

# 技能交換平台

## 設計文件

專案名稱	Exchange
撰寫日期	2017/1/15
發展者	吳宇鴻、鍾子健、施博文、蔡昌廷、羅祐任

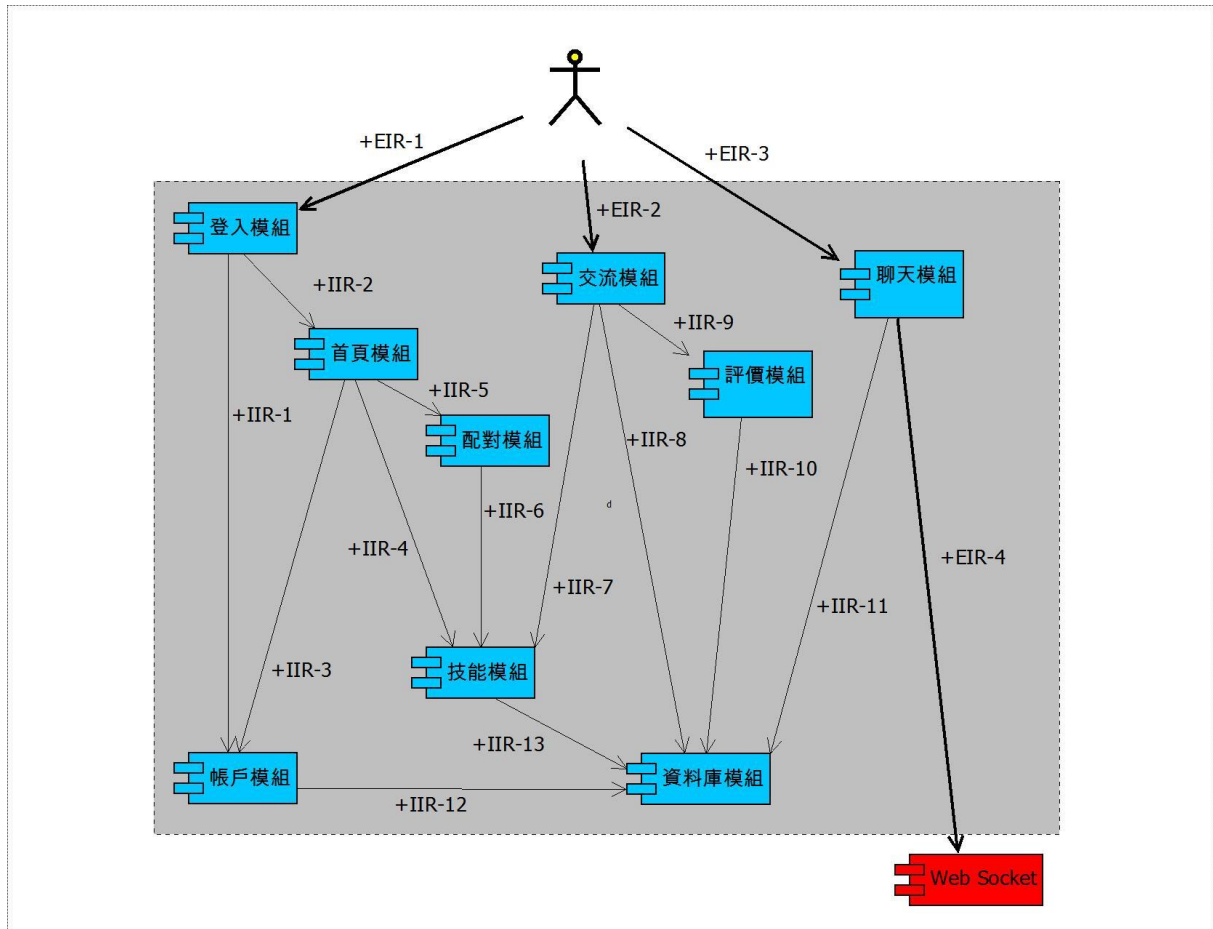
## 版次變更記錄

版次	變更項目	變更日期
1.0	(初版) 資料庫、類別圖初步規劃。	2016/11/18
1.1	系統模型與架、介面需求與設計、流程設計、使用者畫面設計、類別圖設計、資料細部設計、設計議題。	2016/11/25
1.2	各項細部文件內容調整。	2016/11/29
1.3	系統模型圖示調整。 實作技術章節新增。 流程設計：黑名單流程。 設計議題補充：演算法之定義	2016/12/16
1.4	最後修正。	2017/01/15

## 目錄 (Table of Contents)

版次變更記錄.....	2
1. 系統模型與架構(SYSTEM MODEL/SYSTEM ARCHITECTURE) .....	4
2. 介面需求與設計(INTERFACE REQUIREMENT AND DESIGN).....	5
3. 流程設計(PROCESS DESIGN) .....	11
4. 使用者畫面設計(USER INTERFACE DESIGN).....	14
5. 資料細部設計(DATA DESIGN) .....	24
6. 類別圖設計(CLASS DIAGRAM).....	35
7. 實作技術(IMPLEMENTATION LANGUAGES AND PLATFORMS) .....	52
8. 設計議題(DESIGN ISSUE) .....	53

## 1. 系統模型與架構(System Model/System Architecture)



本系統之模型如圖所示。虛線區塊表示系統邊界(System Boundary)，一般實線箭頭表示內部介面(Internal Interface)，粗體實線箭頭則表示外部介面(External Interface)。本系統將系統分為登入模組、首頁模組、帳戶模組、配對模組、技能模組、交流模組、評價模組、資料庫模組、聊天模組等九大模組，其中聊天模組將使用外部模組 Web Socket 完成訊息的傳遞。使用者概念上將直接使用登入模組、交流模組與聊天模組，其餘各項系統子功能皆規劃於此三個模組的下游，但是使用者實際操作上仍然是實際使用本系統的各項子功能。

## 2. 介面需求與設計(Interface Requirement and Design)

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
EIR-1	使用者與登入模組之介面	登入模組	使用者
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
HTTP GET		登入/註冊資訊	資料正確性判斷(bool)
對應之介面需求(Interface Requirement Description)			
用戶端輸入登入資訊或註冊資訊，並由登入模組來驗證登資料合法性及正確性。			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
EIR-2	使用者與交流模組之介面	交流模組	使用者
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
HTTP GET		-	1.收到的邀請資訊 2.送出的邀請資訊 3.交流中的技能資訊
對應之介面需求(Interface Requirement Description)			
顯示所有收到、送出邀請的技能與交流中的技能。			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
EIR-3	使用者與聊天模組之介面	聊天模組	使用者
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
HTTP GET		送出的交流訊息(string)	訊息是否成功傳出(bool)
對應之介面需求(Interface Requirement Description)			
使用者在聊天頁面輸入文字訊息。			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
EIR-4	聊天模組與 Socket API 之介面	Socket API	聊天模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
API Call		訊息封包	訊息封包
對應之介面需求(Interface Requirement Description)			
聊天模組使用 Socket API 完成訊息傳輸。			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-1	登入模組與帳戶模組之介面	帳戶模組	登入模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
Java Method Call		使用者註冊資訊	註冊資訊是否合法
對應之介面需求(Interface Requirement Description)			
登入模組將註冊資訊交給帳戶模組去新增註冊帳戶。			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-2	登入模組與首頁模組之介面	首頁模組	登入模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
Servlet Redirect		使用者帳號資訊	首頁資訊
對應之介面需求(Interface Requirement Description)			
確認登入資料無誤後跳轉至個人首頁。			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-3	首頁模組與帳戶模 組之介面	帳戶模組	首頁模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
Java Method Call		使用者帳號(id : String)	帳戶資訊
對應之介面需求(Interface Requirement Description)			
首頁模組以使用者帳號向帳戶模組要求帳戶資料並將其呈現於個人首頁上。			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-4	首頁模組與技能模 組之介面	技能模組	首頁模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
Java Method Call		使用者帳號(id : String)	使用者技能資訊
對應之介面需求(Interface Requirement Description)			
首頁模組以使用者帳號向技能模組要求使用者擁有的技能資訊。			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-5	首頁模組與配對模 組之介面	配對模組	首頁模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
HTTP GET		使用者興趣技能與技能資 訊	待配對技能集合
對應之介面需求(Interface Requirement Description)			
首頁以使用者技能資訊與興趣技能資訊供配對模組去演算最佳配對技能			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-6	配對模組與技能模組之介面	技能模組	配對模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
Java Method Call		使用者興趣技能與技能資訊	符合配對結果之技能
對應之介面需求(Interface Requirement Description)			
配對模組以使用者技能資訊去要求技能模組給予符合配對結果之技能。			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-7	交流模組與技能模組之介面	技能模組	交流模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
Java Method Call		邀請、交流中的技能 ID	目標技能資訊
對應之介面需求(Interface Requirement Description)			
交流模組向技能模組要求對應的交流中或邀請中技能資訊			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-8	交流模組與資料庫模組之介面	資料庫模組	交流模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
Java Method Call		接受、拒絕、取消邀請訊號與結束交流訊號	-
對應之介面需求(Interface Requirement Description)			
交流模組將接受、拒絕、取消的邀請資料與結束交流的訊息更新到資料庫中。			



介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-9	交流模組與評價模 組之介面	評價模組	交流模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
HTTP GET		評價資訊	是否成功評價(bool)
對應之介面需求(Interface Requirement Description)			
交流模組以評價模組提供之功能實作評價窗口。			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-10	評價模組與資料庫 模組之介面	資料庫模組	評價模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
Java Method Call		評價分數與評論訊息	-
對應之介面需求(Interface Requirement Description)			
評價模組將評價訊息更新置資料庫。			

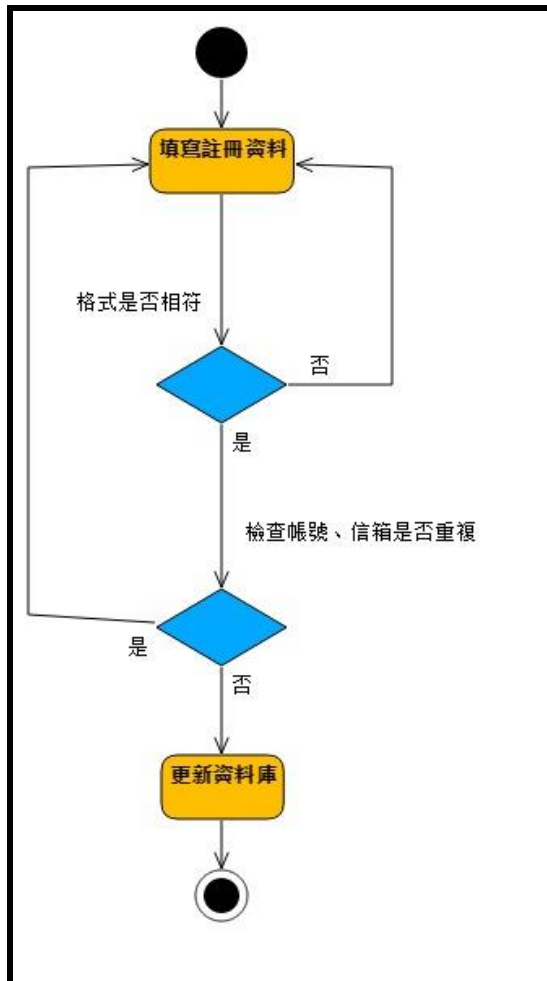
介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-11	聊天模組與資料庫 模組之介面	資料庫模組	聊天模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
Java Method Call		聊天訊息資訊 (內容、時間、發送者)	-
對應之介面需求(Interface Requirement Description)			
將聊天訊息資訊記錄到資料庫中。			

介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-12	帳戶模組與資料庫 模組之介面	資料庫模組	帳戶模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
Java Method Call		帳戶資訊	-
帳戶模組將修改或新增的帳戶資訊更新至資料庫。			

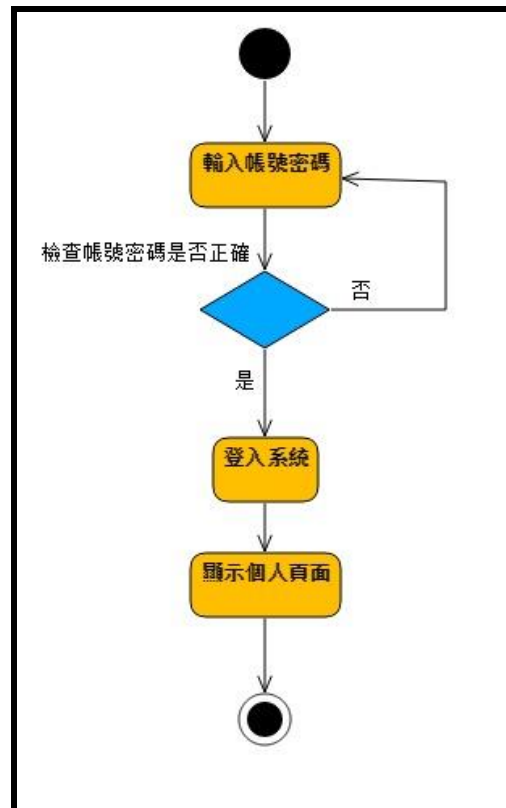
介面編號 (Interface ID)	介面名稱 (Interface Name)	介面提供者 (Interface Provider)	介面使用者 (Interface Consumer)
IIR-13	技能模組與資料庫 模組之介面	資料庫模組	技能模組
連結方式 (Connection Type)		輸入資料 (Input Data)	輸出資料 (Output Data)
Java Method Call		技能資訊	-
對應之介面需求(Interface Requirement Description)			
將修改或新增的技能資訊與興趣技能資訊新增到資料庫中。			

