



SYMSKAN

華藝文獻相似度檢測

分析報告 - 全文報告



## 華藝Symscan文獻相似度檢測分析報告-全文報告

用戶名稱：s10830618@gm.cyut.edu.tw  
文檔名稱：購物直播\_技術報告(最終完成)  
上傳日期：2020/12/25

3%  
總相似度

相似文獻共1篇，1段落 —

1.

---

### 看門狗門禁系統

謝隆斌;薛義郎;陳韋均;陳嘉祥  
TANET2019 臺灣網際網路研討會2019年，2019  
DOI：10.6924/TANET.201909.0073  
來源：  
相似段落共1段

3%  
文章相似度

## 閱讀方式

灰底文字：系統檢測出的相似段落文字。範例：相似段落文字。

實線編號框：與您的文檔相似的文章編號。範例：**1**

虛線編號框：您已手動排除的相似文章編號。範例：**3**

範例：**1****3**相似段落文字。表示您的相似段落文字同時與1號及3號文章相似，但您已手動排除3號文章。

## 相似段落文字 —

資訊與通訊系 2020 年實務專題技術報告購物直播指導教授:李建緯組員:廖俊原、朱家毅、吳柏承、吳松勸、楊証傑朝陽科技大學資訊與通訊系聯絡方式: 413 台中市霧峰區吉峰東路 168 號

Tel:(04) 23323000 ext.

7243Fax:(04)23305539E-mail: lijw@cyut.edu.tw摘要當我們在觀看購物直播時，會發現要購買直播商品時，需要由人工來完成下訂單和建立訂單。

因此如果有一個功能可以讓買家在該網站快速下訂單，而賣家能有完整的訂單資訊就能解決這樣的狀況，本專題透過 MongoDB 來存取及抓取資料，圖一、系統架構圖可讓賣家事先在直播前將要賣的商品先上架，而 2.2 系統說明買家可透過賣家直播時所販賣的商品去下訂單。

關鍵詞: Node.js、MongoDB、Websocket 本專題的系統軟體有 JavaScript、Node.js[1]，套件軟體有 MongoDB[3]、Nginx、WebSocket[9]，Abstracthe 使用者可使用 OBS[5]來擷取畫面或拍攝實況畫面來串流至指定的伺服器網域後，將其畫面推播至When we are watching the shopping live 網頁平台，並將所要販售的商品資訊傳遞到資料broadcast, we will find that when we want to buy live 庫。

買家再透過平台取得已存在資料庫的商品資products, we need to manually complete the order 訊去進行購買，在將雙方的資訊存至資料庫。

placement and order creation.

Therefore, if there is a 2.3 軟體 feature that allows buyers to place orders quickly on the website, and sellers can have complete order information to solve this situation, this topic usesMongoDB to access and capture data, allowing sellers 2.3.1 MongoDB[3] to pre-live broadcast the products to be sold are put onthe shelves first, and buyers can place orders through MongoDB 是一款為 Web 應用程式及網際網路the products sold by the seller during the live 設計的資料庫管理系統，也是目前最多人使用的broadcast.

NoSQL 資料庫的第一選擇。

MongoDB 是文檔和集Keywords: Node.js、MongoDB、Websocket 合的概念，使用了 BSON(Binary JSON)的格式作為儲存數據結構，MongoDB 適合用在網站登入登出、事件紀錄、快取及需要擴充的地方，在我們這次1.前言的專題應用中較多的為網站登入登出及事件紀錄，非常符合我們的需求，故選用 MongoDB。

