

## 2-2-4、Seagull Framework 與 PEAR 的關係

Seagull Framework 使用 PEAR 的套件來開發，讓開發者可以節省許多撰寫程式碼的時間，又可得到強大的功能。這個網站架構以目前 0.3.7 版有使用到的 PEAR 套件有：

1. Archive：可用 PHP 來處理 tar 檔。
  2. Benchmark：可計算網頁腳本程式執行的時間。
  3. Cache：可產生快取。
  4. Config：可幫助管理設定直的工具。
  5. Console\_Getopt：可用 PHP 做文字指令的解析。
  6. Date：可根據地區來決定該地區的日期與時間的時區
  7. DB：資料庫的抽象層。
  8. DB\_DataObject：可根據資料表自動產生實體物件。
  9. DB\_NestedSet：提供方法來建立或刪除或移動資料庫的節點。也提供存取節點的方法
  10. File\_Find：提供搜尋檔案的功能的套件。
  11. HTML\_Javascript：提供常用的 javascript 碼。
  12. HTML\_TreeMenu：提供可簡單產生樹狀目錄結構網頁的套件。
  13. HTML\_Template\_Flexy：網頁模版系統的套件。
  14. HTTP\_Header：提供設定 HTTP 表頭的方法。
  15. HTTP\_Request：提供一個簡單的方法來產生 HTTP 的 Request。
  16. Log：提供一個產生記錄的方法。
  17. Mail：提供傳送電子郵件的方法。
  18. Net\_UserAgent：可偵測使用者的瀏覽器類型與版本。
  19. PHPdocumentor：可用 PHP 來自動產生說明文件。
  20. Text\_Password：可用 PHP 自動產生密碼。
  21. Translate：提供翻譯文字的方法的套件。
  22. Validate：驗證表單被使用者所填寫的內容格式是否正確的套件。
- 共二十二個。

未來還計劃運用其他的 PEAR 套件，希望達到以下的目標：

- 用 xml-rpc 套件建立一個安裝與升級模組的裝置，類似 PEAR 的套件管理網頁，在 Seagull 中的每一個功能都以模組化的方式寫成，因為採物件導向設計，所以每個模組都可分開，若有一個模組升級，

並不會影響其他模組。

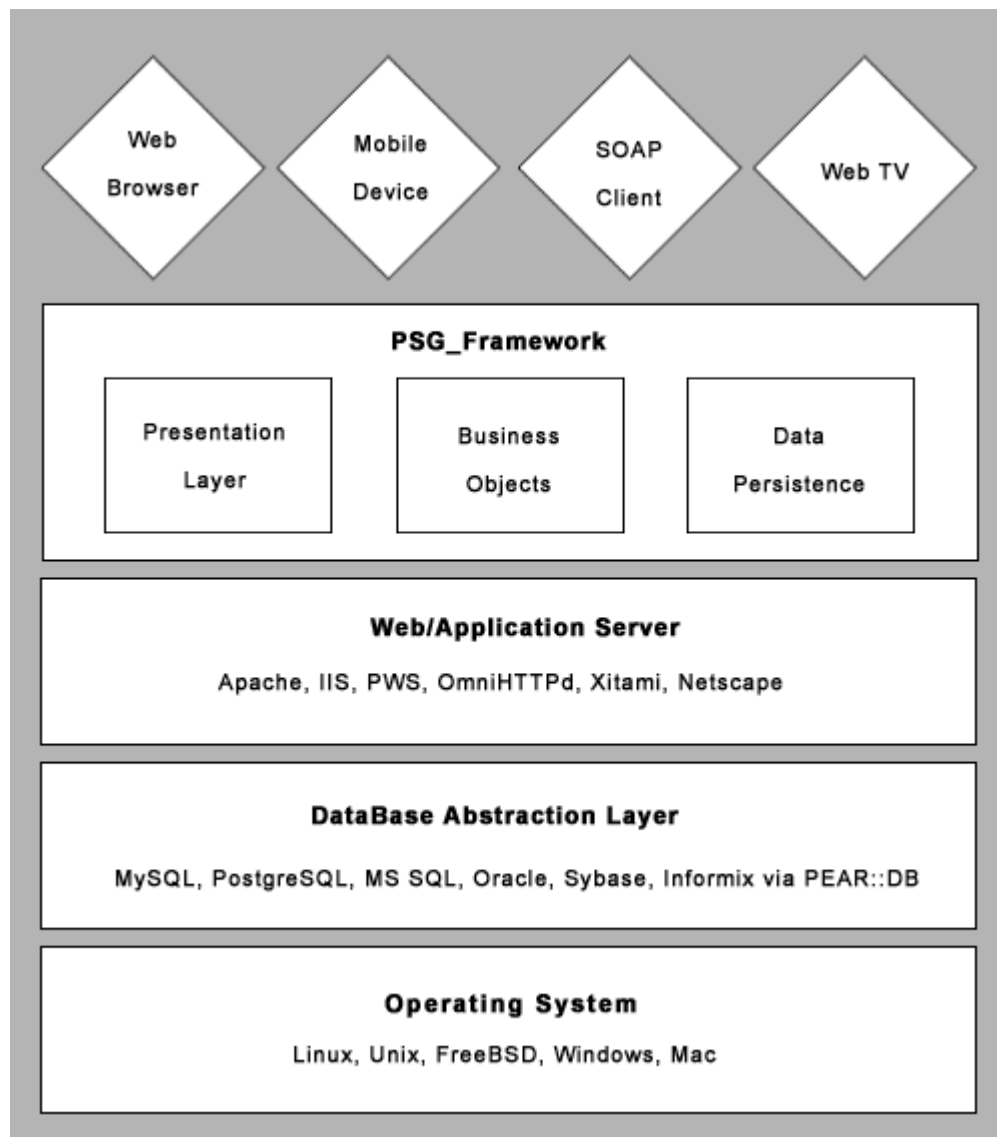
- 藉由 CHAPS/HMAC 套件來建立安全的登入機制。
- 用 HTTP::Upload 類別來取代目前的上傳檔案機制。
- MS Word 的文件可以匯入到文件管理介面中。
- 考慮將資料採用 xml 的格式來供各種不同的裝置都可使用，不只限於瀏覽器。
- 引進發佈文件的版本功能，讓發佈的文件自動產生版本編號。
- 文件管理機制將引進更先進的檔案格式的解析，目前只有六種檔案格式可以上傳，未來希望更多種類的檔案都可上傳，例如：MP3，EXIF 等格式的檔案。
- 若發現 PEAR 有更新的套件，可自動升級。



