**Problem Statement GamEvaluate**

**1. Problem domain**

Il sito GamEvaluate è rivolto alla sempre crescente comunità digitale di videogiocatori. Esso si prefissa di offrire l’accesso ad un ambiente di critica videoludica basato sulle medie delle recensioni e dei voti rilasciati dalla community stessa. Tale obiettivo nasce dalla necessità da parte dei videogiocatori di acquisire informazioni utili riguardo al prodotto che in futuro vorranno acquistare nei negozi, eliminando il rischio di effettuare un acquisto di cui si potrebbero pentire.

GamEvaluate, infatti, permetterà, dato un gioco registrato sul sito, di osservare il parere dell’intera comunità di videogiocatori che hanno in passato provato quel gioco in particolare; parere suddiviso in numerose caratteristiche (ad esempio la trama, il gameplay, la grafica, l’ottimizzazione, ecc.) ognuna delle quali rappresentata dalla media ottenuta dalle votazioni di ogni singolo utente in quell’ambito.

Risolvendo la problematica riguardante acquisti di giochi non rispecchianti le preferenze dell’utente si può ottenere una comunità di videogiocatori di gran lunga più soddisfatta dei prodotti del settore il che, di conseguenza, porterebbe ad un aumento della partecipazione di tali utenti nel miglioramento e nel prolungamento del ciclo di vita dei prodotti in questione: un videogiocatore contento del gioco appena acquistato continuerà con una buona probabilità a far parte della comunità virtuale di tale gioco, comprando più facilmente, in futuro, elementi aggiuntivi riguardanti quest’ultimo. Inoltre aumenta in tal modo la fetta di videogiocatori che acquisteranno le entrate seguenti della serie di cui il videogioco che hanno gradito faceva parte.

**2. Scenarios**

* *Scenario 1 (Amministratore):*

Giovanni è l’amministratore del sito GamEvaluate e può gestire i vari utenti e le recensioni del sito. Giovanni si informa attraverso internet riguardo gli ultimi giochi usciti, così da poterli aggiungere al sito e farli valutare dagli utenti. Trova un gioco da poter aggiungere, effettua il log-in compilando il form nella homepage del sito, inserendo come username: admin\_ge e come password: Admin01, e va nella sezione del sito dedicata per poter compilare un form che ha: titolo del gioco, genere, prezzo medio (basato sulla ricerca su vari siti di e-commerce del prezzo per lo stesso gioco) e piattaforma di gioco.

Dopo aver pubblicato il nuovo gioco da poter recensire, ha un po' di tempo libero per controllare alcune recensioni e nota che tra di esse ci sono alcuni commenti contenenti parole offensive. Giovanni decide quindi di eliminare il commento e assegna il compito a un moderatore tramite e-mail (anche se questo compito può essere eseguito da lui stesso). Paolo, che si occupa di gestire le varie recensioni fatte dagli utenti, controlla la propria mail e nota la segnalazione. Accede sul sito con username: paolo\_mod e password: Pmod01 (e trova il commento segnalato. Attraverso la spunta, visibile solo ai moderatori e agli amministratori, lo elimina. Poiché vuole essere sicuro che l’utente non abbia effettuato altri commenti negativi, Paolo va nella sezione “ricerca utente” e digita nella barra di ricerca l’username dell’utente, clicca sopra il risultato e arriva nel profilo dell’utente dove può vedere tutti i commenti da lui effettuati (questa sezione è visibile anche ad un utente). Dopo aver controllato tutti i suoi commenti nota che non ci sono altri commenti che potrebbero essere eliminati e quindi conclude il suo lavoro.

* *Scenario 2 (Visitatore/Utente):*

Mario vuole acquistare un nuovo gioco per la sua console, ma è indeciso tra alcuni titoli. Cercando in rete una recensione, si imbatte nel sito GamEvaluate; utilizzando la barra di ricerca, Mario riesce a trovare i titoli in questione e nota che, per ciascuno di questi, nella scheda principale vengono riportate data di uscita, prezzo medio, genere, piattaforma e media generale dei voti in decimi, seguiti da recensioni/commenti e valutazioni numeriche decimali su vari punti, come, ad esempio, trama, gameplay, grafica e difficoltà.

Mario, dopo accurati confronti, sceglie quale videogioco acquistare. Dopo averlo completato, decide di registrarsi sul sito, in modo tale da poter recensire anche lui il titolo e riportare una valutazione soggettiva riguardo i vari aspetti in aiuto dei futuri visitatori e/o utenti in cerca di informazioni riguardanti il gioco.

