

電子工程總整第六組期末報告

組員：

陳韋翰 B10902212

陳毓堦 B10902222

劉品宗 B10902229

目錄

壹、 前言	2
一、 研究動機.....	2
二、 研究方法.....	2
貳、 正文	3
一、 系統架構.....	3
二、 系統功能.....	3
三、 系統流程圖.....	4
(一)、 基本功能.....	4
(二)、 權限控制.....	4
四、 具備專業能力	5
(一)、 Microsoft Azure	5
(二)、 Linux	8
(三)、 Docker	9
(四)、 HTTP、HTTPS.....	10
(五)、 WebRTC.....	11
(六)、 Socket.....	11
五、 設計理念.....	13
(一)、 互動性	13
(二)、 實用性	14
(三)、 自由度	14
六、 時程表.....	14
七、 分工.....	14
八、 遇到問題及其解決方式	15
(一)、 虛擬機	15
(二)、 Docker	15
(三)、 程式	15
九、 學到的技術.....	15
參、 結論	16
肆、 參考資料.....	16

壹、前言

一、研究動機

隨著影音娛樂、社群媒體的盛行，市面上有越來越多的影音直播平台問世。而最早的直播平台可以追溯到 2007 年的 Justin.tv，而 Justin.tv 抑是如今當紅直播平台 Twitch 的前身，且除了 Twitch 以外，也出現了更多像是 Facebook、鬥魚、熊貓等直播平台。而網路直播若以廣義的方式去定義，有更多像 Google meet、Zoom、Webex、LINE、Twitter 等亦可認為是一種直播的形式。

而除了一般的一個直播主對許多觀眾的形式，因為疫情的關係使得直播的形式有逐漸轉變的趨勢。以一般的直播會更注重於直播主所帶給觀眾的表演，而觀眾能與直播主互動的方式就僅僅是聊天室、捐獻等功能。如今，觀眾更想要和直播主有除了聊天室互動以外的功能，像是可以透過某些方式一起直播、類似加入一個房間與實況主語音交流，以往皆是透過另外的第三方平台去完成該目標，於是發想，是否能不透過他方平台就達成這個目標？

存在於市面上的直播平台可謂各有千秋，而隨著技術的成熟也許我們也可以著手建立一個簡易的直播平台，以自己使用過其他直播平台所擁有的經驗發掘其優勢與其劣勢做保留與汰除。

