

Elaborazione - Iterazione 3

1 Introduzione

In questa terza iterazione di Elaborazione si prendono in esame i seguenti requisiti:

1. Per ogni post l'utente può definirne la visibilità ed *il contenuto **non** può essere **superiore ai 280 caratteri***.
2. La piattaforma offre la possibilità di ***ricercare i post*** di altri whistleblower sulla base di ***parole chiave***.
3. E' possibile seguire gli account di altri whistleblower per visualizzare i post da loro pubblicati nella piattaforma, ***commentarli*** e mettere like.
4. L'utente, oltre a visualizzare nella propria home i post dei whistleblower seguiti, *può visualizzare post pubblici di qualsiasi whistleblower nella **bacheca***.

I casi d'uso di riferimento sono **UC3: Gestisci Post**, **UC4: Ricerca Post**, **UC5: Gestisci Commento** e **UC7: Visualizza Bacheca**.

Nello specifico ci si concentra sulla:

- implementazione dell'estensione **5a** del caso d'uso **UC3**, relativa alla **verifica del numero di caratteri** consentiti in un post, tralasciata nell'iterazione precedente.
- implementazione dello scenario principale di successo del caso d'uso **UC4** e della sua estensione **4a**, ovvero la **ricerca di un post** contenente una determinata parola chiave e la gestione di una ricerca priva di risultato.
- implementazione dello scenario principale di successo per ciò che riguarda il caso d'uso **UC5**. Prendendo in considerazione anche le estensioni **1a** ed **1b**. Nell'ordine: "inserimento", "modifica" e "rimozione" di un **commento**.
- implementazione del caso d'uso **UC7**, ovvero la visualizzazione della **bacheca**, contenente i post pubblici di qualsiasi whistleblower, anche di coloro che non sono presenti nella cerchia d'interesse del singolo account.

Si rimandano a successive iterazioni:

- la gestione della notifica alla pubblicazione di un post
- l'inserimento del like ad un post

Sono stati riscritti i casi d'uso **UC4**, **UC5** e **UC7** in formato dettagliato.

2 Analisi Orientata agli Oggetti

Nell'analisi orientata agli oggetti si fa uso degli elaborati: **Modello di Dominio**, **SSD** (Sequence System Diagram) e dei **Contratti delle operazioni**, al fine di descrivere il dominio da un punto di vista ad oggetti.

Modello di Dominio

La stesura del modello di dominio ci aiuta a decomporre il dominio in concetti o oggetti significativi. Tale elaborato rappresenta in modo visuale non solo le classi concettuali, ma anche le loro associazioni e gli attributi che le caratterizzano.

Prendendo in esame il caso d'uso **UC4** il modello di dominio rimane invariato rispetto a quello mostrato nell'Elaborazione 2, acquisiscono però maggior importanza le associazioni "E'-noto-a" tra Post e Whistler e tra Keyword e Whistler. Infatti, al fine di ricercare i post contenenti una determinata parola chiave Whistler deve conoscere i post presenti nella piattaforma e le keywords.

Lo stesso avviene per il caso d'uso **UC7** nello specifico per l'associazione "E'-nota-a" tra Keyword e Whistler. Infatti, al fine di poter definire quali sono le Keywords di tendenza, necessarie per la visualizzazione nella bacheca dei post pubblici che le contengono, il Sistema deve conoscere tutte le Keywords inserite nella piattaforma.

Prendendo in considerazione, invece, il caso d'uso **UC5** è possibile identificare la seguente classe concettuale:

- **Commento:** è il messaggio di risposta riferito al contenuto di un Post

Il modello di dominio ricavato è il seguente:

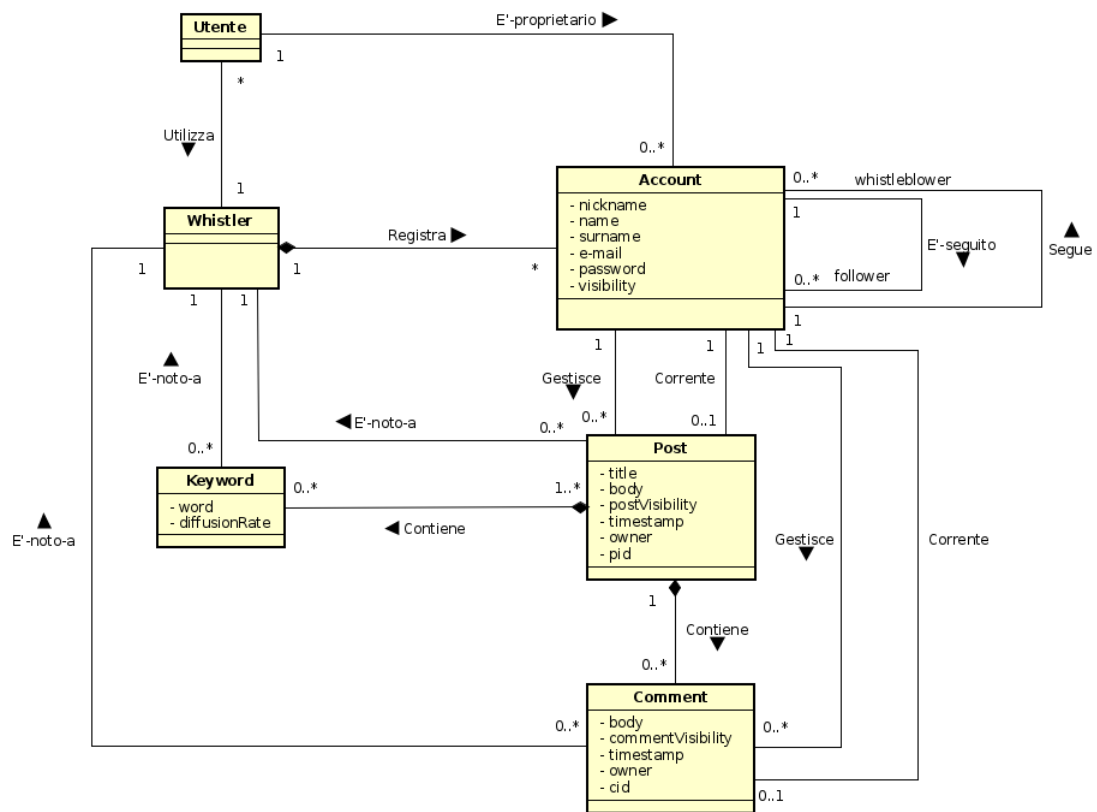
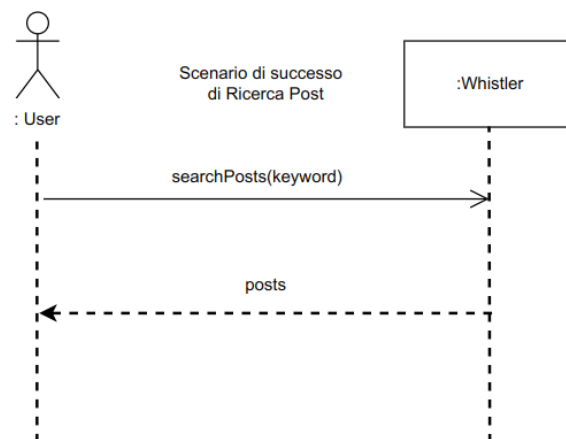


Diagramma di sequenza di sistema (SSD)

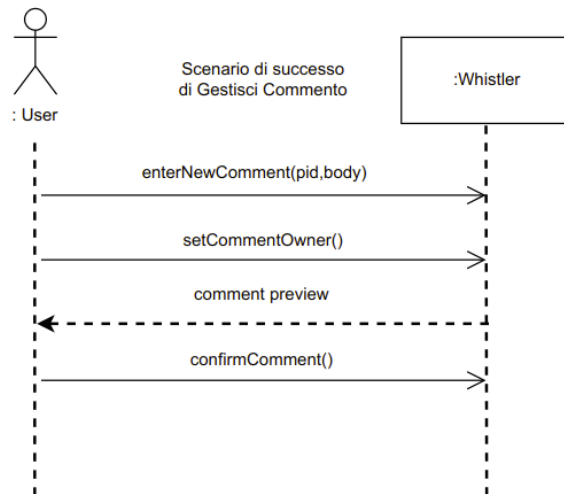
I diagrammi di sequenza di sistema, sono degli elaborati che mostrano gli eventi di input e di output relativi al sistema in discussione (a scatola nera).

