

Progetto di gruppo Ingegneria del software
2014/2015
ProShare

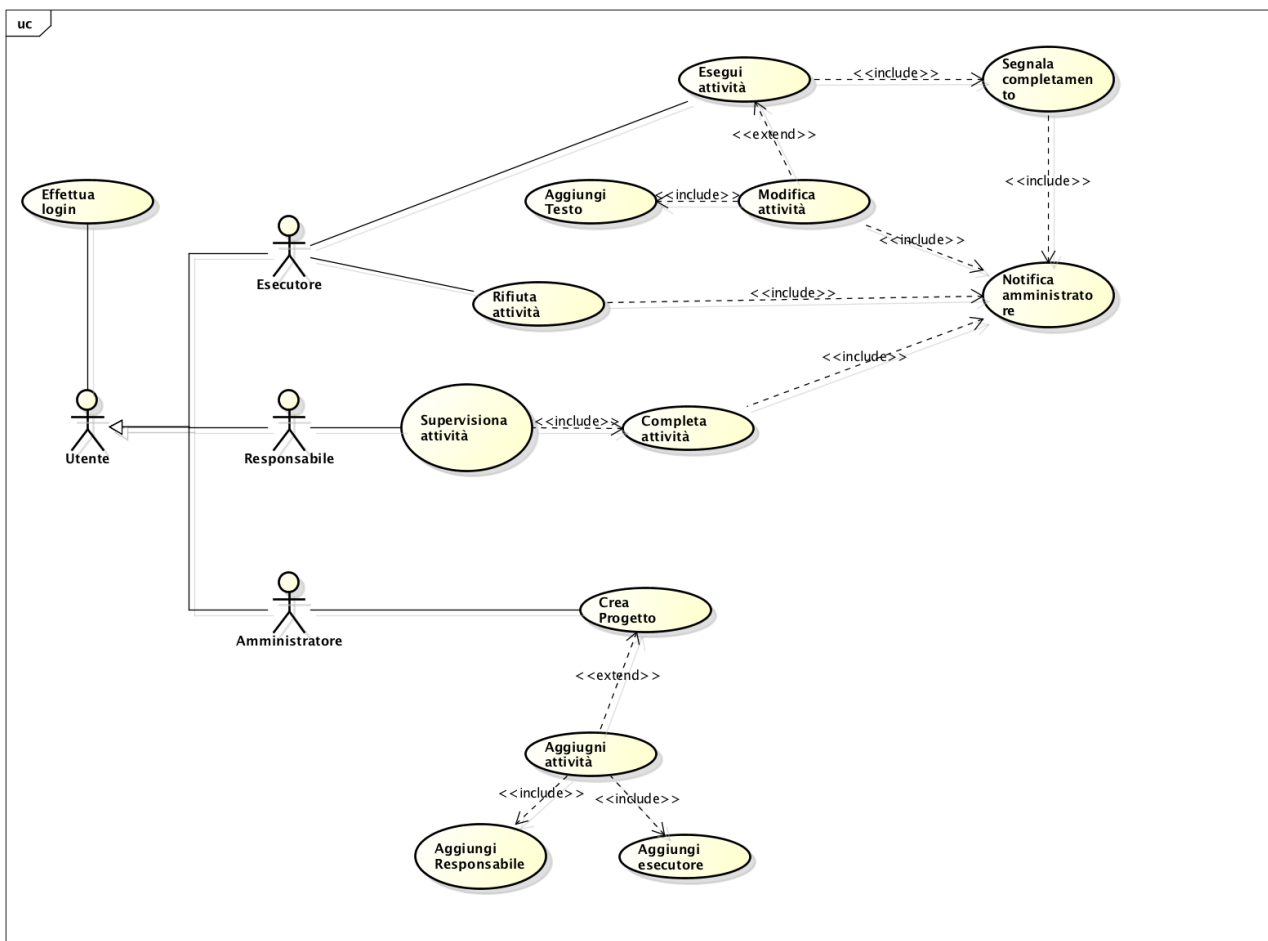
1.Introduzione

ProShare è un' applicazione che consente di creare progetti e condividerli con i propri contatti.
L'applicazione permette di:

- 1) Creare Progetti
- 2) Definire le persone che partecipano al progetto
- 3) Definire le attività che devono essere svolte all'interno del progetto
- 4) Aggiungere o cancellare gli amici

2. UML

2.1 USE CASE



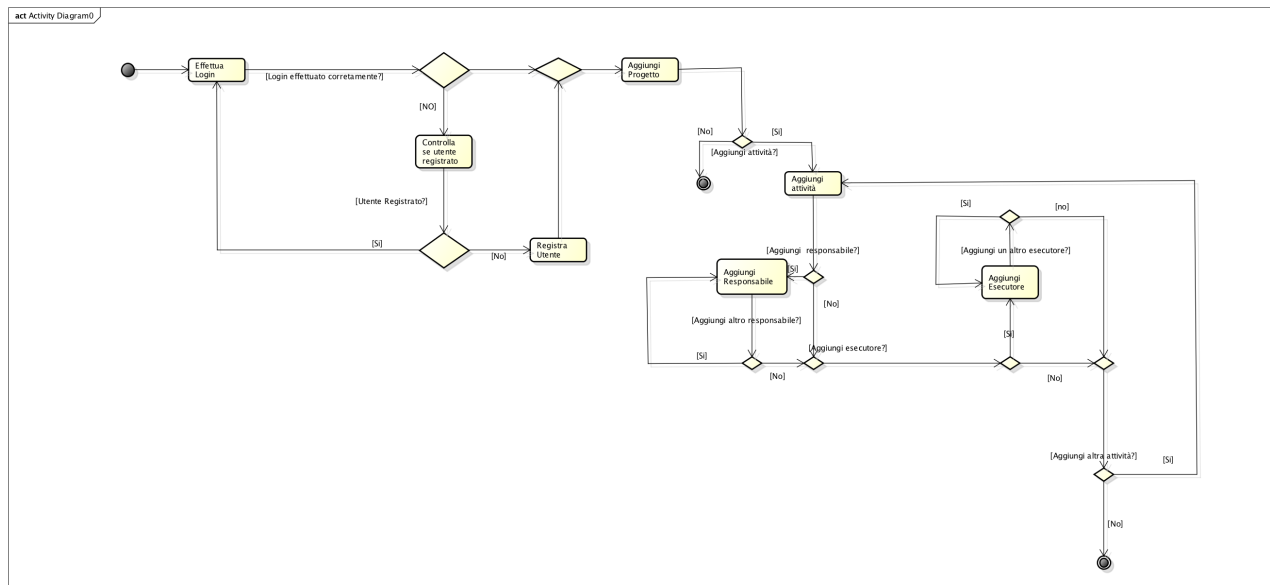
Nello use case vediamo cose un utente può essere : esecutore, responsabile, amministratore.
L'amministratore si occupa della creazione del progetto, delle attività ad esso correlate e di aggiungere ad ogni attività i vari responsabili ed esecutori.

I responsabili si occupano di supervisionare l'attività e nel caso di completarla.

Gli esecutori si occupano di eseguire le attività.

Ogni cambiamento sull'attività verrà notificato all'amministratore.

2.2 Activity Diagram



Nell'activity diagram vengono illustrati i passi per la creazione di un nuovo progetto.

