

# 私人徵才

## 設計文件

專案名稱	私人徵才
撰寫日期	2022/12/06
發展者	江仲恩、黃煥廷、羅思遠、林甯 堃、李汪慶

(紅字均為說明，都可刪除)  
(藍字均為範例，都應刪除或可做  
適度修改)

## 版次變更記錄

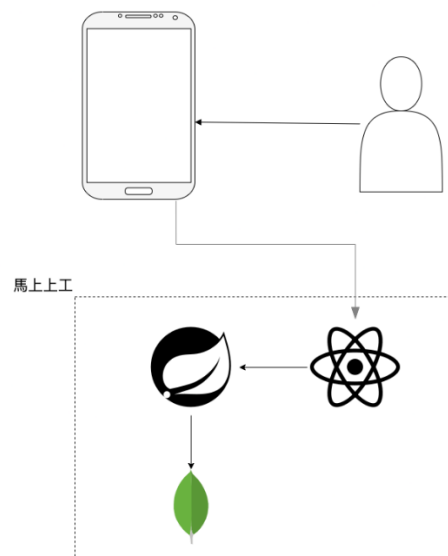
版次	變更項目	變更日期
0.1	初版	2021/11/30
0.2	修正類別圖 修正資料庫設計	2021/12/06
0.3		
0.4		
0.5		
0.6		
1.0		

## 目錄

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) .....	7
3. 流程設計(PROCESS DESIGN) .....	20
4. 使用者畫面設計(USER INTERFACE DESIGN) .....	26
5. 資料設計(DATA DESIGN) .....	46
6. 類別圖設計(CLASS DIAGRAM) .....	49
7. 實作技術(IMPLEMENTATION LANGUAGES AND PLATFORMS) .....	50
8. 設計議題(DESIGN ISSUE) .....	50

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

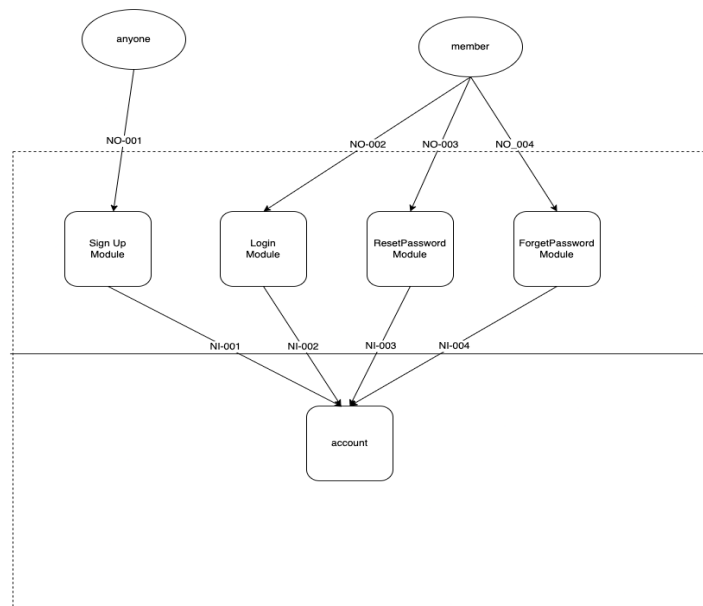
系統高階架構：



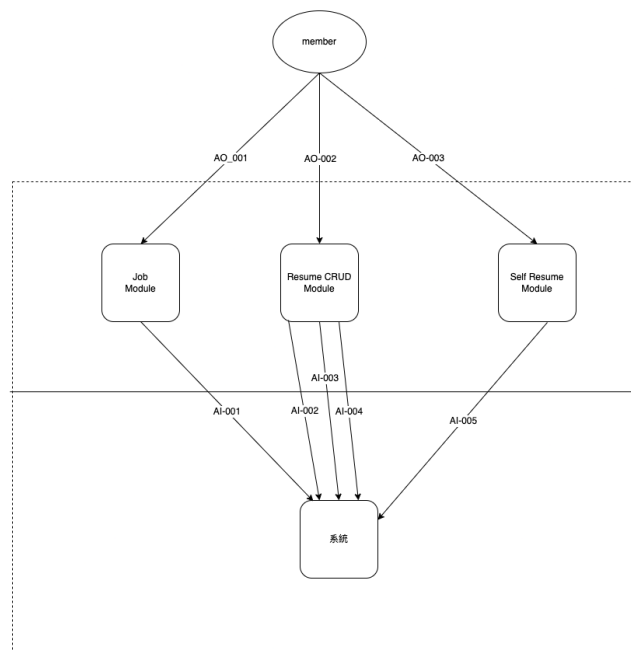
1. 使用者透過手機使用本系統
2. 前端使用 React-Native 實作
3. 後端使用 Spring Boot 實作
4. 資料庫使用 MangoDB

系統內部架構：

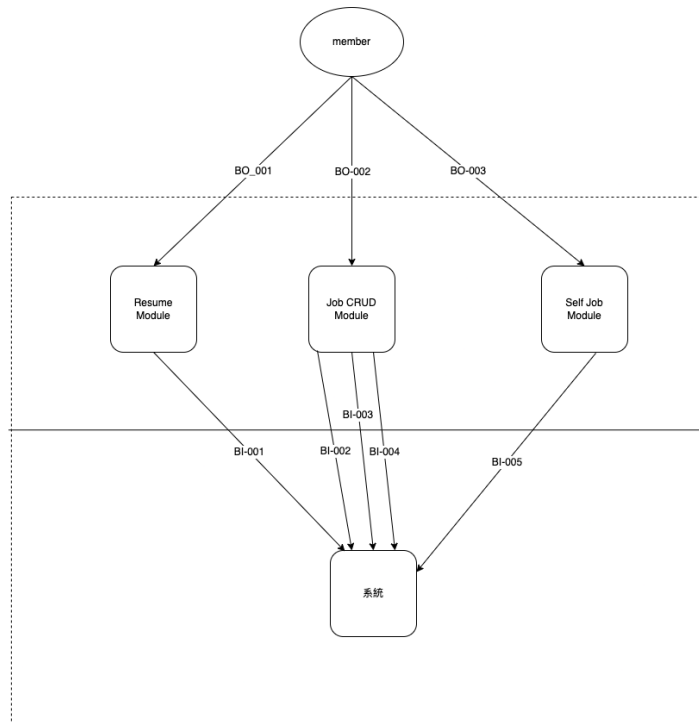
a. 一般功能：包含登入、註冊、忘記密碼、重設密碼



b. 應徵者功能：工作需求、新增履歷、修改履歷、查看履歷、



c. 雇主功能：查看履歷、工作需求、新增工作需求、修改工作需求



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

### 一般功能

#### 外部介面

介面編號	介面名稱	介面提供者	介面使用者
NO-001	註冊	Sign Up Module	Anyone
連結方式	輸入資料		輸出資料
	帳號、名稱、密碼、確認密碼、信箱、驗證碼		註冊成功後提示到信箱接收驗證碼
對應之介面需求			
確認密碼跟再次輸入密碼是否一致，再交給系統驗證帳號是否重複，成功則交給系統新增會員到資料庫，系統再委託 Mail 送出驗證碼			

