

國立臺北商業大學

資訊管理系

110 資訊系統專案設計

系統手冊



組 別：第 110202 組

題 目：yo! 煮廚

指導老師：丁慧瑩老師

組 長：10936036 陳品佑

組 員：10936019 林喆

10936020 林宜潔

10936024 張琇茹

中 華 民 國 1 1 0 年 1 1 月 2 4 日

目錄

第 1 章 背景與動機	1
1-1 簡介	1
1-2 問題與機會	2
1-3 相關系統探討	3
第 2 章 系統目標與預期成果	4
2-1 系統目標	4
2-2 預期成果	4
第 3 章 系統規格	5
3-1 系統架構	5
3-2 系統軟、硬體需求與技術平台	6
3-3 使用標準與工具	6
第 4 章 系統規格	7
4-1 專案時程	7
4-2 專案組織及分工	9
第 5 章 需求模型	11
5-1 使用者需求	11
5-2 使用個案圖	12
5-3 使用個案描述	13

5-4 分析類別圖	19
第 6 章 設計模型	20
6-1 循序圖	20
6-2 設計類別圖	26
第七章 資料模型	27
7-1 佈署圖(DEPLOYMENT DIAGRAM)	27
7-2 套件圖(PACKAGE DIAGRAM)	27
7-3 元件圖(COMPONENT DIAGRAM)	28
7-4 狀態圖(STATE MACHINE)	28
第八章 資料庫設計	32
8-1 資料庫關聯表	32
8-2 表格及其 METADATA	32
第九章 程式規格	37
9-1 元件清單及其規格描述	37
9-2 其他附屬的各種文件	40
第十章 測試模型	59
10-1 測試計畫	59
10-2 測試個案與測試結果資料	61
第十一章 操作手冊	69

11-1 系統安裝	69
第十二章 使用手冊	70
12-1 註冊、登入與忘記密碼.....	70
12-2 搜尋食譜與新增食譜.....	72
12-3 日記.....	75
12-4 個人檔案.....	77
12-4 CHATBOT	79
第十三章 感想	80
10936019 林喆	80
10936020 林宜潔	80
10936024 張琇茹	80
10936036 陳品佑	81
第十四章 參考文獻	82
14-1 參考資料.....	82
附錄 初評之評審建議	85

表目錄

表 1-2-1	SWOT 分析	2
表 1-3-2	競爭者分析	3
表 3-2-1	系統軟、硬體需求與技術平台	6
表 3-3-1	系統環境	6
表 3-3-2	系統開發工具	6
表 3-3-3	管理程式平台	6
表 3-3-4	程式開發技術	6
表 3-3-5	設計工具	6
表 3-3-6	伺服器	6
表 4-1-1	專案時程	7
表 4-2-1	專案組織分工	9
表 5-1-1	功能性需求	11
表 8-2-1	使用者資料表(USER)	32
表 8-2-2	日記資料表(DIARY)	33
表 8-2-3	收藏資料表(LIKES)	33
表 8-2-4	類別資料表(CATEGORY)	33
表 8-2-5	烹調資料表(COOKING)	34
表 8-2-6	食材資料表(INGREDIENT)	34
表 8-2-7	菜系資料表(INTERNATIONAL)	34
表 8-2-8	份量資料表(QUANTITY)	34
表 8-2-9	食譜資料表(RECIPE)	35
表 8-2-10	食譜照片資料表(RECIPE_PICTURE)	36
表 8-2-11	食譜步驟資料表(RECIPE_STEPS)	36
表 9-1-1	使用者管理	37
表 9-1-2	主功能	37
表 9-1-3	日記管理	37
表 9-1-4	個人檔案	38
表 9-1-5	食譜功能	38

表 9-1-6	CHATBOT 功能	39
表 9-1-7	工具	39
表 9-2-1	外部元件清單	40
表 9-2-2	LOGINACTIVITY.JAVA	41
表 9-2-3	FRAGMENT-RECIPE.JAVA	49
表 9-2-3	SEARCHRECIPEADAPTER.JAVA	54
表 10-2-1	註冊會員測試資料	61
表 10-2-2	登入會員測試資料	61
表 10-2-3	登入會員測試資料	61
表 10-2-4	個人密碼修改測試資料	61
表 10-2-5	食譜首頁	62
表 10-2-6	下方導覽列	62
表 10-2-7	新增食譜	62
表 10-2-8	食譜封面照	63
表 10-2-9	撰寫名稱簡介	63
表 10-2-10	公開選項	63
表 10-2-11	類別選項	63
表 10-2-12	烹調方式選項	64
表 10-2-13	烹調時間選項	64
表 10-2-14	份量選項	64
表 10-2-15	葷素選項	64
表 10-2-16	菜系選項	65
表 10-2-17	食譜清單	65
表 10-2-18	食譜步驟照片	65
表 10-2-19	食譜步驟說明	65
表 10-2-20	食譜關鍵字搜尋	66
表 10-2-21	食譜篩選搜尋	66
表 10-2-22	食譜詳細	66
表 10-2-23	收藏食譜	67

表 10-2-24	食譜評分	67
表 10-2-25	留言板	67
表 10-2-26	日記查詢測試	67
表 10-2-27	CHATBOT 使用者輸入測試	68
表 10-2-28	CHATBOT 回覆測試	68

圖目錄

圖 1-1-1 新冠肺炎飲食習慣改變.....	1
圖 1-1-2 POLLSTER 波仕特線上市調.....	1
圖 3-1-1 系統架構.....	5
圖 3-1-2 系統功能詳細圖	5
圖 5-2-1 使用個案圖(USE CASE DIAGRAM).....	12
圖 5-3-1 註冊 ACTIVITY DIAGRAM	13
圖 5-3-2 登入 ACTIVITY DIAGRAM	13
圖 5-3-4 新增日記 ACTIVITY DIAGRAM	14
圖 5-3-3 新增食譜 ACTIVITY DIAGRAM	14
圖 5-3-5 維護日記 ACTIVITY DIAGRAM	15
圖 5-3-6 維護食譜 ACTIVITY DIAGRAM	16
圖 5-3-7 查詢日記 ACTIVITY DIAGRAM	16
圖 5-3-9 修改個人檔案 ACTIVITY DIAGRAM	17
圖 5-3-8 查詢食譜 ACTIVITY DIAGRAM	17
圖 5-3-10 CHATBOT ACTIVITY DIAGRAM.....	18
圖 5-4-1 分析類別圖.....	19
圖 6-1-1 註冊 SEQUENTIAL DIAGRAM.....	20
圖 6-1-3 新增日記 SEQUENTIAL DIAGRAM.....	21
圖 6-1-2 新增食譜 SEQUENTIAL DIAGRAM	21
圖 6-1-4 維護食譜 SEQUENTIAL DIAGRAM.....	22

圖 6-1-5 維護日記 SEQUENTIAL DIAGRAM.....	22
圖 6-1-7 查詢食譜 SEQUENTIAL DIAGRAM.....	23
圖 6-1-6 查詢日記 SEQUENTIAL DIAGRAM.....	23
圖 6-1-9 個人檔案管理 SEQUENTIAL DIAGRAM.....	24
圖 6-1-8 CHATBOT SEQUENTIAL DIAGRAM	24
圖 6-1-10 登入 SEQUENTIAL DIAGRAM.....	25
圖 6-2-1 設計類別圖.....	26
圖 7-1-1 佈署圖 DEPLOYMENT DIAGRAM.....	27
圖 7-2-1 套件圖 PACKAGE DIAGRAM.....	27
圖 7-3-1 元件圖 COMPONENT DIAGRAM.....	28
圖 7-4-1 CHATBOT STATE MACHINE.....	28
圖 7-4-2 日記 STATE MACHINE	29
圖 7-4-4 註冊 STATE MACHINE	30
圖 7-4-3 食譜 STATE MACHINE	30
圖 7-4-5 登入 STATE MACHINE	31
圖 8-1-1 資料庫關聯表.....	32
圖 12-1-2 註冊	70
圖 12-1-1 登入.....	70
圖 12-1-3 GOOGLE 登入.....	71
圖 12-1-4 GOOGLE 驗證	71
圖 12-1-5 GOOGLE 驗證成功	71

圖 12-1-8 變更密碼成功	71
圖 12-1-7 身分驗證	71
圖 12-1-6 忘記密碼	71
圖 12-2-1 食譜首頁	72
圖 12-2-2 搜尋頁面	72
圖 12-2-3 關鍵字搜尋	72
圖 12-2-6 篩選資料	73
圖 12-2-5 篩選頁面	73
圖 12-2-4 搜尋頁面	73
圖 12-2-9 新增食譜步驟	74
圖 12-2-8 新增食譜清單	74
圖 12-2-7 新增食譜簡介	74
圖 12-3-1 日記列表	75
圖 12-3-2 選擇日期	75
圖 12-3-3 查看當天日記	75
圖 12-3-4 新增按鈕	76
圖 12-3-5 新增日記頁面	76
圖 12-3-6 輸入敘述	76
圖 12-4-1 個人檔案首頁	77
圖 12-4-2 編輯資料	77
圖 12-4-3 變更照片	77

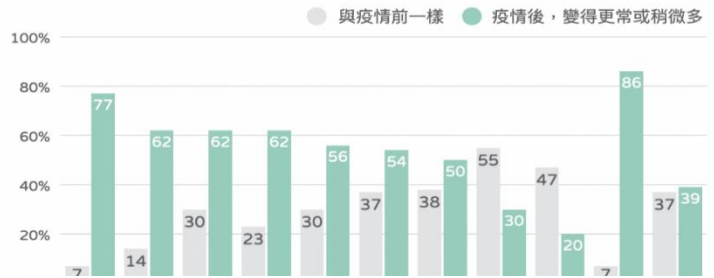
圖 12-4-6 食譜詳細	77
圖 12-4-5 歷屆食譜	77
圖 12-4-4 個人檔案	77
圖 12-4-7 使用說明	78
圖 12-4-8 關於我們	78
圖 12-4-9 發送電子郵件	78
圖 12-4-10 CHATBOT	79

第 1 章 背景與動機

1-1 簡介

2019 年底於武漢爆發新型冠狀肺炎，累計確診病例在 2021 年 1 月達到近八億例，已經超越 2002 年爆發的 SARS 病例。也因武漢肺炎臺灣迅速進入防疫模式，政府至今仍持續不斷的向大眾宣導，要求民眾在公共場所戴口罩、勤洗手、保持安全距離等，以致於大幅降低了民眾出門的慾望。防疫飲食的概念隨著疫情而興起，促使亞洲國家的人們在疫情下，更傾向於在家吃飯、自己煮或外帶回家，以減少在外與他人接觸的情況。

新冠肺炎對亞洲消費者飲食習慣的影響
在家用餐的習慣



▲ 圖 1-1-1 新冠肺炎飲食習慣改變

請問您每隔多久會自己下廚做菜
(煮開水、煮泡麵不算)？

特性	不會 做菜	幾乎 不煮	每天 都會	2-3天 1次	4-5天 1次	6-7天 1次	2-3週 1次	1個月 1次	超過 1個月 1次	總和
性別										
男性	21.7%	24.8%	12.2%	11.5%	6.1%	6.0%	5.0%	5.0%	7.7%	100.0%
女性	12.1%	16.4%	19.5%	16.1%	7.7%	9.3%	6.4%	4.4%	8.1%	100.0%
年齡層										
19歲以下	25.0%	16.6%	16.7%	5.6%	8.3%	5.6%	5.6%	8.3%	8.3%	100.0%
20-29歲	18.8%	22.1%	9.4%	14.1%	6.9%	7.4%	6.2%	7.7%	7.4%	100.0%
30-39歲	16.6%	22.9%	13.1%	14.4%	6.6%	7.2%	6.1%	4.6%	8.5%	100.0%
40-49歲	15.7%	16.3%	17.5%	15.1%	9.0%	9.9%	5.7%	2.7%	8.1%	100.0%
50歲以上	12.2%	15.4%	33.0%	12.5%	5.4%	7.9%	4.3%	2.5%	6.8%	100.0%
總和	16.4%	20.1%	16.2%	14.0%	7.0%	7.8%	5.8%	4.7%	8.0%	100.0%

Pollster 波仕特線上市調

然而，根據 Pollster 波仕特線上市調網對近 2000 位 13 歲以上的國人進行調查，調查資料顯示有將近 40% 的受訪者表示不下廚，不會煮菜與不懂如何拿捏食物分量是年輕族群下廚頻率低的主要因素。

因此，「yo！煮廚」是一個能提供適合年輕族群的食譜平台，使用者能透過平台分享自己的獨門食譜，不用大費周章的尋找食譜書籍查閱食譜，能利用關鍵字快速的查詢到

▲ 圖 1-1-2 Pollster 波仕特線上市調

他人上傳分享的食譜，也可使用菜系、類別、烹調方式來查找食譜。此外「yo！煮廚」還有聊天機器人憑藉簡單的問答，為有選擇障礙的使用者提供熱門的食譜選項。「yo！煮廚」可讓使用者將食譜與日記結合起來，用日記來記錄使用者的每日飲食與相關食譜，以便日後進行查閱，使用者能逐步學習如何煮菜、食材拿捏等技巧，並能以此作為日記記錄生活。

1-2 問題與機會

透過「yo!煮廚」，使用者能夠以很方便的體驗在短時間內可以記錄自己的飲食點滴，以及參考系統建議來為自己的下一餐作決定。為了使「yo!煮廚」有效地協助使用者，因此我們將使用 SWOT 分析了解目前狀況。

▼ 表 1-2-1 SWOT 分析

S 優勢	W 劣勢
<ol style="list-style-type: none">1. 食譜能夠結合日記，使用者便能容易回顧歷史紀錄2. 聊天機器人可以，推薦食譜供使用者選擇3. 多種查詢食譜方式，快速精準4. 畫面精簡好操作，符合年輕人的喜好5. 食譜留言功能，提高互動性	<ol style="list-style-type: none">1. 目前無創建網頁供使用者使用2. 尚未開發 iOS 版本，無法觸及 iOS 使用者
T 機會	O 威脅
<ol style="list-style-type: none">1. 疫情未消退，民眾傾向在家飲食2. 上班族重視食安問題，因此親自下廚能減少擔憂	<ol style="list-style-type: none">1. 市場上食譜相關應用程式過多

1-3 相關系統探討

▼ 表 1-3-2 競爭者分析

		yo! 煮廚	愛料理	cookpad
食譜搜尋	關鍵字	✓	✓	✓
	菜系	✓	✗	✗
	類別	✓	✓	✓
	烹調方式	✓	✗	✗
日記		✓	✗	✗
聊天機器人		✓	✗	✗
留言		✓	✓	✓
網頁		✗	✓	✓

由表 1-3-2 得知本系統之主要優點如下：

相關 APP 大多是以關鍵字和類別進行搜尋，「yo!煮廚」擁有多面向的食譜搜尋方式，可讓使用者能快速且精準的找到所需要的食譜；並提供聊天機器人，人性化的溝通，讓「yo!煮廚」更貼近生活，且有留言功能，提升 App 與使用者之間的互動。最後，食譜結合日記，邊做邊紀錄的方式，作為往後回顧所使用。

第 2 章 系統目標與預期成果

2-1 系統目標

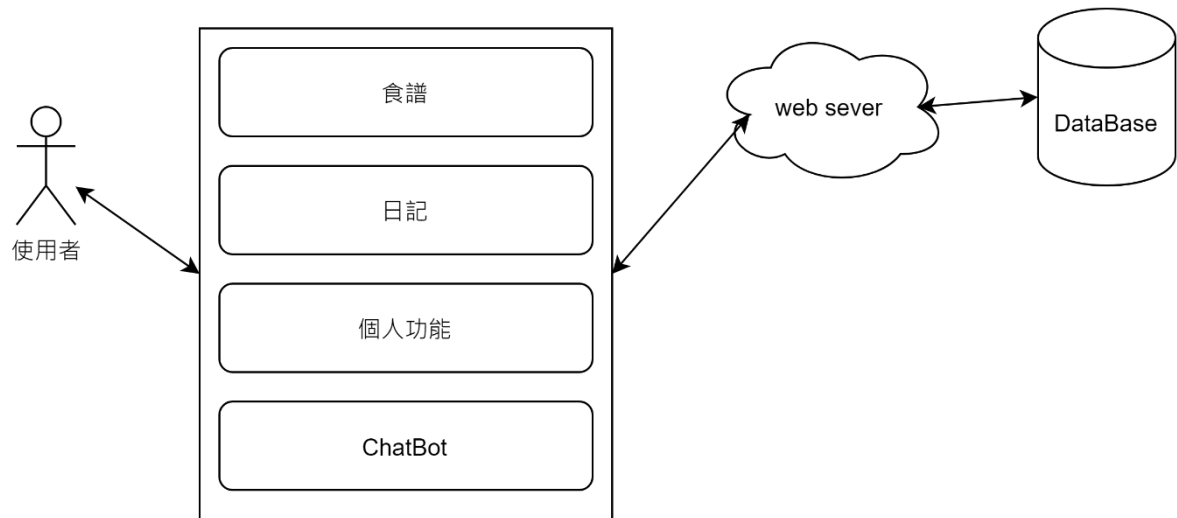
- 快速查詢食譜
用最直覺的分類，關鍵字及烹調模式去尋找想要的食譜，不必耗費太多腦力。
- 機器人推薦
使用機器人制式化的問答能夠縮短要煮甚麼的煩惱時間。
- 食譜日記
透過日記可以將每一次的做菜成品記錄下來，之後搜尋自己做過的相同菜色會發現自己的進步。
- 食譜留言功能
透過食譜留言能夠增加使用者與 app 的互動性。

2-2 預期成果

- 快速便利
很多人會覺得網路上的食譜寫得不清不楚。我們致力改善這一情況，增加使用者的使用體驗。
- 機器人推薦使用者食譜
使用 chatbot 制式化的功能會為現今許多的「選擇困難症」使用者進行縮短要煮甚麼的時間
- 將每一天都記錄下來
人可以一天不運動，但一定要吃飯，把自己吃過的食物、做過的食譜記錄下來，會發現自己一天比一天更加進步。

第 3 章 系統規格

3-1 系統架構



▲ 圖 3-1-1 系統架構

系統功能詳細：

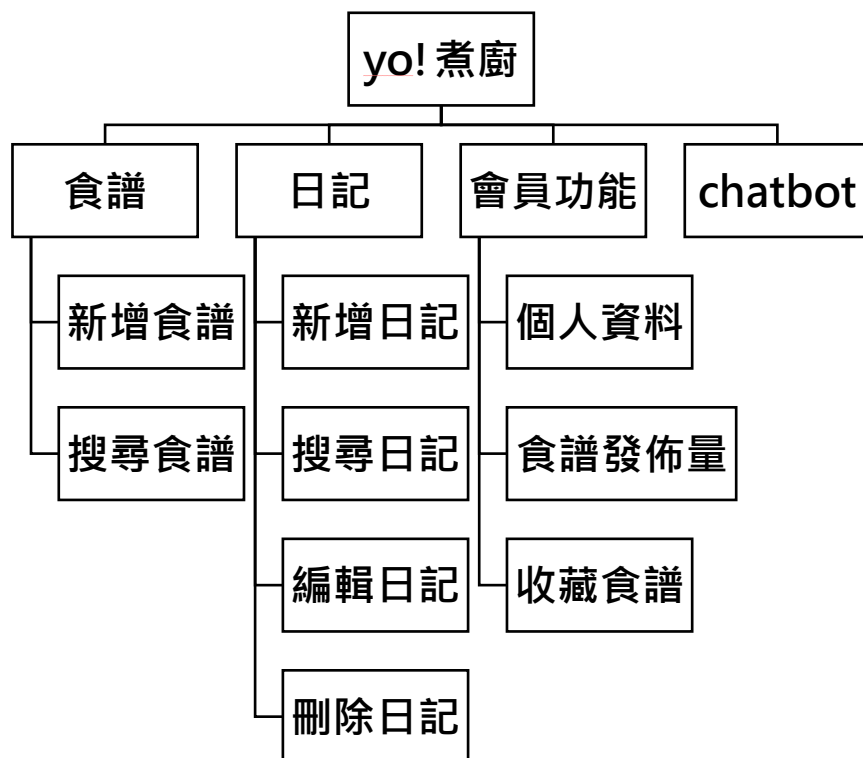


圖 3-1-2 系統功能詳細圖

3-2 系統軟、硬體需求與技術平台

▼ 表 3-2-1 系統軟、硬體需求與技術平台

硬體需求	
作業系統	Android
裝置需求	
網路需求	WiFi/行動網路

3-3 使用標準與工具

▼ 表 3-3-1 系統環境

網路需求	WiFi/行動網路
智慧型行動裝置	Android 版本 6.0 以上

▼ 表 3-3-2 系統開發工具

APP 設計	Android Studio 、AdobeXD
資料庫	MySQL

▼ 表 3-3-3 管理程式平台

專案管理	github、Google Drive、000webhost
------	--------------------------------

▼ 表 3-3-4 程式開發技術

APP	前端技術	Android Studio、Java
	後端技術	php

▼ 表 3-3-5 設計工具

文件製作	Microsoft Word 2019、Google 文件
簡報	Microsoft Power Point 2019
LOGO	Adobe XD、Adobe PS、figma







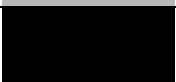










▼ 表 3-3-6 伺服器

伺服器工具	Apache
-------	--------

第 4 章 系統規格

4-1 專案時程

▼ 表 4-1-1 專案時程 ：預定時程、：實際時程

年月份	2021											
任務	1	2	3	4	5	6	7	8	9	10	11	12
主題 發想/擬 定	 											
系統功 能討論		 										
APP 介面 設計			 									
資料庫 設計			 									
資料庫 建置				 								
開發 環境 建置			 									
文件 撰寫					 							
介面 建置			 									

4-2 專案組織及分工

▼ 表 4-2-11 專案組織分工 ●：主要、○：協助

組員 項目	10936036 陳品佑	10936019 林喆	10936020 林宜潔	10936024 張琇茹
1.專題發想				
蒐集資料	●	●	●	●
主題制定	●	●	●	●
需求分析	●	●	●	●
2.資料庫				
資料庫設計	●	○	○	○
資料庫建置	●		○	
3.系統撰寫				
登入介面 設計		○	●	●
登入功能	●	○	○	●
日記功能	○	●	○	●
日記介面 設計	●	○	●	●
食譜功能	○	●	●	○
食譜介面 設計	○	○	●	●
個人功能	○	●	●	●
個人介面 設計	○	○	●	●
Chatbot 功能	●		○	
Chatbot 介面設計	●		○	○
系統整合		○	●	●
測試	●	●	●	●

4.文件製作				
背景與動機		○	●	○
系統目標與 預期成果		●	○	●
系統規格	○	●	○	○
專案管理	○	●	○	○
需求模型		●	●	●
設計模型		●	●	●
活動圖	○	●		○
分析類別圖		●		○
循序圖		○	●	○
佈署圖	●	●		
套件圖	●		●	
元件圖	●			●
狀態機	●	●	●	●
文件統整 編排	●	○	●	○
5.視覺設計				
LOGO 設計			●	○
海報設計			○	●
影片製作	○	●	○	○