## 2-2-4、Seagull Framework 系統各種圖形

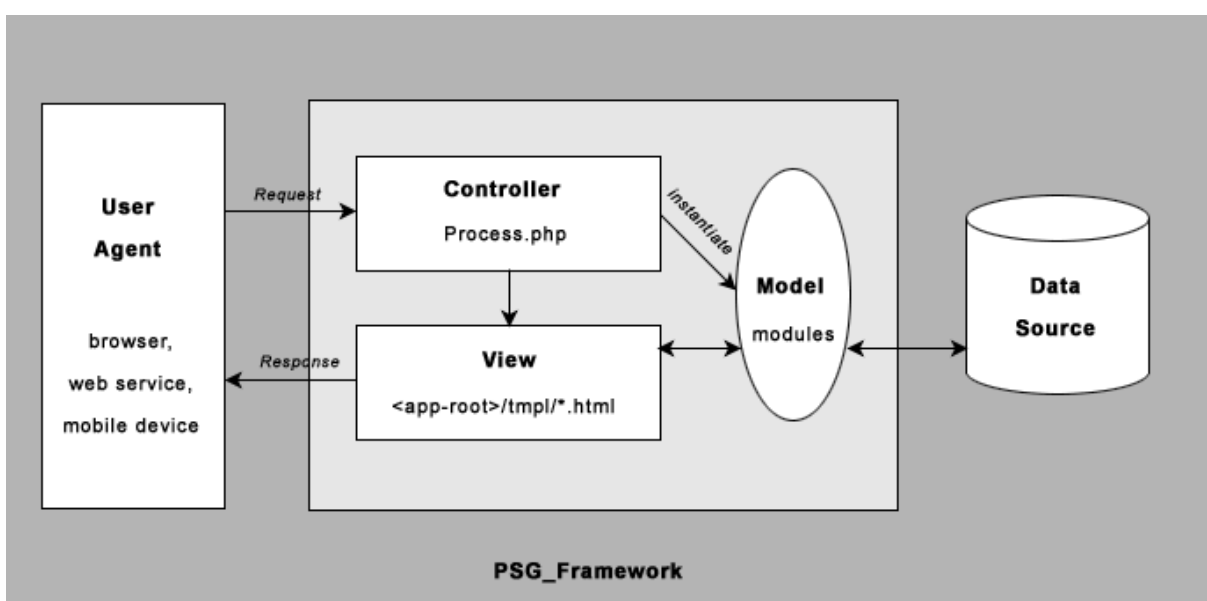
### 2-2-4-1、Seagull Framework 系統綜觀圖(System Overview)

