Progetto di Ingegneria del Software 2

SWIMv2

Design Document

A.A. 2012/2013

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

Docente:
Prof.ssa Raffaela Mirandola

Versione 1.0 del 22/12/2012

Indice

1 Panoramica della piattaforma

La piattaforma sviluppata ha il compito di promuovere, tramite l'utilizzo di una web application, la collaborazione e la condivisione di informazioni utili alla risoluzione di problemi tra gli utenti.

Nel dettaglio, il presente documento di Design definisce la struttura concettuale e funzionale fornendo una precisa descrizione delle guidelines che saranno seguite nello sviluppo e nel deployment dell'applicazione.

Il documento è inoltre conforme con le specifiche presenti nel RASD.

Il sistema è utilizzato da tre categorie di utenti: gli ospiti, gli utenti registrati e gli amministratori. Tramite un sistema di autenticazione il sistema controlla chi può svolgere determinate azioni in base al proprio ruolo.

Per evitare l'accesso a servizi non consentiti si è deciso di utilizzare il Role-Based Access Control. Con tale sistema di autenticazione si consentirà di usufruire solamente dei servizi specifici per il proprio ruolo.

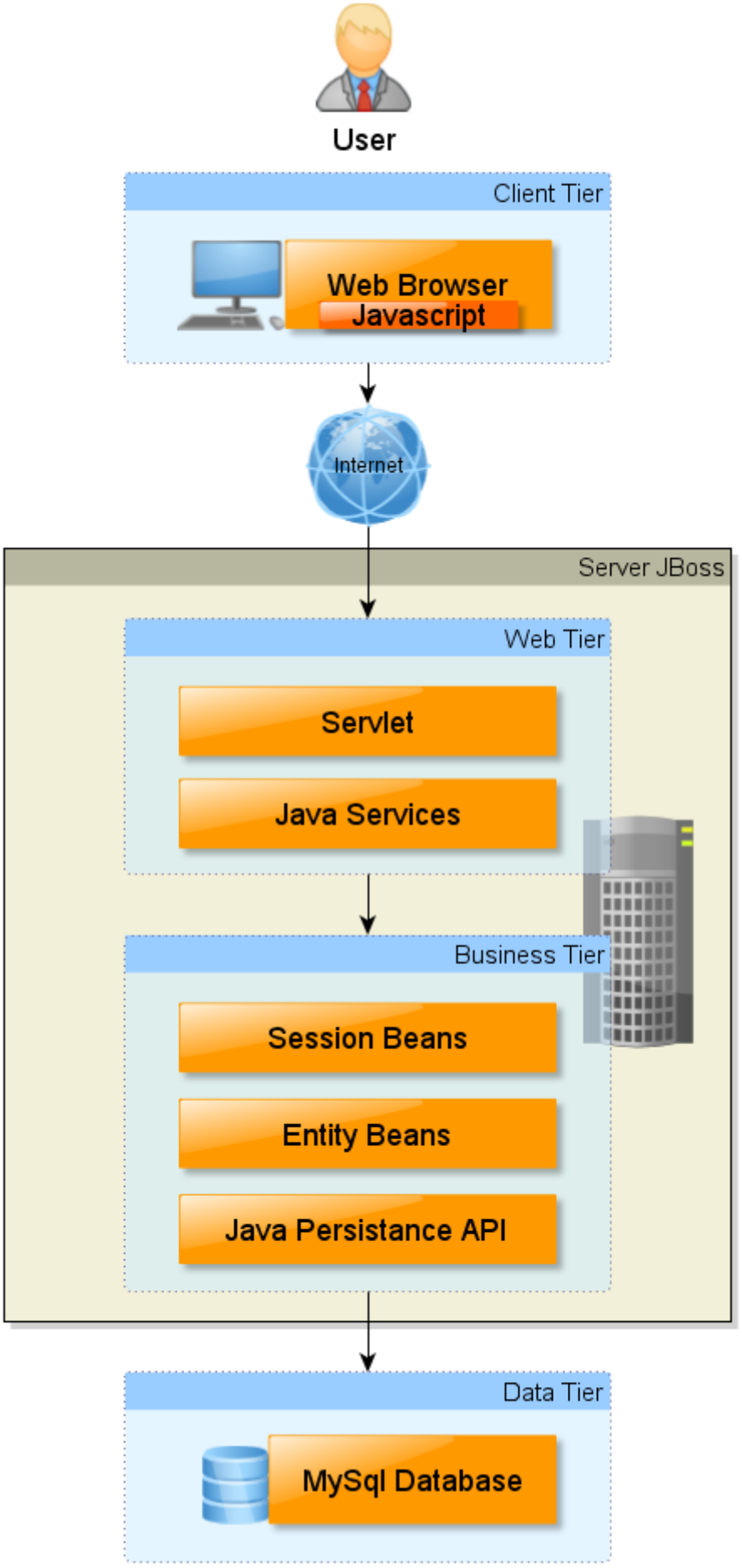
2 Descrizione del sistema