### 3. 流程設計(Process Design)

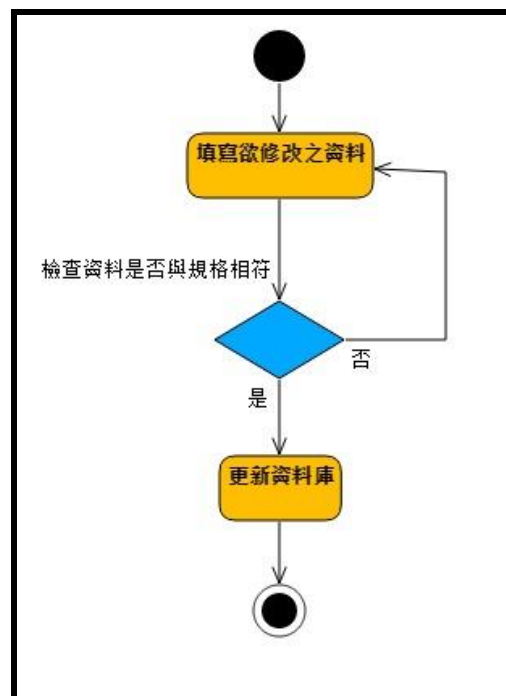
#### 3.1 註冊流程



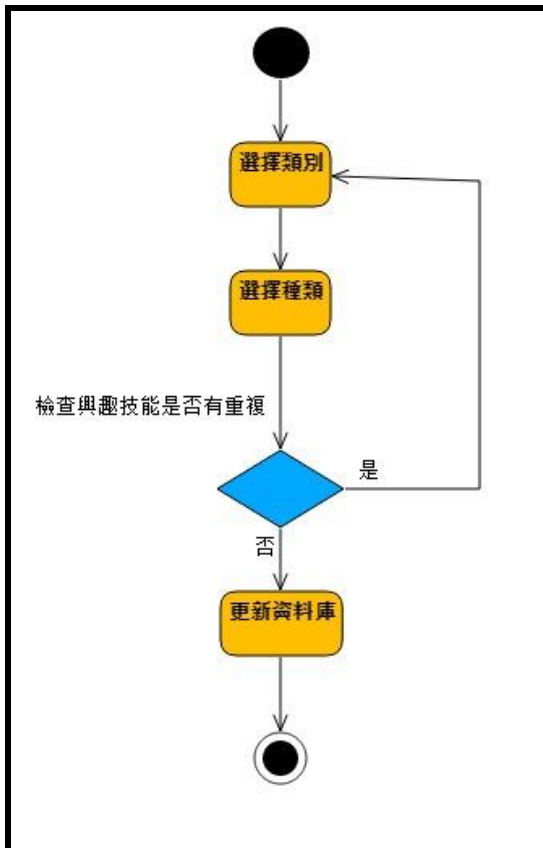
#### 3.2 登入流程



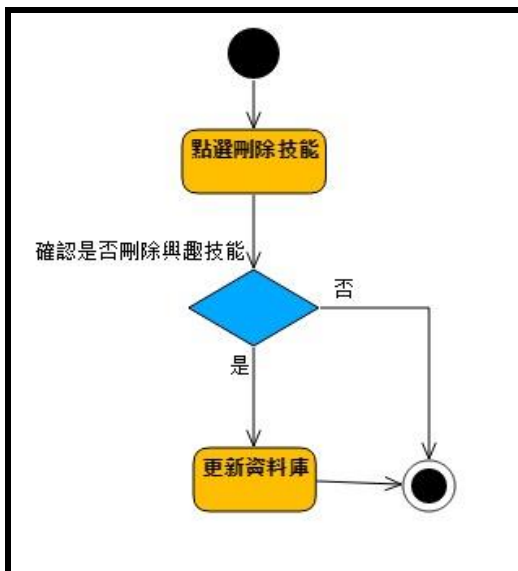
#### 3.3 修改個人資料流程



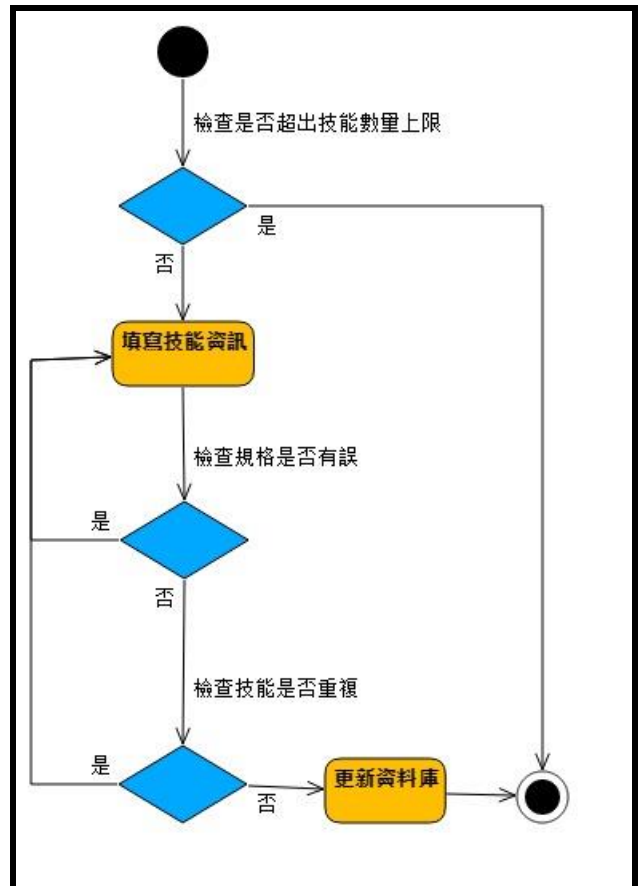
### 3.4 建立興趣技能流程



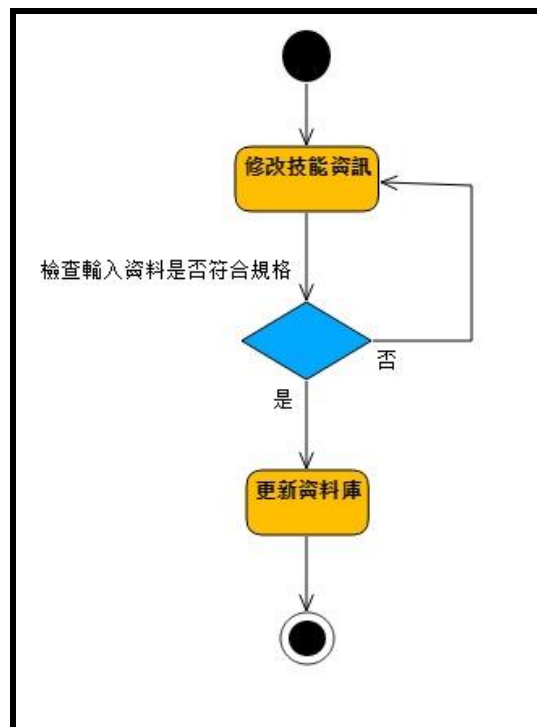
### 3.5 刪除興趣技能流程



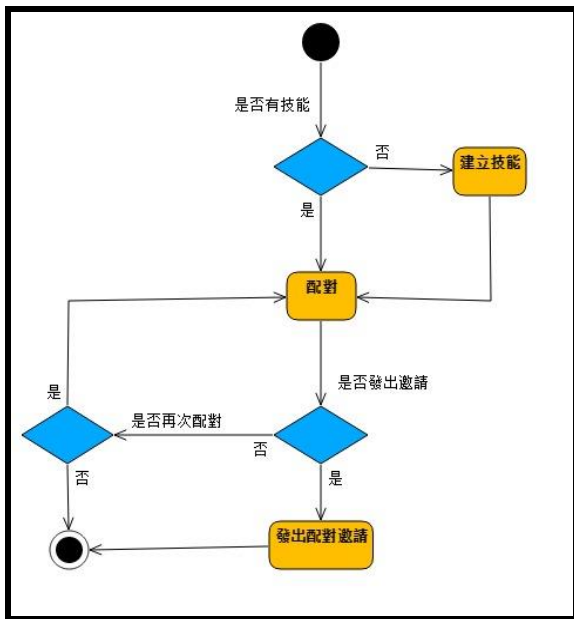
### 3.6 建立技能流程



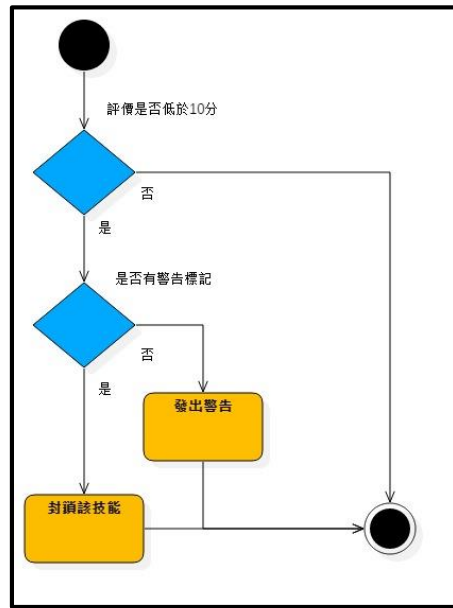
### 3.7 修改技能流程



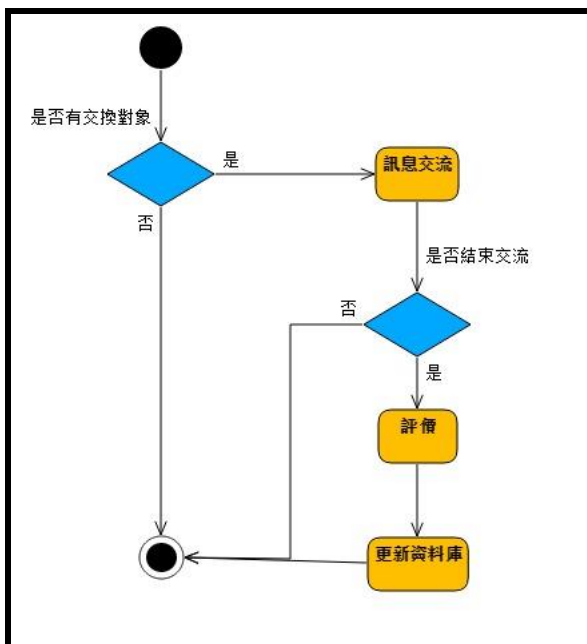
### 3.8 配對流程



### 3.9 黑名單流程

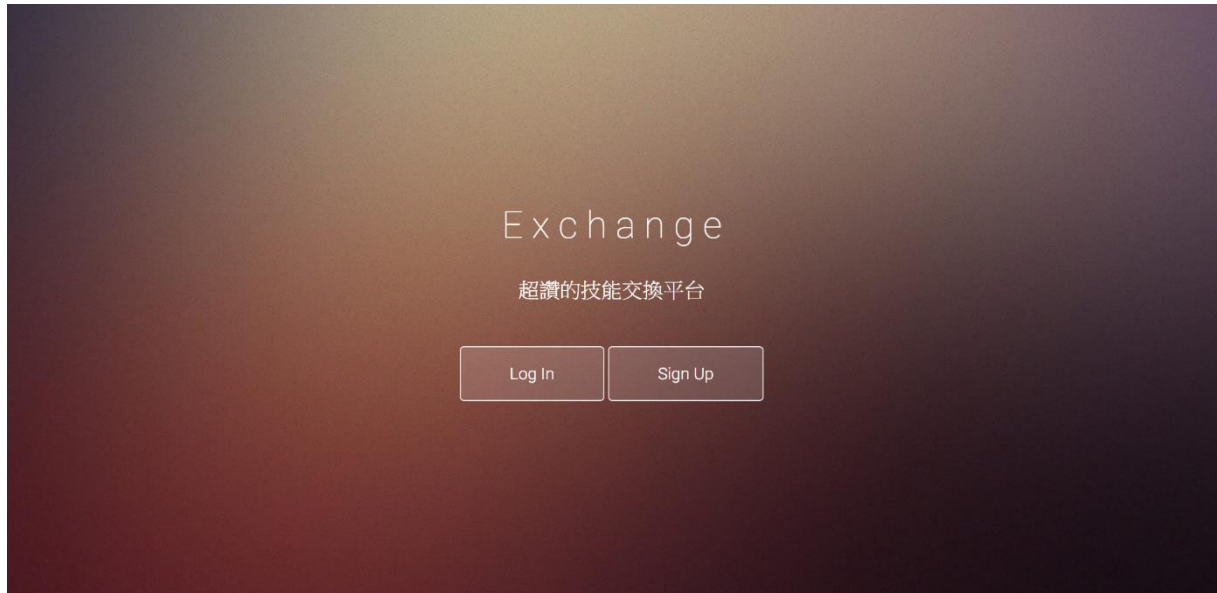


### 3.10 交流管理流程

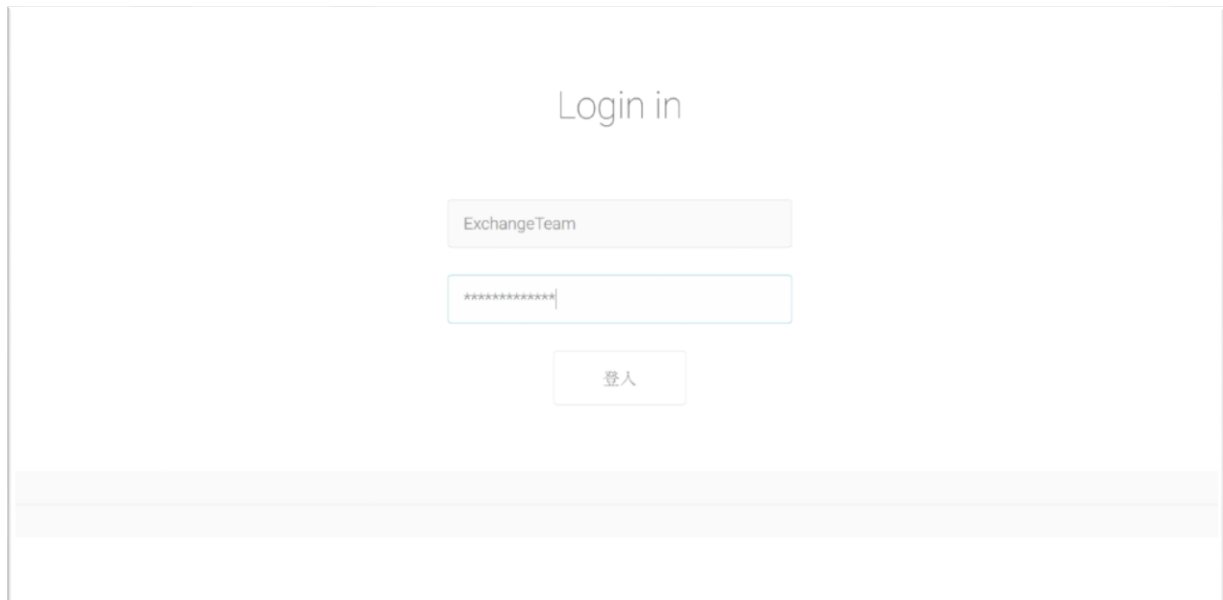


## 4. 使用者畫面設計(User Interface Design)

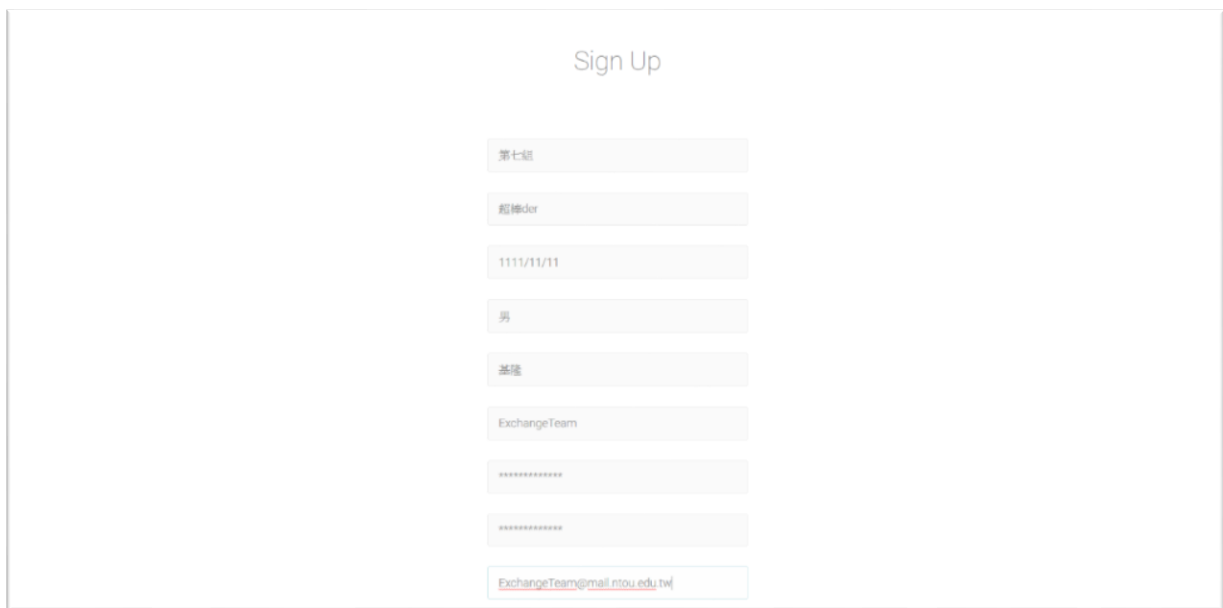
4.1 起始頁面：未登入的使用者進入系統所看到的畫面。



4.2 起始頁面 - 登入：輸入已註冊帳號與密碼後，可使用本系統。



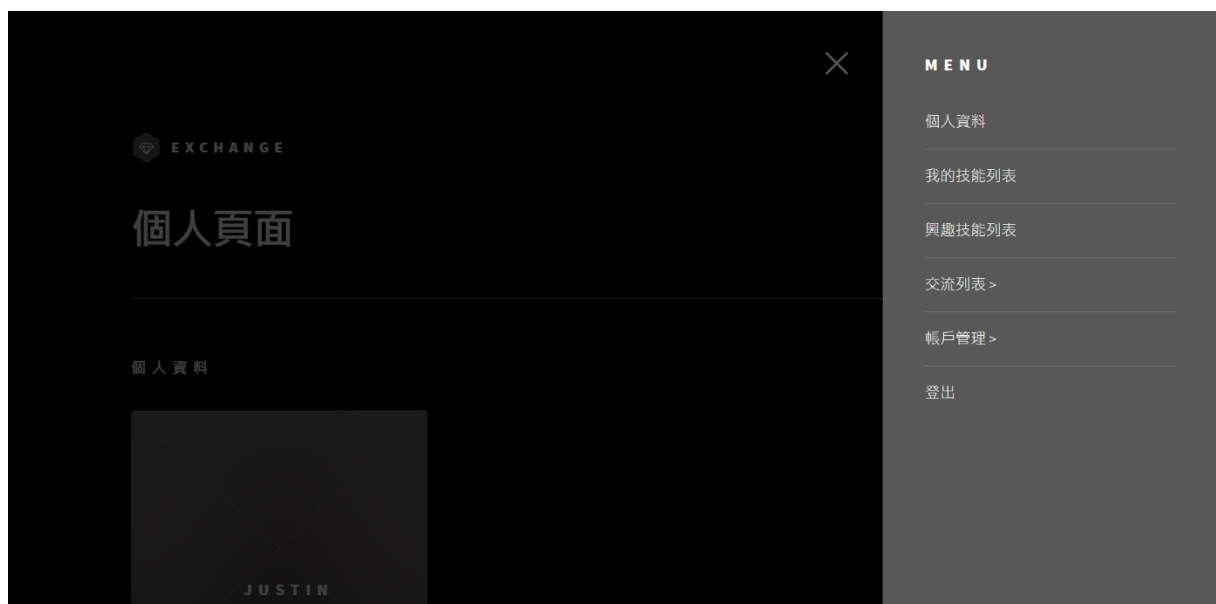
#### 4.3 起始頁面 - 註冊：寫註冊所需的資料。



A registration form titled "Sign Up" with the following fields:

- 第七組
- 超維der
- 1111/11/11
- 男
- 基礎
- ExchangeTeam
- \*\*\*\*\*
- \*\*\*\*\*
- ExchangeTeam@mail.ntou.edu.tw

#### 4.4 個人頁面 - Navigation Bar：顯示出個人頁面提供的各個資訊及連結。



#### 4.5 個人頁面 - 檢視技能：點選技能按鈕，進入技能頁面。





4.6 個人頁面 - 修改個人資料：點選按鈕跳出更改個人資料跳窗。

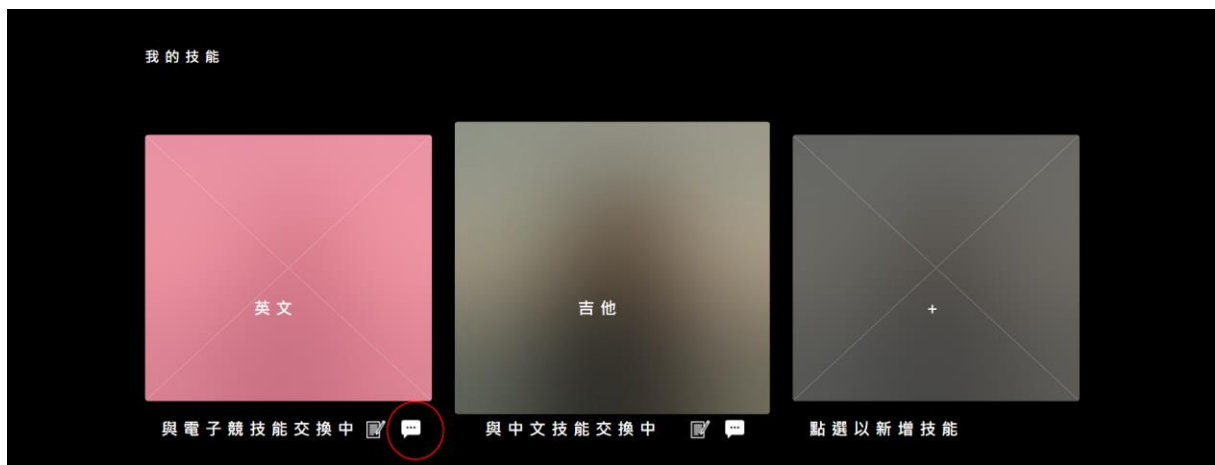
The image shows two parts of the user interface. On the left is the '個人資料' (Personal Information) page, which has a dark background. It features a large grey placeholder for a profile picture with the name 'JUSTIN' in the center. Below the placeholder, the text '男 21歲 基隆市' (Male, 21 years old, Keelung City) is displayed. A red circle highlights a pencil icon in the bottom right corner, indicating the edit function. On the right is the '修改個人資料' (Modify Personal Information) modal window, which has a white background and a close button (X) in the top right. It contains three input fields labeled '暱稱' (Nickname), '地區' (Region), and '信箱' (Email). At the bottom of the modal is a button labeled '確認修改' (Confirm Modification).