Di seguito si riportano gli SSD ottenuti analizzando i casi d'uso presi in esame, sia nei loro **scenari principali di successo** che negli **scenari alternativi** più frequenti o complessi.

Per ciò che riguarda il caso d'uso **UC4**:

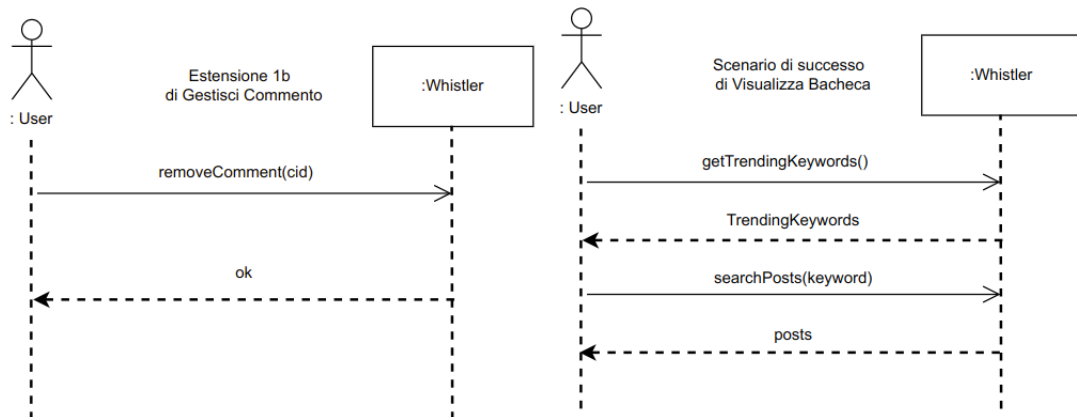


Per il caso d'uso **UC5**:



Durante l'inserimento di ciascun commento: l'operazione d'inserimento del **time-stamp**, la generazione del **cid** (comment_id) e la **visibilità** del commento vengono gestite internamente dal sistema. Nello specifico la visibilità del commento sarà identica a quella del post a cui è riferito.

Si riportano di seguito l'SSD dell'estensione **1b** relativa alla rimozione di un commento e l'SSD dello scenario di successo del caso d'uso **UC7**.



(a) estensione **UC5_1b** relativa alla rimozione di un commento

(b) caso d'uso **UC7**

Contratti delle Operazioni

Di seguito si riportano i contratti delle operazioni di sistema ritenuti di maggiore rilevanza e identificati mediante l'analisi degli SSD precedentemente elaborati. I contratti delle operazioni permettono di fornire maggiori dettagli sull'effetto delle operazioni di sistema.

Contratto CO1: searchPosts

Operazione : searchPosts(keyword:String)

Riferimenti : caso d'uso: Ricerca Post

Pre-condizioni :

- l'Utente possiede un account ed è autenticato nella piattaforma

Post-condizioni :

- il Sistema ha ritornato i posts contenenti la keyword indicata

Contratto CO2: enterNewComment

Operazione : enterNewComment(pid:String, body:String)

Riferimenti : caso d'uso: Gestisci Commento

Pre-condizioni :

- l'Utente possiede un account ed è autenticato nella piattaforma

- è presenta almeno un post, visibile all'Utente, nella piattaforma

Post-condizioni :

- è stata creata una nuova istanza c di Commento;

- gli attributi di c sono stati inizializzati;

- c è stato associato ad Account tramite l'associazione "Corrente"

Contratto CO3: setCommentOwner

Operazione : setCommentOwner()

Riferimenti : caso d'uso: Gestisci Commento

Pre-condizioni :

- è in corso l'inserimento di un nuovo Commento c

Post-condizioni :

- c.owner è diventato account.nickname

Contratto CO4: confirmComment

Operazione : confirmComment()

Riferimenti : caso d'uso: Gestisci Commento

Pre-condizioni :

- è in corso l'inserimento di un nuovo Commento c

Post-condizioni :

- è stata associata l'istanza c di Commento corrente ad Account tramite l'associazione "Gestisce";

- è stata associata l'istanza c di Commento corrente al Post, avente il pid indicato, tramite l'associazione "Contiene";

- l'istanza c di Commento corrente è nota a Whistler.

