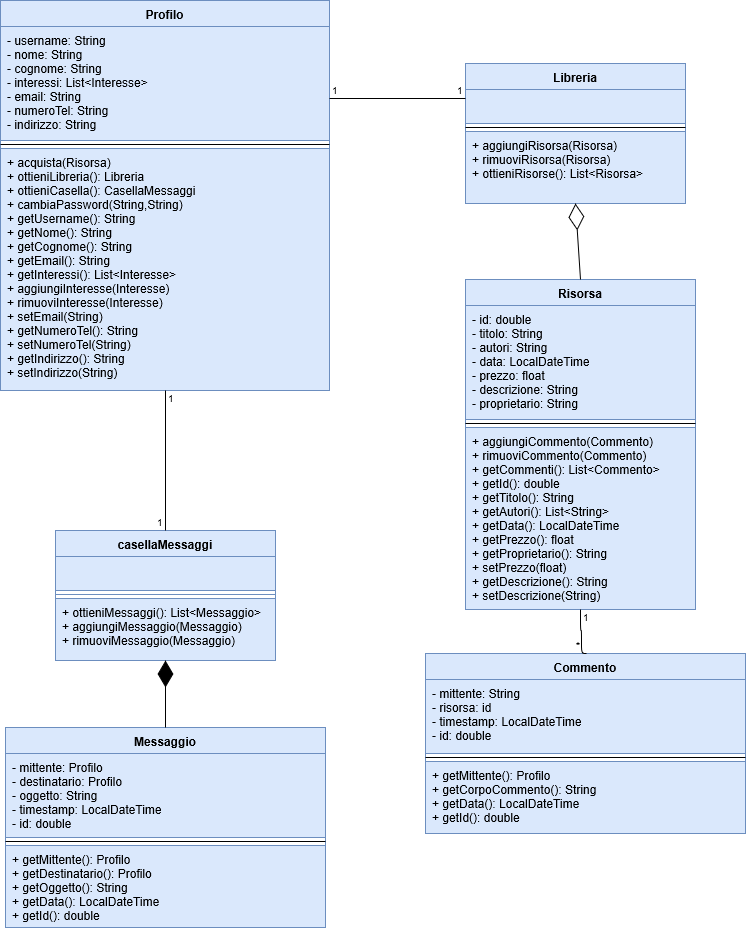
# Progettazione del dettaglio

Saranno riportati di seguito i diagrammi riguardanti il sistema

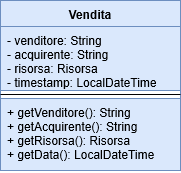
## Struttura

*Diagramma di Dettaglio: Dominio Risorse*



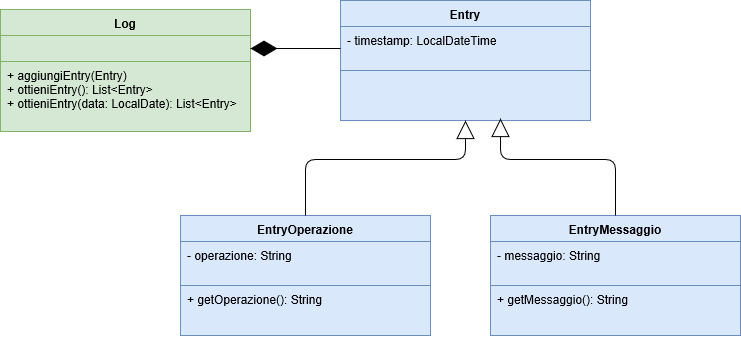
Le modifiche principali riguardano l’eliminazione dei metodi di invio commenti e messaggi, spostati all’interno dei controller. Il metodo *cambiaPassword* si occuperà come prima cosa di cifrare le password in chiaro, successivamente le spedirà e in caso di successo saranno effettivamente modificate. Lo username è da considerarsi unico all’interno del sistema così come l’id dei messaggi, commenti e risorse (non possono esistere per esempio messaggi o risorse con stesso id). Questa unicità ci permette di capire sempre chiaramente tutte le entità coinvolte senza inclusioni circolari (del tipo Profilo contiene Messaggio che contiene Profilo )