本平台是使用會員制，當使用者在登入後可以選擇買賣家，直播主選擇賣家後可以上架商品及創建房間。

觀眾則是選擇買家後可以選擇想要觀看的實況主。

本專題透過 WebSocket 建立了聊天室，讓買賣家可以在聊天室裡互動及殺價。

也透過 OBS 去攝取視訊鏡頭的畫面，而後推播到指定的房間內讓買去觀看。

最後利用 MongoDB 讓賣家事先上船的商品存到資料庫裡，買家則可以觀看或下架已上傳的商品，而買賣家雙方就都能得到完整的訂單資訊。

## 2.系統架構

### 2.1 系統架構圖二、MongoDB資訊與通訊系 2020 年實務專題技術報告2.3.2 Visual Studio Code[4]Visual Studio

Code（簡稱 VS Code）是一個由 2.4.1 RTMP[6]Microsoft 開發的，支援跨平台，VS Code 能提供夠好的效能及穩定，它強大的原因來自於它的外即時訊息協定(Real-Time Messaging Protocol，掛，多虧了許多的 open source，讓 VS Code 幾乎縮寫為：RTMP)，為 adobe 所開發的，主要的功可對所有程式語言來開發，此軟體也自帶許多功能就是讓 Server 與 Flash 之間能夠傳輸串流媒體能，例如可以偵錯，色碼判斷及整合了 Git，讓開音訊及影片資料等等。

RTMP 準備傳輸時會先將資發時方便許多。

料封裝成一小塊，等要實際傳輸時會在分裝更小塊，這樣能夠助於調解傳輸的效率，適合長時間播放，但 RTMP 服務器會將丟失的畫面檔案給緩存起來，以提供較穩定的畫面，減少畫面斷斷續續的發生。

2.4.2 TCP/IP[7]傳輸控制協定(Transmission Control Protocol，縮寫為：TCP)，也是屬於端點對端點(End to End)的傳輸協定，和 IP 兩者幾乎是連結再一起的同一名稱(TCP/IP)，在 TCP 的兩端主機，可透過彼此的溝通，確保資料在傳輸中的正確性及傳輸速率的控制，無須經過任何節點。

### 圖三、Visual Studio 編譯環境 2.4.3 HTTP[8]2.3.3 Open Broadcaster Software[5]超文本傳輸協定(Hyper Text

Transfer Protocol，縮寫：HTTP)是一種用戶端瀏覽器和伺Open Broadcaster Software 是由 OBS Project 開服端伺服器之間溝通的標準協定，由用戶端透過發的自由開源跨平台串流媒體和錄影程式，是一網址超連結向伺服器下達 HTTP 請求，建立一個到個可以用來進行網路串流直播和錄製影片等等的伺服器的指定埠 TCP 連線，當伺服器收到請求，開源套件，操作簡單容易上手，作為直播使用非就會向用戶端傳回一個狀態，使用 MIME 格式回常方便。資料傳輸主要則是通過 RTMP 來完成的，應回用戶端。

可以傳送到支援 RTMP 的軟體，非常符合我們的需求，故使用此軟體。

2.4.4 WebSocket[9]WebSocket 是雙向通訊模式，與 HTTP 不同的地方是建立連線之後，不論是用戶端或伺服器端都可以隨時將資料傳送給對方；而 HTTP 始終都採用請求-回應的方式來通訊，相較之下 HTTP 的效率沒有 WebSocket 高，本專題有使用到網站上的即時通訊，使用 WebSocket 就無需採用輪詢、Comet技術了，避免佔用很多寬頻。

## 2.5 系統功能和實作

### 2.5.1 使用者介面(user interface)圖四、Open Broadcaster Software 環境2.4 通訊設備/協定：RTMP、TCP/IP、HTTP、WebSocket資訊與通訊系 2020 年實務專題技術報告圖八、觀看者進房介面

圖五、平台首頁示意圖圖九、聊天室介面

圖六、直播主、觀看者選擇介面圖十、購物車介面2.5.2 功能說明首先開啟平台，當你進入介面(如圖五)註冊一個帳號，然後登入剛註冊的帳號便可以選擇買賣家介面(如圖六)。

