

Programming Techniques For Distributed System

Project: *Volley Transfer*

Volley Transfer nasce con l'idea di poter essere una piattaforma per atleti e dirigenti pallavolistici (ma facilmente adattabile ad altri sport) che possa permettere l'incontro tra i primi che cercano squadra e i dirigenti che hanno bisogno di giocatori per completare il roster della propria società.

Dopo aver avviato il client, il programma permette di registrarsi al "social". Durante la registrazione viene chiesto se si vuole realizzare un profilo *Athlete* o *Manager*. Una volta completata la registrazione bisognerà loggarsi attraverso user e password impostati durante la registrazione.

A questo punto sarà possibile modificare i propri dati inseriti sulla piattaforma e avere funzionalità specifiche in base alla tipologia di profilo.

Per gli atleti sarà possibile modificare il proprio curriculum sportivo per permettere ai dirigenti di capire con che tipologia di atleta avranno a che fare.

Per i dirigenti sarà possibile cercare tra gli atleti registrati filtrandoli per ruolo, età o provincia di residenza.

Il funzionamento:

L'applicazione utilizza una comunicazione tramite socket.

Il server all'avvio carica in memoria gli utenti registrati alla piattaforma da un file che viene periodicamente aggiornato ogni 60 secondi. Timer che può essere modificato in base alle esigenze.

Ogni utente connesso viene gestito attraverso un thread con un client Manager dedicato.

Il client, in fase di registrazione, dopo aver raccolto i dati utente, si assicura che l'username non sia già presente in memoria e quindi invia il nuovo profilo al server per essere memorizzato.

In fase di login invece scarica le informazioni dell'utente in locale e quindi attua tutte le modifiche eventualmente necessarie. Al logout i dati sul server e sul client vengono sincronizzati.

Classi/Librerie java utilizzate:

java.io.*;	java.time.LocalDateTime;
java.util.*;	java.text.SimpleDateFormat;
java.net.ServerSocket;	java.security.NoSuchAlgorithmException;
java.net.Socket;	java.security.MessageDigest;
java.time.LocalDate;	java.math.BigInteger;
java.time.Period;	

Classi e Gerarchia:

Le classi realizzate per il progetto sono:

- ✓ *User*, classe astratta che rappresenta l'utente generico.
- ✓ *Athlete* e *Manager*, classi che derivano da *User*, e sono le reali implementazioni dei possibili utenti della piattaforma.
- ✓ *AthShow*, una classe che permette di inviare solo le giuste informazioni ai manager che fanno una ricerca.
- ✓ *City*, permette di organizzare le informazioni delle città di residenza degli utenti.
- ✓ *Season*, classe che rappresenta una stagione di un atleta. Usata per realizzare il curriculum sportivo.
- ✓ *Server*, *Client* e *Client Manager*, sono le classi per permettere il funzionamento client server del sistema.
- ✓ *UserHashMap*, gli utenti del sistema sono organizzati in una hashmap, questa classe permette di implementare la sincronizzazione e il salvataggio e caricamento da file.

(Vedi allegati in fondo per diagramma UML)

Riferimenti:

Il materiale è liberamente consultabile su gitHub: <https://github.com/andrealomby10/VolleyTransf>.

Class Diagram



