Embedded System Lab 1 :Messenger

Developer: 電機四 陳家暄 B02502061
電機四 毛學涵 B02901045
電機四 劉德元 B02208028

一、 主要功能

Front End

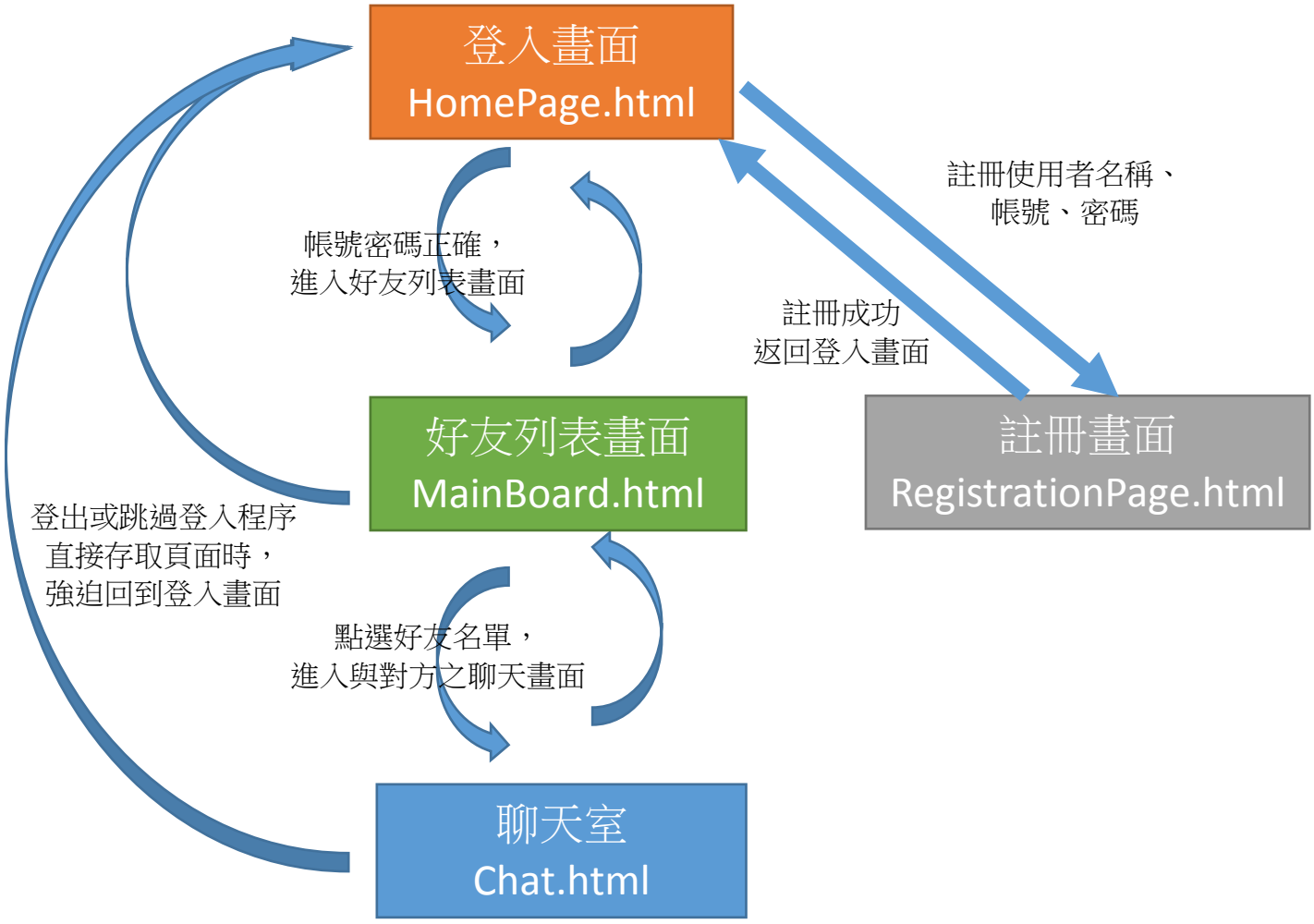
1. **好友名單：**可以在頁面輸入帳號加入其他人為好友，會確認對方帳號是否存在，加入同時對方帳號也會自動加自己好友。成為好友後，對方會出現在下方可聊天名單中。
2. **聊天訊息 preview：**在好友名單上，顯示和此好友最近一筆的聊天內容，過長的部分最後面以…取代，有未讀的新訊息會以黑粗體表示。
3. **顯示上線：**在聊天室以及好友名單上，都有綠點顯示好友是否在線。
4. **一對一聊天：**此為主要的聊天功能，進入聊天頁面會顯示最近五筆聊天內容，在對話框中按下 Enter 訊息會自動傳出，如果需要手動換行可以按 Shift+Enter。
5. **顯示已讀：**所有傳出去的訊息都會顯示是否已讀，會顯示在聊天視窗中或是好友名單的 preview，判斷的標準為對方是否在聊天視窗中。
6. **Public Chatroom：**此為公眾聊天室，進入時可以和所有也在裡面的人聊天。

