

# AlumniBook G3

郝昀彥 R00725051 郭瀚智 R02725023 鄭立民 R02725041 李奕德 B99705021  
張凱涵 B00705027 施淮振 B00705047 倪嘉銘 T02705102

---

## Contents

1	專案背景及目的	1
2	專案假設與限制	2
3	專案範圍	2
3.1	功能性需求	2
3.1.1	登入模組	2
3.1.2	個人資料管理模組	3
3.1.3	搜尋模組	3
3.1.4	關係建立模組	3
3.2	非功能性需求	3
3.2.1	資料可變性	3
3.2.2	系統穩定性	3
3.2.3	系統安全性	4
3.2.4	使用者期望	4
4	解決方案	4
4.1	系統架構	4
4.2	使用套裝軟體及開發工具、程式語言	6
4.3	系統環境需求	6

## 1 專案背景及目的

本系在發展茁壯的同時培養了許多傑出的人才，這些人散布在各個不同領域。在超級專業化的現代若能連結正確領域的人來提供系上一些建議與幫助，必能讓資管系所內的課程內容與訓練更加的紮實。所以除了學生間的相互聯絡，本系統也將會提供系友與教授及行政人員進入本系統，對其所擅長之專業領域提供協助。學生亦可藉由本系統盡早深入了解產業現況，獲得更多正確且即時的資訊並對未來發展方向有更確實精確的判斷。

為了確保使用者身分得正確性，本系統必須要求使用者經由計算機中心 (或其他認證機構) 所提供的身分認證服務來認證使用者和其所宣稱的人為同一個人以防冒用。經正確認證之使用者於進入本系統後，會來到共用之瀏覽區。在這個區塊可以看到目前系統中的所有議題，瀏覽並參與討論。在觀看議題的同時使用者也可針對特定議題發表評論，此評論將會由任何觀看此議題的人共同看見。

若在閱讀討論區時對特定人士產生興趣，也可以在適當的隱私權相關管理下查詢特定的系友的相關資訊。能否能夠看見該使用者所有相關資料取決於資料擁有者的個人隱私條件以及搜尋者之帳號類別，或是搜尋者與被搜尋者之間有在本系統中建立特別的朋友關係。本系統中若存在朋友關係則再搜尋時會由特定對象獨立於依照帳號類別做設計的隱私規則，直接套用該使用者針對朋友設定的隱私權相關設定，讓朋友間能夠更加深入的了解彼此。

除了瀏覽討論區外，使用者亦可編輯自身的相關資料與隱私相關選項來讓自己被更多人看見。本系統將提供預先填入之公開資訊、基礎的個人連絡資訊輸入、以及與職業生涯相關之記錄，三個主要區塊。使用者可依照自身之習慣與想法選擇性的填入資料與決定其公開性。預設的群組別為校友、學生、教授及行政人員、與院系人員及朋友幾種，未來可再依據使用者需求做調整。

本系統旨在於建立一個能夠有效凝聚資管系所內之相關人士，提供一個跨界跨部 (研究所與大學部) 之溝通平台。本系統將階段性提供完整的溝通整合服務，冀望成為連結具專業知識的人幫忙解答疑惑，也希望能夠成為校友找人、學生找事之合作橋梁。

