

Elaborazione - Iterazione 1

1 Introduzione

Rivisti i documenti della fase di ideazione, sono state apportate delle modifiche nella numerazione dei casi d'uso precedentemente individuati, tenendo conto della maggior importanza di ciascuno dal punto di vista dell'architettura e del rischio. E' stato scritto, inoltre, il caso d'uso UC1 in formato dettagliato.

Lo scopo di questa prima iterazione di Elaborazione è quello di implementare i seguenti requisiti:

1. Ogni utente possiede un account identificato da un nickname ed associato ad un profilo.
2. E' possibile seguire gli account di altri whistleblower per visualizzare i post da loro pubblicati nella piattaforma, commentarli e mettere like.

I casi d'uso di riferimento sono **UC1: Gestisci Account** e **UC2: Segui Account**. Nello specifico ci si concentra sulla:

- implementazione dello scenario principale di successo per ciò che riguarda il caso d'uso UC1, tralasciando in questa iterazione, per semplicità, gli scenari alternativi di modifica e rimozione dell'account. Si rimanda, inoltre, l'implementazione del profilo ad iterazioni successive.
- implementazione dello scenario principale di successo per ciò che riguarda il caso d'uso UC2, tralasciando in questa iterazione, per semplicità, lo scenario alternativo di rimozione degli account seguiti dalla cerchia d'interesse. Si rimanda, inoltre, l'integrazione con la gestione dei post, della home e delle notifiche ad iterazioni successive nelle quali verranno presi in considerazione i rispettivi casi d'uso.

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.

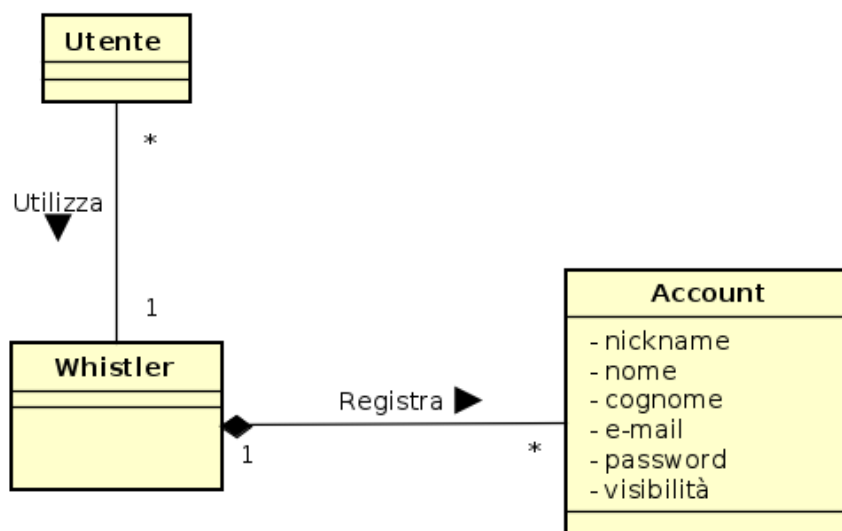
Modello di Dominio

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.

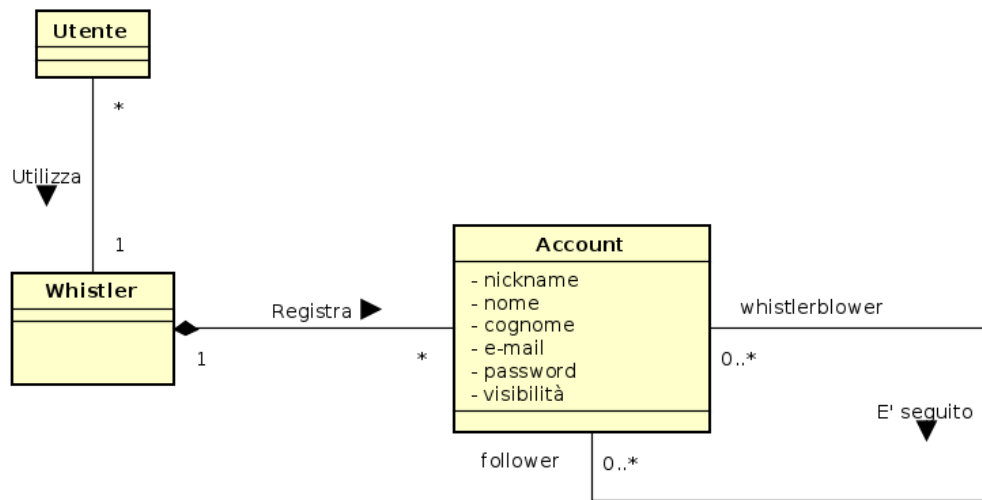
Prendendo in esame il caso d'uso **UC1** è possibile identificare le seguenti classi concettuali:

- **Utente:** è l'attore primario, rappresenta l'utilizzatore del Sistema
- **Whistler:** rappresenta il Sistema ovvero la piattaforma di microblogging Whistler
- **Account:** rappresenta l'identità dell'Utente all'interno della piattaforma di microblogging. Contiene tutte le informazioni personali, i post, le impostazioni ed i follower dell'Utente.

Il modello di dominio ricavato è il seguente:



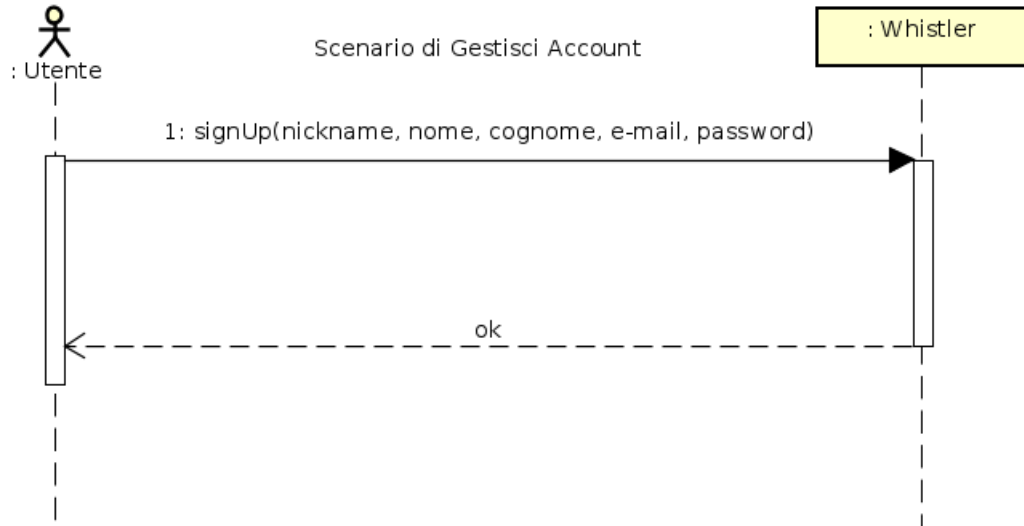
Prendendo in esame, invece, il caso d'uso **UC2** emerge l'**associazione riflessiva**, visibile nel modello di dominio aggiornato:



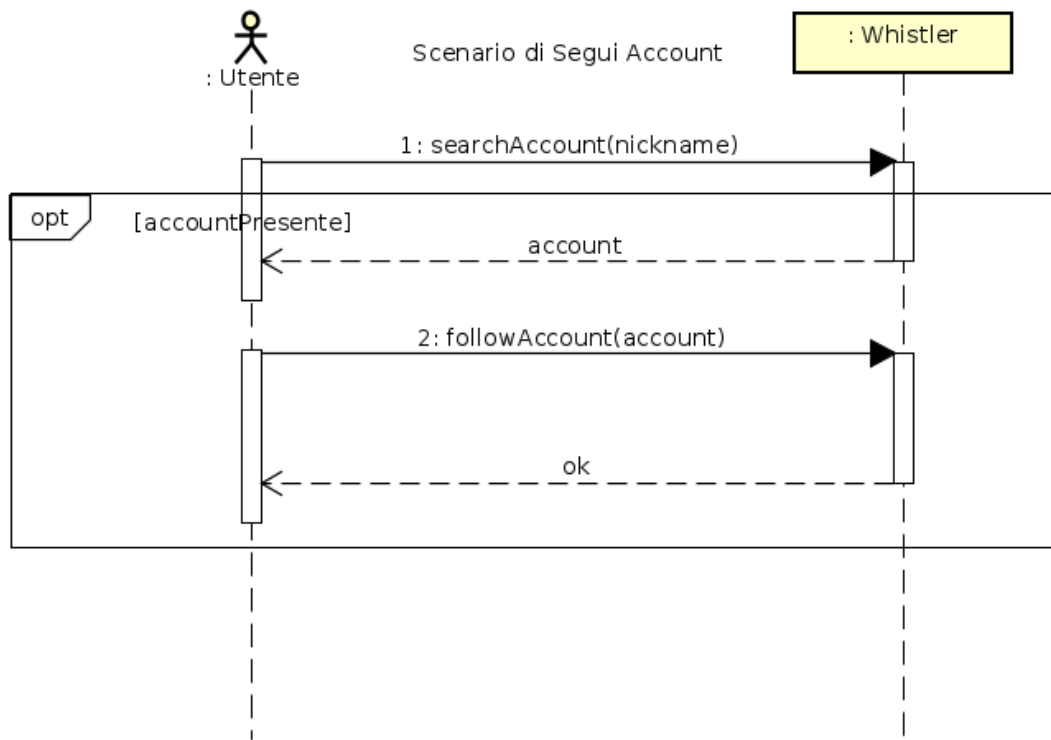
Tale associazione, grazie alla presenza dei **ruoli** associati alle molteplicità, evidenzia che un **Account** può essere seguito da 0 o più **Account** in qualità di Followers e che questo a sua volta può seguire 0 o più **Account** (**Whistlerblowers**). E' importante, inoltre, sottolineare il fatto che un **Account** possa essere seguito non solo da **Account** che esso stesso segue (quindi presenti nella sua cerchia d'interesse), ma anche da **Account** che non vengono seguiti a loro volta.

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). Analizzando lo scenario principale di successo del caso d'uso **UC1** si è ottenuto il seguente SSD:



mentre per ciò che concerne lo scenario principale di successo del caso d'uso **UC2** si è ottenuto il seguente SSD:



Contratti delle Operazioni

Di seguito si riportano i contratti delle operazioni di sistema 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: signUpAccount

Operazione : signUpAccount(a:Account)

Riferimenti : caso d'uso: Gestisci Account

Pre-condizioni : nessuna

Post-condizioni :

- l'account a è stato aggiunto nel Sistema

Contratto CO2: searchAccount

Operazione : searchAccount(nickname:String)

Riferimenti : caso d'uso: Segui Account

Pre-condizioni :

- l'Utente possiede un account ed è autenticato nella piattaforma
- sono presenti account di altri utenti nella piattaforma

Post-condizioni : nessuna (componente di interrogazione)

Contratto CO3: followAccount

Operazione : followAccount(a:Account)

Riferimenti : caso d'uso: Segui Account

Pre-condizioni :

- l'Utente possiede un account ed è autenticato nella piattaforma
- sono presenti account di altri utenti nella piattaforma

Post-condizioni :

- l'account a è stato aggiunto nella cerchia d'interesse dell'account richiedente