4.7 個人頁面 - 編輯技能：點選按鈕編輯技能，進入編輯技能頁面。

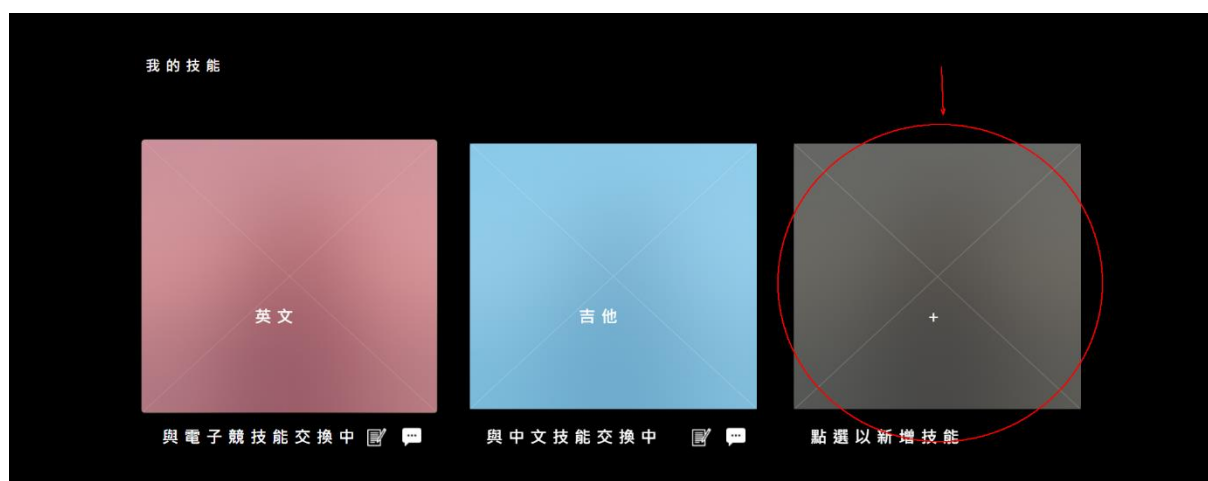
The image shows the '我的技能' (My Skills) page, which has a dark background. It displays three skill cards: '英文' (English) on a red background, '吉他' (Guitar) on a blue background, and a grey card with a '+' sign for adding a new skill. Below each card is a text label and a button with a pencil icon. The labels are '與電子競技能交換中' (Exchanging with Electronic Competition Skills), '與中文技能交換中' (Exchanging with Chinese Skills), and '點選以新增技能' (Click to add skill). The button under the '英文' card is circled in red.

The image shows the skill editing page. At the top left is a cartoon character icon with a '技能' (Skill) label. To its right is a text input field with the placeholder 'Hello my skill is' and two dropdown menus, one showing '音樂' (Music) and the other '吉他' (Guitar). In the top right corner, there is a location pin icon and the text '在基隆市' (In Keelung City). Below this is a section titled '技能的簡介/經歷' (Skill Introduction/Experience) with a large text area for input, labeled '輸入技能簡介及經歷...'. At the bottom right of the page are two buttons: 'Submit' (blue) and 'Cancel' (red).

#### 4.8 個人頁面 - 交換中技能聊天：點選聊天按鈕進入聊天頁面。



#### 4.9 個人頁面 - 新增技能：點選按鈕新增技能，進入新增技能頁面。



#### 4.10 個人頁面 - 配對按鈕：點選進入配對頁面。



Exchange 按鈕：送出請求。

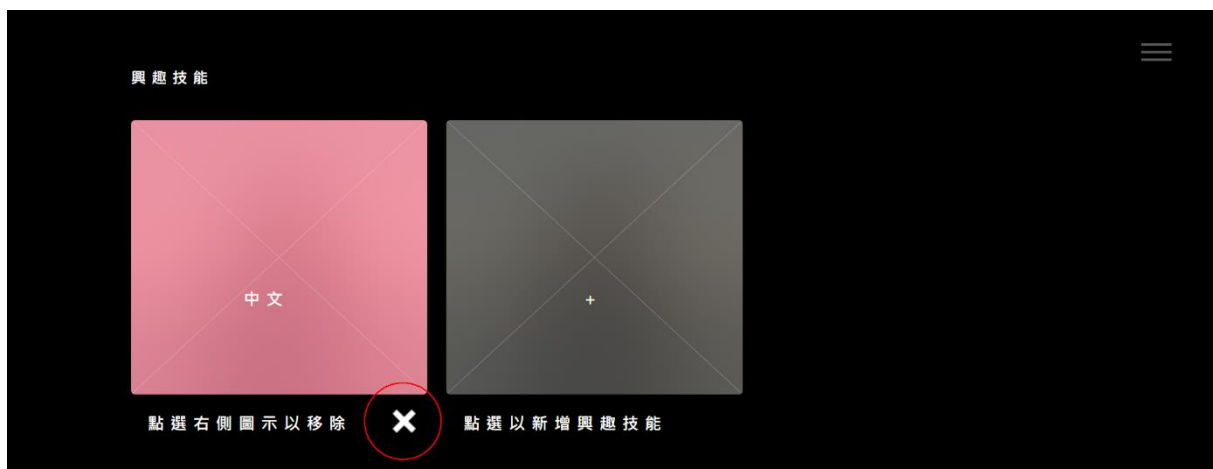
Next 按鈕：下一張技能。

Exit 按鈕：回到個人頁面。

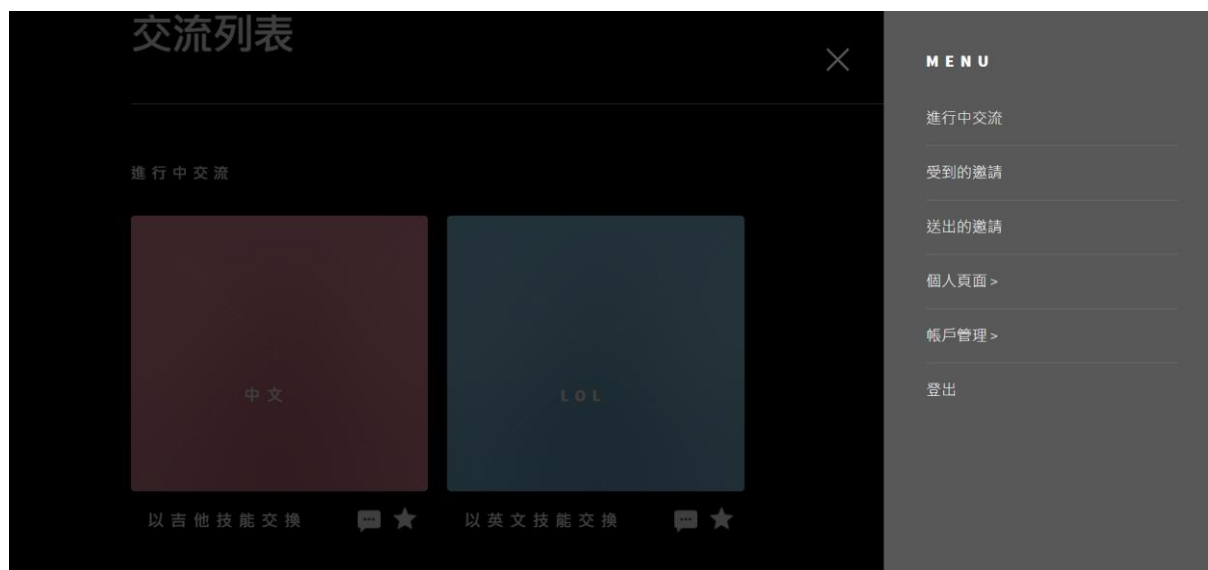
4.11 個人頁面 - 新增興趣技能：點選按鈕跳出新增技能跳窗。



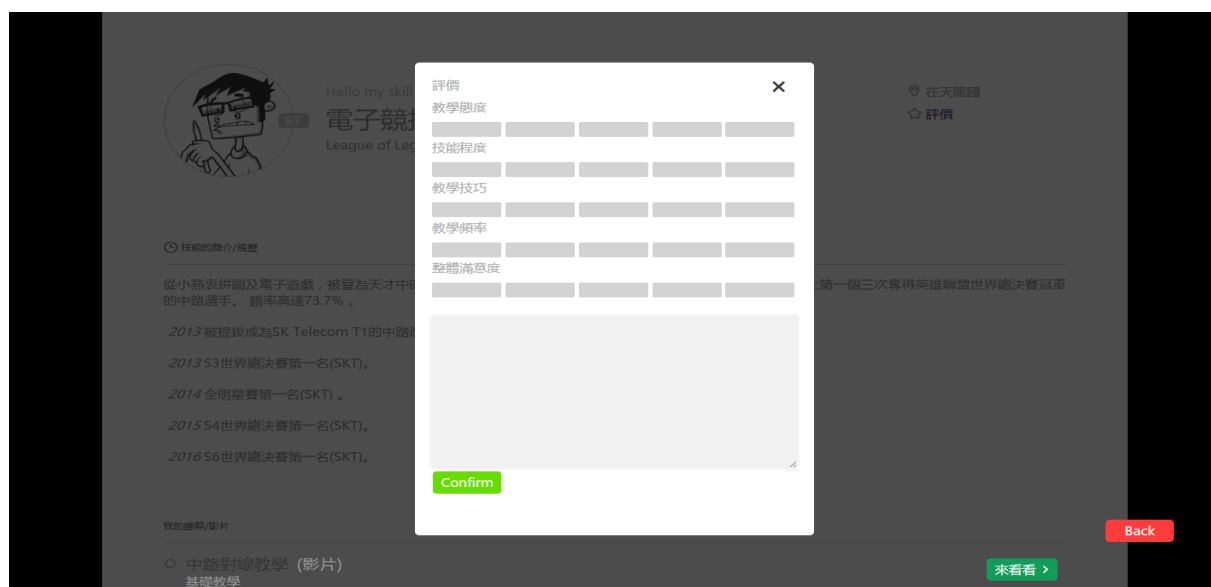
4.12 個人頁面 - 刪除技能：點選按鈕刪除興趣技能。



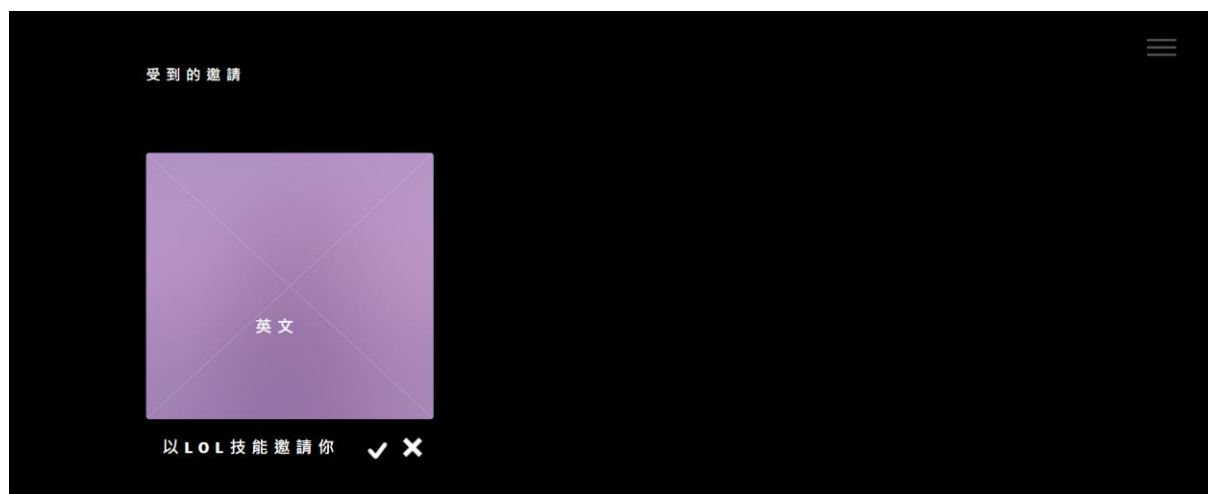
4.13 交流列表 - Navigation Bar：顯示出個人頁面提供的各個資訊及連結。



