**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

中文摘要

因應現代技術的自動化，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](#_Toc122381647)

[ABSTRACT II](#_Toc122381648)

[表 目 錄 VI](#_Toc122381649)

[圖 目 錄 VII](#_Toc122381650)

[第 1 章 研究動機與目標 1](#_Toc122381651)

[1.1、研究動機 1](#_Toc122381652)

[1.2、研究目標 1](#_Toc122381653)

[第 2 章 架構設計與研究項目 4](#_Toc122381654)

[2.1、系統架構 4](#_Toc122381655)

[2.2、軟體設計 4](#_Toc122381656)

[2.2.1、衣物辨識與資料庫 4](#_Toc122381657)

[2.2.2、天氣資料 5](#_Toc122381658)

[2.2.3、衣物推薦演算法 5](#_Toc122381659)

[2.3、硬體設計 5](#_Toc122381660)

[第 3 章 需求研究與文獻探討 6](#_Toc122381661)

[3.1、衣物辨識 6](#_Toc122381662)

[3.2、天氣資料 8](#_Toc122381663)

[3.3、衣物推薦演算法 10](#_Toc122381664)

[3.4、資料庫架設 11](#_Toc122381665)

[3.5、使用者介面需求 13](#_Toc122381666)

[3.6、硬體需求 15](#_Toc122381667)

[3.7、使用需求 16](#_Toc122381668)

[3.6.1、前端 16](#_Toc122381669)

[3.6.2、後端 17](#_Toc122381670)

[3.8、需求關聯 18](#_Toc122381671)

[第4章、實際執行方案 21](#_Toc122381672)

[4.1、軟體 21](#_Toc122381673)

[4.1.1、衣物辨識 21](#_Toc122381674)

[4.1.2、天氣資料 22](#_Toc122381675)

[4.1.3、衣物推薦演算法 24](#_Toc122381676)

[4.1.4、資料庫架設 25](#_Toc122381677)

[4.1.5、人機介面設計 27](#_Toc122381678)

[4.1.6、軟體架構優化 33](#_Toc122381679)

[4.2、硬體 35](#_Toc122381680)

[4.2.1、事前設計 35](#_Toc122381681)

[4.2.2、實際製作 35](#_Toc122381682)

[4.2.3、修正 35](#_Toc122381683)

[4.3、軟硬體連接 35](#_Toc122381684)

[4.4、流程設計 35](#_Toc122381685)

[第5章、研究成果 36](#_Toc122381686)

[第6章、應用範圍與市場分析 37](#_Toc122381687)

[6.1、市場分析 37](#_Toc122381688)

[6.2、應用範圍及市場定位 37](#_Toc122381689)

[第7章、工作進度與分配 39](#_Toc122381690)

[參考文獻 40](#_Toc122381691)

[附錄 41](#_Toc122381692)

# 表 目 錄

# 圖 目 錄

# 研究動機與目標

## 1.1、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。

(下面這些講的是AIOT，跟專題無關)

隨著現代科技逐年的發展進步，醫療裝置、家庭、建築與工業自動化的IOT裝置每年數量呈爆炸性成長，而在這其中人工智慧能夠使這些物聯網更好地發揮作用，收集資料並從中分析數據並學習，做出及時的決策並採取行動，也可以注意到人們可能忽視的小細節，來降低人工成本。

為了更好的將AIOT應用到生活上，大數據分析漸漸地成為了主流，在收集大量資料與數據後來預測及分析使用者的下一步或是最佳解，讓人們可以更專心地在某一件事物上，而不用顧慮太多，皆由機器來幫忙注意或思考，在各個領域與行業中，AI大數據分析已不再是抽象的事物，漸漸地各項領域都朝新型態的數位技術出發，利用科技來協助我們的生活必給予幫助。

## 1.2、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：

一、節省時間

(1) 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，在使用各功能的同時也可查看天氣資料，減少查詢的時間。

(2) 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。

(3) 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。

二、紀錄衣物資訊

(1) 了解衣物使用狀況，使用次數或存放時間等。

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。

(2) 了解自我習慣

會記錄並顯示長存放的衣物樣式、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。

而本研究計畫最後的研究目標是希望可以達到下圖效果，查詢天氣預報獲得冷、熱等天氣資訊後，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。

