

Progetto di Ingegneria del Software 2

SWIMv2

**Requirements Analysis Specification Document**

A.A. 2012/2013

**Autori**: Emanuele Uliana, Gabriele Rufolo, Walter Rubino  
**Docente**: Prof.ssa Raffaela Mirandola

# Descrizione del problema

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).

**Indice**

[Descrizione del problema 2](#_Toc340769562)

[1. Introduzione 4](#_Toc340769563)

[1.1. Glossario 4](#_Toc340769564)

[1.2. Obiettivi 5](#_Toc340769565)

[2. La piattaforma SWIMv2 6](#_Toc340769566)

[2.1. Identificazione degli attori 6](#_Toc340769567)

[3. Considerazioni preliminari 7](#_Toc340769568)

[4. Specifiche funzionali 8](#_Toc340769569)

[5. Specifiche non funzionali 9](#_Toc340769570)

[6. Identificazione degli scenari 10](#_Toc340769571)

[7. Modello UML 15](#_Toc340769572)

[7.1. Sequence diagram 15](#_Toc340769573)

[7.2. Statechart 17](#_Toc340769574)

[8. Modello Alloy 17](#_Toc340769575)

[9. Strumenti utilizzati 17](#_Toc340769576)

# Introduzione

## 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 divisi per categorie e contenenti thread

* **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)

## 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)

# 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 uteti 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.

## 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.

# 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.

# Requisiti 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 la soluzioni non verranno rimossi restando così visibili

# Requisiti non funzionali

Sistema usato per il backup, gruppo di continuità…

fornire un’interfaccia aggiuntiva a quella degli utenti registrati nella quale è possibile visionare le richieste di inserimento di nuove abilità e poter decidere se accettarle o meno.

## Documentazione

* 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

*Continua…*

# Identificazione degli scenari

|  |  |
| --- | --- |
| Nome dello scenario | Un ospite si registra |
| Attori | Ospite |
| Flusso degli eventi | L'ospite entra nel sito  L'ospite clicca sul pulsante per registrarsi  L'ospite compila il form di registrazione  L'ospite conferma i dati inseriti  Il sistema verifica i dati e invia una mail di verifica  L'ospite apre la sua mail, cerca il messaggio e cliccando su un link contentuto in esso conferma la registrazione  L'ospite è finalmente registrato e può loggarsi |

|  |  |
| --- | --- |
| Nome dello scenario | Un ospite effettua il login con successo |
| Attori | Ospite |
| Flusso degli eventi | L'ospite entra nel sito  L'ospite clicca sul pulsante di login  L'ospite inserisce username e password  L'ospite clicca sul pulsante di conferma  Il sistema interroga il database e, trovata una corrispondenza, visualizza la pagina principale dell'utente corrispondente ai dati inseriti  L'ospite è diventato un utente registrato |

|  |  |
| --- | --- |
| Nome dello scenario | Un utente modifica il proprio set di abilità (rimanendo comunque tra quelle predefinite dagli amministratori) |
| Attori | Utente |
| Flusso degli eventi | L'utente clicca sul pulsante di modifica delle abilità  L'utente seleziona le nuove abilità che ritiene di possedere e deseleziona quelle che aveva e ritiene di non possedere più  L'utente clicca sul pulsante di conferma  L'utente ha modificato con successo il proprio set di abilità |

|  |  |
| --- | --- |
| Nome dello scenario | Un utente effettua una ricerca con più condizioni che non trova alcun utente con quelle caratteristiche |
| Attori | Utente |
| Flusso degli eventi | L'utente clicca sul pulsante di ricerca  L'utente nel campo “nome” della ricerca scrive “Ajeje Brazorf”; nel campo “città” scrive “Milano” e lascia vuoto quello riferito alle abilità  L'utente clicca sul pulsante di conferma della ricerca  Il sistema effettua una query basata sulle stringhe scritte dall'utente nei campi della ricerca e tale query restituisce un insieme vuoto di tuple  Il sistema mostra la pagina di “nessun risultato” |

|  |  |
| --- | --- |
| Nome dello scenario | Un utente cancella il proprio account |
| Attori | Utente |
| Flusso degli eventi | L'utente clicca sul pulsante che consente di avviare la procedura di cancellazione del proprio account  Il sistema mostra un prompt di conferma/annullamento dell'azione  L'utente conferma la richiesta  Il sistema rimuove tutte le amicizie dell'utente  Il sistema esegue automaticamente il logout dell'utente  Il sistema cancella dal database tutte le tracce dell'utente  Il sistema notifica con un prompt l'avvenuta cancellazione |

|  |  |
| --- | --- |
| Nome dello scenario | Un utente invia una richiesta di amicizia ad un altro utente dopo averlo cercato; quest'ultimo la accetta |
| Attori | Utente, altro utente |
| Flusso degli eventi | L'utente A ha già effettuato una ricerca e vuole chiedere l'amicizia ad un utente B  L'utente A clicca sul bottone di richiesta di amicizia  Il sistema inoltra la richiesta all'utente B  L'utente B vede una notifica nella sua area personale  L'utente B clicca sul pulsante di accettazione dell'amicizia  L'utente B compare nella lista di amici di A e viceversa |

