**中 國 文 化 大 學**

**資 訊 工 程 學 系**

**資 訊 系 統 專 題**

**智能衣櫃**

**學 生：黃 懷 萱**

**陳 墨 兮**

**詹 惠 茹**

**指導教授：林 世 崧**

**中 華 民 國 111 年 12 月**

**智能衣櫃**

**專題學生：黃懷萱、陳墨兮、詹惠茹**

**指導教授：林世崧 博士**

**中國文化大學 資訊工程學系**

摘要

因應現代技術，大多事務都傾向於自動化趨勢，同時為了緩解忙碌的早晨，本研究計畫完成一台智能衣櫃。其中技術項目包含衣物辨識，讓使用者放入衣物至檢視口後不用手動輸入項目細節便可存入衣物資訊，完畢後再自動運轉將衣物整理擺放至位置，只需要將衣物放入衣櫃就可以在面板上查看所有的衣物資訊，再者是使用時會自動抓取天氣資訊並自動計算出當時最適合的衣物組合，透過UI介面選取心儀的衣物組合，接著會由衣櫃內部的馬達運轉自動推送到使用者面前，不需要翻箱倒櫃的搜尋衣服，利用機器來記錄並完成所有事物。

**Intelligence-Closet**

**Student： Huang Huai-Hsuan, Chen Mo-Hsi, and xxxxx**

**Advisor： Prof. Lin-xxx**

**Department of Computer Science and Information Engineering**

**Chinese Culture University**

ABSTRACT

**英文摘要**

目 錄

[摘要 I](#_Toc122481798)

[ABSTRACT II](#_Toc122481799)

[表 目 錄 VI](#_Toc122481800)

[圖 目 錄 VII](#_Toc122481801)

[第 1 章 研究動機與目標 1](#_Toc122481802)

[1.1 、研究動機 1](#_Toc122481803)

[1.2 、研究目標 2](#_Toc122481804)

[第 2 章 架構設計與研究項目 5](#_Toc122481805)

[2.1 、系統架構 5](#_Toc122481806)

[2.2 、軟體設計 5](#_Toc122481807)

[2.2.1 、衣物辨識與資料庫 5](#_Toc122481808)

[2.2.2 、天氣資料 6](#_Toc122481809)

[2.2.3 、衣物推薦演算法 6](#_Toc122481810)

[2.3 、硬體設計 6](#_Toc122481811)

[第 3 章 需求研究與文獻探討 7](#_Toc122481812)

[3.1 、衣物辨識 7](#_Toc122481813)

[3.2、天氣資料 9](#_Toc122481814)

[3.3、衣物推薦演算法 12](#_Toc122481815)

[3.4、資料庫架設 15](#_Toc122481816)

[3.5、使用者介面需求 17](#_Toc122481817)

[3.6、硬體需求 20](#_Toc122481818)

[3.7、使用需求 21](#_Toc122481819)

[3.6.1、前端 21](#_Toc122481820)

[3.6.2、後端 22](#_Toc122481821)

[3.8、需求關聯 23](#_Toc122481822)

[第4章、實際執行方案 26](#_Toc122481823)

[4.1、軟體 26](#_Toc122481824)

[4.1.1、衣物辨識 26](#_Toc122481825)

[4.1.2、天氣資料 27](#_Toc122481826)

[4.1.3、衣物推薦演算法 29](#_Toc122481827)

[4.1.4、資料庫架設 30](#_Toc122481828)

[4.1.5、人機介面設計 32](#_Toc122481829)

[4.1.6、軟體架構優化 38](#_Toc122481830)

[4.2、硬體 40](#_Toc122481831)

[4.2.1、事前設計 40](#_Toc122481832)

[4.2.2、實際製作 40](#_Toc122481833)

[4.2.3、修正 40](#_Toc122481834)

[4.3、軟硬體連接 40](#_Toc122481835)

[4.4、流程設計 40](#_Toc122481836)

[第5章、研究成果 41](#_Toc122481837)

[第6章、應用範圍與市場分析 42](#_Toc122481838)

[6.1、市場分析 42](#_Toc122481839)

[6.2、應用範圍及市場定位 42](#_Toc122481840)

[第7章、工作進度與分配 44](#_Toc122481841)

[參考文獻 45](#_Toc122481842)

[附錄 46](#_Toc122481843)

表 目 錄

圖 目 錄

# 研究動機與目標

## 研究動機

不論是學生或上班族，面對緊湊的生活步調，在出門前往往需要花費時間在決定日常生活的衣物穿搭，除了要找尋適合自己當日外出場合的服裝之外，也需要利用手機查詢天氣概況，避免因為穿著不夠保暖，導致受風寒而影響身體健康；又或者穿著太多，身體積熱過多而感到不適。因此，如何在出門前，快速地選定適合自己心情的穿搭，並且兼顧美觀與身體舒適是一個值得研究的議題。

為了解決上述問題，本計畫擬設計並開發一個智能衣櫃，此衣櫃具備一個主控系統、一個螢幕、一組馬達控制系統與攝像頭來完成這項研究計畫。首先，使用者需先將添購的衣物放進衣櫃時，點選螢幕後會將衣物送至衣櫃內部，啟動攝像頭將該衣物進行拍照，並透過服飾影像識別演算法自動識別衣物類別，存入資料庫的同時也會啟動馬達將衣物收納擺放好。當使用者要出門時，系統會抓取當地氣候資訊，並依據自訂的演算法選定穿搭衣物，也可以經由使用者定義。最後，當使用者決定該穿搭衣物後，系統會驅動馬達控制系統，將衣物送到方便使用者拿取的位置，使用者拿取後即可穿搭出門。

## 研究目標

鑒於現代技術所帶來的改變與自動化趨勢，同時為了緩解忙碌的早晨，本研究計畫預計將完成一台智能衣櫃。其中技術項目包含衣物辨識，使使用者將衣物放入衣櫃後由系統自動辨識並歸類該衣物，並將衣物通過軌道收入衣櫃。使用時只需要將衣物放入衣櫃就可以在面板上查看所有的衣物資訊，解決衣服的收納與歸類問題。

同時此衣櫃包含一套衣物推薦演算法，該演算法會結合使用者設定的喜好度、顏色、種類等資訊為使用者推薦數套衣物組合，並且系統會時時自動抓取天氣資訊供使用者進行考量，使用者只需點點手指就可從電子面板取得以上資訊。在選擇好心儀的組合後，衣櫃就會自動將衣服透過軌道輸送至使用者眼前，因此使用者不需再翻箱倒櫃的找衣服或選擇困難，利用機器就可完成所有事物。

因此，本研究計畫的智能衣櫃研究計劃將有以下這些目標：

1. 節省時間
2. 自動天氣預報

使用者無需再手動查詢，可透過已設定好的使用者資訊，可藉由該資訊已設定好的地點欄位，自動從中央氣象局API中獲得氣溫、濕度、雨量、天氣狀況等訊息，讓使用者在使用各功能的同時也可迅速查看天氣目前的天氣狀況，不僅節省時間且能讓沒有習慣看天氣預報的使用者，有更好的生活體驗。

1. 自動推薦衣物

根據獲得的氣溫等天氣資訊與衣物類別及使用者喜好度進行演算，按照分數的高低排序排列當日適配的衣物，例如冬天則系統會推薦長袖、長褲等衣物，反之夏天則推薦短袖、短褲等。

