

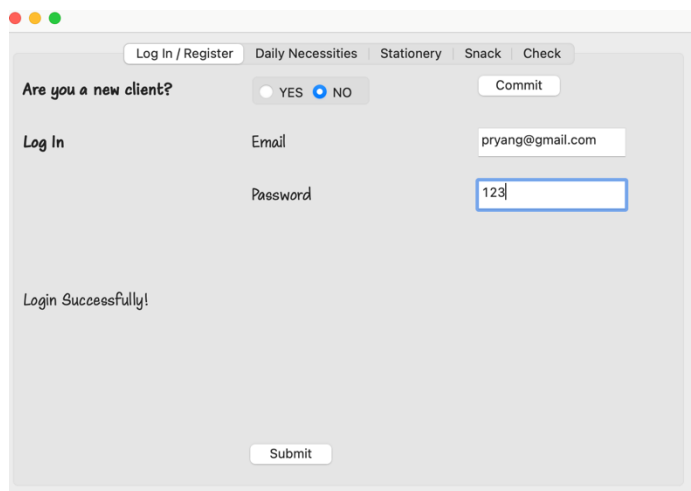
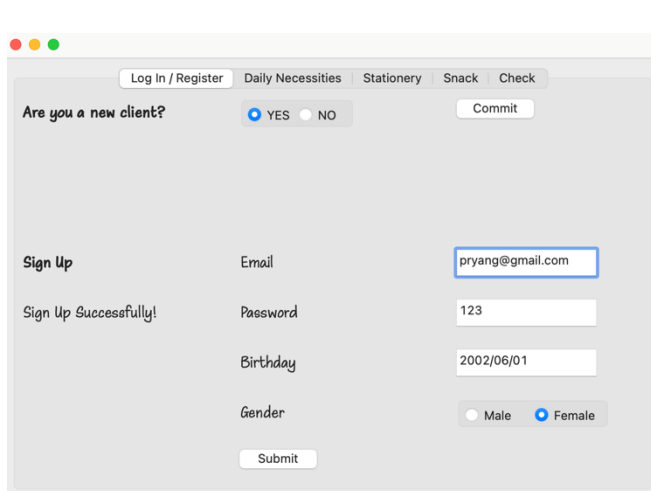
期中專題報告—小電商實作

一、題目發想：

當時在想專題題目的時候，剛好在和同學一起逛蝦皮，所以就決定做一個買東西的小介面。而這個介面也提供一些蝦皮有的服務：註冊登入、買東西、留評論、儲值虛擬錢包等等。

二、功能說明：

1. 登入/註冊頁面：一開始的登入/註冊介面會有個選擇框，詢問是否為新顧客。如果是新顧客，就要先註冊；如果不是，就要先登入。註冊的信箱只能用 **Gmail** 信箱，如果不是用 **Gmail** 信箱註冊，介面會出現一個提示訊息，告訴使用者信箱須包含“@gmail.com”。而如果在登入的時候密碼有錯誤，介面也會出現提示訊息告知使用者密碼錯誤。當註冊或登入成功時，介面都會出現「成功」的標示。



2. 購買頁面：總共分成三種：“Daily Necessities”, “Stationery”, “Snack”（日常用品、文具、點心）。每個頁面都有 2 或 3 種類別：“Daily Necessities”中有 3 種類別分別是：“Kitchen”、“Bathroom”、“Room”；“Stationery”中有 2 種類別分別是：“For school”、“For office”；“Snack”中有 2 種類別分別是：“Cookies”、“Chocolate”。每個頁面的每種類別都有 8 種選擇。類別的右方欄框是給使用者輸入要購買的數量。當使用者在一介面選擇好後，要先按下介面左下角的“Check Storehouse”，系統就會去「倉庫」查看是否有足夠多的數量可以提供給使用者。如果商品都足夠，就會顯示“All products are available!”，如果不夠，就會顯示剩下幾個。（例如盤子只剩下 10 個，介面就會顯示“Plate only 10 left”）。使用者再依據剩下的商品數決定是否要降低購買的數量或是選擇其他商品。當最後商品數都確定足夠後，使用者就可以按下右下角的按鈕（Add to Chart）將商品加入購物車。加入購物車後，原本選擇的商品及商品數量會全部回到初始值，使用者可以再繼續購買同一類的別種商品。

