

逢甲大學

資訊工程系專題報告

以.net 技術實作

逢甲大學.net 討論社群網站

學生：謝清光(四丙)

指導教授：劉安之教授

中華民國九十三年五月

目錄

圖表目錄.....	4
摘要.....	7
第一章－導論.....	8
1-1 簡介.....	8
1-2 問題說明.....	8
1-3 網站規劃.....	10
1-4 網站內容分析.....	12
第二章－.net 技術的研究與討論.....	15
2-1 ASP.NET 簡介.....	15
2-2 .NET 平台資料庫存取技術介紹.....	18
第三章－網站系統規劃.....	25
3-1 第一版的產生.....	25
3-2 Hurricane 與 Asp.net Forum 簡介.....	26
3-3.1 Hurricane 入口網站介紹與分析.....	27
3-3.2 Asp.net Forum 介紹與分析.....	31
3-3 系統整合實作.....	34
3-3.1 資料庫整合實作.....	34
3-3.2 使用者及安全性整合實作.....	35
3-3.3 網站架構整合實作.....	36

第四章－與搜尋引擎的整合與系統展示.....	45
4-1 與個人化搜尋引擎的整合.....	45
4-2 系統展示.....	47
第五章－結論.....	59
5-1 總結.....	59
5-2 遭遇到的困難.....	59
5-3 未來展望與期許.....	59
5-4 心得.....	60
參考文獻.....	62



圖表目錄

圖 1.1 使用者與網站互動圖.....	10
圖 2.1 利用 header 設定連結程式碼.....	16
圖 2.2 Asp.net 運作流程圖.....	17
圖 2.3 .net 平台資料存取架構圖.....	19
圖 2.4 資料、DataSet 與控制項繫結示意圖.....	21
圖 2.5 DataAdatapter 功能示意圖.....	22
圖 2.6 Asp.net 執行預存程序與資料繫結流程圖.....	23
表 3.1 系統開發環境與工具.....	25
圖 3.1 網站初版首頁.....	26
圖 3.2 網站初版討論區.....	26
圖 3.3 網站初版討論區內容.....	27
圖 3.4 Hurricane 網站首頁.....	28
圖 3.5 Hurricane 的資料表[1](Blog 部分).....	29
圖 3.6 Hurricane 的資料表[2](使用者部分).....	30
圖 3.7 Hurricane 的資料表[3](新聞，文章，資源下載部分).....	31
圖 3.8 Asp.net Forum 論壇系統.....	32
圖 3.9 Asp.net Forum 資料表與關聯性.....	33

圖 3.10	整合後的 Hurricane 資料庫.....	34
圖 3.11	FormsAuthentication 表單驗證的運作流程.....	35
圖 3.12	使用者角色整合示意圖.....	36
圖 3.13	整合後的討論區外觀.....	39
圖 3.14	建立全文檢索目錄.....	40
圖 3.15	文章搜尋畫面.....	43
圖 3.16	討論區搜尋功能.....	44
圖 4.1	加入專案.....	45
圖 4.2	將專案引入參考.....	45
圖 4.3	專案相依性與建置順序.....	46
圖 4.4	輸入 Asp.net 關鍵字得到的查詢結果.....	47
圖 4.5	網站首頁.....	47
圖 4.6	新聞分類頁面.....	48
圖 4.7	發送新聞頁面.....	49
圖 4.8	日期選擇以及分類選擇.....	49
圖 4.9	管理新聞頁面.....	50
圖 4.10	修改新聞頁面.....	50
圖 4.11	新聞核准頁面.....	51
圖 4.12	個人日誌首頁.....	52
圖 4.13	閱讀及回應頁面.....	53

圖 4.14 日誌管理功能.....	53
圖 4.15 討論區首頁.....	54
圖 4.16 討論區內標題頁面.....	54
圖 4.17 討論文章頁面.....	55
圖 4.18 建立新討論文章.....	55
圖 4.19 討論區搜尋頁面.....	56
圖 4.20 討論區管理介面.....	56
圖 4.21 資源下載首頁.....	57
圖 4.22 傳送資源頁面.....	57
圖 4.23 下載頁面.....	58



摘要

逢甲大學.net 技術研究群成立於 2003 年 3 月。透過本校資訊處與台灣微軟的協助，和在全台各大專院校的微軟菁英俱樂部的同學們一起互相學習。討論社群網站的成立，除了提供研究群一個討論的空間，也透過資料的吸收整理，文章的撰寫，成爲一個可以學習.net 技術的知識庫，讓大家的學習更有系統、更寬廣。

以下便對報告的各章節做扼要的說明：

第一章：導論，包含簡介，網站分析，網站系統規劃，資料分類與整理的介紹。

第二章：介紹系統實作相關知識.net 平台的研究探討。

第三章：系統實作，將 Hurricane 系統與 Asp.net Forum 的整合與中文化的介紹。

第四章：與個人化搜尋引擎系統的整合與系統展示。

第五章：結論，包含遭遇到的困難，未來展望，以及心得。

第一章－導論

1-1 簡介

微軟學生菁英俱樂部逢甲支部於 2003 年 3 月在逢甲大學成立，正式名稱爲【逢甲大學 .Net 技術研究群】，有鑑於資訊科技的蓬勃發展，.Net 技術研究群致力於提供資訊產業最新之技術與訊息，我們的宗旨如下：

- 促進國內軟體研究及開發人才培育。
- 提供逢甲大學師生前瞻性軟體資訊學習與應用環境。
- 鼓勵逢甲大學學生研究開發第三代網際網路技術之電腦軟體應用。
- 提升逢甲大學學生專題製作之品質。
- 整合校內軟體發展技術，成立知識管理中心。
- 其組織隸屬於逢甲大學資訊處，由趙啓時老師擔任研究群指導老師。

研究群辦公室設在逢甲人言大樓三樓，並設有獨立之圖書室供會員閱讀及討論使用。關於研究群網站與知識管理中心的建置，利用.net 平台技術設計一個網站，讓研究群同學互相討論.net 技術的空間，以及有系統的吸吸知識，成爲我這次專題的題目。

1-2 問題說明

社群網站的最大用途就是將有共同目的、共同興趣的使用者連結在一起。最重要的是分析網站使用者爲什麼會來到這個網站，他們會需要哪些功能或內容，

還有他們在網站裡面活動的情形是如何的。關於這些問題，我訂定了一些主題來討論：

1. 使用者：

定義使用者是網站建構時期非常重要的事情，要知道使用者進入網站會做些什麼活動，必須知道有哪些人會進入這個網站。依照上列問題，將使用者做分類：

- 想學習.net 技術，希望在網站內尋找相關資料。
- 尋找某種功能的設計方式或是 sample code，以完成自己想建立的系統。
- 尋找免費下載的元件，縮短開發時間。
- 有問題需要請別人幫忙解決。
- 研究群成員
- 網站管理人員

2. 使用者的活動：

在列舉出哪些使用者會利用網站後，接下來要分析他們在網站可能會做些什麼事。在這裡我參考了www.asp.net與www.codeproject.com的運作方式，自己實際當個使用者，看看自己在學習的時候用到了哪些功能，自己參與了哪些活動。使用者與管理人員透過討論社群的互動方式，可以由下圖表示：

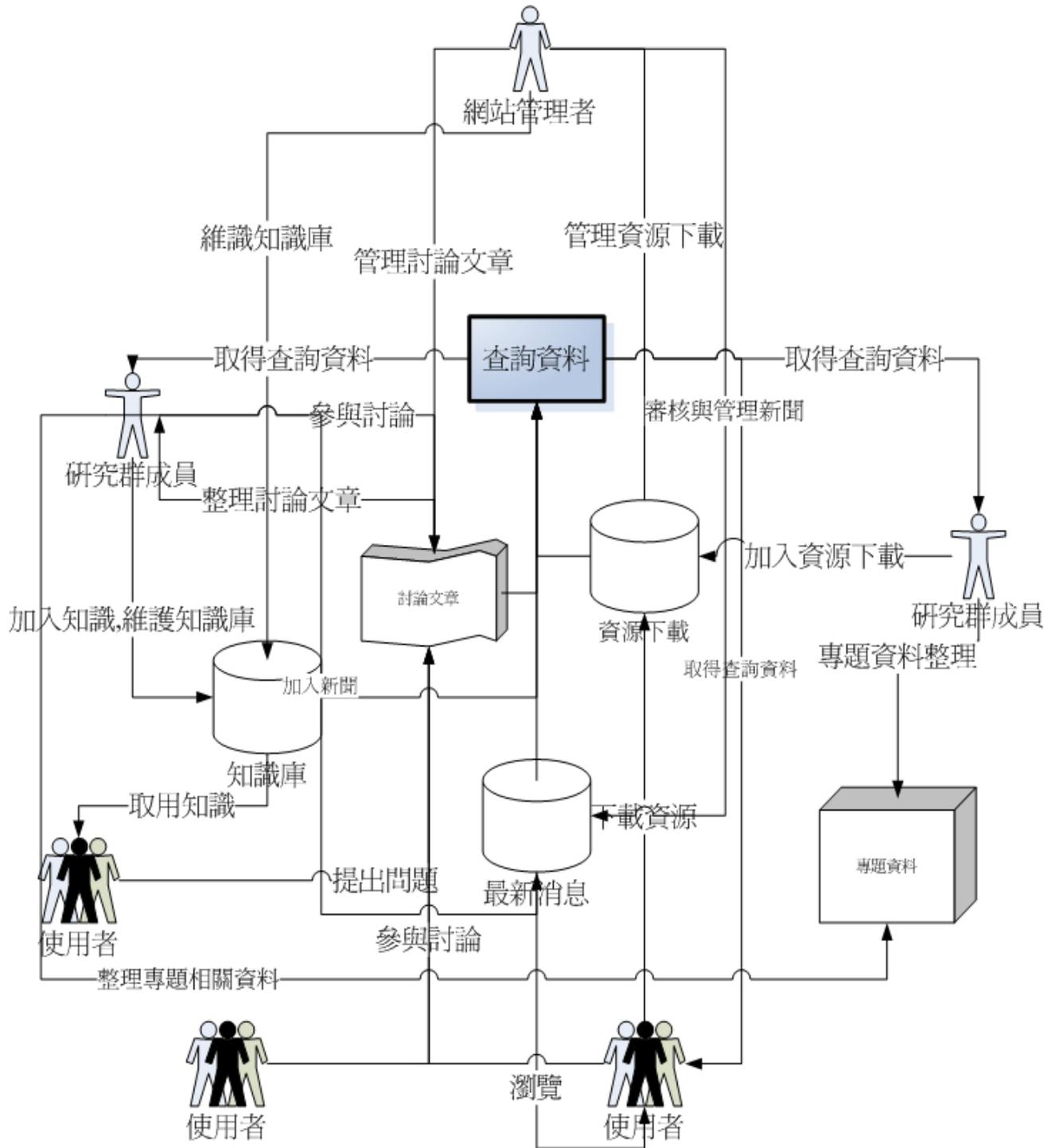


圖 1.1 使用者與網站互動圖

1-3 網站規劃

在確定題目後，便開始參考國內外其他討論有關.net 技術的社群網站，從他們的經營方式來思考研究群的網站該怎麼架構、怎麼與使用者達到互動的功能、

如何促進研究群同學學習知識、提供專題實作技術參考資料，使大家可以順利的完成計劃。以下是我訂定的一些基本需求：

1. 新聞發佈功能：

新聞的發佈、審核、刪除、修改的功能，除了研究群的最新消息之外，還有相關於微軟學生俱樂部的新聞，微軟的新產品發表以及安全性更新通知。

2. 個人資料收集：

希望能讓研究群的同學讓自己可以將找尋到的資料。透過資料整理的功能，可以對自己的專題、想學習的新知識、以及使用的心得做整理，再把它寫到文章中成為知識庫的一部分。

3. 討論區功能：

這是討論社群的主要功能，除了基本針對研究群事務、.net 技術討論、網站的問題及建議做討論之外，對討論串的整理、討論版的管理功能也應該要有很好的介面提供管理者方便操作。

4. 發表文章及評分討論功能：

在整理相關技術資料後，或是在其他討論社群閱讀了不錯的技術文章，可以利用發表文章的功能把知識介紹給其他使用者。討論的功能除了可以讓使用者針對文章內容的問題與發表人互相討論，達到互動的效果，也希望可以透過評分的功能讓發表人能有個肯定。

5. 投票功能：

希望透過投票的功能，能達到使用者與網站的互動。

6. 資源下載：

在閱讀文章時，也許會有附加一個 sample code，或是有一些值得推薦的控制項、免費下載的系統，都可以把它轉回來供大家使用，縮短開發時間。

7. 使用者資料功能：

使用者可以在個人化的介面設定自己的資料。

8. 搜尋功能：

搜尋功能是建立知識管理的重要功能之一。使用者透過搜尋可以免去在一大堆資料中找尋自己想要的麻煩，利用關鍵字就可以取得需要的資料。

9. 管理介面：

良好的管理介面可以大大提升管理者的方便，除了上述功能的管理之外，也加入了網站友站連結管理、網站功能元件的管理等等，可以讓管理者方便地使用。

1-4 網站內容分析

在網站的基本功能需求分析出來後，接下來便要針對網站能提供哪些內容，將內容分類形成知識。有內容、有特色的網站才會讓使用者經常參觀、找尋自己想要的資料。以下對各個功能的內容分析：

- 研究群的內容：

研究群的運作，必然會伴隨著許多的知識，這些知識若被保存下來，可

以形成研究群的資產及特色。以下是有關研究群內容的整理：

- I. 研究群及微軟學生俱樂部最新消息及活動。
 - II. 會議記錄，可以針對開會時大家討論的結果做整理，例如對自己專題的建議、大家針對知識的討論、以及研究群運作的討論等等。
 - III. 專題成果，包括實作的系統、設計方式、實作技術的方法說明等等，可以提供其他同學或是使用者做參考。
 - IV. 個人資料整理，包含在其他網站或是書籍得到的一些參考資料、自己整理的專題想法等等，可以做為專題實作過程的記錄。
- .net 技術內容：

在.net 技術知識部分，我是參考國內外一些較具規模的社群網站，由他們的分類方式經過整理、篩選而得到網站可以得到的分類。這部分的內容也是取自國內外的網站經過整理後轉到研究群網站。以下是我針對.net 技術所做的分類：