1. 自動拿取衣物

不論是推薦衣物或是自選衣物都可藉由面板選擇並藉由韌體系統控制各項馬達裝置，自動從衣櫃內部運送至使用者面前。

1. 紀錄衣物資訊
2. 了解衣物使用狀況，使用次數或存放時間等。

使用人機介面面板就可以查看衣物資訊，資料庫內部會記錄衣物的樣式、顏色、也會記錄該使用者使用該衣物的次數等，一共存放的時間等。

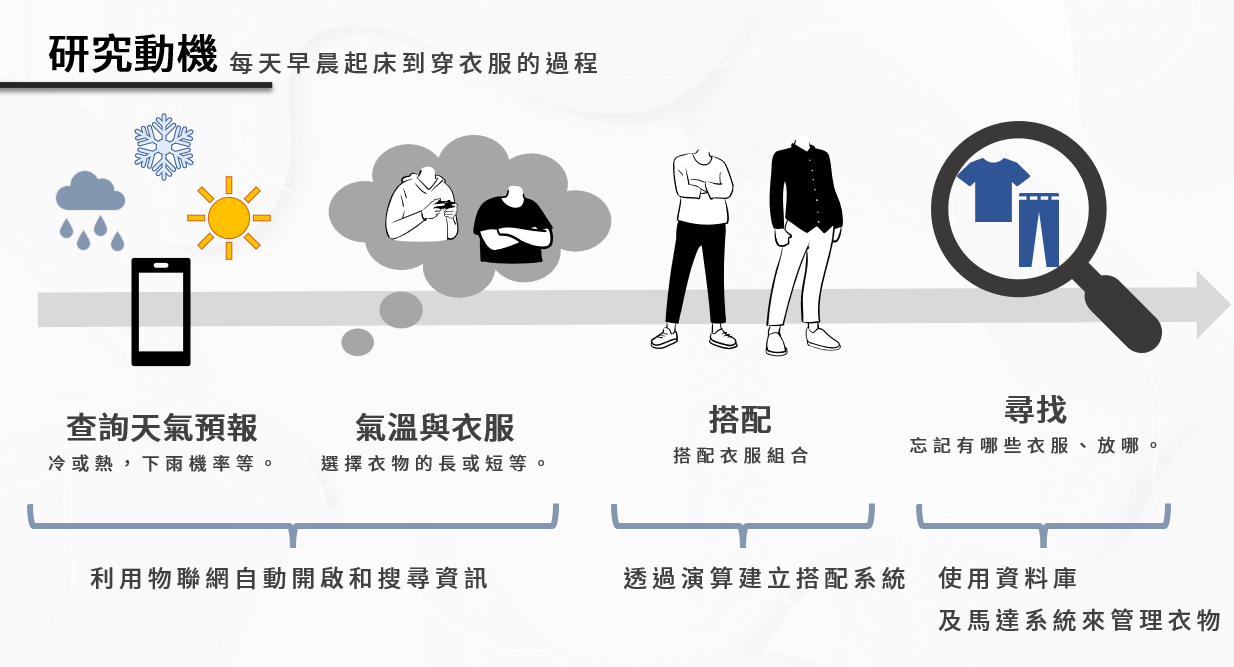
1. 了解自我習慣

會記錄並顯示常存放的衣物樣式、顏色以及使用次數的衣物排行等等，可歸納出常穿搭及喜好的種類，能得出實用性及喜好程度不高衣物，來記錄使用者偏好的衣物樣式，減少購買不必要的衣物節省空間的使用，也可以減少使用者支出的問題，更能了解使用者對於衣物使用的喜好，使用者也可選擇使用次數較少的衣服，來達到使用率平均，減少衣物耗損。

1. 降低衣物耗損

透過使用資料庫查看使用次數，知曉使用者衣物使用頻率、存放時間及存放環境的觀測，對衣物有更好的控管，也可提醒使用者衣物保存的狀況如何，避免過度存放，時間過太久的衣物容易發霉等諸多問題產生，建議汰換。

而本研究計畫最後的研究目標是希望可以達到下圖效果，衣櫃可自動查詢天氣預報從而獲得氣溫狀況等，讓機器能夠更好選擇出該選擇的衣物，並根據先前使用者對衣物的喜好程度與氣溫進行排列組合供使用者作選擇，最後透過韌體與資料庫系統將衣物自動推送至使用者面前。



**圖 1.1研究動機與目標**

而研究內的內容統整成如下圖，該衣櫃是什麼（What）？擁有的功能有哪些，為什麼要建造這個系統（Why）？該如何建立（How）？，與適用的場合與商機可能會在哪裡（Where）？



**圖 1.2研究內容**

# 架構設計與研究項目



## 系統架構

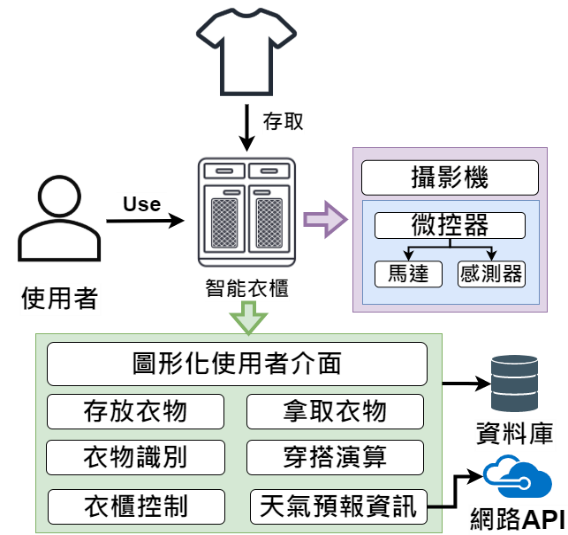


圖 2.1系統架構圖

本計畫的整體架構（如圖2.1），由Web應用框架為概念基礎所建立的圖形化使用者介面負責供使用者使用，這些前端頁面將會發送的請求至後端Python來進行動作，包含資料庫的傳輸、硬體的控制、衣物的辨識、網路API請求以及推薦穿搭衣物演算法的運作等。

## 軟體設計

### 衣物辨識與資料庫

為了方便使用者能夠清晰了解自己所擁有的所有衣服，本研究計畫能在系統中顯示出所有的衣服資料供使用者查閱，資料包含衣服的顏色／類型／樣式（ex. 短袖、無袖、帽T…）以及最重要的「照片」，在傳統的收納中就算使用者記得自己用的衣服，可能也不會記得確切的位置及樣式，因此本研究計畫此功能可以幫助使用者簡單地就可以查詢到自己想的那件衣服，並且將它取出。

為此本研究計畫中，智慧衣櫃的檢視口處放置一台網路攝像頭，除了保存照片外，將會利用人工智慧中的圖樣辨識來快速辨別衣物樣貌，藉此算出衣服的基本資訊（上述的顏色／樣式等），並將所有資訊放入資料庫以完成使用者專屬的衣服資料庫。

### 天氣資料

在平時生活中考量服裝搭配的因素有很多，其中一個較大的因素就是「天氣」狀況，因此本研究會將天氣資訊一同顯示給使用者，不須額外再使用手機查詢。同時，使用者可透過資料庫的紀錄查詢過往的穿衣習慣等，或是透過設置將某些衣物設為收藏，只需設定簡單的搜尋條件與排序就可輕鬆找到想要的衣物，達到資訊整合的效果。

### 衣物推薦演算法

系統在推薦使用者穿衣組合時會根據不同參數進行公式運算，包含「天氣資訊」、「使用者喜好度」等，結合各項資訊後開始進行計算，分數越優則排列順序將會越發靠前，使用者可以透過畫面選擇衣物組合，以此解決現代人選擇困難症的問題。

## 硬體設計

衣櫃最基本的功能便是收納。然而目前衣櫃最容易遇到的問題也正是在收納的部分。傳統的衣櫃（如圖2.2）由於需要同時壘放多層，因此無法容易拿出想要的衣物，有時需要較久以前的衣服時得翻箱倒櫃才能拿到下方的衣服。



圖 .2傳統衣櫃示意圖

本研究計畫透過機械軌道與旋轉輪盤來設計衣櫃存儲空間，使用者只需簡單的將衣服掛上檢視口的位置處衣櫃便會自動透過馬達將衣服存入衣櫃中，要取用時也只需使用點擊螢幕，衣服便會自動被推出，解決找衣服的麻煩。

# 需求研究與文獻探討

## 衣物辨識

功能需求

本研究計畫中，使用者將衣物放入衣櫃後，系統將會自動辨識衣物的樣式與顏色，並且將這些資訊一併存入資料庫中，而針對本研究計畫中的衣物辨識，計畫如表3.1與表3.2的功能性需求與非功能性需求，並對這些功能查找相關功能方法並達成。

表 3.1圖樣辨識功能性需求

|  |  |
| --- | --- |
| Pattern recognition functional requirement | |
| PRFR1 | 辨識衣服 |
|  | 透過攝影機拍下的相片，對衣服的種類與顏色進行辨識，並回傳至系統。 |

