

# 臺灣二〇〇七年國際科學展覽會

科 別：電腦科學

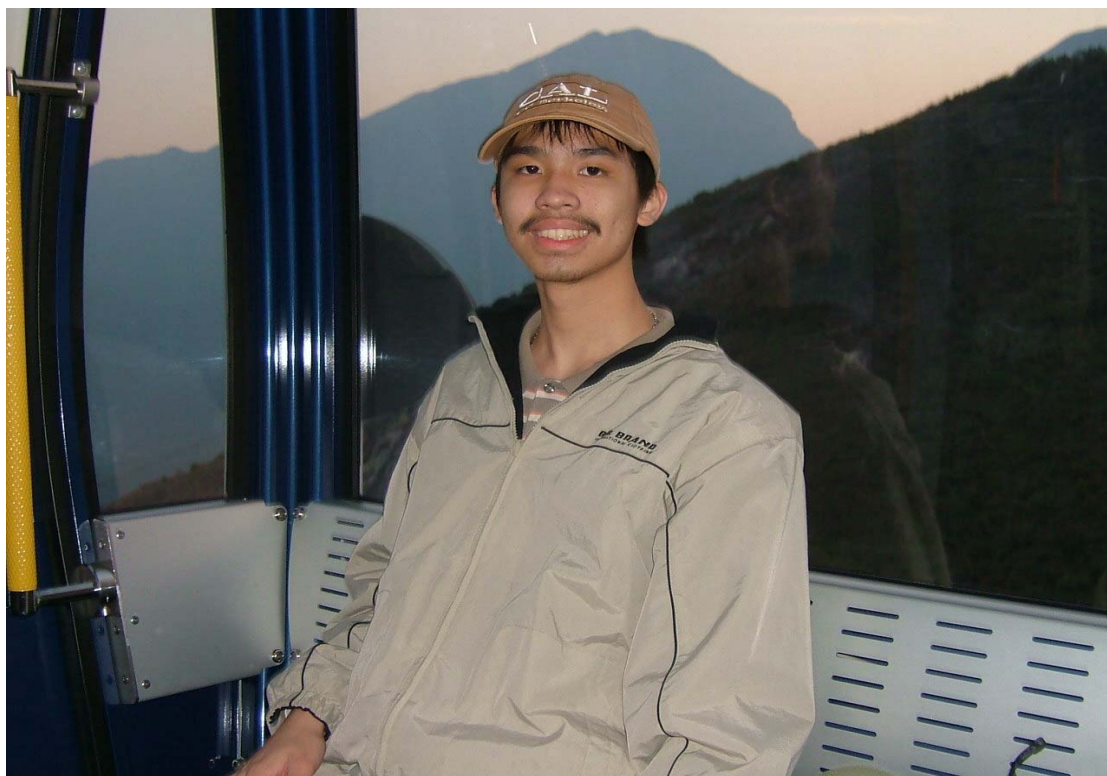
作 品 名 稱：利用 Google Maps 建立高中地理資訊系統

學校 / 作者：國立臺灣師範大學附屬高級中學 陳劭恆  
國立臺灣師範大學附屬高級中學 杜俊翰

## 作者簡介



我是陳劭恆，目前就讀師大附中高中一年級學生，自國中開始接觸電腦程式設計，便產生濃厚興趣，曾於國中二年級參加台北縣以及全國科展，獲得評審肯定，並對此產生興趣。感謝林俊卿老師的指導，使我有機會攝取這個機會，參加國際科展，汲取這次經驗。



我是杜俊翰目前就讀於師大附中一年級。從我小時候，就深刻喜歡研究數理，對電腦科學也是有興趣，曾參加台北市科展，希望透過這次國際科展，學習並研究新知，加上老師指導，使我對網路有更深一步的了解。相信在老師及同學的努力之下，能獲得好成績。

# 利用 Google Maps 建立高中地理資訊系統

## 摘要

台灣通史「婆娑之洋，美麗之島」，身為台灣人，應知台灣事，本研究本維基百科（Wikipedia）所揭示的「海納百川，有容乃大」精神，以 Apache 網頁伺服器為平台，MySQL 為資料庫，利用 AJAX、PHP 網頁技術，並使用 Google Maps 提供之服務，建立符合 Web 2.0 精神的高中地理資訊共享系統。在此系統下，我們允許教師(編輯者)以合作方式上網提供與課程相關的地理資訊，並可進行資訊修改或刪除，希望集合所有教師之力，提供給學生最豐富、最正確的高中地理資訊系統。

## A collaborating geographic information system based on Google Maps for high school geography education

### **Abstract**

In this paper, we propose a new map-based collaboration environment “MyGIS” for geographical information distribution. In MyGIS, the publishers/teachers can easily collaborate to annotate a spot with contents such as messages and links. The annotations of a spot are updated on the map in real-time which enables the receivers/students to correctly study at anywhere and anytime. To ensure the trustfulness of the content, the publishers are authenticated using login mechanism. In this study, several technologies including Apache, PHP, MySQL, Ajax are involved. Finally, We implemented MyGIS on Google Maps and evaluated its feasibility and usefulness.

