PleChat 遠端視訊應用系統

PleChat Remote Video Application System

黃映綺、楊雅淇、莊侃絃、黃家郁、劉婕云

指導老師: 黃豐隆

國立聯合大學 資訊工程學系

苗栗市南勢里聯大 2 號

{U0724048,U0710015,U0714045,U0724042,U0724043}@gm.nuu.edu.tw

摘要

為了防範疫情擴散,地方政府會管制民眾的外出，繼而減少人們之間的接觸。於是人們只能利用通訊或視訊軟體來交流，但現在各種通訊或視訊軟體在使用上有其優缺點。因此我們希望開發出一款介面簡單明瞭且方便使用的通訊網頁，透過網頁即可開始即時傳輸訊息或視訊。

關鍵詞：即時通訊、視訊、跨平台。

Abstract

In order to prevent the spread of the epidemic, local governments will regulate people's go out and then reduce contact with people. So people can only use communication or video application, but now all this application have their advantages and disadvantages in use. Therefore, we hope to develop a simple and easy-to-use communication webpage, through which we can start real-time transmission of messages or video.

Keywords: Instant Messaging、Video、Cross-platform.

一、 前言

1.1 研究背景與動機

去年 (2020) 年初，由於嚴重特殊傳染性肺炎 (COVID-19) 爆發，迅速蔓延至世界各地。在世界各國不斷宣導避免接觸見面的情況下，尋找替代的溝通方式便是現在的一大課題。無論是一般民眾、商家、公司、國家機關，在這個時代中，利用網路來進行交流是人人都有的需求。至今，全球疫情仍然十分嚴峻、不太明朗，在無法控制疫情的情況下，只能藉由網際網路，於線上進行遠距離的通傳訊息、傳輸檔案資料、進行語音視訊通話會議、修習網路課程等方式以完成業務必要的溝通交流、知識的學習。

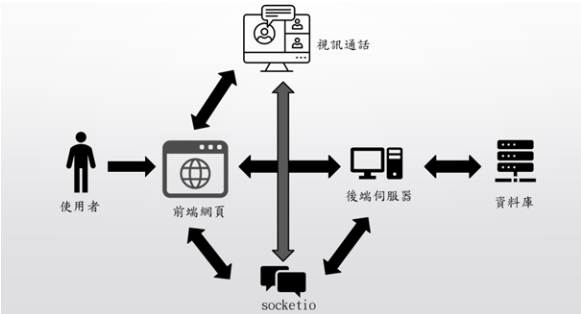
面對此全球疫情發展趨勢，在科技領域上，相關的技術應用及網路的連線需求也進而提升。身為資工系準畢業生的我們也思考著，若能熟悉此領域，或許能為防疫工作盡一份心力。

1.2 研究目的

現今市面上已有許多種可以傳遞訊息跟進行視訊通話的軟體，也各有其優缺點。但在使用上會覺得使用者介面過於繁雜。所以我們的出發點則是透過實作以研究通訊軟體背後的網路技術，並且設計出「輕鬆、方便、容易使用」的通訊網頁，讓使用者能快速且方便的與他人交流。

二、 系統架構與功能

2.1 系統架構



圖（一）：系統架構圖

使用者連線進本專題頁面後，最先接觸到的是本網頁的前端頁面，對使用者而言，所有需要的功能及操作皆只需透過前端頁面就可完整的使用本服務。

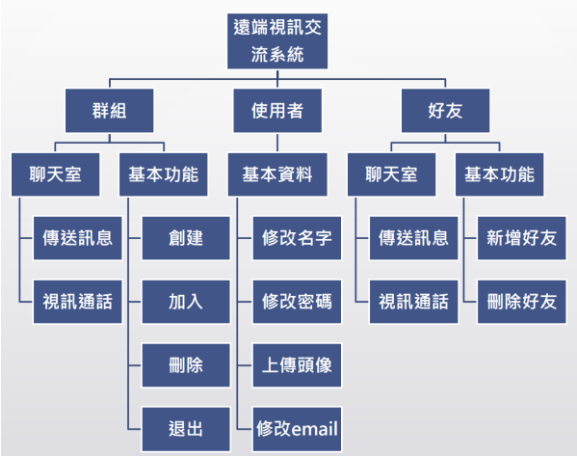
根據使用者使用的服務，前端將會向後端伺服器發送用戶的請求、再透過後端伺服器去抓取資料庫的資料（例如抓取聊天紀錄、驗證登入者身分）。

