Requirements Analysis Document

Progetto GameSquare



Partecipanti:

Francesco Galasso 0512105314

Cosimo Botticelli 0512105460

Aniello Pio La Pietra 0512105716

Sommario

[1. Introduzione 4](#_Toc66208669)

[**1.1 Scopo del sistema** 4](#_Toc66208670)

[**1.2 Ambito del sistema** 5](#_Toc66208671)

[**1.3 Obiettivi e criteri di successo** 5](#_Toc66208672)

[**1.4 Definizioni, acronimi e abbreviazioni** 5](#_Toc66208673)

[**1.5 Riferimenti** 5](#_Toc66208674)

[**1.6 Panoramica** 6](#_Toc66208675)

[2. Sistema corrente 6](#_Toc66208676)

[3. Sistema proposto 6](#_Toc66208677)

[**3.1 Panoramica** 6](#_Toc66208678)

[**3.2 Requisiti funzionali** 7](#_Toc66208679)

[**RF\_AUT: AUTENTICAZIONE** 7](#_Toc66208680)

[**RF\_USER: CONTRIBUTI UTENTE** 7](#_Toc66208681)

[**RF\_NAV: NAVIGAZIONE** 8](#_Toc66208682)

[**RF\_ADMIN: AMMINISTRAZIONE** 8](#_Toc66208683)

[**3.3 Requisiti non funzionali** 8](#_Toc66208684)

[**3.3.1 Usabilità** 8](#_Toc66208685)

[**3.3.2 Affidabilità** 8](#_Toc66208686)

[**3.3.3 Prestazioni** 9](#_Toc66208687)

[**3.3.4 Manutenibilità** 9](#_Toc66208688)

[**3.3.5 Implementazione** 9](#_Toc66208689)

[**3.3.6 Interfaccia** 9](#_Toc66208690)

[**3.3.7 Packaging** 9](#_Toc66208691)

[**3.3.8 Legali** 9](#_Toc66208692)

[**3.4 Modelli di sistema** 9](#_Toc66208693)

[**3.4.1 Scenari** 10](#_Toc66208694)

[**3.4.2 Use Case Model** 12](#_Toc66208695)

[UserRegistration 16](#_Toc66208696)

[IncorrectFields 17](#_Toc66208697)

[ExistingUsername 17](#_Toc66208698)

[UserLogin 17](#_Toc66208699)

[SessionNotActive 18](#_Toc66208700)

[InvalidUsername 18](#_Toc66208701)

[WrongPassword 19](#_Toc66208702)

[UserLogout 19](#_Toc66208703)

[RetrievePassword 19](#_Toc66208704)

[ChangePassword 20](#_Toc66208705)

[InvalidPassword 20](#_Toc66208706)

[SearchGame 21](#_Toc66208707)

[EmptySearch 21](#_Toc66208708)

[SearchByKeyword 21](#_Toc66208709)

[SearchByParameters 22](#_Toc66208710)

[ViewGameInfo 22](#_Toc66208711)

[ViewThread 22](#_Toc66208712)

[SortThreads 23](#_Toc66208713)

[AddGameToList 23](#_Toc66208714)

[AlreadyInList 24](#_Toc66208715)

[ChangeCategoryInList 24](#_Toc66208716)

[RemoveElementFromList 24](#_Toc66208717)

[VoteGame 25](#_Toc66208718)

[OverwriteVote 25](#_Toc66208719)

[PostComment 25](#_Toc66208720)

[LikeComment 26](#_Toc66208721)

[RemoveLike 26](#_Toc66208722)

[RequestThread 26](#_Toc66208723)

[RequestGame 27](#_Toc66208724)

[ReportComment 27](#_Toc66208725)

[ViewUserArea 27](#_Toc66208726)

[AcceptThread 28](#_Toc66208727)

[AddThread 28](#_Toc66208728)

[AcceptGame 29](#_Toc66208729)

[AddGame 29](#_Toc66208730)

[UpdateGame 29](#_Toc66208731)

[AcceptReport 30](#_Toc66208732)

[RefuseRequest 30](#_Toc66208733)

[RemoveThread 31](#_Toc66208734)

[RemoveComment 31](#_Toc66208735)

[**3.4.3 Object Model** 31](#_Toc66208736)

[**3.4.4 Dynamic Model** 37](#_Toc66208737)

1. Introduzione

## **1.1 Scopo del sistema**

GameSquare propone una piattaforma centralizzata e pensata per videogiocatori, a prescindere dalla loro esperienza, e sviluppatori.

Gli utenti registrati potranno aggiungere al proprio profilo i videogiochi in loro possesso, catalogandoli come “acquistati”, “in corso di gioco”, “completati”, e opzionalmente aggiungere a un gioco completato l’accezione di “platinato” (completato al 100%), con possibilità di spostarli da una categoria all’altra. Si potranno inoltre recensire videogiochi che sono stati già aggiunti alla propria lista personale, fornendo un voto in punti percentuali. La recensione dovrà essere significativa in quanto ci sarà la possibilità per gli altri utenti di votare le recensioni per stabilirne l’utilità. Ogni utente potrà proporre al gestore del catalogo di aggiungere un titolo mancante, fornendo della documentazione che prova l’esistenza dello stesso. Infine, i giocatori avranno un punteggio personale, che aumenterà completando o platinando giochi, e terrà conto della difficoltà degli stessi mettendo in relazione la quantità di persone che lo hanno giocato e quanti lo hanno finito per calcolare il punteggio guadagnato per il completamento, e la quantità di persone che lo hanno finito con quanti lo hanno platinato per calcolare il punteggio ottenuto per il conseguimento del platino.

Ogni gioco avrà la sua pagina, in cui sarà possibile visualizzare le informazioni base del titolo, la valutazione generale sulla base dei voti degli utenti e quanto sono stati aggiunti a una determinata categoria rispetto alle altre. La pagina di ogni gioco avrà una sezione commenti suddivisa per categorie (FAQs, discussione generale, tornei, etc.).

Una piattaforma del genere è utile all’utente che vuole avere un’idea d’insieme di un titolo prima di acquistarlo, come anche a coloro che vogliono chiedere a giocatori più esperti come completare sezioni difficoltose, o semplicemente chi vuole discutere di un titolo di suo gradimento.

Inoltre può giovare anche agli sviluppatori, in quanto possono rendersi conto di quante persone apprezzino il loro prodotto tanto da completarlo, e ovviamente leggere e scambiare commenti con la community.

## **1.2 Ambito del sistema**

Il mondo dell’intrattenimento videoludico è diventato negli ultimi anni un fenomeno esteso a centinaia di milioni di persone, grazie a servizi e piattaforme che permettono un semplice accesso a cataloghi molto grandi di videogiochi (e. g. Steam, PSN, Nintendo eShop, etc.), e fenomeni mediatici come la comunità di videogiocatori su YouTube o su Twitch, gli eSports e le molte riviste online specializzate.

Con questa crescita, molti usufruitori del prodotto sentono il bisogno di catalogare i propri giochi e analizzarli per avere un’idea di quanti titoli in proprio possesso debbano ancora essere completati (il cosiddetto backlog). Inoltre la ricerca di una community specifica per un determinato gioco, che sia per fare domande riguardo al completamento dello stesso, organizzare tornei in caso di un titolo competitivo, o semplicemente discussione generale, si fa sempre più difficoltosa. Ciò avviene a causa dell’aumento più che lineare della quantità di titoli sul mercato, grazie anche al successo recente di titoli creati da sviluppatori indipendenti e la riscoperta di giochi di generazioni passate (retrogaming).

Un altro aspetto da considerare è la quantità di servizi diversi che offrono qualche tipo di aggregazione per i giocatori, come gruppi Facebook, 4chan, Discord e le miriadi di forum e imageboards che tuttavia non sono centralizzati, e raramente pensati per ospitare questo specifico tipo di utenza.

## **1.3 Obiettivi e criteri di successo**

Obiettivi primari:

* Fornire una piattaforma centralizzata per tenere traccia del proprio catalogo videoludico.
* Compartimentalizzare le discussioni riguardanti uno specifico gioco per rendere più intuitivo lo scambio di opinioni o l’organizzazione di community.
* Offrire agli sviluppatori l’occasione di avere feedback diretto sulle loro opere.

Criteri di successo:

* Il sito deve essere online 24/7.
* Deve avere un tempo di risposta basso.

## **1.4 Definizioni, acronimi e abbreviazioni**

NA

## **1.5 Riferimenti**

NA

## **1.6 Panoramica**

Il seguente documento “GameSquare RAD” è diviso in sezioni ed ha la seguente composizione:

**- Sezione di Introduzione:**

In questa sezione è spiegato in breve qual è l'ambito del progetto e lo scopo del sistema che verrà implementato definendo gli obiettivi e i criteri di successo del sistema.

**-Sezione Sistema Corrente:**

Questa sezione resterà vuota in quanto il sistema proposto verrà creato da zero e non andrà a sostituire alcun servizio pubblico esistente.

**-Sezione SISTEMA PROPOSTO:**

In questa sezione viene inizialmente data una panoramica del sistema che deve essere realizzato e cosa andrà ad implementare.

Si procede con la sottosezione dei Requisiti Funzionali del sistema dove vengono identificate le funzionalità che Il sistema deve offrire.

I Requisiti Non Funzionali seguono il modello FURPS+, composto da: Usabilità, Affidabilità, Prestazioni, Manutenibilità, Implementazione, Interfaccia, Packaging, Legali. Seguono poi gli scenari, gli use-case e gli use-case diagram.

2. Sistema corrente

NA

3. Sistema proposto

## **3.1 Panoramica**

Il sistema da noi proposto è un sito web che offrirà servizi sia a utenti facenti pare della community videoludica, siano essi sviluppatori, nella forma di discussioni con i fruitori del loro prodotto, o i fruitori stessi, nella forma di liste per tenere traccia del proprio catalogo, discussioni con altri utenti e valutazioni.

L’utente registrato standard potrà:

* Visionare il proprio profilo con i relativi dati più rilevanti.
* Accedere alla lista da lui creata e modificarla a piacimento.
* Valutare i giochi presenti nella propria lista.
* Creare o contribuire a discussioni già esistenti, oltre che richiederne la creazione.
* Proporre l’aggiunta di un gioco, se dispone delle prove della sua esistenza.

Lo sviluppatore potrà:

* Visionare il proprio profilo con una sezione dedicata ai propri giochi sviluppati.
* Creare o contribuire a discussioni già esistenti.
* Aggiungere il proprio il gioco.

I moderatori potranno:

* Moderare e rimuovere qualsiasi tipo di contenuto (discussioni, commenti, etc.) qualora questo sia inopportuno.
* Accettare o eliminare richieste di creazione di discussioni o di cancellazione di contenuti inopportuni da parte degli utenti.

I gestori catalogo potranno:

* Accettare o eliminare richieste da parte degli utenti di aggiunta di un gioco al sito.
* Aggiungere essi stessi un gioco al sito, oltre che modificarne le informazioni.

Gli utenti non registrati potranno:

* Navigare il sito e cercare giochi tramite ricerca testuale, parametri o titoli correlati.
* Visualizzare la pagina di un prodotto e le informazioni relative, comprese le discussioni a riguardo.

## **3.2 Requisiti funzionali**

### **RF\_AUT: AUTENTICAZIONE**

* **RF\_AUT\_01 – Registrazione** –Priorità Media

Il sistema deve permettere la registrazione di un utente.

* **RF\_AUT\_02 – Login** –Priorità Alta

Il sistema deve permettere l’accesso di un utente.

* **RF\_AUT\_03 – Log out** –Priorità Alta

Il sistema deve permettere all’utente di terminare la propria sessione.

* **RF\_AUT\_04 – Recupero Password** –Priorità Medio-Bassa

Il sistema deve permettere all’utente di recuperare la propria password.

* **RF\_AUT\_04 – Modifica Password** –Priorità Medio-Bassa

Il sistema deve permettere all’utente di modificare la propria password.

### **RF\_USER: CONTRIBUTI UTENTE**

* **RF\_USER\_01 – Aggiungi Gioco alla Lista** –Priorità Alta

Il sistema deve permettere all’utente di aggiungere un qualsivoglia gioco alla sua lista, selezionando una categoria a piacimento.

* **RF\_USER\_02 – Vota Gioco** –Priorità Alta

Il sistema deve permettere all’utente di poter aggiungere una valutazione ad un gioco.

* **RF\_USER\_03 – Commenta** –Priorità Alta

Il sistema deve permettere all’utente di contribuire a una discussione tramite un commento.

* **RF\_USER\_03 – Like** –Priorità Medio-Bassa

Il sistema deve permettere all’utente di mettere “mi piace” a un commento.

* **RF\_USER\_04 – Richiesta Discussione** –Priorità Media

Il sistema deve permettere all’utente di richiedere la creazione di una discussione.

* **RF\_USER\_05 – Richiesta Aggiunta Gioco** –Priorità Media

Il sistema deve permettere all’utente di richiedere l’aggiunta di un gioco ai gestori del catalogo.

* **RF\_USER\_06 – Segnalazione contenuto** –Priorità Media

Il sistema deve permettere all’utente di segnalare ai moderatori contenuti ritenuti inopportuni o non pertinenti**.**

### **RF\_NAV: NAVIGAZIONE**

* **RF\_NAV\_01 – Cerca Elemento** –Priorità Media

Il sistema permette all’utente di eseguire una ricerca testuale o tramite parametri sui giochi.

* **RF\_NAV\_02 – Visualizza Elemento** –Priorità Alta

Il sistema deve far visualizzare la pagina dedicata di un gioco con le relative informazioni.

### **RF\_ADMIN: AMMINISTRAZIONE**

* **RF\_ADMIN\_01 – Valuta Richiesta Discussione** –Priorità Media

Il moderatore deve poter valutare (accettare o rifiutare) richieste di creazione di discussioni da parte degli utenti.

* **RF\_ADMIN\_02 – Creazione Discussione** –Priorità Alta

Il moderatore e gli sviluppatori di videogiochi devono poter creare discussioni relative a determinati giochi, senza che esse necessitino di approvazione.

* **RF\_ADMIN\_03 – Valuta Segnalazione**–Priorità Media

Il moderatore deve poter valutare segnalazioni di contenuti inappropriati da parte degli utenti.

* **RF\_ADMIN\_04 – Valuta Richiesta aggiunta gioco** –Priorità Media

Il gestore catalogo deve poter valutare richieste per l’aggiunta di nuovi giochi al sito da parte degli utenti

* **RF\_ADMIN\_05 – Aggiungi o modifica Gioco** –Priorità Media

Il gestore del catalogo deve poter modificare o aggiungere giochi al sistema.

* **RF\_ADMIN\_06 – Rimuovi Elemento** –Priorità Media

Il moderatore deve poter rimuovere elementi creati dall’utenza che ritiene non opportuni.

## **3.3 Requisiti non funzionali**

### **3.3.1 Usabilità**

Il sistema deve:

* Permettere di accedere alle funzioni principali (ricerca di un gioco, accesso al profilo utente) in modo intuitivo e dalla homepage.
* Contenere una sezione “Aiuto” che funge da breve guida per l’utente.

### **3.3.2 Affidabilità**

Il servizio, in quanto sito web, dovrà essere online 24/7 con basso rischio di crash del servizio di hosting.

Inoltre, agli utenti non sarà richiesto di fornire dati sensibili in quanto non utili ai fini dell’utilizzo di GameSquare.

### **3.3.3 Prestazioni**

* Il tempo di risposta di una query deve essere basso, in condizioni medie minore o uguale a un secondo.
* Non c’è un limite realistico ai giochi che è possibile aggiungere al database.
* Non c’è un limite realistico ai giochi che l’utente può avere nella sua libreria.
* Il servizio deve funzionare correttamente su dispositivi con diverse risoluzioni e aspect ratio.

### **3.3.4 Manutenibilità**

* In futuro ci sarà, per ogni gioco, una sezione dei giochi consigliati.
* La rimozione dei bug sarà a carico degli sviluppatori.
* Le modifiche e le migliorie saranno a carico di sviluppatori e progettisti, in base alle direttive e alle richieste dell’utenza.

### **3.3.5 Implementazione**

* La parte back-end deve essere programmata in Java.
* Il database deve essere programmato in SQL.
* La parte front-end deve essere strutturata tramite CSS e JavaScript.

### **3.3.6 Interfaccia**

* Il sistema deve interagire con un database di videogiochi importandone il contenuto.
* Il contenuto del database sarà gestito dalle API fornite dallo stesso.
* Il sito si affiderà a un sito di hosting per essere online.

### **3.3.7 Packaging**

Il sistema, in quanto sito web dovrà essere supportato da tutti i browser più usati.