Back End

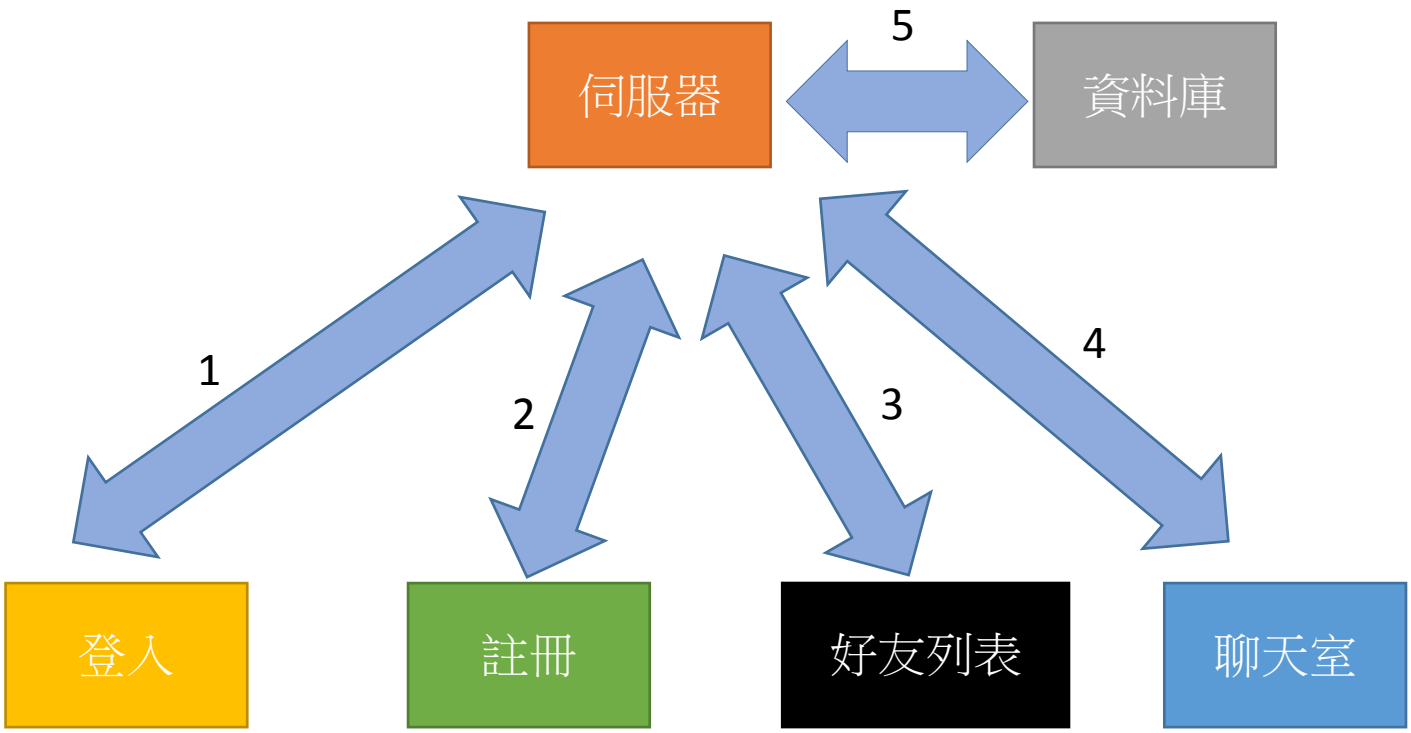
1. **歷史紀錄：**所有傳出去的訊息都會被存進 SQL，會紀錄每一筆訊息的發送者、接收者、內容、發送時間、以及是否已讀。
2. **帳號密碼系統：**SQL 中會紀錄所有使用者資料，包含使用者名稱、帳號和密碼，每次註冊時會檢查是否已重複，登入時會驗證資料是否正確。
3. **Router：**當外部 Client 端連線上伺服器 IP 時，會有 router 決定 response。
   a. 當連線到根目錄時，回傳首頁給使用者。
   b. 當連線到 Chatroom、MainBoard 時，會先驗證使用者是否已登入，若登入即傳回正確頁面，若錯誤則跳出 Warning 並自動跳轉回登入頁面。
   c. 連線到其餘未定義頁面時，回傳 404 Not found。
4. **Cookie：**因為 WEB 的安全性需求，Client 幾乎無法讀寫任何檔案以至於缺乏記憶性，使得 Client 每次連到伺服器時，無法告訴對方自己的身分，Server 無法知道連線進來的用戶是新用戶還是已登入會員，我們用 Cookie 記錄 User 的帳號資料，並在進入 Chat 時記錄傳訊息的對象。

