

# Progetto di Ingegneria del Software 2

## SWIMv2



## Requirements Analysis Specification Document

A.A. 2012/2013

**Autori:**

Emanuele Uliana (799256), Gabriele Rufolo (743695), Walter Rubino (742519)

**Docente:**

Prof.ssa Raffaella Mirandola

Versione 1.1 del 12/12/2012

## Indice

<b>1</b>	<b>Descrizione</b>	<b>3</b>
<b>2</b>	<b>Introduzione</b>	<b>4</b>
2.1	Glossario . . . . .	4
2.2	Obiettivi . . . . .	4
<b>3</b>	<b>La piattaforma SWIMv2</b>	<b>5</b>
3.1	Identificazione degli attori . . . . .	5
<b>4</b>	<b>Considerazioni preliminari</b>	<b>6</b>
<b>5</b>	<b>Requisiti</b>	<b>7</b>
5.1	Funzionali . . . . .	7
5.2	Non funzionali . . . . .	8
5.2.1	Interfacce e usabilità . . . . .	8
5.2.2	Struttura del sistema . . . . .	8
5.2.3	Hardware . . . . .	8
5.2.4	Performance . . . . .	8
5.2.5	Sicurezza . . . . .	9
5.2.6	Documentazione . . . . .	9
<b>6</b>	<b>Identificazione degli scenari e casi d'uso</b>	<b>10</b>
6.1	Scenari . . . . .	10
6.2	Casi d'uso . . . . .	13
<b>7</b>	<b>Modelli UML</b>	<b>19</b>
7.1	Sequence diagrams . . . . .	19
7.2	Statechart . . . . .	26
<b>8</b>	<b>Modello Alloy</b>	<b>28</b>
8.1	Signatures . . . . .	28
8.2	Facts . . . . .	28
<b>9</b>	<b>Strumenti utilizzati</b>	<b>29</b>

# 1 Descrizione

Si vuol sviluppare un social network che consenta agli utenti di aiutarsi reciprocamente inviando richieste di supporto e/o fornendo egli stesso aiuto.

L'utente non registrato (ospite) ha la sola facoltà di lettura dei quesiti, delle risposte e delle soluzioni ma senza possibilità di interagire con alcuno dei soggetti interessati.

Il sistema deve permettere la creazione di nuovi utenti tramite l'immissione di credenziali di accesso quali l'email e la password. Una volta registrato l'utente potrà non solo leggere le risposte ai problemi già noti (pubblicati da terzi) ma anche contribuire sia alla soluzione del problema sia ponendo nuove domande.

Il sistema implementa anche un sistema di gestione dei feedback con il quale è possibile valutare l'operato di chi ha fornito aiuto creando così un forte ed efficiente sistema di collaborazione.

I quesiti possono essere posti sia in privato tramite l'utilizzo di messaggi privati tra due utenti che immissione di domande nella bacheca pubblica e quindi visibile da tutti gli internauti (anche ai non registrati, vedi sopra).

## 2 Introduzione

### 2.1 Glossario

Al fine di migliorare la comprensione del seguente documento si riportano qui sotto alcune tra le principali definizioni dei termini usati all'interno dello stesso:

- **Abilità:** è un'abilità posseduta dall'utente e può riferirsi ad esempio a conoscenze in ambito culinario, sportivo, professionale, accademico, scolastico, hobbistico etc.
- **Account:** vedi *profilo*
- **Argomento:** vedi *thread*
- **Bacheca:** con bacheca intendiamo l'insieme totale dei messaggi pubblici presenti sulla piattaforma SWIMv2
- **Messaggio privato:** è uno strumento che consente agli utenti di inviarsi messaggi tra loro privatamente per chiedere aiuto e/o rispondere a richieste di aiuto ricevute
- **Feedback:** è la possibilità di esprimere un giudizio sull'operato di un utente
- **Post:** è il singolo messaggio scritto da un utente, esso può contenere un domanda, un messaggio di risposta o anche una soluzione
- **Profilo:** è l'insieme delle informazioni presenti nel sistema relative ad un determinato utente. Comprende il nome e cognome, un indirizzo email per effettuare l'accesso e per essere contattato, un feedback assegnatogli dagli utenti del sistema, un'immagine di profilo
- **Thread:** contiene il quesito posto dall'utente in bacheca, con i post di risposta e la soluzione (se esiste)

### 2.2 Obiettivi

La piattaforma sviluppata ha per obiettivi:

- Fornire una piattaforma di supporto agli utenti
- Valutare l'operato degli utenti con il sistema del feedback
- Condividere conoscenze e quindi creare collaborazione tra gli internauti
- Garantire l'iscrizione da parte di nuovi utenti
- Possibilità di immettere nel sistema nuove abilità
- Consentire agli utenti non registrati l'accesso al sistema in sola lettura
- Favorire la collaborazione tra utenti (con suggerimenti di amicizie)

## 3 La piattaforma SWIMv2

Il nuovo sistema realizzato consiste in una piattaforma web accessibile dai vari utenti via Web che offrirà servizi mirati alle varie tipologie di utenti collegati.

I membri del sistema hanno facoltà di accedere alla pagina del proprio profilo, aggiornare i propri dati, leggere i messaggi privati, rispondere alle richieste di supporto, scrivere in bacheca, valutare gli altri utenti con un feedback. Possono anche, in base al gruppo di appartenenza, visionare le richieste d'inserimento di nuove abilità e poter decidere se accettarle o meno.

Il sistema mette inoltre a disposizione un meccanismo di friends suggestion con il quale i membri possono suggerire nuove amicizie ai propri contatti.

### 3.1 Identificazione degli attori

Dopo un'accurata analisi del problema si è provveduto a determinare i protagonisti principali del sistema e le relative funzionalità, riportate qui di seguito:

1. Ospite: Non ha alcun potere di interagire attivamente con il sistema, può solamente cercare nella bacheca pubblica i quesiti (con relative risposte) condivise da altri utenti con lo stesso problema o quantomeno simile.
2. Utente registrato: Usa il sistema per inviare nuove richieste di supporto e per interagire con gli altri utenti. Può formulare domande sia sulla bacheca pubblica che inviare messaggi privati agli altri utenti. Ha facoltà di rispondere alle domande poste dagli altri utenti, può accettare nuove richieste di amicizia, inviarne di nuove e suggerire nuove abilità all'amministratore di sistema. Ha facoltà di valutare gli altri utenti tramite un sistema di feedback.
3. Amministratore: Ha il controllo sul sistema. Può aggiungere o rimuovere abilità, accettare o rifiutare richieste di nuove funzionalità.

## 4 Considerazioni preliminari

Nella specifica in nostro possesso sono state rilevate alcune lacune e imprecisioni riguardo le funzionalità del sistema. Qui di seguito esporremo le assunzioni da noi fatte per ovviare a tali ambiguità.

