**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。

**關鍵詞**：

指導教授 (簽名)

**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](#_Toc122673557)

[ABSTRACT II](#_Toc122673558)

[表 目 錄 VI](#_Toc122673559)

[圖 目 錄 VII](#_Toc122673560)

[第1章 研究動機與目標 1](#_Toc122673561)

[1.1 研究動機 1](#_Toc122673562)

[1.2 研究目標 1](#_Toc122673563)

[第2章 架構設計與研究項目 4](#_Toc122673564)

[2.1 系統架構 4](#_Toc122673566)

[2.2 軟體設計 5](#_Toc122673567)

[2.2.1 衣物辨識與資料庫 5](#_Toc122673568)

[2.2.2 天氣資料 5](#_Toc122673569)

[2.2.3 衣物推薦演算法 5](#_Toc122673570)

[2.3 硬體設計 6](#_Toc122673571)

[第3章 需求研究與文獻探討 6](#_Toc122673572)

[3.1 衣物辨識 6](#_Toc122673573)

[3.2 天氣資料 8](#_Toc122673574)

[3.3 衣物推薦演算法 11](#_Toc122673575)

[3.4 資料庫架設 14](#_Toc122673576)

[3.5 使用者介面需求 16](#_Toc122673577)

[3.6 硬體需求 20](#_Toc122673578)

[3.7 使用需求 24](#_Toc122673579)

[3.7.1 前端 24](#_Toc122673580)

[3.7.2 後端 25](#_Toc122673581)

[3.8 需求關聯 26](#_Toc122673582)

[第4章 實際執行方案 29](#_Toc122673583)

[4.1 軟體 30](#_Toc122673584)

[4.1.1 資料庫架設 31](#_Toc122673585)

[4.1.2 衣物辨識 34](#_Toc122673586)

[4.1.3 天氣資料 35](#_Toc122673587)

[4.1.4 衣物推薦演算法 38](#_Toc122673588)

[4.1.5 人機介面設計 40](#_Toc122673589)

[4.1.6 軟體架構優化 47](#_Toc122673590)

[4.2 硬體 49](#_Toc122673591)

[4.2.1 事前設計 49](#_Toc122673592)

[4.2.2 實際製作 49](#_Toc122673593)

[4.2.3 修正 49](#_Toc122673594)

[4.3 軟硬體連接 49](#_Toc122673595)

[4.4 流程設計 49](#_Toc122673596)

[第5章 研究成果 50](#_Toc122673597)

[第6章 應用範圍與市場分析 51](#_Toc122673598)

[6.1 市場分析 51](#_Toc122673599)

[6.2 應用範圍及市場定位 51](#_Toc122673600)

[第7章 工作進度與分配 53](#_Toc122673601)

[參考文獻 54](#_Toc122673602)

[附錄 55](#_Toc122673603)

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以以下幾點做為研究方向及目標：

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。

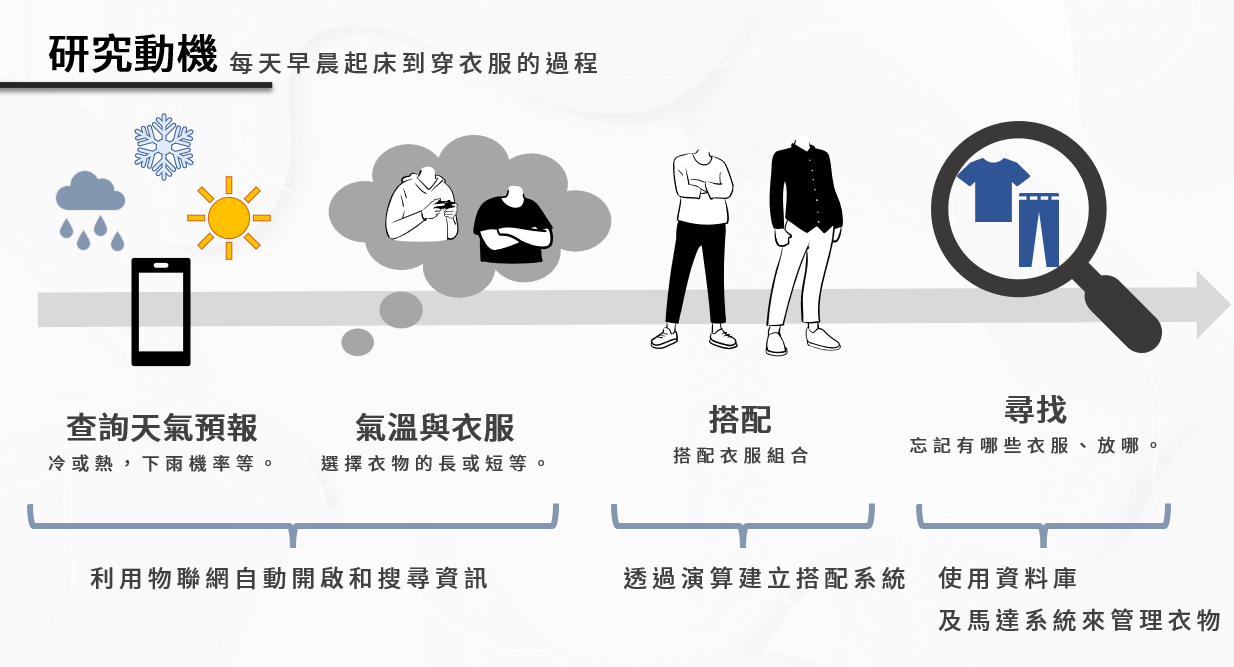
1. 了解自我習慣

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

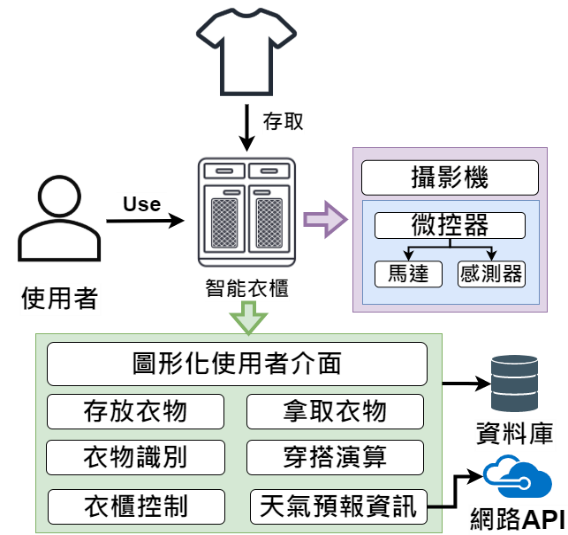


圖 2.1系統架構圖

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 2.2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