表 3.2圖樣辨識非功能性需求

|  |  |
| --- | --- |
| Pattern recognition non-functional requirement | |
| PRNFR1 | 辨識速度 |
|  | 要求在5秒內辨識完成 |
| PRNFR2 | 結果一致 |
|  | 辨識出來的結果與原本的衣服結果一樣。 |
| PRNFR3 | 性能 |
|  | 辨識期間CPU占用率<=50%。 |
| PRNFR4 | 成功率 |
|  | 辨識成功率達八成以上 |
| PRNFR5 | 缺陷率 |
|  | 系統每辨識10000次最多發生1次故障 |

1. 文獻探討

內容

## 天氣資料

1. 功能需求

本研究計畫中，使用者將選用衣物組合時所用的衣物推薦法的氣溫參數，將會依據中央氣象局所提供之網路API來獲取氣溫、濕度、紫外線等多項資訊，因此針對本研究計畫中的抓取網路API功能，計畫如表3.3與表3.4的功能性需求與非功能性需求，並對這些功能查找相關功能方法並達成。

表 3.3網路API功能性需求

|  |  |
| --- | --- |
| Internet API functional requirement | |
| IAFR1 | 抓取天氣資訊 |
|  | 使用爬蟲與中央氣象局提供之API獲取該鄉鎮的天氣資訊，這些資訊包含：   1. T 溫度 2. AT 體感溫度 3. UVI 紫外線指數 4. MaxAT 最高體感溫度 5. MinAT 最低體感溫度 6. Wx 天氣現象 7. WS 風速 8. RH 相對溼度 9. PoP12h 降雨機率 10. Ci舒適度   上述這些將會針對單一鄉鎮（區）獲取該單位的天氣資料，而非一個縣市。 |

表 3.4網路API非功能性需求

|  |  |
| --- | --- |
| Internet API non-functional requirement | |
| IANFR1 | 執行速度 |
|  | 需在規定秒數內執行完畢，若超過須執行替代方案 |
| IANFR2 | 替代方案 |
|  | 以下狀況發生時應有替代方案：   1. 未聯網 2. 執行時間過長 |

1. 文獻探討

交通部中央氣象局為便利民眾共享及應用政府資料、促進及活化政府資料應用、結合民間創意提升政府資料品質及價值、優化政府服務品質，且中央氣象局氣象資料開放平台是由中央氣象局提供的氣象 API 服務，由政府方面提供因此在使用上不需花費金額，介接的方式可以透過官方的使用說明，在網站上申請一個 apiKey 後就能夠依據不同的 API 取得對應的資料了。

適用的對象包含作為個人使用或學術研究用途之資料使用者、一般民眾、輕中量用戶等，都可下載氣象開放資料，該平台將會開放XML、JSON、ZIP、KML、KMZ、CAP 及 GRIB2 等格式供民眾與系統開發業者進行資料加值運用。。

而此中央氣象局開放資料平臺之資料擷取API所使用的是Representational State Transfer API的網頁服務架構工具為 「Swagger」，可藉由輸入金鑰並選定API後輸入指定參數，按下Execute即可使用。



圖 3.1 中央氣象局開放資料平臺API畫面預覽

## 衣物推薦演算法

1. 功能需求

當獲取天氣狀況等資訊後，衣物組合的各項參數與公式結合出的分數將會是推薦依據，進行排列組合後再呈現於使用者面前，因此針對本研究計畫中的衣物推薦演算法，計畫如表3.5與表3.6的功能性需求與非功能性需求，並對這些功能查找相關功能方法並達成。

表 3.5演算法功能性需求

|  |  |
| --- | --- |
| Algorithm functional requirement | |
| AGFR1 | 衣物推薦演算法 |
|  | 演算法根據系統所取得之天氣資訊、使用者偏好度、衣物資訊等進行演算，推薦出數套（至多20套）衣物組合。 |

表 3.6演算法非功能性需求

|  |  |
| --- | --- |
| Algorithm non-functional requirement | |
| AGNFR1 | 執行速度 |
|  | 需在規定秒數內執行完畢 |

1. 文獻探討

關於天氣推薦演算法其一參考根據是由「室內溫度、濕度與風速對 29~60 歲人員感知與舒適度之研究」 （張慧君, 2010）的研究結果，對人體於環境之舒適度最重要的三項參數為「溫度」、「濕度」與「氣流」（意旨風速），其中的權重比如下圖，作為天氣推薦演算法的公式參考項目的其中之一。



圖 3.2 人體與各因子影響比重比較表（張慧君, 2010）

於天氣推薦演算的第二參考由湖北省氣象局於該官方網站上的氣象科普中所發布的「溫度和濕度：影響人體舒適度的最大CP」（湖北省氣象局, 2019-07-30 ）的一文中有提到，溫度至17℃~25℃，濕度落在30%~60%，風速2m/s ~ 4m/s為人體最適合的溫度區間，而超出適合濕度區間段時，會因為身體無法有效排出熱能，容易導致體溫過高等不好影響。

天氣推薦演算法的最後一項參考依據是由近期網路興起的「26度穿衣法則」作為計算公式的參考依據，是台灣於2022年開始流行的一套穿衣法則，由於人體最舒適的平均溫度是26°C，因此會以這個溫度作為基準，透過計算法來搭配出適合當天天氣的穿搭，這個計法的公式就是 26°C－當天的氣溫＝所需的衣服溫度 ，以20°C的天氣為例，26°C減去20°C，當天身上所需的衣服就是6°C，這時候就根據衣服各自的參考溫度來選擇適合的衣服。

表 3.7衣物與衣物代表溫度表

|  |  |
| --- | --- |
| 衣物種類 | 衣物代表溫度 |
| 長T恤／長褲／百褶裙 | 1°C |
| 襯衫／高領 | 2°C |
| 針織衫/毛絨背心 | 3°C |
| 薄毛衣／薄帽T | 3°C |
| 西裝外套 | 4°C |
| 毛帽／圍巾／厚毛衣 | 4°C |
| 風衣 | 5°C |
| 高領厚毛衣／厚帽T | 5°C |
| 發熱衣／輕薄羽絨服 | 6°C |
| 毛大衣 | 7°C |
| 羽絨服 | 9°C |

## 資料庫架設

1. 功能需求

在使用的過程中會有各項資料需要進行儲存的動作，需要有一資料庫架設系統可以為衣物、使用者喜好度、地區等資料進行儲存的動作，因此針對本研究計畫中的資料庫架設上，計畫如表3.8與表3.9的功能性需求與非功能性需求，並對這些功能查找相關功能方法並達成。

表 3.8資料庫與傳輸功能性需求

|  |  |
| --- | --- |
| Database functional requirement | |
| DBFR1 | 儲存資料 |
|  | 儲存系統資料，其中包含：   1. 使用者資料 2. 使用者基本資料 3. 使用者衣物資料 4. 使用者衣物使用資料 5. 天氣資訊 6. 衣櫃資料   註：任何需要保存的資料皆有對應的資料表與欄位 |
| DBFR2 | 修改資料 |
|  | 資料庫須包含以下功能：   1. 新增資料 2. 修改資料 3. 刪除資料 4. 查詢資料   註：每一個table 都要有基本的新增單筆資料的功能 |
| DBFR3 | 查詢資料 |
|  | 資料庫需可供系統查詢資料，同時須因使用者不同而分別提供不同資料，詳情請見（3.5使用者介面） |
| DBFR4 | 架設需求 |
|  | 不同主機需可假設相同資料庫環境，只需安裝本地環境皆可運行，可以執行的環境需包含：   1. MacOS 2. Windows |

表 3.9資料庫與傳輸非功能性需求

|  |  |
| --- | --- |
| Database non-functional requirement | |
| DBNFR1 | 處理速度 |
|  | 90%的請求可以在1秒內完成 |
| DBNFR2 | 資料正規化 |
|  | 正規化並利用視圖串接來優化速度 |
| DBNFR3 | 儲存空間限制 |
|  | 每筆資料將被設定最大儲存大小，超過將被進行壓縮處理 |

1. 文獻探討

本研究計畫預計的資料庫服務為本地端，無須雲端架設，採用的資料庫語言為Microsoft SQL Server，由微軟開發的關聯式資料庫管理系統，SQL為一套標準化的資料庫操作命令，在龐大的數據當中可以幫助我們更方便的搜尋到需要的資料。選在該資料庫語言的其中一項原因包含在學習曲線上也較其他資料庫語言較為容易上手，不需要繁瑣的設定，也因為為本地端資料庫因此在安全性上有嚴格的交易安全與控制，對於資料庫 transaction 的安全控制非常良好。

