國 立 臺 北 商 業 大 學

資 訊 管 理 系

１ ０ ８ 資 訊 系 統 專 案 設 計

系統手冊

組　　別：　　第 108402 組
題　　目：　　今天穿什麼?
指導老師：　　林明毅老師
組　　長：　　10546002 葉秀湘
組　　員：　　10546003 楊婷文　　N1056407 羅方伶
　　　　　　　N1056414 謝佳錚　　1054A037 簡培雯

中 華 民 國 １ ０ ８ 年 １ １ 月 ２ ０ 日

第10章　測試模型

第11章　操作手冊

第12章　使用手冊

第13章　感想

第14章　參考資料

第15章　附錄初評檢討

第一章. 背景與動機

1-1 簡介

　　科技發達的當今時代，在一天早晨睜開眼睛，最先做的不再是刷牙洗臉，而是拿起手機查看訊息、滑滑臉書、IG 等等，當回過神才赫然發現上班、上課或是跟人約好的時間就要到了，可自己卻什麼都還沒準備好，著急情況之下還得從衣櫃當中各式各樣的衣服做挑選跟搭配。因此我們想要設計出一款 APP，讓使用者將擁有的衣物透過拍照方式放入到 APP 裡面，再搭配天氣，來幫助使用者快速搭配出適當的裝扮。

1-2 SWOT 分析

內在環境

優勢	個性化界面增加使用感官 簡單明瞭的界面，方便使用者簡易操作 藉由天氣預報提醒使用者攜帶雨傘，給使用者一些小提示
劣勢	APP 需要好的推廣方式才能為使用者所知 新增一些功能要藉由推廣才能讓更多使用者知道

外在環境

機會	個性化界面增加使用感官 可以自行設定衣櫃內衣服樣式，不單單只有內部樣式 不同於其他 APP，利用拼貼方式呈現搭配效果
威脅	有許多相似 APP 存在 APP 上架門檻低，潛在 APP 隨時會出現

1-3 相關系統探討

(目前既有系統、類似點、不同點)
● Stylebook
　　Stylebook 是由時尚業內部人士參與設計的衣櫃組織管理工具，有超過 90 個功能，可以幫助使用者規劃穿著服裝，記錄你的每日穿著，也可提早計劃你的服裝搭配，不用在出門前煩惱該穿什麼，或是太常穿同樣搭配的服飾。

　　使用軟體時可導入你的實際衣服照片，不止能幫助你計劃穿些什麼，還可以幫你統計各類服飾的數據及衣著成本。甚至當你偶有不同的靈感時也可將點子儲存於靈感圖書館，它會幫你追蹤並保存你的風格靈感。

- StyleShare

    同樣以最簡單直接的 share & follow 方式運作，選擇關心的項目(男、女裝及地區等)就可以看到相關的分享。風格比較大眾化，品牌、設計及潮流都是很容易能消化、學習及購入的級數，因此使用者多是 15-25 歲的年輕人。更會製作特輯，介紹最新的潮流。在日本、香港、台灣等地都設有特派工作人員，以製作和當地潮流有關的特輯。

- Wear

    被稱為日本最大的服裝配備 APP「WEAR」是由 STARTTODAY 公司營運主要負責服裝潮流搭配，在日本擁有 500 萬以上會員以及跟 2000 家以上服裝品牌合作成為日本最大購物網站「ZOZOTOWN」，所負責營運的 APP「WEAR」提供最新的服裝穿搭，推出五個月就在日本突破 200 萬次下載，並高達 50 萬筆穿搭推薦仍持續累積。

    「WEAR」APP 在 21 日正式進軍台灣，也與台灣 FACEBOOK 擁有超過 50 萬粉絲的 EC 網站「LOVFEE」、「ORENDA」等 8 家業者都會跟「WEAR」合作，除了能在「WEAR」 APP 上看到多種穿搭外，還能直接線上購買人氣穿搭商品。另外還有 40 位潮流穿搭影響力高的模特兒與部落客也會以台灣的 WEARISTA 名義一同參與。

    不同點：SNS 功能，可追蹤人氣用戶或喜歡的搭配帳戶；另外還可使用社群關聯了解有在使用的好友

▼ 表 1-3-1 相關系統探討表

相關網站 特色	今天穿什麼	Stylebook	StyleShare	Wear
自行添加衣櫃內容	✓		●	
個人行事曆建置		●	●	●
及時天氣資訊	✓			
穿搭心得分享		●	●	●

第二章 系統目標與預期成果

2-1 系統目標

在每天早上起床時,總是會花很多時間思考今天到底要穿什麼,有時候花了老半天終於選出一套滿意的服裝穿出門,卻因為沒有注意到當天的天氣狀況,出了門才發現穿了太少,或是穿了太多,讓自己美好的一天因為服裝而有所影響。或是因為一天的天氣變化太大,因應穿衣時的天氣穿搭,到了晚上衣服的厚度已經不足以保暖了。

提供給不喜歡看天氣預報的使用者,若是當天會下雨,打開 APP 會立即跳出帶傘小提示,在選擇穿搭的同時也能得知要帶傘的訊息,出門就不會因為沒帶傘而淋成落湯雞。

本專題的目標是讓使用者減少很多決定穿搭的時間,APP 結合當天的天氣與行程,透過預覽的方式在畫面中顯示穿搭時的情況,讓使用者不需要一件一件的試穿就能知道看起來的感覺,用最少的時間完成一天最合適的穿搭。

▼表 2-2-1 主要目標

解決使用者在穿搭上的迷茫:	
解決天氣變化而導致穿衣不適當	開啟 GPS 可得知當地天氣,以及當天的天氣變化,幫助使用者決定適合的穿搭與配件。解決使用者因為天冷穿太少、太熱又穿太多、下雨忘記帶傘等困擾。
針對不想花時間思考穿搭的人	打開 APP 能立即提供穿搭建議,不用再花費時間精力思考或進行無數次的試穿。

2-2 預期成果

- 打開 APP 能立即得到穿搭建議
- 使用者能即時得知天氣資訊
- 以簡單的操作供使用者為衣服建檔
- 提供的穿搭不僅合時宜和天氣還要好看

第三章 系統規格

3-1 系統架構
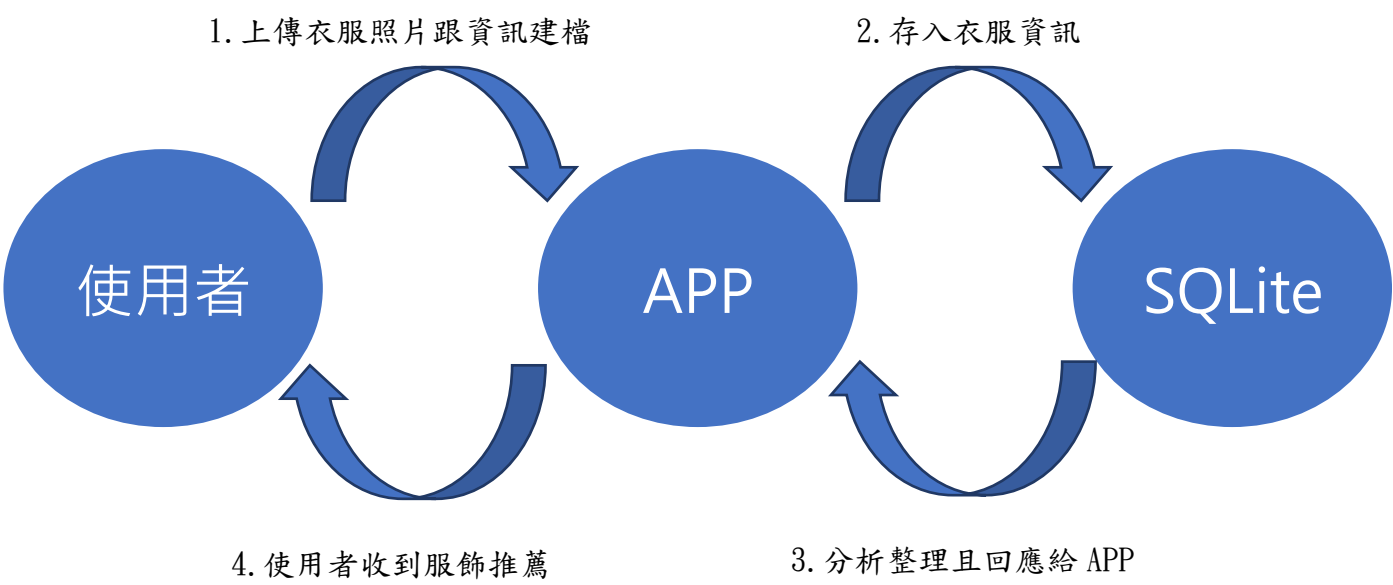
▲圖 3-1-1 系統架構圖

3-2 系統軟硬體需求與技術平台

▼表 3-2-1 系統需求表

軟、硬體需求		
作業系統	最低系統需求	建議系統需求
版本	Android 5.0 以上	Android 7.1.1 以上
處理器	雙核心以上	四核心以上
磁碟可用空間	1GB 以上可用空間	2GB 以上可用空間
RAM	1GB 以上可用記憶體	2GB 以上可用記憶體
網路	3G、4G 行動上網、Wi-Fi 無線網路	

3-3 使用標準與工具

▼表 3-3-1 使用標準與工具表

系統開發環境	
作業系統	Windows 7、Windows 10
資料庫	SQLite
程式開發技術	
前端技術	Android Studio
後段技術	JAVA
編輯器	Android Studio
管理程式平台	
版本控制	Git
專案管理	Github
文件美工程式	
簡報	Microsoft Office PowerPoint
設計圖樣	Adobe Illustrator、Adobe Photoshop
文件	Microsoft Office Word、Google 文件、Visual Paradigm for UML

第四章 專案時程與組織分工

4-1 甘特圖

工作項目 \ 月份	2	3	4	5	6	7	8	9	10	11	12
主題構思	預/實	預/實									
開發工具學習	預/實	預/實									
系統功能分析		預/實									
前端設計			預/實	預/實	預/實	預/實	預/實	預/實			
前端開發			預/實	預/實	預/實	預/實	預/實	預/實	預/實	預	
資料庫建置					預/實	預/實	預/實				
系統模型		預/實				預/實	預				
系統測試								預	預/實	預/實	
系統整合					預/實	預			預/實	預/實	
系統需求分析		預/實				預/實	預				
系統手冊製作			預/實	預/實	預/實		預	預	預	實	
最終檢查										預/實	預/實

▲ 圖 4-1-1 專案時程甘特圖

■ 預期進度
■ 實際進度
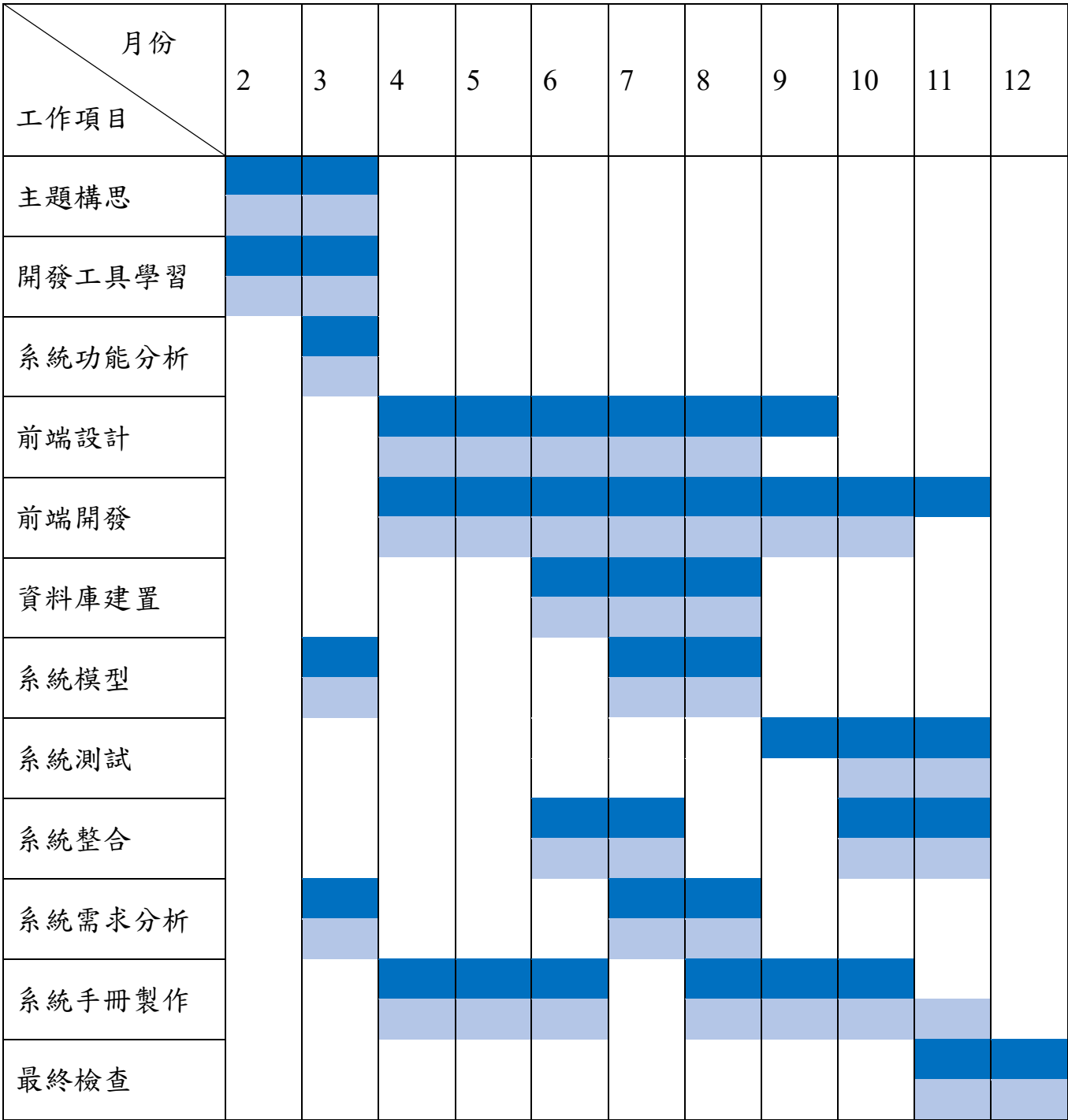
4-2 專案組織與分工

<div align="center">▼表 4-2-1 工作分配表</div>

項目	學號姓名	10546002 葉秀湘	10546003 楊婷文	N1056407 羅方伶	N1056414 謝佳錚	1054A037 簡培雯
主題	決定主題	●	●	●	●	●
	背景與動機	●	●	●	●	●
	需求分析	●	●	●	●	●
	資料蒐集	●	●	●	●	●
資料庫	資料庫規劃	●	○	●	○	○
	資料庫建置	○	○	●	○	○
	資料庫校能	●	○	●	○	○
美術設計	Logo 設計	○	○	○	○	●
	名牌設計	○	○	○	●	○
	Icon 圖示設計	○	○	○	○	●
	介面設計	○	○	○	○	●
	簡報製作	○	○	○	○	●
	海報設計	○	○	○	●	○
文件	手冊編寫	○	●	○	●	●
	手冊編排	○	○	○	●	○
	簡介編寫	○	○	○	●	●
	內容校正	●	●	●	●	●
UML分析	使用個案圖	○	○	○	○	●
	使用活動圖	○	○	○	○	●
	分析類別圖	○	●	●	○	○
	循序圖	○	●	●	○	○
	設計類別圖	●	●	●	○	○
	部屬圖	○	●	●	○	○
	元件圖	○	●	●	○	○
	套件圖	○	●	●	○	○
	狀態機	●	●	●	○	○
程式撰寫	前端介面	●	○	●	○	○
	前端設計	●	○	●	○	●
	程式碼撰寫	●	○	●	○	○
	程式碼整合	●	○	○	○	○
	程式碼偵錯	●	○	●	○	○