第 5 章 需求模型

5-1 使用者需求

- 功能需求

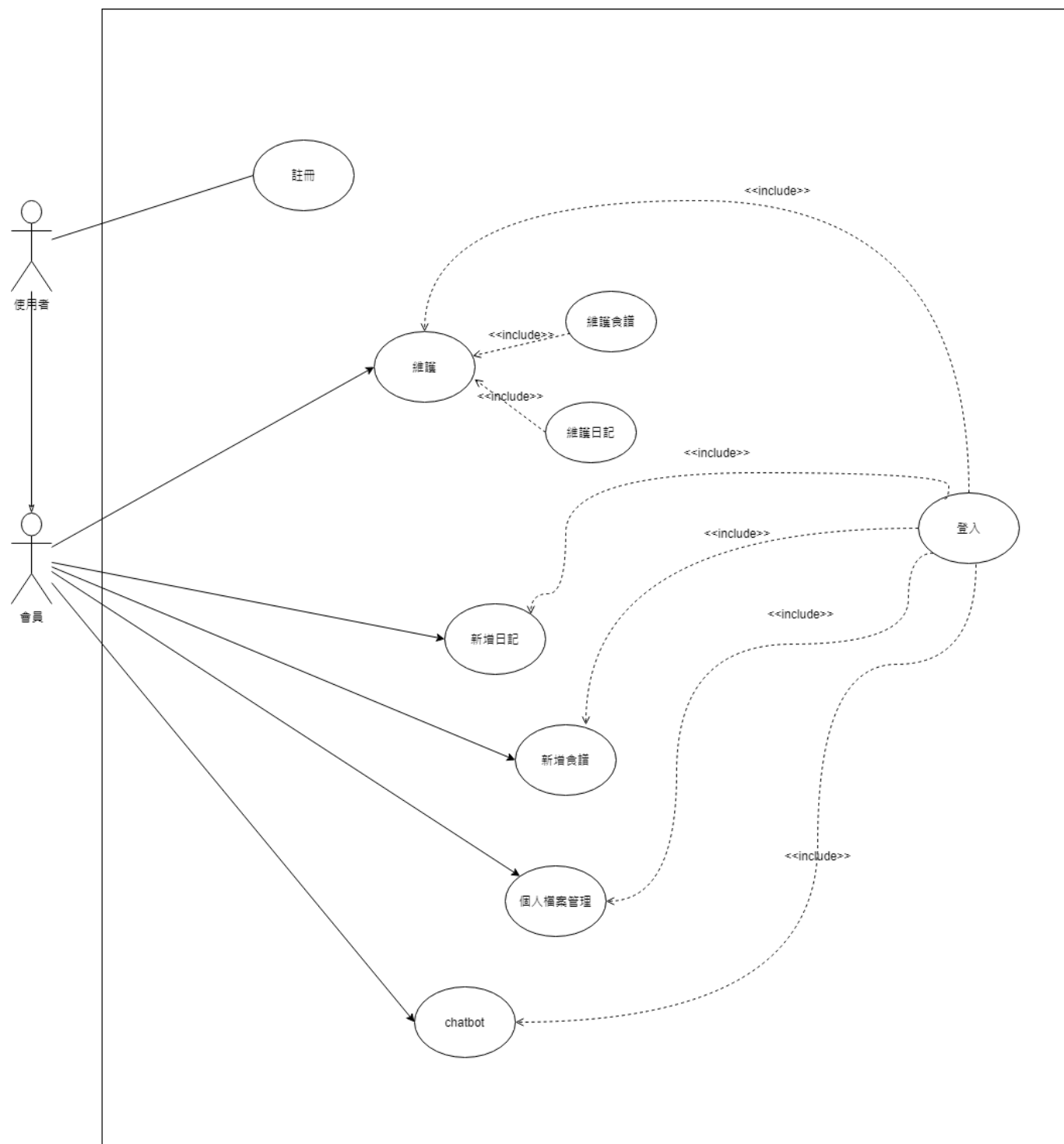
▼表 5-1-1 功能性需求

事件	說明
註冊帳號	使用者創建新會員
會員登入	登入後可開始使用系統
個人檔案管理	使用者可讀取個人資料與修改
新增日記	建立日記內容
查詢現有日記	可利用清單或搜尋功能查看現有日記
修改日記	可編輯現有日記內容
刪除現有日記	可刪除現有日記內容
新增食譜	建立食譜內容
查詢食譜	可利用關鍵字、菜系及種類查詢食譜
修改食譜	可編輯現有食譜內容
刪除現有食譜	可刪除現有食譜內容
chatbot	和機器人互動得出想要的食譜結果

- 非功能需求

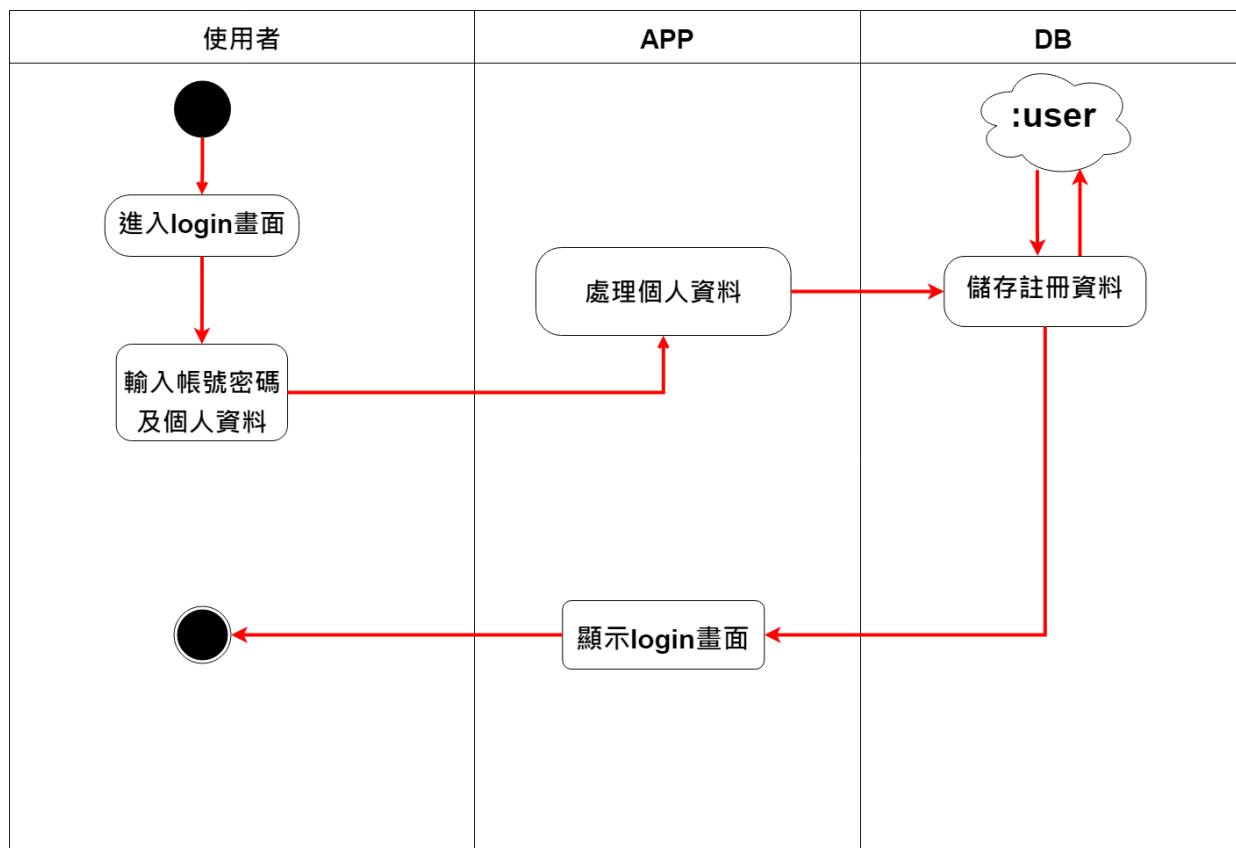
- (1) 裝置需求：支援 WiFi、行動數據
- (2) 使用者必須登入後才可以使⽤日記功能。
- (3) 需開啟網路功能才能更新日記及食譜內容。

5-2 使用個案圖

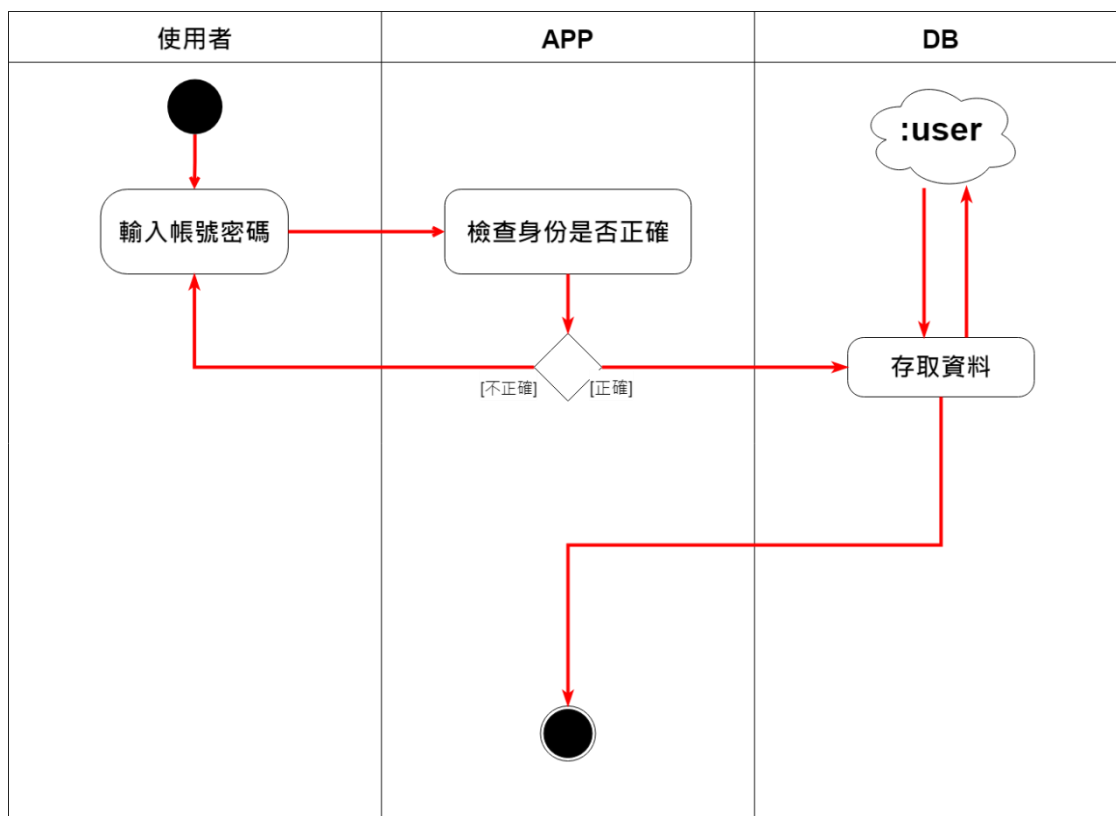


▲圖 5-2-1 使用個案圖(Use Case Diagram)

