Relazione Progetto Ingegneria del Software

## Relatori:

* Filippo Lo Bue
* Pierfrancesco Soffritti
* Paolo Sarti

## Descrizione del problema:

Si vuole realizzare una piattaforma software remota che gestisca delle recensioni di videogiochi, pubblicate da dei recensori che danno una valutazione ai videogiochi, e permetta agli utenti registrati di consultarle avendone una visione personalizzata.

## Documento dei requisiti:

Ciascun videogioco presente nel sistema ha un nome, una data di rilascio, un’immagine e un genere tra: avventura, azione, fantasy, horror, mmorp, rpg, corse, indie, simulazione, sport, stealth e strategia.

Inoltre ogni videogioco può avere al più una recensione inserita nel sistema da uno dei recensori.

Ogni recensione contiene: un testo, il nome del recensore ed una valutazione numerica da 0 a 10, per ciascun aspetto del gioco che il recensore ha deciso di valutare. Ad esempio: grafica, giocabilità, storia, originalità.

Il recensore non indicherà una valutazione numerica complessiva per la recensione, la quale varierà a seconda delle preferenze dell’utente registrato sulla piattaforma. Essa verrà calcolata come media delle valutazioni del recensore, pesate in base all’importanza che ogni utente assegna ad ogni aspetto del gioco.

Il sistema manterrà per ogni utente registrato: il nome, la password e le sue preferenze: le associazioni tra gli aspetti del gioco e relativo peso numerico, tra 0 e 10, assegnato dall’utente.

Al momento della visualizzazione di una recensione, il sistema combinerà le informazioni sulle preferenze dell’utente con le valutazioni assegnate dai recensori, fornendo all’utente registrato una valutazione numerica complessiva e personalizzata del videogioco recensito.

I recensori, in quanto utenti registrati, potranno quindi vedere i voti complessivi dei videogiochi in modo personalizzato secondo le proprie preferenze, anche sulle loro stesse recensioni.

Gli utenti registrati potranno:

* Visualizzare le recensioni con il relativo voto personalizzato.
* Inserire videogiochi nel sistema.
* Gestire le preferenze sul proprio profilo: assegnando un peso numerico agli aspetti scelti tra quelli disponibili nel sistema.
* Rimuovere videogiochi non recensiti dal sistema.

I recensori, oltre alle facoltà degli utenti registrati, potranno:

* Recensire i videogiochi inserendo il testo della recensione.
* Aggiungere, modificare ed eliminare le valutazioni degli aspetti dalle recensioni dei videogiochi da loro recensiti.

Gli aspetti possono essere aggiunti nel sistema solo dai recensori al momento della valutazione degli aspetti delle loro recensioni.

Gli utenti non registrati al sistema:

* Non potranno aggiungere videogiochi.
* Potranno consultare le recensioni, ma al posto della valutazione complessiva personalizzata vedranno la media aritmetica delle valutazioni degli aspetti della recensione.
* Potranno registrarsi al sistema.

Il sistema deve garantire il proprio utilizzo a più utenti contemporaneamente, quindi deve supportare la presenza di più sessioni simultanee. In caso di modifica di informazioni, ad esempio aggiunta di una recensione ad un determinato videogioco, queste devono poter essere visibili a tutti gli utenti. Deve inoltre essere permesso ad uno stesso utente registrato di aprire più sessioni contemporaneamente.

Esempio:

**Videogiochi**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Grafica** | **Giocabilità** | **Storia** |
| Super Mario | 5 | 10 | 3 |

**Utenti**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Grafica** | **Giocabilità** | **Storia** |
| Utente 1 | 2 | 10 | 1 |
| Utente 2 | 9 | 6 | 10 |

Valutazioni Complessive Personalizzate:

|  |  |  |
| --- | --- | --- |
| **Nome Utente** | **Nome Videogioco** | **Valutazione Complessiva** |
| Utente 1 | Super Mario | 8.7\* |
| Utente 2 | Super Mario | 5.4\*\* |

Medie pesate:

\* (5x2+10x10+3x1)/(2+10+1)=8.7

\*\* (5x9+10x6+3x10)/(9+6+10)=5.4

## 

## 

## Glossario

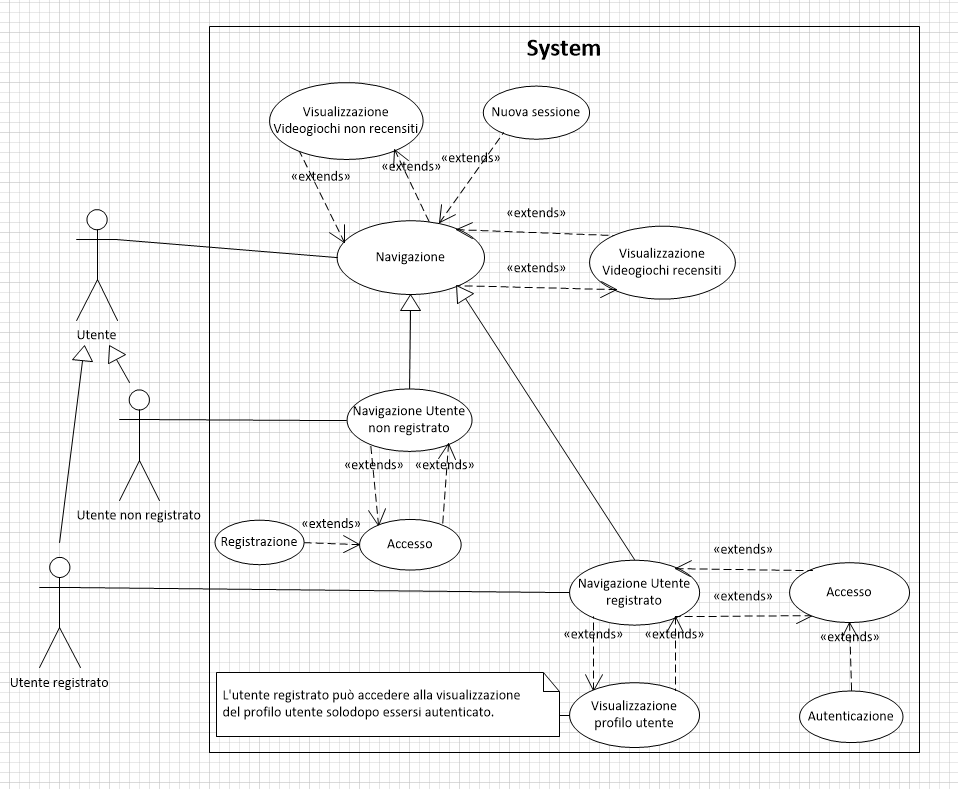
|  |  |
| --- | --- |
| **Termine** | **Significato e caratteristiche** |
| Recensione | Documento che analizza gli aspetti di un videogioco assegnando ad alcuni di essi (considerati rilevanti) una valutazione.  Può essere fatta da uno e un solo recensore per ciascun videogioco.  contiene:   * gli aspetti valutati * un testo * il nome del recensore |
| Videogioco | Inserito da un utente registrato (recensore o non).  Ad ogni videogioco è associata al più una recensione.  contiene:   * nome * data di rilascio * genere * immagine del gioco |
| Utente registrato | Persona registrata al sistema.  Può:   * visualizzare videogiochi e recensioni (con relativo voto complessivo personalizzato) * aggiungere ed eliminare videogiochi al sistema * aggiornare il proprio profilo |
| Recensore | È un utente registrato ed inoltre può:   * aggiungere recensioni ai videogiochi non ancora recensiti * aggiungere e/o valutare aspetti dei videogiochi solo dopo averli recensiti. * modificare gli aspetti valutati delle proprie recensioni * aggiungere aspetti al sistema nel atto della valutazione |
| Profilo dell’utente | Dati memorizzati nel sistema per ogni utente registrato.  Composto da:   * nome utente * password * preferenze sugli aspetti dei videogiochi scelte tra gli aspetti disponibili nel sistema. |
| Aspetto | Caratteristica del videogioco soggetta ad una valutazione numerica sia da parte dell’utente registrato in quanto preferenza, sia da parte del recensore in quanto valutazione.  Possono essere aggiunti nel sistema dal recensore. |
| Aspetto valutato | E’ un aspetto associato ad una recensione a cui il recensore ha attribuito una valutazione numerica. |
| Preferenza | Aspetto (selezionato tra quelli presenti nel sistema), a cui l’utente registrato può associare una valutazione numerica per quantificarne l’importanza in relazione alle altre preferenze. |
| Valutazione numerica | Numero intero da 0 a 10, associato ad un aspetto del videogioco o ad una preferenza. |
| Sessione | Dati legati ad una specifica interazione con il sistema. |
| Visualizzazione della recensione | Presentazione della recensione selezionata. |
| Visualizzazione della recensione personalizzata | Presentazione della recensione selezionata con la valutazione numerica complessiva dipendente dalle preferenze dell’utente registrato. |
| Valutazione numerica complessiva | Media pesata delle valutazioni numeriche degli aspetti assegnati dal recensore in base alle preferenze dell’utente.  Esempio:  **Videogiochi**   |  |  |  |  | | --- | --- | --- | --- | | **Nome** | **Grafica** | **Giocabilità** | **Storia** | | Super Mario | 5 | 10 | 3 |   **Utenti**   |  |  |  |  | | --- | --- | --- | --- | | **Nome** | **Peso Grafica** | **Peso Giocabilità** | **Peso Storia** | | Utente 1 | 2 | 10 | 1 | | Utente 2 | 9 | 6 | 10 |   **Valutazioni Complessive Personalizzate:**   |  |  |  | | --- | --- | --- | | **Nome Utente** | **Nome Videogioco** | **Valutazione Complessiva** | | Utente 1 | Super Mario | 8.7\* | | Utente 2 | Super Mario | 5.4\*\* | |
| Aggiungere videogiochi alla piattaforma | Operazione di aggiunta di un videogioco nel sistema, consentita ad ogni utente registrato. |
| Utente non registrato | Utente non memorizzato nel sistema.  Può:   * visualizzare videogiochi e corrispondenti recensioni, visualizzando una media aritmetica delle valutazioni degli aspetti della recensione. * registrarsi |
| Inserire videogiochi | Azione di inserimento di un videogioco nel sistema da parte di un utente registrato |
| Gestione delle preferenze | Azione di aggiunta/modifica/eliminazione delle proprie preferenze eseguita da un utente registrato |
| Registrazione al sistema | Operazione con la quale un utente non registrato può registrarsi specificando nome utente e password desiderati. |
| Recensire un videogioco | Azione tramite cui un recensore aggiunge la propria recensione di un videogioco al sistema. |
| Modificare una recensione | Azione tramite cui un recensore modifica gli aspetti valutati in una propria recensione nel sistema. |
| Aggiunta valutazione aspetto | Azione di assegnamento di una valutazione ad un aspetto. |
| Modifica valutazione aspetto | Azione di modifica di una valutazione di un aspetto valutato in una recensione. |
| Eliminazione valutazione aspetto | Azione di rimozione di un aspetto valutato da una recensione |
| Aggiunta aspetto | Azione di aggiunta di un aspetto al sistema. |

