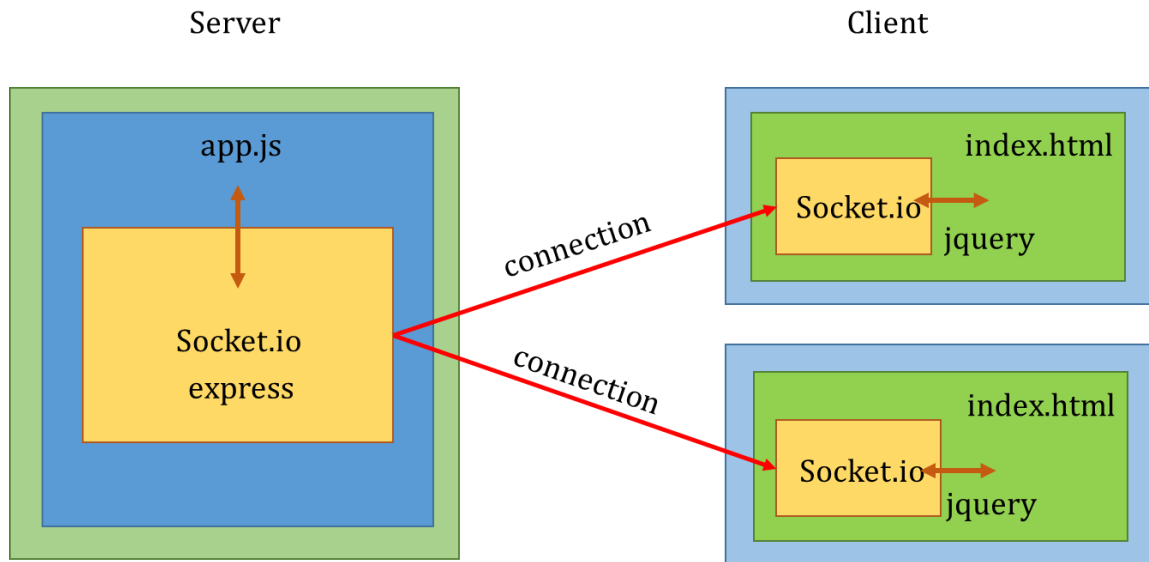


嵌入式系統實驗一

第四組 B02901146 蕭智遠 B02901117 蘇健智

1. 系統架構：

系統架構如下圖所示。Server 端用 Javascript 來撰寫所需要執行的程式，並利用 socket.io 和 express 來建立 HTTP 連線和雙向溝通的機制。Client 則同樣是利用 socket.io 來與 server 端進行連線溝通，並利用 jquery 來做為 javascript 取用 html 網頁檔所需資料的介面。



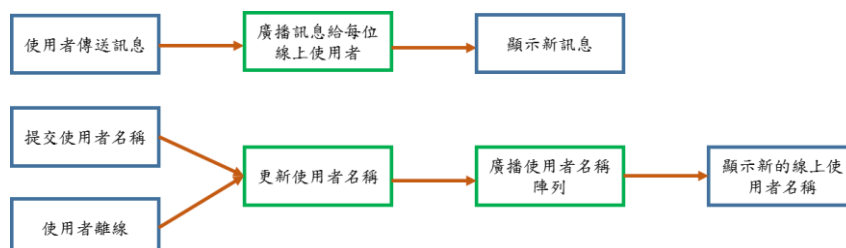
Server 端的行為大致上可再細分為下面三種情形

- (1) 新使用者加入 → 將使用者名稱放入儲存使用者名稱的陣列中，並將使用者名稱陣列廣播給每一位在線上的使用者。
- (2) 使用者傳送新訊息 → 將訊息與傳遞該訊息的使用者名稱寫入一 key-value pair 陣列中，並將該陣列廣播至每一位在線上的使用者。
- (3) 使用者離線 → 將離開的該使用者名稱從存放使用者名稱的陣列中移除，並把更新後的陣列重新廣播至每一位在線上的使用者。

Client 端的運作模式可分為以下種情況：

- (1) 提交使用者名稱 → 將使用者名稱傳送至 server 端。
- (2) 使用者輸入訊息 → 將訊息傳送至 server 端。
- (3) 使用者接收到新訊息 → 將使用者訊息和接收到訊息的時間顯示在螢幕上。
- (4) 使用者接收到使用者名稱的陣列 → 將陣列內的每一位使用者顯示在螢幕上。

