

Elaborazione - Iterazione 2

1 Introduzione

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

1. Gli utenti possono pubblicare dei **post** costituiti da un titolo, dal contenuto e da parole chiave.
2. L'utente, oltre a visualizzare nella propria **home** i post dei whistleblower seguiti, può visualizzare post pubblici di qualsiasi whistleblower nella bacheca.
3. Il **profilo** contiene i post pubblicati dall'utente e le sue informazioni personali, che possono essere modificate e delle quali è possibile definire la visibilità (pubblica e quindi visibile agli altri utenti o privata).
4. L'utente può visualizzare e gestire gli utenti presenti nella sua cerchia di interesse.
5. La piattaforma tiene traccia di tutte le operazioni effettuate registrando le azioni svolte dagli utenti e memorizzandole in un opportuno log.

I casi d'uso di riferimento sono **UC3: Gestisci Post**, **UC6: Visualizza Home** e **UC8: Visualizza Profilo**.

Nello specifico ci si concentra sulla:

- implementazione dello scenario principale di successo per ciò che riguarda il caso d'uso **UC3**. Inoltre si prendono in considerazione anche le estensioni **1a** ed **1b**. Nell'ordine: "inserimento", "modifica" e "rimozione" di un post. In questa iterazione verrà tralasciata l'estensione **5a** relativa alla verifica del numero di caratteri consentiti in un post.
- implementazione del caso d'uso **UC6**, ovvero la visualizzazione della home, contenente i post degli utenti presenti nella cerchia d'interesse dell'account dell'utente.
- implementazione dello scenario principale di successo e degli scenari alternativi del caso d'uso **UC8**, soffermandosi sull'estensione **1a** (visualizzazione del proprio profilo)

- registrazione delle operazioni effettuate nel sistema su un file log.

Inoltre, verranno implementati anche gli scenari tralasciati durante la prima iterazione, relativi ai casi d'uso **UC1: Gestisci Account** e **UC2: Segui Account** in modo da coprirli nella loro interezza.

- implementazione degli scenari alternativi del caso d'uso **UC1**, messi da parte nell'iterazione 1, ovvero la modifica (**1a**) e la rimozione (**1b**) dell'account.
- implementazione dello scenario alternativo, non ancora implementato, del caso d'uso **UC2** relativo alla gestione degli account seguiti presenti nella cerchia d'interesse dell'utente.

Si rimandano a successive iterazioni, l'implementazione della **bacheca** in cui l'utente può visualizzare i post pubblici dei whistleblowers non presenti nella sua cerchia d'interesse e l'invio di una notifica verso i follower dell'utente, alla pubblicazione di un nuovo post.

Sono stati riscritti i casi d'uso **UC6** e **UC8** 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.

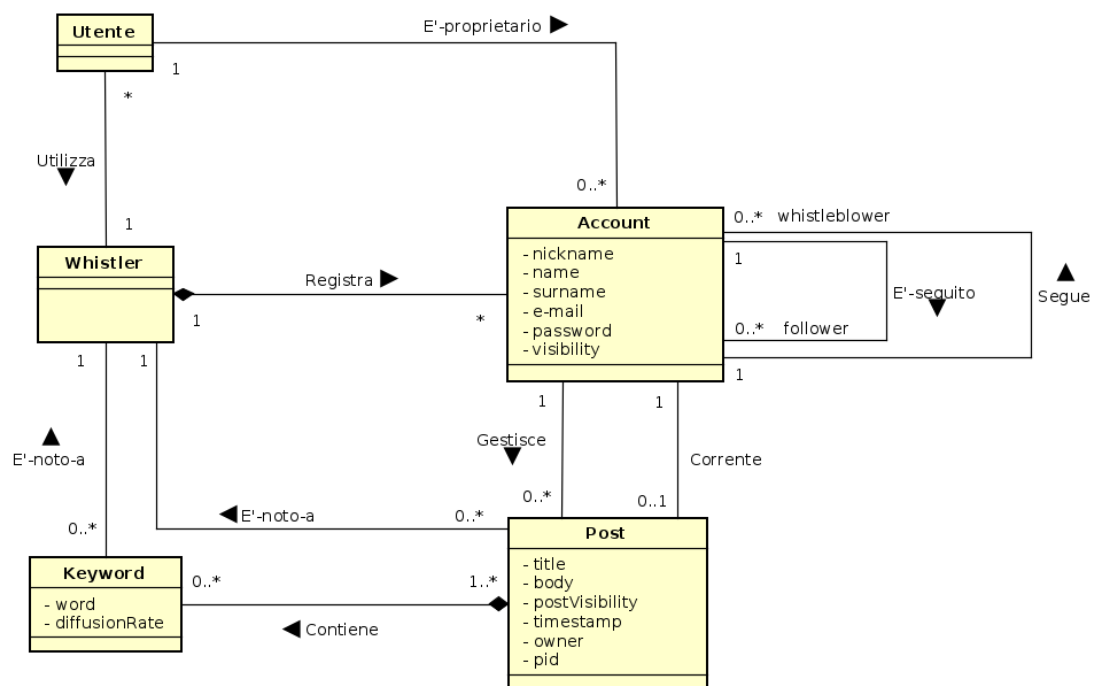
Modello di Dominio

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 **UC3** è possibile identificare le seguenti classi concettuali:

- **Post:** è il messaggio inserito all'interno della piattaforma di microblogging
- **Keyword:** è la parola che identifica al meglio il significato del post nel quale è inserita. E' costituita dalla parola in sé e dal tasso di diffusione (utile per individuare le keywords di tendenza).

