資料庫系統概論

Introduction to Database Systems

期末專題-Codzy 線上自主學習

資工四甲 謝清福 406261690

資工三乙 林鈺恩 407262354

資工三乙 賴婷妤 407262419

資工三乙 呂明洋 407170434

資工三乙 楊 晴 407262249

目錄

第一章 緒論

一、發展背景與動機

今年因為疫情肆虐全球，導致全球各大專院、中小學等實施遠距較學，部分教授會以直接丟現成的開放式影片當作現成的教材，又或是遠距教學因為網速、攝影機品質等造成學生無法受到良好的上課品質；若想要自行尋找開放式課程的影片，以台灣而言，目前全國的開放式課程只有台清交僅提供較完整的開放式課程平台且，其他學校並沒有提供平台及課程，或是都是老師自行上傳且以目前都沒有一個平台是有完整的統合所有學校的資源。因此我們希望可以做出一套**資源完整**且**專屬資工**的線上自主學習平台。

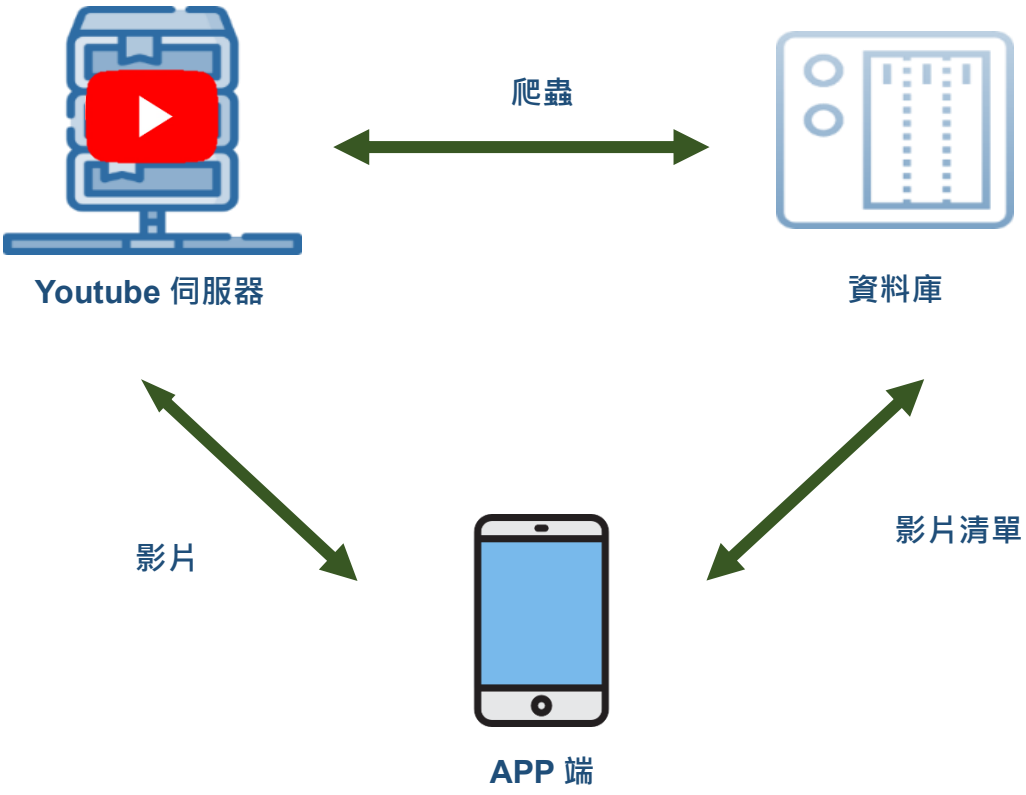
二、系統發展目的

我們希望可以做出一套能以資工系為主，保有 Youtube 豐富性及多元性的影片清單系統，並透過下方的留言結合情感分析去算出該影片的推薦指數，同時也有熱搜榜，讓使用者能清楚的知道最近哪些影片正被大量的用戶觀看，且包含著所有台灣頂尖大學關於資工系課程的的開放式影片，好讓使用者可以有系統性的學習，打造出一套專屬資工系的線上自主學習平台，且能幫助資工系學生們找出最適合自己的自主學習課程，並能用最精簡、最一目了然的方式呈現。