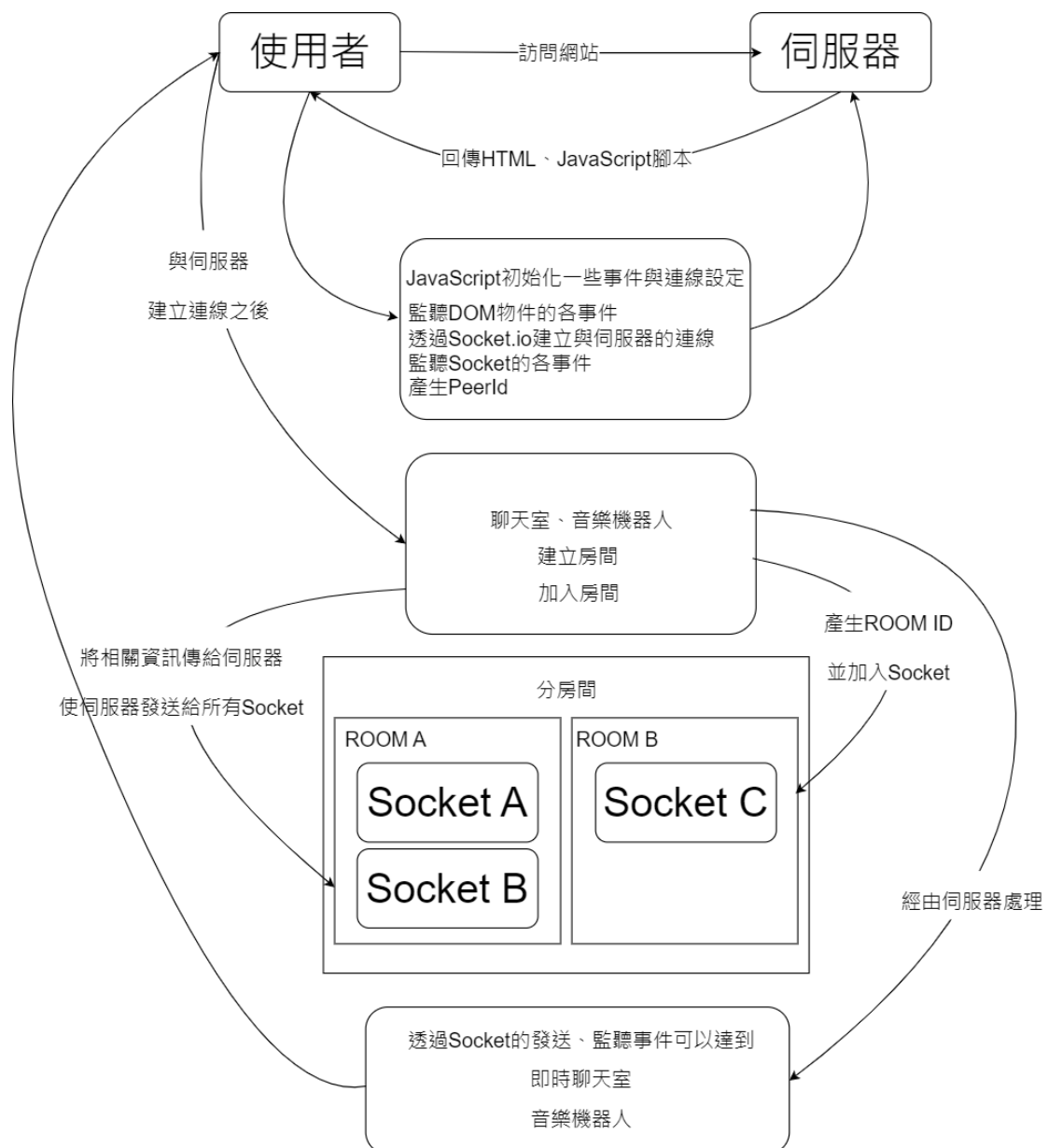
二、研究方法

透過電子工程總整的課程與網路上各式文獻，透過做中學的方式不斷嘗試錯誤並修正。且因為必須透過實務上的操作不斷地從中發現 bug 並解決，且為了不斷修正功能達到上述之保優汰劣的事項，勢必會經歷許多的錯誤，而不斷從錯誤中學習抑是最重要的研究方法。