- *Interazioni tra utenti*: abbiamo assunto che tale attività potesse avvenire tramite una bacheca pubblica o tramite messaggi privati.
- *Suddivisione dei messaggi*: ogni singolo messaggio (post) ha un suo contenitore padre (thread) il quale identifica l'argomento trattato. Il thread può essere stato pubblicato o sulla bacheca pubblica o sulla casella messaggi privati di un utente.
- *Richieste di amicizia*: uno tra gli aspetti sicuramente più ambigui e probabilmente errati presenti nella specifica è certamente quello concernente la gestione delle amicizie; l'esistenza di un'amicizia di classe "A" e una "B" risulta non solo di dubbia definizione ma anche totalmente inutile.
- *Coerenza di dominio*: Si precisa infine che non è possibile suggerire a un proprio amico di stringere amicizia con un contatto non presente nella propria cerchia di contatti.
- *Ricerca degli utenti*: si può effettuare la ricerca all'interno della piattaforma per abilità possedute dai membri e/o per nickname.
- *Ricerca degli argomenti*: si possono individuare i thread tramite una ricerca basata sulla categoria appartenente o direttamente tramite il titolo.
- *Definizione delle abilità*: il sistema è dotato di un insieme di abilità predefinite che può essere arricchito grazie ai suggerimenti dei membri tramite un'esatta procedura.
- *Gestione delle categorie*: gli amministratori possono creare, rinominare, eliminare categorie; Le categorie hanno nomi quali "Informatica", "Motori", "Hobbistica" e così via.
- *Stato dei thread*: i messaggi presenti in bacheca hanno uno stato che può essere "Attivo", "Risolto", "Invalido" ed una relativa icona, ogni amministratore può cambiare lo stato in qualsiasi momento. Un thread può inoltre risultare "Aperto" o "Chiuso" cioè vi è la possibilità di continuare una discussione o meno.

## 5 Requisiti

### 5.1 Funzionali

Vengono riportate qui di seguito alcune specifiche che consentono il mantenimento dei requisiti da parte della piattaforma sviluppata:

- Gli amministratori hanno l'abilità di aggiungere nuove abilità tra quelle predefinite, approvare richieste (ricevute dagli utenti) di inserimento di nuove abilità
- Le abilità inserite da un amministratore nel sistema saranno visibili a tutti gli utenti che potranno decidere se abilitarle o meno sul loro profilo
- Gli amministratori hanno un superset di abilità rispetto a quello posseduto dagli utenti registrati. Essi infatti possono svolgere tutte le azioni che svolgono quest'ultimi: rispondere a richieste di aiuto, inviare e ricevere feedback etc.
- Ogni utente registrato può richiedere aiuto pubblicamente (sulla bacheca) o privatamente (via messaggio privato)
- Un utente può suggerire ad un suo amico di stringere amicizia con un terzo
- L'utente possono definire il proprio set di abilità scegliendo tra quelli predefiniti
- Ogni utente può proporre nuove abilità da inserire anche qualora non siano già presenti tra quelle predefinite dal sistema
- L'utente può pubblicare un feedback sull'aiuto ricevuto
- Ogni utente può inviare o rispondere a richieste di amicizia
- Ogni ospite può cercare risposte ai suoi problemi senza però poter scrivere
- Ogni problema ha un autore, un insieme di risposte e di soluzioni proposte e un stato che può essere Attivo, Invalido, Risolto
- Ogni profilo contiene: il nome dell'utente, una casella di messaggi privati, un indicatore di punteggio relativo al feedback ricevuto
- Ogni utente può cancellare il suo profilo in qualsiasi momento ma i messaggi pubblici, le relative risposte e le soluzioni non verranno rimossi restando così visibili

## 5.2 *Non funzionali*

### 5.2.1 Interfacce e usabilità

Il sistema sarà in tutto e per tutto web-oriented consentendo la massima facilità d'uso possibile e rispettando i più recenti standard web quali HTML4.0, XHTML1.1, CSS2/3, ECMAScript permettendo inoltre sia la massima compatibilità con i browser e i sistemi operativi ad oggi più comuni sia il miglior risultato in termini di affidabilità e robustezza del prodotto stesso.

### 5.2.2 Struttura del sistema

Il sistema userà la tecnologia J2EE per la presentazione e la gestione delle web pages; mentre verrà fatto uso del linguaggio MySQL per la manipolazione e l'estrazione dei dati presenti sul database.

Requisito fondamentale per l'utilizzo della piattaforma SWIM è una connessione a larga banda. Il server centrale e tutti i suoi servizi saranno accessibili all'utente attraverso i comuni web browser quali Mozilla Firefox, Google Chrome, Apple Safari purchè abilitati alla tecnologia Javascript.

### 5.2.3 Hardware

Il server del sistema monterà hard disk in configurazione RAID 1 al fine di scongiurare la perdita dei dati presenti. Il sistema RAID 1 creerà una copia esatta (mirror) di tutti i dati su due o più dischi.

Il server sarà dotato di un microprocessore della famiglia Intel Xeon migliorando così le prestazioni permettendo l'esecuzione di più task contemporaneamente e in modo più efficiente e rapido.

### 5.2.4 Performance

Il sistema, oltre a montare un adeguato microprocessore (come già sopra riportato) dovrà godere di una buona ventilazione del locale nel quale è installato consentendo così di non degradare le prestazioni.

Il server avrà anche bisogno di una connessione internet a larga banda con un throughput proporzionato al carico di lavoro previsto.



### 5.2.5 Sicurezza

La password non verrà salvata in chiaro all'interno del database ma verrà usata una funzione di hash per criptarla e rendere quindi inutile ogni tentativo di furto da parte di malintenzionati. Così facendo gli utenti del sistema avranno la garanzia della sicurezza e della protezione dei loro dati sensibili.

Il server di sistema verrà installato all'interno di un locale appositamente adibito a tale scopo ed opportunamente protetto da sistemi di antifurto, antintrusione e antincendio oltre che ad un meccanismo di backup giornaliero (incrementale) e settimanale (completo dei dischi) che verrà eseguito preferibilmente off site.

Per garantire la continuità del servizio anche in caso di black out di energia elettrica, il sistema dovrà essere dotato di un gruppo di continuità opportunamente dimensionato.

### 5.2.6 Documentazione

Il gruppo di progetto e sviluppo del sistema provvederà a produrre la seguente documentazione al committente:

- Project Plan, per stabilire gli obiettivi e le tempistiche del progetto
- Requirements Analysis Specification Document, per definire il progetto nei suoi requisiti e procedendo con l'analisi delle specifiche
- Design Document (DD), per definire il design del progetto
- Commenti nel codice
- Testing Document

## 6 Identificazione degli scenari e casi d'uso

### 6.1 *Scenari*

**Scenario:** un utente si registra con l'intenzione di cercare aiuto

**Attori:** Cloud Strife

Cloud Strife è un appassionato di spade antiche e ha intenzione di iscriversi a SWIMv2 per trovare un arrotino che possa fare il filo alla sua "buster sword". Di conseguenza segue la procedura guidata e completa la registrazione; in particolare risulta residente a Nibelheim e il suo set di abilità include "negoziazioni con le compagnie energetiche", "maestro di scherma (specialità spada)" e "superatleta".

**Scenario:** un utente dà un feedback negativo

**Attori:** Felice Evacuo, Edward Late

L'utente Felice Evacuo ha richiesto tramite messaggio un aiuto al suo amico inglese Edward Late, ma quest'ultimo per un mese non si è fatto sentire; di conseguenza il signor Evacuo decide di assegnare a mr Late un feedback negativo tramite l'apposita funzionalità presente sulla pagina personale di quest'ultimo.