1. Web Application / Web Service
2. Window Form Application
3. .Net Framework / Security
4. Mobile Application (SmartPhone & PDA,.....)
5. ADO.Net / SQL Server
6. Network Application
7. .Net Language / Object Oriented Program Design
8. Case Study

有關 Case Study 是個很有趣的部分。目前國內的網站尚未針對這項內容發展，不過國外的 Community 會針對自己網站的 Content，例如網站系統、討論區、控制項的設計方式與社群的會員互相討論以增進自己設計系統的能力。

力。寫程式前先從看別人的程式開始，了解好的設計方法，可以讓自己在接下來設計系統時避免很多問題的發生。

- 分享的 code Sample、使用者控制項、以及免費下載的 Web Application Source code，讓開發者可以免去很多自己寫程式的麻煩，因為網路上已經有你想要的這些元件，可以提升開發的速度。



第二章－.net 技術的研究與討論

2-1 ASP.NET簡介

微軟推行的.NET 平台，是利用網路為傳輸媒介，以 XML 為資料傳遞的格式，利用.NET Framework 所建構出的基礎，將所有可執行微軟平台的設備達到資訊互通的目標，這包括支援微軟平台的個人電腦、筆記型電腦、PDA、手機、甚至可能是家電或是其他設備。許多應用程式都以這種架構設計，最典型的 Web Application 也不免的成為微軟主推的一種應用程式架構。利用 Web Browser 在各種平台的共通性，以及以 XML 傳送資料的方式，將原本都是使用 Window Form 的應用程式，多了一種介面。Web Form 的好處，在於不需要安裝 Client 端程式，因為 Browser 就是 Client 端程式。更方便的解決需求，使得 Web Form Application 大受歡迎。Asp.net 的產生，就是為了Web Form Application 而設計的架構。

在 ASP.NET 裡加入了許多新的功能，使得設計方式更加近似於 Window Form：

1. Code Behind－

首先我們先看以下一段程式碼：

在 ASP.NET 中可以使用將程式碼寫在另一個 C#程式檔中，把 html 語法及程式碼分離的設計方式，稱為 Code Behind。在此例中，我們將 Default.aspx 的程式碼寫在 Default.cs。在開發網站程式時，經常是由網頁設計者與程式撰寫人員分工合作，為了獨立這兩大部分的工作，Code Behind 是一個很好的撰寫的設計。

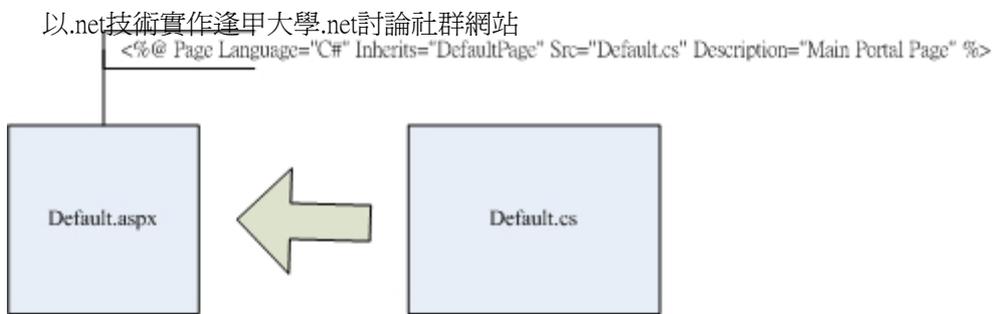


圖 2.1 利用 header 設定連結程式碼

2. 與 ADO.NET 的連結－

在經過了 ADO(Active Data Object)的經驗後，設計人員發現 RecordSet 的缺點。因為 RecordSet 需持續的與資料庫連線一筆一筆取得資料，持續連線造成的與資料庫頻寬不足，造成效能無法提升。在 ADO.NET，這不是 ADO 的改良版本，而是完全與 ADO 不同的資料庫連結設計方式。利用離線存取策略，ADO.NET 將資料存入 DataSet——一個暫時的離線資料庫，我們可以對 DataSet 的資料進行新增刪除修改查詢的動作，而 DataSet 本身就是一個 XML 文件，使用非常方便。當發生資料被修改過後，再與資料庫連結。這種離線存取的方式，大大的降低了資料庫的負擔，提升了效能。

3. 更豐富的設定及使用者控制項－

你可以自訂一個使用者控制項，再利用如下列的方式：

```
<asp:Label id="spanAdd" Text="add" runat="server"/>
```

這是 ASP.NET 內建的 label 控制項，你可以在網頁上直接把這個控制項加入，只要填入需要的參數便可以使控制項啟動。

此外，你可以自己建立自己的命名空間以及控制項，例如

```
<IECS:login id="StudentLogin" runat="server"/>
```

這可以表示為一個在 IECS 命名空間中的一個登入控制項。

4. 編譯而非直譯－

在 ASP 的時代，由於是使用直譯的方式執行，常常被批評效能不佳，這次.NET 平台除了改變以往的編譯模式，使用 IL(Intermediate Language)及 JIT(Just In Time)來增加對其他平台的支援，也將 ASP.NET 同樣的做了變更，第一次的執行較慢，但是可以將之後存取的效能提升。

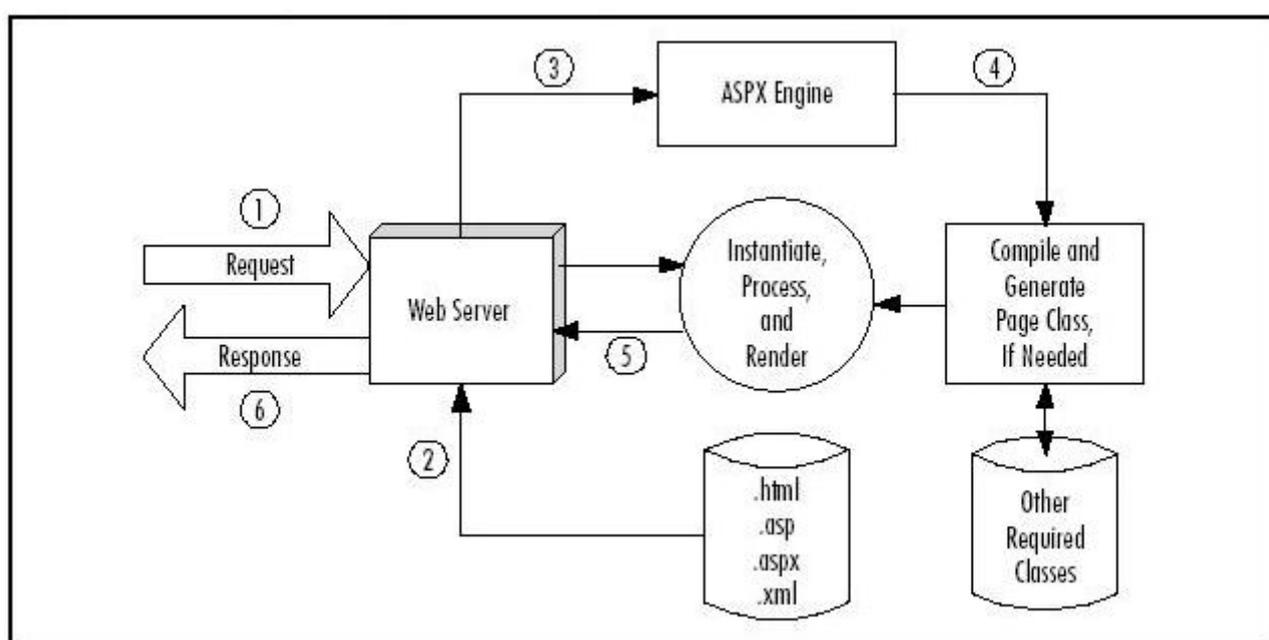


圖 2.2 Asp.net 運作流程圖

5. 多語言支援－

.NET 平台目前所知共可支援常用的程式語言共 25 種之多，包括 C++、JAVA(J#)、C#、VB、COBOL、SMALLTALK 等等，目前 ASP.NET1.0 已支援 C#及 VB.NET，在 1.1 版更會加入 J#為開發 web form 程式的基礎語言。此外，在同一個程式中，可以使用多種語言來共同開發程式，真正達到”跨語言”的目標。

6. 快取的功能－

在 ASP.NET 可以將經常使用的部分，如資料集、網頁程式碼、控制項等，設置成爲快取，常駐在記憶體內，而且若經過變更可以重新更新快取，平衡資料更新程度與網站速度。

7. 更容易的部署－

在 ASP.NET 中，與 IIS Server 的更緊密結合，開發完成的程式碼，可以透過 IIS 設置成爲虛擬目錄後即可使用，或者是更可以利用 Visual Studio.NET 封裝精靈把程式包裝成爲一個安裝檔，即可快速的完成部署網站。

8. 安全性－

.NET 提供了三種驗證模式：Windows 驗證，Form 表單驗證，以及 passport 驗證。

Windows 驗證適合在企業內部使用，因爲使用 Windows 驗證需要在 Windows Server 內存有自己的帳號。而 passport 驗證目前需要向微軟註冊且需要繳費，所以目前不太常被使用。在網站上最被常使用的就是表單驗證。在.NET 裡也有在表單驗證裡自設加密的資料，如 MD5 加密，可以設在你的登入密碼等等較隱密的資料內，增加安全性。

2-2 .NET平台上資料庫存取技術介紹

在.NET 平台說到資料存取，不得不提到 ADO(ActiveX Data Object).NET，它是.NET 各平台存取資料的一個橋樑。我們先由以下的圖來看看 ADO.NET 的架構：

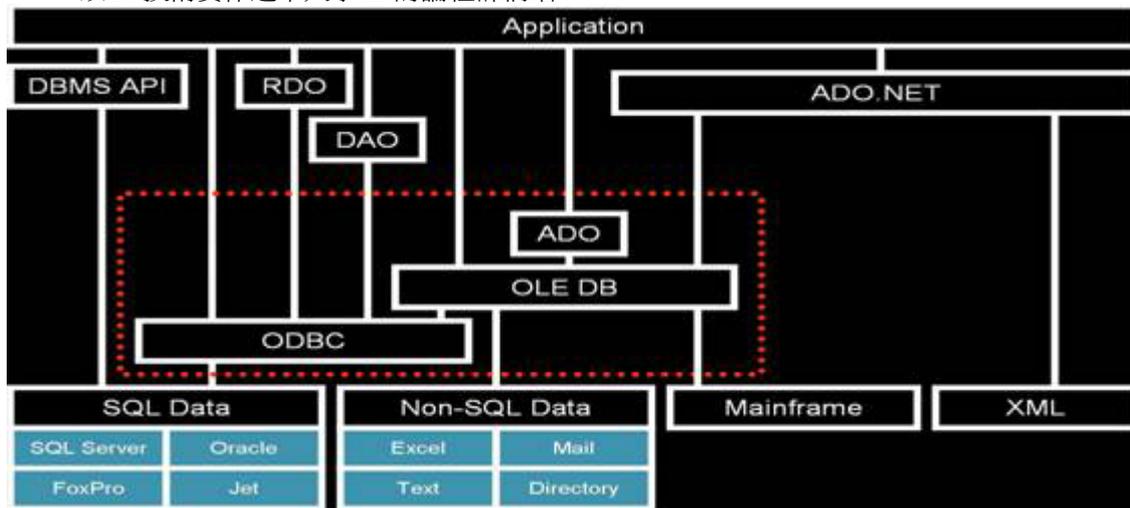


圖 2.3 .net 平台資料存取架構圖

可以看出，ADO.NET 是建立在 ODBC、OLE DB 之上的一個資料存取模式，不僅保有原本 ADO 可存取的資料，更加入 XML，使得 .NET 平台的資料存取的種類更加完整。不管是 Windows Form 或是 Web Form 只要呼叫 ADO.NET 物件即可取得需要的資料。

ADO.NET 常使用到的 Namespace(命名空間)有：

- System.Data
- System.SqlClient
- System.OleDb
- System.odbc
- System.Xml
- System.OracleClient(在 ASP.NET 1.1 加入的新命名空間)

當然,SqlClient 與 OracleClient 是 ADO.NET 針對這兩個 DataBase 的資料存取做特別最佳化，其他的就只能使用 OleDb 去存取資料庫。而在 ADO.NET 中延續 ADO 所使用的物件有以下幾個：

- Connection 物件-用來開啓,關閉對資料庫的連線
- Command 物件-用來下 SQL 指令到資料庫，傳回資料的物件

關於 ADO.NET 有以下三個重要的物件：

- DataSet - 針對一般資料、XML 資料和關聯式資料進行儲存、遠端處理和程式設計。
- DataReader - 針對從 SQL Server 資料來源讀取順向資料流的資料錄。
- DataAdapter - 針對資料庫將資料送回 DataSet，以及整理資料。

接下來我們就一個一個介紹有關這些物件：

1. DataSet-離線資料庫

DataSet 其實就是在 Web Server 所維護的一些 XML 的資料檔,從資料庫中取出資料以 XML 文件的方式傳入 Web Server,再由 Web Server 做管理的一個資料集。 DataSet 物件表示資料的快取，具有類似資料庫的結構，例如資料表、資料行、關聯性和條件約束。

