**學習心得**

**學員：Bob\_Chen**

目前在本公司已經兩個月了，學習到的收穫非常多。

第一天踏進公司開始上班，對大家都是非常陌生，互相不熟悉不認識，透過早上的迎新小團康來認識彼此，互相熟悉對方，了解來自何處，有從事過什麼行業，及興趣等等，下午就開始展開我們的學習行程。

在第一階段的過程中，由於我一開始對於PHP語言不熟悉，加上在老師非常快速的上課趕課下，相當吃力，我只能利用空暇時間，例如：上班休息時間、下班後回家的時間，來進行加以補強，我很害怕跟不上進度。上課學習的日子漸漸過去，到了實作專題時間，先以bootstrap來進行套樣式底，再加以改造完成自己所訂的題目，這個過程中從無到有，也了解到中間很多的產生過程。

從最原始的HTML及PHP混著寫在同一頁，慢慢地改變版本，變成擁有物件導向的寫法，最後完成MVC的架構，並且了解M(model)、V(view)、C(controller)分別的用意是做什麼的，model負責處理資料庫及判斷，view就只要負責顯示，controller就是主要接收指令並且跟model要東西再回傳給view。我的想法是controller很像超商店員、view是客人、model是倉管，店員依照客人要的東西，像倉管拿取詢問是有這東西，最後再將答案和東西給客人。

在第一次評鑑的時候相當緊張，很怕自己會出槌，還好最後順利地完成這次的報告，也透過主管前輩們的問與答了解自己那些方面不足加以補強。這個實作作品，是我PHP第一次完成的作品，成就感非常的大！

在第二階段分為四個禮拜，每個禮拜分別都有不同的部門主管前輩們下來出題目。第一個禮拜是研二部門，首先出的作業就是將第一階段的題目改得更完善一點，並要求MVC架構更清楚明確以及一些語法上的使用該注意的地方，資料庫防範問題使用PDO及SLQ 語法都是大寫和，欄位前後加上 ` ，如 `欄位`，例如： "SELECT \* FROM `NewActivity` WHERE `id` = :ActivityID"

後來出二個加分作業是陣列及報名系統，我是做報名系統，先將基本功能做出來，在考慮報名時間是否超過或者未到、限制總人數、可否攜伴、攜伴人數上限、誰可以參加等等功能，做這作業時候很快就將基本功能完成並形成MVC架構，因為在上個作業步驟耗了滿多的時間在了解MVC內容上。最後還要考慮同時點擊參加或者相差幾毫秒點擊參加，收到的資訊是一樣的，最後送出資料會有誤，一開始想到使用的是FOR UPDATE來進行排隊等待，最後發現如果一個人花費時間0.1秒，只有少數人使用不會影響太大，但是如果人數上萬人搶票報名，等待時間會超級久，這樣就是有問題了。在思考的同時已經進行到下一個禮拜。

第二階段第二禮拜，研發五部，要大家都一致的Coding style風格，讓大家都可以看得更清楚，一看就懂，以及程式要好閱讀，閱讀性高。這禮拜讓我更深入的了解Git的關西，善加利用Git版本控管，讓commit更清楚知道這版本是在寫什麼改什麼的，一個commit只做一件事情，完成一個小段落就commit一次，避免大量修改不同地方都在同一個commit裡面。

作業是寫出簡易銀行系統，每個不同帳戶可以出款、入款、查詢餘額及帳戶明細，基本功能寫完後，一開始一樣是使用FOR UPDATE來做基底，最後回到上禮拜問題，因此找到樂觀鎖的方式來解決排隊衝突問題，用版本來確認目前是最新還是舊的，最新的就給他執行，舊的就告知他無法執行資料有誤。以及在更新資料庫裏面的金額，加減法運算直接在SQL語法內執行，例如：

`balance` = `balance` - :tradeMoney

直接把餘額這個欄位的金額做減交易金額運算。加上beginTransaction搭配commit以及失敗的rollBack，只要中途有錯誤就讓他直接還原到最初的狀態，完成後commit才算完成所有動作，否則也不算完成這筆交易，不會執行。最後再將程式碼能簡化都都簡化掉，盡量不使用else，直接先將錯誤放在前面讓他直接離開後面就不會做，例如 if ($userId != null && $tradeMoney != null) {

( 執行動作 )

}

Else {

請輸入正確資料

}

將它簡化改成

if (!$userId || !$tradeMoney) {

請輸入正確資料

}

( 執行動作 )

將所有程式碼能簡化就簡化，改善她得執行效率，以及可閱讀性。

寫出單元測試，來測試我們的所有function內容的執行是否正確，一開始寫是自行輸入金額多少，還要自己去察看資料庫目前餘額多少來進行改動，非常的麻煩，最後解決辦法是建造一個假帳號資料，先行setUp設定初始，最後再tearDown將所有資料還原，並將每個可能性發生的狀況都寫進去，讓它涵蓋率100%，全部都測到。

第二階段第三個禮拜，研發一部，第一天做出簡報技巧的整理心得，再來就是作業一作業二，我是選擇作業一，抓出下注網站內的數據，並存入資料庫，再使用快取進行存取使用。

這個作業是我覺得目前最難的作業。

一開始使用Xpath試著去抓資料，試了很多次沒有辦法，後來使用CURL去抓全部內容，有抓到其他部份的資訊，但是數據那部分抓不到，試了很久，後來發現他是使用iframe再另外鑲入網站，最後有抓到裡面那個網站，但因為內部有寫判斷被擋下來都被導頁離開，試過幾種偽裝方式進去，仍然不行，最後試到了用cookie方式，成功進去抓到要的所有數據，再依照要的內容依依的切割，最後成功獲得數據存數資料庫。

快取部分兩種Redis和Memcached，我選擇的是Redis，起初先了解它的方式，最後使用到的是，set('todayGame', json\_encode($record));，將資料用JSON格式存入放在'todayGame'內，最後要取出 $result = $redis->get('todayGame');，將內容取出。

第二階段第四個禮拜，研發三部，這禮拜是作業做多的一個禮拜，但是都滿簡單的很容易應付。第一天就很開心得大家先玩踩地雷，了解原理，然後下午就開始寫出亂數產生地雷，並判斷周圍的數字，由於以前寫過老鼠走迷宮，原理類似，很快就寫完這部分，只是稍微粗心了點，有些小細節沒注意到導致判斷錯誤，變成很後面才完成。再來寫出反驗證踩地雷程式，寫出判斷出地雷內部哪裡有問題，以及它是否有輸入特殊符號、炸彈數量、大小寫等等問題。一樣先繪製流程圖，先大概了解原理更清楚在進行下一部，試著操作API使用方式，找出BUG，最後自己自做屬於自己的API，進行一連串驗證，是否正確等等。

總結，從一開始不懂PHP到現在可以寫出這些作品，真的很替自己開心，有所學到東西，在其中也了解，注重的是資料庫維護防範，還有Coding style閱讀性問題，Git版本控管問題，程式碼簡化，思考邏輯能力，自己找出問題自己解決，以及平台與平台間的切換API，不可以有疏失，否則變成洗錢問題之一。

這兩個月來，所學習的內容相當豐富，不僅學到了專業知識，也結交了許多同事朋友，也懂得在團隊生活下完成任務，更懂得時間規劃，多少時間多少產值，如何去改善增加產值，如何有效率完成事情。

謝謝各位主管前輩們，謝謝老師，謝謝各位同仁。