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，省去使用者手動輸入的繁瑣過程。

### 功能需求

為了針對衣物辨識的各項功能開發需求，將衣物辨識的學習分為兩種方向，類別與顏色的辨識，兩者辨識是分別進行，最後在統一將結果回傳，目的是不造成機器學習的負擔，除此之外還要兼顧辨識速度的問題，最後的計畫如表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 |
|  | 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%。 |

表 3.2圖樣辨識非功能性需求(續)

|  |  |
| --- | --- |
| 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 |

### 文獻探討

內容

## 天氣資料

### 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 |

### 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

### 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 |

### 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

### 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 |
| 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 4. 查詢資料資料 |

(待續)

|  |  |
| --- | --- |
| 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 |

### 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

### 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 |

(待續)

表 3.10使用者介面功能性需求（續）

|  |  |
| --- | --- |
| 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 |
| 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 |

### 文獻探討

1. 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。

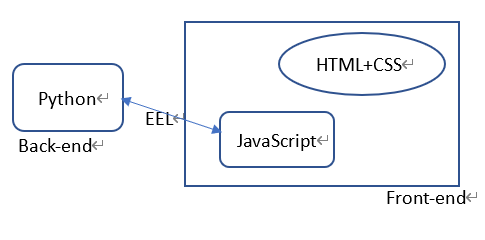


圖 3.3 EEL使用的整體架構方法

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 |

(待續)

表 3.12硬體功能性需求(續)

|  |  |
| --- | --- |
| 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 |

(待續)

表 3.14前端功能性需求(續)

|  |  |
| --- | --- |
| 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 |

(待續)

表 3.16後端功能性需求(續)

|  |  |
| --- | --- |
| 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

一張含有 桌 的圖片

自動產生的描述

圖 3.7系統需求V.S系統需求矩陣

一張含有 桌 的圖片

自動產生的描述

圖 3.8使用需求V.S系統需求矩陣

# 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。

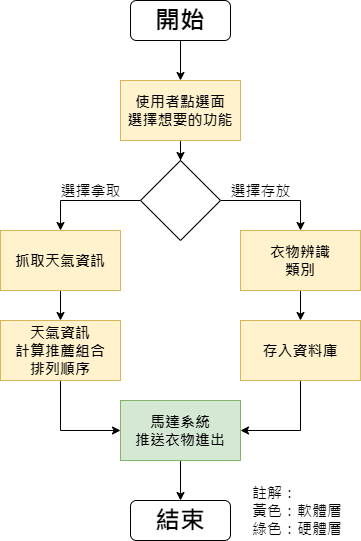


圖 4.1 使用者流程概念圖

## 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輔助其他功能，而人機介面可讓使用者與微控制器進行溝通，從而進行各項動作，衣物辨識與天氣推薦演算法會透過資料庫做為中樞媒介來將資料傳遞給同層級。

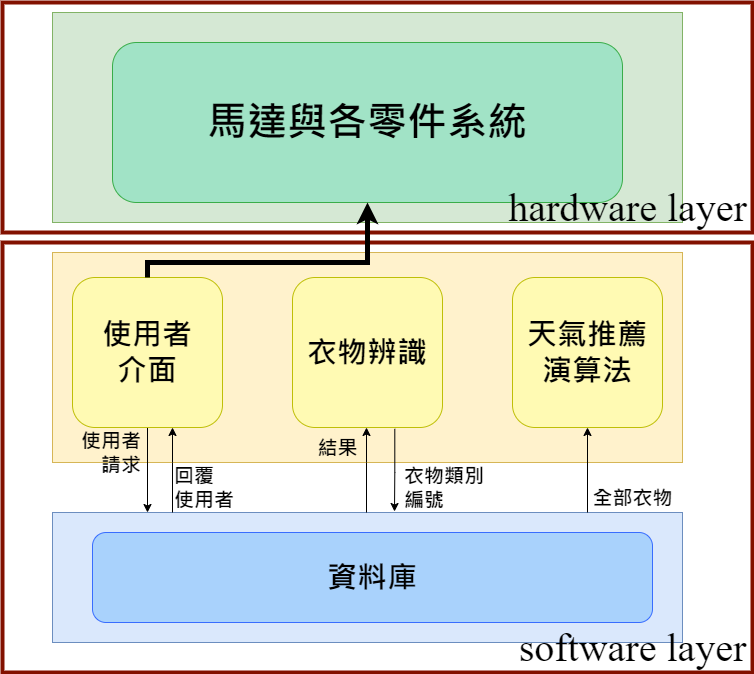


圖 4.2 應用主題各層間關聯圖

### 資料庫架設

資料庫系統所倚賴的工具為Microsoft SQL Server，版本18，建立本地資料庫存放該衣櫃的內部所有資訊，包含的資料庫於表4.1，遵循資料庫正規化規則，正規化的原因是在一堆資料表中很難將完整的狀況表達清楚，最後呈現的ER Diagram（Entity Relationship Model）結果將會非常複雜，且很比對設計的模型是否正確，實作上是否會出現異常問題，因此正規化目的在於將實體描述的資料簡單化，去除冗餘的內容，讓資料能井然有序的且有效率的儲存；且正規化過後的資料庫可提升資料儲存與操作的效率，減少資料的異常問題，讓資料庫更容易維護。

資料庫正規化後有幾項特點：欄位的唯一性，每個欄位只適用儲存一筆資料，能用一個欄位表達清楚的則不使用第二個欄位，可以用兩個欄位表達清楚的則不使用一個；主關鍵欄位，每筆資料都需要有一個主鍵，來區別這些資料的不同，且不可重複；功能關聯性，欄位之間的關聯應該足夠明確，用於區別不同的資料表，關聯則是兩個資料表之間的欄位關聯；欄位獨立性，欄位之間不應該有遞移相依。