## 2 專案假設與限制

為使本計畫順暢運行，其執行環境必須具有以下幾點之限制，其中缺乏任一一項將造成時程延宕或效果不如預期等等缺陷。

- 身分認證  
關於身分認證部分，台大計算機中心必須能夠提供 single sign-on 之服務，並且能夠回應本系統所有身分認證之需求。同時，回傳之 token 必須包含學號等資訊以供系統自動填入使用者帳號詳細內容使用。
- 硬體  
台大資管系必須能夠提供用於最後架站所需之伺服器的硬體設備或虛擬伺服器。
- 帳戶資料  
台大資管系必須提供預設帳戶之資料，以供建立初始使用者名單之用。或系所應能提供學號之判別方式來認證所有正常的用戶，例如其學號範圍、數字上的特徵與限制等等。
- 使用者習慣討論  
本系統需能夠對隸屬於每個不同預設分類項下之使用者做指定形式之訪談，以便了解每個職位之使用者對本系統之期望與意見。此意見將影響預設帳號類別之新增與刪去、朋友關係的用途與系統未來發展之用。
- 階段性開放  
本計畫將以階段性的方式向終點努力，期間所釋出的測試系統並不一定與完成系統之表現、內容完全相同。會依照以下順序依序開放測試：
  - 使用者帳號管理
  - 討論區測試
  - 完整上線系統
- 使用者範圍  
本系統預設之使用者範圍為校友、學生、教授及行政人員。超出本規畫範圍者將必須等到完整系統上線後發現有必要才會以更新版本的新增。
- 行動裝置  
本系統雖有預設 API 開放給未來發展之非特定行動裝置系統中的程式，但是並不包含強制性的程式開發，將再行有餘力之時完成此部分。

## 3 專案範圍

### 3.1 功能性需求

在功能性需求的部分本系統應包含登入模組、個人資料管理模組、討論區模組、搜尋模組、關係建立模組以及行動裝置模組。全部功能性需求條列如表 1 (Table 1)。

#### 3.1.1 登入模組

在登入模組中需包含以下幾點功能:

- 宣稱身分
- 身分認證
- 取得回傳之 token 並解構

在宣稱身分的地方必須由使用者宣稱自己的身分並提供相關認證資料，在經由計算機中心所提供之 single sign-on 之服務完成身分認證之後，回傳給系統一組經認證的特定格式之 Token。若使用者為初次登入本系統，則此 Token 在經系統解構之後必須能夠抽取部分必要資料填入系統之資料庫中做為使用者個基礎資訊。此行為將於前端網頁系統中完成，若有需要填入時則會交由後端資料系統負責。

功能模組	功能需求	功能範圍
登入模組	輸入身分 身分認證 取得 token	讓使用者宣稱自己的身分 透過計中做身分認證 回收計中回傳的 token
個人資料管理模組	管理公開資料 管理個人聯絡資料 職業生涯資料管理 隱私權管理 關係管理 儲存資料異動	讓使用者維護帳號的公開資訊 讓使用者維護帳號的聯絡資料 讓使用者維護職業生涯相關資料 讓使用者管理帳號之公開性選項 讓使用者管理與其他帳號之間的關係 儲存任何使用者新增/修改/刪除之資料
搜尋模組	搜尋特定使用者 回傳指定使用者	依照指定學號搜尋符合帳號 依照指定學號回傳符合條件之帳號
關係建立模組	傳送使用者好友要求 記錄使用者對於要求之回應 記錄使用者與其他用戶之友好狀況	由被要求者決定是否成為該用戶之朋友 提供被要求者回應交友要求 儲存所有與該使用者為好友關係的連結

Table 1: 功能性需求列表

### 3.1.2 個人資料管理模組

個人資料管理模組將負責管理每個使用者各自的資料組，此資料組中包含使用者的公開資料、個人連絡資料、職業生涯相關資料與隱私權管理及關係管理。使用者必須能夠看見目前系統內的資料狀況並且能夠更動大部分的資料，其中如學號等等的則必須要經過系統管理者之認證後方可變更。此更改之行為將於前端網頁系統中完成，若有資料必須儲存則會交由後段資料系統負責。

### 3.1.3 搜尋模組

本系統為了建立一個適合使用者相互溝通之平台，必須提供使用者方便互相連結之功能，搜尋模組正是為了此需求所建立。當使用者在討論區中產生對特定用戶之好奇心，便可藉由本模組使用特定資料 (如: 學號) 做為索引找到該用戶之公開資訊。公開程度將依該用戶之隱私設定做限制，若該用戶認為預設之帳號類別方式無法達成所欲之特殊公開之組合則需透過朋友關係的建立來達成要求。互相成為朋友的兩人將以特別規則優於普通規則的方式優先採用該用戶所設定的特別規則觀看公開資訊。本搜尋功能將全部於前端網頁系統完成。

### 3.1.4 關係建立模組

本系統中基於完成使用者之特定隱私權設計之理念，將在系統中提供建立好友關係的功能，也就是本模組的核心。本模組必須要能傳遞使用者教朋友的邀請、記錄使用者對於要求的回應與記錄使用者與其他用戶之關係。

