

Progetto “Tecnologie Web” 2022

Racca Lorenzo (141202)

Descrizione progetto	2
Requisiti	2
Progettazione	3
Casi d'uso	3
Schema DB	4
Realizzazione	5
Tecnologie scelte	5
Organizzazione logica	5
Applicazioni	5
Templates	5
Casi d'uso d'interesse	5
Gestione like	6
Gestione rimozione da watchlist	6
Film suggeriti	7
Recommendation System	7
Test	8
Test su codice applicativo	8
Test 1	8
Test 2	8
Test 3	8
Test 4	8
Test su una vista	8
Test 1	8
Test 2	8
Test 3	9
Test 4	9
Test 5	9
Test 6	9
Test 7	9
Test 8	9
Screenshot interessanti	10
Pagina dei film	10
Pagina personale	11

Descrizione progetto

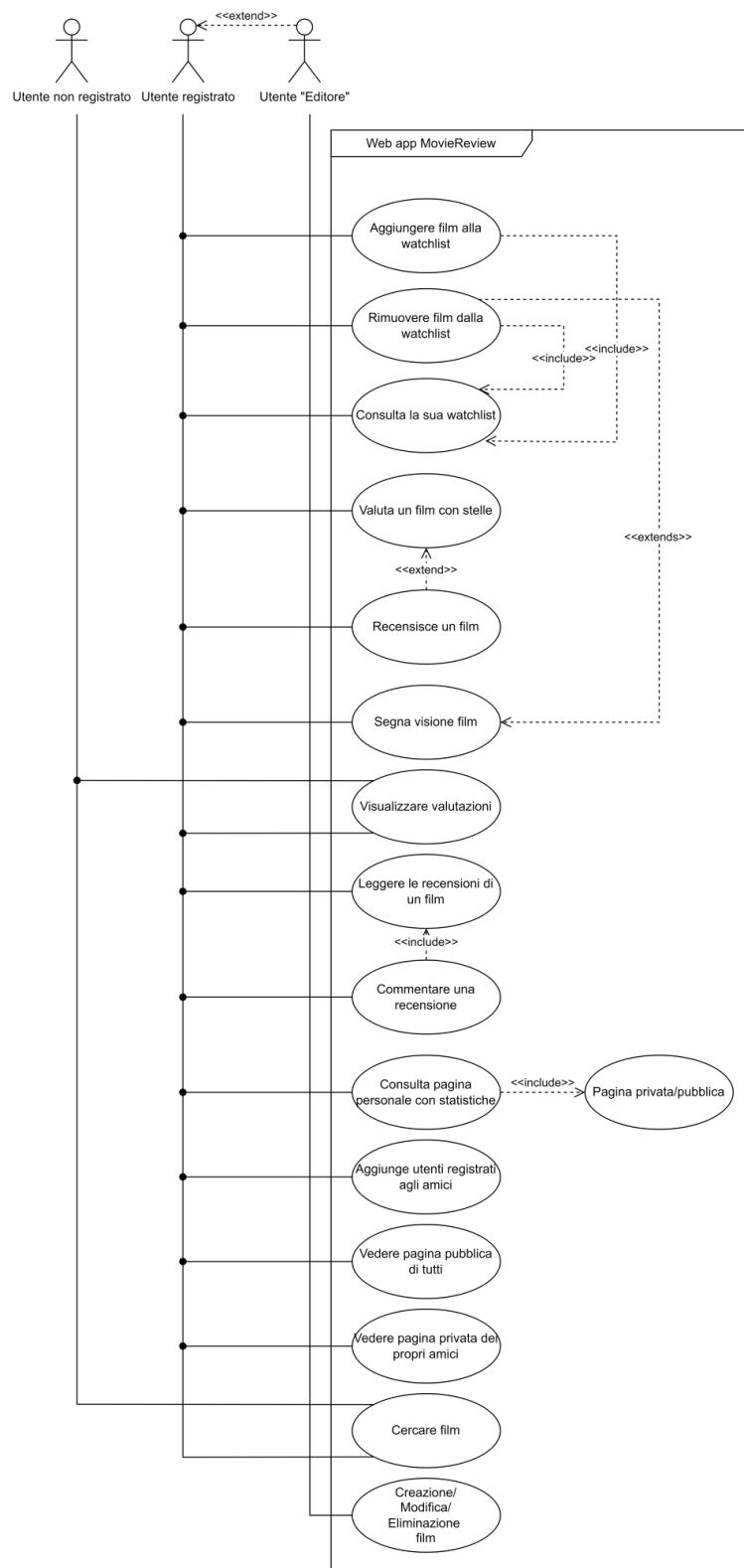
Creare un sito di recensioni di film(stile imdb o rotten tomatoes), dove l'utente possa tenere traccia delle proprie visioni, annotare i film che vuole vedere e recensirli.