本研究計畫的資料庫規畫為，每個資料表皆有主鍵，資料表內的欄位不包含無意義的欄位，每個欄位若包含字串值則將會是單一值，若需重複使用則正規化後引用，並且規劃每個資料表與之欄位的相依性，建立ER-Model清晰可視化。

資料表分類有幾大項，衣物相關、使用者相關及天氣推薦相關等，而衣物相關又可分為衣物的顏色、類別、層級等各項資料存儲，下表(表4.1)將會描述本研究計畫所建立的資料表：

表 4.1資料庫對應說明關係表

|  |  |  |
| --- | --- | --- |
| 資料表 | 說明範例 | 關聯 |
| Category | 衣物的分類如上半身、下半身及大衣等… | 用於分類衣物層級關係，如上半身與＋下半身會必要組合，而外套則是次要等級。 |
| Sub\_Category | 衣物的子分類如長袖、短袖、短褲等… | 可辨識的衣物樣式，且儲存衣物與天氣適配度，用於推薦衣物演算法的參數。 |
| Color | 衣物的顏色如紅、藍、黑 | 可辨識的衣物顏色。 |
| Color\_Graph | 衣物顏色適配圖如白色上衣與白色下褲的總分 | 顏色適配度為衣物推薦演算法的參數。 |
| Clothes\_Node | 衣物的詳細資訊如顏色、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 | 中央氣象局提供之縣市，如台北、高雄…等 | 儲存縣市的API序號，用於查詢天氣。 |
| Village | 中央氣象局提供之鄉鎮地區，可查詢該地天氣。 | 負責用於該地搜索天氣資訊。 |
| User\_Dashboard | 使用者的資料儲存，如姓名、所在位置等…。 | 負責儲存該使用者的資料 |

Entity–relationship Model簡稱ER-Model為實體關聯模型，是用來描述實體與實體之間關聯的模型圖，且會標示關係的基數性（carddinality），所謂的關係所代表的基數性是指實體所能參與關係的案例數，如一對一關係、一對多關係及多對多關係，而本研究計畫共9個資料表，且大多的關係基數性都是一對多關係。

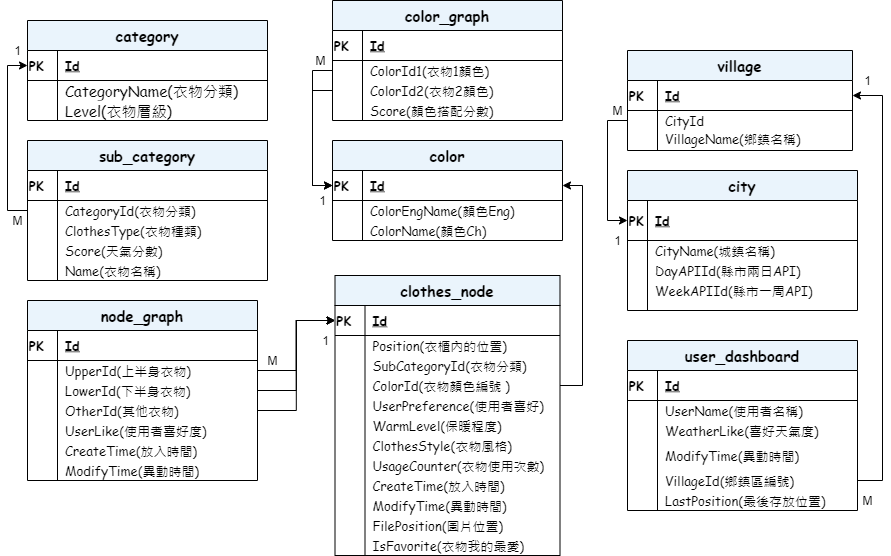


圖 4.3 智慧衣櫃資料庫之ER-Model圖

### 衣物辨識

### 天氣資料

本研究計畫在計算推薦衣物組合時需取得即時的天氣資料，因此需找尋能提供這方面資料的第三方服務平台，以台灣為例，政府的政府資料開放平台中有許多OpenAPI可供民眾自由使用，而抓取天氣資訊的API網站是由交通部中央氣象局的氣象資料開放平台所提供，若要延伸至國際使用還可使用AccuWeather或OpenWeather等服務。

為取得中央氣象局API使用許可，需先註冊為一般會員取得個人「API授權碼」也是俗稱的金鑰，從由Swagger搭建的OpenAPI系統，選擇合適的API確認是否有提供需取得的資料，預計取得的資料有溫度、體感溫度、紫外線指數、最高體感溫度、最低體感溫度、天氣現象、風速、相對溼度、降雨機率、舒適度及天氣描述。

本研究所採用的API為：「/v1/rest/datastore/F-D0047-093」，指定的參數序號為「中央氣象局氣象資料開放平臺－資料擷取使用說明書」所定義，分別為全台灣的不同縣市，包含台北縣、宜蘭縣及基隆縣等，使用畫面如下圖。

一張含有 文字 的圖片

自動產生的描述

圖 4.4 中央氣象局開放資料平臺之資料擷取API測試畫面

在Python抓取API時，將會導入Request與BeautifulSoup4套件，取得的資料格式可根據輸入的網址決定，並且該網址需附帶金鑰與想查詢的參數，之後所有對該服務所發送的請求都需要附帶上這組金鑰，這些服務的供應者就可以透過此把金鑰來辨認你使用他們服務的頻率和情況。本研究計畫所採用的格式為字串型態的JSON格式，方便最後前端框架存取使用，目前截取得天氣資訊如下：

表 4.2擷取天氣資料詳細資料表

|  |  |  |
| --- | --- | --- |
| API內欄位名稱 | 實際代表資料 | 每次抓取時間 |
| T | 溫度 | 每3小時 |
| AT | 體感溫度 | 每3小時 |
| UVI | 紫外線 | 每12小時 |
| MaxAT | 最高體感溫度 | 每12小時 |
| MinAT | 最低體感溫度 | 每12小時 |
| Wx | 天氣現象 | 每3小時 |
| WS | 風速 | 每3小時 |
| RH | 相對溼度 | 每3小時 |
| PoP12h | 降雨機率 | 每3小時 |
| CI | 舒適度 | 每3小時 |
| WeatherDescription | 天氣描述 | 每3小時 |

