

**Corso di laurea in Informatica**  
**Dipartimento di Informatica - Università di Pisa**  
**Progetto – Terzo Appello 2021-22**

***BD: Consegna entro il 13-7-2022 ore 18:00***

***IS: Consegna entro il 14-7-2022 ore 18:00***

**Introduzione comune per i corsi di IS e BD**

## **The Series Hunter**

Si deve progettare una app, The Series Hunter, per permettere a un gruppo di conoscenti (e conoscenti di conoscenti...) di consigliarsi serie offerte dalle più note piattaforme streaming, visto che pochi sono soddisfatti del sistema di raccomandazioni offerto dalle piattaforme stesse.

Un utente può creare un nuovo gruppo. Ogni partecipante a un gruppo può aggiungere un amico (deve essere una persona che conosce nella vita reale) al gruppo. Non è possibile chiedere tramite app di essere uniti a un gruppo esistente, si può solo essere invitati. Un gruppo ha un nome ed eventualmente una descrizione: ad esempio si può creare un gruppo dedicato alle serie europee o alle commedie italiane o ai cartoni animati.

Un utente può lasciare un gruppo se lo vuole. Può anche essere espulso, in caso di spoiler, (che è assolutamente vietato) oppure in caso di inattività prolungata (se per 4 mesi non aggiunge alcuna recensione). Sospetti spoiler possono essere segnalati da qualsiasi partecipante agli amministratori, che decidono se siano effettivamente tali.

Per mantenere lo spirito della app, non sono permessi gruppi troppo grandi: il numero massimo è 200 persone. Oltre questo limite nuove persone possono essere invitate solo dopo abbandono o espulsione.

Ogni gruppo è inizialmente amministrato dal creatore del gruppo. Al raggiungimento di 40 partecipanti, i partecipanti al gruppo eleggono un secondo amministratore. Un terzo viene aggiunto al raggiungimento dei 100 partecipanti, sempre tramite elezione. Un amministratore può abbandonare il suo ruolo, nel qual caso si va a nuove elezioni. Le elezioni:



- sono indette dagli amministratori rimasti o dal decano del gruppo, il partecipante con maggior anzianità (di appartenenza, non anagrafica)
- durano 3 giorni, poi chi ha più voti vince
- chiunque si può aggiungere alla lista dei candidati
- si possono votare più persone (pensate a come funzionano i sondaggi in facebook)

Un partecipante al gruppo può inserire una nuova serie (o film) o aggiungere una revisione a una esistente. Per evitare doppioni, quando si inserisce una nuova serie, il sistema cerca se questa è già esistente. In tal caso chiede all'utente: "intendevi questa?". In caso di risposta affermativa apre in scrittura la scheda della serie, altrimenti viene creata una nuova scheda.

Una scheda aperta in scrittura permette di:

1. aggiungere un secondo/terzo titolo (in una lingua diversa, comprendente un sottotitolo...);
2. indicare un voto tra 0 e 10;
3. aggiungere dei tag (es: #commedia, #cartoon, #francese, #miniserie...) a quelli esistenti;
4. aggiungere un commento testuale;
5. aggiungere la/una piattaforma su cui è offerta;

Un partecipante al gruppo può fare diversi tipo di ricerca, tre sono obbligatoriamente offerti dalla App:

1. ricerca per titolo: si apre in lettura la scheda relativa al titolo indicato;
2. ricerca per voto: vengono restituite le 5 serie non ancora viste (non recensite dal richiedente) con i voti più alti (media aritmetica);
3. ricerca per gusti simili: vengono restituite le 5 serie non ancora viste (non recensite dal richiedente) con i voti più alti (media ponderata usando come peso una misura dell'affinità di giudizi).

## Progetto di Ingegneria del Software

1. Descrivere con un diagramma UML tutti i casi d'uso del Sistema. Per uno di essi, non banale, dare la narrativa.
2. Dare un diagramma di sequenza che descriva il caso d'uso di cui si è data la narrativa.
3. Dare un diagramma delle classi che descriva gli elementi significativi del dominio della App.
4. Dare un diagramma di macchina a stati che modelli gli stati in cui può trovarsi un gruppo.
5. Dare un diagramma di attività che modelli il processo per valutare se una recensione è spoiler.
6. Definire una possibile architettura (dare sia la vista C&C che di dislocazione) per realizzare il sistema in oggetto.
7. Aggiungere un nuovo requisito funzionale e un nuovo requisito non funzionale.
8. Il metodo `int affinitàDiGiudizi (Partecipante a, Partecipante b)` riceve in input i riferimenti a due partecipanti e calcola il grado di affinità di giudizi tra essi. L'implementazione del metodo conta le schede per cui entrambi hanno espresso un voto, se queste sono in numero minore o uguale a 3, restituisce -1, altrimenti restituisce la media delle differenze (in valore assoluto) dei voti dati a ogni serie. Il metodo `affinitàDiGiudizi` invoca i seguenti metodi:
  - `String [] recensioniDi (Partecipante a)` restituisce un array con i titoli delle serie per cui a ha espresso un voto;
  - `int[] votoDi (String s, Partecipante a)` restituisce il voto espresso da a alla serie s

Scrivere un'implementazione del metodo `affinitàDiGiudizi` e definire due stub per i metodi invocati e alcuni casi di test che permettano di avere copertura al 100% delle decisioni. Calcolare la copertura dei cammini ottenuta.

## Regole di consegna del progetto di Ingegneria del Software

1. Deve essere consegnato un archivio contenente:
  - a. il file `.vpp` del progetto (un file unico e non tanti file spezzettati)
  - b. un file pdf che posso stampare per correggere più velocemente i progetti. Deve contenere: nome cognome e matricola dei membri del gruppo; le parti testuali del progetto; i diagrammi prodotti, rispettando l'ordine degli esercizi dati.
2. Sia i file che l'archivio devono chiamarsi col cognome del referente del gruppo (chi fa la sottomissione del progetto).
3. Il progetto deve essere inviato per email entro la data fissata: `laura.semini@unipi.it`
4. Per correttezza nei confronti dei compagni di gruppo, chi sottomette mette in CC tutti i membri del gruppo.
5. La mail deve avere subject: `Progetto SeriesHunter`
6. Tutte le mail (mittente e destinatari) devono essere istituzionali (`@unipi.it / @studenti.unipi.it`).