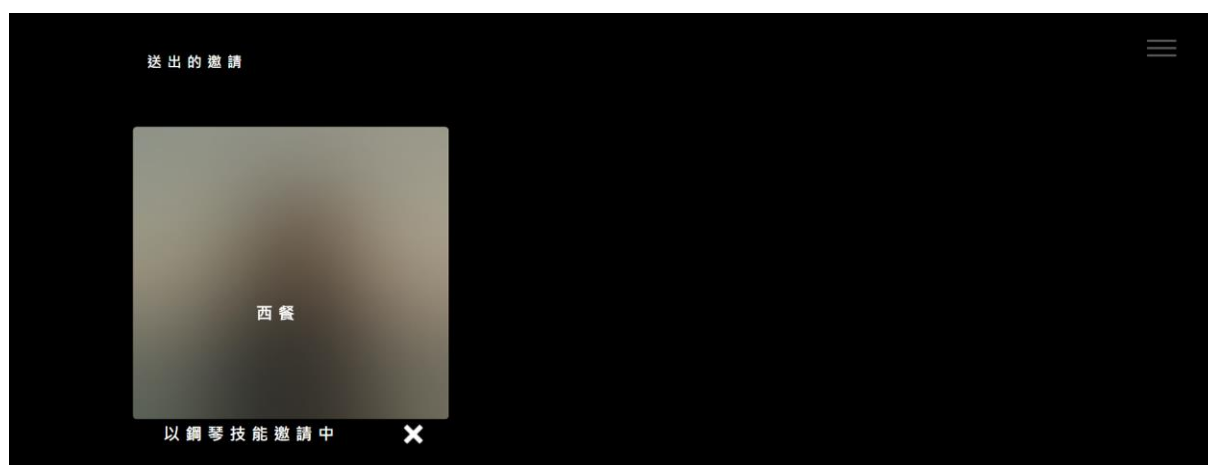
4.14 交流列表 - 評分按鈕：點選按鈕跳到評分頁面。



4.15 交流列表 - 受到的邀請：打勾同意，打叉拒絕。



4.16 交流列表 - 送出的邀請：列出技能配對正在等待的請求。



## 5. 資料細部設計(Data Design)

### 5.1 Account Table

欄位名稱	欄位代號	定義	型態	長度	備註
帳號	user_id	使用者帳號	varchar	20	Primary Key
密碼	password	使用者密碼	varchar	20	Not Null
名稱	user_name	使用者名稱	varchar	20	Not Null
綽號	nick_name	使用者暱稱	varchar	20	Not Null
性別	gender	使用者身體性別	boolean	-	Not Null
電子郵件	email	使用者電子郵件	email	-	Not Null
出生年月日	birthday	使用者生日	date	-	Not Null
居住地區	region	使用者所在地	varchar	20	Not Null
技能數量	skill_number	使用者目前擁有技能數量	int	11	Not Null, Unsigned
技能上限	skill_max	使用者目前可擁有技能上限	int	11	Not Null, Unsigned
最近登入	recent_log	記錄使用者最近登入時間	date	-	Not Null

```

CREATE TABLE accounts (
    user_id      varchar(20),
    password     varchar(20) NOT NULL,
    user_name    varchar(20) NOT NULL,
    nick_name    varchar(20) NOT NULL,
    gender       boolean NOT NULL,
    email        varchar(20) NOT NULL,
    birthday     date NOT NULL,
    region       varchar(20) NOT NULL,
    skill_number int(11)  UNSIGNED NOT NULL,
    skill_max    int(11)  UNSIGNED NOT NULL,
    recent_log   date NOT NULL,
    PRIMARY KEY (user_id)
);

```



## 5.2 Favorite Table

欄位名稱	欄位代號	定義	型態	長度	備註
種類名稱	type_name	使用者興趣技能的名稱	varchar	20	Primary Key
帳號	account	使用者帳號	varchar	20	Primary Key

```
CREATE TABLE favorites (
    type_name    varchar(20),
    account      varchar(20),
    PRIMARY KEY (type_name, account)
);
```

## 5.3 Skill Table

欄位名稱	欄位代號	定義	型態	長度	備註
技能 ID	skill_id	技能的 ID	int	11	Primary Key, Auto_Increment
帳號	account	使用者帳號	varchar	20	Not Null, Unique A
種類名稱	type_name	技能種類名稱 (吉他、英文)	varchar	20	Not Null, Unique A
介紹/經歷	intro_expr	使用者技能程度/經歷描述	text	-	Not Null
技能等級	skill_level	技能等級	int	11	Not Null, Unsigned
已交換次數	times	技能的總交流次數	int	11	Not Null, Unsigned
教學態度總分	attitude_score	該技能教學態度評價項所得總分	int	11	Not Null, Unsigned
技能程度總分	profession_score	該技能技能程度評價項所得總分	int	11	Not Null, Unsigned
教學技巧總分	teaching_score	該技能教學技巧評價項所得總分	int	11	Not Null, Unsigned
教學頻率總分	frequency_score	該技能教學頻率評價項所得總分	int	11	Not Null, Unsigned

整體滿意度 總分	satisfaction_score	該技能整體滿意度評價項所得總分	int	11	Not Null, Unsigned
警告標記	warning_tag	該技能是否已受到警告(上次交流分數總分低於 10 分)	boolean	-	Not Null
黑名單標記	bad_tag	該技能是否已被封鎖(連續兩次交流總分皆低於 10 分)	boolean	-	Not Null

**CREATE TABLE** skills (

skill\_id            **int(11)    AUTO\_INCREMENT,**  
 account            **varchar(20)   NOT NULL,**  
 type\_name          **varchar(20)   NOT NULL,**  
 intro\_expr          **text        NOT NULL,**  
 skill\_level         **int(11)    UNSIGNED   NOT NULL,**  
 times               **int(11)    UNSIGNED   NOT NULL,**  
 attitude\_score      **int(11)    UNSIGNED   NOT NULL,**  
 profession\_score   **int(11)    UNSIGNED   NOT NULL,**  
 teaching\_score      **int(11)    UNSIGNED   NOT NULL,**  
 frequency\_score    **int(11)    UNSIGNED   NOT NULL,**  
 satisfaction\_score **int(11)    UNSIGNED   NOT NULL,**  
 warning\_tag         **boolean   NOT NULL,**  
 bad\_tag             **boolean   NOT NULL,**  
**UNIQUE** (account, type\_name),  
**PRIMARY KEY** (skill\_id)

) **AUTO\_INCREMENT = 1;**

#### 5.4 Invitation Table

欄位名稱	欄位代號	定義	型態	長度	備註
邀請發送端技能	ivt_sender	寄出邀請的技能 ID	int	11	Primary Key(skill_id)
邀請接收端技能	ivt_receiver	受到邀請的技能 ID	int	11	Primary Key(skill_id)

```
CREATE TABLE invitations (
    ivt_sender      int(11),
    ivt_receiver    int(11),
    PRIMARY KEY (ivt_sender, ivt_receiver)
);
```

#### 5.5 Exchange Table

欄位名稱	欄位代號	定義	型態	長度	備註
技能 A	skill_a	交流中技能 ID(發送者)	int	11	Primary Key(skill_id)
技能 B	skill_b	交流中技能 ID(受邀者)	int	11	Primary Key(skill_id)
結束旗標 A	end_a	技能 A 交流是否已經結束	boolean	-	Not Null
結束旗標 B	end_b	技能 B 交流是否已經結束	boolean	-	Not Null

```
CREATE TABLE exchanges (
    skill_a  int(11),
    skill_b  int(11),
    end_a    boolean,
    end_b    boolean,
    PRIMARY KEY (skill_a, skill_b)
);
```

## 5.6 kind Table

欄位名稱	欄位代號	定義	型態	長度	備註
類別代碼	class_code	類別 ID	varchar	20	Primary Key
類別名稱	class_name	技能分類之類別名稱 (語言、體育)	varchar	20	Unique

```
CREATE TABLE classes (
    class_code    varchar(20),
    class_name    varchar(20),
    UNIQUE (class_name),
    PRIMARY KEY (class_code)
);
```

class_code	class_name
MUS	音樂
LNG	語言
PLG	程式語言
DNC	舞蹈
SPT	運動與競技
CKN	烹飪
ART	藝術
DSN	設計
SKL	技藝
CRD	棋牌

## 5.7 Type Table

欄位名稱	欄位代號	定義	型態	長度	備註
種類代碼	type_code	種類 ID	varchar	20	Primary Key
種類名稱	type_name	技能種類之名稱(英文、籃球)	varchar	20	Unique
類別代碼	class_code	技能種類之對應類別 ID	varchar	20	Not Null

```
CREATE TABLE types (
    type_code    varchar(20),
    type_name    varchar(20),
    class_code   varchar(20) NOT NULL,
    UNIQUE (type_name),
    PRIMARY KEY (type_code)
);
```

type_code	type_name	class_code
MUS-001	歌唱	MUS
MUS-002	吉他	MUS
MUS-003	鋼琴	MUS
MUS-004	小提琴	MUS
MUS-005	中提琴	MUS
MUS-006	大提琴	MUS
MUS-007	口琴	MUS
MUS-008	電吉他	MUS
MUS-009	爵士鼓	MUS
MUS-010	法國號	MUS
MUS-011	喇叭	MUS
MUS-012	貝斯	MUS
MUS-013	定音鼓	MUS
MUS-014	笙	MUS
MUS-015	蕭	MUS
MUS-016	單簧管	MUS
MUS-017	豎笛	MUS
MUS-018	橫笛	MUS
MUS-019	直笛	MUS

MUS-020	唢呐	MUS
MUS-021	二胡	MUS
MUS-022	古筝	MUS
LNG-001	英文	LNG
LNG-002	中文	LNG
LNG-003	日文	LNG
LNG-004	德文	LNG
LNG-005	泰文	LNG
LNG-006	法文	LNG
LNG-007	西班牙文	LNG
LNG-008	拉丁文	LNG
LNG-009	韓文	LNG
LNG-010	越南文	LNG
LNG-011	希臘文	LNG
LNG-012	阿拉伯文	LNG
LNG-013	俄文	LNG
LNG-014	夏威夷文	LNG
LNG-015	挪威文	LNG
LNG-016	馬來文	LNG
LNG-017	菲律賓文	LNG
LNG-018	瑞典文	LNG
LNG-019	荷蘭文	LNG
LNG-020	希臘文	LNG
PLG-001	C	PLG
PLG-002	Java	PLG
PLG-003	Python	PLG
PLG-004	C++	PLG
PLG-005	R	PLG
PLG-006	C#	PLG
PLG-007	PHP	PLG
PLG-008	JavaScript	PLG
PLG-009	Ruby	PLG
PLG-010	Go	PLG
PLG-011	Matlab	PLG
PLG-012	Swift	PLG
PLG-013	VB	PLG
PLG-014	Perl	PLG

PLG-015	Html	PLG
PLG-016	Android	PLG
DNC-001	原住民舞蹈	DNC
DNC-002	民俗舞蹈	DNC
DNC-003	現代舞	DNC
DNC-004	芭蕾舞	DNC
DNC-005	華爾滋	DNC
DNC-006	探戈	DNC
DNC-007	踢踏舞	DNC
DNC-008	鬥牛舞	DNC
DNC-009	街舞	DNC
DNC-010	機械舞	DNC
DNC-011	鎖舞	DNC
DNC-012	Breaking	DNC
DNC-013	舊學派嘻哈	DNC
DNC-014	新學派嘻哈	DNC
DNC-015	Freestyle	DNC
DNC-016	電流舞	DNC
DNC-017	鋼管舞	DNC
DNC-018	肚皮舞	DNC
DNC-019	夏威夷草裙舞	DNC
DNC-020	鬥牛舞	DNC
SPT-001	跳水	SPT
SPT-002	游泳	SPT
SPT-003	水上芭雷	SPT
SPT-004	水球	SPT
SPT-005	輕艇	SPT
SPT-006	自行車	SPT
SPT-007	體操	SPT
SPT-008	排球	SPT
SPT-009	場地障礙賽	SPT
SPT-010	射箭	SPT
SPT-011	田徑	SPT
SPT-012	羽毛球	SPT
SPT-013	籃球	SPT
SPT-014	拳擊	SPT
SPT-015	擊劍	SPT

SPT-016	曲棍球	SPT
SPT-017	足球	SPT
SPT-018	高爾夫	SPT
SPT-019	手球	SPT
SPT-020	柔道	SPT
SPT-021	划船	SPT
SPT-022	橄欖球	SPT
SPT-023	帆船	SPT
SPT-024	射擊	SPT
SPT-025	桌球	SPT
SPT-026	跆拳道	SPT
SPT-027	網球	SPT
SPT-028	棒球	SPT
SPT-029	健身	SPT
SPT-030	壘球	SPT
SPT-031	太極拳	SPT
SPT-032	滑板	SPT
SPT-033	直排輪	SPT
SPT-034	電子競技	SPT
SPT-035	溜冰	SPT
CKN-001	料理	CKN
CKN-002	烘焙	CKN
CKN-003	調酒	CKN
ART-001	素描	ART
ART-002	電繪	ART
ART-003	書法	ART
ART-004	雕塑	ART
ART-005	戲劇	ART
ART-006	水彩	ART
ART-007	油畫	ART
ART-008	版畫	ART
ART-009	水墨畫	ART
ART-010	裝置藝術	ART
ART-011	行為藝術	ART
DSN-001	髮型設計	DSN
DSN-002	廣告設計	DSN
DSN-003	服裝設計	DSN



SKL-001	魔術	SKL
SKL-002	扯鈴	SKL
SKL-003	舞龍舞獅	SKL
SKL-004	陀螺	SKL
CRD-001	橋牌	CRD
CRD-002	象棋	CRD
CRD-003	圍棋	CRD
CRD-004	跳棋	CRD
CRD-005	西洋棋	CRD
CRD-006	黑白棋	CRD