介面編號	介面名稱	介面提供者	介面使用者
NO-002	登入	Login Module	member
連結方式	輸入資料		輸出資料
	帳號、密碼		登入成功後跳轉 到應徵者畫面
對應之介面需求			
接收輸入資料交給系統驗證是否正確，如果正確生成 token 給前端			

介面編號	介面名稱	介面提供者	介面使用者
NO-003	重設密碼	ResetPassword Module	member
連結方式	輸入資料		輸出資料
	舊密碼、新密碼、再次確認密碼		修改後跳轉到會員頁面
對應之介面需求			
接收輸入資料交給後端驗證舊密碼是否正確，以及新密碼與再次確認密碼是否一致，都符合的話系統更新資料庫			

介面編號	介面名稱	介面提供者	介面使用者
NO-004	忘記密碼	ForgetPassword Module	member
連結方式	輸入資料		輸出資料
	帳號、信箱		提示系統更新密碼傳送至信箱
對應之介面需求			
系統驗證帳號與信箱是否與註冊一致，若一致將資料庫的密碼更新成亂碼，並傳送信箱給使用者			



## 內部介面

介面編號	介面名稱	介面提供者	介面使用者
NI-001	/register	account	Sign Up Module
連結方式	輸入資料		輸出資料
HTTP POST	{  "username" : "String",  "userID" : "String",  "email" : "String",  "password" : "String"  }	{  Status: "200"  }	
對應之介面需求			
將註冊資料傳送給系統			

介面編號	介面名稱	介面提供者	介面使用者
N1-002	/login	account	Login Module
連結方式	輸入資料		輸出資料
HTTP POST	{ "userlD" : "String", "password" : md5("String") }	{ "token" : "String" "status" : "200" }	
對應之介面需求			
將登入資訊傳給系統進行資料驗證，成功系統生成並返回 token			

介面編號	介面名稱	介面提供者	介面使用者
NI-003	/auth/resetpassword	account	ResetPassword Module
連結方式	輸入資料		輸出資料
HTTP POST	{ "userID" : "String", "oldPassword" : "String", "newPassword" : "String" }	{ Status: "200" }	
對應之介面需求			
傳送新舊密碼，交給系統更新資料庫			

介面編號	介面名稱	介面提供者	介面使用者
NI-004	/forgetpassword	account	ForgetPassword Module
連結方式	輸入資料		輸出資料
HTTP POST	{ "userID" : "String", "email" : "String" }		{ Status: "200" }
對應之介面需求			
傳送 userID 與 email 確認是否一致，一致傳送 email 給 user email			

## 應徵者功能

### 外部介面

介面編號	介面名稱	介面提供者	介面使用者
AO-001	瀏覽工作需求	Job Module	Member(應徵者模式)
連結方式	輸入資料		輸出資料
			顯示所有的工作需求
對應之介面需求			
顯示所有的工作需求			

介面編號	介面名稱	介面提供者	介面使用者
AO-002	履歷 CRUD	Resume CRUD Module	Member(應徵者模式)
連結方式	輸入資料		輸出資料
	點擊新增、修改、刪除		顯示更新後的履歷資料
對應之介面需求			
Resume CRUD Module 傳送資料給系統更新資料庫			

介面編號	介面名稱	介面提供者	介面使用者
AO-003	瀏覽自己的履歷	Self Resume Module	Member(應徵者模式)
連結方式	輸入資料		輸出資料
			顯示應徵者自己所新增的履歷
對應之介面需求			
顯示應徵者自己所新增的履歷			

### 內部介面

介面編號	介面名稱	介面提供者	介面使用者
AI-001	/auth/Jobs	系統	Job Module
連結方式	輸入資料		輸出資料
HTTP GET			{  "Job" : "Array["JobObject"]"  }
對應之介面需求			
將 Job List Object 的所有資料傳遞給 Job Module			

介面編號	介面名稱	介面提供者	介面使用者
AI-002	/auth/Resumes	系統	Resume CRUD Module
連結方式	輸入資料		輸出資料
HTTP POST	{ "title" : "String"		{ Status: "200"

	<pre> "name" : "String" "sex" : "String" "birth" : "String" "phoneNumber" : "String" "email" : "String" "school" : "String" "department" : "String" "status" : "String" "year" : "String" "nature" : "String" "type" : "String" "time" : "String" "salary" : "String" "region" : "String" "introduction" : "String" "id" : "String" "user" : "String" "createTime" : "String" "refreshTime" : "String" "order" "int" } </pre>	}
對應之介面需求		
系統創立新的一筆履歷，並更新給 Resume CRUD Module		

介面編號	介面名稱	介面提供者	介面使用者
AI-003	/auth/Resumes/{user}/{order}	系統	Resume CRUD Module
連結方式	輸入資料		輸出資料
HTTP PUT	{user} : "String"  {order} : "int"		{  Status: "200"  }
對應之介面需求			
系統修改指定的履歷，並更新給 Resume CRUD Module			

介面編號	介面名稱	介面提供者	介面使用者
AI-004	/auth/Resumes/{user}/{order}	系統	Resume CRUD Module
連結方式	輸入資料		輸出資料
HTTP DELETE	{user} : "String"  {order} : "int"		{  Status: "200"  }
對應之介面需求			
系統刪除指定的履歷，並更新給 Resume CRUD Module			