Il sistema è composto da varie pagine HTML ognuna delle quali fornisce uno specifico servizio al cliente. Ogni webpage include la tecnologia Javascript che consente non solo di migliorare l'interfaccia stessa della pagine ma permette anche di ottimizzare la facilità di navigazione.

Per sviluppare il sistema sono stati usati i seguenti tool:

- WaveMaker, per creare l'interfaccia grafica;
- Eclipse Juno, per creare la logica di business;
- Mysql, per gestire la base di dati.

2.1 L'architettura del sistema



La piattaforma SWIMv2 è un applicazione multi-tier composta dai seguenti livelli:

- Client tier: è composta dal client side dell'applicazione: si occupa di inviare le richieste al webserver tramite il browser. Include il codice Javascript e il codice HTML per la presentazione dei servizi.
- Web tier: comprende le servlet per le risposte alle richieste Ajax e quindi con il client tier. Hanno anche il compito di gestire la sessione con gli utenti del sistema e di ricevere da quest'ultimi gli input inviati e di passarli quindi al business tier.
- Business tier: gestisce la logica di sistema
- Data tier: comprende la base di dati della quale si servirà il business tier. Tale base dati comprenderà tutti i dati e tutte le informazioni del sistema.

2.2 Sottosistemi

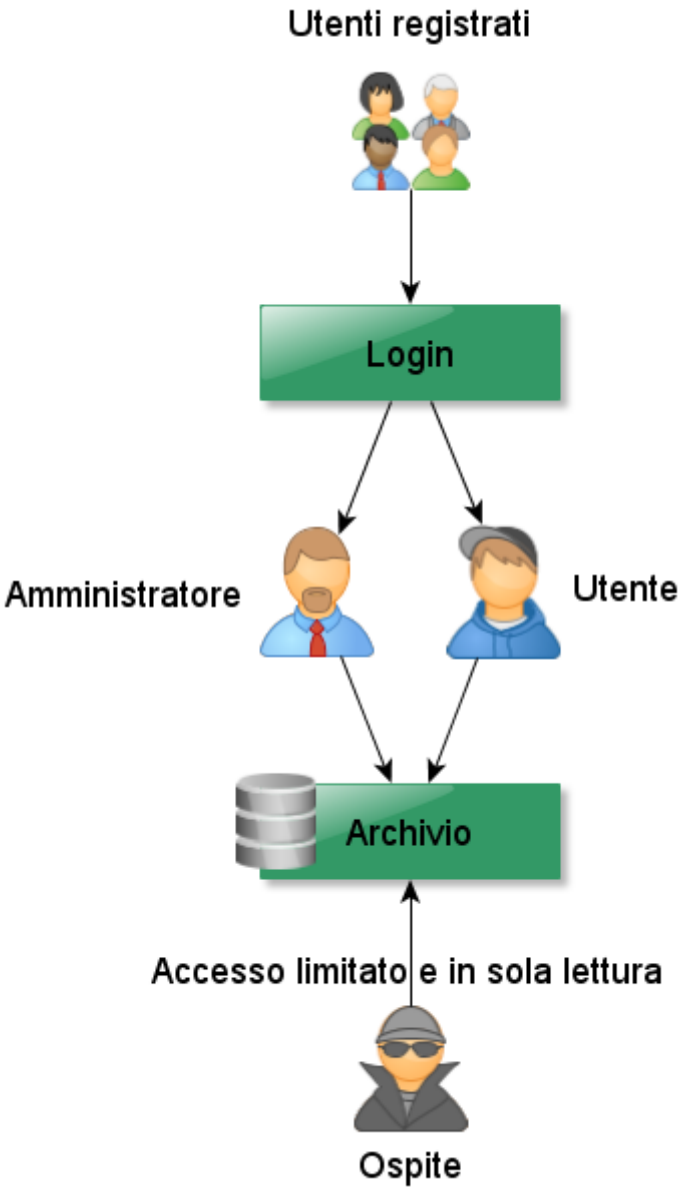
Sistemi che compongono l'applicazione: a noi servirà il login, i sottosistemi di ospite, utente registrato, amministratore e l'archiviazione dei messaggi, discussioni, registrazioni ecc.