## 壹、前言(研究動機和研究目的)

### 一、研究動機

十六世紀初，西方國家海上探險發達，來到東方的印度、印尼、菲律賓、中國、日本一帶尋找新的殖民地，從事貿易。當他們經過臺灣海峽時，看見臺灣山嶽如畫、樹木青蔥、嘆為奇觀。因此，葡萄牙人驚探之為「Ilha Formosa」，也就是「美麗之島」的意思。從此，西方人就以「Formosa」（福爾摩沙）稱呼這個島。

台灣通史「婆娑之洋，美麗之島」，身為台灣人，應知台灣事，本研究以 Apache 網頁伺服器為平台，MySQL 為資料庫，利用 AJAX、PHP 網頁技術，並使用 Google Maps 提供之服務，建立符合 Web 2.0 精神的高中地理資訊共享系統。

### 二、研究目的

1. 認識 Google Maps 及 API 之應用
2. 了解 AJAX 技術之應用
3. 認識資料庫處理
4. 建立高中地理資訊系統(MyGIS)

## 貳、研究方法或過程

### 一、安裝 Apache、PHP、MySQL

Apache 網頁伺服器，是透過超文件傳輸協定(Hypertext Transfer Protocol, HTTP)來服務網路上的要求，它通常會傳回一些檔案，其內容大多是以超文件標示語言(Hypertext Markup Language, HTML)寫成。Apache 是由 "a patchy server" 一詞所衍生，在最早的時候，Apache 便是針對 NCSA 網頁伺服程式，進行一系列修正 (patch) 而來的，目前是當今網際網路上最常被使用的網頁伺服器。

『PHP is a server-side, cross-platform, HTML embedded scripting language.』，也就是在伺服器端的跨平台 HTML 內嵌式描述語言，型態類似 Java script、VB script、IIS 的 ASP (Active Server Page)。有人也把 PHP 說成是『Personal Homepage Program』。PHP 是使用在 Web Server 端的動態網頁 (Dynamic Web) 開發。主要處理 Web Server 端的計算、文字、檔案.....等，且 PHP 內建有與眾多資料庫連結的能力，非常適合作為網頁應用程式的開發工具。

MySQL 是由瑞典斯德哥爾摩一家公司 T.c.X DataKonssultAB (<http://www.tcx.se>) 所開發而的。根據官方說法，MySQL 是一種多使用者(Multi user)、多執行緒(multi thread) 的資料庫系統，支援一般的 SQL 語法，執行速度快，並且支援多種作業系統平台，例如：Win32 平台、Unix-like 系統等。

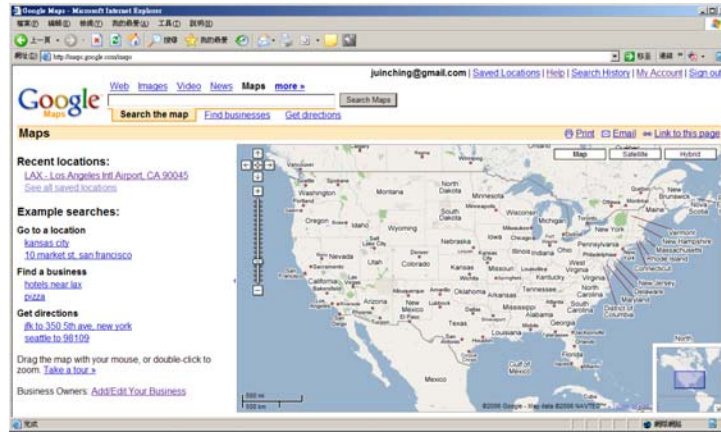
本研究將以 apache 作為網站伺服器，MySQL 作為資料庫伺服器，以 PHP 為伺服器端執行語言，利用 AJAX 技術進行系統的開發。

### 二、Google Maps 及 API

## 1. Google Maps

Google Maps 是 Google 公司提供的網路地圖服務，包含以下三種瀏覽模式：

- Map
- Satellite
- hybrid

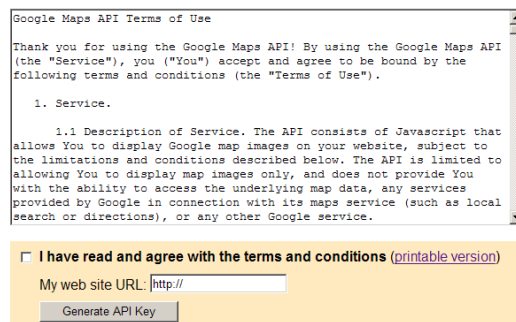


使用者可利用滑鼠拖曳/點選或使用移動/縮放控制項進行地圖的瀏覽。

## 2. Google Maps API 申請流程

在使用 Google Mapss API 之前，你必須向 Google 申請到一組 Key，申請的方法如下：

- 以你的帳號登入 Google，如果你還沒有帳號，請到 <https://www.google.com/accounts/> 申請。
- 連到 Google Mapss API Sign Up 網頁將 (1) 所決定的 URL 填入，並按 "Generate API Key" 取得 Key。
- 最後會出現三個方塊，分別是你取得的 Key，你指定的 URL，以及一個範例。建議是將範例中的 `<script src="..."></script>` 整行 Copy/Paste 下來。