Il modello di dominio ricavato è il seguente:



Prendendo in esame, invece, i casi d'uso **UC6** e **UC8** non emergono particolari classi concettuali. Sebbene la timeline possa sembrare un buon candidato, si sceglie di rilegare le dinamiche di aggiornamento della visualizzazione alla user interface, per cui il modello di dominio rimane invariato.

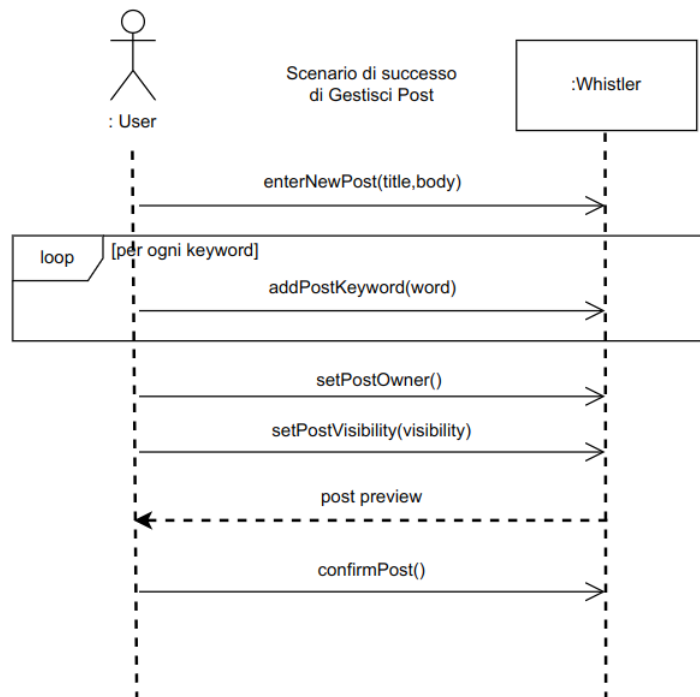
Anche le estensioni dei casi d'uso **UC1** e **UC2** non comportano variazioni nel modello di dominio fin qui ottenuto.

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 **UC3**:



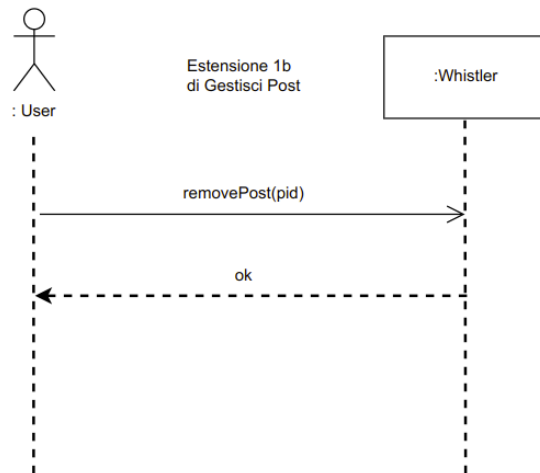
Durante l'inserimento di ciascun post: l'operazione d'inserimento del **timestamp** e della generazione del **pid** (post_id) vengono gestite internamente dal sistema.

Scenari alternativi 1a e 1b:

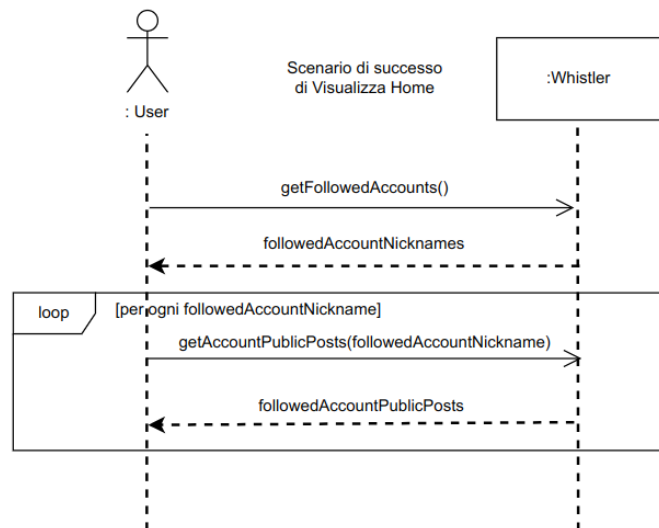
Affinché si possa modificare o eliminare un post è necessario che l'utente richiedente ne sia il proprietario, altrimenti non sarà possibile la modifica o la rimozione. Il controllo in questione viene eseguito direttamente all'interno del sistema.

Relativamente alle operazioni di modifica(**1a**), poiché queste, di volta in volta, si limitano a singole chiamate del tipo "setTitle", "setBody" ecc... in funzione di ciò che l'utente vuole modificare del post, si è deciso di omettere i relativi SSD.