## 使用者介面需求

1. 功能需求

在使用者體驗層面來說，讓系統擁有圖形化使用介面供使用者點選將賦予較優的人性化使用流程，而該使用者介面預計將可操控所有相關功能，並且是個簡易易懂的人機介面。因此針對本研究計畫中的使用者介面建構上，計畫如表3.10與表3.11的功能性需求與非功能性需求，並對這些功能查找相關功能方法並達成。

表 3.10使用者介面功能性需求

|  |  |
| --- | --- |
| User interface functional requirement | |
| UIFR1 | 人機介面 |
|  | 提供使用者一個合理方便的人機介面，此介面需依照觸控面板方向設計，並彙整產品功能，設計清楚易懂的畫面與說明，需求分支請見（UIFR2~UIFR7），功能說明請見第叁章/三、其他/f.人機介面設計 |
| UIFR2 | 主選單 |
|  | 提供跳至各項功能之選單 |
| UIFR3 | 使用者介面 |
|  | 提供使用者登入、登出、些改使用者資料之介面 |
| UIFR4 | 收納介面 |
|  | 提供使用者使用收納功能，頁面須滿足以下幾點：   * + 1. 分為舊衣收納（已經放入過後拿出來的衣服） 與新衣收納     2. 新衣收納需顯示辨識結果與照片並提供修正辨識結果之選項 |
| （待續） | |

表 3.10使用者介面功能性需求（續）

|  |  |
| --- | --- |
| UIFR5 | 拿取介面（系統推薦） |
|  | 顯示系統推薦的衣物組合之圖片供使用者選擇 |
| UIFR6 | 拿取介面（自行選擇） |
|  | 提供使用者選擇欲拿取之衣物選擇介面，頁面須滿足以下幾點：  1.有大項分類選單  2.有細項分類選單  3.有衣物的照片  4.是否存在衣櫃內的衣物需有顯示區別  5.提供單件or雙件拿取選項 |
| UIFR7 | 衣物設定介面 |
|  | 後期設定衣物資訊（顏色、樣式、喜愛度等）之介面 |
| UIFR8 | 使用說明介面 |
|  | 提供使用者操作說明 |

表 3.11使用者介面非功能性需求

|  |  |
| --- | --- |
| User interface non-functional requirement | |
| UINFR1 | 防呆設計 |
|  | 要求介面應有適合的防呆設計與提示 |
| UINFR | 操縱姓 |
|  | 介面應便於操作、易懂、乾淨，並依使用者體驗進行二次優化 |

1. 文獻探討
2. 後端

由於本研究計畫的核心後端預計皆使用Python作為主要核心語言，其中包含衣物辨識、公式演算與呼叫資料庫等功能，因此在網站的開發與架設同樣將使用Python作為網站架構的後端語言，在Python中製作GUI的應用程序有多種選擇，但若只是想使用HTML及JS的話，若要使用一般的Python後端程序通常需要編寫大量樣板代碼才能與客戶端（Javascript ）到服務器端（Python）。EEl不像Electron或Cefpython那樣成熟，它可能不適合製作類似成熟應用程序，但它非常適合製作實用小程序腳本的GUI。

而本研究計畫採用Python的開源套件EEL，EEl是一個python輕量級的GUI第三方的模組，可以透過本地的伺服器實現web與python之間的溝通，所以整個介面會以網頁方式呈現，也就是可以用HTML 與CSS 去控制呈現的風格，然後用JS 與python 互相呼叫。

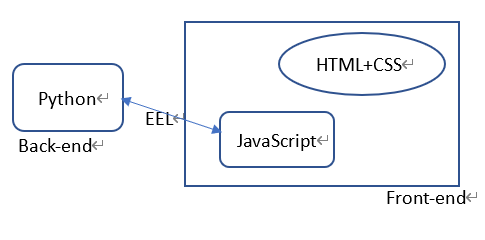


圖 3.3 EEL使用的整體架構方法

該套件的優點有幾點，代碼簡潔、與Python的結合優良、環境與安裝簡單，旨在於結合前端技術開發桌面應用程式，前後端邏輯串接容易。

1. 前端

當後端程式處理完需求之後，網頁是由三大元素所構成：HTML、CSS、JavaScript，最後的結果將回傳給瀏覽器做渲染的動作，在此之前會先將從資料庫介接過來的資料轉成JSON格式，方便前端做使用，為了整體流程與撰寫的流暢度，且要正確的處理使用者的需求。

而本研究計畫的前端框架為AngularJS，相比於其他前端框架，AngularJS的學習曲線較其他Angular與Vue等元件化前端框架較低，且由於是較為老舊的版本，雖然還有在維護但不支援較新的套件，卻也因此長久的歷史所以擁有廣大的開發社群，教程也較為完成，不須花太多的時間在學習該前端框架。

從後端得出的資料為JSON格式，資料的雙向綁定對於前端撰寫與操作功能上有很好的幫助，AngularJS有$scope的協定變數可以做到上述這件事情，使用者只需要將輸入的value綁定到該變數上即可，而前端做任何修改時，該變數雙向綁定同時將會立即做更新的動作。

## 硬體需求

1. 功能需求

當使用者點選人機介面後，為了做到自動系統化因此有一套硬體系統可以替使用者完成應做的所有功能，而該硬體層面相關功能無須使用者在額外進行操作。因此針對本研究計畫中的硬體裝置架構上，計畫如表3.12與表3.13的功能性需求與非功能性需求，並對這些功能查找相關功能方法並達成。

表 3.12硬體功能性需求

|  |  |
| --- | --- |
| Hardware functional requirement | |
| HWFR1 | 衣服存取功能 |
|  | 提供衣服存取功能，該功能需滿足以下條件：   1. 可隨時存放任一一件衣服 2. 可隨時取得任意一件衣服 3. 可同時處理兩件衣服 |
| HWFR2 | 拍照功能 |
|  | 衣櫃需有拍照功能，拍照須滿足以下條件：  1.圖片清晰  2.亮度充足  3.確保拍到整件衣服正面 |
| HWFR3 | 人機介面操控功能 |
|  | 提供觸控螢幕供使用者對衣櫃進行操作，並且可以連接到後端在進行其他如衣服存取功能。 |

表 3.13硬體非功能性需求

|  |  |
| --- | --- |
| Hardware non-functional requirement | |
| HWNFR1 | 耐久度 |
|  | 1.須保證各部分可承重預設的兩倍重量不變形  2.須保證各部分零件在5000次內使用無虞 |
| HWNFR2 | 外觀 |
|  | 外觀需簡潔並包含一定設計感 |
| HWNFR3 | 使用體驗 |
|  | 各部分用作需朝以下目標完成：  1.運作速度快速（ex.拿取需在20秒內完成） |

1. 文獻探討

為了做到輸送衣物的功能，預計採用的馬達共用伺服馬達、減速馬達、步進馬達等，預測最後的承重總重，一件衣物的重量包含上衣或下褲，以較重的男性衣物較為標準，不包含外套與大衣，最輕為短袖的350克至牛仔褲的950克之間，圓盤最後擺放總數件數為12件，合體重量最重會落在10公斤左右，除此之外也需考慮圓盤初始啟動力與轉盤速度等，考慮這些多方面因素，所採用的馬達將會下列幾項：

1. Servo De Alta Velocidad MG995 De 180°、Servo De Alta Velocidad MG996 De 180°，選擇上述兩者伺服馬達型號的原因有幾項：可控制方向（180度以內)、角度偏差在回中時誤差為0度而左右各45°誤差≤ 3、結構材質為金屬齒輪、工作扭距為13G／cm，使用電壓為5V~7.2V。



圖 3.4 MG995伺服馬達構造圖  
(Julpin, Servo De Alta Velocidad MG995 De 180°)

1. Nema 17 Stepper Motor - 17hs6401 - 5mm D Shaft，Nema的型號有非常多種，其中包含2、3、4、6、8代，因應衣物的重量與扭距需求，採用力矩與轉距都最優的6401型號。

一張含有 桌 的圖片

自動產生的描述

圖 3.5 Nema 42步進電機各項參數(ubuy)