Mario, quindi, entra nella sezione di registrazione di GamEvaluate tramite apposita icona nella Homepage; inserisce i suoi dati quali username, e-mail e password. Nota subito come lo username, pur non dovendo rispettare alcun criterio (eccetto la lunghezza massima di 12 caratteri), deve risultare disponibile, quindi non in utilizzo da altri utenti precedentemente registrati, così come l’e-mail; la password, d’altro canto, deve essere di un minimo di 8 caratteri e deve contenere almeno una lettera maiuscola ed un numero. L’username inserito è: mario\_123, la mail utilizzata è: [mario\_123@gmail.com](mailto:mario_123@gmail.com), mentre la password scelta è: Mario123

Una volta registrato, Mario effettua il log-in su GamEvaluate con le credenziali appena inserite. Entrato nell’area utente cerca il titolo che intende recensire aiutandosi con la barra di ricerca e, una volta trovato, clicca su di esso per entrare nell’area relativa al gioco. All’interno di tale area nota la presenza di tutti i campi di valutazione relativi ai vari aspetti del gioco e la possibilità di assegnare una votazione in decimi ad ognuno di essi; nota, inoltre, la possibilità di lasciare un commento generale di valutazione per il gioco.

Mario decide quindi di assegnare le votazioni corrette, a suo parere, ed infine lasciare un commento che descriva la sua esperienza di gioco.

**3.Use Cases**

|  |  |
| --- | --- |
| Use case name | **AddGame** |
| Participating actor | Iniziato dall’*Amministratore* |
| Flow of events | 1. L’*Amministratore attiva la funzione “Aggiungi gioco”* dalla Home*.*    1. GamEvaluate risponde presentando la schermata di aggiunta gioco. 2. L’*Amministratore* compilail form di aggiunta del gioco inserendone il nome, il genere, la descrizione, il prezzo medio, la piattaforma e l’immagine di copertina. Infine l’*Amministratore* conferma i dati ed invia il form.    1. GamEvaluate controlla che non ci siano dati non idonei ed infine crea un nuovo *Gioco* all’interno del database.    2. GamEvaluate informa l’*Amministratore* dell’avvenuto/non avvenuto inserimento con un *pop-up.* |
| Entry condition | * L’*Amministratore* ha effettuato l’accesso a GamEvaluate. |
| Exit condition | * Il *Gioco* viene inserito con successo. * Ci sono stati problemi con l’aggiunta del *Gioco* da parte di GamEvaluate. |
| Quality requirements | * Nel caso di fallimento l’*Amministratore* viene informato tramite *pop-up.* |

|  |  |
| --- | --- |
| Use case name | **DeleteGame** |
| Participating actor | Iniziato dall’*Amministratore* |
| Flow of events | 1. L’*Amministratore attiva la funzione “Elimina gioco”* dalla schermata informativa del gioco da eliminare*.*    1. GamEvaluate chiede all’*Amministratore* di confermare l’operazione. 2. L’*Amministratore* conferma l’operazione.    1. GamEvaluate elimina il *Gioco* dal database.    2. GamEvaluate informa l’*Amministratore* dell’avvenuta/non avvenuta eliminazione con un *pop-up.* |
| Entry condition | * L’*Amministratore* ha effettuato l’accesso a GamEvaluate. |
| Exit condition | * Il *Gioco* viene eliminato con successo. * Ci sono stati problemi con l’eliminazione del *Gioco* da parte di GamEvaluate. |
| Quality requirements | * Nel caso di fallimento l’*Amministratore* viene informato tramite *pop-up.* |

|  |  |
| --- | --- |
| Use case name | **ModifyGame** |
| Participating actor | Iniziato dall’*Amministratore/Moderatore* |
| Flow of events | 1. *L’Amministratore/Moderatore attiva la funzione “Modifica gioco”* dalla schermata informativa del gioco da modificare*.*    1. GamEvaluate risponde trasformando i campi informazione nella schermata del gioco in campi modificabili. 2. L’ *Amministratore/Moderatore* effettua le opportune modifiche e, in seguito, le conferma.    1. GamEvaluate controlla che non ci siano dati non idonei ed infine aggiorna i dati del gioco modificati.    2. GamEvaluate informa l’*Amministratore/Moderatore* dell’avvenuta/non avvenuta modifica con un *pop-up*. |
| Entry condition | * L’ *Amministratore/Moderatore* ha effettuato l’accesso a GamEvaluate. * Il *Gioco* è presente all’interno del database di GamEvaluate. |
| Exit condition | * Il *Gioco* viene modificato con successo. * Ci sono stati problemi con la modifica del *Gioco* da parte di GamEvaluate. |
| Quality requirements | * Nel caso di fallimento l’*Amministratore/Moderatore* viene informato tramite *pop-up.* |