Contratto CO5: removeComment**Operazione :** removeComment(cid:String)**Riferimenti :** caso d'uso: Gestisci Commento**Pre-condizioni :**

- su Whistler è presente un commento c con cid corrispondente a quello in esame;
- l'account che si accinge a rimuovere il commento è proprietario di quel commento

Post-condizioni :

- l'associazione di c con il Post che lo conteneva è stata rimossa;
- l'associazione di c con Account è stata rimossa;
- c è stato rimosso.

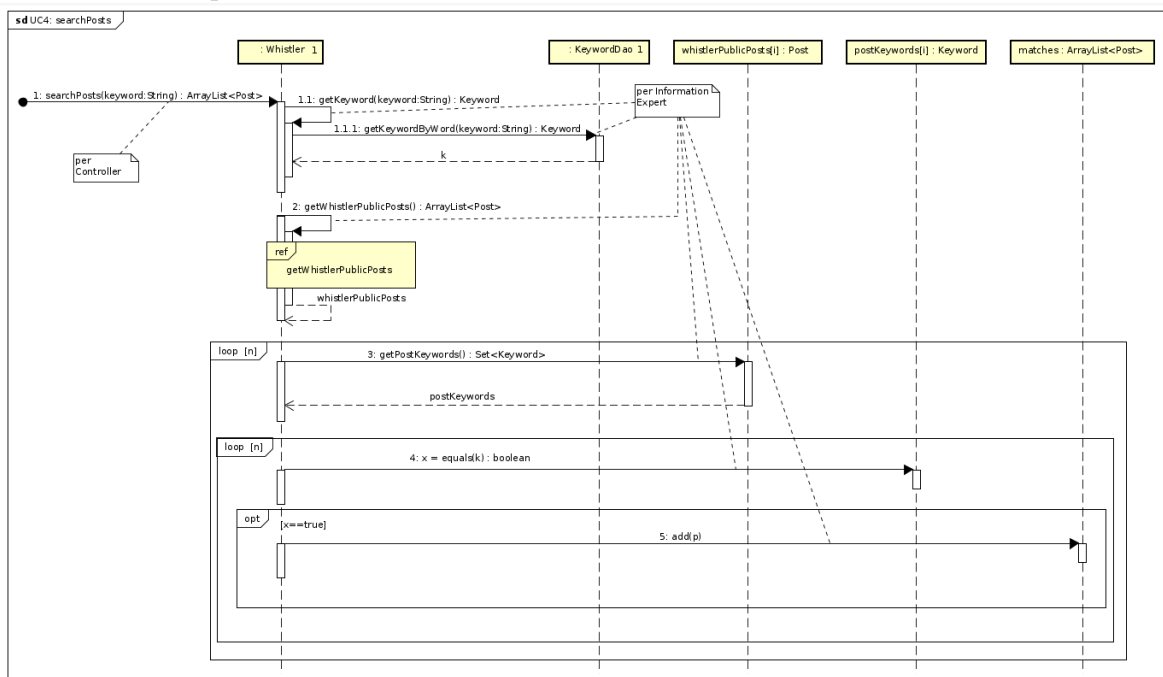
3 Progettazione

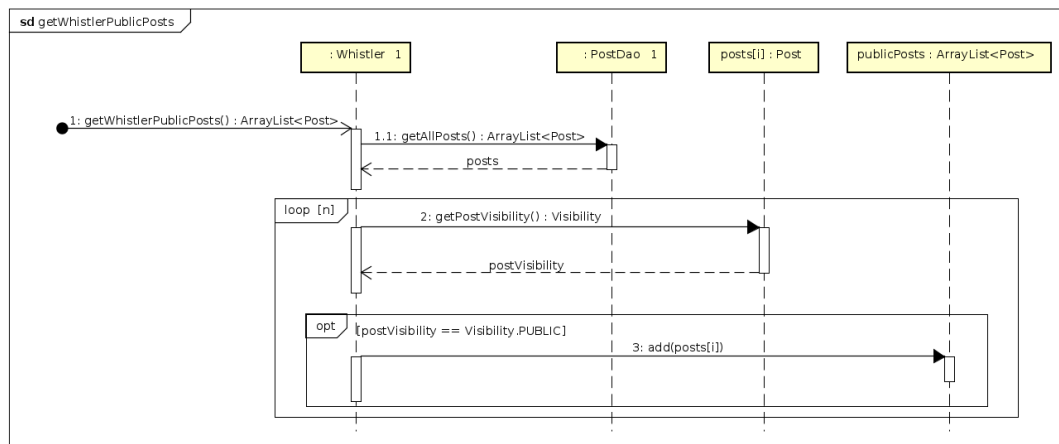
Dopo aver terminato l'analisi orientata ad oggetti per questa iterazione, sfruttando gli elaborati prodotti, si passa alla fase di progettazione. Di seguito si riportano i Diagrammi di Interazione (nello specifico **Diagrammi di Sequenza**) - modellazione dinamica - ed il **Diagramma delle classi** - modellazione statica, tra loro complementari.

Durante la stesura di quest'ultimi sono stati tenuti a mente ed applicati i vari principi di progettazione OO, quali i **patter GRASP** per l'assegnazione delle responsabilità ed i **design pattern Gang-of-Four (GoF)**.