●主要負責　○輔助

第五章 需求模型

5-1 使用者需求

1. 功能需求

　　系統透過使用者所建立衣服中搭配當天的天氣，提供最合適的穿搭建議，使用者也可以針對自己的喜好做修改，同時APP也會提供使用者瀏覽一周的天氣情況。

▼表 5-1-1 功能需求表

功能需求	使用案例
依照衣服形式分類，使用者可以瀏覽與增減每件衣物的細項	衣服管理
提供使用者所在地的一周天氣情況	一周天氣
提供穿搭建議，使用者可以在 APP 上預覽穿搭起來的模樣，也可以自行做修改	預覽穿衣

2. 非功能需求

　　使用者可以透過簡單的圖形化界面，一打開 APP 就可以得到穿搭建議。建檔的方式也只需要透過選取的方式建立，不需要有太多複雜的設定。

▼表 5-1-2 非功能需求表

非功能需求
簡單易操作的使用者介面
圖片讀取時間於 2 秒之內
隱私保護，未經使用者允許圖片不會散布至網路上

5-2 使用個案圖(Use case diagram)
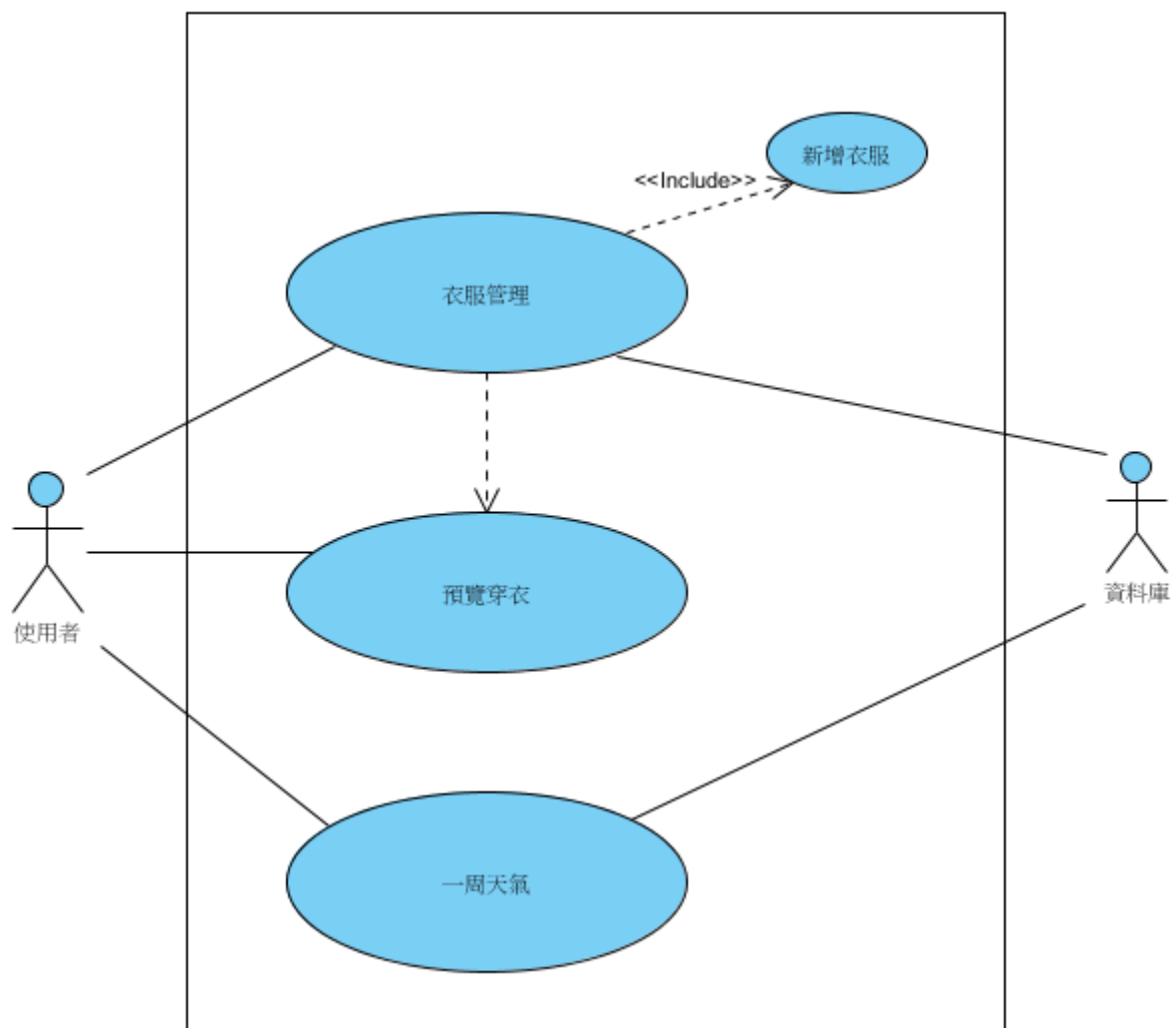
圖 5-2-1 使用案例圖

5-3 使用個案描述

Use Case	管理衣服
介面功能	依照衣服形式分類，使用者可以瀏覽與修改、增減每件衣物的細項
備註	新增衣服的詳細活動圖，由圖 5-3-2 顯示

圖 5-3-1 活動圖：管理衣服

10
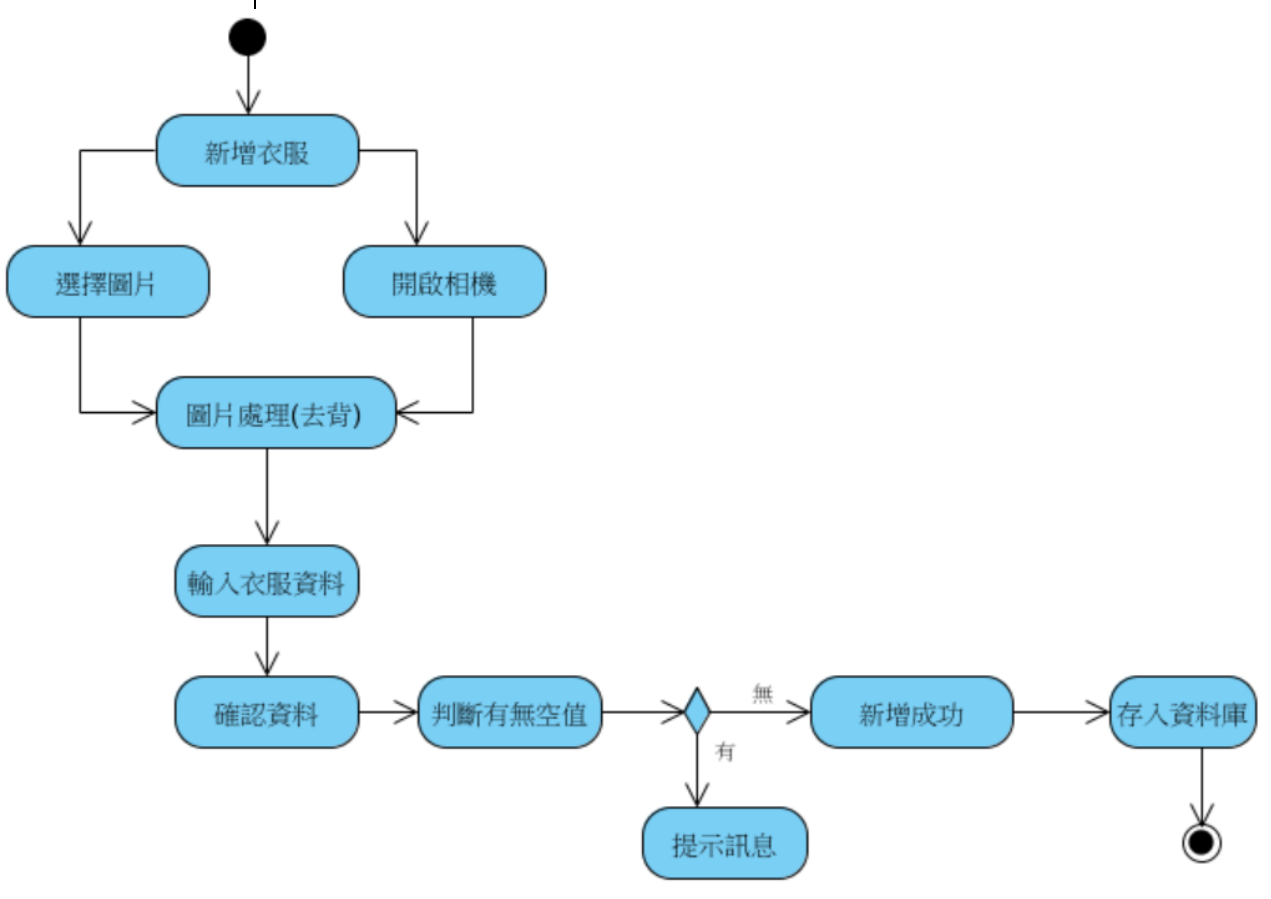
Use Case	新增衣服
介面功能	在新增介面中選取符合單件衣服的相關資訊，系統會自動存入資料庫中，日後可以自行再修改
備註	為管理衣服中的新增衣服活動圖示

圖 5-3-2 活動圖：新增衣服

Use Case	預覽穿衣
介面功能	系統會提供使用者穿衣建議，並將穿搭起來的樣子顯示於畫面上，也可以自行修改穿搭的建議。並依據降雨機率提供溫馨小提醒。
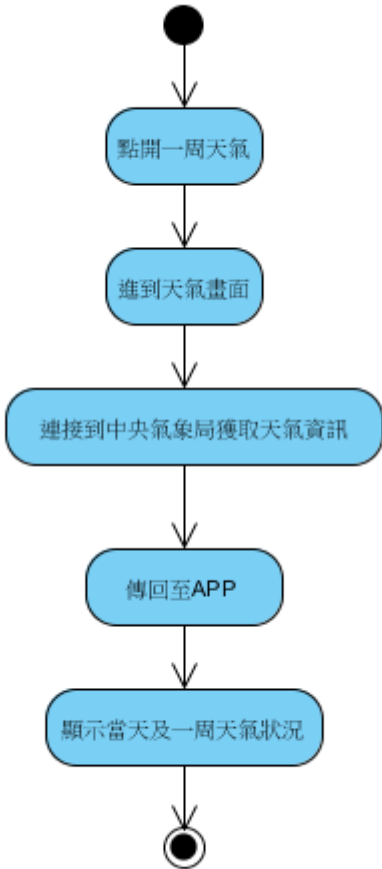
圖 5-3-3 活動圖：預覽穿衣

Use Case	一周天氣
介面功能	透過中央氣象局 API 取得天氣資訊並傳回到 APP，藉此提供給使用者天氣的詳細資訊。

圖 5-3-4 活動圖：一周天氣

5-4 分析類別圖(Analysis class diagram)

備註	使用者上傳資料到 APP，APP 去各程式間獲取資料並回傳。
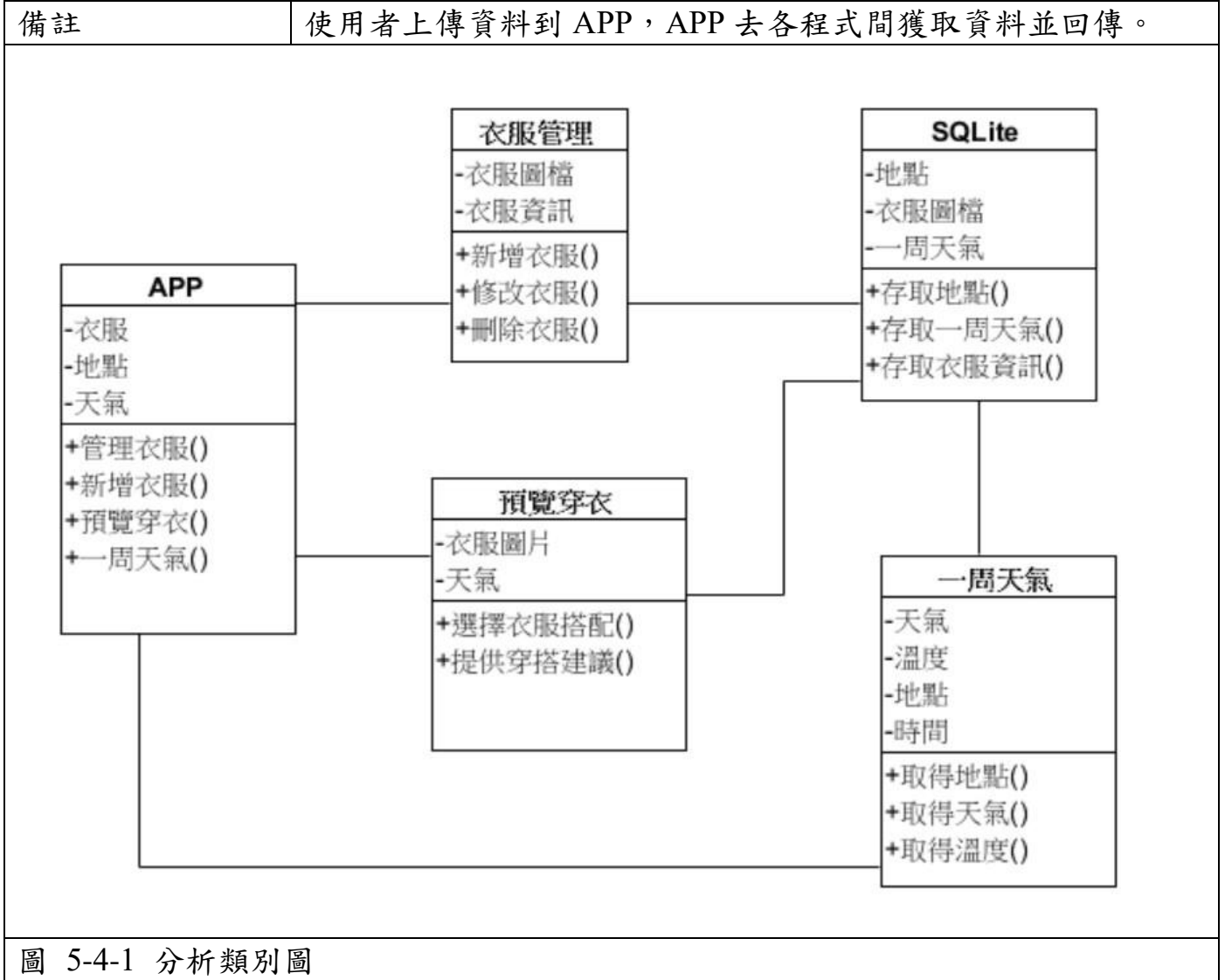
圖 5-4-1 分析類別圖

第六章 設計模型

6-1 循序圖(Sequential diagram)