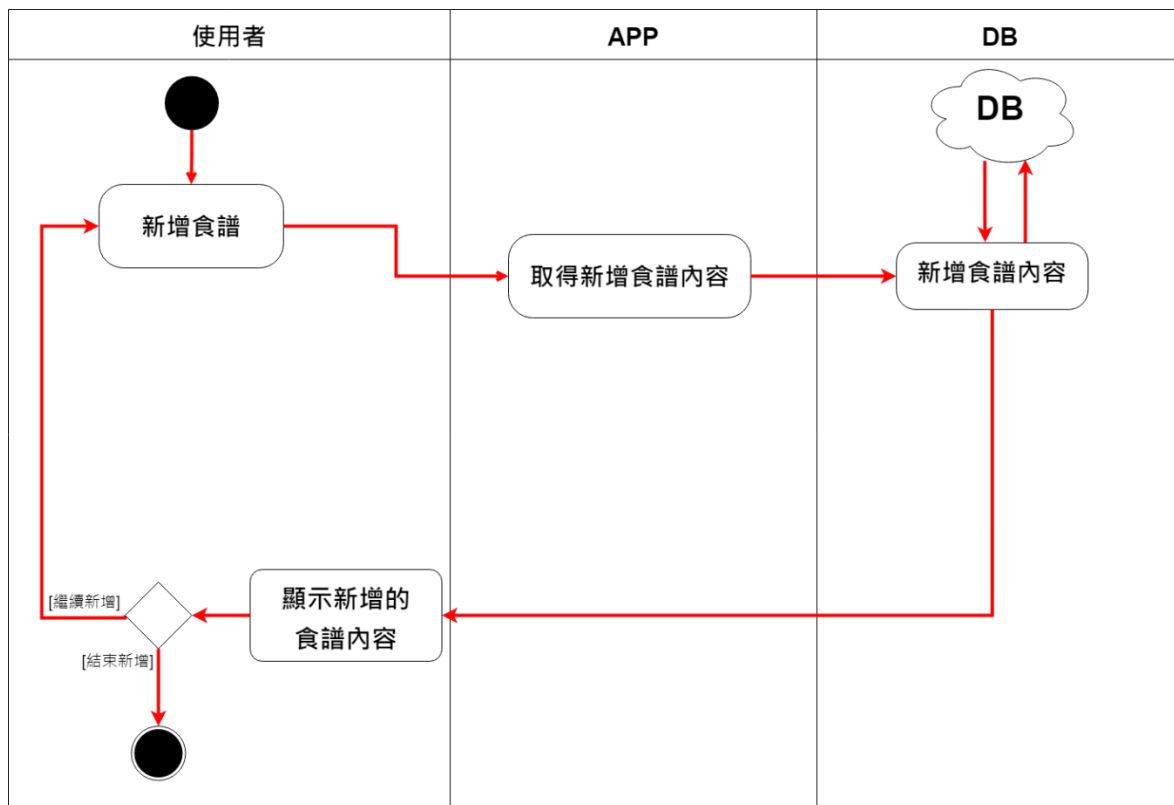
5-3 使用個案描述



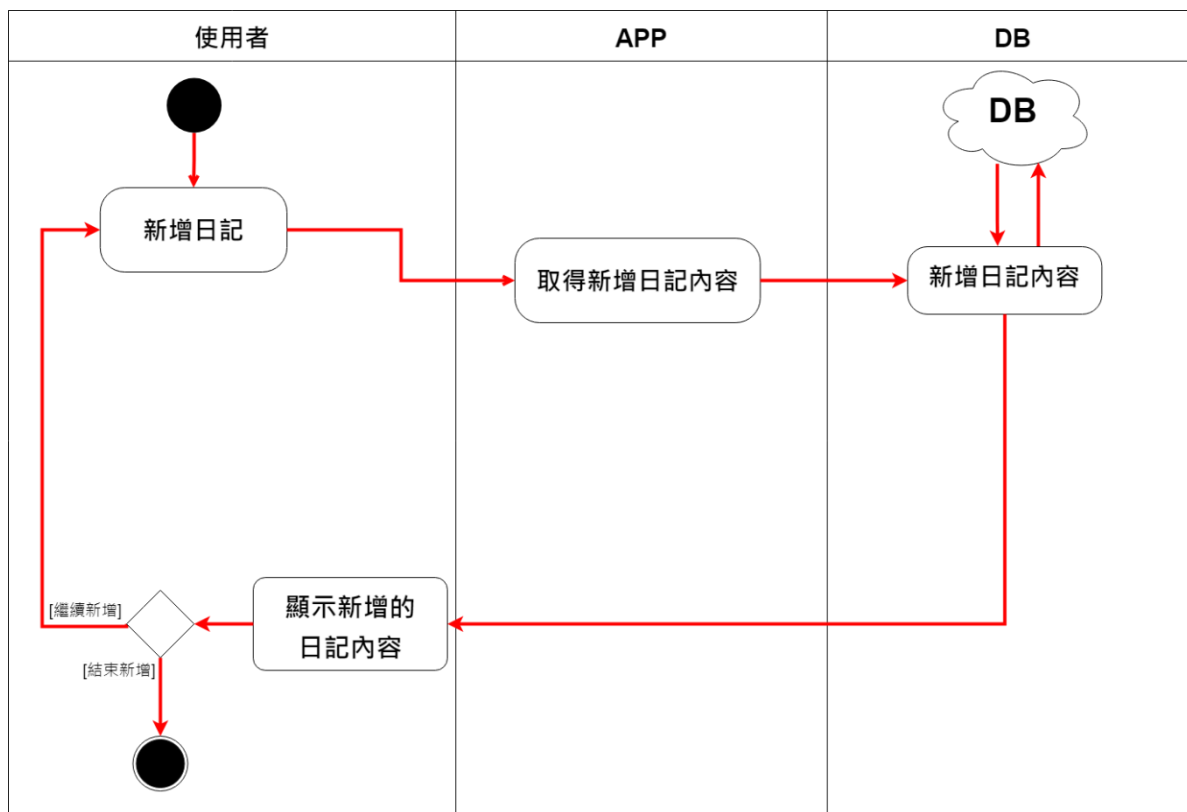
▲圖 5-3-1 註冊 activity diagram



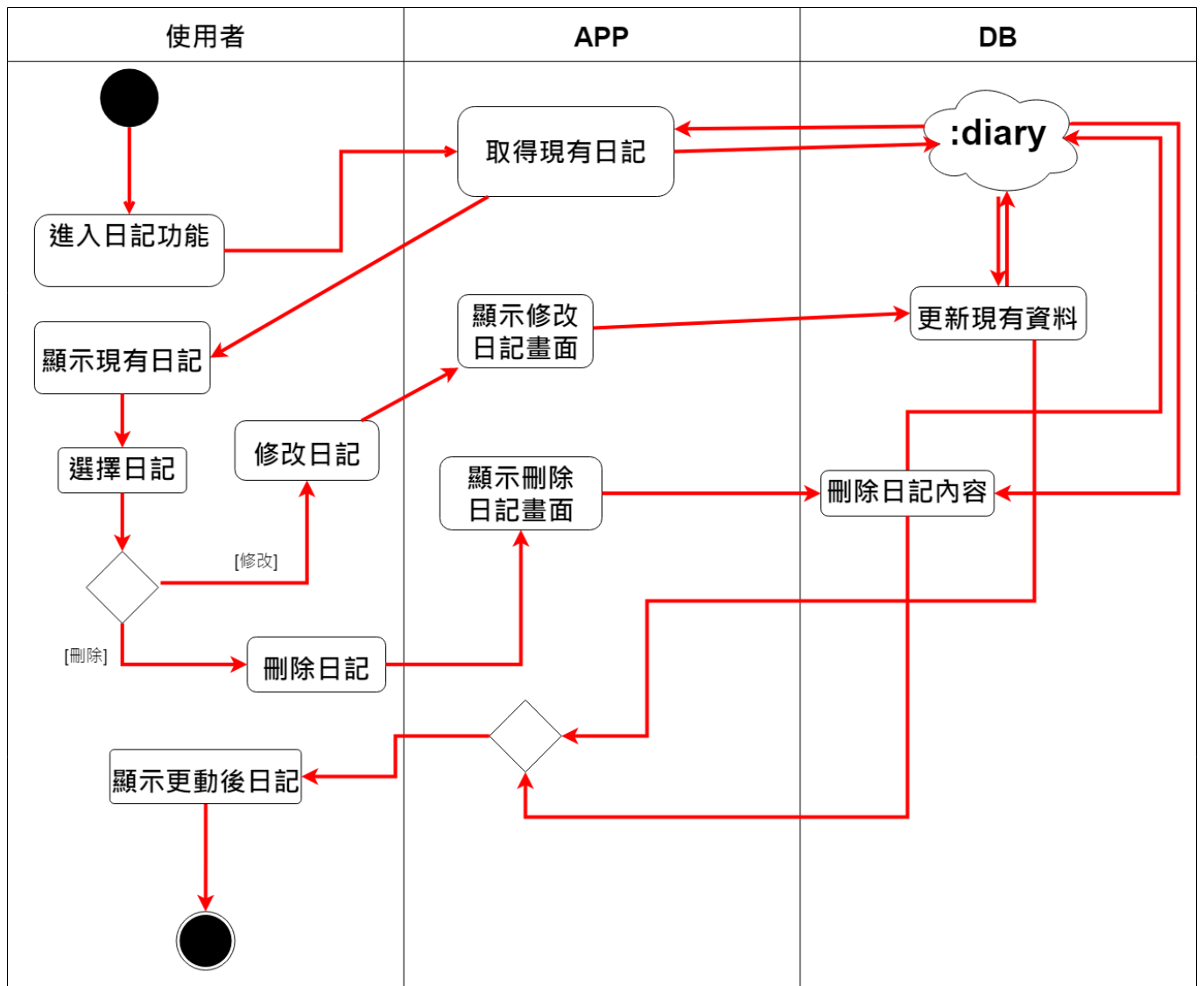
▲圖 5-3-2 登入 activity diagram



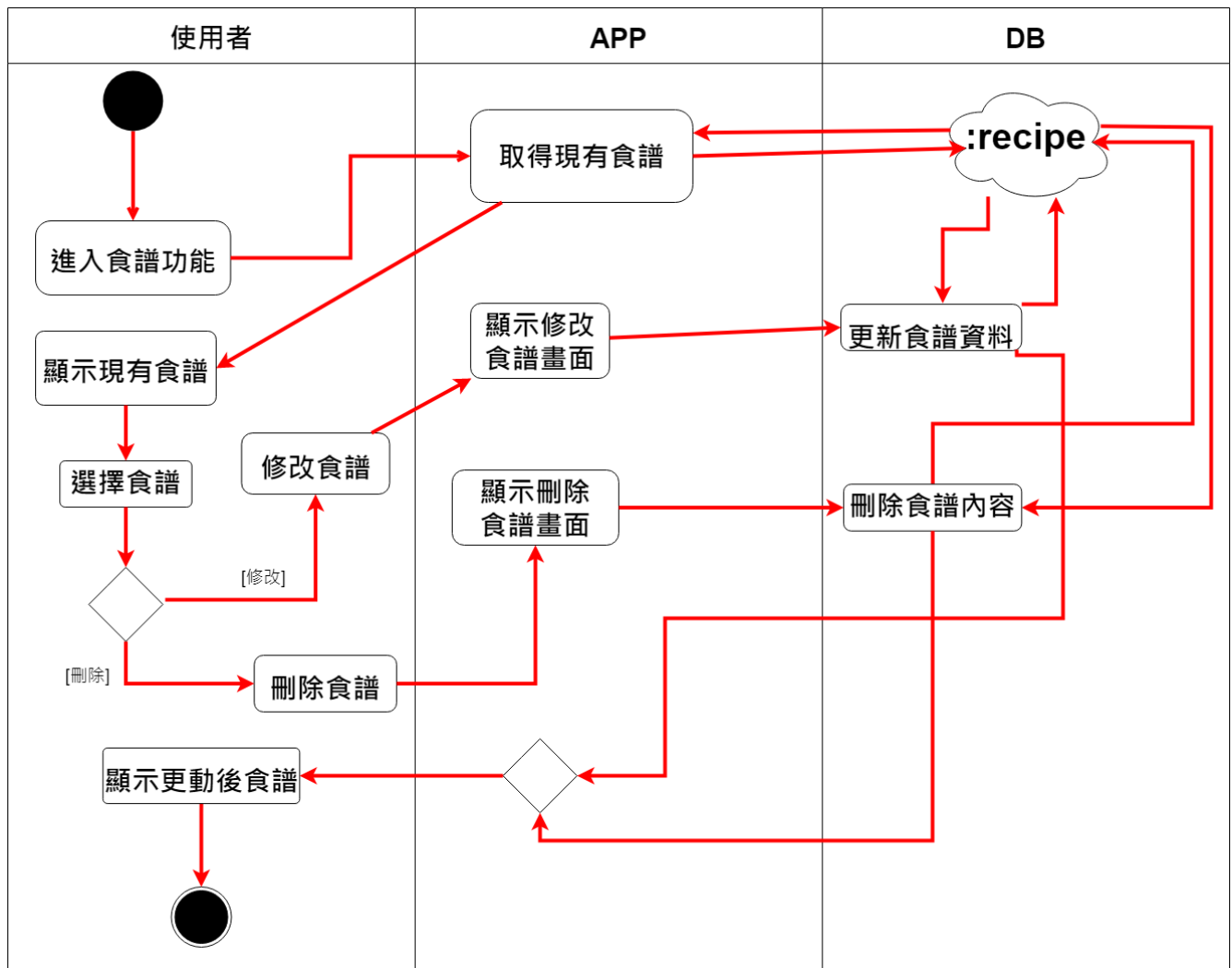
▲圖 5-3-3 新增食譜 activity diagram



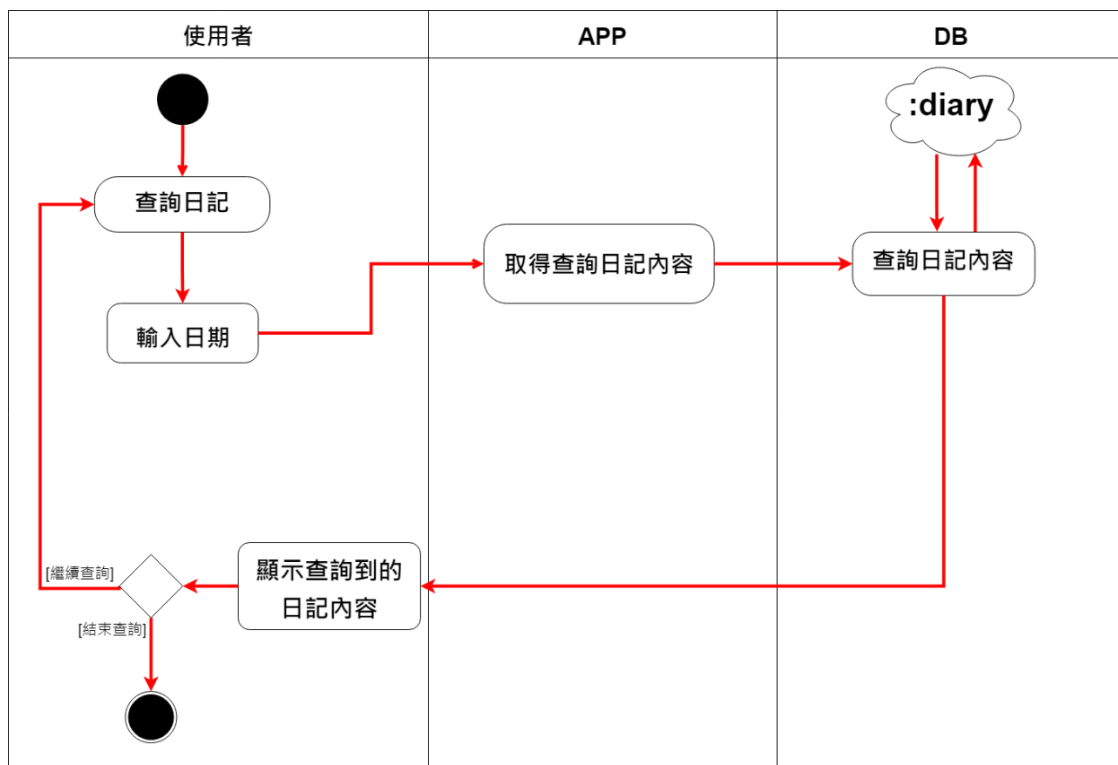
▲圖 5-3-4 新增日記 activity diagram



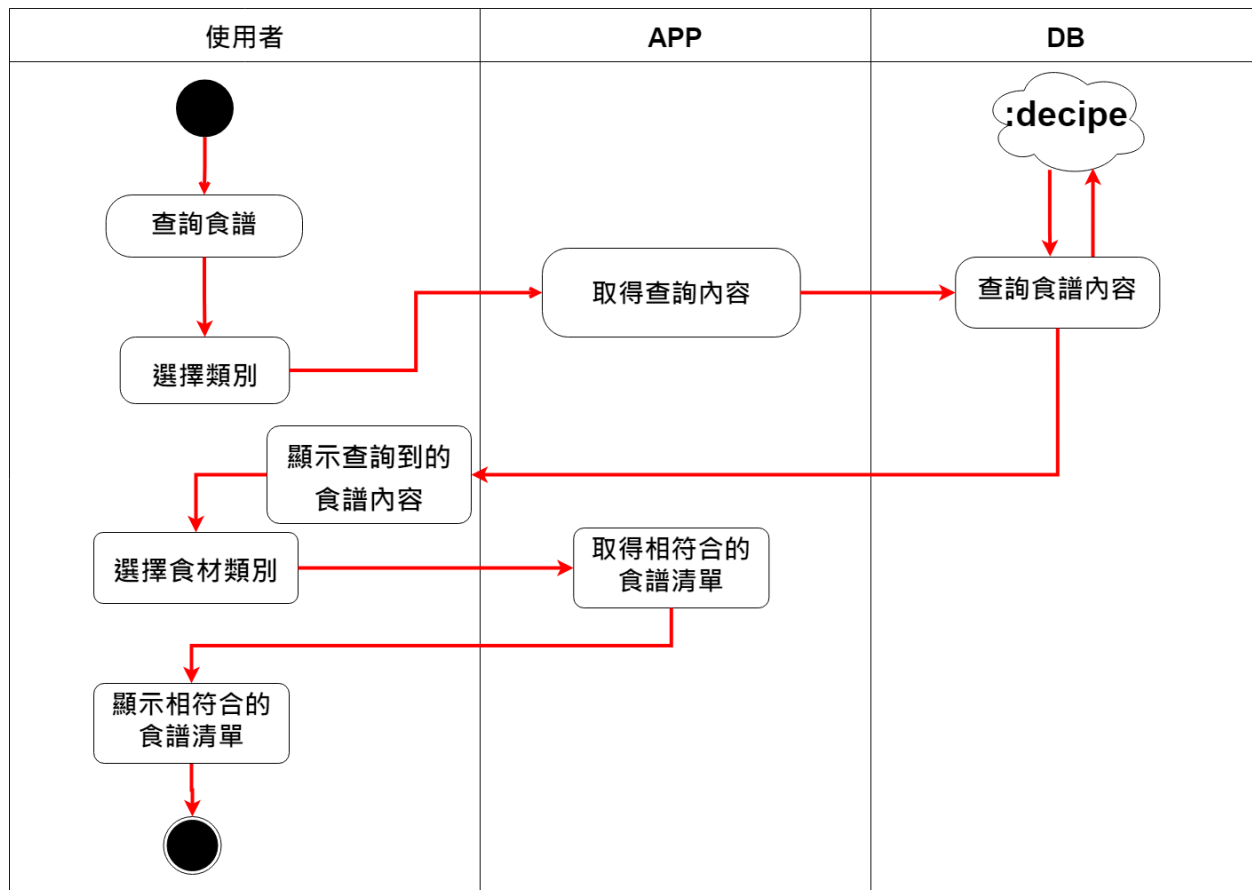
▲圖 5-3-5 維護日記 activity diagram



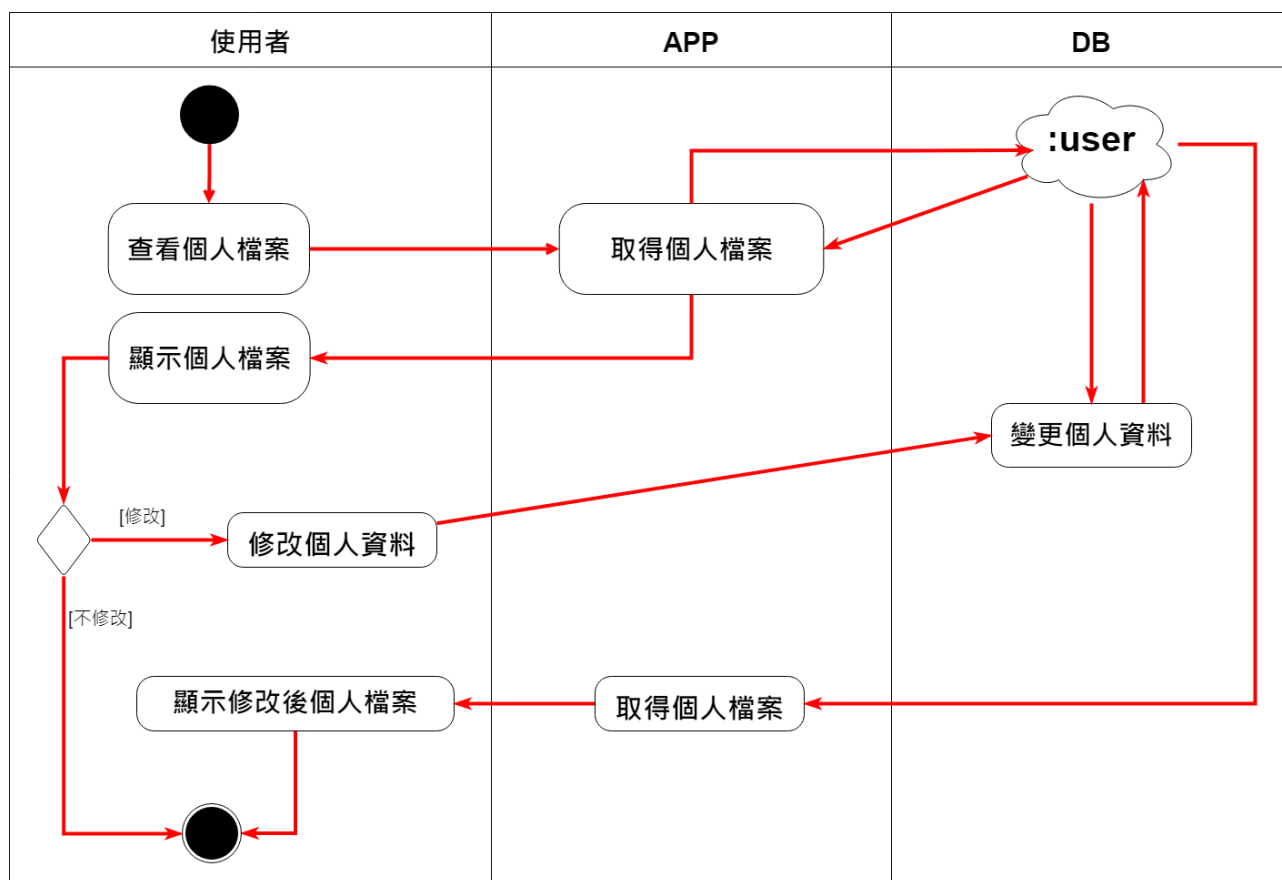
▲圖 5-3-6 維護食譜 activity diagram



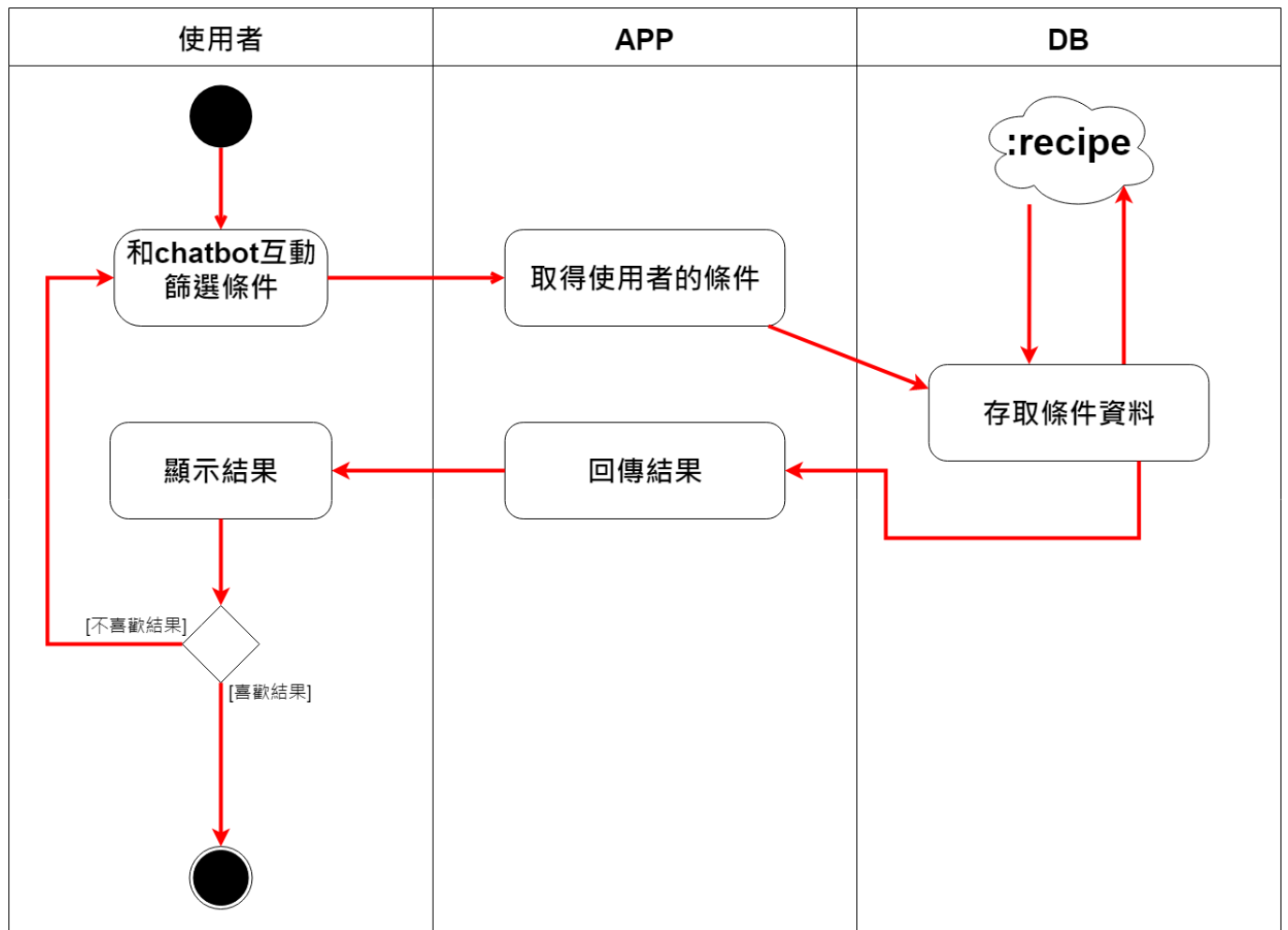
▲圖 5-3-7 查詢日記 activity diagram



▲圖 5-3-8 查詢食譜 activity diagram

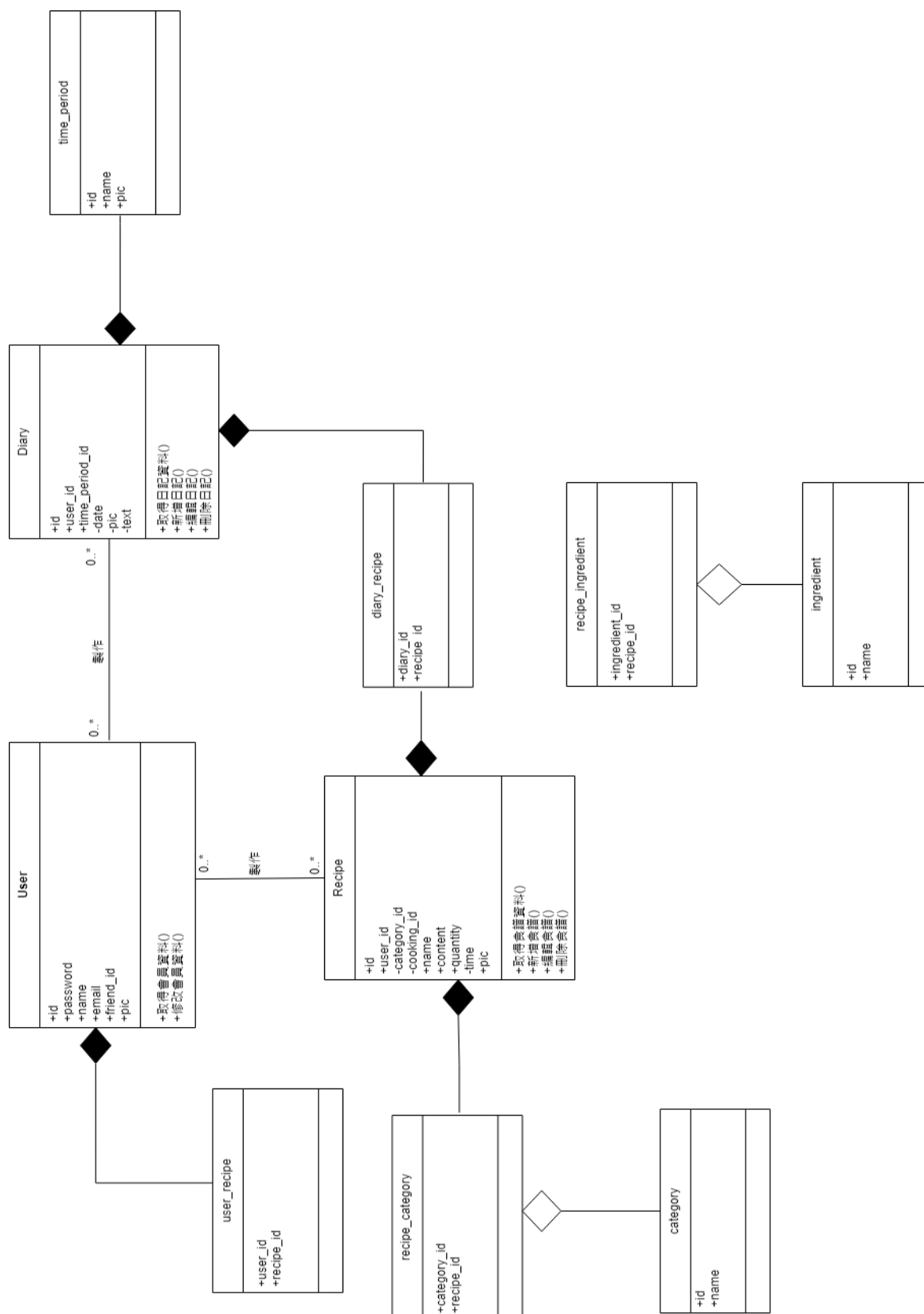


▲圖 5-3-9 修改個人檔案 activity diagram