Requisiti

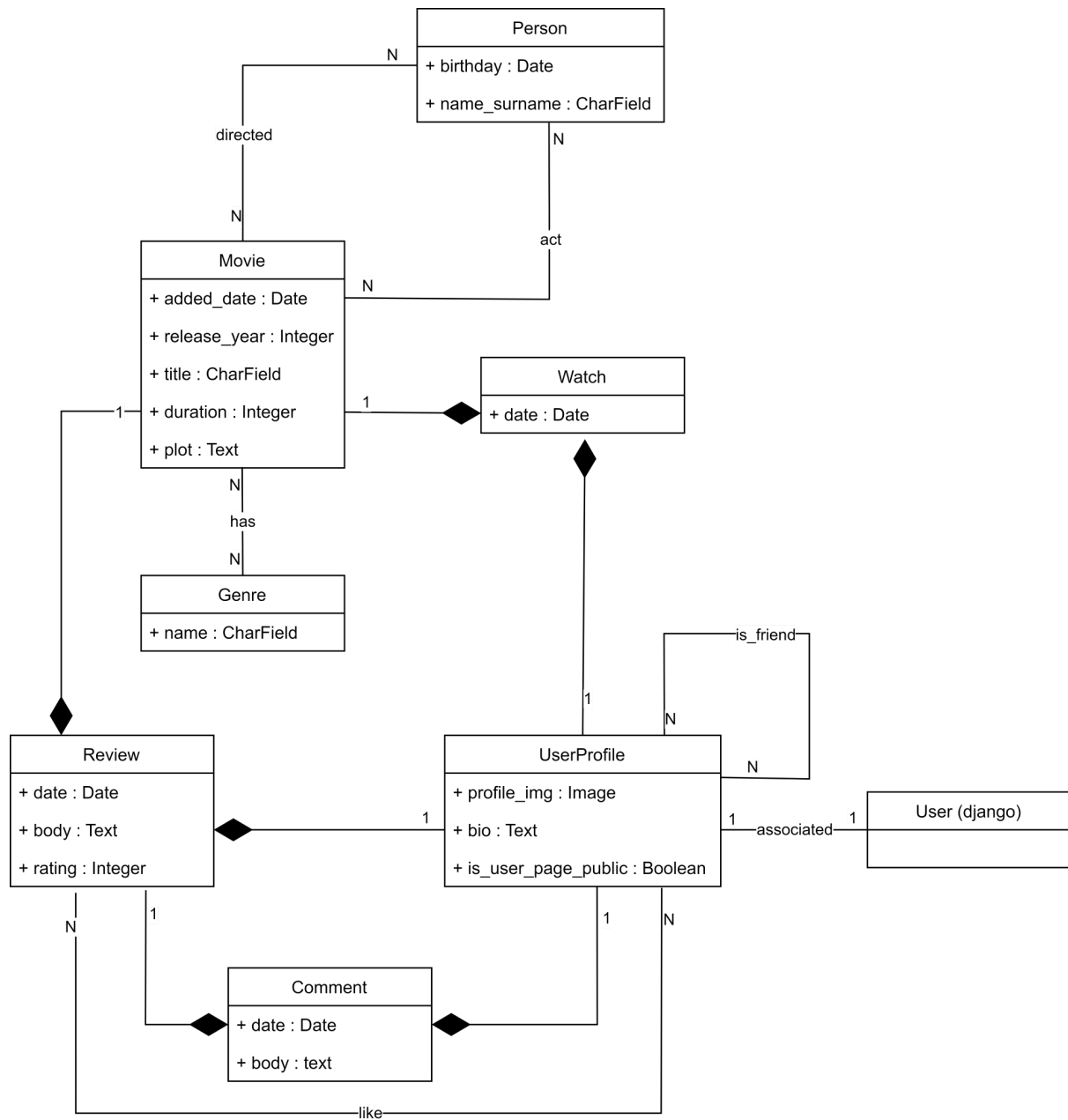
- L'utente iscritto ha una sua watchlist
- L'utente iscritto può valutare i film con una valutazione da 1 a 5 stelle più una breve recensione e segnare la data di visione (sono possibili i rewatch)
- Le recensioni e valutazioni sono visibili da tutti gli utenti iscritti.
- E' possibile commentare o votare(mettere mi piace) le recensioni e creare discussioni
- L'utente iscritto ha una sua pagina dove può avere delle statistiche date dai film che ha visto (ex. Genere preferito, ore passate a guardare film, film preferito etc)
- La pagina dell'utente può essere pubblica o privata
- Gli utenti possono aggiungersi a vicenda tra la lista dei propri amici
- Solo i propri amici possono vedersi le proprie pagine private a vicenda
- In homepage vengono mostrati i nuovi film aggiunti alla piattaforma e i film in tendenza (film più visti nell'ultimo mese). All'utente registrato vengono mostrati anche una serie di film consigliati
- E' possibile ricercare i film(con filtri) e gli utenti
- L'utente non iscritto può vedere le valutazioni degli utenti, ma non le recensioni. Inoltre non può vedere nessuna pagina degli utenti registrati.
- Esistono degli utenti "editore" che possono creare e modificare i film