圖 6-1-1 循序圖
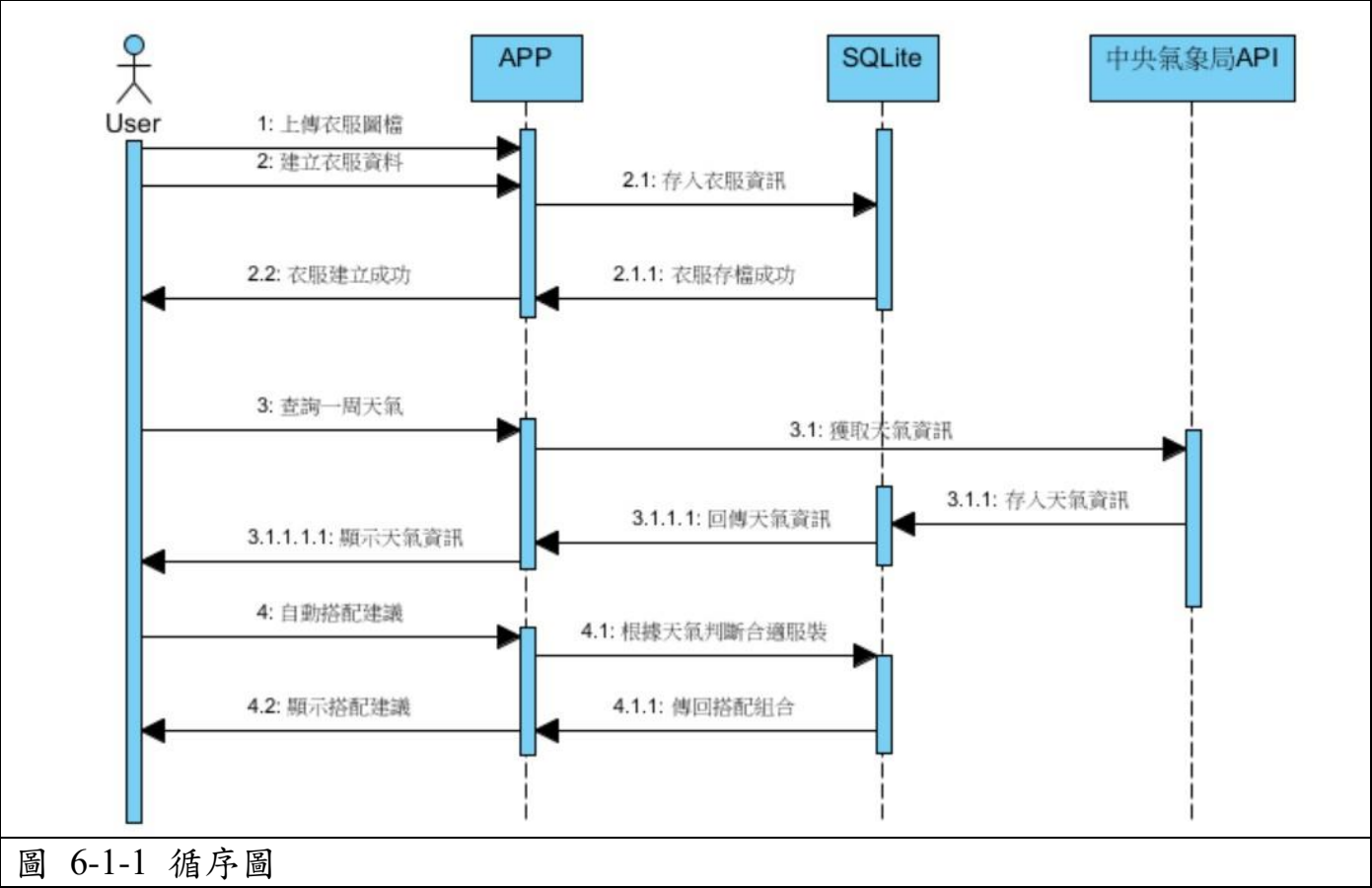
6-2 設計類別圖(Design class diagram)

備註	最終使用者在設定完個人資料，並上傳相關資料後，APP 參照各式各樣的數值去資料分析並回傳結果顯示在 APP 上。
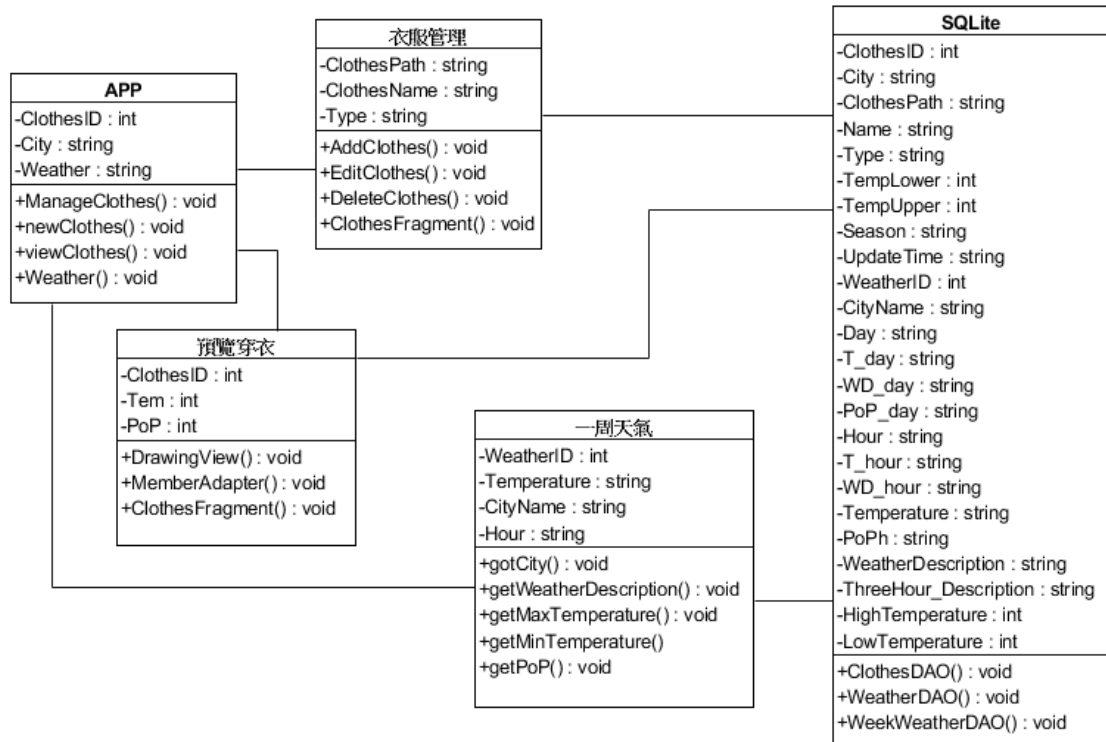
圖 6-2-1 設計類別圖

第七章 實作模型

7-1 佈署圖(Deployment diagram)

圖 7-1 佈署圖
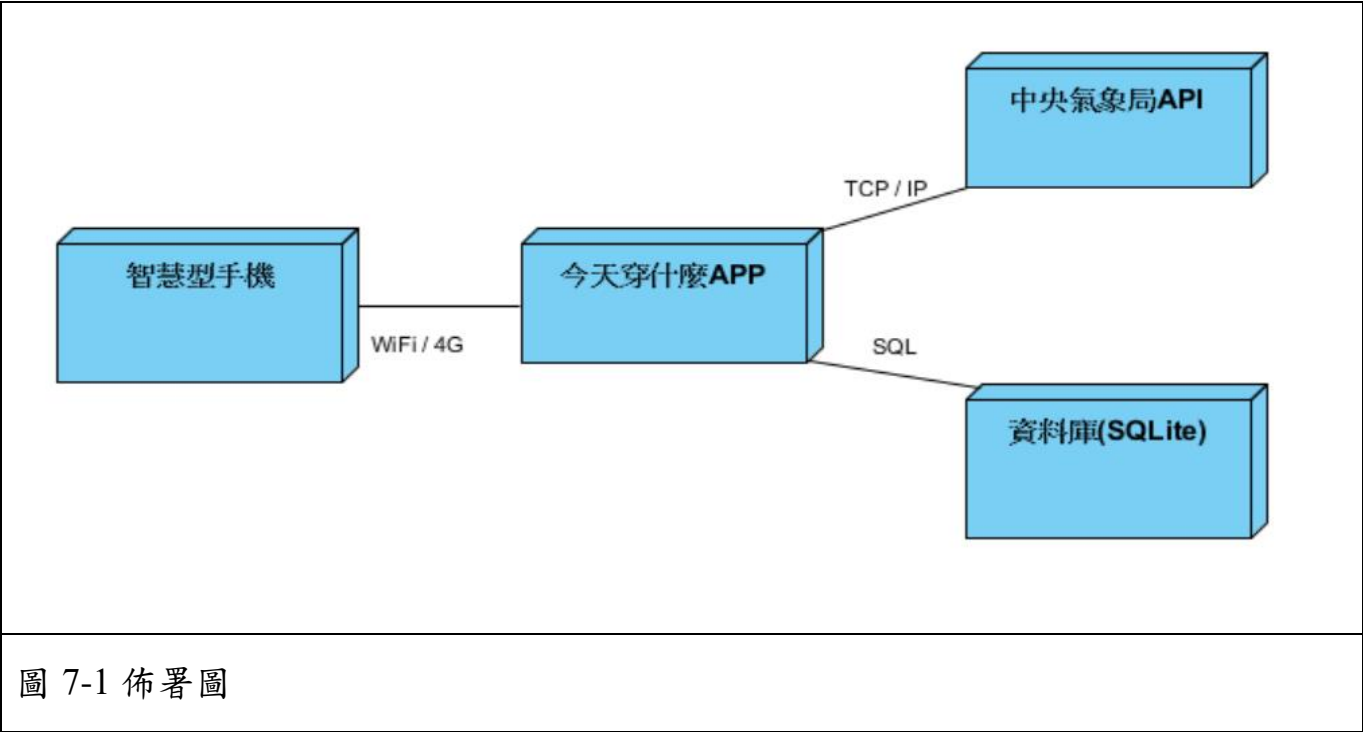
7-2 套件圖(Package diagram)

圖 7-2-1 套件圖
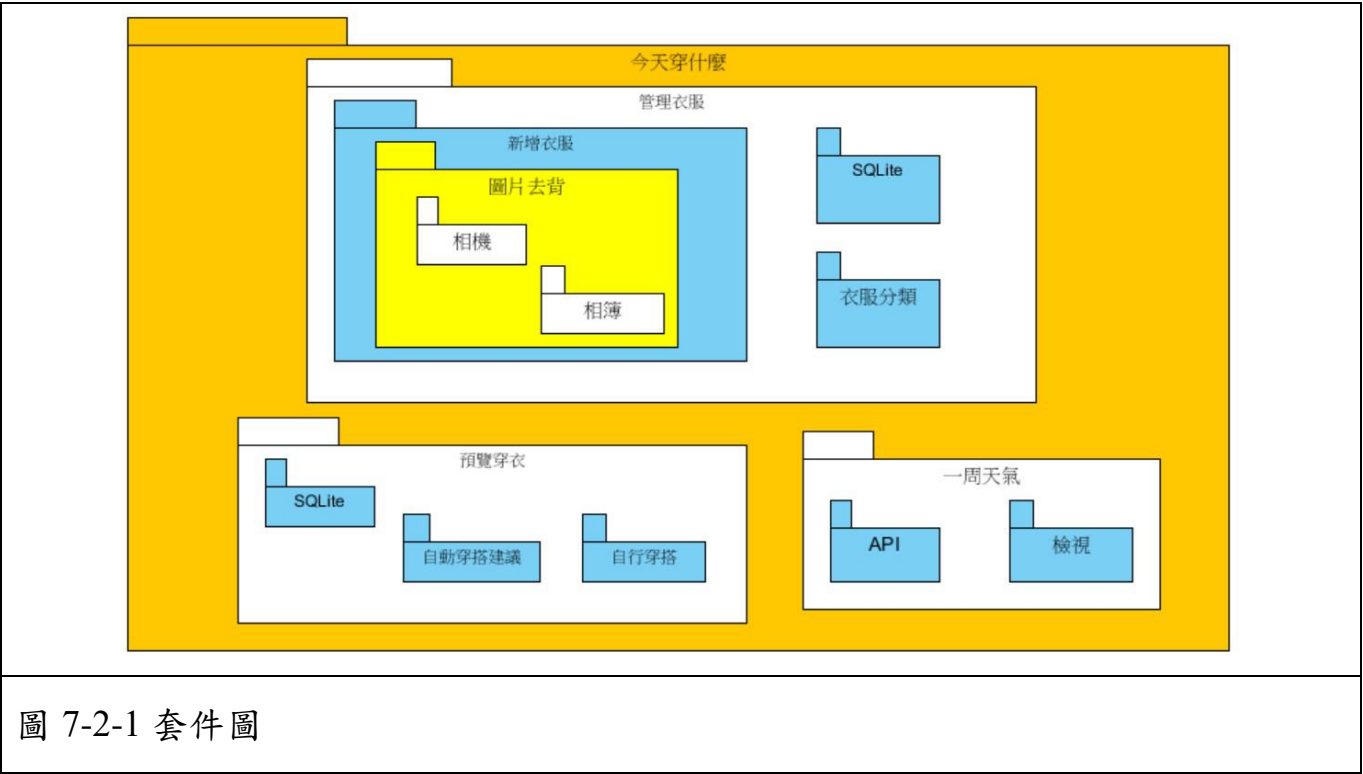
7-3 元件圖(Component diagram)
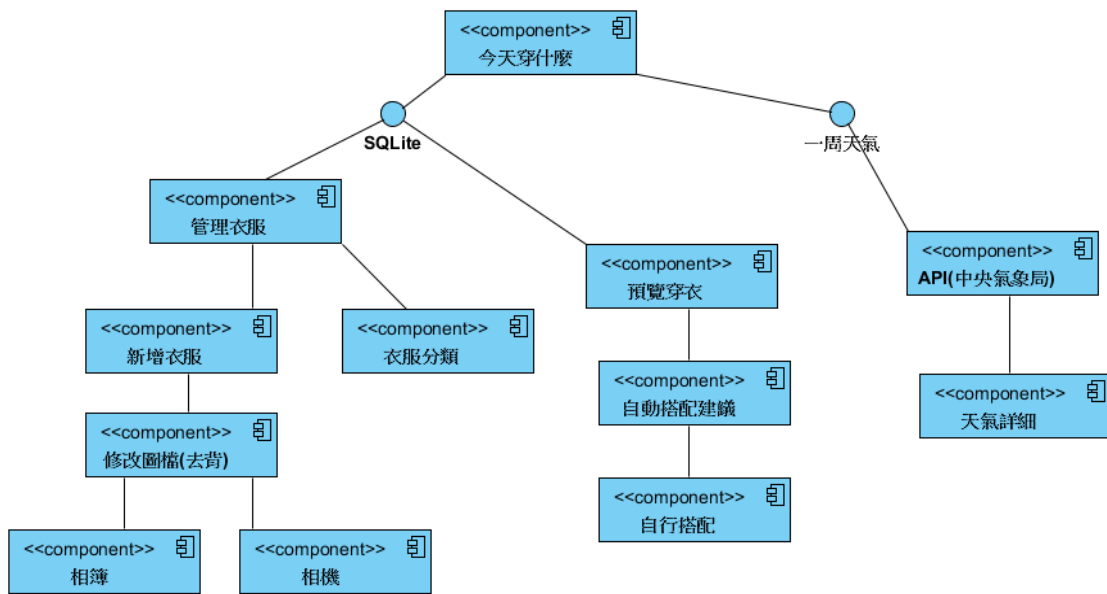
圖 7-3-1 元件圖