第二章 系統說明

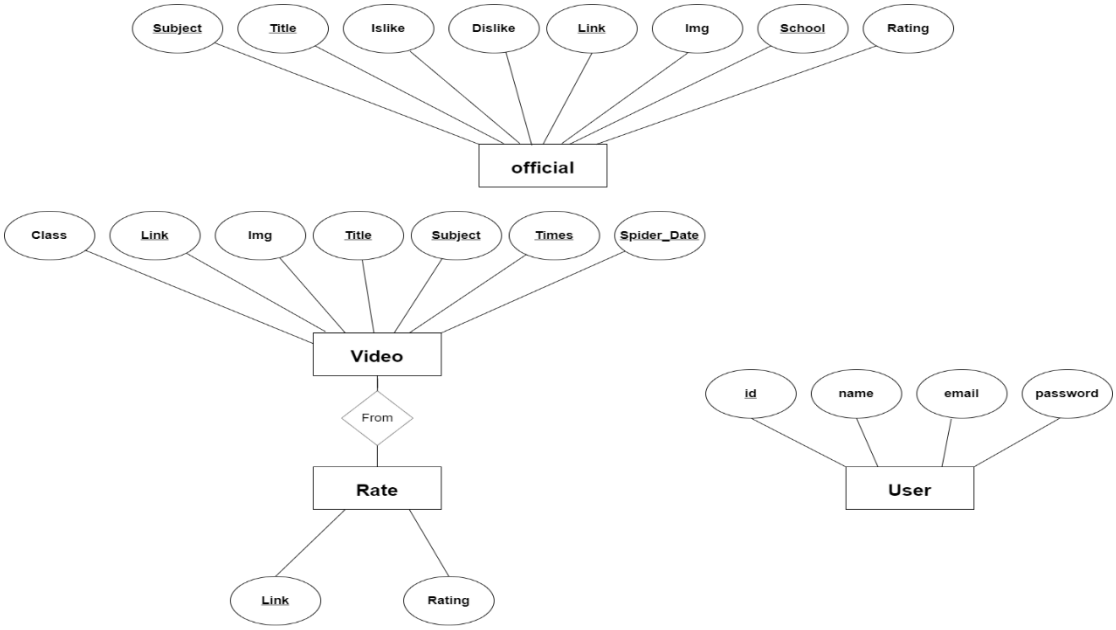
一、系統架構

如下圖所示，當 APP 送出 request 時，APP 會透過 internet 去資料庫尋找使用者所想查詢的影片類別並且回傳給 APP 端對應到的清單及資料，若使用者想看該部影片，點選影片後就可以透過 internet 連到 Youtube 的伺服器觀看想看的影片。而資料庫則會每日的去獲取 Youtube 裡各個類別的影片和相關資訊，包含留言，喜歡/不喜歡數，觀看次數等。

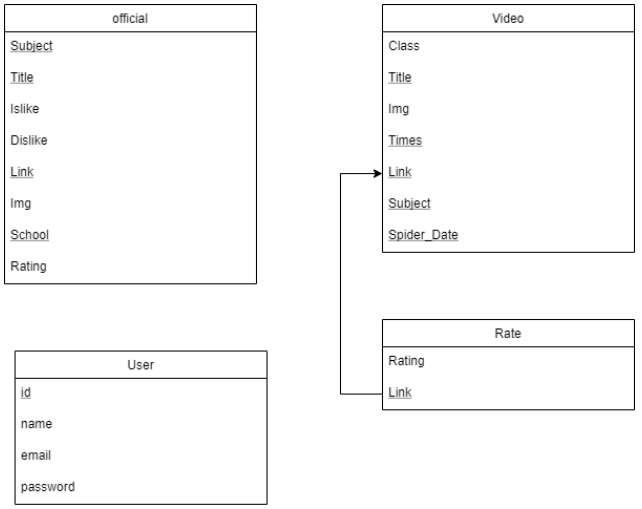


二、資料存取與儲存方式

以下依序為我們的 ER-diagram、資料量、Schema。我們將 Youtube 上的標題、連結、縮圖、點閱率經過爬蟲後會進入到我們的資料庫，並用 User 這張表紀錄所有已註冊的使用者，而 official 則是存著所有由學校拍攝的 Youtube 影片及相關資訊



資料表	動作	資料列數	類型	編碼與排序	大小
official	瀏覽 結構 搜尋 新增 清空 刪除	1,341	InnoDB	utf8mb4_general_ci	1.5 MB
rate	瀏覽 結構 搜尋 新增 清空 刪除	11,095	InnoDB	utf8mb4_general_ci	1.5 MB
users	瀏覽 結構 搜尋 新增 清空 刪除	8	InnoDB	utf8mb4_general_ci	16.0 KB
video	瀏覽 結構 搜尋 新增 清空 刪除	~184,104	InnoDB	utf8_general_ci	40.8 MB
4 張資料表 總計		~196,548	InnoDB	utf8mb4_general_ci	43.9 MB



