

PROPOSTA DI PROGETTO
OBJECT ORIENTED SOFTWARE DESIGN
CORSO DI LAUREA IN INFORMATICA
UNIVERSITA' DEGLI STUDI DELL'AQUILA
A.A. 2016/2017

SPECIFICA

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

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

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

- *Registrazione da parte dell'utente al portale*

L'utente avrà la possibilità di effettuare la registrazione al portale di gaming in modo da effettuare in un secondo momento la login e dare il via quindi alla propria esperienza di gioco.

- *Visualizzazione del profilo e delle pagine dei relativi giochi.*

Sarà possibile visualizzare il proprio profilo con le relative informazioni anagrafiche e con i vari traguardi conquistati durante le sessioni di gioco. Sarà quindi possibile visualizzare la timeline con i livelli conquistati o il bilancio dei propri punti esperienza, che incrementandosi porteranno all'acquisizione di un nuovo livello.

- *Possibilità di votare un gioco o esprimere un giudizio sullo stesso tramite recensione.*

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

- *Effettuare una sessione di gioco e collezionare punti esperienza ad ogni sessione.*

Ad ogni sessione di gioco l'utente vedrà incrementarsi la propria "balance" relativa ai propri punti esperienza. Tale attributo servirà allo stesso per conseguire nuovi livelli e arricchire il proprio profilo di nuovi traguardi.

- *Collezionare trofei relativamente al livello conquistato.*

Ogni livello (eccetto quello iniziale) avrà associato il relativo “award”. Tale trofeo, associato 1:1 con il relativo livello, sarà assegnato insieme a quest’ultimo al relativo profilo utente.

Scenario:

Ipotizziamo che ogni "bet" (spinnata, giocata) fornisca 30XP (punti esperienza) e i livelli sono configurati come segue:

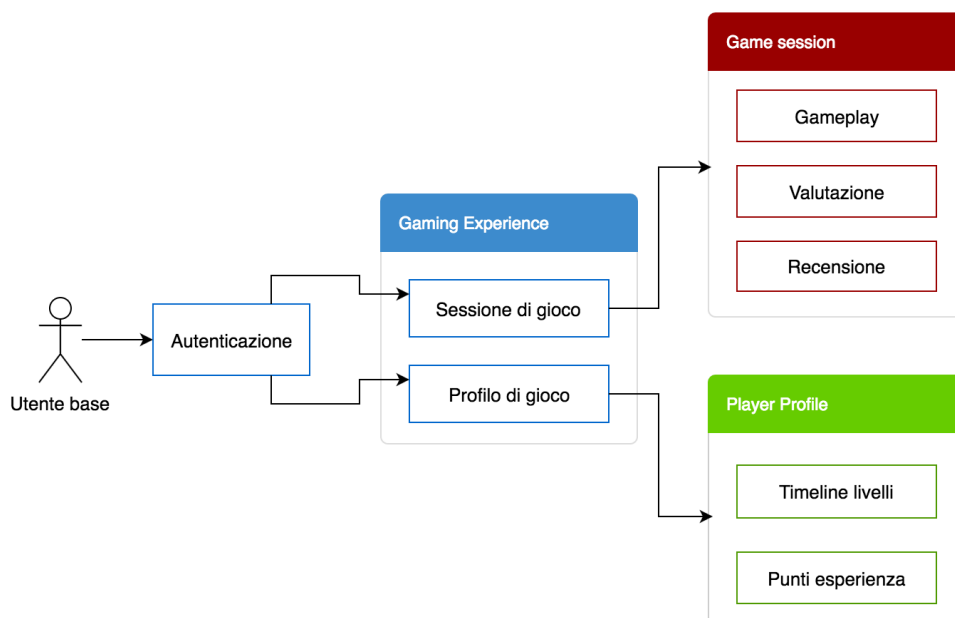
- Livello 0 - 0
- Livello 1 - 100 XP
- Livello 2 - 200 XP
- Livello 3 - 300 XP
- Livello 4 - 400 XP
- Livello 5 - 500 XP

ne segue che serviranno "n" bet per ottenere il livello. Il gioco può essere simulato, ad esempio attraverso una Window in cui ogni click su un determinato bottone rappresenta una "bet". La windows con il bottone che simula la “bet” chiamerà una funzione che incrementerà un'attributo dell'oggetto player (balance) di X punti esperienza e controllerà/aggiognerà controlla il livello.