7-4 狀態機(State machine)
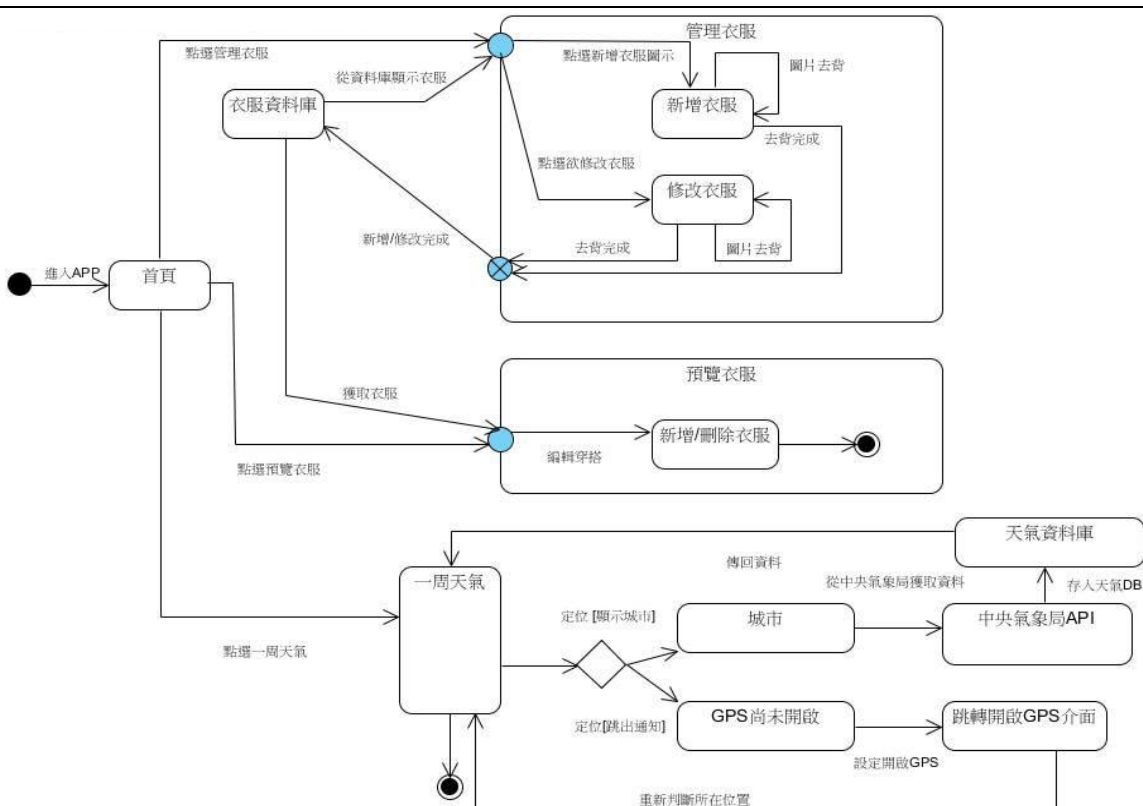
圖 7-4-1 狀態機

第八章 資料庫設置

8-1 資料庫關聯表

▼表 8-1-1 管理衣服資料庫關聯表

ManageClothes			
資料行	中文名稱	資料類型	允許 NULL
_id	編號	Int	N
clothespath	衣服圖片路徑	Text	N
Name	衣服名稱	Text	N
Type	衣服分類	Text	N
Style	衣服風格	Text	N
tempLower	合適溫度下限	Int	N
tempUpper	合適溫度上限	Int	N
Season	季節	Text	N
updatetime	更新日期	Text	N

▼表 8-1-2 天氣資料庫關聯表

Weather			
資料行	中文名稱	資料類型	允許 NULL
_id	編號	Int	N
CityName	城市名稱	Text	N
NowCity	現在城市	Text	N
Day	現在日期	Text	N
T_Day	溫度日期	Text	N
WD_Day	天氣顯示日期	Text	N
PoP_Day	降雨機率日期	Text	N
Hour	現在時間	Text	N
T_Hour	溫度時間	Text	N
WD_Hour	天氣顯示時間	Text	N
Temperature	溫度	Text	N
PoPh	降雨機率	Text	N
WeatherDescription	天氣描述	Text	N
Threehour_Descript	三小時天氣描述	Text	N

Week_Weather			
資料行	中文名稱	資料類型	允許 NULL
_id	編號	Int	N
CityName	城市名稱	Text	N
Day	日期	Text	N
MaxT_Day	最高溫度日期	Text	N
MinT_Day	最低溫度日期	Text	N
WD_Day	天氣顯示日期	Text	N
PoP_Day	降雨機率日期	Text	N
Hour	溫度時間	Text	N
HighTemperature	最高溫	Text	N
LowTemperature	最低溫	Text	N
PoPh	降雨機率	Text	N
WeatherDescription	天氣描述	Text	N

第九章 程式

9-1 元件清單及規格描述

▼表 9-1-1 元件清單

編號	檔案名稱	功能說明
1	Weather	顯示天氣資訊
2	Viewclothes_CustomBitmap	給予推薦穿衣照片
3	Viewclothes_DrawingView	調整推薦照片
4	Viewclothes_MemberAdapter	預覽穿衣中類別細項的適配器
5	Viewclothes_PagerAdapter	預覽穿衣中類別適配器
6	viewclothes	顯示穿衣推薦
7	AddClothes	新增衣服照片
8	MainActivity	APP 首頁
9	Manageclothes	管理衣服
10	Clothes_MemberAdapter	衣服管理中類別細項適配器
11	Clothes_PagerAdapter	衣服管理中類別適配器
12	WelcomeActivity	應用程式的初始畫面
13	editclothes	修改衣服
14	newclothes	新增衣服