▲圖 5-3-10 chatbot activity diagram

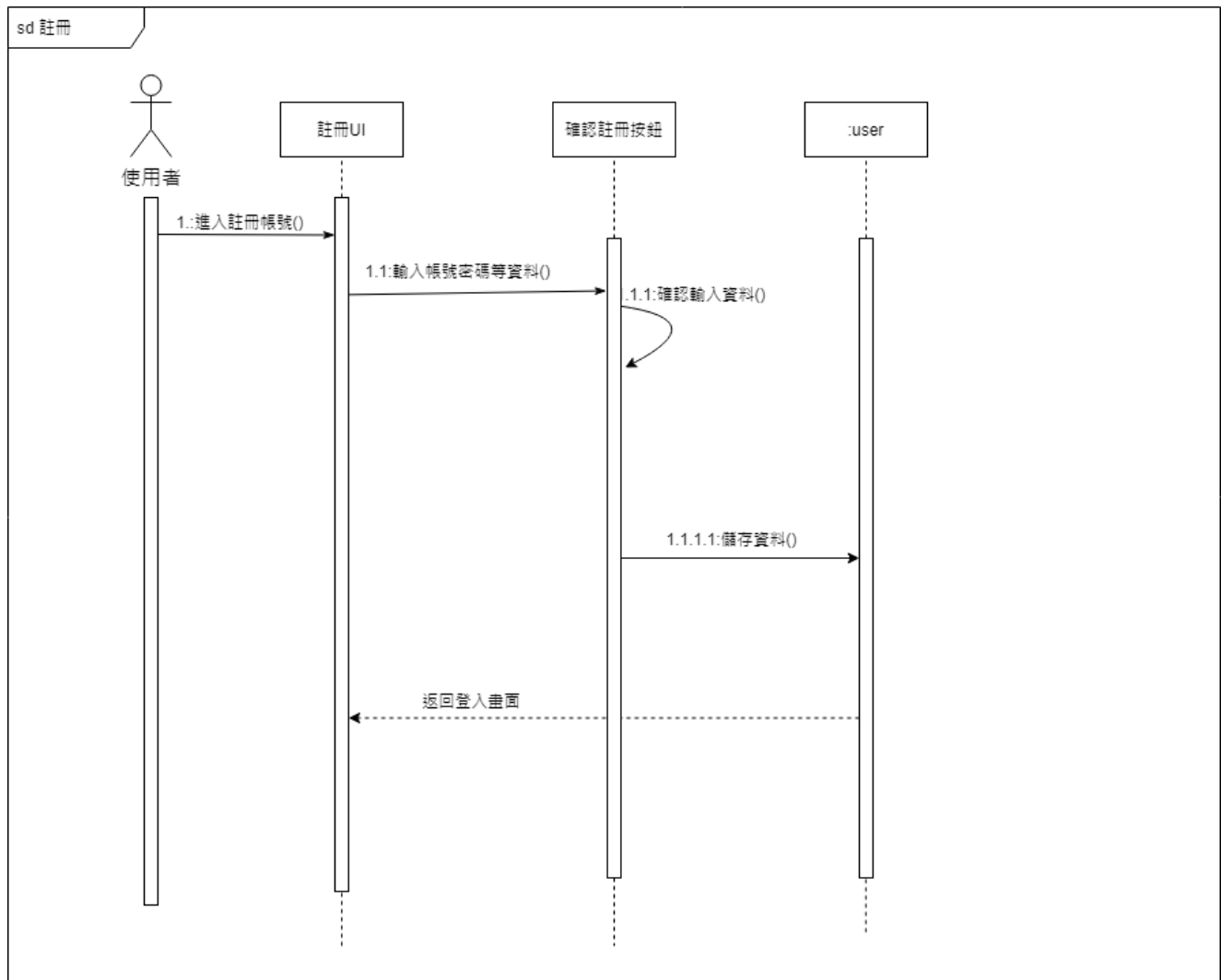
5-4 分析類別圖



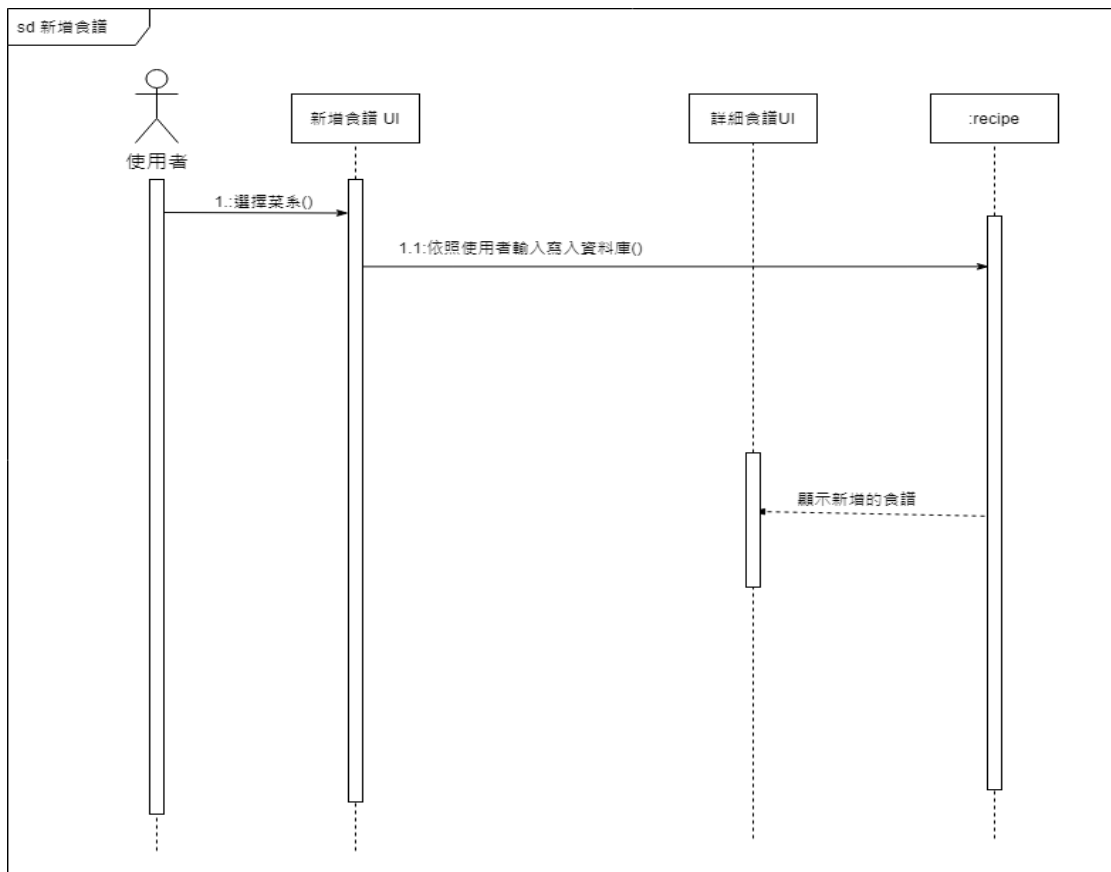
▲圖 5-4-1 分析類別圖

第 6 章 設計模型

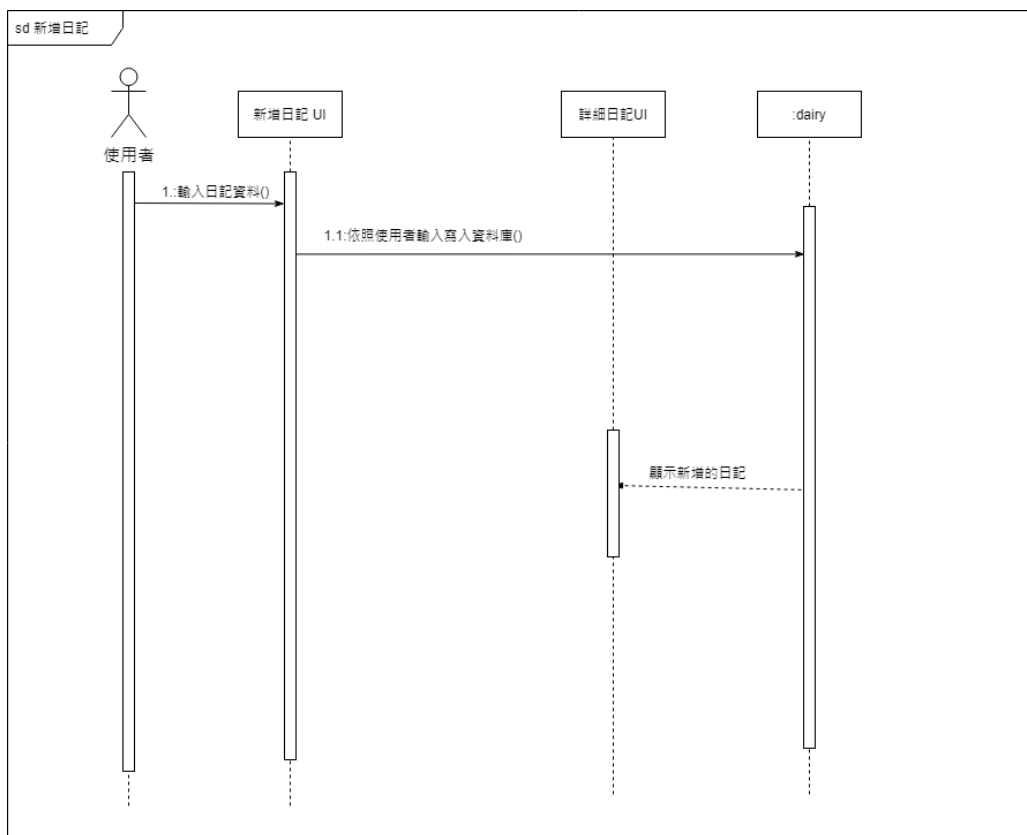
6-1 循序圖



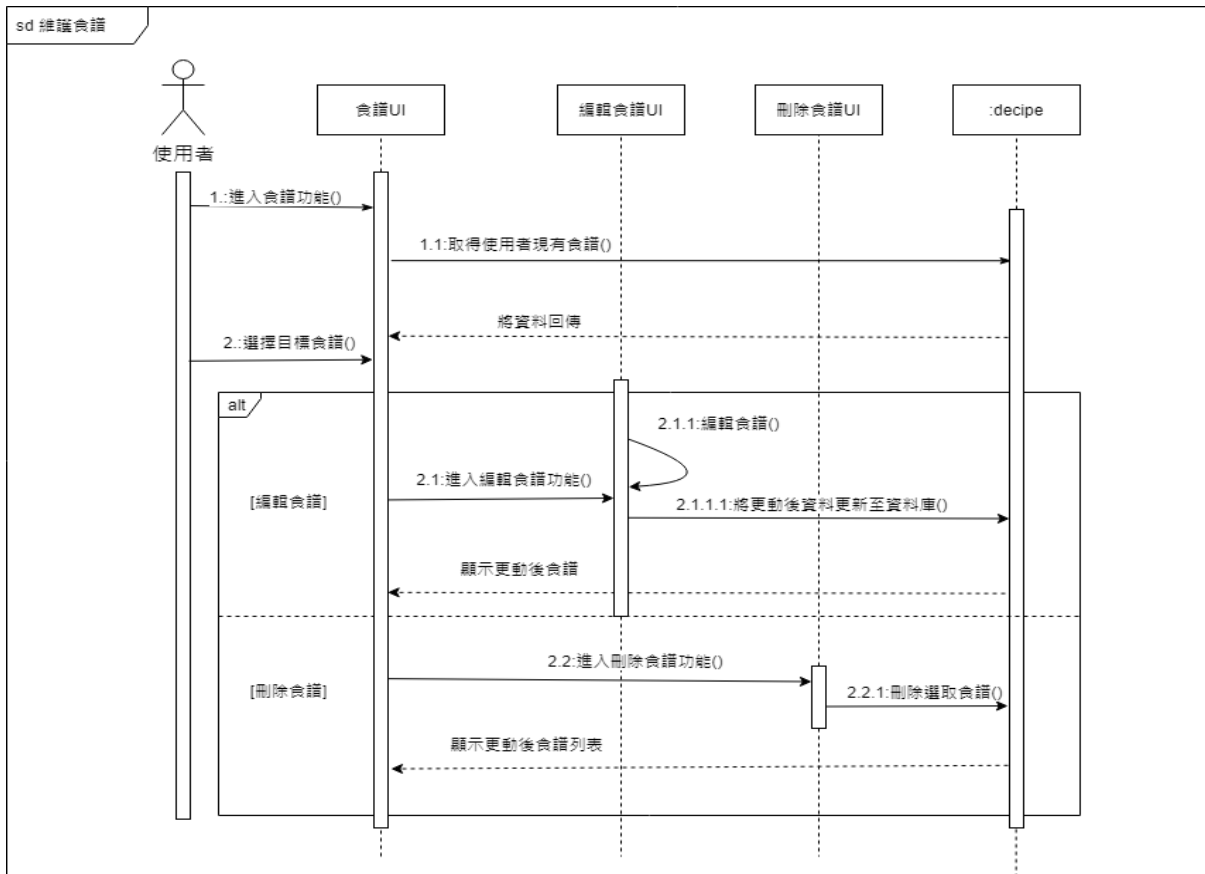
▲圖 6-1-1 註冊 sequential diagram



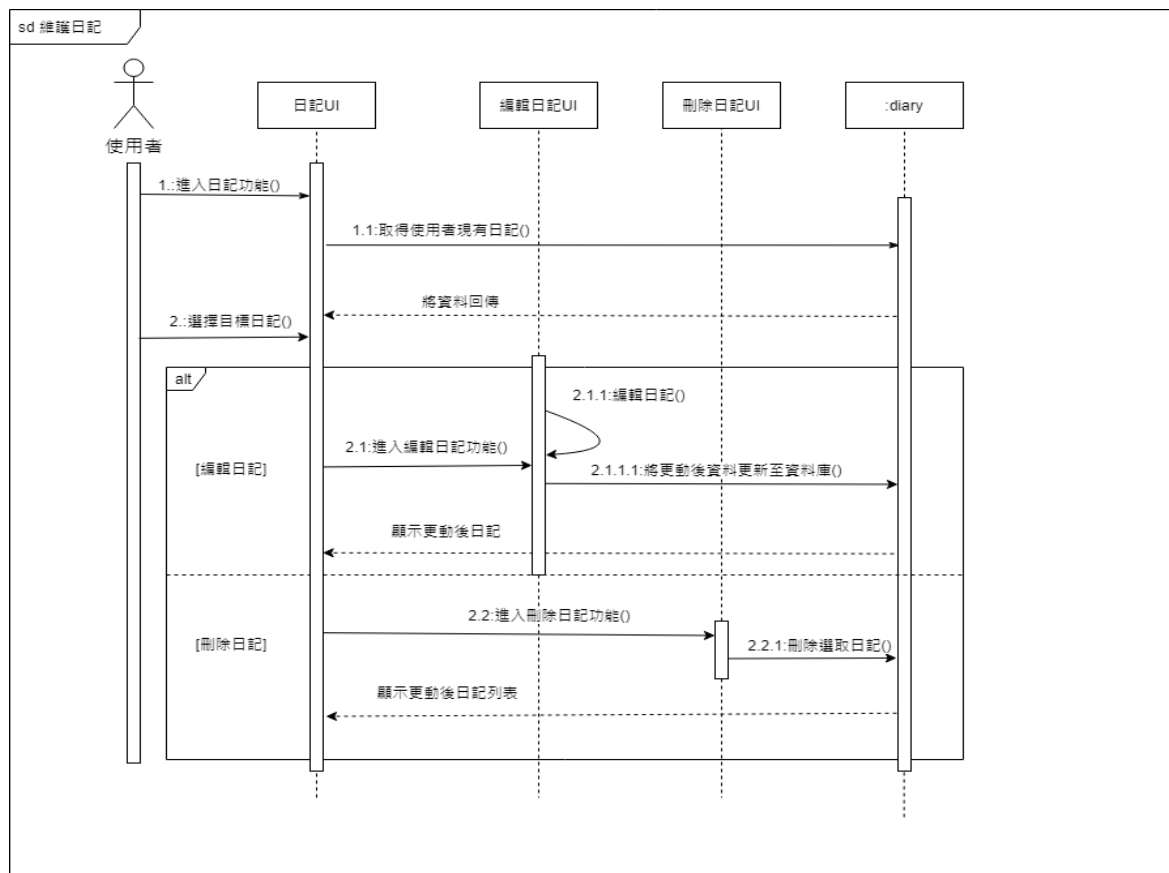
▲圖 6-1-2 新增食譜 sequential diagram



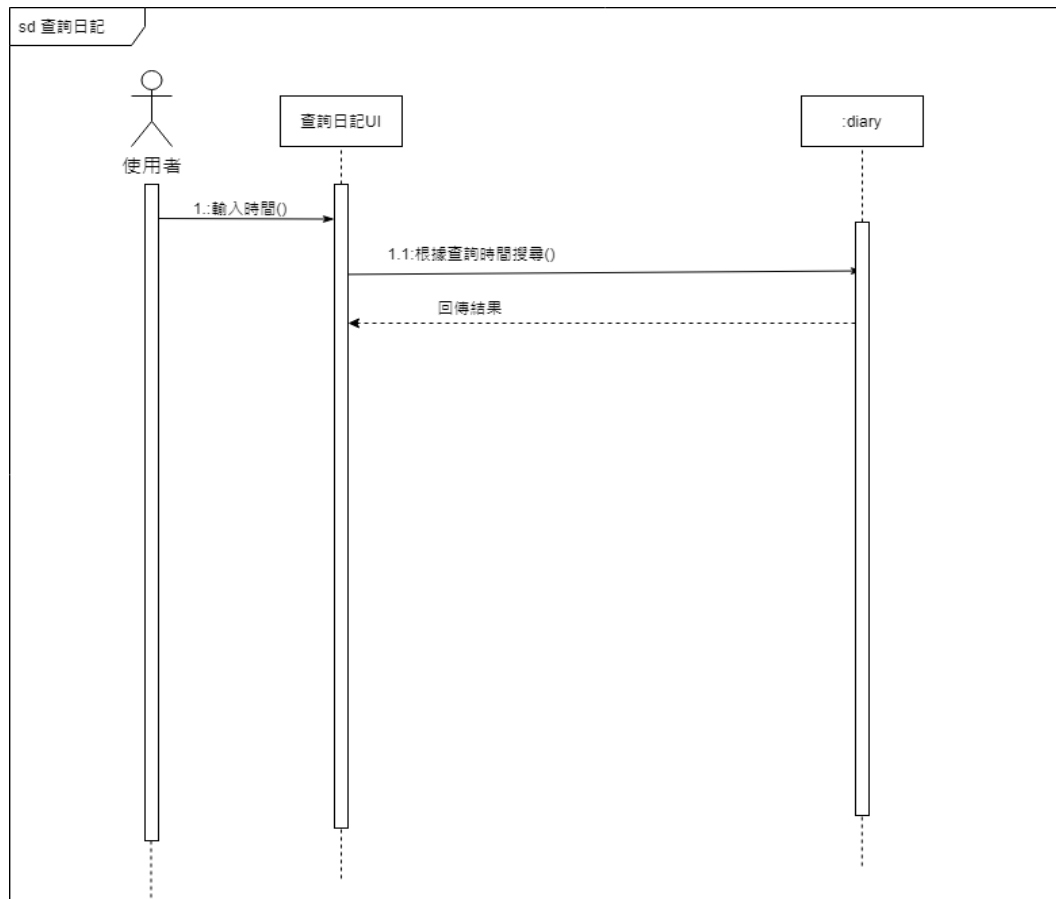
▲圖 6-1-3 新增日記 sequential diagram



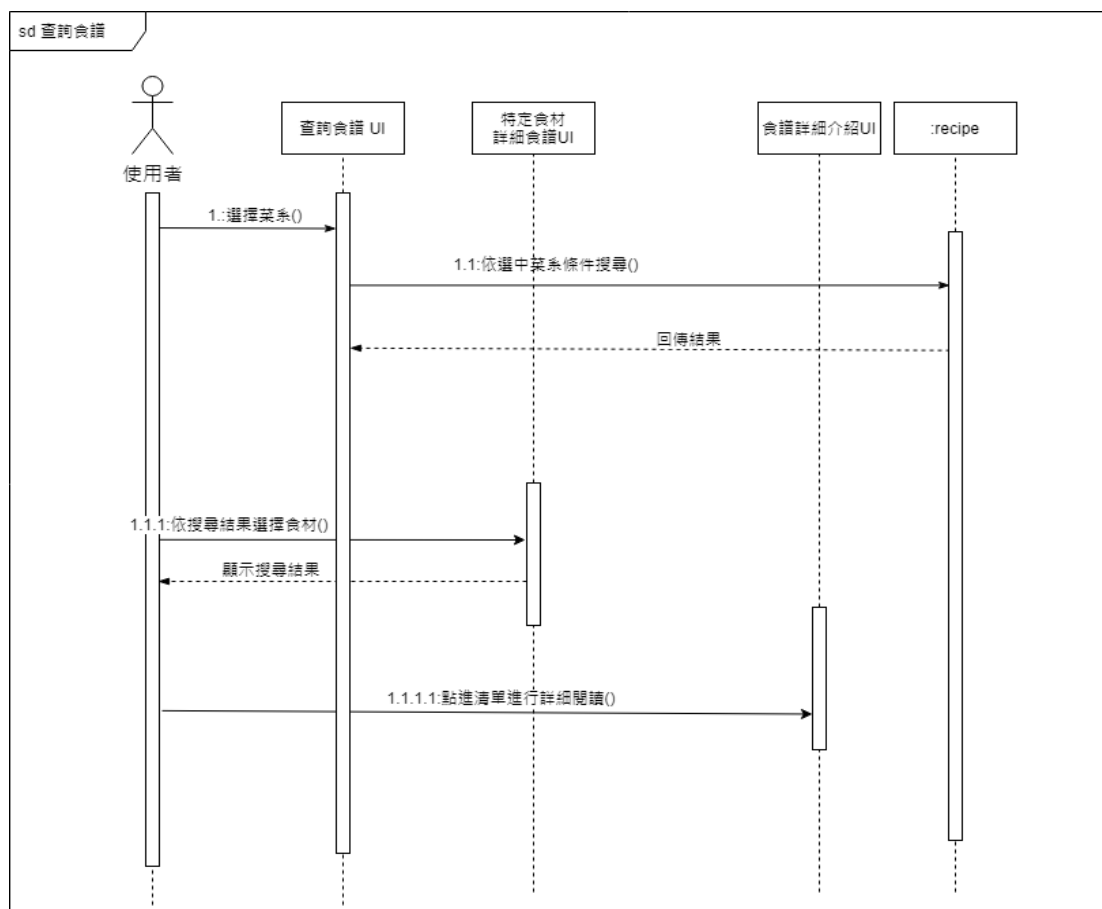
▲圖 6-1-4 維護食譜 sequential diagram



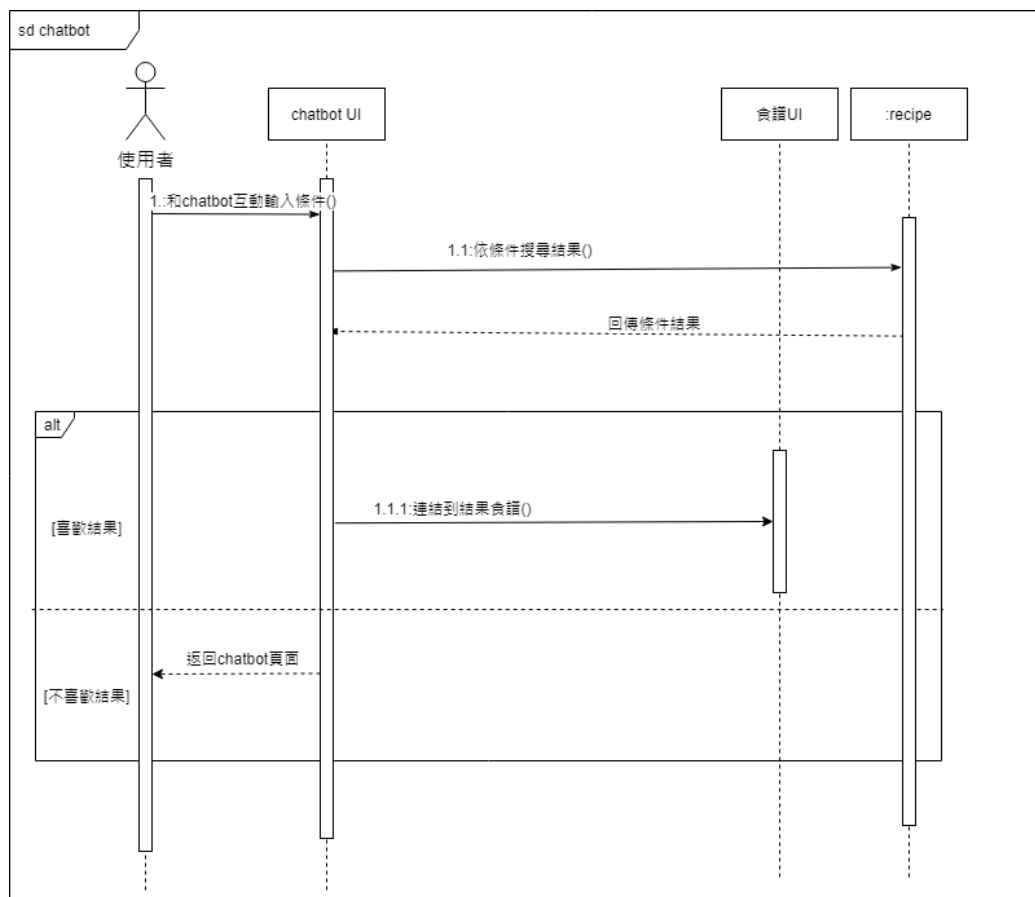
▲圖 6-1-5 維護日記 sequential diagram



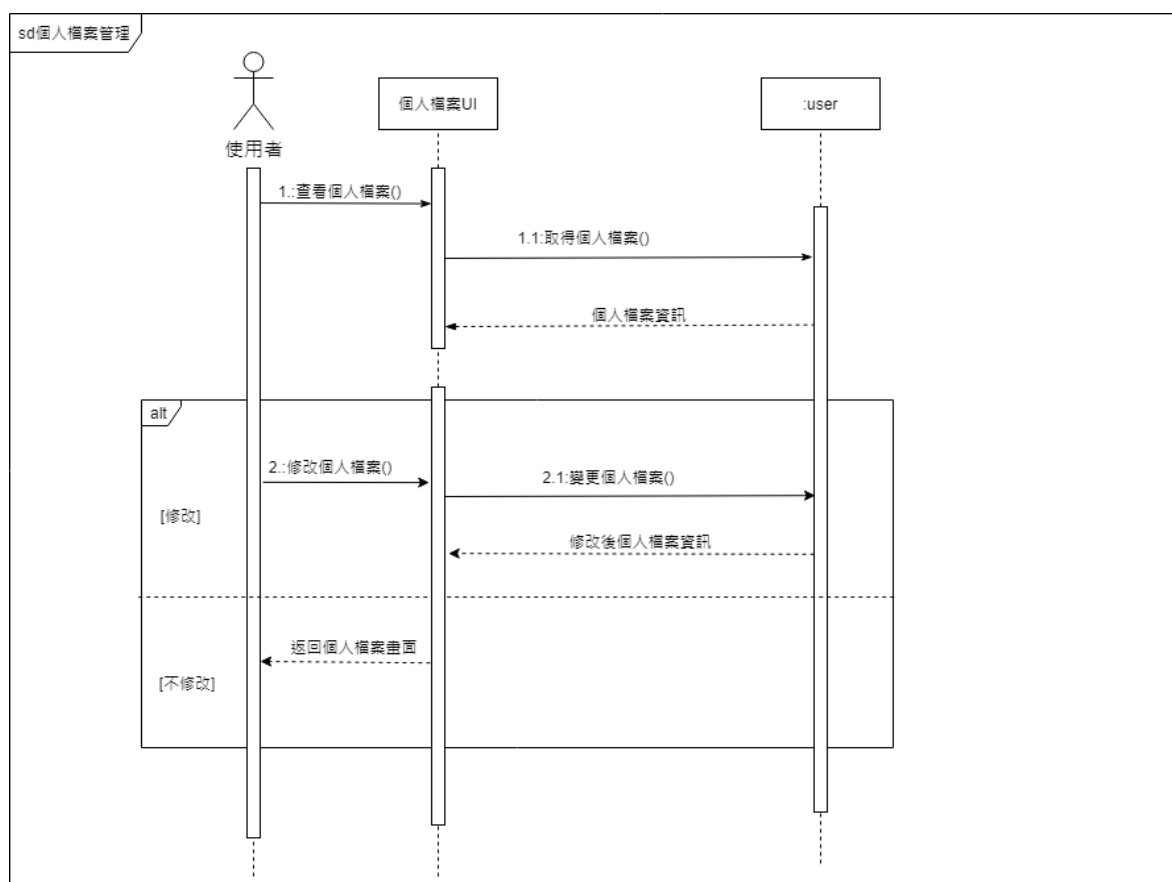
▲圖 6-1-6 查詢日記 sequential diagram



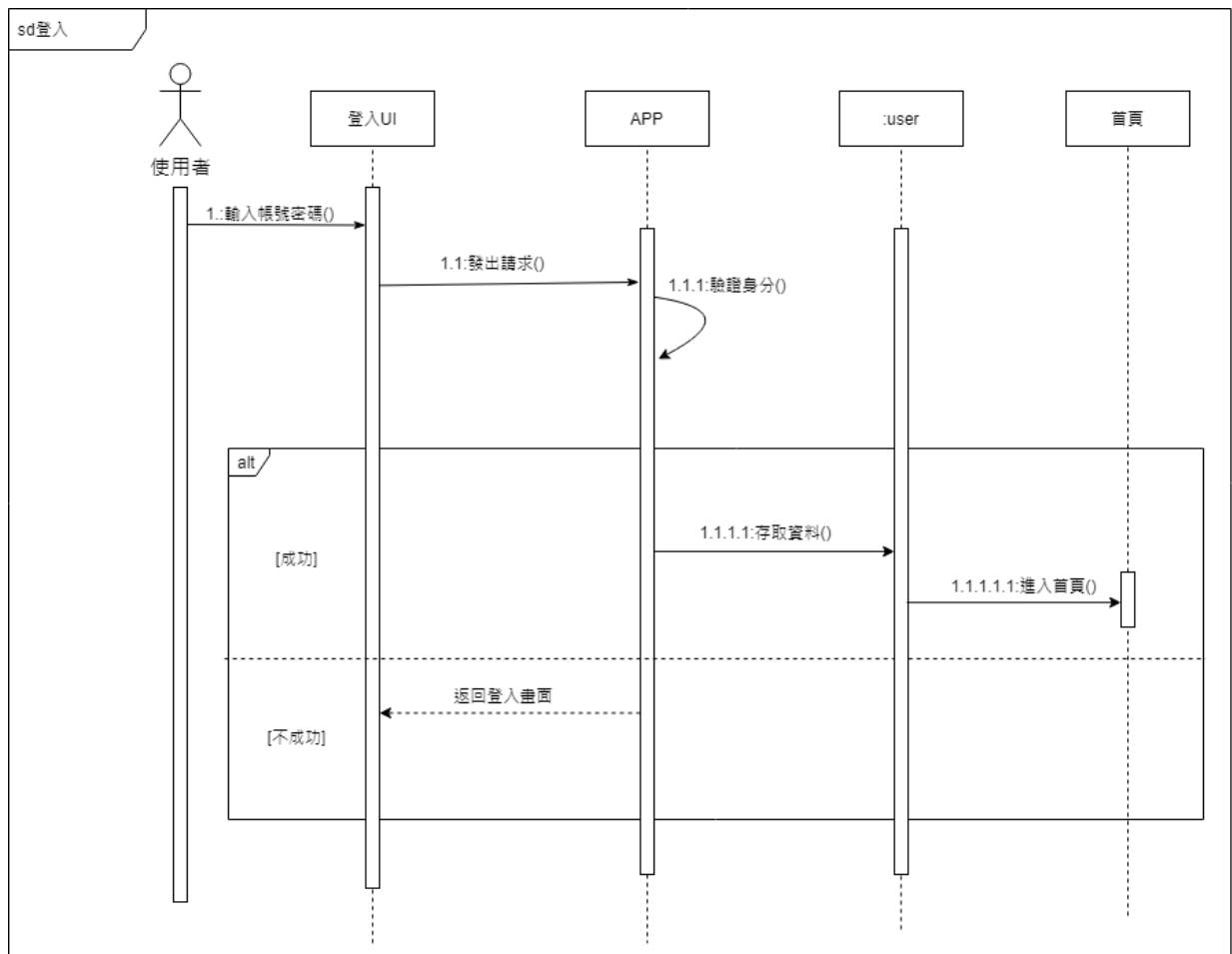