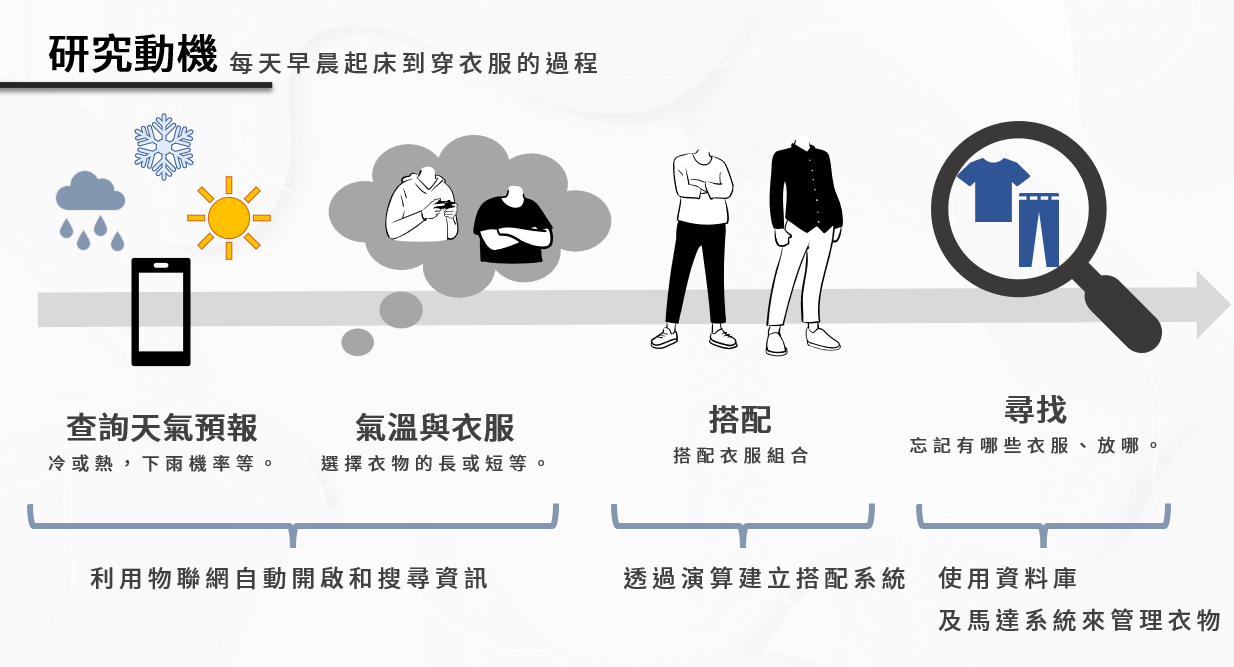


圖 1.1研究動機與目標

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)？

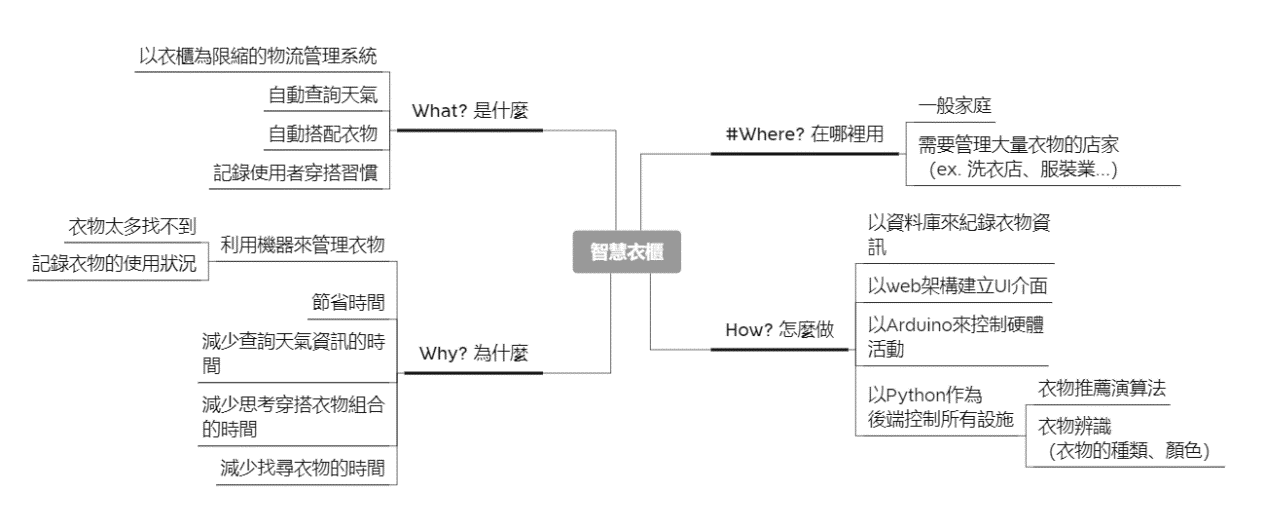


圖 1.2研究內容

# 架構設計與研究項目

## 2.1、系統架構

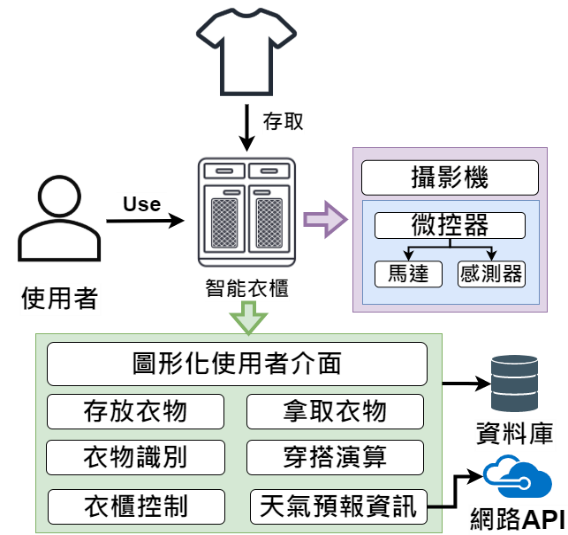


圖 2.1系統架構圖

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的請求以及推薦穿搭衣物演算法的運作等。

## 2.2、軟體設計

### 2.2.1、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(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的就可以查詢到自己想的那件衣服。

為此，研究計畫中，智慧衣櫃的檢視口處放置攝像機，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。

### 2.2.2、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」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。

### 2.2.3、衣物推薦演算法

系統在推薦使用者穿衣組合時會結合使用者所設定的衣服喜好度、穿衣習慣與天氣資訊等數據進行公式計算，透過演算法會為使用者量身推薦四套衣物組合，以此解決現代人選擇困難症的問題。