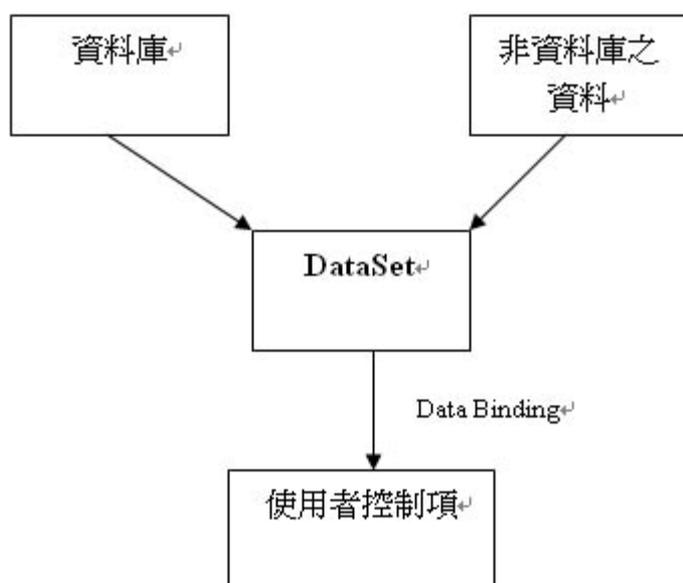


圖 2.4 資料、DataSet 與控制項繫結示意圖

雖然 DataSet 的行為非常像資料庫，但是重要的是，DataSet 物件不會直接與資料庫或其他來源資料互動。這樣開發人員才能使用永遠一致的程式撰寫模型，而不管來源資料所在的位置。來自資料庫、XML 檔案和程式碼的資料，或使用輸入項目都可以放置在 DataSet 物件中。所以，當 DataSet 做了變更時，它們可以在更新資料來源之前被追蹤或驗證。在一個 DataSet 裡可以包含許多的資料，包括資料表，資料間的關係等等的資訊：

- Tables 集合—存放傳回的資料表，包含了 DataRow 集合與 DataColumn 集合。
- Relations 集合—將 DataTable 之間的關聯存放在此集合內。
- ExtendedProperties 集合—用來存放使用者自訂的資訊，如密碼，或建立時間等。

2. DataAdapter

DataAdapter 物件可以當成 DataSet 和資料來源之間的橋樑。

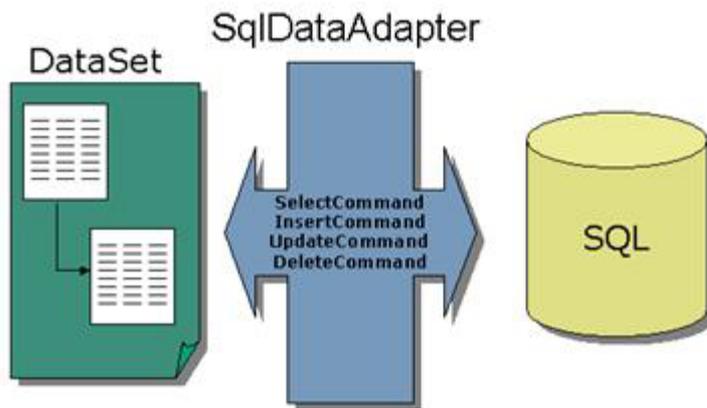


圖 2.5 DataAdapter 功能示意圖

在對 DataSet 做了變更之後，DataAdapter 物件會使用命令更新資料來源。使用 DataAdapter 的 Fill 方法呼叫 SELECT 命令；使用 Update 方法對每一個已變更的資料列呼叫 INSERT、UPDATE 或 DELETE 命令。為了控制 Runtime 所用的陳述式，您可以明確設定這些命令來解析變更，包含預存程序的使用。針對特定的案例，CommandBuilder 物件可以根據 Select 陳述式在 Run-Time 產生這些命令。不過，這個 Run-Time 產生需要額外的伺服器往返來收集所需的中繼資料 (Metadata)，因此在設計階段明確提供 INSERT、UPDATE 和 DELETE 命令，將會產生較佳的 Run-Time 效能。而 DataAdapter 在宣告的方式也非常的便利，如下所示：

宣告方式(C#)

```
SqlConnection myConnection= new  
SqlConnection("server=(local)\VSDotNET;Trusted_Connection=yes;database=northwind");
```

首先先建立 connection，連線字串是存取 local 端的 VSDotNET 的 Server 資料庫是北風資料庫。

```
SqlDataAdapter mySqlDataAdapter = new SqlDataAdapter("select * from customers", myConnection);
```

建立 DataAdapter，把要使用 SQL 命令及使用的連線加入即可回傳資料。

接下來只要使用 DataSet 的 Fill 方法把 DataAdapter 傳回的資料接收即可。

使用 StoreProcedure：

預存程序是一種很常用的方式，可以讓程式設計者與資料庫管理者分工合作，而且在後端執行比建立連線後下多次的 SQL 命令更來得有效率。

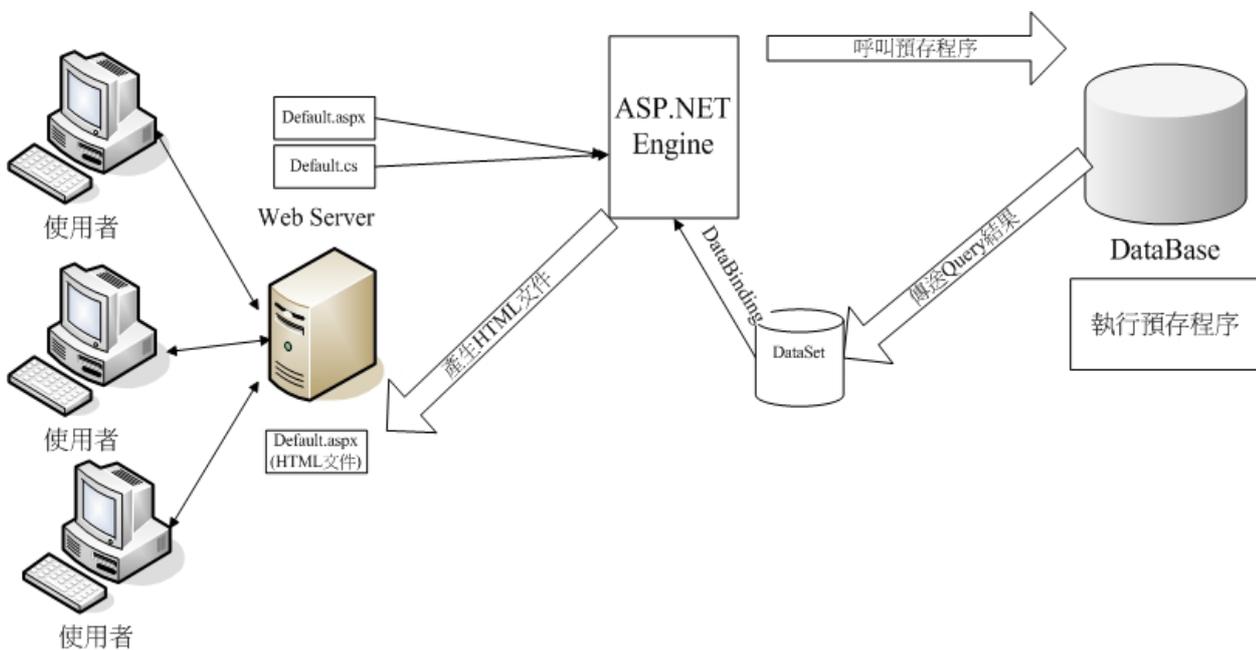


圖 2.6 Asp.net 執行預存程序與資料繫結流程圖

我們來看 ASP.NET 是如何寫的。

```
mySqlDataAdapter.InsertCommand.CommandText = "sp_InsertCustomer";
```

加入我們要使用的 StoreProcedure 名

```
mySqlDataAdapter.InsertCommand.CommandType = CommandType.StoredProcedure;
```

把 Adapter 的 CommandType 改成執行資料庫端的 StoreProcedure 而非 Text。

```
mySqlDataAdapter.DeleteCommand.CommandText = "sp_DeleteCustomer";
```

```
mySqlDataAdapter.DeleteCommand.CommandType = CommandType.StoredProcedure;
```

```
mySqlDataAdapter.UpdateCommand.CommandText = "sp_UpdateCustomers";
```

DataAdapter 物件共有三種指令，Insert，Delete，Update，只要依序將所需要的動作將其填入 DataAdapter 內，當發生新增刪除修改的事件時，DataAdapter 會找尋其相對應的方法去執行。

3. DataReader

DataReader 的運作方式是利用將資料傳回 DataSet，再從 DataSet 內一筆一筆的顯示資料。這種方式不用一次又一次的和資料庫連線，也可以一次只取得一筆資料並顯示在表單中。適合在只回傳資料表中的其中一筆記錄時，或當需要顯示唯讀資料時，如在 Web 網頁展示一些資料時可使用。DataSet 並不能直接顯示在網頁或是表單程式中，所以需要加入控制項以達成資料繫結

(DataBinding)的目的，共分為兩種類型—

- I. 簡單的資料繫結：將控制項繫結到一個單一的資料元素，如 DataSet 中某資料表某欄位的值，連結到 Label 或是 TextBox 上。

例如我們將 DataReader(reader)中的姓名部分(Name)在第 0 個位置，和

Label 控制項(lblName)繫結：

```
lblName.Text = reader.GetString(0);
```

- II. 複雜的資料繫結：將控制項繫結到多個資料元素，如 DataGrid，DataList 等，需要多個欄位或是多個資料表單互相配合的一種繫結方式，我們也只要使用 DataBinding 的方法把資料由 DataSet 傳送至控制項，就會把資料依照我們需要的方式顯現出來了。

```
DataList.DataSource = DataSet.DefaultView
```

第三章 系統實作

3-1 第一版的產生

在規劃功能之後，便立即開始實作網站的第一版。以下是在實作的開發環境與工具：

系統開發平台	.Net Framework 1.1 Asp.net 1.1
主機	CPU:AMD Athlon XP 2400+ Memory:512MB Hard Disk:40G
Web 伺服器	Microsoft Internet Information System (IIS) 6.0
資料庫伺服器	SQL Server 2000 Sp3
主機作業系統	Microsoft Windows Server 2003
系統開發工具	Microsoft Visual Studio.net 2003

表 3.1 系統開發環境與工具

第一版的功能已具基本功能，包含新聞、討論區、投票、資源下載等系統，大部分都是自己看書上的程式碼及現有的資料表拼拼湊湊出來的，基本功能都有出現但是卻不夠完備，尤其是在討論區方面，開討論區、文章的管理、使用者權限的設定、文章搜尋的功能，這些功能如果要自己做，實在是個很大的工程。不過在這一段持續嚐試錯誤的階段，除了熟悉工具，確實了解 asp.net 元件間的運作方式外，也開始尋找現有的系統，希望第二版的網站可以透過整合其他既有的系統來完成。之後找到了 Hurricane 入口網站、以及 Asp.net Forum 論壇系統。

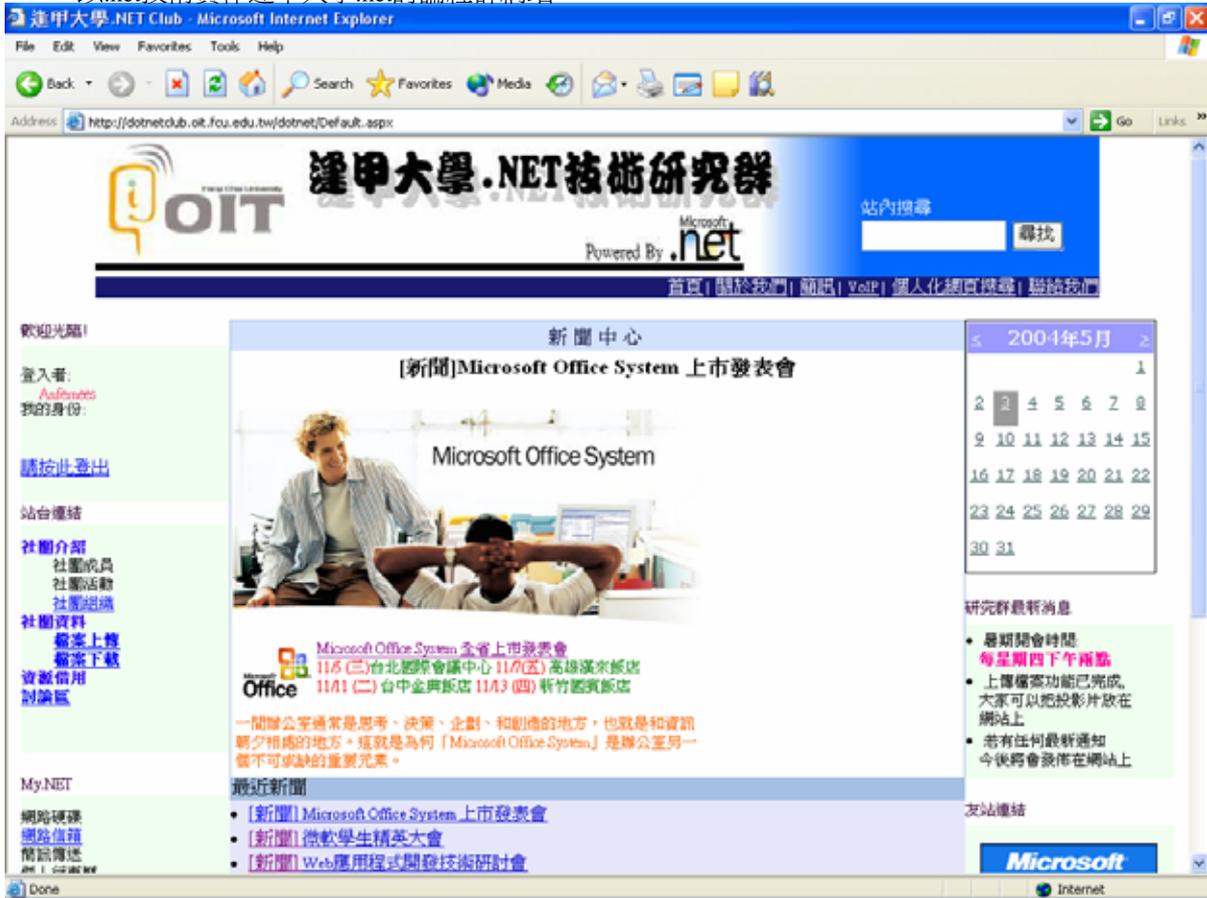


圖 3.1 網站初版首頁



圖 3.2 網站初版討論區



圖 3.3 網站初版討論區內容

3-2 Hurricane與Asp.net Forum 簡介

3-2.1 Hurricane入口網站介紹與分析

Hurricane 入口網站(<http://www.projecthurricane.net>)是一個以學生社群為主題所設計的網站。以這個網站為架構也是因為這個網站具有大部分我所需要的功能。

