國立中央大學

資訊管理學系

系統分析與設計

系統軟體分析規格書

**央央知識共享系統**

KnowledgeNCU

第三組

資管三A 110403023 林均翰

資管三B 110403043 石昀翰

資管三A 110403025 朱博脩

資管三A 110403013 王釋緯

資管三B 110403046 管奕凱

指導教授：許智誠教授

[**第2章 資料庫設計**](#_fdan82pquh8u) **3**

[1. 會員資料表(member):](#_x9a3ai932npr) 4

[2. 貼文資料表(post):](#_cswosjo2qnup) 5

[3. 檔案資料表(file):](#_yylx702vbbr) 6

[4. 關注資料表(follow):](#_yx8jqp1r5u4l) 6

[5. 留言資料表(comment):](#_y57foc9be7s) 7

[6. 通知資料表(notification):](#_bsmtkusdi4i6) 8

[7. 按讚資料表(Like):](#_dh13hyc64rys) 8

[**第3章 類別圖**](#_37lz55wybwth) **9**

[**第4章 系統循序圖**](#_rqdygeqcv4xg) **13**

[4.1 使用案例圖](#_kq0h71e648vx) 13

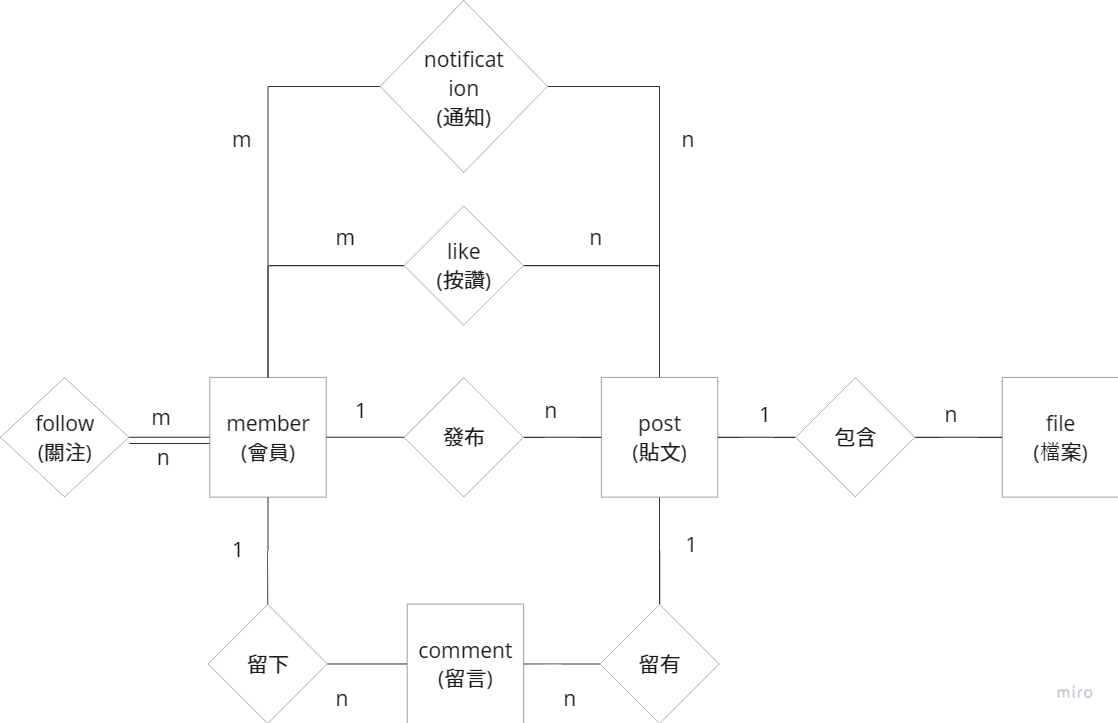
[4.2 Use Case實作之循序圖](#_ng9mpivcsktv) 14

[4.2.1 商業流程編號 1.0: 會員模組](#_gbjvbq7gipch) 14

[4.2.2 商業流程編號 2.0: 管理員模組](#_ec62xkqbw2yc) 15

# **第2章 資料庫設計**

設計階段之資料庫，根據分析文件之實體關係圖（Entity-Relation Diagram），進行確認並依據其規劃資料庫之資料表，共計包含4個實體（Entity）、7個關係（Relationship）、0個複合性實體（Compound Entity），下圖為設計階段之ER圖，亦可使用資料庫綱要圖（Schema Diagram）進行取代：



