

TuTubi

Indice

1	Requisiti	2
2	Specifica dei tipi di dato	4
3	Specifica delle associazioni	5
4	Specifica delle operazioni	6
5	Specifica dei Vincoli	28
6	Diagramma delle classi	29

1 Requisiti

1. Utente

- 1.1. nome: `StringS`
- 1.2. dataIscrizione: `DataOra`
- 1.3. I `video` pubblicati dall'utente
- 1.4. Le `valutazioni` che ha espresso
- 1.5. I `commenti` che ha pubblicato

2. Video

- 2.1. titolo: `StringS`
- 2.2. La durata (operazione)
- 2.3. descrizione: `StringL`
- 2.4. nomeFile: `StringM`
- 2.5. La `categoria` (unica)
- 2.6. I `tag` (almeno uno)
- 2.7. Se è un video in risposta a un altro video:
 - 2.7.1. Il video a cui risponde
 - 2.7.2. Non può essere in risposta a un video pubblicato dallo stesso utente (vincolo)
- 2.8. Numero di visualizzazioni (operazione)
- 2.9. Media valutazioni (operazione)
- 2.10. Se è stato `censurato`
- 2.11. Numero di valutazioni (operazione)

3. Categoria

- 3.1. nome: `StringS`

4. Tag

- 4.1. nome: `StringS`

5. Cronologia

- 5.1. L'`utente` che ha effettuato la visualizzazione
- 5.2. Il `video` visualizzato
- 5.3. dataVisualizzazione: `DataOra`

6. Valutazione

- 6.1. voto: `Voto`
- 6.2. L'utente che ha pubblicato il video non può votarlo (vincolo)

- 6.3. Gli altri utenti possono votare un video al più una sola volta (vincolo)
- 6.4. Un utente può esprimere una valutazione solo su un video che ha visualizzato (vincolo)
- 6.5. L'**utente** che ha espresso la valutazione
- 6.6. Il **video** valutato
- 7. Commento
 - 7.1. testo: **StringL**
 - 7.2. dataPubblicazione: DataOra
 - 7.3. Gli utenti possono esprimere più commenti sullo stesso video (vincolo)
 - 7.4. Un utente può scrivere un commento solo su un video che ha visualizzato (vincolo)
- 8. Playlist
 - 8.1. I **video** ordinati
 - 8.1.1. DataOra di inserimento in playlist
 - 8.2. nome: **StringS**
 - 8.3. dataCreazione: DataOra
 - 8.4. stato: **StatoPlaylist**
- 9. Video Censurato
 - 9.1. La **motivazione** della censura
 - 9.2. Non può essere visualizzato (vincolo)
 - 9.3. Non può essere valutato (vincolo)
 - 9.4. Non può essere commentato (vincolo)
 - 9.5. Non può essere inserito in una **playlist** (vincolo)
 - 9.6. Non può essere restituito come risultato di una ricerca (vincolo)
 - 9.7. Una volta censurato non può più essere ripristinato (vincolo)
 - 9.8. commentoAggiuntivo: **StringL**
- 10. MotivazioneCensura
 - 10.1. nome: **StringS**
 - 10.2. descrizione: **StringL**

2 Specifica dei tipi di dato

1. InteroGZ: $\text{Intero} > 0$
2. InteroGEZ: $\text{Intero} \geq 0$
3. StringS: Stringa di 75 caratteri
4. StringM: Stringa di 600 caratteri
5. StringL: Stringa di 4200 caratteri
6. Voto: Intero tra 0 e 5
7. RealeGEZ: $\text{Reale} \geq 0$
8. StatoPlaylist: {Pubblica, Privata}
9. Video: File video