- Login: sottosistema che si occupa della gestione dell'accesso al sistema, in base alla tipologia dell'utente seleziona il sottosistema di utilizzo;
- Archivio: sottosistema che si occupa della gestione degli accessi e degli utilizzi del DBMS del sistema;
- Ospite: gestisce il sistema con le ovvie restrizioni imposte dallo stato anonimo dell'utente ospite (impossibilità di replicare ai messaggi, inviare feedback. . .)
- Utente registrato: questo sottosistema gestisce tutte le azioni dell'utente autenticato, i suoi permessi e le informazioni del suo profilo
- Amministratore: gestisce sia tutte azioni esclusive dell'amministratore (come l'accettazione di nuove abilità) che quelle comuni all'utente registrato (come l'invio di messaggi)

L'esistenza dei sottosistemi dell'utenza di SWIM è giustificata dalla loro diversa interazione con l'archivio.

Gli amministratori potranno modificare le tabelle all'interno del DBMS, gli utenti registrati interagiranno con l'archiviazione dei messaggi e infine gli ospiti potranno visualizzare il contenuto di una parte dei dati senza poter apportare nessuna modifica.

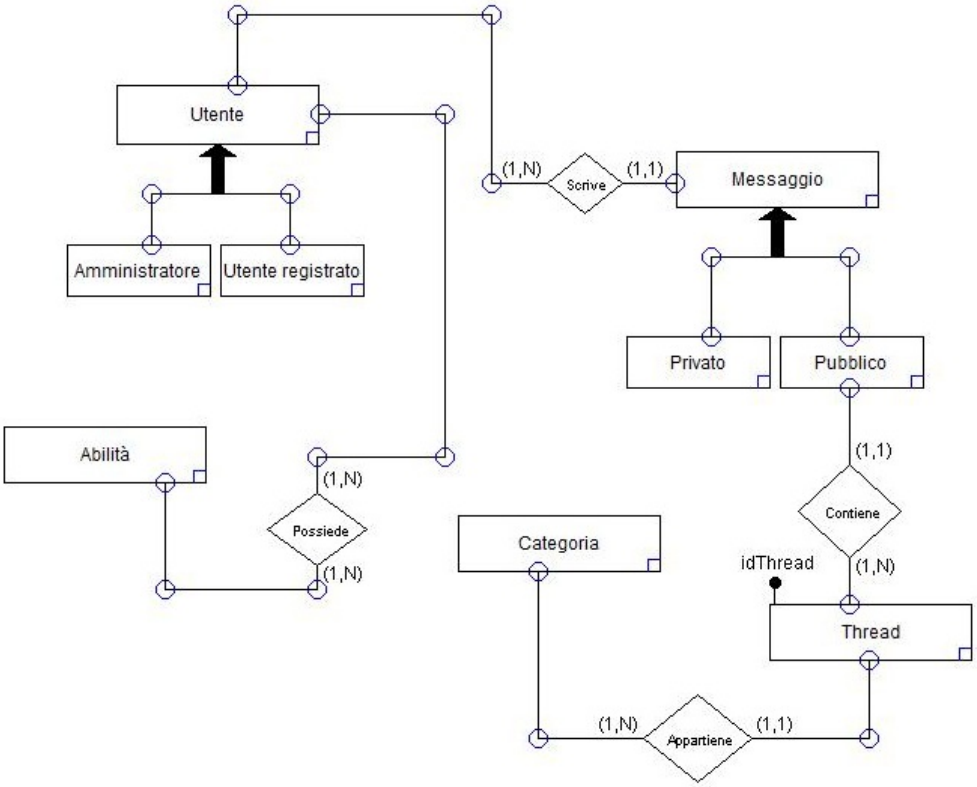
Per gestire tale meccanismo abbiamo scelto una soluzione che prevede di sfruttare la logica dell'applicazione, essa sarà in un certo senso intelligente e secondo l'appartenenza di classe renderà possibili le corrispettive interazioni con il database.


