presentation

Presentation costituisce la parte dell'applicazione che è visualizzata a schermo, ovvero la view.
Essa si occupa di realizzare la comunicazione dei cambiamenti avvenuti all'interno del model all'utilizzatore del sistema, sia che si tratti di operazioni che di dati aggiornati.
La componente view si baserà su una semplice interfaccia realizzata tramite la libreria Swing di Java.

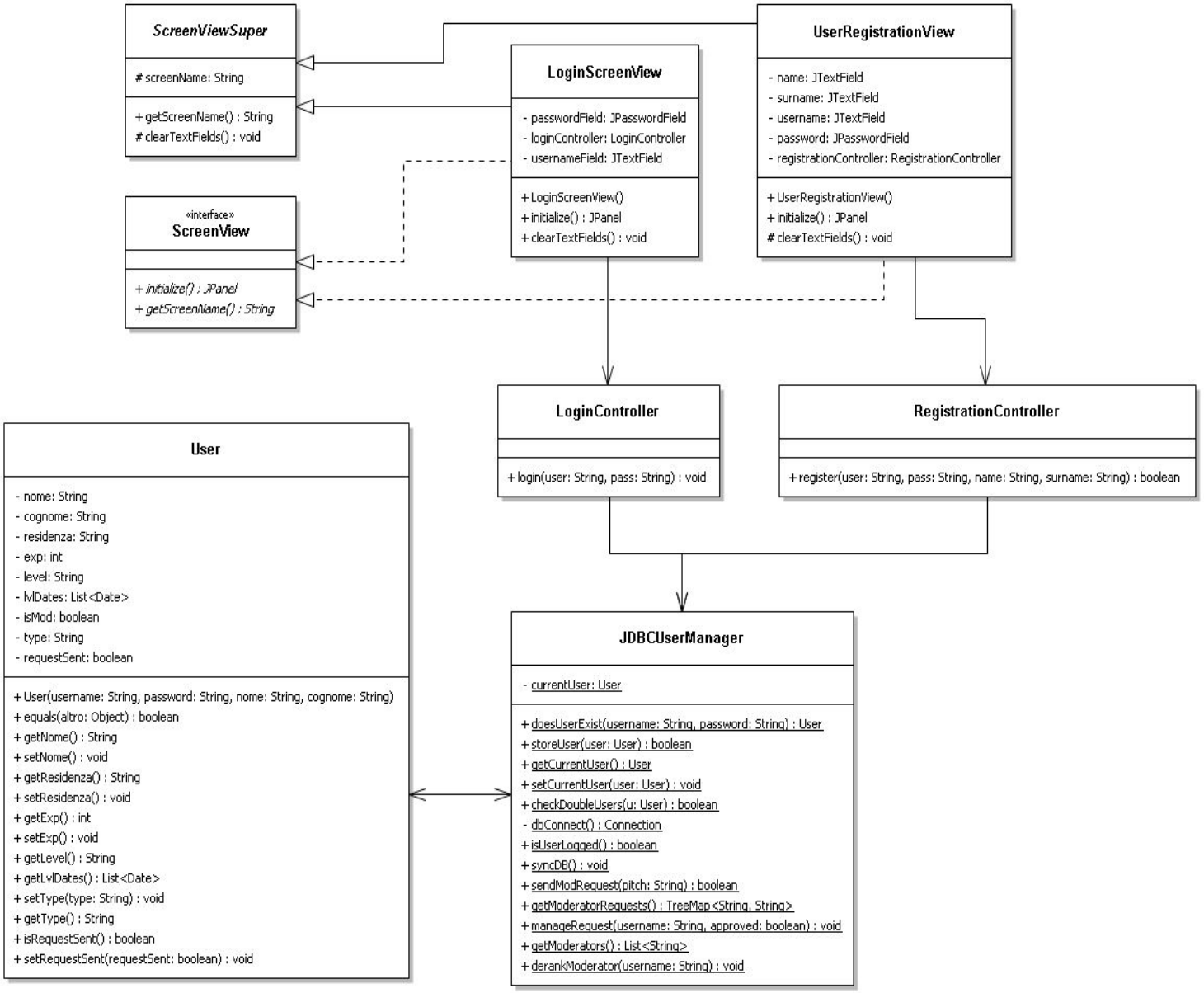
3. Impl

Business.impl raccoglie quella parte del codice che si occupa della logica di controllo e di accesso ai dati dell'applicazione.
In particolare funge da intermediario tra le operazioni avvenute nelle classi di ***presentation*** e i cambiamenti che avvengono nel model, che devono essere memorizzati in database.