## 3. Google Maps API 的使用限制

Google Maps API 使用上有些法律上的限制如下：

- 你的 Google Maps 服務必須免費開放給一般使用者。
- 如果你預估你的地圖服務會有超過每天 50,000 筆頁面的存取量，請主動聯絡 Google。
- Google 會不定期更新 API，你有責任跟著更新你的網頁中的 API。
- 你不能更改或隱藏 Google 的 logo。
- Google 有權力以後在地圖上放廣告，而你不能更改或破壞這些廣告。
- 有些地圖應用是 Google 不希望看到的，例如指明哪些地方可以購買到毒品或是任何違法行爲。

#### 4. 在網頁嵌入 Google Maps

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Google Maps JavaScript API Example</title>
    <script src="http://maps.google.com/maps?file=apis&v=2&key=ABQIAAAA0dPgouXvkNrj7hQgmzI4xS9FEVi-HGTK5xT0kobkY2"
      type="text/javascript"></script>
    <script type="text/javascript">
      //
      function load() {
        if (GBrowserIsCompatible()) {
          var map = new GMap2(document.getElementById("map"));
          map.setCenter(new GLatLng(37.4419, -122.1419), 13);
        }
      }
      //]]&gt;
    &lt;/script&gt;
  &lt;/head&gt;
  &lt;body onload="load()" onunload="GUnload()"&gt;
    &lt;div id="map" style="width: 500px; height: 300px"&gt;&lt;/div&gt;
  &lt;/body&gt;
&lt;/html&gt;</pre>
</div>
<div data-bbox="226 556 447 574" data-label="Section-Header">
<h4>5. 使用 Google Maps API</h4>
</div>
<div data-bbox="266 577 671 595" data-label="Text">
<p>Google Maps 提供為數眾多的 API，如圖所示。</p>
</div>
<div data-bbox="228 605 669 813" data-label="Table">
<table border="0">
<tbody>
<tr>
<td style="vertical-align: top;">
<p><b><a href="#">Audience</a></b></p>
<p><b><a href="#">Introduction</a></b></p>
<p><a href="#">The Hello World of Google Maps</a></p>
<p><a href="#">Browser Compatibility</a></p>
<p><a href="#">XHTML and VML</a></p>
<p><a href="#">API Updates</a></p>
<p><a href="#">Geocoding</a></p>
<p><a href="#">Routing and Local Search</a></p>
<p><a href="#">Marker Manager <sup>new!</sup></a></p>
<p><b><a href="#">Maps Examples</a></b></p>
<p><a href="#">The Basics</a></p>
<p><a href="#">Map Movement and Animation</a></p>
<p><a href="#">Adding Controls to the Map</a></p>
<p><a href="#">Event Listeners</a></p>
<p><a href="#">Opening an Info Window</a></p>
<p><a href="#">Map Overlays</a></p>
<p><a href="#">Click Handling</a></p>
<p><a href="#">Display Info Windows Above Markers</a></p>
<p><a href="#">Tabbed Info Windows</a></p>
<p><a href="#">Creating Icons</a></p>
<p><a href="#">Using Icon Classes</a></p>
<p><a href="#">Draggable Markers</a></p>
<p><a href="#">Encoded Polylines <sup>new!</sup></a></p>
<p><a href="#">Using XML and Asynchronous RPC ("AJAX") with Maps</a></p>
<p><a href="#">Custom Map Controls</a></p>
<p><a href="#">Custom Overlays</a></p>
</td>
<td style="vertical-align: top;">
<p><b><a href="#">Geocoder Examples</a></b></p>
<p><a href="#">Geocoding using JavaScript</a></p>
<p><a href="#">Extracting Structured Address Information</a></p>
<p><a href="#">Caching Geocodes</a></p>
<p><a href="#">HTTP Request</a></p>
<p><b><a href="#">Marker Manager Examples <sup>new!</sup></a></b></p>
<p><a href="#">A Weather Map</a></p>
<p><a href="#">Google offices</a></p>
<p><b><a href="#">Troubleshooting</a></b></p>
<p><a href="#">Other resources</a></p>
<p><b><a href="#">API Overview</a></b></p>
<p><a href="#">The GMap2 class</a></p>
<p><a href="#">Events</a></p>
<p><a href="#">The Info Window</a></p>
<p><a href="#">Overlays</a></p>
<p><a href="#">Controls</a></p>
<p><a href="#">XML and RPC</a></p>
<p><a href="#">Reducing Browser Memory Leaks</a></p>
<p><b><a href="#">Class Reference</a></b></p>
</td>
</tr>
</tbody>
</table>
</div>
<div data-bbox="258 832 519 851" data-label="Text">
<p>本研究主要使用的 API 如下：</p>
</div>
<div data-bbox="258 854 520 893" data-label="List-Group">
<ul>
<li>➤ Adding Controls to the Map</li>
<li>➤ Event Listeners</li>
</ul>
</div>
<div data-bbox="835 922 853 939" data-label="Page-Footer">
<p>7</p>
</div>
```

- Opening an Info Window
- Map Overlays
- Click Handling
- Using XML and Asynchronous HTTP with Maps
- XML and RPC

各 API 的詳細使用方式請見

<http://www.google.com/apis/maps/documentation/>。

### 三、AJAX

#### 1. AJAX 簡介

AJAX 全稱為「Asynchronous JavaScript and XML」（非同步 JavaScript 和 XML），是一種創建互動式網頁應用的網頁開發技術。它使用：

- 使用 XHTML+CSS 來表示信息；
- 使用 JavaScript 操作 DOM（Document Object Model）進行動態顯示及交互；
- 使用 XML 和 XSLT 進行數據交換及相關操作；
- 使用 XMLHttpRequest 對象與 Web 伺服器進行非同步數據交換；

傳統的 Web 應用允許用戶端填寫表單(form)，當提交表單時就向 Web 伺服器發送一個請求。伺服器接收並處理傳來的表單，然後送回一個新的網頁。這個做法浪費了許多頻寬，因為在前後兩個頁面中的大部分 HTML 代碼往往是相同的。由於每次應用的交互都需要向伺服器發送請求，應用的響應時間就依賴於伺服器的響應時間。這導致了用戶界面的響應比本地應用慢得多。

與此不同，AJAX 應用可以僅向伺服器發送並取回必需的數據，它使用 SOAP 或其它一些基於 XML 的頁面服務介面，併在客戶端採用 JavaScript 處理來自伺服器的響應。因為在伺服器和瀏覽器之間交換的數據大量減少（大約只有原來的 5%），結果我們就能看到響應更快的應用。同時很多的處理工作可以在發出請求的客戶端機器上完成，所以 Web 伺服器的處理時間也減少了。

#### 2. XMLHttpRequest

XMLHttpRequest 是一組 API 函數集，可被 JavaScript、JScript、VBScript 以及其它 web 瀏覽器內嵌的腳本語言調用，通過 HTTP 在瀏覽器和 web 伺服器之間收發 XML 或其它數據。XMLHTTP 最大的好處在於可以動態地更新網頁，它無需重新從伺服器讀取整個網頁，也不需要安裝額外的外掛程式。該技術被許多網站使用，以實現快速響應的動態網頁應用。例如：Google Gmail、Google Suggest 以及 Google Maps 等。

本研究將使用 AJAX 的 XMLHttpRequest 技術與伺服器進行資料存



取，如下圖所示。

#### XML and RPC

The Google Maps API exports a factory method for creating `XmlHttpRequest` objects that work in recent versions of IE, Firefox, and Safari. The following example downloads a file called `myfile.txt` and displays its contents in a JavaScript `alert`:

```
var request = GXmlHttp.create();
request.open("GET", "myfile.txt", true);
request.onreadystatechange = function() {
  if (request.readyState == 4) {
    alert(request.responseText);
  }
};
request.send(null);
```

The API also exports a simpler method for typical HTTP GET requests called `GDownloadUrl` that eliminates the need for `XmlHttpRequest` `readyState` checking. The example above could be rewritten using `GDownloadUrl` like this:

```
GDownloadUrl("myfile.txt", function(data, responseCode) {
  alert(data);
});
```

You can parse an XML document with the static method `GXml.parse`, which takes a string of XML as its only argument. This method is compatible with most modern browsers, but it throws an exception if the browser does not support XML parsing natively.

See the [GXmlHttp](#) and [GXml](#) class references for more information.

## 四、資料庫操作與 SQL

本研究使用到的資料庫的查詢，共包含有篩選、新增、更新及刪除等等，其 SQL(Structured Query Language)語法分述如下：

### 1. 篩選查詢

SELECT 欄位名 1, [,欄位名 n]\*

FROM 資料表名稱,...

WHERE 欄位名 1 > 條件 1 [ and | or 欄位名 2 > 條件 2]\*;

你可以列出想要得到的很多個欄位名稱，或是使用 \* (星號)來選取全部的欄位資料。如果你是以欄位名稱下指令，那麼傳回資料是以所列的順序為準。

\* FROM 指定資料來源(資料表)，這部份是一定要鍵入。

\*WHERE 語句 (選項) 判斷各筆資料是否合乎條件，符合條件的筆數才會被傳回。

比較運算子除了>之外，尚有下列幾種：

運算子	說明
=	等於
>	大於
<	小於
>=	大或等於
<=	小或等於
<> or !=	不相同
LIKE	相似(用於字串部份搜尋)

### 2. 新增查詢

INSERT INTO 資料表名稱 (欄位名 1[,欄位名 n]\*) VALUES (值 1[,值 n]\*)  
用來新增一筆資料到資料庫中。

### 3. 更新查詢

UPDATE 資料表名稱

SET 欄位名 1=值 1 [,欄位名 n=值 n]\*

WHERE 欄位名 1 > 條件 1 [ and | or 欄位名 n > 條件 n]\*;