### **3.3.8 Legali**

NA

## **3.4 Modelli di sistema**

### **3.4.1 Scenari**

#### **-Scenario 1:**

***Nome Scenario:*** ricercaTorneo

***Attori Partecipanti:*** Nico:utente

***Descrizione:*** Nico, un giocatore di Tekken 7, vuole partecipare al suo primo torneo competitivo online.

***Flusso di eventi:***

1. Si reca sulla homepage di GameSquare e cerca il gioco interessato.
2. Sulla pagina del gioco in questione e clicca sulla sezione “discussioni”.
3. Filtra le discussioni per argomento selezionando “competitivo”.
4. Nota varie organizzazioni di tornei online, fra cui una per giocatori con poca esperienza in tale ambito.
5. Esprime il suo desiderio di partecipare in quanto orari e modalità gli sono convenienti.
6. L’autore del thread risponde a Nico e si mettono d’accordo sulle modalità di partecipazione.

#### **-Scenario 2:**

***Nome Scenario:*** communityFeedback

***Attori Partecipanti:*** Lorenzo:utente

***Descrizione:*** Lorenzo, uno sviluppatore indipendente, ha pubblicato il suo primo videogioco da pochi giorni e sta cercando del feedback a riguardo.

***Flusso di eventi:***

1. Cerca il proprio gioco sulla piattaforma, visualizzandone la pagina.
2. Nota che l’85% delle persone che lo hanno acquistato lo hanno anche giocato, indice di alta curiosità da parte del pubblico.
3. Nota anche che il 70% delle persone che lo hanno giocato lo hanno anche completato, indice forse di scarsa longevità.
4. Si dirige poi nella sezione discussioni, dove nonostante un generale elogio ha conferma della scarsa difficoltà del titolo.
5. Comincia, infine, egli stesso una discussione dove si annuncia come creatore del titolo in questione e richiede consigli e feedback direttamente ai propri acquirenti.

#### **-Scenario 3:**

***Nome Scenario:*** richiestaAggiuntaGioco

***Attori Partecipanti:*** Francesco:utente

***Descrizione:*** Francesco, un appassionato di videogiochi meno conosciuti, non riesce a trovare su GameSquare il titolo “Garage: Bad Dream Adventure”.

***Flusso di eventi:***

1. Si reca sull’apposita pagina contenente il form per proporre l’aggiunta di un gioco.
2. Inserisce i requisiti, ossia il nome del gioco e una fonte attendibile della sua esistenza, in questo caso un riferimento a un video del gameplay caricato per intero su YouTube.
3. Dopo qualche giorno, la sua proposta viene approvata e il gioco aggiunto al database.
4. Francesco aggiunge il gioco al suo catalogo di titoli completati.

#### **-Scenario 4:**

***Nome Scenario:*** eliminazioneContenuto

***Attori Partecipanti:*** Piero:admin

***Descrizione:*** Piero, un admin di GameSquare, nota che un contenuto è inopportuno.

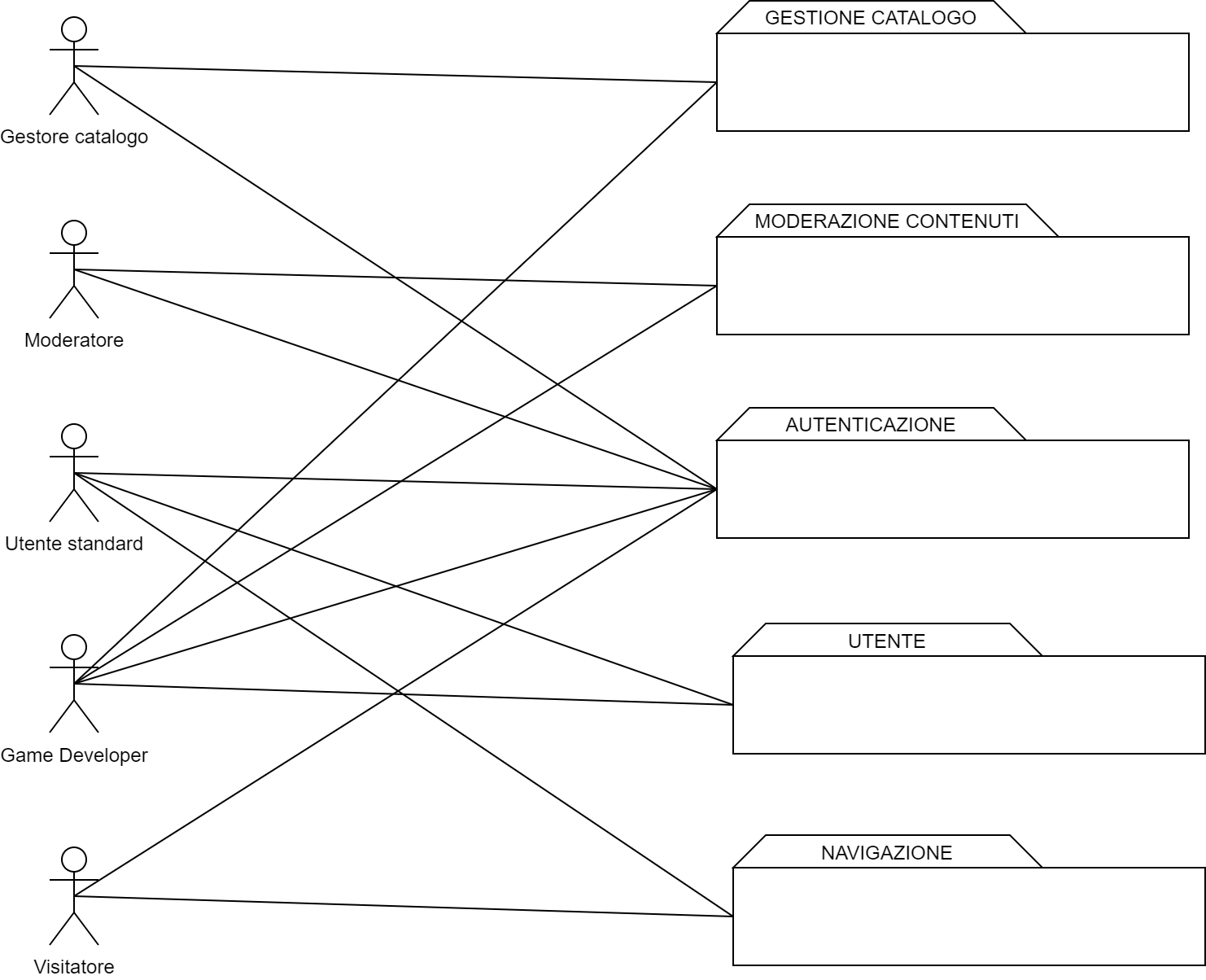
***Flusso di eventi:***

1. Molti utenti hanno segnalato una discussione come non opportuna, a causa della mancanza di contributi significativi alla community e all’eccessiva volgarità.
2. L’admin nota le molte segnalazioni e decide di rimuovere il contenuto nocivo.
3. Tramite un apposito tasto elimina il contenuto inviando una segnalazione privata all’utente che lo ha creato incoraggiandolo a soppesare meglio le proprie parole la prossima volta.