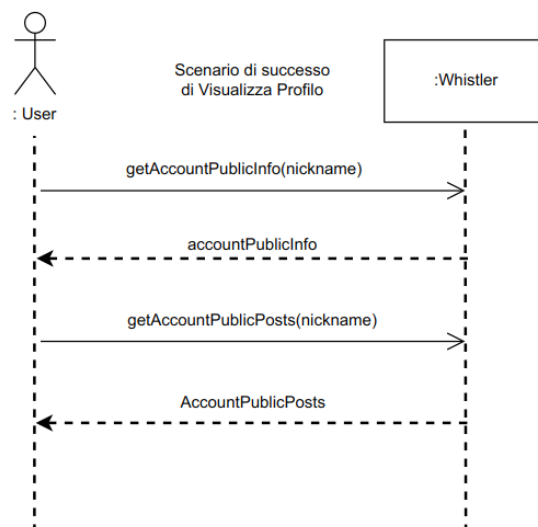
Si riporta di seguito, invece, l'SSD dell'estensione **1b** relativa alla rimozione di un post.



Per il caso d'uso **UC6**:

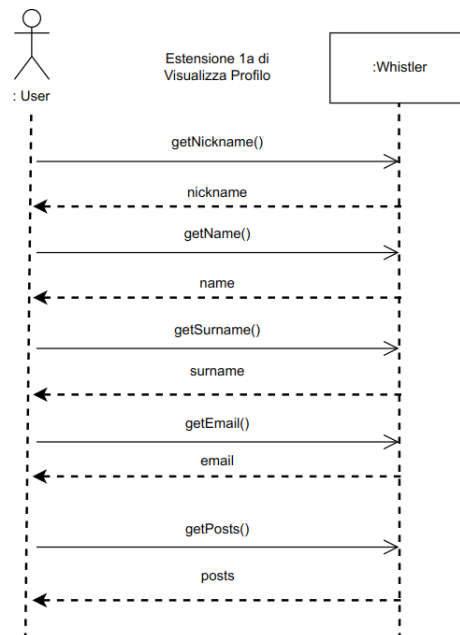


Per il caso d'uso **UC8**:



Tale caso d'uso è stato analizzato adottando come preconditione la presenza, nella piattaforma, dell'account associato al nickname in esame.

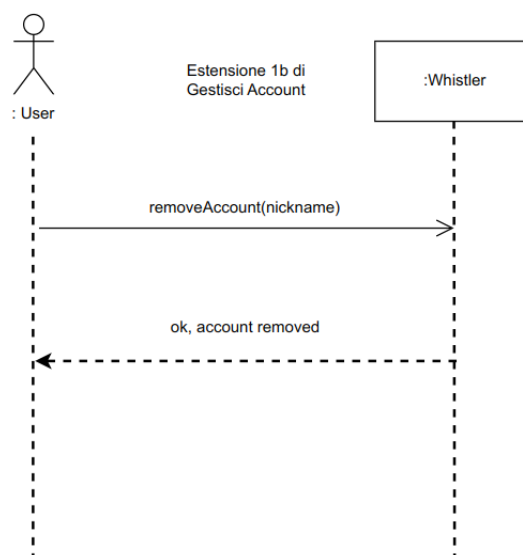
Relativamente all'estensione **1a**:



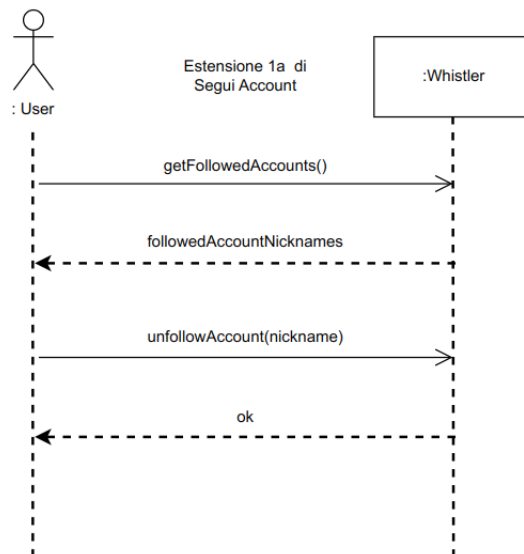
Si noti come l'SSD dell'estensione **1a** di **UC8** rimanga pressoché invariato nel risultato ottenuto rispetto allo scenario principale di successo. La differenza principale consiste nel ritornare tutte le informazioni ed i post (pubblici e privati) dell'account e non solo i post e le informazioni pubbliche.

Analizzando le estensioni dei casi d'uso **UC1** ed **UC2** si è ritenuto importante mostrare i seguenti SSD:

Estensione **UC1 1b**:



Estensione **UC2 1a**:



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: enterNewPost

Operazione : enterNewPost(title:String, body:String)

Riferimenti : caso d'uso: Gestisci Post

Pre-condizioni :

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

Post-condizioni :

- è stata creata una nuova istanza p di Post;
- gli attributi di p sono stati inizializzati;
- p è stato associato ad Account tramite l'associazione "Corrente"

Contratto CO2: addPostKeyword

Operazione : addPostKeyword(word:String)

Riferimenti : caso d'uso: Gestisci Post

Pre-condizioni :

- è in corso l'inserimento di un Post p

Post-condizioni :

- è stata creata una nuova istanza k di Keyword;
- gli attributi di k sono stati inizializzati;
- la keyword k è stata associata al Post corrente p tramite l'associazione "Contiene"

Contratto CO3: setPostVisibility**Operazione :** setPostVisibility(vis:Visibility)**Riferimenti :** caso d'uso: Gestisci Post**Pre-condizioni :**

- è in corso l'inserimento di un nuovo Post p

Post-condizioni :

- p.visibility è diventata vis.