- 結帳介面：最後將要買的全部商品都加入購物車後，就可以進入到“Chart”介面結帳。使用者可以先按左上角的按鈕（**Show My Chart**）使按鈕右邊的文字框顯示購物車內容。購物車內容會顯示使用者買的商品種類、數量、每個商品的總價錢、全部商品的總價錢。使用者確認後按右上角的結帳按鈕（**Pay Now**）來付錢。而如果使用者的帳戶錢包不夠支付這次花費的總金額時，介面會顯示使用者的錢包目前有多少錢，並請使用者先加值。填入加值金額後，按下右邊的按鈕（**Add**）後，介面會顯示加值成功的標示。之後再按下結帳按鈕後，系統會再檢查錢包的前夠不夠支付，如果錢包裡的錢夠，就會顯示“Done! Thank you! Have a nice day!” 標示，表示使用者購買成功。

- 使用者留言回饋：購買結束後，使用者可以在“Leave Comment”的右方欄位裡留言，平台後端的資料庫就會收到使用者的留言。之後使用者再按下右下角的“Bye Bye”按鈕關閉介面。關閉介面後，使用者就會收到 Email。如果有留言，Email 的標題就會是“Comments Received!”；如果沒有留言，Email 的標題就會是“Thanks for buying our products!”。而 Email 的內容會只有使用者買商品的明細。

Thanks for buying our products! :



clairepryang@gmail.com

寄給我 ▾

Hope you enjoy our products!

Here is the receipt :

Turner 3 : 300 dollars

Total : 300 dollars

↩ 回覆

➡ 轉寄

Comments Received!  



clairepryang@gmail.com

寄給我 ▾

Thank you for leaving the comments! Hope you enjoy our products!

Here is the receipt :

Plate 8 : 800 dollars

Tissue 15 : 3000 dollars

Highlighter 5 : 50 dollars

Envelope 5 : 150 dollars

Oreo 3 : 147 dollars

FERRERO ROCHER 1 : 200 dollars

Total : 4347 dollars

↩ 回覆

➡ 轉寄

三、實作技術：

1. wxPython GUI
2. SQLite database
3. Read/ write files
4. Regular expression
5. Sending email via python

四、自我評估：

1. 在將程式和資料庫連結的時候，發現資料庫的數據都不會改變，後來慢慢測試才發現原來是程式沒有抓到使用者介面的選擇欄資料。上網查了一下發現，我以為的“GetSelection”函式會回傳選擇的值，也就是內容本身，但是其實“GetSelection”只會回傳選擇資料的索引值。所以還需要用“GetString”函式轉換成選擇的內容。
2. 另外在測試的時候，發現如果在同個頁面中，如果只選擇一項類別的商品，而不選擇另一項商品，程式就會報錯，會顯示「抓不到資料」。但是一方面我的想法是，如果我沒有選擇商品，程式就不用去抓取資料了。我後來才想到可以設個預設值，使他的索引值為-1。如果索引值為-1的商品，就將其商品跳過，不考慮。
3. 在將購買商品資料呈現在 GUI 上的時候，讀寫檔案花了一段時間。因為我預想的檔案呈現方式是有換行的，但是放到 GUI 上後，資料都在同一行。所以我又花了一段時間研究 wxPython 的 TextCtrl 的官方文件說明。後來才發現，TextCtrl 需要再設一個「多行」的屬性才能達到預期的效果。
4. 這次的期中專題讓我更熟悉 wxPython 及資料庫如何使用，以往除了課堂作業之外，自己都沒做過類似的專題。希望在下半學期，能夠學到更有趣的技術，且在第二次的專題中能夠做出更好玩、有趣的作品！