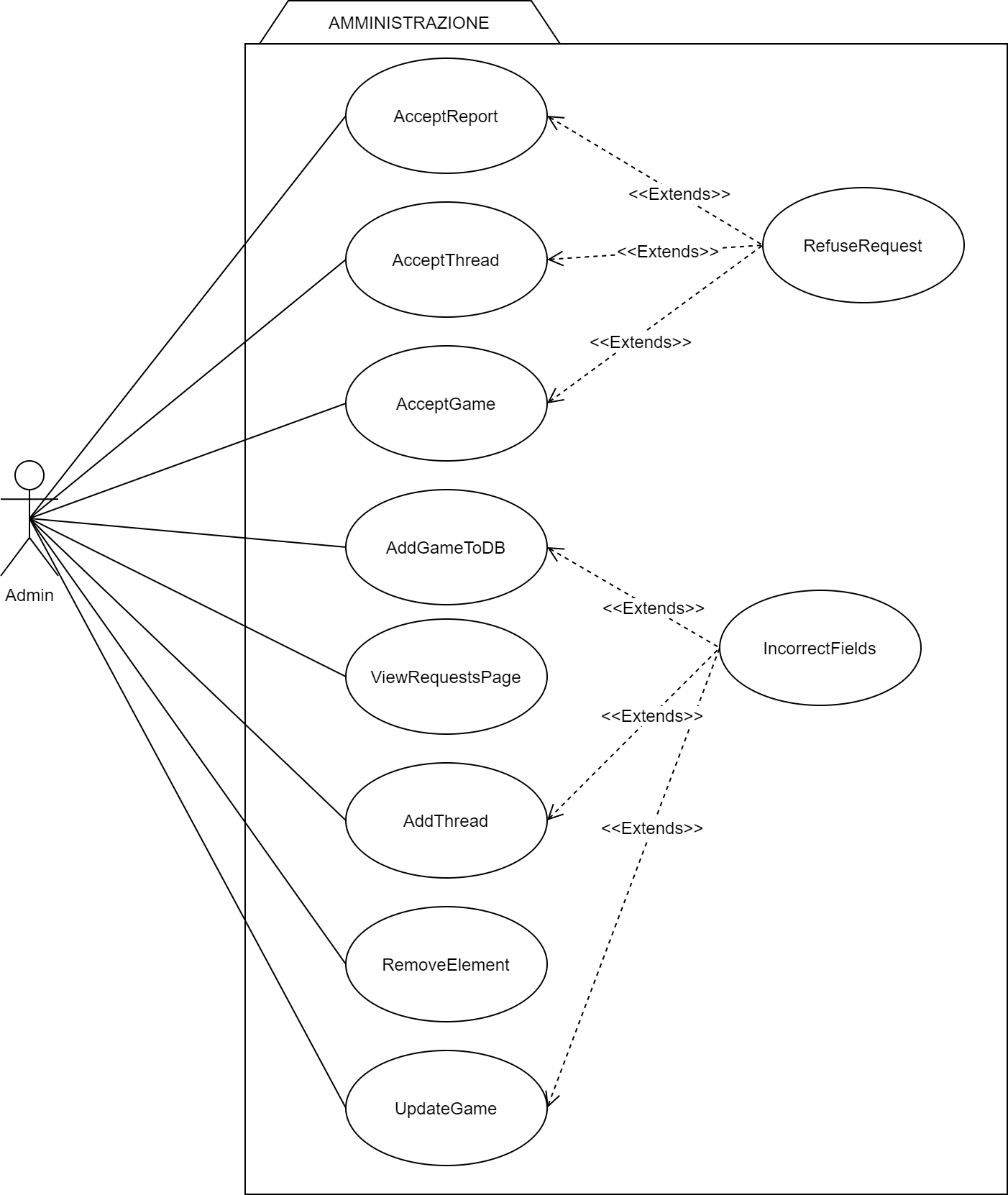
### **3.4.2 Use Case Model**

#### **3.4.2.1 Use Case Model Diagrams**

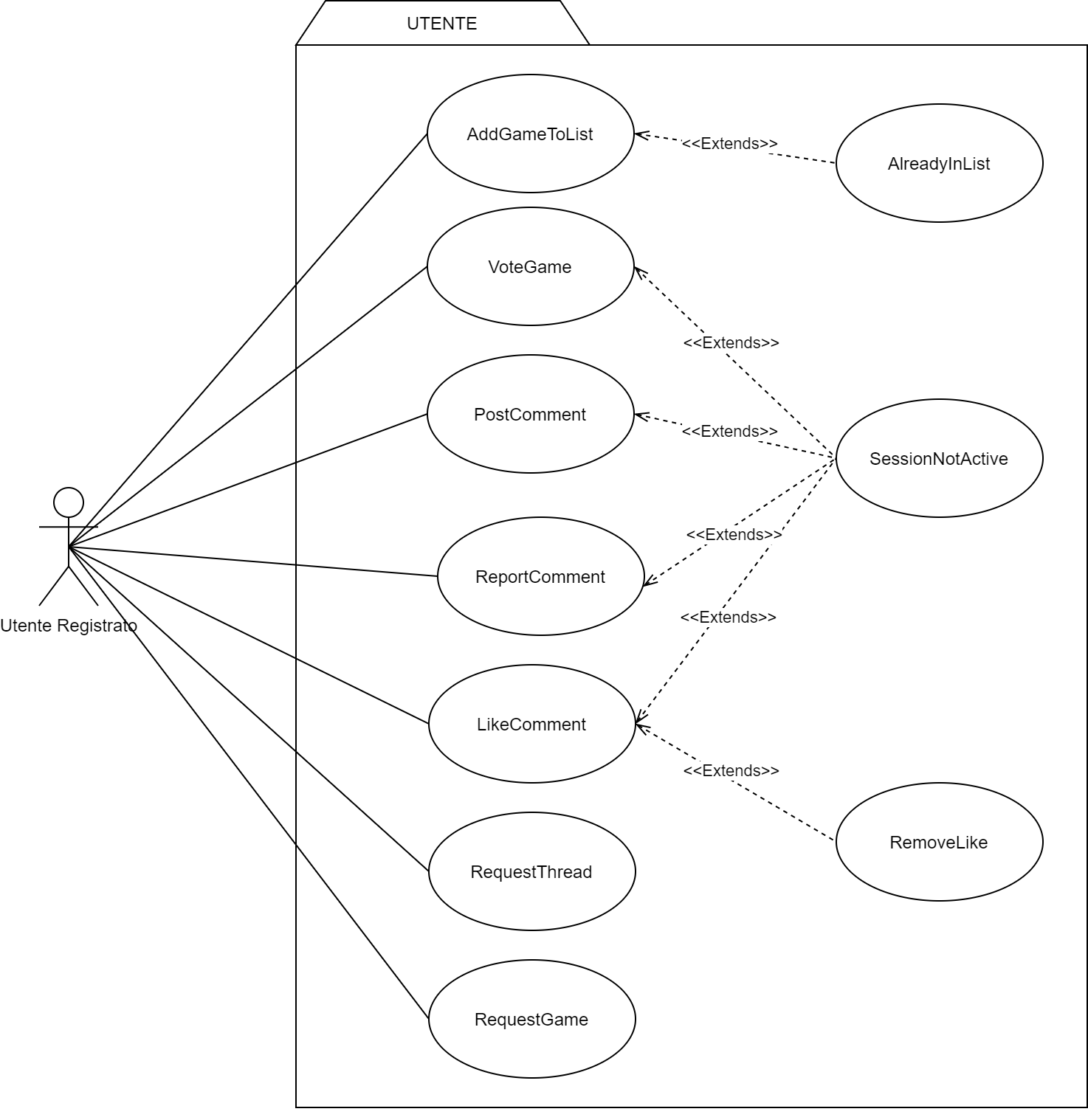
#### **-UCD\_GENERALE**



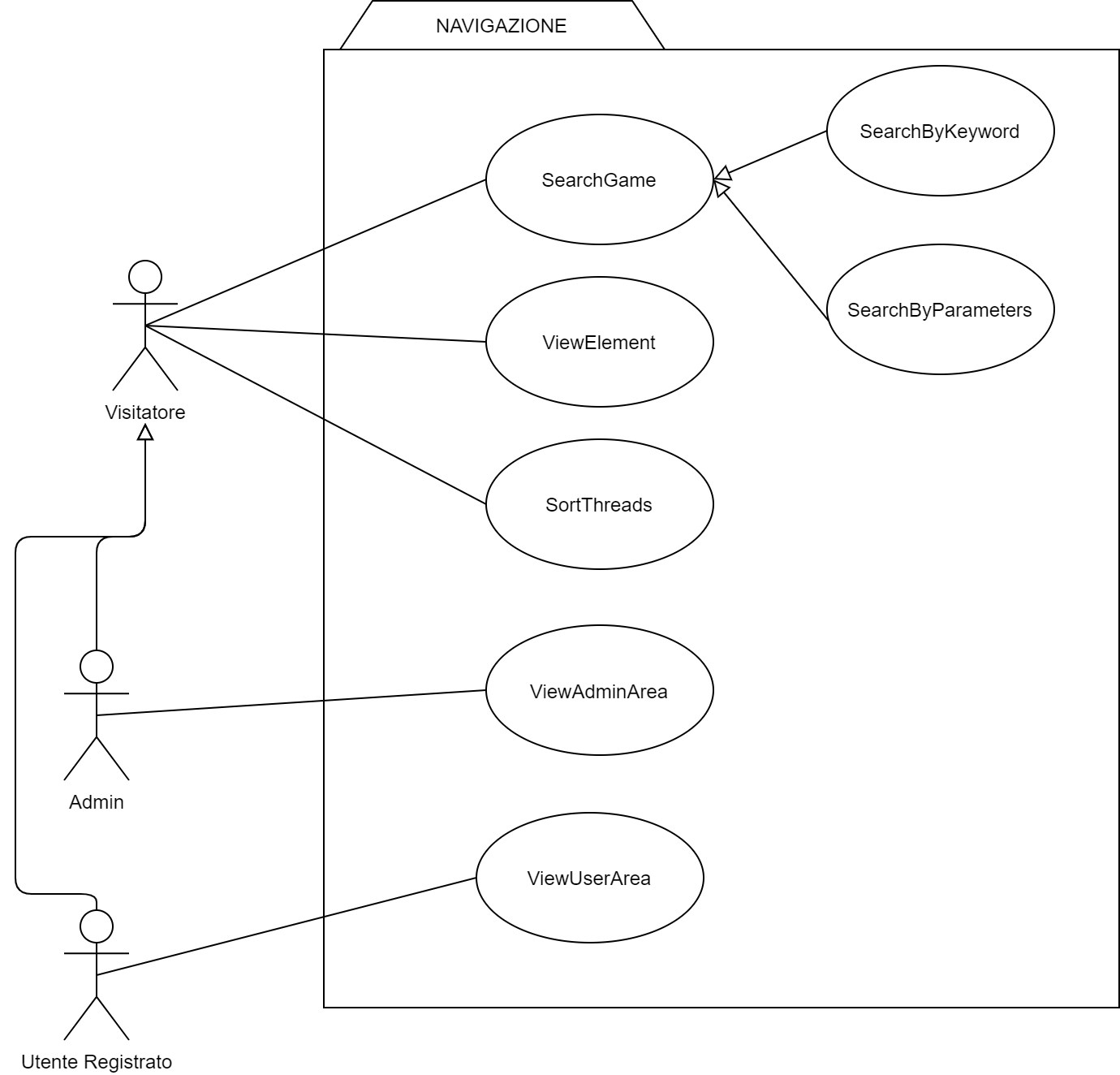
#### **-UCD\_ADMIN**



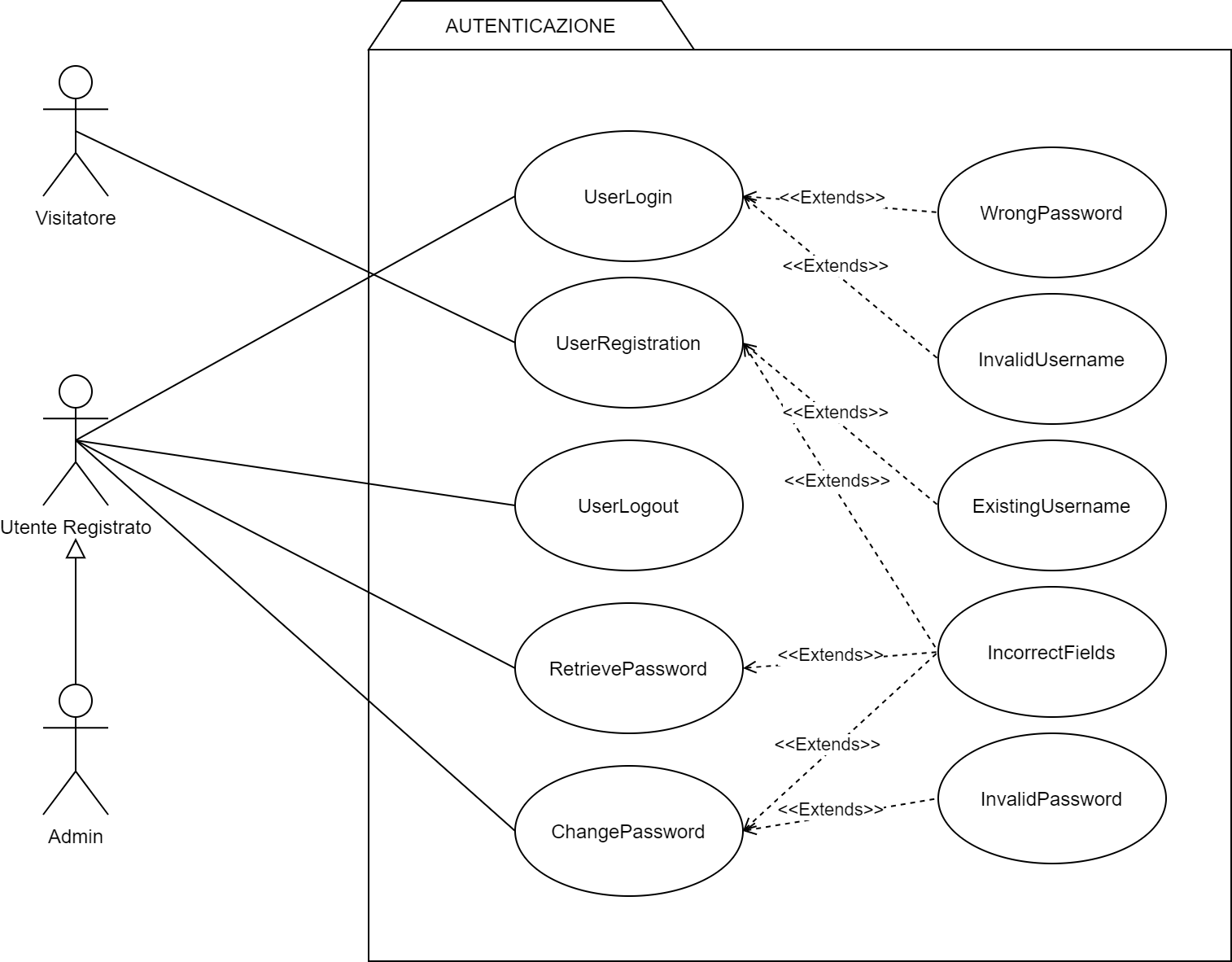
#### **-UCD\_USER**



#### **-UCD\_NAV**



#### **-UCD\_AUT**



#### **3.4.2.2 Use Cases**

UC\_AUT\_1:

|  |  |
| --- | --- |
| Nome del caso d’uso | UserRegistration |
| Attori partecipanti | Visitatore |
| Entry condition | Questo caso d’uso inizia quando un visitatore seleziona la funzionalità di registrazione. |
| Flusso degli eventi | 1. Il visitatore inserisce in un form il nome utente, la password, la data di nascita, il suo indirizzo e-mail. 2. Il visitatore accetta i termini e le condizioni. 3. Dopo aver inserito i dati, clicca sul pulsante “registrati”. 4. Viene effettuato un controllo sulla correttezza dei dati. 5. I dati inviati vengono validati. 6. Se i dati sono corretti l’utente viene registrato. |
| Exit condition | Questo caso d’uso termina quando l’utente si è correttamente registrato e viene quindi reindirizzato alla homepage da utente registrato. |
| Eccezioni | Se all’occorrenza del punto 4 uno dei campi non è stato inserito correttamente, inizierà il caso d’uso “IncorrectFields” (UC\_AUT\_1.1).  Se all’occorrenza del punto 5 il nome utente è già stato usato, inizierà il caso d’uso “ExistingUsername” (UC\_AUT\_1.2). |
|  |  |
|  |  |

UC\_AUT\_1.1:

|  |  |
| --- | --- |
| Nome del caso d’uso | IncorrectFields |
| Attori partecipanti | Utente qualsiasi |
| Entry condition | Questo caso d’uso inizia quando l’utente ha inserito in un form dei dati che non rispettano il formato corretto. |
| Flusso degli eventi | 1. L’operazione di submit del form viene impedita. 2. Il Sistema mostra una notifica a schermo indicando all’utente che I campi del form non sono stati compilati correttamente. |
| Exit condition | Questo caso d’uso termina quando l’utente inserisce i dati che rispettano il formato corretto. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_AUT\_1.2:

|  |  |
| --- | --- |
| Nome del caso d’uso | ExistingUsername |
| Attori partecipanti | Visitatore |
| Entry condition | Questo caso d’uso inizia quando il visitatore prova a registrarsi con un nome utente già presente nel sistema. |
| Flusso degli eventi | 1. L’operazione di registrazione viene impedita.  2. Il Sistema mostra una notifica a schermo indicando al visitatore che il nome utente immesso è già stato usato. |
| Exit condition | Questo caso d’uso termina quando il visitatore effettua correttamente la registrazione. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_AUT\_2:

|  |  |
| --- | --- |
| Nome del caso d’uso | UserLogin |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando un utente registrato seleziona la funzionalità di login. |
| Flusso degli eventi | 1. L’utente, che non ha una sessione attiva, compila il form di login con le sue credenziali. 2. Dopo aver riempito I campi, conferma. 3. Viene effettuato un controllo sulla correttezza dei dati. 4. Le credenziali inviate vengono validate. 5. Se I dati sono corretti, l’utente accede al sito. |
| Exit condition | Questo caso d’uso termina quando l’utente ha effettuato correttamente l’accesso al sito |
| Eccezioni | Se l’indirizzo email inserito non esiste, inizia il caso d’uso “InvalidEmail”(UC\_AUT\_2.2).  Se la password non è corretta, inizia il caso d’uso “WrongPassword”(UC\_AUT\_2.3). |
|  |  |
|  |  |

UC\_AUT\_2.1:

|  |  |
| --- | --- |
| Nome del caso d’uso | SessionNotActive |
| Attori partecipanti | Visitatore |
| Entry condition | Questo caso d’uso inizia quando l’utente, che non ha una sessione attiva, richiede di effettuare un’azione per cui è necessaria avere una sessione attiva. |
| Flusso degli eventi | 1. Reindirizzamento dell’utente alla pagina di login. 2. Visualizzazione della notifica a schermo “sessione non attiva”. |
| Exit condition | Questo caso d’uso termina immediatamente, iniziando il caso d’uso “UserLogin” (UC\_AUT\_2). |
| Eccezioni |  |
|  |  |
|  |  |

UC\_AUT\_2.2:

|  |  |
| --- | --- |
| Nome del caso d’uso | InvalidEmail |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente registrato inserisce un indirizzo email non corretto nel form di login. |
| Flusso degli eventi | 1. L’operazione di login viene impedita. 2. Il Sistema mostra una notifica a schermo indicando all’utente che l’indirizzo email non è valido. |
| Exit condition | Questo caso d’uso termina quando l’utente inserisce un indirizzo email valido ed effettua il login. |
| Eccezioni |  |
|  |  |
|  |  |

**UC\_AUT\_2.3:**

|  |  |
| --- | --- |
| Nome del caso d’uso | WrongPassword |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente inserisce una password sbagliata al momento del login. |
| Flusso degli eventi | 1. L’operazione di login non viene effettuata. 2. Visualizzazione della notifica a schermo “password errata”. |
| Exit condition | Questo caso d’uso termina quando l’utente inserisce la password corretta ed effettua il login |
| Eccezioni |  |
|  |  |
|  |  |

UC\_AUT\_3:

|  |  |
| --- | --- |
| Nome del caso d’uso | UserLogout |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente, che ha una sessione attiva, seleziona la funzionalità di logout |
| Flusso degli eventi | 1. L’utente clicca sul button relativo al logout. 2. Sullo schermo appare un messaggio che conferma l’avvenuta disconnessione. |
| Exit condition | Questo caso d’uso termina quando l’utente è nella homepage senza una sessione attiva. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_AUT\_4:

|  |  |
| --- | --- |
| Nome del caso d’uso | RetrievePassword |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente, che non ha una sessione attiva, clicca sul pulsante “recupera password”. |
| Flusso degli eventi | 1. L’utente indica su un form l’indirizzo e-mail di recupero (obbligatoria) dove inviare il link per rinnovare la password. 2. Dopo aver inserito correttamente i dati, essi vengono inviati e validati. 3. Se l’indirizzo e-mail è corretto e corrisponde a quello di un utente registrato, viene inviata una e-mail per il recupero della password all’indirizzo e-mail specificato. |
| Exit condition | Questo caso d’uso termina quando la richiesta per la modifica della password viene correttamente inviata all’email di recupero specificata. |
| Eccezioni | Se l’utente non specifica l’indirizzo e-mail nel campo obbligatorio, inizierà il caso d’uso “IncorrectFields” (UC\_AUT\_1.1) |
|  |  |
|  |  |

UC\_AUT\_5:

|  |  |
| --- | --- |
| Nome del caso d’uso | ChangePassword |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente, che ha una sessione attiva, si trova nella schermata per il rinnovo della password. |
| Flusso degli eventi | 1. L’utente compila un form con due campi in cui inserire la nuova password e confermare la stessa (entrambi obbligatori). 2. Il submit viene effettuato. |
| Exit condition | Questo caso d’uso termina quando la password viene modificata correttamente. |
| Eccezioni | Se la password immessa nel campo di conferma è diversa da quella proposta, inizierà il caso d’uso “IncorrectFields” (UC\_AUT\_1.1).  Se almeno uno dei due campi del form è vuoto, inizierà il caso d’uso “IncorrectFields” (UC\_AUT\_1.1).  Se la password non sarà composta da almeno 6 caratteri, inizierà il caso d’uso “IncorrectFields” (UC\_AUT\_1.1).  Se la nuova password corrisponde alla precedente, inizierà il caso d’uso “InvalidPassword” (UC\_AUT\_5.1). |
|  |  |
|  |  |

UC\_AUT\_5.1:

|  |  |
| --- | --- |
| Nome del caso d’uso | InvalidPassword |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente registrato ha cercato di rinnovare la sua password inserendo quella precedente. |
| Flusso degli eventi | 1. Il Sistema non procederà alla modifica della password. 2. L’utente verrà notificato del fatto che la password è uguale alla precedente. |
| Exit condition | Questo caso d’uso termina quando l’utente modifica correttamente la password. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_NAV\_1:

|  |  |
| --- | --- |
| Nome del caso d’uso | SearchGame |
| Attori partecipanti | Utente qualsiasi |
| Entry condition | Questo caso d’uso inizia quando l’utente ha accesso alla modalità di ricerca. |
| Flusso degli eventi | 1. L’utente decide di effettuare una ricerca per parola chiave, inizia il caso d’uso “SearchByKeyword” (UC\_NAV\_1.2); oppure, l’utente decide di effettuare una ricerca per parametri, inizia il caso d’uso “SearchByParameters” (UC\_NAV\_1.3). |
| Exit condition | Questo caso d’uso termina quando l’utente è sulla pagina che mostra i titoli correlati alla sua ricerca. |
| Eccezioni | Se la ricerca effettuata non porta a nessun risultato, inizierà il caso d’uso “EmptySearch” (UC\_NAV\_1.1) |
|  |  |
|  |  |

UC\_NAV\_1.1:

|  |  |
| --- | --- |
| Nome del caso d’uso | EmptySearch |
| Attori partecipanti | Utente qualsiasi |
| Entry condition | Questo caso d’uso inizia quando la ricerca effettuata dall’utente non porta a nessun risultato. |
| Flusso degli eventi | 1. Il Sistema mostra una pagina dei risultati vuota. 2. L’utente verrà notificato del fatto che la ricerca non contiene risultati. |
| Exit condition | Questo caso d’uso termina quando l’utente è sulla pagina dei risultati della ricerca |
| Eccezioni |  |
|  |  |
|  |  |

UC\_NAV\_1.2:

|  |  |
| --- | --- |
| Nome del caso d’uso | SearchByKeyword |
| Attori partecipanti | Utente qualsiasi |
| Entry condition | Questo caso d’uso inizia quando l’utente ha effettuato una ricerca per parola chiave. |
| Flusso degli eventi | 1. Il Sistema effettua una ricerca in base alla parola chiave immessa dall’utente. 2. L’utente viene reindirizzato alla pagina che mostra i risultati correlati ad essa. |
| Exit condition | Questo caso d’uso termina quando l’utente è sulla pagina dei risultati di ricerca |
| Eccezioni |  |
|  |  |
|  |  |