貳、正文

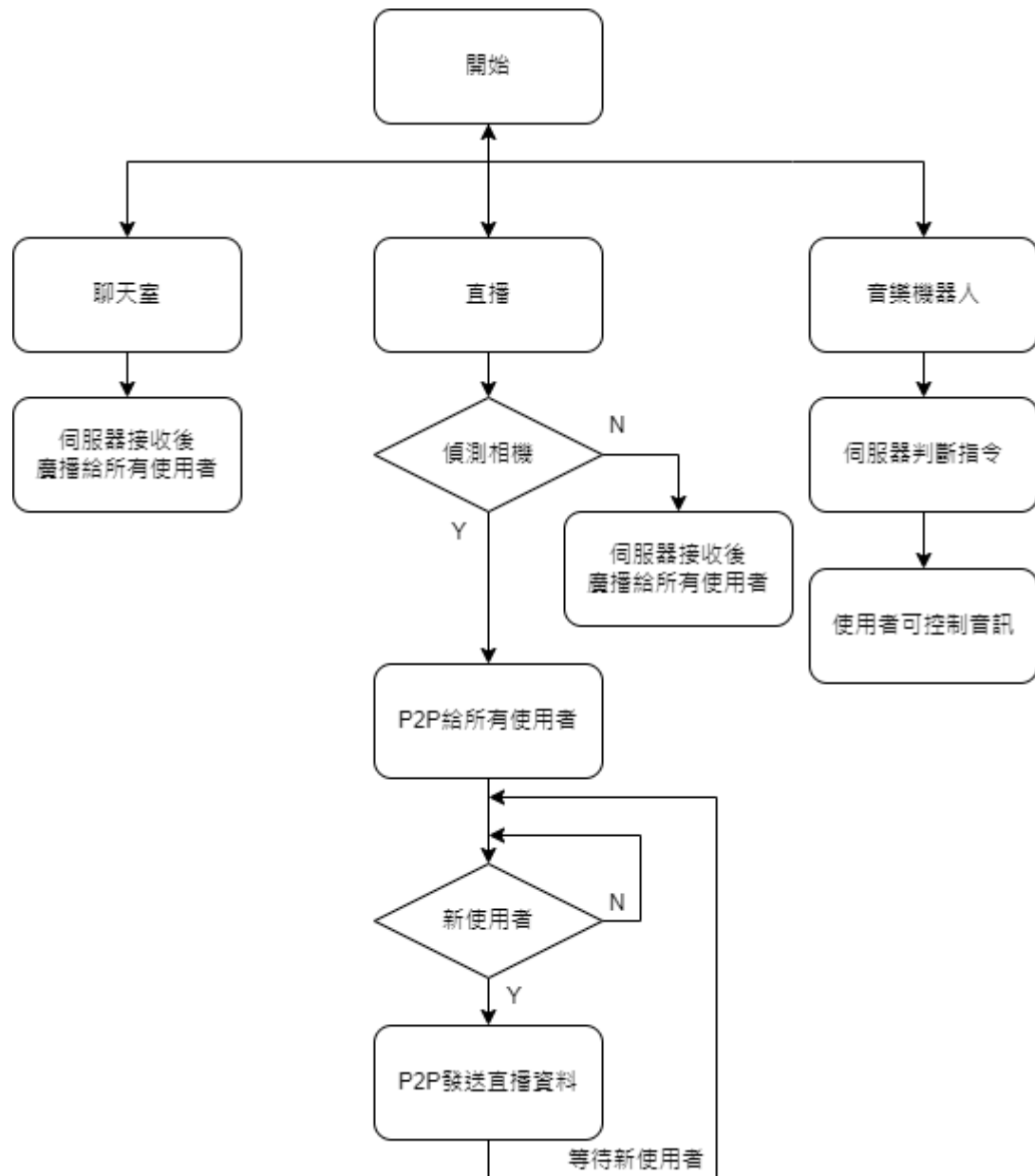
一、系統架構

二、系統功能

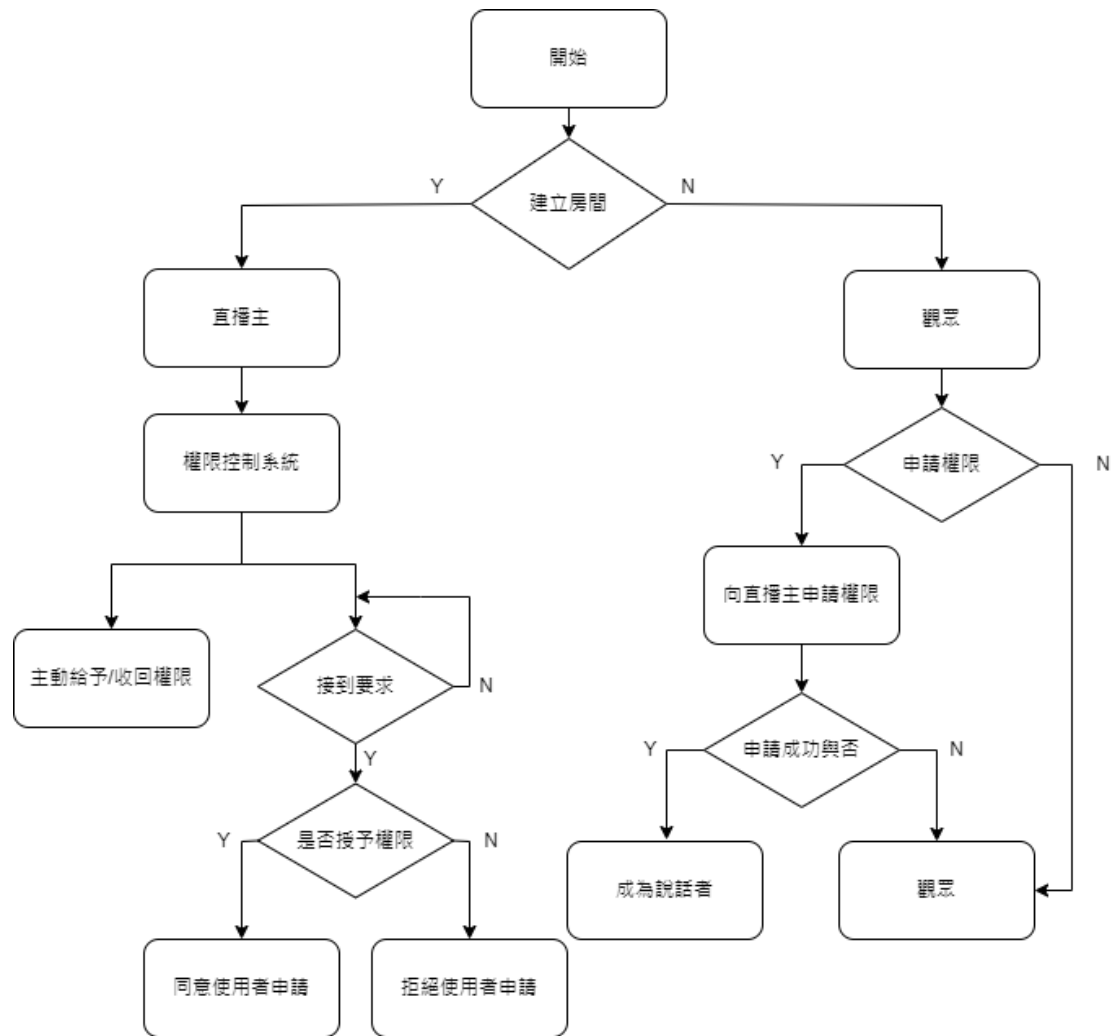


三、系統流程圖

(一)、基本功能



(二)、權限控制



四、具備專業能力

(一)、Microsoft Azure

1、簡介

Azure 是 Microsoft 在 2008 年開始發展並在 2010 年正式推出的功用雲端服務平台，而所謂公用雲端(Public cloud)是指有些組織具有雲端的設備進而以租用的方式出售給大眾，使各公司可以使用該組織所提供的雲端服務。



2、雲端運算

雲端運算是基於網際網路的運算方式，他可以共享軟硬體的資源並且依照終端需求分配所需資源。而一般雲端業務都具備有以下特徵：

- 降低使用者對 IT 專業知識之依賴
- 任意地點皆可存取
- 多人共享共同資源
- 快速部屬之靈活度
- 可被監控之服務

(1)、服務模式

可以根據服務模式的不同進而區分出 IaaS、PaaS 與 SaaS。