▲圖 6-1-7 查詢食譜 sequential diagram



▲圖 6-1-8 chatbot sequential diagram

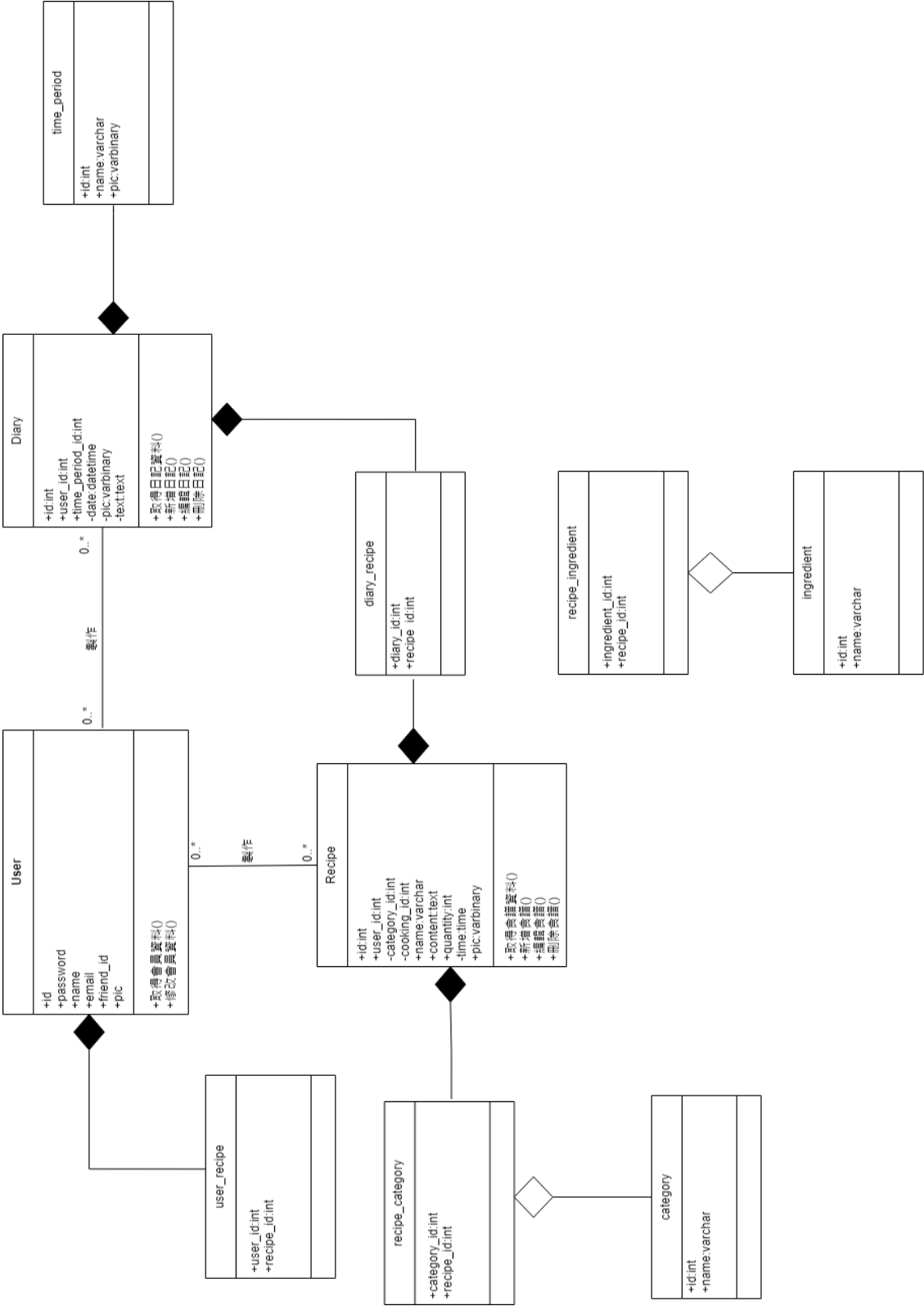


▲圖 6-1-9 個人檔案管理 sequential diagram



▲圖 6-1-10 登入 sequential diagram

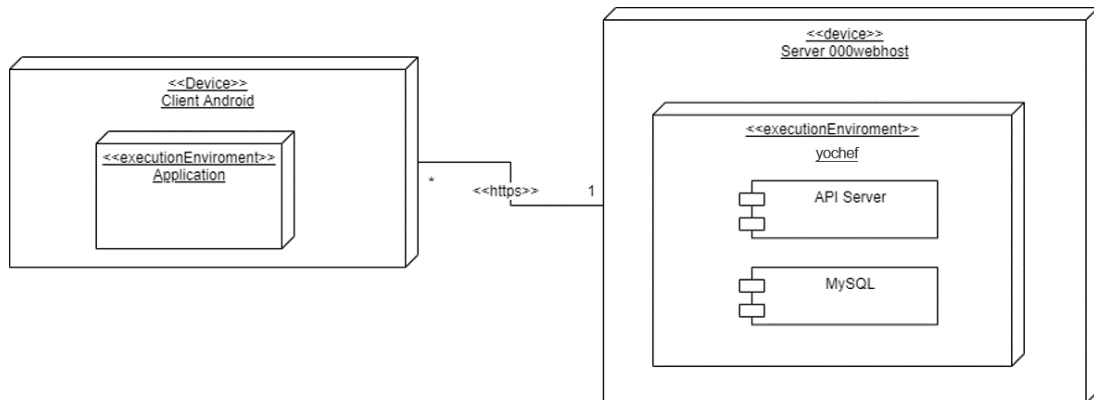
6-2 設計類別圖



▲圖 6-2-1 設計類別圖

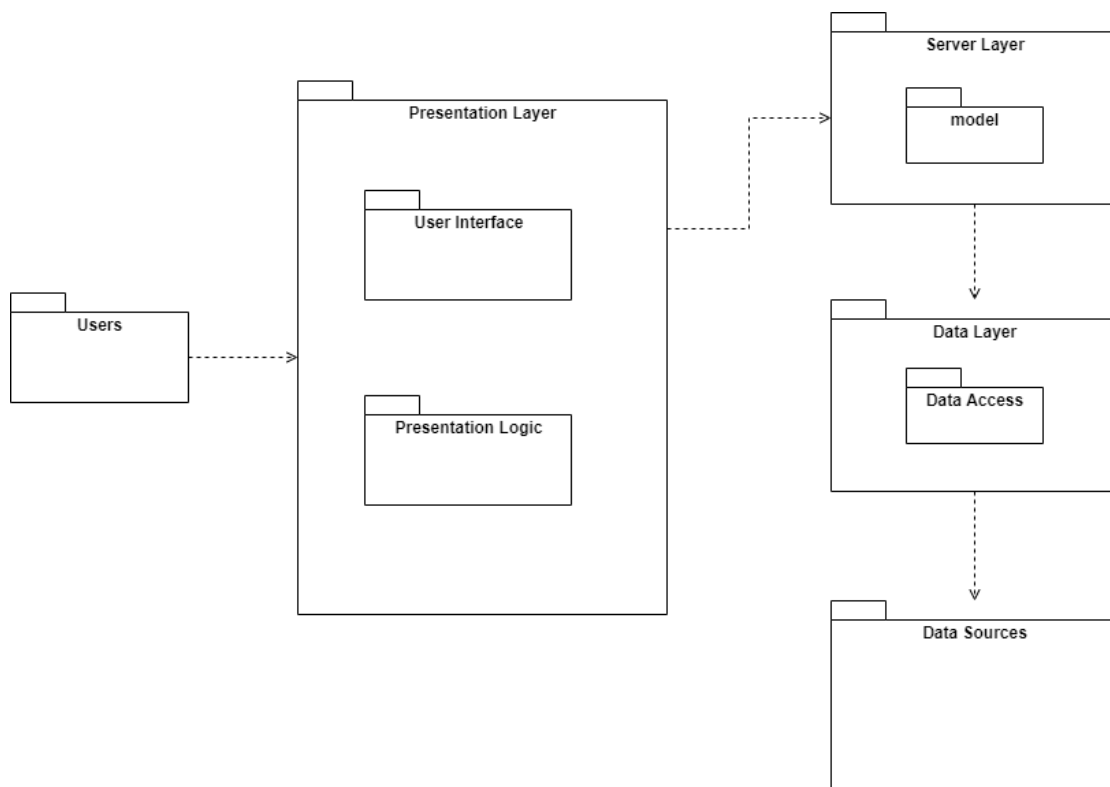
第七章 資料模型

7-1 佈署圖(Deployment diagram)



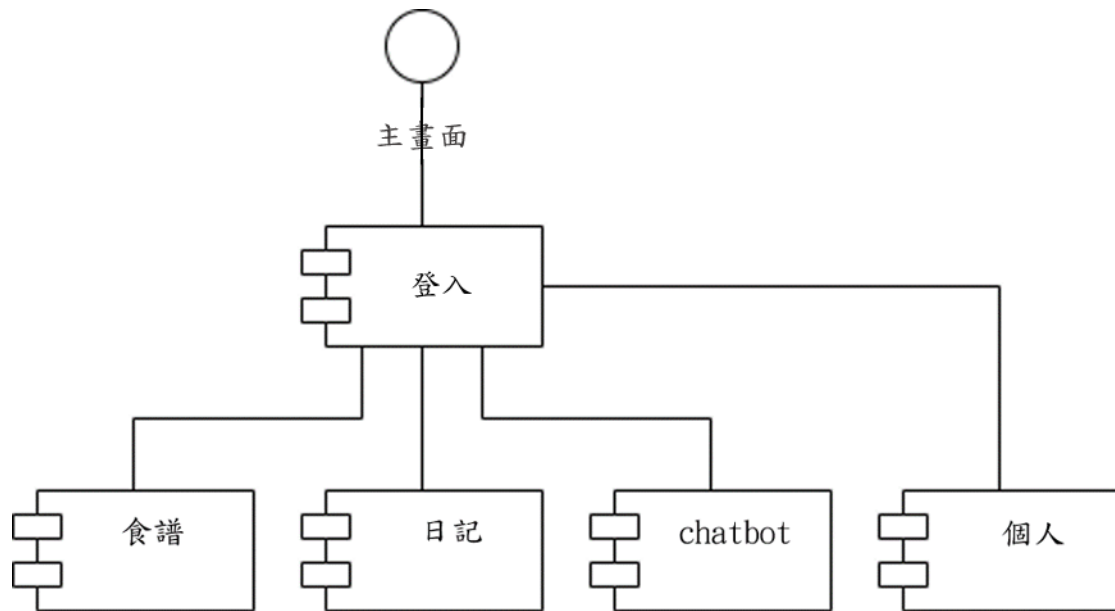
▲圖 7-1-1 佈署圖 deployment diagram

7-2 套件圖(Package diagram)



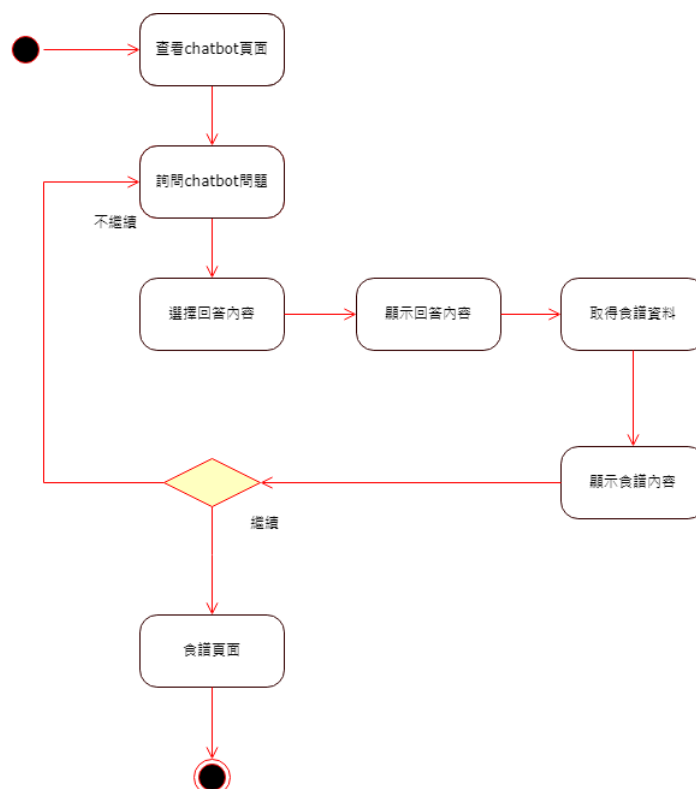
▲圖 7-2-1 套件圖 Package diagram

7-3 元件圖(Component diagram)

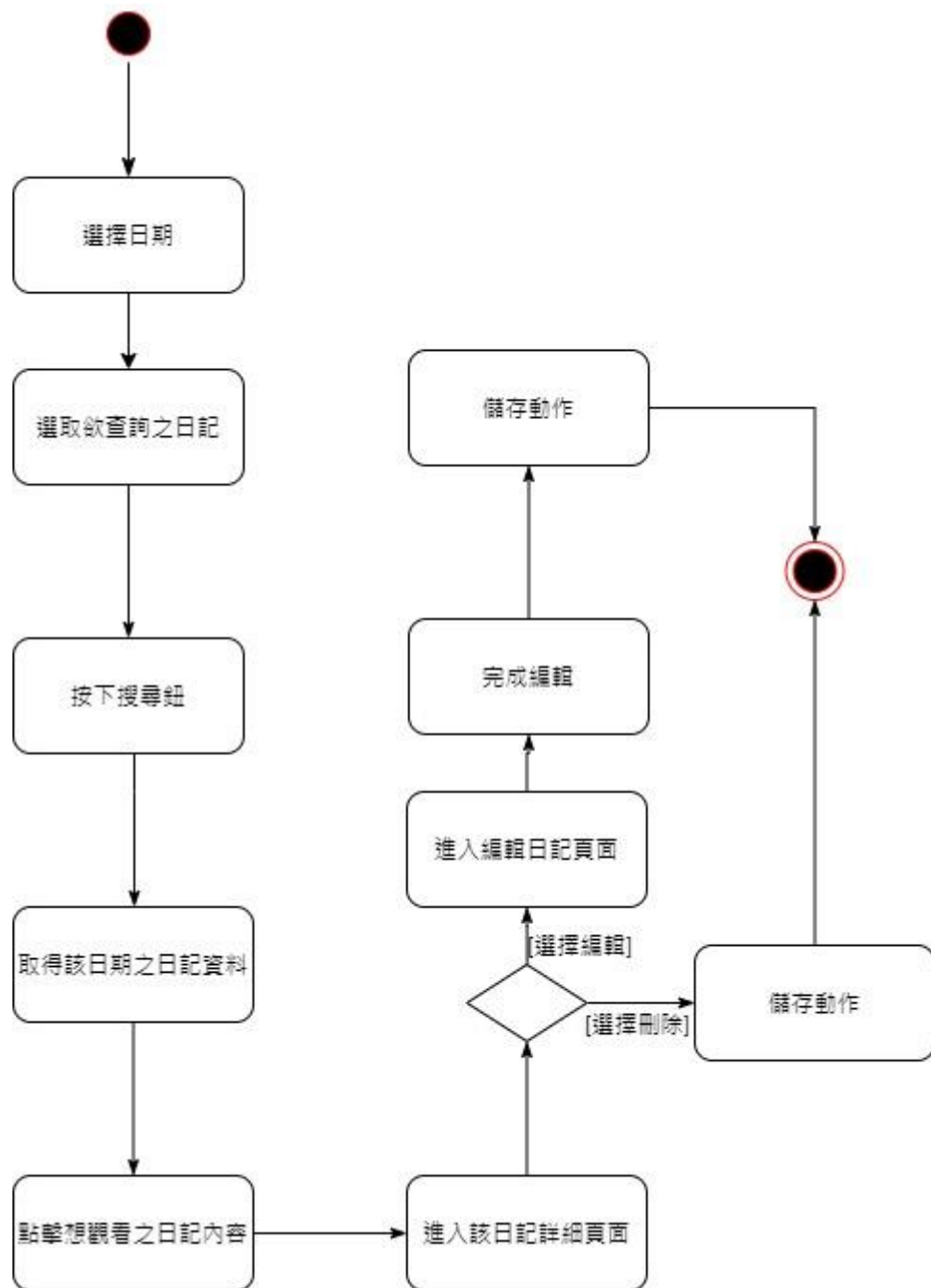


▲圖 7-3-1 元件圖 component diagram

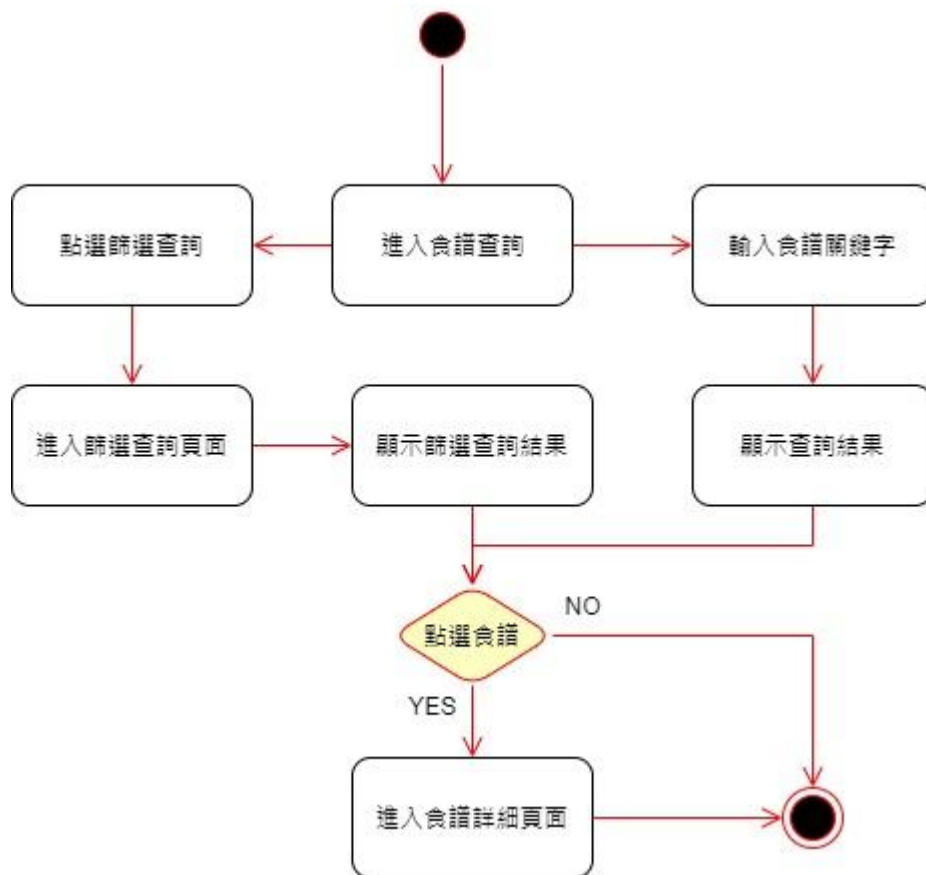
7-4 狀態圖(State machine)