UC\_NAV\_1.3:

|  |  |
| --- | --- |
| Nome del caso d’uso | SearchByPublisher |
| Attori partecipanti | Utente qualsiasi |
| Entry condition | Questo caso d’uso inizia quando l’utente ha effettuato una ricerca per publisher (editore del gioco). |
| Flusso degli eventi | 1. Il Sistema effettua una ricerca in base al parametro selezionato dall’utente. 2. L’utente viene reindirizzato alla pagina che mostra i risultati correlati ad esso. |
| Exit condition | Questo caso d’uso termina quando l’utente è sulla pagina dei risultati della ricerca. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_NAV\_1.4:

|  |  |
| --- | --- |
| Nome del caso d’uso | SearchByGenre |
| Attori partecipanti | Utente qualsiasi |
| Entry condition | Questo caso d’uso inizia quando l’utente ha effettuato una ricerca per genere |
| Flusso degli eventi | 1. Il Sistema effettua una ricerca in base al parametro selezionato dall’utente. 2. L’utente viene reindirizzato alla pagina che mostra i risultati correlati ad esso. |
| Exit condition | Questo caso d’uso termina quando l’utente è sulla pagina dei risultati della ricerca. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_NAV\_2:

|  |  |
| --- | --- |
| Nome del caso d’uso | ViewGameInfo |
| Attori partecipanti | Utente qualsiasi |
| Entry condition | Questo caso d’uso inizia quando l’utente è su una pagina in cui sono presenti link che indirizzano a pagine di giochi. |
| Flusso degli eventi | 1. L’utente clicca sulla voce di un determinato gioco. 2. Il sistema mostra la pagina relativa al gioco, con tutte le informazioni su di esso. |
| Exit condition | Questo caso d’uso termina quando l’utente è sulla pagina richiesta |
| Eccezioni |  |
|  |  |
|  |  |

UC\_NAV\_3:

|  |  |
| --- | --- |
| Nome del caso d’uso | ViewThread |
| Attori partecipanti | Utente qualsiasi |
| Entry condition | Questo caso d’uso inizia quando l’utente clicca sul nome di un thread. |
| Flusso degli eventi | 1. L’utente clicca sulla voce di una determinata discussione relativa al gioco in questione. 2. Il Sistema mostra la pagina relativa all’intera discussione. |
| Exit condition | Questo caso d’uso termina quando l’utente è sulla pagina rischiesta |
| Eccezioni |  |
|  |  |
|  |  |

UC\_NAV\_4:

|  |  |
| --- | --- |
| Nome del caso d’uso | SortThreads |
| Attori partecipanti | Utente qualsiasi |
| Entry condition | Questo caso d’uso inizia quando l’utente è sulla pagina di un gioco e ha accesso alle modalità di ricerca delle discussioni per parametro. |
| Flusso degli eventi | 1. L’utente seleziona determinati parametri dall’apposito menu. 2. Il sistema effettua una ricerca in base ai parametri selezionati dall’utente. 3. Il sistema mostra le discussioni correlate ai parametri, se sono presenti. |
| Exit condition | Questo caso d’uso termina quando la lista delle discussioni è stata aggiornata correttamente. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_USER\_1:

|  |  |
| --- | --- |
| Nome del caso d’uso | AddGameToList |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente è sulla pagina di un gioco e decide di aggiungerlo alla propria lista. |
| Flusso degli eventi | 1. Selezione della categoria alla quale aggiungere il gioco. 2. Aggiunta del gioco. 3. Calcolo del punteggio abilità e aggiornamento del relativo dato utente. 4. Aggiornamento delle statistiche sulla pagina del gioco. 5. Il sistema mostra una notifica a schermo per indicare che il gioco è stato aggiunto alla lista. |
| Exit condition | Questo caso d’uso termina quando il gioco è correttamente inserito nella lista e le statistiche aggiornate. |
| Eccezioni | Se il gioco è già presente nella lista con la categoria selezionata, inizierà il caso d’uso “AlreadyInList” (UC\_USER\_1.1).  Se il gioco era già presente nella lista ma con una categoria diversa, inizierà il caso d’uso “ChangeCategoryInList” (UC\_USER\_1.2). |
|  |  |
|  |  |

**UC\_USER\_1.1:**

|  |  |
| --- | --- |
| Nome del caso d’uso | AlreadyInList |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente tenta di aggiungere un gioco già presente nella lista con la medesima categoria. |
| Flusso degli eventi | 1. L’operazione di aggiunta non viene effettuata. 2. Il sistema mostra una notifica a schermo per indicare che il gioco era già presente nella lista con la medesima categoria. |
| Exit condition | Questo caso d’uso termina immediatamente, e la posizione del gioco nella lista e le statistiche dell’utente non vengono modificate. |
| Eccezioni |  |
|  |  |
|  |  |

**UC\_USER\_1.2:**

|  |  |
| --- | --- |
| Nome del caso d’uso | ChangeCategoryInList |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente tenta di aggiungere un gioco già presente nella lista ma in una categoria differente. |
| Flusso degli eventi | 1. Il gioco rimarrà nella lista dell’utente ma la sua categoria verrà aggiornata. 2. Il sistema aggiorna il punteggio dell’utente e le statistiche del gioco aggiunto in base alla categoria selezionata. 3. Il sistema mostra una notifica a schermo per indicare che la categoria è stata aggiornata. |
| Exit condition | Questo caso d’uso termina immediatamente, e la sua categoria viene aggiornata. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_USER\_1.3:

|  |  |
| --- | --- |
| Nome del caso d’uso | RemoveElementFromList |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente è sulla pagina di un gioco presente nella sua lista e decide di rimuoverlo cliccando sull’apposito button. |
| Flusso degli eventi | 1. L’utente clicca sull’apposito bottone per rimuovere il gioco dalla lista 2. Il sistema aggiorna la lista dell’utente e i dati del gioco in questione. |
| Exit condition | Questo caso d’uso termina quando il gioco viene rimosso dalla lista dell’utente e le sue statistiche aggiornate. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_USER\_2:

|  |  |
| --- | --- |
| Nome del caso d’uso | VoteGame |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente, che ha una sessione attiva, è sulla pagina del gioco. |
| Flusso degli eventi | 1. Votazione dell’elemento con un valore in decimi. 2. Il sistema calcola la media dei voti per quel determinato gioco. |
| Exit condition | Questo caso d’uso termina quando la media dei voti del gioco viene aggiornata e il voto dell’utente viene salvato |
| Eccezioni | Se l’utente non ha una sessione attiva, inizierà il caso d’uso “SessionNotActive” (UC\_AUT\_2.1) |
|  | Se l’utente aveva già votato il gioco, inizierà il caso d’uso “OverwriteVote” (UC\_USER\_2.1) |
|  |  |
|  |  |

UC\_USER\_2.1:

|  |  |
| --- | --- |
| Nome del caso d’uso | OverwriteVote |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente valuta un gioco per il quale aveva già espresso un voto. |
| Flusso degli eventi | 1. Il voto precedente viene sovrascritto. 2. Il sistema calcola la media dei voti per quel determinato gioco. |
| Exit condition | Questo caso d’uso termina quando la media dei voti del gioco viene aggiornata e il nuovo voto dell’utente viene salvato |
| Eccezioni |  |
|  |  |
|  |  |

**UC\_USER\_3:**

|  |  |
| --- | --- |
| Nome del caso d’uso | PostComment |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente, che ha una sessione attiva, desidera commentare un elemento esistente. |
| Flusso degli eventi | 1. Selezione dell’elemento da commentare. 2. Scrittura del commento. 3. Il submit viene effettuato. |
| Exit condition | Questo caso d’uso termina quando il commento viene correttamente postato. |
| Eccezioni | Se l’utente non ha una sessione attiva, inizierà il caso d’uso “SessionNotActive” (UC\_AUT\_2.1) al momento del submit. |
|  |  |
|  |  |

UC\_USER\_3.1:

|  |  |
| --- | --- |
| Nome del caso d’uso | LikeComment |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente, che ha una sessione attiva, vuole lasciare un like su un commento esistente. |
| Flusso degli eventi | 1. L’utente clicca sul tasto “like” relativo al commento. 2. Se non aveva già aggiunto il like, esso viene aggiunto al numero di like totali del commento. |
| Exit condition | Questo caso d’uso termina quando il numero di like totali al commento aumenta di uno. |
| Eccezioni | Se l’utente non ha una sessione attiva, inizierà il caso d’uso “SessionNotActive”(UC\_AUT\_2.1).  Se l’utente ha già aggiunto un like al commento, si passa al caso d’uso “RemoveLike” (UC\_USER\_3.2). |
|  |  |

UC\_USER\_3.2:

|  |  |
| --- | --- |
| Nome del caso d’uso | RemoveLike |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente ha già messo like ad un commento vuole rimuoverlo. |
| Flusso degli eventi | 1. L’utente clicca sul tasto “like” relativo al commento. 2. Il like precedentemente aggiunto viene eliminato |
| Exit condition | Questo caso d’uso termina quando il like viene tolto e il numero complessivo decrementa di uno. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_USER\_4:

|  |  |
| --- | --- |
| Nome del caso d’uso | RequestThread |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente registrato è sulla pagina del gioco, esistente, e vuole iniziare una discussione su esso. |
| Flusso degli eventi | 1. L’utente clicca sulla voce “nuova discussione” nella pagina di un determinato gioco. 2. Viene richiesta la compilazione di un form in cui inserire un titolo (obbligatorio) e una tipologia (facoltativa). 3. Il submit viene effettuato. |
| Exit condition | Questo caso d’uso termina quando la richiesta per la nuova discussione, assieme allo username dell’utente che l’ha effettuata, viene inviata agli admin. |
| Eccezioni | Se l’utente non inserisce un nome nel form di richiesta della discussione, inizierà il caso d’uso “IncorrectFields” (UC\_AUT\_1.1). |
|  |  |
|  |  |

UC\_USER\_5:

|  |  |
| --- | --- |
| Nome del caso d’uso | RequestGame |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente è sulla pagina di richiesta per l’aggiunta di un nuovo gioco, che non deve essere già presente nel sito. |
| Flusso degli eventi | 1. L’utente compila un form in cui inserire un titolo(obbligatorio), il publisher del gioco, il genere, l’anno e un link a una fonte che attesta l’esistenza del gioco (obbligatorio). 2. Il submit viene effettuato |
| Exit condition | Questo caso d’uso termina quando la richiesta per l’aggiunta del gioco viene inviata agli admin. |
| Eccezioni | Se l’utente non inserisce un campo obbligatorio nel form di richiesta del gioco, inizierà il caso d’uso “IncorrectFields” (UC\_AUT\_1.1). |
|  |  |
|  |  |

UC\_USER\_6:

|  |  |
| --- | --- |
| Nome del caso d’uso | ReportComment |
| Attori partecipanti | Utente registrato |
| Entry condition | Questo caso d’uso inizia quando l’utente, che ha una sessione attiva, intende segnalare un commento esistente. |
| Flusso degli eventi | 1. L’utente clicca sul pulsante “segnala” presente sul commento. 2. L’utente segna in una checkbox nel form le motivazioni per cui sta segnalando il commento (facoltativo). 3. Il sumbit viene effettuato. |
| Exit condition | Questo caso d’uso termina quando la richiesta per la rimozione del commento viene inviata agli admin. |
| Eccezioni | Se l’utente che non ha una sessione attiva, inizierà il caso d’uso “SessionNotActive” (UC\_AUT\_2.1). |
|  |  |
|  |  |

UC\_USER\_7:

|  |  |
| --- | --- |
| Nome del caso d’uso | ViewUserArea |
| Attori partecipanti | Utente registrato e Admin |
| Entry condition | Questo caso d’uso inizia quando l’utente registrato o l’admin accede alla propria “Area Utente”. |
| Flusso degli eventi | 1. L’utente visualizza la propria area personale, contenente i propri dati, il proprio punteggio abilità, le informazioni relative alle sue liste di giochi e la lista delle discussioni da lui iniziate. |
| Exit condition | Questo caso d’uso termina quando l’utente ha visualizzato correttamente la sua area personale |
| Eccezioni |  |
|  |  |
|  |  |

UC\_ADMIN\_1:

|  |  |
| --- | --- |
| Nome del caso d’uso | AcceptThread |
| Attori partecipanti | Moderatore |
| Entry condition | Questo caso d’uso inizia quando il moderatore è sulla pagina contenente le richieste di discussioni da approvare e la richiesta in questione esiste. |
| Flusso degli eventi | 1. Il moderatore verifica la richiesta. 2. Il moderatore clicca sull’apposito button per accettare o rifiutare la richiesta. 3. Se la richiesta viene accettata, la discussione viene creata nella apposita pagina del gioco, con i dati presenti nella richiesta dell’utente. |
| Exit condition | Questo caso d’uso termina quando la richiesta viene eliminata dalla pagina delle richieste e la discussione viene creata correttamente. |
| Eccezioni | Se il moderatore rifiuta la richiesta, si passa al caso d’uso “RefuseRequest” (UC\_ADMIN\_7). |
|  |  |
|  |  |

**UC\_ADMIN\_2:**

|  |  |
| --- | --- |
| Nome del caso d’uso | AddThread |
| Attori partecipanti | Moderatore o Game Developer |
| Entry condition | Questo caso d’uso inizia quando l’utente è sulla pagina del gioco sul quale vuole cominciare una discussione. |
| Flusso degli eventi | 1. L’utente clicca sulla voce “nuova discussione” nella pagina di un determinato gioco. 2. Viene richiesta la compilazione di un form in cui inserire un titolo (obbligatorio) e una tipologia (facoltativa). 3. Il submit viene effettuato. |
| Exit condition | Questo caso d’uso termina quando la nuova discussione viene creata correttamente. |
| Eccezioni | Se l’utente non inserisce un nome nel form di creazione della discussione, inizierà il caso d’uso “IncorrectFields” (UC\_AUT\_1.1). |
|  |  |
|  |  |

UC\_ADMIN\_3:

|  |  |
| --- | --- |
| Nome del caso d’uso | AcceptGame |
| Attori partecipanti | Gestore catalogo |
| Entry condition | Questo caso d’uso inizia quando il gestore è sulla pagina contenente le richieste di nuovi giochi da approvare e la richiesta in questione esiste. |
| Flusso degli eventi | 1. Il gestore verifica la richiesta. 2. Il gestore clicca sull’apposito button per accettare o rifiutare la richiesta. 3. Se la richiesta viene accettata, il gioco viene aggiunto al sistema con i dati specificati nella richiesta. |
| Exit condition | Questo caso d’uso termina quando la richiesta viene eliminata dalla pagina delle richieste e il gioco viene aggiunto correttamente al sistema. |
| Eccezioni | Se il gestore rifiuta la richiesta, si passa al caso d’uso “RefuseRequest” (UC\_ADMIN\_7). |
|  |  |
|  |  |

UC\_ADMIN\_4:

|  |  |
| --- | --- |
| Nome del caso d’uso | AddGame |
| Attori partecipanti | Gestore catalogo o Game Developer |
| Entry condition | Questo caso d’uso inizia quando l’utente è sulla pagina per l’aggiunta di un gioco che non deve essere già presente nel sistema. |
| Flusso degli eventi | 1. L’utente compila un form in cui aggiunge tutti i campi relativi ad un determinato gioco. 2. Il submit viene effettuato. |
| Exit condition | Questo caso d’uso termina quando il gioco viene aggiunto correttamente al sistema. |
| Eccezioni | Se l’utente non inserisce un campo obbligatorio negli attributi del gioco, inizierà il caso d’uso “IncorrectFields” (UC\_AUT\_1.1). |
|  |  |
|  |  |

UC\_ADMIN\_5:

|  |  |
| --- | --- |
| Nome del caso d’uso | UpdateGame |
| Attori partecipanti | Gestore catalogo |
| Entry condition | Questo caso d’uso inizia quando il gestore è sulla pagina di modifica del gioco presente nel sistema. |
| Flusso degli eventi | 1. Il gestore compila un form in cui modifica i campi relativi ad un determinato gioco. 2. Il submit viene effettuato. |
| Exit condition | Questo caso d’uso termina quando le informazioni del gioco vengono aggiornate correttamente. |
| Eccezioni | Se il gestore lascia vuoto un campo obbligatorio negli attributi del gioco, inizierà il caso d’uso “IncorrectFields” (UC\_AUT\_1.1). |
|  |  |
|  |  |

UC\_ADMIN\_6:

|  |  |
| --- | --- |
| Nome del caso d’uso | AcceptReport |
| Attori partecipanti | Moderatore |
| Entry condition | Questo caso d’uso inizia quando il moderatore è sulla pagina contenente le richieste di commenti e/o thread da eliminare e la richiesta in questione esiste. |
| Flusso degli eventi | 1. Il moderatore verifica la richiesta. 2. Il moderatore clicca sull’apposito button per accettare o rifiutare la richiesta. 3. Se la richiesta viene accettata, il commento specificato nella richiesta viene eliminato. |
| Exit condition | Questo caso d’uso termina quando la richiesta viene eliminata e il commento in questione non è più presente. |
| Eccezioni | Se il moderatore rifiuta la richiesta, si passa al caso d’uso “RefuseRequest” (UC\_ADMIN\_7). |
|  |  |
|  |  |