1. N20微型金屬減速馬達，全金屬齒輪、體積小、堵轉扭力7Kg/cm，額定力矩1kg/cm，空轉轉速200rpm/min。



圖 3.6 N20微型金屬減速馬達

## 使用需求

### 前端

表 3.14前端功能性需求

|  |  |
| --- | --- |
| Fornt-end functional requirement | |
| FEFR1 | 建立使用者資料 |
|  | 每個使用者皆可建立自己的使用者資料，其中包含：  1.使用者名稱  2.使用者所在地  3.使用者的衣物與使用狀況 |
| FEFR2 | 儲存衣服 |
|  | 使用者可將衣服存入衣櫃，並且只需將衣物掛入即可 |
| FEFR3 | 拿取衣服（推薦） |
|  | 使用者可獲取數套系統推薦之衣物組合，並依此拿取 |
| FEFR4 | 拿取衣服（自選） |
|  | 使用者可自行透過分類選單拿取衣服 |
| FEFR5 | 獲取天氣資訊 |
|  | 使用者可查詢所在地的天氣資訊 |
| FEFR6 | 查詢、修改衣物資訊 |
|  | 使用者可對衣物資料進行以下動作：  1.查詢衣物資料  2.修改衣物資料  3.將衣物設為最愛  4.查詢使用狀況（註：需可進行類似top10排名之功能） |

表 3.15前端非功能性需求

|  |  |
| --- | --- |
| Fornt-end non-functional requirement | |
| FENFR1 | 操作性 |
|  | 各系統（硬體、軟體）皆需便於操作，並朝向後期優化UX體驗為目標 |

### 後端

表 3.16後端功能性需求

|  |  |
| --- | --- |
| Back-end functional requirement | |
| BEFR1 | Model架構 |
|  | 系統之各項功能應滿足Model架構，以對應所有資料庫欄位進行傳輸 |
| BEFR2 | Controller架構 |
|  | 系統之各項功能應滿足Controller架構，利用firmata協定與Arduino進行連接 |
| BEFR3 | View架構 |
|  | 系統之各項功能應滿足View架構，利用EEL 讓JavaScript 與Python溝通 |

表 3.17後端非功能性需求

|  |  |
| --- | --- |
| Back-end non-functional requirement | |
| BENFR1 | Controller架構 |
|  | 所有須運算之物件接使用Controller架構作為介接工具，在優化空間與延伸的擴充性上有更大的空間。 |

## 需求關聯

敘述

表 3.18系統需求V.S系統需求矩陣1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 需求 | DBFR1 | DBFR2 | DBFR3 | DBFR4 | IAFR1 | AGFR1 | PRFN1 | UIFR1 | UIFR2 |
| DBFR1 | **--** | **V** | **V** |  | **V** |  | **V** | **V** |  |
| DBFR2 |  | **--** |  |  |  |  |  | **V** |  |
| DBFR3 |  |  | **--** |  | **V** | **V** |  | **V** |  |
| DBFR4 |  |  |  | **--** |  |  |  |  |  |
| IAFR1 |  |  |  |  | **--** | **V** |  |  |  |
| AGFR1 |  |  |  |  |  | **--** |  | **V** |  |
| PRFN1 |  |  |  |  |  |  | **--** | **V** |  |
| UIFR1 |  |  |  |  |  |  |  | **--** | **V** |
| UIFR2 |  |  |  |  |  |  |  | **V** | **--** |
| UIFR3 |  |  |  |  |  |  |  | **V** |  |
| UIFR4 |  |  |  |  |  |  |  | **V** |  |
| UIFR5 |  |  |  |  |  |  |  | **V** |  |
| UIFR6 |  |  |  |  |  |  |  | **V** |  |
| UIFR7 |  |  |  |  |  |  |  | **V** |  |
| UIFR8 |  |  |  |  |  |  |  | **V** |  |
| HWFR1 |  |  |  |  |  |  |  |  |  |
| HWFR2 |  |  |  |  |  |  | **V** |  |  |
| HWFR3 |  |  |  |  |  |  |  |  |  |

表 3.19系統需求V.S系統需求矩陣2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 需求 | UIFR3 | UIFR4 | UIFR5 | UIFR6 | UIFR7 | UIFR8 | HWFR1 | HWFR2 | HWFR3 |
| DBFR1 | **V** | **V** |  |  |  |  | **V** | **V** |  |
| DBFR2 | **V** | **V** | **V** | **V** | **V** |  |  |  |  |
| DBFR3 | **V** |  |  | **V** | **V** |  |  |  |  |
| DBFR4 |  |  |  |  |  |  |  |  |  |
| IAFR1 |  |  | **V** | **V** |  |  |  |  |  |
| AGFR1 |  |  | **V** |  |  |  |  |  |  |
| PRFN1 |  | **V** |  |  |  |  | **V** |  |  |
| UIFR1 | **V** | **V** | **V** | **V** | **V** | **V** | **V** |  | **V** |
| UIFR2 |  |  |  |  |  |  |  |  | **V** |
| UIFR3 | **--** |  |  |  |  |  |  |  | **V** |
| UIFR4 |  | **--** |  |  |  |  |  |  | **V** |
| UIFR5 |  |  | **--** |  |  |  |  |  | **V** |
| UIFR6 |  |  |  | **--** |  |  |  |  | **V** |
| UIFR7 |  |  |  |  | **--** |  |  |  | **V** |
| UIFR8 |  |  |  |  |  | **--** |  |  | **V** |
| HWFR1 |  |  |  |  |  |  | **--** |  |  |
| HWFR2 |  |  |  |  |  |  | **V** | **--** |  |
| HWFR3 |  |  |  |  |  |  |  |  | **--** |

表 3.20使用需求V.S系統需求 矩陣

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 需求 | FEFR1 | FEFR2 | FEFR3 | FEFR4 | FEFR5 | FEFR6 | BEFR1 | BEFR2 | BEFR3 |
| DBFR1 | V | V |  |  |  |  | V | V | V |
| DBFR2 | V | V |  |  |  | V | V | V | V |
| DBFR3 |  |  | V | V | V | V | V | V | V |
| DBFR4 | V |  |  |  |  |  | V | V | V |
| IAFR1 |  |  | V |  | V |  | V | V | V |
| AGFR1 |  |  | V |  |  |  | V | V | V |
| PRFN1 |  | V |  |  |  |  | V | V | V |
| UIFR1 | V | V | V | V |  | V | V | V | V |
| UIFR2 | V | V | V | V |  | V | V | V | V |
| UIFR3 | V |  |  |  |  |  | V | V | V |
| UIFR4 |  | V |  |  |  |  | V | V | V |
| UIFR5 |  |  | V |  |  |  | V | V | V |
| UIFR6 |  |  |  | V |  |  | V | V | V |
| UIFR7 |  |  |  |  |  | V | V | V | V |
| UIFR8 | V | V | V | V |  |  | V | V | V |
| HWFR1 |  | V | V | V |  |  | V | V | V |
| HWFR2 |  | V |  |  |  |  | V | V | V |
| HWFR3 | V | V | V | V | V | V | V | V | V |

# 實際執行方案

根據需求研究與文獻探討中所提到的各功能項目，將需要完成的步驟分類成一個又一個主題，根據系統架構圖（圖2.1）所示，根據使用者與功能需求可將各部分大致分類為「軟體層」與「硬體層」兩者，兩者功能皆與彼此息息相關。

當使用者點選介面功能後，若選擇存放，系統會先將衣物進行辨識動作，辨識完後的衣服類別可自動存入至資料庫當中；而當使用者選擇拿取功能時，需聯網獲取外部API得到天氣資訊，並根據該天氣資訊計算推薦衣物組合，上述這些步驟皆為軟體層可達到的目標。而最後須由馬達與各項零件配合，來將衣物進行推送進出等動作，而這些為硬體層的功能建造與配置。

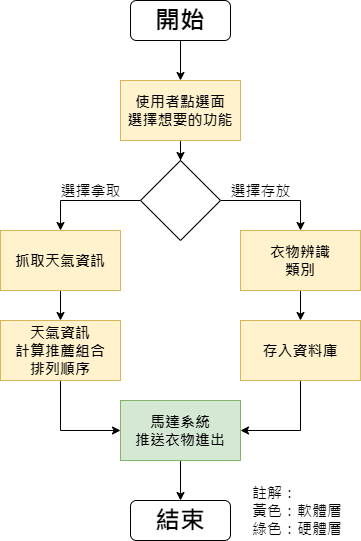


圖 4.1 使用者流程概念圖

## 軟體

根據上述描寫，本研究計畫中軟體層可共分為四大主題，「前端使用者介面」、「衣物辨識」、「資料庫系統」及「天氣推薦衣物演算法」，最底層的支撐架構為資料庫系統，由資料庫提供的各項數據給予其他功能。