介面編號	介面名稱	介面提供者	介面使用者
AI-005	/auth/Resumes/{user}	系統	Self Resume Module
連結方式	輸入資料		輸出資料
HTTP GET	{user} : "String"		[[ "Resume" : "Array["ResumeObject"]" ]]
對應之介面需求			
將指定使用者的 Resume List Object 的所有資料傳遞給 Self Resume Module			

## 雇主功能

### 外部介面

介面編號	介面名稱	介面提供者	介面使用者
BO-001	瀏覽履歷	Resume Module	Member(雇主模式)
連結方式	輸入資料		輸出資料
			顯示所有的履歷
對應之介面需求			
顯示所有的履歷			

介面編號	介面名稱	介面提供者	介面使用者
BO-002	工作需求 CRUD	Job CRUD Module	Member(雇主模式)
連結方式	輸入資料		輸出資料
	點擊新增、修改、刪除		顯示更新後的工作需求資料
對應之介面需求			
Job CRUD Module 傳送資料給系統更新資料庫			

介面編號	介面名稱	介面提供者	介面使用者
BO-003	瀏覽自己的工作需求	Self Job Module	Member(雇主模式)
連結方式	輸入資料		輸出資料
			顯示雇主自己所新增的工作需求
對應之介面需求			
顯示雇主自己所新增的工作需求			



## 內部介面

介面編號	介面名稱	介面提供者	介面使用者
BI-001	/auth/Resumes	系統	Resume Module
連結方式	輸入資料		輸出資料
HTTP GET			{ "Resume" : "Array["ResumeObject"]" }
對應之介面需求			
將 Resume List Object 的所有資料傳遞給 Resume Module			

介面編號	介面名稱	介面提供者	介面使用者
BI-002	/auth/Jobs	系統	Job CRUD Module
連結方式	輸入資料		輸出資料
HTTP POST	<pre>{   "title" : "String"   "name" : "String"   "sex" : "String"   "phoneNumber" : "String"   "email" : "String"   "nature" : "String"   "type" : "String"   "content" : "String"   "date" : "List&lt;String&gt;"   "time" : "List&lt;String&gt;" }</pre>		<pre>{   Status: "200" }</pre>

	<pre> "salary" : "String" "region" : "String" "salaryMethod" : "String" "salaryDate" : "String" "id" : "String" "user" : "String" "createTime" : "String" "refreshTime" : "String" "order" : "int" } </pre>	
對應之介面需求		
系統創立新的一筆工作需求，並更新給 Job CRUD Module		

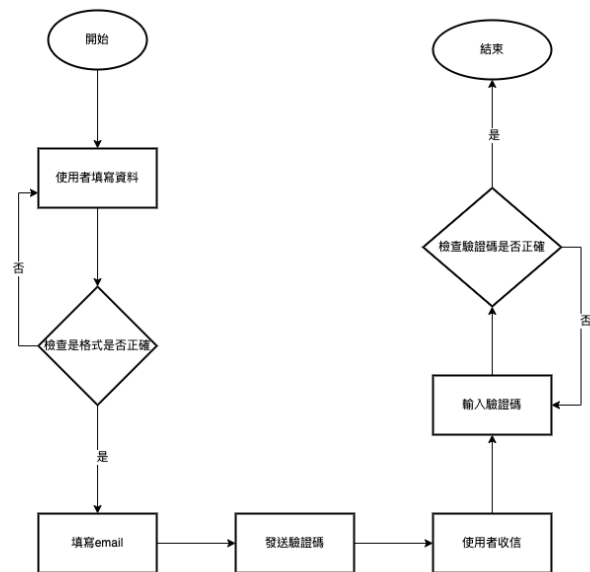
介面編號	介面名稱	介面提供者	介面使用者
BI-003	/auth/Jobs/{user}/{order}	系統	Job CRUD Module
連結方式	輸入資料		輸出資料
HTTP PUT	{user} : "String"  {order} : "int"		{  Status: "200"  }
對應之介面需求			
系統修改指定的工作需求，並更新給 Job CRUD Module			

介面編號	介面名稱	介面提供者	介面使用者
BI-004	/auth/Jobs/{user}/{order}	系統	Job CRUD Module
連結方式	輸入資料		輸出資料
HTTP DELETE	{user} : "String"  {order} : "int"		{  Status: "200"  }
對應之介面需求			
系統刪除指定的工作需求，並更新給 Job CRUD Module			

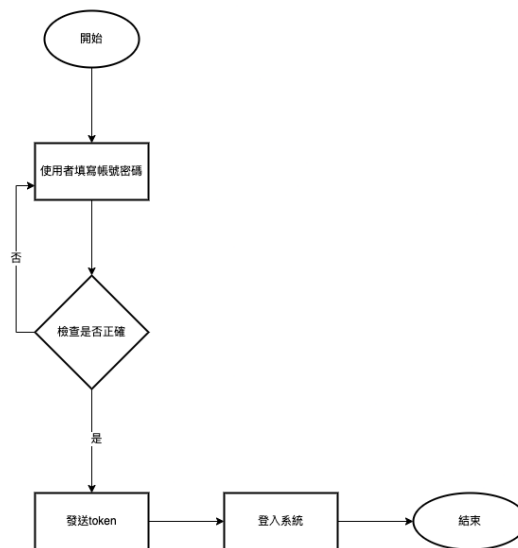
介面編號	介面名稱	介面提供者	介面使用者
BI-005	/auth/Jobs/{user}	系統	Self Job Module
連結方式	輸入資料		輸出資料
HTTP GET	{user} : "String"	[{  "Job" : "Array["JobObject"]"  }]	
對應之介面需求			
將指定使用者的 Job List Object 的所有資料傳遞給 Self Job Module			