UC\_ADMIN\_7:

|  |  |
| --- | --- |
| Nome del caso d’uso | RefuseRequest |
| Attori partecipanti | Gestore catalogo o moderatore |
| Entry condition | Questo caso d’uso inizia quando l’utente clicca sul buttone per rifiutare una richiesta. |
| Flusso degli eventi | 1. La richiesta viene rifiutata ed eliminata. |
| Exit condition | Questo caso d’uso termina quando la richiesta non è più presente nella relativa pagina delle richieste. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_ADMIN\_8:

|  |  |
| --- | --- |
| Nome del caso d’uso | RemoveThread |
| Attori partecipanti | Moderatore |
| Entry condition | Questo caso d’uso inizia quando il moderatore è sulla pagina relativa al thread da eliminare. |
| Flusso degli eventi | 1. Il moderatore clicca sul pulsante “rimuovi” presente nell’elemento da eliminare. 2. Il thread viene eliminato, così come tutti I suoi commenti relativi ed appare una notifica a schermo per segnalarlo. |
| Exit condition | Questo caso d’uso termina quando la discussione non è più presente nel sistema. |
| Eccezioni |  |
|  |  |
|  |  |

UC\_ADMIN\_9:

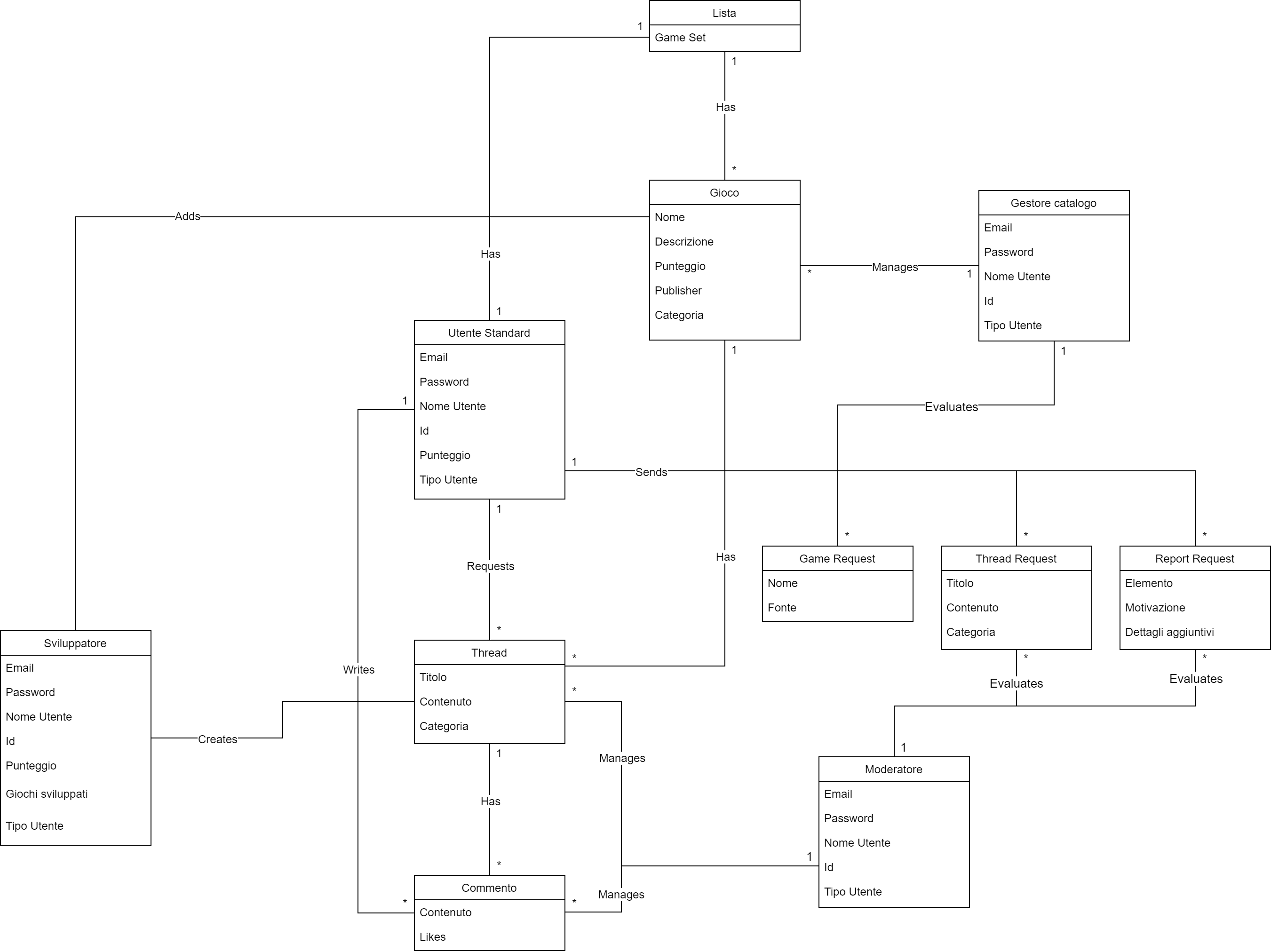
|  |  |
| --- | --- |
| Nome del caso d’uso | RemoveComment |
| Attori partecipanti | Moderatore |
| Entry condition | Questo caso d’uso inizia quando il moderatore è sulla pagina della discussione relativa al commento da eliminare |
| Flusso degli eventi | 1. Il moderatore clicca sul pulsante “rimuovi” presente nell’elemento da eliminare. 2. Il commento viene eliminato ed appare una notifica a schermo per segnalarlo. |
| Exit condition | Questo caso d’uso termina quando il commento non è più presente nella discussione. |
| Eccezioni |  |
|  |  |
|  |  |

### **3.4.3 Object Model**

|  |  |  |
| --- | --- | --- |
| Nome oggetto | Tipologia | Descrizione |
| Registration\_Boundary | Boundary | Form da compilare per la registrazione, con campi da riempire e submit |
| Login\_Boundary | Boundary | Pagina per effettuare il login da utente registrato o da admin |
| Notification\_boundary | Boundary | Notifica a schermo che appare per segnalare un cambiamento o un errore |
| Homepage | Boundary | Pagina principale visualizzata se non si ha una sessione attiva |
| LoggedHomePage | Boundary | Pagina principale a cui si viene reindirizzati una volta attivata la sessione |
| UserAreaLink |  |  |
| UserArea\_boundary | Boundary | Area accessibile dall’utente registrato, con tutte le informazioni su di esso |
| Like\_Boundary | Boundary | Icona che viene creata quando viene messo like a un commento da un utente; se cliccata permetterà di rimuoverlo |
| Comment\_boundary | Boundary | Uno specifico commento in una discussione |
| SearchBar\_boundary | Boundary | Barra di ricerca per effettuare una ricerca di giochi per parola chiave |
| GameSort\_Boundary | Boundary |  |
| ResultsPage | Boundary | Pagina che mostra i risultati di una ricerca effettuata nel sito |
| GameLink\_boundary | Boundary | Link che reindirizza alla pagina di uno specifico gioco |
| GamePage | Boundary | Pagina che mostra tutte le informazioni relative ad un determinato gioco, oltre a consentire le operazioni sullo stesso, da utente registrato |
| ThreadLink | Boundary |  |
| ThreadPage | Boundary |  |
| ThreadForm | Boundary | Form da compilare per richiedere una nuova discussione (utente registrato) o crearne una (admin) |
| GameRequestForm | Boundary | Form con cui l’utente registrato può richiedere l’aggiunta di un nuovo gioco |
| ReportRequestForm | Boundary | Form con cui l’utente registrato può richiedere la segnalazione di un commento inopportuno |
| Logout\_Boundary | Boundary | Permette di terminare la propria sessione attiva |
| RetrievePassword\_boundary | Boundary | Form da compilare per indicare l’indirizzo e-mail presso il quale inviare il link di recupero della password |
| PasswordModifier\_boundary | Boundary | Form da compilare per la modifica della password |
| ThreadRequestPage | Boundary | Pagina accessibile ai moderatori con le richieste di nuove discussioni da parte degli utenti |
| GameRequestPage | Boundary |  |
| ReportRequestPage | Boundary |  |
| NewGame\_boundary | Boundary | Form che permette all’admin di inserire un nuovo gioco nel database |
| GameUpdaterForm | Boundary | Form che permette all’admin di aggiornare i dati relativi ad un determinato gioco |
| Registration\_Control | Control | Si occupa di controllare il corretto inserimento dei campi in un form per quando riguarda la registrazione di un nuovo utente e la sua effettiva operazione |
| Login\_Control | Control | Si occupa di tutte le operazioni di controllo per il login di un utente registrato |
| Logout\_Control | Control | Si occupa di tutte le operazioni di controllo per il logout di un utente registrato avente una sessione attiva |
| RetrievePassword\_Control | Control | Si occupa di tutte le operazioni di controllo per la richiesta del recupero della password di un utente registrato |
| ChangePassword\_Control | Control | Si occupa di tutte le operazioni di controllo per la richiesta da parte di un utente registrato avente sessione attiva di cambiare la password |
| Search\_Control | Control | Consente al sistema di effettuare una ricerca tra tutti i giochi o thread presenti nel sistema secondo i parametri o la keyword inserita dall’utente, oppure per un gioco specifico nel caso di indirizzamento tramite link |
| Game\_Control | Control | Si occupa di tutte le operazioni di controllo per l’aggiunta di un nuovo gioco da parte di un admin e per la sua creazione e/o update |
| Request\_Control | Control | Si occupa di tutte le operazioni di controllo per la richiesta da parte di un utente di aggiungere un nuovo gioco, di creare un nuovo thread riguardo un gioco oppure di segnalare un commento o thread sgradito |
| Thread\_Control | Control | Si occupa di tutte le operazioni di controllo per l’aggiunta di un nuovo thread da parte di un admin e per la sua creazione |
| Vote\_control | Control |  |
| Comment\_Control | Control | Si occupa delle operazioni di controllo per l’aggiunta di un commento a una specifica discussione da parte dell’utente registrato |
| Like\_control |  | Si occupa delle operazioni di controllo per l’aggiunta di un like a un commento, come la verifica della sessione dell’utente e se un like era già stato inserito da un utente specifico |
| List\_Control | Control | Consente al sistema di organizzare le liste di un determinato utente, verificando la presenza di oggetti ed eventualmente aggiungendoli, togliendoli o spostandoli da una lista all’altra, calcolando conseguentemente il punteggio associato |
| Game | Entity | Rappresenta un singolo gioco |
| GameSet | Entity | Rappresenta un insieme di giochi |
| User | Entity | Rappresenta un singolo utente registrato, con tutte le sue informazioni e il suo punteggio abilità |
| Admin | Entity |  |
| List | Entity | Rappresenta una lista di giochi di un determinato utente |
| Vote | Entity | Rappresenta un voto aggiunto da un utente a uno specifico gioco |
| Like | Entity | Rappresenta un like inserito da un utente in un commento specifico |
| Comment | Entity | Rappresenta un commento aggiunto da un giocatore in una determinata discussione |
| Thread | Entity | Rappresenta una singola discussione creata da un utente per un gioco |
| ThreadSet | Entity | Rappresenta un insieme di discussioni per un gioco |
| GameRequest | Entity | Rappresenta una richiesta per l’aggiunta di un nuovo gioco inviata a un admin da un utente registrato |
| ThreadRequest | Entity | Rappresenta una richiesta per l’aggiunta di un nuovo thread inviata a un admin da un utente registrato |
| ReportRequest | Entity | Rappresenta una segnalazione di un commento o un thread inviata a un admin da un utente registrato |
| User\_manager | Manager | Si occupa della gestione dei dati persistenti relativi all’entità “User” |
| Game\_manager | Manager | Si occupa della gestione dei dati persistenti relativi all’entità “Game” |
| Thread\_manager | Manager | Si occupa della gestione dei dati persistenti relativi all’entità “Thread” |
| List\_manager | Manager |  |
| Vote\_manager | Manager |  |
| Comment\_manager | Manager |  |
| Like\_manager | Manager |  |
| GameRequest\_manager | Manager |  |
| ThredRequest\_manager | Manager |  |
| ReportRequest\_manager | Manager |  |
| Session\_manager | Manager |  |
| Field\_manager | Manager |  |