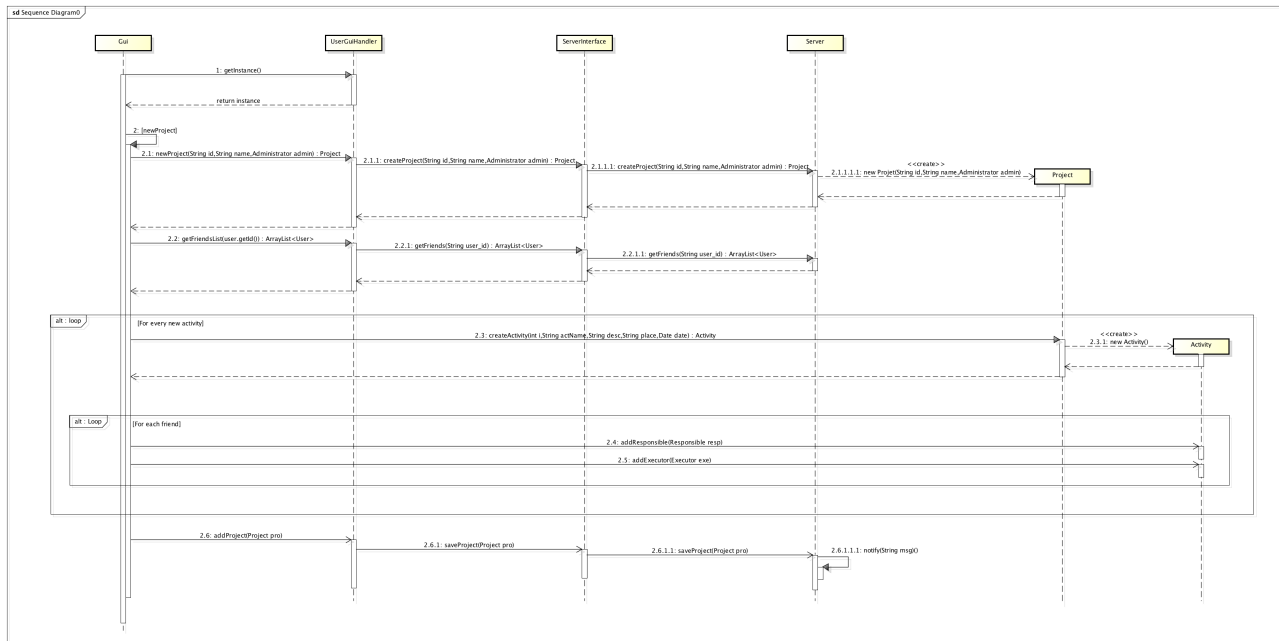
Inizialmente l'utente deve effettuare il login, se l'utente non effettua correttamente il login o non è registrato, deve ripetere o la procedura di login o registrarsi.

Una volta completato questo passo l'utente può aggiungere il progetto, se decide di non farlo, la procedura si arresta, altrimenti si procede oltre passando ad aggiungere un'attività.

Per ogni attività verrà richiesto di aggiungere sia esecutori che responsabili.

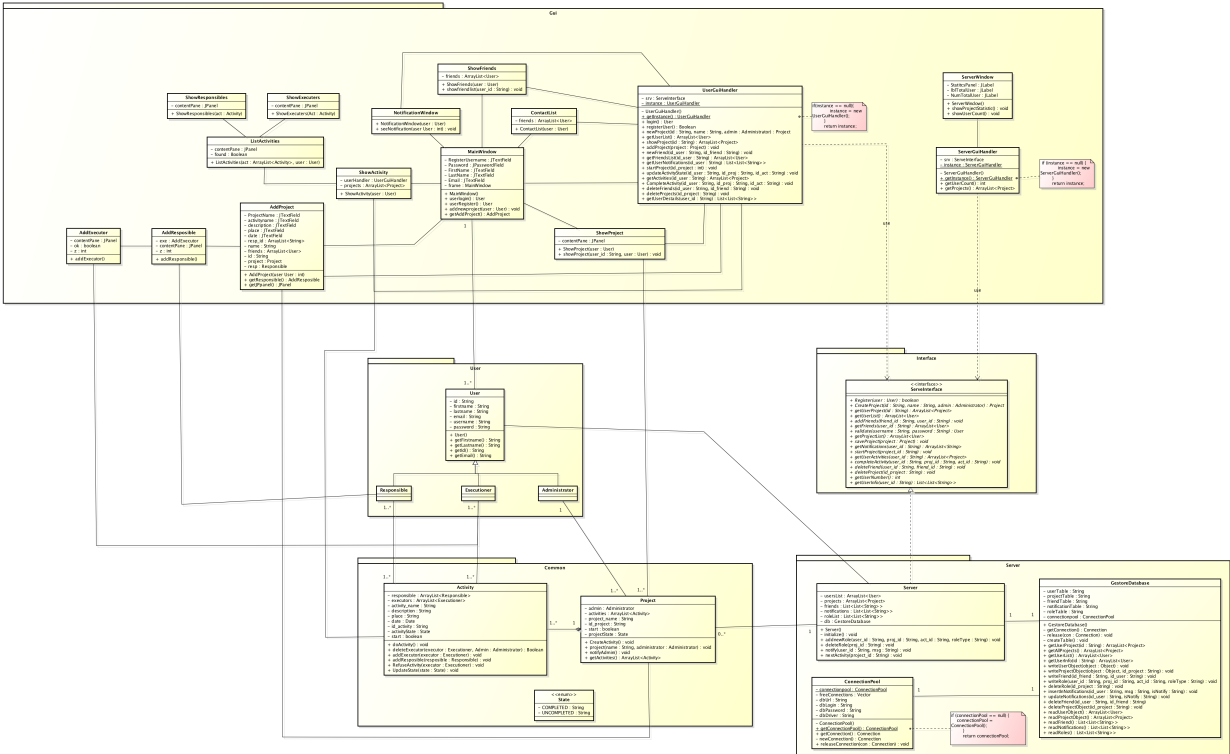
Una volta completata la creazione verrà chiesto all'utente se desidera aggiungere al progetto una nuova attività, nel caso affermativo si ripete la procedura di aggiunta attività, altrimenti la creazione è terminata con successo.