## 3.2 非功能性需求

非功能性需求的部分主要牽涉到關於資料未來可能的彈性、系統運作的穩定性、系統元件的重複利用可能、以及資訊安全等。

### 3.2.1 資料可變性

針對例如說使用者種類或是文章類型等等資訊，設計出未來較容易擴充之資料儲存模型。

### 3.2.2 系統穩定性

硬體方面，需配置不斷電系統以及資料自動備份系統以維持系統不斷線的能力。另一方面，也需要設計具有平衡流量能力的伺服器架構，以有效降平均回應時間。

### 3.2.3 系統安全性

軟體方面，為防止已知的幾種攻擊方式，本系統在設計時必須確保使用者不會再不恰當的地方使用”“等特殊符號。包含 SQL injection 在內的幾種攻擊方式都必須要在系統設計時避免。

### 3.2.4 使用者期望

為確保本系統朝向正確且易學易用的方向發展，在發展過程中將不定期的針對特定使用者族群做抽樣調查。取得之資料將在適當的分析之後，用於增進系統效能與新增功能或是修改錯誤等等面向。

## 4 解決方案

### 4.1 系統架構

依照先前文件之功能描述，我們試圖以 USE CASE 圖描述各種使用者與系統的互動關係，以及系統功能的連結關係。

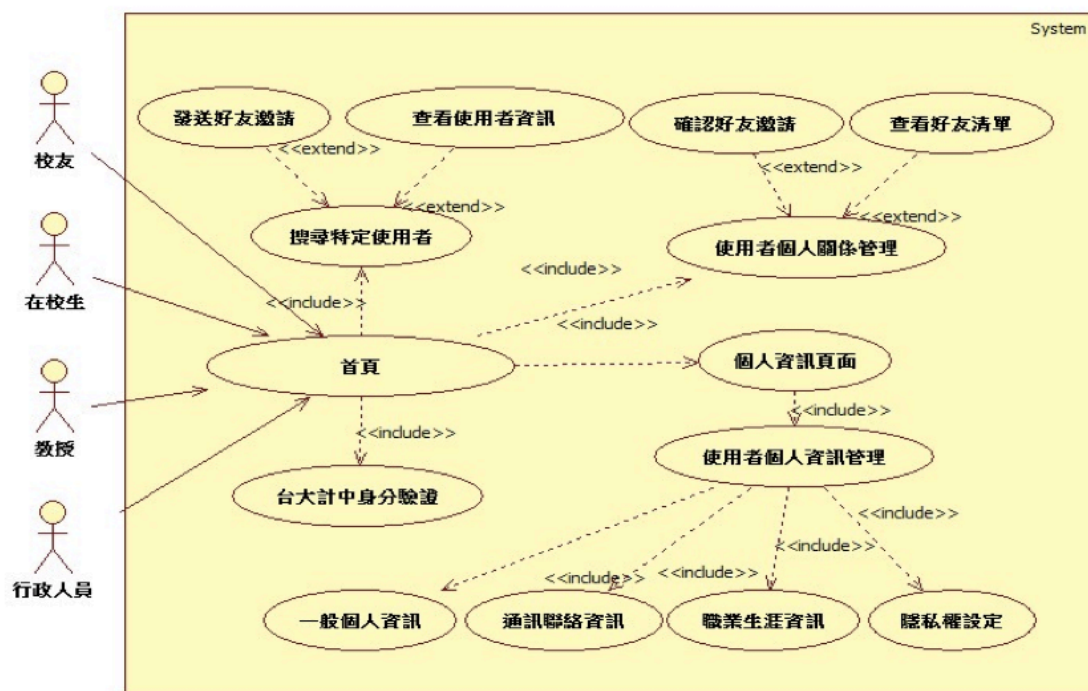


Figure 1: Use Case Diagram (使用者資訊管理)

然而目前有關好友相關的功能 (也就是關係管理、確認好友邀請、查看好友清單、發送好友邀請，這四個 USE CASE) 尚未規劃與實作。所以以下只做其他 USE CASE 之 Sequence Diagram。