## 

## 

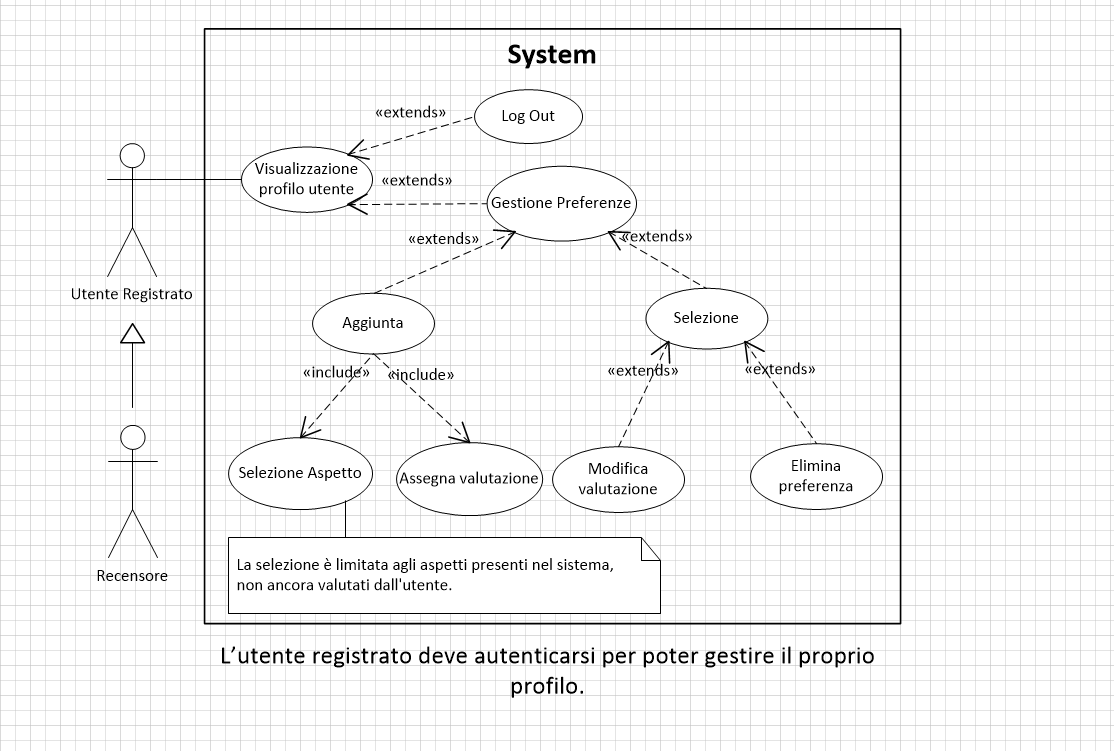
## Casi d’uso

### Navigazione



L’utente registrato nel sistema deve autenticarsi per accedere a qualsiasi altra funzionalità specifica degli utenti registrati.