前端頁面的聊天室功能並配合名為 Socket.io 的套件來實現未讀訊息提醒、聊天室內訊息的即時更新，且即時抓取各好友及群組的相關列表。

透過點擊前端的 UI 來呼叫視訊通話服務，會將不同使用者之間的攝影機影像及麥克風聲音傳送給彼此。

使用此架構，不同服務之間的互相溝通，進而讓此網頁順利運作，讓使用者們順利運用。

2.2 系統功能



圖（二）：系統功能圖

本網頁的功能主要分成三個部分：使用者的基本資料、使用者的好友、使用者的群組。

基本資料部分，使用者的帳號將會有名字、密碼、頭像、Email 等資料，也可透過前端頁面進行更新及修改。

使用者的好友部分，有新增好友及刪除好友的基本功能，並且有著可讓使用者跟該好友進行一對一訊息傳送及視訊通話的平台（聊天室）。

使用者的群組部分，有新增群組、刪除群組、加入群組、退出群組等基本功能，並且有著可以讓使用者與群組其他所有成員進行多對多訊息傳送及視訊通話的平台（聊天室）。

2.3 開發環境

1. Vue.js 前端頁面架構

是一款開發使用者介面的 Javascript 框架，也可以建立單頁 Web 應用框架。

2. Node.js

是個利用 JavaScript 編寫的模組，它提供的 API 形式簡單，方便開發伺服器端或網路相關的應用。

3. MongoDB

是一款用 C++寫的 NoSQL 資料庫，它是用文檔的方式來儲存資料，可以儲存 T 級量的資料。

4. Express 函式庫

是個 Node.js 的 Web 應用框架，與 Node.js 原本提供的函式又更為簡潔，可以快速地架設伺服器。

5. Socket.io

是個可以開發即時雙向通信的套件，它類似於 Node.js 的處理方法，使開發者可以輕鬆處理即時事件。

6.WebRTC

是款用 C++和 JavaScript 開發的 API，

它可以支援網頁瀏覽器使用即時視訊或語音。

7.VS Code

是個可支援多種語言的程式編輯器，它的介面簡單舒適並且有內建命令列，還能安裝內建的擴充程式來拓展軟體功能。

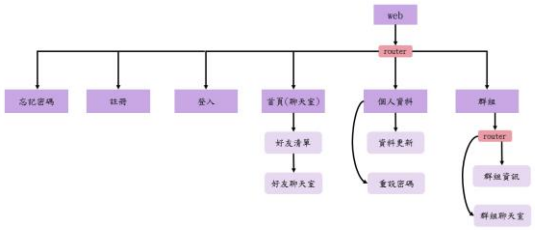
8. Postman

是一個模擬 HTTP Request 的工具，撰寫 API 時可以用此工具來測試功能。

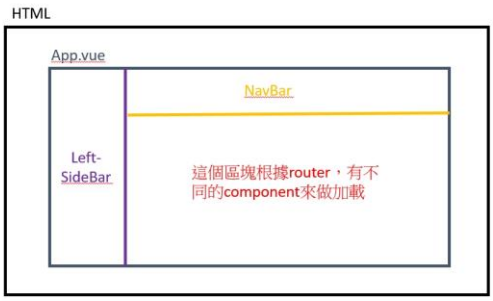
三、 實作方式

3.1 頁面功能實作

1. 前端頁面 — Vue 實作:



圖(三)：網頁前端系統架構



圖(四)：網頁框架示意圖

Vue 是一種 JavaScript 漸進式框架。透過 router(路由)在單一頁面的 html 上加載各種 component(元件)，而元件也可以重複使用，如下方圖片所示，形成完整的使用者介面。各個 component 之間傳遞資料的方式以父子組件來傳值與繼承。

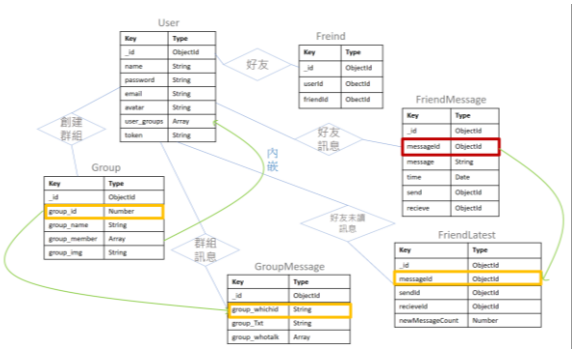
2. JSON Web Token (JWT):

過去的 session 和 cookie 驗證機制是使用者登入並驗證成功後，伺服器會建立 session 來記錄使用者驗證狀態，並將 sessionID 回傳給客戶端，讓客戶端在後續的請求中附帶有此 sessionID 的 cookie 給伺服器驗證。但此方法需要為每位使用者建立 session，若使用者增多會使資料庫的負擔增加，因此則選擇採用 jwt。

jwt 驗證機制是收到使用者登入的請求後，伺服器會驗證用戶資料，成功後會產生 Token 並回傳給客戶端，客戶端再存入 Storage，往後客戶端發送請求時需附帶此 token 給伺服器驗證，若沒有附帶 token 則會回傳錯誤。與上述 session 與 cookie 驗證機制相比，jwt 不會將用戶驗證狀態用 session 儲存起來，減輕資料庫負擔。

3.2 資料庫架構設計

資料庫使用的是 Mongo DB ，擁有延展性、彈性及高可用性的特色，特別針對需要大量資料的應用程式，在這邊分成了使用者、好友、好友訊息、好友未讀訊息、群組以及群組訊息六個資料表。



圖(五)：資料庫實體關聯圖

在使用者資料表紀錄了使用者的基本資料以及 token，再登入後會以 token 作為驗證，確認其身份。在群組的欄位中則存有陣列型態的成員，即為加入群組的使用者，在使用者的欄位中也存有群組資料。在查詢的時候省去關聯式查詢的時間，凸顯了 NoSQL 的特色，使得更方便並且有效率。

3.3 後端 API

API 運作上，提供了基本使用者註冊、登入、忘記密碼功能，這三個功能不需要 token 驗證，客戶端只要將使用者輸入的資

料打包回傳給後端對應的 API，之後再進入資料庫撈取資料比對資訊，將比對後的訊息或資料包成 JSON 回傳給客戶端頁面。

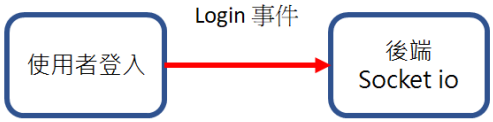
使用者登入之後 login API 會產生 token 回傳給客戶端，之後使用者使利用登入後的使用者資料、好友與群組的相關功能時，客戶端不只會回傳所需的資料，還需要將 token 放在 header 裡，才能成功的操作功能，否則會視為未登入狀態。

3.4 通知即時更新

使用 Socket.io 實現聊天室即時更新：

1. 連線到 socket.io 就會有一組屬於使用者 socket，而我們可以拿取 socket 中的 socketId 來當作使用者登入的證件。

2. 使用者登入時，就會連線到後端的 socket.io，並發送 login 事件，此時後端會收到 login 傳送過來 userId token，並將使用者的 socket、socketid、ip、userid、token 記錄在後端的線上名單中。