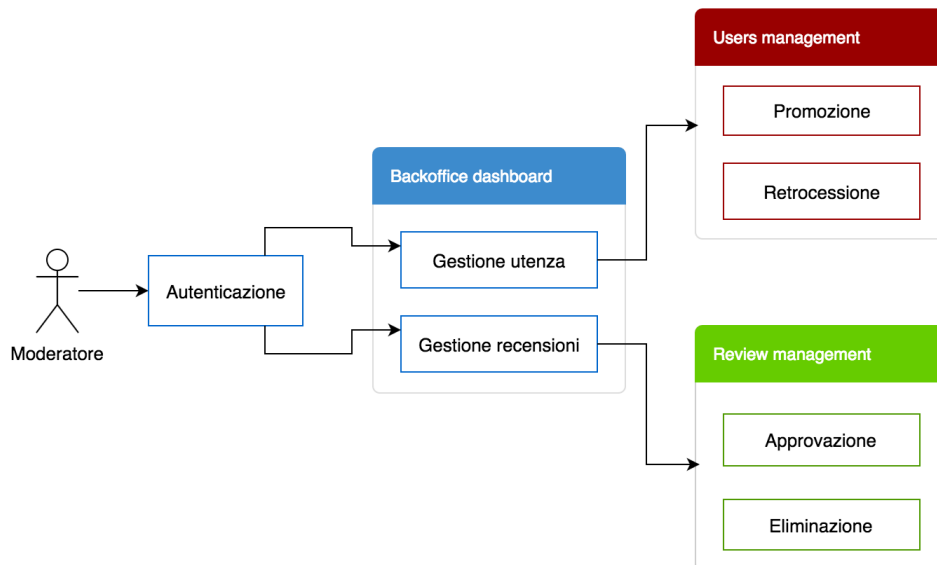
Attori del sistema:

- Amministratore: gestione generale del sistema
- Moderatore: approvazione recensioni/gestione utenza
- Utente base: effettuare sessione di gioco/votare e recensire gioco

Processo di gaming experience da parte di un *Utente base*:



Processo di gaming experience da parte di un *Moderatore*:



Link utili:

- www.lottomatica.it/

REQUISITI

Parte 1

1. Requirements

Se necessario, riorganizzare la specifica data sotto forma di requisiti.

Documenti richiesti: (1.1) documento dei requisiti.

2. System design

Definire gli obiettivi di design, decomporre il sistema, selezionare le strategie per la costruzione del sistema, definire l'architettura software del sistema, scegliere design patterns.

Documenti richiesti: (2.1) modello dell'architettura software (o più modelli con diverso grado di dettaglio), (2.2) descrizione dell'architettura, (2.3) descrizione delle scelte e strategie adottate (compresi design patterns)

3. Software/Object design

Definire gli oggetti del dominio, definire in modo dettagliato componenti, classi, interfacce e membri.

Documenti richiesti: (3.1) modelli rappresentanti l'object design con classi, interfacce e membri, (3.2) descrizione dei dettagli di design scelti.

4. Deliverable (deadline 21/4/2017)

Il progetto deve essere mantenuto in un repository GIT, che sarà strutturato come segue.

<nome_progetto> : root del progetto

- *src* : questa directory conterrà il codice sorgente del progetto
- *doc* : questa directory conterrà i documenti di design del progetto (in format pdf)
- *javadoc* : questa directory conterrà il codice javadoc del progetto

La URL del repository sarà consegnata al docente, la parte di competenza di questo deliverable sarà contenuta nella directory doc.

Note: Gli strumenti a disposizione per svolgere la parte 1 del progetto riguardano il programma del corso relativo alle lezioni 1-13 (OO design, Classi e membri, Package, modificatori di accesso, final/static, Ereditarietà, Polimorfismo, Interfacce)

Parte 2

1. Refining the design

Il design del sistema e il design software devono essere raffinati ed integrati alla base delle nuove conoscenze acquisite nella seconda parte del corso.

Documenti da raffinare: (2.1), (2.2), (2.3), (3.1)

2. Implementation

Implementazione del sistema software alla base delle scelte di design effettuate. Il lavoro deve essere svolto all'interno di un ambiente di sviluppo (IDE), Eclipse. Il lavoro deve essere sviluppato in modo collaborativo attraverso un repository GIT.

L'implementazione del sistema deve necessariamente far uso dei seguenti strumenti:

- Classi, package e modificatori di accesso per una corretta strutturazione del sistema
- Interfacce
- Ereditarietà
- Polimorfismo e overriding/overloading di metodi con almeno un esempio di dinamic binding
- Uso di final, static
- Eccezioni
- Collection/Map
- JDBC

3. Documentation

La documentazione da allegare al progetto sarà costituita dai documenti ai punti 2.* e 3.* raffinati e inoltre da Javadoc ed eventuali annotazioni (ciò implica opportuna formattazione e commenti).

4. Deliverable

A ultimazione del progetto la comunicazione della URL del repository da aggiornare sarà consegnata al docente, le parti di competenza di questo deliverable saranno contenute nelle directory src, doc, javadoc.

Note: Gli strumenti a disposizione per svolgere la parte 2 del progetto riguardano il programma del corso relativo alle lezioni 14-20 (Eccezioni, Collections/Maps, Generics, I/O, JDBC, Threads, Design patterns).

Tutti i progetti non conformi ai sopra indicati requisiti non saranno accettati.