Contratto CO4: confirmPost**Operazione :** confirmPost()**Riferimenti :** caso d'uso: Gestisci Post**Pre-condizioni :**

- è in corso l'inserimento di un nuovo Post p

Post-condizioni :

- è stata associata l'istanza p di Post corrente ad Account tramite l'associazione "Gestisce".
- l'istanza p di Post corrente e le Keywords ad esso associate sono note a Whistler.

Contratto CO5: removePost**Operazione :** removePost(pid:String)**Riferimenti :** caso d'uso: Gestisci Post**Pre-condizioni :**

- è presente un post p con pid corrispondente a quello in esame.
- l'account che si accinge a rimuovere il post è proprietario di quel post.

Post-condizioni :

- il diffusionRate delle keywords associate al post è stato ridotto;
- le associazioni di p con le keywords che conteneva sono state rimosse;
- l'associazione di p con Account è stata rimossa;
- p è stato rimosso.

Contratto CO6: removeAccount**Operazione :** removeAccount(nickname:String)**Riferimenti :** caso d'uso: Gestisci Account**Pre-condizioni :**

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

Post-condizioni :

- le associazioni del tipo "Gestisce" tra l'account a associato al nickname e ciascun post, di cui a è proprietario, sono state rimosse;
- i post di cui a è proprietario sono stati rimossi;
- le associazioni "Segue" ed "E'-seguito" tra l'account a ed altri account presenti nella piattaforma sono state rimosse;
- l'associazione "E'-proprietario" tra Utente ed Account è stata rimossa;
- l'account a è stato rimosso dalla piattaforma Whistler.

Contratto CO7: unfollowAccount**Operazione :** unfollowAccount(nickname:String)**Riferimenti :** caso d'uso: Segui Account**Pre-condizioni :**

- l'account b associato al nickname fornito è presente nella cerchia d'interesse dell'account a dell'Utente

Post-condizioni :

- l'associazione "Segue" tra l'account a e l'account b è stata rimossa;
- l'associazione "E'-seguito" tra l'account b e l'account a è stata rimossa.

3 Errata corrige

Durante la stesura del contratto **unfollowAccount - C07** dell'elaborazione corrente, mi sono reso conto di non avere opportunamente sottolineato nel contratto **followAccount** dell'elaborazione 1 la creazione dell'associazione "E'-seguito" tra l'account a e l'account b.

Si riporta di seguito la versione rivista e corretta:

Contratto CO2: followAccount - Elaborazione 1**Operazione :** followAccount(nickname:String)**Riferimenti :** caso d'uso: Segui Account**Pre-condizioni :**

- l'Utente possiede un account "a" ed è autenticato nella piattaforma;
- è presente almeno un altro account nella piattaforma oltre a quello dell'Utente;

Post-condizioni :

- l'account a è stato associato all'account b identificato dal nickname fornito tramite l'associazione "Segue";
- *l'account b è stato associato all'account a tramite l'associazione "E'-seguito".*

Questa dimenticanza si è propagata anche nella stesura del diagramma di sequenza **sd UC2: followAccount** - Elaborazione 1 e nel refactoring dell'Elaborazione 1, in cui si è aggiunto lo strato di persistenza. Per tale motivo nella sezione di progettazione verrà riportata anche la versione rivista e corretta del corrispettivo diagramma di sequenza.