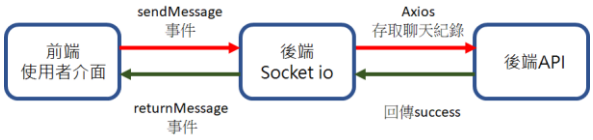


圖(六)：Socketio-Login 事件

3. 當使用者點擊好友，前端會發送 getChat 事件，將好友的 userId 傳送給後端 Socketio，後端 Socketio 再利用 axios 函式向後端 API 取得聊天紀錄資料，之後將聊天紀錄發送 returnChat 事件回傳給前端。

圖(七)：Socketio-getChat 事件

4. 當使用者發送訊息，前端會發送 sendMessage 事件，將聊天資訊 message、receive(好友 Id)、send(使用者 Id)傳送給後端 Socketio，然後將資料用 axios 給後端 API，後端 API 再回傳訊息給後端 Socketio，後端 Socketio 在廣播給使用者和好友，達到即時聊天的功能。

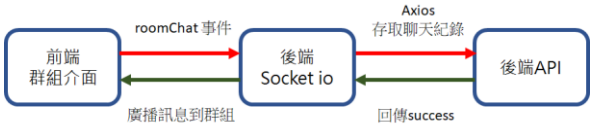


圖(八)：Socketio-sendMessage 事件

5. 當使用者登出或離開時，會將使用者從線上名單內會刪除(socket、socketid、ip、userid、token)。

圖(九)：Socketio-Disconnect 事件

6. 當使用者點擊群組，前端會先去後端 API 取得聊天紀錄。當使用者在群組裡發送訊息，發送 roomChat 事件，將聊天資訊 message、gId(群組 Id)、userId(使用者 Id)傳送給後端 API，後端 API 會回傳訊息，後端 Socketio 再將訊息廣播回群組，讓群組內的成員都看能到此條訊息，達到即時聊天的功能。



圖(十)：Socketio-roomChat 事件

以下為透過 Socketio 的前後端溝通示意圖：

前端發送事件 ：

圖(十一)：Socketio-前端發送事件

後端回傳資料 ：

圖(十二)：Socketio-後端 API 回傳資料

3.5 視訊通話

1.WebRTC(網路即時通訊技術):

一個提供 Web 應用程式及網站進行錄影或隨選播放串流音訊與影像的技術，可以直接使用瀏覽器進行資料交換而無須透過中介服務。作為一個標準規格，WebRTC 可以提供任何瀏覽器在不需要安裝外掛程式或第三方軟體下，分享應用程式的資料和進行電話會議。

四、 結論

在疫情的肆虐下，人們被迫減少彼此之間的接觸，雖然相較於其他疫情嚴峻的國家而言，國內在一開始對於防疫的警覺和積極對策避免了疫情的擴大，但繼而改以利用視訊軟體來交流仍是全世界的一大趨勢。

我們也透過此次專題的實作，對於伺服器的建置、網頁架構及溝通方式更為熟練，也理解了許多網路連線背後的技術應用。相信此份專題對於我們來說必是難能可貴的一次經驗。雖說本網頁是為一次專題的題目，也望將此份系統更完善，提升系統的完整性及加入特色功能後，推廣到供學校其他師生，甚至讓有需求的所有人使用。

五、 參考文獻

[1] JWT

https://medium.com/%E9%BA%A5%E5%85%8B%E7%9A%84%E5%8D%8A%E8%B7%AF%E5%87%BA%E5%AE%B6%E7%AD%86%E8%A8%98/%E7%AD%86%E8%A8%98-%E9%80%8F%E9%81%8E-jwt-%E5%AF%A6%E4%BD%9C%E9%A9%97%E8%AD%89%E6%A9%9F%E5%88%B6-2e64d72594f8

[2] WebRTC API

https://developer.mozilla.org/zh-TW/docs/Web/API/WebRTC_API

[3] Node.js

https://zh.wikipedia.org/wiki/Node.js

[4] MongoDB

http://mongodbcanred.blogspot.com/2015/01/mongodb.html

[5] Socket.io

https://ithelp.ithome.com.tw/articles/10102886

[6] Visual Studio Code

https://zh.wikipedia.org/wiki/Visual_Studio_Code