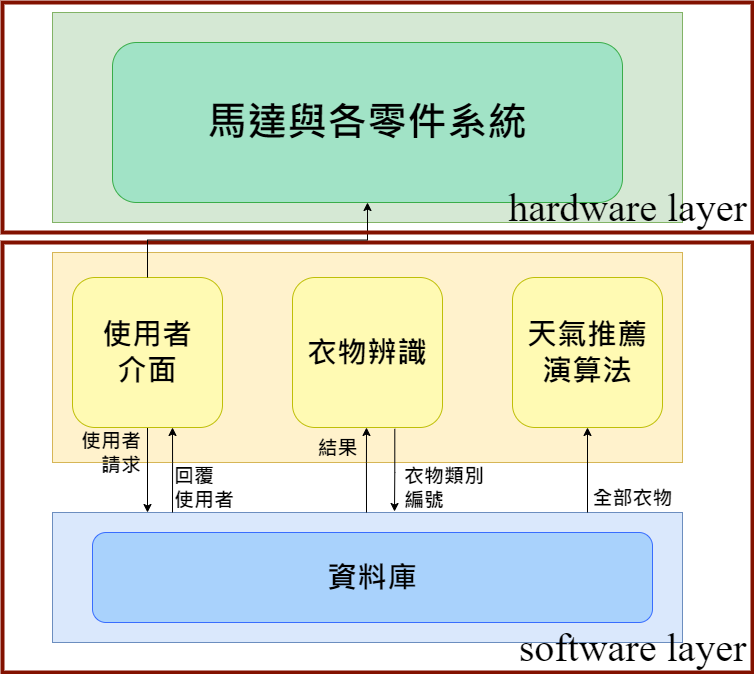


圖 4.2 應用主題各層間關聯圖

### 資料庫架設

表 4.1料庫表格關係

資料庫系統所倚賴的工具為Microsoft SQL Server，版本18，考量資料建表（如表）與關連ER圖（如圖）如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 資料庫 | 說明 | 範例 | 關聯 |
| Category | 衣物的分類：  Ex. 上半身、下半身… | | 用於分類衣物 |
| Sub\_Category | 衣物的子分類  Ex. 長袖、短袖、短褲… | | 可辨識的衣物樣式，用於分類  衣物與天氣適配度用於推薦衣物演算法的參數 |
| Color | 衣物的顏色  Ex. 紅、藍、黑 | | 可辨識的衣物顏色，用於分類 |
| Color\_Graph | 衣物顏色的適配圖  Ex. 白（上） + 白（下） = 5 | | 顏色適配度為衣物推薦演算法的參數 |
| Clothes\_Node | 衣物的詳細資訊  Ex. 顏色、樣式、拿取次數、資料存放位置… | | 負責用於管理衣物的詳細資訊  作為衣物關聯圖形的節點 |
| City | 中央氣象局所提供縣市  Ex. 台北、高雄 | | 負責用於搜索天氣資訊 |
| （待續） | | | |
| **表 4.1料庫表格關係（續）** | | | |
| Station | 中央氣象局所提供站別  Ex. 埔心、楊梅 | | 負責用於搜索天氣資訊 |
| User\_Dashboard | 使用者的資料儲存  EX. 姓名 位置 使用資料 | | 負責儲存該使用者所有資料 |

一張含有 桌 的圖片

自動產生的描述

資料庫ER-Model

### 衣物辨識

補

### 天氣資料

在天氣資訊抓取方面Python會使用Request與BeautifulSoup4套件透過中央氣象局所提供的API進行網頁爬蟲抓取中央氣象局所提供之現存測站。

為求處理速度與解決不定時更新站別所造成的不穩定性，我們事先寫了排程演算法在固定時間內重新抓取站別資料，並將縣市與站別分為兩個部分存入資料庫做正規化方便取用。

隨後，利用中央氣象局所提供API來獲取天氣資訊（所獲得之資訊為JSON格式檔案，包含當日當時所需天氣資訊）。

所使用的API有兩者，分別為：

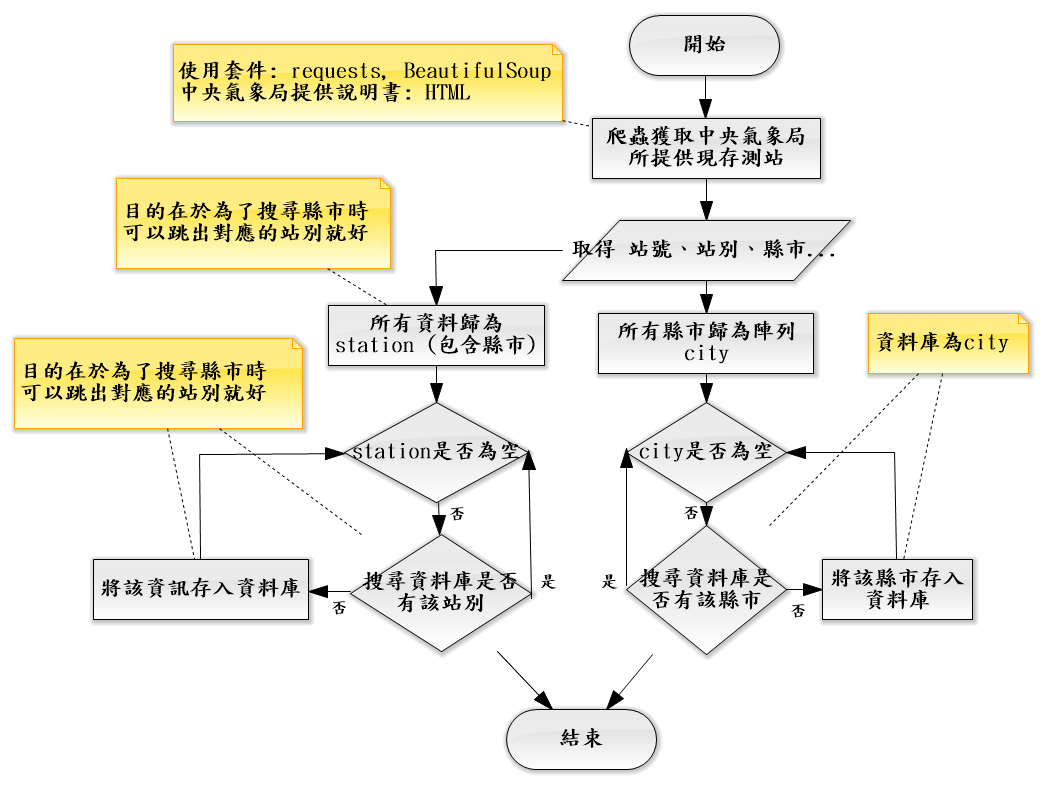
1.「​/v1​/rest​/datastore​/O-A0001-001自動氣象站-氣象觀測資料」

2.「​/v1​/rest​/datastore​/O-A0003-001局屬氣象站-天氣觀測報告」

由於兩者所包含測站各不相同，因此先使用了上述用正規表示法整理之資訊，在確認是否正確擷取後進行雙重認證，認證完成即將資料存儲送至前端介面中。

目前主要截取得天氣資訊如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | 補滿 |  |
|  |  |  |
|  |  |  |

** 存入資料庫流程圖

<實際畫面>

### 衣物推薦演算法

在獲得了天氣資訊後，我們使用了「26度穿衣法則」，將人體最適溫度定在了26度，並與當時溫差得比較，依據此參數來選擇衣物的種類，比如當日氣溫為10度，則溫差為16度，因此需選擇保暖度優先級較高的衣物（如毛衣、長褲等），反之若當日氣溫為30度，則溫差為-4度，則不應選擇保暖度優先級較高的衣物。

除此之外演算法也會透過在資料庫中存儲的使用者個人喜好資料來做為依據調整其標準值。

公式如下：，T為26度標準值，可利用使用者喜好設定值U進行加減，N為目前溫度與每件衣物的天氣適配度（WS）值進行相乘後，加上最後經由使用指設定的衣物搭配準則的分數後，進行平方即可獲得衣物的穿搭準則分數。