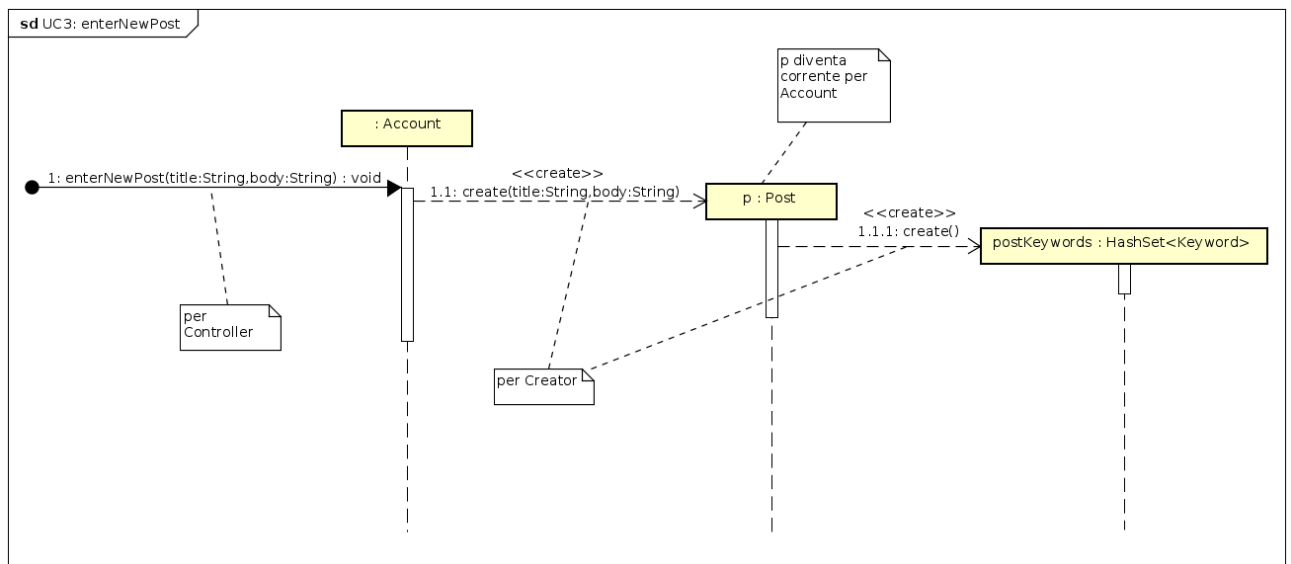
4 Progettazione

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.

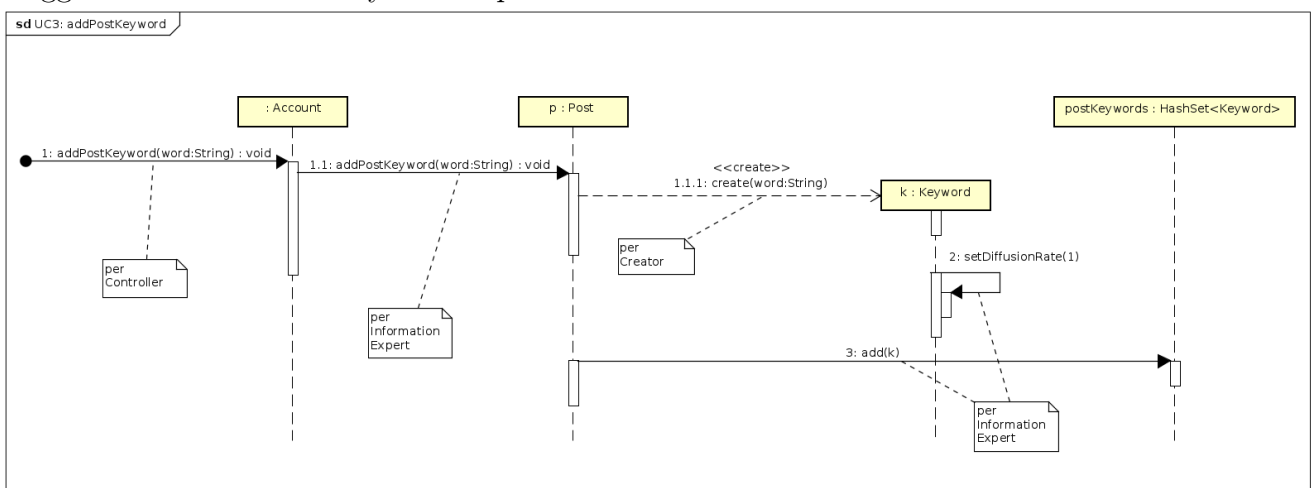
Durante la stesura di quest'ultimi sono stati tenuti a mente ed applicati i vari principi di progettazione OO, quali i **patter GRASP** per l'assegnazione delle responsabilità ed i **design pattern Gang-of-Four (GoF)**.