賣家:當選擇賣家進入創建房間介面(如圖七)此時可在這邊上架要販售的商品，而後買家如果有下訂單便會收到買家所下商品的完整資訊。

買家:當選擇買家就會進入選房介面(如圖八)，選擇想觀看的直播主就會進入聊天室介面(如圖九)，圖七、直播主創房介面如果觀看到喜愛的商品便可以直接下訂單，而後就會跑出完整的商品資訊(如圖十)。

## 3.參考文獻

[1] Node.js –維基百科，自由的百科全書–Wikipedia 檢自:<https://zh.wikipedia.org/wiki/Node.js>[2] Linux –維基百科，自由的百科全書–Wikipedia 檢自:<https://zh.wikipedia.org/wiki/Linux>[3] MongoDB –維基百科，自由的百科全書–資訊與通訊系 2020 年實務專題技術報告Wikipedia 檢自:<https://zh.wikipedia.org/wiki/MongoDB>[4] Visual Studio Code –維基百科，自由

的百科全書-

Wikipedia 檢自:[https://zh.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://zh.wikipedia.org/wiki/Visual_Studio_Code)[5] Open Broadcaster Software -維基百科，自由的百科全書-

Wikipedia 檢自:[https://zh.wikipedia.org/wiki/Open\\_Broadcaster\\_Software](https://zh.wikipedia.org/wiki/Open_Broadcaster_Software)[6]即時訊息協定-維基百科，自由的百科全書-

Wikipedia 檢自:[https://en.wikipedia.org/wiki/Real-Time\\_Messaging\\_Protocol](https://en.wikipedia.org/wiki/Real-Time_Messaging_Protocol)[7] TCP/IP 協定棧組成-維基百科，自由的百科全書- Wikipedia 檢自:[https://en.wikipedia.org/wiki/Internet\\_protocol\\_suite](https://en.wikipedia.org/wiki/Internet_protocol_suite)[8] HTTP | MDN – Mozilla

<https://developer.mozilla.org/zh-TW/docs/Web/>

HTTP

**1** [9] WebSocket -維基百科，自由的百科全書-Wikipedia 檢自:<https://zh.wikipedia.org/wiki/WebSocket>4.致謝

首先，誠摯的感謝專題指導老師李建緯教授和指導的學長，我們平時在教授跟學長細心的指導和適時的提供建議下，讓我們可以順利的完成本次專題。

在製作專題的期間，讓我們學習到許多的知識，像是思考方向的能力及遇到問題的態度，也會告訴我們專題可能會面臨到的困難還有其中的應對方式，都讓我們收穫良多。

另外要感謝的是本組參與的組員，不斷地進行溝通和內部討論，致使本專題可以順利的完成，回首專題的研究過程，從初期時我們的都不太會，只會一直在網路上查詢相關資料，到專題的完成並且順利參展，不僅提高小組成員的專業能力，更重要的是在於每位組員都能互相的幫忙，不怕辛苦去完成每個功能。

最後我們要再次的感謝李建緯老師的用心指導，也讓我們知道專題是大家一起努力的，謝謝大家！

---

## 引文疏漏檢測 —

缺少文內註		
作者	篇名	年代
未檢測出缺少文內註		

缺少參考書目	
作者	年代
未檢測出缺少參考書目	

# 購物直播

指導教授: 李建緯 組員: 廖俊原、朱家毅、吳柏承、吳松勸、楊証傑

朝陽科技大學資訊與通訊系

聯絡方式: 413 台中市霧峰區吉峰東路 168 號

Tel: (04) 23323000 ext. 7243

Fax: (04)23305539

E-mail: lijw@cyut.edu.tw

## 摘要

當我們在觀看購物直播時，會發現要購買直播商品時，需要由人工來完成下訂單和建立訂單。因此如果有一個功能可以讓買家在該網站快速下訂單，而賣家能有完整的訂單資訊就能解決這樣的狀況，本專題透過 MongoDB 來存取及抓取資料，可讓賣家事先在直播前將要賣的商品先上架，而買家可透過賣家直播時所販賣的商品去下訂單。

**關鍵詞:** Node.js、MongoDB、Websocket

## Abstract

When we are watching the shopping live broadcast, we will find that when we want to buy live products, we need to manually complete the order placement and order creation. Therefore, if there is a feature that allows buyers to place orders quickly on the website, and sellers can have complete order information to solve this situation, this topic uses MongoDB to access and capture data, allowing sellers to pre-live broadcast the products to be sold are put on the shelves first, and buyers can place orders through the products sold by the seller during the live broadcast.