**Scenario:** un utente risponde ad una richiesta di aiuto

**Attori:** Gabriel Zufolo, Horatio Caine

L'utente Gabriel Zufolo, appassionato di musica ha ricevuto una richiesta di aiuto da parte del suo amico Horatio Caine, un commissario di polizia che è da qualche mese sulle tracce di una banda. Tale banda, con la copertura dei concerti da essa tenuti, in realtà permette a dei complici di svaligiare le case degli ignari spettatori. Caine ha chiesto a Zufolo di dargli delle lezioni di flauto per potersi infiltrare nella banda come musicista e investigare. Zufolo risponde e si dichiara disponibile a collaborare con la polizia, lasciando il proprio numero di telefono per ulteriori contatti.

**Scenario:** un team di specialisti hardware cerca un esperto in compilatori

**Attori:** Bobby Bianchetti

Bobby Bianchetti, specialista in microprocessori, in particolare ricercatore sulle pipeline, decide di integrare nel suo gruppo anche uno specialista in compilatori. Egli non è registrato su SWIM e sebbene provi a scrivere sulla bacheca pubblica riceve in risposta dall'applicazione un popup che gli ricorda di non essere abilitato alla scrittura all'interno di SWIM e l'ho invita a registrarsi.

**Scenario:** un gruppo di utenti cerca un esperto nella gestione di database per realizzare il loro social network

**Attori:** Mark Westfold, Michael Sommer

Mark Westfold, programmatore esperto, nello specifico membro di un team di utenti di SWIM che vorrebbero realizzare un loro social network dedicato all'hacking puro, si incarica di cercare una figura capace di gestire in maniera ottimale il database dell'applicazione, decide quindi di pubblicare nella categoria informatica un thread dove esplicita tale richiesta di aiuto. Michael Sommer, informatico capace e gestore di diversi database presenti in applicazioni a carattere industriale, risponde a tale richiesta mostrandosi notevolmente interessato.

**Scenario:** un calciatore tedesco cerca un preparatore atletico per la riabilitazione dopo l'ennesimo infortunio

**Attori:** Thomas Strunz, Johannes Trapp

Thomas Strunz, calciatore tedesco militante nella squadra dell'oratorio della cattedrale di Monaco di Baviera, si è infortunato per la quarta volta in un anno e decide di cambiare preparatore atletico. Dato che è da tempo un utente registrato su SWIM e si ricorda che un suo vecchio compagno di classe, presente nella sua lista amici del social network, gli aveva parlato di un amico italo-tedesco che aveva intenzione di diventare un preparatore atletico, chiede a costui tramite messaggio il nome del preparatore e, ricevendo come risposta Johannes Trapp, lo cerca tra gli amici dell'ex compagno di classe e, una volta trovato, gli invia la richiesta di amicizia.

**Scenario:** lo stesso calciatore invia all'amministratore Emanuele Uliana un messaggio di lamentela

**Attori:** Emanuele Uliana, Thomas Strunz, Johannes Trapp

Thomas Strunz, dopo un iniziale impegno a tempo pieno nella riabilitazione, per negligenze proprie trascura la seconda fase degli esercizi e decide di rientrare in campo troppo presto, rompendosi alla prima partita il legamento crociato anteriore. A questo punto Johannes Trapp gli invia un messaggio: "Hai visto? Certo che il tuo cognome è proprio un presagio del tuo carattere!" Strunz, si ritiene offeso e invia all'amministratore Emanuele Uliana un messaggio in cui riporta le parole del preparatore atletico e chiede provvedimenti.

**Scenario:** un rappresentante della nazionale piloti cerca un allenatore in vista della sfida di beneficenza contro la nazionale cantanti, ma riesce solo al secondo tentativo

**Attori:** Michele Lumache, Giuseppe Morino, Antonio Marchese

Michele Lumache, esponente di spicco della nazionale calcistica dei piloti di formula 1, e utente registrato su SWIM, è alla ricerca di un allenatore per la squadra che consenta di arrivare preparati alla sfida di beneficenza contro la nazionale cantanti. Inizialmente pensa di aver trovato la persona giusta in Giuseppe Morino, tuttavia dopo la richiesta di amicizia e il primo scambio di messaggi, Michele si accorge che i metodi di Giuseppe non sono troppo ortodossi, in quanto si basano sulla corruzione arbitrale e sull'arroganza nei confronti degli avversari. Di conseguenza si rivolge ad un brillante neoallenatore di nome Antonio Marchese che, grazie alla sua professionalità, convince Michele.

**Scenario:** un utente è costretto alla cancellazione coatta da parte del suo datore di lavoro

**Attori:** Montgomery Arde, Smith ER. S.

Montgomery Arde è il capo di una società energetica americana a cui è stata commissionata l'apertura di una centrale nucleare in Namibia. Non potendo trattare di persona con gli indigeni del posto, decide di inviare il suo fedele assistente Smith ER. S. in sua vece. Tuttavia Smith ha un piccolo difetto: se non lo si distoglie a forza dal social network SWIM (su cui è iscritto in quanto "grande esperto di lingue autoctone dell'Africa subsaariana" e "ottimo mediatore con i selvaggi Boscimani"), ci passa tutta la giornata. Di conseguenza Mr Arde gli impone di disiscriversi, promettendogli uno stipendio di 1201 dollari al mese contro i 1200 che già guadagnava. Smith fedelmente segue la procedura guidata e cancella il proprio account, allettato dalla straordinaria offerta economica del suo capo.

**Scenario:** un utente chiede informazioni sulle domande che Google effettua durante i colloqui di lavoro

**Attori:** Lorenzo Pagina, Giorgia Angurie

Giorgia Angurie è un'abile organizzatrice di eventi con esperienze politiche alle spalle e vorrebbe diventare parte della famosa azienda di Mountain View. Tra i suoi amici su SWIM ce n'è uno che come abilità ha "esperto in colloqui di lavoro impossibili" e "ex dipendente di google": si chiama Lorenzo Pagina. Giorgia chiede a Lorenzo se può inviarle qualche fac-simile delle domande dei colloqui di Google. Lorenzo risponde con tre delle domande che ha visto fare agli aspiranti lavoratori: 1) Sei ridotto alle dimensioni di una moneta, anche la tua massa viene ridimensionata per mantenere la tua densità al livello che avevi prima e vieni gettato in un frullatore che tra 60 secondi entrerà in funzione. Che cosa faresti? 2) Quanto dovresti pagare per lavare tutte le finestre di Seattle? 3) Quante palline da golf ci stanno in uno scuolabus?

**Scenario:** un utente chiede di modificare il proprio set di abilità

**Attori:** Walter Rubino, Gabriele Rufolo, Alex Ziggie

Alex Ziggie è un utente di SWIM che vorrebbe aggiungere al proprio set di abilità le seguenti: "venditore di aria fritta" e "capacità di apparire dal nulla" e, a tal proposito invia un messaggio all'amministratore Walter Rubino per la prima e all'amministratore Gabriele Rufolo per la seconda. Walter ritiene che la richiesta a lui pervenuta non rispetti il vincolo di realtà delle abilità: dubita infatti che una miscela di gas possa essere fritta a mo' di patatine; Gabriele per lo stesso motivo boccia la seconda richiesta. Entrambi notificano ad Alex il fatto che le sue richieste non possano essere accolte.

