RELAZIONE ASSIGNMENT 4 PCD

**Semprini Luca - 0000854447**

**Riciputi Jacopo**

Questo assignment richiedeva di implementare un servizio di chat distribuito, prendendo come riferimento le architetture orientate ai servizi e reattive, e aspetti metodologici correlati.

-INTRODUZIONE-

…

Si è scelto di utilizzare il linguaggio Scala per l’implementazione dell’esercizio, integrandolo con la parte grafica scritta con JavaFX.

…

-SERVER-

-CLIENT-

Il modello del client è strutturato prevalentemente in maniera Actor-based. Si usa infatti il framework Akka per definire i principali comportamenti del sistema, dalle interazioni REST alla GUI.

…

-RestClient-

…

Gli attori responsabili del comportamento della GUI sono GUIActor e PreGUIActor, i quali si relazionano rispettivamente con le classi di controllo MainViewController e InitialWindowController. PreGUIActor si occupa principalmente di comunicare al RestClient l’intenzione da parte dell’utente di accedere al sistema, con uno username specificato; se lo username non verrà trovato all’interno del database, si richiederà all’utente se desideri creare un nuovo account o inserire un differente username. Nel caso la richiesta dell’utente abbia successo (nuovo account creato oppure login effettuato con uno username esistente), l’attore si occuperà di caricare la finestra principale; se la richiesta fallisce verrà mostrato un messaggio di errore e sarà possibile per l’utente riprovare.

Una volta caricata la View principale, verrà mandato un messaggio di attivazione al GUIActor, che, grazie a questo trigger, si occuperà di richiedere al RestClient la lista delle chat globali (quindi tutte le chat presenti sul database, non solo quelle a cui è iscritto l’utente loggato) e, una volta ricevute le esporrà nell’apposita lista, su cui sarà possibile fare una operazione di filtraggio se si desidera visualizzare unicamente le chat a cui si è iscritti. L’attore GUIActor interagisce quasi continuamente con il RestClient, in particolare nelle operazioni di creazione di una nuova chat, di rimozione di una chat, di adesione ad una chat esistente di cui non si faceva parte, di abbandono di una chat di cui si faceva parte e di invio di un messaggio su una chat selezionata. Quest’ultima operazione richiede l’intervento di un ulteriore attore (ChatActor), che è wrappato all’interno della classe di modello ChatWrapper, e che gestirà l’invio dei messaggi sulla chat e la notifica di eventuali errori.