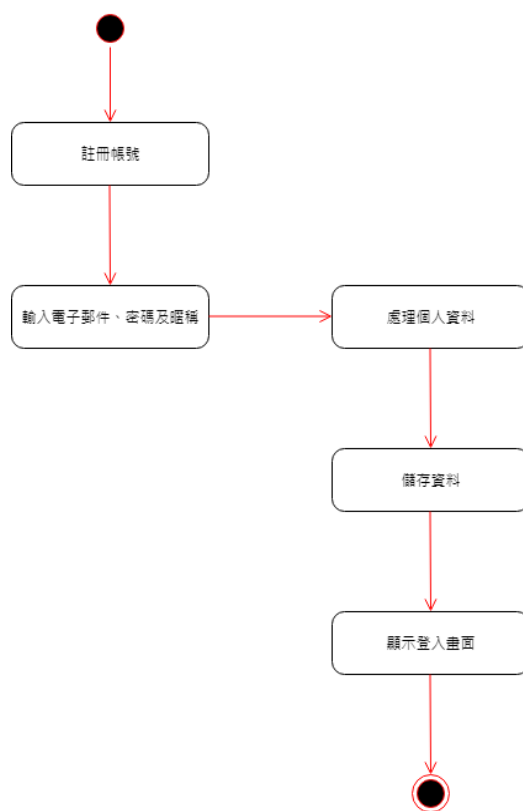
▲圖 7-4-1 chatbot state machine



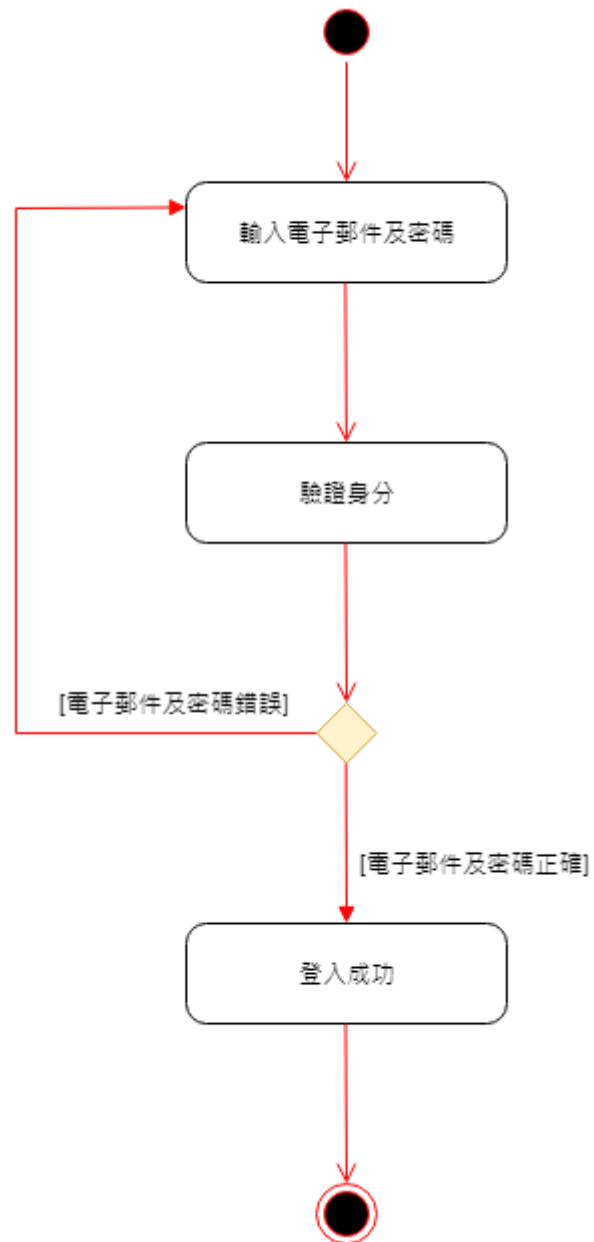
▲圖 7-4-2 日記 state machine



▲圖 7-4-3 食譜 state machine



▲圖 7-4-4 註冊 state machine



▲圖 7-4-5 登入 state machine

第八章 資料庫設計

8-1 資料庫關聯表



▲圖 8-1-1 資料庫關聯表

8-2 表格及其 Meta data

▼表 8-2-1 使用者資料表(user)

資料表名稱	user	
資料表描述	使用者	
欄位名稱	型態	註解
id	int(11)	使用者編號
name	varchar(50)	姓名
email	varchar(50)	電子郵件
password	varchar(50)	密碼
user_pic	varchar(300)	個人圖片
user_cover	varchar(300)	個人背景

▼表 8-2-2 日記資料表(diary)

資料表名稱	diary	
資料表描述	日記	
欄位名稱	型態	註解
id	int(11)	日記編號
user_id	int(11)	使用者編號
time_period_id	text	餐期
date	text	日期戳記
title	text	日記標題
text	text	日記內文
picture	varchar(225)	日記圖片
recipe_id	varchar(20)	日記對應的食譜編號

▼表 8-2-3 收藏資料表(likes)

資料表名稱	likes	
資料表描述	食譜收藏	
欄位名稱	型態	註解
id	int(11)	收藏流水編號
user_id	int(11)	使用者 id
recipe_id	int(11)	食譜 id

▼表 8-2-4 類別資料表(category)

資料表名稱	category	
資料表描述	類別	
欄位名稱	型態	註解
cate_id	int(11)	類別編號
category	varchar(10)	類別
catePic	varchar(255)	類別圖片

▼表 8-2-5 烹調資料表(cooking)

資料表名稱	cooking	
資料表描述	烹調方式	
欄位名稱	型態	註解
cookid	int(11)	烹調方式編號
cooking	varchar(10)	烹調方式
cookPic	varchar(255)	烹調方式圖片

▼表 8-2-6 食材資料表(ingredient)

資料表名稱	ingredient	
資料表描述	食材	
欄位名稱	型態	註解
ingreid	int(11)	食材編號
Ingre	varchar(50)	食材名稱

▼表 8-2-7 菜系資料表(international)

資料表名稱	international	
資料表描述	菜系	
欄位名稱	型態	註解
interid	int(11)	菜系編號
inter	varchar(10)	菜系
interIMG	varchar(255)	菜系圖片

▼表 8-2-8 份量資料表(quantity)

資料表名稱	quantity	
資料表描述	份量	
欄位名稱	型態	註解
quanid	int(11)	份量編號

recipe_id	int(11)	食譜編號
ingredient_id	int(11)	食材編號
ingredient_quantity	varchar(20)	份量

▼表 8-2-9 食譜資料表(recipe)

資料表名稱	recipe	
資料表描述	食譜	
欄位名稱	型態	註解
id	int(11)	食譜編號
userid	varchar(11)	發佈者編號
category_id	varchar(10)	類別
cooking_id	varchar(10)	烹調方式
recipe_name	varchar(20)	食譜名稱
quantity	varchar(10)	份量
release_time	varchar(100)	發佈日期
international	varchar(10)	菜系
vegetarian	varchar(10)	葷素
all_star	varchar(10)	累計評分
average	varchar(10)	平均評分
rating	varchar(10)	評分次數
introduction	varchar(300)	食譜簡介
cooking_time	varchar(10)	烹調時間
likes	varchar(10)	收藏數
privacy	varchar(10)	公開

▼表 8-2-10 食譜照片資料表(recipe_picture)

資料表名稱	recipe_picture	
資料表描述	食譜照片	
欄位名稱	型態	註解
id	int(11)	照片編號
recipe_id	int(11)	食譜編號
recipe_picture	varchar(255)	食譜照片

▼表 8-2-11 食譜步驟資料表(recipe_steps)

資料表名稱	recipe_steps	
資料表描述	食譜步驟	
欄位名稱	型態	註解
id	int(11)	照片編號
recipe_id	int(11)	食譜編號
step_number	int(11)	步驟順序
step_description	varchar(255)	步驟敘述
step_picture	varchar(255)	步驟照片

第九章 程式規格

9-1 元件清單及其規格描述

▼表 9-1-1 使用者管理

使用者管理	
檔案名稱	功能說明
LoginActivity.java	登入/所有個人資料載入
Register.java	註冊
GoogleLoginActivity.java	第三方登入
ForgotPassActivity.java	忘記密碼
Updatepassword.java	變更密碼

▼表 9-1-2 主功能

主功能	
檔案名稱	功能說明
fragment_recipe.java	食譜功能頁面
fragment_diary.java	日記頁面+查詢
fragment_chatbot.java	聊天機器人頁面
fragment_profile.java	個人檔案總列
MainActivity.java	控制 fragment 檔案
MainPageAdapter.java	控制 fragment 切換功能

▼表 9-1-3 日記管理

日記管理	
檔案名稱	功能說明
diary_detail.java	日記資料詳細頁面
diary_edit.java	編輯日記
new_diary.java	新增日記
DiaryAdapter.java	日記功能 class
SearchDiary.java	日記 array list

▼表 9-1-4 個人檔案

個人檔案	
檔案名稱	功能說明
EditProfile.java	編輯個人資料
ProfileAdapter.java	個人資料功能
SearchProfile.java	個人資料 array list
Edit_userRecipe.java	編輯個人發布之食譜
post_recipe_list.java	個人發布食譜清單介面
SearchRecipePost.java	個人發布食譜 array list
recipePostAdapter.java	個人發布食譜功能
like_recipe_list.java	個人收藏食譜清單介面
likeRecipeAdapter.java	個人收藏食譜功能
SearchLikeRecipe.java	個人收藏食譜 array list
NotiActivity.java	通知推播設定

▼表 9-1-5 食譜功能

食譜功能	
檔案名稱	功能說明
AllFragment.java	篩選食譜功能
CreateActivity.java	新增食譜分頁控制
creIntroFragment.java	新增食譜簡介
creListFragment.java	新增食譜清單
creStepFragment.java	新增食譜步驟
detailActivity.java	食譜資料 viewPager 分頁控制
FilterActivity.java	篩選食譜搜尋結果
Fragment_recipe.java	食譜功能首頁
introFragment.java	食譜詳細資料
listFragment.java	食譜清單資料
stepFragment.java	食譜步驟資料
searchRecipeActivity.java	食譜關鍵字搜尋

▼表 9-1-6 chatbot 功能

Chatbot 功能	
檔案名稱	功能說明
ChatMessageAdapter.java	傳送機器人訊息功能
ChatMessage.java	機器人訊息 Model

▼表 9-1-7 工具

工具	
檔案名稱	功能說明
creList.java	新增清單資料儲存
creStep.java	新增步驟資料儲存
deListAdapter.java	食譜清單資料顯示在 RecyclerView 中
deStepAdapter.java	食譜步驟資料顯示在 RecyclerView 中
FilterAdapter.java	篩選清單顯示在 ExpandableListView 中
FilterHeader.java	篩選資料儲存
FilterRecipeAdapter.java	篩選食譜資料顯示在 RecyclerView 中
getRecipeAdapter.java	食譜資料顯示在 RecyclerView 中
List.java	食譜清單資料儲存
Recipe.java	食譜資料儲存
searchRecipeAdapter.java	食譜資料顯示在 RecyclerView 中
Step.java	食譜步驟資料儲存
creList.java	新增清單資料儲存
creStep.java	新增步驟資料儲存

9-2 其他附屬的各種文件

▼表 9-2-1 外部元件清單

套件名稱	版本	功能
android.volley:volley	1.2.1	Volley 套件
github.bumptech.glide:glide	4.12.0	glide 套件
github.bumptech.glide:compiler	4.12.0	glide 套件
karumi:dexter	6.2.3	dexter 套件
github.dhaval2404:imagepicker	2.1	imagepicker 套件
jakewharton:butterknife	10.2.3	Butterknife 套件
migueltcatalan:materialsearchview	1.4.0	搜尋套件
androidx.appcompat:appcompat	1.3.0	Androidx 套件
com.google.android.material:material	1.3.0	Google 套件
com.google.android.gms:play-services-auth	19.2.0	Google Login API 套件
com.android.support:gridlayout-v7	29.0.0	
com.android.support:recyclerview-v7	30.0.0	Recyclerview 套件
androidx.recyclerview:recyclerview:1.0.0	1.0.0	Androidx 套件
com.github.bumptech.glide:glide:4.12.0	4.12.0	Glide 套件
androidx.core:core:1.6.0	1.6.0	Core 套件
com.karumi:dexter:6.2.3	6.2.3	dexter 套件
com.android.volley:volley:1.2.0	1.2.0	Volley 套件
androidx.legacy:legacy-support-v4	1.0.0	support libraries
junit:junit:4.+	4.+	單元測試工具
androidx.test.ext:junit	1.1.3	
me.himanshusoni.chatmessageview:chat-message-view	1.0.3	Chatview 套件
androidx.fragment:fragment:1.3.0-alpha08	1.3.0	fragment 套件

▼表 9-2-2 LoginActivity.java

程式名稱	LoginActivity.java
目的	登入 / 所有個人資料載入
<pre> package com.example.home; import androidx.appcompat.app.AppCompatActivity; import androidx.appcompat.widget.Toolbar; import android.app.ProgressDialog; import android.content.Intent; import android.net.Uri; import android.os.Bundle; import android.util.Log; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.TextView; import android.widget.Toast; import com.android.volley.AuthFailureError; import com.android.volley.Request; import com.android.volley.RequestQueue; import com.android.volley.Response; import com.android.volley.VolleyError; import com.android.volley.toolbox.StringRequest; import com.android.volley.toolbox.Volley; import com.google.android.gms.auth.api.signin.GoogleSignIn; import com.google.android.gms.auth.api.signin.GoogleSignInAccount; import com.google.android.gms.auth.api.signin.GoogleSignInClient; import com.google.android.gms.auth.api.signin.GoogleSignInOptions; import com.google.android.gms.common.SignInButton; import com.google.android.gms.common.api.ApiException; import com.google.android.gms.tasks.Task; </pre>	

```

import com.google.android.material.textfield.TextInputLayout;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class LoginActivity extends AppCompatActivity {
    //google 登入
    GoogleSignInClient mGoogleSignInClient;
    private static int RC_SIGN_IN=100;
    public static ArrayList<Employee> employeeArrayList = new ArrayList<>();
    EditText email,password;
    //    TextInputLayout ;
    Button signup;
    Button login;
    TextView forgotPass;
    Toolbar toolbar;
    String str_email, str_password;
    String url = "https://louis32132118.000webhostapp.com/login.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle(" 登入");

        email = findViewById(R.id.txtvalue);
        password = findViewById(R.id.txtPassword);
        signup = findViewById(R.id.regName);
        login = findViewById(R.id.login);
    }
}

```

```

forgotPass = findViewById(R.id.viewForgot);
forgotPass.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        setForgotPass();
    }
});

//google 登入
GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestEmail()
    .build();

mGoogleSignInClient = GoogleSignIn.getClient(this, gso);
GoogleSignInAccount account =
GoogleSignIn.getLastSignedInAccount(this);
// Set the dimensions of the sign-in button.
SignInButton signInButton = findViewById(R.id.sign_in_button);
signInButton.setSize(SignInButton.SIZE_STANDARD);

signInButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        signIn();
    }
});

}

private void signIn() {
    Intent signInIntent = mGoogleSignInClient.getSignInIntent();
    startActivityForResult(signInIntent, RC_SIGN_IN);
}

@Override

```

```

public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    // Result returned from launching the Intent from
    GoogleSignInClient.getSignInIntent(...);
    if (requestCode == RC_SIGN_IN) {
        // The Task returned from this call is always completed, no need to
attach
        // a listener.
        Task<GoogleSignInAccount> task =
        GoogleSignIn.getSignedInAccountFromIntent(data);
        handleSignInResult(task);
    }
}

private void handleSignInResult(Task<GoogleSignInAccount> completedTask) {
    try {
        GoogleSignInAccount account =
        completedTask.getResult(ApiException.class);

        GoogleSignInAccount acct =
        GoogleSignIn.getLastSignedInAccount(this);
        if (acct != null) {
            String personName = acct.getDisplayName();
            String personGivenName = acct.getGivenName();
            String personFamilyName = acct.getFamilyName();
            String personEmail = acct.getEmail();
            String personId = acct.getId();
            Uri personPhoto = acct.getPhotoUrl();

            Toast.makeText(this,"會員帳號：
            "+personEmail,Toast.LENGTH_LONG).show();
        }
    }
}

```

```

        startActivity(new Intent(LoginActivity.this,MainActivity.class));
        // Signed in successfully, show authenticated UI.

    } catch (ApiException e) {
        // The ApiException status code indicates the detailed failure reason.
        // Please refer to the GoogleSignInStatusCodes class reference for more
information.

        Log.d("Message",e.toString());
    }
}

public void Login(View view) {

    if (!validatePassword() | !validateEmail()) {
        return;
    } else {

        final ProgressDialog progressDialog = new ProgressDialog(this);
        progressDialog.setMessage("Please Wait..");

        progressDialog.show();

        str_email = email.getText().toString().trim();
        str_password = password.getText().toString().trim();

        StringRequest request = new StringRequest(Request.Method.POST, url,
new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                progressDialog.dismiss();
            }
        }) {
            @Override
            protected void onErrorResponse(VolleyError error) {
                progressDialog.dismiss();
            }
        };
        Volley.newRequestQueue(this).add(request);
    }
}

```



```

        if (response.equalsIgnoreCase("logged in successfully")) {

            email.setText("");
            password.setText("");
            startActivity(new Intent(getApplicationContext(),
MainActivity.class));

            Toast.makeText(LoginActivity.this, "登入成功",
Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(LoginActivity.this, "登入失敗，請重新
嘗試", Toast.LENGTH_SHORT).show();
        }

    }

}, new Response.ErrorListener() {

    @Override
    public void onErrorResponse(VolleyError error) {
        progressDialog.dismiss();
        Toast.makeText(LoginActivity.this,
error.getMessage().toString(), Toast.LENGTH_SHORT).show();
    }
}

) {

    @Override
    protected Map<String, String> getParams() throws
AuthFailureError {
        Map<String, String> params = new HashMap<String,
String>();

        params.put("email", str_email);
        params.put("password", str_password);
        return params;
    }
}

```

```

        }
    };

    RequestQueue requestQueue =
Volley.newRequestQueue(LoginActivity.this);
    requestQueue.add(request);

    }
}
public void moveToRegistration(View view) {
    startActivity(new Intent(LoginActivity.this,Register.class));
    finish();
}

public void setForgotPass() {
    startActivity(new Intent(LoginActivity.this,ForgotPassActivity.class));
    finish();
}

//密碼格式方法
private Boolean validatePassword() {
    String val = password.getText().toString();
    String passwordVal = "^" +
        //"(?=.*[0-9])" +           //at least 1 digit
        "(?=.*[a-z])" +           //at least 1 lower case letter
        //"(?=.*[A-Z])" +           //at least 1 upper case letter
        "(?=.*[a-zA-Z])" +         //any letter
//        "(?=.*[@#$%^&+=])" +       //at least 1 special character
        "(?=\S+$)" +              //no white spaces
        ".{6,}" +                  //at least 4 characters

```

```

        "$";

        if (val.isEmpty()) {
            password.setError("密碼不可空白");
            return false;
        } else if (!val.matches(passwordVal)) {
            password.setError("密碼長度必須包含英數且 6 碼以上");
            return false;
        } else {
            password.setError(null);
//            password.setErrorEnabled(false);
            return true;
        }
    }

    //電子郵件格式方法
    private Boolean validateEmail() {
        String val = email.getText().toString();
        String emailPattern = "^[a-zA-Z0-9_\\.]+@([a-zA-Z0-9_\\.]+\\.[a-zA-Z0-9_\\.]+)+$";

        if (val.isEmpty()) {
            email.setError("電子郵件不可空白");
            return false;
        } else if (!val.matches(emailPattern)) {
            email.setError("電子郵件格式錯誤");
            return false;
        } else {
            email.setError(null);
//            email.setErrorEnabled(false);
            return true;
        }
    }
}

```

▼表 9-2-3 Fragment-recipe.java

程式名稱	Fragment-recipe.java
目的	食譜、建立清單功能
<pre> package com.example.home; import android.app.ProgressDialog; import android.content.Intent; import android.os.Bundle; import androidx.annotation.NonNull; import androidx.appcompat.app.AppCompatActivity; import androidx.appcompat.widget.Toolbar; import androidx.fragment.app.Fragment; import androidx.fragment.app.FragmentTransaction; import androidx.recyclerview.widget.LinearLayoutManager; import androidx.recyclerview.widget.RecyclerView; import android.view.LayoutInflater; import android.view.Menu; import android.view.MenuInflater; import android.view.MenuItem; import android.view.View; import android.view.ViewGroup; import android.widget.Toast; import com.android.volley.DefaultRetryPolicy; import com.android.volley.Request; import com.android.volley.RequestQueue; import com.android.volley.Response; import com.android.volley.RetryPolicy; import com.android.volley.VolleyError; import com.android.volley.toolbox.JsonObjectRequest; import com.android.volley.toolbox.Volley; </pre>	

```

import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.miguelcatalan.materialsearchview.MaterialSearchView;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class fragment_recipe extends Fragment {

    private RecyclerView recyclerView;
    private Toolbar toolbar;
    private getRecipeAdapter myAdapter;
    private FloatingActionButton floatingActionButton;
    private MaterialSearchView searchView;
    private List<Recipe> arrayList;
    private ProgressDialog progressDialog;
    FragmentTransaction transaction;
    Fragment fragment;
    Boolean like_true=false;

    public fragment_recipe() {

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment

```

```

View view = inflater.inflate(R.layout.fragment_recipe, container, false);

toolbar=view.findViewById(R.id.toolbar);
((AppCompatActivity)getActivity()).setSupportActionBar(toolbar);
searchView=view.findViewById(R.id.serach_view);
((AppCompatActivity)getActivity()).getSupportActionBar().setTitle("食譜");

progressDialog=new ProgressDialog(getActivity());
progressDialog.setMessage("Loading...");
progressDialog.setCanceledOnTouchOutside(false);
progressDialog.show();

arrayList=new ArrayList<>();
recyclerView=view.findViewById(R.id.recyclerView);
recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));
recyclerView.setHasFixedSize(true);

floatingActionButton=view.findViewById(R.id.FAB_creat);
floatingActionButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(getActivity(), CreateActivity.class));
    }
});
getData();

return view;
}

private void getData() {
    String url="https://louis32132118.000webhostapp.com/getRecipe.php";