## 6.2 Casi d'uso

<b>Nome del caso</b>	Un utente si registra
<b>Attori</b>	Ospite
<b>Condizione di entrata</b>	L'ospite clicca sul pulsante "Registrati"
<b>Flusso degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'ospite inserisce negli appositi campi nome, cognome, città, il suo indirizzo mail e la sua password (quest' ultima scelta sul momento).</li> <li>2. L'ospite seleziona dal set predefinito le abilità che ritiene di possedere.</li> <li>3. L'ospite clicca sul pulsante di conferma.</li> <li>4. Il sistema effettua apposite query sul database per verificare che non esista già la mail indicata ed esegue un controllo sulla password per verificare che abbia la lunghezza minima e contenga solo caratteri permessi.</li> <li>5. Il sistema inserisce nel database i dati del neo-utente ma non mette il flag nel campo corrispondente all'attributo "confermato"</li> <li>6. Il sistema invia una mail all'indirizzo fornito dall'utente con un link di conferma.</li> <li>7. L'utente apre la mail e clicca sul link di conferma.</li> <li>8. Il sistema mette il flag nel campo di cui sopra.</li> <li>9. Il sistema mostra la pagina di avvenuta registrazione (il login comunque non è ancora stato effettuato).</li> </ol>
<b>Condizione di uscita</b>	Il sistema ha mostrato la pagina di avvenuta registrazione
<b>Eccezioni</b>	<ol style="list-style-type: none"> <li>1. La mail inserita dall'utente esiste già nel database</li> <li>2. La password è lunga meno di 8 caratteri o contiene caratteri diversi da lettere, numeri o underscore.</li> </ol>

<b>Nome del caso</b>	Un ospite effettua il login
<b>Attori</b>	Ospite
<b>Condizione di entrata</b>	L'ospite clicca sul pulsante "Login"
<b>Flusso degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'ospite inserisce negli appositi campi mail e password.</li> <li>2. L'ospite preme il pulsante di conferma.</li> <li>3. Il sistema effettua una query nel database con i dati inseriti e, in caso la query ritorni un valore non vuoto, visualizza la pagina personale dell'utente corrispondente.</li> </ol>
<b>Condizione di uscita</b>	L'utente si trova nella sua pagina personale
<b>Eccezioni</b>	La query del sistema ritorna null

<b>Nome del caso</b>	Un utente modifica il proprio set di abilità
<b>Attori</b>	Utente
<b>Condizione di entrata</b>	L'utente clicca sul pulsante di modifica del set di abilità presente nella sua home
<b>Flusso degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente eventualmente (non è obbligato a farlo) toglie il flag da una o più abilità che ha e mette il flag su una o più nuove abilità. (Non c'è limite al numero di abilità possedibili ed è possibile avere zero abilità)</li> <li>2. L'utente clicca sul pulsante di conferma.</li> <li>3. Il sistema aggiorna il database</li> <li>4. Il sistema notifica all'utente il successo dell'operazione e lo reindirizza alla sua pagina personale</li> </ol>
<b>Condizione di uscita</b>	L'utente si ritrova nella sua pagina personale e il suo set è modificato correttamente
<b>Eccezioni</b>	Nessuna: gli unici errori possono essere dovuti a guasti non deterministici della macchina o a cadute della connessione internet

<b>Nome del caso</b>	Un ospite tenta di pubblicare una richiesta di aiuto in una categoria
<b>Attori</b>	Ospite
<b>Condizione di entrata</b>	L' ospite clicca sul pulsante "Inserisci messaggio" presente sulla bacheca pubblica
<b>Flusso degli eventi</b>	<ol style="list-style-type: none"> <li>1. Il sistema risponde con un popup</li> <li>2. L' ospite legge il contenuto del popup e clicca sul pulsante ok per chiuderlo</li> </ol>
<b>Condizione di uscita</b>	L'ospite si ritrova nella pagina della categoria
<b>Eccezioni</b>	Nessuna, vedi caso precedente

<b>Nome del caso</b>	Un utente pubblica una richiesta di aiuto sulla bacheca pubblica
<b>Attori</b>	Utente
<b>Condizione di entrata</b>	L' utente clicca sul pulsante "Inserisci messaggio" presente sulla bacheca pubblica
<b>Flusso degli eventi</b>	<ol style="list-style-type: none"> <li>1. L' utente compone il messaggio</li> <li>2. L' utente clicca sul pulsante di pubblicazione del messaggio</li> <li>3. Il sistema fa in modo che il messaggio diventi visibile a chiunque</li> </ol>
<b>Condizione di uscita</b>	Il messaggio è visibile in bacheca
<b>Eccezioni</b>	Nessuna

<b>Nome del caso</b>	Un ospite effettua una ricerca su tutto l'insieme di utenti di SWIM
<b>Attori</b>	Ospite
<b>Condizione di entrata</b>	L'ospite clicca sul pulsante di ricerca
<b>Flusso degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'ospite riempie almeno uno dei seguenti campi: "nome e cognome", "città" e/o seleziona almeno un'abilità dalla scroll-list.</li> <li>2. L'ospite clicca sul pulsante di conferma.</li> <li>3. Il sistema effettua una query con i dati ricevuti.</li> <li>4. Il sistema mostra una finestra con una lista (eventualmente vuota) di risultati</li> </ol>
<b>Condizione di uscita</b>	La finestra dei risultati è visibile
<b>Eccezioni</b>	<ol style="list-style-type: none"> <li>1. Viene effettuata una ricerca senza parametri.</li> <li>2. I dati inseriti contengono caratteri non consentiti.</li> </ol>

<b>Nome del caso</b>	Un utente chiede ad un amministratore di aggiungere una nuova abilità al set predefinito di sistema
<b>Attori</b>	Utente, Amministratore
<b>Condizione di entrata</b>	L'utente clicca sul pulsante "Invia messaggio" dal profilo pubblico dell'amministratore
<b>Flusso degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente compone il corpo del messaggio includendo una chiara descrizione della nuova abilità, il suo nome e le motivazioni per cui ha proposto di aggiungerla.</li> <li>2. L'utente clicca sul pulsante di invio messaggio.</li> <li>3. Il sistema invia il messaggio all'amministratore e fa comparire una notifica sulla sua pagina personale.</li> </ol>
<b>Condizione di uscita</b>	Il messaggio è inviato correttamente e sulla pagina dell'amministratore compare una notifica
<b>Eccezioni</b>	Nessuna



<b>Nome del caso</b>	Un utente invia una richiesta di amicizia ad un altro utente senza passare dai suggerimenti
<b>Attori</b>	Utente A, Utente B
<b>Condizione di entrata</b>	L'utente A clicca sul pulsante "Aggiungi" dalla pagina pubblica dell'utente B
<b>Flusso degli eventi</b>	Il sistema notifica all'utente B che l'utente A gli ha inviato una richiesta di amicizia.
<b>Condizione di uscita</b>	All'utente B è arrivata la notifica
<b>Eccezioni</b>	Nessuna

<b>Nome del caso</b>	Un utente risponde positivamente ad una richiesta di amicizia
<b>Attori</b>	Utente C, Utente D
<b>Condizione di entrata</b>	L'utente C clicca su "Conferma" dalla notifica della richiesta
<b>Flusso degli eventi</b>	<ol style="list-style-type: none"> <li>1. Il sistema aggiunge D agli amici di C e viceversa.</li> <li>2. Il sistema notifica C del buon fine dell'azione e D del fatto che C abbia accettato la sua richiesta.</li> </ol>
<b>Condizione di uscita</b>	Le notifiche sono state spedite agli utenti coinvolti
<b>Eccezioni</b>	Nessuna