#### **3.4.3.1 Class Diagrams**

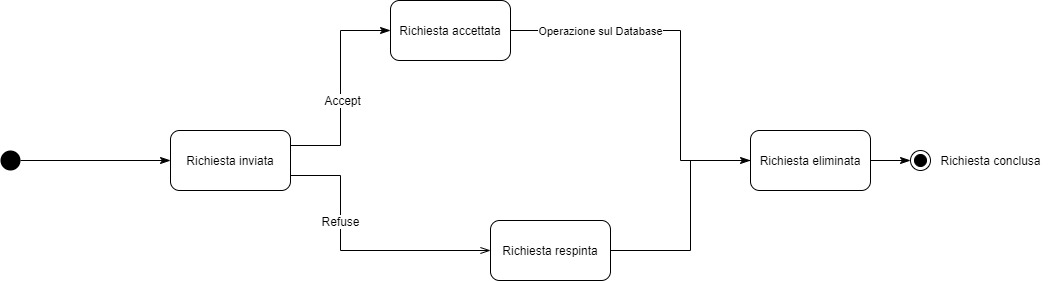
##### **- CD\_ENTITIES**



### **3.4.4 Dynamic Model**

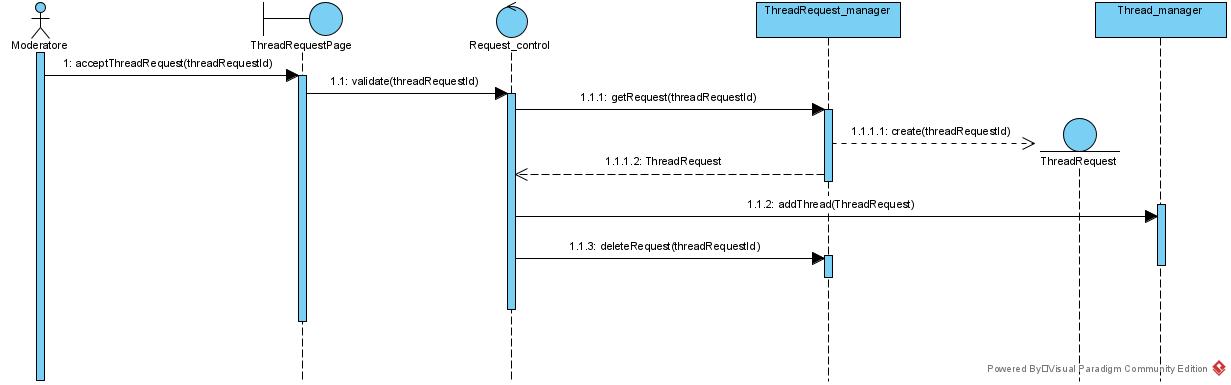
#### **3.4.4.1 Statechart Diagrams**

##### **- STATECHART RICHIESTA:**

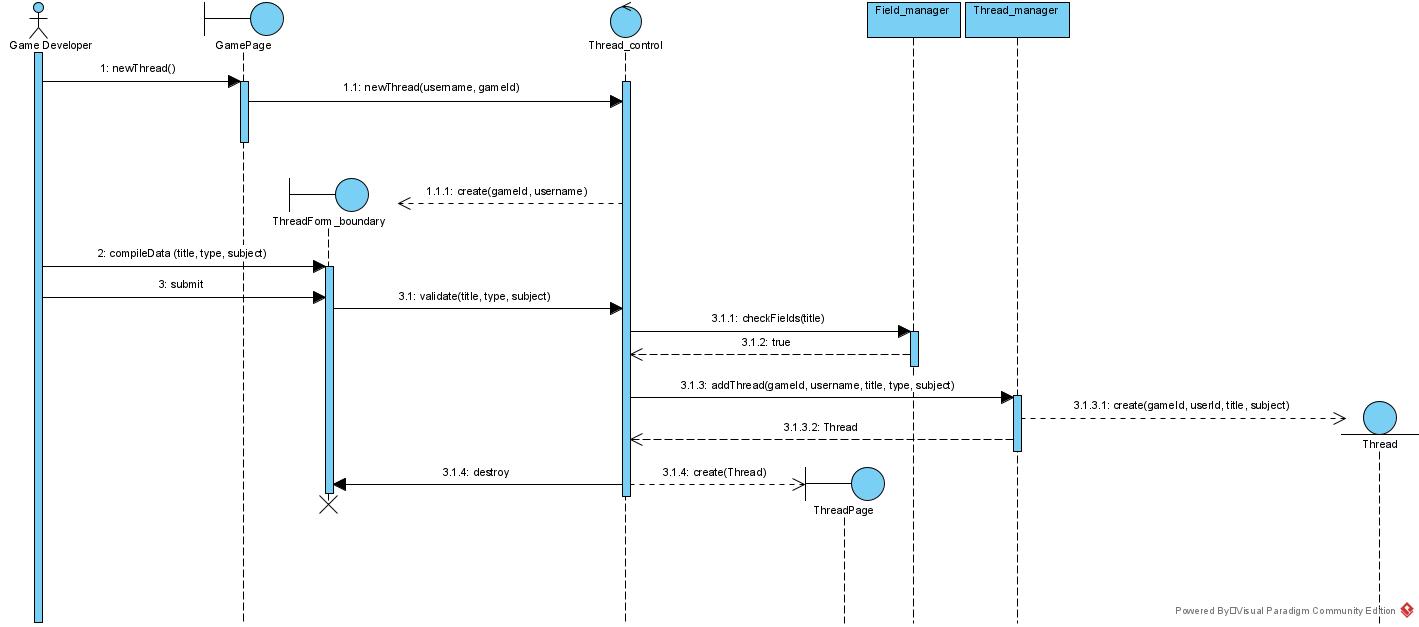


#### **3.4.4.2 Sequence Diagrams**

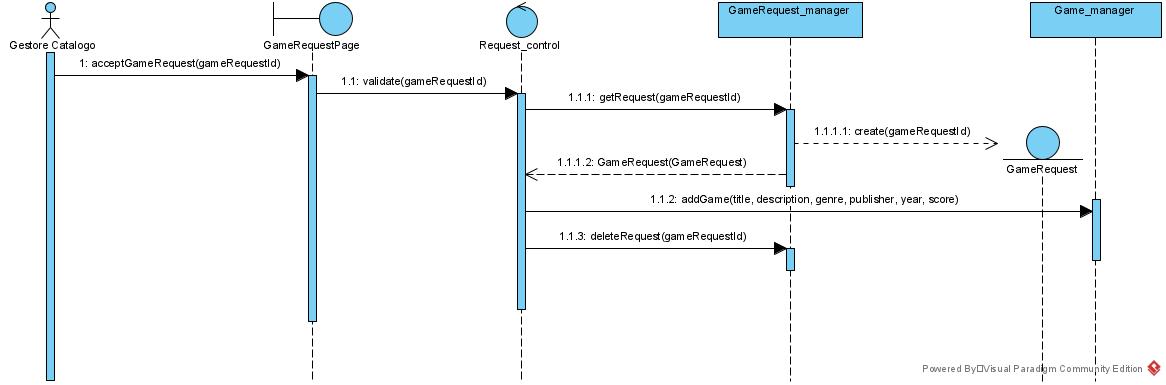
#### **-SD\_UC\_ADMIN\_1**



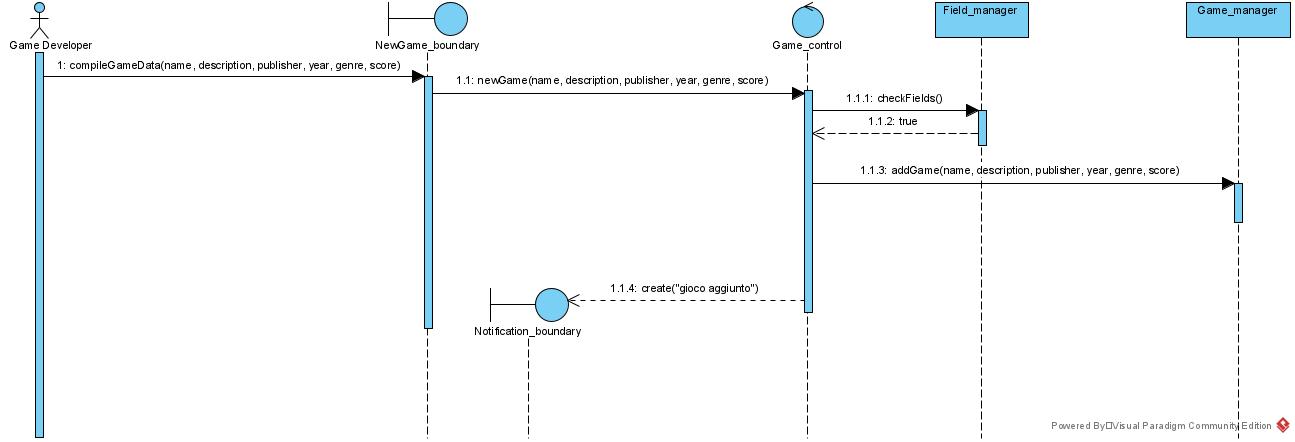
#### **-SD\_UC\_ADMIN\_2**



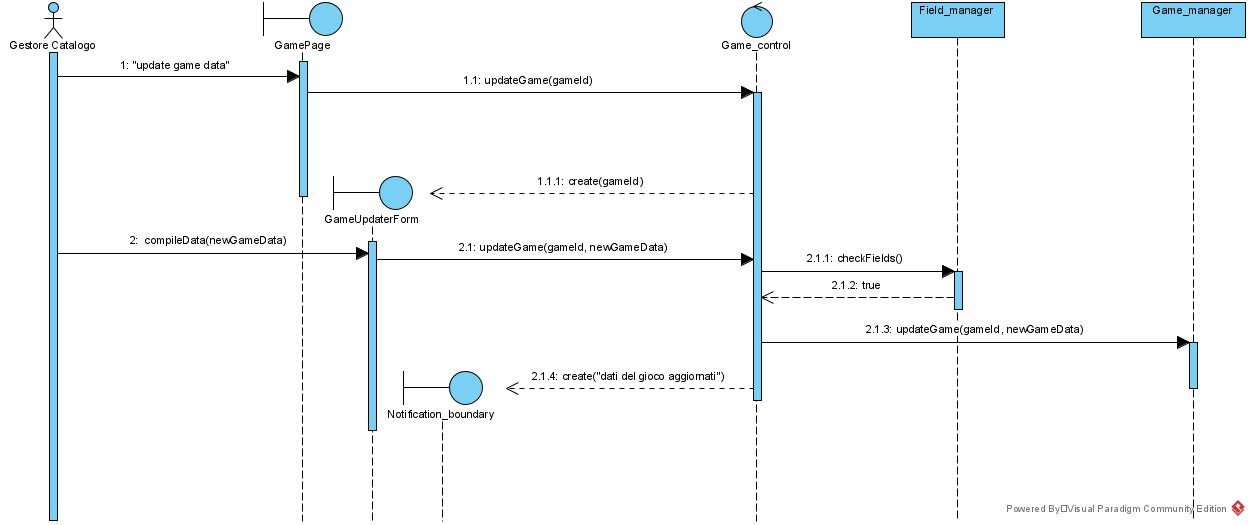
#### **-SD\_UC\_ADMIN\_3**



#### **-SD\_UC\_ADMIN\_4**



#### **-SD\_UC\_ADMIN\_5**



#### **-SD\_UC\_ADMIN\_6**

#### **-SD\_UC\_ADMIN\_7**

#### **-SD\_UC\_ADMIN\_8**

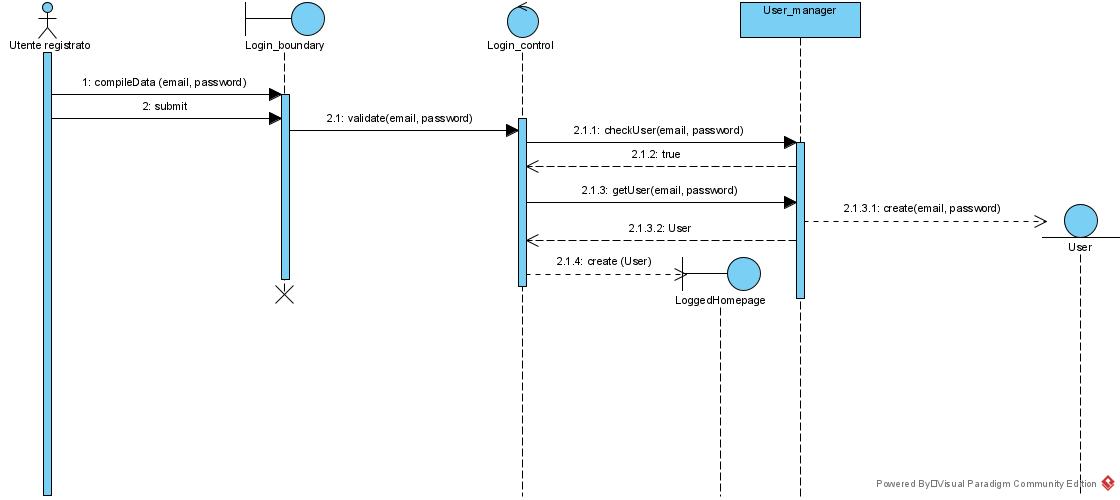
#### **-SD\_UC\_ADMIN\_9**

#### **-SD\_UC\_AUT\_1**

#### **-SD\_UC\_AUT\_1.1**

#### **-SD\_UC\_AUT\_1.2**

#### **-SD\_UC\_AUT\_2**



#### **-SD\_UC\_AUT\_2.1**

#### **-SD\_UC\_AUT\_2.2**

#### **-SD\_UC\_ADMIN\_2.3**

#### **-SD\_UC\_AUT\_3**

#### **-SD\_UC\_AUT\_4**

#### **-SD\_UC\_AUT\_4.1**

#### **-SD\_UC\_AUT\_5**

#### **-SD\_UC\_AUT\_5.1**

#### **-SD\_UC\_NAV\_1.1**

#### **-SD\_UC\_NAV\_1.2**

#### **-SD\_UC\_NAV\_1.3**

#### **-SD\_UC\_NAV\_1.4**

#### **-SD\_UC\_NAV\_2**

#### **-SD\_UC\_NAV\_3**

#### **-SD\_UC\_NAV\_4**

#### **-SD\_UC\_USER\_1**

#### **-SD\_UC\_USER\_1.1**

#### **-SD\_UC\_USER\_1.2**

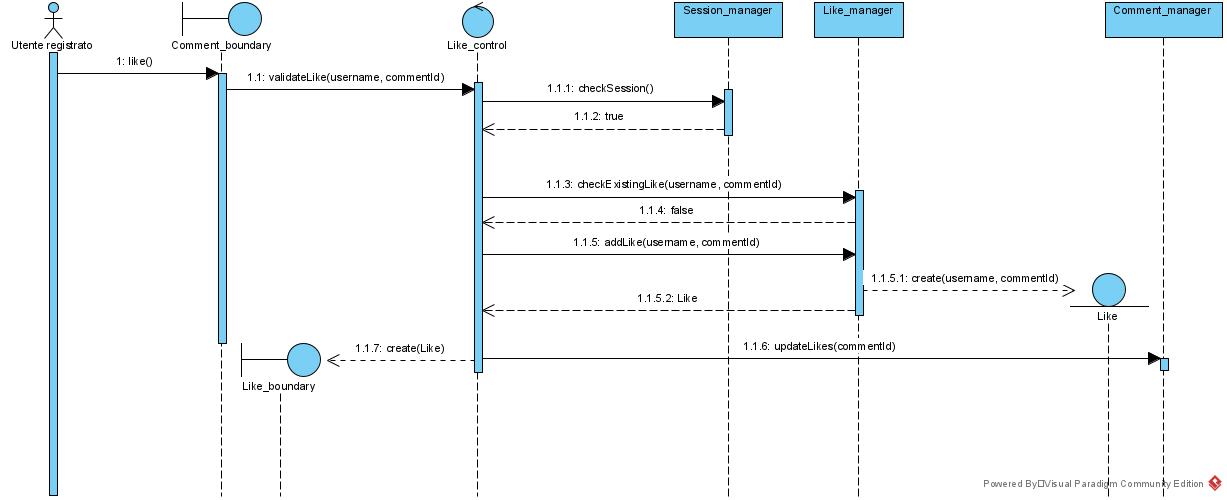