## 2.3、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3)由於需要同時壘放多層，因此無法容易拿出想要的衣服，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。

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。

圖 2.2傳統衣櫃示意圖

# 需求研究與文獻探討

## 3.1、衣物辨識

引文

表 3.1圖樣辨識功能性需求

|  |  |
| --- | --- |
| Pattern recognition functional requirement | |
| PRFN1 | 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，並將數值分類至相近的預設選項作為結果，最終回傳至系統 |

表 3.2圖樣辨識非功能性需求

|  |  |
| --- | --- |
| Pattern recognition non-functional requirement | |
| PRNFN1 | 辨識速度 |
|  | 要求在5秒內辨識完成 |
| PRNFN2 | 結果一致 |
|  | 辨識出來的結果與原本的衣服結果一樣。 |
| PRNFN3 | 性能 |
|  | 辨識期間CPU占用率<=50%。 |
| PRNFN4 | 成功率 |
|  | 辨識成功率達八成以上 |
| PRNFN5 | 缺陷率 |
|  | 系統每辨識10000次最多發生1次故障 |

內容

## 3.2、天氣資料

引文

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獲取所有站別的天氣資訊並送入資料庫，資料包含:  1.溫度  2.濕度  3.最高溫  4.最低溫  5.最高溫時間段  6.最低溫時間段 |