三、程式開發與使用工具

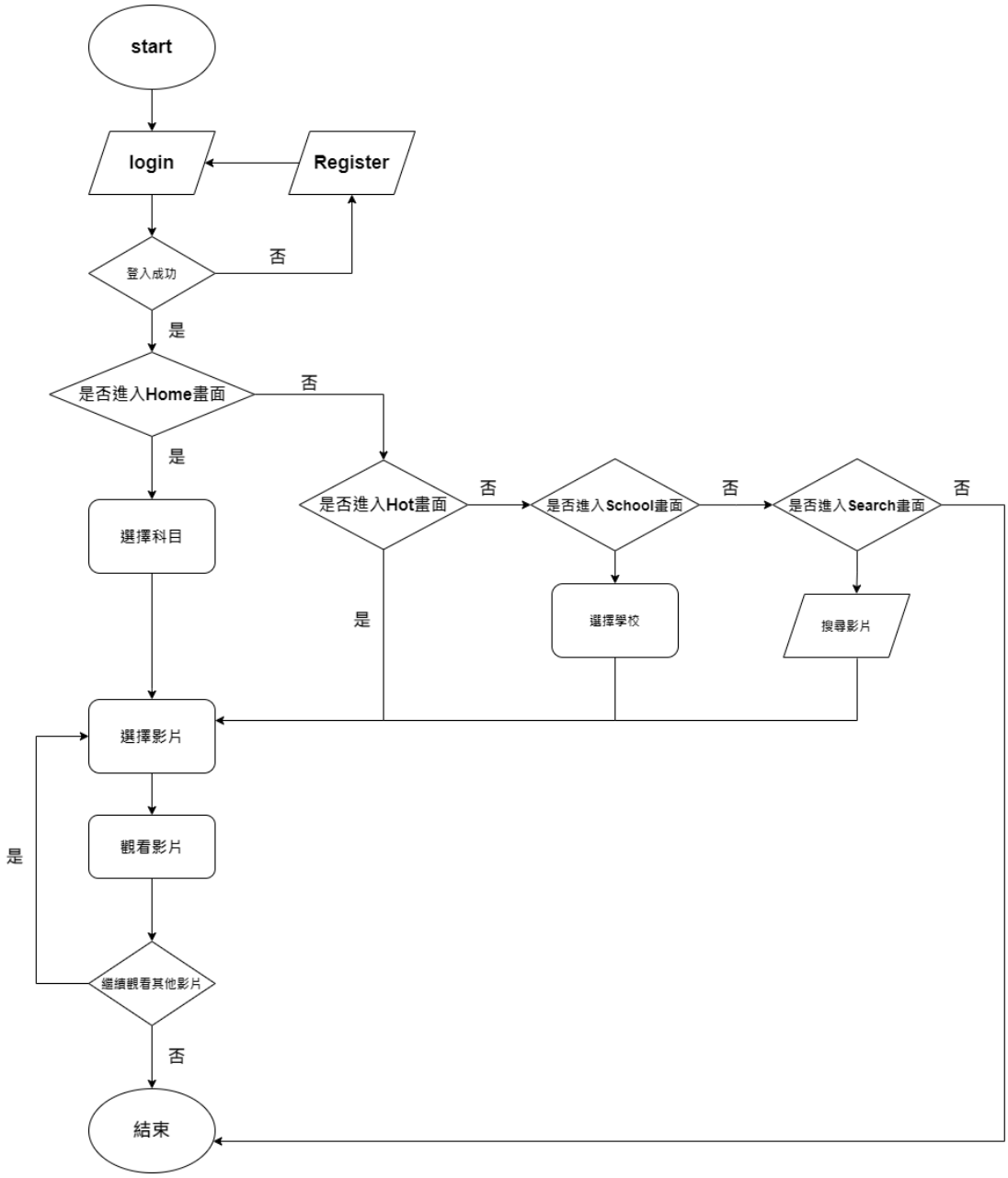
本專題為了達成線上自主學習平台。，使用多套系統開發工具，以下圖所列，

以下根據開發內容詳述製作所使用之工具

- APP 端:使用 Android Studio 搭配 Java 開發 Android 的 APP
- 爬蟲:使用 Python 搭配 Selenium 獲取 Youtube 各影片的資訊
- 情感分析:使用 Python 搭配 Textblob 和 SnowNLP 作為判斷留言是否對影片是正/負面之評論
- 資料庫:使用 XAMPP 架設 MySQL 和 phpmyadmin, MySQL 紀錄使用者資訊及影片資訊，且透過 phpmyadmin 提供的 GUI 介面對 MySQL 做管理，並可以透過 PHP 讓 APP 從資料庫中抓取資料

第三章 系統實作與資料分析

一、系統流程



二、系統介面

- 類別:同時也是我們的主畫面，使用者可以在這邊點選自己想看的類別、科目，就會有關於這個科目的所有影片，並且這些影片都會包含我們自製的評分系統。因為開發對象是針對輔大資工系的學生，因此我們的類別是以輔大資工系的必修及選修課程為主。我們針對輔大資工系的課表分成六個類別

