# Requirements :

M : Mandatory

O : Optionnal

## Application :

* En langage JAVA (APPJAVA\_001\_M)
* Application de messagerie instantanée (APPJAVA\_002\_M)
* Ecrire le nom du pseudo d’une couleur selon le statut (Rouge pour Hors ligne par exemple)

## Client :

* Doit se trouver dans le package com.irc.client
* 50 lignes maximum
* Avoir un pseudo
* Créer une socket TCP pour se connecter sur la machine/port du serveur
* Reçoit adresse IP et port du serveur
* Affiche le message de bienvenue
* Envoie la réponse au serveur
* Changer de statut (En ligne, absent, Hors ligne, Occupé)

## Serveur IRC:

* Doit se trouver dans le package com.irc.server
* Vérifications des pseudos afin d’éviter les doublons
* Envoie un message de bienvenue
* Envoyer la liste de tous les connectés
* Envoyer la liste de tous les clients connectés
* Faire passer chaque message provenant du client au serveur, qui à son tour l’envoi à tous les clients connectés

## IHM :

* Une fenêtre d’identification
* Identification utilisateur ou administrateur
* Une fenêtre de messagerie
* Un bouton « Connect »
* Un bouton «Send »

C:\Users\ROMOLI\Downloads\Untitled Diagram.png

# Fonctionnement

Connexion serveur clients par socket TCP et une base de donnée PostgreSQL.

# Rôles :

## Administrateur :

* Client normal +
* Kicker
* Fermer la discussion
* Déconnecter le serveur
* Se connecte avec un mot de passe

## Client :

* Se connecter au serveur
* Envoyer des messages
* Se déconnecter du serveur
* Affiche l’historique de la discussion
* Se reconnecte automatiquement au serveur en cas de crash de celui-ci

## Serveur :

* En écoute des ports
* Envoyer un message d’accueil au client
* Envoyer la discussion lors de la connexion
* Renvoyer des messages vers tous les clients connectés

# Functionnality

* The user can log in
* The user can log out
* The user can shall be able to broadcast message in conversation
* The user can consult the historic of the conversation (10 messages)
* The user can his status (Offline, Absent, Online)
* The user see the other clients connected on the discussion
* The Admin log in with password
* Admin can stop the server
* Admin can kick out a client
* Admin can delete a message of the conversation
* A Traceability can be consult by an Admin

# Deployment

* JRE 1.8 must be support by the OS machine
* OS Client : Windows 10 (VM parallel desktop)
* OS Server : Unbuntu 16
* JMS incubator-activemq-4.0
* Launch of the application by script .bat
* Server deployment with a tar file that contains: folders bin, script, conf, lib and a file ReadMe in which one explains the server installation.

# High Availability

* Several servers are available
* Data are saved in a PostgreSQL database

# Performance

* Low RAM consumption (see XMS and XMX)
* Number maximum of clients connected : 100

# Communication Protocol

# 

|  |  |  |  |
| --- | --- | --- | --- |
| Modifier | Category | Message | Description |
| + | u | *pseudonym* | add *pseudonym* to the conversation |
| - | u | *pseudonym* | kick *pseudonym* from the conversation |
| + | a | *pseudonym* | ask a connection as administrator |
| + | m | *pseudonym*|*message* | send a *message* from *pseudonym* |
| - | m | *pseudonym*|*message* | remove *message* said by *pseudonym* from the conversation |
| + | p | *receiver*|*sender*|*message* | send a private *message* to *receiver* from *sender* |
| ! | i |  | join the conversation, ask for list of connected user and message history |
| ! | o |  | connection accepted |
| ! | n | *message* | refuse connection with *message* reason |
| ! | k |  | keep alive frame |
| ! | l | *pseudonym* | change status of *pseudonym* to “online” |
| ! | b | *pseudonym* | change status of *pseudonym* to “busy” |
| ! | f | *pseudonym* | change status of *pseudonym* to “away” |
| ! | c | *command* | send *command* to interpret to server |

The character “|” is used as separator mark to identify fields in the frame