

Raffinamento dei requisiti

1. Requisiti sui Video:

- 1.1. Titolo
- 1.2. Durata
- 1.3. Descrizione
- 1.4. Utente che lo ha pubblicato (v. req. 2)
- 1.5. Nome del file
- 1.6. Categoria (solo una)
 - 1.6.1. Nome della categoria
- 1.7. Tag (almeno una, anche più di una)
 - 1.7.1. Nome del tag
- 1.8. Il video al quale risponde (nessuno o solo uno). Non può rispondere ad un video dello stesso utente che lo ha pubblicato
- 1.9. Numero di visualizzazioni (v. req. 2.3.)
- 1.10. Se è censurato
 - 1.10.1. Il motivo della censura
 - 1.10.2. L'istante di censura

2. Requisiti sugli Utenti:

- 2.1. Nome
- 2.2. Data di iscrizione
- 2.3. Video visualizzati (v. req. 1)
 - 2.3.1. Commenti sui video visualizzati (anche più di uno)
 - 2.3.1.1. Istante del commento
 - 2.3.2. Valutazioni sui video visualizzati (da 0 a 5, solo una)
 - 2.3.2.1. Non dall'utente che la ha pubblicata
 - 2.3.3. Istante di visualizzazione
- 2.4. Video pubblicati (v. req. 1)

3. Requisiti sulle Playlist:

- 3.1. Nome
- 3.2. Data di creazione
- 3.3. Utente che la ha creata (v. req. 2.)
- 3.4. Video che contiene (anche più di uno, v. req. 1)
- 3.5. Se è pubblica o privata

999. Use case:

- 999.A. Utenti
- 999.B. Redazione
 - 999.1. Registrazione utenti

999.2. Playlist pubbliche di un utente

999.3. Pubblicazione video

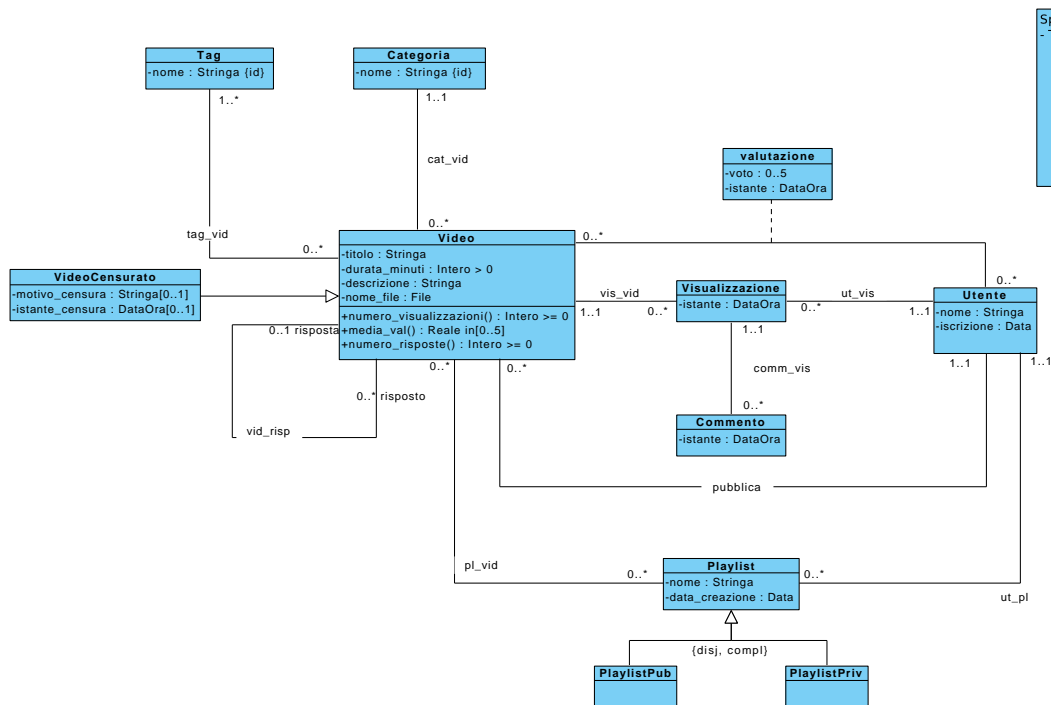
999.4. Creazione e modifica playlist

999.5. Commenti e valutazioni sui video visionati

999.6. ricerca di video: data una categoria, un insieme di tag, ed un intero v tra 0 e 5, restituire tutti i video non censurati disponibili di quella categoria che posseggono almeno uno tra i tag indicati, e che abbiano una valutazione media di almeno v (se un video non ha ancora alcuna valutazione, deve essere restituito comunque).