### 5.8 Comment Table

欄位名稱	欄位代號	定義	型態	長度	備註
技能 ID	skill_id	該評論對應之技能 ID	int	11	Not Null
評論內容	comment	評論的文字內容	varchar	20	Not Null
時間	date	評論時間	date	-	Not Null

```
CREATE TABLE comments (
    skill_id      int(11)  NOT NULL,
    comment       varchar(20) NOT NULL,
    date          date     NOT NULL
);
```

### 5.9 Message Table

欄位名稱	欄位代號	定義	型態	長度	備註
訊息 ID	msg_id	該訊息的 ID	int	11	Primary Key
訊息發送端技能	msg_sender	該訊息發送端技能 ID	int	11	Not Null (skill_id)
內容	content	訊息文字內容	text	-	Not Null

```
CREATE TABLE messages (
    msg_id  int(11),
    msg_sender int(11) NOT NULL,
    content text     NOT NULL,
    PRIMARY KEY (msg_id)
);
```

### 5.10 Image Table

欄位名稱	欄位代號	定義	型態	長度	備註
技能 ID	skill_id	該圖片所在之技能 ID	int	11	Not Null
圖片內容	image	圖片 URL	varchar	2083	Not Null

```
CREATE TABLE images (  
    skill_id int(11) NOT NULL,  
    image varchar(2083) NOT NULL  
);
```

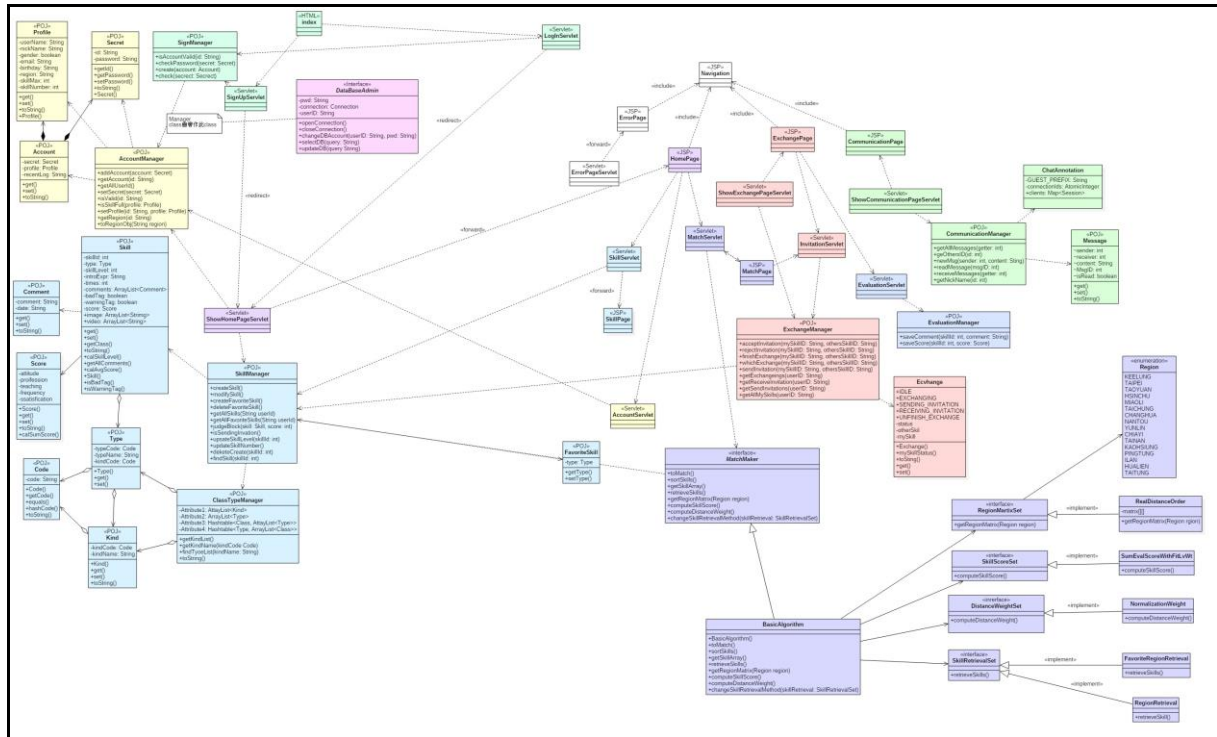
### 5.11 Video Table

欄位名稱	欄位代號	定義	型態	長度	備註
技能 ID	skill_id	該影片所在之技能 ID	int	11	Not Null
影片內容	video	影片 URL	varchar	2083	Not Null

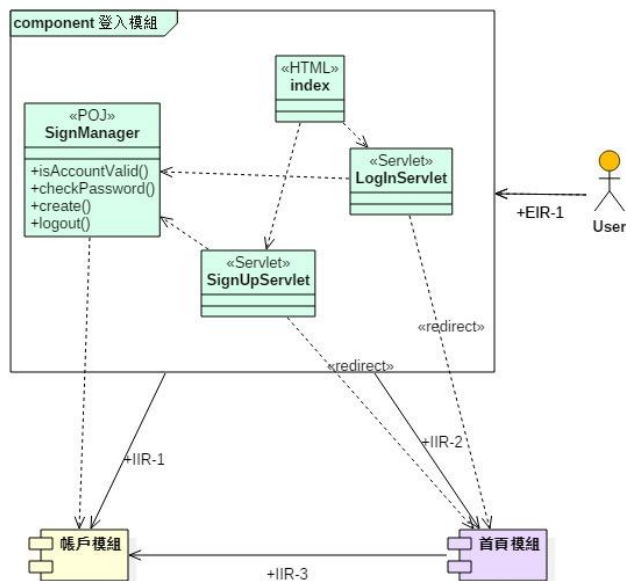
```
CREATE TABLE videos (  
    skill_id int(11) NOT NULL,  
    video varchar(2083) NOT NULL  
);
```

## 6. 類別圖設計(Class Diagram)

### 6.1 類別圖總覽



## 6.2 登入模組

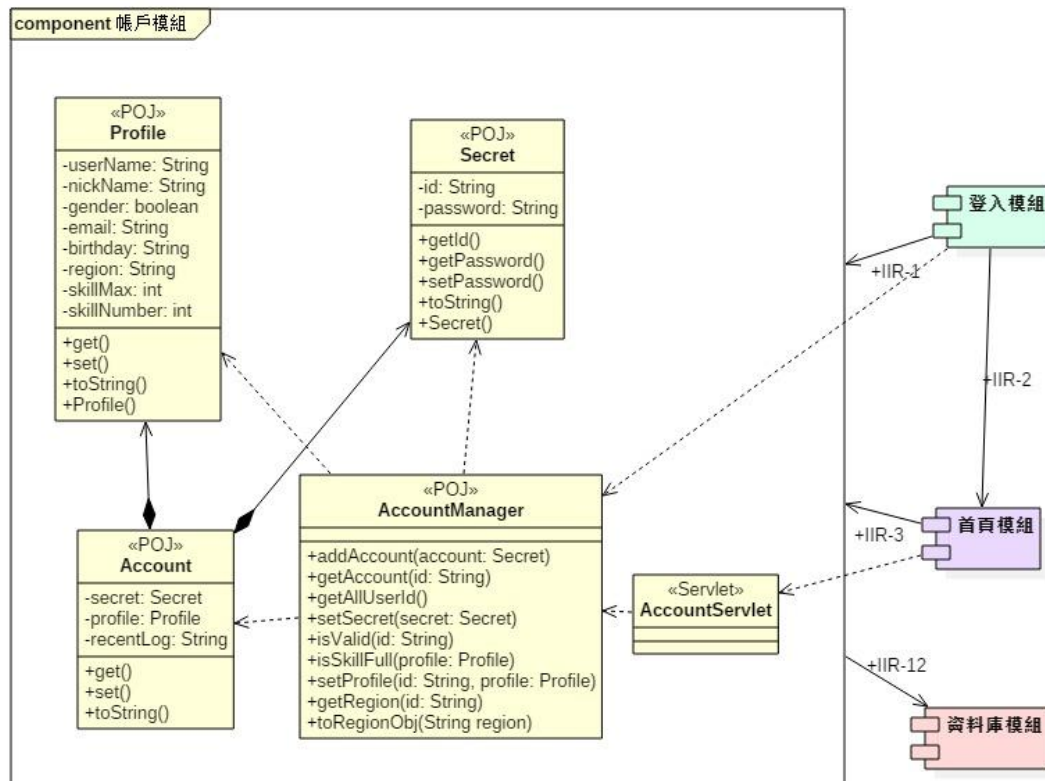


<b>Class Name</b>	SignUpServlet
<b>Category</b>	Servlet
<b>Module</b>	登入模組
<b>Description</b>	將註冊資料交付給 POJ 處理後判斷是否跳轉至首頁
<b>Relationship with other classes</b>	index、SignManager、ShowHomePageServlet
<b>Attributes</b>	無
<b>Traceability with other use case</b>	註冊帳戶(E-UCA-UC-001)

<b>Class Name</b>	LoginServlet
<b>Category</b>	Servlet
<b>Module</b>	登入模組
<b>Description</b>	將登入資料交付給 POJ 處理後判斷是否跳轉至首頁
<b>Relationship with other classes</b>	index、SignManager、ShowHomePageServlet
<b>Attributes</b>	無
<b>Traceability with other use case</b>	登入系統(E-UCA-UC-002)

<b>Class Name</b>	SignManager
<b>Category</b>	POJ
<b>Module</b>	登入模組
<b>Description</b>	管理登入(確認帳號有效、認證密碼正確)、註冊、登出
<b>Relationship with other classes</b>	SignUpServlet、LoginServlet、AccountManager
<b>Attributes</b>	-
<b>Traceability with other use case</b>	註冊帳戶(E-UCA-UC-001)、登入系統(E-UCA-UC-002)

## 6.3 帳戶模組



<b>Class Name</b>	Profile
<b>Category</b>	POJ
<b>Module</b>	帳戶模組
<b>Description</b>	儲存使用者名字、暱稱、性別、E-mail、生日、地區、目前技能上限、擁有技能數量。
<b>Relationship with other classes</b>	Account、AccountManager
<b>Attributes</b>	userName、nickName、gender、email、birthday、region、skillMax、skillNumber
<b>Traceability with other use case</b>	註冊帳戶(E-UCA-UC-001)、檔案管理(E-UCA-UC-003)

<b>Class Name</b>	Secret
<b>Category</b>	POJ
<b>Module</b>	帳戶模組
<b>Description</b>	儲存使用者帳號、密碼。
<b>Relationship with other classes</b>	Account、AccountManager
<b>Attributes</b>	id、password
<b>Traceability with other use case</b>	註冊帳戶(E-UCA-UC-001)、登入系統(E-UCA-UC-002)、檔案管理(E-UCA-UC-003)

<b>Class Name</b>	Account
<b>Category</b>	POJ
<b>Module</b>	帳戶模組
<b>Description</b>	儲存個人資料與帳戶資料
<b>Relationship with other classes</b>	Profile、Secret、AccountManager
<b>Attributes</b>	secret、profile、recentLog
<b>Traceability with other use case</b>	註冊帳戶(E-UCA-UC-001)、登入系統(E-UCA-UC-002)、檔案管理(E-UCA-UC-003)

<b>Class Name</b>	AccountManager
<b>Category</b>	POJ
<b>Module</b>	帳戶模組
<b>Description</b>	管理使用者所有個人帳戶資料：存取使用者資料、新增使用者帳戶、確認使用者帳號密碼是否有效、確認技能數是否已達上限
<b>Relationship with other classes</b>	SignManager、ShowHomePageServlet、AccountServlet、Profile、Account、Secret
<b>Attributes</b>	-
<b>Traceability with other use case</b>	註冊帳戶(E-UCA-UC-001)、登入系統(E-UCA-UC-002)、檔案管理(E-UCA-UC-003)、技能管理(E-UCA-UC-006)

<b>Class Name</b>	AccountServlet
<b>Category</b>	Servlet
<b>Module</b>	帳戶模組
<b>Description</b>	指派個人首頁所有與使用者個人帳戶資訊的資料相關任務給AccountManager
<b>Relationship with other classes</b>	HomePage、AccountManager
<b>Attributes</b>	無
<b>Traceability with other use case</b>	檔案管理(E-UCA-UC-003)

The diagram illustrates the architecture of a Skill Management System, divided into two main sections: **component 技能模块** (Skill Module) and **component 交互模块** (Interaction Module).

**Component 技能模块 (Skill Module):**

- «POJ» Skill**: Contains attributes like skillId, type, skillLevel, introExpr, times, comments, badTag, warningTag, score, image, and video. It has methods get(), set(), getClass(), toString(), calSkillLevel(), getAllComments(), calAvgScore(), skill(), isBadTag(), and isWarningTag().
- «POJ» Comment**: Contains comment and date attributes, with methods get(), set(), and toString().
- «POJ» Score**: Contains attributes like attitude, profession, teaching, frequency, and satisfaction. It has methods Score(), get(), set(), toString(), and calSumScore().
- «POJ» Type**: Contains typeCode, typeName, and kindCode attributes, with methods Type(), get(), and set().
- «POJ» Code**: Contains code attribute, with methods Code(), getCode(), equals(), hashCode(), and toString().
- «POJ» Kind**: Contains kindCode and kindName attributes, with methods Kind(), get(), set(), and toString().
- «POJ» SkillManager**: A central management class with methods like createSkill(), modifySkill(), createFavoriteSkill(), deleteFavoriteSkill(), getAllSkills(), getAllFavoriteSkills(), judgeBlock(), isSendingInvation(), upstateSkillLevel(), updateSkillNumber(), deleteCreate(), and findSkill().
- «POJ» Class TypeManager**: Manages class types with methods like getKindList(), getKindName(), findTypeList(), and toString().
- «JSP» SkillPage (from Model)**: A presentation layer component.
- «Servlet» SkillServlet (from Model)**: A controller layer component.
- «POJ» FavoriteSkill**: Contains type attribute, with methods getType() and setType().

**Component 交互模块 (Interaction Module):**

- 首页模块** (Home Module)
- 配对模块** (Matching Module)
- 交流模块** (Communication Module)
- 资料库模块** (Library Module)

**Interactions:**

- 首页模块** interacts with **配对模块** via **+IIR-4** and **+IIR-5**.
- 配对模块** interacts with **交流模块** via **+IIR-6** and **+IIR-7**.
- 交流模块** interacts with **资料库模块** via **+IIR-8** and **+IIR-13**.
- 技能模块** components interact with the **交互模块** components via various interfaces (IIR-4 to IIR-13).