#### **-SD\_UC\_USER\_1.3**

#### **-SD\_UC\_USER\_2**

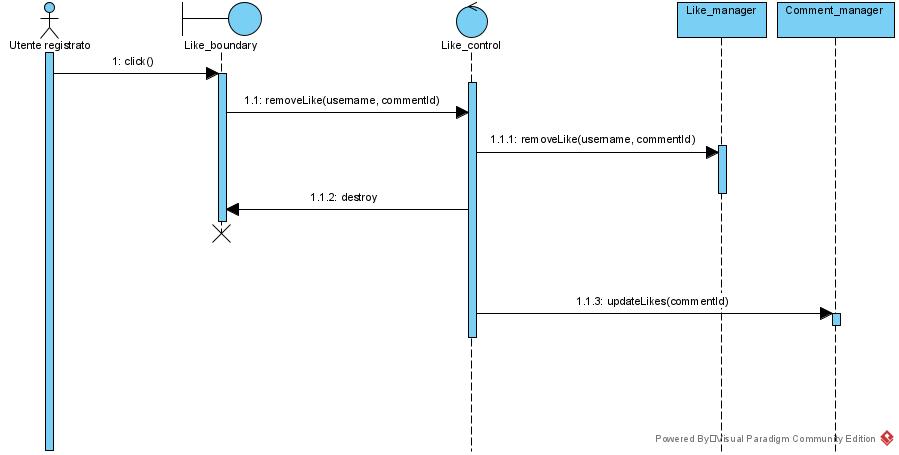
#### **-SD\_UC\_USER\_2.1**

#### **-SD\_UC\_USER\_3**

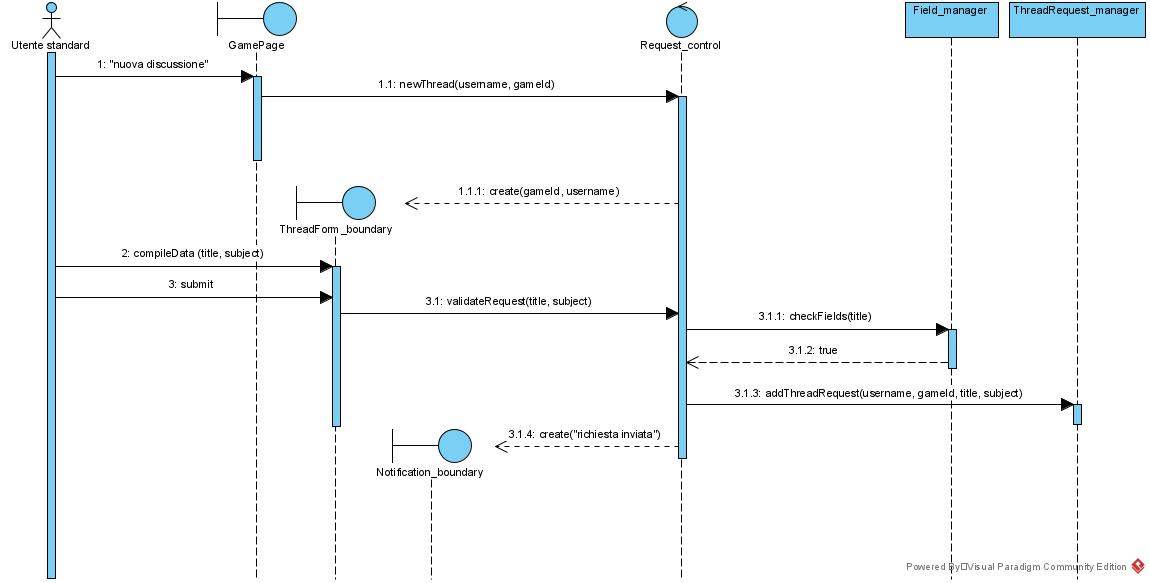
#### **-SD\_UC\_USER\_3.1**



#### **-SD\_UC\_USER\_3.2**



#### **-SD\_UC\_USER\_4**



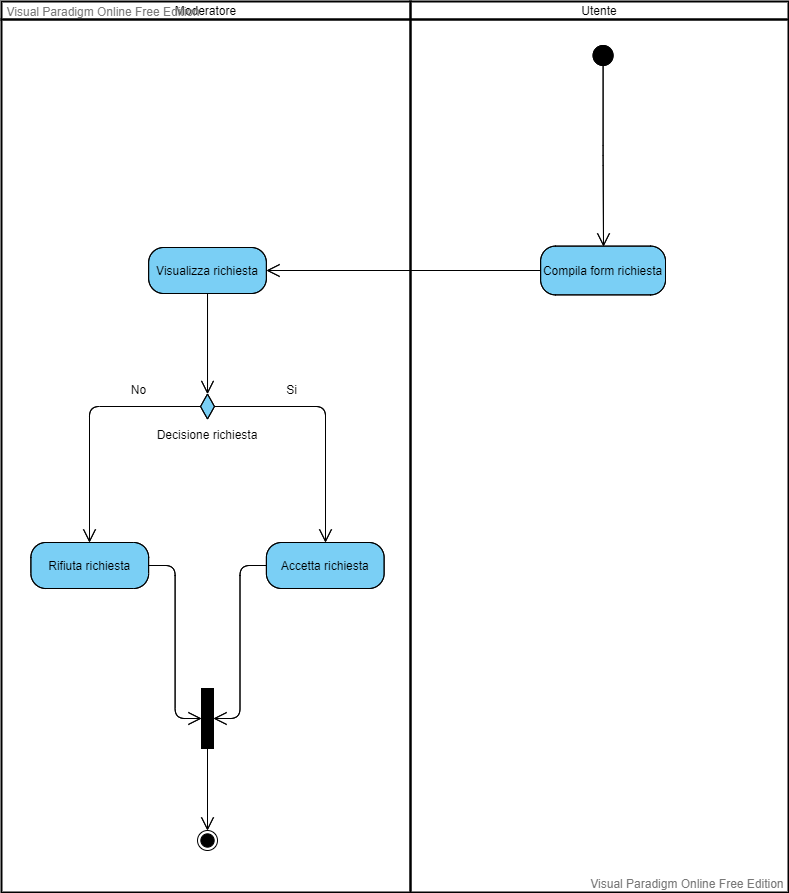
#### **-SD\_UC\_USER\_5**

#### **-SD\_UC\_USER\_6**

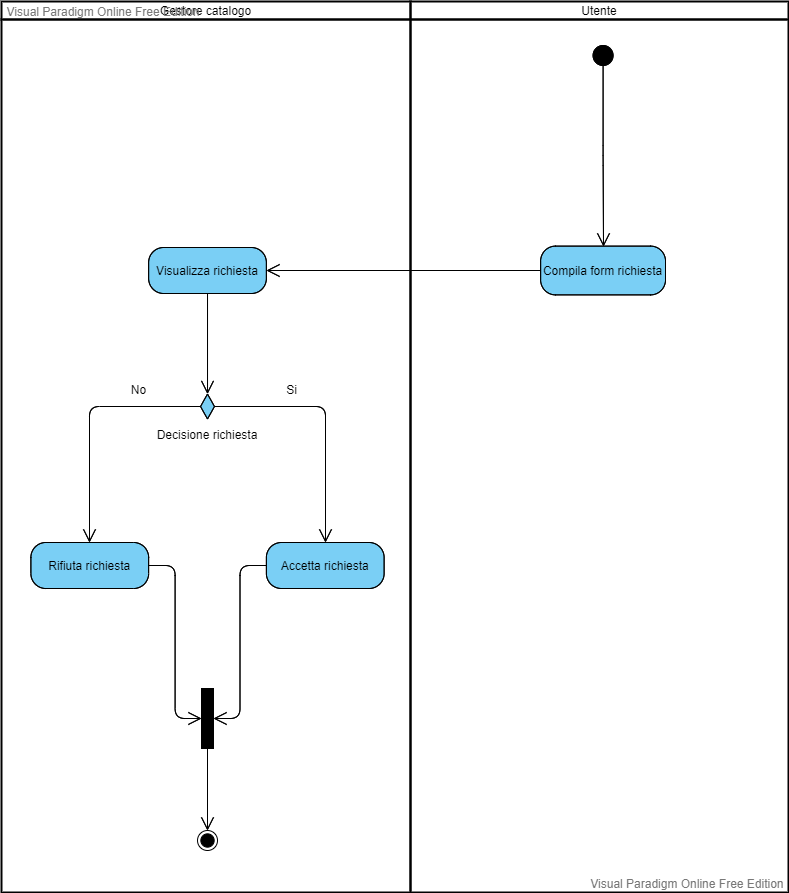
#### **-SD\_UC\_USER\_7**

#### **3.4.4.3 Activity Diagrams**

#### **-AD\_1**



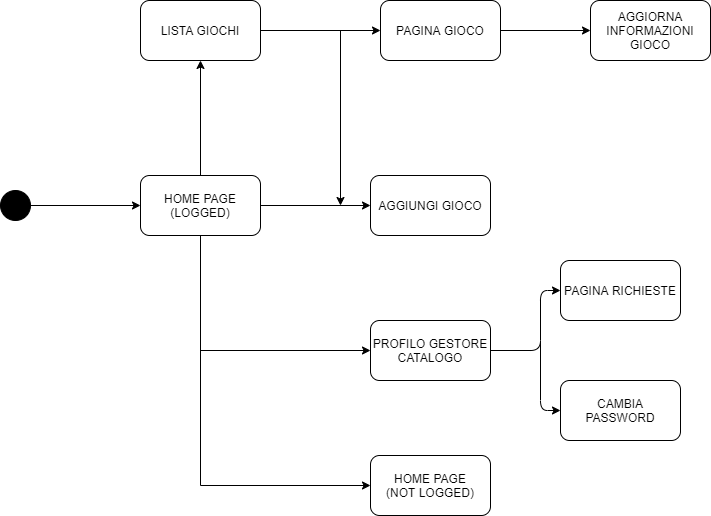
#### **-AD\_2**



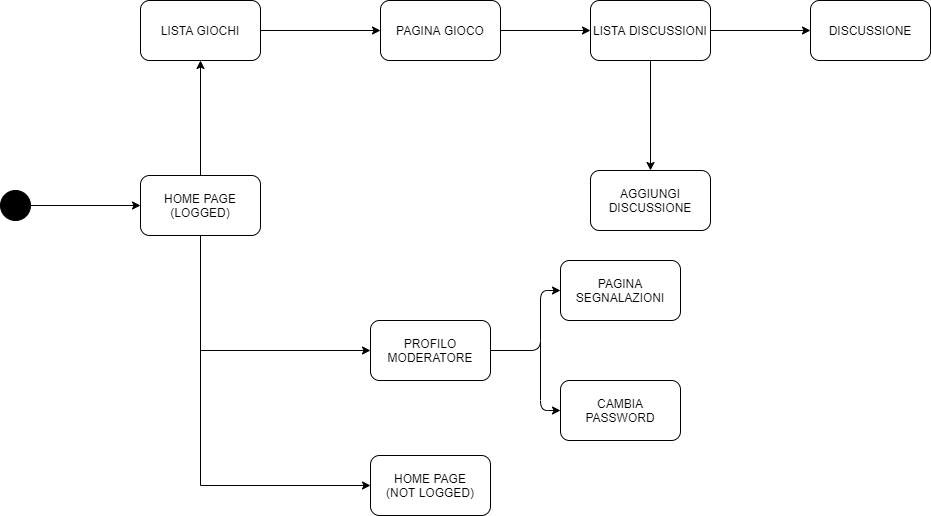
### **3.4.5 User Interface**

#### **3.4.5.1 Navigation Paths**

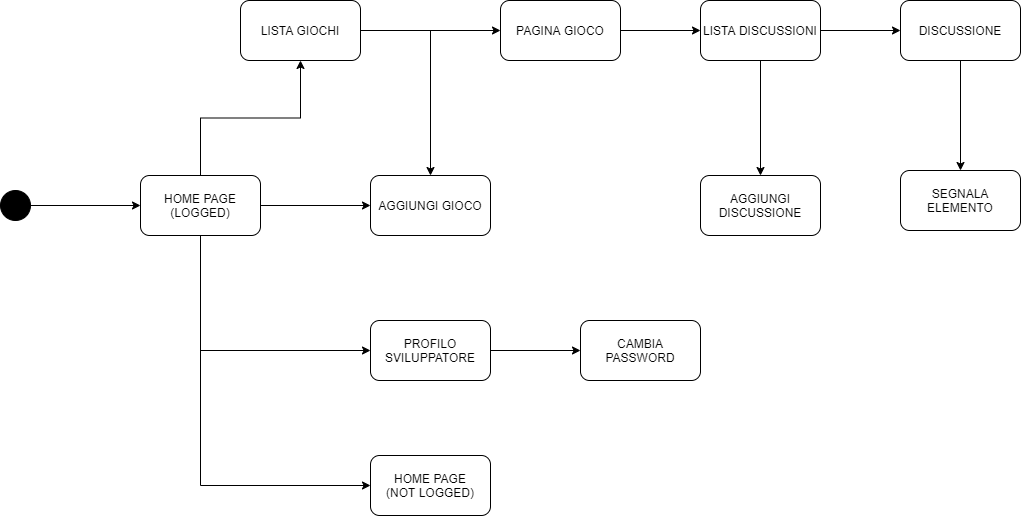
#### **-NP\_GESTORE**



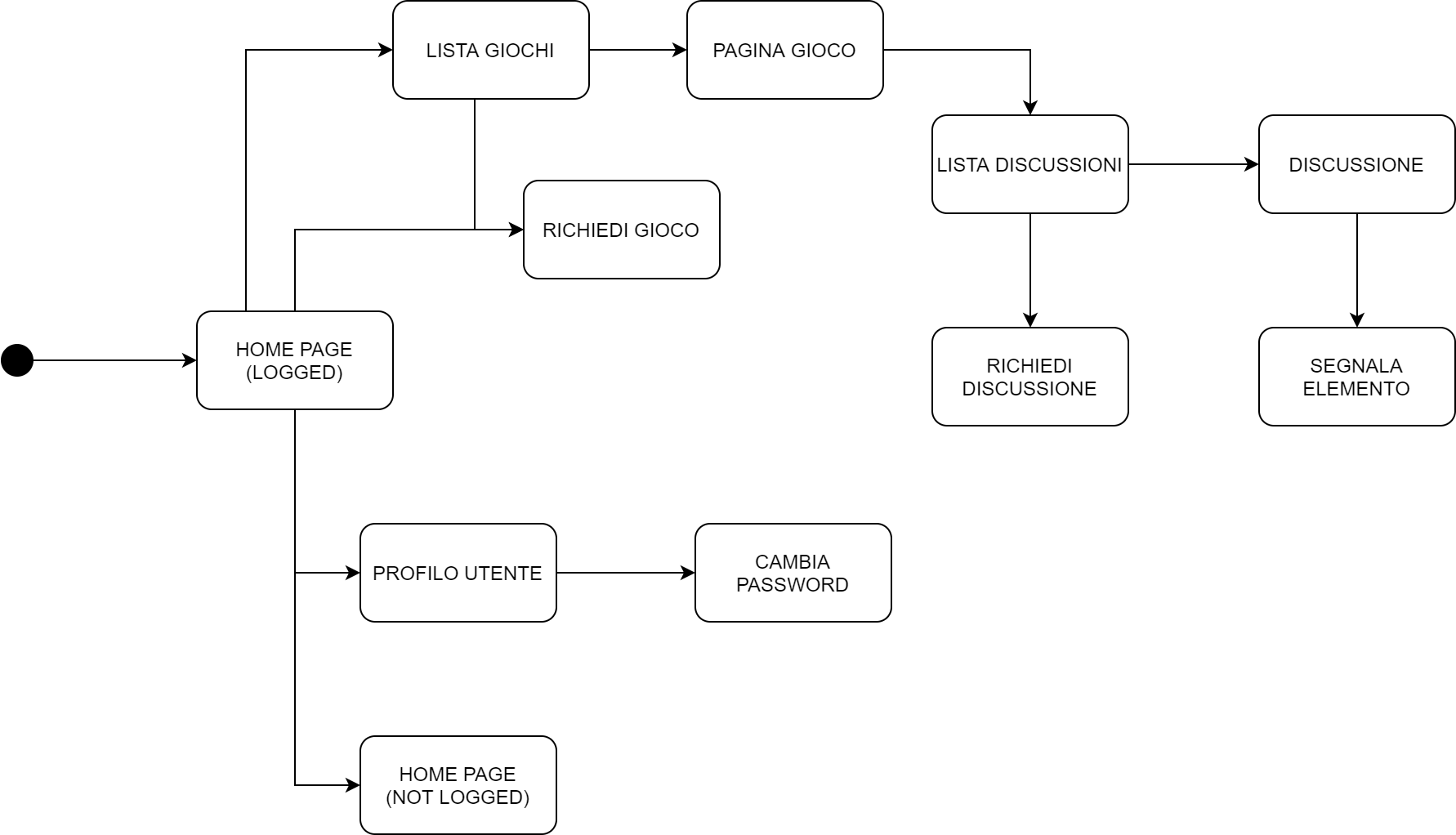
#### **-NP\_MODERATORE**



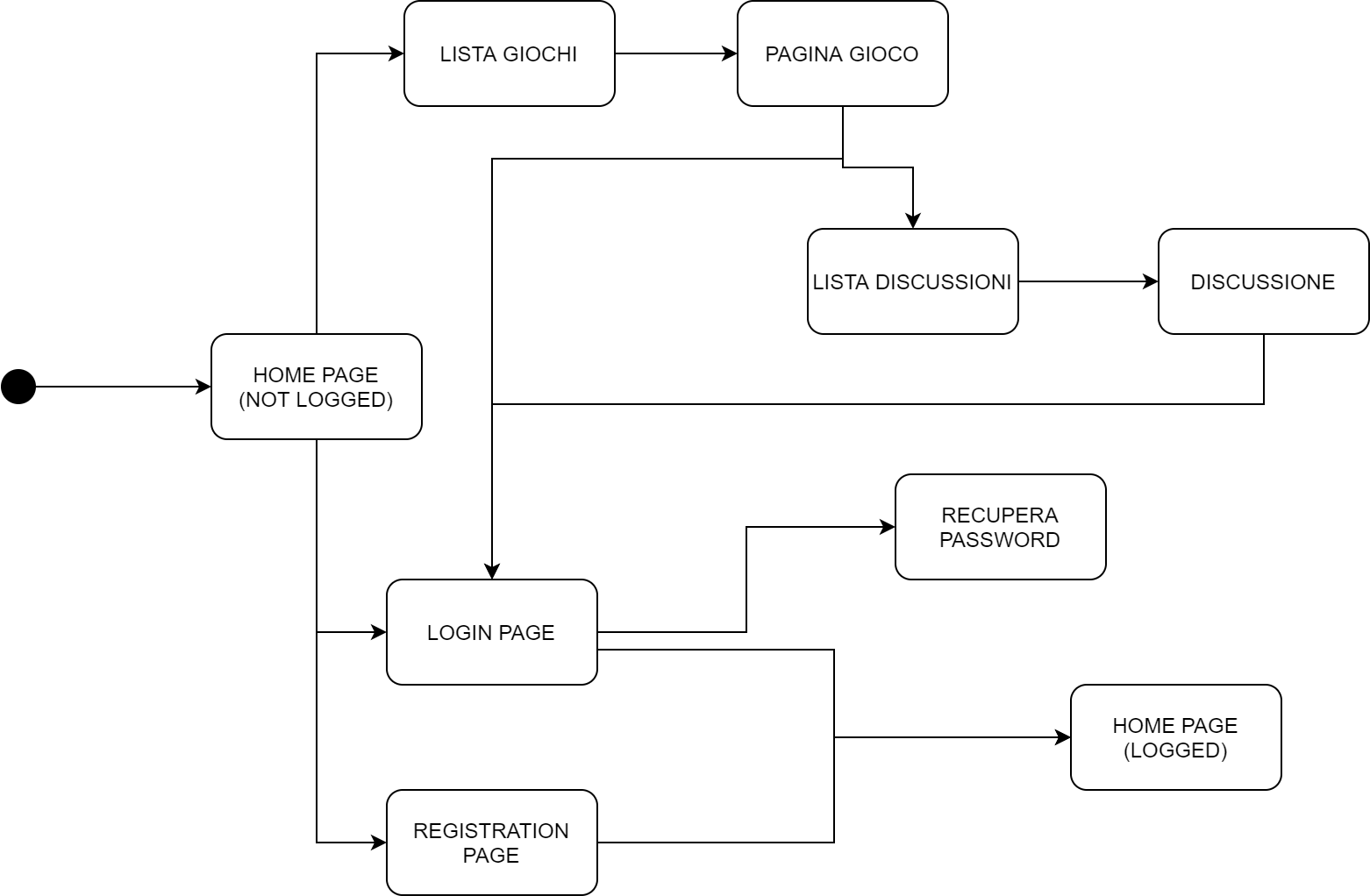
#### **-NP\_SVILUPPATORE**



#### **-NP\_UTNTE**

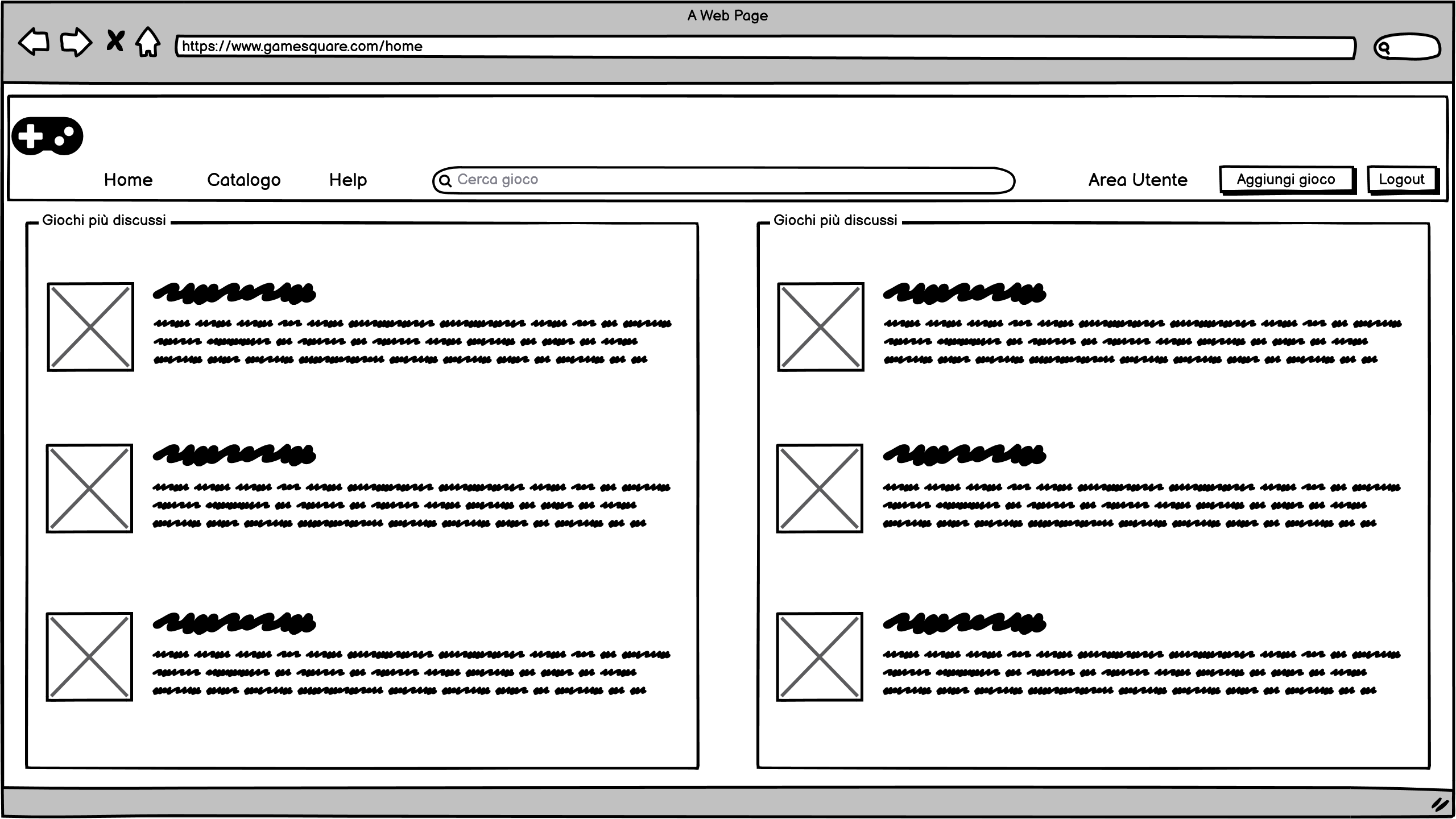


#### **-NP\_VISITATORE**

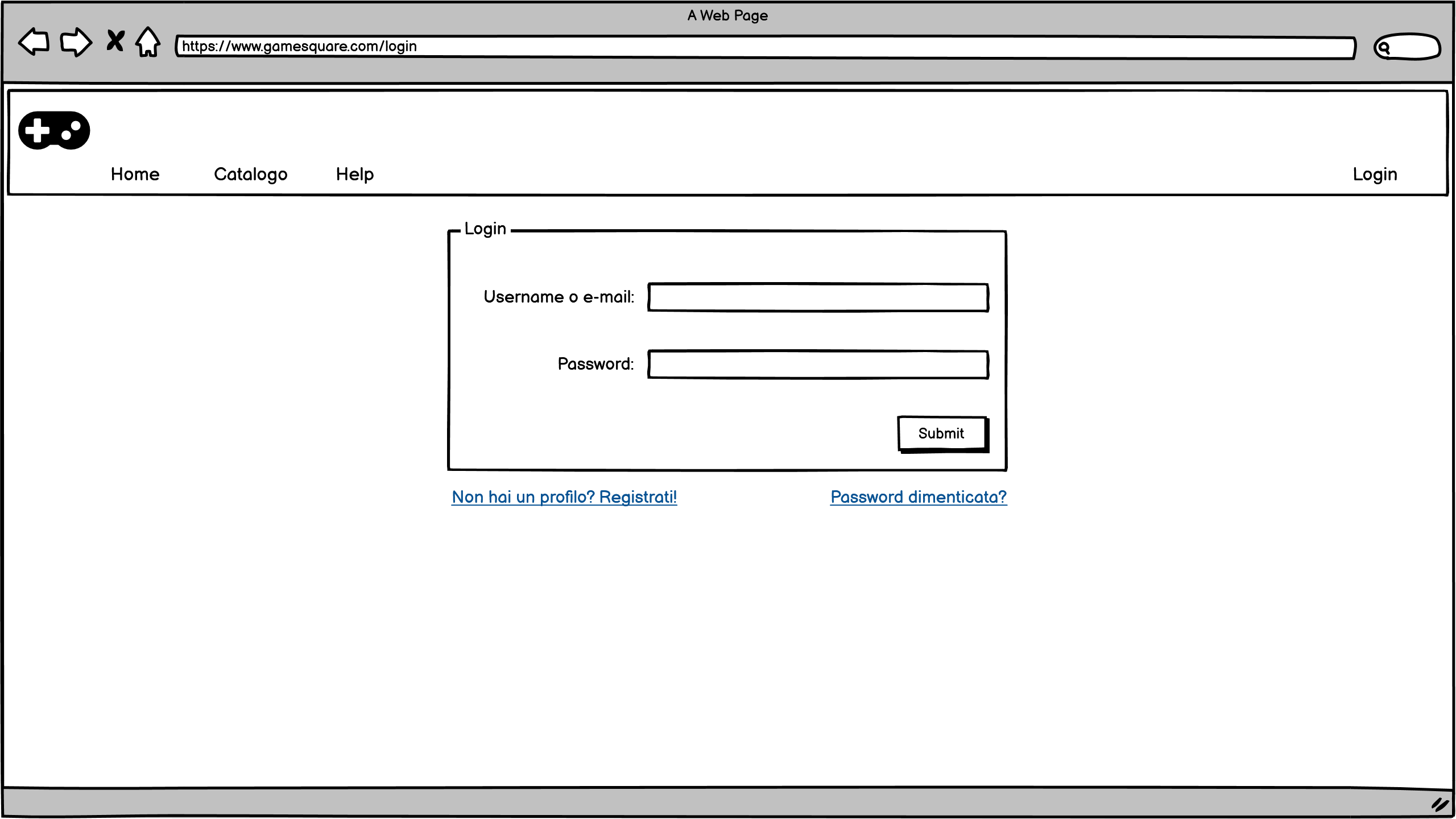


#### **Mock-Ups**

#### **-HOME\_PAGE\_LOGGED**



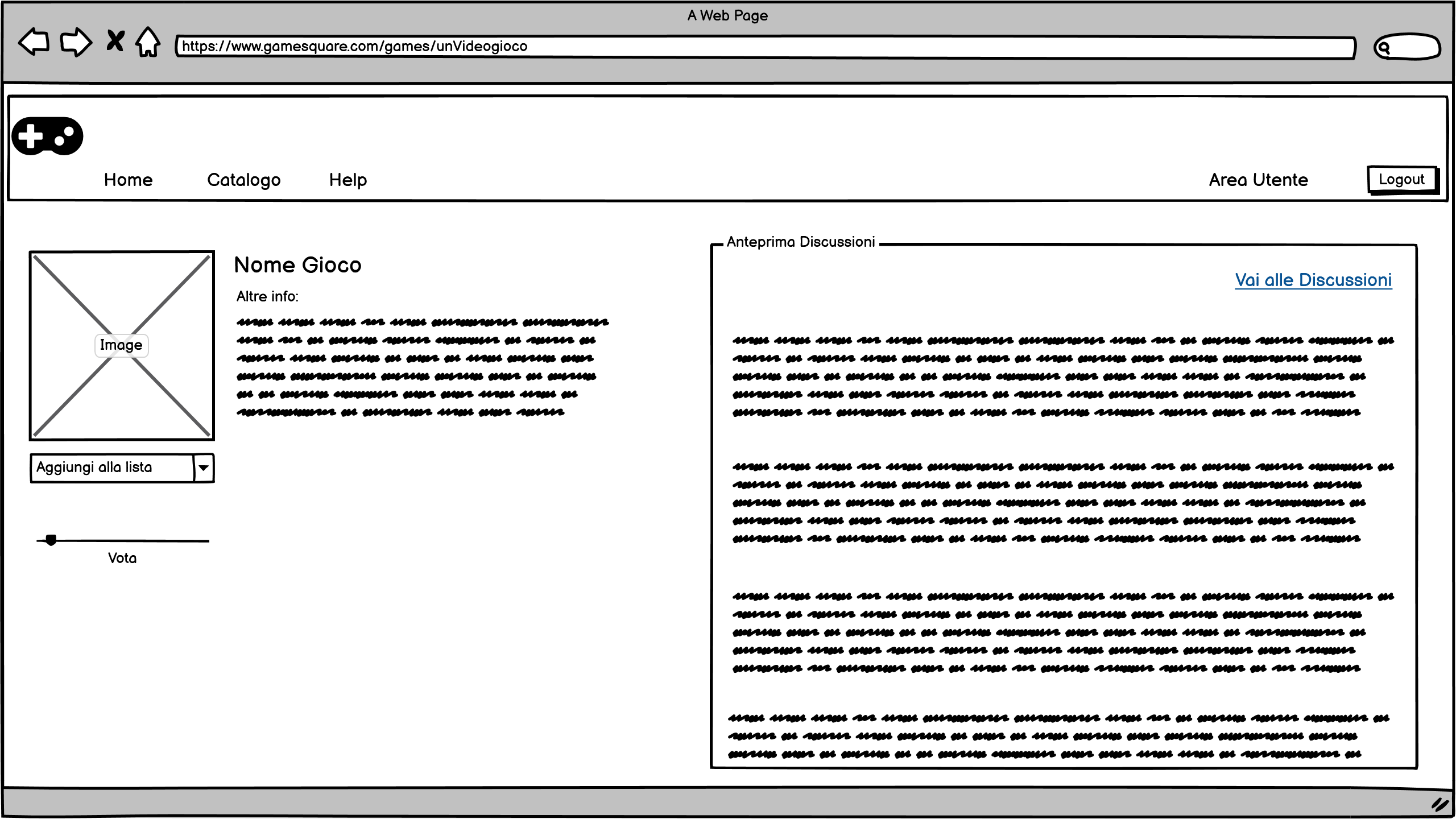
