

受訪者姓名：吳少宇學長

從系統分析與設計課程中遇到甚麼問題？如何解決？

系統分析是第一次有正式的分組然後進行一個很完整的專案開發，從寫文件到成品做出來。所以一開始遇到最大的問題就是沒有寫過這麼完整的東西，主要是技術上會有比較大的困難，至於怎麼解決就是上網多看多學囉，因為應該說學校到目前為止的課程教的東西是遠遠不夠你去完成一個完整的系統的。

(補充)定題目的範圍廣闊會影響到開發的難易程度嗎？

題目如果訂的很廣很多就是範圍很大你開發起來的工作相對會多，但是你也可以一開始先訂的比較大一點，然後因為老師不是會有一次一次的發表嗎？你可以透過每一次的發表、大家的回饋去慢慢修正你的系統

從系統開發中遇到甚麼問題？如何解決？

我自己覺得比較大的問題是大家的程式能力就是有強有弱啦所以可能你們分工出來之後其他人的成果可能會不如預期這樣。那解決方法就是要透過常常開會去追蹤大家的進度，有人遇到困難可能就是要多幫忙吧畢竟成績是大家的。

使用 git 嗎？遇到甚麼問題？

GIT 很好用，但是要大家都會用，記得在開發的時候大家各自開分支，然後在自己的分支裡做完，然後在開會的時候再把它合到主分支就是 Main 裡面，而且過程中可能會有遇到程式碼有衝突吧或是什麼的，你們就是在開會的時候一起解決它。然後盡量不要很多人動到同一個檔案，因為這樣可能會發生很嚴重的衝突會處理的眼花撩亂。

使用哪個語言、資料庫開發？為何？優缺點是？

我們那時候就是後端加資料庫就是用 Firebase 都處理掉，因為 Firebase 蠻方便，你可以直接在

前端的部分就寫好它的函式，它會幫你後端都處理得很好，你就不用再去煩惱這些東西，而且它的資料庫是 NoSQL 的，比較不用處理關聯式資料庫 Primary Key 然後 Foreign Key，那種關聯在 Firebase 裡面都沒有，它就是不用先設定好比較自由，以我們的大小來說也不會遇到那種讀很久讀不出來的問題，Firebase 很方便，然後前端的話我們那時候有嘗試過用框架像 Vue 或 React 那種，我是蠻建議用那種東西的，但是我們那時候大家有點學不來，因為我們也不太常就是不太有接觸過 JavaScript，所以像 Vue 或 React 這種框架又是更進階，你應

該要比較熟悉 JavaScript 之後再去使用的東西，所以我們最後前端是用原生的 JavaScript 語法去做開發的

，然後搭配基本的 HTML CSS。

JavaScript 的優點是它蠻好懂的，但是缺點就是一旦累積很多程式碼之後，它看起來會很亂很難維護，當你要修改某項功能的時候會很花時間，就是會很難維護，所以能學的話推薦還是用 React 或是 Vue 去做，早一點接觸也比較好，因為專題也會用到。

如何定期追蹤組員進度？遇到甚麼問題？

我們那時候開會的頻率是一個禮拜就會開一次會，然後開會最重要的是要知道你這次開會的目的是什麼，比如說前一次開會你已經要分配好每個人做哪些事情，所以我們開會最重要的就是看大家有沒有做到我們上禮拜分配好的工作，就是說如果有人沒有辦法趕上那個進度、完成大家分配好給他的工作，我們也不能做什麼，我們只能給他道德上、良心上的譴責，所以當你遇到有人每一個禮拜都在擺爛的時候，最後我們也只能把他的工作發出去給其他人，這樣其他的工作量就是相對的增加，我們唯一能對他做的制裁就是我們期末的時候，有一個那個分工表我們只會給他可能 5%到 10%。

( 到目前為止， ) 專題遇到的最大問題是甚麼？如何解決？

因為我們專題是先把文件上學期先把文件寫好，現在最大問題就是進度問題，如何解決就是拼命寫。你規劃一個系統需要先寫一個系統的計劃書我們都叫他系統文件，裡面就會寫你的系統的開發目的、要解決什麼問題然後系統的設計、使用者的流程、使用者會遇到哪些問題、會遇到哪些情境，然後你們提供什麼樣的服務，你要詳細的寫你整個系統要幹嘛。因為你不能不知道系統寫了是為什麼然後就一直寫程式。

(補充)那寫文件是類似一個發想的過程嗎，那這個想法是你個人有這個想法還是說你跟其他人就是共同覺得這個想法好像不錯然後決定做這個主題才開始寫的嗎？

當然你的主題一定是要跟全部的人都討論過，因為大家都覺得有意思有辦法做的下去，才能繼續往前走，因為如果有人覺得很無聊，那會不會做到一半他開始擺爛了。就是匯聚大家的想法然後開始寫文件。

( 到目前為止, ) 系統開發中遇到甚麼問題? 如何解決?