由表4.2可以看出不同欄位的抓取時間各不相同，可分成每三小時更新與每12小時更新的資料，為了一次抓取這些所有的資料，呼叫的API也分為兩種，並可與表4.2進行對應：

1. 未來1週天氣預報：每12小時更新
2. 未來2天天氣預報：每3小時更新

一縣市的鄉鎮天氣預報可拆分為兩個API做使用，舉例來說，彰化縣的天氣預報API可分為兩種，一種為兩天天氣預報，另一種為一週天氣預報，而兩種所提供的欄位各不相同，詳細資料可察看「中央氣象局氣象資料開放平臺－資料擷取使用說明書」，因此可根據不同欄位的資料時可採用不同的天氣預報API，如下圖：



圖 4.5中央氣象局資料API同縣市不同欄位所需API說明圖

<實際畫面>

### 衣物推薦演算法

衣物推薦演算法所要做到的事情便是從各項資訊中，通過演算取得對人體最適合的穿衣搭配，搭配的準則包含天氣資訊、使用者個人喜好度及衣物搭配準則，除此之外使用者也可對該搭配組合進行評分，最後成為具有個人搭配特色的衣物推薦演算法。

在獲得了天氣資訊後，可藉由天氣中的溫度、濕度與風速等進行天氣推薦演算法的計算，由圖3.2所提供的人體最適環境的溫度、濕度及氣流權重比，調整公式參數比例，採用的權重比率歲數抓取20歲～60歲之間的大眾權重比，同樣參考表3.7之26度穿衣法則參考值，並與圖4.3使用者資料表中的天氣喜好度進行結合。

第一步簡單根據天氣資訊的溫度與26度穿衣法則各衣物的保暖程度所搭配出的基礎公式，其中NT代表的目前的氣溫；CWL代表衣物的保暖程度，保暖程度的範圍訂定在1°C～9°C之間；UWL代表使用者喜好的溫度傾向，範圍落在-5°C～5°C之間，公式推導如下：

查看不同溫度對於衣物推薦分數的變化，更接近於0的數值將會是最佳解。X軸為溫度變化，Y軸為計算的分數結果，整體演算結果如下圖表示，：

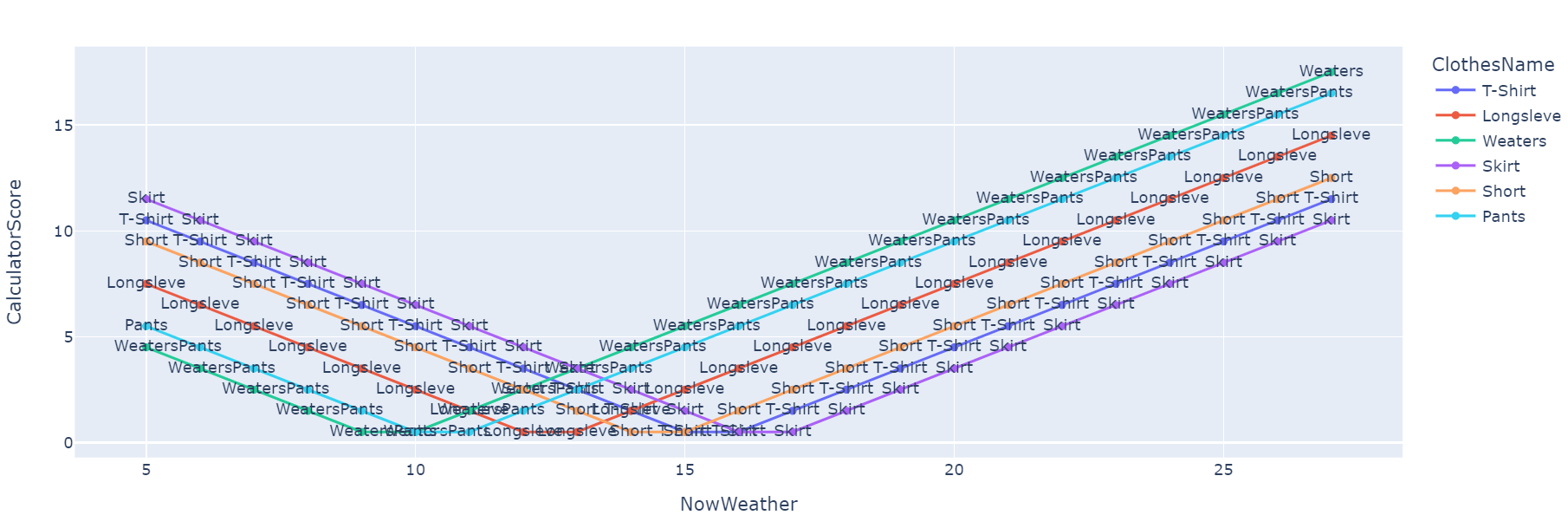


圖 4.6 天氣推薦演算法溫度驗證圖

第二步加入濕度，人體最適合的溼度為60%，當濕度過高的時候會因為人體無法有效的排除熱量，進而導致體溫升高或身體不適的狀況加重，因此最佳狀況為濕度越高的情況下，期望穿著更加清涼的衣物，當濕度降低的時候將做好保暖與保濕的動作，反之當溫度太低時，濕度越高感知的溫度將會越低，需穿著更保暖的衣物。NT為現在偵測溫度，NH為現在偵測濕度，且需要將兩者的單位皆進行百分比轉換才可以進行加減的動作，公式推導如下：

天氣組合的推薦如下圖所示，濕度在50%與60%的時候浮動程度較為激烈，直接查看兩者最高峰的數值，當濕度為50%時，保暖程度較高的衣物的選擇層級較高，當濕度為60%時保暖程度較低（意旨較清涼的衣物）的選擇層級將會提升。

