Sistema CRM

Consegna 3: implementazione in Java

Abbiamo realizzato il progetto sfruttando come richiesto la libreria RMI di Java per la comunicazione col server remoto. Il nostro progetto è dunque composto da due parti principali, Server e Client, realizzati in due progetti Eclipse differenti che includono entrambi un progetto *Shared* nel *Build Path*. Quest'ultimo progetto include le interfacce richieste per il funzionamento di RMI e le classi condivise tra client e server, che implementano l'interfaccia *java.io.Serializable* per permetterne il passaggio delle istanze fra client e server.

Il server

Il Server ha due compiti principali:

- La gestione dei client connessi e degli utenti loggati: tramite un *HashMap* il server associa ogni client registrato all'istanza *Employee* del dipendente che ha effettuato il login; controlla che un utente non si connetta in contemporanea su due client differenti.
- La connessione al database per la gestione dei dati: il Server riceve le richieste dal client e si occupa di chiamare le query sul database tramite la classe singleton *DBManager*.

Il progetto Shared

Il progetto *Shared* è strutturato nel seguente modo: sono presenti le due interfacce *CRMServerInterface* e *CRMClientInterface.* La prima estende le interfacce implementate dai vari gestori presenti sul server, ed è implementata a sua volta dalla classe CRMServer. La seconda è utile solamente per passare al server l'istanza di *CRMClient,* dato che in essa non esiste alcun metodo (non esistono chiamate server->client). Sono presenti inoltre tutte le classi le cui istanze sono utili sia a client e server, come ad esempio *Employee, Customer, Event.* Infine sono presenti le *Exception* che il server lancia al client in caso di errori.

Il Client

Per quanto riguarda il client abbiamo realizzato l'interfaccia grafica utilizzando Swing. Sono presenti 3 diverse finestre: la finestra di login, la finestra di gestione cliente e la finestra di gestioni impiegati. Inoltre esiste una finestra di dialogo resa disponibile all'impiegato per modificare i propri dati personali.

Gestione delle Exception

Nel corso della realizzazione del progetto abbiamo dedicato particolare attenzione alla gestione delle Exception, gestendo quelle già generate durante l'utilizzo delle librerie Java (SQLException, RemoteException eccetera) ma anche creando delle Exception lanciate in caso di situazioni particolari, come password errata, dati inseriti non validi. Le Exception in arrivo dal database vengono gestite sul server che a sua volta le rilancia al client. Quest'ultimo le gestisce nell'interfaccia grafica mostrando una finestra di dialogo con i dettagli sull'errore e causando la chiusura del programma nel caso in cui un errore impedisca al programma di continuare a funzionare normalmente.

Testing – Junit

Per quanto riguarda il testing abbiamo deciso di fare un test sull'Exception che riguarda il tentativo di esecuzione di azioni da parte di utenti che non ne hanno l'autorizzazione. Nella fase di *setUp* viene avviata una connessione al server, dopodichè vengono testati alcuni metodi in circostanze in cui dovrebbero lanciare l'exception *NotAuthorizedUserException.*