還是會有能力不均然後分工有點不公平的問題, 還是會有人比較不擅長寫程式然後我們有些人去看他的程式碼就會覺得頭很痛, 或是他開發進度就會很緩慢, 但我已經專題的組員都已經選下去了, 大家要互相督促, 我還沒有聽到有人做專題做到吵架的, 就是分工很重要。

使用 git 嗎? 遇到甚麼問題?

其實比較沒有遇到問題, 因為有 SA 的經驗之後, 你要確保你的組員都是知道 Git 怎麼用的, 那從 SA 就可以開始練習了, Git 很重要, 我之前用 Git 的時候我也覺得有點複雜, 所以還沒有摸懂 Git 到底要怎麼用的話, 趕快去摸懂。

使用哪個語言、資料庫開發? 跟 SA 所使用的一樣嗎? 為何? 優缺點是?

現在的前端用的就是 React, 因為他的元件化很方便然後還有 State 也很方便, 這些名詞去了解一下, 反正能用框架一定用。資料庫我們改用那個 MongoDB, 因為我們後端也不用 Firebase 了, 我們用的是 Python 處理後端, MongoDB 好像是很多人在用, 我們還沒摸透 MongoDB, 就只是因為我們想用 NoSQL, 因為他寫起來比較簡單一點。

如何定期追蹤組員進度? 跟 SA 一樣嗎? 遇到甚麼問題?

新的問題沒有遇到, 但是跟 SA 的差別就是經過 SA 之後我們比較知道開會要幹嘛, 因為我們當時在 SA 開會的時候很常大家坐在那邊不知道要幹嘛, 開會的時候一定要先想好你們討論的主題是什麼, 然後要處理什麼問題, 因為專題開會遇到最大的問題是我們很容易分心, 因為大家都熟的人, 然後可能講一講就開始聊天, 我覺得這樣有點浪費時間, 所以開會不要分心滿重要的, 然後要做什麼事情想清楚。那開會的主題舉例, 定期追蹤進度, 就是我們可能前一週的會議會先分好大家這一週要幹嘛, 每個人這一週的進度在哪裡, 然後開會的時候就是大家會講自己的成果, 然後有什麼困難大家可以出來討論, 或是有什麼一定要討論的問題可以討論這樣。

對各位的建議是?

這堂課跟你們以前遇到的課都截然不同, 它不是那種上課去聽老師講可以過的一堂課, 這堂課要花很多自己的時間上網學習知識, 所以大家要積極一點, 不然你的其他組員會很難過, 然後你可能專題就不會看到他們了。

---

觀看 Deeplink 學長姊與專題相關之分享

### 小專題那些事

聽完學姐的分享我認為和訪問學長得到的結論大同小異。相同的部分是上這堂課自學是很重要的一件事情。因為這堂課並不是像以前的前/後端設計老師已經給定一個主題然後做出來，是要自己從頭發現問題，然後自己找適當的工具去解決問題。還有團隊默契該如何培養，必須先取得共識，有共同目標希望團隊可以做出怎樣的東西，而不是用友情來維繫團隊。如果大家的目標一致就應該就事論事的把話說出來，講出來也是幫助團隊成品更好的一個關鍵畢竟每個人看事情的角度不盡相同。還有工作管理軟件可以幫助大家知道每個人的進度在哪，工作分配給誰，開會每次的紀錄，是一個非常實用的軟體。

### 專題分享!深度學習-居家智慧照顧系統

用到的東西:label img(用於深度學習影像標記(annotation) 的軟體)、python、linenotify

注意事項:版本相容問題，例如 python 每改一個版有些語法可能會不一樣

心得:我覺得其實學長講的東西主題內容偏少主要都在講過程中遇到的問題以及如何解決(也是這個時候的我們最需要的)，裡面我印象最深的是學校內的專題主要考量的是完成度，而校外的比賽有些是創意組的可以不用那麼在意完成度，這也是現在的我們需要注意的點，不要到時候理想太美好，現實太殘酷。至於其他部分就定期開會、設定目標、充分溝通以及共同努力。希望這學期的 SA 能借取學長經驗少走一些彎路。