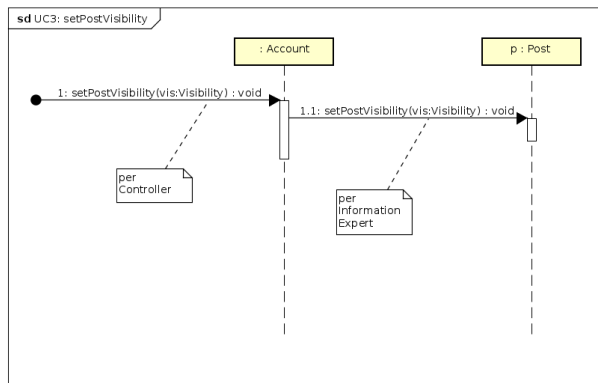
Diagrammi di Sequenza

- Creazione di un nuovo post su Whistler

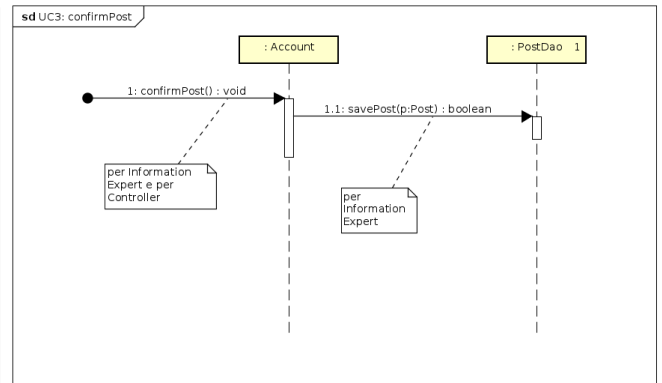


- Aggiunta di una nuova keyword al post corrente





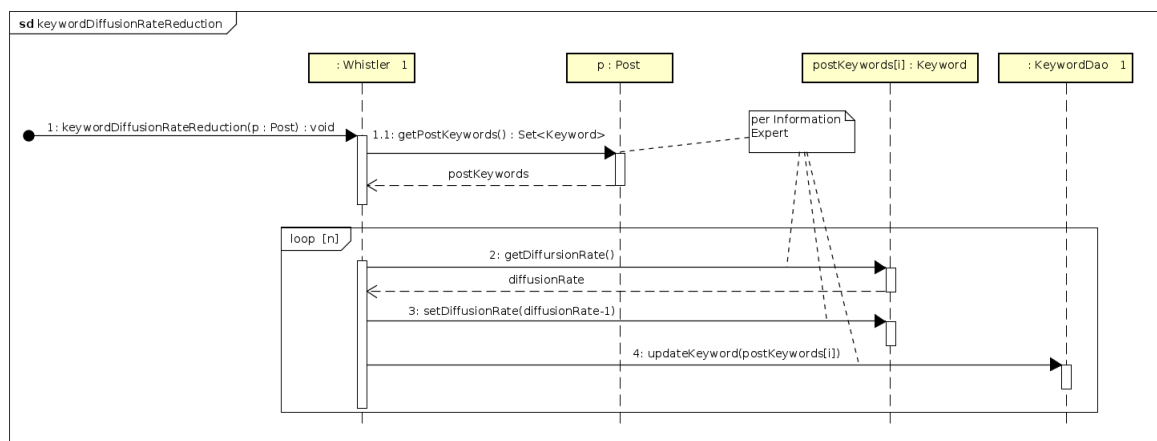
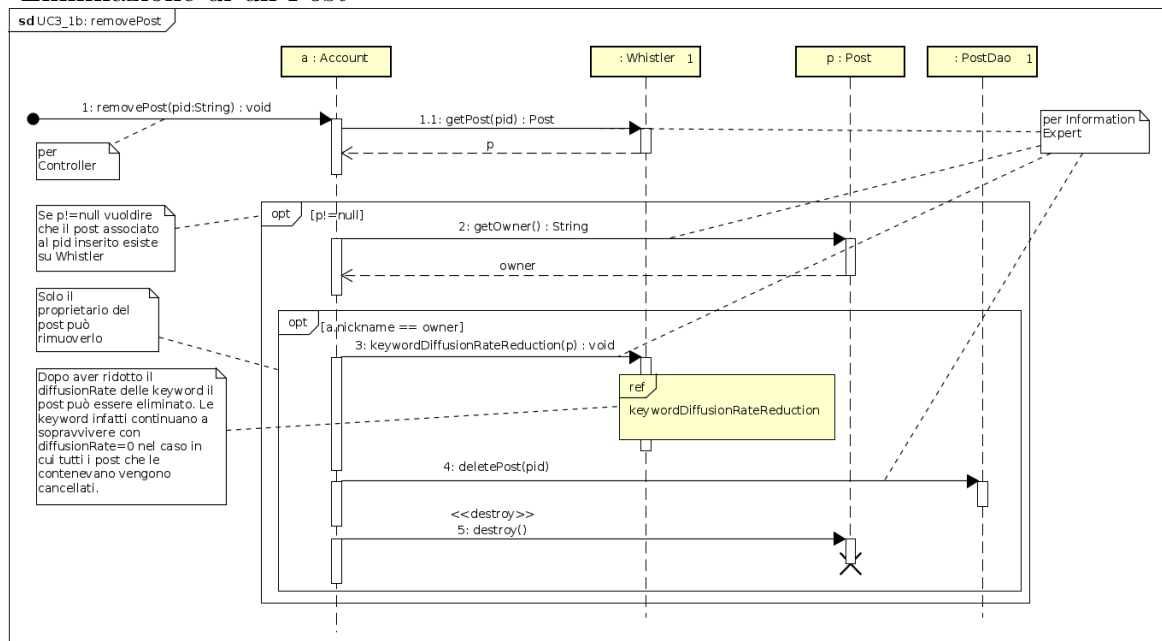
(a) Modifica della visibilità del post corrente