以下為驗證公式的數據可視化圖，此濕度公式分為三種場合驗證，極度寒冷、適中、極度炎熱，而圖4.7、圖4.8及圖4.9 讀Y軸數值為濕度變化量，X軸為為計算結果。

第一種情況是極度寒冷的環境下，在這種時候濕度的關係就相對沒有那麼重要了，而是以保暖度為優先考量，環境為極度寒冷驗證分數結果如下圖：

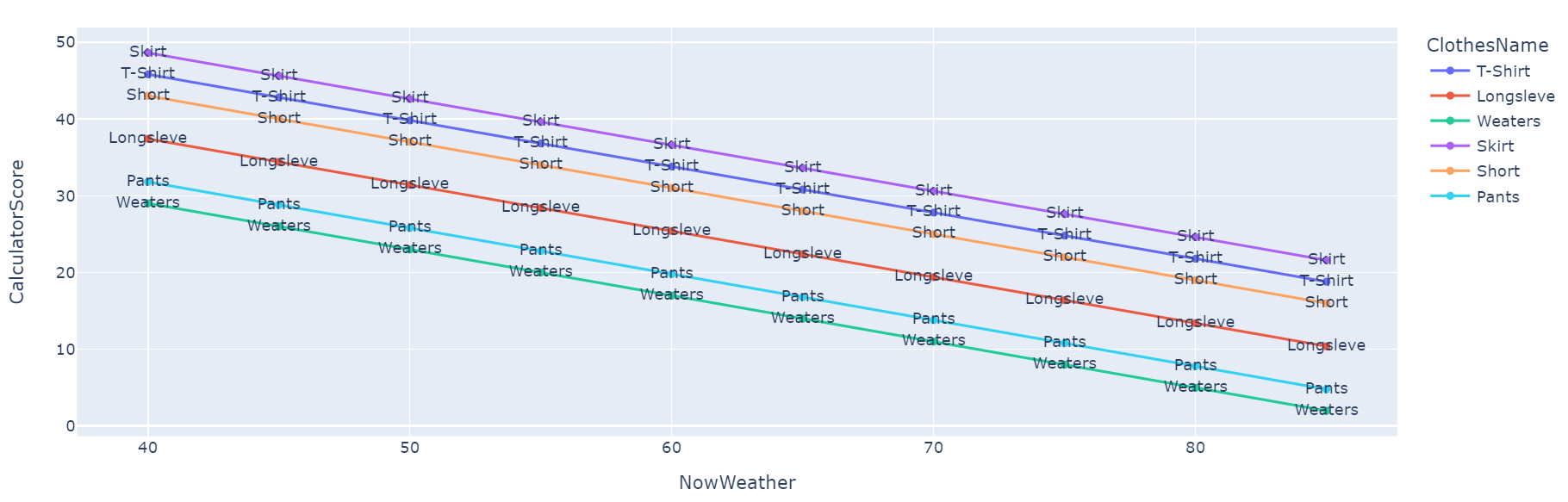


圖 4.7 天氣推薦演算法濕度驗證圖，環境為寒冷溫度約莫8°C

第二種情況是溫度適中，這種狀況下的溼度就相對來說比較重要了，當濕度較高的時候建議穿著更為清涼的衣物，藉此將體內的溫度進行排開，若當濕度較低的情況下為了保濕與保暖的動作，皮膚會教為乾澀，因此可選擇穿著長袖但並不會太厚重的衣物，環境溫度適中的驗證結果如下圖：

一張含有 文字, 地圖, 差異, 多種 的圖片

自動產生的描述

圖 4.8 天氣推薦演算法濕度驗證圖，環境為適中溫度約莫23°C

最後一種情況是溫度炎熱，這種狀況下就如同極度寒冷一樣，散熱的重要性會大於濕度給的權重值，因此在這種情況下會更著重於選擇清涼且易排汗的衣物如短袖、短褲等衣物為核心，環境溫度炎熱的驗證結果如下圖：

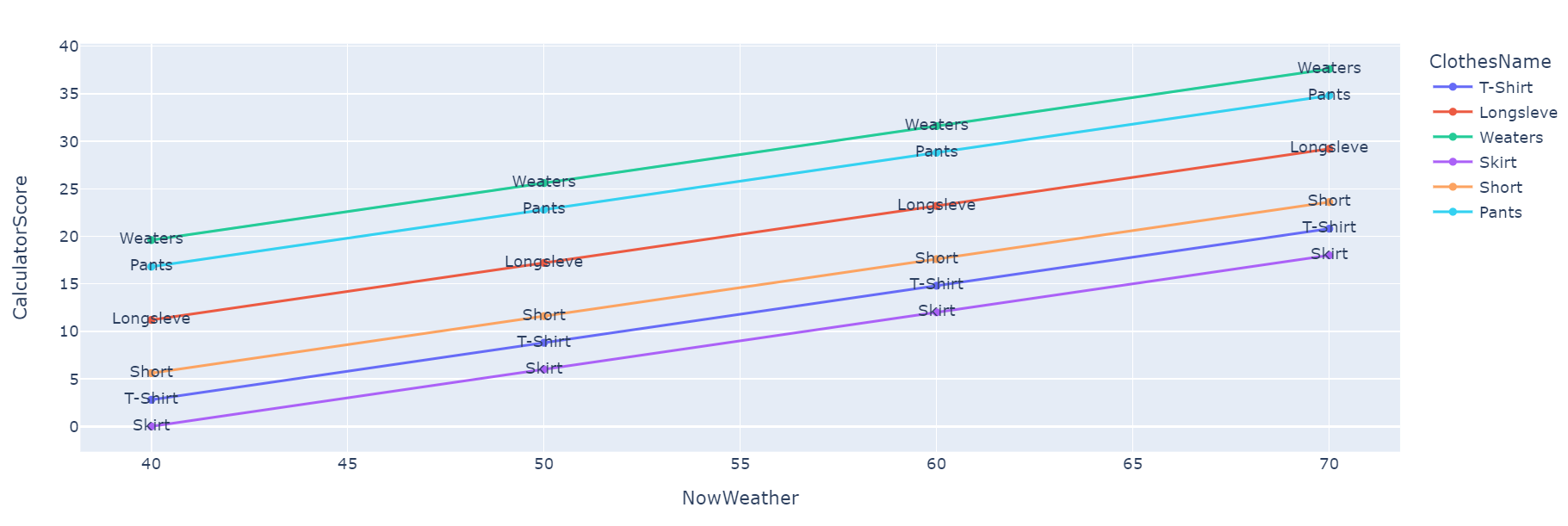


圖 4.8 天氣推薦演算法濕度驗證圖，環境為適中溫度約莫33°C

上述的都是以單件衣物作為預設演算，最後實際還需在將上半身與下半身的保暖程度做相加的動作，除此之外在將使用者喜好度做相加的動作，將全身保暖程度設為TCWL，使用者喜好度設為UL，公式推導如下：