網站在 1.0 版的功能有：

- 新聞發佈與管理功能。
- 活動發佈與管理功能。
- Blog（網路日誌）。
- 討論區的功能。
- 專欄文章發佈、評分、討論功能。

以.net技術實作逢甲大學.net討論社群網站

- 資源下載、討論功能。
- 投票功能。
- 其他站台管理功能。

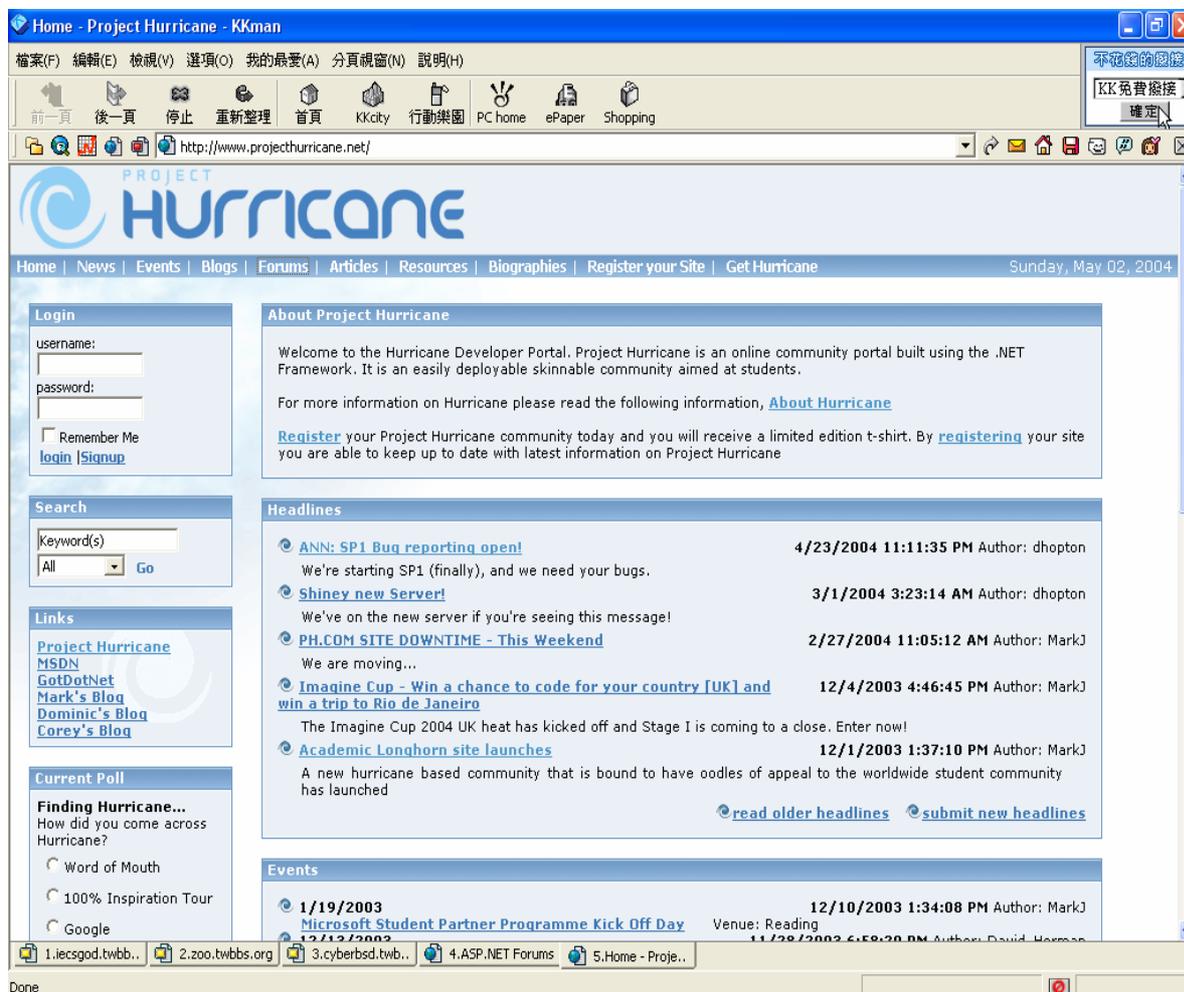


圖 3.4 Hurricane 網站首頁

經過安裝以及實際操作後，我整理了資料表與網站類別的架構，以下是使用 Reverse Engineering 的方式把資料庫的資料表，索引與關聯性顯示的結果：

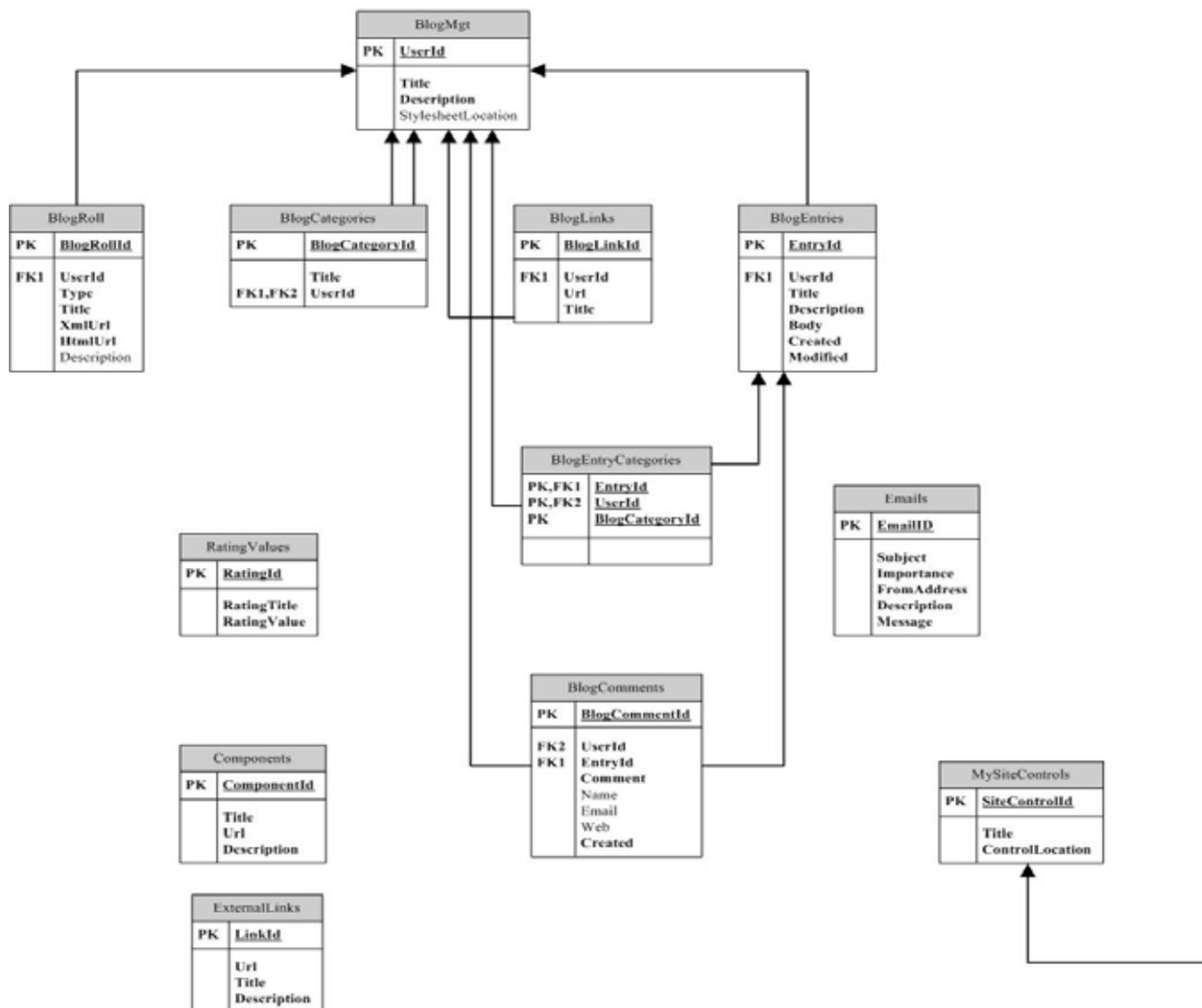


圖 3.5 Hurricane 的資料表[1](Blog 部分)

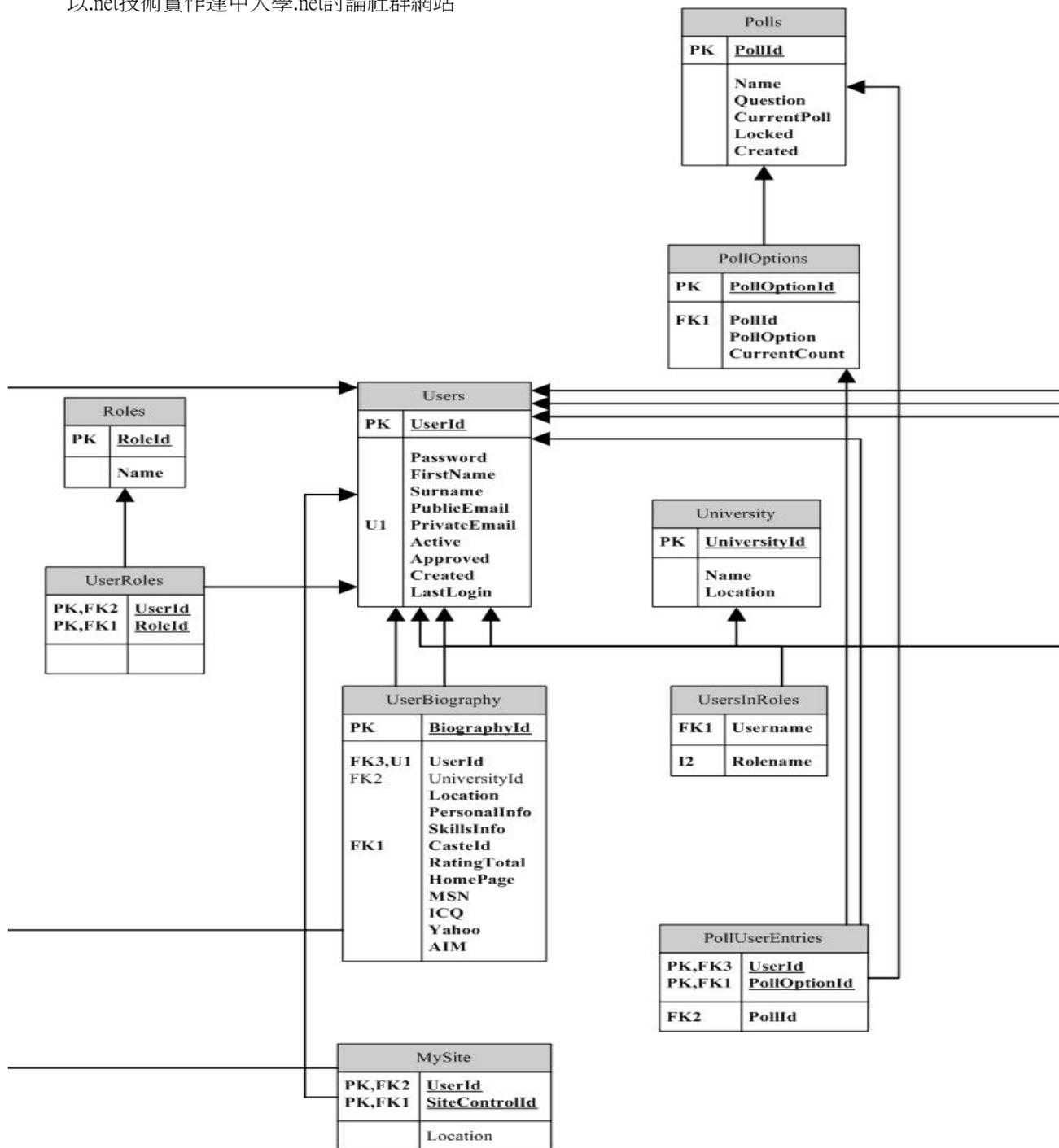


圖 3.6 Hurricane 的資料表[2](使用者部分)

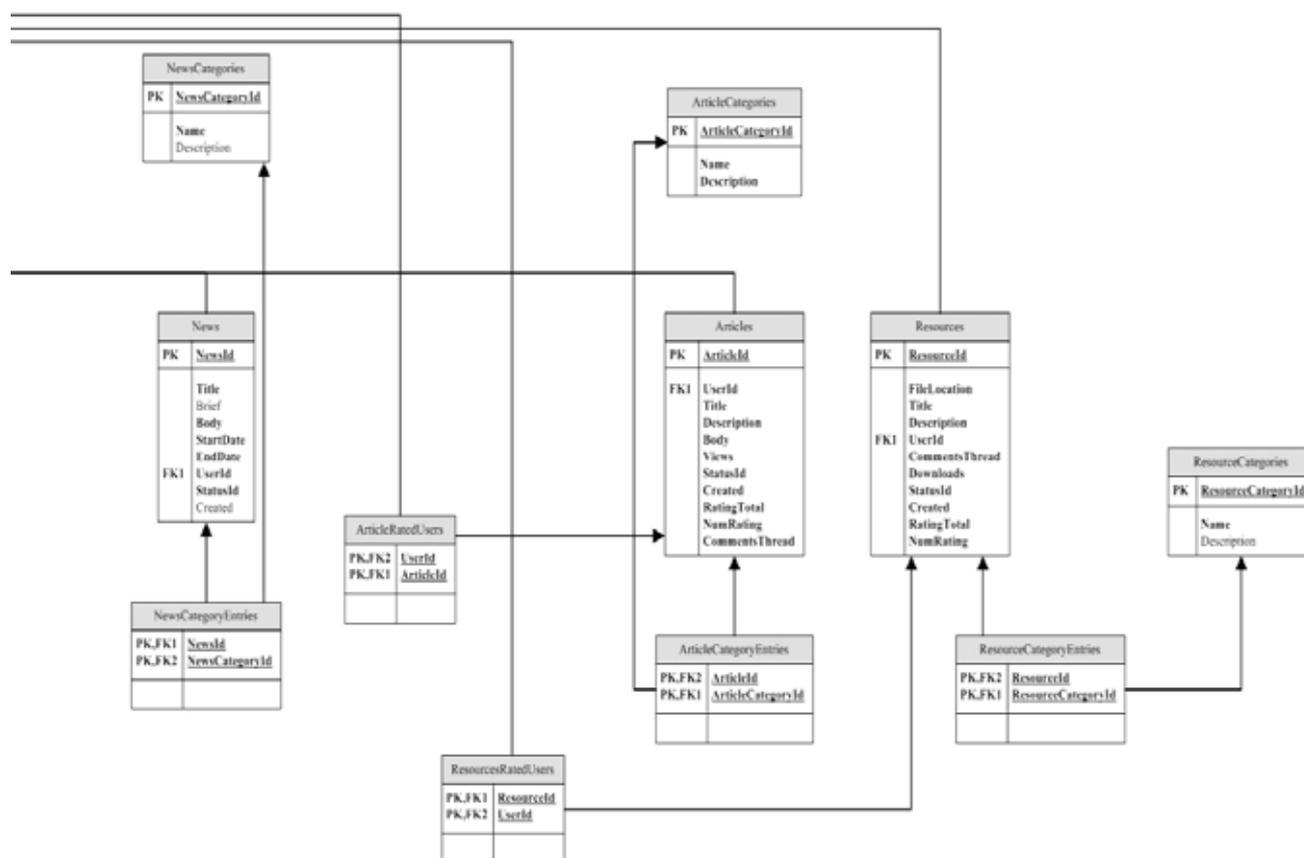


圖 3.7 Hurricane 的資料表[3](新聞，文章，資源下載部分)