由此圖可看出，最底層代表 Seagull Framework 可相容於各種不同的作業系統，接著在上一層植基於作業系統之上的是資料庫的抽象層藉由 PEAR::DB 來存取資料庫，許多不同種類的資料庫皆可使用同一個抽象層不管是免費的或商用的資料庫。再上一層為網站伺服器的平台目前可支援的有 Apache，IIS，PWS 等，又上一層才是 Seagull 架構，內含三個部份分別為，展示層，商業物件(或稱邏輯物件)，實體層將不同的功能分割為不同的層次方便開發者分工合作。最上面是各種使用介面，包括瀏覽器，行動裝置，SOAP(Simple Object Access Protocol)的使用者，網路電視，如圖五十三。



圖五 系統綜觀圖(取自 Seagull Framework 官方網站)

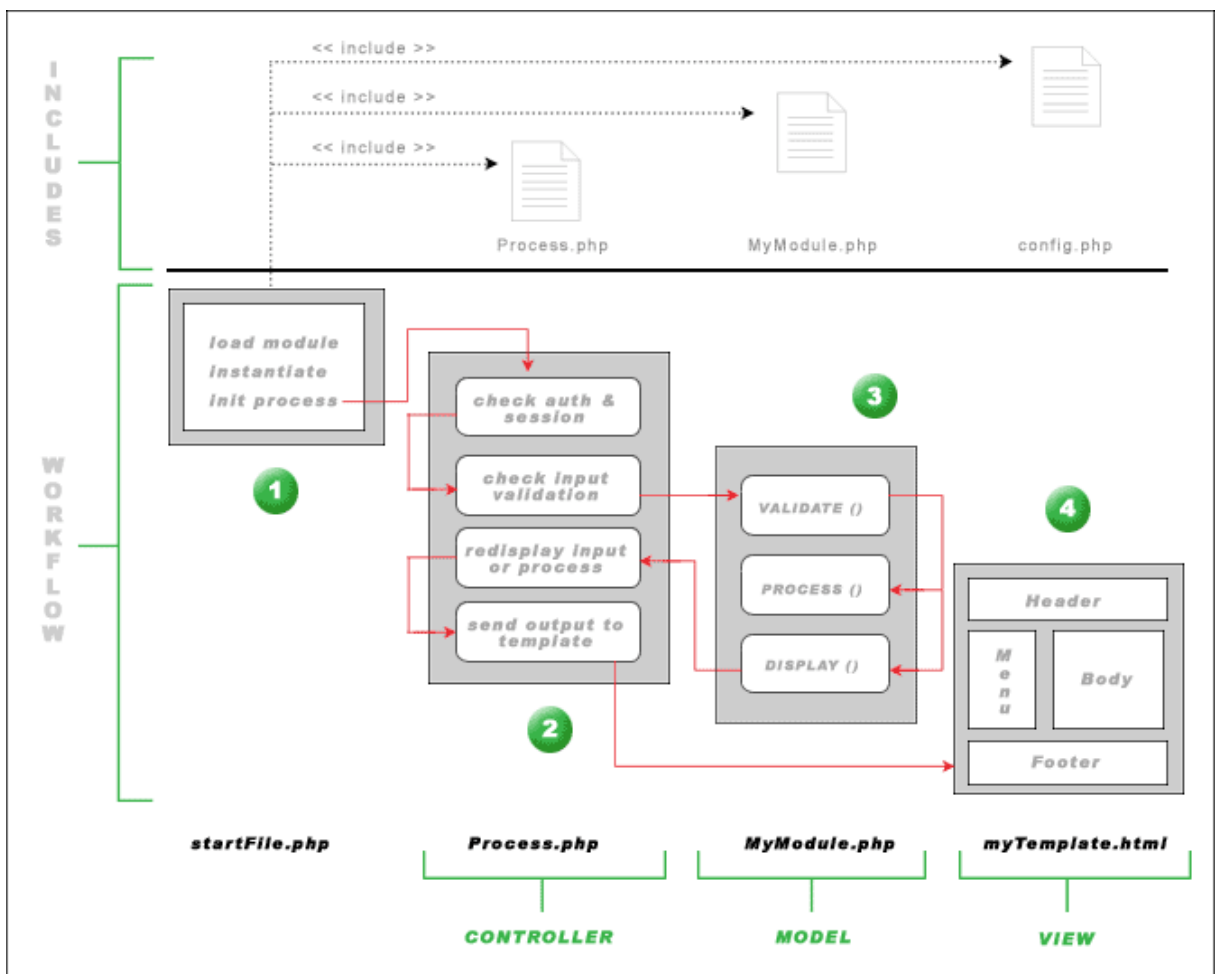
再用另一個觀點來看系統模型，Seagull Framework 是與資料庫分開的，如下圖，三個模組構成了整個架構，分別是控制項(controller)，外觀(view)，模組(model)，當使用者 request 進來後，控制項會判斷 request 的內容，以決定要用那一個模組來處理，控制項會將要處理的模組實體化，若有需要模組會去資料庫存取資料。模組處理的結果會傳送回控制項以決定下一個流程，若流程已跑完則會傳送給外觀類別來處理，將處理結果結合模板變成網頁輸出於螢幕上，如圖六。



