國立臺北商業大學

資 訊 管 理 系

113資訊系統專案設計



**系統手冊**

**組 別：第113407組**

**題 目：智慧侍茶師**

**指導老師：林俊杰老師**

**組 長：11046020 顏聿茗**

**組 員：11046006 張家愷 11046035 徐若蓁**

**11046036 洪藝芸**

**中華民國113年10月16日**

目錄

[第1章 前言 7](#_Toc184138784)

[第2章 營運計畫 10](#_Toc184138785)

[2-1可行性分析 10](#_Toc184138786)

[2-2商業模式－Business model 11](#_Toc184138787)

[2-3市場分析－STP 13](#_Toc184138788)

[2-4五力分析 13](#_Toc184138789)

[第3章 系統規格 14](#_Toc184138790)

[3-1 系統架構 14](#_Toc184138791)

[3-2 系統軟、硬體需求與技術平台 14](#_Toc184138792)

[3-3 使用標準與工具 15](#_Toc184138793)

[第4章 專案時程與組織分工 16](#_Toc184138794)

[4-1 專案時程:甘特圖 16](#_Toc184138795)

[4-2 專案組織與分工 17](#_Toc184138796)

[第5章　設計模型 22](#_Toc184138797)

[5-1 使用者需求 22](#_Toc184138798)

[5-2使用個案圖 23](#_Toc184138799)

[5-3使用個案描述 24](#_Toc184138800)

[5-4 分析類別圖 25](#_Toc184138801)

[第6章　設計模型 26](#_Toc184138802)

[6-1 循序圖 26](#_Toc184138803)

[6-2 設計類別圖 31](#_Toc184138804)

[第7章　實作模型 32](#_Toc184138805)

[7-1 佈署圖 32](#_Toc184138806)

[7-2 套件圖 32](#_Toc184138807)

[7-3 元件圖 33](#_Toc184138808)

[7-4 狀態機、時序圖 35](#_Toc184138809)

[第8章　資料庫設計 43](#_Toc184138810)

[8-1 資料庫關聯圖 43](#_Toc184138811)

[8-2 表格及其Meta data 44](#_Toc184138812)

[第9章　程式 49](#_Toc184138813)

[9-1 元件清單及其規格描述 49](#_Toc184138814)

[9-2 其他附屬之各種元件 52](#_Toc184138815)

[第10章　測試模型 76](#_Toc184138816)

[10-1 測試計畫 76](#_Toc184138817)

[10-2 測試個案與測試結果資料 77](#_Toc184138818)

[第11章　操作手冊 81](#_Toc184138819)

[11-1 系統元件 81](#_Toc184138820)

[11-2 系統下載 81](#_Toc184138821)

[第12章　使用手冊 82](#_Toc184138822)

[12-1 顧客操作 82](#_Toc184138823)

[12-2 員工操作 85](#_Toc184138824)

[12-3管理員操作 86](#_Toc184138825)

[第13章　感想 88](#_Toc184138826)

[第14章　參考資料 90](#_Toc184138827)

[附錄 91](#_Toc184138828)

圖目錄

[圖1-2-1 106年~110年我國飲料業之營利事業家數 8](#_Toc179658171)

[圖3-1-1 系統架構 14](#_Toc179658138)

[圖4-2-1 GitHub文件commits紀錄 19](#_Toc179658139)

[圖4-2-2 GitHub前端commits紀錄 20](#_Toc179658140)

[圖4-2-3 GitHub 後端commits紀錄 21](#_Toc179658141)

[圖5-2-1 使用個案圖 23](#_Toc179658142)

[圖5-4-1 分析類別圖 25](#_Toc179658143)

[圖6-1-1 循序圖－使用者註冊 26](#_Toc179658144)

[圖6-1-2 循序圖-使用者登入 26](#_Toc179658145)

[圖6-1-3循序圖-使用者填寫表單(登入版) 27](#_Toc179658146)

[圖6-1-4 循序圖-使用者填寫表單(未登入版) 27](#_Toc179658147)

[圖6-1-5 循序圖-店員POS機點餐 28](#_Toc179658148)

[圖6-1-6 循序圖-庫存資料 28](#_Toc179658149)

[圖6-1-7 循序圖-銷售資料 29](#_Toc179658150)

[圖6-1-8 循序圖-進貨資料 29](#_Toc179658151)

[圖6-1-9 循序圖-退貨資料 30](#_Toc179658152)

[圖6-2-1 類別圖 31](#_Toc179658153)

[圖7-1-1 佈署圖 32](#_Toc179658154)

[圖7-2-1 套件圖 32](#_Toc179658155)

[圖7-3-1 元件圖(顧客) 33](#_Toc179658156)

[圖7-3-2 元件圖(員工) 33](#_Toc179658157)

[圖7-3-3 元件圖(管理員) 34](#_Toc179658158)

[圖7-4-1 狀態機－使用者註冊 35](#_Toc179658159)

[圖7-4-2 狀態機－使用者登入 36](#_Toc179658160)

[圖7-4-3 狀態機－使用者填表單(登入版) 36](#_Toc179658161)

[圖7-4-4 狀態機－使用者填表單(未登入版) 37](#_Toc179658162)

[圖7-4-5 狀態機－店員POS機點餐 38](#_Toc179658163)

[圖7-4-6 狀態機－庫存資料 39](#_Toc179658164)

[圖7-4-7 狀態機－銷售資料 39](#_Toc179658165)

[圖7-4-8 狀態機－進貨資料 40](#_Toc179658166)

[圖7-4-9 狀態機－退貨資料 41](#_Toc179658167)

[圖7-4-10 時序圖 42](#_Toc179658168)

[圖8-1-1 關聯式資料庫ER圖 43](#_Toc179658172)

[圖8-1-2 關聯式資料庫ER圖 43](#_Toc179658173)

[圖12-1-1 註冊畫面 82](#_Toc184138677)

[圖12-1-2 登入畫面 82](#_Toc184138678)

[圖12-1-3 歡迎畫面 82](#_Toc184138679)

[圖12-1-4客製化問卷畫面 84](#_Toc184138680)

[圖12-1-5舌頭辨識畫面 84](#_Toc184138681)

[圖12-2-12員工首頁 85](#_Toc184138682)

[圖12-2-13 POS機畫面 85](#_Toc184138683)

[圖12-3-1登入首頁 86](#_Toc184138684)

[圖12-3-2管理員首頁 86](#_Toc184138685)

[圖12-3-3庫存資料畫面 86](#_Toc184138686)

[圖12-3-4銷售資料畫面 87](#_Toc184138687)

[圖12-3-5進/退貨資料畫面 87](#_Toc184138688)

表目錄

[表3-2-1系統軟硬體需求 14](#_Toc179658174)

[表3-3-1使用標準與工具表 15](#_Toc179383491)

[表4-1-1專案時程表 16](#_Toc179383492)

[表4-2-1 專業組織與分工表 17](#_Toc179383493)

[表4-2-2專題成果工作內容與貢獻度表 18](#_Toc179383494)

[表5-3-1 填寫問卷 24](#_Toc179383495)

[表5-3-2 查看歷史紀錄 24](#_Toc179383496)

[表5-3-3 新增進/退貨 24](#_Toc179383497)

[表5-3-4 更新庫存 25](#_Toc179383498)

[表8-2-1 資料表 44](#_Toc179383499)

[表8-2-2 資料表描述－01用戶資料 45](#_Toc179383500)

[表8-2-3 資料表描述－02顧客問題資料 46](#_Toc179383501)

[表8-2-4 資料表描述－03 進貨資料 47](#_Toc179383502)

[表8-2-5 資料表描述－04銷售資料 47](#_Toc179383503)

[表8-2-6 資料表描述－05對比銷售資料 48](#_Toc179383504)

[表8-2-7 資料表描述－06藥草庫存資料 48](#_Toc179383505)

[表9-1-1元件清單及其規格描述表 49](#_Toc179383480)

[表9-2-1部分程式碼- form.py 52](#_Toc179383457)

[表9-2-2 部分程式碼- utils.py 55](#_Toc179383458)

[表9-2-3 部分程式碼-views.py 61](#_Toc179383459)

[表9-2-4 部分程式碼-views.py 69](#_Toc179383460)

[表9-2-5部分程式碼-views.py 72](#_Toc179383461)

[表10-2-1功能描述-會員註冊 77](#_Toc179656095)

[表10-2-2 功能描述-會員登入 77](#_Toc179656096)

[表10-2-3 功能描述-登出 77](#_Toc179656097)

[表10-2-4 功能描述-填寫客製化表單(未登入版) 77](#_Toc179656098)

[表10-2-5 功能描述-填寫客製化表單(登入版) 78](#_Toc179656099)

[表10-2-6 功能描述-歷史紀錄 78](#_Toc179656100)

[表10-2-7 功能描述-影像辨識 78](#_Toc179656101)

[表10-2-8 功能描述- POS機點餐 79](#_Toc179656102)

[表10-2-9 功能描述- POS機接收訂單 79](#_Toc179656103)

[表10-2-10 功能描述- POS機收到庫存不足 79](#_Toc179656104)

[表10-2-11 功能描述-庫存資料表 79](#_Toc179656105)

[表10-2-12 功能描述-銷售資料表 80](#_Toc179656106)

[表10-2-13 功能描述-進貨/退貨資料表 80](#_Toc179656107)

[表11-1-1 系統元件表 81](#_Toc179656108)

[表12-1-1 顧客操作使用說明表 82](#_Toc184138689)

[表12-2-1 員工操作使用說明表 85](#_Toc184138690)

[表12-3-1 管理員操作使用說明表 86](#_Toc184138691)

**第1章 前言**

**1-1 背景介紹**

在現代社會中，人們對健康和養生方式的關注日益增加。許多人不僅尋求有效的養生方法，更希望這些方法是天然的。在這樣的需求下，草藥茶作為一種傳統且具有保健功能的飲品，受到了廣泛關注。傳統的草藥店以其提供的草藥飲品和個性化服務吸引了大量的顧客。

為了提供一種更現代化、更智能化的草藥店體驗，我們提出了一個創新方案，旨在讓客戶在感冒或不適時得到更適合的飲品建議，同時享受到更舒適、更個性化的服務。透過科技的應用，我們可以提高服務的效率和準確性，同時保留傳統草藥店的溫馨和人情味，為客戶提供一個全新的草藥養生體驗。

目前，天然療法的需求不斷增加，草藥店作為一種提供草藥茶以及個性化服務的場所，擁有獨特的吸引力。然而，傳統的紙卡溝通方式已經無法滿足現代客戶的需求。因此，我們提出了將草藥店服務智能化的方案，以提供更快速、更個性化的服務。

我們計劃改善草藥店的服務流程。客戶將通過問卷的方式，提供有關他們的症狀、健康需求以及過往的草藥使用歷史等信息。系統將根據這些信息提出相應的問題，以幫助客戶確定最適合他們需求的草藥茶。若客戶不滿意推薦，我們也會提供其他相似選擇，直到客戶滿意為止。

此外，我們計劃整合製作流程，與製作草藥飲品的員工之間的溝通更加流暢。這將有助於確保草藥的配方更加準確並滿足客人的需求。我們還將建立客戶和草藥種類的資料庫，前者用以搜集客戶的資料和反饋數據，了解客戶過往的紀錄和喜好種類，確保客戶獲得更好的體驗；後者可以在系統判斷完成後，給出相應的草藥和比例配方。

**1-2 動機**

目前，這間草藥店依賴紙卡溝通和侍茶師的經驗來推薦草藥茶和調配比例，這種方式存在一些效率低和人員不足的問題。隨著顧客對於快速和個性化服務需求的增加，需要一種更有效的解決方案。

結合現代科技，利用人工智慧和大數據分析，我們可以更準確地了解顧客的需求和症狀，根據個人資料和過往紀錄推薦最適合的草藥茶，並調配出相應的比例。這不僅能提高服務效率，還能解決侍茶師不足的問題，確保每位顧客都能享受到專業和個性化的服務。

根據財政部資料中心106年~110年我國飲料業之營利事業家數，台灣的飲料業發展迅速，飲料業的營利事業家數從106年的21346家增加至110年27414家，營業額接近千億新台幣。這顯示出市場對於飲料的需求量非常大，草藥茶作為飲料市場的一部分，潛力巨大。然而，競爭也非常激烈，消費者對於飲料的健康問題越來越重視。

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述結合市場調研，全球功能性飲料市場預計將在2020到2024年間以6%的年複合增長率增長。這意味著草藥茶這種健康飲品在未來有很大的市場增長空間。此外，技術的引入，如智慧侍茶師，可以顯著提升個性化服務和顧客滿意度。

▲圖1-2-1 106年~110年我國飲料業之營利事業家數

資料來源：財政部資料中心

**1-3 系統目的與目標**

我們系統的主要目的是：

1.有效與顧客溝通：通過線上問卷系統收集顧客的症狀和健康需求信息。

2.及時產出正確的草藥配方：根據顧客提供的信息，系統自動生成草藥茶配方，供員工使用。

3.掌控藥草庫存：利用資料庫管理草藥庫存，確保供應充足。

4.管理顧客歷史紀錄：保存顧客的草藥茶配方和反饋數據，方便下次光臨時提供更個性化的服務。

**1-4 預期成果**

我們預期通過以下方式達成系統目標：

1.網頁前台：顧客可以通過網頁填寫相關問卷，提供健康需求和症狀信息。

2.員工後台：員工通過系統獲取判斷結果，並按照系統提供的配方製作草藥茶飲。

3.數據管理：建立顧客和草藥種類的資料庫，系統自動記錄顧客的問卷歷史數據，提供精確的配方建議。

4.提升服務體驗：通過科技手段，提高服務效率和準確性，確保顧客獲得更好的草藥養生體驗。

**第2章 營運計畫**

**2-1可行性分析**

* 1. 市場可行性分析:

市場調查:調查市場內對於健康飲品的需求，以及喜好程度。

競爭分析:探討當地的手搖飲競爭狀況，以及其他類似的健康茶飲，去了解他們的銷售模式、產品特色和推廣狀況。

市場規模：健康和養生市場規模龐大，且持續增長。草藥產品在這一市場中占有重要地位，特別是在亞健康人群和中老年人中。

目標市場定位:調查目標市場的客源，像是追求健康的顧客，要去保留住這一個目標客群。

* 1. 技術可行性分析:

藥草的供應鏈:可以選擇穩定的供應商，像是跟種植這方面的藥草農民合作。

製作的設備和器具:確保設備運作正常，以及器具沒有損壞。

保持藥草的品質:嚴格控管藥草的品質，確保其藥草的新鮮度、純度等。

技術需求包括開發智能問卷系統，用於收集顧客的症狀和健康需求信息，並結合數據分析技術進行處理。此外，還需建立和管理顧客與藥草配方的資料庫管理系統，以確保數據的有效存儲和高效管理。

* 1. 財務可行性分析:

成本分析:去計算開店的租金、人力成本、進貨成本等等，去推估每杯藥草飲成本大概多少。

盈利預測:根據自身成本，以及市場競爭狀況來去推測銷售量預期和銷售額，以及預測在初期投入後的幾年內，店面的營業收入、成本支出和預期利潤。

風險評估:考慮可能發生的風險因素，例如市場競爭變動、原物料成本提高，去應對這些不確定的風險所制定的計畫。

**2-2商業模式－Business model**

1. 關鍵合作夥伴(Key Partners)

* 社群媒體
* 物流配送
* 藥草供應商和農民
* 本組組員

1. 關鍵活動(Key Activities)

* 藥草品質控管
* 行銷推廣
* 數位行銷策略
* 內容行銷：撰寫健康茶飲的相關文章，教導顧客如何根據自己的需求選擇合適的茶飲。
* 網路廣告：提高在搜尋引擎上的排名，同時投放關鍵字廣告，吸引目標客群。
* 實體推廣活動
* 店內促銷活動：推出期間限定的折扣或其他活動，吸引顧客進店消費。

1. 關鍵資源(Key Resources)

* 良好的機器設備
* 高品質的藥草
* 專業的服務人員

1. 價值主張(Value Propositions)

* 提供天然健康的飲品
* 根據顧客的健康狀況和口味訂製的茶飲
* 健康問卷與顧客檔案建立：提供線上的健康問卷調查，收集顧客的健康狀況、生活習慣和口味偏好，建立詳細的顧客檔案。
  + 客製化：根據顧客問卷回答，調配屬於自己的專屬茶飲。

1. 客戶關係(Customer Relationships)

* 客源分析
* 建立顧客茶飲紀錄

1. 通路(Channels)

* 廣告
* 媒體
* 可透過IG、FB等宣傳

1. 客戶群體(Customer Segments)

* 追求健康的人
* 想喝健康茶飲而不是手搖飲

1. 成本結構(Cost Structure)

* 租金和設備成本
* 管理成本
* 藥草原物料進貨成本
* 人力成本行銷
* 推廣成本

1. 收入來源(Revenue Streams)

* 藥草茶的銷售收入
* 其他產品銷售收入

**2-3市場分析－STP**

|  |  |  |
| --- | --- | --- |
| 市場區隔  Segmentation | 目標市場  Targeting | 市場定位  Positioning |
| 人口區隔:顧客年齡、性別  行為區隔:健康意識、品牌忠誠度、購買行為  心理區隔:追求健康、重視養身的人 | 對健康意識高、追求天然食品的消費者。進行重點銷售 | 獨特的藥草茶飲店、客製化的飲品、舒適的用餐環境、良好的品牌形象 |

**2-4五力分析**

1. 消費者的議價能力 Power of customers:

對於追求健康的消費者來說，議價能力不會高，藥草茶飲店目前沒有太多類似的店家，所以大部分可能會選擇此店家來作消費，但是對健康需求不高的消費者則議價能力就相對提高，市場上會有更多的替代店家可以選擇。

1. 供應商的議價能力 Power of suppliers:

種植藥草的供應商不多，代表供應商議價就會提高，建立跟供應商長期合作關係，或者取得更優惠的價格以及其他折扣，就會降低供應商議價能力的影響。

1. 潛在進入者的威脅 Potential of new entrants into the industry:

高威脅，因為目前藥草茶飲沒有到飽和的狀態、進入這市場的門檻也較低，像是低起步成本以及容易取得的供應商資源，進入市場後也會讓現有的藥草茶飲店的銷售額以及利潤造成影響。

* 1. 替代品的威脅 Threat of substitute products:

手搖飲、果汁、水、能量飲料。

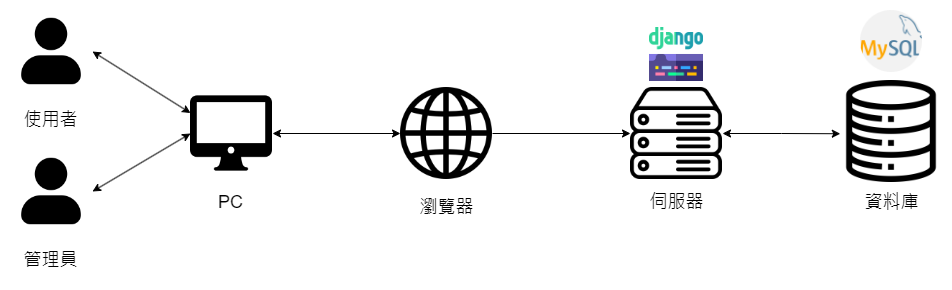
* 1. 現有競爭者的威脅 Competition in the industry:

競爭較高，競爭對手有各式的手搖飲店，手搖飲店還會不定時推出新品。

**第3章 系統規格**

3-1 系統架構

現在的手搖飲店會有各種推銷新飲品的活動，並沒有從顧客的健康思考，本組組員則跟藥草茶飲店合作，藉由傳統的紙卡問卷調查改成線上填寫問卷，計算出最適合每位顧客的茶飲，讓每位來消費的顧客都可以體驗到藥草類的手搖飲然而並不會對身體造成負擔。使用者會先在平板設備填寫問卷，問卷填寫結束後則會將結果回傳至員工介面上以及MySQL中，並經由此系統來去做進銷存系統。



▲圖3-1-1 系統架構

3-2 系統軟、硬體需求與技術平台

網頁應用可以在任何設備上運行，只要有瀏覽器，無論是桌面電腦、筆記本電腦、平板電腦還是手機，使用者都能夠訪問，以及網頁也可以通過響應式設計來調整介面，提供使用者更好的體驗，基於這幾種原因，我們本組決定以網頁開發此系統。

▼表3-2-1系統軟硬體需求

|  |  |
| --- | --- |
| 系統軟硬體需求-電腦 | |
| 軟體需求 | Web瀏覽器 |
| 網路需求 | Wifi/4G/5G網路 |

3-3 使用標準與工具

下表為本組的開發工具，各項工具的選用說明如下：

* Python：最易解讀的程式，也能夠去方便整理，Django也是採用Python語言來做運行。
* Html：只要有瀏覽器都可以去做訪問，作為本組的前端開發工具。
* Django：Django採用Python 語言編寫，框架組織良好，以其快速開發和整潔，作為本組的後端開發工具。
* Github：建立共同合作的軟體開發平台，結合Fork以及GitKraken工具，可方便管理本組以及各自的資料夾，能清楚追蹤與紀錄每位成員的進度與動態，兼具審視檔案的功能，提供團隊開發更多的便利性，作為本組管理專案進度的工具。

▼表3-3-1使用標準與工具表

|  |  |
| --- | --- |
| 系統開發環境 | |
| 作業系統 | Windows11、Ubuntu22.04 |
| 程式開發工具 | |
| 前端 | Html、CSS、JavaScript |
| 後端 | Python、MySQL、Django |
| 文件美工工具 | |
| 文件 | Microsoft Word |
| 簡報 | Microsoft PowerPoint |
| 圖樣 | Canva、Draw.io |
| 影片 | HeyGen、faceswapper |
| 專案管理平台 | |
| 專案管理 | Github、Fork、GitKraken |
| 檔案存放 | Github、Google Drive |
| 溝通工具 | Line、Discord |

**第4章 專案時程與組織分工**

4-1 專案時程:甘特圖

▼表4-1-1專案時程表

▅ 預期進度 / ▅ 實際進度

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 年 | 113年 | | | | | | | | |
| 月 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 技術學習 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 題目構想 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 後端框架編寫 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 資料庫建置 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 伺服器架設 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 前端開發 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| UI/UX設計 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 前後端串接 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 決策樹分析 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 語音辨識 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 影像辨識 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 系統測試 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 系統整合 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 文件製作 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 簡報製作 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

**4-2 專案組織與分工**

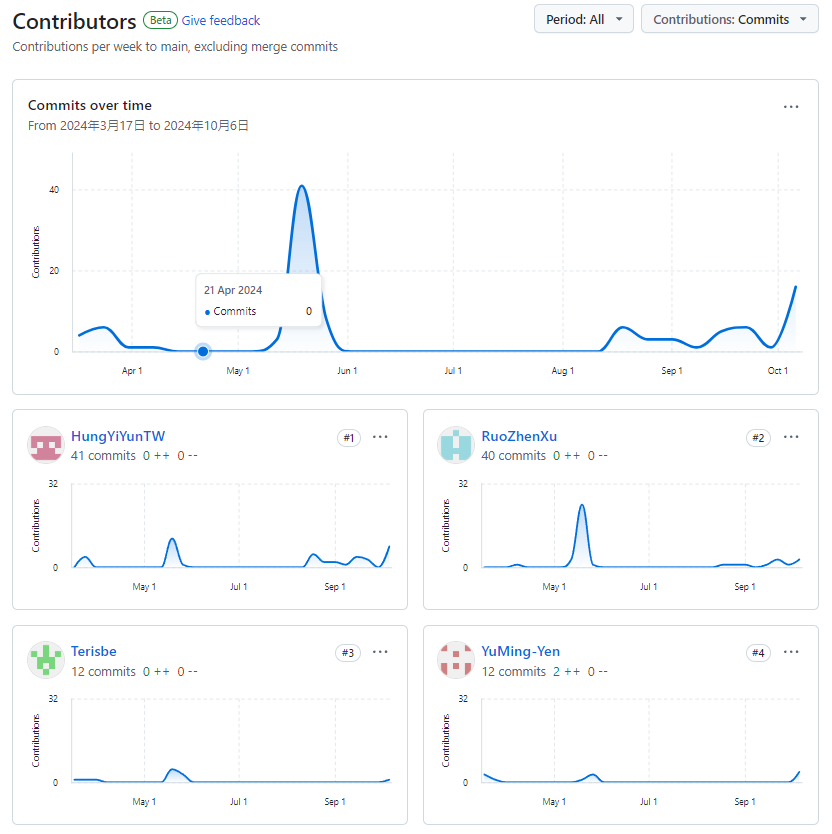
●主要負責人 〇次要負責人 (每一項只能有1位主要負責人，次要負責人最多2位)

▼表4-2-1 專業組織與分工表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 項目/組員 | | 11046020  顏聿茗 | 11046006  張家愷 | 11046035  徐若蓁 | 11046036  洪藝芸 |
| 後端開發 | 資料庫建置 | 〇 | ● |  |  |
| 資料庫Trigger使用 |  | ● |  |  |
| 伺服器架設 | ● | 〇 |  |  |
| Django 環境架設 | ● |  |  |  |
| 登入/註冊功能 |  | ● |  |  |
| 前後端串接 | ● | 〇 |  |  |
| 進貨資料表 | 〇 | ● |  |  |
| 庫存資料表 | ● | 〇 |  |  |
| 銷貨資料表 | ● | 〇 |  |  |
| 歷史紀錄 | 〇 | ● |  |  |
| 分類演算法 |  |  | ● |  |
| 語音辨識 |  |  | ● |  |
| 演算法CART |  | ● | 〇 |  |
| 舌頭影像辨識 | 〇 | ● |  |  |
| 前端開發 | 客製化表單 |  |  | 〇 | ● |
| POS介面 | 〇 |  |  | ● |
| AI虛擬主播影片 |  |  |  | ● |
| 響應式網頁 | 〇 |  |  | ● |
| 登入/註冊介面設計 | ● |  |  |  |
| 美術設計 | UI/ UX | 〇 |  |  | ● |
| Web介面設計 |  | 〇 |  | ● |
| 色彩設計 | ● |  |  |  |
| Logo設計 |  |  | ● |  |
| 文件撰寫 | 統整 | ● | 〇 |  | 〇 |
| 第1章 前言 |  |  | ● |  |
| 第2章 營運計畫 |  |  | ● |  |
| 第3章 系統規格 | 〇 | ● |  |  |
| 第4章 專題時程與組織分工 | ● |  |  |  |
| 第5章 需求模型 | 〇 | ● | 〇 |  |
| 第6章 設計模型 |  |  | 〇 | ● |
| 第7章 實作模型 |  | 〇 | 〇 | ● |
| 第8章 資料庫設計 |  | ● |  |  |
| 第9章 程式 | 〇 | ● |  |  |
| 第10章 測試模型 |  |  |  | ● |
| 第11章 操作手冊 |  |  |  | ● |
| 第12章 使用手冊 |  |  |  | ● |
| 報告 | 簡報製作 |  |  | ● |  |

▼表4-2-2專題成果工作內容與貢獻度表

|  |  |  |  |
| --- | --- | --- | --- |
| 序號 | 姓名 | 工作內容<各限100字以內> | 貢獻度 |
| 1 | 組長  顏聿茗 | 後端:Django環境架設、進貨資料表、庫存資料表、銷貨資料表、前後端資料串接、伺服器架設  前端:POS介面調整、登入/註冊介面製作、響應式網頁調整  文件撰寫、統整 | 25.5 % |
| 2 | 組員  張家愷 | 後端:進貨資料表調整(限制輸入類型)、登入&註冊功能製作、顧客歷史紀錄、前後端串接、舌頭辨識功能、決策樹演算法  資料庫：MySQL資料庫與設計&建置、Trigger建置  文件撰寫 | 29 % |
| 3 | 組員  徐若蓁 | 分類演算法:判斷及調整藥材克數  美編：logo設計、簡報製作  語音辨識、決策樹演算法  文件撰寫 | 20 % |
| 4 | 組員  洪藝芸 | 前端:設計客製化表單網站介面、設計POS機的網站介面、設計製作虛擬主播影片、Web介面設計: UI/ UX設計、響應式網頁、系統簡介撰寫  文件撰寫 | 25.5 % |
|  | | | 總計:100% |



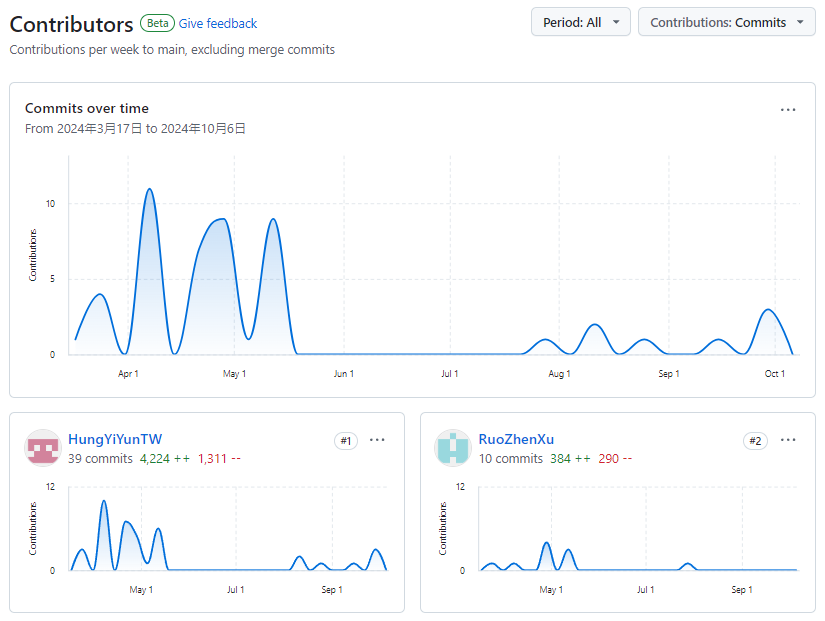
洪藝芸

徐若蓁洪藝芸

顏聿茗

張家愷

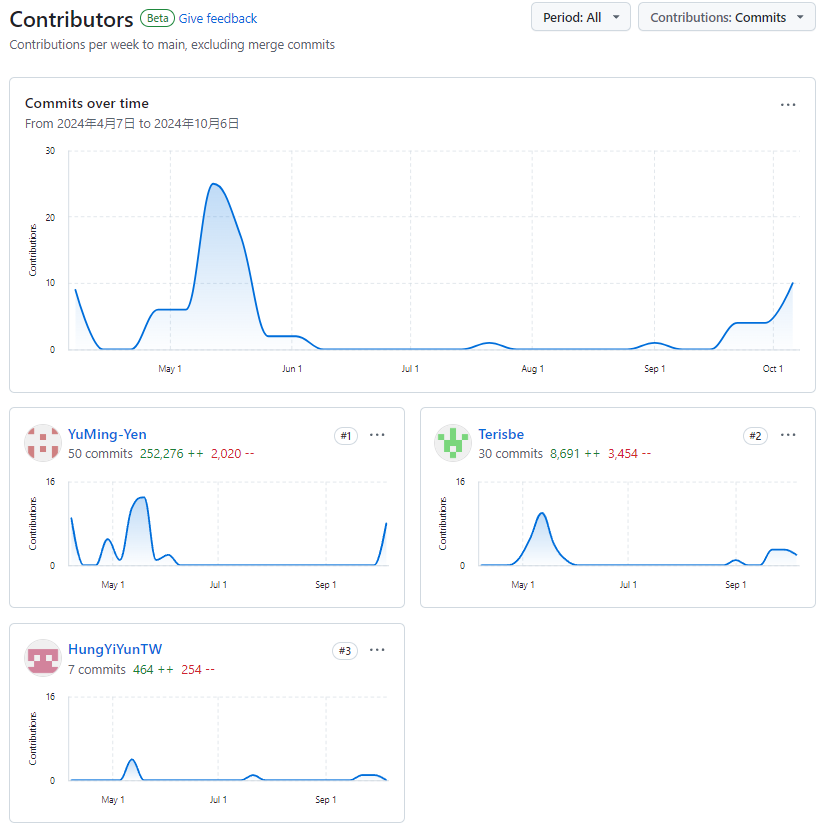
▲圖4-2-1 GitHub文件commits紀錄



徐若蓁

洪藝芸

▲圖4-2-2 GitHub前端commits紀錄



張家愷

洪藝芸

顏聿茗

▲圖4-2-3 GitHub 後端commits紀錄

**第5章　設計模型**

5-1 使用者需求

* 功能性需求：
* 使用者：

1. 用戶註冊和登入：驗證用戶名和生日的正確性。
2. 問卷填寫和提交：在前端頁面上填寫、提交問卷，並得到提交成功的確認。

* 系統：

1. 配方管理：根據問卷數據自動生成藥草配方。
2. 庫存管理：管理藥草庫存數據，並記錄每次更新的時間和數量。

* 非功能性需求：
* 性能：系統在2秒內響應用戶請求，例如登入、問卷提交和配方生成。
* 兼容性：兼容主流的網頁瀏覽器（例如Chrome、Firefox、Safari和Edge）。
* 權限管理：帳號有設定權限限制，只能進入相對應權限的頁面。若進入沒有權限的網址，會進入登入頁面。

**5-2使用個案圖**

****

▲圖5-2-1 使用個案圖

**5-3使用個案描述**

▼表5-3-1 填寫問卷

|  |  |
| --- | --- |
| 使用案例說明 | 填寫問卷 |
| 功能描述 | 顧客可以填寫一份調查問卷，且可以自由選擇要不要舌頭辨識，以便系統根據問卷生成藥草配方 |
| 參與者 | 使用者 |
| 前提 | 進到問卷頁面 |
| 成功條件 | 填寫所有問題症狀 |
| 失敗條件 | 未填寫所有問題症狀 |

▼表5-3-2 查看歷史紀錄

|  |  |
| --- | --- |
| 使用案例說明 | 查看歷史紀錄 |
| 功能描述 | 顧客可以查看過去生成的配方歷史記錄 |
| 參與者 | 使用者 |
| 前提 | 顧客已經成功登入並曾經填寫過表單 |
| 成功條件 | 進到登入頁面 |
| 失敗條件 | 尚未登入 |