3 Specifica delle associazioni

1. $\forall u, v \text{ utVid}(u, v) \rightarrow \text{Utente}(u) \wedge \text{Video}(v)$
2. $\forall c, v \text{ cronVid}(c, v) \rightarrow \text{Cronologia}(c) \wedge \text{Video}(v)$
3. $\forall u, v \text{ valutazione}(u, v) \rightarrow \text{Utente}(u) \wedge \text{Video}(v)$
4. $\forall c, v \text{ comVid}(c, v) \rightarrow \text{Commento}(c) \wedge \text{Video}(v)$
5. $\forall c, u \text{ comUt}(c, u) \rightarrow \text{Commento}(c) \wedge \text{Utente}(u)$
6. $\forall p, v \text{ plVid}(p, v) \rightarrow \text{Playlist}(p) \wedge \text{Video}(v)$
7. $\forall p, u \text{ plUt}(p, u) \rightarrow \text{Playlist}(p) \wedge \text{Utente}(u)$
8. $\forall c, v \text{ catVid}(c, v) \rightarrow \text{Categoria}(c) \wedge \text{Video}(v)$
9. $\forall t, v \text{ tagVid}(t, v) \rightarrow \text{Tag}(t) \wedge \text{Video}(v)$
10. $\forall v', v'' \text{ vidInRispostaA}(v', v'') \rightarrow \text{Video}(v') \wedge \text{Video}(v'') \wedge \text{rispostaVid}(v') \wedge \text{superVid}(v'')$
11. $\forall v, m \text{ motCenVidCen}(m, v) \rightarrow \text{MotivazioneCensura}(m) \wedge \text{VideoCensurato}(v)$

4 Specifica delle operazioni

1. L'Utente deve poter:

1.1. pubblicaVideo(u: Utente, t: `StringS`, d: `StringL`, f: `Video`, cat: `StringS`, tag: `StringS`)

1.1.1. precondizioni:

1.1.1.1. $\forall v \text{ utVid}(u,v) \rightarrow \neg \text{titolo}(v,t)$

1.1.2. postcondizioni:

1.1.2.1. Viene creata una nuova istanza v di Video e viene associata all'utente u tale che:

1.1.2.2. `Video(v)`

1.1.2.3. `titolo(v,t)`

1.1.2.4. `descrizione(v,d)`

1.1.2.5. `file(v,f)`

1.1.2.6. `catVid(cat,v)`

1.1.2.7. `tagVid(tag,v)`

1.1.2.8. `utVid(u,v)`

- 1.2. pubblicaVideoInRisposta(u: Utente, t: **StringS**, d: **StringL**, f: **Video**, cat: **StringS**, tag: **StringS**, v': Video)
 - 1.2.1. precondizioni:
 - 1.2.1.1. $\forall v \text{ utVid}(u,v) \rightarrow \neg \text{titolo}(v,t)$
 - 1.2.1.2. Sia u' l'utente che ha pubblicato v', allora $u' \neq u$
 - 1.2.1.3. $\neg \text{VideoCensurato}(v')$
 - 1.2.2. postcondizioni:
 - 1.2.2.1. Viene pubblicato un nuovo video: pubblicaVideo(u,t,d,f,cat,tag)
 - 1.2.2.2. Viene creata una nuova associazione: vidInRispostaA(v,v')

1.3. visualizzaVideo(u: Utente, v: Video)

1.3.1. precondizioni:

1.3.1.1. $\neg \text{VideoCensurato}(v)$

1.3.2. postcondizioni:

1.3.2.1. Viene creata una nuova istanza c di Cronologia e viene associata all'utente u e al video v tale che:

1.3.2.2. $\text{Cronologia}(c)$

1.3.2.3. $\text{dataVisualizzazione}(c, \text{istanteCorrente})$

1.3.2.4. $\text{cronVid}(c, v)$

1.3.2.5. $\text{cronUt}(c, u)$

1.4. valutaVideo(u: Utente, v: Video, val: **Voto**)

1.4.1. precondizioni:

1.4.1.1. Sia u' l'utente che ha pubblicato v , allora $u' \neq u$

1.4.1.2. $\neg \text{VideoCensurato}(v)$

1.4.2. postcondizioni:

1.4.2.1. Viene creata una nuova istanza dell'associazione Valutazione associata all'utente u e al video v tale che:

1.4.2.2. $\text{Valutazione}(u,v)$

1.4.2.3. $\text{voto}(u,v,\text{val})$

1.4.2.4. $\text{dataValutazione}(u,v,\text{istanteCorrente})$

1.5. pubblicaCommento(u: Utente, v: Video, com: [StringL](#))

1.5.1. precondizioni:

1.5.1.1. \neg VideoCensurato(v)

1.5.2. postcondizioni:

1.5.2.1. Viene creata una nuova istanza c di Commento e viene associata all'utente u e al video v tale che:

1.5.2.2. Commento(c)

1.5.2.3. testo(c,com)

1.5.2.4. dataPubblicazione(c,istanteCorrente)

1.5.2.5. comVid(c,v)

1.5.2.6. comUt(c,u)

1.6. creaPlaylist(u: Utente, n: [StringS](#), d: DataOra, s: StatoPlaylist)

1.6.1. precondizioni:

1.6.1.1. $\forall p \text{ plUt}(p,u) \rightarrow \neg \text{nome}(p,n)$

1.6.2. postcondizioni:

1.6.2.1. Viene creata una nuova istanza p di Playlist e viene associata all'utente u tale che:

1.6.2.2. $\text{Playlist}(p)$

1.6.2.3. $\text{nome}(p,n)$

1.6.2.4. $\text{dataCreazione}(p,d)$

1.6.2.5. $\text{stato}(p,s)$

1.6.2.6. $\text{plUt}(p,u)$

- 1.7. aggiungiVideoAPlaylist(u: Utente, p: Playlist, v: Video)
 - 1.7.1. precondizioni:
 - 1.7.1.1. $plUt(p,u)$
 - 1.7.2. postcondizioni:
 - 1.7.2.1. viene creata una nuova associazione $plVid(p,v)$ tale che:
 - 1.7.2.2. $plVid(p,v)$
 - 1.7.2.3. $dataInserimento(p,v,istanteCorrente)$

1.8. rimuoviVideoDaPlaylist(u: Utente, p: Playlist, v: Video)

1.8.1. precondizioni:

1.8.1.1. $plUt(p,u)$

1.8.1.2. $plVid(p,v)$

1.8.2. postcondizioni:

1.8.2.1. viene rimossa l'associazione $plVid(p,v)$

- 1.9. ottieniVideoInPlaylist(p: Playlist): Video [0..*]
 - 1.9.1. precondizioni:
 - 1.9.1.1. Sia s tale che $\text{stato}(p,s) \rightarrow s = \text{Pubblica}$
 - 1.9.2. postcondizioni:
 - 1.9.2.1. no side effects
 - 1.9.3. return:
 - 1.9.3.1. Sia $V = \{(d,v) \mid \forall v,d \text{ plVid}(p,v) \wedge \text{dataInserimento}(p,v,d)\}$

- 1.10. ottieniPlaylist(u: Utente): Playlist [0..*]
 - 1.10.1. precondizioni:
 - 1.10.1.1. nessuna
 - 1.10.2. postcondizioni:
 - 1.10.2.1. No side effects
 - 1.10.3. return:
 - 1.10.3.1. Sia $P = \{p \mid \forall p,s \text{ plUt}(p,u) \wedge \text{stato}(p,s) \rightarrow s = \text{Pubblica}\}$
 - 1.10.3.2. return P

1.11. iscrizioneUtente(u: Utente, nome: [StringS](#))

1.11.1. precondizioni:

1.11.1.1. $\forall u' \text{ Utente}(u') \rightarrow \neg \text{nome}(u', \text{nome})$

1.11.2. postcondizioni:

1.11.2.1. Viene creata una nuova istanza u di Utente tale che:

1.11.2.2. $\text{Utente}(u)$

1.11.2.3. $\text{nome}(u, \text{nome})$

1.11.2.4. $\text{dataIscrizione}(u, \text{istanteCorrente})$

1.12. F.videoDaCatTagVal(cat: [StringS](#), tag: [StringS](#), val: [Voto](#)): Video [0..*]

1.12.1. precondizioni: nessuna

1.12.2. postcondizioni: nessuna

1.12.3. return:

1.12.3.1. Sia $C = \text{ottieniVideoDaCategoria}(\text{cat})$

1.12.3.2. Sia $T = \text{ottieniVideoDaTag}(\text{tag})$

1.12.3.3. $CT = C \cap T$

1.12.3.4. Sia $V = \{v \mid v \in CT \wedge (\text{mediaValutazioni}(v) \geq \text{val} \vee \text{numeroValutazioni}(v) = 0)\}$

1.12.3.5. return V

1.13. F.videoConMaggiorNumeroVideoInRisposta(c: Categoria): Video [0..*] Ordinati in modo decrescente

1.13.1. precondizioni: nessuna

1.13.2. postcondizioni: nessuna

1.13.3. return:

1.13.3.1. $V = \text{ottieniVideoDaCategoria}(c)$

1.13.3.2. Sia $V' = \{(\text{contaNumeroVideoInRisposta}(v), v) \mid v \in V\}$

1.13.3.3. return V' ordinato in modo decrescente

2. La Redazione deve poter:

2.1. `censuraVideo(v: Video, c: StringL, m: StringS)`

2.1.1. precondizioni:

2.1.1.1. $\neg \text{VideoCensurato}(v)$

2.1.2. postcondizioni:

2.1.2.1. Viene creata una nuova istanza di `VideoCensurato` e viene associata a una nuova istanza di `MotivazioneCensura` tale che:

2.1.2.2. `VideoCensurato(v)`

2.1.2.3. `dataCensura(v,istanteCorrente)`

2.1.2.4. `commentoAggiuntivo(v,c)`

2.1.2.5. `motCenVidCen(m,v)`

3. Il Sistema deve poter:

3.1. ottieniVideoDaCategoria(cat: [StringS](#)): Video [0..*]

3.1.1. precondizioni: nessuna

3.1.2. postcondizioni: nessuna

3.1.3. return:

3.1.3.1. Sia $V = \{v \mid \text{catVid}(\text{cat}, v) \wedge \neg \text{VideoCensurato}(v)\}$

3.1.3.2. return V

- 3.2. ottieniVideoDaTag(tag: **StringS**): Video [0..*]
 - 3.2.1. precondizioni: nessuna
 - 3.2.2. postcondizioni: nessuna
 - 3.2.3. return:
 - 3.2.3.1. Sia $V = \{v \mid \text{tagVid}(\text{tag}, v) \wedge \neg \text{VideoCensurato}(v)\}$
 - 3.2.3.2. return V

3.3. `contaNumeroVideoInRisposta(v: Video)`: [InteroGEZ](#)

3.3.1. precondizioni: $\neg \text{VideoCensurato}(v)$

3.3.2. postcondizioni: nessuna

3.3.3. return:

3.3.3.1. Sia $V = \{v' \mid \text{vidInRispostaA}(v', v) \wedge \neg \text{VideoCensurato}(v')\}$

3.3.3.2. $t = \sum_{v' \in V} \text{contaNumeroVideoInRisposta}(v')$

3.3.3.3. return t

3.4. sommaValutazioni(v: Video): [InteroGEZ](#)

3.4.1. precondizioni: $\neg \text{VideoCensurato}(v)$

3.4.2. postcondizioni: nessuna

3.4.3. return:

3.4.3.1. Sia $V = \{(u,v, \text{val}) \mid \forall u,v, \text{val} \text{ valutazione}(u,v) \wedge \text{voto}(u,v, \text{val})\}$

3.4.3.2. $t = \sum_{(x,y,z) \in V} \text{contaNumeroVideoInRisposta}(z)$

3.4.3.3. return t

3.5. numeroValutazioni(v: Video): [InterroGEZ](#)