圖六 控制項的模型架構

#### 2-2-4-2、網頁執行的流程圖(Page Workflow)

圖中可看出 Seagull Framework 是運用控制項(controller)物件來控制各個模組，控制項是用來判斷並決定流程的，由判斷的結果來決定該呼叫那一個模組來做後續的工作，再將模組執行的結果傳回控制項做判斷下一步要呼叫那一個模組，依此方式直到整個流程結束，例如若使用者登入，將輸入的文字或數值，送往 validate 模組查核身分，檢查輸入的欄位是否正確。若傳回值為正確，則導向顯示網頁畫面。模組是功能的所在，但必須由控制項來啟動。Seagull 設計上都將控制項放在一個網頁，模組放在另一個網頁，輸出的內容又是一個網頁，不混雜在一起，這是開發者將程式分門別類的好設計，這種設計模式值得學習，如圖七。



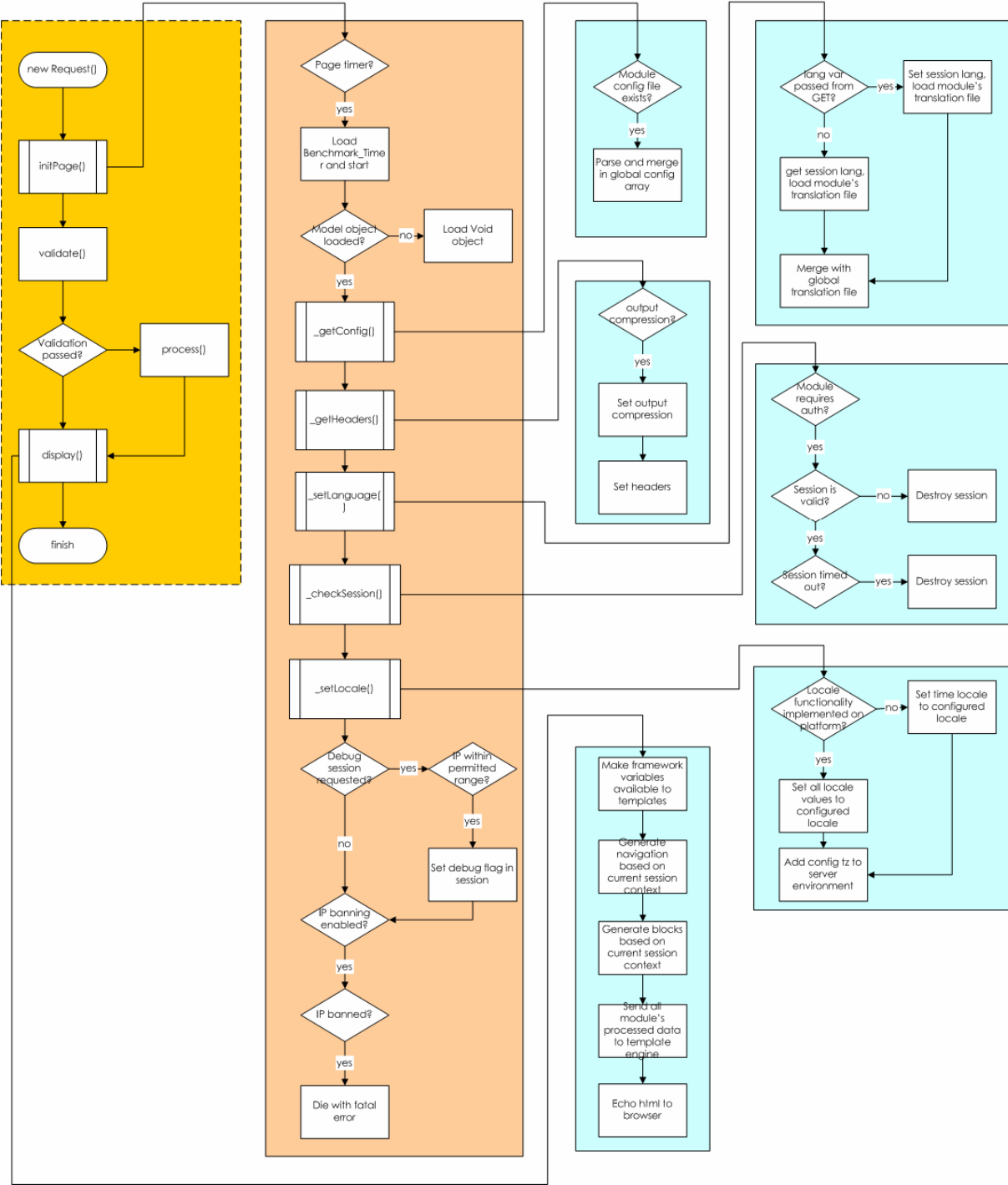
圖七 網頁執行流程圖(取自 Seagull Framework 官方網站)