應用:人工智慧、網路概論、資料庫系統、雲端計算、物聯網、網頁、視窗設計

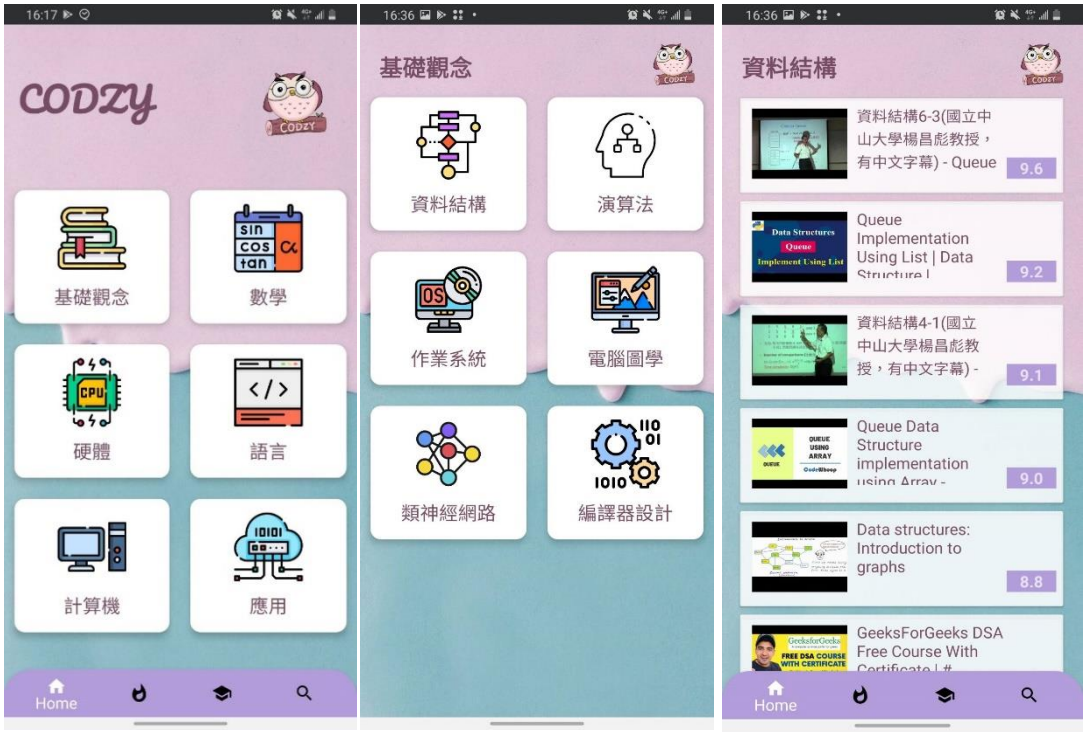
數學:微積分、機率與統計、離散數學、線性代數、工程數學、數值方法

硬體類:微處理機系統、數位邏輯設計、數位電子學

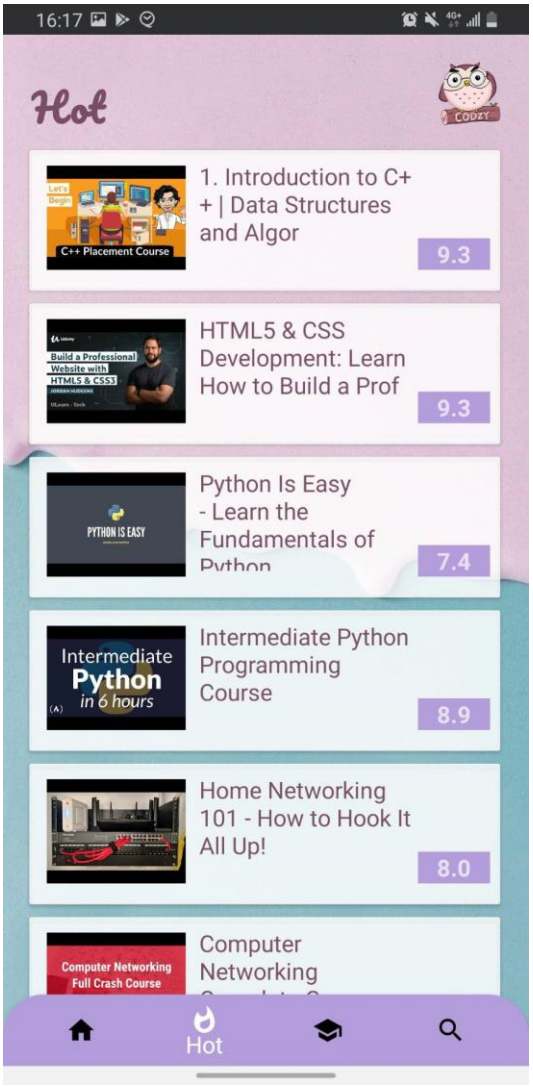
語言類:C#、C、C++、JAVA、PYTHON、R 語言、MATLAB、ASSEMBLY