a、IaaS

IaaS 又稱基礎即運算，他提供了使用者處理、儲存、各種運算資源以及網路的功能。而 IaaS 又是在雲服務的最底層，需要用戶自行處理一些基礎設施的基本運作邏輯。可以控制作業系統、儲存裝置等。

b、PaaS

PaaS 又稱平台即服務，他是介於 IaaS 與 SaaS 中間的一種服務方式。PaaS 提供使用者將基礎的雲端設施部屬至使用者端，使用程式碼、程式庫等。而使用者不需要管理基礎的雲端設施。作業系統、網路等。

c、SaaS

SaaS 又稱軟體即服務，他僅需透過網路，不用任何安裝步驟即可使用。而通常用在商業應用，像是會計系統、人力管理系統等。而提供此服務的公司最大的賣點就是透過將軟硬體的維護來降低 IT 成本。

(2)、部屬模型

據部屬模型可以分大約 3 至 4 種:

a、公用雲

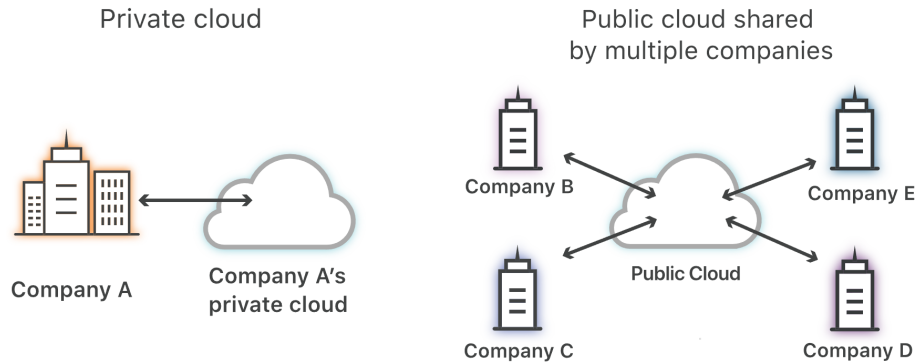
公用雲端服務指所有使用者皆在管理的共同伺服器上運作，但不同客戶不會知道彼此的資訊。且有優點像是成本較低、省下管理伺服器時間、安全性等。