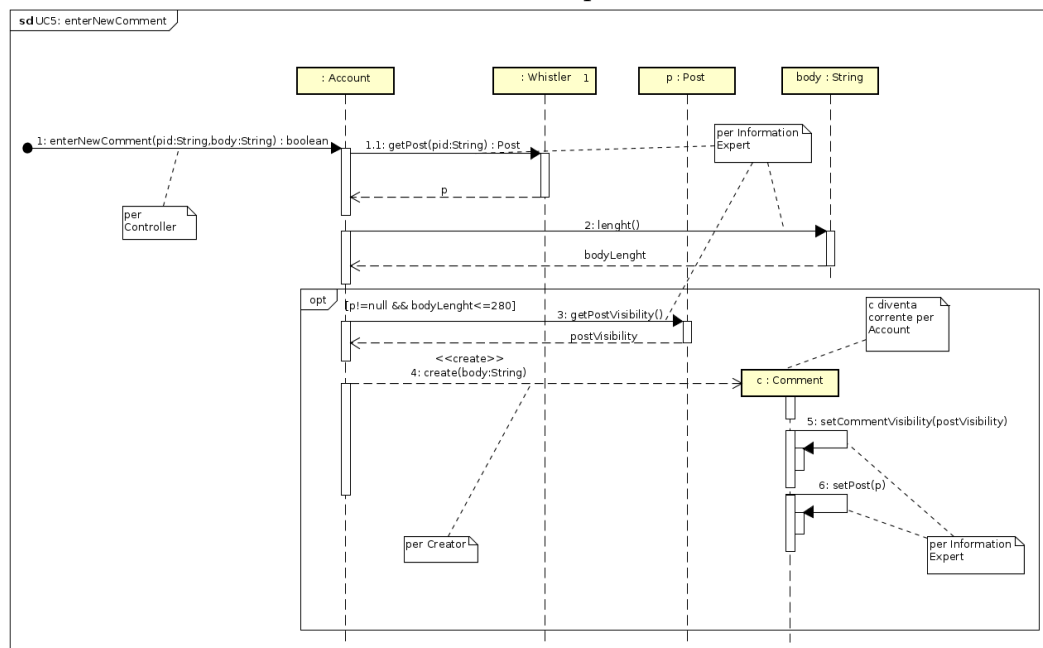
Diagrammi di Sequenza

- Ricerca di un post su Whistler

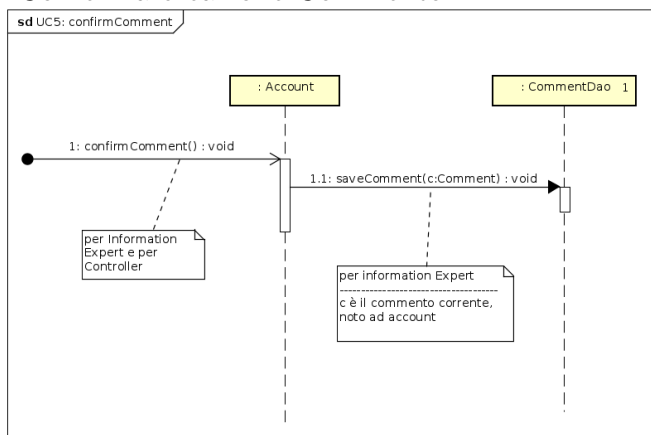




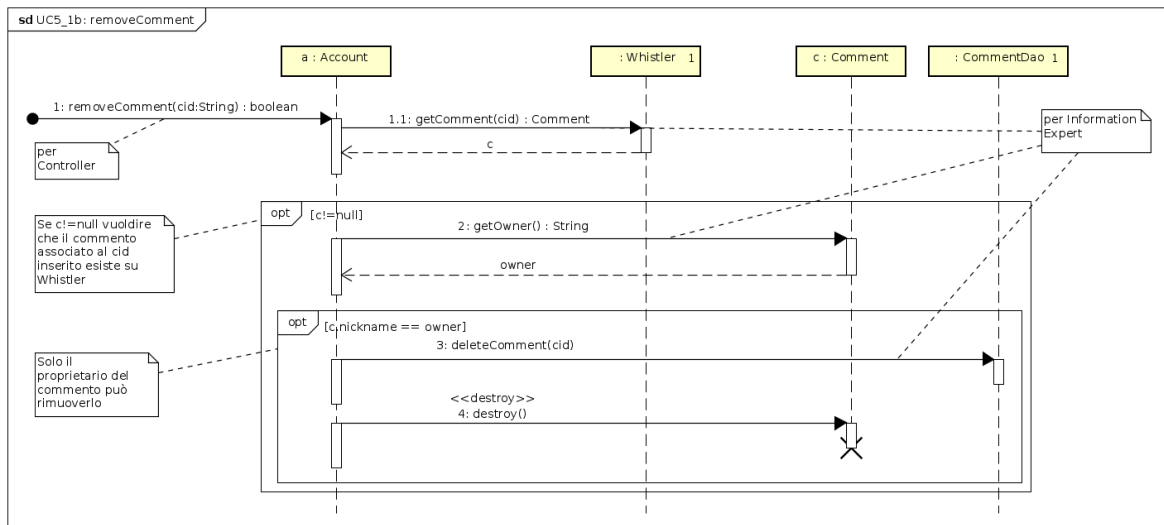
- Inserimento di un Commento su un Post presente su Whistler