二、　系統架構

使用者介面


登入畫面
HomePage.html
註冊使用者名稱、
帳號、密碼
帳號密碼正確，
進入好友列表畫面
註冊成功
返回登入畫面
好友列表畫面
MainBoard.html
註冊畫面
RegistrationPage.html
登出或跳過登入程序
直接存取頁面時，
強迫回到登入畫面
點選好友名單，
進入與對方之聊天畫面
聊天室
Chat.html

Client/Server 互動



1. **登入：**

   於登入頁面的 html 中，將使用者輸入的帳號密碼存入 cookie 再透過 socket 傳送至伺服器，伺服器收到 cookie 後拆解出使用者欲用來登入的帳號密碼，再進入資料庫(Maria DB)中的 UserInfo table 查詢看看是否有找到相符的使用者。如果找到了，伺服器即回傳一個 permission=1 至 client；反之，則傳送 permission=0，client 再透過 permission 的值來判定是否讓使用者進入好友列表頁面並且儲存登入者的身分。在登入頁面也有存在一個按鈕讓使用者進入註冊的頁面，也就是下一段的內容。

2. **註冊：**

   於註冊頁面的 html 中，使用者輸入的使用者名稱、帳號、密碼三樣資訊會被檢查是否有任何一項是空白的。若是有至少一個是空白的，則跳出小視窗進行警告；否則即將資訊傳給伺服器，伺服器在資料庫中進行查詢，確定沒有重複的帳號存在後即將傳來的資訊寫入資料庫，並回傳一個成功的訊息，要不然伺服器會回傳錯誤訊息，讓使用者再試一次；client 接收到成功訊息後將頁面切換回登入頁面。