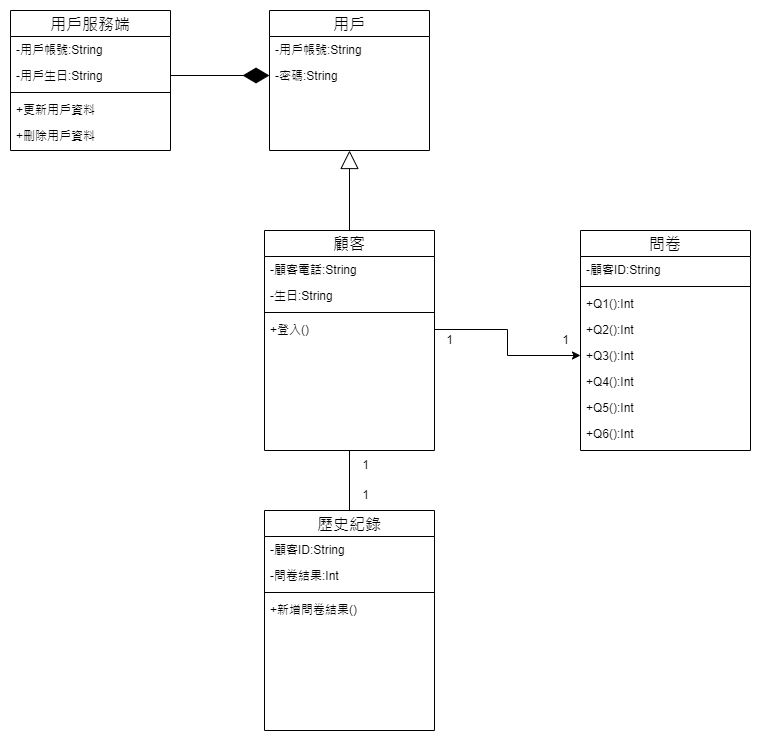
▼表5-3-3 新增進/退貨

|  |  |
| --- | --- |
| 使用案例說明 | 新增進/退貨 |
| 功能描述 | 輸入進/退貨資料 |
| 參與者 | 系統 |
| 前提 | 使用管理員權限登入 |
| 成功條件 | 進貨資料都要填寫完整 |
| 失敗條件 | 填寫的資料格式錯誤 |

▼表5-3-4 更新庫存

|  |  |
| --- | --- |
| 使用案例說明 | 更新庫存 |
| 功能描述 | 自動更新當前庫存狀況 |
| 參與者 | 系統 |
| 前提 | 資料庫中有這筆庫存 |
| 成功條件 | 新增進/退貨資料或新增銷貨資料 |
| 失敗條件 | 無 |

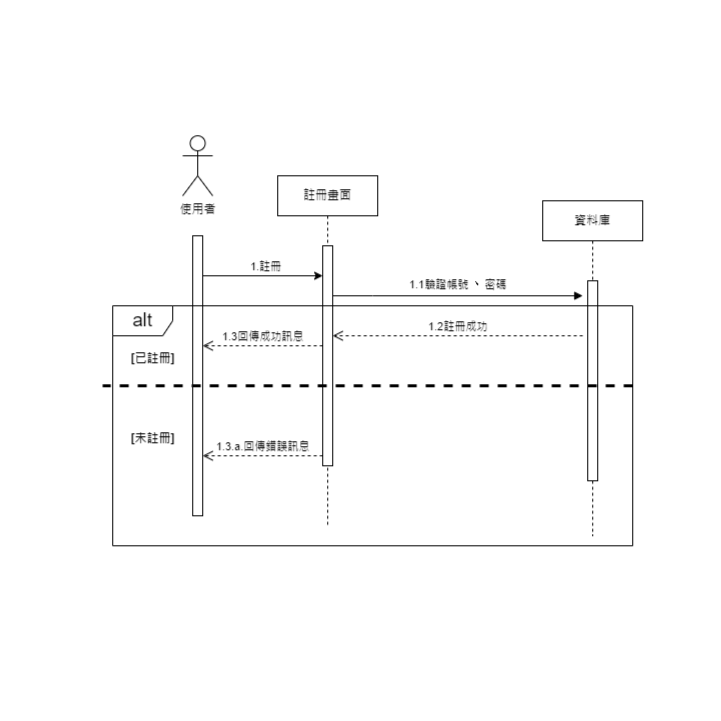
**5-4 分析類別圖**

****

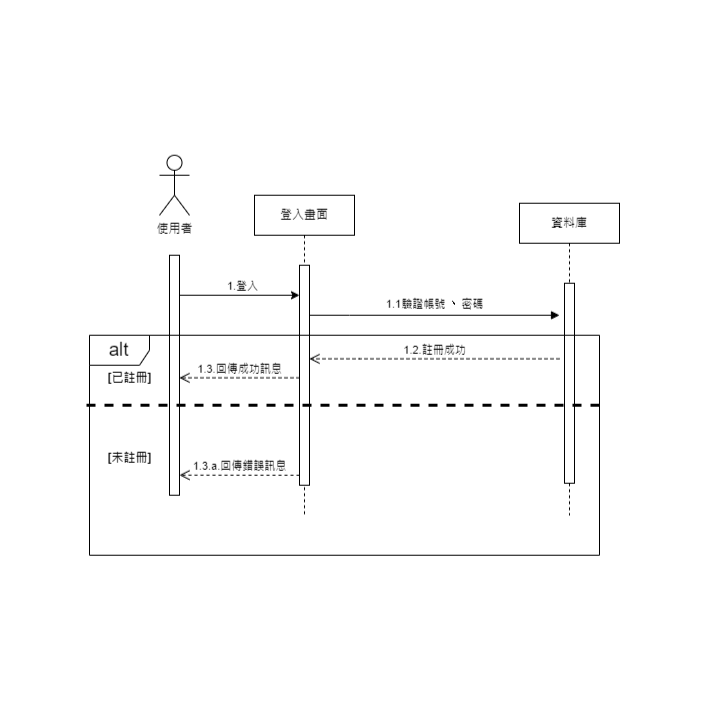
▲圖5-4-1 分析類別圖

**第6章　設計模型**

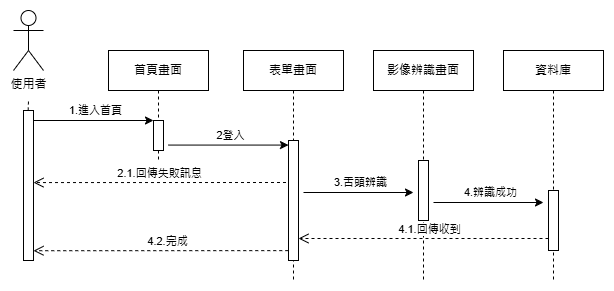
**6-1 循序圖**



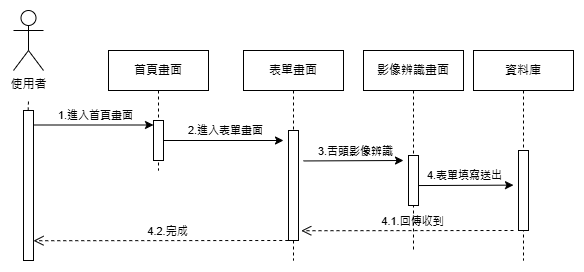
▲圖6-1-1 循序圖－使用者註冊



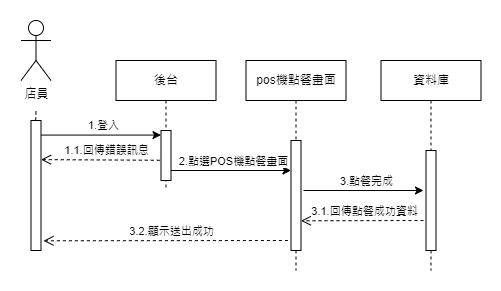
▲圖6-1-2 循序圖-使用者登入



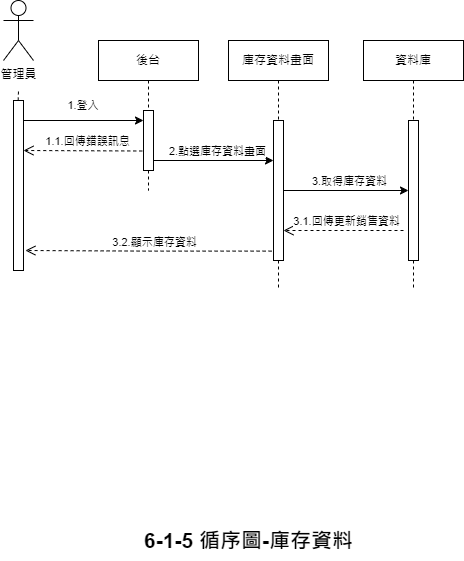
▲圖6-1-3循序圖-使用者填寫表單(登入版)



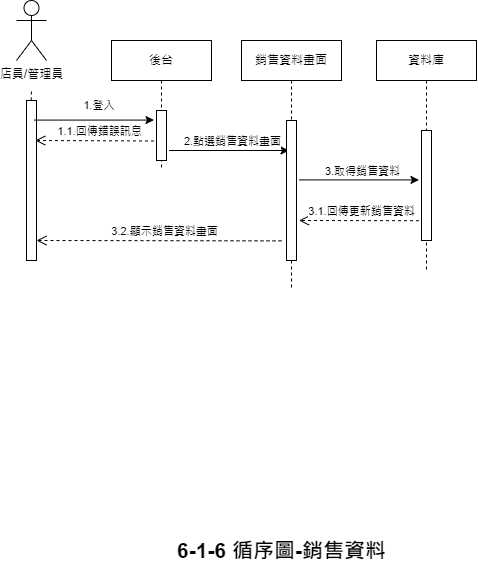
▲圖6-1-4 循序圖-使用者填寫表單(未登入版)



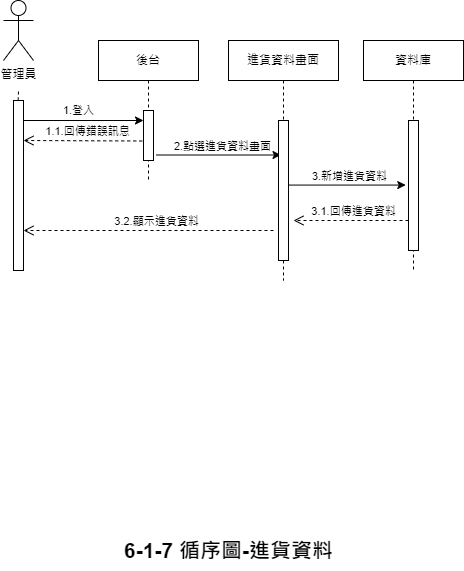
▲圖6-1-5 循序圖-店員POS機點餐

****

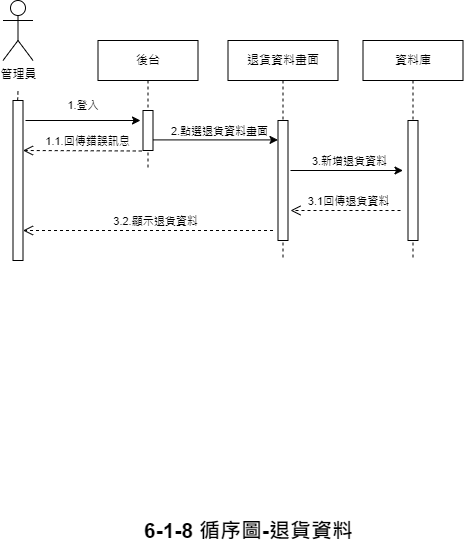
▲圖6-1-6 循序圖-庫存資料

****

▲圖6-1-7 循序圖-銷售資料

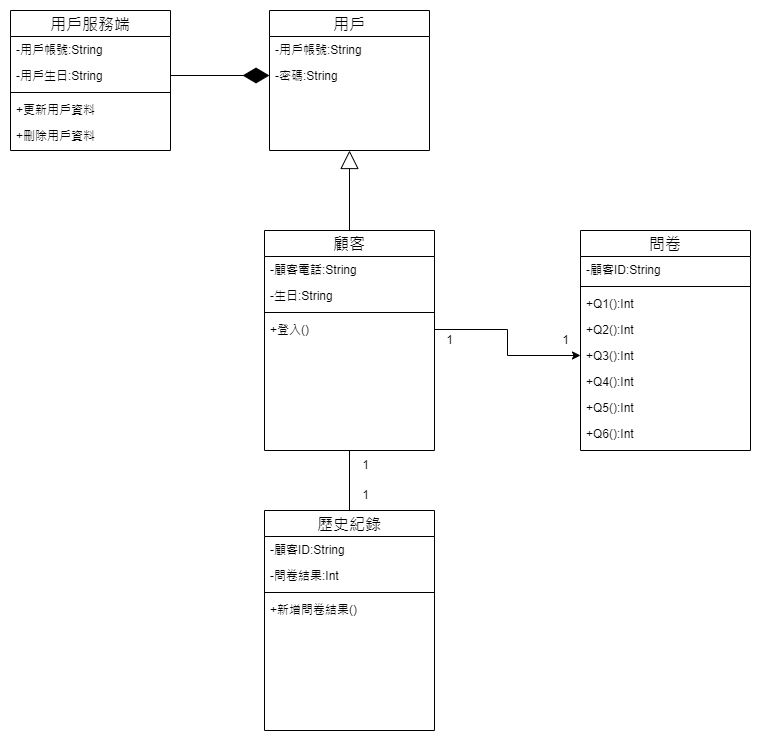
****

▲圖6-1-8 循序圖-進貨資料

****

▲圖6-1-9 循序圖-退貨資料

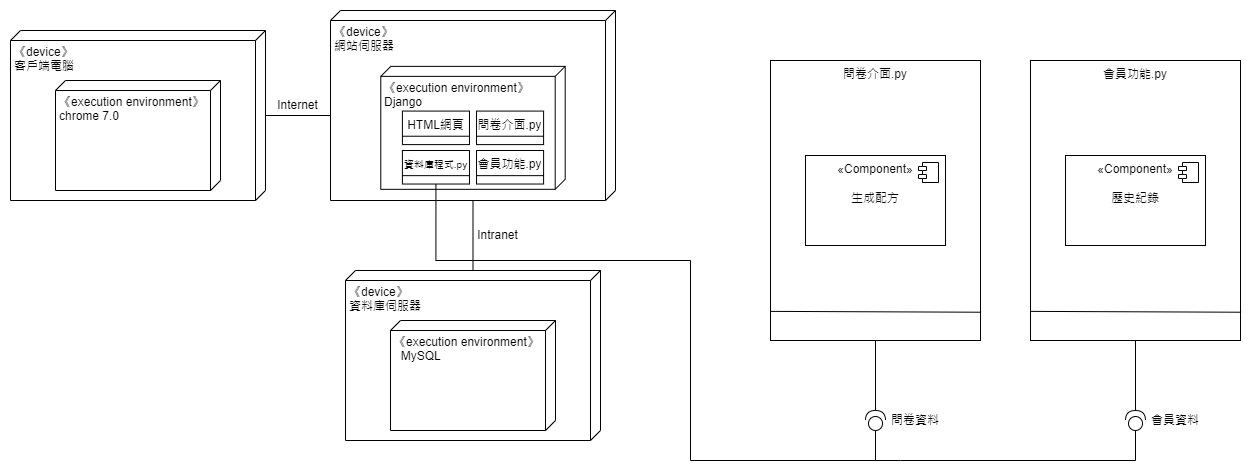
**6-2 設計類別圖**



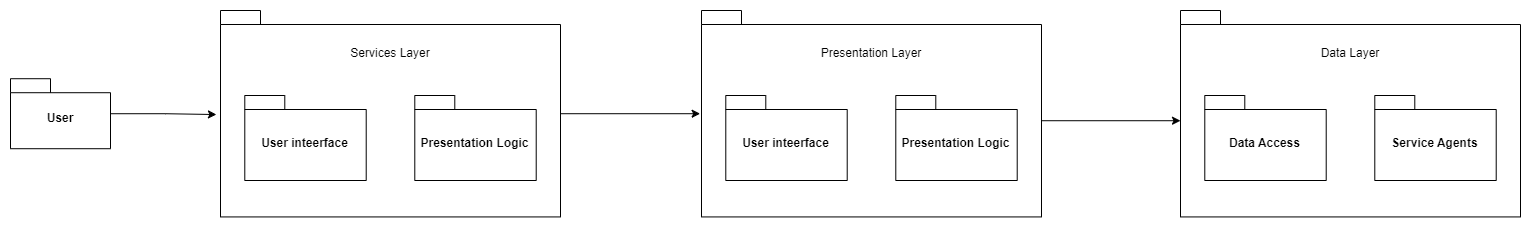
▲圖6-2-1 類別圖

**第7章　實作模型**

**7-1 佈署圖**

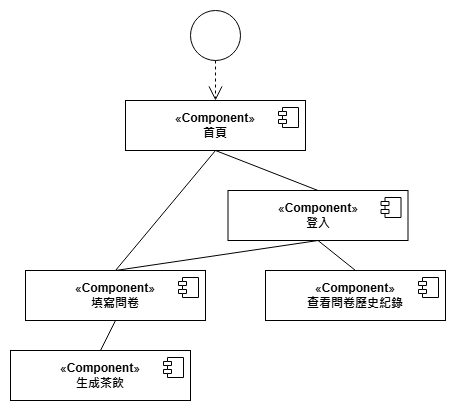
****

▲圖7-1-1 佈署圖

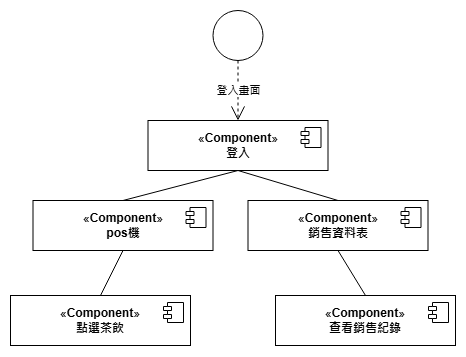
**7-2 套件圖**

▲圖7-2-1 套件圖

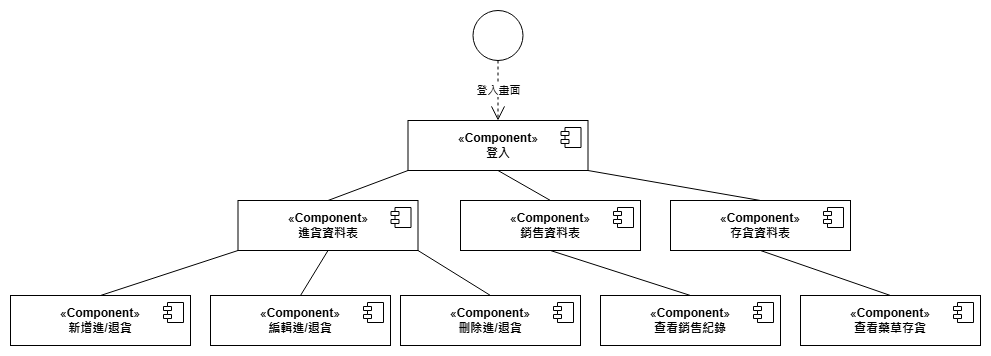
**7-3 元件圖**

****

▲圖7-3-1 元件圖(顧客)