b、私有雲

私有雲端服務如字面意思，為單一使用者組織所有。而相較於公用雲它具有更高的安全性以及更高的權限去控制私有雲。而私有雲除了更高的安全性以外還有以下優點，可預測的效能、成本並且能夠節省長期成本，但相對應的需要有更高的 IT 能力。

c、混合雲

混和雲結合了公私有雲的一些特性，且一般而言公有雲是提供給多個使用者共用，私有雲則是裝載於組織專屬的私人伺服器中。



3、使用的功能

在此次的課程中我們使用了虛擬機的方式去達成一個伺服器的功能，而所謂的虛擬機則是透過虛擬化的技術去分割出 Azure 伺服器之 CPU、記憶體等具有運運算能力的資源。而在 Azure 中主要分為兩種方式，分別是 IaaS 與 PaaS。

(1)、IaaS

Virtual Machine (VM):提供標準的虛擬機器映像或是使用者上船的方式客製化虛擬機器。可支援 Windows 與 Linux 等作業系統。而此次伺服器是使用此方式建立。

(2)、PaaS

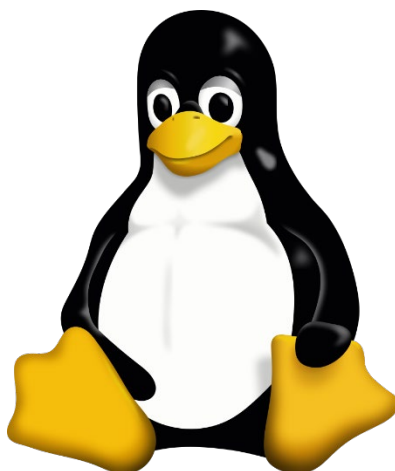
Web APPs: 提供部署網頁應用程式的 Web Apps, 以及具無伺服器 (Serverless) 應用開發能力的 Logic Apps 與 Functions 服務。

(二)、Linux

1、簡介

Linux 是一個開源的類 UNIX 作業系統的核心。而要成為一套完整的作業系統必須再加上使用者空間、應用程式等。所以有人會把它包成一個完整的作業系統，稱為發行版本，常見的有 Fedora Ubuntu、openSUSE 等。且因為本身開源的關係，只要符合

規範所有人皆可以使用。



2、常用指令

cd 目錄	可以前往該目錄
ls	列出當前目錄之檔案
rm 檔案	移除檔案
mkdir 名稱	在當前目錄建立該名稱的目錄
sudo 指令	提升權限去執行指令

(三)、Docker

Docker 是一個源自於 2013 年的開源專案，而他最主要的目的是實作輕量級的作業系統虛擬化解決方案。而 Docker 大概會涉及到三個基本的觀念，映像檔(Image)、容器(Containers)、倉庫(Repository)。而 Docker 與傳統的虛擬機所使用的虛擬化技術類似但有所不同，虛擬化的技術是虛擬化硬體，而 Docker 則是基於作業系統之上的虛擬化，因此有可攜式性、高效地利用伺服器等優勢。因此無論是環境的分離或者是跨設備的運行皆可以輕鬆完成。

1、映像檔(Image)

Docker 的映像檔像是一個唯獨的模板，而他可以像是包含一整個 ubuntu 作業系統的環境，裡面僅僅安裝了一些需要的應用程式和 Apache 等。而映像檔可以用來建立容器。使用者亦可直接使用一個已經由他人做好的映像檔。

2、容器(Container)

Docker 利用容器來執行應用。**Docker** 的每個容器都是互相隔離的，彼此的環境不會受到對方的影響，可以把他當成一個簡易版的 **Linux** 環境。

3、倉庫(Repository)

倉庫集中存放了數量極大的映像檔，常見的有 **Docker Hub** 和 **Docker Pool** 等。而使用者亦可於本地端建立自己的私人倉庫，可以用 **push** 上傳映像檔至倉庫需要使用時再 **pull** 下來即可使用。