根據上圖進行資料表之設計，以下將逐一說明資料庫每張資料表之欄位:

## 會員資料表(member):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | 名稱 | 類型 | 預設值 | 自動增加 | 空值 | 編碼 |
| P.K | member\_id | Int | 無 | V | 否 |  |
|  | member\_name | Varchar(45) | 無 |  | 否 | utf8\_general\_ci |
|  | member\_account | Varchar(45) | 無 |  | 否 | utf8\_general\_ci |
|  | hash\_pwd | Varchar(100) | 無 |  |  | utf8\_general\_ci |
|  | salt | Varchar(45) | 無 |  | 否 | utf8\_general\_ci |
|  | member\_bio | Varchar(400) | 無 |  |  | utf8\_general\_ci |
|  | is\_admin | Int | 0 |  | 否 |  |
|  | member\_img\_path | Varchar(100) | 無 |  | 否 | utf8\_general\_ci |
|  | join\_time | datetime | 無 |  | 否 |  |

✔ member\_id: 為自動增加作為會員編號，不可更動由資料庫系統自動產生

✔member\_name: 會員的名稱

✔member\_account: 會員的帳號

✔salt: 隨機產生的字串穿插於密碼間，雜湊後增加密碼的安全性

✔hash\_pwd: 加上salt雜湊後的密碼

✔member\_bio: 使用者的個人介紹

✔is\_admin: 判斷使用者是否具有管理員權限

✔member\_img\_path: 用於紀錄使用者大頭照圖檔之路徑位置

✔join\_time: 會員註冊的時間日期

## 貼文資料表(post):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | 名稱 | 類型 | 預設值 | 自動增加 | 空值 | 編碼 |
| P.K | post\_id | Int | 無 | V | 否 |  |
|  | post\_title | Varchar(45) | 無 |  | 否 | utf8\_general\_ci |
|  | post\_type | Enum(‘post\_note’,’post\_course\_material’,’post\_feedback’) | 無 |  | 否 |  |
|  | post\_description | Varchar(800) | 無 |  | 否 | utf8\_general\_ci |
|  | post\_time | Datetime | 無 |  | 否 |  |
|  | edit\_time | Datetime | 無 |  |  |  |
| F.K | member\_id | Int | 無 |  | 否 |  |

✔post\_id: 為自動增加作為貼文編號，不可更動由資料庫系統自動產生

✔post\_title: 用於紀錄貼文的名稱

✔post\_type: 用於紀錄貼文的類型，有筆記(post\_note)、教材(psot\_course\_material)和課程評價(post\_feedback)3種

✔post\_decription: 用於紀錄貼文的介紹

✔post\_time: 用來記錄貼文上傳時間點

✔edit\_time: 資料上次被修改的時間

✔member\_id: 用來記錄貼文屬於哪個會員id，為外來鍵(FK)，對應到Member資料表的member\_id(PK)

## 檔案資料表(file):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | 名稱 | 類型 | 預設值 | 自動增加 | 空值 | 編碼 |
| P.K | file\_id | Int | 無 | V | 否 |  |
|  | file\_path | Varchar(100) | 無 |  | 否 | utf8\_general\_ci |
|  | post\_time | Datetime | 無 |  | 否 |  |
| F.K | post\_id | Int | 無 |  | 否 |  |
| F.K | member\_id | Int | 無 |  | 否 |  |

✔ file\_id: 為自動增加的貼文的附件檔案id，不可更動由資料庫系統自動產生

✔ file\_path: 附件檔案之路徑位置

✔ post\_id: 為外來鍵(FK)，對應到post資料表的post\_id(PK)

✔ post\_time: 用來記錄貼文甚麼時候上傳的，為外來鍵(FK)，對應到Post資料表的post\_time (PK)