3. 好友列表：
    a. 進入頁面後，client 向 server 送出搜尋好友名單的請求，server 到資料庫的 Friends table 中選出所有使用者的好友，以及該好友是否在 socket 列表中(也就是上線中)，並回傳好友名單以及每一好友是否上線的資訊，client 接收到回傳的好友列表後動態新增 html 物件，顯示好友名單(帳號+小綠點)。
    b. client 針對每一個好友帳號分別向 server 送出預覽該帳號最後一筆訊息的請求，server 到 ChatHistory table 中選出 Sender=使用者且 Receiver=好友帳號、或是 Sender=好友帳號且 Receiver=使用者的所有資料，依照 Time 排序後，取出最新一筆回傳，client 再將訊息內容及是否已讀(利用 IsRead 判斷)顯示於好友帳號中。
    c. 輸入欲加入的好友帳號並按下新增按鈕時，client 將欲新增的帳號送至 server，server 到資料庫的 UserInfo table 中檢查是否有此帳戶，若有此帳號，則到 Friends table 中確認此帳號是否已成為使用者好友，若不是，則在 Friends table 中新增好友 pair，並通知 client 可以新增好友，client 便動態新增 html 物件，顯示新增的好友名單；否則回傳警告訊息，client 以小視窗顯示不成功的原因。同時 server 亦會通知被新增的好友，若對方也在好友列表畫面中，則可以即時收到此通知，顯示新的好友名單。
    d. client 接收到 server 傳來的已讀/上線/離線/新訊息通知，即時更新對應好友欄位的呈現內容。
    e. 點選欲聊天的好友帳號，將好友帳號存進 cookie 中 receiver 欄位，並切換頁面至聊天室頁面。

4. 聊天室：
    a. 進入頁面後，client 向 server 送出開始聊天的請求，server 便到 ChatHistory table 中選出 Sender=使用者且 Receiver=接收者、或是 Sender=接收者且 Receiver=使用者(使用者/接收者分別對應至 cookie 中存放的 account/receiver)的所有資料，依照 Time 排序後，取出最新五筆回傳，並檢查 receiver 是否在 socket 列表中(是否在線)，將以上資訊回傳給 client，client 便將聊天對象的上線狀態、是否已讀、及聊天內容呈現出來。另外，server 亦在此步驟中將 ChatHistory table 中所有 Sender=接收者且 Receiver=使用者(也就是對方傳給自己的訊息)中的 IsRead 設為 true，並通知接收者訊息已讀，接收者若在好友列表或聊天室頁面收到通知，便會更改對應好友的已讀狀態。
    b. client 收到他人傳來的訊息通知時，判斷此訊息的傳送者是否就是使用者正在聊天的對象，是的話便回傳一個已讀通知給 server，server 便至 ChatHistory 中將對應訊息更新為已讀，並且傳送已讀通知給傳送者；否則不做任何回應。收到他人傳

來的已讀通知時，一樣判斷傳送者是否為使用者聊天的對象，是的話在頁面中顯示已讀。

c. 輸入訊息並按下發送鍵後，將訊息送至 server，server 將該訊息新增至 ChatHistory table 中並預設為未讀，而後 server 向接收者發送有新訊息的通知，接收者若在好友列表頁面收到此通知，則將對應好友欄位中的訊息預覽更新；接收者若在聊天頁面，則如 b 點所述。

d. 公共聊天室則是將 receiver 設為 Public，在公共聊天中的訊息會被 broadcast 到其他 socket，所有在該頁面的使用者皆可接收到訊息。

5. **資料庫：**

使用 mariasql，分為三個 table：

a. UserInfo

此 table 有三個欄位：username、account、password，用來管理帳號登入、註冊，以及檢查帳號是否存在。

b. Friends

此 table 有兩個欄位：usr、friend，用來記錄所有好友配對，若 A 與 B 為好友，則 table 中會同時存有 usr=A, friend=B 以及 usr=B, friend=A 兩筆資料。

c. ChatHistory

此 table 有五個欄位：Sender、Receiver、Content、Time、IsRead，用來記錄所有在聊天室中送出的訊息。

6. **好友列表/聊天室共同功能：**

a. 使用者進入好友列表以及聊天室頁面時，都會先通知 server 已進入頁面，server 便會將使用者的 socket 加進 global 的 socket 列表中，並 broadcast 到所有連線至 server 的 socket，通知有用戶已上線。

b. 從頁面離開時，會通知 server 已離開頁面，server 便會將使用者從 socket 列表中移除，並 broadcast 至所有 socket，通知有用戶已離線。

c. 點選登出按鈕時，client 會將登入時存的 cookie 清空，並重新導向至登入畫面。

d. 進入頁面時，判斷 cookie 中是否有 account 的資訊，若無，則代表使用者尚未登入，重新導向至登入畫面。