補完

### 人機介面設計

**1）、功能列表**

設計UI與UX需求前，首先列出系統可能會需要用的所有功能之列表，再以此為核心建立供使用者使用的前端介面。

為欄位說明 為互動選項 為資訊

|  |  |
| --- | --- |
| 功能分類 | **0主選單（註0-1）** |
| 細項 | 1~7之功能切換選項 |

註0-1：主選單將在畫面左邊以可最小化之方式始終存在

|  |  |  |  |
| --- | --- | --- | --- |
| 功能分類 | 1使用者登入介面（註1-1） | | |
| 細項 | 使用者列表 | 登入 | 管理使用者 |
| 細項分支-1 | 使用者基本資料 | 新增/修正/刪除 |

註1-1：使用者必須登入才可使用2~6之功能

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 功能分類 | **2收納介面** | | | | | |
| 細項 | 收納新衣服 | | 收納已存入過之衣服 | | | |
| 細項分支-1 | 提供辨識結果 | 選擇按鈕 | | | 提供辨識結果 | 提供比對 結果 |
| 細項分支-2 | 提供辨識細項修正選項（註2-1） | 收入 | | 退回 | 提供辨識比對修正選項（註2-2） | |

註2-1：如將辨識結果從（短袖）手動修正為（T-shirt）

註2-2：如比較結果不為同件衣服，可手動尋找對應衣服

|  |  |  |  |
| --- | --- | --- | --- |
| 功能分類 | **3拿取介面（系統推薦）** | | |
| 細項 | 系統推薦選項列表（圖片形式）（註3-1） | 送出按鈕 | 天氣資訊 |

註3-1：點選後將框選該圖樣>可按送出

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能分類 | **4拿取介面（自行選擇）** | | | |
| 細項 | 大項分類選單（註4-1） | 單鍵拿取/整套拿取 | 拿取按鈕 | 詳情按鈕 |
| 細項分支-1 | 細項分類選單（註4-2） | 已選擇之衣物（註4-4） | 衣物詳情 |
| 細項分支-2 | 衣服列表（圖片形式）（註4-3） |

註4-1：大項（收藏/上半身/下半身等）

註4-2：細項（短袖/長袖/Polo衫/帽T等）

註4-3：不可拿取的衣服將被特別標註

註4-4：整套拿取時將顯示每件已選衣物圖片

|  |  |  |  |
| --- | --- | --- | --- |
| 功能分類 | **5衣物設定介面** | | |
| 細項 | 衣服分類選單（註5-1） | 選擇 | 刪除（註5-3） |
| 細項分支-1 | 衣服列表（圖片形式） （註5-2） |
| 細項分支-2 | 該衣物之資訊與更改選項（註5-4） | | |

註5-1：比照{拿取介面}中的大項/細項分類選單

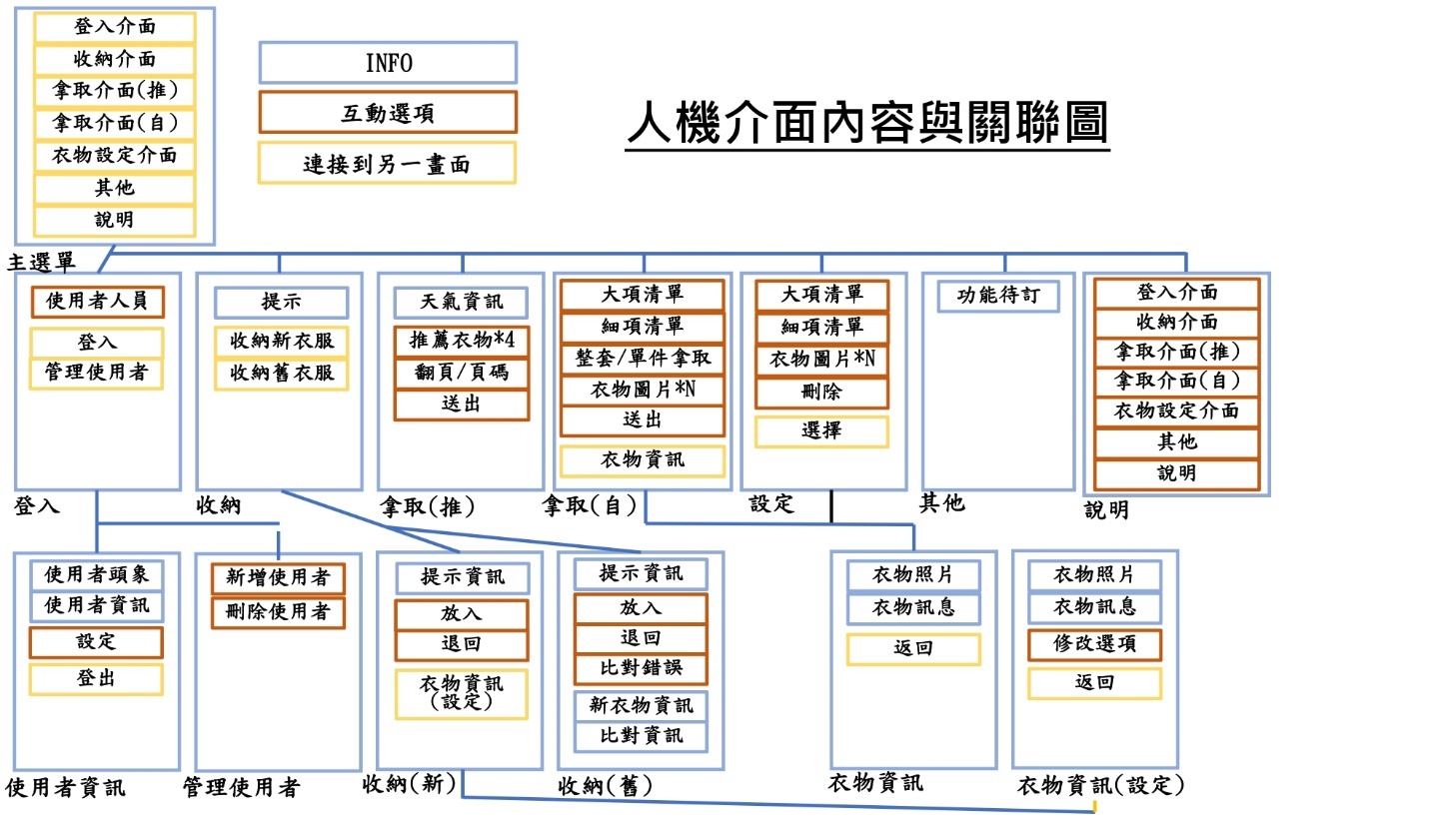
註5-2：點選後將框選該圖樣>可按選擇

註5-3：只有未在衣櫃中的衣物可被刪除資料

註5-3：比照{收納介面}中的{辨識細項修正選項}

|  |  |
| --- | --- |
| 功能分類 | **6使用說明** |
| 細項 | 欲察看之說明選項（1~5介面） |
| 細項分支-1 | 該介面之使用說明 |