✔ member\_id: 來記錄貼檔案屬於哪個會員id，為外來鍵(FK)，對應到Member資料表的member\_id(PK)

## 關鍵字資料表(keyword):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | 名稱 | 類型 | 預設值 | 自動增加 | 空值 | 編碼 |
| P.K | keyword\_id | Int | 無 |  | 否 |  |
|  | word | Varchar(60) | 無 |  | 否 |  |
| F.K | post\_id | Int | 無 |  | 否 |  |

✔ keyword\_id: 用來記錄此會員id，為外來鍵(FK)，對應到Member資料表的member\_id(PK)

✔ word: 用來記錄資料的關鍵字，方便搜尋

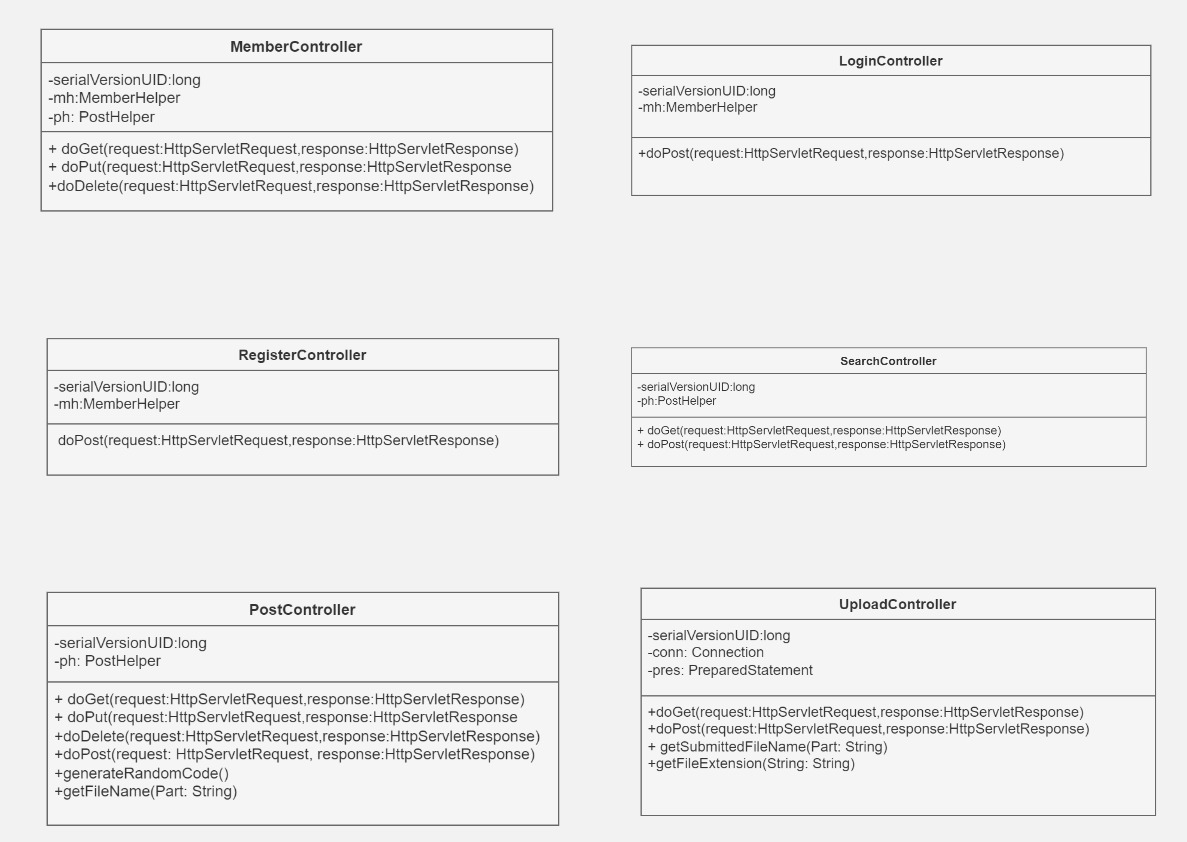
✔ post\_id: 為外來鍵(FK)，對應到post資料表的post\_id(PK)