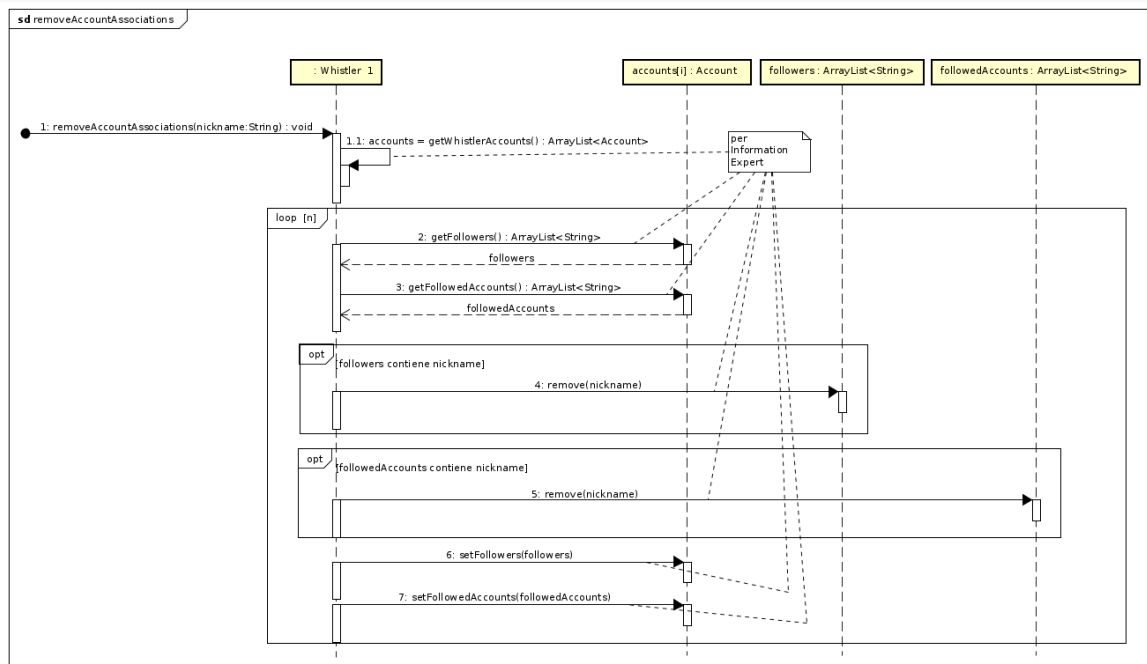
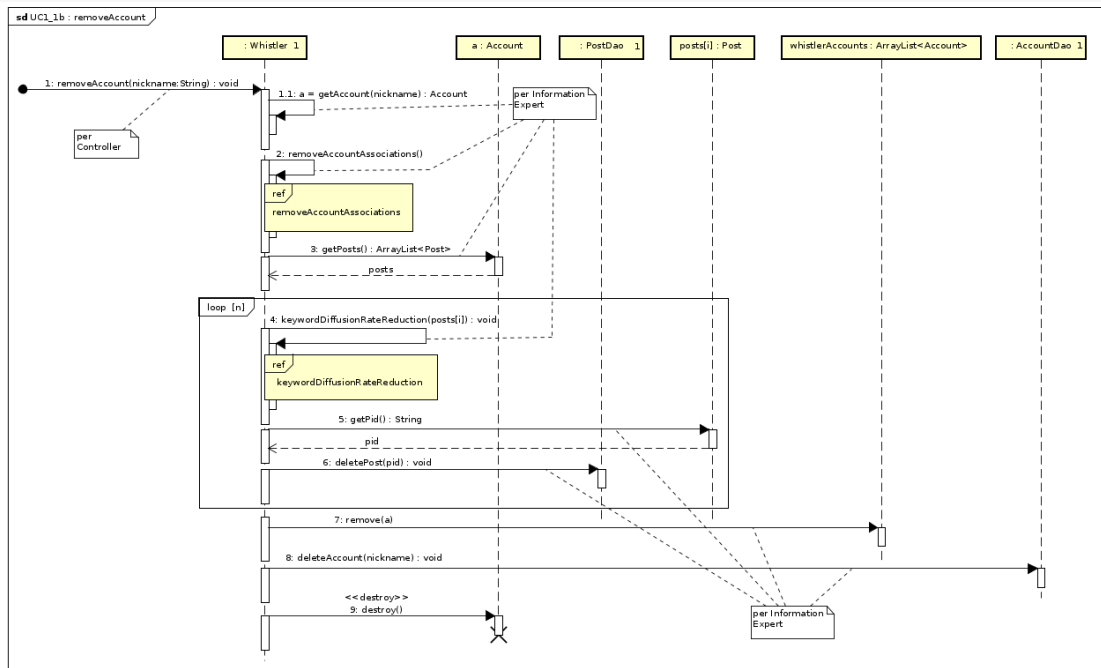
(b) Conferma creazione Post

Si è deciso di gestire la conferma salvando il post direttamente in modo persistente, e successivamente riempire **posts: ArrayList<Post>** con i post presenti sul Database.