*Diagramma di dettaglio: Dominio Vendita*



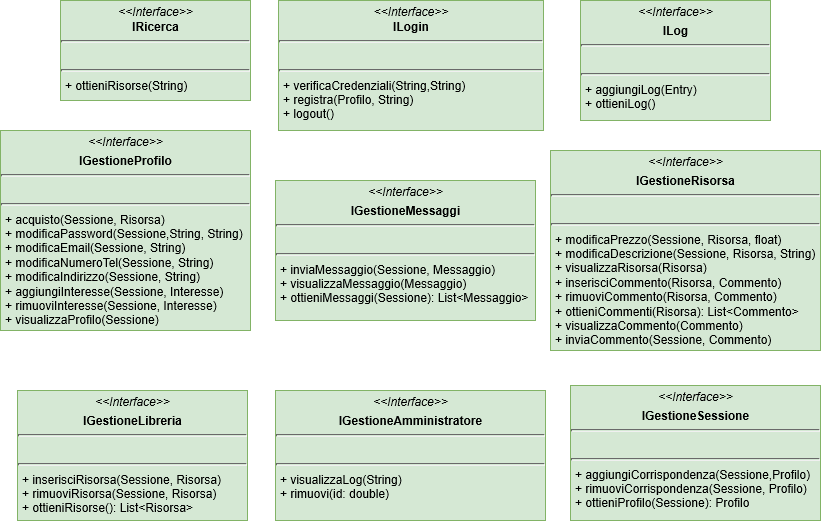
Resta praticamente uguale a quello dell’Analisi del Problema

*Diagramma di dettaglio: Dominio Log*



L’entità di log in questa fase diventa un controller che ci permette di interagire con il dominio delle entry. Il Controller Log qui presentato è un implementazione dell’interfaccia ILog mostrata di seguito

*Diagramma di dettaglio: Interfacce nel Server*



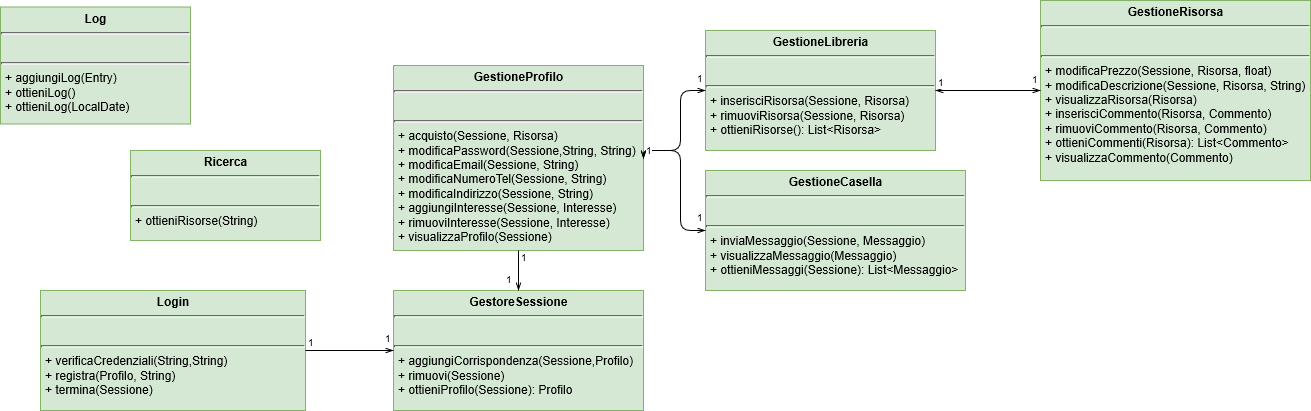
In queste interfacce è stato aggiunto un elemento non citato precedentemente, ovvero Sessione.