Progettazione

Casi d'uso



Schema DB



Realizzazione

Tecnologie scelte

Per la realizzazione del progetto è stato usato **Python 3.10** con **Django 4**.

Pacchetti extra:

- *django-braces* → Pacchetto per la gestione dei privilegi, su Classi e Funzioni, in base al gruppo di appartenenza
- *django-crispy-forms* → Pacchetto che permette di renderizzare con un aspetto più gradevole i form generati in automatico
- *Pillow* → Pacchetto che permette una più facile gestione delle immagini all'interno del progetto django

Dal punto di vista del front end è stato scelto di utilizzare **Bootstrap 4** per il **CSS**.

Per rendere più piacevole l'esperienza utente è stato scelto di utilizzare alcuni script **JavaScript** con **AJAX**

Organizzazione logica

Applicazioni

L'applicazione è costituita da due sotto applicazioni:

- **movies**
- **users**

La scelta di quante e quali sotto applicazioni creare ha richiesto tempo, soprattutto per capire quale fosse la best practice. Alla fine basandomi su alcune risposte su StackOverflow e analizzando qualche repository su Github ho deciso di dividere l'applicazione in tante sotto applicazioni quante sono le entità "forti" del mio database: **Movie** e **UserProfile**.

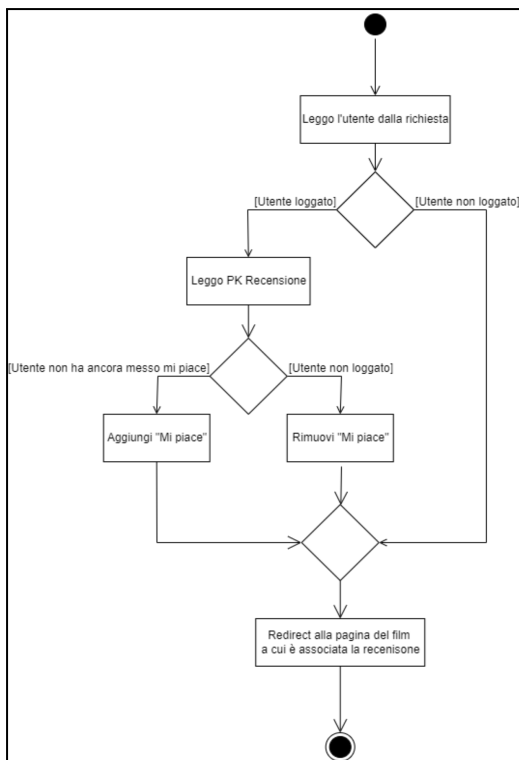
Templates

Per quanto riguarda i templates è stato costruito un template base, che disegna una semplice navbar. Tutti gli altri template sono indipendenti tra loro ma ereditano da quello base.