2.3 Sequence Diagram



Il sequence diagram descrive l'andamento nel tempo di una particolare azione. In questo caso è stato utilizzato per descrivere l'aggiunta di un nuovo progetto. L'utente inserisce i dati e il server crea e un nuovo progetto al client. Poi l'utente inserisce le attività e per ciascuna aggiunge i responsabili ed gli esecutori. Infine l'utente salva il progetto e il client chiama il metodo "saveProject" del server per salvare il progetto. Il server chiama il metodo "notify" per notificare tutti i responsabili e gli esecutori.

2.4 Class Diagram



Il class diagram fornisce un'idea generale sulle caratteristiche e le relazioni tra le classi da implementare nel progetto. Per facilitare la progettazione sono stati usati i seguenti design patterns: Singleton.

Nel class diagram si notano 4 package principali: Common , User, Interface,Server.

Nel package GUI sono contenute tutte le classi dell'interfacciamento con l'utente, attraverso la classe `userguihandler`, avviene l'interfacciamento con il server, dove sarà possibile creare i progetti e le attività collegate ad essa. Attraverso la classe `Server` tutti i dati creati tramite le informazioni inviate dalla classe GUI ci si interfaccia con il database in cui verranno salvati tutti i dati creati da ogni singolo utente.

3. JAVA

Le classi principali sono MainWindow.java e ServerWindow.java. MainWindow.java avvia il client e invece ServerWindow avvia il server. Al primo invio del server, nel database vengono create tutte le tabelle necessarie all'applicazione. Per gestire le connessioni al database è stata usata la classe connection pool. Il connection pool si collega al database mySql. I parametri del collegamento al database vanno messi nella classe "ConnectionPool", nel metodo "loadParameters()" (riga 39).

- dbUrl = "jdbc:mysql://localhost:3306/Social_Project";
- dbLogin = "root";
- dbPassword = "";

NB. Prima di avviare l'applicazione è necessario creare manualmente il database con nome "Social Project".

3.1 JUnit Test:

JUnit Test server per controllare i singoli pezzi di codice che svolgono le funzioni seguendo i requisiti. Sono stati implementati tre metodi di test:

- registerUserTest(): per controllare se l'utente viene registrato oppure no.
- userLoginTest(): per controllare se il login viene effettuato in modo corretto.
- singletonTest(): controlla che non esistano due istanze diverse di una classe singleton.

Sviluppatori:
Michele Carnovale
Manan Tariq