**Keywords:** Node.js、MongoDB、Websocket

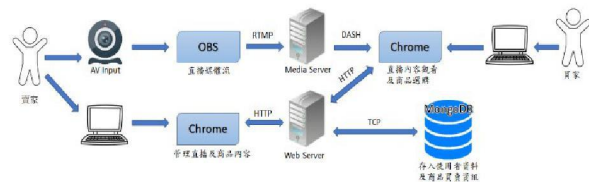
## 1. 前言

本平台是使用會員制，當使用者在登入後可以選擇買賣家，直播主選擇賣家後可以上架商品及創建房間。觀眾則是選擇買家後可以選擇想要觀看的實況主。本專題透過 WebSocket 建立了聊天室，讓買賣家可以在聊天室裡互動及殺價。也透過 OBS 去攝取視訊鏡頭的畫面，而後推播到指定的房間內讓買家去觀看。

最後利用 MongoDB 讓賣家事先上船的商品存到資料庫裡，買家則可以觀看或下架已上傳的商品，而買賣家雙方就都能得到完整的訂單資訊。

## 2. 系統架構

### 2.1 系統架構



圖一、系統架構圖

### 2.2 系統說明

本專題的系統軟體有 JavaScript、Node.js[1]，套件軟體有 MongoDB[3]、Nginx、WebSocket[9]，使用者可使用 OBS[5]來擷取畫面或拍攝實況畫面來串流至指定的伺服器網域後，將其畫面推播至網頁平台，並將所要販售的商品資訊傳遞到資料庫。買家再透過平台取得已存在資料庫的商品資訊去進行購買，在將雙方的資訊存至資料庫。

### 2.3 軟體

#### 2.3.1 MongoDB[3]

MongoDB 是一款為 Web 應用程式及網際網路設計的資料庫管理系統，也是目前最多人使用的 NoSQL 資料庫的第一選擇。MongoDB 是文檔和集合的概念，使用了 BSON(Binary JSON)的格式作為儲存數據結構，MongoDB 適合用在網站登入登出、事件紀錄、快取及需要擴充的地方，在我們這次的專題應用中較多的為網站登入登出及事件紀錄，非常符合我們的需求，故選用 MongoDB。

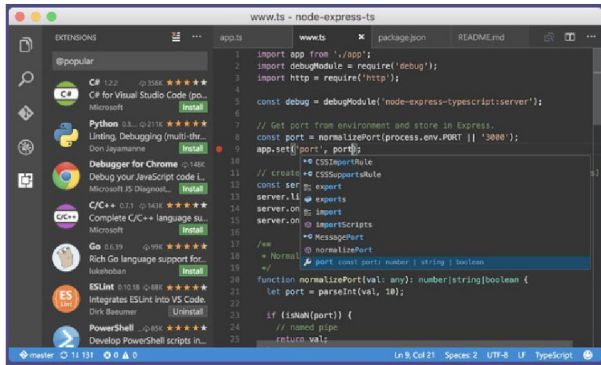
Database Name	Storage Size	Collections	Indexes
admin	52.0KB	3	0
config	26.0KB	3	0
local	44.0KB	1	1
loginlog	72.0KB	2	0
normal	62.0KB	1	1
test001	0.0KB	4	0
test002	0.0KB	0	0