Casi d'uso d'interesse

Di seguito alcuni casi d'uso che hanno richiesto soluzioni ad-hoc

Gestione like



Caso d'uso: L'utente registrato può mettere mi piace alle recensioni

Soluzione:

Per rendere l'esperienza utente più piacevole è stato deciso di implementare questo caso d'uso sfruttando AJAX. E' stata predisposta una chiamata backend "like" che accetta la pk di una recensione e legge l'utente loggato dalla richiesta.

Viene poi effettuata una chiamata AJAX alla suddetta richiesta e in base alla risposta (like, dislike) viene settata l'apposita icona ad interfaccia

Gestione rimozione da watchlist

Caso d'uso: L'utente registrato può rimuovere film dalla sua watchlist

Soluzione:

Un utente può rimuovere un film dalla propria watchlist per due motivi: perché lo ha visto o perché lo vuole semplicemente rimuovere. Questo diverso comportamento dal punto di vista dell'utente viene mostrato come due button diversi con il quale scegliere il tipo di rimozione. A livello di codice i due tasti appartengono a due form distinti che richiamano lo stesso url, l'unica differenza è un campo hidden "is_viewed" all'interno del form.

Film suggeriti

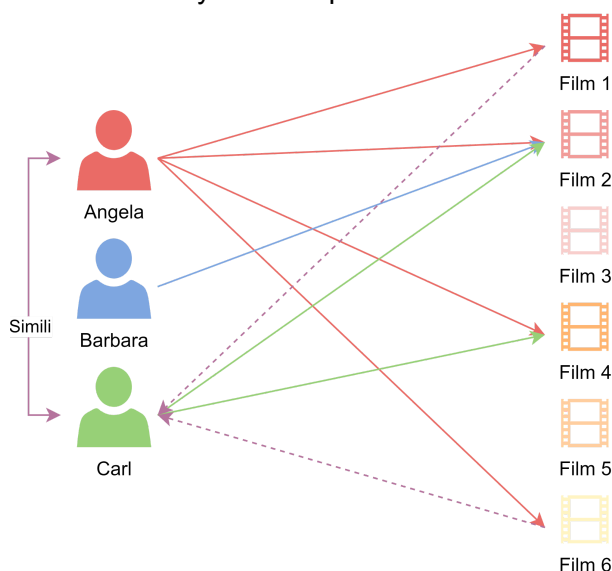
Caso d'uso: L'utente registrato vede in homepage dei film suggeriti

Soluzione:

Questo caso d'uso prevede la realizzazione di un **recommendation system**

Recommendation System

E' stato scelto un Recommendation System di tipo "User-Based Collaborative".



Come mostra l'immagine Angela e Carl hanno più film in comune per cui a Carl verranno suggeriti i film che sono piaciuti ad Angela ma che non ha ancora visto.

Nella mia webapp è possibile fare uno step successivo poiché ci sono più sfumature di gradimento (quante stelline assegnate).

La formula che ho usato prende il nome di *Similarità di Pearson*¹:

$$\text{similarity}(i, j) = \frac{\sum_u r_{(u,i)} r_{(u,j)}}{\sqrt{\sum_u r_{(u,i)}^2} \sqrt{\sum_u r_{(u,j)}^2}}$$

Una volta calcolata la similarità con gli altri utenti, al nostro utente vengono consigliati prima i film dell'utente più simile e poi man mano quello degli utenti meno simili.

Possibile miglioria: Una possibile miglioria potrebbe essere tenere conto del numero di visualizzazioni all'interno del calcolo.

¹ Articolo che ho seguito:

<https://aulascienze.scuola.zanichelli.it/multimedia-scienze/come-te-lo-spiego-scienze/i-suggeriti-per-te-come-la-matematica-stabilisce-i-nostri-gusti-online>

Test

Test su codice applicativo

Come test su codice applicativo ho scelto i metodi *is_actor* e *is_director* della classe *Person*. Questi due metodi controllano rispettivamente se una persona è un attore (ha recitato in almeno un film) e se è un regista (ha diretto almeno un film). Queste due cose non si escludono a vicenda ovviamente.