3-2.2 Asp.net Forum介紹與分析

Asp.net Forum 論壇系統是由微軟線上社群 www.asp.net 的 Asp.Net Forum Team 所設計的。以許多外面的 PHP 論壇系統的功能為基礎，發展.net 平台的論壇程式。現在的版本是 1.2 版，而 2.0 版(<http://forums.asp.net/Forums/>)也已經在測試中，加入了表情、MSN Messenger 是否登入等等更棒的功能。除了多樣化的功能外，在管理上 Asp.net Forum 也提供了許多方便的介面，讓管

以.net技術實作逢甲大學.net討論社群網站

理者可以指派版主、自定管理層級、自定使用者角色以進入需經授權的討論版等，可以說是功能非常完備。而在台灣也已經有人將這個系統中文化，也省去了需要中文化的工夫，現在只要分析如何整合即可。

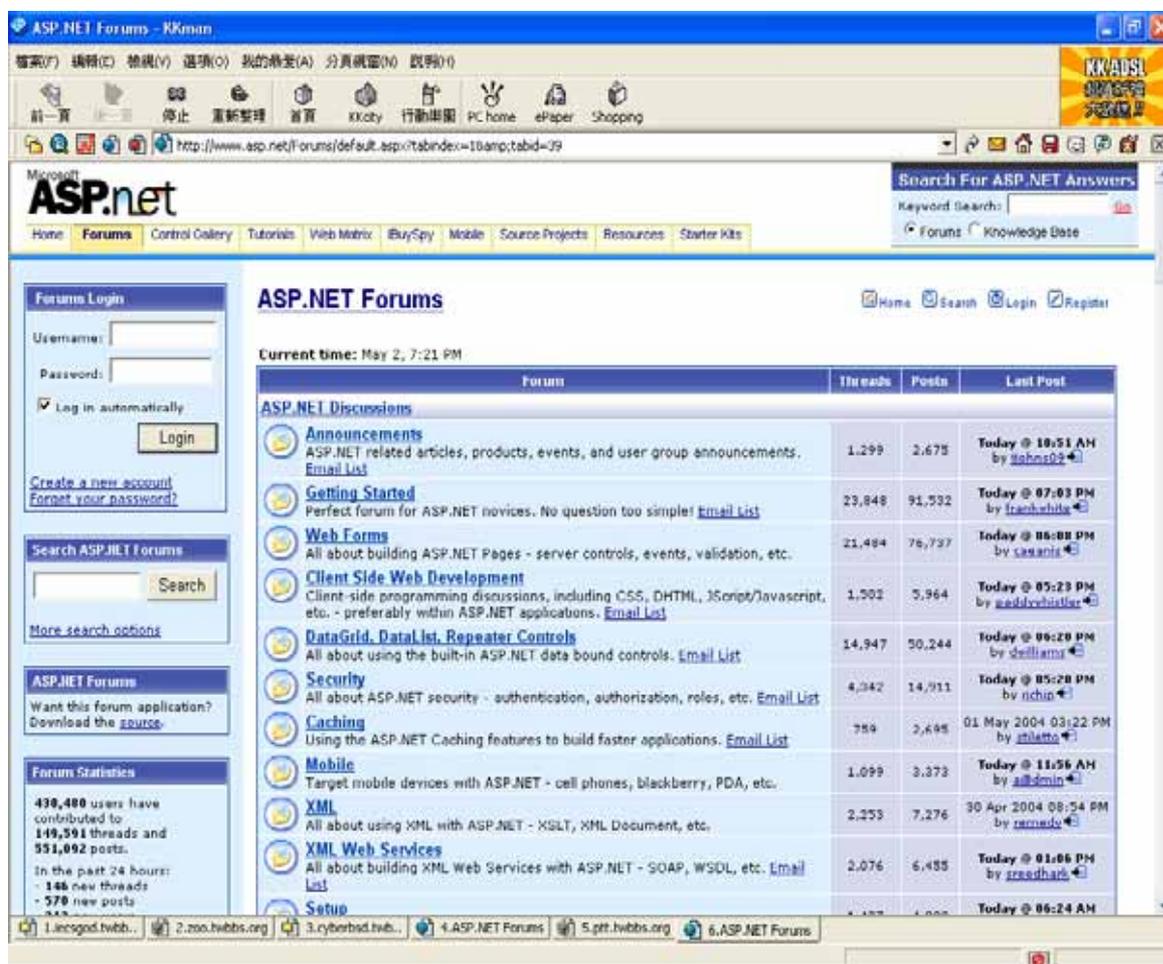


圖 3.8 Asp.net Forum 論壇系統

我也使用 Reverse Engineering 的方式把資料庫關聯性顯現出來，做為系統整合方法的重要資料。以下是 AspNetForum 的資料表及關聯：

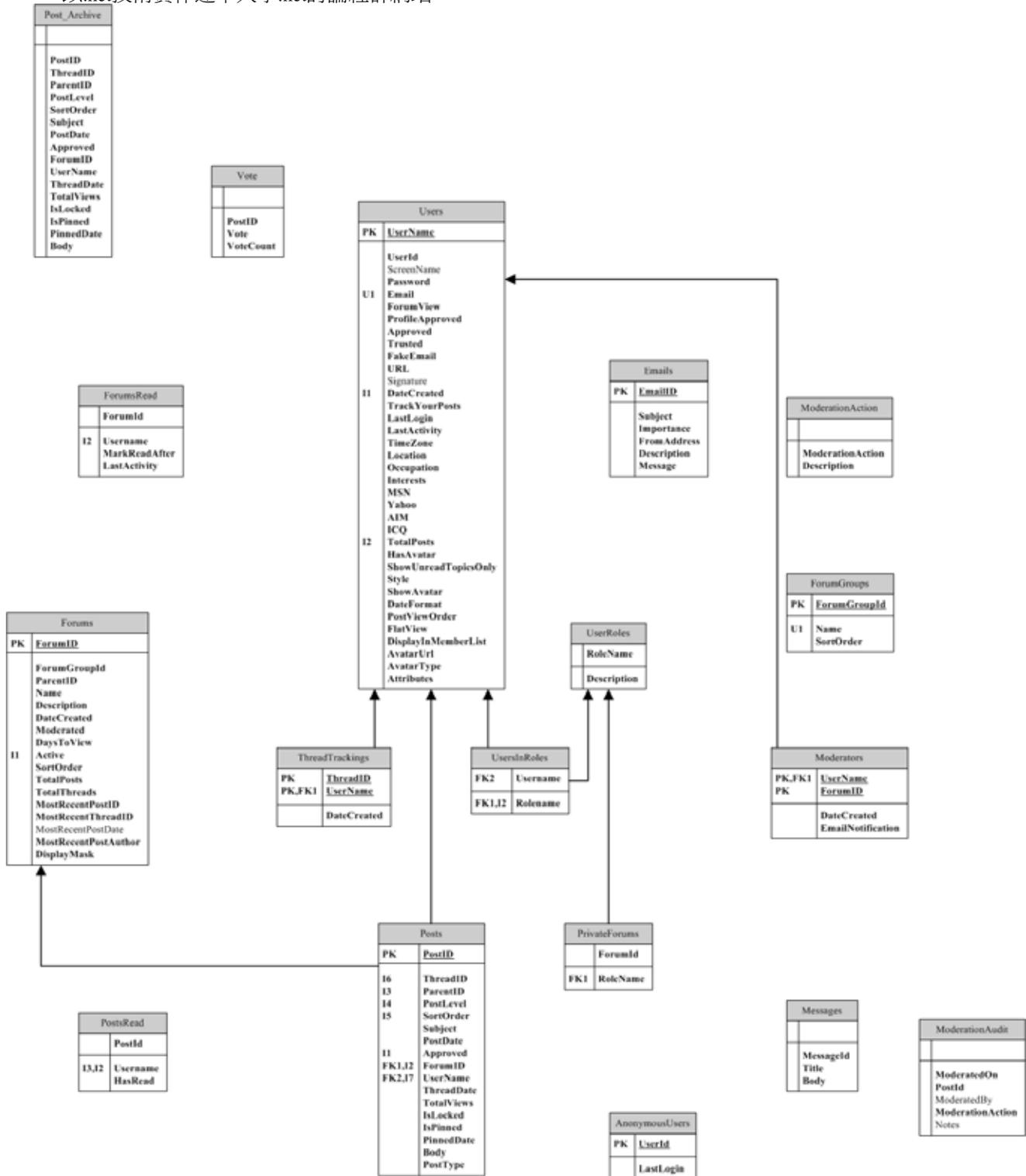


圖 3.9 Asp.net Forum 資料表與關聯性

可以看出整個系統的資料表是分層的，依照

- ForumGroup
- >Forums
- >Posts

這樣的方式，並透過與 User 資料表的關聯完成資料庫的設計。

3-3 系統整合實作

在這一節將介紹的是兩個系統在資料庫的整合，使用者及安全性上的整合，與網站系統架構的整合實作。

3-3.1 資料庫整合實作

在圖 3.6 與圖 3.9 中，我們看到了兩個資料表與它們的 Users 資料表的關聯情形。利用以下方式，可以將使用者資料表做整合：

1. 將 AspNetFroum 的資料表以及預存程序，檢視表都加入 Hurricane 資料庫中。
(Users 資料表除外)
2. 將關聯改爲針對 Hurricane 的 Users 資料表的 UserName 做索引鍵。

包括 Forums，Posts，ThreadTracking，UserInRoles，Moderators。

整合得到的資料表如下圖所示，紅色的部分是整合後的資料表關聯情形：

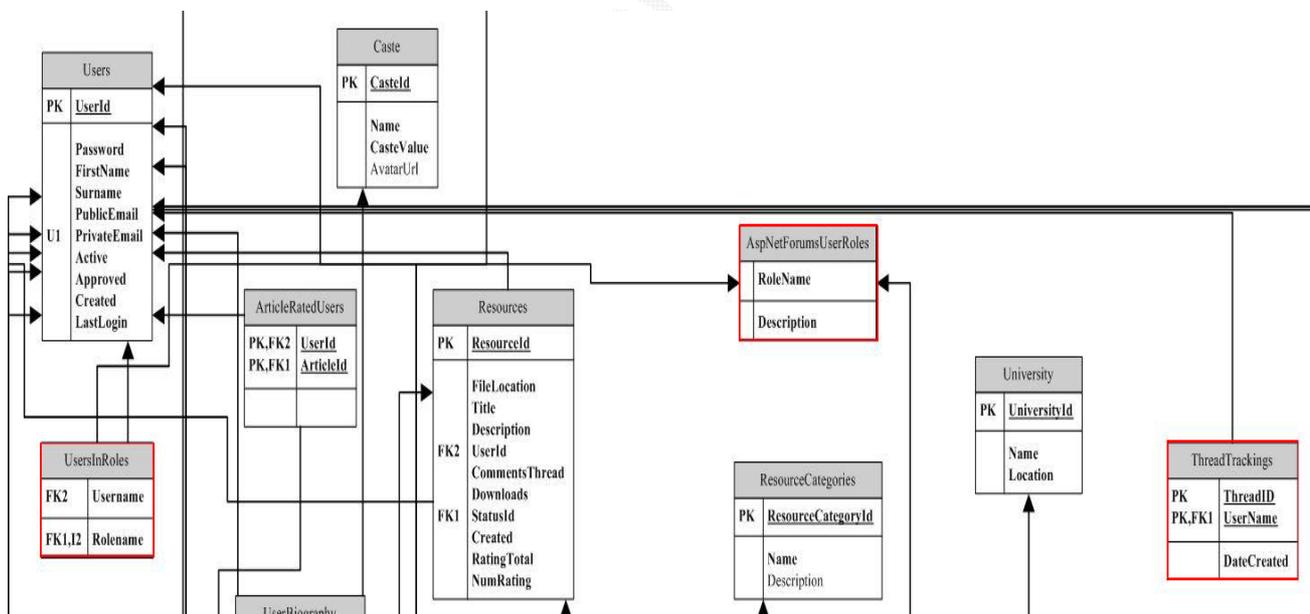


圖 3.10 整合後的 Hurricane 資料庫

3-3.2 使用者與安全性整合實作

兩個系統都是使用 Forms 表單驗證的方法，也就是在資料庫中查詢使用者是否在資料庫中，以及帳號密碼是否正確。在 Asp.net 的安全性類別，使用了 FormsAuthentication 類別來負責使用者認證與權限的建立。