(四)、HTTP、HTTPS

1、HTTP

HTTP 最初的目的是用於接收、發布 **HTML** 頁面的方法。而 **HTTP** 一般預設的序列埠是 **80**，且通常使用 **TCP** 協定去處理客戶端和伺服器的連接問題。且 **HTTP** 是屬於 **OSI** 架構中應用層中的協定。在常見的版本 **HTTP/1.1** 中，定義了八種方式，其中包括 **GET**、**HEAD**、**POST**、**PUT**、**DELETE**、**TRACE**、**OPTIONS**、**CONNECT** 和 **OPTIONS**。而一般 **HTTP** 伺服器至少要有 **GET** 和 **PUT**。

而 **HTTP** 因為設計較為簡便，既不會驗證通訊方的身分亦無法保證其內容在傳輸過程中未經竄改。因此衍伸出了加密版本的 **HTTP**，即 **HTTPS**。

2、HTTPS

HTTPS 預設的序列埠是 **443**，且使用了 **SSL/TLS** 的方式去對網站進行加密，而加密一類分為兩種，對稱加密與非對稱加密。對稱加密即雙方擁有同一把鑰匙，也就是無論加密與解密皆使用同一把密鑰，而如何安全傳輸密鑰是重點。而非對稱加密則是每個終端各自有自己的私鑰與公鑰，公鑰可以任意傳輸，而加密的過程中則是使用接收者的公鑰進行加密再傳輸給該使用者，並用該使用者之私鑰進行解密，因此解決對稱式加密密鑰傳輸的問題。

而 HTTPS 則使用了混和驗證的機制，其原因是非對稱式加密效率過低，因此透過證書的方式讓首次使用非對稱式加密，等接收端接受到證書後再改為效率較高的對稱式加密。

而一般憑證是由數位憑證認證機構(CA)所頒發的，而他們承擔公鑰體系中公鑰的合法性檢驗的責任。而數位憑證可以驗證身分以外也可確認數據的完整性。

(五)、WebRTC

WebRTC 是一個支援網頁瀏覽器進行即時語音與通訊的 API，而其主要由兩項核心技術所組成，分別是 Media Capture and Streams API 和 peer-to-peer connection。

1、Media Capture and Streams API

可以理解這項技術為基於 MediaStreamsObject(代表一個影像與聲音)的同步串流，而開發者亦可使用該功能取得使用者得多媒體串流，進而顯示在瀏覽器上或對他處理。

2、peer-to-peer connection

可以簡稱為 P2P，主要是使伺服器不需要處理兩個個端之間的資料傳輸，而是透過兩個使用者自行處理。而前述中有提到 HTTP 伺服器則是與此截然不同的傳輸方式。一般 HTTP 伺服器是 A 使用者上傳資料到伺服器再由 B 使用者向伺服器請求獲得 A 的資料。而 P2P 則是第一次連接時確認彼此連線後往後皆由兩使用者端自行傳輸。

(六)、Socket

Socket 是一個可以讓開發者不用管理任何讀取 IP、傳到目的地的任何細項便可以連接到外部系統進行溝通的一個工具。以電話系統為例 Socket 就像其中的電話號碼與分機號碼，只要記得該設備的 IP 位址和 PORT 號碼便可以進行傳輸，其也可以選擇用幾種不同的傳輸協定，TCP 與 UDP。