用來修改資料庫內已存在的資料，SET 子句指定要修改的值，WHERE 子句指定符合修改的條件。

### 4. 刪除查詢

DELETE FROM 資料表名稱

WHERE 欄位名 1 > 條件 1 [ and | or 欄位名 n > 條件 n]\*;

用來刪除資料庫內已存在的資料，WHERE 子句指定符合刪除的條件。

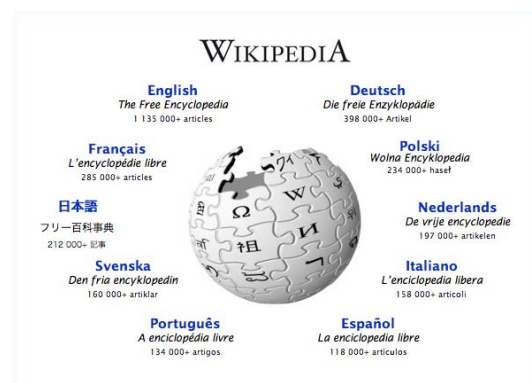
## 五、Wikipedia

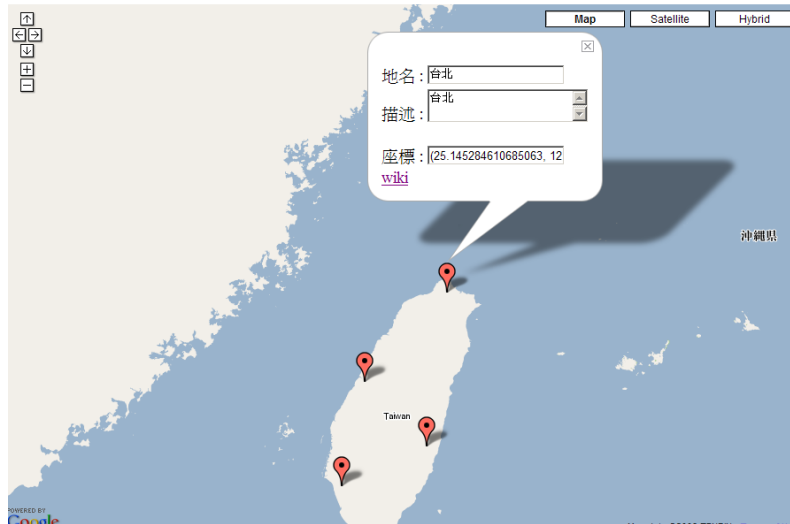
維基百科（Wikipedia）是一個基於 wiki 技術的多語言的百科全書協作計劃，也是一部用不同語言寫成的百科全書，其目標及宗旨是為地球上的每一個人提供自由的百科全書——用他們選擇的語言所書寫的，全世界知識的總和。

維基百科全書，自 2001 年 1 月 15 日正式成立，截至 2006 年 11 月，維基百科條目數第一的英文維基百科[1]已有 150 萬個條目，而所有 250 種語言的版本共超過 570 萬個條目[2]，其中條目數前 15 名的維基百科共佔總條目數的 79%，大部分頁面都可以由任何人使用瀏覽器進行閱覽及修改，維基百科的普及也促成了其它計劃，例如維基新聞、維基教科書等計劃的產生，但是也造成對這些所有人都可以編輯的內容的準確性的爭議。

中文維基百科，自 2002 年 10 月 24 日正式成立，截至 2006 年 11 月 29 日，已擁有 103,018 個條目。

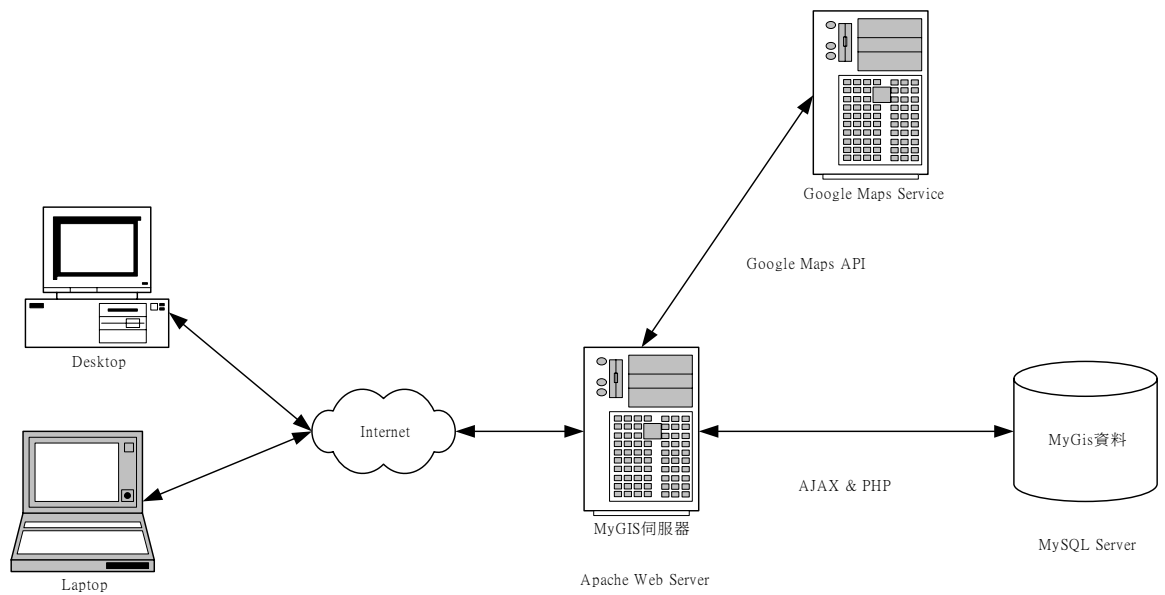