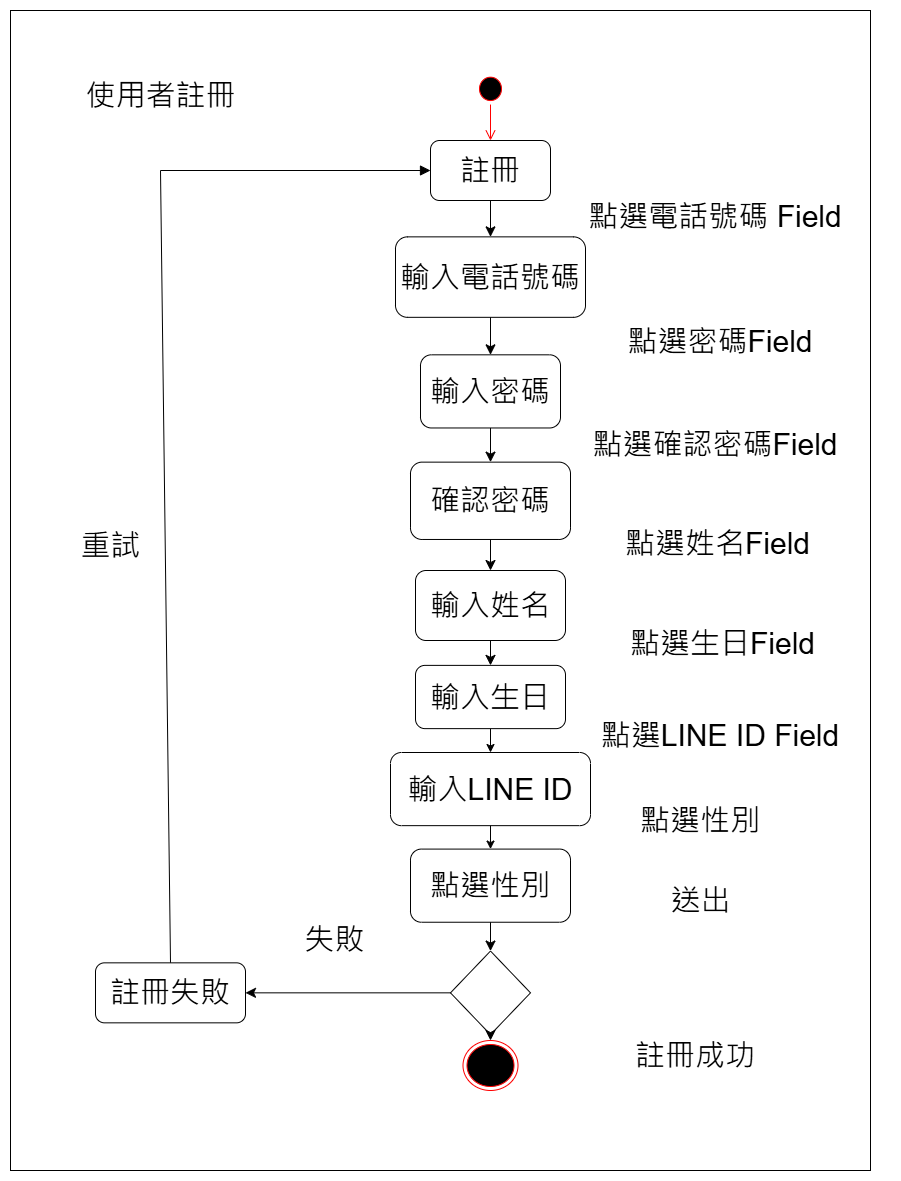
****

▲圖7-3-2 元件圖(員工)

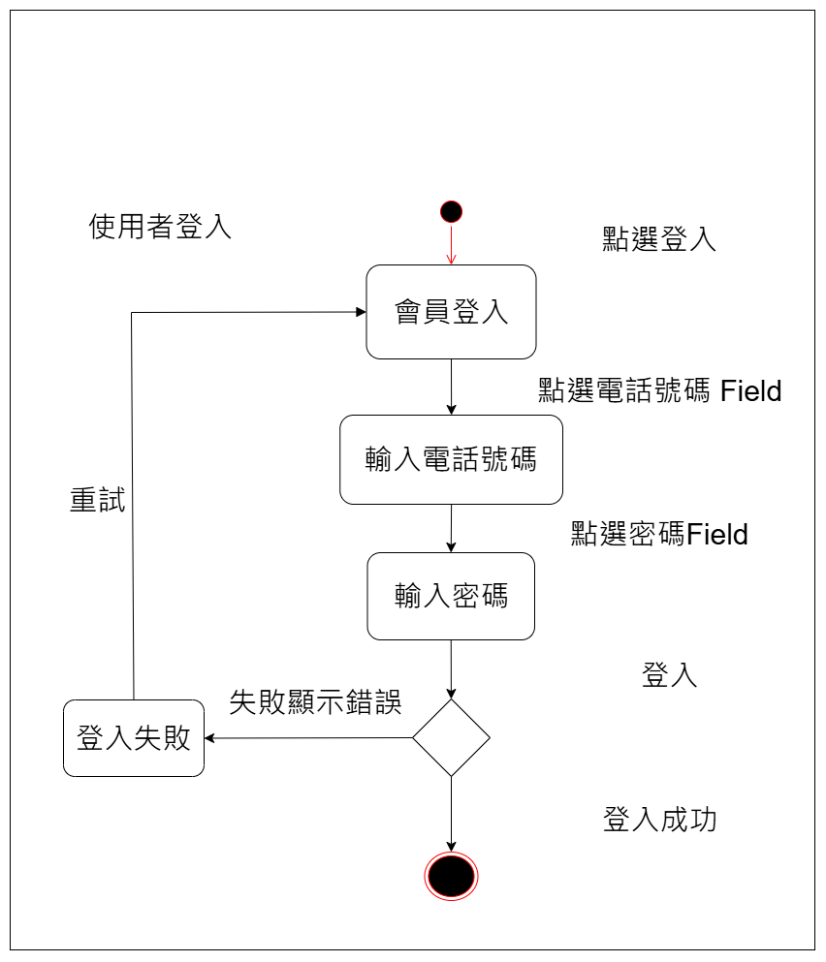
****

▲圖7-3-3 元件圖(管理員)

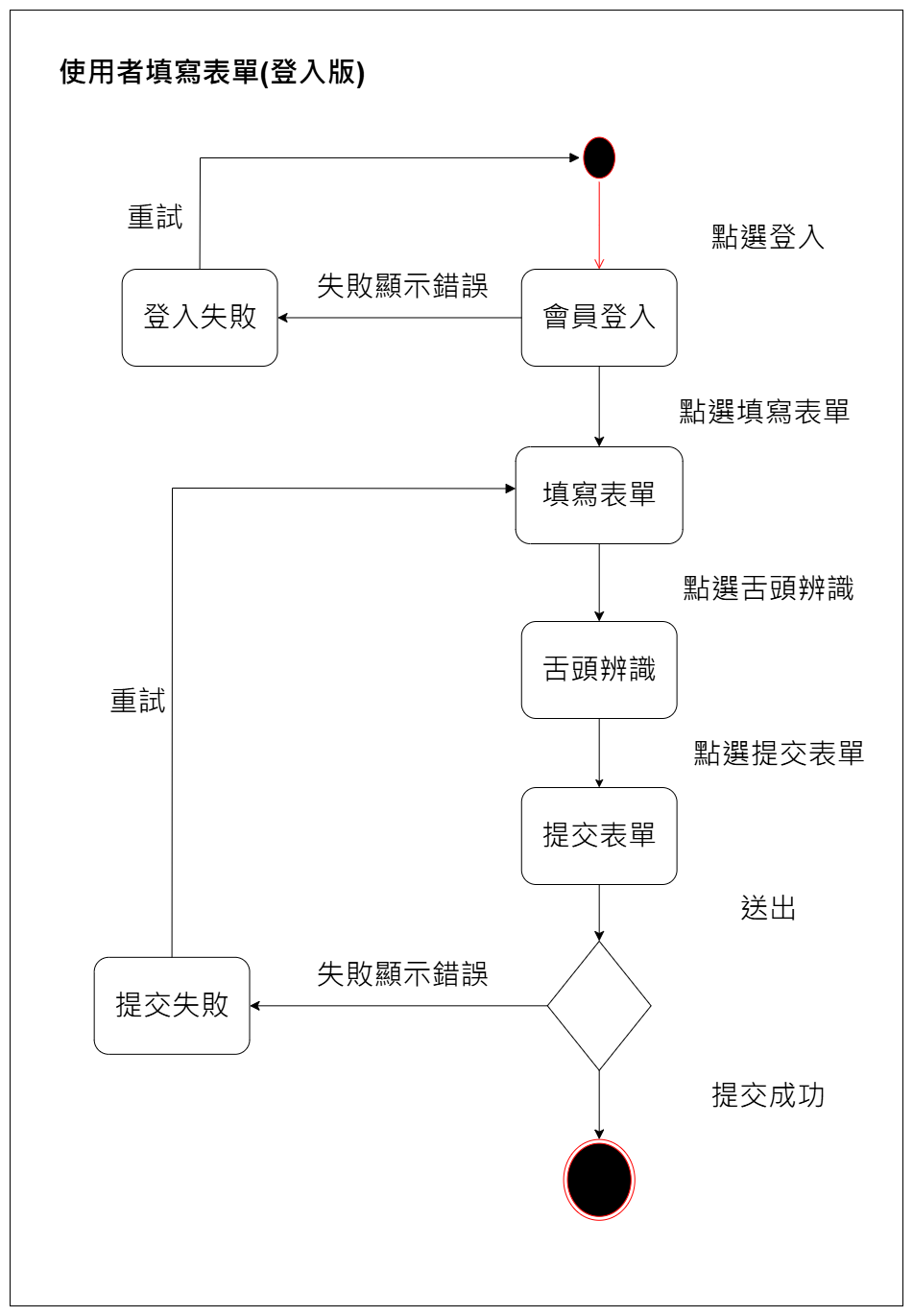
**7-4 狀態機、時序圖**



▲圖7-4-1 狀態機－使用者註冊



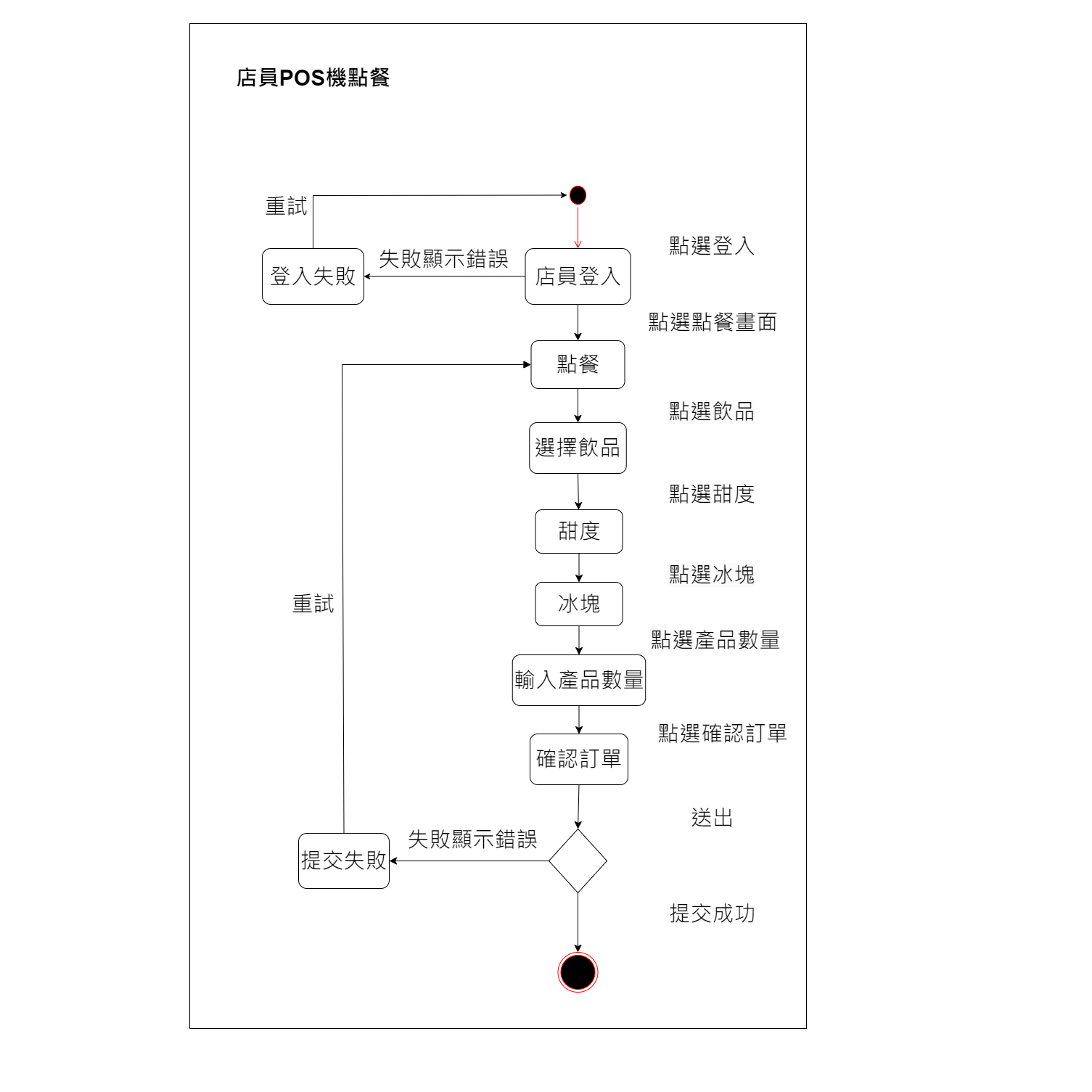
▲圖7-4-2 狀態機－使用者登入



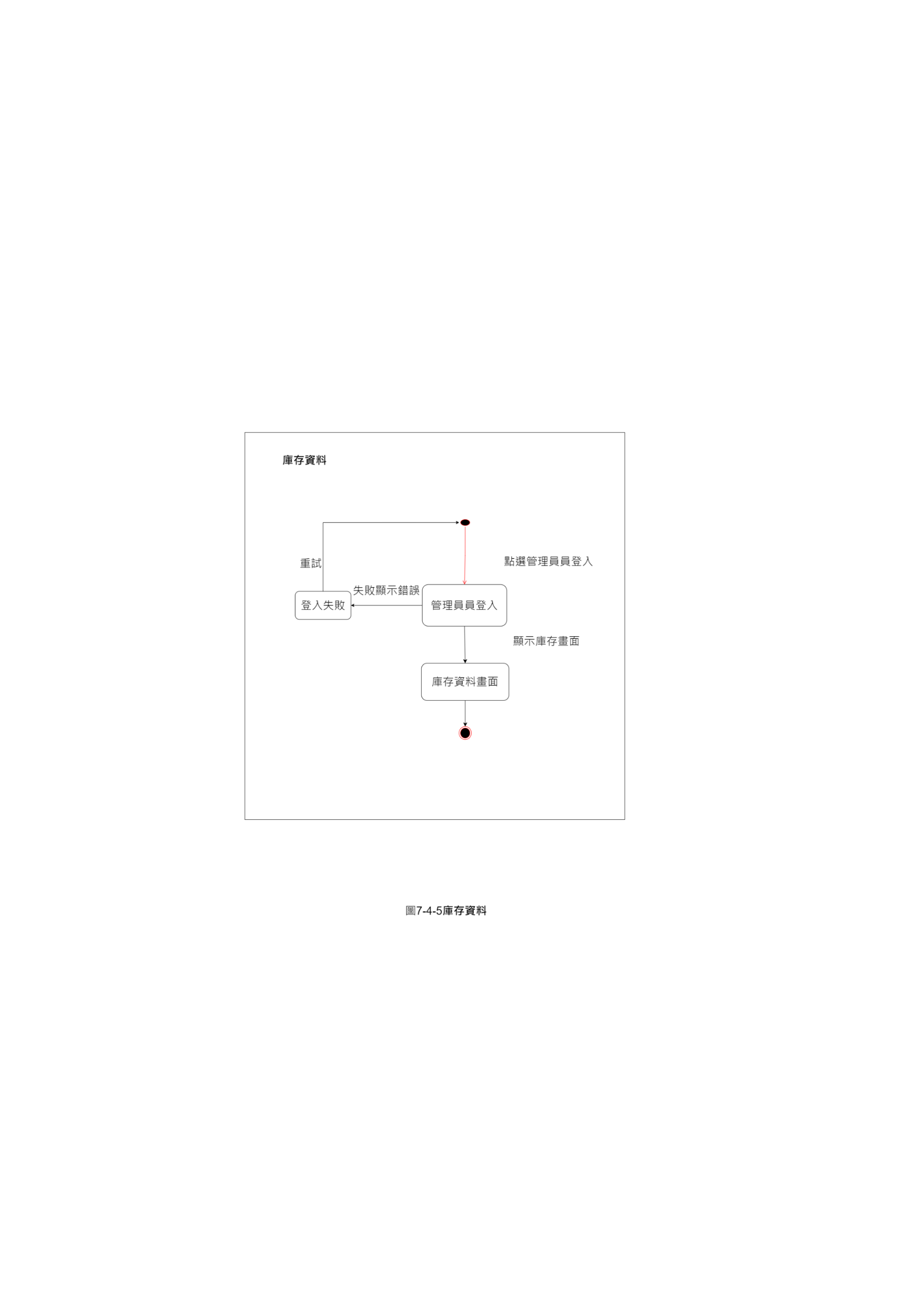
▲圖7-4-3 狀態機－使用者填表單(登入版)



