

### CHAT (uso del protocollo UDP multicast)

Realizzare un'applicazione che, mediante il protocollo UDP multicast, consenta di chattare con altri utenti connessi alla stessa "chat room".

Per comunicare tra loro, i client dovranno inviare e ricevere all'indirizzo multicast **224.0.0.1**, porta **10000**. La comunicazione dovrà avvenire soltanto tra i client collegati alla stessa chatroom. Le chatroom sono:

- SPORT
- MUSICA
- CINEMA

Ogni messaggio è rappresentato da una stringa; questa deve rispettare il seguente formato:

**#TEXT | <chatroom> | <utente> | <messaggio>**

nel quale il campo **#TEXT** è sempre presente.

Ad esempio:

**#TEXT | SPORT | filippo.bianchi | grande vittoria di Valentino!**

**#TEXT | CINEMA | andrea.verdi | Ho visto il film diretto da...**

Nel ricevere i messaggi, il programma dovrà filtrare quelli non validi o che non sono indirizzati alla chatroom dell'utente.

### Requisiti

L'utente deve poter:

- Entrare/uscire da una chatroom.
- Spedire e ricevere messaggi.

Segue il *mockup* di una ipotetica UI:

The mockup shows a chat interface with the following elements:

- Chat room:** A dropdown menu currently showing 'sport'.
- Nick:** A text input field containing 'pinturicchio'.
- Buttons:** 'Entra' (Join) and 'Esci' (Leave) buttons are located below the Nick field.
- Messaggio:** A section for sending messages, including a text input with the placeholder text 'Anche l'honda non scherza però!' and a '>>' button.
- Chat Log:** A list of received messages:
  - nickmanofreda: grande Marquez!
  - pinturicchio: il circuito era favorevole alla honda.
  - nickmanofreda: si va bé, ma lui è il più forte!
- Status Bar:** A bar at the bottom indicating 'Chat: sport - 224.0.0.1 - 10000'.

Nota bene: mediante un *textbox* si devono visualizzare sia messaggi inviati che ricevuti, comprensivi di mittente, in modo da preservare la cronostoria della conversazione.