### 2-2-4-3、控制項流程圖(Controllor Workflow)

Seagull Framework 的官方網站提供了首頁的控制項物件的內部詳細流程圖。與圖六的差別在於上圖為一種示意圖，而下圖是實際的執行過程的流程圖。不同顏色的區塊代表不同的網頁。可清楚看出將各項功能切割的結果，可讓開發者很直覺的看懂程式碼，日後除錯或增加新功能也很方便，更重要的是不會讓程式碼混雜在一起。

由此圖可看出，Seagull Framework 的開發者群們前置作業相當仔細，先做過全盤的思考，討論，將之形成圖案，才開始動手分配工作給所有開發者撰寫程式。若將此方式移植到化學教學網站的開發，將有助於網站讓許多人參與製作，或者後續有人中途參與網站的開發也可以藉由圖形的說明很快進入狀況，如圖八。

# Seagull Controller Workflow



圖八 控制項的流程圖(取自 Seagull Framework 官方網站)

### 2-2-4-4、UML 的順序圖(Sequence Diagram)

所謂的 UML，目前軟體工程學中，針對物件導向的程式開發，提出的

一套撰寫軟體藍圖的標準語言，稱為統一塑模語言（Unified Modeling Language，UML）。所謂的 UML 根據 Grandy Booch，James Rumbaugh，Ivar Jacobson(1998)三位物件導向系統分析方法的學者所下的定義如下：

UML 是撰寫軟體藍圖的標準語言

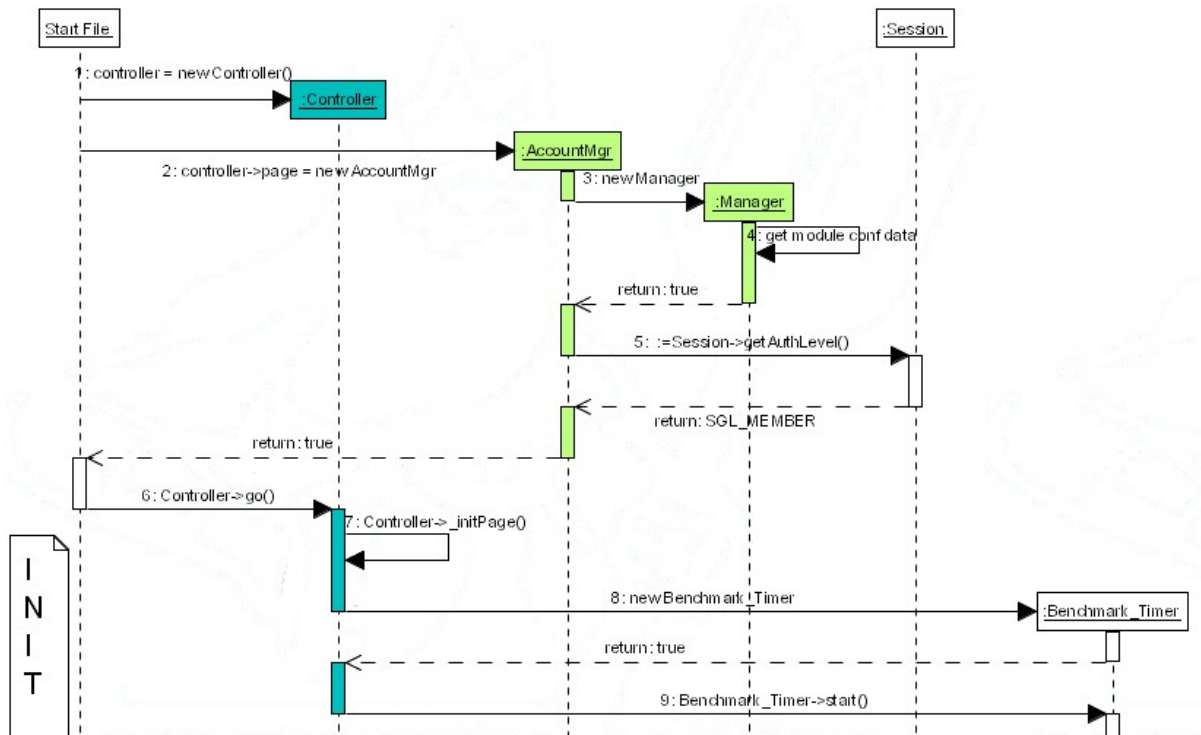
UML 可以視覺化方式訂定，建構以及記錄軟體系統的產出。

就如同蓋房子一樣，若沒有事先將建築藍圖畫好，做好整體規劃，將來怎麼蓋的出功能齊全的好房子。軟體也是一樣，沒有先將設計圖畫好，怎能建構一套好的應用程式。

UML 的產出，包括文件，檔案，程式以及函式庫，最重要的是圖形等等。這個語言定義了許多圖形，這些圖形可用來建構軟體的架構，也就是所謂的「塑模」。用圖形來塑模有簡單，易懂，易溝通等好處。目前有許多廠商開發出 UML 的繪圖軟體，例如：MicroSoft Visio，IBM 的 Rational Rose，免費的共享軟體 ArgoUML 等，其中 Rational Rose 可以將圖形直接轉換為程式碼，功能最強大，可以節省非常多的時間，但也相對的昂貴。