最後預測的結果，當溫度為8度的情況，長褲與毛衣的優先層級較高，而當溫度為26度情況下，則短裙與短袖的推薦優先度層級較其他搭配來的高，可視化結果如下圖：

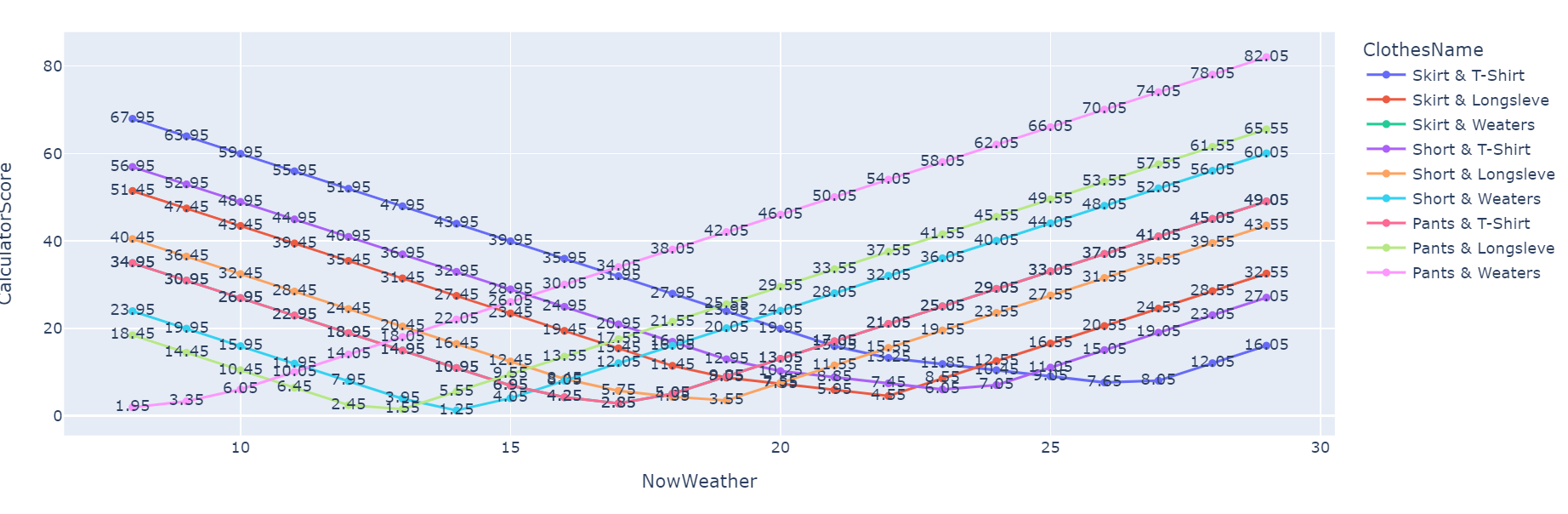


圖 4.10 天氣推薦演算法濕度驗證結果圖

### 人機介面設計

**1）、需求分析**

**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

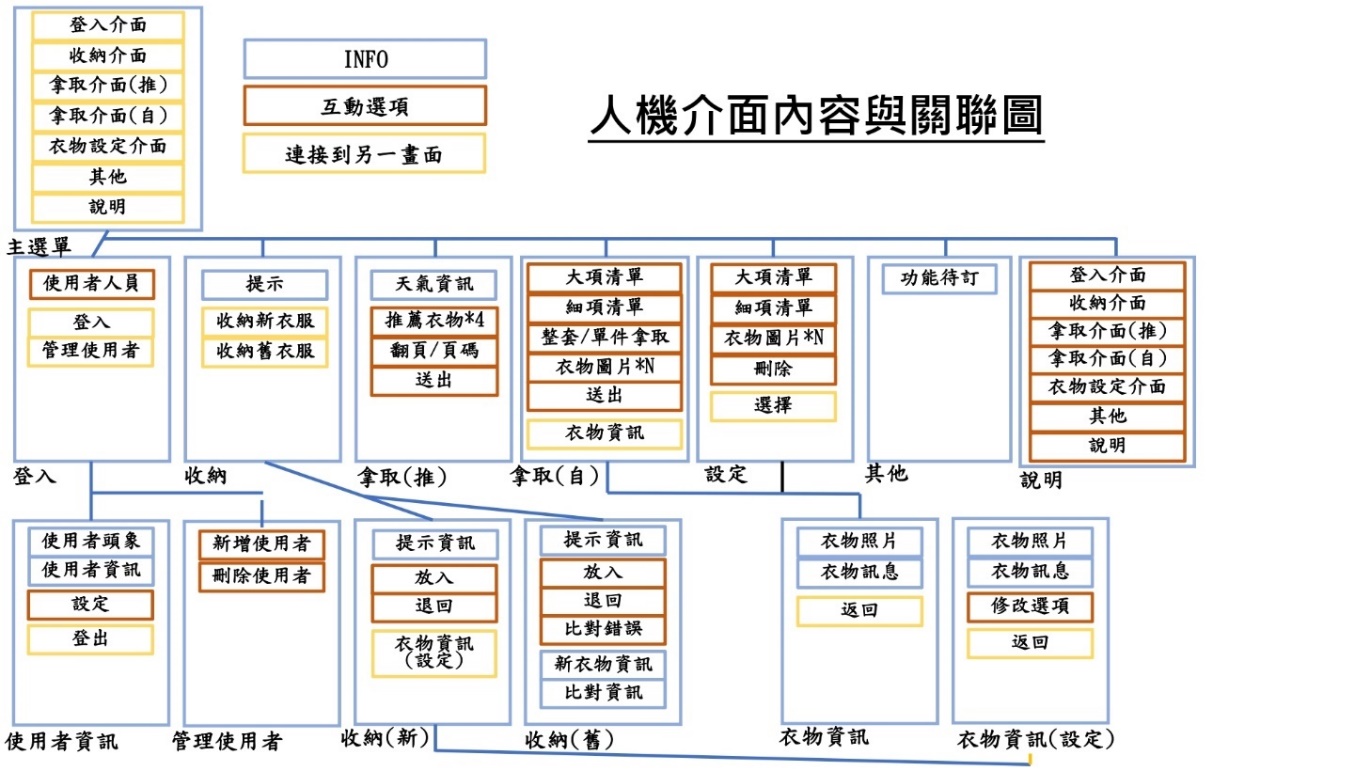
該頁面與其他頁面之關聯性，基本的UI介面關聯圖如下。

圖 4.8人機介面關聯圖

**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。

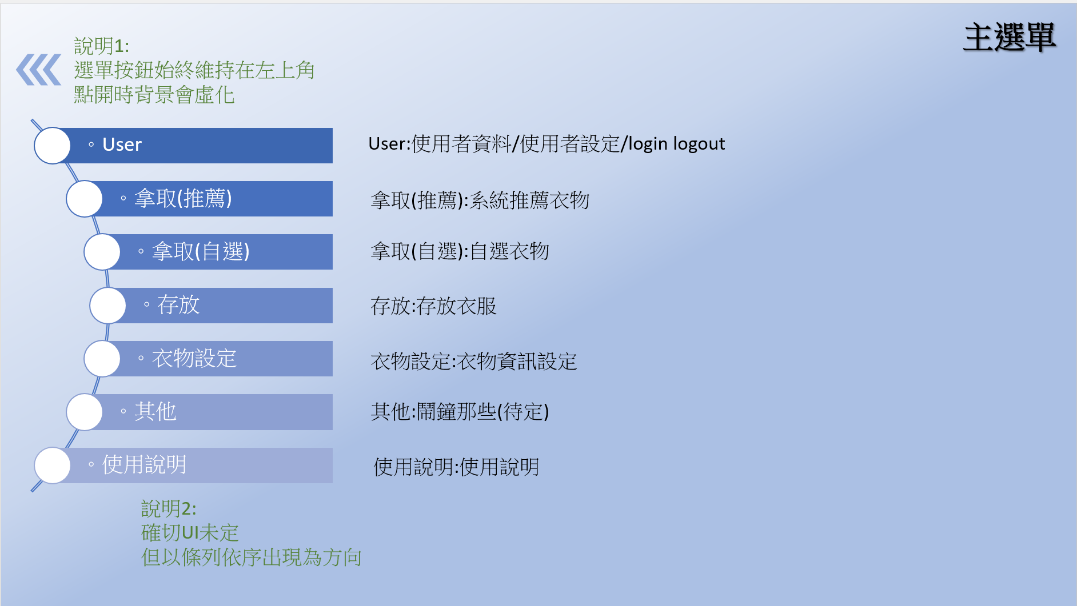


圖 4.9UI範例-主選單

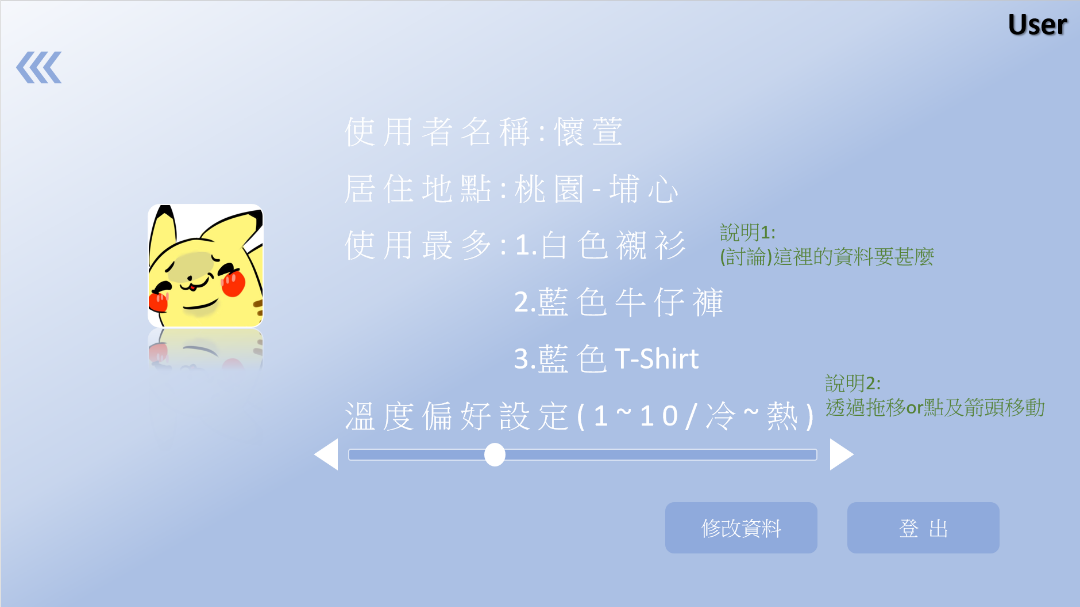