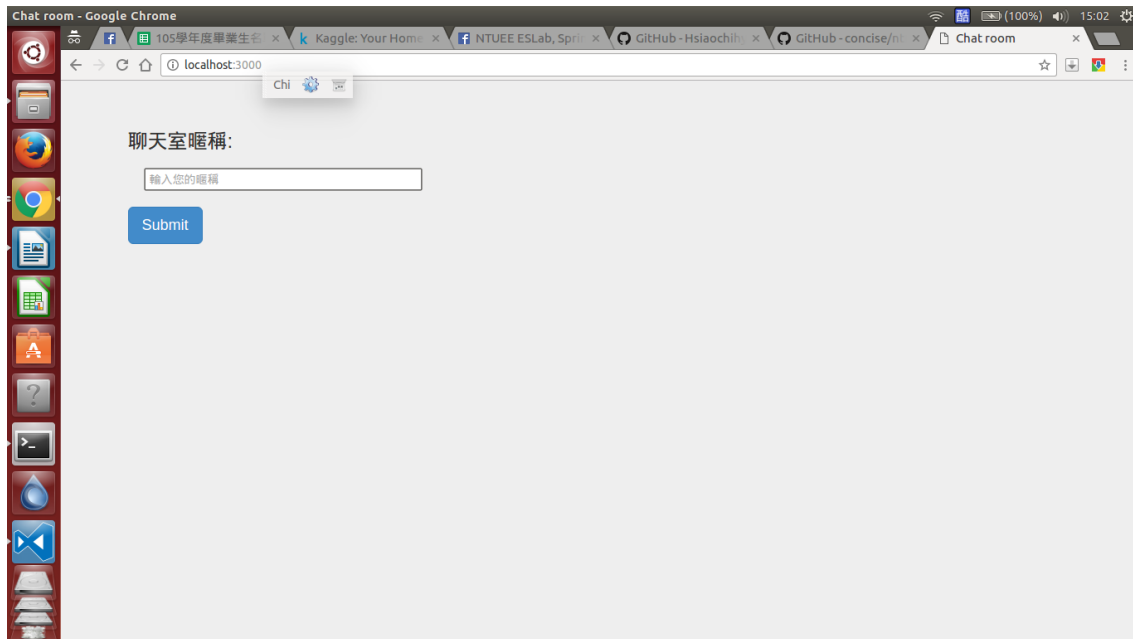
下圖為 Server 與 Client 端運作的流程圖，其中藍框為 Client、綠框為 Server