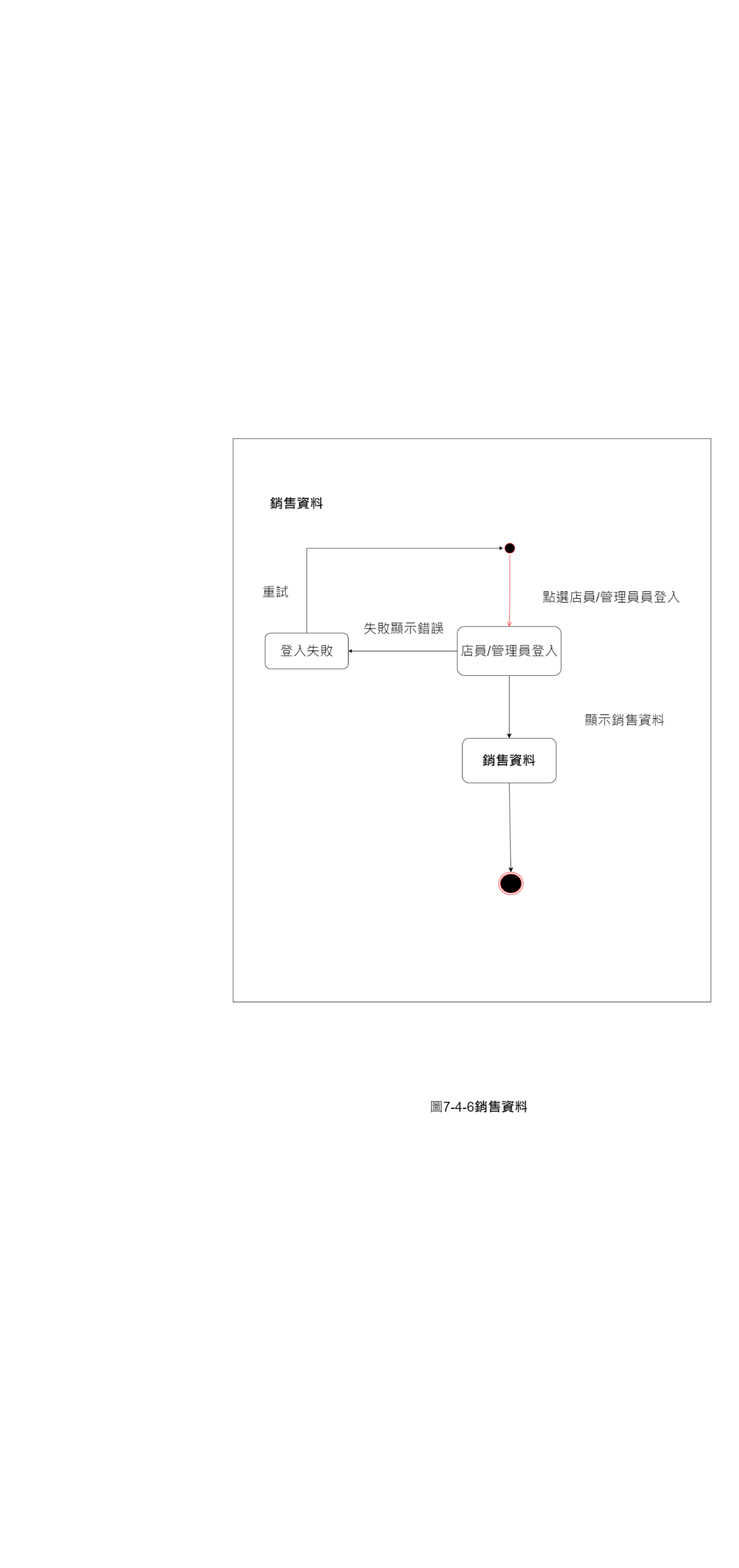
▲圖7-4-4 狀態機－使用者填表單(未登入版)



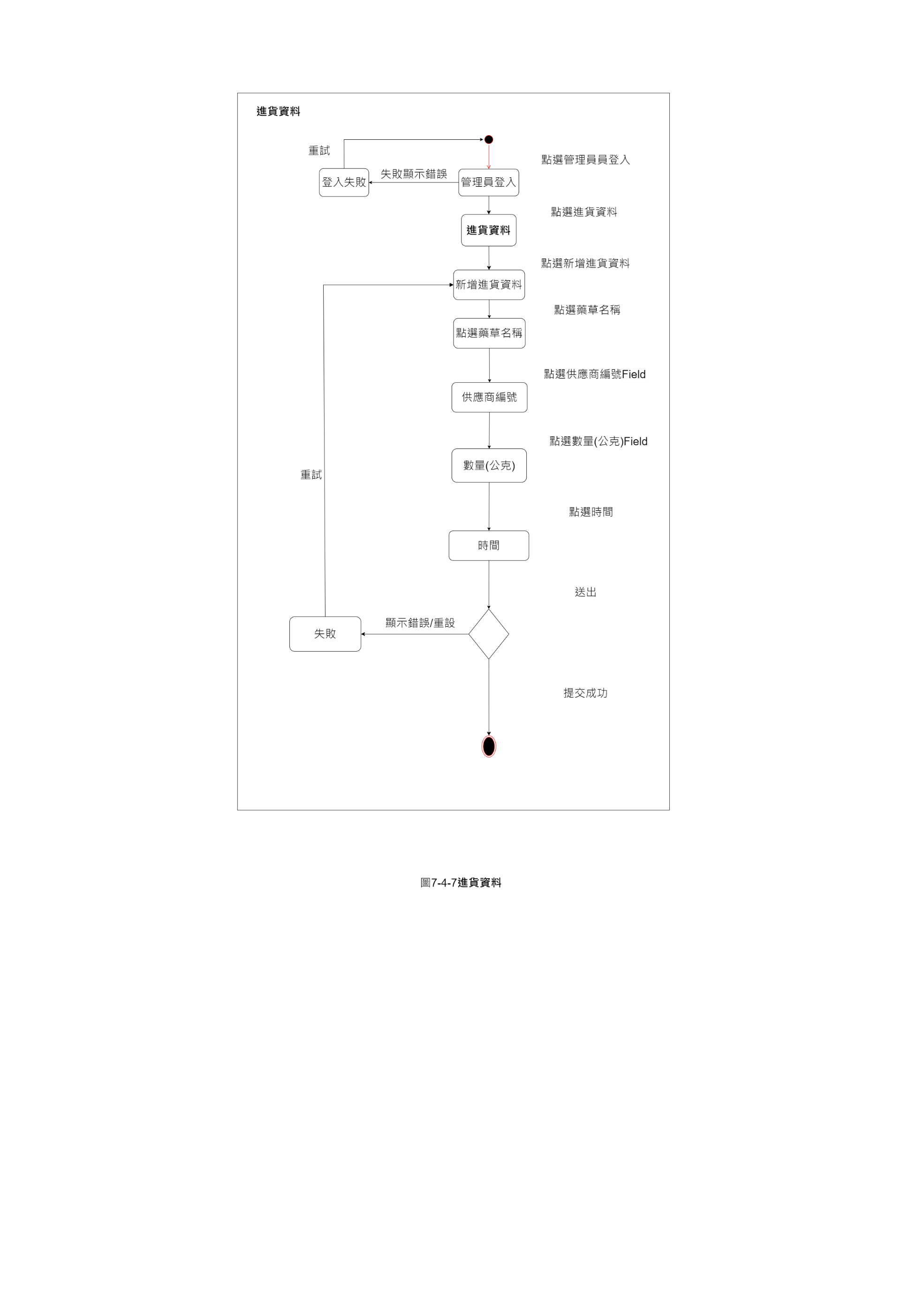
▲圖7-4-5 狀態機－店員POS機點餐



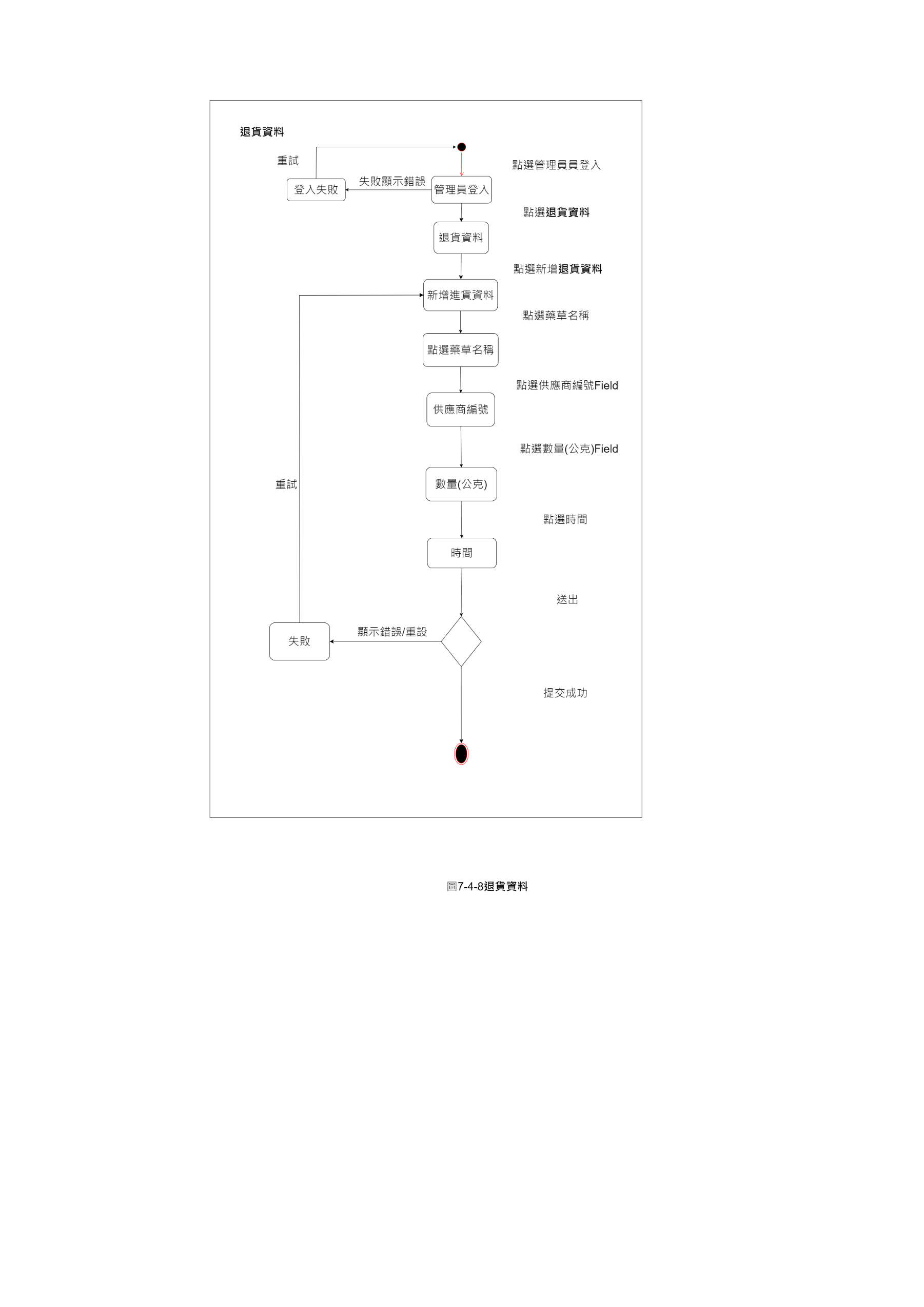
▲圖7-4-6 狀態機－庫存資料



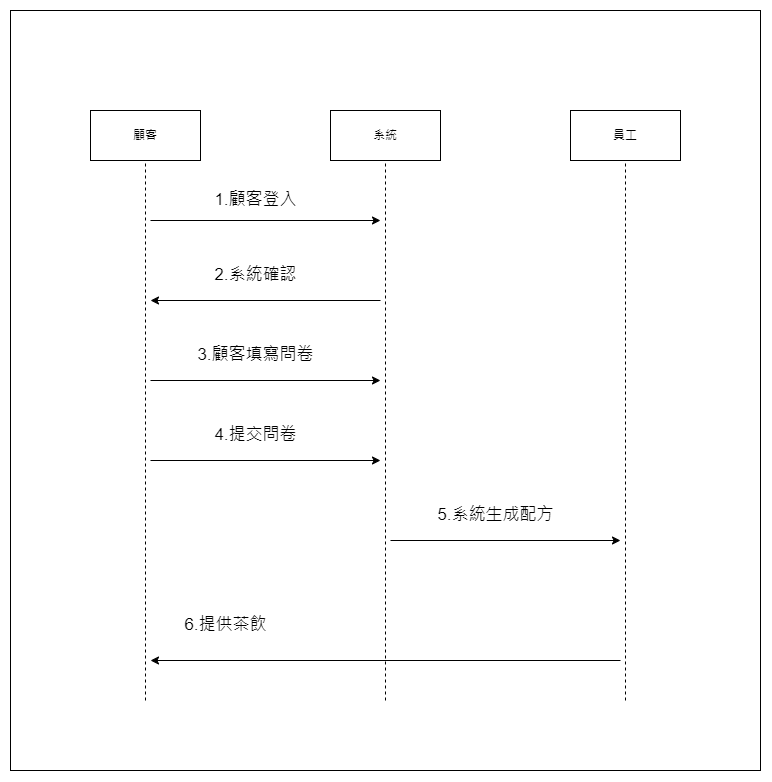
▲圖7-4-7 狀態機－銷售資料



▲圖7-4-8 狀態機－進貨資料



▲圖7-4-9 狀態機－退貨資料



▲圖7-4-10 時序圖

第8章　資料庫設計

8-1 資料庫關聯圖

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

▲圖8-1-1 關聯式資料庫ER圖

一張含有 文字, 螢幕擷取畫面, 陳列, 圖表 的圖片

自動產生的描述

▲圖8-1-2 關聯式資料庫ER圖

8-2 表格及其Meta data

customer存放顧客每位顧客的電話、密碼、名字、性別、生日、line id。symptom of question為存放顧客的問題結果，sale為存放銷貨基本資料，sale\_2則是跟sale作對應，若customer\_id、product\_name、order\_time跟sale相同，則可以對照出顧客上次點的茶飲紀錄。herb\_stock為存放藥草庫存，紀錄藥草的變動數。Purchase為記錄進貨的藥草量以及時間，時間可以查看藥草是否有無過期。

▼表8-2-1 資料表

|  |  |  |
| --- | --- | --- |
| 資料表編號 | 資料表英文名稱 | 資料表中文名稱 |
| 01 | customer | 用戶資料(登入) |
| 02 | symptom of question | 顧客問題資料 |
| 03 | purchase | 進貨資料 |
| 04 | sale | 銷售資料 |
| 05 | sale\_2 | 對比銷售資料 |
| 06 | herb\_stock | 藥草庫存資料 |

01 customer(用戶資料)

▼表8-2-2 資料表描述－01用戶資料

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 中文名稱 | 用戶資料 | | 資料表編號 | | 01 | | |
| 英文名稱 | customer | | 主索引 | |  | | |
| 資料表描述 | 顧客資料 | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 允許空值 | 中文 | | 備註 | 預設值 |
| customer\_id | VARCHAR | 15 | 否 | 用戶id(電話) | | PK |  |
| password | VARCHAR | 128 | 否 | 密碼 | |  |  |
| customer\_name | VARCHAR | 10 | 否 | 用戶姓名 | |  |  |
| birthday | VARCHAR | 7 | 否 | 用戶生日 | |  |  |
| sex | VARCHAR | 5 | 否 | 用戶性別 | |  |  |
| line\_id | VARCHAR | 45 | 是 | 用戶line id | |  |  |
| last\_login | DATETIME |  | 是 | 最後登入時間 | |  |  |
| is\_active | TINYINT | 1 | 否 | 帳號狀態 | | 0/否  1/是 |  |
| is\_staff | TINYINT | 1 | 否 | 員工權限 | | 0/否  1/是 |  |
| is\_superuser | TINYINT | 1 | 否 | 管理員權限 | | 0/否  1/是 |  |

02 symptom of question (顧客問題資料)

▼表8-2-3 資料表描述－02顧客問題資料

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 中文名稱 | 顧客問題資料 | | 資料表編號 | | 02 | | |
| 英文名稱 | symptom of question | | 主索引 | |  | | |
| 資料表描述 | 顧客問題資料 | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 允許空值 | 中文 | | 備註 | 預設值 |
| customer\_id | VARCHAR | 15 | 否 | 顧客電話 | | PK/FK |  |
| question\_time | DATETIME |  | 否 | 填寫問卷時間 | |  |  |
| Q1 | INT |  | 是 | Q1問題 | |  |  |
| Q2 | INT |  | 是 | Q2問題 | |  |  |
| Q3 | INT |  | 是 | Q3問題 | |  |  |
| Q4 | INT |  | 是 | Q4問題 | |  |  |
| Q5 | INT |  | 是 | Q5問題 | |  |  |
| Q6 | INT |  | 是 | Q6問題 | |  |  |