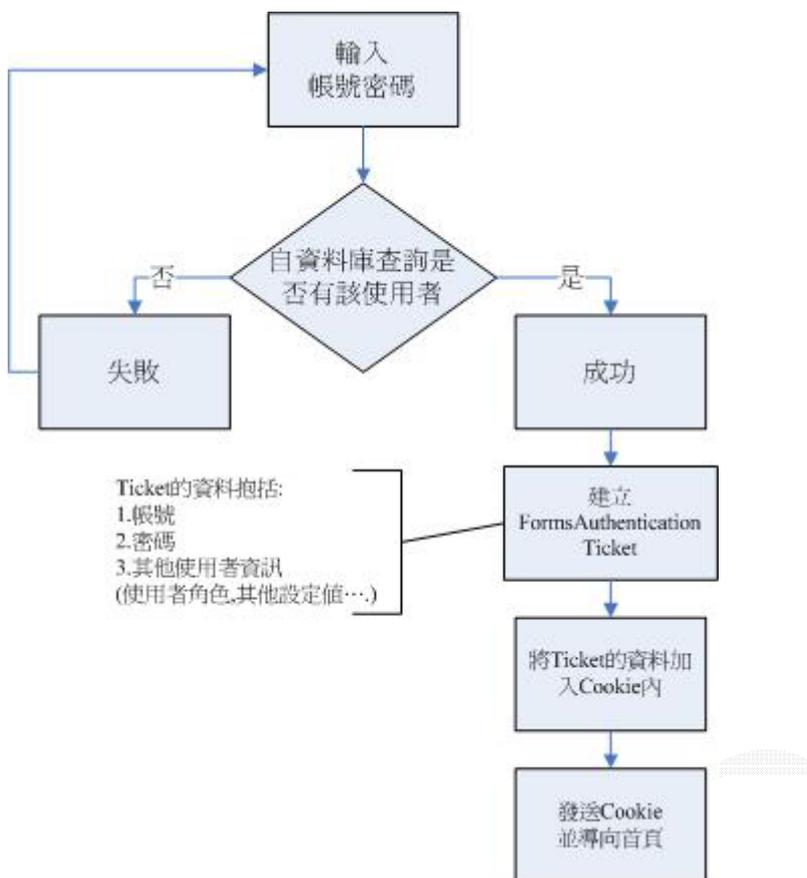


圖 3.11 FormsAuthentication 表單驗證的運作流程

兩個系統需要登入兩次，造成不必要的麻煩。修改登入及註冊模組，讓使用者可以自由的通行這兩個系統。所以在 Ticket 裡的 UserData 部分也加入了 AspNetForum 的一些設定值，讓 Cookie 可以通用。並且將 AspNetForm 的 Login 部分去掉，省略認證的步驟，把兩種 UserRole 加入 UserData 中，像 Admin 的帳號就可能是 Administrator 的 Role，而且也是 Forum Admin 的 Role。

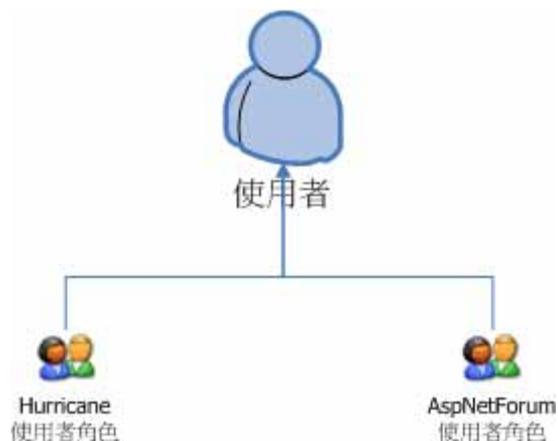


圖 3.12 使用者角色整合示意圖

3-3.3 網站架構的整合實作

1. 討論區加入網站架構的整合：

這個部分，首先我們要先將討論區加到/Forums 資料夾。

因為應用程式的設定值都保存在 Web.config 的關係，我們必須修改

Web.config，並加到網站的 Web.config。以下是有關討論區的設定值，包括換頁設定，連線字串，以及站台其他設定：

```
<!--
*****
Application Settings
*****
-->
<add key="availableSkins" value="default" />
<add key="defaultPageSize" value="25" />
<add key="connectionString" value="server=localhost;Trusted_Connection=true;database=Hurricane" />
<add key="defaultDateFormat" value="dd MMM yyyy" />
<add key="defaultTimeFormat" value="hh:mm tt" />
<add key="pathToTransformationFile" value="/transform.txt" />
<add key="smtpServer" value="localhost" /> <!-- Can specify SMTP Server to use to send out emails. Use
"default" to use the default Windows 2000 SMTP Server -->
```

以.net技術實作逢甲大學.net討論社群網站

```
<add key="allowDuplicatePosts" value="false" /><!-- Whether or not you wish to allow messages with
duplicate bodies being posted in various forums -->
<add key="dbTimeZoneOffset" value="0" /><!-- The timezone offset of your database server. (GMT is +0; EST
= -5;) -->
<add key="siteName" value="逢甲大學.net研究群討論區" /><!-- The name of your AspNetForums.NET Web site.
-->
<add key="DataProviderClassName" value="AspNetForums.Data.SqlDataProvider" />
```

在這裡修改路徑：

```
<add key="urlWebSite" value="http://localhost/forums" />
<add key="forumsDirectory" value="/forums"/>
<add key="forumsName" value="討論區"/>

<!--
If the ASP.NET Forums are configured to run in a directory that is not
an IIS VRoot, provide the name of that directory here.
<add key="forumsDirectory" value="/Forums" />
```

這樣就可以確保我們的更改不會造成程式錯誤。

2. 其他細部功能的加入：

包括加入 **HtmlEdit** 控制項方便討論區文章編輯，修改 **NavigationMenu**，版面整合，以及站內搜尋功能的加入。其他小項的修改便不多敘述。

● 修改 **Controls/CreateEditPost.cs**，最重要的是加入 **GetTextBox** 方法：

```
/*
// GetTextBox
//
/// <summary>
/// Returns the control contents as a 'TextBox'.
/// This is to handle possibilities for various types of control as the content editor
/// </summary>
/// <param name="edit_control">Content editor control</param>
*/
private TextBox GetTextBox(Control edit_control)
{
    if (edit_control is HtmlEdit)
    {
        return (TextBox)((HtmlEdit)edit_control);
    }
}
```

```
以.net技術實作逢甲大學.net討論社群網站
}
else
{
    return (TextBox)edit_control;
}
}
```

HtmlEdit 是一個我們加入的控制項，在 HtmlEdit.cs 裡設定了他的功能。

除此之外，我們必須在 AddPost.aspx 及 EditPost.aspx 中，設定允許 HTML 標籤的的傳送：

```
<%@ Page validateRequest="False" %>
```

如此便不會對 HTML 標籤做檢驗的動作。未做此設定，會使得程式發生錯誤，因為 Asp.net 預設是不允許傳送 HTML 的。

- 修改討論區的 Navigation Menu：

在討論區的上方有幾個提供的連結，其中有些是用來修改會員資料的部分，關於這部分把它的設定改成 “Visible = False” 避免使用者再去動到論壇程式的會員部分，這樣會造成程式無效。

在整合程式碼後，接下來我們要針對網頁表單做版面的整合。首先，Hurricane 系統利用到的版面控制項共有以下四個：

1. Header 控制項，用來顯示網站的 Logo，放在最上面。
2. SideBar 控制項，由許多控制項集合而成，顯示在最左邊，共有 Login、SearchBox、SiteLink、Polls 等四個控制項，以及動態顯示的 TaskPlane 工作列控制項。
3. Footer 控制項，放在網頁的最下面

以.net技術實作逢甲大學.net討論社群網站

4. Stylesheet 控制項，目的是用來與 CSS 檔案做連結達到佈景的功能。

利用最頂端的控制項使用宣告：

```
<%@ Register TagPrefix="ucl" TagName="Header" Src="MainControls/Header.ascx" %>
```

可以在網頁表單內加入一個 Header 控制項

```
<ucl:Header id="Header1" runat="server"></ucl:Header>
```

如此便可以達成版面的整合，其他控制項依此類推。有關與討論區的整合功能修改大致完成。



圖 3.13 整合後的討論區外觀

- 站內搜尋功能

這個部分就要靠自己動手去實作了。首先需要被搜尋的資料有：

以.net技術實作逢甲大學.net討論社群網站

- ✓ 新聞
- ✓ 文章
- ✓ 資源下載
- ✓ 日誌(Blog)
- ✓ 討論區

利用 SQL SERVER 內建的全文檢索功能，可以讓我很快的完成站內搜尋。

首先，在資料庫內部啓動全文檢索功能，這個功能只有在 SQL SERVER 2000 Enterprise Edition 才有，而且需要啓動。接下來設定全文檢索的目錄，在下圖中，我一共設定了 Resources，News，BlogEntries，Articles 這四個資料表，並且在每個資料表的 Title 及 Body 欄位建立全文檢索的索引。如此一來便完成建立全文檢索索引的功能，接下來就要開始加入預存程序，讓程式呼叫全文檢索並回傳結果。

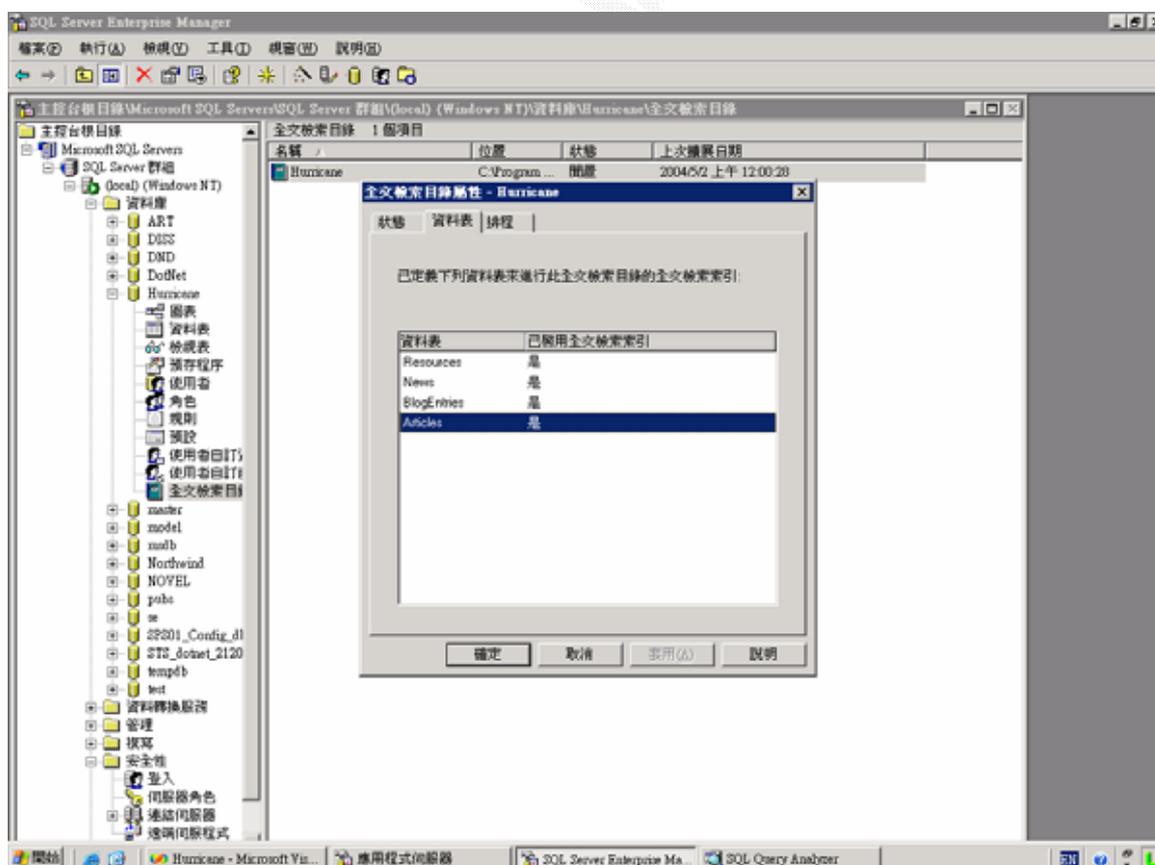


圖 3.14 建立全文檢索目錄

以下是一個建立全文檢索查詢的預存程序，可以取得文章的資料。

```
CREATE PROCEDURE dbo.Articles_SearchArticles
@Search nvarchar(200)
AS
SELECT
    ArticleId,
    UserId,
    Title,
    Description,
    Body,
    Views,
    Created,
    RatingTotal,
    NumRating,
    CommentsThread,
    StatusId
FROM Articles
WHERE FREETEXT(Body, @Search) OR FREETEXT(Description, @Search) OR FREETEXT(Title, @Search)
AND StatusId != 1

GO
```

用藍色字標示的是全文檢索的語法，利用 FreeText 我們可以不用加入 And 或是 Or 的額外字元，只需要把關鍵字加上空格區分即可。其他在新聞、Blog、資源下載的搜尋依此類推。

接下來是修改 IArticlesData.cs 還有 ArticlesData.cs。前者是一個類別的介面，我們必須明確地宣告一個方法，並在後者的程式碼實作這個方法。

```
public Article[] GetArticlesBySearchText(string search)
{
    //產生SQL連線的物件
    SqlConnection conn = this.GetConnection();
    //建立SQL命令物件，並且執行剛才寫好的預存程序。
    SqlCommand cmd = new SqlCommand("Articles_SearchArticles", conn);
```

以.net技術實作逢甲大學.net討論社群網站

```
SqlDataReader reader = null;

Article[] result = null;

ArrayList tempResult = new ArrayList();

//加入參數

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@Search", SqlDbType.NVarChar).Value = search;

conn.Open();

reader = cmd.ExecuteReader();

while(reader.Read())

{

//取得資料並存入暫存的資料陣列中

tempResult.Add(new Article((int)reader["ArticleID"], (string)reader["Title"],

(string)reader["Description"], (string)reader["Body"], (string)reader["UserId"],

(string)reader["CommentsThread"],

(int)reader["Views"], (int)reader["StatusId"], (DateTime)reader["Created"],

(int)reader["RatingTotal"], (int)reader["NumRating"],

this.GetArticleCategories((int)reader["ArticleId"]))));

}

if(tempResult.Count > 0)

result = (Article[])tempResult.ToArray(typeof(Article));

reader.Close();

conn.Close();

//回傳結果

return(result);

}
```