圖 4.10UI範例-使用者介面



圖 4.11UI範例-衣物比對介面

圖 4.12UI範例-衣物辨識結果

圖 4.13UI範例-推件衣服選單

**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。

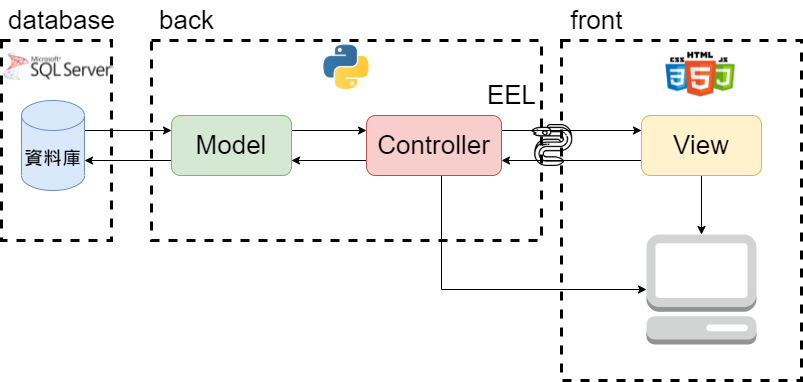


圖 4.17軟體架構圖

## 硬體

未將本專題所製作之功能實際運用，我們將製作一實體以實現智慧衣櫃之功能，包含：

1. 自動收納衣服

使用找只需將衣服放置於出入口即可由衣櫃自動將衣服放入，拿取時亦由衣櫃自動拿出。

1. 結合資料庫選取衣服

透過資料庫保存的存放位置，衣櫃可自行選擇出需要的衣服。

1. 拍照功能

衣櫃內部須待有拍照功能

### 事前設計

### 實際製作

### 修正

## 軟硬體連接

所有硬體控制能夠皆由前端的顯示畫面來操作，使用者只需要點擊按鈕就可以使用硬體設備，包含所有的馬達控制流程，當硬體的所有操作都結束之後再回傳給使用者結果，完成一次的操作流程。