```

```

JsonObjectRequest jsonObjectRequest=new
JsonObjectRequest(Request.Method.GET, url, null, new
Response.Listener<JSONObject>() {
    @Override
    public void onResponse(JSONObject response) {
        try {
            JSONArray jsonArray = response.getJSONArray("data");
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                Recipe recipe = new Recipe(
                    jsonObject.getString("id"),
                    jsonObject.getString("recipename"),
                    jsonObject.getString("username"),
                    jsonObject.getString("time"),
                    jsonObject.getString("category"),
                    jsonObject.getString("cooking"),
                    jsonObject.getString("cookingid"),
                    jsonObject.getString("inter"),
                    jsonObject.getString("intro"),
                    jsonObject.getString("likes"),
                    jsonObject.getString("people"),
                    jsonObject.getString("cookingtime"),
                    jsonObject.getString("picture")

                );
                arrayList.add(recipe);
            }
            myAdapter=new getRecipeAdapter(getActivity(),arrayList);
            myAdapter.setHasStableIds(true);
            recyclerView.setAdapter(myAdapter);
            progressDialog.dismiss();
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

```

Toast.makeText(getActivity(),e.toString(),Toast.LENGTH_SHORT).show();
                progressDialog.dismiss();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

Toast.makeText(getActivity(),error.getMessage(),Toast.LENGTH_SHORT).show();
                progressDialog.dismiss();
            }
        });
    int socketTime=7000;
    RetryPolicy retryPolicy=new DefaultRetryPolicy(socketTime,
        DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
        DefaultRetryPolicy.DEFAULT_BACKOFF_MULT);
    jsonObjectRequest.setRetryPolicy(retryPolicy);
    RequestQueue requestQueue= Volley.newRequestQueue(getActivity());
    requestQueue.add(jsonObjectRequest);
}
}

```


▼表 9-2-3 searchRecipeAdapter.java

程式名稱	searchRecipeAdapter.java
目的	查詢食譜功能
<pre> package com.example.home; import android.content.Context; import android.content.Intent; import android.os.Bundle; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.Filter; import android.widget.ImageButton; import android.widget.ImageView; import android.widget.TextView; import androidx.annotation.NonNull; import androidx.cardview.widget.CardView; import androidx.recyclerview.widget.RecyclerView; import com.bumptech.glide.Glide; import java.util.ArrayList; import java.util.List; public class searchRecipeAdapter extends RecyclerView.Adapter<searchRecipeAdapter.MyViewHolder> { private List<Recipe> arrayList; Context context; private List<Recipe> fullArrayList; int counter=1; public searchRecipeAdapter(Context context, List<Recipe> arrayList){ </pre>	

```

        this.arrayList=arrayList;
        this.context=context;
        this.fullArrayList=arrayList;
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View
view=LayoutInflater.from(context).inflate(R.layout.search_list_item,parent,false);
        MyViewHolder myViewHolder=new MyViewHolder(view);
        myViewHolder.txtid.setText(String.valueOf(counter++));
        return myViewHolder;
    }

    @Override
    public void onBindViewHolder(@NonNull searchRecipeAdapter.MyViewHolder
holder, int position) {
        final String name=fullArrayList.get(position).getRecipeName();
        final String username=fullArrayList.get(position).getUsername();
        final String retime=fullArrayList.get(position).getTime();
        final String id=fullArrayList.get(position).getId();
        final String cotime=fullArrayList.get(position).getCookingTime();
        final String people=fullArrayList.get(position).getPeople();
        final String like=fullArrayList.get(position).getLike();

        holder.txtname.setText(name);
        holder.txtcon.setText(username);
        holder.txtmsg.setText(retime);
        holder.txtlike.setText(like);
        holder.txtcotime.setText("烹調時間："+cotime);
        holder.txtpeo.setText("份量："+people);
    }

```

```

        holder.imageView(fullArrayList.get(position).getPicture());

        holder.cardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
//
Toast.makeText(context,"Clicked"+name,Toast.LENGTH_SHORT).show();
                Intent intent =new Intent(context, detailActivity.class);
                Bundle bundle=new Bundle();
                bundle.putString("id",id);
                intent.putExtras(bundle);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                context.startActivity(intent);
            }
        });
    }

    @Override
    public int getItemCount() {
        return fullArrayList.size();
    }

    public Filter getFilter(){
        return new Filter(){
            @Override
            protected FilterResults performFiltering(CharSequence charSequence){
                String charString =charSequence.toString();
                if(charString.isEmpty()){
                    fullArrayList=arrayList;
                }else {
                    List<Recipe> filteredList= new ArrayList<>();
                    for(Recipe row: arrayList){

```

```

        if(row.getRecipename().toLowerCase().contains(charString.toLowerCase())){
            filteredList.add(row);
        }
    }
    fullArrayList=filteredList;
}
FilterResults fil=new FilterResults();
fil.values=fullArrayList;

return fil;
}

@Override
protected void publishResults(CharSequence constraint, FilterResults
filterResults) {
    fullArrayList=(ArrayList<Recipe>) filterResults.values;
    notifyDataSetChanged();
}

};
}

public class MyViewHolder extends RecyclerView.ViewHolder {
    TextView txtid,txtname,txtcon,txtmsg,txtpeo,txtcotime,txtlike;
    CardView cardView;
    ImageView imageView;
    ImageButton Button;
    View view;
    public MyViewHolder(@NonNull View itemview){
        super(itemview);
        view=itemview;
    }
}

```

```

        this.txtlike=itemview.findViewById(R.id.heartCount1);
        this.cardView=itemview.findViewById(R.id.card_view1);
        this.txtid=itemview.findViewById(R.id.txt_id1);
        this.txtname=itemview.findViewById(R.id.txt_name1);
        this.txtcon=itemview.findViewById(R.id.txt_con1);
        this.txtmsg=itemview.findViewById(R.id.txt_msg1);
        this.Button=itemview.findViewById(R.id.heart1);
        this.txtcotime=itemview.findViewById(R.id.txt_cotime);
        this.txtpeo=itemview.findViewById(R.id.txt_peo);