其他類別的資料亦是如此。

在網頁表單後置的程式碼中，我們加入了呼叫此方法以及將回傳結果與 Repeater

控制項繫結的動作：

```
//呼叫GetArticlesBySearchText方法，並把回傳的結果指定給Repeater控制項
rptSearchArticles.DataSource = ad.GetArticlesBySearchText(sqlSearchText);
//讓Repeater控制項與資料繫結。
rptSearchArticles.DataBind();
//如果回傳的資料是Nothing，就不要顯示這個類別。
if(rptSearchArticles.Items.Count == 0 )
{
```

```
以.net技術實作逢甲大學.net討論社群網站  
ArticleDIV.Visible = false;
```

```
}
```

如此便成功的完成了文章搜尋的功能，如下圖：

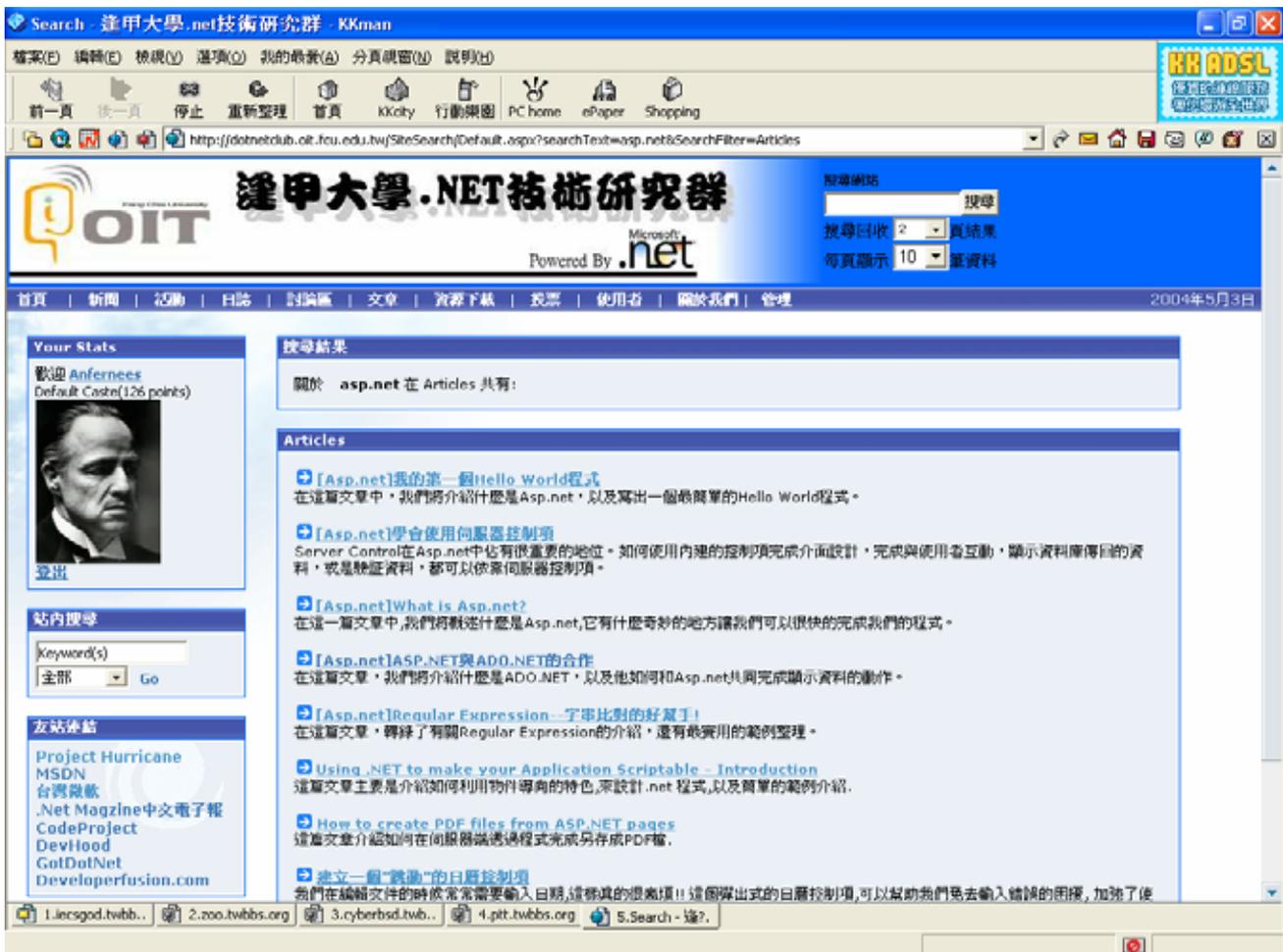


圖 3.15 文章搜尋畫面

這是找尋 asp.net 產生的結果。而有關討論區的搜尋只要呼叫已經寫好的搜尋功能，把 URL 更改成爲/Forum/Search.aspx，再加上搜尋字的參數便可以啓動，如下圖所示：



圖 3.16 討論區搜尋功能

第四章-與搜尋引擎的整合與系統展示

4-1 與個人化搜尋引擎的整合

由於兩邊都是以 Visual Studio.Net 2003 開發，整合的功能變得比較方便。首先我們先將已經做好的 Search 專案加入至原來網站的專案中，接下來要讓 Search 專案與 Hurricane 專案連結。利用專案之間的依存性，我們可以發現專案之間的建置是有順序的。例如說，DataAccess 專案是 Hurricane 的後置程式碼，所以它必須在 Hurricane 之前先編譯完成，Search 專案也應如此。

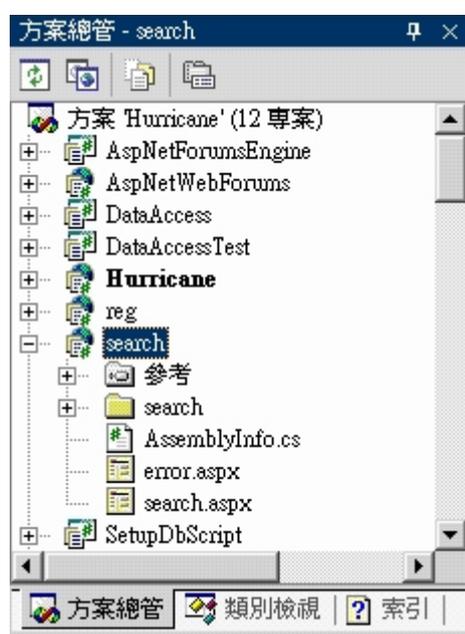


圖 4.1 加入專案

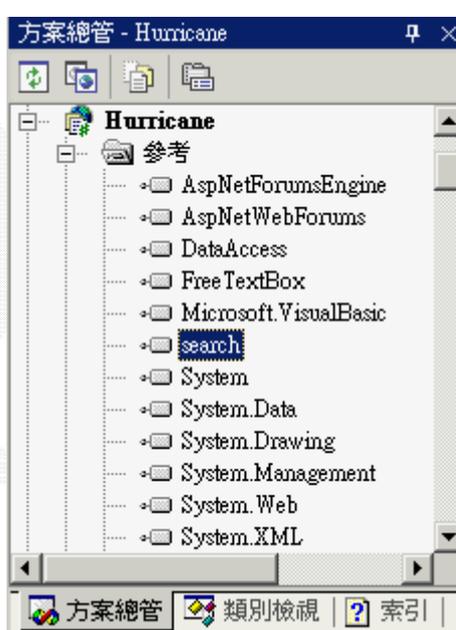


圖 4.2 將專案引入參考

在網站路徑的部分，也需要做些許修改。由於原本自己就建了 Search 資料夾，用來做站內搜尋，不過因為與系統整合造成名稱衝突，需要重新修改命名空間才能達到完成整合的目標。

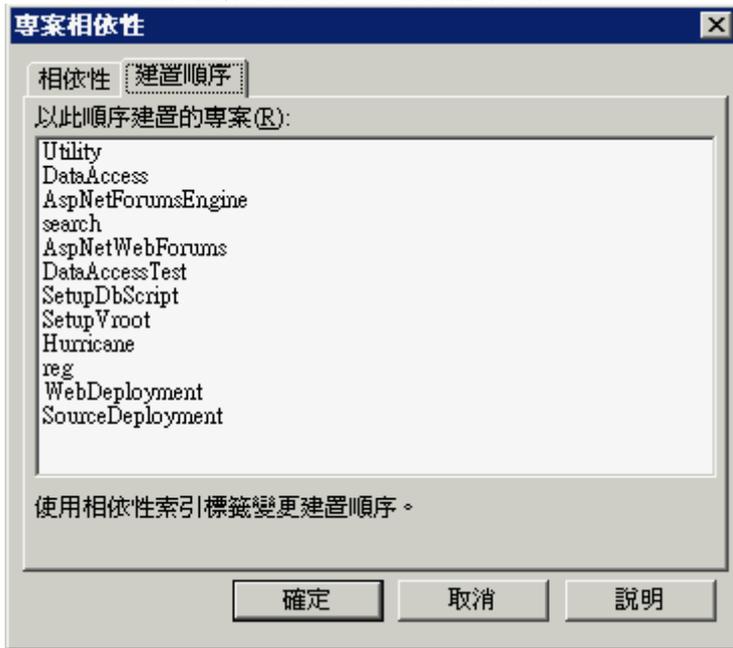


圖 4.3 專案相依性與建置順序

我們在建置專案後，利用專案的參考，我們可以把 Search 專案變成網站的一個 Library，並且可以呼叫他的功能，而且在 Search 寫好的控制項可以引入網站，網站上的套版控制項－Header.ascx、Footer.ascx 等等，也都可以套用至 Search 專案中。

接下來我們將 SendQuery.ascx 控制項實際加入網站 Sidebar.ascx 控制項中。

首先宣告引入控制項：

```
<% @Register TagPrefix="uc1" TagName="sendQuery" Src="/search/search/sendQuery.ascx" %>
```

在網站中加入控制項標籤：

```
<uc1:sendQuery id="SendQuery1" runat="server"></uc1:sendQuery>
```

經過套版面後，我們順利地將系統完成整合。

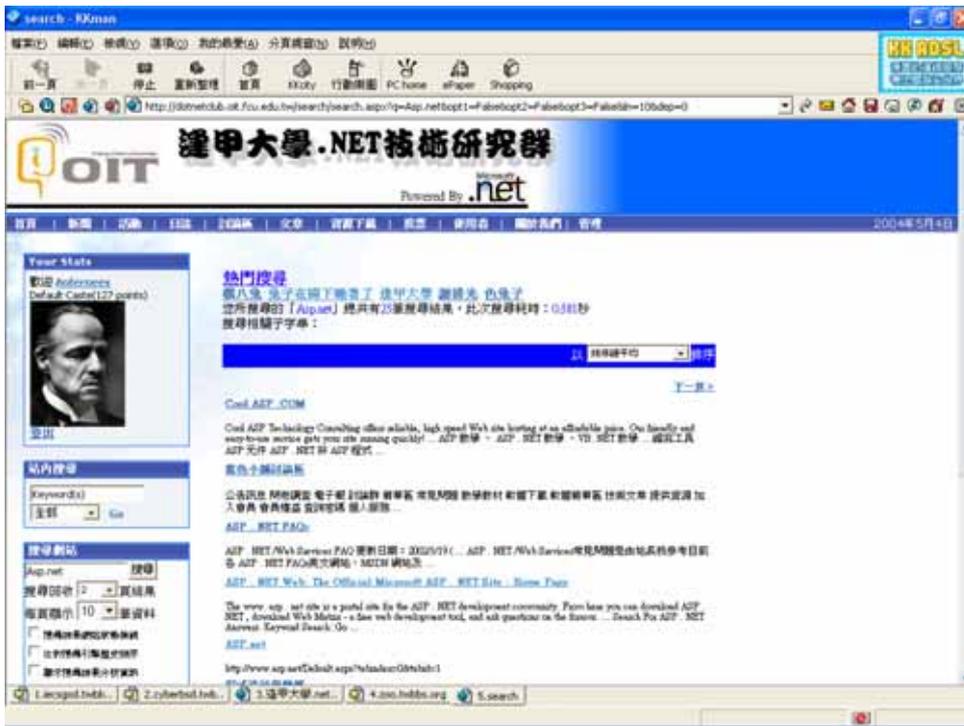
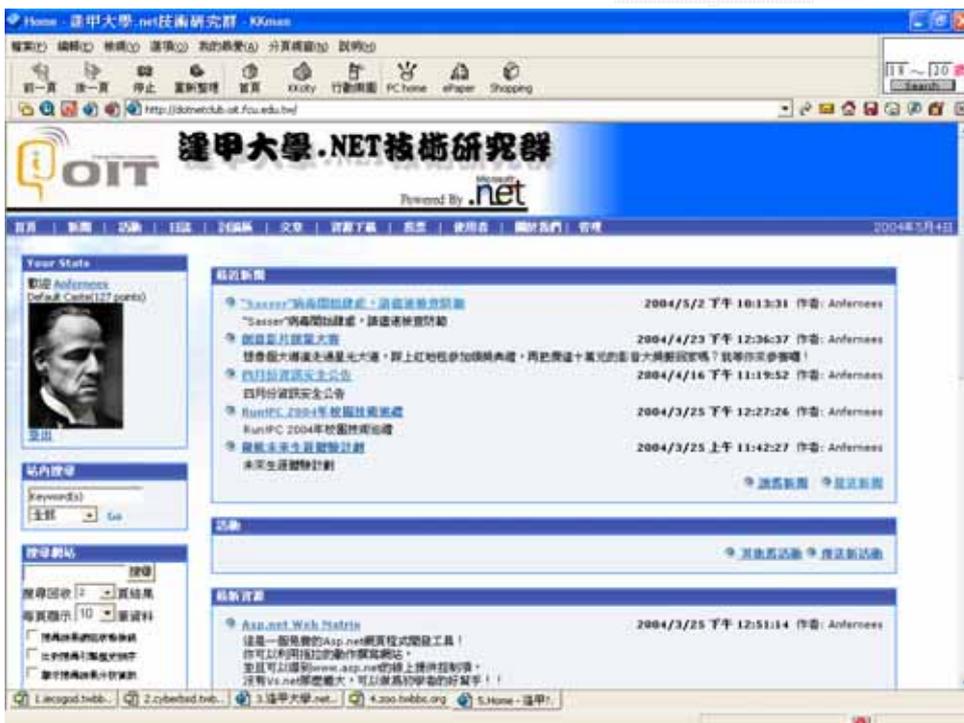