程式名稱	Weather
目的	顯示天氣資訊

程式碼

```
package com.example.weather;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Gravity;
import android.view.View;
import android.widget.HorizontalScrollView;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import com.example.clothes.MainActivity;
import com.example.clothes.R;
import com.example.clothes.database.WeekWeatherDAO;
import com.example.clothes.database.getWeather;
import com.example.clothes.database.getWeekWeather;
import com.example.clothes.database.weatherDAO;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
public class weather extends AppCompatActivity {

    // 宣告資料庫功能類別欄位變數

    private WeekWeatherDAO dao_Week;
    private weatherDAO dao_Today;
```

```java
    long dbcount = 1;
    getWeekWeather getWeekWeather = new getWeekWeather();
    public TextView Location, week_location; // 顯示城市

    public TextView Temperature, LowTemperature, HighTemperature; // 顯示溫度

    public TextView WeatherDescription, WeatherDescription_array; // 顯示天氣敘
述

    public TextView Today_date, Today_Time, WD_Day, PoP_Day, MaxT_Day,
MaxT_Hour, MinT_Day; // 顯示日期時間

    public ImageView weather_image, MON_image, TUE_image, WED_image,
THU_image, FRI_image, SAT_image, SUN_image;
    public TextView MON_Min, TUE_Min, WED_Min, THU_Min, FRI_Min,
SAT_Min, SUN_Min;
    public TextView MON_Max, TUE_Max, WED_Max, THU_Max, FRI_Max,
SAT_Max, SUN_Max;
    public TextView week_PoP, PoPh, MON_PoP, TUE_PoP, WED_PoP, THU_PoP,
FRI_PoP, SAT_PoP, SUN_PoP; // 降雨機率

    public    ArrayList<String>    locationList,    WD_Day_list,    PoP_Day_list,
MaxT_Day_list, WeatherDescription_array_list, week_PoP_list, HighTemperature_list,
MinT_Day_list, LowTemperature_list, MaxT_Hour_list;
    public ArrayList<getWeather> dao_Today_list;
    private HorizontalScrollView scrollView;
    private LinearLayout linear;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_weekweather);
        dao_Week = new WeekWeatherDAO(getApplicationContext());
        dao_Today = new weatherDAO(getApplicationContext());
        locationList = new ArrayList<String>();
        WD_Day_list = new ArrayList<String>();
        PoP_Day_list = new ArrayList<String>();
        MaxT_Day_list = new ArrayList<String>();
        WeatherDescription_array_list = new ArrayList<String>();
        week_PoP_list = new ArrayList<String>();
        HighTemperature_list = new ArrayList<String>();
```

```java
        MinT_Day_list = new ArrayList<String>();
        LowTemperature_list = new ArrayList<String>();
        MaxT_Hour_list = new ArrayList<String>();
        scrollView = (HorizontalScrollView) this.findViewById(R.id.scroll_view);
        linear = (LinearLayout) this.findViewById(R.id.linear);

        // 顯示所在位置

        Location = (TextView) findViewById(R.id.City);
        week_location = (TextView) findViewById(R.id.week_location);

        // 顯示最高最低溫

        Temperature = (TextView) findViewById(R.id.Temperature);
        HighTemperature = (TextView) findViewById(R.id.HighTemperature);
        LowTemperature = (TextView) findViewById(R.id.LowTemperature);
        PoPh = (TextView) findViewById(R.id.Rainfall_probability);
        week_PoP = (TextView) findViewById(R.id.week_PoP);
        WeatherDescription = (TextView) findViewById(R.id.weather);
        WeatherDescription_array                    =                    (TextView)
findViewById(R.id.WeatherDescription_array);
        Today_date = (TextView) findViewById(R.id.Today_date);
        Today_Time = (TextView) findViewById(R.id.Today_Time);
        WD_Day = (TextView) findViewById(R.id.WD_Day);
        PoP_Day = (TextView) findViewById(R.id.PoP_Day);
        MaxT_Day = (TextView) findViewById(R.id.MaxT_Day);
        MinT_Day = (TextView) findViewById(R.id.MinT_Day);
        MaxT_Hour = (TextView) findViewById(R.id.MaxT_Hour);
        weather_image = (ImageView) findViewById(R.id.weather_image);
        MON_image = (ImageView) findViewById(R.id.MON_image);
        MON_Min = (TextView) findViewById(R.id.MON_Min);
        MON_Max = (TextView) findViewById(R.id.MON_Max);
        MON_PoP = (TextView) findViewById(R.id.MON_PoP);
        TUE_image = (ImageView) findViewById(R.id.TUE_image);
        TUE_Min = (TextView) findViewById(R.id.TUE_Min);
        TUE_Max = (TextView) findViewById(R.id.TUE_Max);
        TUE_PoP = (TextView) findViewById(R.id.TUE_PoP);
        WED_image = (ImageView) findViewById(R.id.WED_image);
        WED_Min = (TextView) findViewById(R.id.WED_Min);
        WED_Max = (TextView) findViewById(R.id.WED_Max);
        WED_PoP = (TextView) findViewById(R.id.WED_PoP);
        THU_image = (ImageView) findViewById(R.id.THU_image);
        THU_Min = (TextView) findViewById(R.id.THU_Min);
```

```java
        THU_Max = (TextView) findViewById(R.id.THU_Max);
        THU_PoP = (TextView) findViewById(R.id.THU_PoP);
        FRI_image = (ImageView) findViewById(R.id.FRI_image);
        FRI_Min = (TextView) findViewById(R.id.FRI_Min);
        FRI_Max = (TextView) findViewById(R.id.FRI_Max);
        FRI_PoP = (TextView) findViewById(R.id.FRI_PoP);
        SAT_image = (ImageView) findViewById(R.id.SAT_image);
        SAT_Min = (TextView) findViewById(R.id.SAT_Min);
        SAT_Max = (TextView) findViewById(R.id.SAT_Max);
        SAT_PoP = (TextView) findViewById(R.id.SAT_PoP);
        SUN_image = (ImageView) findViewById(R.id.SUN_image);
        SUN_Min = (TextView) findViewById(R.id.SUN_Min);
        SUN_Max = (TextView) findViewById(R.id.SUN_Max);
        SUN_PoP = (TextView) findViewById(R.id.SUN_PoP);
        Today_date.setText(new        SimpleDateFormat("yyyy-MM-dd").format(new
Date()));
        Today_Time.setText(new SimpleDateFormat("HH").format(new Date()));

        // 讀取各縣市一週天氣預報

        new
WeekTask().execute("https://opendata.cwb.gov.tw/fileapi/v1/opendataapi/F-D0047-091
?Authorization=CWB-6BB38BEE-559E-42AB-9AAD-698C12D12E22&downloadTyp
e=WEB&format=JSON");
        String city = dao_Today.getoneID(1).getNowCity();
        Location.setText(city);
        dao_Today_list = dao_Today.getWDweather(Location.getText().toString());
        String Description;
        String[] Description_array;
        for(int y = 0 ; y < dao_Today.getWDweather(city).size() ; y++){

if(dao_Today.getWDweather(city).get(0).getDay().equals(dao_Today.getWDweather(ci
ty).get(y).getT_Day()) &&

dao_Today.getWDweather(city).get(0).getHour().equals(dao_Today.getWDweather(cit
y).get(y).getT_Hour())){

Temperature.setText(dao_Today.getWDweather(city).get(y).getTemperature()  +  " °C
");
                Description = dao_Today_list.get(y).getWeatherDescription();

                Description_array = Description.split("。");
```

```java
                WeatherDescription.setText(Description_array[0] + "\n" + "\n" +
Description_array[1]  +  "\n"  +  "\n"  +  Description_array[2]  +  "\n"  +  "\n"  +
Description_array[3]  +  "\n"  +  "\n"  +  Description_array[4]  +  "\n"  +  "\n"  +
Description_array[5]);
                    PoPh.setText(dao_Today_list.get(y).getPoPh() + " % ");

                    if(Description_array[0].equals("      晴      ")              ||

Description_array[0].equals("晴時多雲") || Description_array[0].equals("多雲時晴")){

weather_image.setImageDrawable(getResources().getDrawable( R.drawable.sunny));
                }else     if(Description_array[0].equals("    多    雲    ")     ||

Description_array[0].equals("多雲時陰") || Description_array[0].equals("陰時多雲") ||

Description_array[0].contains("陰")){

weather_image.setImageDrawable(getResources().getDrawable(  R.drawable.cloudybac
kground));
                }else if(Description_array[0].contains("雨")){

weather_image.setImageDrawable(getResources().getDrawable( R.drawable.rainy));
                }else     if(Description_array[0].contains("    雷    ")     ||

Description_array[0].contains("雷") && Description_array[0].contains("雨")){

weather_image.setImageDrawable(getResources().getDrawable( R.drawable.lightning))
;
                }else if(Description_array[0].contains("雪")){

weather_image.setImageDrawable(getResources().getDrawable( R.drawable.snowy));
                }else if(Description_array[0].contains("霧")){

weather_image.setImageDrawable(getResources().getDrawable( R.drawable.windy));
                }
                break;
```

```java
                }
            }
        for (int n = 0; n < 8; n++) {
            String[] threehour_array =
dao_Today_list.get(n).getThreehour_Description().split("。");

            LinearLayout.LayoutParams linearLp = new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
LinearLayout.LayoutParams.WRAP_CONTENT);
            LinearLayout myLinear = new LinearLayout(this);
            linearLp.setMargins(30, 30, 30, 30);
            myLinear.setOrientation(LinearLayout.VERTICAL);
            myLinear.setTag(n);
            linear.addView(myLinear, linearLp);

            // 小時

            LinearLayout.LayoutParams textViewLp = new
LinearLayout.LayoutParams(150, 150);
            TextView textView = new TextView(this);
            textView.setText(dao_Today_list.get(n).getWD_Hour() + ": 00");
            textView.setGravity(Gravity.CENTER);
            myLinear.addView(textView, textViewLp);

            // 天氣圖示

            LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(150,
150);

            ImageView imageView = new ImageView(this);

            if(threehour_array[0].equals("晴")){

                imageView.setBackgroundResource(R.drawable.sun);

            }else if(threehour_array[0].equals("晴時多雲") ||

threehour_array[0].equals("多雲時晴")){

                imageView.setBackgroundResource(R.drawable.sunpartlyclear);

            }else if(threehour_array[0].contains("多雲") ||

threehour_array[0].contains("多雲時陰") || threehour_array[0].contains("陰時多雲")){

                imageView.setBackgroundResource(R.drawable.mostlycloudy);
```

```java
            }else if(threehour_array[0].contains("陰") ){
                imageView.setBackgroundResource(R.drawable.cloudy);
            }else if(threehour_array[0].contains("雨") ){
                imageView.setBackgroundResource(R.drawable.rain);
            }else    if(threehour_array[0].equals("多雲陣雨")    ||
threehour_array[0].equals("多雲短暫雨") || threehour_array[0].equals("多雲短暫陣雨
") || threehour_array[0].equals("多雲時陰短暫雨") || threehour_array[0].equals("多雲
時陰短暫陣雨")    ||    threehour_array[0].equals("陰時多雲短暫雨")    ||
threehour_array[0].equals("陰時多雲短暫陣雨") || threehour_array[0].equals("陰時多
雲有雨")    ||    threehour_array[0].equals("陰時多雲有陣雨")    ||
threehour_array[0].equals("陰時多雲陣雨")){
                    imageView.setBackgroundResource(R.drawable.cloudywithrain);
            }else   if(threehour_array[0].equals("多雲時晴短暫陣雨")   ||
threehour_array[0].equals("多雲時晴短暫雨") || threehour_array[0].equals("晴時多雲
短暫陣雨") || threehour_array[0].equals("晴短暫陣雨") || threehour_array[0].equals("
晴午後陰短暫雨")   ||   threehour_array[0].equals("晴午後陰短暫陣雨")   ||
threehour_array[0].equals("晴時多雲陣雨") || threehour_array[0].equals("多雲時晴陣
雨")){

imageView.setBackgroundResource(R.drawable.afternooncloudywithshowers);
            }    else    if    (threehour_array[0].contains("雷")    &&
threehour_array[0].contains("雨") || threehour_array[0].contains("雷")) {
                imageView.setBackgroundResource(R.drawable.thunderstorms);
```

```java
            }else if(threehour_array[0].contains("霧")){

                imageView.setBackgroundResource(R.drawable.fog);

            }else if(threehour_array[0].contains("雪")){

                imageView.setBackgroundResource(R.drawable.snow);

            }
            imageView.setScaleType(ImageView.ScaleType. CENTER_CROP);
            myLinear.addView(imageView, lp);

            // 溫度

            LinearLayout.LayoutParams          textTem         =         new
LinearLayout.LayoutParams(150, 150);
            TextView threehour_Tem = new TextView(this);
            threehour_Tem.setText(threehour_array[2].substring(4,6) + "°C");
            threehour_Tem.setGravity(Gravity.CENTER);
            myLinear.addView(threehour_Tem, textTem);

        }
    }

    // 取得伺服端傳來回應

    private class WeekTask extends AsyncTask<String, Void, String> {
        @Override
        protected String doInBackground(String... params) {
            StringBuilder sb = new StringBuilder ();
            try {
                URL url = new URL ( params[0] );
                BufferedReader in = new BufferedReader (new InputStreamReader
( url.openStream () ) );
                String line = in.readLine ();
                while (line != null) {
                    sb.append ( line );
                    line = in.readLine ();
                }
            } catch (MalformedURLException e) {
                e.printStackTrace ();
            } catch (IOException e) {
                e.printStackTrace ();
            }
            return sb.toString ();
        }
```

```java
        protected void onPostExecute(String week_data) {
            super.onPostExecute ( week_data );
            parseJSON ( week_data );
        }
        private void parseJSON(String week_data)    {
            JSONObject Ob;
            try{
                Ob = new JSONObject(week_data);
                JSONObject                 locations                 =
Ob.getJSONObject("cwbopendata").getJSONObject("dataset").getJSONObject
("locations");
                JSONArray location_array = locations.getJSONArray("location");
                for (int i = 0; i < location_array .length (); i++) {
                    JSONObject JsonObject = location_array.getJSONObject(i);
                    String locationName = JsonObject.getString("locationName");
                    week_location.setText(locationName);
                    locationList.add(week_location.getText().toString());
                }
                for (int z = 0; z < location_array .length (); z++) {
                    JSONObject JsonObject = location_array.getJSONObject(z);
                    String locationName = JsonObject.getString("locationName");
                    for  (int  citycount  =  0;  citycount  <  locationList.size();
citycount++) {
                        if (locationName.equals(locationList.get(citycount))) {
                            JSONArray          weatherElement          =
JsonObject.getJSONArray("weatherElement");
                            for (int j = 0; j < weatherElement.length(); j++) {
                                JSONObject            jsonObject2            =
weatherElement.getJSONObject(j);
                                String            elementName            =
jsonObject2.getString("elementName");
                                JSONArray                time                =
jsonObject2.getJSONArray("time");
                                switch (elementName) {

                                    // 天氣現象

                                    case "WeatherDescription":
                                        for (int k = 0; k < time.length(); k++)
{
                                            JSONObject    jsonObject3    =
time.getJSONObject(k);
```

```java
jsonObject3.getString("startTime");

String startTime =
String WD_startTime =
startTime.substring(0, 10);

WD_Day.setText(WD_startTime);

WD_Day_list.add(WD_Day.getText().toString());

JSONObject WDelementValue =
jsonObject3.getJSONObject("elementValue");

String value =
WDelementValue.getString("value");

WeatherDescription_array.setText(value);

WeatherDescription_array_list.add(WeatherDescription_array.getText().toString());

String[] Description_array =
value.split("。");

switch
(getWeek(WD_startTime)) {

    case "天":

        if
(Description_array[0].equals("晴")) {

SUN_image.setImageDrawable(getResources().getDrawable(R.drawable.sun));
        } else if
(Description_array[0].equals("晴時多雲") || Description_array[0].equals("多雲時晴"))
{

SUN_image.setImageDrawable(getResources().getDrawable(R.drawable.sunpartlyclear
));
        } else if
(Description_array[0].equals("多雲") || Description_array[0].equals("多雲時陰") ||

Description_array[0].equals("陰時多雲")) {
```

```java
SUN_image.setImageDrawable(getResources().getDrawable(R.drawable.mostlycloudy));
                                                    } else if
(Description_array[0].contains("陰")) {

SUN_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudy));
                                                    } else if
(Description_array[0].equals("多雲陣雨") || Description_array[0].equals("多雲短暫雨
") || Description_array[0].equals("多雲短暫陣雨") || Description_array[0].equals("多雲
時陰短暫雨") || Description_array[0].equals("多雲時陰短暫陣雨") ||
Description_array[0].equals("陰時多雲短暫雨") || Description_array[0].equals("陰時
多雲短暫陣雨") || Description_array[0].equals("陰時多雲有雨") ||
Description_array[0].equals("陰時多雲有陣雨") || Description_array[0].equals("陰時
多雲陣雨")) {

SUN_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudywithrain));
                                                    } else if
(Description_array[0].equals("多雲時晴短暫陣雨") || Description_array[0].equals("多
雲時晴短暫雨") || Description_array[0].equals("晴時多雲短暫陣雨") ||
Description_array[0].equals("晴短暫陣雨") || Description_array[0].equals("晴午後陰
短暫雨") || Description_array[0].equals("晴午後陰短暫陣雨") ||
Description_array[0].equals("晴時多雲陣雨") || Description_array[0].equals("多雲時
晴陣雨")) {

SUN_image.setImageDrawable(getResources().getDrawable(R.drawable.afternooncloudywithshowers));
```

```java
                                                                        } else if
(Description_array[0].contains(" 雷 ")  &&  Description_array[0].contains(" 雨 ")  ||
Description_array[0].contains("雷")) {

SUN_image.setImageDrawable(getResources().getDrawable(R.drawable.thunderstorms
));
                                                                        } else if
(Description_array[0].contains("霧")) {

SUN_image.setImageDrawable(getResources().getDrawable(R.drawable.fog));
                                                                        } else if
(Description_array[0].contains("雪")) {

SUN_image.setImageDrawable(getResources().getDrawable(R.drawable.snow));
                                                                        }
                                                                        break;
                                                            case "一":
                                                                if
(Description_array[0].equals("晴")) {

MON_image.setImageDrawable(getResources().getDrawable(R.drawable.sun));
                                                                        } else if
(Description_array[0].equals("晴時多雲") || Description_array[0].equals("多雲時晴"))
{

MON_image.setImageDrawable(getResources().getDrawable(R.drawable.sunpartlyclea
r));
                                                                        } else if
(Description_array[0].equals(" 多 雲 ") || Description_array[0].equals(" 多 雲 時 陰 ") ||
Description_array[0].equals("陰時多雲")) {

MON_image.setImageDrawable(getResources().getDrawable(R.drawable.mostlycloudy
```

```java
));
                                                } else if
(Description_array[0].contains("陰")) {

MON_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudy));
                                                } else if
(Description_array[0].equals("多雲陣雨") || Description_array[0].equals("多雲短暫雨
") || Description_array[0].equals("多雲短暫陣雨") || Description_array[0].equals("多雲
時陰短暫雨") || Description_array[0].equals("多雲時陰短暫陣雨") ||
Description_array[0].equals("陰時多雲短暫雨") || Description_array[0].equals("陰時
多雲短暫陣雨") || Description_array[0].equals("陰時多雲有雨") ||
Description_array[0].equals("陰時多雲有陣雨") || Description_array[0].equals("陰時
多雲陣雨")) {

MON_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudywithrain));
                                                } else if
(Description_array[0].equals("多雲時晴短暫陣雨") || Description_array[0].equals("多
雲時晴短暫雨") || Description_array[0].equals("晴時多雲短暫陣雨") ||
Description_array[0].equals("晴短暫陣雨") || Description_array[0].equals("晴午後陰
短暫雨") || Description_array[0].equals("晴午後陰短暫陣雨") ||
Description_array[0].equals("晴時多雲陣雨") || Description_array[0].equals("多雲時
晴陣雨")) {

MON_image.setImageDrawable(getResources().getDrawable(R.drawable.afternooncloudywithshowers));
                                                } else if
```

```java
(Description_array[0].contains("雷") && Description_array[0].contains("雨") ||
Description_array[0].contains("雷")) {

MON_image.setImageDrawable(getResources().getDrawable(R.drawable.thunderstorms));
                                        } else if
(Description_array[0].contains("霧")) {

MON_image.setImageDrawable(getResources().getDrawable(R.drawable.fog));
                                        } else if
(Description_array[0].contains("雪")) {

MON_image.setImageDrawable(getResources().getDrawable(R.drawable.snow));
                                        }
                                        break;

                                case "二":
                                        if
(Description_array[0].equals("晴")) {

TUE_image.setImageDrawable(getResources().getDrawable(R.drawable.sun));
                                        } else if
(Description_array[0].equals("晴時多雲") || Description_array[0].equals("多雲時晴"))
{

TUE_image.setImageDrawable(getResources().getDrawable(R.drawable.sunpartlyclear));
                                        } else if
(Description_array[0].equals("多雲") || Description_array[0].equals("多雲時陰") ||
Description_array[0].equals("陰時多雲")) {

TUE_image.setImageDrawable(getResources().getDrawable(R.drawable.mostlycloudy));
```

```java
                                                        } else if
(Description_array[0].contains("陰")) {

TUE_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudy));
                                                        } else if
(Description_array[0].equals("多雲陣雨") || Description_array[0].equals("多雲短暫雨
") || Description_array[0].equals("多雲短暫陣雨") || Description_array[0].equals("多雲
時陰短暫雨") || Description_array[0].equals("多雲時陰短暫陣雨") ||
Description_array[0].equals("陰時多雲短暫雨") || Description_array[0].equals("陰時