基礎觀念:、資料結構、演算法、作業系統、電腦圖學、類神經網路、編譯器設計

計算機類:計算機概論、計算機組織、計算機結構

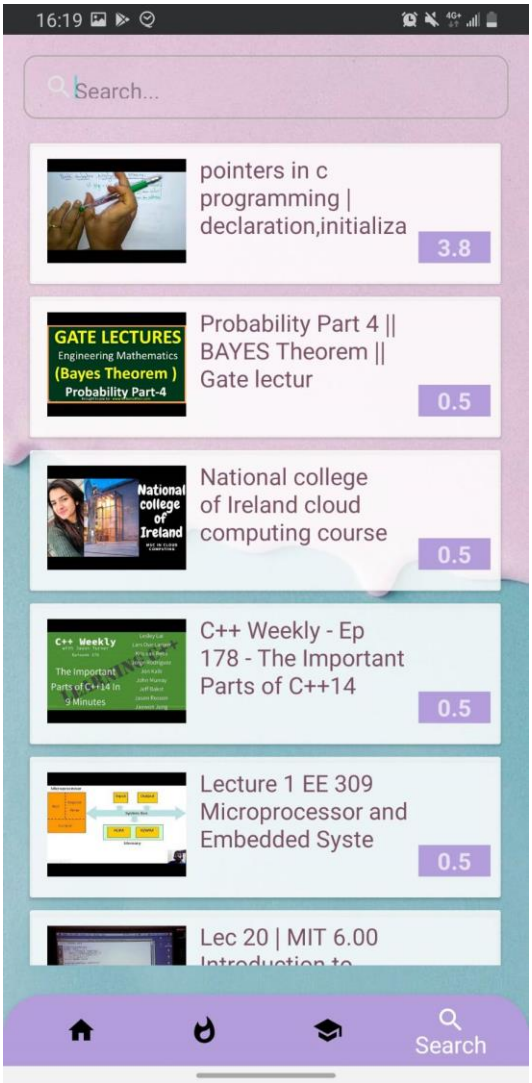


- 熱門: 會這此類別顯示近一個月的最熱門影片十大影片，為了更精確地找出熱門影片，採用的計算方式是當月最大的點閱率減到最小的點閱率以便能更明顯看出這部影片在點閱率上的成長，並設計，除了要點閱率高之外，在影片的自製評分系統下還要大於 7 以上，才有資格成為熱門影片



- 學校:目前已將全台有具有系統系的開放式課程列在 APP 上，並且目前也只有台大、清大、清大具有較完整的開放式課程資源，與 TOCEC 不同的是，我們已經各科目的各部影片使用人力+電腦爬蟲將其資料收集下來，以便使用者可以直接透過 APP 去觀看該校的開放式課程，而不需要像 TOCEC 需要在連回原校網站後才能進行觀看

- 搜尋:透過此頁面可以搜尋列在本 APP 上的所有影片，使用者可以透過影片上的標題做搜尋。由於使用各大學的開放式課程系統發現在搜尋上十分不方便，或是常常無法找到使用者希望找到的，因此在編寫此功能時，特別將搜尋功能加以優化，除了增加準確率，還改善語法，讓搜尋速度能夠到達無延遲。



三、情感分析原理及實際留言分析

在本次過程中，我們使用了 SnowNLP 的套件去對我們每一部影片的留言做情感分析，SnowNLP 是一套基於 TextBlob 所延伸的情感分析模組。與 TextBlob 的差異是，TextBlob 通常處理英文，而 SnowNLP 在處理中文方面比 TextBlob 的準確率高很多。而情感分析是五大部分組成的，分別是 entity-實體,aspect-屬性,opinion-觀點,holder-觀點持有者,time-時間。假設有一個留言是”我覺得近年來蘋果手機續行力非常好”那在這個留言中，蘋果手機代表實體，續航力是屬性，觀點是非常好，觀點持有者是我，而時間則是近來

情感分析其實是藉由實際的留言的正向情感以及負面情感文本數據，重新訓練出的此次分析所使用的原始模型，訓練中會逐行讀取正面以及負面樣本，進行分詞、去除停用詞並加上正負樣本的標記，最後以 Naoive Bayes 模型統計各詞的頻率進而達成分析模型。

模型訓練完成後，逐一讀入所收集到的各則評論，使用我們自行訓練的情感分析模型加上 Naoive Bayes 模型預測出數值，其數值介於 0-10 之間的浮點數，越靠近 10 表示越積極(正面)，若數值愈接近 0 代表情感越消極(負面)。舉例來說，假設一則留言是“怎麼影片錄製時背景音樂卻開很大聲 真吵 教學成效 直接扣分。得出原始成績為 0.3，因為有”吵”、”扣分”因此判定為較負面的留言；假設另一則留言是“教得很細 很受用 感恩”，得出原始成績為 9.8，因為有”感恩”、”受用”，因此判斷為較正面的留言

第四章 結論

一、開發時遇到的問題:

在開發此 APP 時，我們遇到了許多難題並設法改善，以下做說明

- 官方影片未公開:在收集部分學校的開放式課程影片時，我們發現校方可能會將影片的權限設為未公開以防止不相干的人士觀看，但這會導致我們直接用關鍵字去做尋找，我們的解決方法是，將影片的網址一一記錄在一個 txt 檔裡，並讓程式去讀這個檔裡每部影片的相關資訊，以收集校方的開放式課程影片
- 爬蟲過於耗時:因為本次我們所需要爬得類別數量不少，並且每次爬蟲因為我們所需要的影片資訊過多，無法單獨直接在 Youtube 搜尋頁面上做讀取，也就是說我們每部影片都必須點進去影片才可以做爬蟲，後來經過多方嘗試，發現將爬蟲時的等待時間降低，並且挑在網路品質較穩定的場所爬蟲會有利於爬蟲速度的提升。
- 不相關的影片太多:在爬各類別影片時，我們發現很多科目的詞彙常常出現在各種影片的關鍵字中，像人工智慧常常出現在各大電視節目以及報導、機率統計常常出現在賭博、遊戲類的影片標題中、作業系統常常獲取到關於各家作業系統的說明、安裝等等。因此讓我們在下關鍵字時更為小心謹慎，同時要增加條件，例如不要有新聞之類的關鍵字等