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 |

內容

## 3.3、衣物推薦演算法

引文

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(4~8)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 |

內容

表 3.7資料庫與傳輸功能性需求

## 3.4、資料庫架設

|  |  |
| --- | --- |
| Database functional requirement | |
| DBFR1 | 儲存資料 |
|  | 儲存系統資料，其中包含:  1.使用者資料  1.1使用者基本資料  1.2使用者衣物資料  1.3使用者衣物使用資料  2.天氣資訊  3.衣櫃資料  \*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  \*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(2.5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 3.8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 |

內容

## 3.5、使用者介面需求

引文

表 3.9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)，功能說明請見企畫書(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  3.舊衣收納需顯示當前照片與比對結果之照片與資訊，並可供使用者手動替換比對結果 |
| (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 |

## 3.6、硬體需求

表 3.12硬體功能性需求

|  |  |
| --- | --- |
| Hardware functional requirement | |
| HWFR1 | 衣服存取功能 |
|  | 提供衣服存取共能，該功能需滿足以下條件:  1.可隨時取得任意一件衣服  2.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 |

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) |

## 3.7、使用需求

表 3.14前端功能性需求

### 3.6.1、前端

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

表 3.16後端功能性需求

### 3.6.2、後端

|  |  |
| --- | --- |
| 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 |
|  | 補 |

## 3.8、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 |

# 第4章、實際執行方案

## 4.1、軟體

### 4.1.1、衣物辨識

補

### 4.1.2、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:

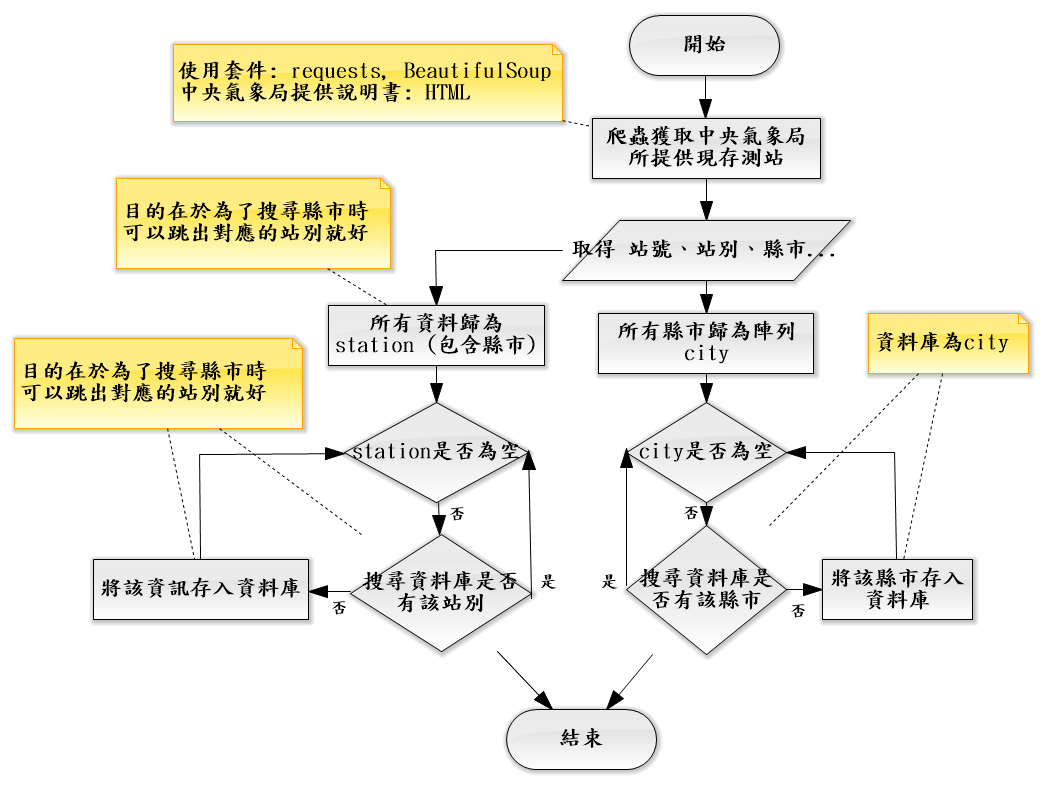
1.「​/v1​/rest​/datastore​/O-A0001-001自動氣象站-氣象觀測資料」