除了本研究參考高國中地理課程所建立的地理資訊外，本研究為提供使用者更多的資訊，將自動提供維基百科全書的連結，如下圖所示。



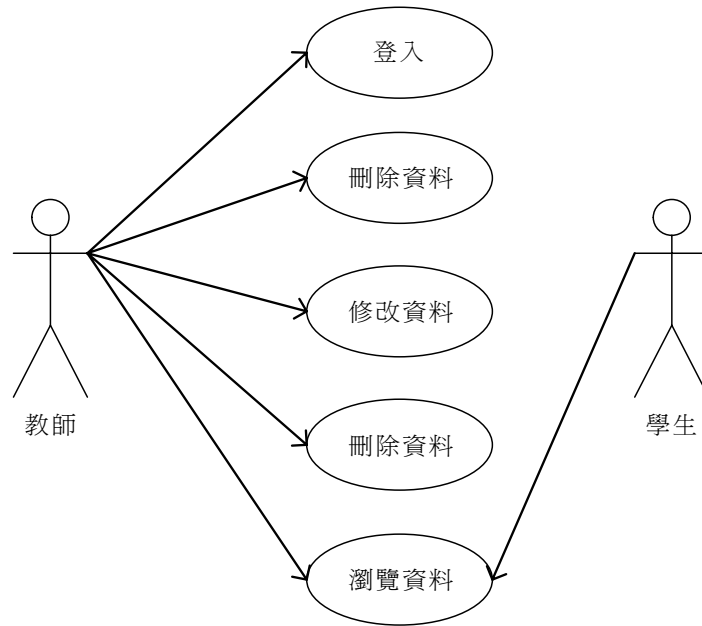


## 參、研究結果與討論

### 1. 系統架構圖

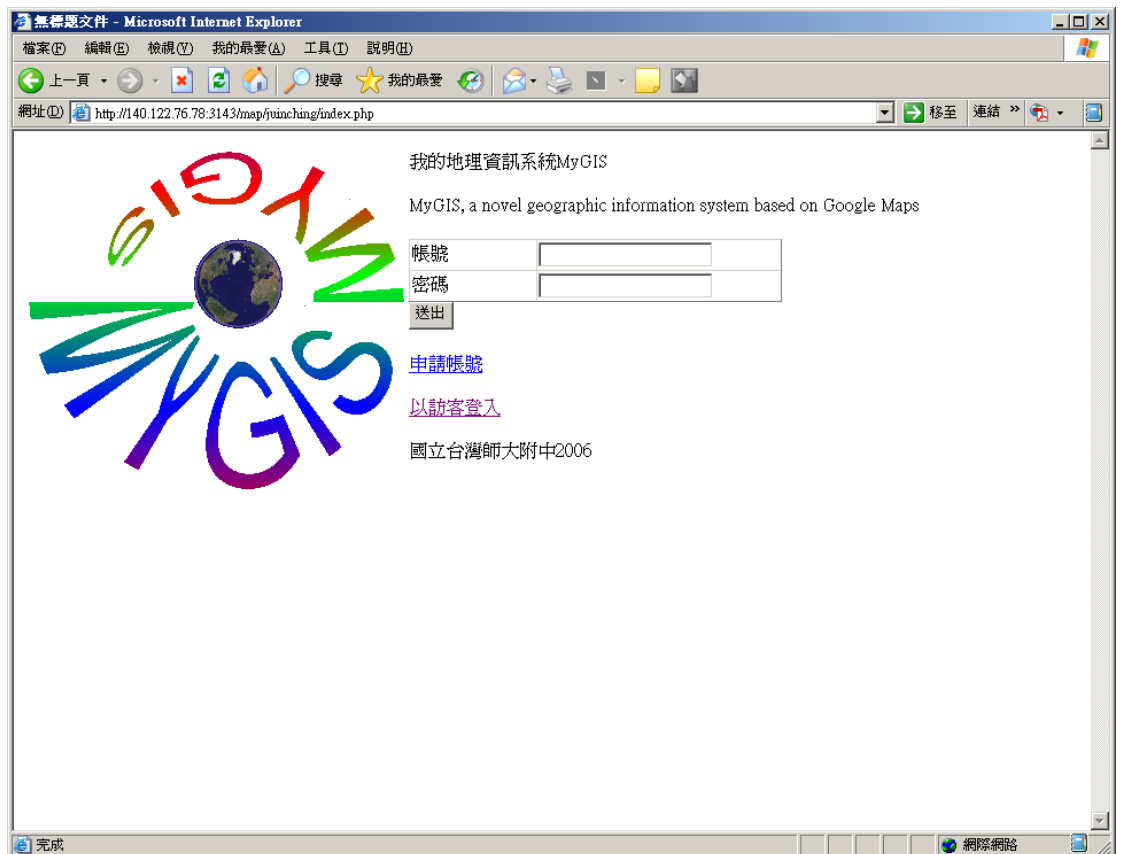


### 2. Use Case

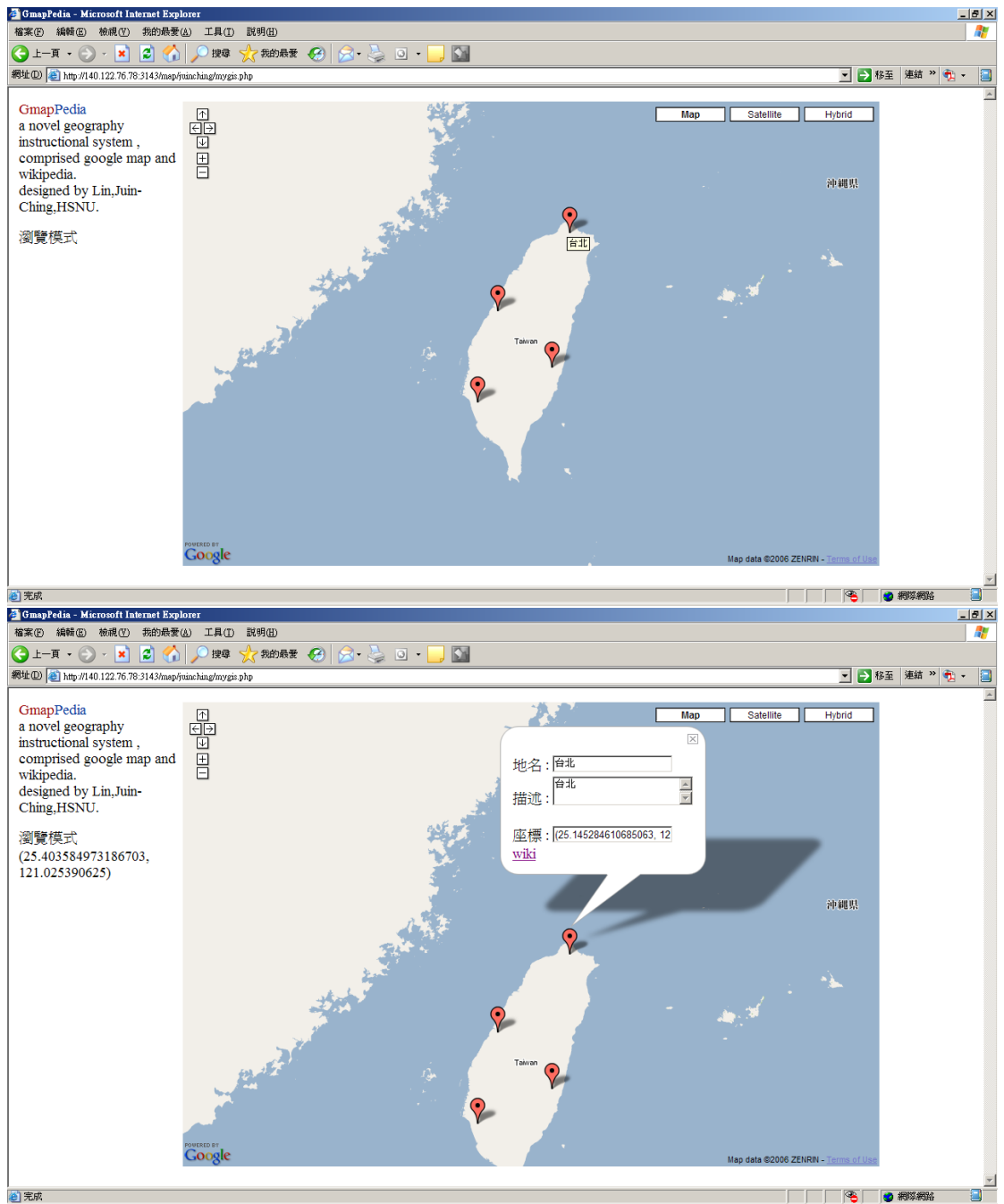


## 一、系統操作

### 1. 首頁

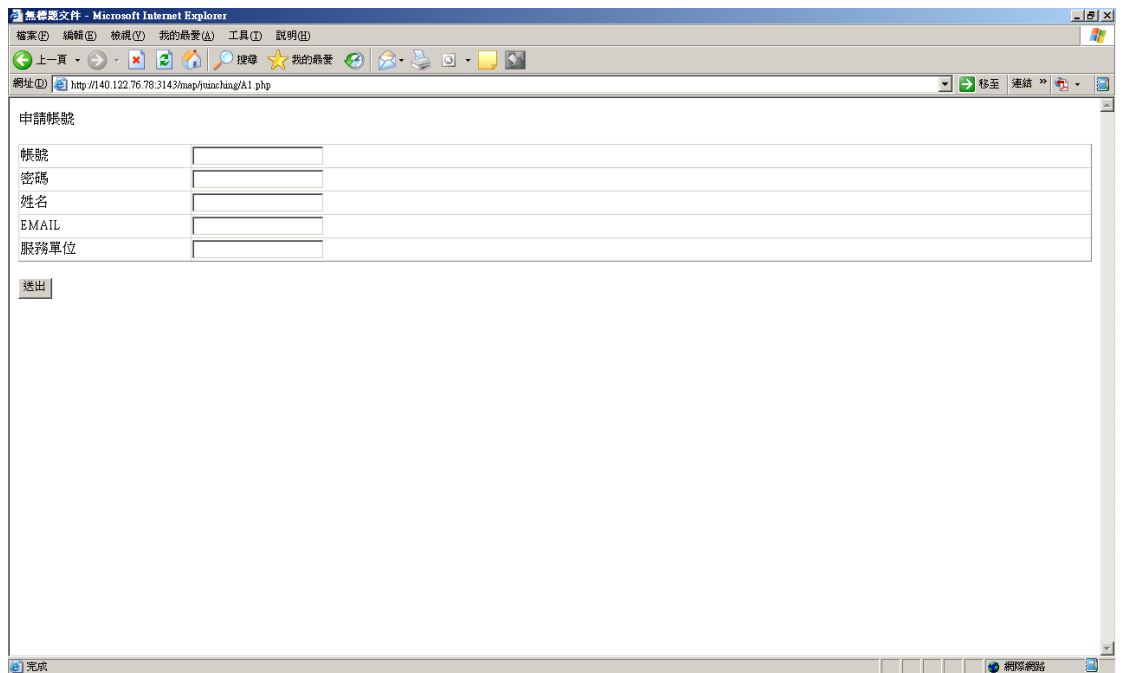


### 1. 學生(一般使用者)



## 2. 教師(編輯者)

- 申請帳號

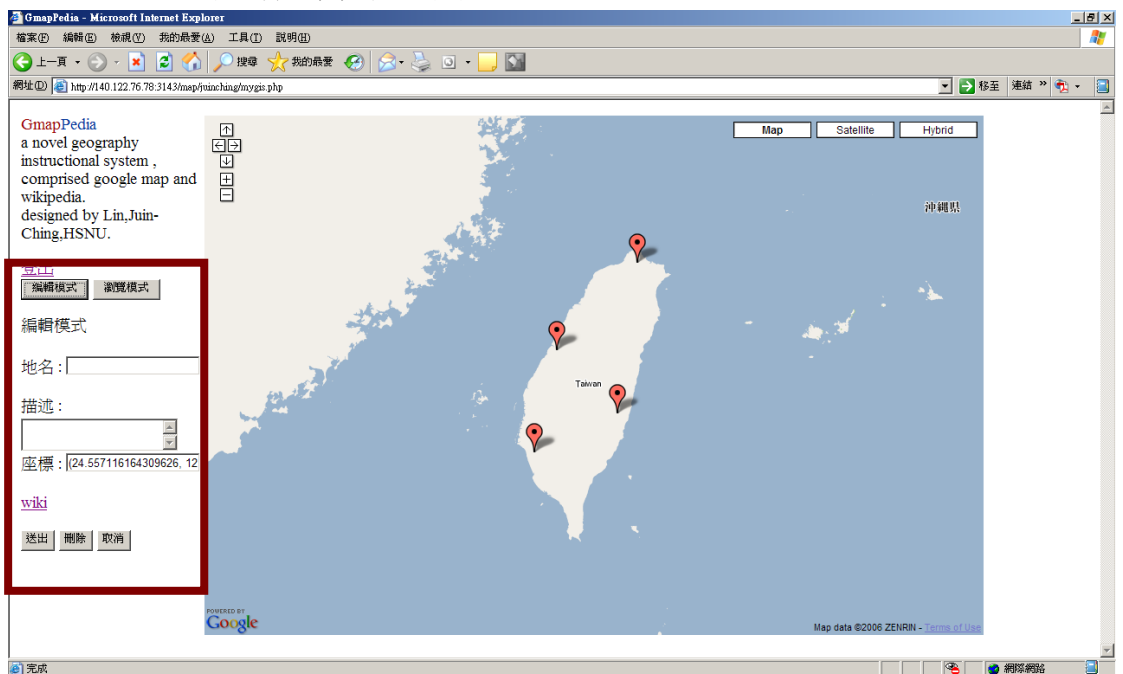


➤ 瀏覽模式：同學生(一般使用者操作) 操作。

➤ 編輯模式：

● 新增

- ✓ 在地圖上以滑鼠左鍵選定要加註地理資訊的地點。
- ✓ 出現資料輸入表單。
- ✓ 填寫表單後按送出即可。



● 修改

- ✓ 在地圖上以滑鼠左鍵選定要修改地理資訊的地點。
- ✓ 出現資料輸入表單。
- ✓ 填寫表單後按修改即可。

- 刪除

- ✓ 在地圖上以滑鼠左鍵選定要刪除地理資訊的地點。
- ✓ 出現資料表單。
- ✓ 直接後按刪除即可。

## 肆、結論與應用

如同維基百科（Wikipedia）所揭示的「海納百川，有容乃大」，本研究以 Apache 網頁伺服器為平台，MySQL 為資料庫，利用 AJAX、PHP 網頁技術，並使用 Google Maps 提供之服務，建立符合 Web 2.0 精神的高中地理資訊共享系統。在此系統下，我們允許教師(編輯者)以合作方式上網提供與課程相關的地理資訊，並可進行資訊修改或刪除，希望集合所有教師之力，提供給學生最豐富、最正確的高中地理資訊系統。

本研究未來發展方向如下：

- 一、加入 wiki engine：wiki engine 具有文件內容版本管控機制，同時支援 RSS，本研究未來擬納入 wiki engine 以取代現有的輸入機制。
- 二、高中地理資訊系統的推廣：推廣此系統到全國各高級中學，集合全國教師之力，建立最豐富、最正確的高中地理資訊系統，提供學生自修最佳之平台。

## 伍、參考文獻

1. Apache software foundation. <http://www.apache.org/>
2. Google Maps API.<http://www.google.com/apis/maps/>.
3. MySQL. [http://linux.tnc.edu.tw/techdoc/mysql/mysql\\_doc/manual\\_toc.html](http://linux.tnc.edu.tw/techdoc/mysql/mysql_doc/manual_toc.html)
4. Wikipedia.<http://zh.wikipedia.org/w/index.php?title=%E9%A6%96%E9%A1%B5&variant=zh-tw>.

## 評語

1. 利用 Google 提供的應用程式，建構「地理」網站方便高中生學習地理。
2. 利用 web2.0 的技術，任何人都可以至系統內涵提供貢獻。
3. 這套系統能夠產生的影響有待觀察，但預期有實用性。
4. 就創意及科學的面向而言不是十分明顯。