<b>Nome del caso</b>	Un utente dà un feedback
<b>Attori</b>	Utente A, Utente B
<b>Condizione di entrata</b>	L'utente A ha ricevuto un messaggio di risposta dall'utente B
<b>Flusso degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente A clicca sul pulsante "Dai feedback" contenuto nel messaggio di risposta di B.</li> <li>2. L'utente A seleziona il livello del feedback (da 1 a 10) dal popup che il sistema ha aperto</li> <li>3. L'utente clicca su "Conferma".</li> <li>4. Il sistema notifica B del feedback ricevuto con un messaggio automatico e aggiorna la media dei suoi feedback.</li> </ol>
<b>Condizione di uscita</b>	La media è correttamente aggiornata
<b>Eccezioni</b>	Nessuna

<b>Nome del caso</b>	Un utente cancella il proprio account
<b>Attori</b>	Utente
<b>Condizione di entrata</b>	L'utente preme il pulsante "Cancella account" dalla propria pagina personale
<b>Flusso degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente conferma la volontà di cancellazione dal popup che il sistema apre.</li> <li>2. Il sistema slogga l'utente, cancella tutti i suoi legami di amicizia (il che vuol dire che rimuove tutti i suoi amici e dalle loro liste di amici rimuove l'utente stesso) e cancella i suoi dati dal database.</li> <li>3. Il sistema notifica con un popup l'avvenuta cancellazione e reindirizza alla pagina principale di SWIM.</li> </ol>
<b>Condizione di uscita</b>	Il sistema ha notificato l'avvenuta cancellazione
<b>Eccezioni</b>	Nessuna

