**一、技術主題**

**何謂 MVC？**

Model-View-Controller(MVC)是一種設計模式(design pattern)，主要目的是用來簡化應用程式的開發與增強程式的可維護性，其作法是將應用程式分割成以下三個邏輯的元件:

**Model** -主要負責應用程式中的商業邏輯(Business Logic),商業邏輯是一個非技術用語，用來描述應用程式功能性的演算法以及資料庫與使用者介面之間資料的交換。Model層封裝了應用程式中對資料的存取並提供可重複使用的函式庫，像是資料庫存取的抽象化，郵件的遞送，資料的驗證與稽核都會發生在此。

**View** -簡單解釋就是我們用來呈現自Model層的資料與蒐集使用互動資料的網頁，HTML/CSS/Javascript都是使用於此層的技術。

**Controller** -蒐集使用者於View元件中所輸入的資料並決定由哪一支程式進行資料的處理。接收Model元件所回傳的資料並在解析後傳遞給View元件作呈現。所有程式的例外處理以及流程控制。

使用MVC來進行網路應用程式開發有很多優點，包含程式碼簡潔與程式升級的彈性，但最為人稱讚的優點在於可以將一個專案的開發分成三個不同的角色來增加協同作業的效率。

**Development** -對應到MVC中的Model元件，這部分主要是由熟悉伺服器程式語言(ex:PHP/Python/Ror/ASP.Net/Perl…etc)，資料庫管理與設計，資訊架構，演算法，以及資料驗證等技術的工程師所負責建構應用程式運作細節並提供APIs以及如何與資料互動的規則。

**Design -**對應到MVC中的View元件，一般來說負責此部分的是熟悉圖像製作，HTML、CSS以及Javascript的視覺設計人員。

**Integration -**對應到MVC中的Controller元件，整合者主要負責將視覺設計師所開發出的靜態頁面轉換成嵌有程式的動態頁面，將資料利用表單發送到Model Layer，接收並轉換Model Layer所回傳的資料並呈現在View Layer之上。

舉個例子來說，工作流程是這樣，Designer會先根據企劃部門所提出的企劃輸進行網頁的圖面設計，圖面設計完之後會交給Developer,Developer會評估功能需求是否可行且符合公司的隱私與安全性政策，一但確認之後，Developer就會著手開發存取後端資料庫的APIs，最後網頁靜態頁面與APIs都會交予到Integrator手上進行套板的工作，不論是採用將程式內嵌入網頁(ex.Plain PHP)或者使用樣板引擎(ex:Smarty)，Integrator會處理自網站前台所傳遞來諸如表單，URL Request參數，Cookie等等資料並將其傳遞到後端伺服器進行處理，最後將伺服器端所傳遞回來的資料進行處理後呈現在前台的動態網頁上。

MVC的缺點是由於它沒有明確的定義，所以完全理解MVC並不是很容易。使用MVC需要精心的規劃，由於它內部原理比較複雜，所以需要花費一些時間去思考。

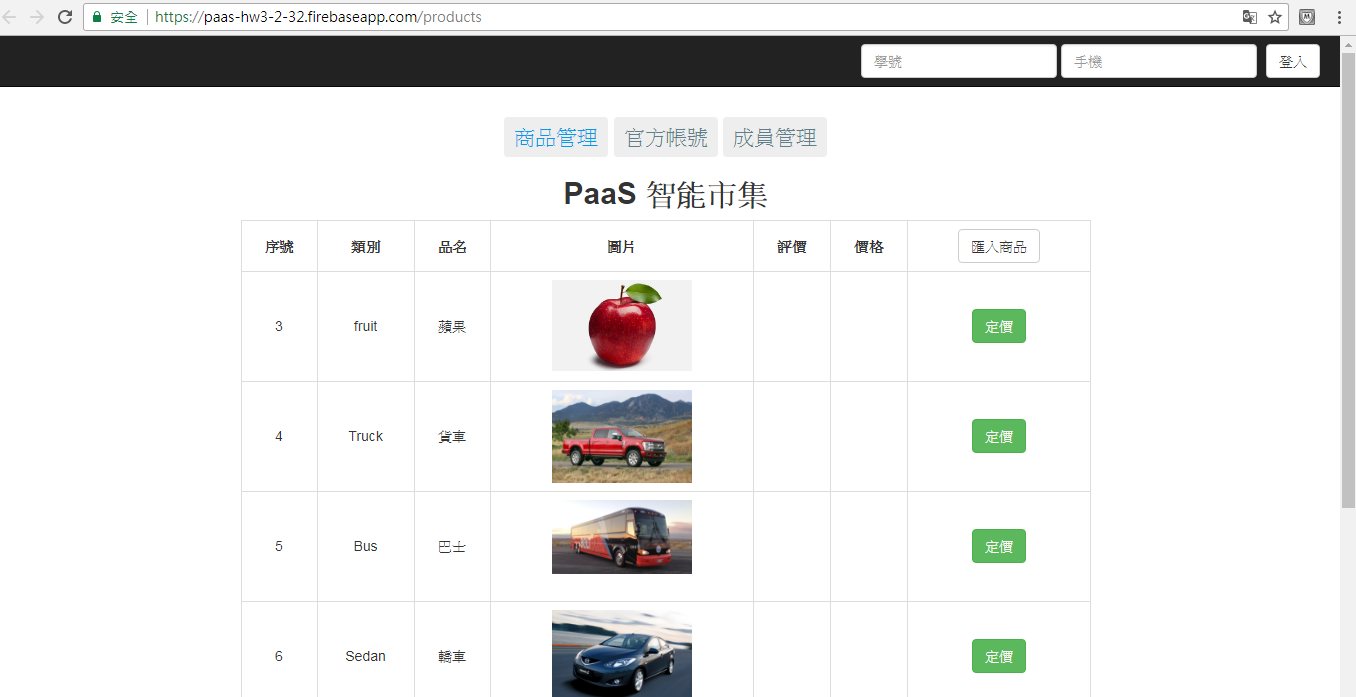
你不得不花費相當的時間去思考如何將MVC套用到你要開發的東西商品，同時由於模型和顯示要嚴格分離，這也是一件相當困難的事情。每個物件載使用之前都需要經過測底的測試。

