

Trillium

Documento di architettura - D3

Indice

Indice	1
Scopo del documento	1
Diagramma delle classi	1
1. Utenti e Opzioni	2
2. Gestione dei post e Ricerca	2
3. Registrazione ed Autenticazione	2
Codice in Object Constraint Language	3
1. Utenti e Opzioni	3
Diagramma delle classi con codice OCL	4

Scopo del documento

Il presente documento riporta l'analisi dell'architettura del progetto “**Trillium**” tramite i diagrammi delle classi in **UML** (Unified Model Language) e codice **OCL** (Object Constraint Language). L'analisi prende in considerazione il documento precedente, contenente la definizione dei componenti, diagramma use-case e quello di contesto. Le classi verranno rappresentate tramite UML, mentre la logica e tutti i concetti che ne conseguono saranno rappresentati tramite OCL.

Diagramma delle classi

Nella presente sezione verranno analizzate le classi.

Ogni componente del documento precedente verrà utilizzato per formare una o più classi ed ogni classe sarà rappresentata da un nome, degli attributi (dati gestiti dalla classe) e dei metodi, inoltre ogni classe potrà essere relazionata con altre classi.

Il diagramma delle classi può essere suddiviso in tre componenti principali:

1. Utenti e Opzioni

La classe **User** contiene tutte le informazioni che un utente può inserire nella piattaforma più altre informazioni utili per la gestione dello stesso, ad esempio le date di creazione, dell'ultima modifica e dell'eliminazione dell'account.

Inoltre **User** contiene una lista di amici e di altri utenti che segue e una lista dei propri post e commenti per facilitarne la gestione.

La classe **UserPage** permette di creare uno **User** inserendo tutte le informazioni necessarie oppure di modificarne uno già esistente.

Tale classe non memorizza permanentemente i dati sensibili inseriti dall'utente ma funge solo da interfaccia alla classe **User**.

Un'ulteriore classe **UserSettings** è stata aggiunta per gestire le varie opzioni dell'account dell'utente e per facilitarne l'ampliamento.

2. Gestione dei post e Ricerca

La classe **Post** contiene tutte le informazioni presenti in un post creato da un utente.

In **Post** vengono memorizzate la data di creazione, dell'ultima modifica e della cancellazione se avvenuta e deve essere salvato anche un ID che permetta di identificare i post tra di loro.

Come per lo **User** è stata creata una classe **PostPreferences** per gestire le opzioni del post come la loro visibilità.