圖二、MongoDB



### 2.3.2 Visual Studio Code[4]

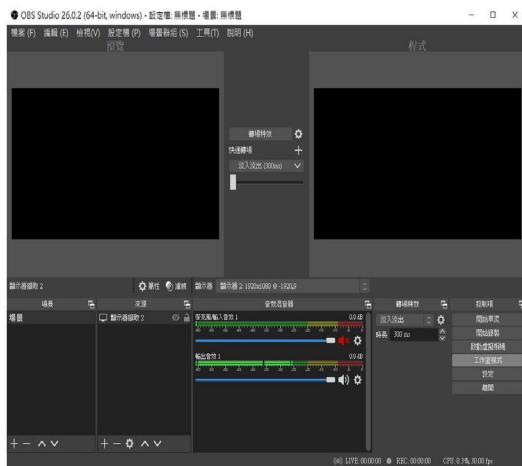
Visual Studio Code (簡稱 VS Code) 是一個由 Microsoft 開發的，支援跨平台，VS Code 能夠好的效能及穩定，它強大的原因來自於它的外掛，多虧了許多的 open source，讓 VS Code 幾乎可對所有程式語言來開發，此軟體也自帶許多功能，例如可以偵錯，色碼判斷及整合了 Git，讓開發時方便許多。



圖三、Visual Studio 編譯環境

### 2.3.3 Open Broadcaster Software[5]

Open Broadcaster Software 是由 OBS Project 開發的自由開源跨平台串流媒體和錄影程式，是一個可以用來進行網路串流直播和錄製影片等等的開源套件，操作簡單容易上手，作為直播使用非常方便。資料傳輸主要則是通過 RTMP 來完成的，可以傳送到支援 RTMP 的軟體，非常符合我們的需求，故使用此軟體。



圖四、Open Broadcaster Software 環境

## 2.4 通訊設備/協定：RTMP、TCP/IP、HTTP、WebSocket

### 2.4.1 RTMP[6]

即時訊息協定(Real-Time Messaging Protocol，縮寫為：RTMP)，為 adobe 所開發的，主要的功能就是讓 Server 與 Flash 之間能夠傳輸串流媒體音訊及影片資料等等。RTMP 準備傳輸時會先將資料封裝成一小塊，等要實際傳輸時會在分裝更小塊，這樣能夠助於調解傳輸的效率，適合長時間播放，但 RTMP 服務器會將丟失的畫面檔案給緩存起來，以提供較穩定的畫面，減少畫面斷斷續續的發生。

### 2.4.2 TCP/IP[7]

傳輸控制協定(Transmission Control Protocol，縮寫為：TCP)，也是屬於端點對端點(End to End)的傳輸協定，和 IP 兩者幾乎是連結再一起的同一名稱(TCP/IP)，在 TCP 的兩端主機，可透過彼此的溝通，確保資料在傳輸中的正確性及傳輸速率的控制，無須經過任何節點。

### 2.4.3 HTTP[8]

超文本傳輸協定(Hyper Text Transfer Protocol，縮寫：HTTP)是一種用戶端瀏覽器和伺服器端伺服器之間溝通的標準協定，由用戶端透過網址超連結向伺服器下達 HTTP 請求，建立一個到伺服器的指定埠 TCP 連線，當伺服器收到請求，就會向用戶端傳回一個狀態，使用 MIME 格式回應用戶端。

### 2.4.4 WebSocket[9]

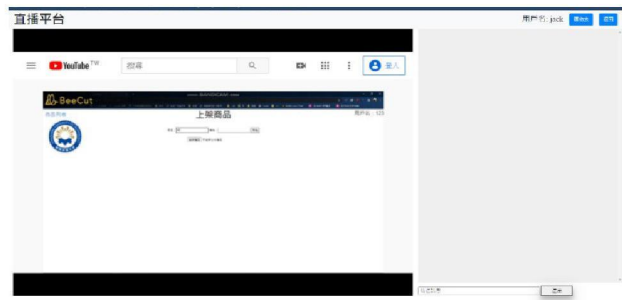
WebSocket 是雙向通訊模式，與 HTTP 不同的地方是建立連線之後，不論是用戶端或伺服器端都可以隨時將資料傳送給對方；而 HTTP 始終都採用請求-回應的方式來通訊，相較之下 HTTP 的效率沒有 WebSocket 高，本專題有使用到網站上的即時通訊，使用 WebSocket 就無需採用輪詢、Comet 技術了，避免佔用很多寬頻。