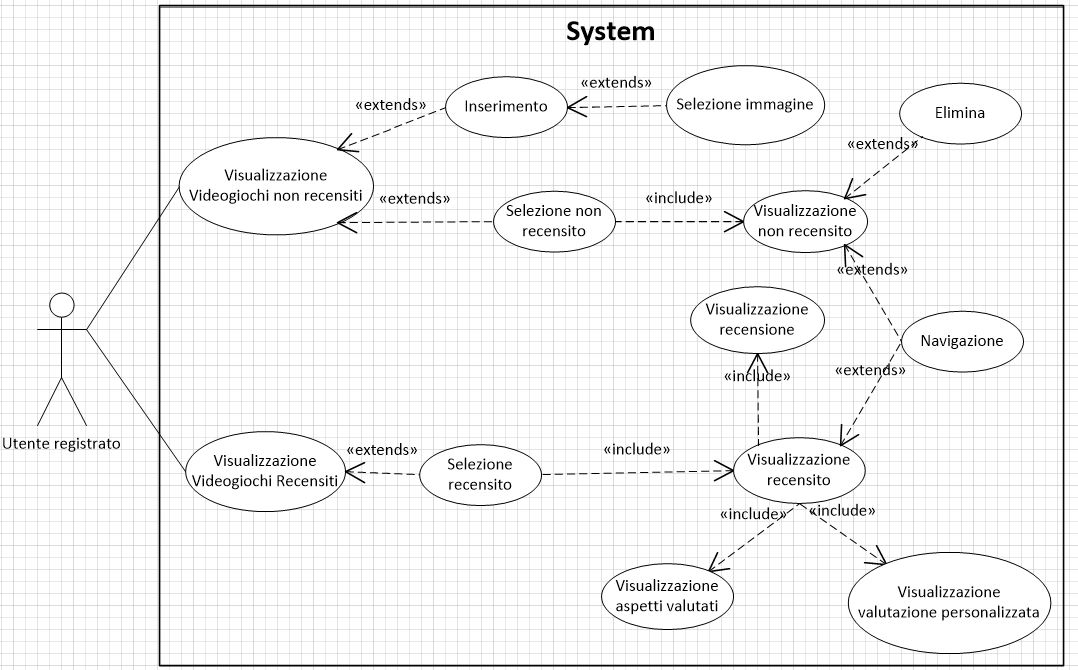
## Visualizzazione profilo



## Visualizzazione e Gestione Videogiochi recensiti e non

## 

## 



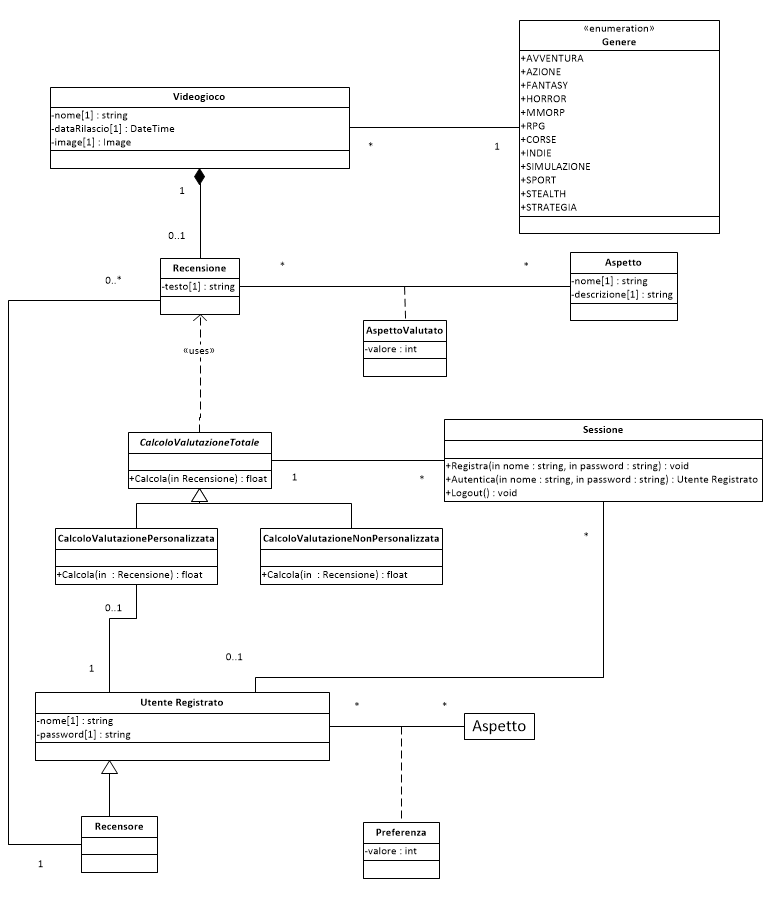
## 

## Scenari

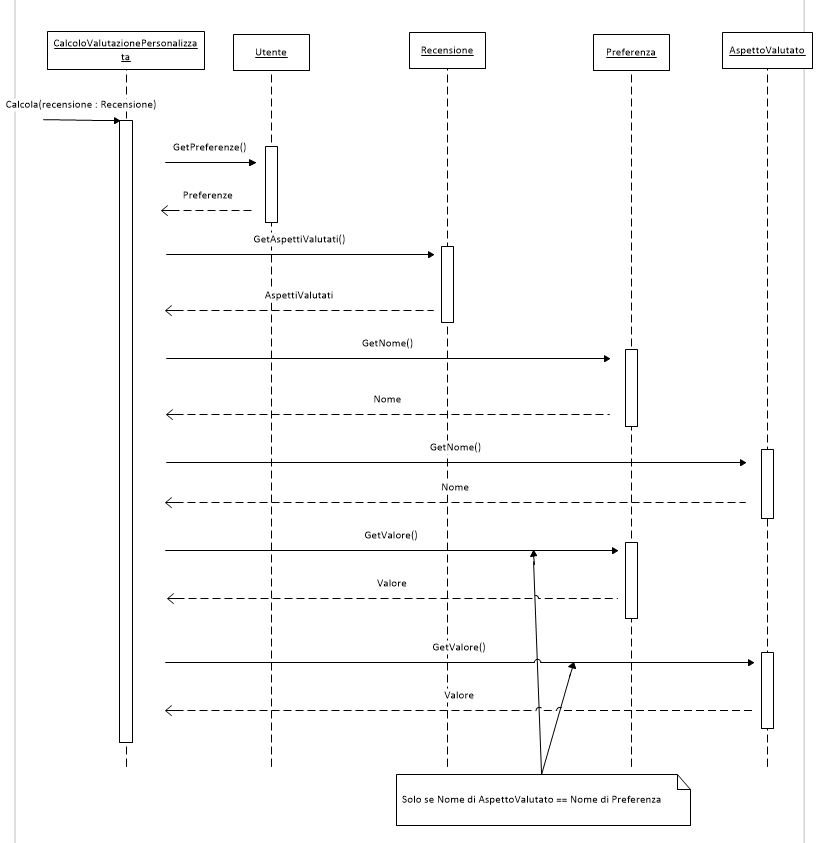
|  |  |
| --- | --- |
| Titolo | Aggiunta preferenza. |
| Descrizione | Aggiunta di una nuova preferenza (aspetto e relativa valutazione) nel profilo dell’utente. |
| Relazioni |  |
| Attori | Utente registrato, Recensore. |
| Precondizioni | * L’utente registrato deve essersi autenticato. * Il sistema deve contenere almeno un aspetto non ancora valutato nel profilo dell’utente. |
| Postcondizioni | * Il profilo dell’utente contiene una nuova preferenza. |
| Scenario principale | 1. Il sistema presenta gli aspetti presenti nel sistema tranne quelli già valutati dall’utente. 2. L’utente seleziona uno tra gli aspetti presentati dal sistema. 3. L’utente assegna un peso numerico intero da 0 a 10 all’aspetto selezionato. 4. L’utente conferma. 5. Il sistema salva i dati. |
| Scenari alternativi |  |
| Requisiti non funzionali | Il valore deve per forza essere un valore intero compreso tra 0 e 10 |
| Punti aperti |  |

|  |  |
| --- | --- |
| Titolo | Inserimento videogioco |
| Descrizione | Inserimento di un nuovo videogioco nel sistema. |
| Relazioni |  |
| Attori | Utente registrato, Recensore. |
| Precondizioni | * L’utente deve essersi autenticato. * Il sistema non deve contenere un videogioco con stesso nome e stessa data di rilascio. |
| Postcondizioni | * Il sistema contiene un nuovo videogioco. |
| Scenario principale | * L’utente aggiunge nome, data rilascio, genere del videogioco. * L’utente opzionalmente seleziona una immagine per il videogioco. * L’utente conferma l’aggiunta del videogioco. * Il sistema salva i dati. |
| Scenari alternativi |  |
| Requisiti non funzionali | La selezione del genere è limitata ad una lista finita. |
| Punti aperti |  |