Software & Object Design

1. Login & Registrazione



1.a Class Diagram Login/Registrazione raffinato in base alle scelte implementative

1.1 View:

LoginScreenView e **RegistrationView** sono le classi che si occupano della visualizzazione a video delle operazioni di Login e di Registrazione. Esse estendono la classe **ScreenViewSuper** e implementano l'interfaccia **ScreenView**. Inoltre utilizzano i metodi delle classi LoginController e RegistrationController.

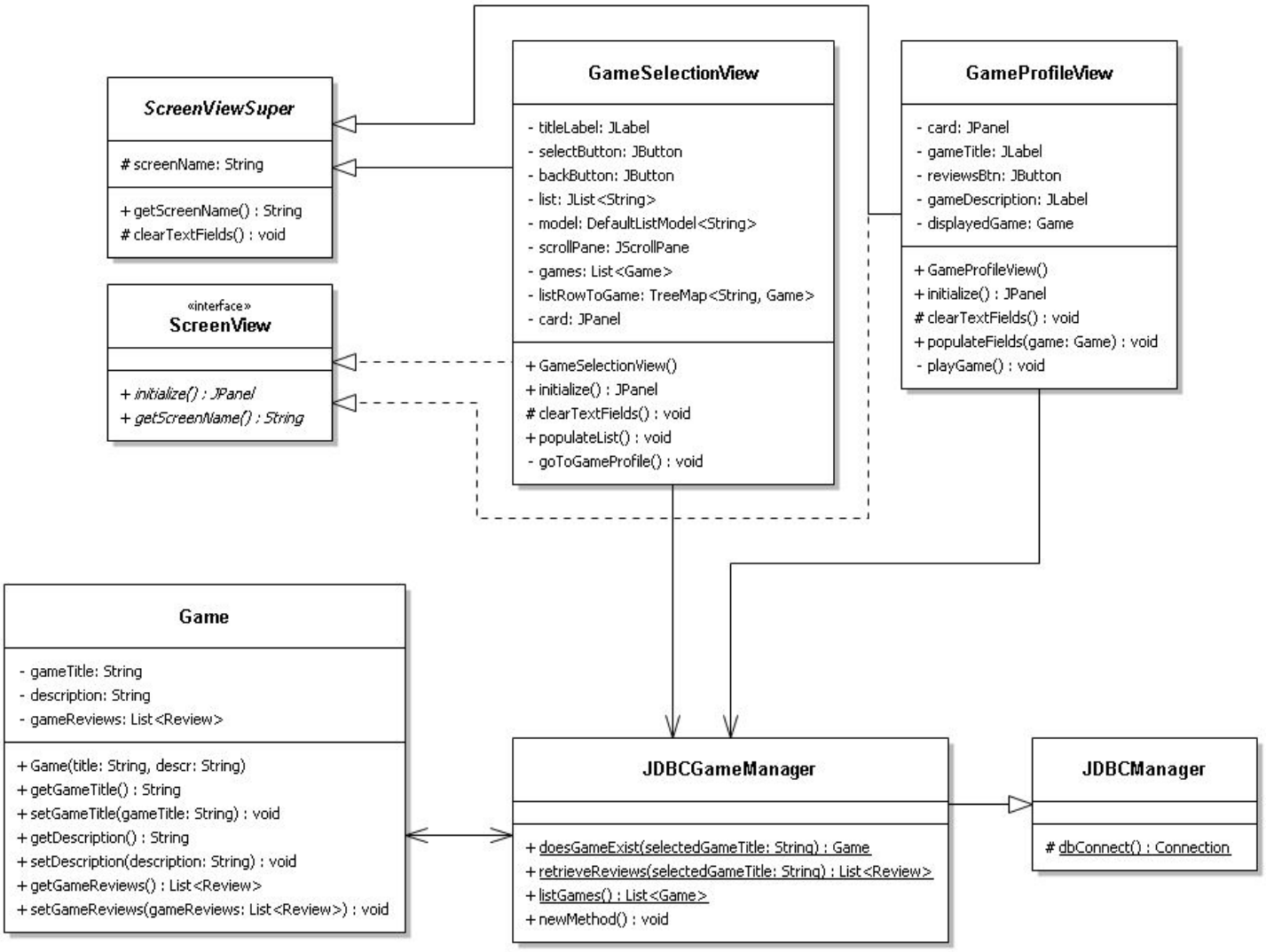
1.2 Controller:

LoginController e **RegistrationController** sono le classi che si occupano della comunicazione delle due Screen sopracitate con il Model dell'applicazione. Esse utilizzano i metodi della classe **JDBCUserManager**.

1.3 Model:

User è la classe che rappresenta un utente all'interno del Model e pertanto contiene tutte le informazioni al riguardo. La classe JDBCUserManager istanzia ed utilizza oggetti di tipo User.

2. Sessione di Gioco



2.a Class Diagram Gioco raffinato in base alle scelte implementative

2.1 View:

GameSelectionView è la classe che si occupa della visualizzazione a video della pagina di selezione dei giochi. Contiene una *List* di oggetti **Game** che viene popolata ogni volta che la pagina è aperta.
GameProfileView è invece la classe che visualizza a video la pagina specifica del gioco selezionato tramite GameSelectionView. Entrambe estendono la classe **ScreenViewSuper** e implementano l'interfaccia **ScreenView**.

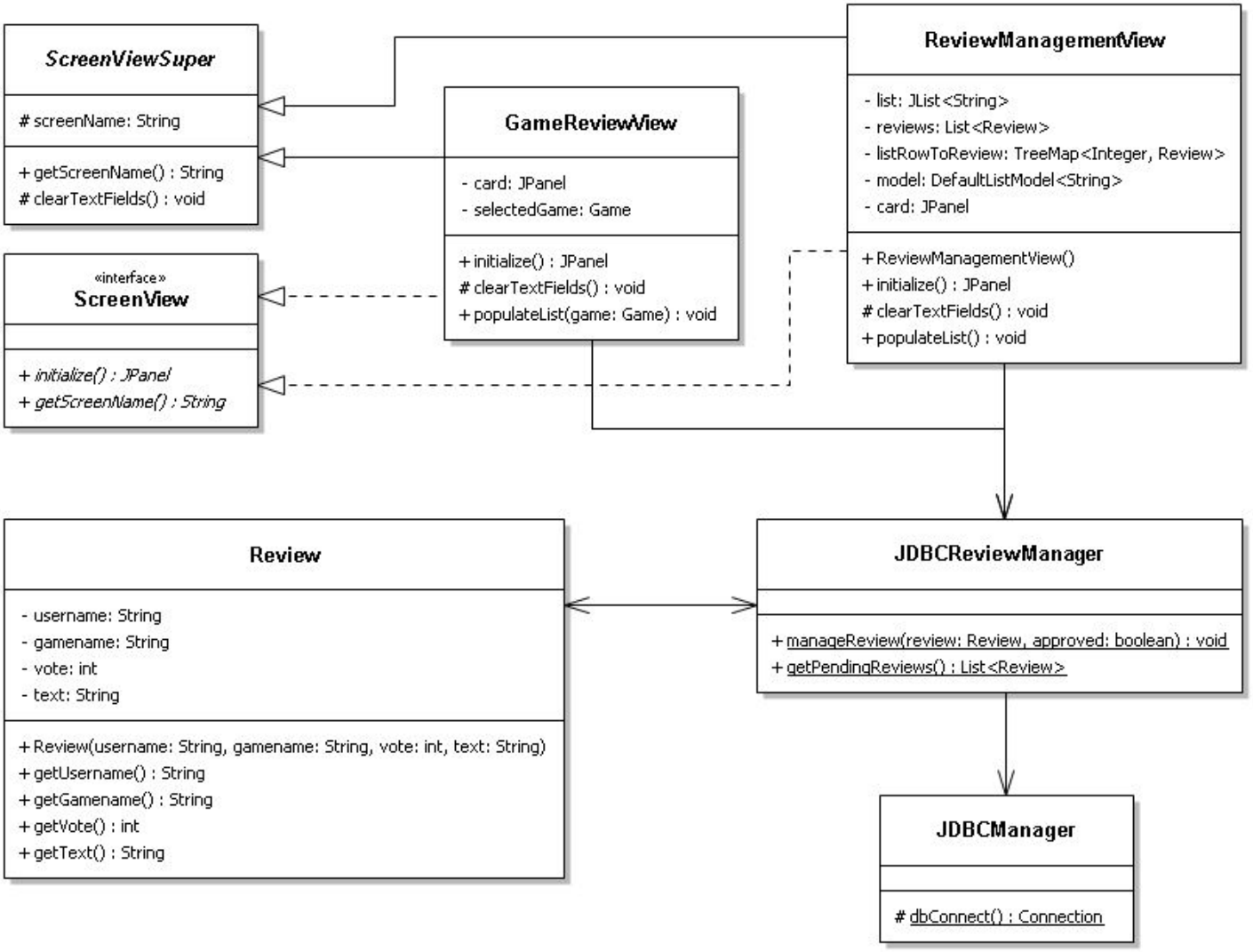
2.2 Controller:

La comunicazione tra parte Model e parte View in questa sezione è realizzata dalla classe **JDBCGameManager.**

2.3 Model:

Game è la classe che rappresenta un gioco all'interno del Model dell'applicazione e contiene informazioni relative ai giochi disponibili al sistema. La classe JDBCGameManager istanzia ed utilizza oggetti di tipo Game.

3. Recensione Giochi



3.a Class Diagram Review raffinato in base alle scelte implementative

3.1 View:

GameReviewView è la classe che si occupa della visualizzazione a video della pagina delle recensioni di un gioco selezionato. Contiene una *List* di oggetti **Review** che viene popolata ogni volta che la pagina è aperta.
ReviewManagementView è invece la classe che visualizza a video la pagina per la gestione delle recensioni (che fa quindi parte delle funzioni di un moderatore). Contiene una *List* di oggetti **Review** che costituiscono quelle recensioni che sono ancora in attesa di approvazione. Entrambe le classi estendono la superclasse **ScreenViewSuper** e implementano l'interfaccia **ScreenView**.

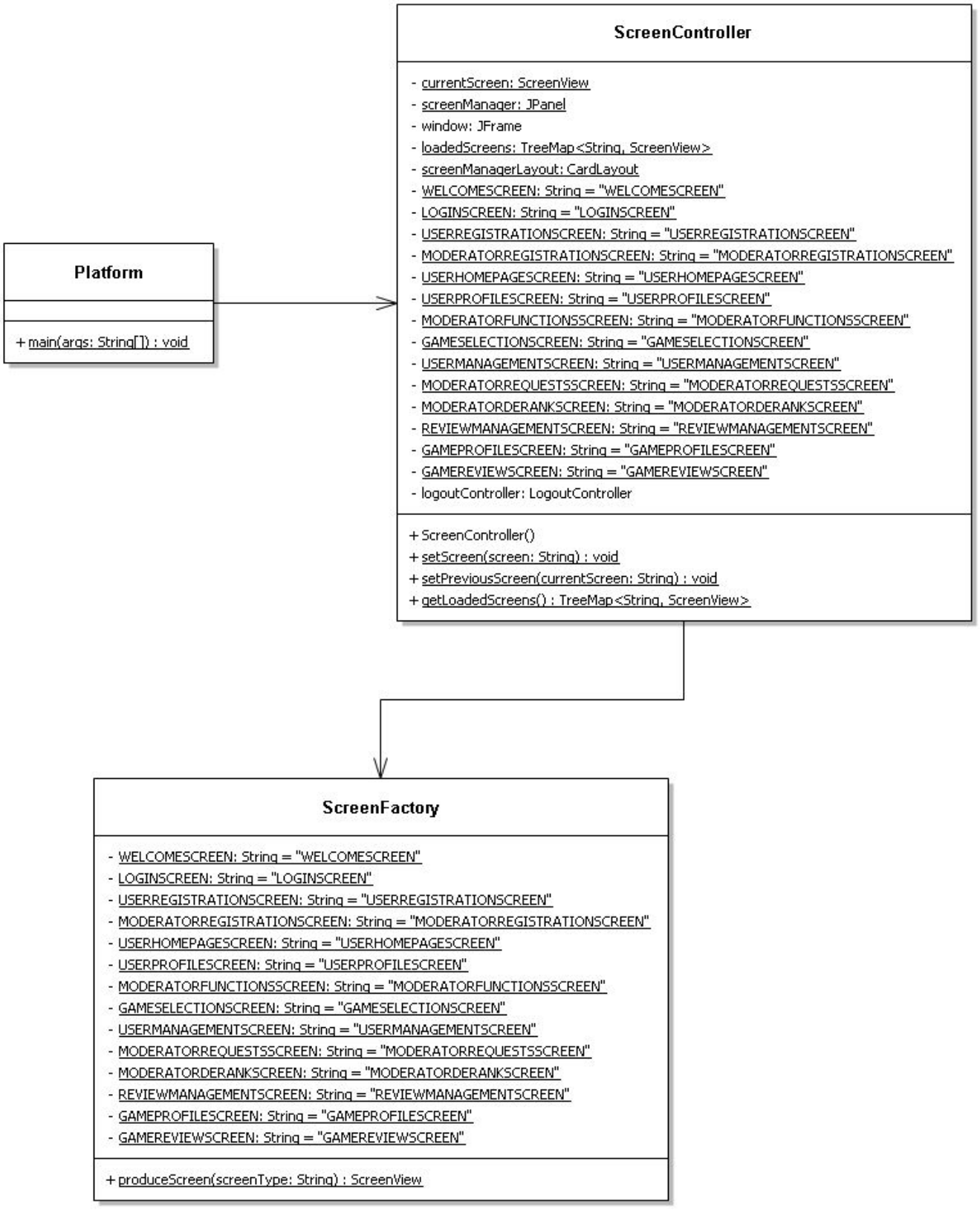
3.2 Controller:

La comunicazione tra parte Model e parte View in questa sezione è realizzata dalla classe **JDBCReviewManager.**

3.3 Model:

Review è la classe che rappresenta una recensione all'interno del Model. Serve a immagazzinare le informazioni di ogni recensione. La classe JDBCReviewManager istanzia ed utilizza oggetti di tipo Review.

4. Platform (Main)



Platform è la classe che contiene il *main*, all'interno del quale non si fa altro che creare un nuovo oggetto di tipo **ScreenController**. Questo oggetto costituisce la finestra dell'applicazione, che viene inizializzata al fine di contenere e visualizzare ogni successiva schermata determinata dalle scelte dell'utente.