999.7. Cercare, data una categoria, i video di quella categoria che hanno il numero maggiore di video in risposta, al fine di isolare le discussioni più animate tra gli utenti.

999.8. Censura di video da parte della redazione



Specifica dei tipi di dato:
- Tipo File: Tipo Stringa secondo standard

Specifica delle classi

Specifica della classe Video

numero_visualizzazioni() : Intero ≥ 0 :

- Precondizioni:
 - Nessuna
- Postcondizioni:
 - Sia V l'insieme di tutte le $vis:Visualizzazione$, tali che $(vis, this):vis_vid$
 - $result = |V|$

media_val() : Reale in $[0..5]$:

- Precondizioni:
 - $this$ è coinvolto almeno in un link dell'associazione valutazione
- Postcondizioni:
 - Sia $result$ definito come segue:
 - Sia V l'insieme di tutti i link dell'associazione valutazione che coinvolgono $this$
 - Sia S la somma di tutti i valori per gli attributi voto dei link in V
 - $result = S / |V|$

numero_risposte() : Intero ≥ 0 :

- Precondizioni:
 - Nessuna
- Postcondizioni:
 - Sia V l'insieme di tutti i $vid2:Video$, tali che partecipano al link dell'associazione vid_resp , nel quale $vid2$ ha ruolo di risposto e $this$ ha ruolo di risposto
 - Sia $result = |V|$

[V.Video.risposta.utente_non_autorisposta] Un utente non può rispondere con un video ai suoi stessi video

Per ogni $ut1:Utente$, $ut2:Utente$, $vid1:Video$ e $vid2:Video$, tali che:

- $vid1$ partecipa con il ruolo di risposto e $vid2$ partecipa con il ruolo di risposta al link dell'associazione $(vid1, vid2):vid_resp$,
- $(vid1, ut1):pubblica$
- $(vid2, ut2):pubblica$

Non deve essere che $ut1 = ut2$

Specifica della classe Utente

[V.Utente.Valutazione.valuta_visualizzati] Un utente non può valutare un video che non ha visualizzato

Per ogni $ut:Utente$ e $vid:Video$, tali che $(ut, vid):valutazione$, deve esistere una $vis:Visualizzazione$ tale che $(ut, vis):ut_vis$ e che $(vis, vid):vis_vis$.

[V.Utente.Valutazione.no_autovalutazione] Un utente non può valutare un video pubblicato da se stesso.

Per ogni $ut:Utente$ e $vid:Video$, non deve essere contemporaneamente che $(ut, vid):valutazione$ e che $(ut, vid):pubblica$

[V.Utente.Valutazione.non_su_video_censurati] Un utente non può valutare video dopo che sono stati censurati

Per ogni $ut:Utente$ e $vid:Video$, tali che $(ut, vid):valutazione$, non deve essere che vid è anche istanza di `VideoCensurato` e che $(ut, vid).istante \geq (ut, vid).istante_censura$

Specifica della classe Visualizzazione

[V.Visualizzazione.non_su_video_censurati] Un utente non può visualizzare video dopo che sono stati censurati

Per ogni vis:Visualizzazione e vid:Video, tali che (vis, vid):vis_vid, non deve essere che vid è anche istanza di VideoCensurato e che vis.istante >= vid.istante_censura

[V.Visualizzazione.Commento.commenta_visualizzati] L'istante del commento deve venire dopo l'istante della visualizzazione

Per ogni vis:Visualizzazione e comm:Commento, tali che (vis, comm):comm_vis, deve essere che vis.istante <= comm.istante