    }

    public void setImageView(String uri) {
        this.imageView=view.findViewById(R.id.image1);
        Glide.with(context).load(uri).into(imageView);
    }
}

```

第十章 測試模型

10-1 測試計畫

我們以使用者的角度測試實際操作各功能的各個功能是否正常操作，以下針對各功能測試：

會員功能：

1. 註冊會員：確認使用者註冊是否成功。
2. 登入會員：確認使用者登入是否成功。
3. Google 登入：確認使用者使用 Google 登入是否成功。
4. 使用者資料：確認使用者所有資料是否成功載入。

食譜功能：

5. 食譜首頁：確認食譜是否抓取成功。
6. 下方導覽列：確認是否能切換至其他功能頁面。
7. 新增食譜：確認是否能新增食譜。
8. 食譜封面照：確認抓取封面照片是否成功。
9. 撰寫名稱簡介：確認能否成功輸入食譜名稱及簡介。
10. 公開選項：確認公開選項能否選取。
11. 類別選項：確認是否可以選取類別選項。
12. 烹調方式選項：確認是否可以選取烹調方式選項。
13. 烹調時間選項：確認是否可以選取烹調時間選項。
14. 份量選項：確認是否可以選取份量選項。
15. 葷素選項：確認是否可以選取葷素選項。
16. 菜系選項：確認是否可以選取菜系選項。
17. 食譜清單：確認是否可以新增食譜清單。
18. 食譜步驟照片：確認步驟照片是否抓取成功。
19. 食譜步驟說明：確認步驟說明是否輸入成功。
20. 食譜關鍵字搜尋：確認關鍵字搜尋是否成功。
21. 食譜篩選搜尋：確認食譜篩選搜尋是否成功。
22. 食譜詳細：確認食譜詳細頁面顯示是否成功。
23. 收藏食譜：確認是否能新增食譜。
24. 食譜評分：確認食譜能否提交評分。
25. 留言板：確認是否可留言並顯示留言。

Chatbot 功能：

- 26. 使用者輸入訊息：確認使用者輸入訊息是否成功
- 27. chatbot 回覆訊息：確認 chatbot 回覆訊息是否成功

日記功能：

- 28. 查詢日記：確認使用者日記查詢是否成功。

10-2 測試個案與測試結果資料

▼表 10-2-1 註冊會員測試資料

測試編號	1
功能名稱	註冊會員
測試目的	確認使用者註冊是否成功
測試流程	輸入帳號、密碼並送出
預期結果	註冊成功
測試成果	正常

▼表 10-2-2 登入會員測試資料

測試編號	2
功能名稱	登入會員
測試目的	確認使用者登入是否成功
測試流程	輸入帳號、密碼並送出
預期結果	進入系統
測試成果	正常

▼表 10-2-3 登入會員測試資料

測試編號	3
功能名稱	Google 登入
測試目的	確認使用者使用 Google 登入是否成功
測試流程	使用 Google 第三方登入模組
預期結果	進入系統
測試成果	正常

▼表 10-2-4 個人密碼修改測試資料

測試編號	4
------	---

功能名稱	個人密碼修改
測試目的	確認是否可以成功修改個人密碼
測試流程	按下修改個人密碼按鈕後，再次登入
預期結果	可修改個人密碼，並可以登入成功
測試成果	正常

▼表 10-2-5 食譜首頁

測試編號	5
功能名稱	食譜首頁
測試目的	確認食譜是否抓取成功
測試流程	進入頁面有顯示出資料庫之食譜
預期結果	顯示成功
測試成果	正常

▼表 10-2-6 下方導覽列

測試編號	6
功能名稱	下方導覽列
測試目的	確認是否能切換至其他功能頁面
測試流程	點選下列導覽列之功能
預期結果	切換成功
測試成果	正常

▼表 10-2-7 新增食譜

測試編號	7
功能名稱	新增食譜
測試目的	確認是否能新增食譜
測試流程	資料輸入後能夠新增食譜
預期結果	新增成功
測試成果	正常

▼表 10-2-8 食譜封面照

測試編號	8
功能名稱	食譜封面照
測試目的	確認抓取封面照片是否成功
測試流程	可開啟相簿或相機，選取拍攝照片並返回 APP
預期結果	抓取成功
測試成果	正常

▼表 10-2-9 撰寫名稱簡介

測試編號	9
功能名稱	撰寫名稱簡介
測試目的	確認能否成功輸入食譜名稱及簡介
測試流程	輸入名稱、簡介並送出
預期結果	輸入成功
測試成果	正常

▼表 10-2-10 公開選項

測試編號	10
功能名稱	公開選項
測試目的	確認公開選項能否選取
測試流程	勾選或取消勾選公開
預期結果	輸入成功
測試成果	正常

▼表 10-2-11 類別選項

測試編號	11
功能名稱	類別選項
測試目的	確認是否可以選取類別選項
測試流程	選取類別並送出
預期結果	選取成功

測試成果	正常
------	----

▼表 10-2-12 烹調方式選項

測試編號	12
功能名稱	烹調方式選項
測試目的	確認是否可以選取烹調方式選項
測試流程	選取烹調方式並送出
預期結果	選取成功
測試成果	正常

▼表 10-2-13 烹調時間選項

測試編號	13
功能名稱	烹調時間選項
測試目的	確認是否可以選取烹調時間選項
測試流程	選取烹調時間並送出
預期結果	選取成功
測試成果	正常

▼表 10-2-14 份量選項

測試編號	14
功能名稱	份量選項
測試目的	確認是否可以選取份量選項
測試流程	選取份量並送出
預期結果	選取成功
測試成果	正常

▼表 10-2-15 葷素選項

測試編號	15
功能名稱	葷素選項
測試目的	確認是否可以選取葷素選項
測試流程	選取葷素並送出

預期結果	選取成功
測試成果	正常

▼表 10-2-16 菜系選項

測試編號	16
功能名稱	菜系選項
測試目的	確認是否可以選取菜系選項
測試流程	選取菜系並送出
預期結果	選取成功
測試成果	正常

▼表 10-2-17 食譜清單

測試編號	17
功能名稱	食譜清單
測試目的	確認是否可以新增食譜清單
測試流程	輸入清單並送出
預期結果	輸入成功
測試成果	正常

▼表 10-2-18 食譜步驟照片

測試編號	18
功能名稱	食譜步驟照片
測試目的	確認步驟照片是否抓取成功
測試流程	可開啟相簿或相機，選取拍攝照片並返回 APP
預期結果	抓取成功
測試成果	正常

▼表 10-2-19 食譜步驟說明

測試編號	19
功能名稱	食譜步驟說明

測試目的	確認步驟說明是否輸入成功
測試流程	輸入步驟說明並送出
預期結果	輸入成功
測試成果	正常

▼表 10-2-20 食譜關鍵字搜尋

測試編號	20
功能名稱	食譜關鍵字搜尋
測試目的	確認關鍵字搜尋是否成功
測試流程	輸入關鍵字並送出，查看搜尋結果
預期結果	搜尋成功
測試成果	正常

▼表 10-2-21 食譜篩選搜尋

測試編號	21
功能名稱	食譜篩選搜尋
測試目的	確認食譜篩選搜尋是否成功
測試流程	輸入食譜篩選並送出，查看搜尋結果
預期結果	搜尋成功
測試成果	正常

▼表 10-2-22 食譜詳細

測試編號	22
功能名稱	食譜詳細
測試目的	確認食譜詳細頁面顯示是否成功
測試流程	點選食譜查看頁面
預期結果	顯示成功
測試成果	正常

▼表 10-2-23 收藏食譜

測試編號	23
功能名稱	收藏食譜
測試目的	確認是否能新增食譜
測試流程	點選食譜收藏鈕
預期結果	收藏成功
測試成果	正常

▼表 10-2-24 食譜評分

測試編號	24
功能名稱	食譜評分
測試目的	確認食譜能否提交評分
測試流程	點選星星進行評分並送出
預期結果	評分成功
測試成果	正常

▼表 10-2-25 留言板

測試編號	25
功能名稱	留言板
測試目的	確認是否可留言並顯示留言
測試流程	輸入留言並送出，進行查看
預期結果	顯示成功
測試成果	正常

▼表 10-2-26 日記查詢測試

測試編號	26
功能名稱	查詢日記
測試目的	查詢日記是否成功
測試流程	按下日期並送出

預期結果	查詢成功
測試成果	正常

▼表 10-2-27 Chatbot 使用者輸入測試

測試編號	27
功能名稱	使用者輸入
測試目的	使用者輸入訊息是否成功
測試流程	輸入訊息並送出
預期結果	輸入成功
測試成果	正常

▼表 10-2-28 Chatbot 回覆測試

測試編號	28
功能名稱	Chatbot 回覆
測試目的	Chatbot 回覆訊息是否成功
測試流程	使用者輸入訊息，chatbot 回覆訊息
預期結果	回覆成功
測試成果	正常

第十一章 操作手冊

11-1 系統安裝



▼ 表 11-1-1 系統安裝說明書

第十二章 使用手冊

以下是功能操作介面圖及使用方法，能清楚了解本系統之使用方式。

12-1 註冊、登入與忘記密碼

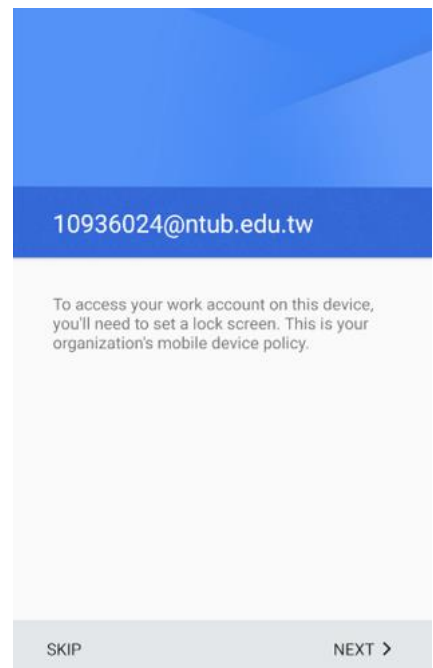
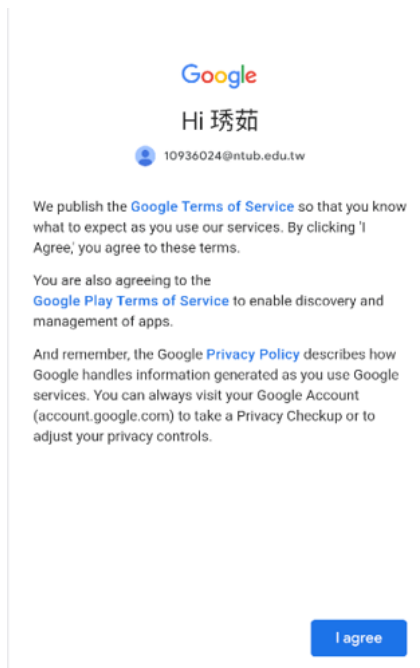
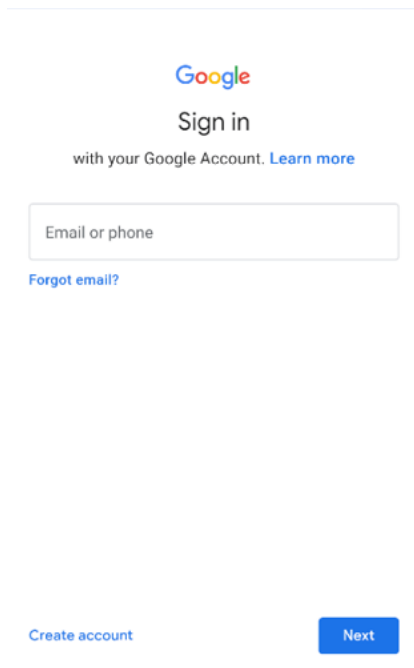
開啟 yo!煮廚後，必須先註冊帳號，註冊帳號後回到登入畫面進行登入，回到首頁另外也可以用 Google 帳號登入。如果忘記密碼，可輸入電子郵件確認身份後，即可變更密碼。



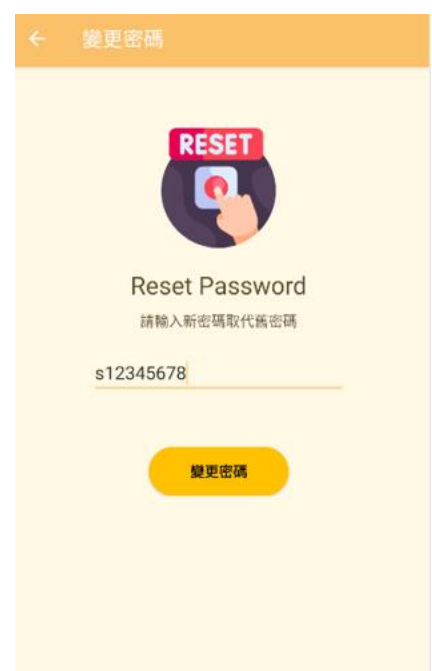
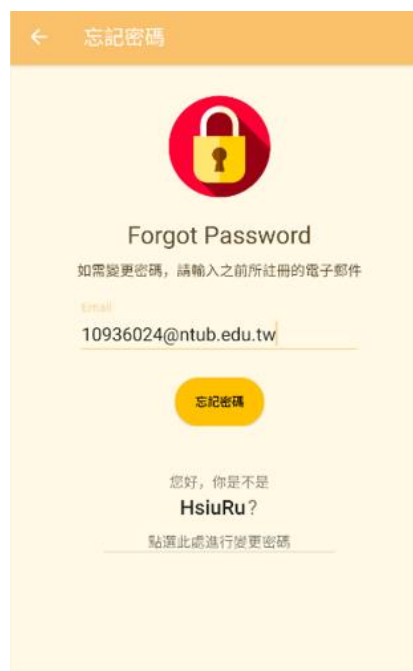
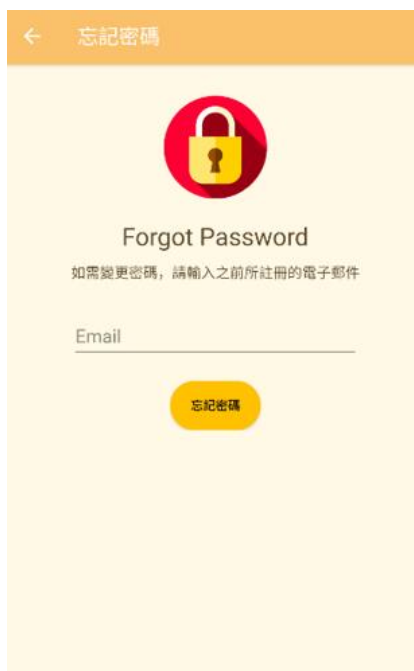
▲圖 12-1-1 登入



▲圖 12-1-2 註冊



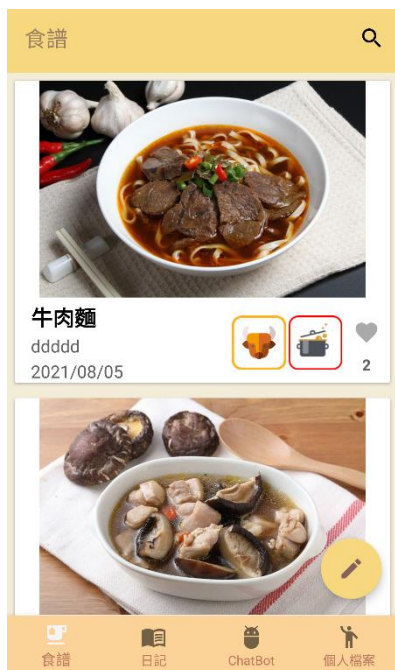
▲圖 12-1-3 Google 登入 ▲圖 12-1-4 Google 驗證 ▲圖 12-1-5 Google 驗證成功



▲圖 12-1-6 忘記密碼 ▲圖 12-1-7 身分驗證 ▲圖 12-1-8 變更密碼成功

12-2 搜尋食譜與新增食譜

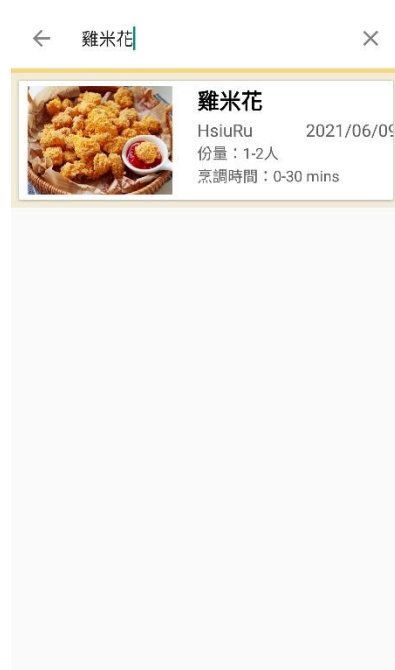
食譜關鍵字搜尋：在食譜首頁右上選單點選放大鏡進入搜尋頁面，再點選放大鏡即可輸入關鍵字進行搜尋。



▲圖 12-2-1 食譜首頁



▲圖 12-2-2 搜尋頁面

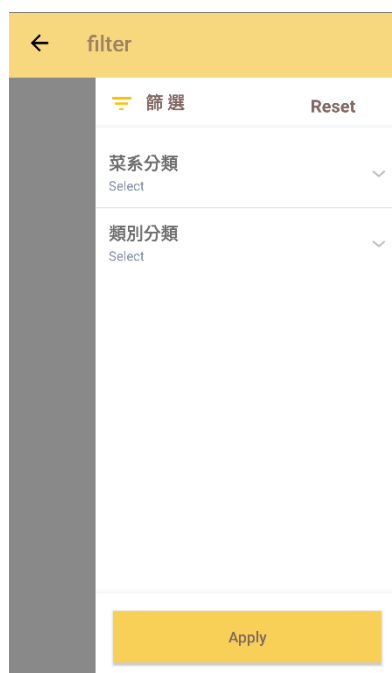


▲圖 12-2-3 關鍵字搜尋

食譜篩選搜尋：在食譜首頁右上選單點選放大鏡進入搜尋頁面，點選 ALL 即可進入篩選頁面，進行篩選提交篩選資料，即可顯示篩選食譜結果。



▲圖 12-2-4 搜尋頁面

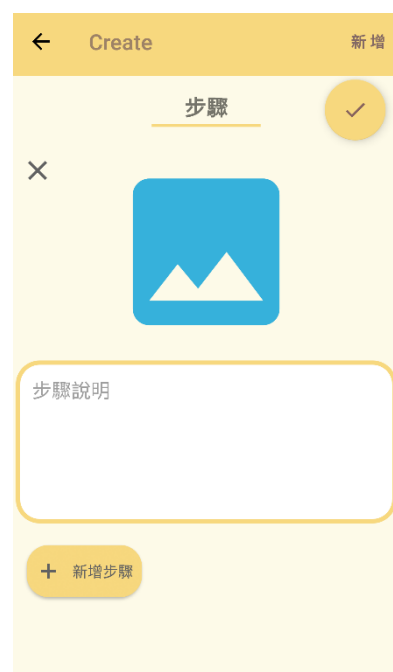
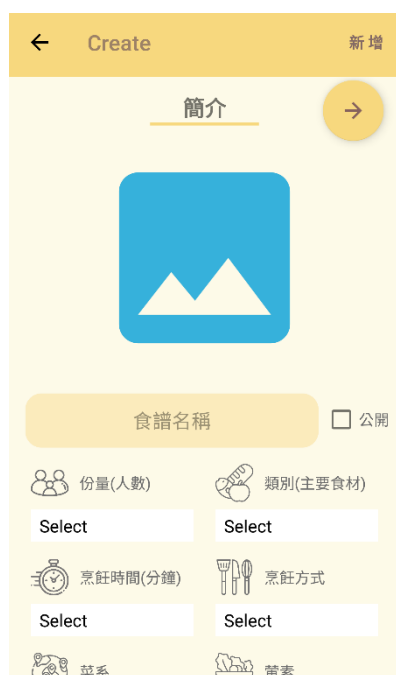


▲圖 12-2-5 篩選頁面



▲圖 12-2-6 篩選資料

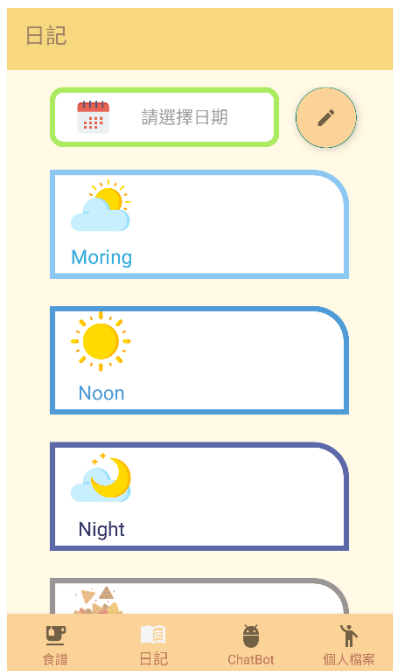
新增食譜：在食譜首頁右下角點選新增按鈕進入新增食譜頁面，填寫好便可點選右上角新增按鈕新增食譜。



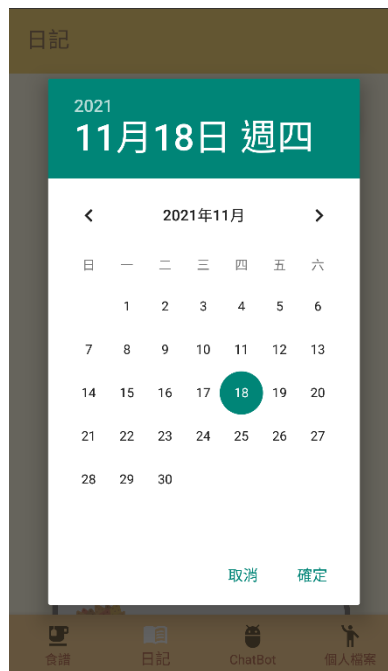
▲圖 12-2-7 新增食譜簡介 ▲圖 12-2-8 新增食譜清單 ▲圖 12-2-9 新增食譜步驟

12-3 日記

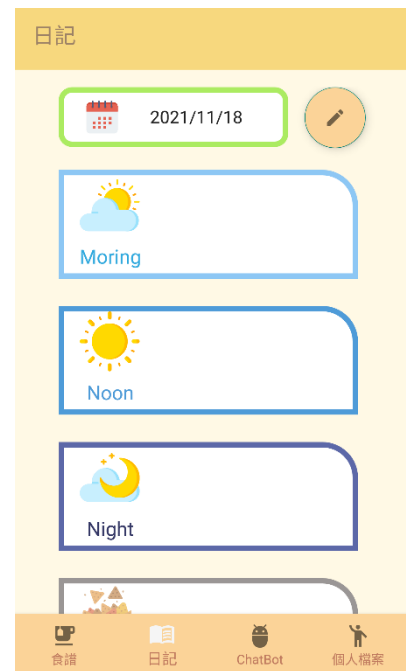
日記搜尋：在日記頁面中提供日期選擇，使用者選擇想查詢的日期，按下搜尋便能查看自己在該日期寫的日記。



▲圖 12-3-1 日記列表

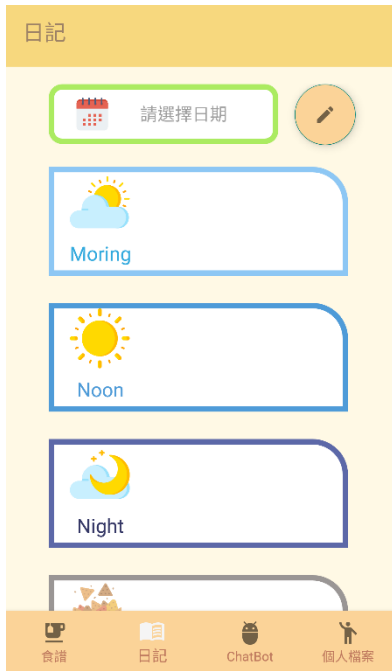


▲圖 12-3-2 選擇日期



▲圖 12-3-3 查看當天日記

新增日記：日記首頁右上角點選新增按鈕進入新增日記頁面，進入後可以依照自己的情況填入日記資料，按下儲存便成功新增日記。



▲圖 12-3-4 新增按鈕



▲圖 12-3-5 新增日記頁面



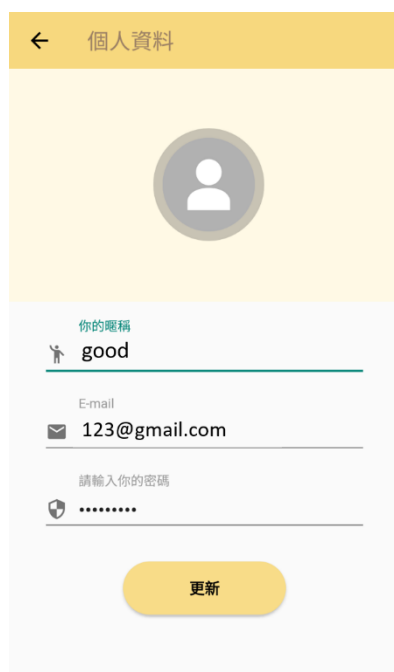
▲圖 12-3-6 輸入敘述

12-4 個人檔案

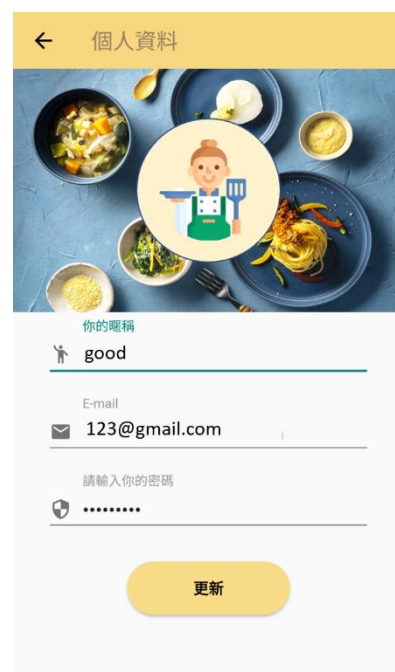
編輯個人檔案：按下個人資料按鈕可以進入到編輯頁面，在此可以變更自己的大頭照及封面照片。



▲圖 12-4-1 個人檔案首頁



▲圖 12-4-2 編輯資料

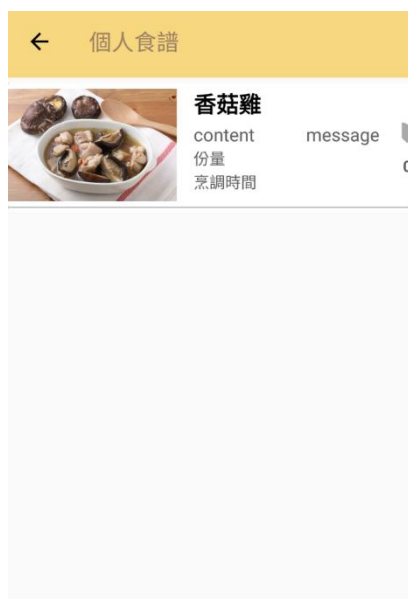


▲圖 12-4-3 變更照片

個人食譜：在個人檔案頁面有個食譜發佈量按鈕，點選後可察看所有發佈過的食譜。



▲圖 12-4-4 個人檔案



▲圖 12-4-5 歷屆食譜

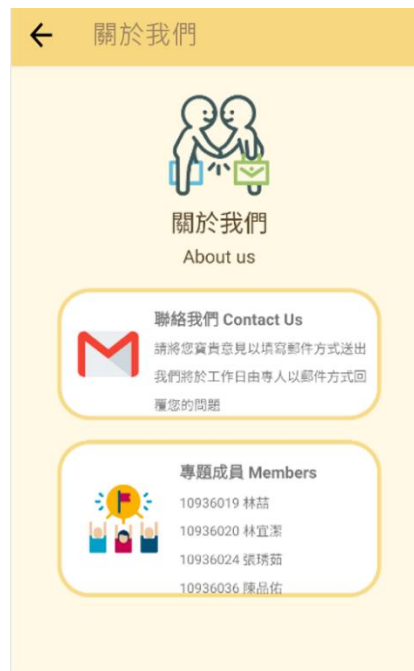


▲圖 12-4-6 食譜詳細

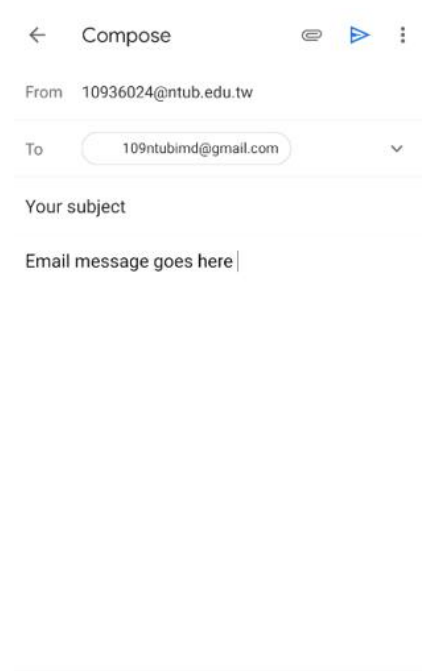
其他功能：可點選使用說明，查看功能；關於我們、聯絡我們，可發送電子郵件與我們聯繫。



▲圖 12-4-7 使用說明



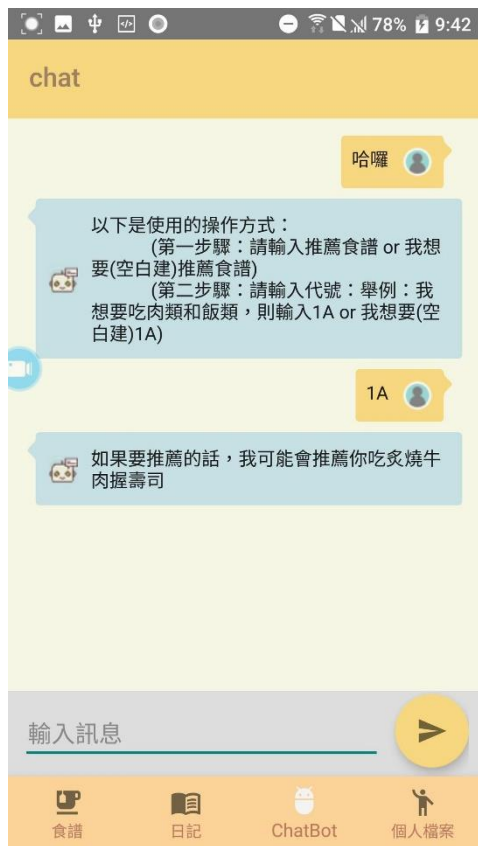
▲圖 12-4-8 關於我們



▲圖 12-4-9 發送電子郵件

12-4 chatbot

chatbot 功能：使用 chatbot 制式化的功能能夠幫助使用者作決定。



▲圖 12-4-10 chatbot

第十三章 感想

10936019 林喆

專題製作初期到中期時我常常被「我辦不到」、「我做不出來」的念頭給困擾，五月時因為突然疫情爆發，在時程上多少有受到影響，每天都待在電腦前上課及製作專題讓身心其實都有一定的壓力，因為全程都是待在家且線上實作，對自制力非常差的我是一個非常大的挑戰。

幸好和老師跟組員的固定開會討論，妥善將所有功能分門別類的安排時程逐步完成，在後期慢慢看到系統逐漸成形後，心裡其實放心不少，也比較踏實，而且在這段路中，我的程式能力不管是前端及後端都提升了不少，也體會到做中學的道理：網路/書本上教的是一回事，怎麼實際應用在自己需要的地方，更考驗平常對程式邏輯的靈敏度。

謝謝丁老師跟組員們包容不盡優秀的我，很高興能在專題中看到我們的成長！

10936020 林宜潔

相較於其他同學，我對程式可說是入門級新手，突然之間要我寫專題，我可是一臉茫然又不知所措。組員之中雖有五專也是就讀資管系的同學，卻並非北商資管五專生升上的，因此，我靠著指導老師的悉心協助以及組員間的協作摸索，並按照老師的時程計畫逐步地完成了專題製作。其中，當然遇到了許多困難，例如明明跟跟著資料做卻總是出現錯誤，版本的交替讓我本身就不熟悉程式語言與邏輯的人，只能花出更多時間來研究嘗試，努力了這半年，我很明顯地感覺到了自己的成長。剛開始組員之間雖有一些小摩擦，後來大家互相體諒、換位思考，每個人的努力成就了個專題，也許這個專題的主題不夠商業、不夠創新，但對於我來說，這份寶貴的經驗是我學生時期最濃厚的一抹色彩。

10936024 張琇茹

透過此次專題，我重新認識 Android studio 這套軟體，更加了解前端與後端的關係及串接方式，並且學習到學校中不曾聽過的概念及技術，如遇到不懂或不會時，會自己的主動上網查找資料且鑽研技術，再與組員相互討論，雖然過程中有爭辯、不愉快，但幸好都迎刃而解，讓我更了解到組員的重要性，我們不僅僅是合作夥伴，更是彼此的革命伴侶。

回想到一評時，我們因程式未與資料庫串聯，以致於成績位於吊車尾，但幸好專題導師丁慧瑩老師沒有放棄我們，且給我們許多良好的建議及方向，讓此次專題起死而生，謝謝老師的鼓勵，成為我們最大的動力，為此次專題畫上美好的句點。

10936036 陳品佑

經過這一年的專題製作，我認為我又成長了一大步。我是這個團隊的組長，首先做很多決定要想得更遠以及預想到未來可能會發生的事情，而這我在前半年並沒有做得很稱職，因為有太多事情是自己從未碰觸到的，因此花了很多的時間去研究沒有辦法立刻就下策略，像是 android studio 連上資料庫、抓取 api、資料庫的欄位設計還有我所負責的 chatbot 機器人。再來是還要掌控進度，而這些我優秀的隊友們，都很稱職地做好自己的工作。而這次的專題，從無到有，從爭吵到協同合作，我認為最重要的不是最後的成果，而是我們在共同製作專題的過程。

第十四章 參考文獻

14-1 參考資料

STACK OVERFLOW-send id then get registration data with volley and php

<https://stackoverflow.com/questions/47765717/send-id-then-get-registration-data-with-volley-and-php>

How to Implement Volley Library in Android Studio | VolleyLibrary | Android Coding

<https://www.youtube.com/watch?v=4dPqMrbP7xc&t=388s>

[教學] 字符串分割--java 中 String.split()用法 和 分割空白字符(正則)

<http://www.tshopping.com.tw/thread-247963-1-1.html>

Android 延时操作的三种方式

https://blog.csdn.net/qq_33833327/article/details/70946086

STACK OVERFLOW-Php to android getting the return value

<https://stackoverflow.com/questions/28670525/php-to-android-getting-the-return-value>

Github-droidcodes/DynamicViews

<https://github.com/droidcodes/DynamicViews>

ExpandableListView with CheckBox-RadioButton in Android

<https://medium.com/@makkenasrinivasarao1/expandablelistview-with-checkbox-radiobutton-in-android-af9ec9d81ddf>

Github- Dhaval2404/ImagePicker

<https://github.com/Dhaval2404/ImagePicker>

Android ImageButton 圖片按鈕

<https://www.flaticon.com/>

How to Integrate Google Sign In Your Android App using Android Studio

<https://www.youtube.com/watch?v=uPg1ydmnzpk&list=WL&index=6>

Validate Registration and login form in android studio - Form validation android 2020

<https://www.youtube.com/watch?v=o9Y7HDkopHg&list=WL&index=8>

Validate Email & Password with Regular Expression - Android Studio Tutorial

https://www.youtube.com/watch?v=cnD_7qFeZcY&list=WL&index=10

Android Layout 五大佈局

<https://fiend1120.pixnet.net/blog/post/191809863>

Android Studio 連線 MySQL 取得資料

<https://yayar.medium.com/android-studio->

[%E9%80%A3%E7%B7%9Amysql%E5%8F%96%E5%BE%97%E8%B3%87%E6%96%](https://yayar.medium.com/android-studio-%E9%80%A3%E7%B7%9Amysql%E5%8F%96%E5%BE%97%E8%B3%87%E6%96%)

[99-47bcebf60c88](#)

[Android Studio 菜鳥的學習分享]Fragment 應用(下)-BottomNavigationView 選單 + ViewPager

<https://ithelp.ithome.com.tw/articles/10250784>

Java：日期時間格式化輸出入處理：Date、Calendar、SimpleDateFormat

<https://www.ewdna.com/2009/01/javadatecalendardateformatsimpledatefor.html>

Android-TextView 多行本文滾動輕鬆實現

<https://sunylin.pixnet.net/blog/post/91068996>

select count(*) from table of mysql in php

<https://stackoverflow.com/questions/6907751/select-count-from-table-of-mysql-in-php>

sql php order by 用法

<https://pclevinblog.pixnet.net/blog/post/314563483-%5Bmysql%5D1-8->

[_%E6%8E%92%E5%BA%8F%E7%9A%84%E7%94%A8%E6%B3%95%28order-by%29](https://pclevinblog.pixnet.net/blog/post/314563483-%5Bmysql%5D1-8-%E6%8E%92%E5%BA%8F%E7%9A%84%E7%94%A8%E6%B3%95%28order-by%29)

MySQL 學習筆記 (CONSTRAINT 預設值篇)

<https://ithelp.ithome.com.tw/articles/10212142>

Android RecyclerView Using JSon Parsing PHP MySQL Data

<https://androidjson.com/recyclerview-json-listview-example/>

[Android Studio] 連線 MySQL 新增修改刪除與 ListView 自設畫面 (2019/1/18 更新)

<https://devildemon20100423.blogspot.com/2018/11/android-studio-mysqllistview.html>

TimePickerDialog - Android Studio Tutorial

https://www.youtube.com/watch?v=QMwaNN_aM3U&t=136s

Part 1 : How to build a ChatBot in Android (full Guide)

<https://www.youtube.com/watch?v=z5JILJO64sk&list=PLBYLctkttE9krFN1BZOvwHQrnSAs1qc7g>

Part 2 : How to build a ChatBot in Android (full Guide)

<https://www.youtube.com/watch?v=RSwfvHdIbpM&list=PLBYLctkttE9krFN1BZOvwHQrnSAs1qc7g&index=2>

Part 3 : How to build a ChatBot in Android (full Guide)

<https://www.youtube.com/watch?v=njIDMdsNiMg&list=PLBYLctkttE9krFN1BZOvwHQrnSAs1qc7g&index=3>

Part 4 : How to build a ChatBot in Android (full Guide)

<https://www.youtube.com/watch?v=GIogekyiFls&list=PLBYLctkttE9krFN1BZOvwHQrnSAs1qc7g&index=4>

Part 5 : How to build a ChatBot in Android (full Guide)

<https://www.youtube.com/watch?v=0E1rKBHMq8Y&list=PLBYLctkttE9krFN1BZOvwHQrnSAs1qc7g&index=5>

Creating a ChatBot App in Android | GeeksforGeeks

https://www.youtube.com/watch?v=7_Cc36c7EW0&t=2002s

Office Hours with Steve Worswick: Using the callapi tag - part 1

https://www.youtube.com/watch?v=nTqDY3xG_Ek&list=PLCzO7nFcslZEgvFFoSEVNzFh6-zqTBBXU&index=22

Office Hours with Steve Worswick: Using the callapi tag - part 2

<https://www.youtube.com/watch?v=EDmmPCcJpcl&list=PLCzO7nFcslZEgvFFoSEVNzFh6-zqTBBXU&index=23>

Office Hours with Steve Worswick: Using the callapi tag - part 3

https://www.youtube.com/watch?v=sp4IbY_zg8I&list=PLCzO7nFcslZEgvFFoSEVNzFh6-zqTBBXU&index=24

How to Create a Chatbot in Android with BrainShop API?

<https://www.geeksforgeeks.org/how-to-create-a-chatbot-in-android-with-brainshop-api/>

Android — Baking a simple ChatBot in 30 minutes (AIML)

<https://medium.com/@harivigneshjayapalan/android-baking-a-simple-chatbot-in-30-minutes-aiml-ff43c3269025>

附錄 初評之評審建議

評審建議事項	修正情形
1. 查詢若能提供複數食材輸入,再輸出建議食譜會較符合實際使用需求	已修正，已新增多種查詢功能
2. 建議與 SNS 連動	已修正，新增留言板功能
3. 系統登錄的部分建議提供以 SNS 帳號或 google 登錄	已修正，新增 google 登入
4. 動機有提到兩大族群，上班族的部分提及食安，系統功能無相關部分	已修正，此款 app 是讓上班族在家裡自己動手做料理，且 app 有選擇食譜以及食譜步驟去降低食安風險
5. 以國家分類菜色，在食譜數量不多的情況，會顯得不是那麼需要	已修正，因此透過此 app 去增加顧客的使用率，新增食譜的豐富性以及食譜量
6. 建議加上食譜製作困難度或耗時評等	已修正，食譜頁面已增加耗費時間