多雲短暫陣雨") || Description_array[0].equals("陰時多雲有雨") ||
Description_array[0].equals("陰時多雲有陣雨") || Description_array[0].equals("陰時
多雲陣雨")) {

TUE_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudywithrai
n));
                                                        } else if
(Description_array[0].equals("多雲時晴短暫陣雨") || Description_array[0].equals("多
雲時晴短暫雨") || Description_array[0].equals("晴時多雲短暫陣雨") ||
Description_array[0].equals("晴短暫陣雨") || Description_array[0].equals("晴午後陰
短暫雨") || Description_array[0].equals("晴午後陰短暫陣雨") ||
Description_array[0].equals("晴時多雲陣雨") || Description_array[0].equals("多雲時
晴陣雨")) {

TUE_image.setImageDrawable(getResources().getDrawable(R.drawable.afternoonclou
dywithshowers));
                                                        } else if
```

```java
(Description_array[0].contains("雷") && Description_array[0].contains("雨") ||
Description_array[0].contains("雷")) {

TUE_image.setImageDrawable(getResources().getDrawable(R.drawable.thunderstorms
));
                                        } else if
(Description_array[0].contains("霧")) {

TUE_image.setImageDrawable(getResources().getDrawable(R.drawable.fog));
                                        } else if
(Description_array[0].contains("雪")) {

TUE_image.setImageDrawable(getResources().getDrawable(R.drawable.snow));
                                        }
                                        break;

                                case "三":
                                        if
(Description_array[0].equals("晴")) {

WED_image.setImageDrawable(getResources().getDrawable(R.drawable.sun));
                                        } else if
(Description_array[0].equals("晴時多雲") || Description_array[0].equals("多雲時晴"))
{

WED_image.setImageDrawable(getResources().getDrawable(R.drawable.sunpartlyclea
r));
                                        } else if
(Description_array[0].equals("多雲") || Description_array[0].equals("多雲時陰") ||
Description_array[0].equals("陰時多雲")) {

WED_image.setImageDrawable(getResources().getDrawable(R.drawable.mostlycloudy
));
```

```java
                                                        } else if
(Description_array[0].contains("陰")) {

WED_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudy));
                                                        } else if
(Description_array[0].equals("多雲陣雨") || Description_array[0].equals("多雲短暫雨
") || Description_array[0].equals("多雲短暫陣雨") || Description_array[0].equals("多雲
時陰短暫雨") || Description_array[0].equals("多雲時陰短暫陣雨") ||
Description_array[0].equals("陰時多雲短暫雨") || Description_array[0].equals("陰時
多雲短暫陣雨") || Description_array[0].equals("陰時多雲有雨") ||
Description_array[0].equals("陰時多雲有陣雨") || Description_array[0].equals("陰時
多雲陣雨")) {

WED_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudywithra
in));
                                                        } else if
(Description_array[0].equals("多雲時晴短暫陣雨") || Description_array[0].equals("多
雲時晴短暫雨") || Description_array[0].equals("晴時多雲短暫陣雨") ||
Description_array[0].equals("晴短暫陣雨") || Description_array[0].equals("晴午後陰
短暫雨") || Description_array[0].equals("晴午後陰短暫陣雨") ||
Description_array[0].equals("晴時多雲陣雨") || Description_array[0].equals("多雲時
晴陣雨")) {

WED_image.setImageDrawable(getResources().getDrawable(R.drawable.afternoonclo
udywithshowers));
                                                        } else if
```

```java
(Description_array[0].contains("雷") && Description_array[0].contains("雨") ||
Description_array[0].contains("雷")) {

WED_image.setImageDrawable(getResources().getDrawable(R.drawable.thunderstorms));
                                                } else if
(Description_array[0].contains("霧")) {

WED_image.setImageDrawable(getResources().getDrawable(R.drawable.fog));
                                                } else if
(Description_array[0].contains("雪")) {

WED_image.setImageDrawable(getResources().getDrawable(R.drawable.snow));
                                                }
                                                break;
                                        case "四":
                                                if
(Description_array[0].equals("晴")) {

THU_image.setImageDrawable(getResources().getDrawable(R.drawable.sun));
                                                } else if
(Description_array[0].equals("晴時多雲") || Description_array[0].equals("多雲時晴"))
{

THU_image.setImageDrawable(getResources().getDrawable(R.drawable.sunpartlyclear));
                                                } else if
(Description_array[0].equals("多雲") || Description_array[0].equals("多雲時陰") ||
Description_array[0].equals("陰時多雲")) {

THU_image.setImageDrawable(getResources().getDrawable(R.drawable.mostlycloudy));
```

```java
                                        } else if
(Description_array[0].contains("陰")) {

THU_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudy));
                                        } else if
(Description_array[0].equals("多雲陣雨") || Description_array[0].equals("多雲短暫雨
") || Description_array[0].equals("多雲短暫陣雨") || Description_array[0].equals("多雲
時陰短暫雨") || Description_array[0].equals("多雲時陰短暫陣雨") ||
Description_array[0].equals("陰時多雲短暫雨") || Description_array[0].equals("陰時
多雲短暫陣雨") || Description_array[0].equals("陰時多雲有雨") ||
Description_array[0].equals("陰時多雲有陣雨") || Description_array[0].equals("陰時
多雲陣雨")) {

THU_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudywithrai
n));
                                        } else if
(Description_array[0].equals("多雲時晴短暫陣雨") || Description_array[0].equals("多
雲時晴短暫雨") || Description_array[0].equals("晴時多雲短暫陣雨") ||
Description_array[0].equals("晴短暫陣雨") || Description_array[0].equals("晴午後陰
短暫雨") || Description_array[0].equals("晴午後陰短暫陣雨") ||
Description_array[0].equals("晴時多雲陣雨") || Description_array[0].equals("多雲時
晴陣雨")) {

THU_image.setImageDrawable(getResources().getDrawable(R.drawable.afternooclou
dywithshowers));
                                        } else if
```

```java
(Description_array[0].contains("雷") && Description_array[0].contains("雨") ||
Description_array[0].contains("雷")) {

THU_image.setImageDrawable(getResources().getDrawable(R.drawable.thunderstorms
));
                                            } else if
(Description_array[0].contains("霧")) {

THU_image.setImageDrawable(getResources().getDrawable(R.drawable.fog));
                                            } else if
(Description_array[0].contains("雪")) {

THU_image.setImageDrawable(getResources().getDrawable(R.drawable.snow));
                                            }
                                            break;

                                        case "五":
                                            if
(Description_array[0].equals("晴")) {

FRI_image.setImageDrawable(getResources().getDrawable(R.drawable.sun));
                                            } else if
(Description_array[0].equals("晴時多雲") || Description_array[0].equals("多雲時晴"))
{

FRI_image.setImageDrawable(getResources().getDrawable(R.drawable.sunpartlyclear)
);
                                            } else if
(Description_array[0].equals("多雲") || Description_array[0].equals("多雲時陰") ||
Description_array[0].equals("陰時多雲")) {

FRI_image.setImageDrawable(getResources().getDrawable(R.drawable.mostlycloudy))
;
```

```java
                                                                } else if
(Description_array[0].contains("陰")) {

FRI_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudy));
                                                                } else if
(Description_array[0].contains("短暫陣雨") || Description_array[0].contains("陣雨") ||

Description_array[0].contains("短暫雨") || Description_array[0].contains("雨")) {

FRI_image.setImageDrawable(getResources().getDrawable(R.drawable.rain));
                                                                } else if
(Description_array[0].equals("多雲陣雨") || Description_array[0].equals("多雲短暫雨

") || Description_array[0].equals("多雲短暫陣雨") || Description_array[0].equals("多雲

時陰短暫雨") || Description_array[0].equals("多雲時陰短暫陣雨") ||

Description_array[0].equals("陰時多雲短暫雨") || Description_array[0].equals("陰時

多雲短暫陣雨") || Description_array[0].equals("陰時多雲有雨") ||

Description_array[0].equals("陰時多雲有陣雨") || Description_array[0].equals("陰時

多雲陣雨")) {

FRI_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudywithrain
));
                                                                } else if
(Description_array[0].contains("晴午後") || Description_array[0].equals("多雲時晴短

暫陣雨") || Description_array[0].equals("多雲時晴短暫雨") ||

Description_array[0].equals("晴時多雲短暫陣雨") || Description_array[0].equals("晴

短暫陣雨") || Description_array[0].equals("晴午後陰短暫雨") ||

Description_array[0].equals("晴午後陰短暫陣雨") || Description_array[0].equals("晴
```

```java
時多雲陣雨") || Description_array[0].equals("多雲時晴陣雨")) {

FRI_image.setImageDrawable(getResources().getDrawable(R.drawable.afternooncloud
ywithshowers));
                                                          } else if
(Description_array[0].contains(" 雷 ") && Description_array[0].contains(" 雨 ") ||
Description_array[0].contains("雷")) {

FRI_image.setImageDrawable(getResources().getDrawable(R.drawable.thunderstorms)
);
                                                          } else if
(Description_array[0].contains("霧")) {

FRI_image.setImageDrawable(getResources().getDrawable(R.drawable.fog));
                                                          } else if
(Description_array[0].contains("雪")) {

FRI_image.setImageDrawable(getResources().getDrawable(R.drawable.snow));
                                                          }
                                                          break;
                                                  case "六":
                                                          if
(Description_array[0].equals("晴")) {

SAT_image.setImageDrawable(getResources().getDrawable(R.drawable.sun));
                                                          } else if
(Description_array[0].equals("晴時多雲") || Description_array[0].equals("多雲時晴"))
{

SAT_image.setImageDrawable(getResources().getDrawable(R.drawable.sunpartlyclear
));
                                                          } else if
```

```java
(Description_array[0].equals("多雲") || Description_array[0].equals("多雲時陰") || Description_array[0].equals("陰時多雲")) {

SAT_image.setImageDrawable(getResources().getDrawable(R.drawable.mostlycloudy));
                                        } else if (Description_array[0].contains("陰")) {

SAT_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudy));
                                        } else if (Description_array[0].equals("多雲陣雨") || Description_array[0].equals("多雲短暫雨") || Description_array[0].equals("多雲短暫陣雨") || Description_array[0].equals("多雲時陰短暫雨") || Description_array[0].equals("多雲時陰短暫陣雨") || Description_array[0].equals("陰時多雲短暫雨") || Description_array[0].equals("陰時多雲短暫陣雨") || Description_array[0].equals("陰時多雲有雨") || Description_array[0].equals("陰時多雲有陣雨") || Description_array[0].equals("陰時多雲陣雨")) {

SAT_image.setImageDrawable(getResources().getDrawable(R.drawable.cloudywithrain));
                                        } else if (Description_array[0].equals("多雲時晴短暫陣雨") || Description_array[0].equals("多雲時晴短暫雨") || Description_array[0].equals("晴時多雲短暫陣雨") || Description_array[0].equals("晴短暫陣雨") || Description_array[0].equals("晴午後陰短暫雨") || Description_array[0].equals("晴午後陰短暫陣雨") || Description_array[0].equals("晴時多雲陣雨") || Description_array[0].equals("多雲時
```

```java
晴陣雨")) {

SAT_image.setImageDrawable(getResources().getDrawable(R.drawable.afternooncloudywithshowers));
                } else if
(Description_array[0].contains("雷") && Description_array[0].contains("雨") ||

Description_array[0].contains("雷")) {

SAT_image.setImageDrawable(getResources().getDrawable(R.drawable.thunderstorms));
                } else if
(Description_array[0].contains("霧")) {

SAT_image.setImageDrawable(getResources().getDrawable(R.drawable.fog));
                } else if
(Description_array[0].contains("雪")) {

SAT_image.setImageDrawable(getResources().getDrawable(R.drawable.snow));
                }
                break;
            }
        }
        break;
    // 降雨機率
    case "PoP12h":
        for (int k = 0; k < time.length(); k++)
{
            JSONObject jsonObject3 =
time.getJSONObject(k);
            String startTime =
jsonObject3.getString("startTime");
            String PoP_startTime =
startTime.substring(0, 10);

PoP_Day.setText(PoP_startTime);
```

```java
PoP_Day_list.add(PoP_Day.getText().toString());

jsonObject3.getJSONObject("elementValue");

elementValue.getString("value");

");

week_PoP_list.add(week_PoP.getText().toString());

(getWeek(PoP_startTime)) {

SUN_PoP.setText(value + " % ");

MON_PoP.setText(value + " % ");

TUE_PoP.setText(value + " % ");

WED_PoP.setText(value + " % ");

THU_PoP.setText(value + " % ");

JSONObject   elementValue   =

String          value          =

if (value == "null") {
    value = "0";
}
week_PoP.setText(value  +  "  %
");

switch

        case "天":

            break;
        case "一":

            break;
        case "二":

            break;
        case "三":

            break;
        case "四":

            break;
```

```java
                                        case "五":
                                            FRI_PoP.setText(value + " % ");
                                            break;
                                        case "六":

SAT_PoP.setText(value + " % ");

                                            break;
                            }
                        }
                        break;
                    case "MaxT":
                        for (int k = 0; k < time.length(); k++)
{
                            JSONObject jsonObject3 =
time.getJSONObject(k);
                            String startTime =
jsonObject3.getString("startTime");
                            String MaxT_startTime =
startTime.substring(0, 10);

MaxT_Hour.setText(startTime.substring(11, 13));

MaxT_Hour_list.add(MaxT_Hour.getText().toString());

MaxT_Day.setText(MaxT_startTime);

MaxT_Day_list.add(MaxT_Day.getText().toString());
                            JSONObject elementValue =
jsonObject3.getJSONObject("elementValue");
                            String value =
elementValue.getString("value");
                            switch
(getWeek(MaxT_startTime)) {
                                case "天":

SUN_Max.setText(value + " °C ");
                                    break;
```

```java
                                                  case "一":
MON_Max.setText(value + " °C ");
                                                    break;
                                                  case "二":
TUE_Max.setText(value + " °C ");
                                                    break;
                                                  case "三":
WED_Max.setText(value + " °C ");
                                                    break;
                                                  case "四":
THU_Max.setText(value + " °C ");
                                                    break;
                                                  case "五":
FRI_Max.setText(value + " °C ");
                                                    break;
                                                  case "六":
SAT_Max.setText(value + " °C ");
                                                    break;
                                                  }
HighTemperature.setText(value);

HighTemperature_list.add(HighTemperature.getText().toString());
                                                  }
                                                break;
                                            case "MinT":
                                                for (int k = 0; k < time.length(); k++)
{
                                                    JSONObject    jsonObject3    =
```

```java
time.getJSONObject(k);
String startTime = jsonObject3.getString("startTime");
String MinT_startTime = startTime.substring(0, 10);
MinT_Day.setText(MinT_startTime);
MinT_Day_list.add(MinT_Day.getText().toString());
JSONObject elementValue = jsonObject3.getJSONObject("elementValue");
String value = elementValue.getString("value");
switch (getWeek(MinT_startTime)) {
    case "天":
        SUN_Min.setText(value + " °C ");
        break;
    case "一":
        MON_Min.setText(value + " °C ");
        break;
    case "二":
        TUE_Min.setText(value + " °C ");
        break;
    case "三":
        WED_Min.setText(value + " °C ");
        break;
    case "四":
        THU_Min.setText(value + " °C ");
        break;
```

```java
                                                    case "五":

FRI_Min.setText(value + " °C ");

                                                        break;

                                                    case "六":

SAT_Min.setText(value + " °C ");

                                                        break;
                                                }
                                            LowTemperature.setText(value);

LowTemperature_list.add(LowTemperature.getText().toString());
                                        }
                                    break;
                                }
                            }
                        }
                        CompleteAdd(citycount);
                    }
                }
                dbcount = 1;
                setTemperatureText();
            }catch(JSONException e) {
                e.printStackTrace();
            }
        }
    }
    public void setTemperatureText(){
        Integer                          current_Time                          =
Integer.valueOf(Today_Time.getText().toString()).intValue();
        String S_current_Time = "";
        if(current_Time >= 6 && current_Time < 18){

if(dao_Week.getWDweather(Location.getText().toString()).get(0).getHour().equals("12
"))
                S_current_Time = "12";
            else
                S_current_Time = "06";
        }else{
```

```java
if(dao_Week.getWDweather(Location.getText().toString()).get(0).getHour().equals("00
"))
                    S_current_Time = "00";
                else
                    S_current_Time = "18";
        }
        for(int          y          =          0          ;          y          <
dao_Week.getWDweather(Location.getText().toString()).size() ; y++) {
            if
(dao_Week.getWDweather(Location.getText().toString()).get(y).getDay().equals(dao_
Week.getWDweather(Location.getText().toString()).get(y).getMaxT_Day()) &&

dao_Week.getWDweather(Location.getText().toString()).get(y).getHour().equals(S_cur
rent_Time)) {
                LowTemperature.setText("L"                                    +
dao_Week.getWDweather(Location.getText().toString()).get(y).getLowTemperature()
+ "°");
                HighTemperature.setText("H"                                   +
dao_Week.getWDweather(Location.getText().toString()).get(y).getHighTemperature()
+ "°");
                break;
            }
        }
    }
    private String getWeek(String pTime) {
        String Week = "";
        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
        Calendar c = Calendar.getInstance();
        try {
            c.setTime(format.parse(pTime));
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        if (c.get(Calendar.DAY_OF_WEEK) == 1) {

            Week += "天";

        }if (c.get(Calendar.DAY_OF_WEEK) == 2) {

            Week += "一";
```

```java
        }if (c.get(Calendar.DAY_OF_WEEK) == 3) {

            Week += "二";

        }if (c.get(Calendar.DAY_OF_WEEK) == 4) {

            Week += "三";

        }if (c.get(Calendar.DAY_OF_WEEK) == 5) {

            Week += "四";

        }if (c.get(Calendar.DAY_OF_WEEK) == 6) {

            Week += "五";

        }if (c.get(Calendar.DAY_OF_WEEK) == 7) {

            Week += "六";

        }
        return Week;
    }

    //整理進入 DB

    public void CompleteAdd(int citycount){
        getWeekWeather.setDay(Today_date.getText().toString());
        getWeekWeather.setCityName(locationList.get(citycount));
        for (int a = 0; a < WD_Day_list.size(); a++) {
            getWeekWeather.setWD_Day(WD_Day_list.get(a));

getWeekWeather.setWeatherDescription(WeatherDescription_array_list.get(a));
            getWeekWeather.setPoP_Day(PoP_Day_list.get(a));
            getWeekWeather.setPoPh(week_PoP_list.get(a));
            getWeekWeather.setMaxT_Day(MaxT_Day_list.get(a));
            getWeekWeather.setHour(MaxT_Hour_list.get(a));
            getWeekWeather.setHighTemperature(HighTemperature_list.get(a));
            getWeekWeather.setMinT_Day(MinT_Day_list.get(a));
            getWeekWeather.setLowTemperature(LowTemperature_list.get(a));
            if(dao_Week.getCount() != 330){
                dao_Week.insert(getWeekWeather);
            }else{
                getWeekWeather.setId(dbcount);
                dao_Week.update(getWeekWeather);
                dbcount++;
            }
```

```java
        }
        WD_Day_list = new ArrayList<String>();
        WeatherDescription_array_list = new ArrayList<String>();
        PoP_Day_list = new ArrayList<String>();
        week_PoP_list = new ArrayList<String>();
        MaxT_Day_list = new ArrayList<String>();
        HighTemperature_list = new ArrayList<String>();
        MinT_Day_list = new ArrayList<String>();
        LowTemperature_list = new ArrayList<String>();
        MaxT_Hour_list = new ArrayList<String>();
    }

    // 回到主頁按鈕

    public void toHome(View view) {
        Intent intent = new Intent();
        intent.setClass( weather.this   , MainActivity.class);
        startActivity(intent);
        weather.this.finish();
    }

    // 重新整理按鈕

    public void reLoad(View view) {
        Intent intent=new Intent(this, weather.class);
        startActivity(intent);

        finish(); // 關閉此檔案

        overridePendingTransition(0, 0);
    }
}
```