|  |  |
| --- | --- |
| Use case name | **Upgrade/DowngradeUser** |
| Participating actor | Iniziato dall’*Amministratore* |
| Flow of events | 1. L’*Amministratore* ricerca l’utente da nominare/declassare con l’apposita barra di ricerca nella Home*.*    1. GamEvaluate risponde mostrando la lista degli utenti corrispondenti al testo inserito nella barra. 2. L’*Amministratore* nomina/declassa l’utente scelto a *Moderatore* tramite l’apposita icona di fianco all’utente.    1. GamEvaluate chiede conferma dell’operazione all’*Amministratore*. 3. L’*Amministratore* conferma la scelta.    1. GamEvaluate cambia i privilegi dell’utente/moderatore a quelli di *Moderatore/Utente.*    2. GamEvaluate informa l’*Amministratore* dell’avvenuto/non avvenuto *Upgrade/Downgrade* tramite *pop-up*. |
| Entry condition | * L’*Amministratore* ha effettuato l’accesso a GamEvaluate. * L’*Utente* è presente all’interno del database di GamEvaluate. |
| Exit condition | * *L’Utente* viene nominato/declassato con successo. * Ci sono stati problemi con la nominazione/declassamento dell’*Utente* da parte di GamEvaluate. |
| Quality requirements | * Nel caso di fallimento l’*Amministratore* viene informato tramite *pop-up.* * L’*Amministratore* è in grado di effettuare un *Upgrade/Downgrade* di un Utente mediante un singolo bottone. Se l’utente selezionato gode di privilegi di *Utente* avverrà un *Upgrade*, altrimenti avverrà un *Downgrade*. |
| Use case name | **Ban/UnbanUser** |
| Participating actor | Iniziato dall’ *Amministratore/Moderatore* |
| Flow of events | 1. L’ *Amministratore/Moderatore* ricerca l’utente da bandire/ripristinare con l’apposita barra di ricerca nella Home*.*    1. GamEvaluate risponde mostrando la lista degli utenti corrispondenti al testo inserito nella barra. 2. L’ *Amministratore/Moderatore* bandisce/ripristina l’utente scelto tramite l’apposita icona di fianco all’utente.    1. GamEvaluate chiede conferma dell’operazione all’ *Amministratore/Moderatore*. 3. L’ *Amministratore/Moderatore* conferma la scelta.    1. GamEvaluate rende l’utente *Bandito* o, nel caso lo fosse già, ne ripristina le funzioni.    2. GamEvaluate informa l’*Amministratore/Moderatore* dell’avvenuto/non avvenuto *Ban/Unban* tramite *pop-up*. |
| Entry condition | * L’ *Amministratore/Moderatore* ha effettuato l’accesso a GamEvaluate. * L’*Utente* è presente all’interno del database di GamEvaluate. |
| Exit condition | * *L’Utente* viene bandito/ripristinato con successo. * Ci sono stati problemi con il *Ban/Unban* dell’*Utente* da parte di GamEvaluate. |
| Quality requirements | * Nel caso di fallimento l’*Amministratore/Moderatore* viene informato tramite *pop-up.* * L’ *Amministratore/Moderatore* è in grado di effettuare un *Ban/Unban* di un Utente mediante un singolo bottone. Se l’utente selezionato è banditoavverrà un *Ban*, altrimenti avverrà un *Unban*. |

**\*** Il *Moderatore* può bandire solo *Utenti* base di GamEvaluate, e non altri *Moderatori*.

|  |  |
| --- | --- |
| Use case name | **VoteGame** |
| Participating actor | Iniziato dall’ *Utente* |
| Flow of events | 1. L’*Utente* seleziona l’opzione di votazione del gioco.    1. GamEvaluate presenta all’Utente il form con la possibilità di assegnare un voto ai singoli aspetti del gioco. 2. L’*Utente* assegna una votazione a tutti gli aspetti e conferma l’operazione.    1. GamEvaluate controlla che tutti gli aspetti siano stati votati.       1. Se degli aspetti non sono stati votati, informa l’*Utente* con un *pop-up*.       2. Se tutti gli aspetti sono stati votati, aggiunge la votazione al database della piattaforma e informa l’*Utente* dell’avvenuta votazione con un *pop-up*. |
| Entry condition | * L’Utenteha effettuato l’accesso a GamEvaluate. * L’*Utente* è all’interno dell’area informativa del gioco da votare. |
| Exit condition | * La votazione va a buon fine e l’*Utente* ne viene informato con un *pop-up*. * La votazione non va a buon fine e l’*Utente* ne viene informato con un *pop-up*. |
| Quality requirements | * Nel caso di fallimento l’Utente viene informato tramite *pop-up.* |