最常使用的 UML 標準圖形有：使用案例圖(Use case diagram)，類別圖(class diagram)，順序圖(Sequence diagram)，狀態圖(statechart diagram)，活動圖(activity diagram)，元件圖(component diagram)，部署圖(deployment diagram)。每一種圖代表一種系統架構的觀點，為何需要這麼多種圖形，因為單靠一種圖形無法將整個架構表現的很清楚，透過不同圖形所代表的不同的觀點可以讓客戶與開發者互相溝通與瞭解，並進一步做好開方前的前置規劃。

以下從 Seagull Framework 官方網站所提供的就是 UML 的順序圖，順序圖是以實際程式執行的順序所繪製的圖形，其中白色方框代表 PEAR 物件，負責系統最底層的工作，深綠色方框代表控制物件，用來控制流程的走向，淺綠色代表模組物件，是實際執行工作的單位，透過箭頭的方向與編號，可以瞭解程式執行的順序與物件間的關係，如圖九。



圖九 Seagull Framework 首頁的順序圖(取自 Seagull Framework 官方網站)

順序圖已經可以隱約看出程式最細微的實際執行方式，對開發者而言，有了這個圖就可以著手撰寫程式了。

補充說明另外兩種非常重要的 UML 圖形，名為使用案例圖與類別圖。

若要建構一個架構好又具發展性的網站，除了之前的圖形還需要製作這兩種圖形。

#### 2-2-4-5、使用案例圖 (Use-case diagram):

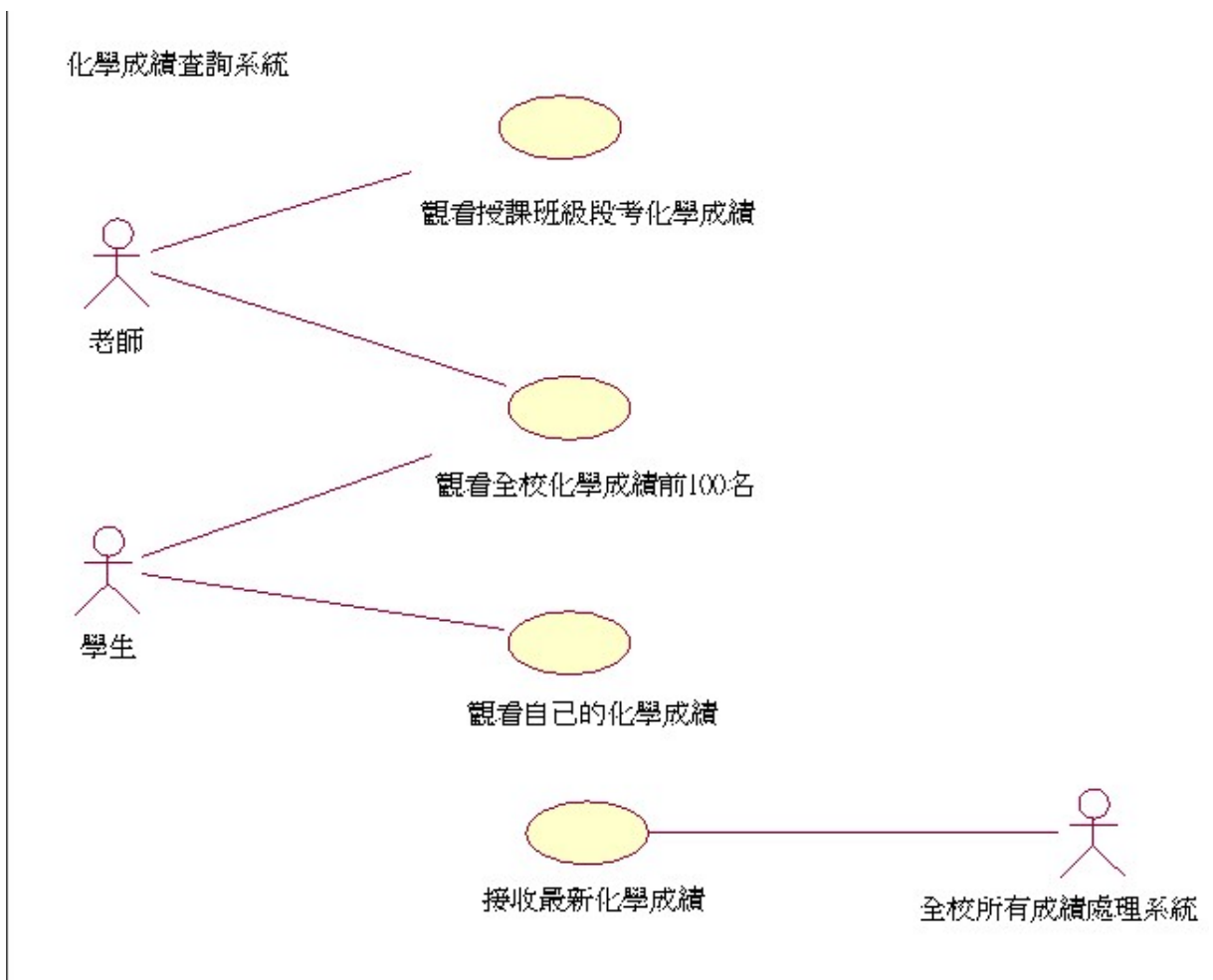
根據 Donald Bell 的解釋(2003):「一個使用案例可用來說明系統所提供的一項功能，使用案例圖主要的目的在幫助開發團隊設想系統功能的需求，其中包含了參與者(actor，也就是跟系統互動的人)與基本處理程序的關係，還有不同使用案例之間的關係。使用案例圖通常會表示出一組使用案例：不是整個系統所有的使用案例，就是與使用案例相關的一組特殊的功能(例如所有和安全管理相關的使用案例)。你必須在圖的中間畫一個橢圓，並在橢圓的中間或是下面填入這個使用案例的名稱，用來表示使用案例圖中的一個使用案例。另外你要在圖的左邊或是右邊畫一個棒狀的小