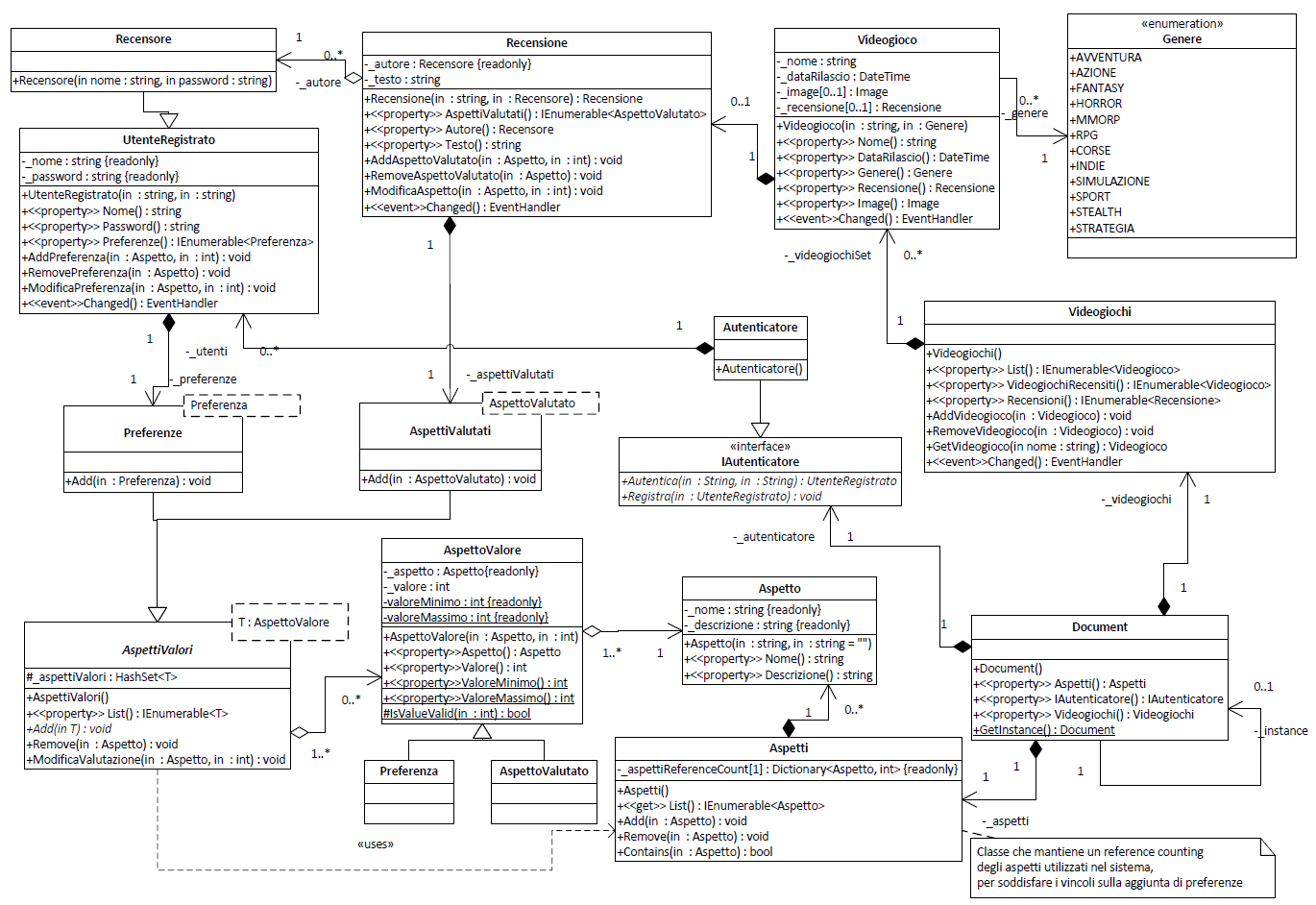
## Modello statico



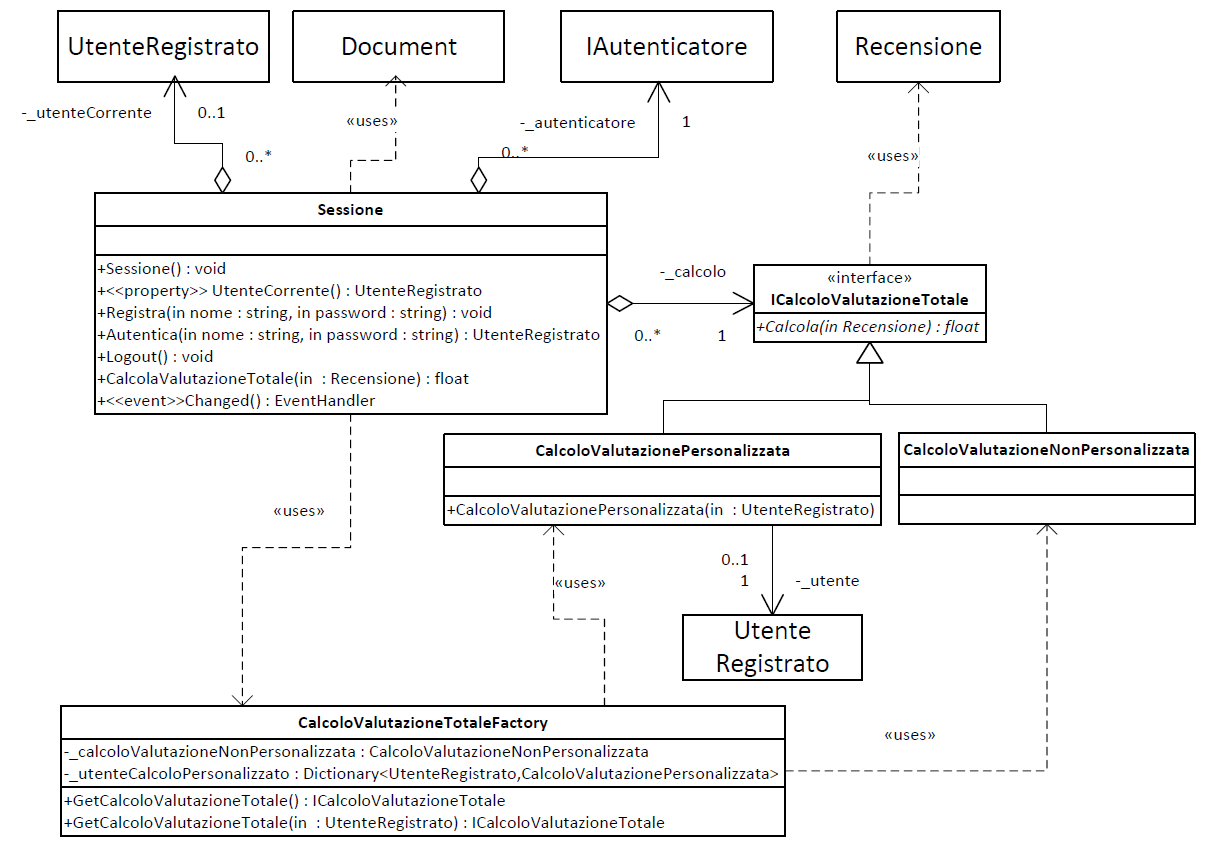