圖 4.4 輸入 Asp.net 關鍵字得到的查詢結果

4-2 系統展示

4-2.1 網站首頁



以.net技術實作逢甲大學.net討論社群網站

圖 4.5 網站首頁

網站首頁包括了最近新聞、活動、最新資源、最新文章以及其他連結部分。

使用者可以在首頁開始找尋資料或是直接進入其他功能連結。

4-2.2 新聞功能

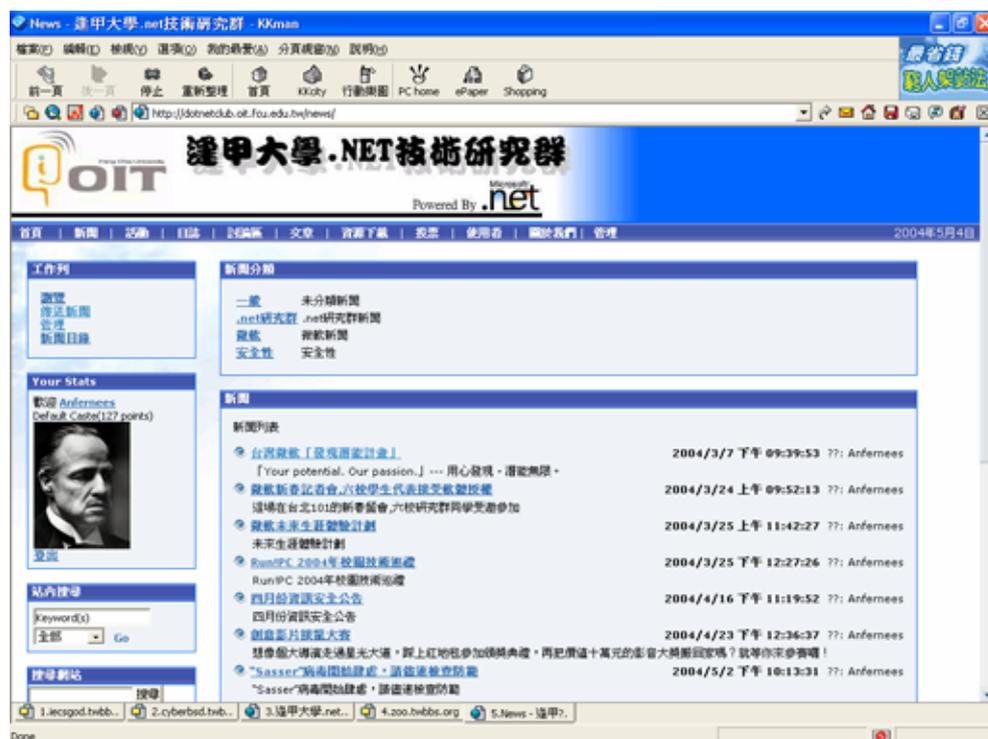


圖 4.6 新聞分類頁面

在每個功能頁面都會有符合你的權限的工作列。如果你只是一般訪客，那麼

你就沒有發送新聞的能力，也不能有管理者審核新聞的能力。

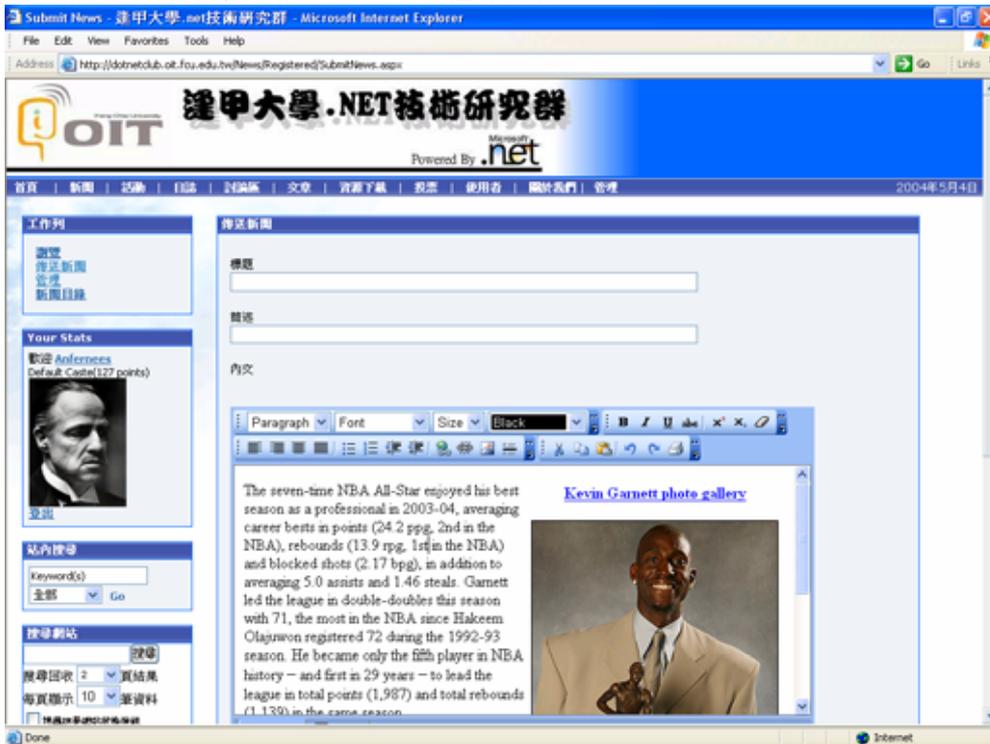


圖 4.7 發送新聞頁面

在這個部分我去結合一個控制項，可以讓使用者在線上編輯 Html 文件，並且提供所見即所得的環境。



圖 4.8 日期選擇以及分類選擇

以.net技術實作逢甲大學.net討論社群網站

這部分是傳送新聞功能的下半部，可以選擇輸入日期或是利用日曆點選。下面是分類，可以重複選取多個分類。



圖 4.9 管理新聞頁面

管理者可以在這裡修改新聞、審核新聞、刪除新聞等功能。



圖 4.10 修改新聞頁面



圖 4.11 新聞核准頁面

在點選更改新聞狀態後，可以選擇核准文章。

其他相同的功能如活動、文章的操作方式都如同新聞，在此不多做贅述。

4-2.3 日誌功能展示



圖 4.12 個人日誌首頁

在這裡可以看到左邊出現的工作列、加入站台連結、將自己的 Blog 做資料分類，以及個人日曆。右邊是顯示自己編輯的資料。有關編輯 Blog 的相關功能也如同新聞。



圖 4.13 閱讀及回應頁面

在這裡可以針對日誌的內容與其他人討論。



圖 4.14 日誌管理功能

在這裡可以加入 Blog 的連結、分類、資料的新增刪除功能

4-2.4 討論區功能展示



圖 4.15 討論區首頁

在這裡可以看到所有的討論區以及在右上角的功能列，點選後便可進入討論目錄或是討論區。



以.net技術實作逢甲大學.net討論社群網站

圖 4.16 討論區內標題頁面



圖 4.17 討論文章頁面



圖 4.18 建立新討論文章



圖 4.19 討論區搜尋頁面



圖 4.20 討論區管理介面

共有使用者管理、論壇(討論區)管理、群組管理、角色管理、信息管理等項目。

4-2.5 資源下載功能展示



圖 4.21 資源下載首頁



圖 4.22 傳送資源頁面

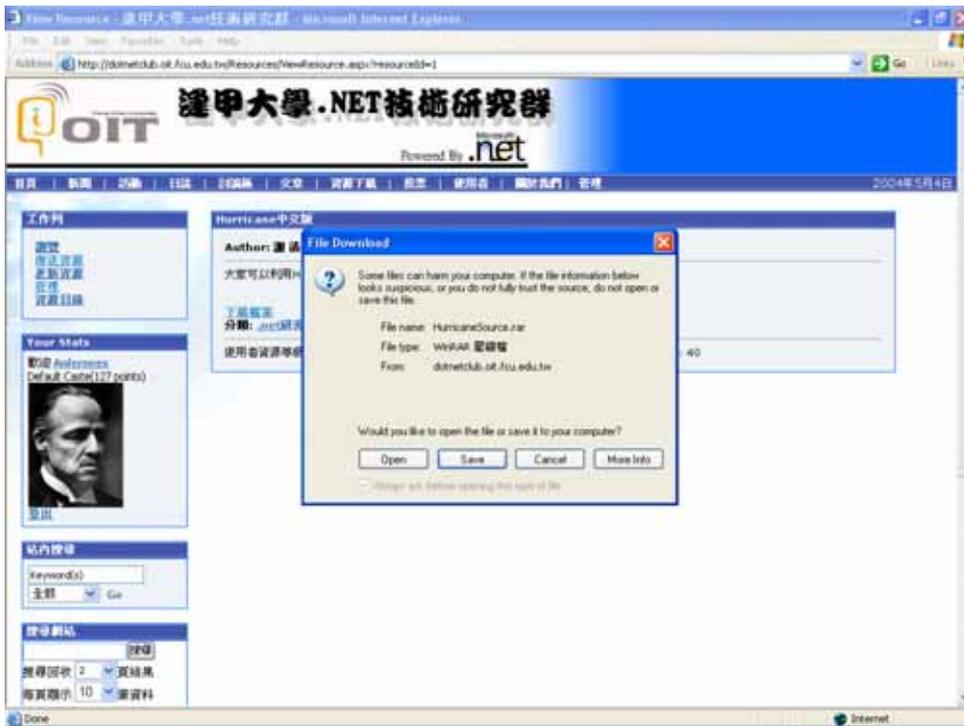


圖 4.23 下載頁面

當下載完成後，會增加下載次數。



第五章－結論

5-1 總結

目前網站已經順利運作中，而網站的原始程式也提供給各校微軟俱樂部同學使用，並且在使用的過程中也很感謝各位研究群同學給我意見，以及錯誤修正。希望可以建立起每個研究群的知識庫，對日後各校研究群的發展會有更多的幫助。

5-2 遭遇到的困難

其實困難是持續發生的。從規劃、網站的第一版產生，確實讓我了解萬事起頭難的道理。不熟悉的工具和架構，還有對.net 平台的茫茫然，確實讓我花了不少時間了解。最大的困難，應該是第二版整合的過程。裡面最大的論壇系統，程式碼不只是多，而且程式設計架構是我一直不能理解的。剛開始對物件導向程式設計的不熟悉，也是造成網站進度落後的一個原因。

5-3 未來展望與期許

目前逢甲大學.net 討論社群網站已經算是功能完整，不過接下來仍需要和研究群同學共同發展網站的特色。在招收新進學弟學妹之後，網站的維護經營已經不再是我一個人單打獨鬥，以後將會和研究群一起持續經營下去。以下便略述幾項未來展望：

1. 討論區轉信功能：

雖然研究群的網站會員人數持續增加，不過以討論區而言，提問人提出問題後最希望的是立刻有人可以解決他的問題。透過各校微軟學生俱樂部網站的

建立，希望可以利用 **Web Service** 達到互相轉信的機制，讓同學間學習的方向更廣。

2. 個人化的功能：

目前研究群的經營方向將朝著手持式智慧型裝置的系統開發發展。未來希望可以提供一連串與網站互動的功能，讓使用者在任何地方都可以使用網站的服務。

5-4 心得

提筆至此，這份書面報即將結束。回想起升上大三的暑假開始到現在，也快要兩年的時間了。過程中因為自己外務太多的問題，沒有放下全部的心力去做，使得專題進度一延再延，也讓我了解自己的缺失。現在完成了，心情真是複雜阿！

大三開學後，和宗碩到台北恆毅資訊受訓兩星期 **Asp.net** 和 **SQL SERVER** 的訓練課程，每天往返中壢與台北的過程是很充實的，也開始了我對.net 平台的學習之旅。在這個過程中，養成了自己習慣去 **MSDN** 找新資料、查 **MSDN Library**，看一些介紹.net 知識的英文文章，去國外的討論社群問問題，和 **Wrox** 的作者互通信件。一個人做專題，很多時候很容易被自己內在的惰性影響，自己也好幾次進度走走停停，而且在過程中我發現自己不是那麼有想法的人，我應該時時刻刻激發自己的思維，並且和大家分享討論。最後的這個學期讓我學到很多，不只是程式實作上，自己的求學心態上也有不小的轉變。與宗碩的系統整合工作進行的順利，也是後來在學習的時候變得比較扎實，可以馬上上手的原因。

最後要感謝校長及趙啓時老師的指導。從剛開始對任何事情都懵懵懂懂的我

以.net技術實作逢甲大學.net討論社群網站

們，老師花了很多的心力讓我們成長，糾正我們不正確的求學方式。除此之外，專題之外的問題，老師也很樂意聽我的煩惱，給我建議。我發覺我自己在做專題的過程成長了一些。也非常感謝政巨學長在網站規劃時給我的許多建議，還有宗碩和其他學弟們的建議與測試。謝謝大家！



參考文獻

書目資料

- Alex Homer David Sussman, Brian Francis , Robert Howard, Wrox Author Team, Karli Watson, Richard J Anderson, ” **Professional ASP.NET 1.0, Special Edition**”. wrox.
- Marco Bellinaso , Kevin Hoffman, ” **ASP.NET Website Programming: Problem - Design - Solution, C# Edition**”, wrox.
- 董大偉, "ASP.NET 程式設計徹底研究". 文魁資訊
- “IIS 5 網站建置與管理”, 碁峰資訊.

網站資料

- Devhood.com
<http://www.devhood.com>
- Asp.net Community
<http://www.asp.net>
- ProjectHurricane
<http://www.projecthurricane.com>
- 北縣技術論壇
<http://forum.tpc.edu.tw/AspNetForums>
- MSDN Library
<http://msdn.microsoft.com>
- .Net Magzine 中文電子報
<http://www.netmag.com.tw/>
- CodeProject.com
<http://www.codeproject.com/>
- GotDotNet.com
<http://www.gotdotnet.com/>
- DeveloperFusion.com
<http://www.developerfusion.com>
- Move-to.Net
<http://www.move-to.net>