

資料庫專題心得

這次的專題我們這組做的是音樂 APP，而我主要是負責所有的前端工作，包含使用者介面的設計和功能的程式碼撰寫，另外還有一部分歌詞和排行榜的爬蟲及海報的設計。前端的介面和功能實作起來沒有太大的問題，最有問題的是排行榜的曲線圖生成和呈現的方式，因為我所使用的開發 IDE 並沒有原生的圖表、canvas 元件，因此當時在做的時候花了好多時間才找到了較適合我們使用 API - iOS charts，而排行榜的呈現方式組內也討論的好幾次，原先呈現出來的會是當前小時第一、二、三名歌曲在 24 小時內的曲線變化，但最後發現實作出來的效果不佳，我自己在使用者的角度來看的話，會不太直觀，因此最後才定案決定呈現出前三名的歌曲變化，有哪些歌曲在 24 小時內都爬升至前三名的資訊都顯示出來。

而我遇到較棘手的問題有兩個，其中一個是透過後端的 API 拿取資料庫的資料時，當資料量過多的時候會因為 JSON response 回來的速度慢，而前端頁面 Table 顯示的時間較早而導致錯誤，最後我是用了原生的 activity indicator 在使用者點開頁面的時候先有個等待 Loading 好的轉圈圈 view，接著在背後與後端 API 做 JSON 資料的傳遞，等到資料都準備好才顯示 Table，雖然原本的錯誤能夠解決，但就會變成使用端等待的時間變長了。另外一個問題為，在實作正在收聽頁面時，原本應該在每個歌單中都能下載歌曲和播放，然後現在收聽的頁面會接到使用者點擊了哪一首歌曲的資訊後更新頁面裡的歌曲資訊(專輯封面、歌名、歌手等)，但由於在一開始建立的時候沒有說定好和本身對 iOS App 開發的背景知識不足，導致在跳離開頁面的時候原本載入過的此頁面就會被釋放掉，進而使得在子頁面傳遞資料的時候會發生錯誤，但找了很多資料也嘗試過許多方法仍然沒有在學期結束前把問題解決，所以最後我將這個頁面的功能移植到了別的頁面並使用了別的方式呈現。

除了透過這一次的專題增進了自己打程式碼的能力外，還藉此次機會學習到的新的語言 Swift，並對一個 App 的架構多少有了認識，同時也透過網路爬蟲磨練了我原本不太熟的 python 和 node.js，也因此多認識了兩個套件 - BeautifulSoup 和 cheerio。另外，最讓我有深刻感受的是最初的提案報告、期中的進度報告以及期末的發表報告，這三次的報告從準備投影片到實際上台都讓我收穫不少，正如老師您所說，平常幾乎沒有太多能夠上台報告的機會，因此口才和上台報告的膽量等都很欠缺，經過這三次的磨練不僅讓我學會了如何將自己較為零碎的想法統整成一句話，並透過自己的話把我想表達的重點傳

達清楚，讓台下的大家都能够理解。

經過了這個學期的資料庫專題，我深刻的體會到了分工合作的價值與重要性，這個囊括前後端和資料庫的專題大小，真的沒辦理自己獨立完成。除此之外我也學習到了如何說出自己的想法與組員溝通、討論，藉此來得出彼此都能接受的結論。最重要的是讓我有了更多實作的經驗及作品，提升了自己畢業以後的競爭力。