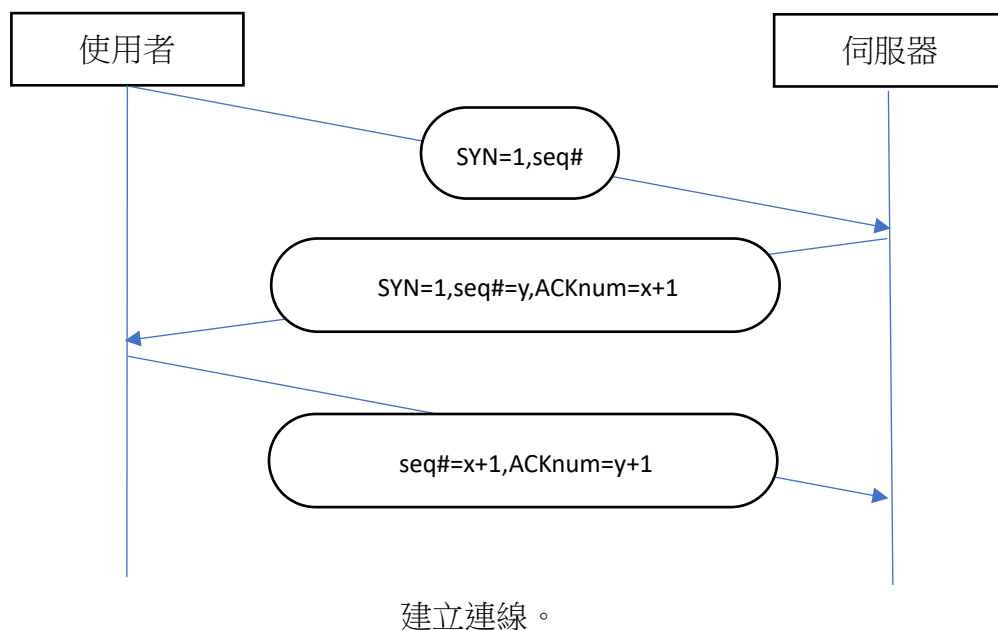
1、TCP

TCP 是一種網路傳輸的協定，其全名為 Transmission Control Protocol。這個傳輸協定具有可靠性高等優點。而其最著名的屬於他的連接方式，三次握手以及四次揮手。

(1)、三次握手

三次握手主要是建立連線的過程，其步驟如下：

- 使用者先傳送一個訊號，且會使 $SYN=1$ 與自己的 $Seq\#=x$ 給伺服器。
- 伺服器接到訊號後會回傳一個 ACK，而同時 $SYN=1$ 與自己的 $Seq\#=y$ ，同時會有一個 ACKnum 預計下一次收到 $seq\#$ 應為多少。
- 使用者接受到再回傳給伺服器一個訊號即成功

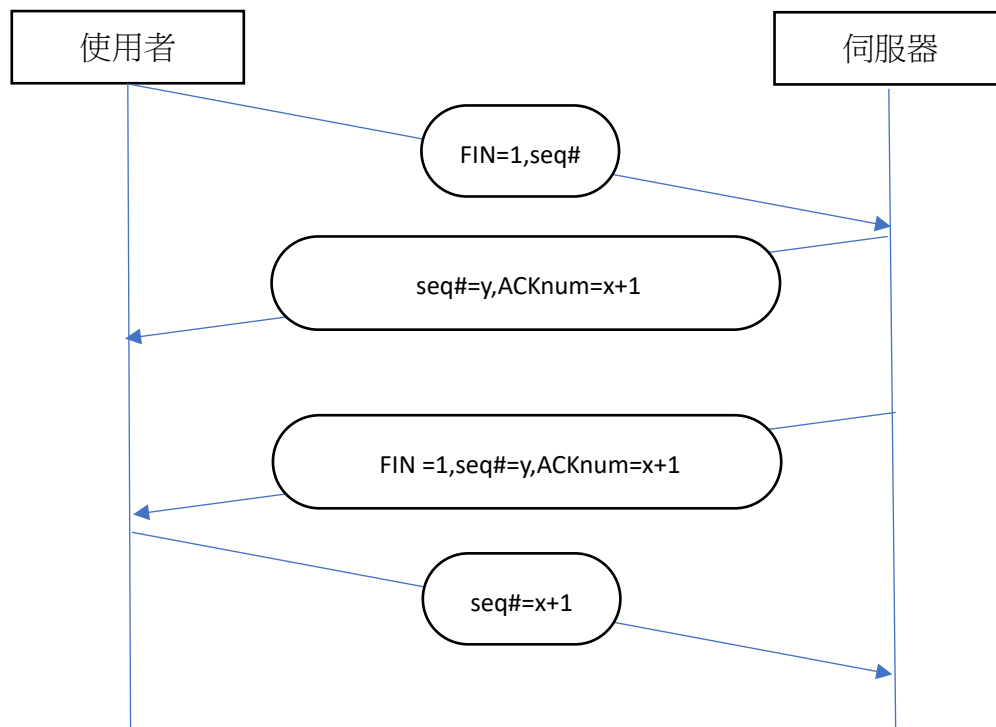


(2)、四次揮手

四次揮手主要是用於切斷連線，其步驟如下：

- 使用者先將 FIN 設為 1 並傳送給伺服器。
- 伺服器得知要關閉連線時先進行關閉，且關閉前後各送出一次 $seq\#=y$ 的回應。第一個回應會使客戶端等待伺服器關閉連線
- 第二個則是伺服器已關閉完成。