<b>Class Name</b>	SkillManager
<b>Category</b>	POJ
<b>Module</b>	技能模組
<b>Description</b>	管理使用者技能：新增、修改技能、新增興趣技能、刪除興趣技能，另外還包括讀取使用者所有技能及興趣技能、修改技能狀態(封鎖)等功能
<b>Relationship with other classes</b>	ShowHomePageServlet、SkillServlet、ExchangeManager、Match-Maker、Skill、ClassTypeManager、FavoriteSkill
<b>Attributes</b>	-
<b>Traceability with other use case</b>	檔案管理(E-UCA-UC-003)、技能管理(E-UCA-UC-006)、技能配對(E-UCA-UC-007)

<b>Class Name</b>	Comment
<b>Category</b>	POJ
<b>Module</b>	技能模組
<b>Description</b>	儲存評論資料
<b>Relationship with other classes</b>	Skill
<b>Attributes</b>	comment、date
<b>Traceability with other use case</b>	技能評價(E-UCA-UC-004)

<b>Class Name</b>	Code
<b>Category</b>	POJ
<b>Module</b>	技能模組
<b>Description</b>	記載一個類別或種類的編碼(代號)
<b>Relationship with other classes</b>	Type、Kind
<b>Attributes</b>	code
<b>Traceability with other use case</b>	檔案管理(E-UCA-UC-003)、技能管理(E-UCA-UC-006)

<b>Class Name</b>	Type
<b>Category</b>	POJ
<b>Module</b>	技能模組
<b>Description</b>	儲存技能種類資料
<b>Relationship with other classes</b>	Skill、ClassTypeManager、Kind、Code
<b>Attributes</b>	typeCode、typeName、kindCode
<b>Traceability with other use case</b>	檔案管理(E-UCA-UC-003)、技能管理(E-UCA-UC-006)

<b>Class Name</b>	Kind
<b>Category</b>	POJ
<b>Module</b>	技能模組
<b>Description</b>	儲存技能類別資料
<b>Relationship with other classes</b>	Type、ClassTypeManager、Code
<b>Attributes</b>	classCode、className
<b>Traceability with other use case</b>	檔案管理(E-UCA-UC-003)、技能管理(E-UCA-UC-006)

<b>Class Name</b>	Score
<b>Category</b>	POJ
<b>Module</b>	技能模組
<b>Description</b>	儲存技能各項分數
<b>Relationship with other classes</b>	Skill
<b>Attributes</b>	attitude、profession、teaching、frequency、satisfaction
<b>Traceability with</b>	檔案管理(E-UCA-UC-003)、技能管理(E-UCA-UC-006)



<b>other use case</b>	
-----------------------	--

<b>Class Name</b>	KindTypeManager
<b>Category</b>	POJ
<b>Module</b>	技能模組
<b>Description</b>	各個技能類別與種類的對照表，提供迅速查找
<b>Relationship with other classes</b>	SkillManager、Type、Class
<b>Attributes</b>	classSet、typeSet、types、class
<b>Traceability with other use case</b>	檔案管理(E-UCA-UC-003)、技能管理(E-UCA-UC-006)

<b>Class Name</b>	FavoriteSkill
<b>Category</b>	POJ
<b>Module</b>	技能模組
<b>Description</b>	儲存興趣技能資料
<b>Relationship with other classes</b>	SkillManager
<b>Attributes</b>	type
<b>Traceability with other use case</b>	檔案管理(E-UCA-UC-003)、技能配對(E-UCA-UC-007)

<b>Class Name</b>	SkillServlet
<b>Category</b>	Servlet
<b>Module</b>	技能模組
<b>Description</b>	指派個人首頁所有與技能資訊的資料相關任務給 SkillManager
<b>Relationship with other classes</b>	HomePage、SkillManager、SkillPage
<b>Attributes</b>	無
<b>Traceability with other use case</b>	檔案管理(E-UCA-UC-003)、交流管理(E-UCA-UC-005)、技能管理(E-UCA-UC-006)

<b>Class Name</b>	SkillPage
<b>Category</b>	JSP
<b>Module</b>	技能模組
<b>Description</b>	顯示技能詳細資料頁面
<b>Relationship with other classes</b>	SkillServlet
<b>Attributes</b>	無
<b>Traceability with other use case</b>	技能評價(E-UCA-UC-004)、技能管理(E-UCA-UC-006)

The diagram illustrates the component configuration for the MatchMaker system. It shows the following components and their relationships:

- MatchServer** (Component) depends on **MatchPage** (Component) and **MatchMaker** (Interface).
- MatchPage** (Component) depends on **MatchMaker** (Interface).
- MatchMaker** (Interface) defines methods: `+toMatch()`, `+sortSkills()`, `+getSkillArray()`, `+retrieveSkills()`, `+getRegionMatrix(Region region)`, `+computeSkillScore()`, `+computeDistanceWeight()`, and `+changeSkillRetrievalMethod(skillRetrieval: SkillRetrievalSet)`.
- BasicAlgorithm** (Component) implements **MatchMaker** (Interface) and depends on **RegionMatrixSet** (Interface), **SkillScoreSet** (Interface), **DistanceWeightSet** (Interface), **SkillRetrievalSet** (Interface), and **RegionRetrieval** (Interface).
- RegionMatrixSet** (Interface) defines the method: `+getRegionMatrix(Region region)`.
- SkillScoreSet** (Interface) defines the method: `+computeSkillScore()`.
- DistanceWeightSet** (Interface) defines the method: `+computeDistanceWeight()`.
- SkillRetrievalSet** (Interface) defines the method: `+retrieveSkills()`.
- RegionRetrieval** (Interface) defines the method: `+retrieveSkill()`.
- RealDistanceOrder** (Component) implements **RegionMatrixSet** (Interface) and depends on **Region** (Enumeration).
- SumEvalScoreWithFitLVWt** (Component) implements **SkillScoreSet** (Interface) and depends on **Region** (Enumeration).
- NormalizationWeight** (Component) implements **DistanceWeightSet** (Interface) and depends on **Region** (Enumeration).
- RegionRetrieval** (Component) implements **RegionRetrieval** (Interface) and depends on **Region** (Enumeration).
- FavoriteRegionRetrieval** (Component) implements **SkillRetrievalSet** (Interface) and depends on **Region** (Enumeration).
- Region** (Enumeration) lists the following values: KEELUNG, TAIPEI, TAOYUAN, HSIENCHU, MIAOLI, TAICHUNG, CHANGHUA, NANTOU, YUNLIN, CHIAYI, TAINAN, KAOHSIUNG, PINGTUNG, ILAN, HUALIEN, and TAITUNG.
- 首頁模組** (Component) depends on **MatchServer** (Component) and **技能模組** (Component).
- 技能模組** (Component) depends on **MatchMaker** (Interface) and **資料庫模組** (Component).
- 資料庫模組** (Component) depends on **MatchMaker** (Interface).

<b>Class Name</b>	MatchMaker
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	依據配對演算法來取得符合配對標準的技能資訊
<b>Relationship with other classes</b>	MatchServlet、BasicAlgorithm、SkillManager
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

42

<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)
---	--------------------

<b>Class Name</b>	BasicAlgorithm
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	配對演算法的實作
<b>Relationship with other classes</b>	MatchMaker
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

<b>Class Name</b>	RegionMatrixSet
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	依據地區獲得一系列距離表
<b>Relationship with other classes</b>	BasicAlgorithm、RegionDiatanceOrder、Region
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

<b>Class Name</b>	RegionDiatanceOrder
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	依據地區獲得一系列距離表的實作
<b>Relationship with other classes</b>	BasicAlgorithm、RegionDiatanceOrder、Region
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

<b>Class Name</b>	Region
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	存放地區所相對應的數字
<b>Relationship with other classes</b>	RegionDiatanceOrder
<b>Attributes</b>	KEELUNG, TAIPEI, TAoyuan, HSinchu, MIAOLI, TAIHUNG, CHANGHUA, NANTOU, YUNLIN, CHIAYI, TAI-NAN, KAOHSIUNG, PINGTUNG, ILAN, HUALIEN, TAITUNG
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

<b>Class Name</b>	SkillScoreSet
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	計算技能的權重分數
<b>Relationship with other classes</b>	BasicAlgorithm、SumEvalScoreWithFitLvWt
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

<b>Class Name</b>	SumEvalScoreWithFitLvWt
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	計算技能的權重分數實作
<b>Relationship with other classes</b>	SkillScoreSet
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

<b>Class Name</b>	DistanceWeightSet
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	計算技能的距離權重
<b>Relationship with other classes</b>	BasicAlgorithm、NormalizationWeight
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

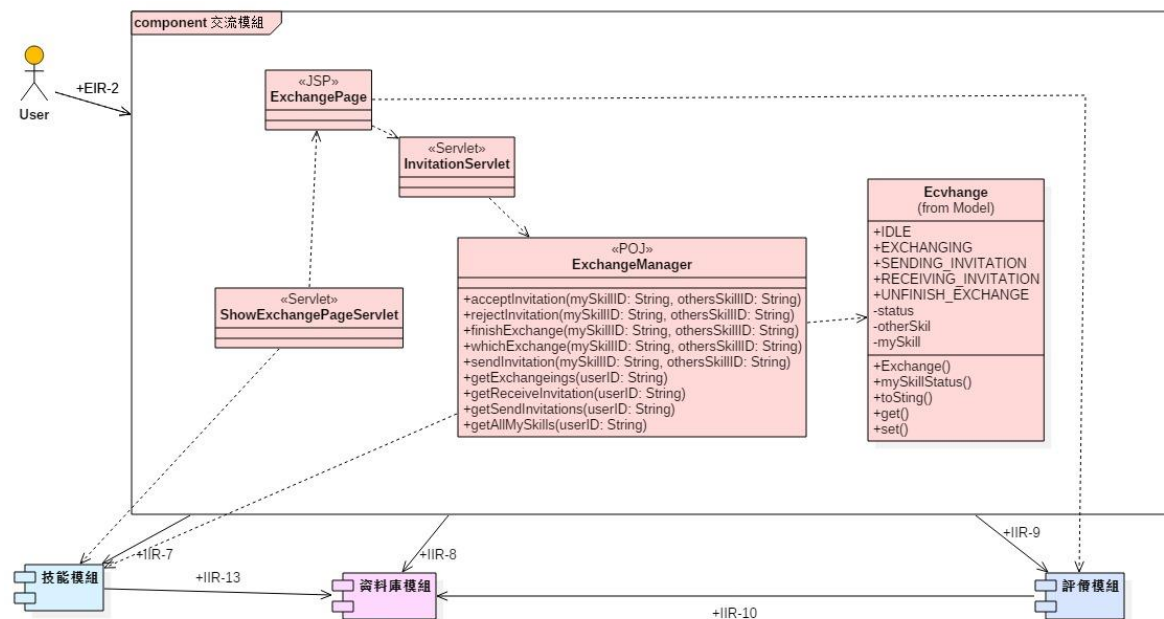
<b>Class Name</b>	NormalizationWeight
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	計算技能的距離權重實作
<b>Relationship with other classes</b>	DistanceWeightSet
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

<b>Class Name</b>	SkillRetriviealSet
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	以興趣與地區撈取資料庫取得技能
<b>Relationship with other classes</b>	BasicAlgorithm、FavoriteRegionRetrieval、RegionRetrieval
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

<b>Class Name</b>	FavoriteRegionRetrieval
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	以興趣與地區撈取資料庫取得技能實作
<b>Relationship with other classes</b>	SkillRetriviealSet
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

<b>Class Name</b>	RegionRetrieval
<b>Category</b>	POJ
<b>Module</b>	配對模組
<b>Description</b>	以地區撈取資料庫取得技能實作
<b>Relationship with other classes</b>	SkillRetriviealSet
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能配對(E-UCA-UC-007)

## 6.6 交流模組



<b>Class Name</b>	ExchangeManagr
<b>Category</b>	POJ
<b>Module</b>	交流模組
<b>Description</b>	管理交流功能；接受與拒絕邀請、交流結束、讀取所有邀請。
<b>Relationship with other classes</b>	ShowExchangePageServlet、InvitationServlet、SkillManager
<b>Attributes</b>	-
<b>Traceability with other use case</b>	交流管理(E-UCA-UC-005)

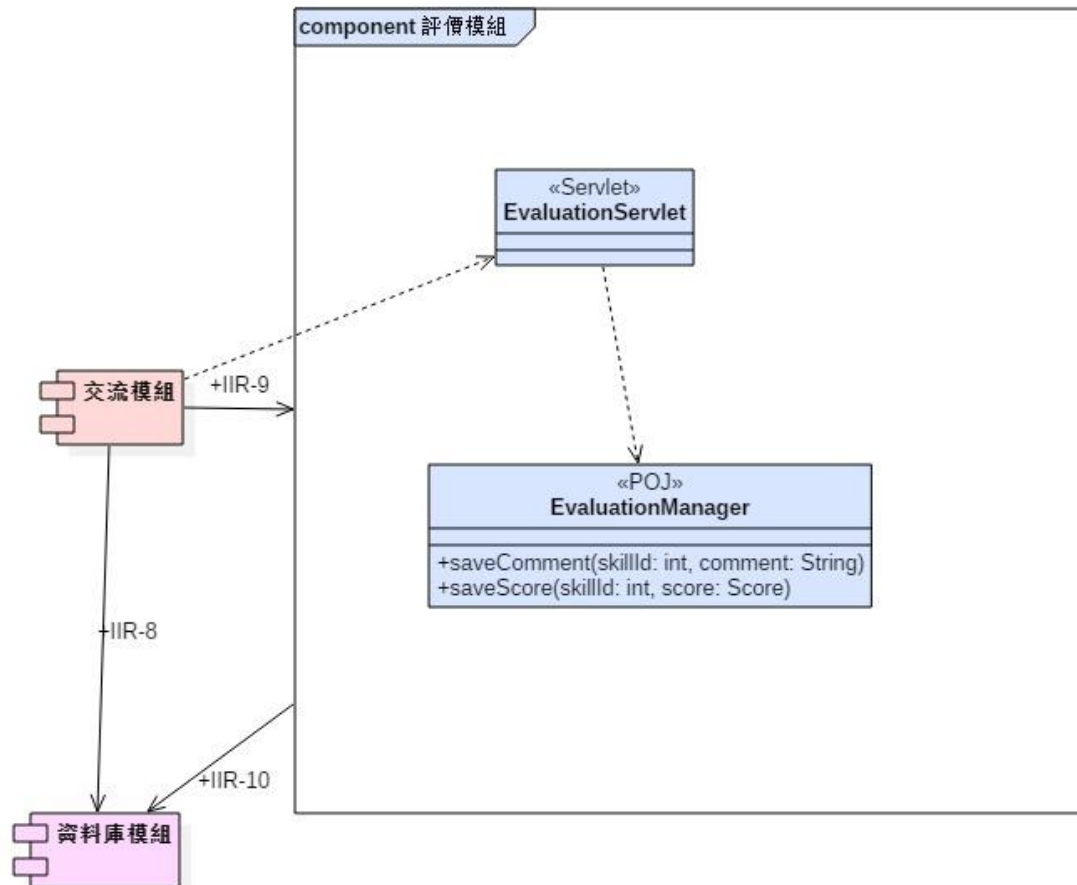
<b>Class Name</b>	ExchangePage
<b>Category</b>	JSP
<b>Module</b>	交流模組
<b>Description</b>	交流頁面：顯示收到的邀請、送出的邀請、進行中的交流
<b>Relationship with other classes</b>	ShowExchangePageServlet、Navigation、InvitationServlet、EvaluationServlet
<b>Attributes</b>	無
<b>Traceability with other use case</b>	技能評價(E-UCA-UC-004)、交流管理(E-UCA-UC-005)

<b>Class Name</b>	ShowExchangePageServlet
<b>Category</b>	Servlet
<b>Module</b>	交流模組
<b>Description</b>	交付使用者資訊並指派 ExchangeManager 取得呈現在 ExchangePage 中的所有資訊
<b>Relationship with other classes</b>	ExchangePage、ExchangeManager
<b>Attributes</b>	無
<b>Traceability with other use case</b>	交流管理(E-UCA-UC-005)

<b>Class Name</b>	InvitationServlet
<b>Category</b>	Servlet
<b>Module</b>	交流模組
<b>Description</b>	取得配對頁面資料後指派 ExchangeManager 送出邀請(存取 DB)
<b>Relationship with other classes</b>	ExchangePage、MatchPage
<b>Attributes</b>	無
<b>Traceability with other use case</b>	交流管理(E-UCA-UC-005)