## 2.5 系統功能和實作

### 2.5.1 使用者介面 (user interface)



圖五、平台首頁示意圖



圖九、聊天室介面



圖六、直播主、觀看者選擇介面



圖十、購物車介面



圖七、直播主創房介面



## 2.5.2 功能說明

首先開啟平台，當你進入介面(如圖五)註冊一個帳號，然後登入剛註冊的帳號便可以選擇買賣家介面(如圖六)。

賣家: 當選擇賣家進入創建房間介面(如圖七)此時可在這邊上架要販售的商品，而後買家如果有下訂單便會收到買家所下商品的完整資訊。

買家: 當選擇買家就會進入選房介面(如圖八)，選擇想觀看的直播主就會進入聊天室介面(如圖九)，如果觀看到喜愛的商品便可以直接下訂單，而後就會跑出完整的商品資訊(如圖十)。

## 3. 參考文獻

- [1] Node.js – 維基百科，自由的百科全書 – Wikipedia 檢自：  
<https://zh.wikipedia.org/wiki/Node.js>
- [2] Linux – 維基百科，自由的百科全書 – Wikipedia 檢自：  
<https://zh.wikipedia.org/wiki/Linux>
- [3] MongoDB – 維基百科，自由的百科全書 –



- Wikipedia 檢自：  
<https://zh.wikipedia.org/wiki/MongoDB>
- [4] Visual Studio Code – 維基百科，自由的百科全書 –  
 Wikipedia 檢自：  
[https://zh.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://zh.wikipedia.org/wiki/Visual_Studio_Code)
- [5] Open Broadcaster Software - 維基百科，自由的百科全書 –  
 Wikipedia 檢自：  
[https://zh.wikipedia.org/wiki/Open\\_Broadcaster\\_Software](https://zh.wikipedia.org/wiki/Open_Broadcaster_Software)
- [6] 即時訊息協定 – 維基百科，自由的百科全書 –  
 Wikipedia 檢自：  
[https://en.wikipedia.org/wiki/Real-Time\\_Messaging\\_Protocol](https://en.wikipedia.org/wiki/Real-Time_Messaging_Protocol)
- [7] TCP/IP 協定棧組成 – 維基百科，自由的百科全書 –  
 Wikipedia 檢自：  
[https://en.wikipedia.org/wiki/Internet\\_protocol\\_suite](https://en.wikipedia.org/wiki/Internet_protocol_suite)
- [8] HTTP | MDN – Mozilla  
<https://developer.mozilla.org/zh-TW/docs/Web/HTTP>
- [9] WebSocket – 維基百科，自由的百科全書 –  
 Wikipedia 檢自：  
<https://zh.wikipedia.org/wiki/WebSocket>

#### 4. 致謝

首先，誠摯的感謝專題指導老師 李建緯 教授和指導的學長，我們平時在教授跟學長細心的指導和適時的提供建議下，讓我們可以順利的完成本次專題。在製作專題的期間，讓我們學習到許多的知識，像是思考方向的能力及遇到問題的態度，也會告訴我們專題可能會面臨到的困難還有其中的應對方式，都讓我們收穫良多。另外要感謝的是本組參與的組員，不斷地進行溝通和內部討論，致使本專題可以順利的完成，回首專題的研究過程，從初期時我們的都不太會，只會一直在網路上查詢相關資料，到專題的完成並且順利參展，不僅提高小組成員的專業能力，更重要的是在於每位組員都能互相的幫忙，不怕辛苦去完成每個功能。最後我們要再次的感謝李建緯老師的用心指導，也讓我們知道專題是大家一起努力的，謝謝大家！