### 3. 流程設計(Process Design)

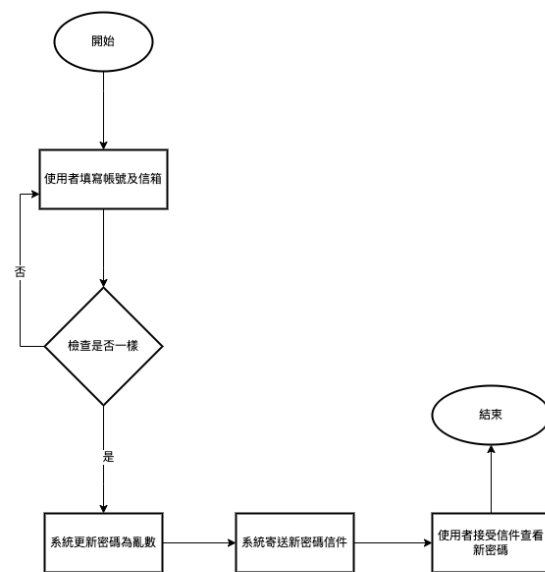
#### 1. 註冊流程：



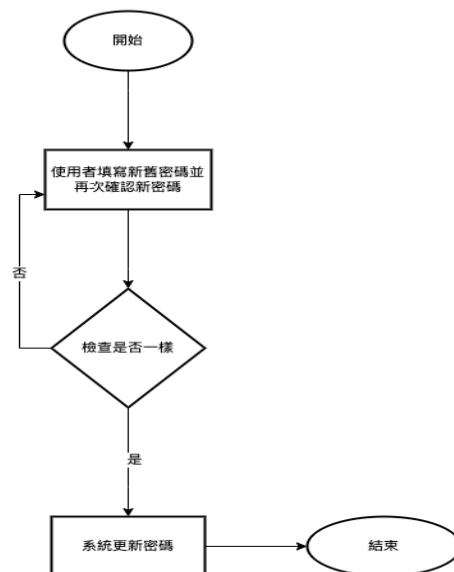
#### 2. 登入流程：



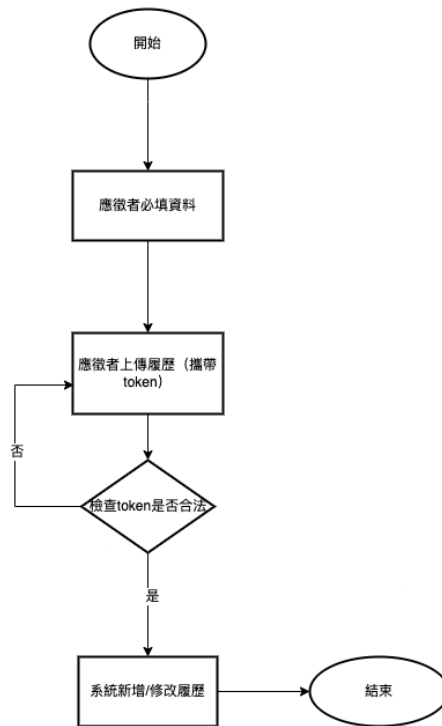
### 3. 忘記密碼：



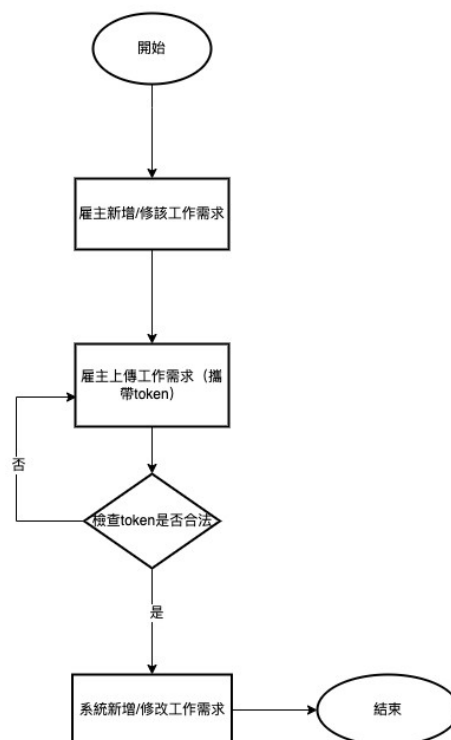
### 4. 重設密碼：



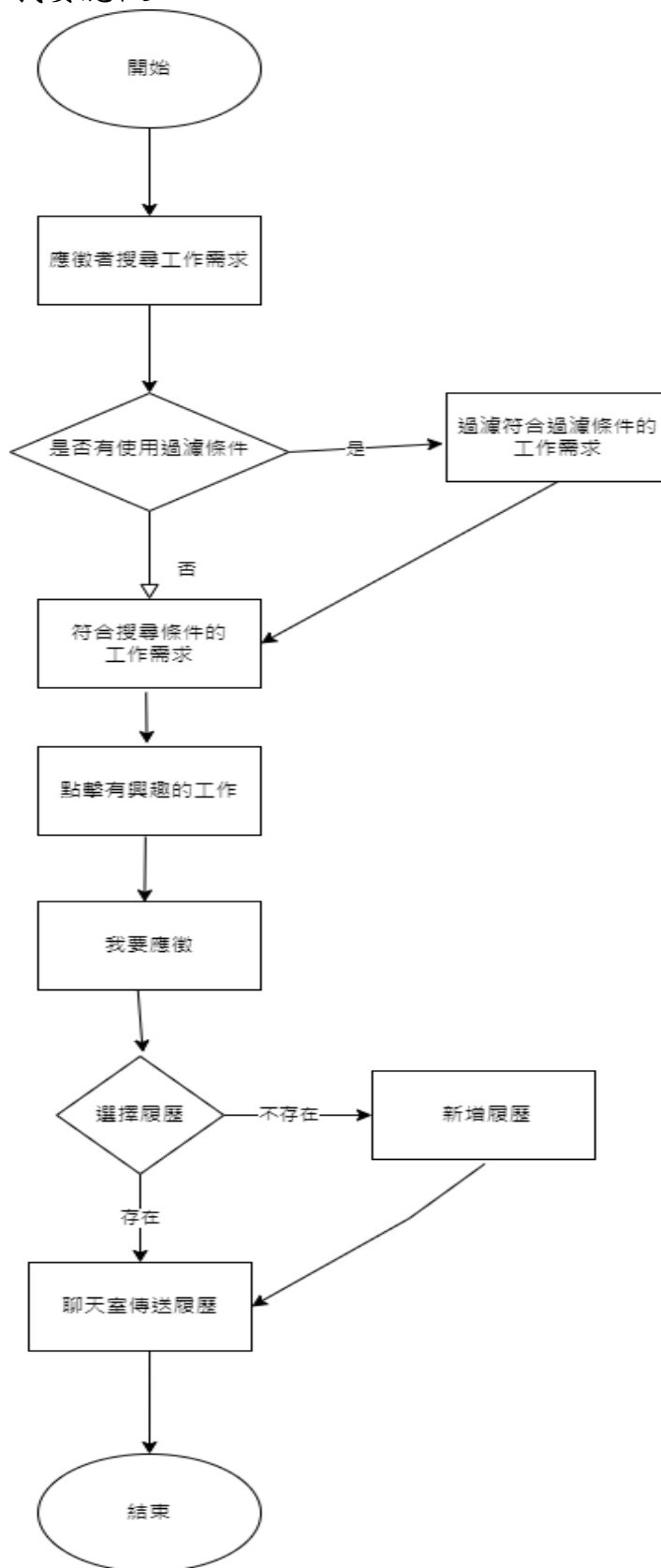
## 5. 新增/修改履歷:



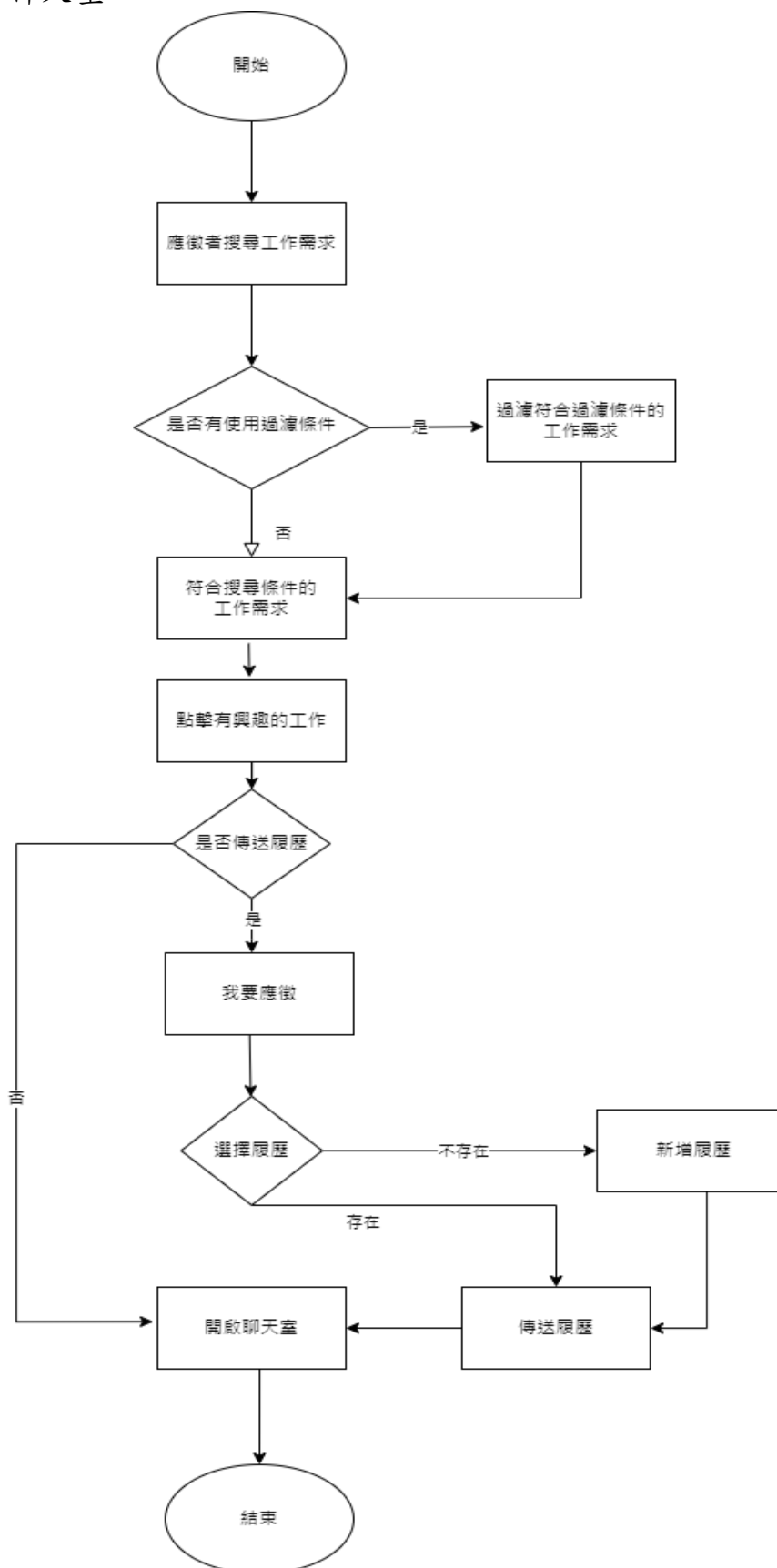
## 6. 新增/修改工作需求:



## 7. 我要應徵：

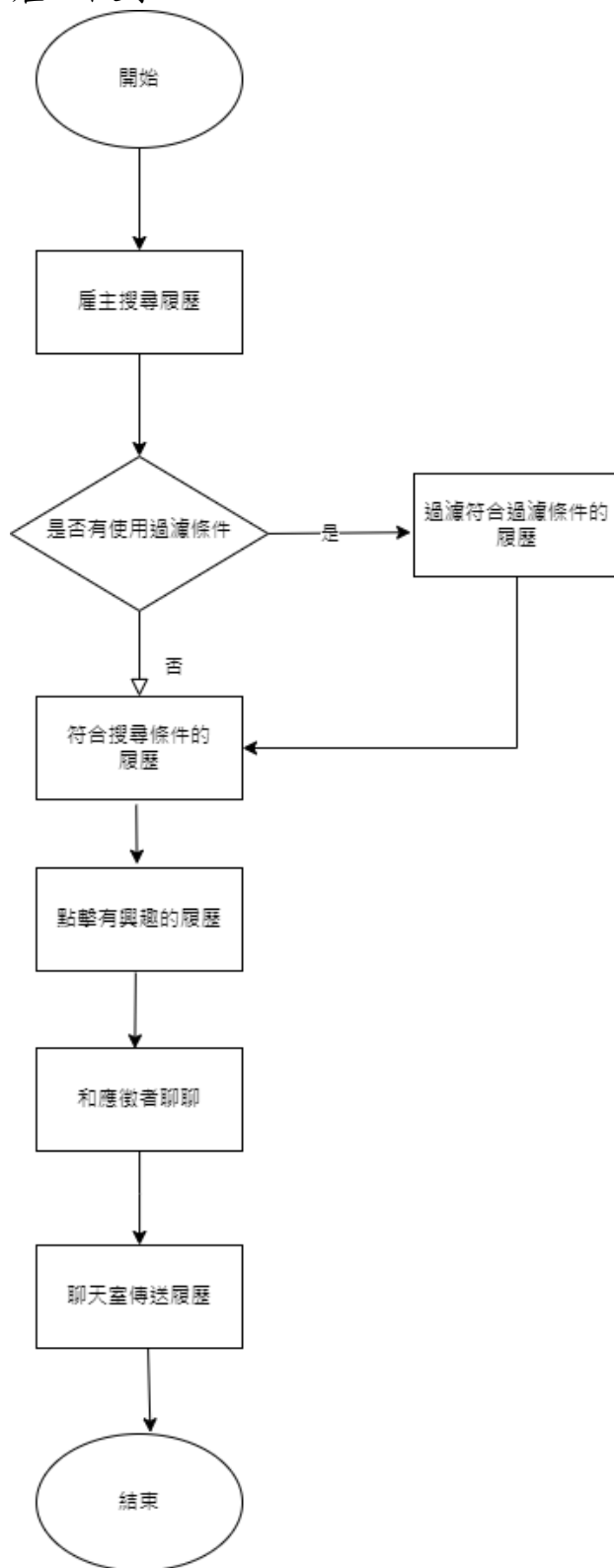


## 8. 聊天室





## 9. 雇主徵才：



#### 4. 使用者畫面設計(User Interface Design)

登入頁面

A hand-drawn sketch of a login page on a grid background. The design is enclosed in a blue rectangular border. At the top, the text "馬上上工" (Start work immediately) is written in a large, stylized font. Below this, there is a large rectangular box labeled "photo". Underneath the photo box, there are two smaller rectangular boxes: the first is labeled "帳號" (Username) and the second is labeled "密碼" (Password). Below these two boxes is a single rectangular box labeled "Login". At the bottom of the page, there is a horizontal line, and below it, the text "沒有帳號? 註冊" (No account? Register) is written.

## 註冊頁面

馬工上上

Sign Up

帳號

email

密碼

密碼 again

註冊

The image shows a hand-drawn registration form on a yellow grid background. At the top, the text '馬工上上' is written in a stylized, handwritten font, followed by 'Sign Up' in a similar font. Below this, there are four input fields, each with a label to its left: '帳號' (Username), 'email', '密碼' (Password), and '密碼 again' (Password again). At the bottom, there is a button labeled '註冊' (Register).

## 應徵者搜尋工作需求頁面



點任何 工作需求 都會跳轉至 工作需求的詳細資訊頁面

點 工作需求 的星星會將其加入收藏


搜尋按鈕：應徵者搜尋工作需求頁面

聊天按鈕：到聊天室頁面

履歷按鈕：應徵者存在履歷列頁面

個人按鈕：個人資料頁面

工作需求的詳細資訊業面

 小箱作業員

工作種類: 正職

工作內容: XXXXXXXXX

工作需求: XXXXXXXXX


工作地區: XXX

薪資待遇: XXXXX~XXXXX

聯絡電話: 0987-089-487

聯絡信箱: mashangice@gmail.com

---

 收藏 | 詢問 | 我要應徵

收藏按鈕：收藏這份工作需求

詢問按鈕：跳到和雇主的聊天室

我要應徵按鈕：跳到選擇履歷的頁面


## 選擇履歷頁面


The wireframe is enclosed in a blue border and set against a light yellow grid background. At the top, the text '應徵' (Apply) is centered. Below it, the text '選擇履歷:' (Select Resume:) is positioned. The main content area contains two identical resume entry boxes, each with a title field labeled '履歷title' and an '編輯' (Edit) button to its right. Below these boxes is a '新增履歷' (Add Resume) button. At the bottom right of the page is a '送出' (Submit) button.

編輯按鈕：編輯存在的履歷

新增履歷按鈕：新增一份新的履歷

送出按鈕：送出已選定的履歷到與雇主的聊天室



 基本資料

姓名: 小冰

性別:

☒ 男      ☐ 女      ☐ 其他

出生: 1970-01-01

電話: 09XX-XXX-XXX

信箱: XXXX@gmail.com

下一步



履歷主旨:

test

學經歷

學校名稱:

中央大學



科系名稱:

資工系



就學狀態:

• 就讀中 • 畢業 • 肄業

年級:

• 一 • 二 • 三 • 四 • 五

下一步





## 求職條件

工作類別:

+

應徵種類:

+

工作時段:

+

希望待遇:

500 / hr

希望工作地點:

台北

自傳

確定

應徵者將履歷送到雇主聊天室的畫面

← 上上

已送出履歷

Name

Age

Introduction...

Up

XXXXX

9:27

13:10

00000

19:21

Up

XXXXX

19:61

你好~~

Thanks!!!

請問缺人嗎?

輸入訊息

📎

點擊履歷：跳到應徵者詳細履歷頁面

詳細履歷頁面

←

英文家教

姓名: XXX

性別: 男

age: XX

聯絡電話: 0978-094-998

聯絡信箱: xiaomin@gmail.com

學經歷: NTU-CSE

工作時段: 18:00 ~ 20:00

自我簡介:

XXXX = XXX

XXXXXXXXXX

XXXXXX

XXXXXXXXXXXXXX

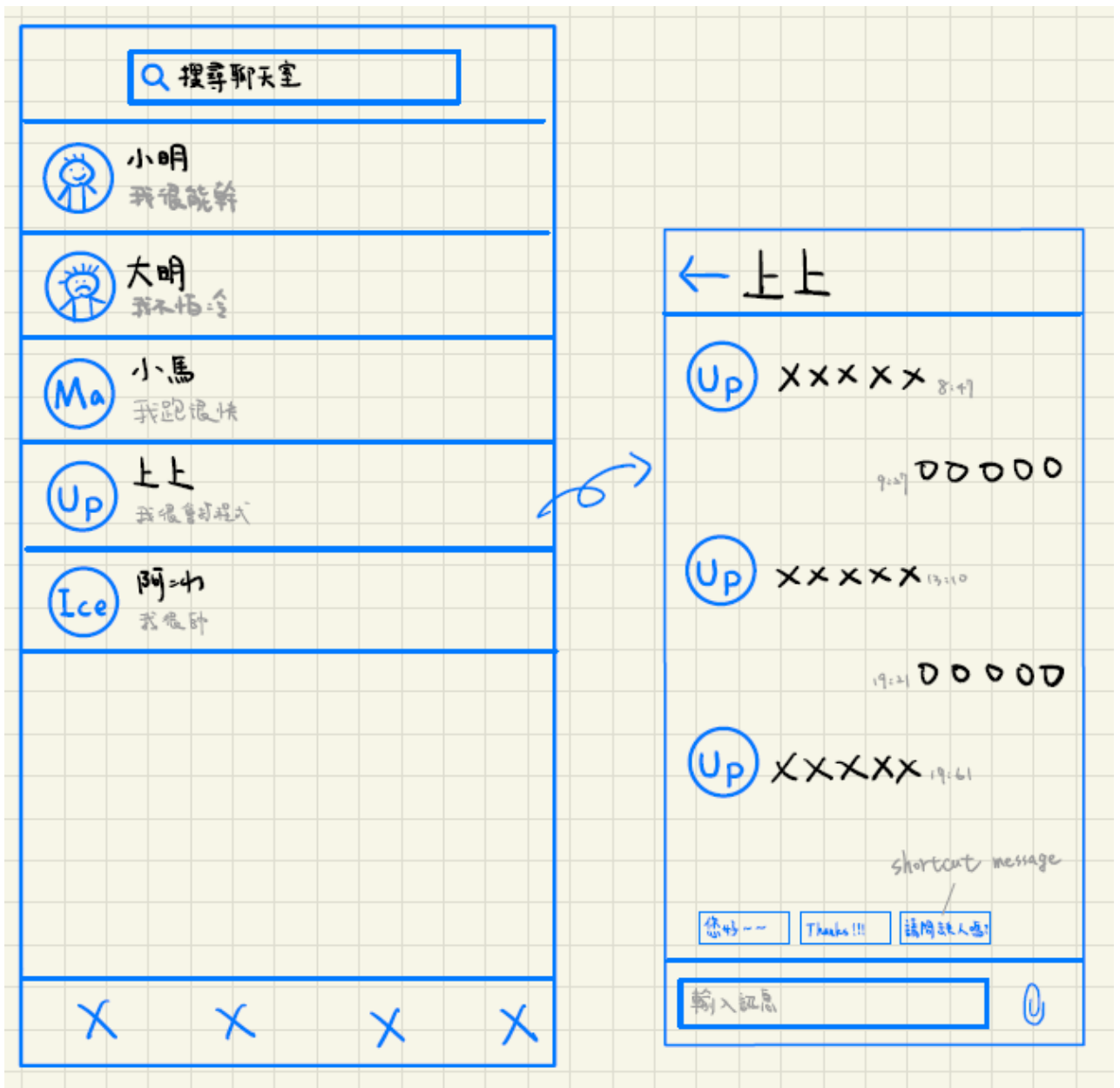
評價: 4.2

✪

XXX

XXXX

## 聊天室頁面



## 應徵者存在履歷列頁面



新增履歷/修改按鈕：切換到編輯/新增履歷頁面

刪除履歷按鈕：刪除選定的履歷

履歷可視開關按鈕：可使履歷隱藏於應徵者履歷頁面中

個人資料頁面

[illegible]

我的收藏按鈕：切換到收藏工作需求頁面

切換為雇主模式按鈕：切換成雇主搜尋應徵者頁面

問題回報按鈕：切換到問題會報頁面

## 問題回報頁面

← 問題回報

email: xxxxxx@xxx.xxx

問題type

xxxxxx

V

問題內容:

xxxxxxxxxxxx

送出

## 雇主搜尋應徵者頁面



點任何 履歷 都會跳轉至 詳細履歷頁面

點 工作需求 的星星會將其加入收藏

搜尋按鈕：應徵者搜尋工作需求頁面

聊天按鈕：到聊天室頁面

履歷按鈕：雇主存在工作需求列頁面

個人按鈕：個人資料頁面



## 存在工作需求列頁面



新增工作/修改按鈕：切換到新增/編輯工作需求頁面

刪除履歷按鈕：刪除選定的工作需求

履歷可視開關按鈕：可使工作需求隱藏於應徵者搜尋工作需求頁面中



**基本資料**

姓名：  
例：黃小米

性別：  
☒ 男      ☐ 女

年齡：  
例：2000/01/01 >

電話：  
例：0912345678

信箱：  
例：xxxx@gmail.com

瀏覽

儲存



工作主旨:

**工作條件**

工作性質:

+

工作日期:

+

工作時段:

+

瀏覽

儲存



工作需求：

請輸入您工作的需求

工作内容：

請輸入您工作的内容

薪資待遇：

請輸入您工作的薪資待遇

工作地點：

找找看您工作的地點



支薪方式：

找找看您支薪的方式



瀏覽

儲存

個人資料頁面

[illegible]

我的收藏按鈕：切換到收藏工作需求頁面

切換為雇主模式按鈕：切換成應徵者搜尋工作需求頁面

問題回報按鈕：切換到問題會報頁面

## 5. 資料設計(Data Design)

### a. MemberAccount

A. 用途：儲存使用者帳號

B. 資料欄位

欄位代碼	欄位名稱	欄位內容	欄位型態
id	MongoDB ID	MongoDB ID	String
username	使用者名稱	使用者名稱	String
userID	使用者帳號	使用者帳號	String
email	使用者電子信箱	使用者電子信箱	String
password	使用者密碼	使用者登入密碼	String
roles	角色	Admin or User	List<String>

### b. Mail

A. 用途：儲存和驗證使用者的電子信箱

B.

欄位代碼	欄位名稱	欄位內容	欄位型態
receivers	收件者	收件者	String
verifyCode	驗證碼	驗證使用者的驗證碼	String
email	電子信箱	驗證碼寄送之電子郵件	String

### c. ChatMessage

A. 用途：存放訊息接收者、傳送者和訊息內容

B.