<b>Class Name</b>	Exchange
<b>Category</b>	POJ
<b>Module</b>	交流模組
<b>Description</b>	紀錄技能之間的交流關係與狀態
<b>Relationship with other classes</b>	ExchangeManager
<b>Attributes</b>	<i>IDLE</i> 、 <i>EXCHANGING</i> 、 <i>SENDING_INVITATION</i> 、 <i>RECEIVING_INVITATION</i> 、 <i>UNFINISH_EXCHANGE</i> 、status、othersSkill、mySkill
<b>Traceability with other use case</b>	交流管理(E-UCA-UC-005)

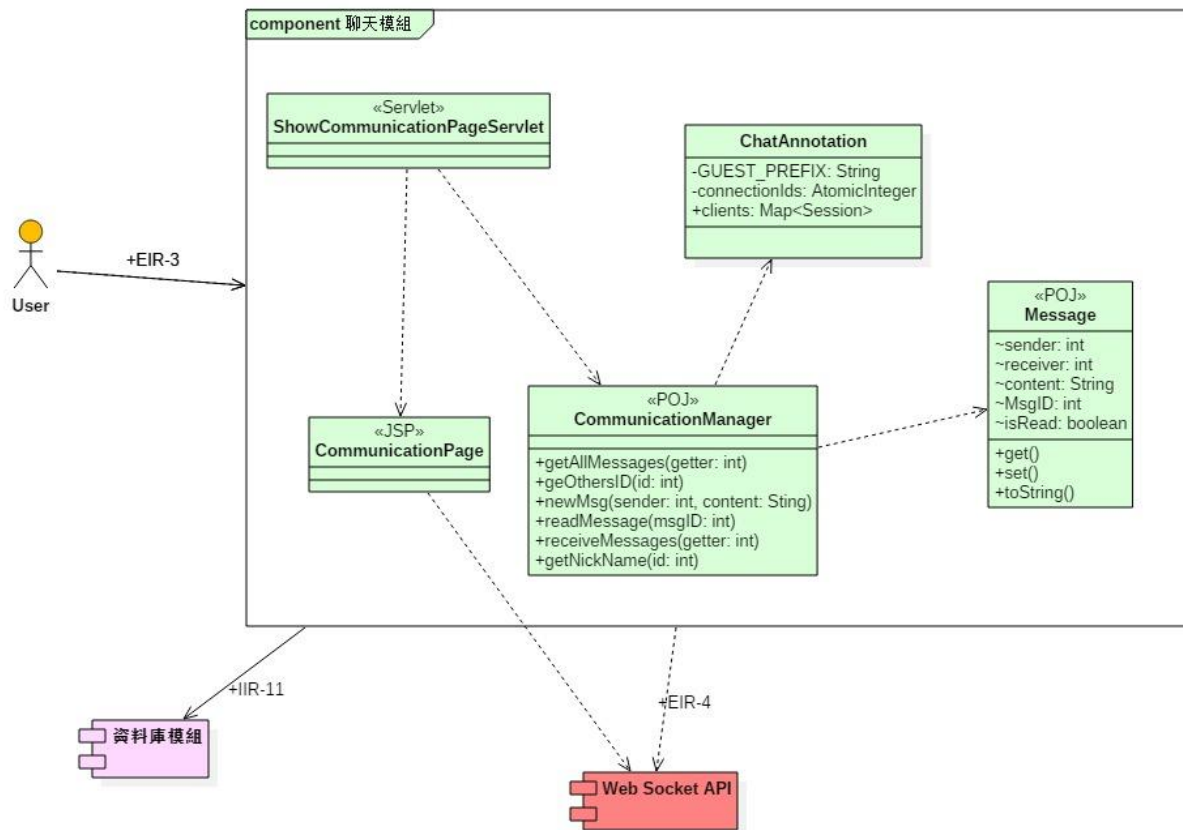
## 6.7 評價模組



<b>Class Name</b>	EvaluationManager
<b>Category</b>	POJ
<b>Module</b>	評價模組
<b>Description</b>	儲存更新評價細項分數、新增評語(存取 DB)。
<b>Relationship with other classes</b>	EvaluationServlet
<b>Attributes</b>	-
<b>Traceability with other use case</b>	技能評價(E-UCA-UC-004)

<b>Class Name</b>	EvaluationServlet
<b>Category</b>	Servlet
<b>Module</b>	評價模組
<b>Description</b>	取得欲評論技能資訊並指派 EvaluationManager 去新增評論
<b>Relationship with other classes</b>	ExchangePage、EvaluationManager
<b>Attributes</b>	無
<b>Traceability with other use case</b>	技能評價(E-UCA-UC-004)

## 6.8 聊天模組



<b>Class Name</b>	Message
<b>Category</b>	POJ
<b>Module</b>	聊天模組
<b>Description</b>	儲存訊息資料。
<b>Relationship with other classes</b>	CommunicationManager
<b>Attributes</b>	sender、receiver、msgID、content、isRead
<b>Traceability with other use case</b>	交流管理(E-UCA-UC-005)

<b>Class Name</b>	CommunicationManager
<b>Category</b>	POJ
<b>Module</b>	聊天模組
<b>Description</b>	取得所有歷史訊息、儲存新訊息(DB)
<b>Relationship with other classes</b>	ShowCommunicationPageServlet、Message
<b>Attributes</b>	-
<b>Traceability with other use case</b>	交流管理(E-UCA-UC-005)

<b>Class Name</b>	CommunicationPage
<b>Category</b>	JSP
<b>Module</b>	聊天模組

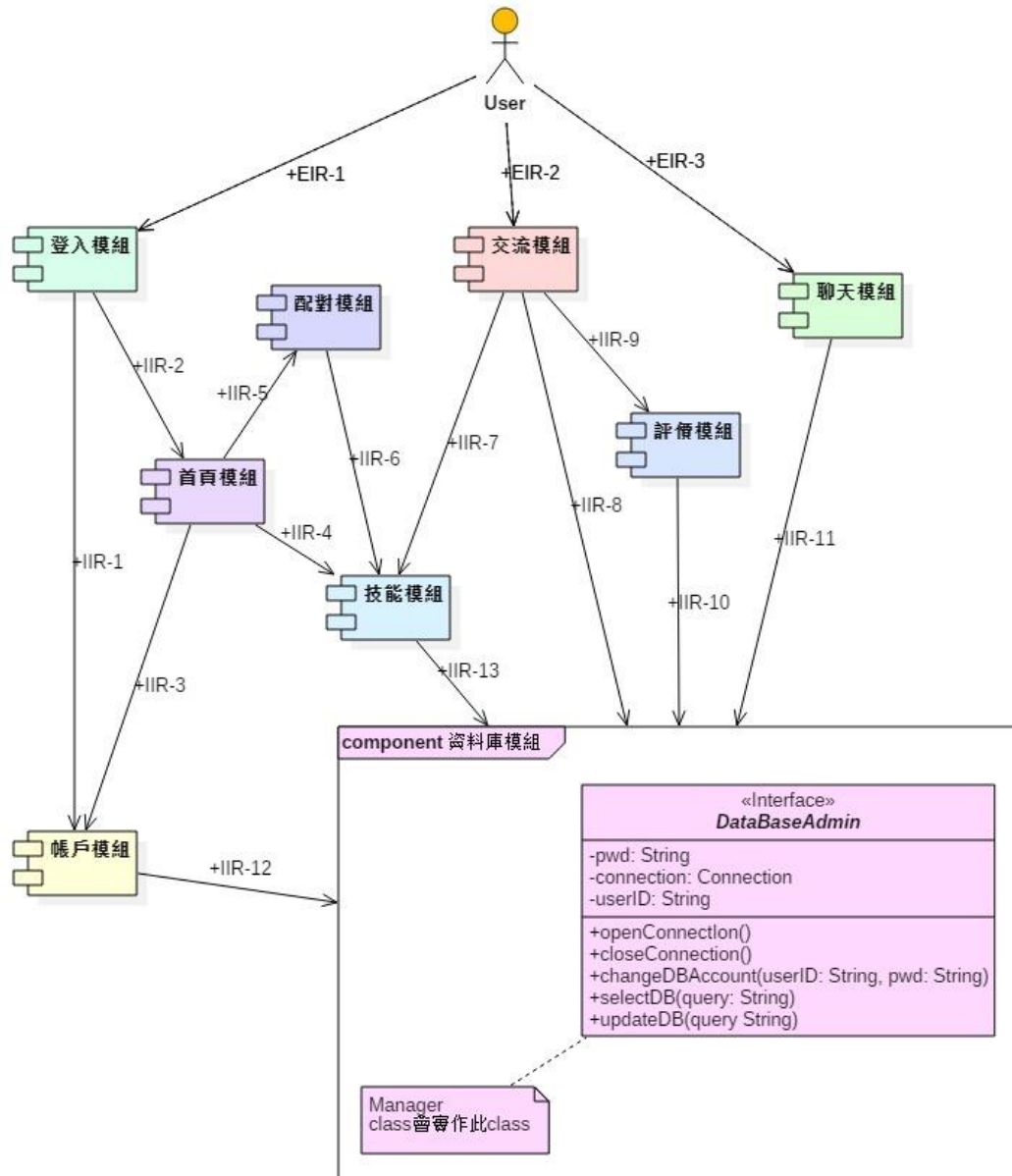


<b>Description</b>	聊天室頁面，顯示所有聊天訊息
<b>Relationship with other classes</b>	ShowCommunicationPageServlet、Navigation
<b>Attributes</b>	無
<b>Traceability with other use case</b>	交流管理(E-UCA-UC-005)

<b>Class Name</b>	ShowCommunicationPageServlet
<b>Category</b>	Servlet
<b>Module</b>	聊天模組
<b>Description</b>	指派 CommunicationManager 取得所有訊息資訊並交由 CommunicationPage 顯示
<b>Relationship with other classes</b>	CommunicationPage、CommunicationManager
<b>Attributes</b>	無
<b>Traceability with other use case</b>	交流管理(E-UCA-UC-005)

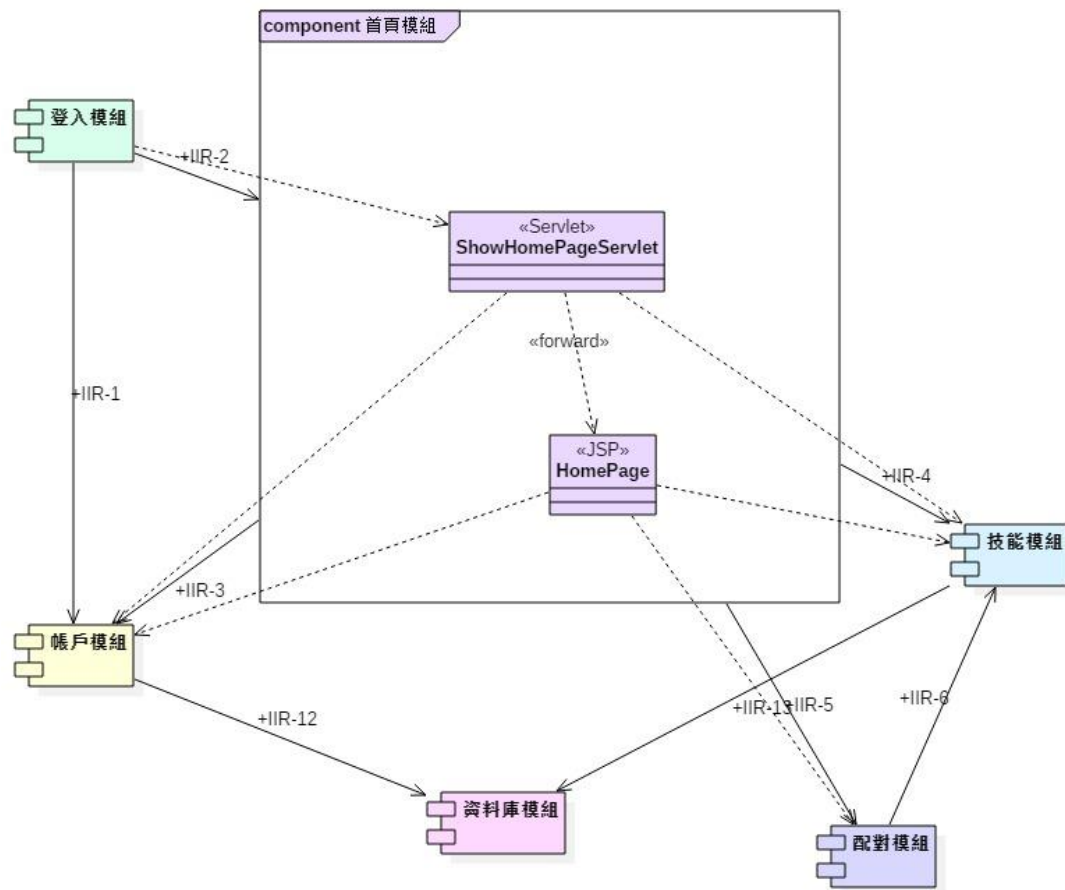
<b>Class Name</b>	ChatAnnotation
<b>Category</b>	Servlet
<b>Module</b>	聊天模組
<b>Description</b>	聊天室的伺服端
<b>Relationship with other classes</b>	CommunicationManager
<b>Attributes</b>	GUEST_PREFIX、connectionIds、clients
<b>Traceability with other use case</b>	交流管理(E-UCA-UC-005)

## 6.9 資料庫模組



<b>Class Name</b>	DataBaseAdmin
<b>Category</b>	POJ (Interface)
<b>Module</b>	資料庫模組
<b>Description</b>	唯一個存取資料庫的介面、由需要存取資料庫的所有 Manager(POJ)來實作
<b>Relationship with other classes</b>	-
<b>Attributes</b>	Connection、pwd、userID
<b>Traceability with other use case</b>	註冊系統(E-UCA-UC-001)、登入系統(E-UCA-UC-002)、技能評價(E-UCA-UC-004)、交流管理(E-UCA-UC-005)、技能管理(E-UCA-UC-006)、技能配對(E-UCA-UC-007)

## 6.10 首頁模組



<b>Class Name</b>	ShowHomePageServlet
<b>Category</b>	Servlet
<b>Module</b>	首頁模組
<b>Description</b>	取得使用者資訊並交付多個 POJ 來取得所有顯示首頁所需資訊
<b>Relationship with other classes</b>	SignUpServlet、LoginServlet、AccountManager、SkillManager、HomePage
<b>Attributes</b>	無
<b>Traceability with other use case</b>	檔案管理(E-UCA-UC-003)、技能評價(E-UCA-UC-004)、技能管理(E-UCA-UC-006)、技能配對(E-UCA-UC-007)

<b>Class Name</b>	HomePage
<b>Category</b>	JSP
<b>Module</b>	首頁模組
<b>Description</b>	個人頁面：顯示個人資訊、擁有技能與狀態、興趣技能
<b>Relationship with other classes</b>	ShowHomePageServlet、Navigation、SkillServlet、AccountServlet、MatchServlet
<b>Attributes</b>	無
<b>Traceability with other use case</b>	檔案管理(E-UCA-UC-003)、技能評價(E-UCA-UC-004)、技能管理(E-UCA-UC-006)、技能配對(E-UCA-UC-007)

## 7. 實作技術(Implementation Languages and Platforms)

本系統預計以網站為系統平台，採用響應式網頁(Responsive Web Design, RWD)的技術，提供使用者可以在多種裝置上使用本網站。本系統預計採用 Tomcat 作為平台伺服器的基礎，以 MVC(Model-View-Controller)的概念建構模組化的系統架構。Tomcat 是在 Java 環境下執行的網頁伺服軟體架構，因此，本系統將以 Java 程式語言撰寫物件化的網站系統，其包含以 POJ、JSP、Servlet 分別實現 MVC 概念的 Model、View 以及 Controller，並透過 Tomcat 本身規劃的物件化方式傳遞封包、交付封包與轉送請求。系統後台之資料庫建構為 MySQL 環境，系統預計使用 JDBC 套件連接 MySQL 資料庫，並透過 SQL 的方式取得所需的資料集。平台前端將採用開源的前端樣板，再應用 Bootstrap、jQuery 等前端框架輔助調整以符合本系統的功能與架構。