人（為了避免你會納悶，事實上有一些人會畫比棒狀小人好看的人形圖），用來表示使用案例圖的一個參與者（包括系統使用者）。還要利用簡單的線條來描述參與者和使用案例之間的關係。」簡單的說就是要畫出

系統要設計給那些使用者以及那些相關的系統，他們之間如何互動。

例如底下的一個使用案例圖範例：

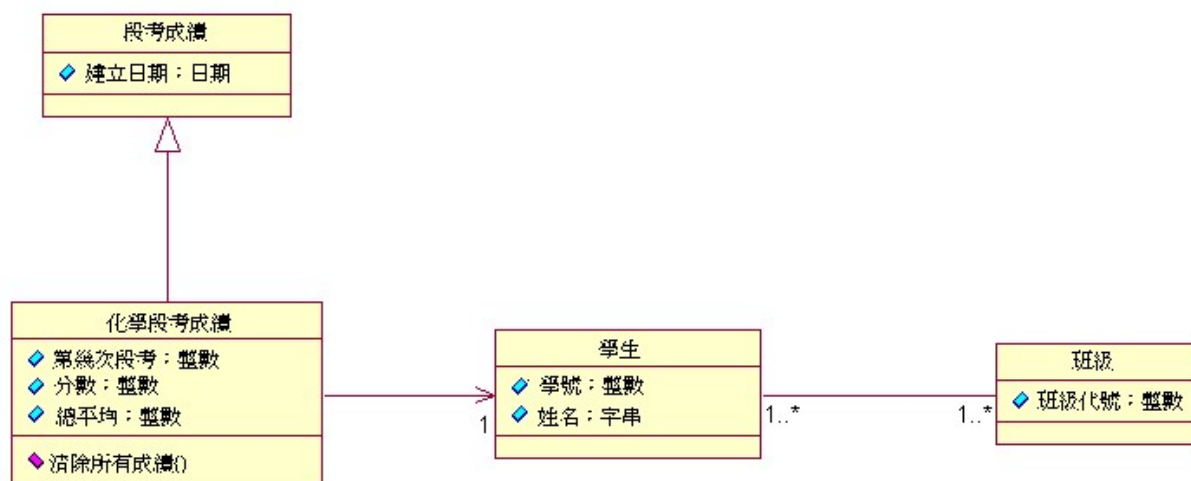


圖十 使用案例圖範例

假如研究者想做一個化學成績查詢系統，研究者將與此系統有互動的使用者與其他相關系統畫成上圖的使用案例圖，此圖顯示老師可以從此系統觀看授課班級的化學成績，也可觀看全校前100名的化學成績。而學生可以從此系統觀看全校前100名的化學成績，也可觀看自己的化學成績。另外，這個查詢系統還可從全校成績處理系統處接收到最新化學成績。若在設計系統前先畫出使用案例圖將有助於釐清系統將提供的功能，與系統將服務或互動的對象。