## **7 Modelli UML**

### ***7.1 Sequence diagrams***

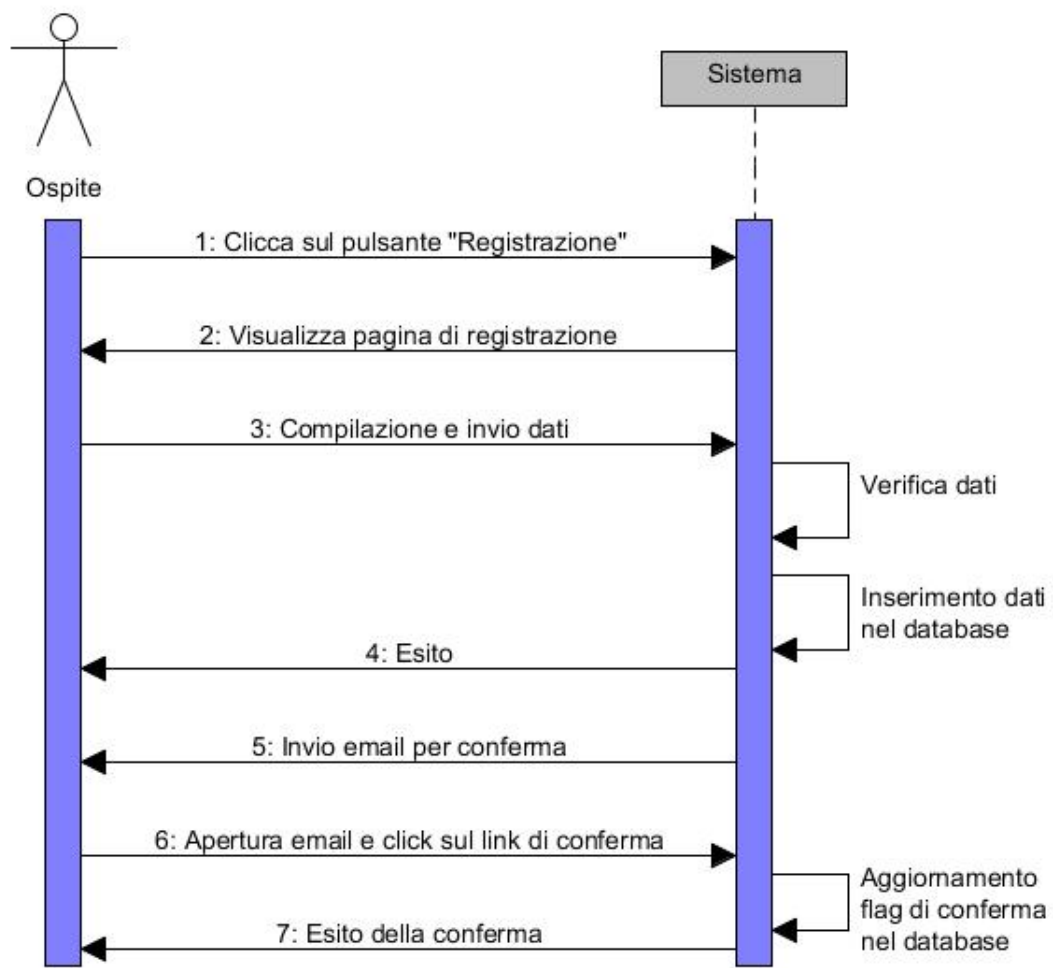


Figura 1: Creazione di un utente

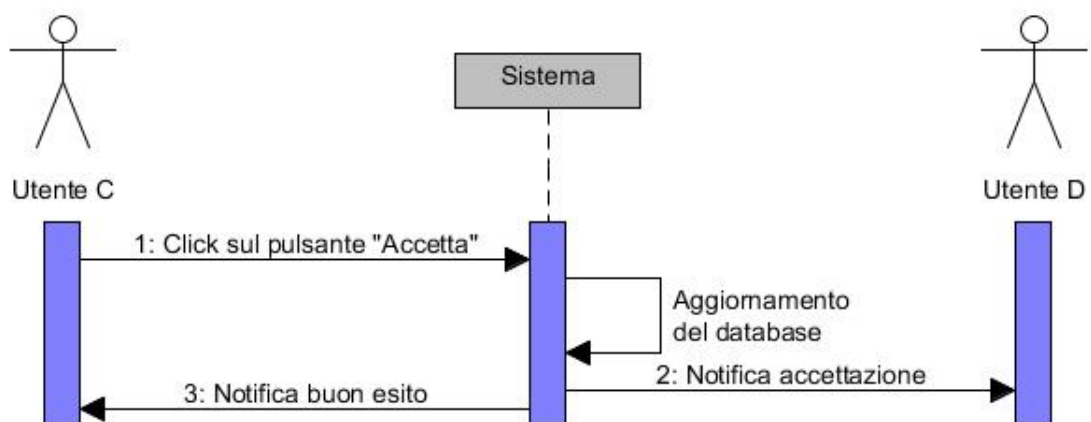


Figura 2: Accettazione di una richiesta di amicizia

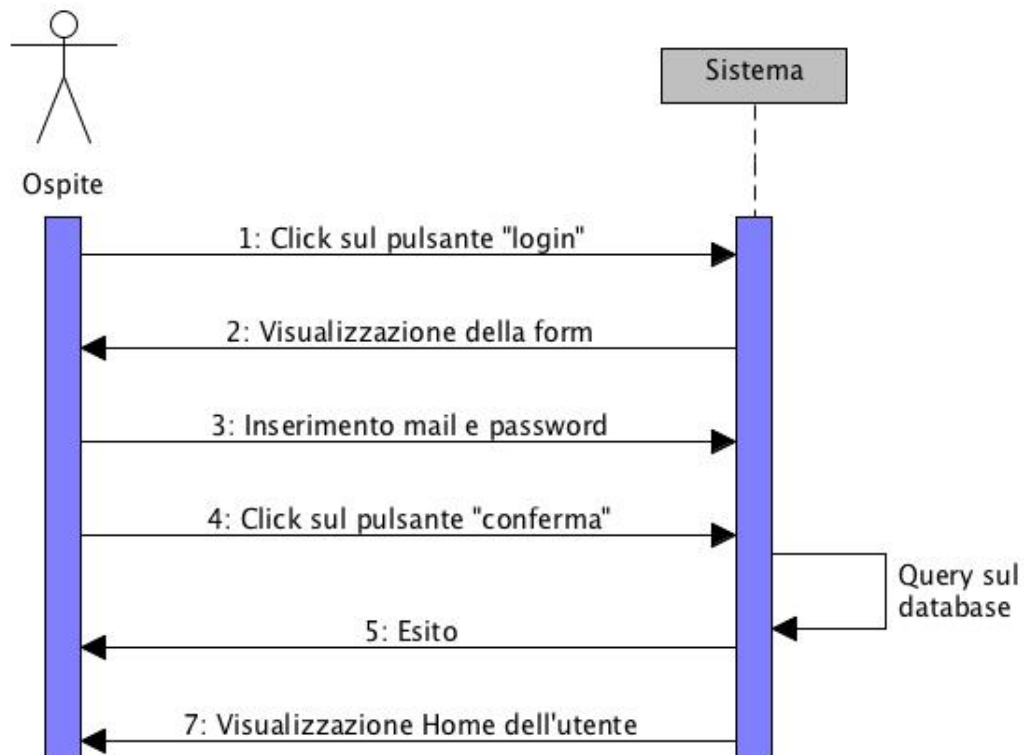


Figura 3: Login di un utente

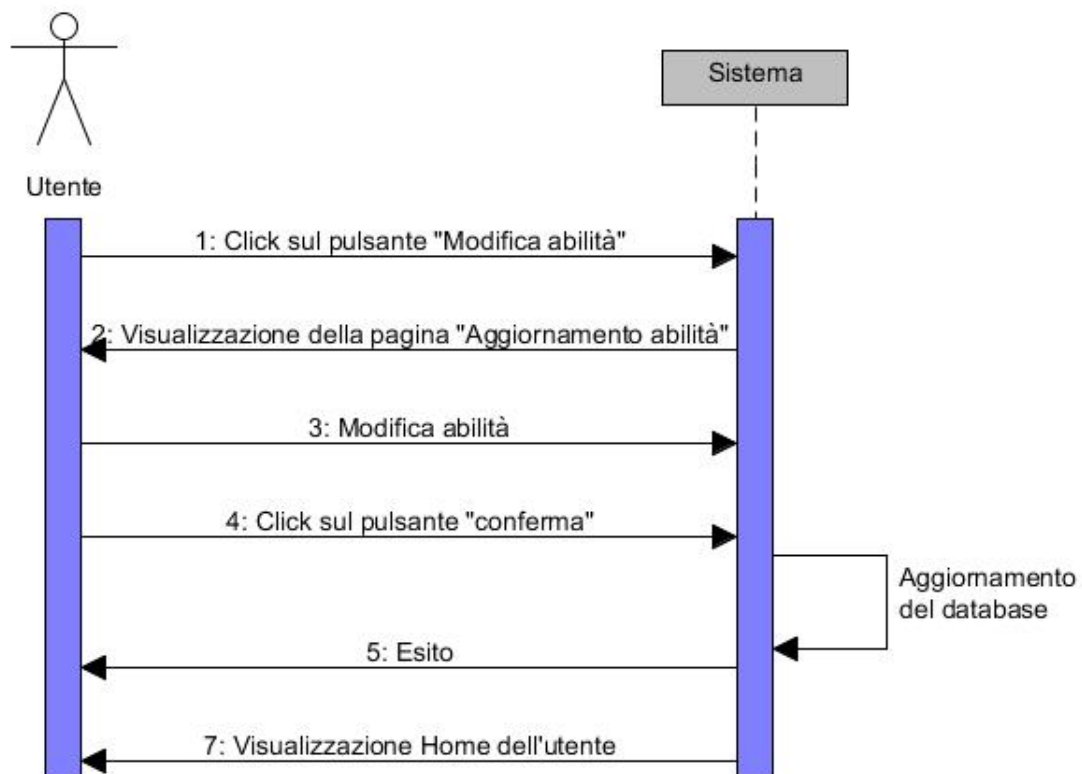


Figura 4: Modifica del set di abilità



Figura 5: Invio di una richiesta di amicizia

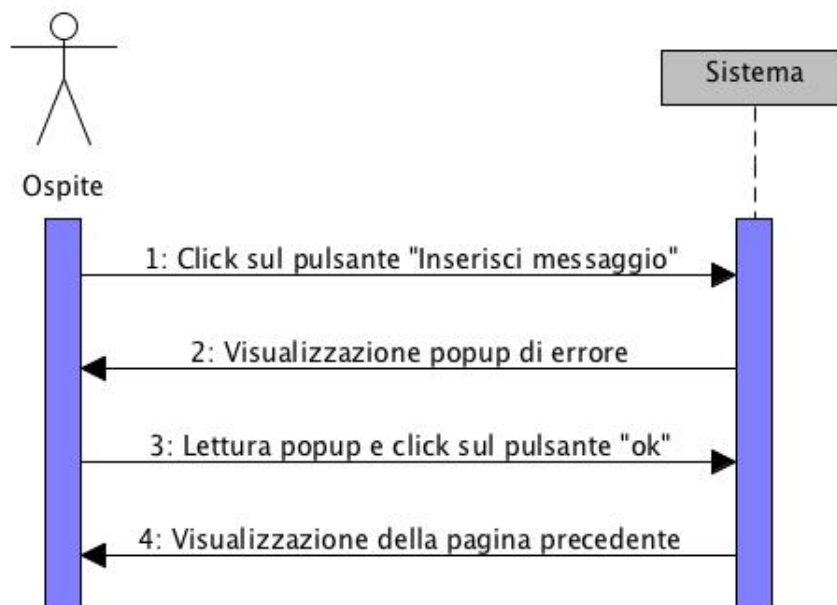


Figura 6: Un ospite tenta di creare un thread

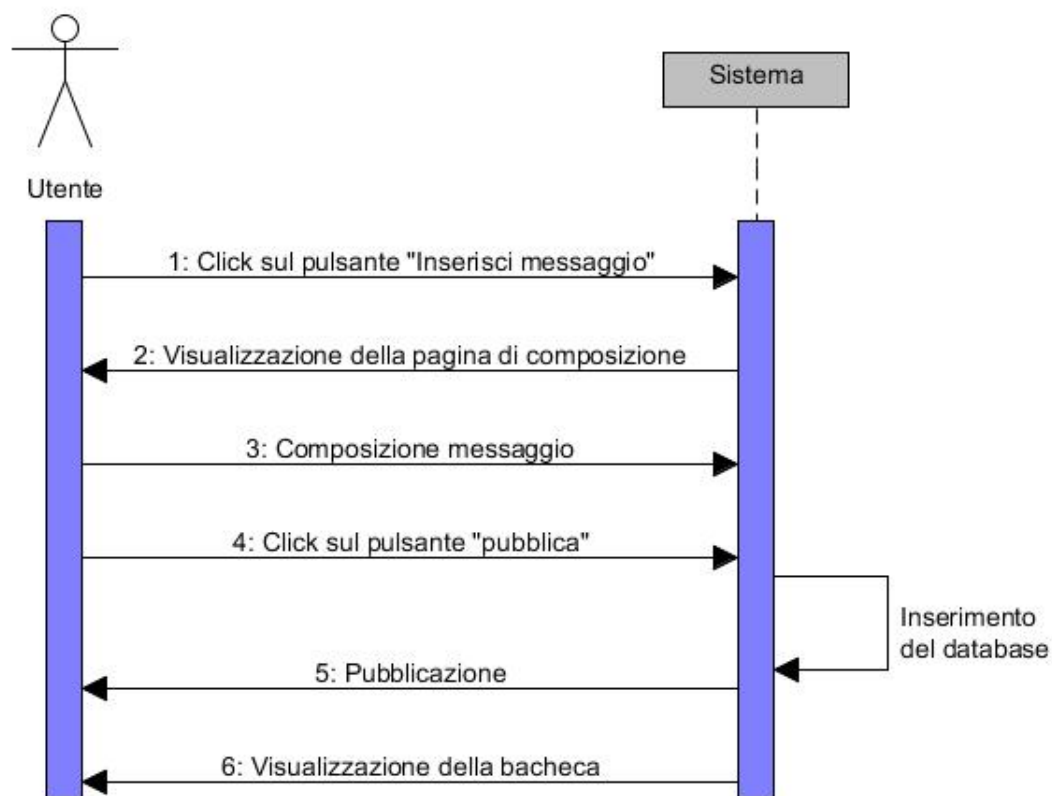


Figura 7: Invio di un nuovo messaggio sulla bacheca pubblica

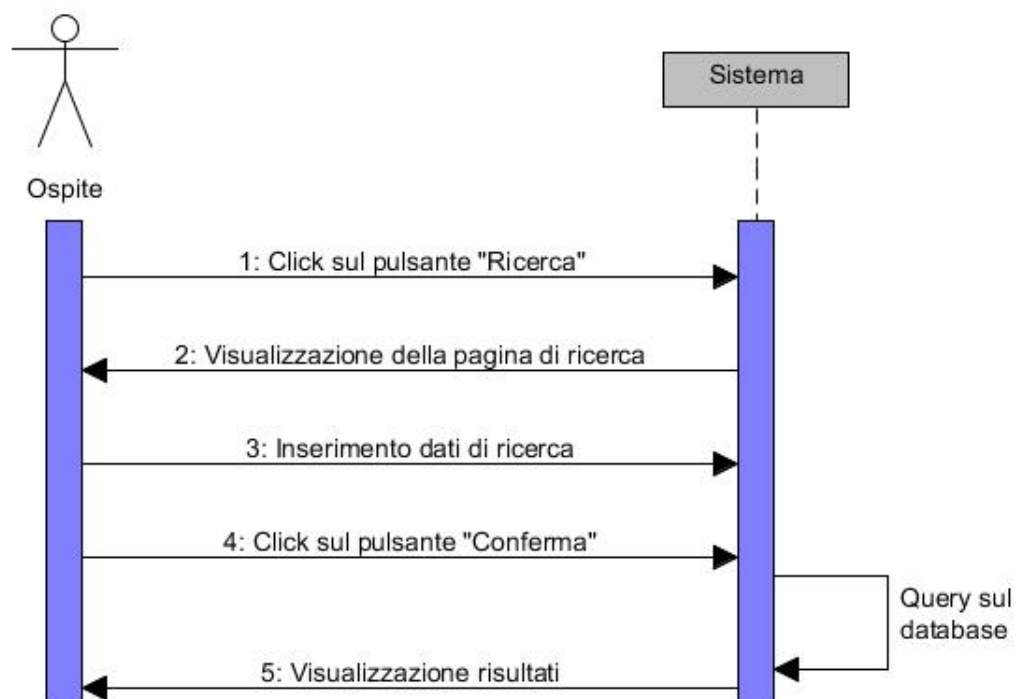


Figura 8: Ricerca di un utente

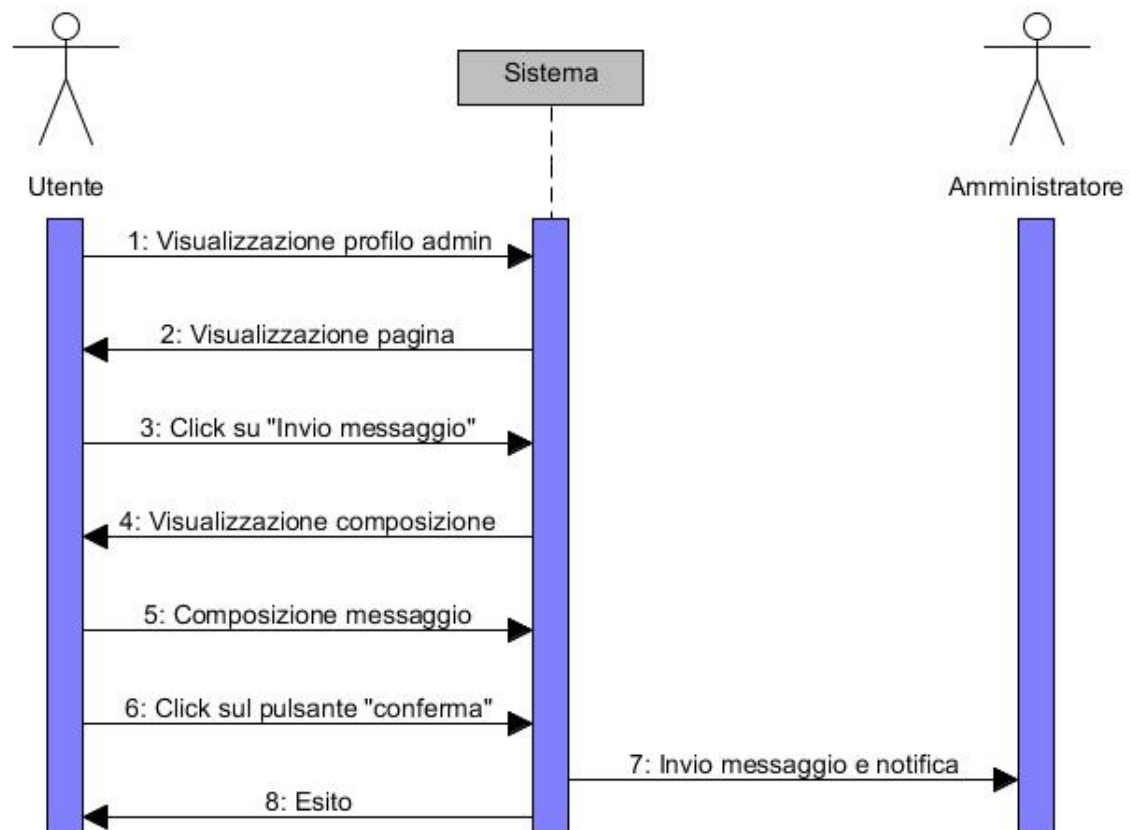


Figura 9: Invio di un messaggio privato