03 purchase (進貨資料)

▼表8-2-4 資料表描述－03 進貨資料

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 中文名稱 | 進貨資料 | | 資料表編號 | | 03 | | |
| 英文名稱 | purchase | | 主索引 | |  | | |
| 資料表描述 | 進貨資料 | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 允許空值 | 中文 | | 備註 | 預設值 |
| purchases\_id | INT |  | 否 | 藥草進貨編號 | | PK/AI |  |
| herbs\_id | INT |  | 否 | 藥草編號 | |  |  |
| supply\_id | VARCHAR | 45 | 否 | 供應商編號 | |  |  |
| herbs\_name | VARCHAR | 10 | 否 | 藥草名稱 | |  |  |
| purchases\_value | FLOAT |  | 否 | 進貨藥草量 | |  |  |
| purchases\_time | DATETIME |  | 否 | 進貨時間 | |  |  |

04 sale (銷售資料)

▼表8-2-5 資料表描述－04銷售資料

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 中文名稱 | 銷售資料 | | 資料表編號 | | 04 | | |
| 英文名稱 | sale | | 主索引 | | customer\_id  product\_name  order\_time | | |
| 資料表描述 | 銷售資料 | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 允許空值 | 中文 | | 備註 | 預設值 |
| sale\_id | INT |  | 否 | 銷售編號 | | PK/AI |  |
| customer\_id | VARCHAR | 15 | 否 | 顧客電話 | |  |  |
| product\_name | VARCHAR | 10 | 否 | 產品名稱 | |  |  |
| herbs\_id | INT |  | 否 | 藥草編號 | |  |  |
| herbs\_name | VARCHAR | 10 | 否 | 藥草名稱 | |  |  |
| sales\_value | FLOAT |  | 否 | 銷售藥草量 | |  |  |
| order\_time | DATETIME |  | 否 | 點餐時間 | |  |  |

05 sale\_2 (對比銷售資料)

▼表8-2-6 資料表描述－05對比銷售資料

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 中文名稱 | 對比銷售資料 | | 資料表編號 | | 05 | | |
| 英文名稱 | sale\_2 | | 主索引 | | product\_name  order\_time | | |
| 資料表描述 | 對比銷售資料 | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 允許空值 | 中文 | | 備註 | 預設值 |
| customer\_id | VARCHAR | 15 | 否 | 顧客電話 | | PK/FK |  |
| product\_name | VARCHAR | 10 | 否 | 產品 | | FK |  |
| order\_time | DATETIME |  | 否 | 點餐時間 | | FK |  |
| price | INT |  | 否 | 價格 | |  |  |

06 herbs\_stock (藥草庫存資料)

▼表8-2-7 資料表描述－06藥草庫存資料

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 中文名稱 | 藥草庫存資料 | | 資料表編號 | | 06 | | | |
| 英文名稱 | herbs\_stock | | 主索引 | |  | | | |
| 資料表描述 | 藥草庫存資料 | | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 允許空值 | 中文 | | 備註 | | 預設值 |
| herbs\_id | INT |  | 否 | 藥草編號 | | | PK |  |
| herbs\_name | VARCHAR | 10 | 否 | 藥草編號名字 | | |  |  |
| current\_stock | FLOAT |  | 否 | 藥草當前庫存 | | |  |  |

第9章　程式

**9-1 元件清單及其規格描述**

▼表9-1-1元件清單及其規格描述表

|  |  |  |  |
| --- | --- | --- | --- |
| Html檔(前端) | | | |
| 編號 | 群組 | 檔案名稱 | 功能 |
| 1-1-1 | 登入與註冊 | login | 登入 |
| 1-1-2 | register | 註冊 |
| 1-2-1 | 首頁 | url | 三個使用者登入模式 |
| 1-2-2 | index | 主畫面 |
| 1-3-1 | 使用者功能 | question | 問卷表單 |
| 1-3-2 | detect | 舌頭檢測 |
| 1-4-1 | 顧客權限 | history | 歷史紀錄 |
| 1-5-1 | 員工權限 | staff | 員工畫面 |
| 1-5-2 | pos | Pos功能 |
| 1-6-1 | 管理者權限 | manage | 管理者畫面 |
| 1-6-2 | perchaselist | 進貨畫面 |
| 1-6-3 | purchasepostform | 新增進貨 |
| 1-6-4 | edit | 編輯進貨 |
| 1-6-5 | delete | 刪除進貨 |
| 1-6-6 | herbstocklist | 庫存畫面 |
| 1-6-7 | 管理者權限 | salelist | 銷貨畫面 |
| Python 檔案（後端） | | | |
| 2-1-1 | 進銷存模組 | views.py | 進銷存視圖 |
| 2-1-2 | urls.py | 進銷存網址 |
| 2-1-3 | models.py | 進銷存模型 |
| 2-2-1 | 決策樹模組 | views.py | 決策樹視圖 |
| 2-2-2 | models.py | 決策樹模型 |
| 2-2-3 | herbs.csv | 訓練集資料 |
| 2-3-1 | 問卷模組 | views.py | 問卷視圖 |
| 2-3-2 | models.py | 問卷模型 |
| 2-3-3 | urls.py | 問卷網址 |
| 2-4-1 | 首頁模組 | views.py | 首頁視圖 |
| 2-4-2 | models.py | 首頁模型 |
| 2-4-3 | urls.py | 首頁網址 |
| 2-5-1 | 使用者模組 | views.py | 使用者視圖 |
| 2-5-2 | models.py | 使用者模型 |
| 2-5-3 | urls.py | 使用者網址 |
| 2-5-4 | form.py | 使用者驗證 |

|  |  |  |  |
| --- | --- | --- | --- |
| 2-6-1 | 管理員模組 | views.py | 管理員視圖 |
| 2-6-2 | models.py | 管理員模型 |
| 2-6-3 | urls.py | 管理員網址 |
| 2-6-4 | admin.py | 管理員後台 |
| 2-7-1 | 辨識模組 | views.py | 辨識視圖 |
| 2-7-2 | models.py | 辨識模型 |
| 2-7-3 | urls.py | 辨識網址 |
| 2-7-4 | utils.py | 辨識功能 |

**9-2 其他附屬之各種元件**

▼表9-2-1部分程式碼- form.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-5-4 | 程式名稱 | form.py |
| 目的 | 判斷註冊以及登入時是否有符合輸入的格式，以及在進貨和編輯時讓料號能夠對應藥草的編號 | | |
| 部分程式碼 | | | |
| class PostForm(forms.Form): #進貨資料驗證  HERBS\_CHOICES = [  ('', '--- 請選擇藥草 ---'),  ('1', '魚腥草'),  ('2', '白鶴靈芝'),  ('3', '積雪草'),  ('4', '金銀花'),  ('5', '蒲公英'),  ('6', '忍冬'),  ('7', '野茄樹'),  ('8', '金錢薄荷'),  ('9', '紫蘇'),  ('10', '鴨舌癀'),  ('11', '益母草'),  ('12', '薄荷'),  ('13', '甜菊'),  ('14', '咸豐草')  ]    herbs\_name = forms.ChoiceField(choices=HERBS\_CHOICES, required=True)  herbs\_id = forms.IntegerField(required=True, min\_value=1)  supply\_id = forms.CharField(label='供應商編號', max\_length=45)  purchases\_value = forms.FloatField(required=True)  purchases\_time = forms.DateTimeField(required=True, widget=forms.DateTimeInput(attrs={'type': 'datetime-local'}))  class CustomerRegistrationForm(forms.ModelForm): #註冊（處理用戶輸入）  password = forms.CharField(widget=forms.PasswordInput(attrs={'autocomplete': 'new-password', 'id': 'password' }),label='密碼')  #autocomplete': 'new-password告訴瀏覽器這是新的密碼 不應該保存任何值  password\_confirmation = forms.CharField(widget=forms.PasswordInput(), label='確認密碼')  birthday = forms.CharField(max\_length=7, label='生日')  sex = forms.ChoiceField(  choices=Customer.GENDER\_CHOICES,  widget=forms.RadioSelect,  label='性別'  )  customer\_id = forms.CharField(label='電話號碼', max\_length=10,)  class Meta:  model = Customer  fields = ['customer\_id', 'password', 'customer\_name', 'sex','line\_id','birthday']  labels = {  'customer\_name': '姓名',  'line\_id': 'LINE ID（可選填）'  }  def clean\_customer\_id(self):  customer\_id = self.cleaned\_data['customer\_id']  if not customer\_id.startswith('09'):  raise forms.ValidationError("電話號碼必須為09開頭")  if Customer.objects.filter(customer\_id=customer\_id).exists():  raise ValidationError("此電話號碼已被註冊")  return customer\_id  def clean\_customer\_name(self):  customer\_name = self.cleaned\_data.get('customer\_name')  if re.search(r'\d', customer\_name): # 使用正規表達檢查是否有數字  raise ValidationError("姓名不應包含數字")  return customer\_name  def clean(self):  cleaned\_data = super().clean()  password = cleaned\_data.get("password")  password\_confirmation = cleaned\_data.get("password\_confirmation")  if password and password\_confirmation:  if password != password\_confirmation:  self.add\_error('password\_confirmation', "密碼不一致")  return cleaned\_data  def clean\_birthday(self):  birthday = self.cleaned\_data.get('birthday')  # 確保生日為7位數字格式  if not re.match(r'^\d{7}$', birthday):  raise forms.ValidationError("生日必須是7位數字格式")    year = int(birthday[:4]) # 提取年  month = int(birthday[4:6]) # 提取月  day = int(birthday[6:7]) # 提取日  return birthday    class LoginForm(forms.Form): #登入系統  username = forms.CharField(label='電話號碼', max\_length=10,)  password = forms.CharField(label='　　密碼', widget=forms.PasswordInput(attrs={'autocomplete': 'new-password'}))  class PurchaseForm(forms.Form): #進貨編輯驗證  herbs\_id = forms.ChoiceField(  choices=[  ('', '--- 請選擇藥草 ---'),  ('1', '魚腥草'),  ('2', '白鶴靈芝'),  ('3', '積雪草'),  ('4', '金銀花'),  ('5', '蒲公英'),  ('6', '忍冬'),  ('7', '野茄樹'),  ('8', '金錢薄荷'),  ('9', '紫蘇'),  ('10', '鴨舌癀'),  ('11', '益母草'),  ('12', '薄荷'),  ('13', '甜菊'),  ('14', '咸豐草')  ],  required=True,  )  supply\_id = forms.CharField(max\_length=20, required=True)  purchases\_value = forms.IntegerField( required=True)  purchases\_time = forms.DateTimeField(required=True, widget=forms.DateTimeInput(attrs={'type': 'datetime-local'})) | | | |

▼表9-2-2 部分程式碼- utils.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-7-4 | 程式名稱 | utils.py |
| 目的 | 通過影像辨識獲得舌頭圖像，分析舌頭的顏色並判斷其狀態。最後，根據分析結果將顏色數據儲存到資料庫中。 | | |
| 部分程式碼 | | | |
| import numpy as np  import cv2  from matplotlib import pyplot as plt  import time  import threading  from django.utils import timezone  from myapp.models import TongueColor  # 函數功能:求出輸入矩陣中所有不為0元素的最大值與最小值  # 輸入:矩陣  # 輸出:最大值與最小值  def getMyMaxAndMin(I):  myMax = 0  myMin = 255  rows,columns= I.shape  for i in range(rows):  for j in range(columns):  if I[i][j] != 0:  if I[i][j] > myMax:  myMax = I[i][j]  if I[i][j] < myMin:  myMin = I[i][j]  return myMax,myMin  # 函數功能:迭代法確定色度法中的閾值，忽略矩陣中所有為0的元素  # 輸入:矩陣  # 輸出:符合要求的閾值  def iterativeThreshold(I):  rows,columns= I.shape  myMax,myMin = getMyMaxAndMin(I)  T = (myMax + myMin) / 2  temp = T  while True:  #計算所有小於temp且不為0元素的平均值  leftCount = 0  leftSum = 0  for i in range(rows):  for j in range(columns):  if I[i][j] != 0 and I[i][j] < temp:  leftCount += 1  leftSum += I[i][j]  if leftCount != 0:  leftAvg = leftSum / leftCount  else:  leftAvg = 0  #計算所有大於temp且不為0元素的平均值  rightCount = 0  rightSum = 0  for i in range(rows):  for j in range(columns):  if I[i][j] != 0 and I[i][j] >= temp:  rightCount += 1  rightSum += I[i][j]  if rightCount != 0:  rightAvg = rightSum / rightCount  else:  rightAvg = 0  T = (leftAvg + rightAvg) / 2  if temp == T:  break  temp = T  return T  # 函數功能:將輸入圖像拆分為H,S,V通道,並進行一系列處理提出舌頭的輪廓  # 輸入:原始圖像  # 輸出:二值化的closing矩陣，舌頭區域為255，非舌頭區域為0  def hsvDeal(img):  hsv = cv2.cvtColor(img,cv2.COLOR\_BGR2HSV)  h = hsv[:,:,0]  s = hsv[:,:,1]  v = hsv[:,:,2]  #二值化H通道與V通道  \_,thresh\_h = cv2.threshold(h,127,255,cv2.THRESH\_BINARY)  \_,thresh\_v = cv2.threshold(v,127,255,cv2.THRESH\_BINARY)  #對H通道與V通道進行“與”運算  hAndV = cv2.bitwise\_and(thresh\_h,thresh\_v)  #進行形態學“閉”運算，先膨脹後腐蝕  kernel = np.ones((5, 5), np.uint8)  closing = cv2.morphologyEx(hAndV, cv2.MORPH\_CLOSE, kernel)  #返回提取出的舌頭輪廓  return closing  # 函數功能:色度閾值法判斷圖像是否是舌頭  # 輸入:原圖像  # 輸出:bool值  def tongueColorDetect(img):  #提取出舌頭區域  closing = hsvDeal(img)  #舌頭的掩膜,舌頭區域為1，非舌頭區域為0  mask = closing.copy() / 255  #計算“舌頭”區域的占整個圖像的比例，太低的話說明不是舌頭  effectiveRate = np.sum(mask == 1) / (mask.shape[0] \* mask.shape[1])  #非舌頭區域置為白色  tongueArea = img.copy()  tongueArea[mask == 0] = [255,255,255]  cv2.imshow('Tongue',tongueArea)  #色度閾值法，I = R - (G + B) / 2  #分離舌質與舌苔  I = tongueArea[:,:,2] - (tongueArea[:,:,0] + tongueArea[:,:,1]) / 2  I = I \* mask  iterT = iterativeThreshold(I) #調用函數獲得閾值  colorResult = tongueArea.copy()  colorResult[I >= iterT] = [255,255,255]  # cv2.imshow("Split",colorResult)  avg\_r = np.mean(tongueArea[(I > 0) & (I < iterT),2])  avg\_g = np.mean(tongueArea[I >= iterT,1])  avg\_compare = np.mean(I[(I > 0) & (I < iterT)])  # print("The average value of compare is:" + str(avg\_compare))  # print("The average value of r is:" + str(avg\_r))  # print("The average value of g is:" + str(avg\_g))  # print("The rate of the effective area is:" + str(effectiveRate))  # 新增：分析舌頭顏色  hsv = cv2.cvtColor(tongueArea, cv2.COLOR\_BGR2HSV)  avg\_hue = np.mean(hsv[mask == 1, 0])  avg\_sat = np.mean(hsv[mask == 1, 1])  avg\_val = np.mean(hsv[mask == 1, 2])  # 計算相對亮度  overall\_brightness = np.mean(img)  relative\_brightness = avg\_val / overall\_brightness  # 調試輸出  # print(f"avg\_hue: {avg\_hue}, avg\_sat: {avg\_sat}, avg\_val: {avg\_val}")  # print(f"relative\_brightness: {relative\_brightness}")  # 定義顏色範圍（這些值可能需要根據實際情況調整）  if relative\_brightness > 1.3 and avg\_sat < 50: # 相對亮度高且飽和度低表示過白  color = "white"  elif 0 <= avg\_hue < 20 or 330 <= avg\_hue <= 360: # 紅色的色相範圍  if avg\_sat > 100: # 高飽和度表示過紅  color = "red"  else:  color = "pink"  else:  color = "pink"  # 可以調節的四個參數  if avg\_r >= 120 and avg\_compare >= 20 and avg\_g < 200 and effectiveRate >= 0.1:  return (True, color)  else:  return (False, None)  def run\_tongue\_detection(user):  WIDTH = 320  HEIGHT = 240  cap = cv2.VideoCapture(0)  cap.set(3, WIDTH)  cap.set(4, HEIGHT)  last\_color = None  color\_start\_time = None  while True:  ret, frame = cap.read()  cv2.rectangle(frame, (WIDTH // 3 - 5, HEIGHT // 5 \* 3 - 5), (WIDTH // 3 \* 2, HEIGHT), [0, 255, 0], 1)  isTongue, tongue\_color = tongueColorDetect(frame[HEIGHT // 5 \* 3:HEIGHT, WIDTH // 3:WIDTH // 3 \* 2, :])  if isTongue:  if tongue\_color:  cv2.putText(frame, f' {tongue\_color}', (5, 30), cv2.FONT\_HERSHEY\_COMPLEX, 0.8, (0, 0, 255), 2, 8, 0)  if last\_color == tongue\_color:  if time.time() - color\_start\_time >= 2:  save\_color\_to\_db(tongue\_color, user)  break  else:  last\_color = tongue\_color  color\_start\_time = time.time()  else:  cv2.putText(frame, 'ok', (5, 30), cv2.FONT\_HERSHEY\_COMPLEX, 0.8, (0, 0, 255), 2, 8, 0)  else:  cv2.putText(frame, 'cant found tongue', (5, 30), cv2.FONT\_HERSHEY\_COMPLEX, 0.8, (0, 0, 255), 2, 8, 0)  cv2.imshow('Frame', frame)  if cv2.waitKey(10) & 0xFF == ord('q'):  break  cap.release()  cv2.destroyAllWindows()  def save\_color\_to\_db(color, user):  customer\_id = user.customer\_id if user.is\_authenticated else '0'  # 使用Django ORM存儲顏色數據  TongueColor.objects.create(customer\_id=customer\_id, color=color) | | | |