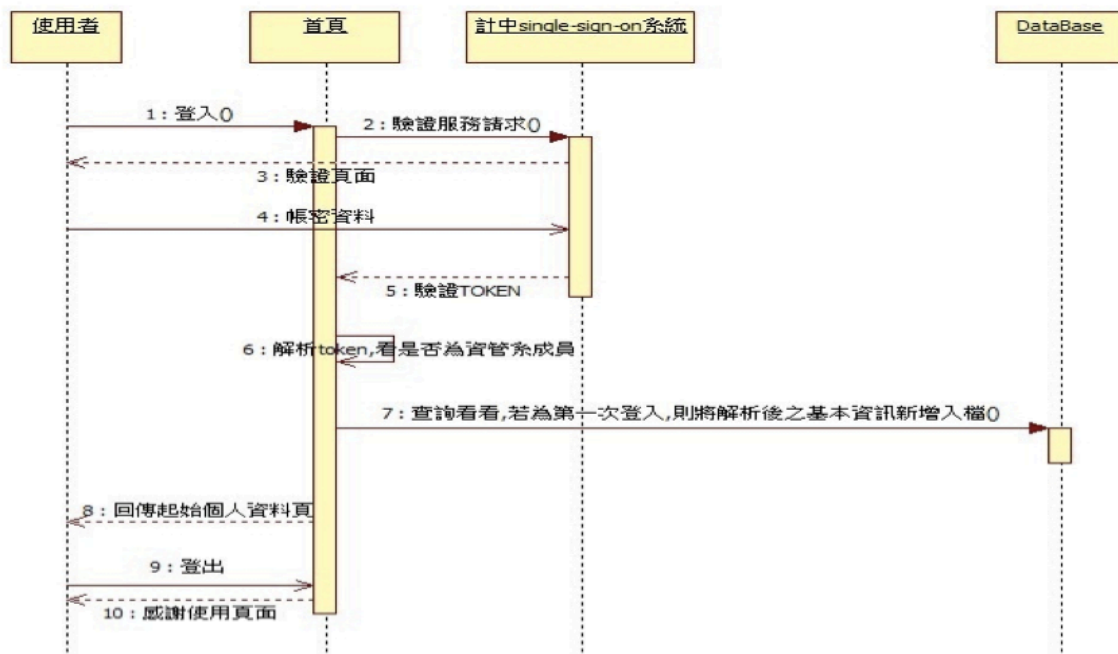


Figure 2: Sequence Diagram (首頁登入)

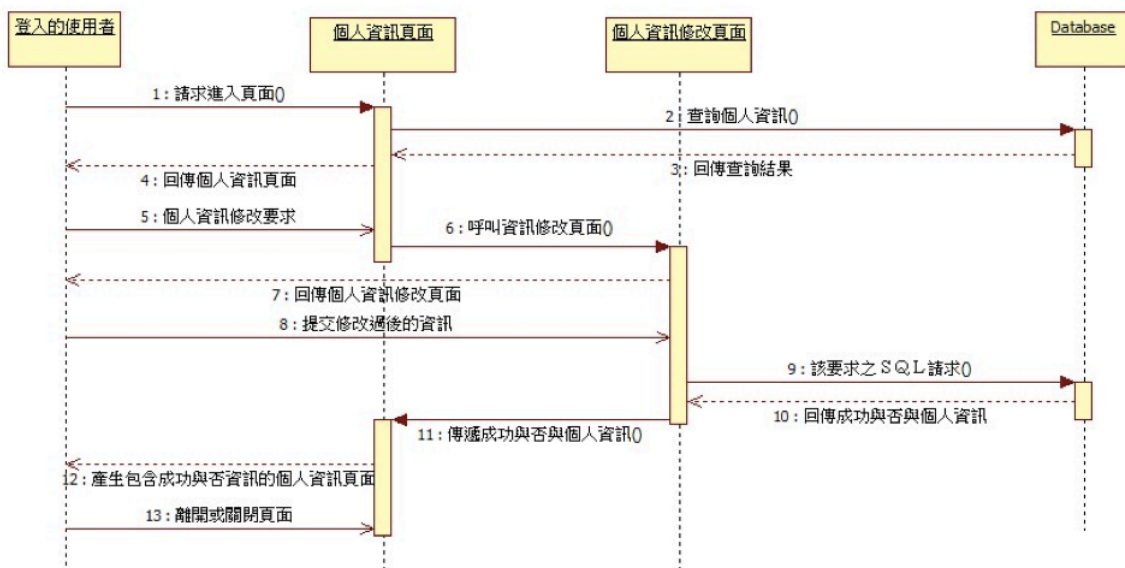


Figure 3: Sequence Diagram (個人資訊管理與個人資訊修改)