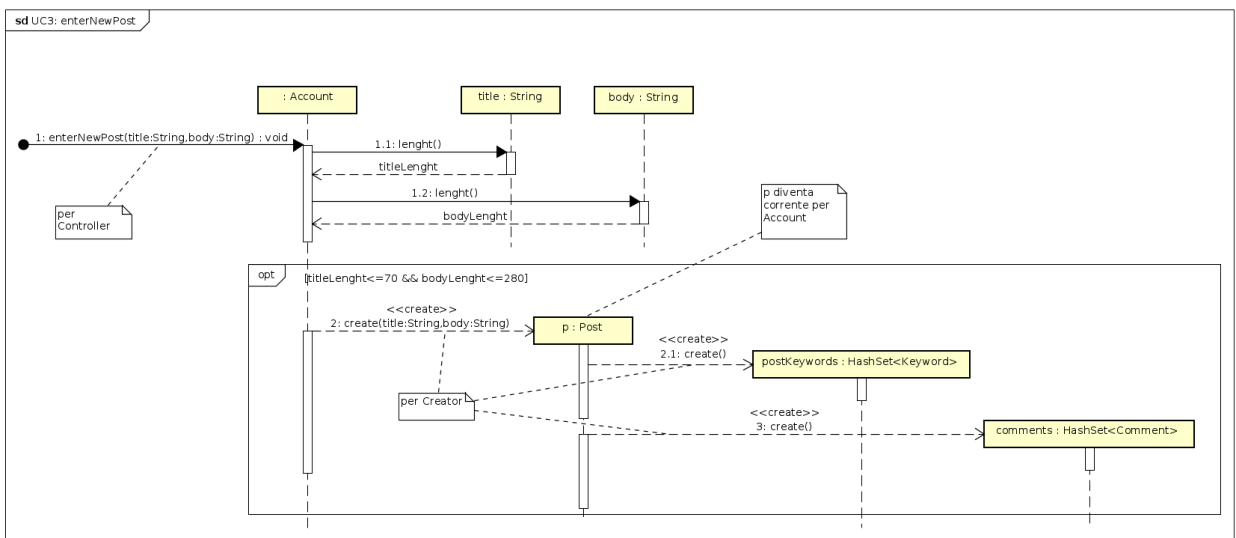
- Conferma creazione Commento



- Rimozione di un Commento



Si riporta di seguito la versione del Diagramma di Sequenza del caso d'uso **UC3 - enterNewPost** aggiornata all'iterazione corrente con il controllo sulla lunghezza del titolo e del corpo del post.



Diagrammi delle Classi

- DCD : Elaborazione 3

Si rimanda al file **Whistler_E3.asta** nella cartella 04_Elaborazione_3 per la consultazione del diagramma delle Classi relativo all'elaborazione 3.

Testing

Si riportano di seguito i test unitari che sono stati aggiunti durante l'elaborazione corrente.

Test Unitario

La strategia adottata consiste nel definire una classe di test per ciascuna classe da verificare. Questa classe di test contiene uno o più metodi di test per ciascun metodo pubblico della classe da verificare.

Classi di test:

Account Class

- 1) testEnterNewPost_TitleLenghtMoreThanSeventy (CF)(VE)
- 2) testEnterNewPost_TitleLenghtEqualSeventy (CF)(VE)
- 3) testEnterNewPost_TitleLenghtLessThanSeventy (CF)(VE)
- 4) testEnterNewPost_BodyLenghtMoreThanTwoHundredAndEighty (CF)(VE)
- 5) testEnterNewPost_BodyLenghtEqualTwoHundredAndEighty (CF)(VE)
- 6) testEnterNewPost_BodyLessThanTwoHundredAndEighty (CF)(VE)
- 7) testEnterNewComment_BodyLenghtMoreThanTwoHundredAndEighty (CF)(VE)
- 8) testEnterNewComment_BodyLenghtEqualTwoHundredAndEighty (CF)(VE)
- 9) testEnterNewComment_BodyLenghtLessThanTwoHundredAndEighty (CF)(VE)
- 10) testConfirmComment (CF)
- 11) testRemoveComment_owner (CE)(CF)
- 12) testRemoveComment_notOwner (CE)(CF)