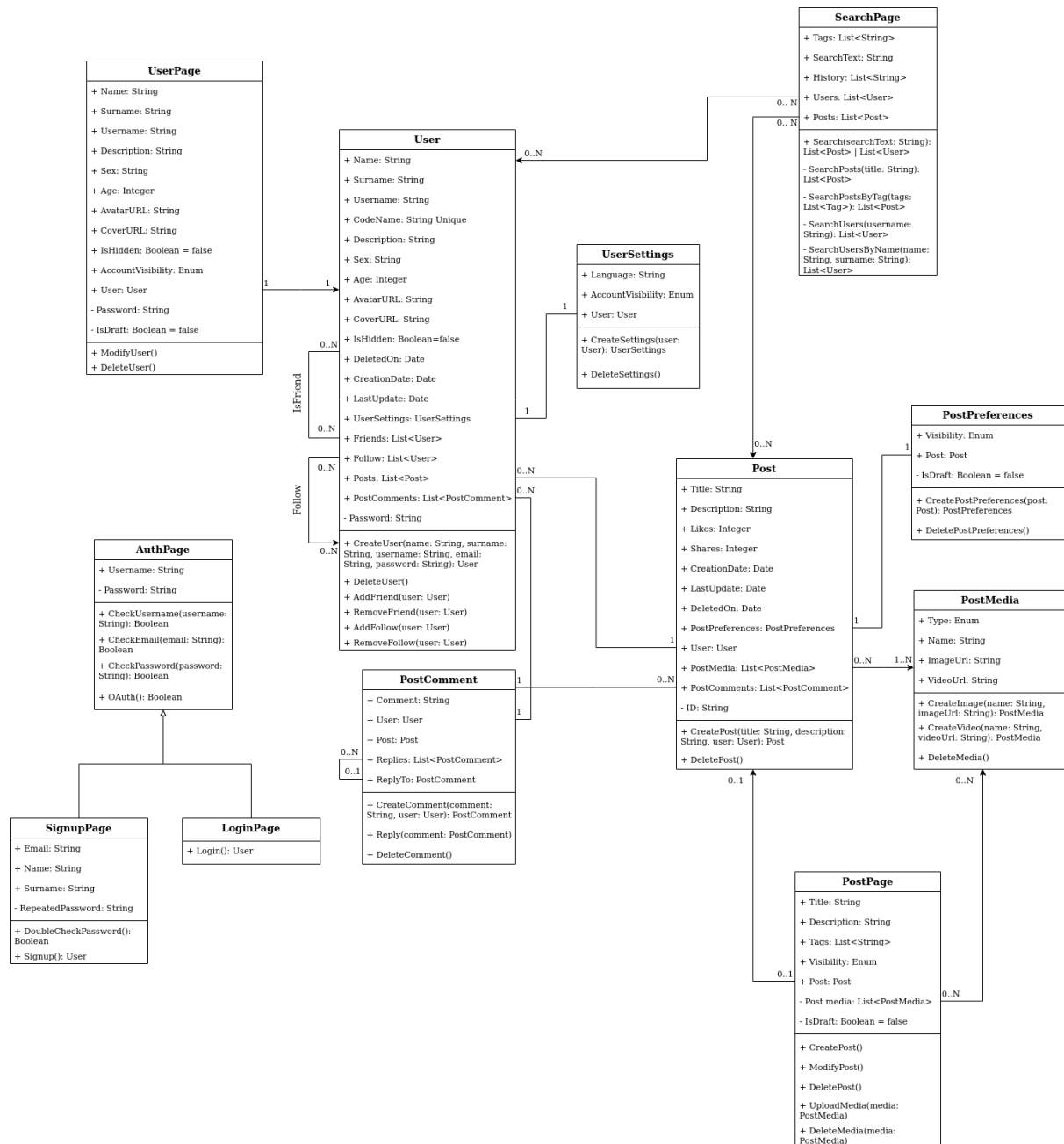
La classe **PostMedia** serve a gestire tutti i media, come immagini, video, ecc..., che vengono aggiunti al post, similmente è stata creata **PostComment** per gestire tutti i commenti che gli utenti possono aggiungere al post.

La classe **PostPage** permette di creare, modificare ed eliminare un post inserendo tutte le informazioni necessarie.

Infine è stata introdotta la classe **SearchPage** che permette ad un utente di cercare dei post tramite il loro titolo, eventualmente filtrandoli per tag, oppure trovare utenti tramite il loro username o nome e cognome

3. Registrazione ed Autenticazione

La registrazione di un account alla piattaforma avviene tramite la classe **SignupPage** mentre l'accesso all'account tramite **LoginPage**, entrambe le pagine sono estensioni della classe **AuthPage** che contiene tutte le operazioni che permettono la verifica dei dati e l'autenticazione.



Codice in Object Constraint Language

In questa sezione verrà illustrata in modo formale la logica delle classi. Tale logica verrà descritta tramite OCL.

1. Utenti e Opzioni

Nella classe **User** l'anno dell'utente non può essere un numero negativo e lo username e l'email non possono essere vuoti.

Ad ogni aggiunta o rimozione di un'amicizia o di un “*follow*” deve essere rimosso il corrispettivo amico (*seguito*) nella lista **Friends** o **Follow** rispettivamente.

Diagramma delle classi con codice OCL

[illegible]