2. 功能介紹：

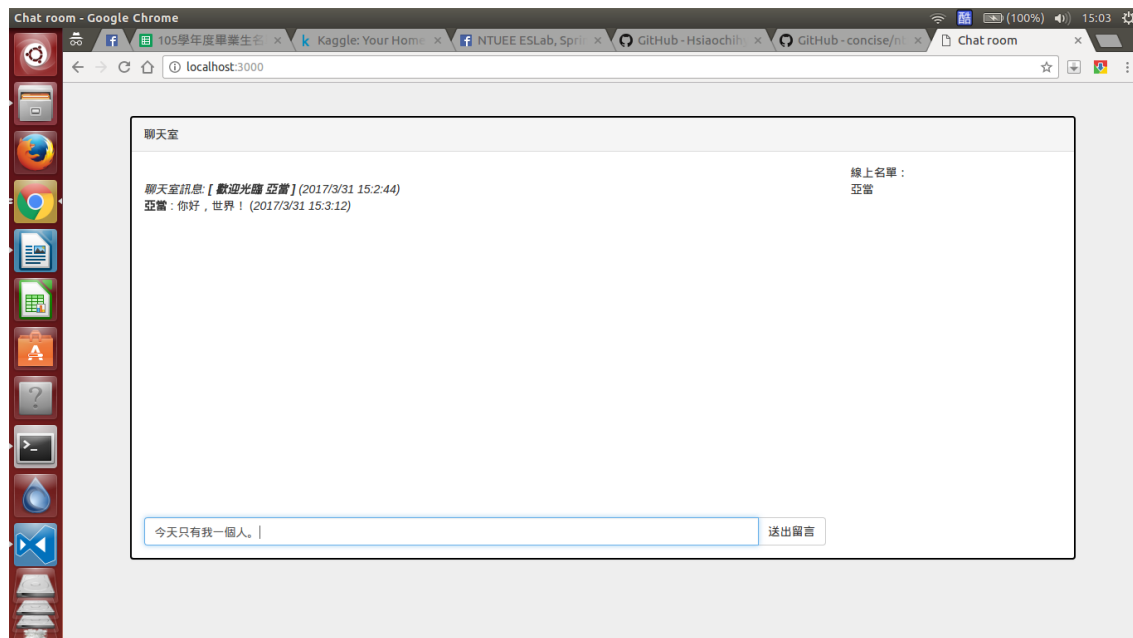
(1) 開始畫面：

輸入暱稱，按下 Submit 即可登入。



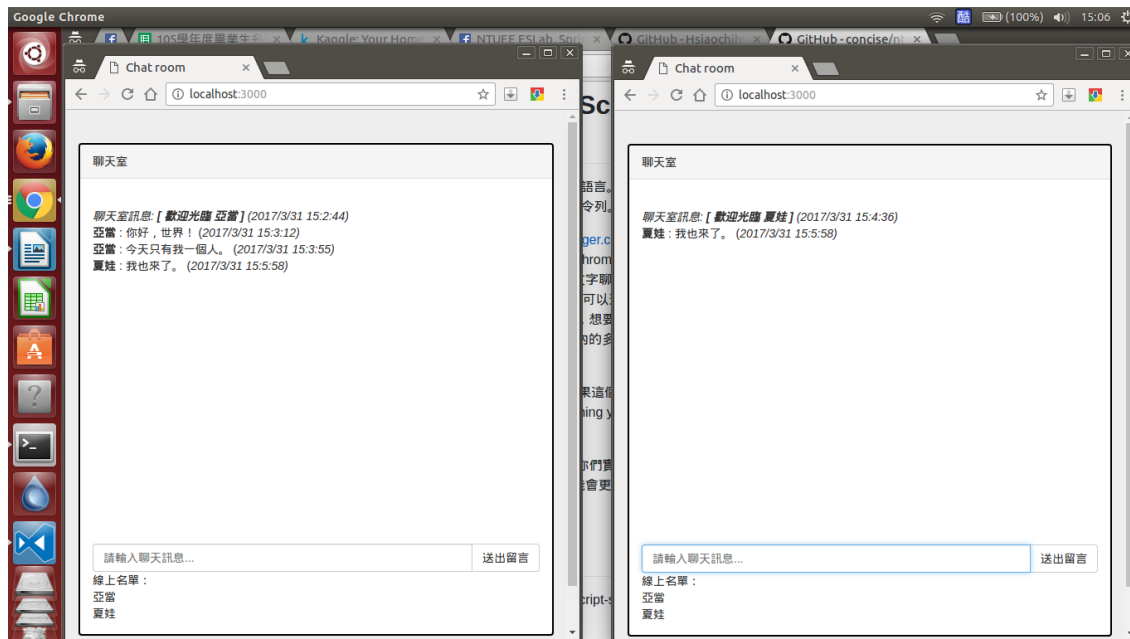
(2) 對話框界面：

右側是目前在聊天室中的成員名單，弱瀏覽器寬度變窄會轉到下方顯示。
在下方輸入框輸入文字，輸入完畢按下發送即可送出，



(3) 第二位成員加入聊天室：

下方可以見到現在有兩位使用者在聊天室裡。
新的使用者並不會看到前面發送的訊息。



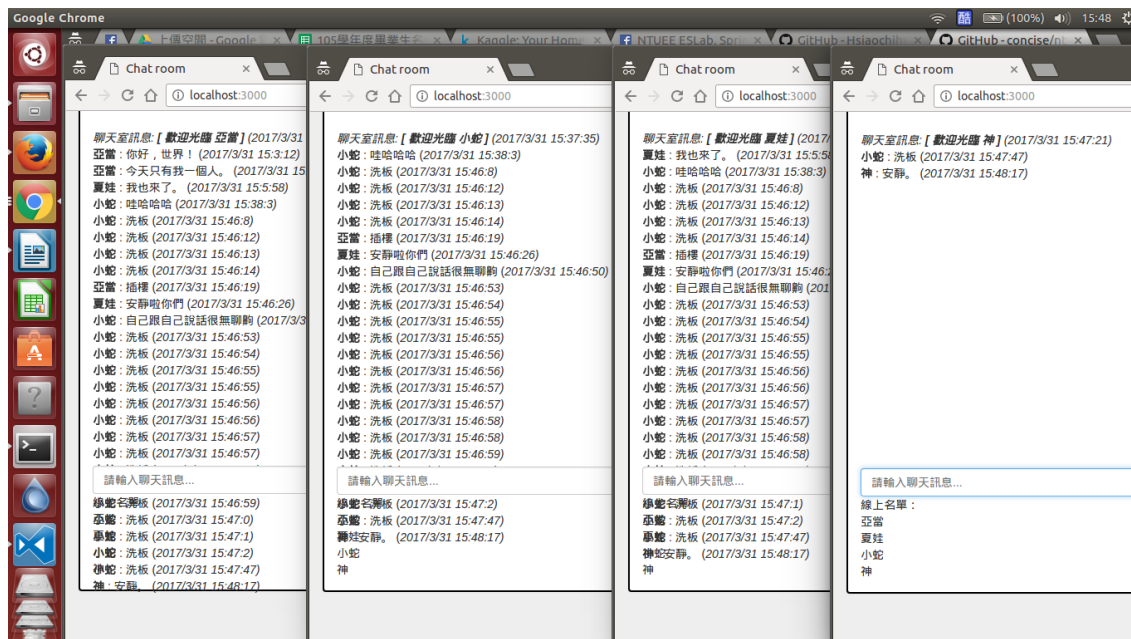
(4)再加一位也不是問題:

輸入的訊息會立即同步到三位使用者的視窗中顯示。