## 2-2-4-6、類別圖 (Class diagram)

類別圖描述不同的個體（人、事物和資料）相互的關係；換句話說，它表示了系統的靜態結構。以上述成績查詢系統為例，可畫出部分的類別圖如下圖：



圖十一 類別圖範例

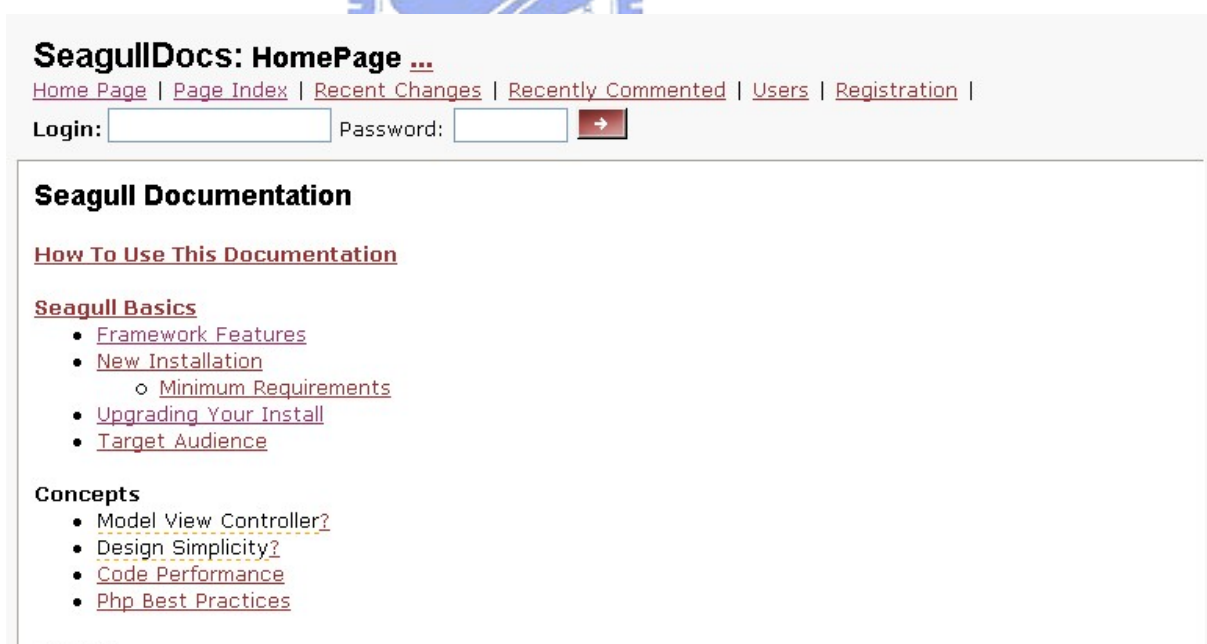
每一個矩形代表一個類別，每個類別都分成三層，最上一層代表類別名稱，中間層代表類別的「資料」或稱作「成員」，冒號後面代表資料的格式，最下一層代表類別的「方法」。上圖化學段考成績類別乃繼承自段考成績類別，化學段考成績類別會聯繫到學生類別，而班級與學生類別會互相聯繫。

我們可以透過將 Seagull Framework 的原始碼內容來畫成類別圖，將會更清楚看出架構的運作模式，也更能幫助研究者了解 PHP 網站物件導向的程式設計概念並將其模式加以學習。

## 2-2-5、Seagull Framework 使用說明手冊

Seagull 官方網站還提供了一份以 Wiki 架構而成的 Seagull 使用說明文件與教學說明文件網站，內容包括架構說明，如何擴充模組，如何升級，如何與設計者連絡，如何修改模板，如何使用或更改資料庫…等，裡面的文章都是許多人的集體創作，也是研究 Seagull Framework 的絕佳資料，更妙的是這些文章純粹是程式開發者與使用者將研究心得寫出，並沒有透過計畫與開會，大家自由創作。

簡單補充說明，Wiki 是一種共同創作(collaborative)環境的網站，只要去 wiki 的官方網站就可下載套裝程式，安裝也很簡單。所謂的共同創作是指每個人都可以任意修改網站上的頁面資料，並用一些很直覺的語法來建立文章的格式例如：只要在 wiki 中寫入 [[<http://www.wakkawiki.com>]] 後按 Submit，即可出現超連結如：<http://www.wakkawiki.com>。wiki 的語法尚有許多格式，可參閱 wiki 的官方網站。Seagull 專案藉由這種方式讓所有使用者共同創作參與撰寫各式各樣的使用說明與教學文件。如下圖：



圖十二 Seagull Framework 的 wiki 文件區