二、未來展望:

這次的專題因為時間上較為不夠，留下了很多遺憾，以下做說明

- 做出網頁版 方便桌上型電腦使用:希望有機會可以開發出網頁版本，好讓桌上型電腦可以一同使用，讓學生們不管是用在家還是在外都可以學習，達到更加的效率
- 提高英文語意辨識的準確度:因為目前所使用的 SnowNLP 套件是較偏向分析中文語言的部分，但畢竟學術無國界，許多良好的影片都是以英語為主，因此希望未來有機會透過更多的演算法，去判斷中文留言或是英文留言，以及針對不同的語系可以使用不同的情感分析套件做使用
- 拓大使用範圍:因為目前是針對輔大資工系的學生做設計，希望未來可以針對輔大理工學院，或是全球的資工系學生，甚至是全球的大學生，讓台灣更多優秀、內容精闢的影片可以被世界各地的學生看到。而因為理工科系在許多必修科目上重複性蠻高，因此希望可以讓不只是資工的學生，全理工的學生都能一同使用此 APP。
- 推薦影片:希望在註冊時可以設置年級，這樣登入後系統便可以透過使用者的年級，去推斷說該名學生目前正在上哪些課程，會需要哪些課程的影片及相關背景知識，並推薦使用者說哪些影片是目前使用者會需要、正在進行的，讓自主學習不再是自己一個人，而是手機會主動的提醒你

三、Q&A:

Q:十萬筆資料哪來的?

A:我們每天會去抓去各個類別的影片及相關的影片資訊，目的是要找出最熱門的影片並呈現給使用者看。而系統獲得影片連結後，也會抓取相關的留言並去判斷其分數，但留言因美觀及流程設計方面，所以不呈現給使用者看，也是因為這個原因所以我們沒有將留言放在資料庫裡。但光每日爬各類別影片的點閱率，其實就已經超過 10 萬筆了。

Q:為什麼 APP 檢視資料如此少?

A:考慮到不管呈現的資料多與少，使用者都必須點進去觀看影片，因此為了美化整體的設計，我們就沒有將所有我們爬下來的資料做呈現。

Q:為什麼沒有做到期初說的收藏功能?

A:考量到以實際層面來說，通常都是點進去影片，有觀看才會知道這部影片是否適合自己、是否值得推薦，因此我們就將收藏的功能取消，改成顯示各大學得開放式課程。並且若真的想要將影片收藏，其實點進去 Youtube 做收藏會更為方便及直觀

Q:萬一影片沒有留言怎麼辦?

A:我們有將影片裡喜歡數跟不喜歡數的資訊抓取下來，若發現影片本身是沒有留言的，我們變會用喜歡及不喜歡數做評分的計算

四、心得

楊晴:

這次專題遇到很多障礙，第一個困難就是從來沒有寫過 Android Studio，所以一開始完全不知道怎麼下手，只能跟著別人的實作，邊做邊熟悉，所以可能一個簡單的 BUG 就要好幾天才找出答案，第二個困難是，開發 APP 的部分有三個人一起，各自做出一些功能後，發現要合在一起才是最困難的，因為有些環境、變數會不一樣，在一個我們都不熟悉的開發環境下，要找出問題真的很難，這也讓我第一次體會到 Error 訊息的重要，因為沒有 Error 的 Bug 才是最可怕的，因為好不容易可以執行程式後，出現閃退或一片空白的情況，真的不知所措，後來才發現要連後端的資料，必須要讓 PHP 抓的東西跟 JAVA 要的東西一樣才行，而 layout 顯示的東西也都必須在 JAVA 宣告才不會閃退，這些問題都是試了好幾次才發現，經過這些測試，感覺未來在 debug 時會比較有方向，我覺得這次專題是一個很好的學習經驗，學習到要如何與組員溝通協調以及合作，組內分工很明確，大家也都盡全力完成自己的工作，最後才能有一個成果出來。

賴婷妤:

上資料庫這堂課前還不知道資料庫是什麼，上了幾堂課後學會 SQL 的語法，慢慢了解資料庫在開發應用程式是很重要的，每個頁面的資料都是要去資料庫抓的。起初，我們組討論了很多題目，最後由於今年疫情的關係，想到線上自主平台 APP，想要讓不只是我們資工系，其他系對資工有興趣的人能更有制度的學習，也利用語言分析幫他們篩選較好的課程。