程式名稱	Viewclothes_CustomBitmap
目的	給予推薦穿衣照片
程式碼	

```java
package com.example.viewclothes;

import android.graphics.Bitmap;
import android.graphics.Matrix;
import android.graphics.PointF;
public class CustomBitmap {
    private long id;
    public float startDis;
    public PointF midPoint;
    public float oldRotation = 0;
    public float rotation = 0;
    public PointF startPoint = new PointF();
    public Matrix matrix = new Matrix();
    public Bitmap bitmap;
    public CustomBitmap(Bitmap bitmap) {
        this.bitmap = bitmap;
    }
    public Bitmap getBitmap() {
        return bitmap;
    }
    public long getId() {
        return id;
    }
    public void setId(long id) {
        this.id = id;
    }
    public void setMatrix(Matrix matrix) {
        this.matrix = matrix;
    }
}
```

程式名稱	Viewclothes_DrawingView
目的	調整推薦照片
程式碼	

```java
package com.example.viewclothes;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Matrix;
import android.graphics.Paint;
import android.graphics.PointF;
import android.graphics.Rect;
import android.util.AttributeSet;
import android.view.MotionEvent;
import android.view.View;
import java.util.ArrayList;
import java.util.List;

public class DrawingView extends View{

    private List<CustomBitmap> _bitmaps;
    private Context _context;
    private CustomBitmap _curCustomBitmap;
    private viewclothes viewClothes;
    private Matrix currentMatrix = new Matrix();
    private enum MODE {
        NONE, DRAG, ZOOM
    };
    private MODE mode = MODE.NONE;
    public DrawingView(Context context, AttributeSet attrs) {
        super(context,attrs);
        this._context = context;
        _bitmaps = new ArrayList<>();
    }

    public void addBitmap(CustomBitmap bitmap){
        _bitmaps.add(bitmap);
    }
    public List<CustomBitmap> getViews(){
        return _bitmaps;
    }
}
```

```java
    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        Paint paint = new Paint();
        paint.setStyle(Paint.Style.FILL);
        paint.setAntiAlias(true);
        for(CustomBitmap bitmap:_bitmaps){
            canvas.drawBitmap(bitmap.getBitmap(), bitmap.matrix, paint);
        }
    }
    public float distance(MotionEvent event) {
        float dx = event.getX(1) - event.getX(0);
        float dy = event.getY(1) - event.getY(0);
        return (float) Math.sqrt(dx * dx + dy * dy);
    }
    public PointF mid(MotionEvent event) {
        float midX = (event.getX(1) + event.getX(0)) / 2;
        float midY = (event.getY(1) + event.getY(0)) / 2;
        return new PointF(midX, midY);
    }
    public float rotation(MotionEvent event) {
        double delta_x = (event.getX(0) - event.getX(1));
        double delta_y = (event.getY(0) - event.getY(1));
        double radians = Math.atan2(delta_y, delta_x);
        return (float) Math.toDegrees(radians);
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        switch (event.getAction() & MotionEvent.ACTION_MASK) {

            case MotionEvent.ACTION_DOWN:// 手指压下屏幕

                mode = MODE.DRAG;
                if(_curCustomBitmap == null && _bitmaps.size() > 0){
                    _curCustomBitmap = _bitmaps.get(_bitmaps.size()-1);
                }

                boolean isChanged = false;// 当前操作 bitmap 是否改变

                for(CustomBitmap bitmap:_bitmaps){
                    float[] values = new float[9];
                    bitmap.matrix.getValues(values);
```

```java
                    float globalX = values[Matrix.MTRANS_X];
                    float globalY = values[Matrix.MTRANS_Y];
                    float                    width                    =
values[Matrix.MSCALE_X]*bitmap.getBitmap().getWidth();
                    float                    height                    =
values[Matrix.MSCALE_Y]*bitmap.getBitmap().getWidth();
                    Rect    rect    =    new    Rect((int)globalX,    (int)globalY,
(int)(globalX+width), (int)(globalY+height));
                    if(rect.contains((int)event.getX(), (int)event.getY())){
                        _curCustomBitmap = bitmap;
                        isChanged = true;
                    }
                }

                //切换操作对象，只要把这个对象添加到栈底就行

                if(isChanged){
                    _bitmaps.remove(_curCustomBitmap);
                    _bitmaps.add(_curCustomBitmap);
                }

                currentMatrix.set(_curCustomBitmap.matrix);// 记录 ImageView
```

当前的移动位置

```java
                _curCustomBitmap.matrix.set(currentMatrix);
                _curCustomBitmap.startPoint.set(event.getX(), event.getY());
                postInvalidate();
                break;


            case MotionEvent.ACTION_POINTER_DOWN:// 当屏幕上还有触
```

点（手指），再有一个手指压下屏幕

```java
                mode = MODE.ZOOM;
                _curCustomBitmap.oldRotation = rotation(event);
                _curCustomBitmap.startDis = distance(event);
                if (_curCustomBitmap.startDis > 10f) {
                    _curCustomBitmap.midPoint = mid(event);

                    currentMatrix.set(_curCustomBitmap.matrix);//    记    录
```

ImageView 当前的缩放倍数

```java
                    }
                    break;

            case MotionEvent.ACTION_MOVE:// 手指在屏幕移动，该 事件会
不断地触发

                    if (mode == MODE.DRAG) {
                        float dx = event.getX() - _curCustomBitmap.startPoint.x;//
得到在 x 轴的移动距离
                        float dy = event.getY() - _curCustomBitmap.startPoint.y;//
得到在 y 轴的移动距离

                        _curCustomBitmap.matrix.set(currentMatrix);// 在 没 有 进
行移动之前的位置基础上进行移动
                        _curCustomBitmap.matrix.postTranslate(dx, dy);
                    } else if (mode == MODE.ZOOM) {// 缩放与旋转

                        float endDis = distance(event);// 结束距离
                        _curCustomBitmap.rotation        =        rotation(event)      -
_curCustomBitmap.oldRotation;
                        if (endDis > 10f) {
                            float scale = endDis / _curCustomBitmap.startDis;// 得
到缩放倍数

                            _curCustomBitmap.matrix.set(currentMatrix);
                            _curCustomBitmap.matrix.postScale(scale,        scale,
_curCustomBitmap.midPoint.x, _curCustomBitmap.midPoint.y);

_curCustomBitmap.matrix.postRotate(_curCustomBitmap.rotation,
_curCustomBitmap.midPoint.x, _curCustomBitmap.midPoint.y);
                        }
                    }
                    break;
```

```
                case MotionEvent.ACTION_UP:// 手指离开屏

                    break;

                case  MotionEvent.ACTION_POINTER_UP:// 有手指离开屏幕,但屏
幕还有触点（手指）

                        mode = MODE.NONE;
                        break;
            }
            invalidate();
            return true;
        }
        public void delectpic(){
            _bitmaps.remove(_curCustomBitmap);
            viewClothes.wearlist.remove(_curCustomBitmap.getId());
            postInvalidate();
        }
        public void refresh(){
            postInvalidate();
        }

}
```

程式名稱	Viewclothes_MemberAdapter
目的	預覽穿衣中類別細項的適配器
程式碼	

```java
package com.example.viewclothes;
import android.content.Context;
import android.graphics.BitmapFactory;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import com.example.clothes.R;
import com.example.clothes.database.clothesDAO;
import com.example.clothes.database.getClothesMember;
import java.util.ArrayList;
public class MemberAdapter extends
RecyclerView.Adapter<MemberAdapter.ViewHolder>{
    private Context context;
    private ArrayList<getClothesMember> ClothesList;
    private LayoutInflater inflater;

    // 宣告資料庫功能類別欄位變數

    private clothesDAO dao;
    public MemberAdapter(Context context, ArrayList<getClothesMember>
ClothesList) {
        this.context = context;
        this.ClothesList = ClothesList;
        dao = new clothesDAO(context);
        inflater = LayoutInflater.from(context);
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        //View itemView = View.inflate(context, R.layout.cardviewlayout,null);
        View                    itemView                    =
inflater.inflate( R.layout.cardviewlayout_view,parent,false);
        return new ViewHolder(itemView);
    }
```

```java
        @Override
        public void onBindViewHolder(ViewHolder holder, int position) {
                getClothesMember member = ClothesList.get(position);

holder.imageId.setImageBitmap( BitmapFactory.decodeFile(member.getImgPath()));
        }
        @Override
        public int getItemCount() {
                return ClothesList.size();
        }
```

//Adapter 需要一個 ViewHolder，只要實作它的 constructor 就好，保存起

來的 view 會放在 itemView 裡面

```java
    class ViewHolder extends RecyclerView.ViewHolder{
        private ImageView imageId;
        public ViewHolder(View itemView) {
                super(itemView);
                imageId = (ImageView) itemView.findViewById(R.id.imageId);

                //在 adapter 中设置点击事件(長按)

                itemView.setOnLongClickListener(new   View.OnLongClickListener()
{

                        @Override
                        public boolean onLongClick(View v) {
                                if(onItemLongClickListener!=null){
                                        onItemLongClickListener.OnItemLongClick(v,
ClothesList.get(getLayoutPosition()));
                                }
                                return false;
                        }
                });

                //在 adapter 中设置点击事件(短按)

                itemView.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                                if(onItemClickListener!=null){
                                        onItemClickListener.OnItemClick(v,
ClothesList.get(getLayoutPosition()));
                                }
```

```java
                    return;
                }
            });
        }
    }

    //设置 item 的监听事件的接口(長)

    public interface OnItemLongClickListener {
        public void OnItemLongClick(View view, getClothesMember data);
    }
    private OnItemLongClickListener onItemLongClickListener;
    public        void        setOnItemLongClickListener(OnItemLongClickListener
onItemLongClickListener) {
        this.onItemLongClickListener = onItemLongClickListener;
    }

    //设置 item 的监听事件的接口(短)

    public interface OnItemClickListener {
        public void OnItemClick(View view, getClothesMember data);
    }
    private OnItemClickListener onItemClickListener;
    public void setOnItemClickListener(OnItemClickListener onItemClickListener)
{
        this.onItemClickListener = onItemClickListener;
    }
}
```

程式名稱	Viewclothes_PagerAdapter
目的	預覽穿衣中類別適配器

程式碼

```java
package com.example.viewclothes;

import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import com.example.viewclothes.fragment.clothes1Fragment;
import com.example.viewclothes.fragment.clothes2Fragment;
import com.example.viewclothes.fragment.clothes3Fragment;
import com.example.viewclothes.fragment.clothes4Fragment;
import com.example.viewclothes.fragment.clothes5Fragment;
import com.example.viewclothes.fragment.clothes6Fragment;
import com.example.viewclothes.fragment.clothes7Fragment;
import com.example.viewclothes.fragment.clothes8Fragment;
import java.util.List;
public class PagerAdapter extends FragmentPagerAdapter {
    private List<Fragment> fragmentList;
    public PagerAdapter(FragmentManager fm, List<Fragment> fragmentList) {
        super(fm);
        this.fragmentList = fragmentList;
    }
    @Override
    public Fragment getItem(int position) {
        switch(position)
        {
            case 0:
                clothes1Fragment tab1=new clothes1Fragment();
                return tab1;
            case 1:
                clothes2Fragment tab2=new clothes2Fragment();
                return tab2;
            case 2:
                clothes3Fragment tab3=new clothes3Fragment();
                return tab3;
            case 3:
                clothes4Fragment tab4=new clothes4Fragment();
                return tab4;
```

```java
                    case 4:
                        clothes5Fragment tab5=new clothes5Fragment();
                        return tab5;
                    case 5:
                        clothes6Fragment tab6=new clothes6Fragment();
                        return tab6;
                    case 6:
                        clothes7Fragment tab7=new clothes7Fragment();
                        return tab7;
                    case 7:
                        clothes8Fragment tab8=new clothes8Fragment();
                        return tab8;
            }
            return null;
        }
        @Override
        public int getCount() {
            return fragmentList.size();
        }
    }
}
```