|  |  |
| --- | --- |
| Use case name | **ReviewGame** |
| Participating actor | Iniziato dall’ *Utente* |
| Flow of events | 1. L’*Utente* seleziona l’opzione di recensione del gioco.    1. GamEvaluate presenta all’Utente il form con la possibilità di recensire il gioco. 2. L’*Utente* scrive una recensione riguardante il gioco e conferma l’operazione.    1. GamEvaluate aggiunge la recensione al database della piattaforma.    2. GamEvaluate informa l’*Utente* dell’avvenuto inserimento della recensione con un *pop-up*. |
| Entry condition | * L’Utenteha effettuato l’accesso a GamEvaluate. * L’*Utente* è all’interno dell’area informativa del gioco da votare. |
| Exit condition | * La recensione va a buon fine e l’*Utente* ne viene informato con un *pop-up*. * La recensione non va a buon fine e l’*Utente* ne viene informato con un *pop-up*. |
| Quality requirements | * Nel caso di fallimento l’*Utente* viene informato tramite *pop-up.* |

|  |  |
| --- | --- |
| Use case name | **Registrazione** |
| Participating actor | Iniziato dal Visitatore |
| Flow of events | 1. Il *Visitatore* accede all’area registrazione 2. GamEvaluate presenta all’utente il form con la possibilità di inserire username, e-mail, password 3. Il *Visitatore* dopo aver completato tutti i campi del form conferma i dati 4. GamEvaluate controlla la presenza dei dati (username e e-mai) all’interno del database e se già esistono avverte l’utente 5. GamEvaluate informa l’*Utente* dell’avvenuta registrazione dopo essere stato reindirizzato alla pagina di login |
| Entry condition | * Il *Visitatore* non è già registrato a GamEvaluate. |
| Exit condition | * La registrazione va a buon fine e l’*Utente* ne viene informato. * La recensione non va a buon fine e il *Visitatore* ne viene informato con un *pop-up*. |
| Quality requirements | * Nel caso di fallimento il *Visitatore* viene informato tramite *pop-up.* |

**4. Functional requirements**

* *L’amministratore* deve poter aggiungere/eliminare giochi e modificare le informazioni di questi ultimi.
* *L’amministratore* deve poter vietare l’accesso a determinati utenti con un apposito meccanismo di ban. Tali utenti rimarranno registrati ma non potranno effettuare l’accesso fino a futura decisione dell’*amministratore.*
* L’*amministratore* deve poter nominare un utente a moderatore e, viceversa, declassare un moderatore a utente.
* Il *moderatore* deve poter aggiungere/eliminare giochi e modificare le informazioni di questi ultimidc.
* L’*utente* deve poter ricercare un gioco ed accedere all’area informativa di quest’ultimo. \*
* *L’utente* deve poter inserire le proprie votazioni personali e pubblicare una recensione all’interno dell’area informativa di un gioco. \*
* *L’utente* deve poter salvare un gioco all’interno della propria Wish List. \*

\* I requisiti funzionali per l’*utente* riguardano anche il *moderatore*.

**5. Nonfunctional requirements**

* *Usability.* L’utente deve poter utilizzare GamEvaluate con facilità senza conoscenze implicite se non una conoscenza di base del funzionamento di un sito web.
* *Supportability.* GamEvaluate deve poter essere utilizzato sui principali browser (Chrome, Mozilla, Explorer, Safari).
* *Portability.* L’utentedeve poter accedere a GamEvaluate sia da un PC sia da un dispositivo mobile.
* *Implementation.* La piattaforma è implementata tramite linguaggio Java e SQL per quanto riguarda il back-end, e linquaggio HTML, Javascript e CSS per quanto riguarda il front-end.

6. Target environment

7. Deliverable & deadlines

* **Mercoledì 25 settembre**: consegna schede informative partecipanti e proposta di progetto
* **Mercoledì 2 ottobre**: Invio informazioni al docente
  + Creazione repository GitHub di progetto
  + Invio file excel con informazioni su progetto e partecipanti e file della proposta di progetto
  + Invito collaboratori, docente e tutor su repository GitHub
* **Mercoledì 3 ottobre:** Kick-off meeting (in lab) 30 Scadenze intermedie (documenti)
* **11 ottobre**: Problem Statement
* **25 ottobre**: Requisiti e casi d’uso
* **8 novembre**: Requirements Analysis Document
* **29 novembre**: System Design Document
* **13 dicembre**: piano di test e specifica interfacce dei moduli del sistema