### 連接設計

後端的核心語言為Python，操作所有機械流程的微控制器單元為Arduino，前端控制語言為JavaScript，理想狀態為當使用者透過HTML所渲染出來的畫面，使用者可以透過按鈕事件（JavaScript），透過Controller傳遞請求（Python），當控制器單元（Arduino）接收成功並完成該請求時，再回傳結束訊息，流程如下圖：

補圖

### 連接方法

對於沒有Wi-Fi的微控制器版，需使用USB連接主控裝置（電腦），並利用序列通訊的方法連接Python與Arduino，Python的序列埠通訊套件是pySerial，該套件提供初始化序列埠（設定統一序列埠即可連接）、傳送和接收序列數據的指令（原始資料是byte格式）。

序列埠的設定取決於操作系統的設備名稱，例如再Windows上就會稱為COM3、COM4或COM5等，而Linux上會稱為dev/ttyUSB0，除了序列埠設定之外還需要給予波特率的參數，包含9600、19200或38400等等，以上也皆取決於平台而有所不同，本研究計畫採用的設備名稱微COM4，9600波特率。

### 實際操作方法

為了建立與Arduino連接的方法，Python需先載入serial套件與time套件，由於喚醒Arudino控制器的時間約莫1秒～2秒不等，time套件可以做為緩衝使用。

填寫完需連接的設備名稱與序列埠參數後即可與Arduino進行溝通，而Arduino則需要是等待接收的狀態，待Python傳送指令後執行，成功執行的畫面如下圖：

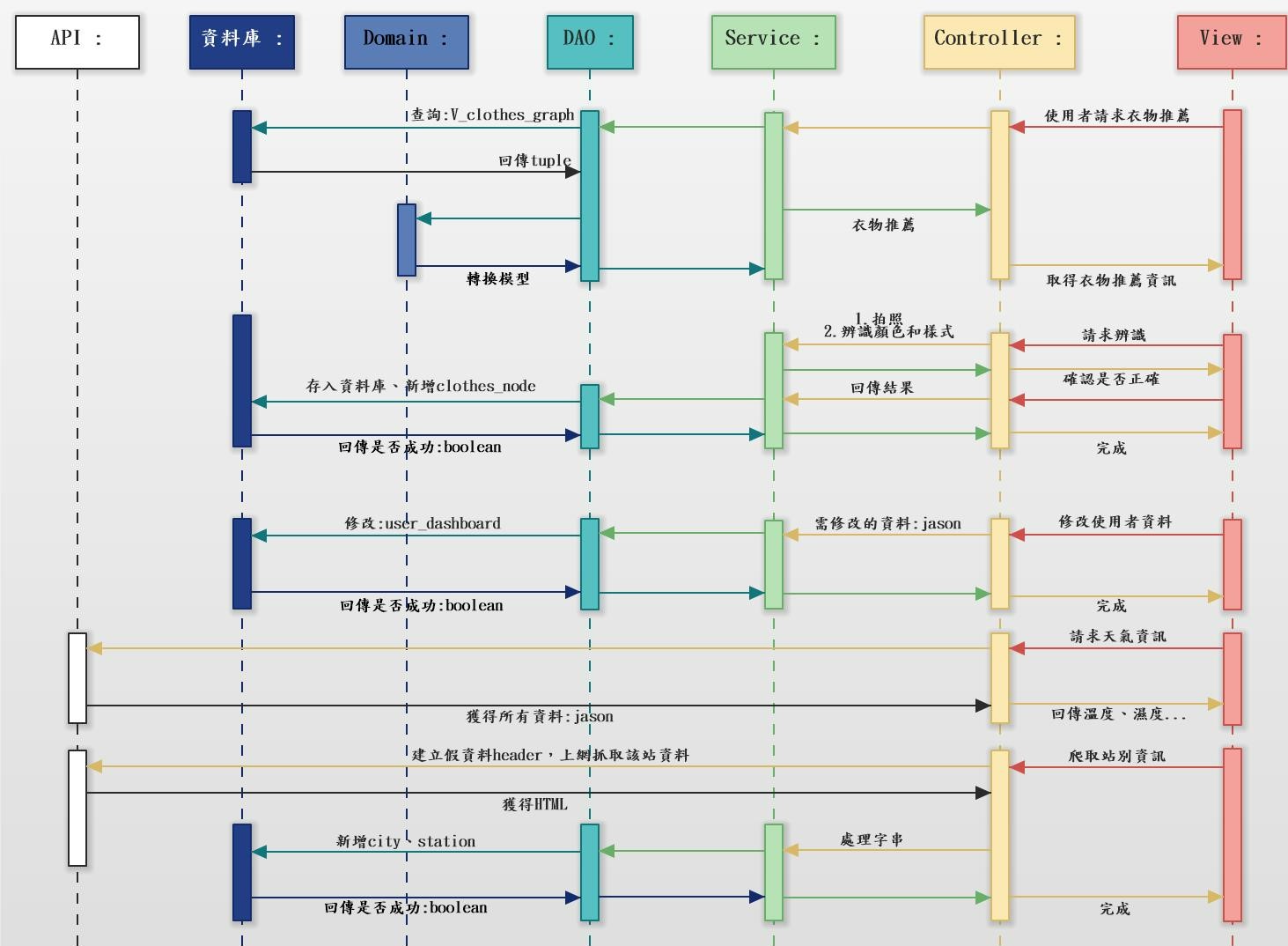
一張含有 文字 的圖片

自動產生的描述

圖 4.18 pySerial執行成功擷取

## 流程設計

回過來看系統架構圖（如圖2.1），目前已建置的系統與功能包含了：資料庫、衣物辨識、網路API、衣物推薦演算法、圖形化使用者介面及實體架構。



# 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 6.0.1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。

# 工作進度與分配

待寫

參考文獻

附錄