在完成專題的過程中，我們從無到有，中間也遇到許多困難，像是 Android Studio 頁面我們碰壁了很多次，無論是頁面互相連接，功能無法呈現，Android Studio 閃退，不斷的嘗試多次慢慢一個一格成功，APP 介面方面顏色配置、icon 製作組員之間也討論很久，在資料庫連接 Android Studio 方面，一開始，僅有幾千筆測試資料時可以顯示很快，但當十萬筆資料放進資料庫 SQL 就會跑很久，好幾次因為寫 SQL 虛擬機當掉，因此我們不斷的試怎麼寫出更好的 SQL 與調整資料庫，最後呈現資料速度有進步了。

很感謝我們組討論時都很認真討論，我們幾乎每周周末都約出來討論，大家約出來沒有人反抗也沒有人缺席，在平時上課放學後也會約一起做，一步一步的完成，APP 慢慢成形成我們想像中的樣子，雖然中間遇到困難時，大家都很迷茫，很慶幸大家都能堅持下去，完成最後的成品。

呂明洋:

此次期末專題是以 codzy 線上自主學習，我主要處理的部分是資料收集的爬蟲部分以及各影片評論的情感分析，使用的語言皆為 PYTHON，使用的相關套件有 selenium 的 webdriver 網頁自動化抓取原始碼處理影片資訊與留言的收集 ，以及 SnowNLP 實現情感分析語意分析。

起初，資料收集遇到障礙，一開始連一點需要的原始碼都抓不好，很多爬蟲相關套件都不支援 Youtube 爬取，過時而無法使用， 所以後來回去一步步看原始碼抓取的方法並找到適合處理本次目標的方式，xpath 擷取到我需要的部分。最初不考慮使用 youtube api 抓取是因為會限制每日存取資訊的數量，我想要達到資料量怕會受限於此，所以選擇以網頁自動化來收集所需的數據，包含影片的標題、點閱率到喜歡和不喜歡等數據，並且達到點閱成長率的部分，由於要同影片需要多次不同時間的數據，所以在學期初就設定好必須提前在二個半月前開始測試連續爬取的穩定性以及是否有例外需要處理的情況，例如如同第二組報告所提到的，有些影片是停用留言或是並無喜歡及不喜歡的數據，這些也都是在測試中有所發現的，要一一檢視是否能夠將每個 youtube 網頁會有的情形處理好，免於錯誤無法繼續爬取，並且提早開始將 py 架設到自己另外準備的設備，這邊又面臨一個難題是由於租屋處的網路不太穩定，沒有另外獨立的線路，所以抽空回到家中架了一台 24 小時不間斷收集的設備，並分時分檔的保留原始數據以便之後提供給主要負責 php 管理的組員以及 app

端想要呈現的資料內容。

至於情感分析的部分是期中進度報告以及期初提案教授有所提到的部分，最初是使用 Textblob 去處理文檔的部分，但有礙於不提供中文留言的分析，因此改由選擇使用 SnowNLP 來處理留言，並自己重新提供正面以及反面的文本，重新訓練出的此次分析所使用的原始模型，他的雛型是從 Naïve Bayes 的一套算法來的，經過多種嘗試最後還是以提供更充足並更加貼切的正面及反面 Data，來做為主要訓練模型進而提升語意情感分析準確度的方法。

此次資料庫專題提升很多爬蟲實際案例操作的經驗，以及初次接觸情感分析的部分，先提爬蟲的部分，以往做爬蟲實作時，爬取的內容相對更簡單，可能是單一的表格或是簡單的標題等，這次更深入活用抓取我要的資料該如何撰寫，由於是自動化網頁，所以也花了很多時間在測試爬取的穩定，起初常常爬到三更半夜還在爬或還在等，因為能提早越多完成不間斷的爬取就能收集到更多的成長數舉，後續也花蠻多時間在將資料處理成其他組員所需要的的部分，但單論爬取的資料種類很充足，任何希望爬取的影片內容，只要下其關鍵字就可以做資料抓取，像是今天聆聽報告時，第二組做廣告推薦 app，有蠻多內容是我已經達到的部分，但我相對劣勢的部分是我沒有做圖形視覺化的資料處理，並且沒有發現 youtube 有關鍵字的功能，如果之後寒假有時間可以將此部分的爬取加到自己的爬蟲的 code 內，至於語意分析的部分，這邊算是第一次接觸，以往頂多處理文本是統計關鍵字詞相關的自然語言撰寫，第一次學習要

分析語意，遇到比較多的困難點是一開始訓練的模型實在是不太準確，花了不少時間處理到符合預期得模型，進而還發現有許多影片留言者會留有一些奇奇怪怪無法分析的留言。例如很不通順的中文語義及各式的表情符號,又或是留他自己撰寫的 code 放在在留言板與影片作者討論，導致準確度一直無法達到預期目標，花了很多時間校正。

林鈺恩:

在剛開始學習 android studio 時，遇到蠻多瓶頸。因為要去學習一套新的軟件加上有很多功能都不熟悉，所以剛開始都是看教學影片一步一步跟著做的，但是常常不知道其製作過程的原理，因此事後也花了不少時間去學習。中間過程中遇到了許多 fragment 和其他 activity 連接上的困難，於是請教了同學，發現很多同學都有相關的問題， 所以後來我們都使用 activity 做分頁。整體而言，我覺得我學習到了很多東西，不僅是 android studio 上的技術，我還學習到了其他的學習方式，因為以往的我可能會想從基礎到進階慢慢一步一步去學習如何操作，但是在時間有限的情況下是不可能這樣學習的，也因為這樣我也開始慢慢學會從實做的過程中學習，而我也發現這樣的學習方式更有效率，且學習的成果也不會比較差。另外，在 app 的 ui 介面設計上，有出現組員之間有不同意見的情況，所以我也從中學習到了如何在遇到意見分歧時和組員溝通。

謝清福:

我覺得這次的專題，讓我發現我同時是個 RD，也是個 PM。

在上資料庫系統概論這門課之前，因為畢專有使用到資料庫紀錄病患資料的關係，因此對資料庫算是有初步的認知及了解。但因為資料庫專題要做一個資料量高達 10 萬筆以上的 APP，因此我將重心都放在開發 APP 上。其實早在暑假時，我就已經開始請我們組員們開始思考專題主題，也有請我們組員們先利用暑假將各自會用到的能力先有一定的基礎。也因為有著暑假的討論，所以我們期初主題很快就確定下來。

我們原始的分工是我負責資料處理及功能規劃與介面設計、每次的報告及書面資料，明洋及鈺恩負責開發 APP 前端，楊晴和婷妤負責後端。但後來發現我們的主題在後端不太需要太多人力，因此就將開發 APP 給鈺恩、楊晴和婷妤。一開始的架站、前後端連線是我一手處理的。其實我們期中時就已經將資料庫成功連接到 APP 端，但因我們那時候在討論該不該用新的元件-RecycleView，以達到更美觀、更直覺的 UI/UX 體驗，並且因為是使用一個 2014 年才推出的元件，論資源豐富量及對我們對元件的熟悉度來說，都沒有 ListView 來為適合。但 RecycleView 帶來的動畫及畫面配置是 ListView 無法達到的，因此我們最後使用 RecycleView。而準備要使用 RecycleView 時已經是期中報告前一周了，因此我們只好先確認 APP 端能確實抓到資料、並且 RecycleView 能確實帶給我們想要達到的效果(但尚未連資料庫)，而期中報告

後我因為要趕畢業專題的關係，因此將開發 APP 的部分暫時交給我的組員，若到時候進度不會，我會再回來繼續開發。

雖然將開發 APP 的重責交給我的其他組員且要忙著準備畢專，但我不曾不知道他們無時無刻的開發進度，平日及假日就算自己要趕畢專，還是會與組員們一同約在咖啡廳，他們可以將遇到的問題與我討論，而我也可以確定的掌握他們的進度，若遇到 BUG 我們就會一起討論、替她們想解決方法以及可行性，例如一開始的無法點擊無法很精準，我們就在每一列上加上隱形的 button，方便使用者做點選，我們的畫面架構也是我一點一點指導他們的。

而爬蟲部分，我們早在 9 月就開始爬蟲，抓取 Youtube 相關的資料，但由於明洋之前有爬蟲的相關經驗，因此就將爬蟲的部分交給他，而爬蟲初期也遇到很多問題，例如 Youtube 改版，使的我們無法用之前的語法去抓取我們需要的資料、留言區停用、影片不公開、等等，這些都是我們一起解決的。也因為我們的資料量很大，因此體驗到正規化的重要，減少重複欄位以及減少兩張 table 相乘的機會真的大大降低的 query 的時間，從原本的 30sec 變成 3sec，

情感分析的部分，這因為當初在期中報告時教授就有叮嚀說我們說，想要透過留言去算分數，但我們對於這個概念一直很模糊，直到我去研究所找指導教授時，我的指導教授與我提起情感分析的概論，我才意識到我們可以使用這個方法分析我們的留言，並與明洋著手討論該怎麼進行，包含留言該怎麼被記錄下來，怎麼樣可以花最少的時間，爬最多的留言，並算出最精確的分數。

資料處理的部分，可能一般人會覺得這邊沒什麼，但若真正從事管理 DB 資料時，就會了解到其重要性。一開始我們並沒有做任何整理，而是直接將爬下來的資料做成 CSV 檔後匯出到資料庫，而產生的問題是資料庫效能很差，常常發現錯誤或是發現找不到相關的資料。後來發現原來是爬蟲時常常在同一時間爬到同一筆資料，而這些重複的資料並沒有任何 key 去做管理，因此我就將所有的資料刪除，並用 excel，每一個類別都檢查是否有無資料重複、錯誤，或是跑版(爬蟲時抓錯欄位)，並將重新整後的檔案再次匯入到資料庫。也因為如此，讓後來在 query 和匯入資料時效能大幅提升。

在這次的專題中，我不僅學到了如何用 Android Studio 開發 APP、SQL 語法、情感分析，更學到了如何團隊分工、合作、溝通，版本控制，我想這些都會有助於我們往後能更快的融入職場。