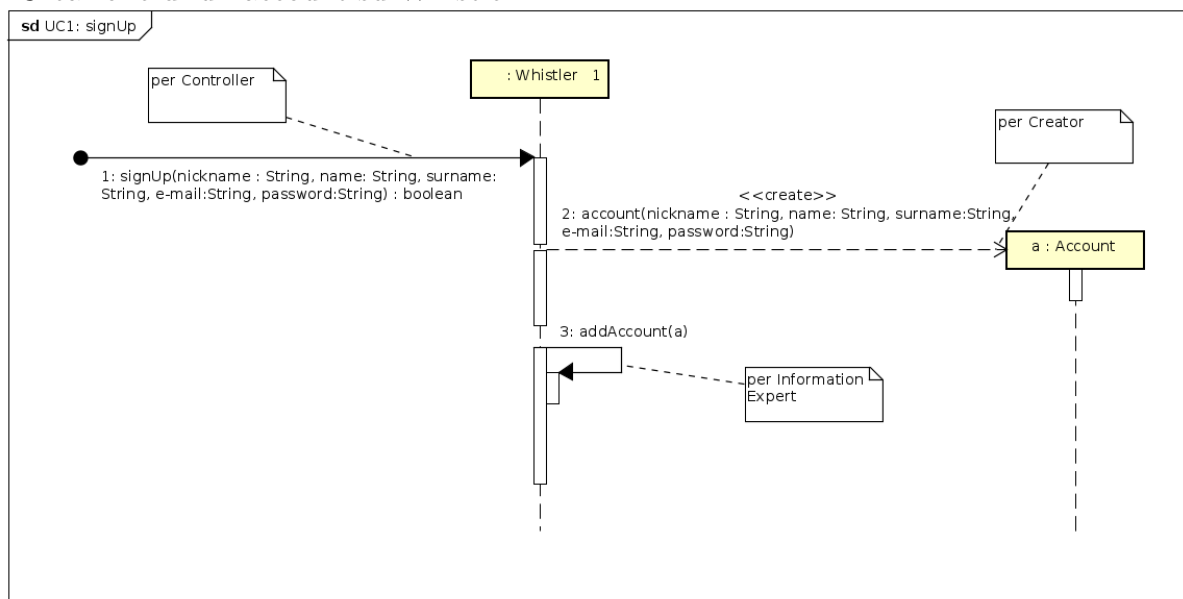
3 Progettazione

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.

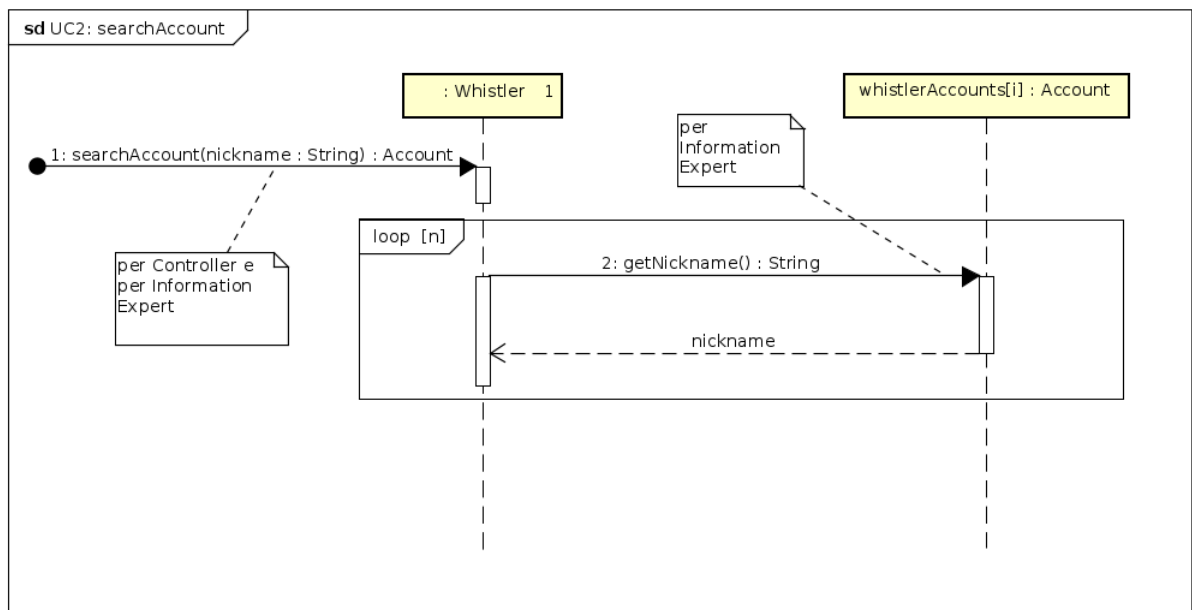
Durante la stesura di quest'ultimi sono stati tenuti a mente ed applicati i vari principi di progettazione OO, quali i **patter GRASP** per l'assegnazione delle responsabilità ed i **design pattern Gang-of-Four (GoF)**.