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>

### 4.1.3、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

### 4.1.4、資料庫架設

表 4.1料庫表格關係

資料庫的部分使用的是Microsoft SQL Server，使用工具為DBeaver，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

### 4.1.5、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

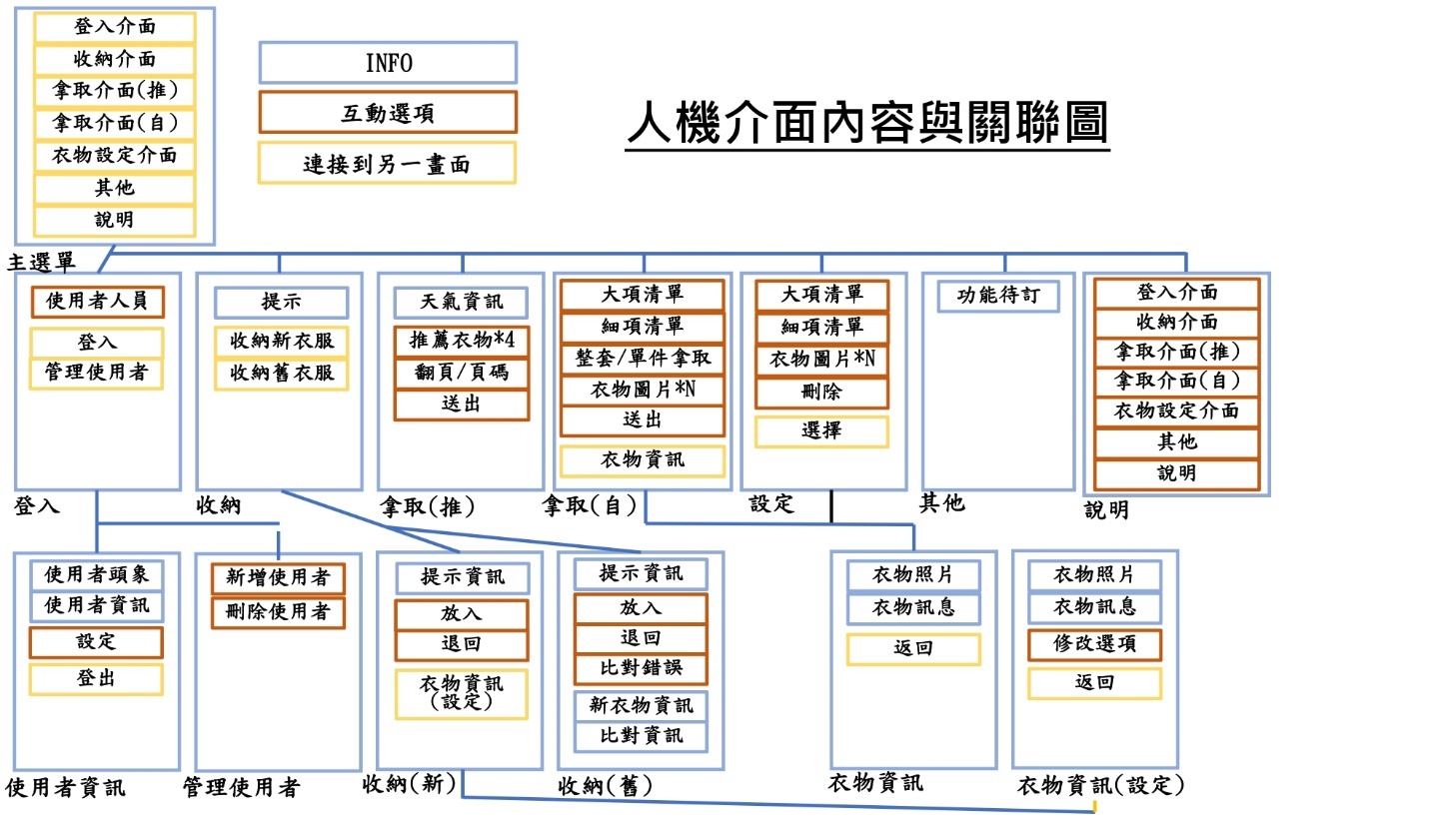
註5-2:點選後將框選該圖樣>可按選擇

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。

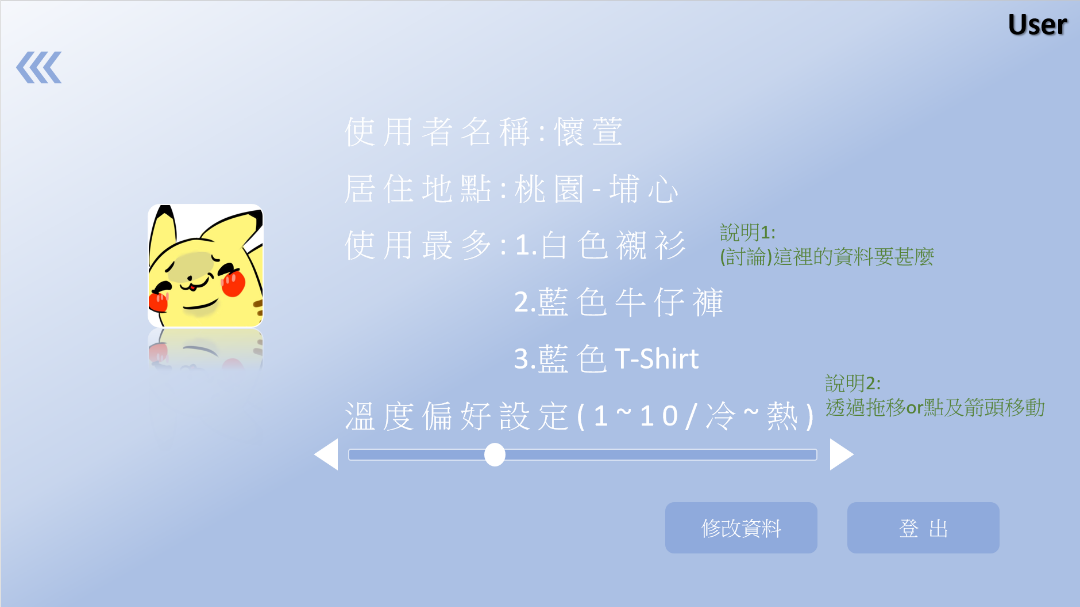
<標號>