# 

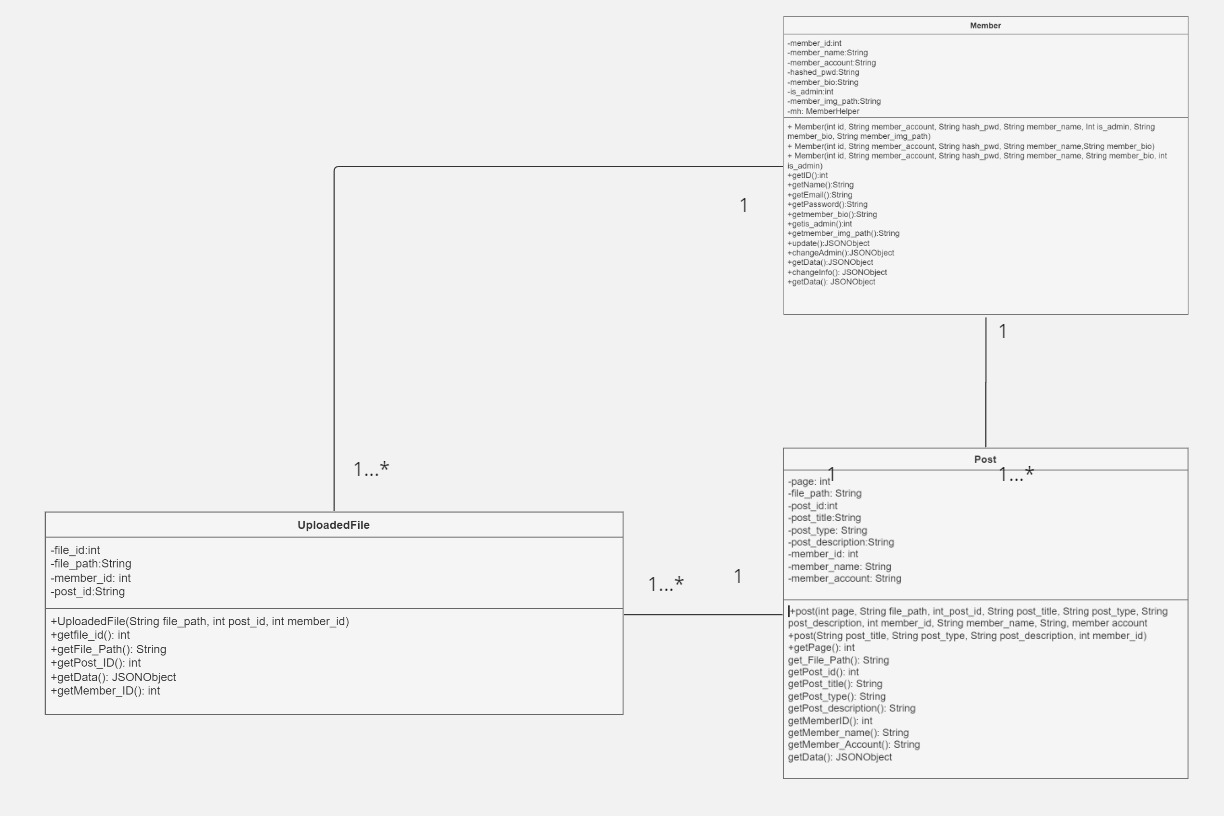
# **第3章 類別圖**

下列圖片依據知識共享平台的分析模型和建立的互動圖，以及實體關係圖(Entity-Relation Diagram) 所設計之類別圖 (Class Diagram) 用於描述系統的類別集合，包含其中之屬性，與類別之間的關係。