由於我們將一個要開發的軟體分成了三個部分，所以使用MVC同時也意味著你要管理比以前更多的文件，這一點是顯而易見的。這樣好像我們的工作量增加了，但請記住這筆他所帶來的好處是不值得一提的。

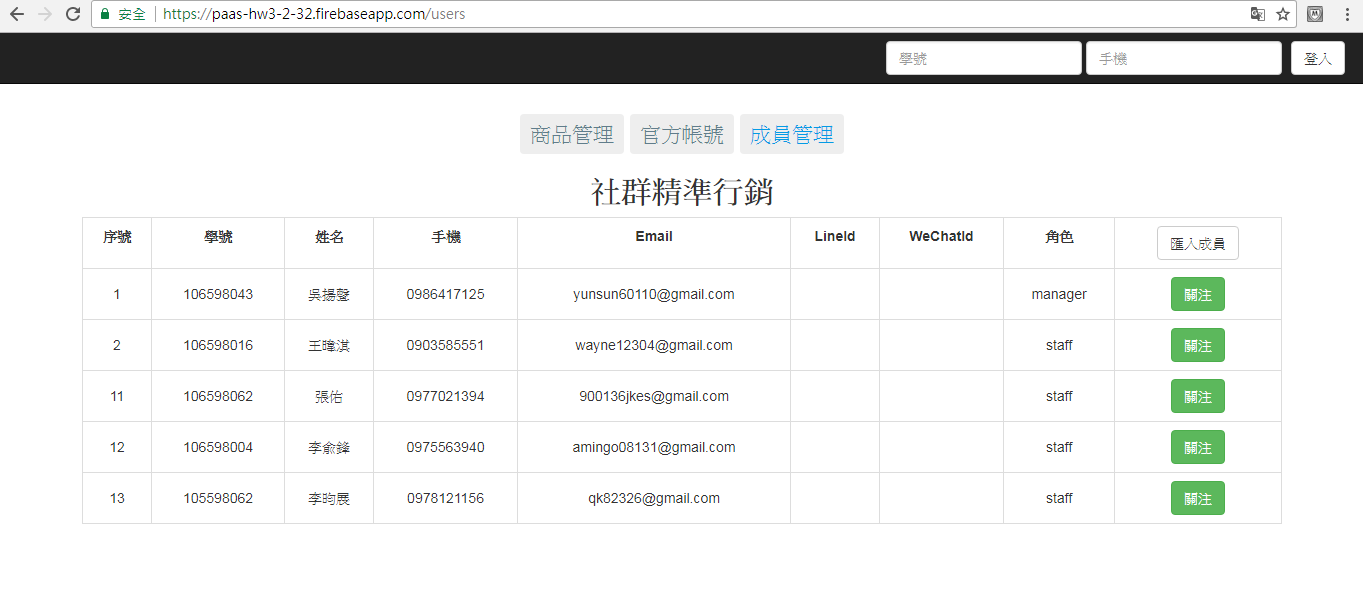
**What is AngularFire?**

* **Observable based** - Use the power of RxJS, Angular, and Firebase.
* **Realtime bindings** - Synchronize data in realtime.
* **Authentication** - Log users in with a variety of providers and monitor authentication state in realtime.
* **Offline Data** - Store data offline automatically with AngularFirestore.
* **ngrx friendly** - Integrate with ngrx using AngularFire's action based APIs.

**二、作業輸入、輸出畫面**







**三、參考資料**

1. 什麼是MVC? What is model-view-Controller?

<http://expect7.pixnet.net/blog/post/36446413-%5B%E8%B3%87%E8%A8%8A%5D-%E4%BB%80%E9%BA%BC%E6%98%AFmvc%EF%BC%9Fwhat's-model-view-controller%EF%BC%9F>

1. The Angular Firebase Survial Guide技術文件

<https://leanpub.com/angularfirebase/read_sample>

1. Angular框架官網

<https://github.com/angular/angularfire2>

**四、學習心得**

我認為MVC並不適合小型甚至中等規模的開發程序，因為會花費大量的時間將MVC應用到規模並不是很大的開發程序通常會得不償失。

在3-1還有3-2的作業當中，起初在3-1的當中，因為Firebae的環境沒有改好，所以遲遲沒有跑出新增的功能，然而在與同學的討論之下，第一個禮拜也沒有如願的把功能做出來，直到了在開始做3-2作業的時候，總算能夠把新增這個功能給順利得做出來了，並且如期的繳交，在這當中，我不僅學到了與同學互相討論、互相幫助、互相研究的精神，最重要的是你想要把東西解決出來，自己會利用手邊的資源(ex.google)，想盡辦法想要把它做出來的感覺，我認為這是更可貴的，畢竟現在要如何善用GOOGLE，來幫助自己，這也是很重要的技術之一。