#### **-LOGIN**



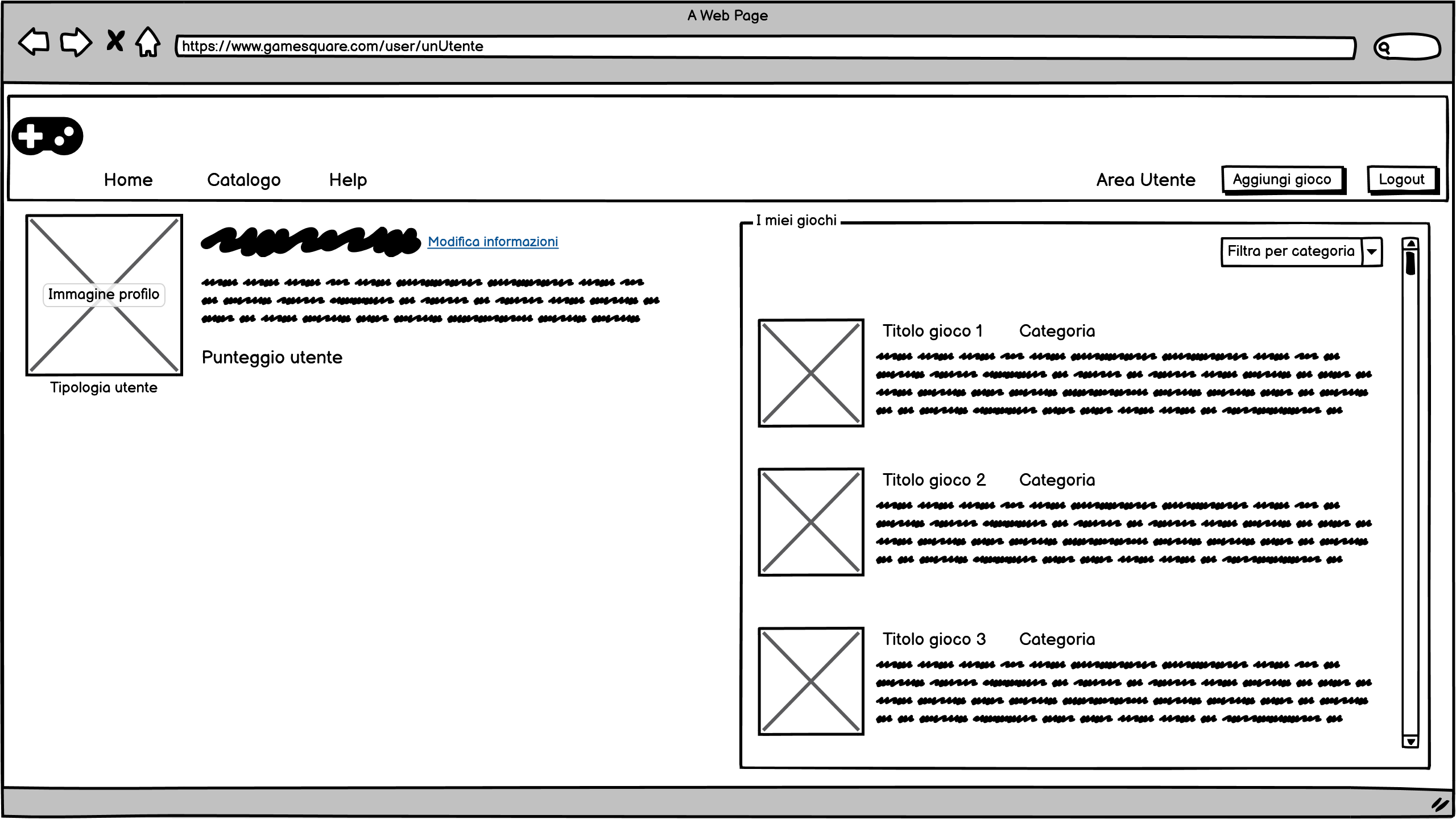
#### **-REGISTRAZIONE**



#### **-PAGINA\_GIOCO**



#### **-PAGINA\_UTENTE**



## 3.5 Glossario dei termini

|  |  |
| --- | --- |
| **Nome** | **Definizione** |
| Gioco | Un videogioco memorizzato nel sistema i cui dati possono essere visualizzati dall’applicazione. |
| Thread | Un post, un annuncio o una discussione avviata nell’applicazione per un determinato gioco. |
| Lista | Un insieme di giochi appartenenti ad un utente iscritto al sito e catalogati in base al loro livello di completamento. |
| Voto | Una valutazione in decimi da parte di un utente per un determinato gioco. |
| Utente Standard | Utente comune iscritto al sito, può gestire la sua lista, valutare i giochi e partecipare a discussioni. |
| Game Developer | Uno sviluppatore di videogiochi, che può aggiungere liberamente giochi e thread all’applicazione. |
| Moderatore | Un utente di GameSquare che si occupa della moderazione dell’aspetto social del sito, aggiungendo/rimuovendo thread e accettando richieste per la loro pubblicazione. |
| Gestore Catalogo | Un utente di GameSquare che si occupa esclusivamente della gestione dei giochi nel sito, aggiungendoli, modificandoli e accettando richieste per la loro aggiunta. |