E' stato inoltre aggiornato il test: *testRemovePost* in *testRemovePost_owner* ed è stato aggiunto *testRemovePost_notOwner*

- 11) testRemovePost_owner (CE)(CF)
- 12) testRemovePost_notOwner (CE)(CF)

Whistler Class

- 1) testSearchPosts (CF)
- 2) testGetKeyword_IsPresent() (CE)(CF)
- 3) testGetKeyword_NotPresent (CE)(CF)
- 4) testGetWhistlerPublicPosts (CF)

Sono stati individuati i casi di test tenendo a mente la definizione di **Black-box Testing** (Funzionale), ovvero la determinazione dei casi di test sulla base della specifica di ciascun componente, non tenendo conto della sua struttura interna. I **dataset** usati sono stati scelti tenendo a mente le seguenti tecniche:

- **tecnica di copertura delle classi di equivalenza (CE)**
- **tecnica di analisi dei valori estremi (VE)**
- **tecnica di copertura delle funzionalità (CF)**

Per ciascun metodo di test elencato in precedenza si è indicato l'acronimo della tecnica di riferimento adottata accanto al suo nome.

Test di Sistema

Si è scelto di eseguire dei test di sistema manuali, al fine di testare la struttura dell'interfaccia utente a caratteri - basata su console - provando a portare a termine i requisiti funzionali indicati nell'iterazione corrente.

A seguito dei **test di Unità e di Sistema** sono state apportate opportune modifiche per rendere più robusta la gestione degli input inseriti dall'utente, durante l'interazione con il Sistema, e per migliorare il comportamento dei metodi presi in esame.

Test di Regressione

A valle di ogni modifica sono stati eseguiti i test unitari e di sistema, per verificare di non aver introdotto ulteriori e nuovi difetti.