- 3.5.1. precondizioni: $\neg \text{VideoCensurato}(v)$
- 3.5.2. postcondizioni: nessuna
- 3.5.3. return:
 - 3.5.3.1. Sia $V = \{(u,v) \mid \forall \text{ valutazione}(u,v)\}$
 - 3.5.3.2. return $|V|$

4. Il Video deve poter:

4.1. `durataSecondi()`: [InteroGEZ](#)

4.1.1. precondizioni: nessuna

4.1.2. postcondizioni: nessuna

4.1.3. return: durata in secondi

- 4.2. numeroVisualizzazioni(): [InterroGEZ](#)
- 4.2.1. precondizioni: nessuna
- 4.2.2. postcondizioni: nessuna
- 4.2.3. return:
 - 4.2.3.1. Sia $C = \{c \mid \text{cronVid}(c, \text{this})\}$
 - 4.2.3.2. return $|C|$

4.3. mediaValutazioni(): [RealeGEZ](#)

4.3.1. precondizioni:

4.3.1.1. numeroValutazioni() > 0

4.3.2. postcondizioni:

4.3.2.1. nessuna

4.3.3. return:

4.3.3.1. Sia $s = \text{sommaValutazioni}(\text{this})$

4.3.3.2. sia $n = \text{numeroValutazioni}(\text{this})$

4.3.3.3. return $\frac{s}{n}$

5 Specifica dei Vincoli

1. V.Video:

1.1. V.Video.VideoInRisposta.AutoreRispostaDiversoDaAutoreVideo:

$$1.1.1. \forall v, v', u, u' \text{ vidInRispostaA}(v, v') \wedge \text{utVid}(u, v) \wedge \text{utVid}(u', v') \rightarrow u \neq u'$$

1.2. V.Video.VideoCensuratoNonVisualizzabile:

$$1.2.1. \forall v, dc, dv, c \text{ VideoCensurato}(v) \wedge \text{dataCensura}(v, dc) \wedge \text{cronVid}(c, v) \wedge \text{dataVisualizzazione}(c, dv) \rightarrow dv <_{dc}$$

1.3. V.Video.VideoCensuratoNonValutabile:

$$1.3.1. \forall v, dc, val, dv \text{ VideoCensurato}(v) \wedge \text{dataCensura}(v, dc) \wedge \text{valutazione}(val, v) \wedge \text{dataValutazione}(val, dv) \rightarrow dv <_{dc}$$

1.4. V.Video.VideoCensuratoNonCommentabile:

$$1.4.1. \forall v, dc, com, dp \text{ VideoCensurato}(v) \wedge \text{dataCensura}(v, dc) \wedge \text{comVid}(com, v) \wedge \text{dataPubblicazione}(com, dp) \rightarrow dp <_{dc}$$

2. V.Cronologia:

2.1. V.Cronologia.VideoVisualizzatoDopoCreazioneUtente

$$2.1.1. \forall u, c, dv, di \text{ cronUt}(c, u) \wedge \text{dataVisualizzazione}(c, di) \wedge \text{dataIscrizione}(u, di) \rightarrow dv \geq di$$

3. V.Valutazione:

3.1. V.Valutazione.VideoAutoreNonPuòValutareSeStesso:

$$3.1.1. \forall u, v \wedge \text{valutazione}(u, v) \rightarrow \neg \text{utVid}(u, v)$$

3.2. V.Valutazione.UtentePuòValutareSoloVideoCheHaVisualizzato:

$$3.2.1. \forall u, v \text{ valutazione}(u, v) \rightarrow \exists c \text{ cronVid}(c, v) \wedge \text{cronUt}(c, u)$$

4. V.Commento:

4.1. V.Commento.UtentePuòCommentareSoloVideoCheHaVisualizzato:

$$4.1.1. \forall u, com, v \text{ comVid}(com, v) \wedge \text{comUt}(com, u) \rightarrow \exists c \text{ cronVid}(c, v) \wedge \text{cronUt}(c, u)$$