**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

### 4.1.6、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。

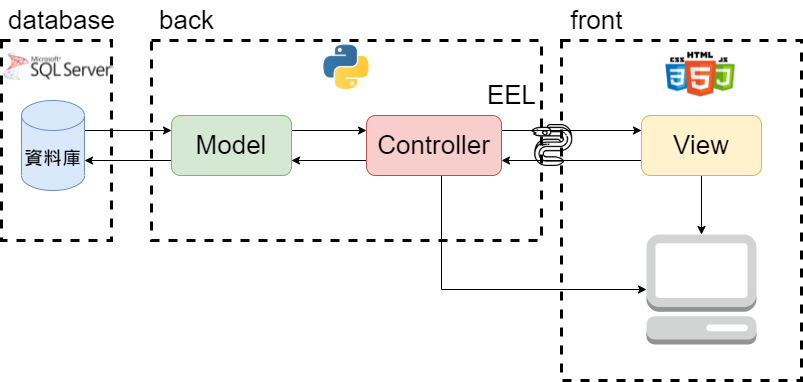
1. **Service**

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

## 4.2、硬體

### 4.2.1、事前設計

### 4.2.2、實際製作

### 4.2.3、修正

## 4.3、軟硬體連接

## 4.4、流程設計

待寫

# 第5章、研究成果

待寫

# 第6章、應用範圍與市場分析

## 6.1、市場分析

表 6.0.1市場產品比較

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。

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

## 6.2、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

# 第7章、工作進度與分配

待寫

# 參考文獻

待寫

# 附錄