Specifica della classe Commento

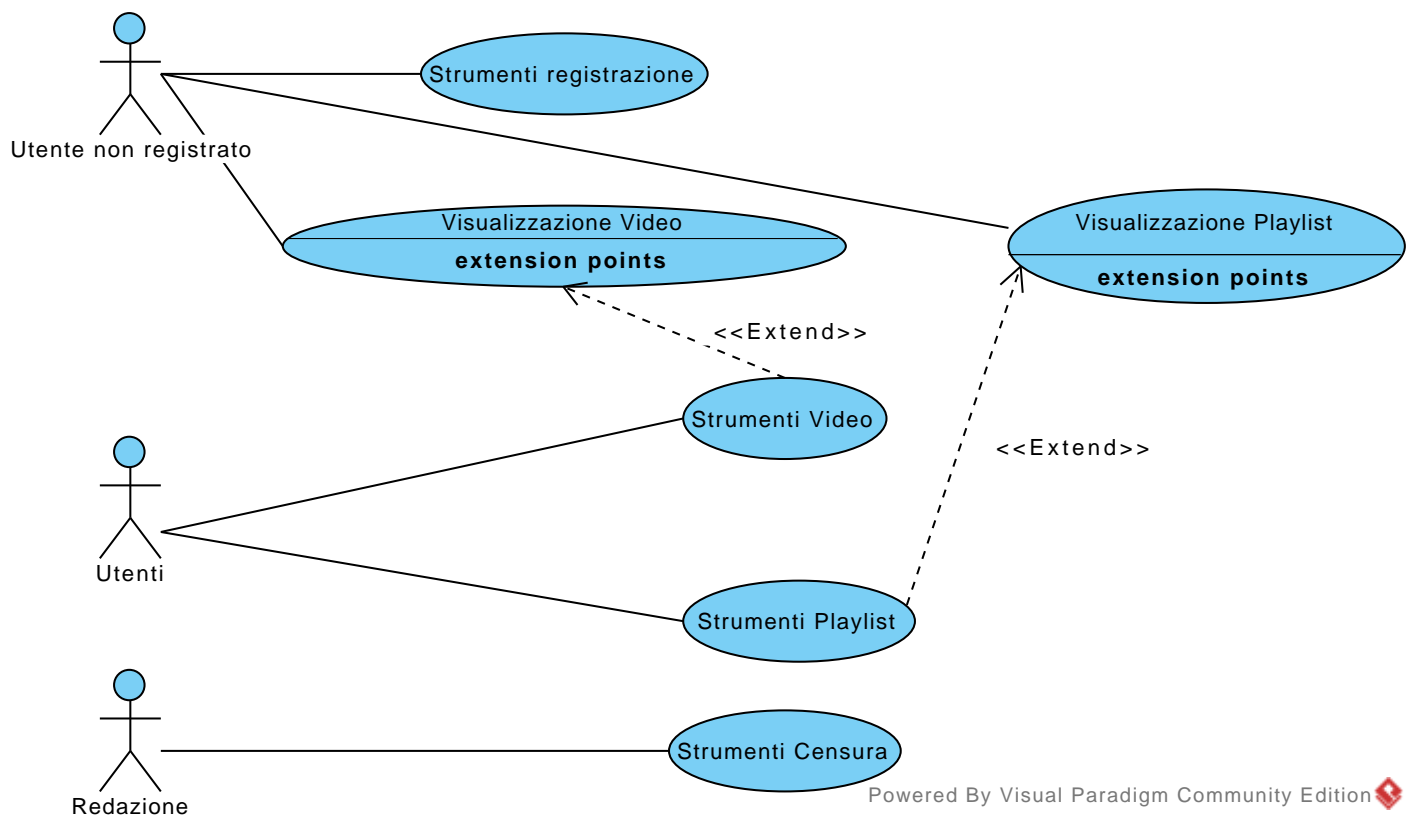
[V.Commento.Video.non_su_video_censurati] Un utente non può commentare video dopo che sono stati censurati

Per ogni comm:Commento, vis:Visualizzazione e vid:Video, tali che (comm, vis):comm_vis e che (vis, vid):vis_vid, non deve essere che vid è anche istanza di VideoCensurato e che comm.istante >= vid.istante_censura

Specifica della classe Playlist

[V.Playlist.Video.non_contiene_censurati] Una playlist non può contenere video censurati

Per ogni pl:Playlist e vid:Video, tali che (pl, vid):pl_vid, non deve essere che vid sia anche istanza di VideoCen



Specifica di Use Case

Strumenti Playlist

trova_playlists(*ut*: Utente) : PlaylistPub[0..*]:

- Precondizioni:
 - Nessuna
- Postcondizioni:
 - Sia result l'insieme di tutte le *pl*:PlaylistPub tali che (*ut*, *pl*):*ut_pl*

Strumenti Video

ricerca_video(*cat*: Categoria, *tags*: Tag[0..*], *v*: 0..5) : Video[0..*]:

- Precondizioni:
 - Nessuna
- Postcondizioni:
 - Sia result l'insieme di tutti i *vid*:Video, tali che:
 - *vid* NON è anche istanza di VideoCensurato
 - Esista il link (*cat*, *vid*):*cat_vid*
 - Esiste almeno un *t*:Tag in *tags* tale che esista il link (*t*, *vid*):*tag_vid*
 - *vid* non è coinvolto in alcun link dell'associazione valutazione oppure *vid*.*media_val()* >= *v*

vid_piu_discussi(*cat*: Categoria) : Video[0..*]:

- Precondizioni:
 - Nessuna
- Postcondizioni:
 - Sia *V* l'insieme di tutti le coppie (*vid*, *n*), dove:
 - *vid*:Video, tale che (*cat*, *vid*):*cat_vid*
 - *n* è il valore di *vid*.*numero_risposte()*
 - Sia *V_Max* l'insieme di tutte le coppie (*vid*, *n*) in *V* che hanno valore di *n* pari al massimo
 - Sia result l'insieme di tutti i valori di *vid* in *V_Max*