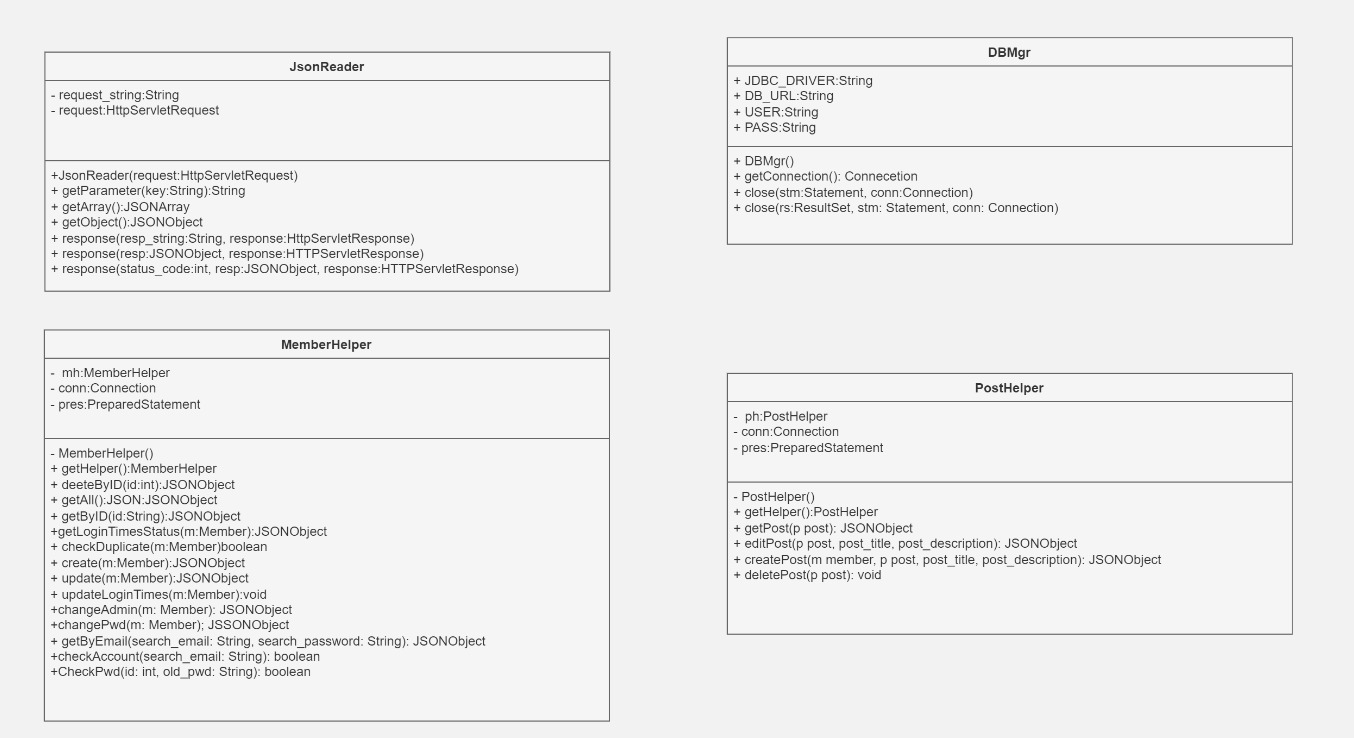
本階段之類別圖屬於細部 (detail) 之設計圖，說明如下列: 類別圖除包含與資料庫相對應知物件外，亦包含相關物件之控制物件 (controller) 、DBMgr與各功能相對應資料庫操作類別(例如: MemberHelper) 和相對應之類別工具 (JsonReader)。