## Diagramma di sequenza



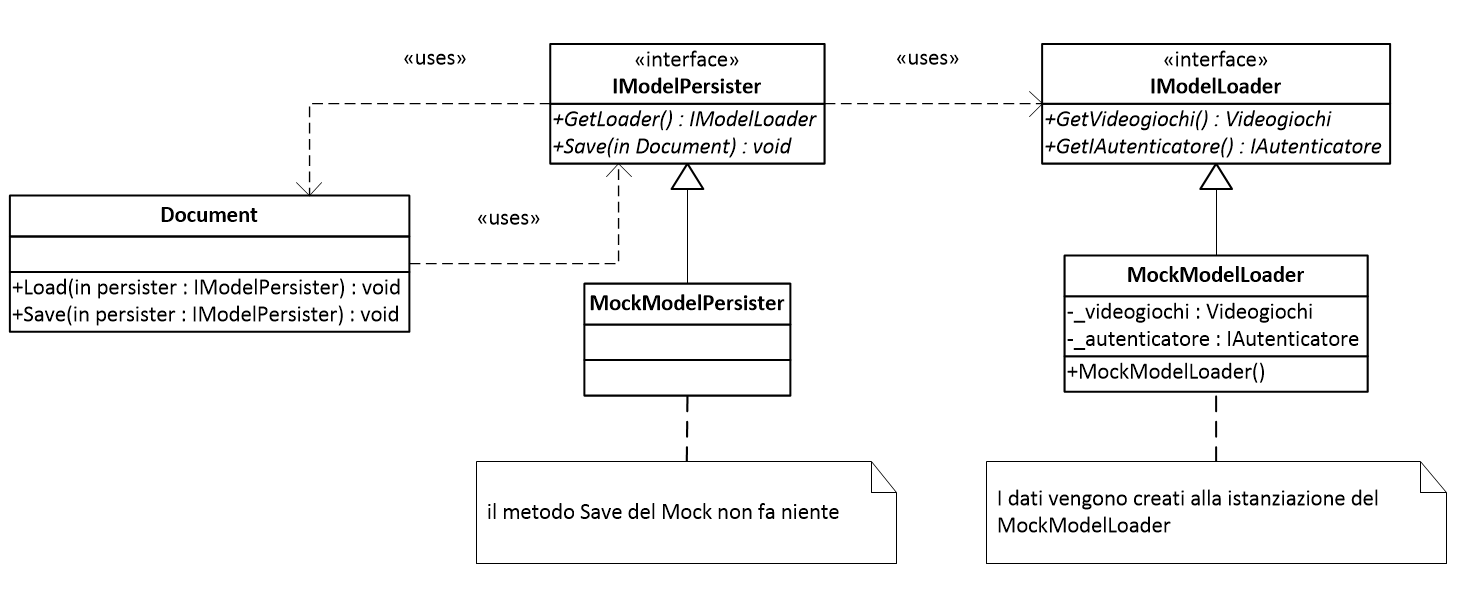
## Diagramma delle classi di progettazione