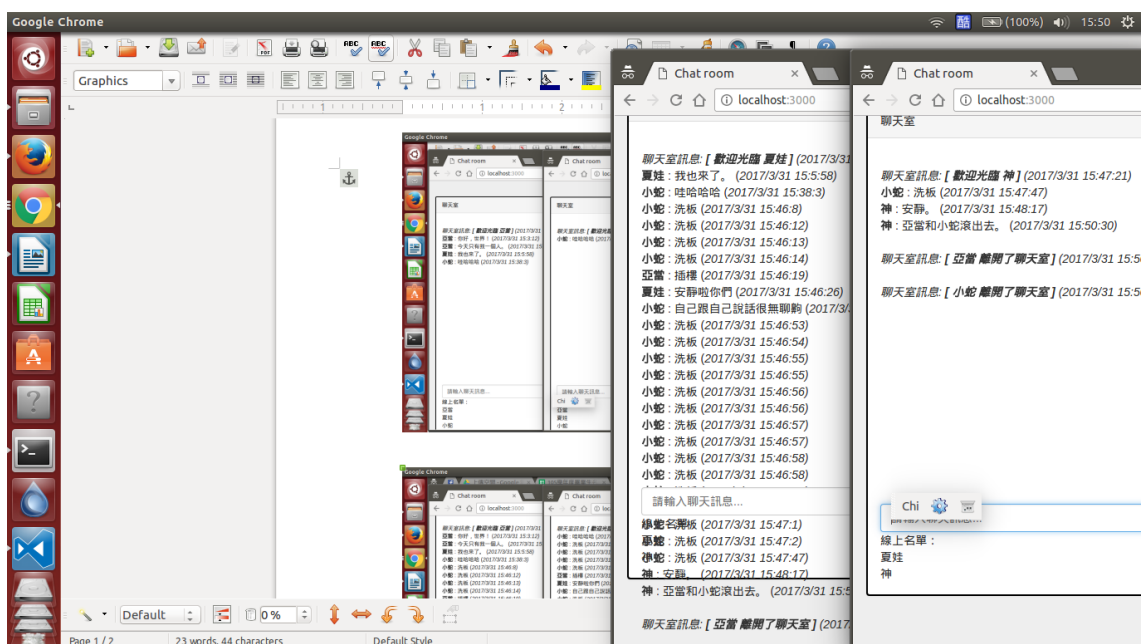
(5)再加入一位使用者:

目前沒有滾輪功能, 訊息會一行行往下疊, 超出的部份就看不到了。



(6)兩位使用者退出:

所有人都會看到有人退出的訊息。與之相對的，有人加入聊天室時並不會直接看到加入的訊息，只能從線上名單裡看到有新的人。



(7)接著嘗試以相同的名稱登入

第一張圖是第一位登入的亞當離開聊天室前的畫面

第二張圖是第二位登入的亞當離開聊天室前的畫面

當使用者(亞當)仍然在線時，任何人以「亞當」登入都會沒有名字，另外雖然沒有截圖，沒有名字的登入者在線上名單裡看不到「亞當」。

直到使用者(亞當)離開後，下一個人以「亞當」登入才會正常的叫做亞當。