Il mondo *Servlet* possiede un oggetto chiamato *HttpSession* che ci permette di identificare stato della connessione ed altri dettagli aggiuntivi ricavati dall’interazione con l’utente. Questo oggetto fornisce un identificatore che è accessibile attraverso il metodo *getSessionId* . Uno stesso utente può acquisire più ID diversi in più interazioni con il sistema, ma fintanto che rimane connesso ne possiederà uno e sarà unico nel contesto degli utenti attivi.

Al contempo, ogni utente fisico è virtualmente rappresentato da un Profilo al quale si dovrebbero associare tutte le azioni che esegue sul sito. Considerato che ogni richiesta a una *Servlet* incorpora in sé l’oggetto *HttpSession*, si è pensato di mantenere solo lato *Server* le corrispondenze “Sessione - Profilo” (che si ricavano all’atto del login)*.* Questo metodo è utilizzato per fornire ulteriori meccanismi di protezione agli utenti, quali:

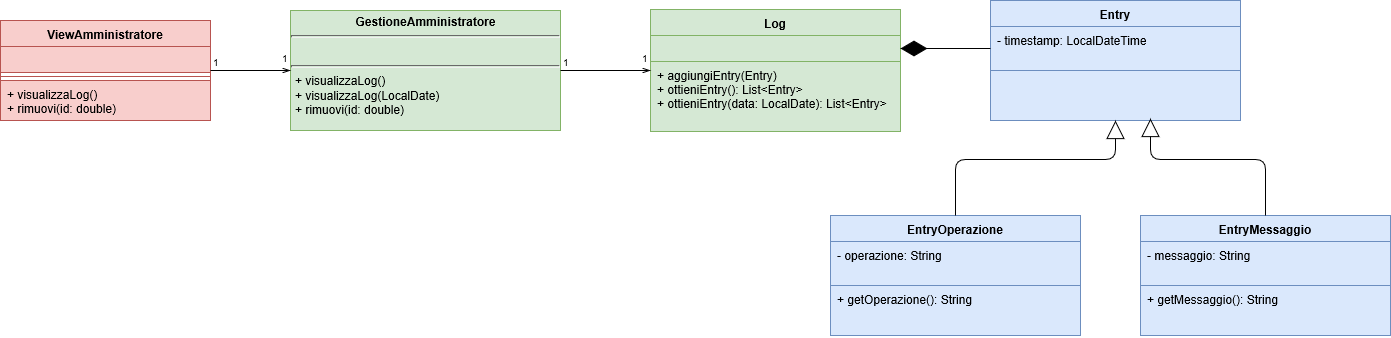
1. Il Server è utilizzabile solo da quei Client che posseggono una sessione valida (ovvero, appartenente a un utente che ha effettuato il login ).
2. La ridondanza introdotta nei metodi (per esempio in *inviaMessaggio( Sessione, Messaggio)* ) permette di verificare che la sessione produttrice di traffico appartenga all’utente che nominalmente lo produce (corrispondenza difficile da trovare vista la relativa dinamicità delle associazioni).

*Diagramma del Dettaglio: Server*

Alcuni appunti:

1. Il controller log è collegato a tutti i controller presenti all’interno del Server
2. Il controller ricerca è collegato solo al controller log. Si è pensato infatti che il miglior modo di ottenere dati dal DBMS sia affidare a lui la ricerca , e quindi il controller interagirà direttamente con lui attraverso query.
3. Il gestore sessione è collegato anche a GestioneLibrerie, GestioneCasella e GestioneRisorsa. Non sono stati rappresentati i collegamenti per motivi di visibilità.

*Diagramma di dettaglio: Amministratore*

 Il programma di amministrazione è del tutto separato dal sito, in concordanza con il vincolo di massima sicurezza. Tutto il programma con username e password risiedono sulla stessa macchina. Username e password saranno inseriti all’interno di un file cifrato e accessibile solo da programmi con privilegi di root.

Per quanto riguarda il ControllerAmministratore, possiede un accesso diretto al DBMS attraverso il quale può effettuare le rimozioni necessarie

*Diagramma di dettaglio: Client Utente*