- 客戶端接到訊息後會再傳送一個訊息給伺服器，伺服器接收後正式斷開連接。



2、UDP

UDP 是一種網路傳輸的協定，其全名為 User Datagram Protocol。這個傳輸協定雖然可靠度較低，但同時他的速度較 TCP 快上許多。因為會遺漏一些封包，且沒有流量控管的手段，因此使用此協定的應用程式必須能夠接受一些封包遺漏的情況，像是直播串流、語音通訊等。

五、設計理念

根據平常使用直播平台的習慣，我們希望能夠把直播平台設計成互動性高、實用性高、自由度高的一款直播平台。

(一)、互動性

隨著社群生態的改變，實況變得不再只是單對多的一種活動，越來越多直播平台會加上更多可以使直播主與觀眾互動的功

能，而同時也可以看到實況主用各種方式增加與觀眾互動的機會，無論是透過第三方平台抑或是遊戲中，越能拉近與觀眾的距離越能夠造就好的直播環境。因此我們吸取的各種平台互動的優點，像是引進了可以申請權限加入互動、不同客群的使用者可以在不同的房間等。

(二)、實用性

實況平台光有虛華的功能並不符合長久的計畫，以外觀吸引大眾眼光是較為簡易能看到成效的方法，而若虛有其表並無法維持長期的發展，因此我們嘗試先以高實用性的方式去設計、製作需要的功能。像是 yt 機器人能夠背景播放音樂，個別麥克風音量調整、偵測等等。透過微小的細節讓使用者在使用直播平台時有更好的體驗。

(三)、自由度

自由度與管理難易程度該如何權衡仍是各大市面上直播平台面臨的嚴重問題，無論是各種仇恨言論會傷害少數族群或是太過嚴格管控導致的文字獄都令人困擾。我們根據兩害相權取其輕的情況下去判斷出想要設計出一款自由度更高的直播平台，因此使用了匿名登入的方式免除了登入所造成的麻煩，但為了方便管理也加入了一點權限的概念。

六、時程表

時間	歷程
09/05~10/17	學習相關技術
10/18~10/26	製作出簡易版本的網頁
10/27~11/02	架設伺服器、更新網頁的 CSS、處理憑證問題
11/03~11/30	修正 BUG、測試各種功能、新增 yt 機器人
12/01~	包 Docker

七、分工

八、遇到問題及其解決方式

(一)、虛擬機

1、已安裝 peerjs 卻無法使用 sudo peerjs

輸入 `sudo chmod -R 777`，777 代表了最高權限，在輸入此指令後就沒有問題了。

2、Web Apps

原本使用 **Web Apps** 去建立我們的伺服器，他可以連結 **github** 並自動部署該網頁。但我們開啟伺服器的方式與他預設不同導致只打得開用 **HTML** 設計的頁面而無法讓伺服器順利開啟。因此改用 **VM** 當做架設伺服器的設備。

3、無法使用攝影機、麥克風

因為 **HTTP** 的關係，瀏覽器會把網頁設置為不安全的網頁，因此無法讀取麥克風與攝影機，因此需要申請 **SSL** 憑證讓 **HTTP** 變為 **HTTPS** 並且改通訊埠由 80 至 443。

(二)、Docker

(三)、程式

九、學到的技術

- JavaScript
- Linux
- Node.js
- Socket.IO

- Docker
- Virtual Machine
- HTML
- CSS

參、結論

肆、參考資料

- [什麼是基礎架構即服務？ IaaS 說明 - AWS \(amazon.com\)](#) 檢索時間(20221205)
- [什麼是公用雲端？ | 公用與私人雲端 | Cloudflare](#) 檢索時間(20221205)
- [\[Cloud Computing\]Public Cloud v.s. Private Cloud](#) 檢索時間(20221205)
- [認識 Microsoft Azure](#) 檢索時間(20221205)
- [雲端運算](#) 檢索時間(20221205)
- [HTTP 為什麼不安全 | 前端隨筆](#) 檢索時間(20221206)
- [基本概念 - 《Docker —— 從入門到實踐》](#) 檢索時間(20221206)
-