Design Patterns Adottati

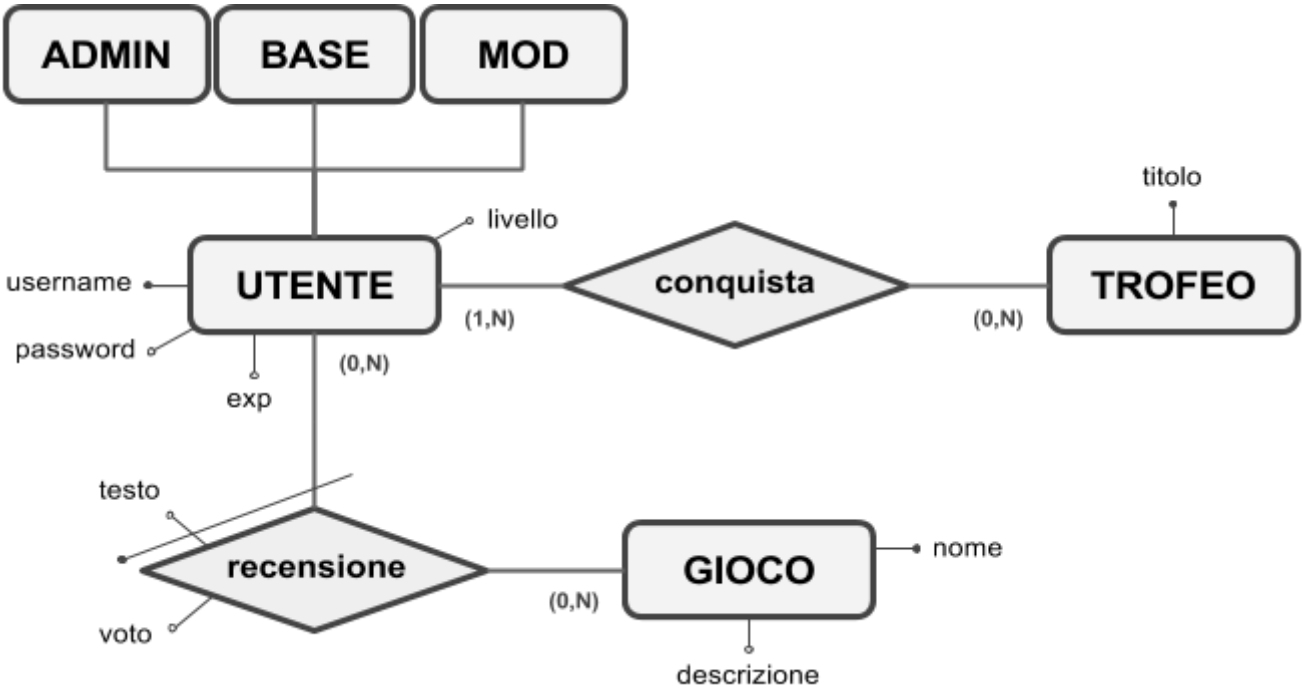
- **Factory Method DP:** per la creazione e la gestione delle schermate (screen) abbiamo deciso di utilizzare questo design pattern. La scelta è stata dettata dalla necessità dell'applicazione di avere a disposizione un gran numero di schermate differenti, la cui creazione e visualizzazione a schermo dipendono completamente dall'utilizzo dell'utente finale. Data la genericità della situazione abbiamo optato per il factory method "*produceScreen(String screenType)".*

Scelte di Design

- **API per comunicazione gioco - piattaforma:** la problematica di come realizzare la *comunicazione gioco - piattaforma* è sorta nel momento in cui il team ha dovuto decidere quali informazioni utili un gioco deve essere in grado di inviare al sistema Orion. L'obiettivo era realizzare un metodo di comunicazione che fosse comune a tutti i giochi e pertanto comune per tutti gli sviluppatori esterni che volessero decidere di far uso della piattaforma. La soluzione è stata ispirata da quella della famosa software house Valve per la sua piattaforma **Steam**, ovvero l'utilizzo di una API distribuita agli sviluppatori dalla società stessa. L'API di Orion (sotto forma di libreria importabile jar) stabilisce una sorta di intesa tra società e sviluppatore. Lo sviluppatore resta libero di decidere di non far comunicare il suo gioco

con la piattaforma. Nel caso in cui però esso voglia realizzare questa comunicazione è obbligato a far uso dell'interfaccia fornita all'interno del suo codice per realizzare il salvataggio dati.