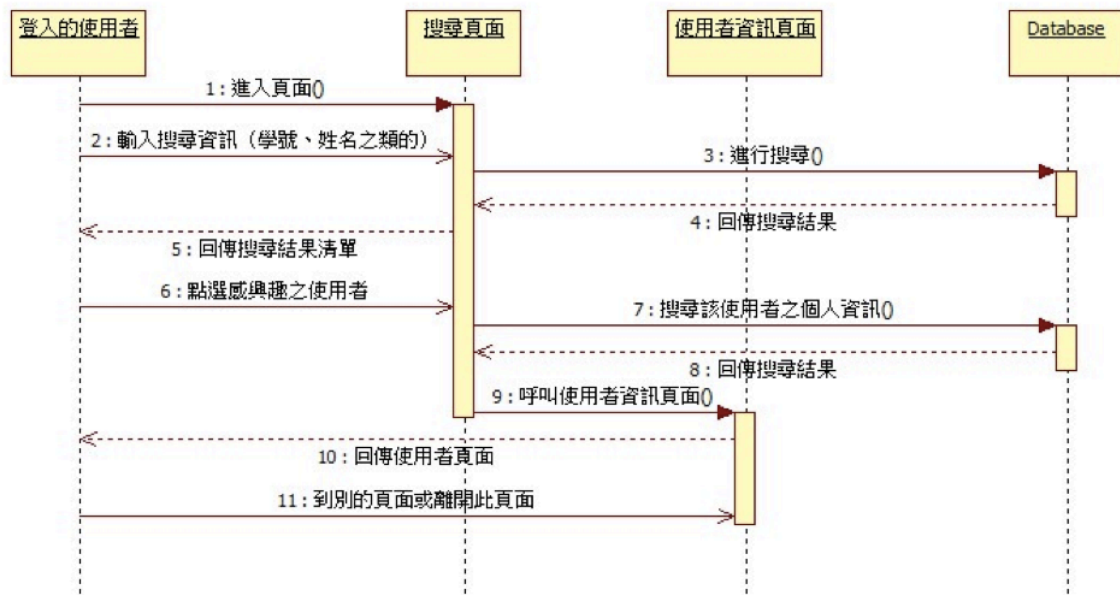


Figure 4: Sequence Diagram (搜尋查看他人使用者資訊)

## 4.2 使用套裝軟體及開發工具、程式語言

本系統採用 Ruby on Rails 作為主要伺服器端開發架構，使用 Ruby 2.0.0-p247 版本，Rails 4.0.0 版本，以 Grape 這個 Ruby REST-like API micro-framework 建立起主要 API server，提供網頁端與手機 JSON API，連結 MySQL(5.5.29) 資料庫儲存索取更新並維持資料正確與一致性，前端網頁頁面以 HTML5(HTML, CSS, JavaScript) 撰寫而成，採用 Angular.js(1.2.0) 一個前端 JavaScript MVC 架構增加程式碼模組化與維護性，使用介面以 Bootstrap(3.0.0) 作為主要網頁前端外觀元件框架，加速雛形開發。

## 4.3 系統環境需求

目前 Ruby on Rails Server 架設在 Heroku 一個雲端應用平台，提供高度延展性，以亞馬遜 AWS EC2 為基礎建構而成的伺服器叢集，使用 git 上傳即可快速部屬，伺服器主要需要運行在 Ruby 2.0.0 與 Rails4.0.0 環境下。網頁前端頁面則放置在 Http server (如 Apache, Nginx) 下，並與 MySQL 資料庫放在資管系主機。前端網頁則相容於多數瀏覽器 (Chrome, Safari, Firefox...)，提供使用者多樣的使用選擇。