|  |  |
| --- | --- |
| Nome dello scenario | Un utente invia ad un amministratore la richiesta di aggiungere una nuova abilità (prima non esistente del tutto) alla propria lista; l'amministratore non accetta |
| Attori | Utente, Amministratore |
| Flusso degli eventi | L'utente preme il pulsante di composizione di un nuovo messaggio all'amministratore  L'utente nel corpo del messaggo chiede che gli venga aggiunta l'abilità di “venditore di fumo”  L'utente clicca sul pulsante di invio messaggio  Il sistema inoltra il messaggio all'amministratore  L'amministratore, ricevuta la notifica, legge il messaggio e giudica sconveniente la richiesta  L'amministratore non modifica le abilità e invia all'utente un messaggio in cui spiega le sue ragioni  Nulla è cambiato né nelle abilità dell'utente, né in quelle predefinite di sistema |

|  |  |
| --- | --- |
| Nome dello scenario | Un utente invia un messaggio ad un suo amico |
| Attori | Utente, altro utente |
| Flusso degli eventi | L'utente A clicca sul pulsante che permette di inviare un messaggio ad un suo amico B  L'utente a compone il corpo del messaggio  L'utente A clicca sul pulsante di conferma invio messaggio  Il sistema inoltra il messaggio all'utente B |

|  |  |
| --- | --- |
| Nome dello scenario | Un ospite pubblica una richiesta di aiuto sulla bacheca pubblica |
| Attori | Ospite |
| Flusso degli eventi | Un ospite clicca sul pulsante di accesso alla bacheca pubblica  L'ospite clicca sul pulsante di inserimento messaggio  L'ospite compone ul suo messaggio  L'ospite preme sul pulsante di conferma di invio del messaggio  Il messaggio è visibile nella bacheca pubblica |

|  |  |
| --- | --- |
| Nome dello scenario | Un utente dà un feedabck riguardo l'aiuto ricevuto |
| Attori | Utente |
| Flusso degli eventi | L'utente ha appena ricevuto un messaggio di risposta a una sua richiesta e desidera inviare un feedback  L'utente clicca sul pulsante che permette di inviare un feedback, pulsante presente sulla pagina del messaggio ricevuto.  L'utente è soddisfatto dell'aiuto ricevuto e quindi decide di assegnare il massimo dei punti 10/10 cliccando sulla stelletta corrispondente  L'utente conferma l'invio del feedback cliccando sull'apposito bottone |

# Modello UML

## Sequence diagram

Abbiamo analizzato il sistema in alcuni casi di utilizzo:

1. *Creazione di un nuovo utente*

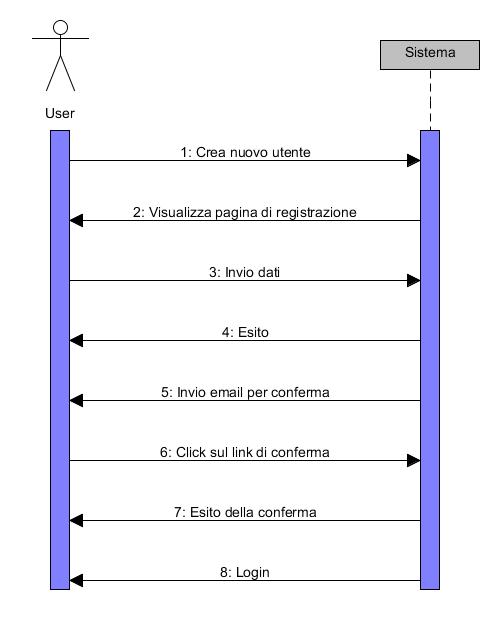


Figura 1 - Creazione utente

1. *Creazione di un nuovo thread*

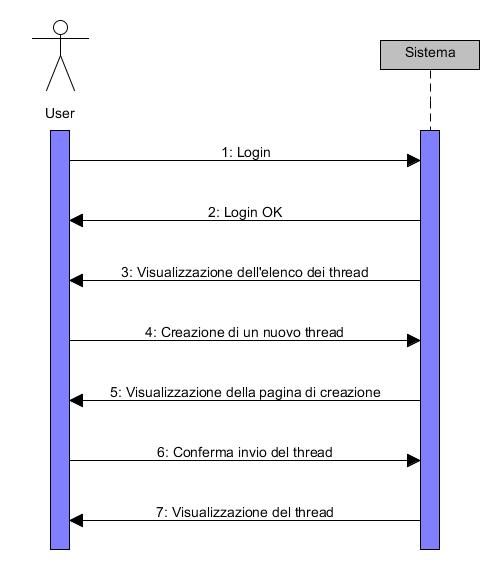


Figura 2 - Creazione di un thread

## Statechart

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

1. *Inserimento di una risposta ad un messaggio*

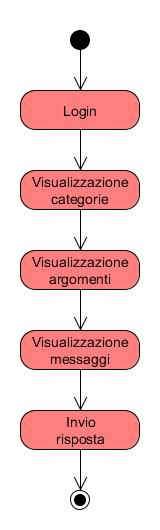


Figura 3 - Inserimento di una risposta

1. *Invio del feedback dopo l’aiuto ricevuto*

**

Figura 4 - Invio del feedback

# Modello Alloy

# Strumenti utilizzati

* Microsoft Office 2013
* Umlet
* Alloy Analizer 4.2