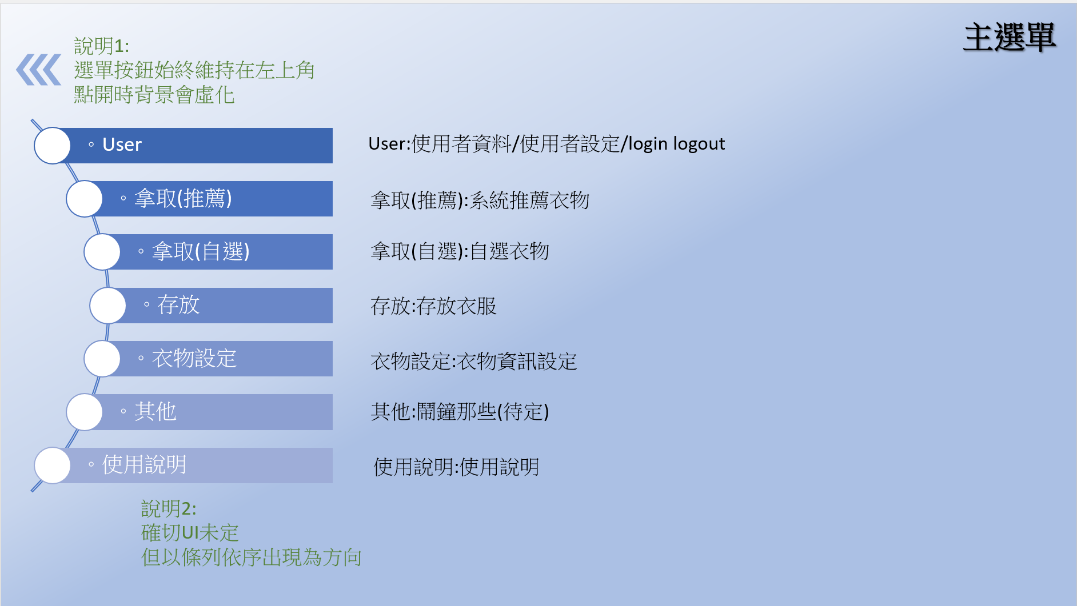
**2）、UI介面關聯圖**

在有了確立了系統功能列表後，為每一頁介面進行整理，並繪製出其該頁面與其他頁面之關聯性，基本的UI介面關聯圖如下。

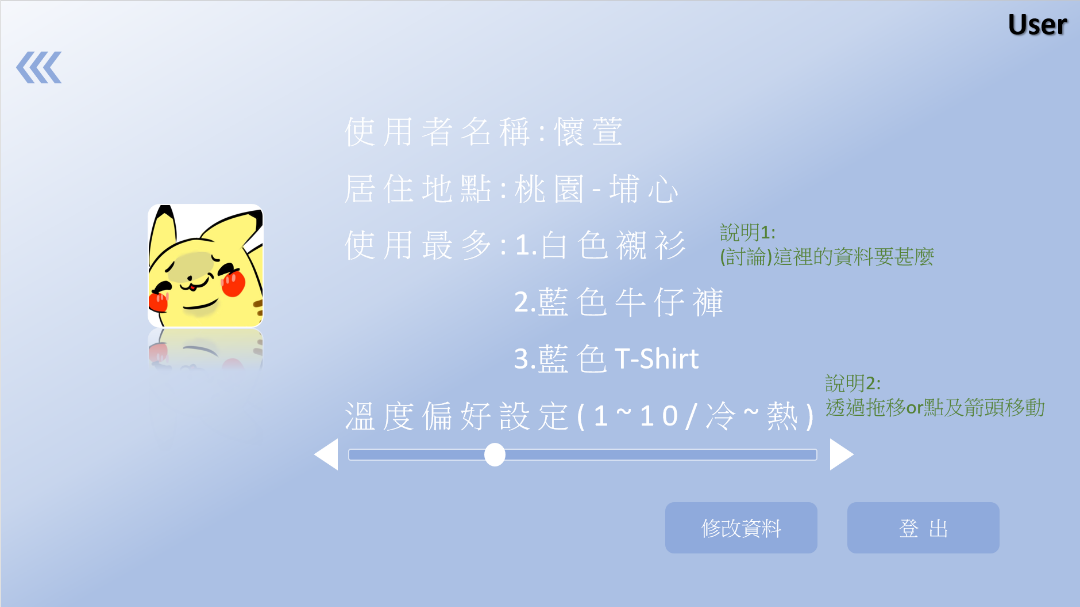
<標號>

**3）、UI模板繪製及心得回饋**

在實際製作UI界面前先利用PowerPoint對各項功能做出UI介面的參考模板，在製作完模板後，透過PowerPoint動畫來模擬使用，並外部人員實際測試並回收使用者心得用以優化最終的UI與UX，最終使用html製作人機互動介面。



UI範例-主選單



UI範例-使用者介面

UI範例-主

UI範例-主

**4）、Web架構建置**

在製作完模板後，透過PowerPoint動畫來模擬使用，由外部人員實際測試並回收使用者心得用以優化最終的UI與UX，最終使用html製作介面。

補圖

### 軟體架構優化

為了將不同功能更直觀的整合及同時方便後期維護，本計畫所建置的 軟體層面皆由Model-View-Controller（MVC）架構做為軟體架構的基礎框架，利用所謂的MVC架構在未來維護或是擴充性上將會有更好的彈性，雖然因MVC架構過於強調職責分離，因此在佔存空間上會有一定 的劣勢，但好處在開發階段會使程式結構更加直覺化，方便進行整合。

以此做為架構核心，可將Model細分為**Domain**、**DAO**、**Service、Controller**等四層，作為動作及EEL的主要控制中樞，並包含所有硬體的控制功能，最後**View**層則由JavaScript與HTML撰寫。

1. **Domain**

會根據資料庫的所有欄位及型態建立一domain層，負責對從資料庫傳輸出來的資料進行內容的存放。

1. **DAO**

對資料庫下搜索的指令語法，原理類似於HQL查詢語法，並對每一份從資料庫傳輸回來的資料裡用domain層包裝好。

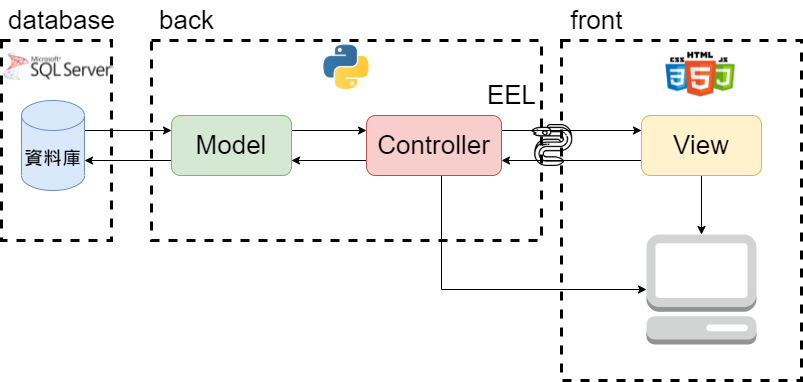
1. **Service**

將DAO包裝好的資料，為了方便傳輸至View中JavaScript，會在此層將所有物件再進行一步分析並將資料轉乘字典型態。

1. **Controller**

可靈活運用所有DAO, Service層的功能。

1. **View**

畫面與顯示的邏輯皆由view所處理，view表單送出的請求皆由Controller接收，再決定給哪個model進行處理，Controller回傳相應的結果至view，並現給使用者。

軟體架構圖

補內容

## 硬體

### 事前設計

### 實際製作

### 修正

## 軟硬體連接

## 流程設計

待寫

# 研究成果

待寫

# 應用範圍與市場分析

## 市場分析

相較於傳統衣櫃與市面上的衣物搭配APP，本專體主要在於整合了兩者的優點並解決其缺點加以優化。

表 6..1市場產品比較

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法 | 收納 | 找衣服 | 天氣資訊 | 穿什麼 |
| 傳統收納 | 慢慢疊慢慢塞 | 翻箱倒櫃 | 拿手機查 | 慢慢試 |
| 手機APP | 使用者自己收 | 使用者自己收 | APP自動查詢 | APP自動搭配 |
| 智慧衣櫃 | 自動收納 | 點點手指 | 自動顯示 | 系統推薦 |

資料來源：自行整理

## 應用範圍及市場定位

智慧衣櫃主要特點在於可以利用資料庫有效的管理衣物和取得資訊，且有互動式介面可供使用者使用，即使是大量衣物的情況下也可以輕鬆控制。

然而其缺點也較為明顯，由於需要加裝機械結構空間上勢必會有所浪費，因此專門分析出兩種此產品的主要的針對客戶群：

（1）、需要有效利用空間的房屋

若家庭空間受限，可挪用一處空間存放智慧衣櫃，利用帳號管理不同成員衣物，不需要每間房都存放體積較大的衣櫃。

（2）、針對擁有大量衣物的工作室或是服裝業相關工作者

利用類似的倉儲系統來做到衣物管理及有UI介面的提供可以更方便的找尋目前有什麼樣的衣物正在存放，也可以看到存入的時間或預覽衣物的樣貌等。

由於目前技術層面的限制，可應用的範圍也因此受限，未來若可將衣櫃馬達與感測技術微小化設計，本計畫的願景是將智能衣櫃也可以推廣至一般家庭居家使用，增加其商業價值。

補滿

# 工作進度與分配

待寫

參考文獻

待寫

附錄