Database Design



Schema ER Iniziale

Requisiti del database

Abbiamo deciso di implementare un database per questi motivi fondamentali:

1. la necessità di immagazzinare tutte le informazioni relative agli utenti registrati al sistema;
2. la necessità di mantenere le informazioni delle recensioni realizzate dagli utenti;
3. la possibilità di memorizzare i giochi disponibili;
4. la possibilità di memorizzare i trofei esistenti;

La tecnologia scelta ed utilizzata è quella MySQL perché:

1. si adatta bene al modello MVC;
2. è ottimale per la gestione di una quantità di dati relativamente esigua;
3. era precedentemente conosciuta dal team di progettazione;

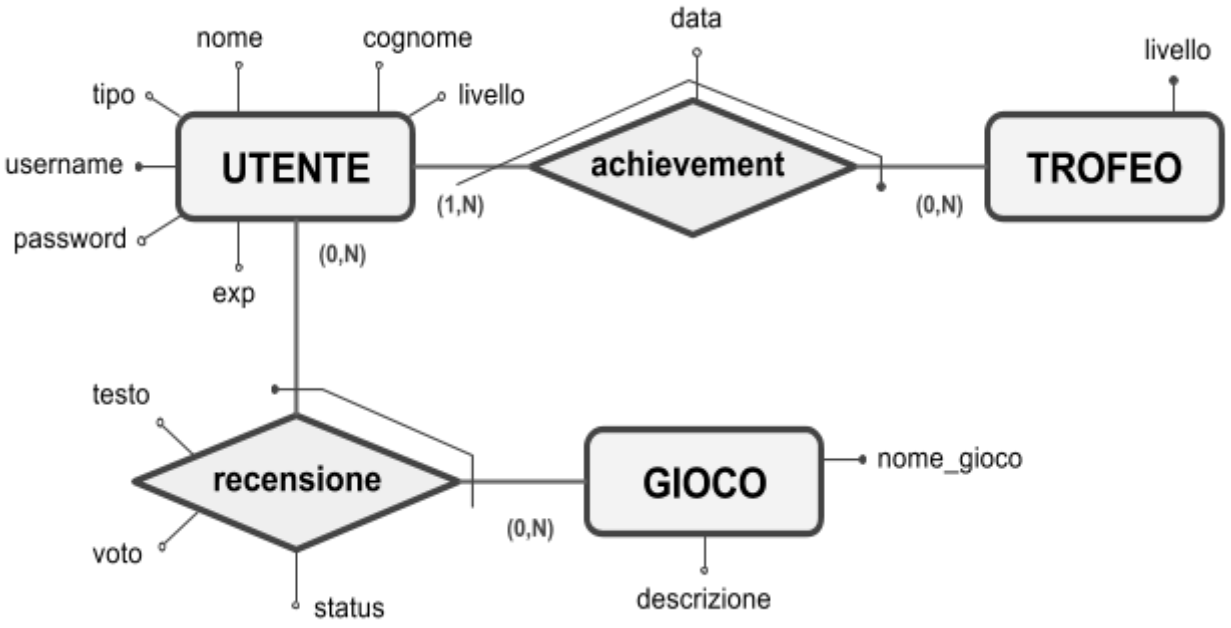
Schema Logico

User(Username, Password, Nome, Cognome, Level, Exp, Type**)**
Trophy(Level**)**
Game(Gamename, Description**)**
Achievement(Player, Trophy, Time**)**
Message(ID, Text, Sender, Receiver**)**
Mod_request(Mod_name, Pitch**)**
Review(Player, Game, Vote, Text, Status**)**

Vincoli

- L'attributo Player della tabella Achievement referenzia Username di User.
- L'attributo Trophy della tabella Achievement referenzia Level di Trophy.
- Gli attributi Sender e Receiver della tabella Message referenziano Username di User.
- L'attributo Sender può assumere un valore speciale detto "Sistema" che indica che la notifica è inviata automaticamente dal sistema all'utente.

- L’attributo Mod_name della tabella Mod_request referenzia Username di User con la condizione che il tipo di User sia Base.
- L’attributo Player della tabella Review referenzia Username di User.
- L’attributo Game della tabella Review referenzia Gamename di Game.



Schema ER Raffinato

Le tabelle “Message” e “Mod_request” non sono rappresentate nel modello ER perché i due concetti sono stati introdotti all’interno del progetto in fase implementativa, per motivi di utilità e necessità.