6 Diagramma delle classi

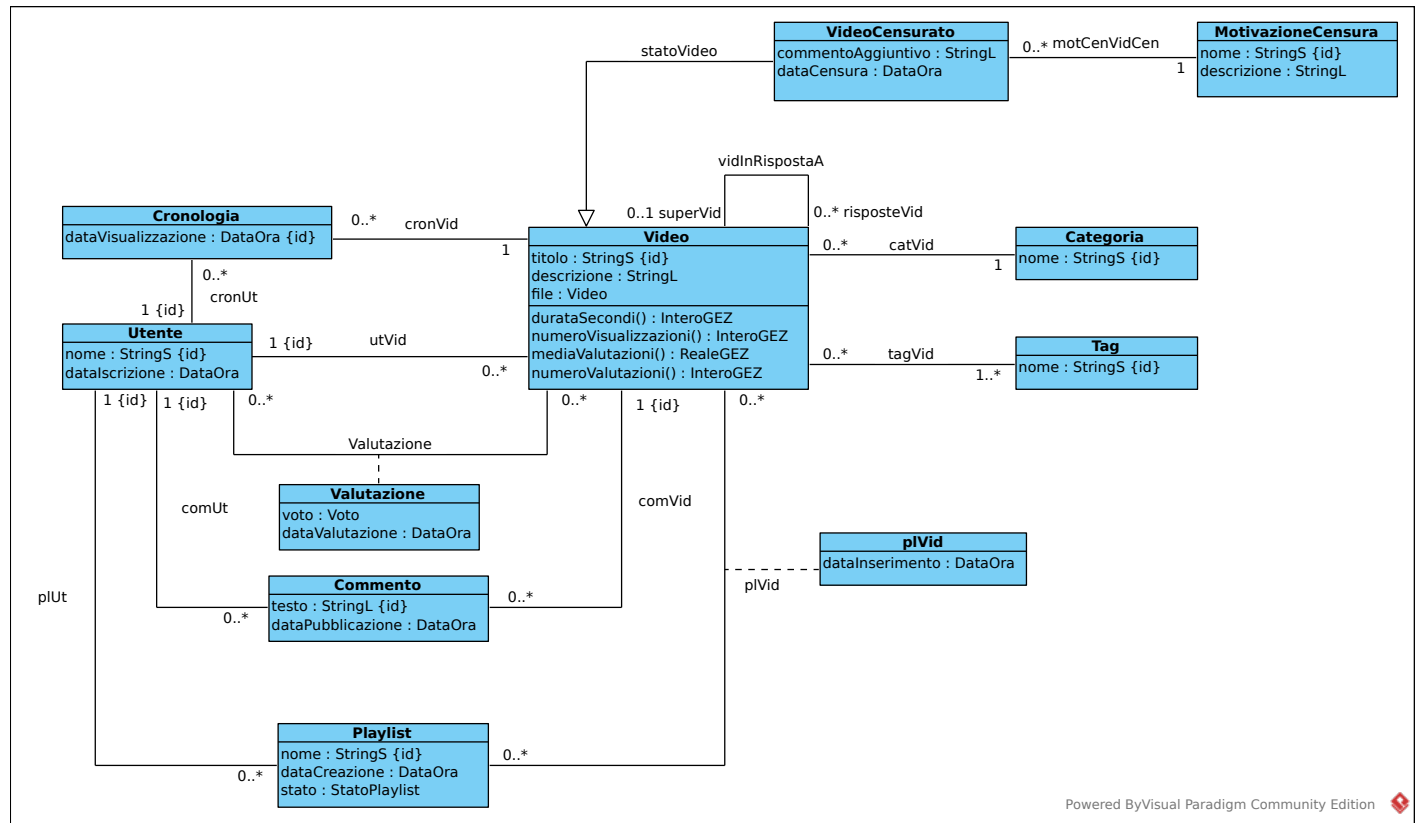


Figure 1: Diagramma delle classi