▼表9-2-3 部分程式碼-views.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-2-1 | 程式名稱 | views.py |
| 目的 | 使用CART決策樹去抓取最合適的藥材，再經由平均相加讓藥材克數達到5g。 | | |
| 部分程式碼 | | | |
| csv\_path = os.path.join(settings.BASE\_DIR, 'static', 'csv', 'herbs.csv')  df = pd.read\_csv(csv\_path, header=None)  df.columns = ['睡不好', '半暝還在嗨', '早上哈啾', '癢癢', '胃生氣', '厭世生理期', '結果']  # 定義選項對應  options\_mapping = {  'nosleep': {  '1': '感到憂鬱',  '2': '感到焦慮',  '3': '容易緊張',  '4': '無'  },  'semi\_darkness': {  '1': '22點以前入睡',  '2': '22-24點入睡',  '3': '24-3點入睡',  '4': '3點以後入睡'  },  'sneezing': {  '1': '長期，伴有呼吸胸悶',  '2': '偶發，伴有呼吸胸悶',  '3': '長期，無呼吸胸悶',  '4': '無'  },  'itchiness': {  '1': '長期過敏',  '2': '短期過敏',  '3': '有就醫拿藥',  '4': '無過敏'  },  'stomach\_anger': {  '1': '胃脹氣',  '2': '反胃',  '3': '胃食道逆流',  '4': '無'  },  'menstrual\_anguish': {  '1': '重度疼痛',  '2': '輕度疼痛',  '3': '不會痛',  '4': '無生理期'  }  }  def question(request):  id\_result = None  if request.method == 'POST' and 'confirm\_button' in request.POST: # 新增按下按鈕才能更改資料庫中的數值  print(request.POST)  nosleep = options\_mapping['nosleep'][request.COOKIES.get('nosleep')]  semi\_darkness = options\_mapping['semi\_darkness'][request.COOKIES.get('semi\_darkness')]  sneezing = options\_mapping['sneezing'][request.COOKIES.get('sneezing')]  itchiness = options\_mapping['itchiness'][request.COOKIES.get('itchiness')]  # 獲取胃生氣的選擇  stomach\_anger = request.POST.get('stomach\_anger', '')  stomach\_anger\_choices = stomach\_anger.split(', ') if stomach\_anger else []  # 將選擇的值轉換為字符串  stomach\_anger\_str = ', '.join(stomach\_anger\_choices)  print("stomach\_anger\_choices:", stomach\_anger\_choices) # 用於調試  print("stomach\_anger\_str:", stomach\_anger\_str) # 用於調試  menstrual\_anguish = options\_mapping['menstrual\_anguish'][request.COOKIES.get('menstrual\_anguish')]  bitter = request.COOKIES.get('bitter')  # 獲取顧客編號  customer\_id = None  if request.user.is\_authenticated:  customer\_id = request.user.customer\_id  id\_result = customer\_id  else:  next\_counter = DailyCounter.get\_next\_counter()  customer\_id = f'{next\_counter:03d}'  id\_result = customer\_id  # 取得台北的時區  taipei\_tz = pytz.timezone('Asia/Taipei')  # 獲取現在的台北時間  taipei\_now = datetime.now(taipei\_tz)  # 將台北時間轉換為 UTC 時間  utc\_now = taipei\_now.astimezone(pytz.utc)  # 將症狀資料存入資料庫  SymptomOfQuestion.objects.create(  customer\_id=request.user.customer\_id if request.user.is\_authenticated else 0,  question\_time=utc\_now,  q1=nosleep,  q2=semi\_darkness,  q3=sneezing,  q4=itchiness,  q5=stomach\_anger,  q6=menstrual\_anguish  )  if nosleep:  def get\_herbs\_result(nosleep, semi\_darkness, sneezing, itchiness, stomach\_anger, menstrual\_anguish):  # 準備特徵數據  input\_data = pd.DataFrame({  '睡不好': [nosleep],  '半暝還在嗨': [semi\_darkness],  '早上哈啾': [sneezing],  '癢癢': [itchiness],  '胃生氣': [stomach\_anger],  '厭世生理期': [menstrual\_anguish]  })  # 將分類變數轉換為數值型  input\_data\_encoded = pd.get\_dummies(input\_data)  # 準備訓練數據  X = df[['睡不好', '半暝還在嗨', '早上哈啾', '癢癢', '胃生氣', '厭世生理期']]  y = df['結果']  # 將訓練數據轉換為數值型  X\_encoded = pd.get\_dummies(X)  # 創建決策樹分類器  clf = DecisionTreeClassifier(criterion='entropy')  clf.fit(X\_encoded, y)  # 進行預測  predicted\_result = clf.predict(input\_data\_encoded.reindex(columns=X\_encoded.columns, fill\_value=0))  # 返回預測結果  return predicted\_result[0]  result = get\_herbs\_result(nosleep, semi\_darkness, sneezing, itchiness, stomach\_anger, menstrual\_anguish)  print(f"預測結果: {result}")  # 處理結果並輸出  result = result.strip("[]")  herbs\_list = [herb.strip().strip("'") for herb in result.split(",")]  if '無' not in stomach\_anger\_choices:  selected\_options = [option.strip() for option in stomach\_anger.split(',')]  num\_options = len(selected\_options)    # 根據選擇的數量調整推薦的藥材  if num\_options == 1:  if '’積雪草 0.25g’' in herbs\_list:  herbs\_list[herbs\_list.index('’積雪草 0.25g’')] = '’積雪草 0.1g’'  else:  herbs\_list.append('’積雪草 0.1g’')  elif num\_options == 2:  if '’積雪草 0.25g’' in herbs\_list:  herbs\_list[herbs\_list.index('’積雪草 0.25g’' )] = '’積雪草 0.25g’'  else:  herbs\_list.append('’積雪草 0.25g’' )  elif num\_options == 3:  if '’積雪草 0.25g’' in herbs\_list:  herbs\_list[herbs\_list.index('’積雪草 0.25g’' )] = '’積雪草 0.5g’'  else:  herbs\_list.append('’積雪草 0.5g’' )  print(f"解析後的藥材列表: {herbs\_list}")    if not herbs\_list:  print("無法解析藥材列表，請檢查數據格式。")  return redirect('/question/')  else:  # 轉換為小數  herbs\_dict = {}  for herb in herbs\_list:  match = re.search(r'(.\*) (\d+\.?\d\*)g', herb)  if match:  name, amount = match.groups()  herbs\_dict[name] = float(amount)  else:  print(f"無法解析藥材: {herb}")    # 計算總量並調整克數  total\_amount = sum(herbs\_dict.values())    if total\_amount == 0:  print("藥材總量為零，無法調整。")  return redirect('/question/')  else:  scale\_factor = 5 / total\_amount    adjusted\_herbs = []  for name, amount in herbs\_dict.items():  adjusted\_amount = round(amount \* scale\_factor, 2)  adjusted\_herbs.append(f"{name} {adjusted\_amount}g")    print("調整後的藥材配方:")  for herb in adjusted\_herbs:  print(herb)  print(f"總量: {sum(float(h.split()[-1][:-1]) for h in adjusted\_herbs)}g")    # 儲存調整後的配方  final\_herbs = [herb.split()[0].replace('‘', '').replace('’', '') for herb in adjusted\_herbs]  dosages=[float(h.split()[-1][:-1]) for h in adjusted\_herbs]  # 在這裡處理最終藥材的選擇  herbs = []  if herbs\_list:  for item in herbs\_list:  parts = item.split()  herbs.append(parts[0])  herbs\_mapping = {  "魚腥草": 1, "白鶴靈芝": 2, "積雪草": 3,  "金銀花": 4, "蒲公英": 5, "忍冬": 6,  "野茄樹": 7, "金錢薄荷": 8, "紫蘇": 9,  "鴨舌黃": 10, "益母草": 11, "薄荷": 12,  "甜菊": 13, "咸豐草": 14  }  # 根據按鈕值進行處理  final\_herbs = []  for herb in herbs:  herb = herb.replace('‘', '').replace('’', '').strip()  if "or" in herb:  options = herb.split("or")  if bitter == "True":  if "蒲公英" in options:  final\_herbs.append("蒲公英")  if "益母草" in options:  final\_herbs.append("益母草")  elif bitter == "False":  if "白鶴靈芝" in options:  final\_herbs.append("白鶴靈芝")  if "鴨舌黃" in options:  final\_herbs.append("鴨舌黃")  else:  final\_herbs.append(herb)  print('最終藥材列表:', final\_herbs) # 輸出最終藥材列表  product\_name = "客製化"  order\_time = utc\_now # 現在時間  customer\_id = request.user.customer\_id if request.user.is\_authenticated else id\_result  global show\_result  show\_result = []  for i in range(len(final\_herbs)):  show\_result.append(final\_herbs[i] + ":" + str(dosages[i]) + "g")  Sale.objects.create(  customer\_id=customer\_id,  product\_name=product\_name,  herbs\_id=herbs\_mapping.get(final\_herbs[i]),  herbs\_name=final\_herbs[i],  sales\_value=dosages[i],  order\_time=order\_time  )    show = Customer.objects.filter(customer\_id=customer\_id).first()  if request.user.is\_authenticated:  customer\_name = show.customer\_name  show\_result.insert(0, "顧客名字：" + customer\_name)  show\_result.insert(0, "顧客電話：" + customer\_id)  else:  show\_result.insert(0, "顧客名字：" + "Guest")  show\_result.insert(0, "未登入顧客編號：" + customer\_id)    a = ShowResult.objects.get(show\_id=0)  a.data = show\_result  a.save()  print('show\_result:',show\_result)  return redirect('/question/')  return render(request, "question.html") | | | |

▼表9-2-4 部分程式碼-views.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-6-1 | 程式名稱 | views.py |
| 目的 | 處理pos機接取前端數據，以及顯示點餐記錄在pos機上面。 | | |
| 部分程式碼 | | | |
| @ensure\_csrf\_cookie  @user\_passes\_test(lambda user:user.is\_staff,login\_url='/accounts/login/')  def pos(request):  if request.method == 'POST':  try:  # 解析 JSON 資料，注意要使用 request.body  data = json.loads(request.body)  orders = data.get('orders', []) # 獲取 orders 陣列  # 取得當前時間  current\_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")  # 初始化顧客編號  customer\_id = None  # 取得表單提交的顧客信息  customer\_info = data.get('customer\_info')  if customer\_info and 'customer\_id' in customer\_info:  customer\_id = customer\_info['customer\_id']  last\_order\_time = Sale.objects.filter(customer\_id=customer\_id).aggregate(last\_order\_time=Max('order\_time'))['last\_order\_time']  # 如果上一個訂單的時間與目前時間相同，則繼續使用該顧客編號  if last\_order\_time and last\_order\_time == current\_time:  pass  else:  customer\_id = None # 置空顧客編號，以便後面重新生成  # 循環處理訂單  for order in orders:  symptom = order.get('symptom')  quantity = order.get('quantity')  if not customer\_id:  # 如果顧客編號為空，則產生一個新的顧客編號  next\_counter = DailyCounter.get\_next\_counter()  customer\_id = f'{next\_counter:03d}'  sale\_value = 2.5 \* int(quantity)  if symptom == "星夜寧靜":  product = "星夜寧靜"  herb = '魚腥草'  herbs\_id = 1  elif symptom == "宵福調和":  product = "宵福調和"  herb = '白鶴靈芝'  herbs\_id = 2  elif symptom == "鼻福寧茶":  product = "鼻福寧茶"  herb = '金銀花'  herbs\_id = 4  elif symptom == "悅膚寧茶":  product = "悅膚寧茶"  herb = '忍冬'  herbs\_id = 6  elif symptom == "慰胃來茶":  product = "慰胃來茶"  herb = '積雪草'  herbs\_id = 3  else:  product = "月悅茶"  herb = '鴨舌癀'  herbs\_id = 10  Sale.objects.create(customer\_id=customer\_id, product\_name=product, herbs\_id=herbs\_id, herbs\_name=herb, sales\_value=sale\_value, order\_time=current\_time)    return JsonResponse({'message': '點餐成功！', 'refresh': True})  except json.JSONDecodeError as e:  return JsonResponse({'error': '無效的 JSON 資料'}, status=400)  show=ShowResult.objects.filter(show\_id=0).first()  show\_result=show.data  print(show\_result)  return render(request, "POS介面 new.html", {'show\_result': show\_result}) | | | |

▼表9-2-5部分程式碼-views.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-1-1 | 程式名稱 | views.py |
| 目的 | 進貨、庫存和銷售的管理功能，包括新增、編輯和刪除進貨，以及檢查庫存和顯示銷售記錄，使用者可以通過不同的表單界面進行操作 | | |
| 部分程式碼 | | | |
| #--------------------------------------------------------------------------進貨表單  def perchaselist(request): #進貨表單設定  purchases = Purchase.objects.all().order\_by('purchases\_id')  return render(request, "purchaselist.html", locals())  def purchasepostform(request): #新增進貨資料  if request.method == "POST":  postform = PostForm(request.POST)  if postform.is\_valid():  herbs\_id = postform.cleaned\_data['herbs\_name']  supply\_id = postform.cleaned\_data['supply\_id']  purchases\_value = postform.cleaned\_data['purchases\_value']  purchases\_time = postform.cleaned\_data['purchases\_time']    # 從 choices 中獲取對應的名稱  herbs\_name = dict(PostForm.HERBS\_CHOICES).get(herbs\_id)    if herbs\_id: # 確保herbs\_id不是空的  unit = Purchase.objects.create(  herbs\_name=herbs\_name,  supply\_id=supply\_id,  herbs\_id=herbs\_id,  purchases\_value=purchases\_value,  purchases\_time=purchases\_time  )  return redirect('/perchaselist/')  else:  postform = PostForm()  return render(request, "purchasepostform.html", {'postform': postform})  def delete(request,id=None): #刪除進貨資料  if id!=None:  if request.method == "POST":  id=request.POST['cId']  try:  unit = Purchase.objects.get(purchases\_id=id) #主鍵傳入的值跟網址的id比對  unit.delete()  return redirect('/perchaselist/')  except:  message = "讀取錯誤！"  return render(request, "delete.html", locals())  def edit(request, id=None): # 編輯進貨資料  unit = get\_object\_or\_404(Purchase, purchases\_id=id)  if request.method == "POST":  form = PurchaseForm(request.POST)  if form.is\_valid():  unit.herbs\_id = form.cleaned\_data['herbs\_id']  herbs\_name\_mapping = {  '1': '魚腥草',  '2': '白鶴靈芝',  '3': '積雪草',  '4': '金銀花',  '5': '蒲公英',  '6': '忍冬',  '7': '野茄樹',  '8': '金錢薄荷',  '9': '紫蘇',  '10': '鴨舌癀',  '11': '益母草',  '12': '薄荷',  '13': '甜菊',  '14': '咸豐草',  }  unit.herbs\_name = herbs\_name\_mapping.get(unit.herbs\_id, '')  unit.supply\_id = form.cleaned\_data['supply\_id']  unit.purchases\_value = form.cleaned\_data['purchases\_value']  unit.purchases\_time = form.cleaned\_data['purchases\_time']  unit.save()  return redirect('/perchaselist/')  else:  form = PurchaseForm(initial={  'herbs\_id': unit.herbs\_id,  'supply\_id': unit.supply\_id,  'purchases\_value': unit.purchases\_value,  'purchases\_time': unit.purchases\_time  })    return render(request, "edit.html", {'form': form, 'unit': unit})  #----------------------------------------庫存表單  def herbstocklist(request): #庫存表單設定  herbs = models.HerbStock.objects.all().order\_by('herbs\_id')  return render(request, "herbstocklist.html", locals())  def check\_inventory(request):  low\_inventory\_herbs = HerbStock.objects.filter(current\_stock\_\_lt=30) #藥草存貨30以下發出警告  low\_inventory\_list = [{'herbs\_name': herb.herbs\_name, 'current\_stock': herb.current\_stock} for herb in low\_inventory\_herbs]  return JsonResponse({'low\_inventory': low\_inventory\_list})  #----------------------------------------銷售表單  def salelist(request): #銷售表單設定  sort\_order = request.GET.get('saleid')  if sort\_order == 'ascending':  sales = Sale.objects.all().order\_by('sale\_id')  else:  sales = Sale.objects.all().order\_by('-sale\_id')  return render(request, "salelist.html", {'sales': sales})  def salelist\_staff(request): #銷售表單(員工)設定  sort\_order = request.GET.get('saleid')  if sort\_order == 'ascending':  sales = Sale.objects.all().order\_by('sale\_id')  else:  sales = Sale.objects.all().order\_by('-sale\_id')  return render(request, "salelist\_staff.html", {'sales': sales}) | | | |