Utenti registrati
Login
Amministratore
Utente
Archivio
Accesso limitato e in sola lettura
Ospite

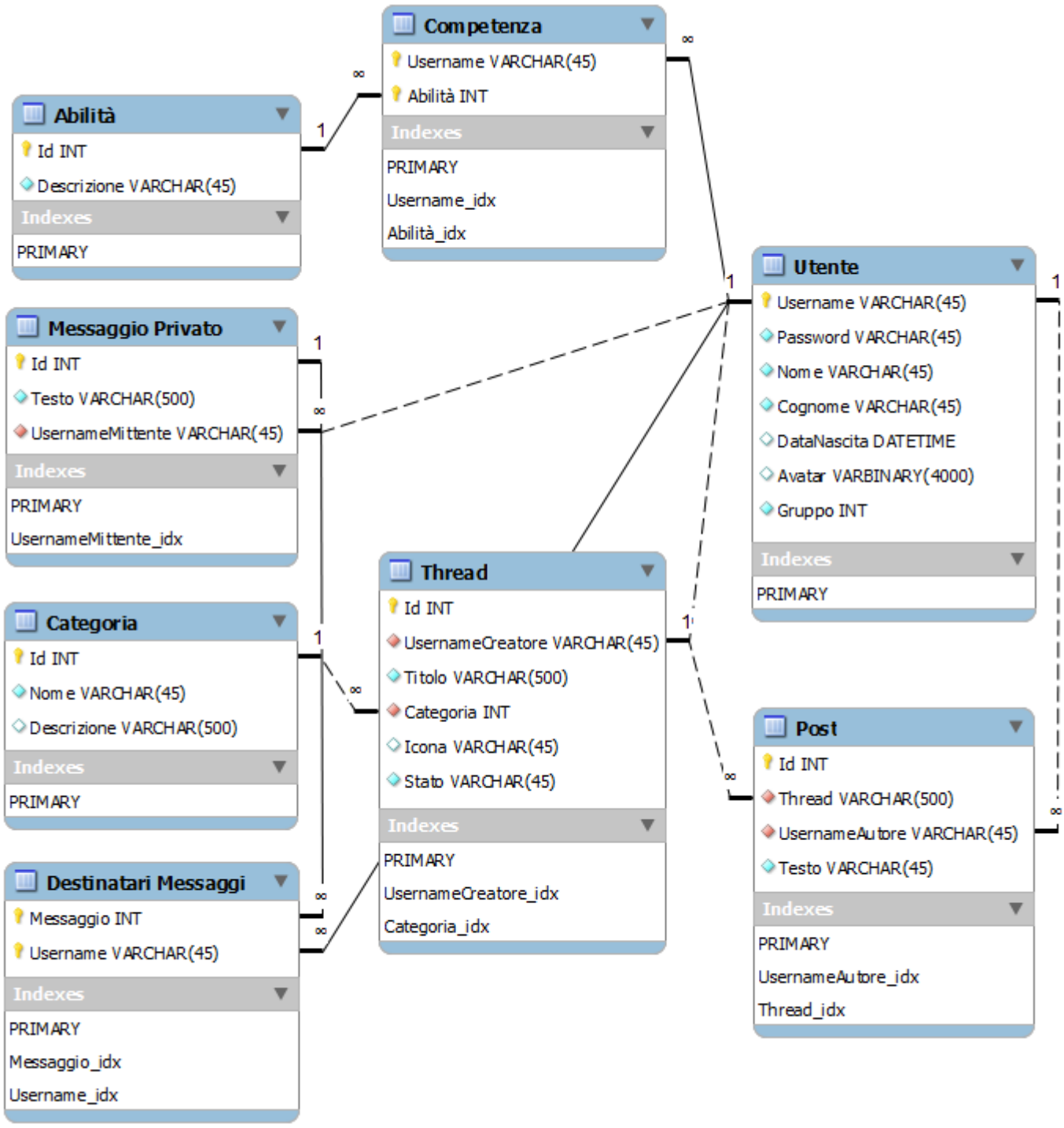
3 Design del Database

3.1 Modello concettuale

Lo schema concettuale del Database in oggetto è stato modellato con un diagramma ER riportante le varie entità, i loro attributi e le relazioni presenti.



3.2 Modello logico



Utente (Username, Password, Nome, Cognome, DataNascita, Avatar, Gruppo);
Competenza (Username, Abilità);
Abilità (Id, Descrizione);
MessaggioPrivato (Id, Testo, UsernameMittente);
DestinatariMessaggio (Messaggio, Username);
Post (Id, Thread, UsernameAutore, Testo);
Thread (Id, UsernameCreatore, Titolo, Categoria, Icona, Stato);
Categoria (Id, Nome, Descrizione);