Diagrammi di Sequenza

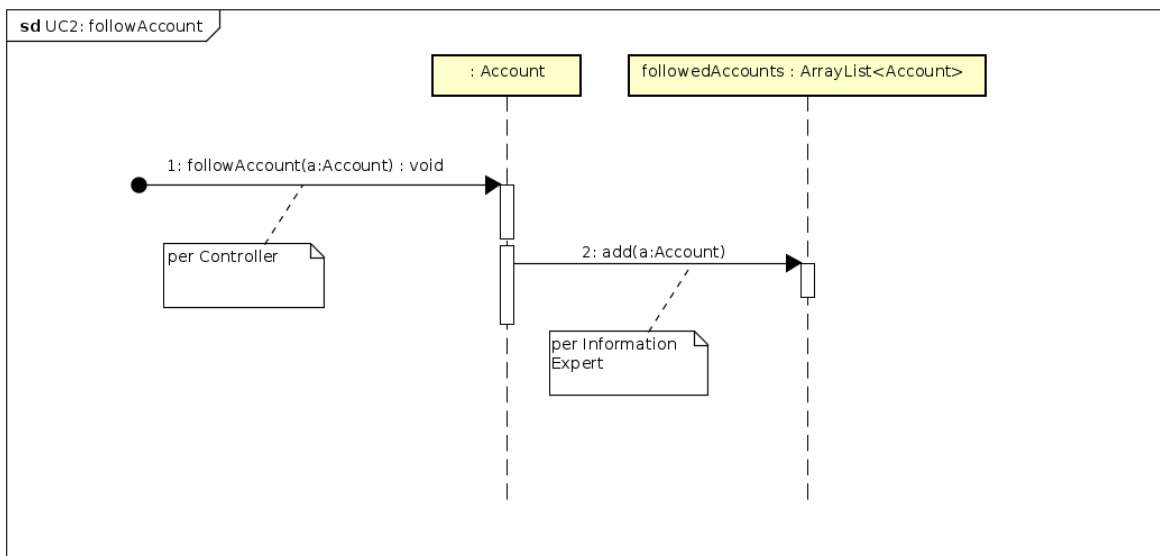
- Creazione di un account su Whistler



- Ricerca di un account

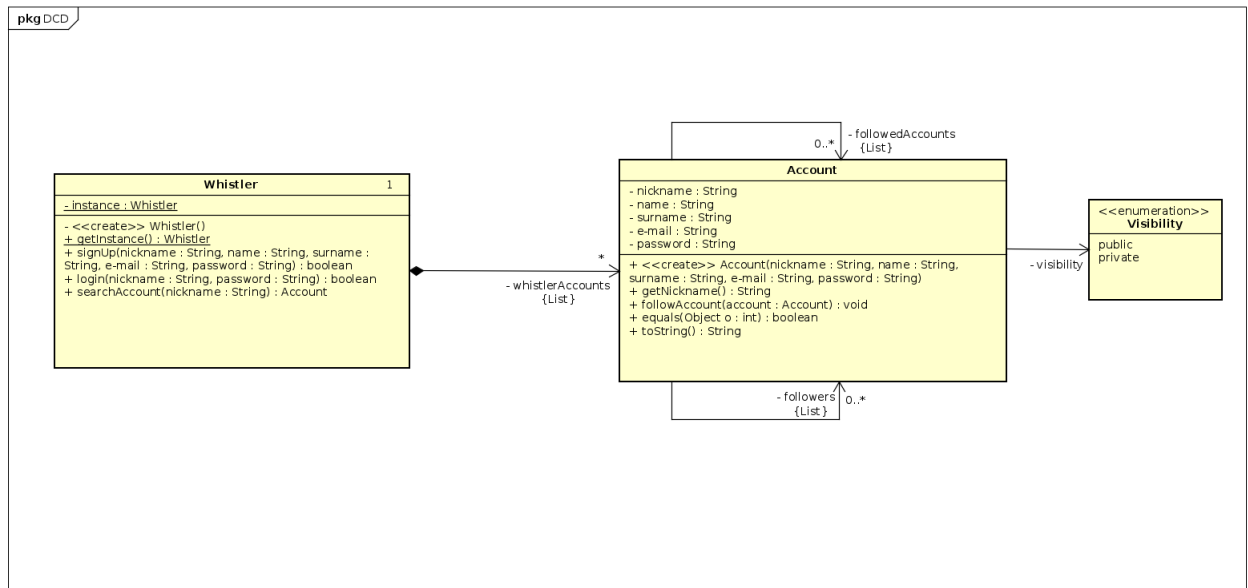


- Seguire un account



Diagrammi delle Classi

- DCD - Elaborazione 1



E' stato utilizzato il **Pattern Singleton (GoF)** sull'oggetto Whistler perché è necessario che esista un'unica istanza della piattaforma di microblogging.

Testing

Lo sviluppo guidato dai test (**TDD** - Test-Driven Development) è un'ottima pratica applicabile su UP. In generale il TDD prevede l'utilizzo di diversi tipi di test (**test unitari**, **test di Integrazione**, **test di Sistema**, **test di Accettazione**, **test di Regression**).

I **test unitari** permettono di verificare il funzionamento delle piccole parti ("unità") del sistema, senza però verificare il sistema nel suo complesso, che vengono verificate invece, per mezzo dei **test di Sistema**. Esempi di unità sono: una classe oppure un metodo.

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:

Whistler Class

- **SignUp Tests**

- 1) testSignUp_ValidNicknameAndPassword (**CE**), (**VE**),(**CF**)
- 2) testSignUp_NicknameAlreadyExists (**CF**)
- 3) testSignUp_NicknameWithSpaces (**CE**)
- 4) testSignUp_PasswordLengthEqualEight (**VE**)
- 5) testSignUp_PasswordLengthLessThanEight (**VE**)