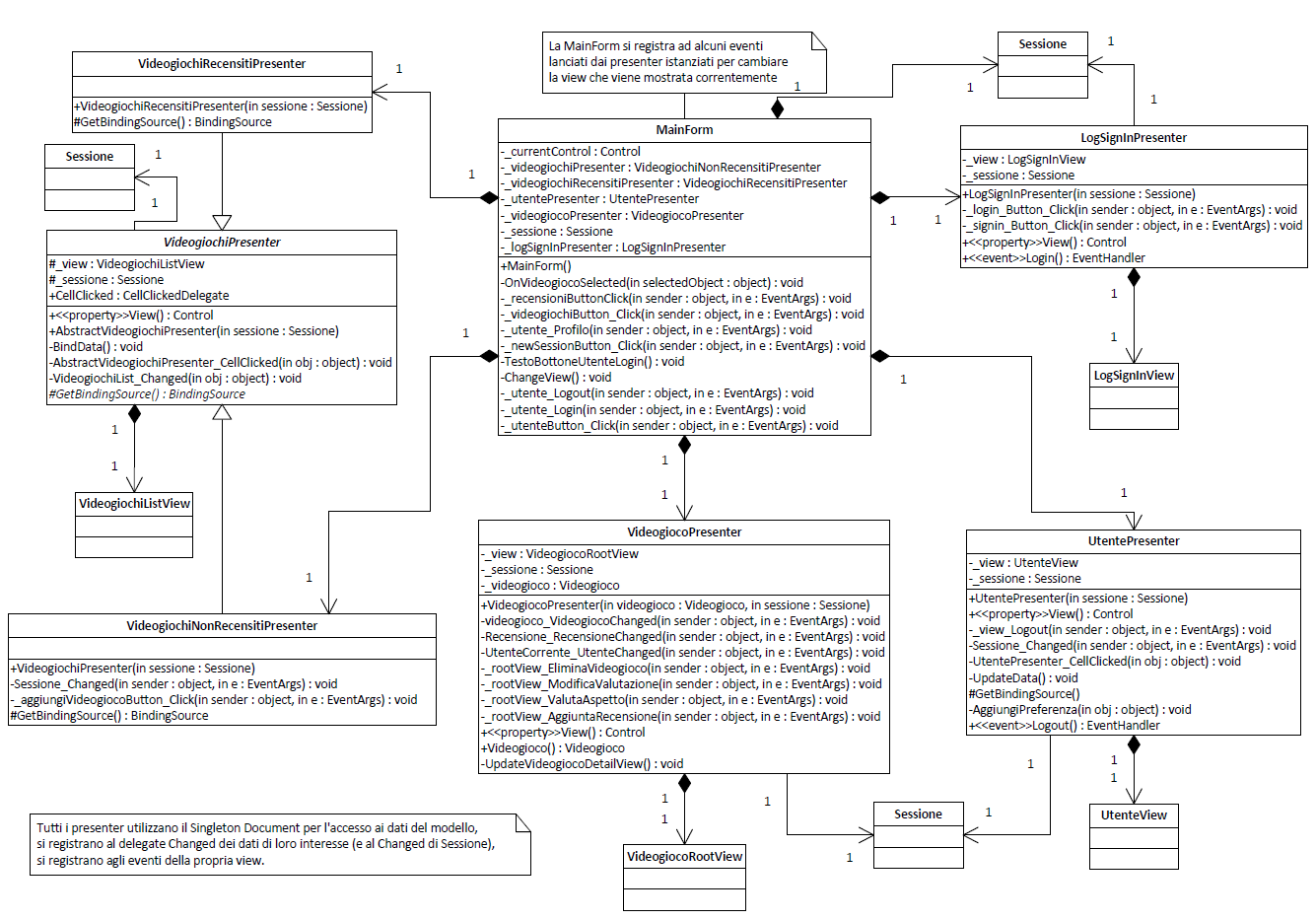
## Sessione e Calcolo



## Persistenza



## MainForm e Presenter



# Pattern e principi di progettazione

## Pattern

I pattern utilizzati in progettazione sono i seguenti:

### Singleton

La classe Document è realizzata come un Singleton per presentare una via di accesso unica ai dati del modello “remoti”, cioè quelli non legati alla sessione corrente.

### Strategy

La classe Sessione, per fornire il metodo CalcolaValutazioneTotale(Recensione):float*,* utilizza,tramite composizione e delega, l’interfaccia ICalcoloValutazioneTotale, le cui concretizzazioni definiscono diversi algoritmi di calcolo alternativi. L’aggiunta o il cambiamento di un algoritmo di calcolo si può ottenere quindi semplicemente aggiungendo una classe che implementa quella interfaccia, o modificando una classe esistente, senza dover modificare la classe Sessione.

### Flyweight

Le istanze delle classi CalcoloValutazionePersonalizzata e CalcoloValutazioneNonPersonalizzata si ottengono tramite una factory CalcoloValutazioneTotaleFactory.

La classe CalcoloValutazioneNonPersonalizzata non ha stato e per questo può essere condivisa senza conflitti, quindi la factory ne mantiene una istanza sola (creata alla prima richiesta) e restituisce sempre quella.

La classe CalcoloValutazionePersonalizzata è invece specifica per un UtenteRegistrato, quindi le sue istanze vengono mantenute in un Dictionary<UtenteRegistrato, ICalcoloValutazioneTotale>. In questo modo ogni sessione dello stesso utente condividerà l’algoritmo di calcolo personalizzato, sia nel caso di più sessioni contemporanee, sia nel caso un utente faccia autenticazioni successive.

L’UtenteRegistrato è quindi lo stato intrinseco, la recensione da valutare è lo stato estrinseco.

### Model View Presenter

L’applicazione è stata organizzata secondo il meta-pattern Model View Presenter, in cui i Presenter si registrano agli eventi scatenati dal Model con una granularità abbastanza fine. Tutti i presenter infatti si registrano solo agli eventi del modello lanciati dalle istanze di loro competenza, ad esempio UtentePresenter con l’istanza di UtenteRegistrato corrente (con deregistrazione al Logout), oltre che al cambiamento della Sessione corrente. Inoltre grazie ai Presenter la MainForm non conosce nessuna classe di View.

### Template Method

Il presenter astratto VideogiochiPresenter ha il metodo *GetBindingSource():BindingSource* astratto, utilizzato all’interno del metodo BindData() concreto. Esso definisce la sorgente dei dati per il popolamento della lista dei videogiochi, ed è astratto per permettere alle classi derivate di ridefinirla con i propri criteri. la classe concreta VideogiochiNonRecensitiPresenter selezionerà solo i videogiochi senza recensione, mentre la classe VideogiochiRecensitiPresenter selezionerà solo quelli con la recensione.

## Principi di progettazione

Nella progettazione è stato molto utilizzato il principio di composizione e delega, ad esempio nel caso di UtenteRegistrato e Recensione, che delegano rispettivamente alle classi contenitore Preferenze e AspettiValutati la gestione delle liste.

Per il Single Responsibility Principle infatti ogni classe non dovrebbe avere più di una responsabilità.

Nella progettazione della autenticazione è stato seguito il principio della inversione delle dipendenze. La sessione non dipende direttamente da una classe concreta per fornire le funzionalità di autenticazione (anche qui SRP), ma dall’interfaccia IAutenticatore, che permette quindi una migliore estendibilità a metodi di autenticazione differenti.

Lo stesso principio è stato applicato grazie al pattern Strategy nelle classi del calcolo, e per la gestione della persistenza: il Document dipende dall’interfaccia IModelPersister, non dalla implementazione concreta (fittizia).

L’applicazione del principio di inversione delle dipendenze permette anche di rispettare pienamente l’Open-Closed, rendendo queste parti facili da estendere tramite l’aggiunta di nuove classi.