欄位代碼	欄位名稱	欄位內容	欄位型態
sender	發信者	發信的使用者	String
receiver	收件者	收件的使用者	String

Message	訊息	發信者傳送給收件者的訊息	String
---------	----	--------------	--------

d. Job

A. 用途：存放雇主的工作需求

B.

欄位代碼	欄位名稱	欄位內容	欄位型態
title	工作主旨	雇主工作需求之主旨	String
name	雇主姓名	雇主的姓名	String
sex	性別	雇主的性別	String
phoneNumber	電話號碼	雇主的電話號碼	String
email	信箱	雇主的信箱	String
nature	工作性質	雇主工作需求之工作性質	String
type	工作需求種類	雇主工作需求種類	String
content	工作內容	工作需求的內容	String
date	工作日期時間	工作需求日期、時間	List< String>
time	工作時段	工作需求時段	List< String >
salary	薪資待遇	雇主提出的薪資待遇	String
region	工作地點	工作需求的地點	String
salaryMethod	支薪方式	雇主給應徵者的支薪方式	String
salaryDate	支薪日	支薪日	String
id	MongoDB ID	MongoDB ID	String
user	用戶名	雇主用戶名	String
createTime	創立時間	工作需求的創立時間	String
refreshTime	更新時間	工作需求的更新時間	String

e. Resume

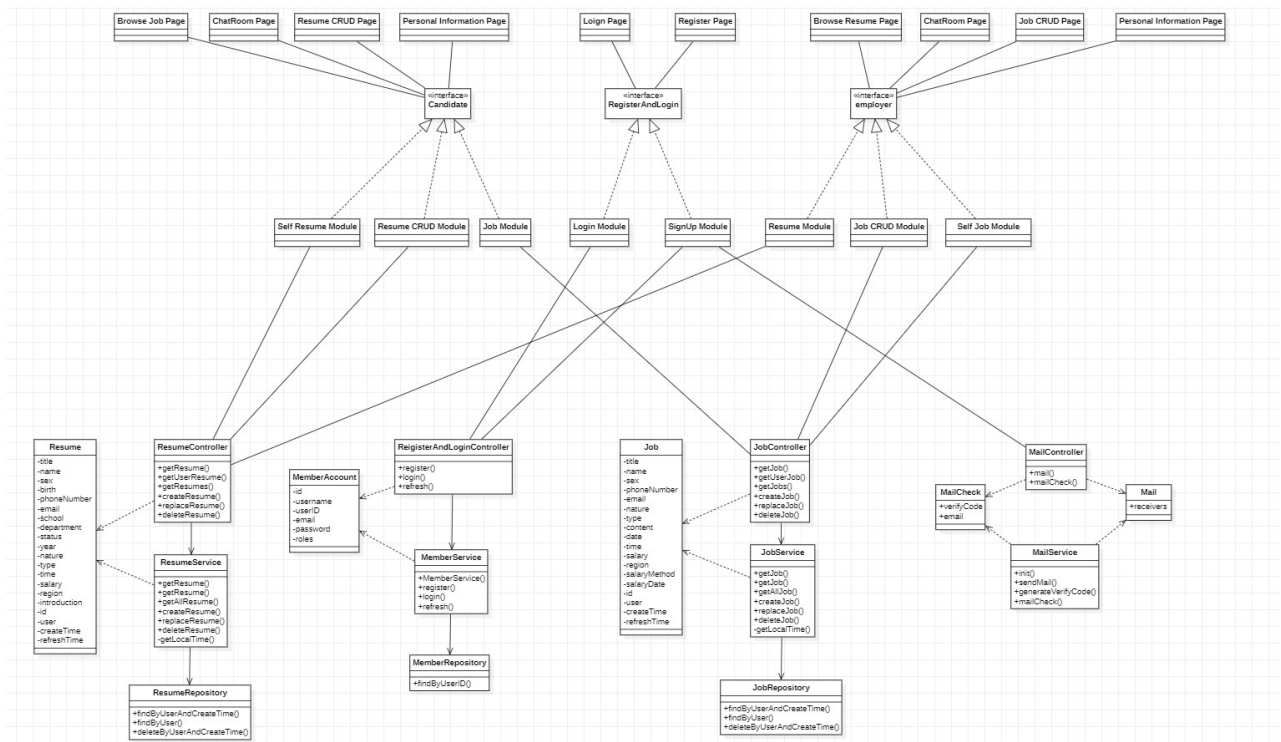
A. 用途：存放應徵者之履歷

B.

欄位代碼	欄位名稱	欄位內容	欄位型態
title	主旨	應徵者的履歷主旨	String
name	姓名	應徵者的姓名	String
sex	性別	應徵者的性別	String
birth	出生年	應徵者的出生年	String
phoneNumber	電話號碼	應徵者的聯絡電話	String
email	信箱	應徵者的電子信箱	String
school	學校名稱	應徵者的最高學歷學校	String
department	科系名稱	應徵者就讀的科系	String
status	就學狀態	應徵者的就學狀況	String
year	年級	應徵者的年齡	String
nature	工作性質	應徵者的履歷的工作性質	String
type	工作種類	應徵者履歷的工作種類	String
time	工作時段	應徵者的可安排工作時段	String
salary	希望待遇	應徵者預期的薪資待遇	String
region	地區	應徵者所在的地區	String
introduction	自我介紹	應徵者的自我介紹	String
id	MongoDB ID	MongoDB ID	String
user	用戶名	應徵者的用戶名	String
createTime	創立時間	履歷的創立時間	String
refreshTime	更新時間	履歷的更新時間	String



## 6. 類別圖設計(Class Diagram)



## 7. 實作方案(Implementation Languages and Platforms)

系統平台	手機 App
前端	React Native
前端框架	React Native
後端	Spring boot
後端框架	Spring boot
資料庫	MongoDB
前後端溝通方式	Restful API, Websocket

## 8. 設計議題(Design Issue)

### 1. 議題一

A. 後端實作語言

B. 解決方案

a. Nodejs

b. Spring boot

C. 最後定案

Spring boot

D. 原因：想要學習 spring boot 的框架

### 2. 議題二

A. 關聯式資料庫 vs 非關聯式資料庫

B. 解決方案

a. 關聯式資料庫

b. 非關聯式資料庫

C. 最後定案

非關聯式資料庫

D. 原因：因為 MongoDB 很好呈現資料