- **Login Tests**

- 1) testLogin_InvalidNicknameAndPassword (**CF**)
- 2) testLogin_InvalidNickname (**CF**)
- 3) testLogin_InvalidPasswordLength (**VE**), (**CF**)
- 4) testLogin_ValidNicknameAndPasswordLengthButNotValidPassword (**CF**),(**VE**)
- 5) testLogin_ValidNicknameAndPassword (**CF**), (**VE**)

- **SearchAccount Tests**

- 1) testSearchAccount_IsPresent (**CE**), (**CF**)
- 2) testSearchAccount_NotPresent (**CE**), (**CF**)

Account Class

- 1) testFollowAccount_TwoAccountInWhistler (**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** scelti sono basati sulla:

- **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 al fine di rendere più robusta la gestione degli input inseriti dall'utente, durante l'interazione con il Sistema, e di migliorare il comportamento dei metodi presi in esame. In particolare sono stati gestiti i seguenti casi:

- (Login) L'inserimento di nickname non esistenti nel sistema
- (Login) L'inserimento di password errata
- (SignUp) L'inserimento di un nickname già esistente
- (Signup/Login) L'inserimento di nickname contenenti spazi
- (SignUp/Login) L'inserimento di una password contenente un numero di caratteri inferiore ad 8
- (In tutte le console) L'inserimento di comandi non validi o fuori range

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.