## 8. 設計議題(Design Issue)

### 8.1 議題一

討論日期：2016/10/21

問題定義：系統進行配對的基本單位為何？

解決方案：

#### (1) 方案 A

以帳號為導向進行配對，對象資訊包含大頭貼、年齡、地區、性別等個人資訊，另外也顯示對象的所有技能，幫助從多個技能中考量是否要配對。

#### (2) 方案 B

以技能為導向進行配對，配對資訊為技能資訊，包含技能、等級、各項評比的分數等，而不過分顯示對方的個人資訊，但是地區資訊除外。

最終定案：方案 B

定案理由：依照系統**技能交換**的初衷，也避免選擇方案 A 而與交友系統太過雷同。

### 8.2 議題二

討論日期：2016/10/21

問題定義：系統要採取何種配對模式？

解決方案：

#### (1) 方案 A

即時配對。使用技能的配對功能後，系統隨機尋找線上也正使用配對功能的技能，之後雙方便可以直接開始交流，直到其中一人離開連線。

#### (2) 方案 B

非即時配對。使用技能的配對功能後，系統根據配對演算法在所有技能中尋找較適配的技能，依使用者發出配對邀請，待對方核准後才可以交流。

最終定案：方案 B

定案理由：

一是避免採取方案 A 而因為使用人數太少導致等待過久，二來遵循**技能交換**的宗旨，期望加深交換技能的實質意義，而非無助益於實現技能交換的方式。

### 8.3 議題三

討論日期：2016/10/21

問題定義：配對技能的篩選方法為何？

解決方案：

#### (1) 方案 A

配對前使用者設定 HTML 的篩選器，項目包含技能類別等篩選條件。

#### (2) 方案 B

使用者新增興趣技能資訊於個人資料，系統使用設計好的配對演算法自動取得所需的條件資料進行篩選。

最終定案：方案 B

定案理由：配對篩選自動化可以減少使用者使用系統的負擔。

### 8.4 議題四

討論日期：2016/10/21

問題定義：配對演算法如何定義？

解決方案：

#### (1) 方案 A

配對演算法會先透過地區、興趣技能與最近一次上線時間在資料庫上進行篩選，篩選過程中，會優先過濾超過三天為上限的使用者，再依照使用者所新增的興趣技能進行挑選，而後是地區，進而篩選出 50 組配對技能，最後利用技能各項分數搭配權重進行排序，作為配對的先後順序。

順序：

- 1 扣除三天未上線的技能(資料庫)
- 2 依照興趣與距離相近的 Select(資料庫)
- 3 依照分數、距離順序計算權重分數，然後排序(array)

距離係數：

以距離表上的距離資訊排序後，依照距離遠近等分比例後得出距離係數舉例來說：

假設基隆、台北、台北、宜蘭與使用者的所在地於距離表上顯示 0,1,1,2

則三地點的距離係數可以計算成  $1/3, 1/4, 1/4, 1/6$

而計算方式：

集合中所有卡片數為距離係數的值，必須先以距離排序，若距離不同則距離係數-1，再將所有距離係數加總為分母，個別距離係數為分子，將所有的地點依比例等分

集合中所有卡片數：4

基隆：4

台北：4 - 1 = 3

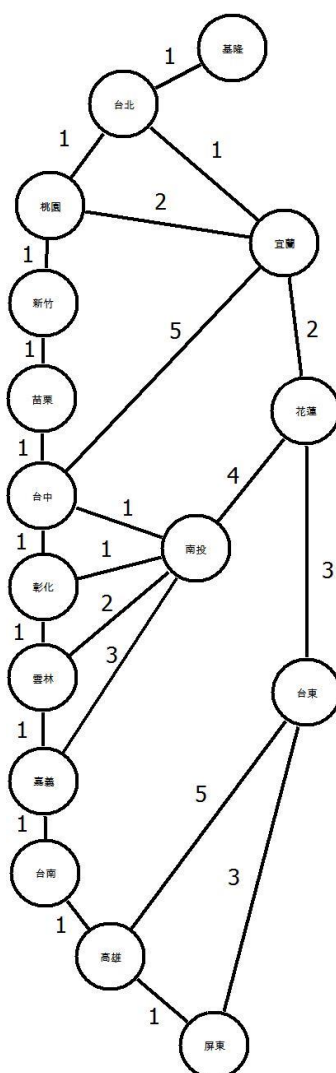
台北：4 - 1 = 3

宜蘭：4 - 2 = 2

分母：4 + 3 + 3 + 2 = 12

基隆：4/12、台北：3/12、台北：3/12、宜蘭：2/12

縣市距離：



路程距離：

縣市	基隆	台北	桃園	新竹	苗栗	台中	彰化	南投	雲林	嘉義	台南	高雄	屏東	宜蘭	花蓮	台東
基隆	0	1	2	3	4	5	6	6	7	8	9	10	11	2	4	7
台北	1	0	1	2	3	4	5	5	6	7	8	9	10	1	3	6
桃園	2	1	0	1	2	3	4	4	5	6	7	8	9	2	4	7
新竹	3	2	1	0	1	2	3	3	4	5	6	7	8	3	5	8
苗栗	4	3	2	1	0	1	2	2	3	4	5	6	7	4	6	9
台中	5	4	3	2	1	0	1	1	2	3	4	5	6	5	5	8
彰化	6	5	4	3	2	1	0	1	1	2	3	4	5	6	5	8
南投	6	5	4	3	2	1	1	0	2	3	4	5	6	6	4	7
雲林	7	6	5	4	3	2	1	2	0	1	2	3	4	7	6	7
嘉義	8	7	6	5	4	3	2	3	1	0	1	2	3	8	7	6
台南	9	8	7	6	5	4	3	4	2	1	0	1	2	9	8	5
高雄	10	9	8	7	6	5	4	5	3	2	1	0	1	9	7	4
屏東	11	10	9	8	7	6	5	6	4	3	2	1	0	8	6	3
宜蘭	2	1	2	3	4	5	6	6	7	8	9	9	8	0	2	5
花蓮	4	3	4	5	6	5	5	4	6	7	8	7	6	2	0	3
台東	7	6	7	8	9	8	8	7	7	6	5	4	3	5	3	0

權重：（依照對方技能卡分數之平均去乘）

假設自己是一張新卡，教學態度、教學頻率、整體滿意度權重設為 3。

假設對方是一張新卡，則不在此限。

教學態度 權重 1,2,3,4,5 以自己教學態度平均(無條件捨去)星等改變權重

技能程度 權重：2,4 如果已擁有該技能權重為 4,反之權重為 2

教學技巧 權重：2,4 如果已擁有該技能權重為 2,反之權重為 4

教學頻率 權重：3,1 依照等級來調整，0-5 等權重為 3,其他權重為 1

整體滿意度 權重：1,2,3,4,5 以自己整體滿意度平均星等改變權重

最終定案：方案 A

定案理由：讓使用者盡可能的配對到地區相近且符合自己想學的技能。

範例：

技能卡(自己)

持有者：Justin

等級：LV1

教學態度分數：0

整體滿意度：0



居住地：基隆

擁有技能：吉他、英文、爵士鼓

感興趣技能：中文、吉他

編號、技能卡、居住地、教學態度、技能程度、教學技巧、教學頻率、整體滿意度

[001、中文、基隆、3、3、3、3、3]

[002、中文、台北、4、2、3、5、3]

[003、中文、宜蘭、3、2、3、2、3]

[004、吉他、基隆、4、2、3、3、4]

[005、吉他、台中、5、5、5、5、5]

[006、吉他、台東、1、1、1、1、1]

[007、吉他、基隆、0、0、0、0、0]

[008、英文、基隆、3、4、3、4、3]

先挑興趣技能符合且距離查表為 0 的：001、004、007

因為不滿 50 筆，再將查表距離為 1 且符合興趣技能的:002

因為不滿 50 筆，再將查表距離為 2 且符合興趣技能的:003

依然不滿 50 筆，依此類推:005、006

由於資料庫內已經沒有符合興趣的技能，所以目前得到的技能集合有:001、002、003、004、005、006、007。

$(\text{教學態度權重係數} \times \text{對方教學態度平均} + \text{技能程度權重係數} \times \text{對方技能程度平均} + \text{教學技巧權重係數} \times \text{對方教學技巧平均} + \text{教學頻率權重係數} \times \text{對方教學頻率平均} + \text{整體滿意度權重係數} \times \text{對方整體滿意度分數}) \times \text{距離係數}$

因為自己為新卡，教學態度權重係數為 3，整體滿意度權重係數為 3；

因為此卡等級為 1，教學頻率權重係數為 3。

距離權重係數：

計算集合所有卡片數:7

所有距離差:0,0,0,1,2,5,7

距離係數:7,7,7,6,5,4,3

加總為分母:39

001 :

因為自己沒有中文技能，技能程度權重係數為 2，教學技巧權重係數為 4。

距離權重係數:7/39

$$(3*3+2*3+4*3+3*3+3*3)* 7/39 = 8.07$$

002 :

因為自己沒有中文技能，技能程度權重係數為 2，教學技巧權重係數為 4。

距離權重係數:6/39

$$(3*4+2*2+4*3+3*5+3*3)* 6/39 = 8$$

003 :

因為自己沒有中文技能，技能程度權重係數為 2，教學技巧權重係數為 4。

距離權重係數:5/39

$$(3*3+2*2+4*3+3*2+3*3)* 5/39 = 5.12$$

004 :

因為自己有吉他技能，技能程度權重係數為 4，教學技巧權重係數為 2。

距離權重係數:7/39

$$(3*4+4*2+2*3+3*3+3*4)* 7/39 = 8.43$$

005 :

因為自己有吉他技能，技能程度權重係數為 4，教學技巧權重係數為 2。

距離權重係數:4/39

$$(3*5+4*5+2*5+3*5+3*5)* 4/39 = 7.69$$

006 :

因為自己有吉他技能，技能程度權重係數為 4，教學技巧權重係數為 2。

距離權重係數:3/39

$$(3*1+4*1+2*1+3*1+3*1)* 3/39 = 1.15$$

007 :

因為自己有吉他技能，技能程度權重係數為 4，教學技巧權重係數為 2。

距離權重係數:7/39

$$(3*0+4*0+2*0+3*0+3*0)* 7/39 = 0$$

依照分數排序

004>001>002>005>003>006> 007

## 8.5 議題五

討論日期：2016/10/21

問題定義：聊天室的訊息的時效性為何？

解決方案：

### (1) 方案 A

保存聊天紀錄，交流結束後刪除所有先前的聊天紀錄。

### (2) 方案 B

不保存聊天紀錄，關閉聊天視窗將失去所有訊息。

最終定案：方案 A

定案理由：

選擇方案 A 符合**技能交換**的主題，是有益於提升交流的連貫性，例如進行了很多次交流後，如果忘記先前幾次交流的特定內容，則可以查詢過去的聊天紀錄。

## 8.6 議題六

討論日期：2016/11/04

問題定義：技能建立數量是否要進行限制？

解決方案：

### (1) 方案 A

限制每個帳號一開始只可以新增三個技能，交流後就不可刪除，且不可以重複創建同項技能，且不再增加數量。

### (2) 方案 B

限制每個帳號一開始只可以新增三個技能，交流後就不可刪除，且不可以重複創建同項技能，且任一個技能上升 5 個等級就增加一個新增技能欄位。

### (3) 方案 C

限制每個帳號一開始只可以新增三個技能，可刪除技能，且可以重複創建同項技能，且任一個技能上升 5 個等級就增加一個新增技能欄位。

最終定案：方案 B

定案理由：

讓使用者在使用技能的時候，會較專注培養於擅長的技能，且能夠增加使用者使用 EXCHANGE 的興趣，而不選擇方案三的原因是因為重複創建同個技能，一個得了差評就創建一個新的，反而會使培養同樣的技能變得沒有意義，我們設計的初衷是希望使用者可以用真正了解與熟悉自己本來就會的技能，甚至能夠學會新的技能，而不是為了洗評價而創建同樣的技能。

## 8.7 議題七

討論日期：2016/11/04

問題定義：評價方案

解決方案：

(1) 方案 A

只有一個綜合評價，分級為 1-5 個評價分。

(2) 方案 B

分為五個評價類別，有教學態度、技能程度、教學技巧、教學頻率、整體滿意度等，每項 5 個評價分，共 25 個評價分。

最終定案：方案 B

定案理由：

方案二的評價方式比較全面，不會讓人在交流後不知道自己的問題在哪，有利於在下次交流可以改善缺點。

## 8.8 議題八

討論日期：2016/11/18

問題定義：在類別圖的設計上，個人資料是否與密碼分開？

解決方案：

(1) 方案 A

不分開。

(2) 方案 B

分開。

最終定案：方案 B

定案理由：

從資訊安全的角度去看，兩者分開較為安全，例如顯示個人資料的功能不用一併取得密碼資料。

## 8.9 議題九

討論日期：2016/11/18

問題定義：經驗條與等級顯示與否？

解決方案：

(1) 方案 A

顯示經驗條與等級。

(2) 方案 B

僅顯示等級。

最終定案：方案 B

定案理由：

加上經驗條，版面配置太壅擠，於美工考量下決定僅顯示等級而不顯示經驗條。

#### 8.10 議題十

討論日期：2016/11/18

問題定義：等級定義方案

解決方案：

(1) 方案 A

每 50 個評價分一個等級。

(2) 方案 B

第一次升級僅需要 15 個評價分，之後每次升級需 25 個評價分。

最終定案：方案 B

定案理由：比較好讓新手去適應等級成長，不會有升第一級要很久的感覺。

#### 8.11 議題十一

討論日期：2016/11/28

問題定義：懲罰制度方案

解決方案：

(1) 方案 A

連續兩次評價分數 10 分以下，就永久封鎖該技能。

(2) 方案 B

連續兩次評價分數 5 分，封鎖三天，且累積一次警告，再連續兩次評價 5 分，封鎖七天，累積二次警告，累積三次警告該技能永久封鎖。

(3) 方案 C

經人舉報即封鎖該帳號。

最終定案：方案 A

定案理由：

投票表決的結果。這樣懲罰力度才夠，不會讓惡意使用者太過放肆，而不使用方案 C 的原因是因為人都會犯錯，如果一次舉報就會讓他被封鎖，這樣會沒有讓人改過自新的機會。

## 8.12 議題十二

討論日期：2016/11/28

問題定義：活躍程度加入演算法與否？

解決方案：

(1) 方案 A

加入上次登入時間的紀錄到資料庫中，供配對演算法運算使用。若三天沒上線便不會被配對演算法挑選到。

最終定案：方案 A

定案理由：

如果有人很多天沒上，這樣配對到他的人會變成發送邀請卻遲遲沒有人審核，這樣是一種很浪費時間的行為，為了增加技能交流的效率，所以加入活躍程度到演算法中。

## 8.13 議題十三

討論日期：2016/11/28

問題定義：使用者發送配對邀請後，要經過多久沒被審核才會失效？

解決方案：

(1) 方案 A

三天。

(2) 方案 B

一天。

最終定案：方案 A

定案理由：

一天都沒回應的話，就使邀請失效，會比較有利於頻繁使用的用戶，讓他們可以在盡量短的時間得到回覆，若沒有回應就盡快去重新配對一個新的對象。