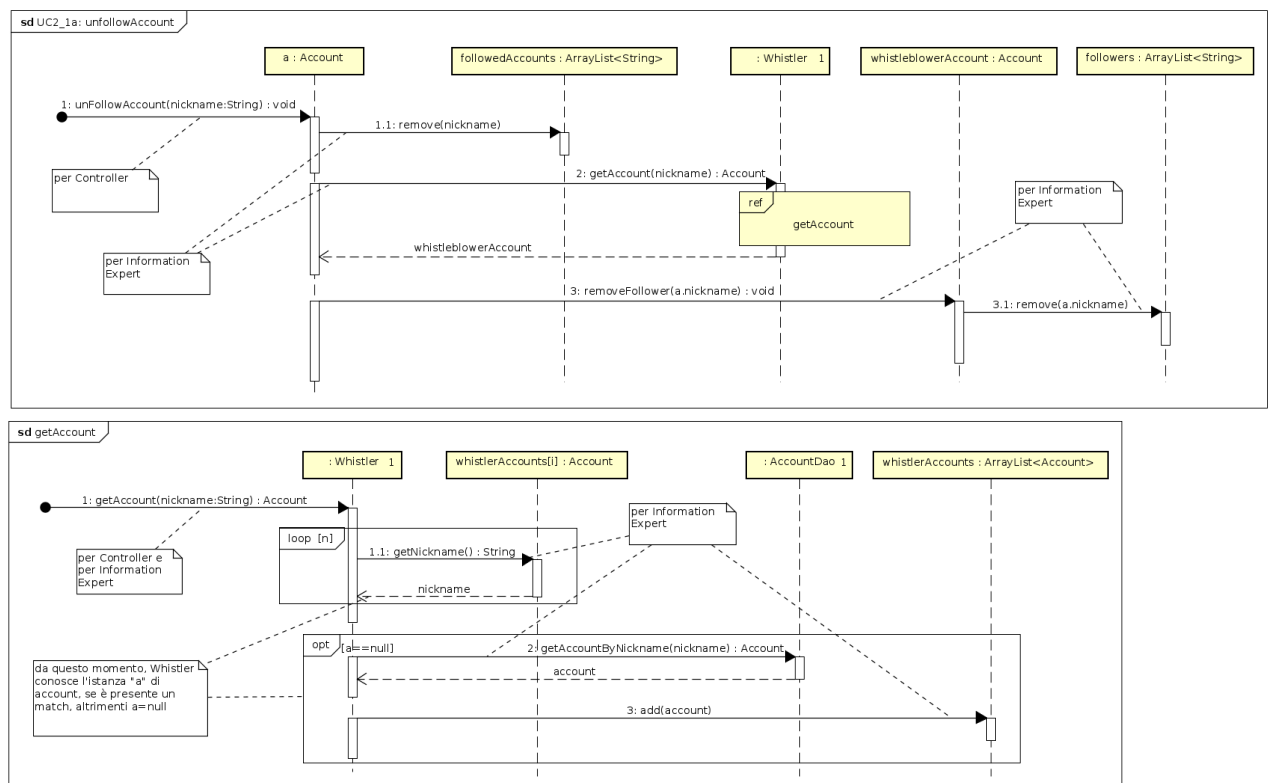
- Eliminazione di un Post



- Eliminazione dell'Account



- Rimozione dalla cerchia d'interesse di un Account

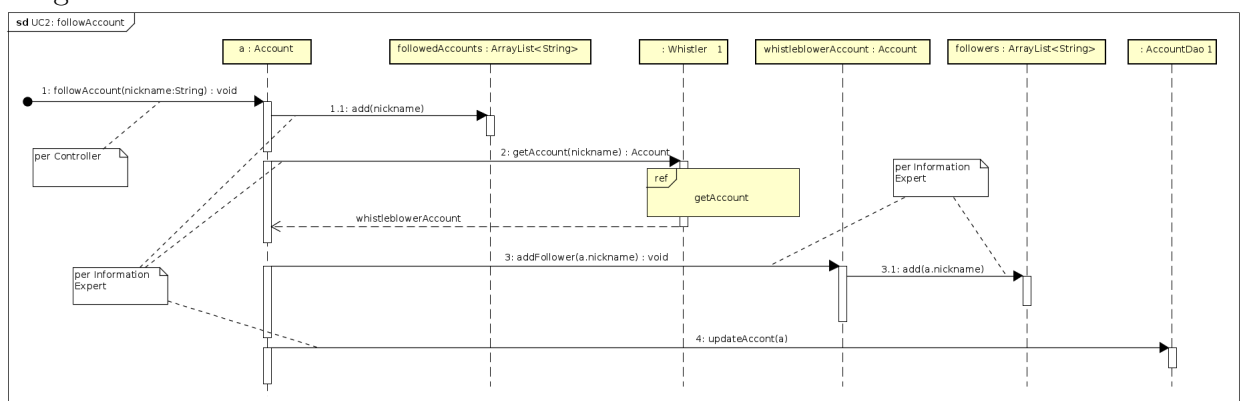


Errata Corrigere - Diagrammi di sequenza - Elaborazione 1 e Refactoring

Si riporta di seguito il diagramma di sequenza **sd UC2: followAccount** rivisto e corretto aggiungendo anche la gestione dell'associazione "E'-seguito" da parte del whistleblower che l'account dell'Utente decide di seguire.

Si noti che il Diagramma di Sequenza mostra l'implementazione attuale, con **followedAccount** e **follower** di tipo `ArrayList<String>` e non `ArrayList<Account>` come era stato pensato inizialmente nella prima iterazione.

- Seguire un account



Diagrammi delle Classi

- DCD : Elaborazione 2

Si rimanda al file **Whistler_E2.asta** nella cartella 03_Elaborazione_2 per la consultazione del diagramma delle Classi relativo all'elaborazione 2.

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) testUnFollowAccount (CF)
- 2) testConfirmPost (CF)
- 3) testRemovePost (CF)

Post Class

- 1) testAddPostKeyword (CF)

Whistler Class

- 1) testGetPost_IsPresent (CE)(CF)
- 2) testGetPost_NotPresent (CE)(CF)
- 3) testGetAccountPublicPosts (CF)
- 4) testGetAccountPublicInfo (CF)
- 5) testKeywordDiffusionRateReduction (CF)
- 6) testRemoveAccount (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.