類別圖（1/3）



類別圖（2/3）



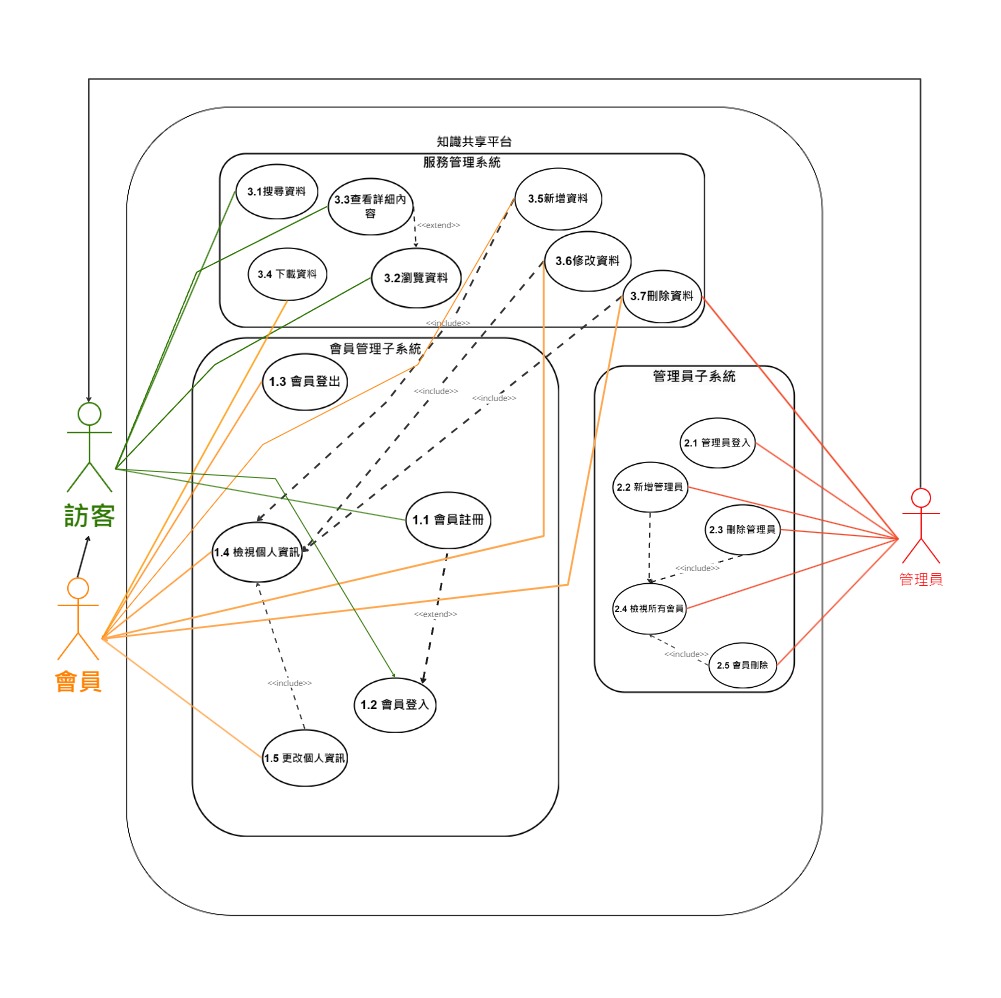
類別圖（3/3）

# **第4章 系統循序圖**

本章節主要依照第一份文件需求所產生之使用案例 (use case) 與第二份文件分析之邏輯階段活動圖與強韌圖為基礎，進行設計階段之循序圖設計，將每個使用案例進行闡述。於此階段，需要有明確之類別 (class) 名稱與呼叫之方法 (method) 與傳入之變數名稱與型態等細部設計之內容。

## **4.1 使用案例圖**

本章節主要依據第一份文件針對專案之需求進行確定，本知識共享系統預計共有 3 位動作者與 20 個使用案例，並依照不同之模組區分成不同子系統共計四'個子系統，其中包含以下：會員管理子系統、管理員子系統、服務管理系統、會員通知子系統。下圖為本系統之使用案例圖：