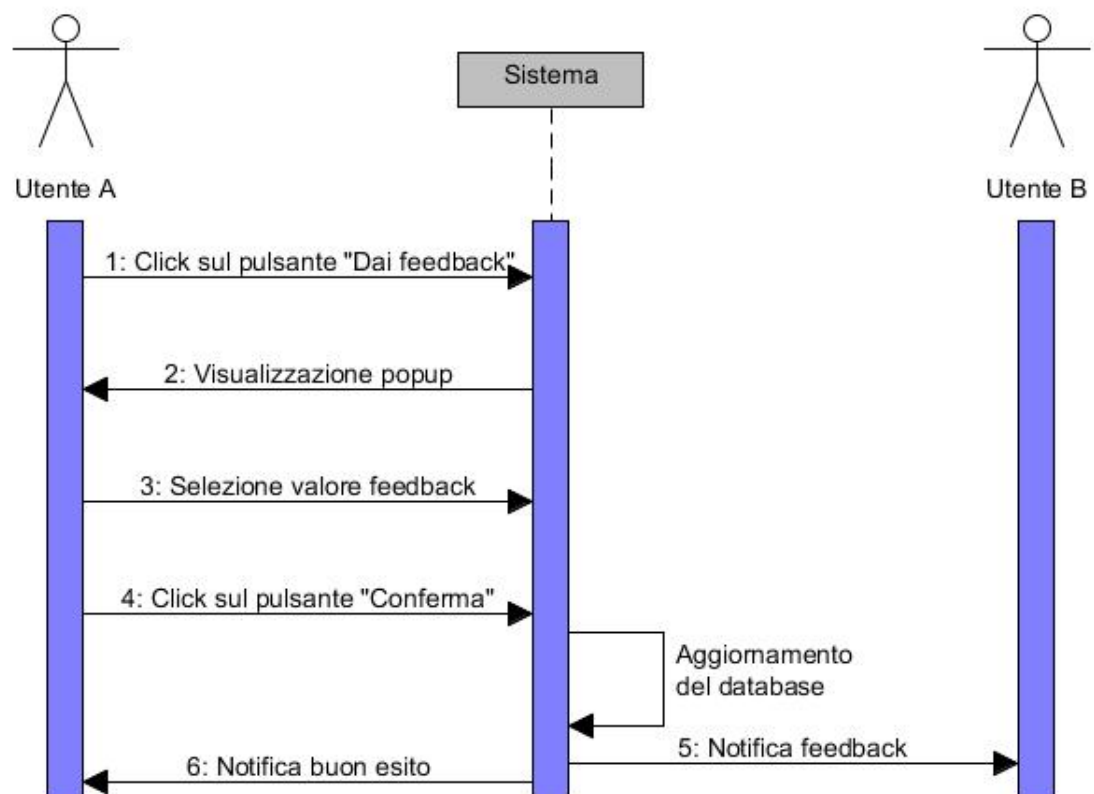


Figura 10: Invio di un feedback



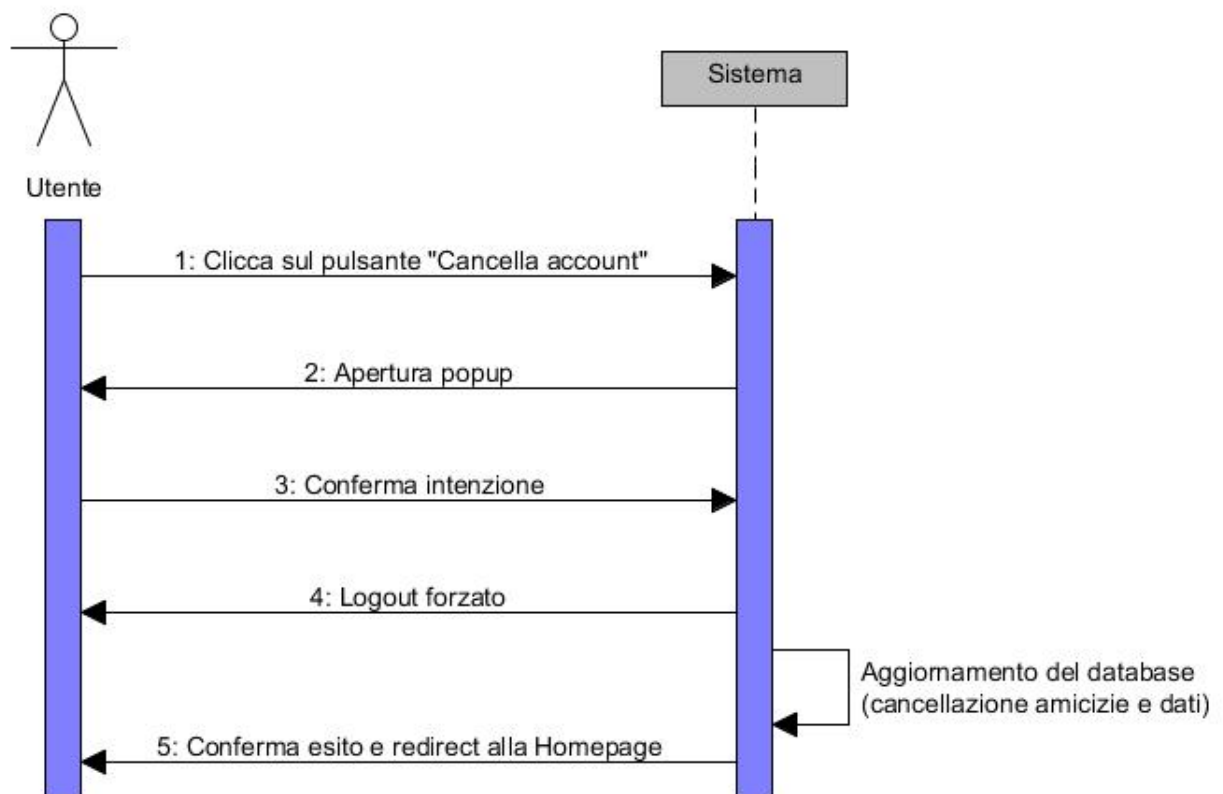


Figura 11: Cancellazione dell'account

## 7.2 Statechart

Con i seguenti statechart abbiamo controllato il flusso delle operazioni del sistema in alcuni casi specifici:

1. *Invio del feedback dopo l'aiuto ricevuto:*



2. *Inserimento di una risposta ad un messaggio:*



## 8 Modello Alloy

### 8.1 Signatures

```
//SIGNATURES

sig Utente{
  inviati: some Messaggio,
  ricevuti: some Messaggio,
  abilità: set Abilità,
}

sig Ospite{
}

sig Admin extends Utente{
}

sig Messaggio{
}

sig Amicizia{
  u1: one Utente,
  u2: one Utente
}

sig Abilità{
}

sig Feedback{
  valore: one Int,
  u1: one Utente,
  u2: one Utente
}
```

### 8.2 Facts

```
//FACTS

//un utente non può essere amico di sè stesso
fact{
  all a: Amicizia {a.u1 != a.u2}
}

//un messaggio deve essere stato inviato da qualcuno, in particolare da uno e un solo utente
fact{
  all m: Messaggio | one u: Utente | m in u.inviati
}

//se un messaggio è stato ricevuto, qualcuno, in particolare non il destinatario, deve averlo inviato
fact{
  all m: Messaggio | all dest: Utente | one mitt:Utente | {dest != mitt && {m in dest.ricevuti implies m in mitt.inviati}}
}

//un utente non può dare feedback a sè stesso
fact{
  all f: Feedback | f.u1 != f.u2
}
```

## 9 Strumenti utilizzati

1. **L<sup>A</sup>T<sub>E</sub>X**: TeXstudio, Kile e MacTeX
2. **Alloy Analyzer 4.2**: per la gestione delle specifiche scritte in Alloy
3. **Umllet**: per la gestione dei grafici e dei diagrammi UML