Test 1

Test: Chiamo la *is_actor* su una persona che non ha recitato in nessun film
Valore atteso: False

Test 2

Test: Chiamo la *is_actor* su una persona che ha recitato in un film
Valore atteso: True

Test 3

Test: Chiamo la *is_director* su una persona che ha mai diretto un film
Valore atteso: False

Test 4

Test: Chiamo la *is_director* su una persona che ha diretto un film
Valore atteso: True

Test su una vista

Come vista da testare ho scelto *ProfileDetails*. Questa vista permette di visualizzare la pagina personale degli utenti. La pagina può essere pubblica o privata, dove pubblica la possono vedere tutti (solo se registrati) mentre privata la può vedere solo chi è nella lista amici dell'utente. La vista *ProfileDetails* permette inoltre al superuser di nominare gli utenti editori.

Test 1

Test: Devo poter accedere alla mia pagina anche se privata
Valore atteso: La richiesta HTTP mi deve ritornare 200 e non devo leggere a video "Pagina privata"

Test 2

Test: Devo poter accedere alla mia pagina pubblica
Valore atteso: La richiesta HTTP mi deve ritornare 200 e non devo leggere a video "Pagina privata"

Test 3

Test: Un utente non loggato non deve vedere le pagine degli utenti loggati anche se pubbliche

Valore atteso: La richiesta HTTP mi deve ritornare 302

Test 4

Test: Da utente loggato non devo poter vedere la pagina degli utenti non amici che hanno la pagina privata

Valore atteso: La richiesta HTTP mi deve ritornare 200 e devo vedere "Pagina privata"

Test 5

Test: Da utente loggato devo poter vedere la pagina degli utenti non amici che hanno la pagina pubblica

Valore atteso: La richiesta HTTP mi deve ritornare 200 e non devo vedere a video "Pagina privata"

Test 6

Test: Da utente loggato devo poter vedere la pagina degli utenti amici che hanno la pagina pubblica

Valore atteso: La richiesta HTTP mi deve ritornare 200 e non devo vedere a video "Pagina privata"

Test 7

Test: Da utente loggato devo poter vedere la pagina degli utenti amici che hanno la pagina privata

Valore atteso: La richiesta HTTP mi deve ritornare 200 e non devo vedere a video "Pagina privata"

Test 8

Test: Da utente admin devo poter nominare editori gli utenti

Valore atteso: La richiesta HTTP mi deve ritornare 200 e devo vedere "Promuovi ad editore"

Screenshot interessanti

Pagina dei film

Modifica

Elimina

5

1979

Halloween

Durata: 91 min



★★★★ 4

Aggiungi alla watchlist 3

2 0

1

Data visione 29/06/2022

Aggiungi visione

1. Possibilità di segnare i watch/rewatch
2. Numero di visualizzazioni totali
3. Possibilità di aggiungere alla watchlist
4. Valutazione media
5. Possibilità di modificare o rimuovere un film

INFOs

Genres

horror

Trama

Un criminale già condannato per l'omicidio della sorella scappa di prigione e torna nella città natale in cerca della prossima vittima durante la notte di Halloween del 1978.


Directors

John Carpenter

Cast

Jamie Lee Curtis

Cliccando su uno dei tasti azzurri è possibile effettuare una ricerca per genere, regista o cast



★★★★★

prova
June 3, 2022

wow

0


👍

Modifica

Elimina

Aggiungi commento

Nascondi



prova
June 27, 2022


wow3

Rimuovi

Così è come si presenta una recensione. Essendo una mia recensione posso modificarla o eliminarla. Inoltre posso aggiungere dei commenti o metterle mi piace. Anche il commento, essendo mio, può essere eliminato.

Pagina personale

admin



Modifica profilo

1

Film preferito

movie15

Genere preferito


animazione

Tempo (ultimo mese)

1 h 30 m

2

Watchlist



Il signore degli anelli: La compagnia dell'anello

3

Visto

Rimuovi

1. Statistiche di visione
2. Watchlist
3. Possibilità di rimuovere film dalla watchlist sia con visione che senza

11