第10章　測試模型

**10-1 測試計畫**

以顧客、店員、管理者角度測試各功能是否能正常運作，而測試方向主要為以下列出的測試清單，確認每項功能是否皆能正確執行。

1. 顧客

會員註冊：確認使用者能夠註冊。

會員登入：確認使用者能夠登入到首頁。

會員登出：確認使用者能夠登出回到首頁。

歷史紀錄：有登入會員可知道前一次點餐紀錄。

客製化表單：確認使用者能夠填寫表單、提交問卷，並得到提交成功的確認。

舌頭影像辨識：確認使用者能夠辨識舌頭顏色，舌頭辨識成功傳送到資料庫。

1. 店員

POS機點餐: 確認店員能從POS機幫顧客點餐。

POS機收到訂單: 確認店員能從POS機收到其他客人訂單餐點。

POS機收到庫存不足: 讓店員知道庫存不夠,需要補貨或跟客人說無法製作餐點。

1. 管理者

進貨/退貨資料表: 確認管理者能夠看到進貨/退貨資料。

庫存資料表: 確認管理者能夠看到庫存資料。

銷貨資料表: 確認管理者能夠看到銷貨資料。

**10-2 測試個案與測試結果資料**

▼表10-2-1功能描述-會員註冊

|  |  |
| --- | --- |
|  | |
| 功能名稱 | 會員註冊 |
| 測試目標 | 確認使用者能夠註冊 |
| 測試作業 | 1. 輸入帳號、生日、姓名、LINEID(可選填)、性別與密碼。 2. 驗證資料庫是否有相同的資訊。 |
| 測試結果 | 成功 |

▼表10-2-2 功能描述-會員登入

|  |  |
| --- | --- |
|  | |
| 功能名稱 | 會員登入 |
| 測試目標 | 確認使用者能夠登入 |
| 測試作業 | 1.輸入帳號與密碼。  2.驗證資料庫是否有相同的資訊。 |
| 測試結果 | 成功 |

▼表10-2-3 功能描述-登出

|  |  |
| --- | --- |
|  | |
| 功能名稱 | 登出 |
| 測試目標 | 確認使用者能夠登出回到首頁 |
| 測試作業 | 1.點選【登出】。  2.測試是否能夠登出。  3.是否有完成。 |
| 測試結果 | 成功 |

▼表10-2-4 功能描述-填寫客製化表單(未登入版)

|  |  |
| --- | --- |
|  | |
| 功能名稱 | 填寫客製化表單(未登入版) |
| 測試目標 | 確認使用者能夠填寫客製化表單、提交問卷，並得到提交成功的確認。 |
| 測試作業 | 1.點選【客製化表單】。  2.測試是否能夠填寫客製化表單、提交問卷。  3.是否有完成提交。 |
| 測試結果 | 成功 |

▼表10-2-5 功能描述-填寫客製化表單(登入版)

|  |  |
| --- | --- |
|  | |
| 功能名稱 | 填寫客製化表單(登入版) |
| 測試目標 | 確認使用者能夠登入會員填寫客製化表單、提交問卷，  並得到提交成功的確認。 |
| 測試作業 | 1.點選【客製化表單】。  2.測試是否能夠登入會員填寫客製化表單、提交問卷。  3.是否有登入會員且完成提交。 |
| 測試結果 | 成功 |

▼表10-2-6 功能描述-歷史紀錄

|  |  |
| --- | --- |
|  | |
| 功能名稱 | 歷史紀錄 |
| 測試目標 | 登入會員可知道前一次點餐紀錄 |
| 測試作業 | 1.點選【登入】。  2.點選【歷史紀錄】。  3.測試是否能夠知道前一次點餐紀錄。  4.是否有看到前一次點餐紀錄。 |
| 測試結果 | 成功 |

▼表10-2-7 功能描述-影像辨識

|  |  |
| --- | --- |
|  | |
| 功能名稱 | 影像辨識 |
| 測試目標 | 使用者可以利用影像辨識來分辦舌頭顏色 |
| 測試作業 | 1.點選【前往檢測】  2.測試是否能夠辨識舌頭顏色  3.是否有完成辨識送到資料庫 |
| 測試結果 | 成功 |

▼表10-2-8 功能描述- POS機點餐

|  |  |
| --- | --- |
|  | |
| 功能名稱 | POS機點餐 |
| 測試目標 | 確認店員能從POS機幫顧客點餐 |
| 測試作業 | 1.點選【登入】  2.點選【POS機】  3.測試是否能夠用POS機點餐  4.是否有完成點餐 |
| 測試結果 | 成功 |

▼表10-2-9 功能描述- POS機接收訂單

|  |  |
| --- | --- |
|  | |
| 功能名稱 | POS機接收訂單 |
| 測試目標 | 確認店員能從POS機收到其他客人訂單餐點 |
| 測試作業 | 1.點選【POS機】  2.測試是否能夠用POS機收到訂單  3.是否有收到訂單 |
| 測試結果 | 成功 |

▼表10-2-10 功能描述- POS機收到庫存不足

|  |  |
| --- | --- |
|  | |
| 功能名稱 | POS機收到庫存不足 |
| 測試目標 | 讓店員知道庫存不夠,可能需要補貨或跟客人說無法製作餐點 |
| 測試作業 | 1.點選【POS機】  2.測試是否能夠用POS機收到庫存不足  3.是否有收到庫存不足 |
| 測試結果 | 成功 |

▼表10-2-11 功能描述-庫存資料表

|  |  |
| --- | --- |
|  | |
| 功能名稱 | 庫存資料表 |
| 測試目標 | 確認管理者能夠看到庫存資料 |
| 測試作業 | 1.點選【庫存資料】  2.測試是否能夠看庫存資料  3.是否有看到庫存資料表 |
| 測試結果 | 成功 |

▼表10-2-12 功能描述-銷售資料表

|  |  |
| --- | --- |
|  | |
| 功能名稱 | 銷售資料表 |
| 測試目標 | 確認管理者能夠看到銷售資料 |
| 測試作業 | 1.點選【銷售資料】  2.測試是否能夠看銷售資料  3.是否有看到銷售資料 |
| 測試結果 | 成功 |

▼表10-2-13 功能描述-進貨/退貨資料表

|  |  |
| --- | --- |
|  | |
| 功能名稱 | 進貨/退貨資料表 |
| 測試目標 | 確認管理者能夠看到進貨/退貨資料表 |
| 測試作業 | 1.點選【進貨/退貨資料】  2.測試是否能夠看到進貨/退貨資料  3.是否有看到進貨/退貨資料 |
| 測試結果 | 成功 |

第11章　操作手冊

**11-1 系統元件**

▼表11-1-1 系統元件表

|  |  |
| --- | --- |
|  | |
| 元件名稱 | 智慧侍茶師 |
| 費用 | 免費 |
| 網路需求 | Wifi/4G/5G網路 |
| 軟體需求 | Web瀏覽器 |

**11-2 系統下載**

輸入以下網址即可使用: <https://yumingyen.pythonanywhere.com/>

第12章　使用手冊

**12-1 顧客操作**

▼表12-1-1 顧客操作使用說明表

|  |  |  |  |
| --- | --- | --- | --- |
| 註冊畫面 | | | |
| ▲圖12-1-1 註冊畫面 | 開啟歡迎首頁影片，  點進去客製化頁面  有登入以及註冊  點選註冊，  輸入電話號碼、密碼  以及確認密碼和LINE ID後  即可按下下方註冊。 | | |
| 登入畫面 | | | |
| ▲圖12-1-2 登入畫面 | | 開啟網頁首先會有  歡迎首頁影片，  有登入以及註冊，  點選登入輸入電話號碼及密碼後，  即可按下下方登錄按鈕進入。 | |
| 歡迎畫面 | | | |
| ▲圖12-1-3 歡迎畫面 | | | 開啟網頁首先會有  歡迎首頁影片。 |

|  |  |
| --- | --- |
| 客製化問卷畫面 | |
| ▲圖12-1-4客製化問卷畫面 | 看完影片可點選開始填表單，  就會進入客製化問卷畫面，  選擇完畢後點選下一題 則會前往下一題，  可選擇登入會員或不登入。 |
| 舌頭辨識畫面 | |
| ▲圖12-1-5舌頭辨識畫面 | 填選完表單可點選前往檢測，  就會進入舌頭辨識畫面。 |

**12-2 員工操作**

▼表12-2-1 員工操作使用說明表

|  |  |  |  |
| --- | --- | --- | --- |
| 登入首頁 | | | |
| ▲圖12-2-11登入首頁 | | 先輸入帳號、密碼，  即可按下下方登錄按鈕登入。 | |
| 員工首頁 | | | |
| ▲圖12-2-12員工首頁 | | | 登入之後，  會看到員工系統目錄。 |
| POS機畫面 | | | |
| ▲圖12-2-13 POS機畫面 | 登入之後，  看到員工系統目錄，  點選POS系統，  即可進入到POS系統裡做點餐。 | | |

**12-3管理員操作**

▼表12-3-1 管理員操作使用說明表

|  |  |  |  |
| --- | --- | --- | --- |
| 登入首頁 | | | |
| ▲圖12-3-1登入首頁 | | | 先輸入帳號、密碼，  即可按下下方登錄按鈕登入。 |
| 管理員首頁 | | | |
| ▲圖12-3-2管理員首頁 | 登入之後即可點選，  庫存、銷售、進貨、退貨資料。 | | |
| 庫存資料畫面 | | | |
| ▲圖12-3-3庫存資料畫面 | | 登入之後即可點選，  庫存資料做查看。 | |

|  |  |  |
| --- | --- | --- |
| 銷售資料畫面 | | |
| ▲圖12-3-4銷售資料畫面 | 登入之後即可點選，  銷售資料做查看。 | |
| 進/退貨資料畫面 | | |
| ▲圖12-3-5進/退貨資料畫面 | | 登入之後即可點選，  進/退貨資料做查看。 |

第13章　感想

* 11046006張家愷

這次的專題製作讓我學習到了每個人對於分工合作的能力，不僅僅是自己的程式能力要有一定的基礎外，分工合作也顯得格外重要，一開始我也是甚麼程式都沒有一定的水準，後端我連碰都沒碰過，結果也是主打一個邊做邊學的概念，有的時候一定要處理debug，那個一處理下來就是好幾個小時，甚至到天亮也有過，畢竟我們專題也沒有到太難，自己邊摸索邊修改還是可以完成的，整個製作流程下來，雖然非常的麻煩，但是到最後看到自己的系統完成度很高的時候也是蠻有成就感的，畢竟這是大家一起努力製作而成的。專題也蠻像是一種小型社會，大家都是需要各做各的，到了出社會之後，可能這種情況會更加的常見，專題是一個讓我覺得很有挑戰性的一個畢業門檻。

* 11046020顏聿茗

經過這次的專題製作，從一開始的找組員、指導教授到訂定題目，現在已經接近專題製作的尾聲了，這期間我們經歷許多挑戰，也都順利完成了!!

專題製作就是把我們大學這幾年所學的東西串在一起，從前端後端到資料庫，我們都是第一次從頭開始架設，雖然在這期間有許多的困難跟問題困擾我們很久，但現在回頭想，我們從完全沒概念到自己一步一步把問題找出來並解決，是一段很寶貴的經驗。

我主要負責的部分是後端的架設跟各種功能的串接，還有協助組員使用git工具，在串接時發現，其實還是得看懂其他部分在做什麼才能將功能好好的串在一起，所以也花了許多時間在研究大家寫了些什麼，也感謝大家這一年來的努力，讓我們的專題能夠順利完成。

* 11046035徐若蓁

這次的專題製作對我來說是一段非常寶貴的學習旅程，從一開始尋找組員和指導老師，到專題的構思與實作，每一個階段都充滿了挑戰，也收穫了許多成長。專題初期，我主要集中於演算法的研究。然而，在深入演算法之前，我還需先整理出大量資料，將所有藥草的功效及不同的組合方式記錄並分析出來。這些資料的處理量非常龐大，為後續的研究奠定了堅實的基礎。

而後我負責專題中的演算法設計與語音辨識技術，這部分挑戰性極高，涉及大量資料處理和模型優化。每當遇到技術瓶頸，我都會不斷調整演算法進行測試，這過程大大提升了我的專業能力，也讓我更深刻理解了語音辨識技術的應用與複雜性。

這次經歷讓我深刻體會到團隊合作的價值。在實作過程中，儘管遇到了一些困難，但我們透過有效的溝通和互相協助，成功克服了挑戰。組員們的默契和配合，使得專題得以順利推進。團隊合作不僅提高了我們的工作效率，也讓解決問題的過程更加順暢。

每當我們在研究過程中遇到困惑或問題，林俊杰老師和施智文老師總是能給予我們深入的指導與耐心的點撥，讓我們能夠跨越技術和理論的難題。他們的專業見解和豐富經驗，幫助我們在專題的每一個階段都能更好地前行。我們的專題能夠圓滿完成，離不開他們的無私付出與不懈鼓勵，這段成長的旅程充滿了他們的智慧與啟發。這次專題製作，不僅讓我完成了學業的最後一個重要里程碑，也讓我在專業領域內有了更深的體會與成長。

* 11046036洪藝芸

這次的專題讓我獲益良多，從一開始的找組員、尋找老師到專案的構思，接著進入編寫程式和製作網頁的階段，過程中遇到了許多挑戰。這次的經歷讓我深刻體會到團隊合作的重要性，雖然在合作過程中發生了一些意外，但最終大家都能達成共識，互相幫助解決問題，感謝各位組員的努力，讓我們能夠順利的完成。

在這次專題中，我主要負責前端的網站頁面設計和優化以及虛擬主播影片等，網站設計不僅是美觀的外觀編排，更需要從使用者的角度出發，考慮使用者在使用過程中的體驗，確保整體流程流暢且易於使用，這讓我學會了如何運用，並在實際操作中不斷調整和改進。虛擬主播影片介紹，從原本的一個簡單影片介紹，到最後把所有組員的臉都放上去影片中輪流播放，也讓我學到許多不一樣的新東西。

最後，我特別感謝林俊杰老師和施智文老師的悉心指導，在我們遇到問題和困難時，老師總是能迅速提供解釋和建議，為整個專題的順利完成提供了重要的支持，完成了大學的畢業專題製作。

第14章　參考資料

1. <https://www.kocpc.com.tw/archives/516823>
2. <https://bootstrap5.hexschool.com/docs/5.0/layout/grid/>
3. <https://getbootstrap.com/docs/4.0/components/buttons/>
4. <https://www.google.com/search?q=heygen%E5%9C%96%E7%89%87+AI&sca_esv=1c705a24ecbc2b2f&sca_upv=1&rlz=1C1RXQR_zh-TWTW972TW972&sxsrf=ADLYWIIq11-h2Vv8D7LIuuUejS_82xuaHQ%3A1724394339174&ei=YyvIZr-wCuvj1e8PqfXJoQc&ved=0ahUKEwj_7afAvYqIAxXrcfUHHal6MnQQ4dUDCA8&uact=5&oq=heygen%E5%9C%96%E7%89%87+AI&gs_lp=Egxnd3Mtd2l6LXNlcnAiD2hleWdlbuWclueJhyBBSUjlIlCiDFiOH3ABeACQAQCYAZoBoAHVA6oBAzEuM7gBA8gBAPgBAZgCA6ACrgLCAgoQABiwAxjWBBhHwgIIEAAYgAQYogSYAwCIBgGQBgqSBwMxLjKgB8IH&sclient=gws-wiz-serp#fpstate=ive&vld=cid:2462cc0e,vid:cvLHF97BlM8,st:0>
5. <https://github.com/Valdecy/C4.5>
6. <https://roger010620.medium.com/%E6%B1%BA%E7%AD%96%E6%A8%B9-decision-tree-%E5%B8%B8%E8%A6%8B%E7%9A%84%E4%B8%89%E7%A8%AE%E7%AE%97%E6%B3%95-id3-c4-5-cart-54091ca85044>
7. <https://steam.oxxostudio.tw/category/python/ai/opencv.html>
8. <https://github.com/GanonYou/TongueDetection>
9. <https://medium.com/jordanttcdesign/%E5%89%8D%E7%AB%AF%E5%AD%B8%E7%BF%92-css-flex-%E6%AF%92%E8%97%A5%E7%AD%86%E8%A8%98-e229db815bf2>
10. <https://mitblog.pixnet.net/blog/post/44380918>
11. <https://stackoverflow.com/questions/50436596/what-is-add-fieldsets-for-in-useradmin-in-django>
12. <https://www.digitalocean.com/community/tutorials/how-to-install-the-django-web-framework-on-ubuntu-22-04>
13. <https://ithelp.ithome.com.tw/articles/10246331>
14. <https://ithelp.ithome.com.tw/articles/10304538>

附錄

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| 密碼用生日，不宜 | 針對密碼部分，考慮到資安問題，密碼使用生日確實有風險，最後經過修改，設計為給客人自行輸入密碼。 |

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| "智慧"要凸顯 | 我們增加了opencv影像辨識，  辨識舌頭顏色**，**  偵測出來結果**，**  輸出成文字儲存到後端**，**  後端在依照文字去做調整藥材。 |

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| 僅根據簡單問卷難以稱為智慧 | 我們使用CART演算法，結果會跟窮舉一模一樣，但使用決策樹的話可擴充性會變更高，窮舉只是治標不治本的方法而已，前端輸入的結果六個都回答訓練集裡面的資料時，會準確的跑出訓練集的結果藥材出來，還新增了複選的功能，在決策樹判斷的同時，再去判斷複選的症狀強弱去額外新增相對應的藥草。 |