Project: Chatting App Author: Jiyoon Seo – June 2015

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case (ID number) | Steps | Expected Result | Actual Result | Pass/Fail |
| Access the Chatting Room (01) | 1. Go to initial page (spa-ui.html) and specified port (7000) | Open with full-screen and prompt user enter a user name | Same as expected result | P |
| Enter Chatting room (02) | 1. A user enters unique ID (not used by other users in the chatting room) 2. Click ‘Enter the chatting room’ | User can go into chatting room and ready to enter messages | Same as expected | P |
| Enter messages (03) | 1. User can enter messages 2. Click ‘send’ button | User can enter messages and the message is shown by the user and other people in the room | Same as expected | P |
| Leaving the chatting room (04) | 1. The user leaves the room by just click ‘X’ on the browser. | As soon as a user leaves the room, other users don’t see the user ID from the list | .. | P |
| Sending private messages (05) | 1. Type ‘/p [user id] [message]’ | Only the sender and recipient should be able to view the message | .. | P |
| Save public messages in the chatting room in database (06) | 1. Send messages to the chatting room 2. Leave the chatting room 3. Re-enter the room again | See previously made messages in the room | .. | P |
| Delete messages sent by the user(07) | 1. Send messages to the chatting room 2. Click ‘Delete’ button | Messages by the user should be deleted from database | .. | P |
| Generate real-time in JavaScript – local test(08) | 1. In local developing environment, enter some messages 2. Check if the time after message is real-time in your location | After message, it shows real time in your local area | .. | P |
| Generate real-time in JavaScript – server test; online (09) | 1. After install and setup server, upload files 2. Iterate all the steps in the test case ID 08 | After message, it shows real time in your local area | Instead of real-time, random time has been generated by server system | F |
| Install required files in server and run | 1. Install required setup files (node, express, socket, mongo) 2. Upload server side and client side files | Working online properly! | .. | P |