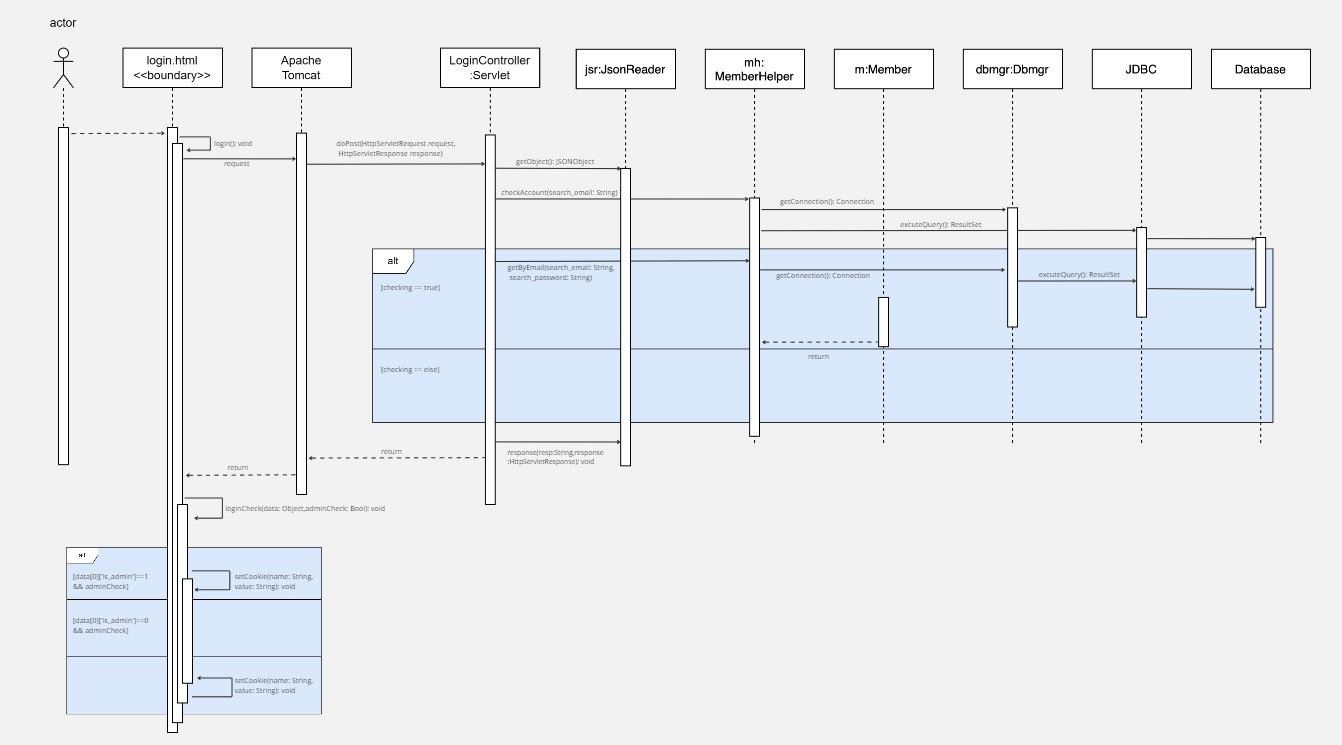
## **4.2 Use Case實作之循序圖**

### **4.2.1 商業流程編號 1.0: 會員模組**

在Use Case 1.0 中所使用到的功能(包含: 會員註冊、會員登入、會員登出等)，以下簡要說明Use Case 1.1 會員登入的功能:

* **Use Case 1.1: 會員登入**

此使用案例說明訪客在在已註冊帳號的情況下，藉由輸入所需欄位(帳號、密碼)來登入會員的帳號，除了輸入所需欄位，需勾選是否以管理員的身分登入 (若無管理員權限且勾選則會顯示登入失敗)，登入的會員比訪客擁有更高的權限，如上傳和下載資料等。

****

1. 訪客完成「1.1 會員註冊」後會自動跳轉到登入頁面(login.html)。
2. 會員輸入帳號、密碼、勾選是否以管理員身分登入並按下「登入」按鈕後，JavaScript裡的login()會送出POST請求。
3. 後端用LoginController的doPost()處理，並以JsonReader讀取Json資料。
4. 使用MemberHelper的checkAccount()檢查帳號是否存在。
5. 用MemeberHelper的getbyemail()取得會員資料放在新創的Member物件，封裝並回傳。
6. 登入成功後跳轉到網站首頁(index.html)

* 回傳有三種情況:

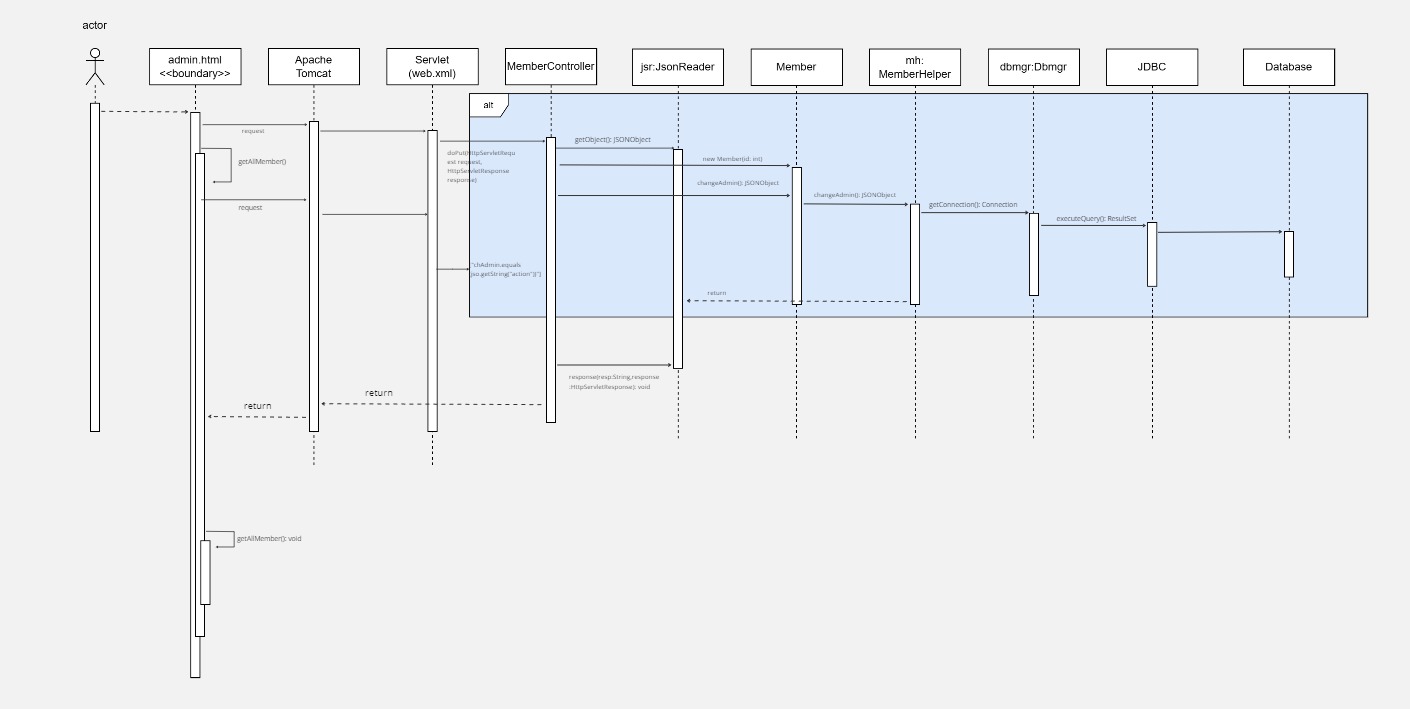
|  |  |  |
| --- | --- | --- |
| message | status code | detail |
| Login\_Successful | 200 | 登入成功 |
| No\_Such\_Account | 403 | 此帳號未註冊 |
| Wrong\_password | 414 | 有此帳號但密碼錯誤 |

### **4.2.2 商業流程編號 2.0: 管理員模組**

在Use Case 2.0 中所使用到的功能(包含: 管理員登入、新增管理員、刪除管理員等)，以下簡要說明**Use Case 2.2 新增管理員**的功能:

* **Use Case 2.2: 新增管理員:**

此使用案例說明管理員在完成「2.4 檢視所有會員」後才可執行此使用案例，管理員可以透過拉動開關來新增會員的管理員權限。



1. 訪客完成「2.1 管理員登入」和「2.4檢視所有會員」後會跳轉到所有會員頁面(admin.html)。
2. 管理員拉動會員的管理員權限後，JavaScript裡的changeAdmin()會送出PUT請求。
3. 後端用MemberController的doPut()處理，並以JsonReader讀取Json資料。
4. 確認前端request的message符合後創一個Memeber物件，並使用MemberHelper的changeAdmin()更改會員的管理員權限。
5. 進行封裝並回傳資料，呼叫JavaScript的getAllMember方法(自動更新顯示的會員資料)後留在所有會員頁面(admin.html)

* 回傳成功:

|  |  |  |
| --- | --- | --- |
| message | status code | detail |
| change\_Successful | 200 | 成功更改會員身分 |