程式名稱	viewclothes
目的	顯示穿衣推薦
程式碼	

```java
import android.content.Intent;
import android.content.SharedPreferences;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Matri
import android.support.design.widget.TabLayout;
import android.support.v4.app.Fragment;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import com.example.clothes.MainActivity;
import com.example.clothes.R;
import com.example.clothes.database.clothesDAO;
import com.example.clothes.database.getClothesMember;
import com.example.clothes.database.weatherDAO;
import com.example.viewclothes.fragment.clothes1Fragment;
import com.example.viewclothes.fragment.clothes2Fragment;
import com.example.viewclothes.fragment.clothes3Fragment;
import com.example.viewclothes.fragment.clothes4Fragment;
import com.example.viewclothes.fragment.clothes5Fragment;
import com.example.viewclothes.fragment.clothes6Fragment;
import com.example.viewclothes.fragment.clothes7Fragment;
import com.example.viewclothes.fragment.clothes8Fragment;

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

import java.text.SimpleDateFormat;
import java.util.ArrayList;
```

```java
import java.util.Date;
import java.util.HashSet;
import java.util.List;
import java.util.Locale;
import java.util.Set;
public class viewclothes extends AppCompatActivity {

    // 宣告資料庫功能類別欄位變數

    private static clothesDAO dao;
    private static weatherDAO weatherDAO;
    private static DrawingView mView;
    private ViewPager myViewPager;
    private TabLayout tabLayout;
    private int[] IconResID = {R.drawable.selector_one, R.drawable.selector_two,
R.drawable.selector_three,
            R.drawable.selector_four,                        R.drawable.selector_five,
R.drawable.selector_six,
            R.drawable.selector_seven, R.drawable.selector_eight};
    private TextView hinttext;
    public static ArrayList wearlist = new ArrayList<>();
    Integer PoP,Tem;
    Set<String> wearSet = new HashSet<String>();
    SharedPreferences WEAR;

    //定義 getClothesMember 的集合(上衣下衣)

    private ArrayList<getClothesMember> UPclothes ,DOWNclothes;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_viewclothes);
        mView = findViewById(R.id.drawingView);
        dao = new clothesDAO(getApplicationContext());
        weatherDAO = new weatherDAO(getApplicationContext());
        WEAR = getSharedPreferences("WEAR", MODE_PRIVATE);
        initdata();
        for(int i=0 ;i < wearlist.size() ; i++){
            getpicture((Long) wearlist.get(i),getApplicationContext());
        }
        processView();
    }
    public void initdata(){
```

```java
//抓取天氣 db
try {
    Tem                                                            =
Integer.valueOf(weatherDAO.getWDweather(weatherDAO.getoneID(1).getNowCity())
.get(0).getTemperature());
    }catch (StringIndexOutOfBoundsException e){

        Toast.makeText(  viewclothes.this,  "獲取天氣資料錯誤" ,
Toast.LENGTH_LONG).show();
    }
    try {
        PoP                                                        =
Integer.valueOf(weatherDAO.getWDweather(weatherDAO.getoneID(1).getNowCity())
.get(0).getPoPh());
    }catch (StringIndexOutOfBoundsException e){

        Toast.makeText(  viewclothes.this,  "獲取天氣資料錯誤" ,
Toast.LENGTH_LONG).show();
    }
    //抓取 db 資料

    int temp = 0;
    do{
        UPclothes = dao.getUPclothes(Tem+temp , Tem-temp);
        temp++;
        if(temp == 200){

            Toast.makeText(  viewclothes.this, "請先至少新增一件上衣" ,
Toast.LENGTH_LONG).show();
            break;
        }
    }while (UPclothes ==null);
    temp = 0;
    do {
        DOWNclothes = dao.getDOWNclothes(Tem+temp , Tem-temp);
        temp++;
        if(temp == 200){

            Toast.makeText(  viewclothes.this, "請先至少新增一件下衣" ,
Toast.LENGTH_LONG).show();
```

```java
                    break;
                }
        }while (DOWNclothes == null);
        SharedPreferences        DATE        =        getSharedPreferences("DATE",
Context.MODE_PRIVATE);
        String        timeStamp        =        new        SimpleDateFormat("yyyyMMdd",
Locale.getDefault()).format(new Date());

        if(!timeStamp.equals(DATE.getString("todayis",null)) ||
                Tem != DATE.getInt("todayTem",0) ||
                !String.valueOf(dao.getlastID().get(0).getId()).equals(DATE.getString("Lastclothes",null)) ||
                dao.getUPCount() != DATE.getInt("getUPCount",0) ||
                dao.getDOWNCount() != DATE.getInt("getDOWNCount",0)){

            //修改成今天日期跟現在溫度

            DATE.edit().clear();
            DATE.edit().remove("todayis");
            DATE.edit().remove("Lastclothes");
            DATE.edit().remove("todayTem");
            DATE.edit().remove("getUPCount");
            DATE.edit().remove("getDOWNCount");
            DATE.edit()
                    .putString("todayis",timeStamp)
                    .putString("Lastclothes",
dao.getlastID().get(0).getId().toString())
                    .putInt("todayTem", Tem)
                    .putInt("getUPCount", dao.getUPCount())
                    .putInt("getDOWNCount", dao.getDOWNCount())
                    .commit();
            if(wearlist != null)wearlist = null;
            wearlist = new ArrayList<>();

            //決定衣服

            int index = (int)(Math.random() * UPclothes.size());

            if(!UPclothes.get(index).getType().equals("連身裙")){

                wearlist.add(UPclothes.get(index).getId());

                //花上衣+素下衣
```

```java
                    if(!UPclothes.get(index).getStyle().equals("素         色         ")
&& !UPclothes.get(index).getStyle().equals("圖案或文字")){
                        index = (int) (Math.random() * DOWNclothes.size());
                        while (    !DOWNclothes.get(index).getStyle().equals("格子")
||
                                !DOWNclothes.get(index).getStyle().equals("條紋")
||
                                !DOWNclothes.get(index).getStyle().equals("花  紋
")  )  {
                            if(DOWNclothes.size() == 1) break; //如果有很前衛的
人(花上+花下)

                            DOWNclothes.remove(index);
                            index = (int) (Math.random() * DOWNclothes.size());
                        }
                        wearlist.add(DOWNclothes.get(index).getId());
                    }else{ //素上衣+任意下衣
                        index = (int) (Math.random() * DOWNclothes.size());
                        wearlist.add(DOWNclothes.get(index).getId());
                    }
                }else{ //是連衣直接加入
                    wearlist.add(UPclothes.get(index).getId());
                }
            }else {
                if(wearlist != null)wearlist = null;
                wearlist = new ArrayList<>();
                wearSet = WEAR.getStringSet("wearSet",null);
                if(wearSet.size() > 0){
                    String[]     data     =     (String[])     wearSet.toArray(new
String[wearSet.size()]);
                        for (int i = 0; i < data.length; i++){
                            wearlist.add(Long.parseLong(data[i]));
                        }
```

```java
                }
            }
        }
    public void processView(){
        hinttext = findViewById(R.id.hint);
        try {
            if(PoP >= 50)

                hinttext.setText("降雨機率為" + PoP + "%，記得要攜帶雨具出門

喔！");

        }catch (NullPointerException e){

            hinttext.setText("抓不到資料我們深感抱歉 QQ");

        }
        myViewPager = findViewById( R.id.myViewPager);
        tabLayout = findViewById( R.id.TabLayout);
        setViewPager();
        tabLayout.setupWithViewPager(myViewPager);
        setTabLayoutIcon();
    }
    public void setTabLayoutIcon(){

        //新增 tab&icon

        for(int i =0; i < IconResID.length;i++){
            tabLayout.getTabAt(i).setIcon(IconResID[i]);
        }
    }
    private void setViewPager(){

        //新增各 Fragment 到 ViewPager----需同步更動 PagerAdapter.java

        clothes1Fragment myFragment1 = new clothes1Fragment();
        clothes2Fragment myFragment2 = new clothes2Fragment();
        clothes3Fragment myFragment3 = new clothes3Fragment();
        clothes4Fragment myFragment4 = new clothes4Fragment();
        clothes5Fragment myFragment5 = new clothes5Fragment();
        clothes6Fragment myFragment6 = new clothes6Fragment();
        clothes7Fragment myFragment7 = new clothes7Fragment();
        clothes8Fragment myFragment8 = new clothes8Fragment();
        List<Fragment> fragmentList = new ArrayList<Fragment>();
        fragmentList.add(myFragment1);
```

```java
        fragmentList.add(myFragment2);
        fragmentList.add(myFragment3);
        fragmentList.add(myFragment4);
        fragmentList.add(myFragment5);
        fragmentList.add(myFragment6);
        fragmentList.add(myFragment7);
        fragmentList.add(myFragment8);
        PagerAdapter        myFragmentAdapter        =        new
PagerAdapter(getSupportFragmentManager(), fragmentList);
        myViewPager.setAdapter(myFragmentAdapter);
    }
    public static void getpicture(long id,Context ctxt){
        getClothesMember member = dao.getoneID(id);
        Bitmap bmp= BitmapFactory.decodeFile( member.getImgPath());
        CustomBitmap customBitmap = new CustomBitmap(bmp);
        customBitmap.setId(id);
        if (getSavedMatrix(id,ctxt) != null){
            customBitmap.setMatrix(getSavedMatrix(id,ctxt));
        }
        mView.addBitmap(customBitmap);
        mView.refresh();
    }
    private void saveMatrix(CustomBitmap customBitmap){
        SharedPreferences.Editor                 editor                 =
getSharedPreferences("matrix",Context.MODE_PRIVATE).edit();
        Matrix matrix = customBitmap.matrix;
        float[] values = new float[9];
        matrix.getValues(values);
        JSONArray array = new JSONArray();
        for (float value:values){
            try {
                array.put(value);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        editor.putString(String.valueOf(customBitmap.getId()), array.toString());
        editor.commit();
    }
    //獲取 matrix
```

```java
    private static Matrix getSavedMatrix(long id ,Context ctxt){
        SharedPreferences sp = ctxt.getSharedPreferences("matrix",
Context.MODE_PRIVATE);
        String result = sp.getString(String.valueOf(id), null);
        if (result != null){
            float[] values = new float[9];
            Matrix matrix = new Matrix();
            try {
                JSONArray array = new JSONArray(result);
                for (int i = 0; i < array.length(); i++) {
                    values[i] = Float.valueOf(String.valueOf(array.getDouble(i)));
                }
                matrix.setValues(values);
            } catch (JSONException e) {
                e.printStackTrace();
            }
            return matrix ;
        }
        return null;
    }
    public void save(View view){
        List<CustomBitmap> list = mView.getViews();
        for (CustomBitmap customBitmap:list){
            saveMatrix(customBitmap);
        }
        wearSet = new HashSet<String>();
        WEAR.edit().clear();
        WEAR.edit().remove("wearSet");
        for(int i=0 ;i < wearlist.size() ; i++){
            wearSet.add(wearlist.get(i).toString());
        }
        Set<String> result = new HashSet<>(wearSet);
        WEAR.edit().putStringSet("wearSet",result).commit();
    }
    public void deletepic(View view){
        try {
            mView.delectpic();
        }catch (NullPointerException e){

            Toast.makeText(this, " 請 先 點 選 一 件 衣 服 ",
Toast.LENGTH_LONG).show();
```

```java
        }
    }

    // 回到主頁按鈕

    public void toHome(View view) {
        Intent intent = new Intent();
        intent.setClass( viewclothes.this    , MainActivity.class);
        startActivity(intent);
        viewclothes.this.finish();
    }

    // 重新整理按鈕

    public void reLoad(View view) {
        Intent intent=new Intent(this, viewclothes.class);
        startActivity(intent);

        finish(); // 關閉此檔案

        overridePendingTransition(0, 0);
    }
    @Override
    public void finish() {
        dao.close();
        weatherDAO.close();
        save(getCurrentFocus());
        super.finish();
    }

    //禁用原生返回鍵

    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if (keyCode == KeyEvent.KEYCODE_BACK) {
            return true;
        }
        return false;
    }
```

程式名稱	AddClothes
目的	使用者新增衣服照片

程式碼

```
import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.os.ParcelFileDescriptor;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.widget.Button;
import android.widget.ImageView;
import com.github.gabrielbb.cutout.CutOut;
import java.io.File;
import java.io.FileDescriptor;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
public class AddClothes extends AppCompatActivity {
    public static Long clothesID ;
    private ImageView imageView;
    Bitmap bitmap = null;
    String filepath;
    public static AddClothes finishself = null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_clothes);
        finishself = this;
        Button b = findViewById(R.id.button);
        imageView = findViewById(R.id.imageView);
        b.setOnClickListener(view -> {
            CutOut.activity()
                    .bordered()
```

```java
                    .noCrop()
                    .start(this);
        });

        if(clothesID == null){    //去新增衣服(頁面)

                Button updata = (Button) findViewById( R.id.updata);
                updata.setOnClickListener(v -> {
                    newclothes.PicPath = filepath;
                    Intent intent = new Intent ();
                    intent.setClass( AddClothes.this, newclothes.class);
                    startActivity(intent);
                });

        }else{                    //去修改衣服(頁面)

                Button updata = (Button) findViewById( R.id.updata);
                updata.setOnClickListener(v -> {
                    editclothes.PicPath = filepath;
                    editclothes.load = false;
                    Intent intent = new Intent ();
                    intent.setClass( AddClothes.this, editclothes.class);
                    startActivity(intent);
                });
        }
    }
    @Override
    protected  void  onActivityResult(int  requestCode,  int  resultCode,  @Nullable
Intent data) {
        if (requestCode == CutOut.CUTOUT_ACTIVITY_REQUEST_CODE) {
            switch (resultCode) {
                case Activity.RESULT_OK:
                    Uri Uri = CutOut.getUri(data);

                    // 存 圖 到 指 定 資 料 夾
-------------------------------------------------------------
                    try {
                        bitmap = getBitmapFromUri(Uri);
                        imageView.setImageBitmap(bitmap);
                        saveToLocal(bitmap);
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
```

```java
//---------------------------------------------------------------------------------
                            break;
                    case CutOut.CUTOUT_ACTIVITY_RESULT_ERROR_CODE:
                            Exception ex = CutOut.getError(data);
                            break;
                    default:
                            System.out.print("User cancelled the CutOut screen");
            }
        }
    }

    //1.Uri 轉成 Bitmap

    private Bitmap getBitmapFromUri(Uri uri) throws IOException {
        ParcelFileDescriptor parcelFileDescriptor =
                getContentResolver().openFileDescriptor(uri, "r");
        FileDescriptor fileDescriptor = parcelFileDescriptor.getFileDescriptor();
        Bitmap image = BitmapFactory.decodeFileDescriptor(fileDescriptor);
        parcelFileDescriptor.close();
        return image;
    }

    //2.儲存到指定資料夾

    private void saveToLocal(Bitmap bitmap) throws IOException {
        String          timeStamp     =        new         SimpleDateFormat
("yyyyMMdd_HHmmss").format(new Date ());
        String imageFileName = "clothesPNG_" + timeStamp;
        File path = new File ( Environment.getExternalStoragePublicDirectory(
                Environment.DIRECTORY_DCIM) + "/clothes");
        if (!path.exists()) {
            path.mkdir();
        }
        File file = new File(path +"/"+
                        imageFileName +
                        ".png");
        if (file.exists()) {
            file.delete();
        }
        filepath = file.toString();
        FileOutputStream out;
        try {
```

```java
                out = new FileOutputStream(file);
                if (bitmap.compress(Bitmap.CompressFormat.PNG, 90, out)) {
                    out.flush();
                    out.close();

                    //保存图片后发送广播通知更新数据库

                    Intent intent = new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);
                    Uri uri = Uri.fromFile(file);
                    intent.setData(uri);
                    this.sendBroadcast(intent);
                }
            } catch (FileNotFoundException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
}
```

程式名稱	MainActivity
目的	APP 首頁，提供主要天氣資訊與時間，可連結至各個功能頁面
程式碼	

```java
package com.example.clothes;
import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.AsyncTask;
import android.provider.Settings;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import com.example.clothes.database.clothesDAO;
import com.example.clothes.database.getWeather;
import com.example.clothes.database.weatherDAO;
import com.example.viewclothes.viewclothes;
import com.example.weather.weather;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Locale;
```

```java
public class MainActivity extends AppCompatActivity {
    // 宣告資料庫功能類別欄位變數

    private weatherDAO dao;
    private clothesDAO clothesDAO;
    getWeather getWeather = new getWeather();
    long dbcount = 1;

    // GPS 定位

    public static final int MY_PERMISSION_ACCESS_COARSE_LOCATION = 11;
    private String commadStr;
    private LocationManager locationManager;
    private int GPS_REQUEST_CODE = 10;

    public TextView CityName; // 顯示城市

    public TextView Today_Temperature; // 顯示氣溫

    public TextView TodayWeek; // 顯示星期

    public TextView date, Today_Time, Today_date, T_day, T_hour, WD_Day,
WD_Hour, PoP_Day; // 顯示日期時間

    public TextView PoP; // 顯示降雨量

    public TextView Description, threehour_Description; // 顯示天氣敘述

    public ArrayList<String> CityName_list, T_day_list, T_hour_list,
Today_Temperature_list, WD_Day_list, WD_Hour_list, Description_list,
threehour_Description_list, PoP_Day_list, PoP_list;
    String Hour_three,city;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        dao = new weatherDAO(getApplicationContext());
        clothesDAO = new clothesDAO(getApplicationContext());
        CityName_list = new ArrayList<String>();
        T_day_list = new ArrayList<String>();
        T_hour_list = new ArrayList<String>();
        Today_Temperature_list = new ArrayList<String>();
        WD_Day_list = new ArrayList<String>();
```

```java
            WD_Hour_list = new ArrayList<String>();
            Description_list = new ArrayList<String>();
            threehour_Description_list = new ArrayList<String>();
            PoP_Day_list = new ArrayList<String>();
            PoP_list = new ArrayList<String>();
            TodayWeek = (TextView) findViewById(R.id.week);
            date = (TextView) findViewById(R.id.date);
            CityName = (TextView) findViewById(R.id.CityName);
            Today_Temperature = (TextView) findViewById(R.id.C);
            PoP = (TextView) findViewById(R.id.PoP);
            Description = (TextView) findViewById(R.id.WeatherDescription);
            Today_Time = (TextView) findViewById(R.id.Today_Time);
            Today_date = (TextView) findViewById(R.id.Today_date);
            threehour_Description                     =                   (TextView)
    findViewById(R.id.threehour_Description);
            T_day = (TextView) findViewById(R.id.T_day);
            T_hour = (TextView) findViewById(R.id.T_hour);
            WD_Day = (TextView) findViewById(R.id.WD_Day);
            WD_Hour = (TextView) findViewById(R.id.WD_Hour);
            PoP_Day = (TextView) findViewById(R.id.PoP_Day);
            Today_Time.setText(new SimpleDateFormat("HH").format(new Date()));
            switch (Today_Time.getText().toString()){
                case "00":
                case "01":
                case "02":
                    Hour_three ="00";
                    break;
                case "03":
                case "04":
                case "05":
                    Hour_three ="03";
                    break;
                case "06":
                case "07":
                case "08":
                    Hour_three ="06";
                    break;
                case "09":
                case "10":
                case "11":
                    Hour_three ="09";
```

```
                    break;
            case "12":
            case "13":
            case "14":
                Hour_three ="12";
                break;
            case "15":
            case "16":
            case "17":
                Hour_three ="15";
                break;
            case "18":
            case "19":
            case "20":
                Hour_three ="18";
                break;
            case "21":
            case "22":
            case "23":
                Hour_three ="21";
                break;
        }
        Today_Time.setText(Hour_three);
        Today_date.setText(new        SimpleDateFormat("yyyy-MM-dd").format(new
Date()));
        date.setText(new SimpleDateFormat("yyyy / MM / dd").format(new Date()));

        // 獲取當前系統星期

        long time = System.currentTimeMillis();
        Date date = new Date(time);
        SimpleDateFormat format = new SimpleDateFormat("EEEE");
        TodayWeek.setText(format.format(date));

        // 鄉鎮天氣預報-臺灣未來 2 天天氣預報

        new                                              TodayTask().execute
(   "https://opendata.cwb.gov.tw/fileapi/v1/opendataapi/F-D0047-089?Authorization=C
WB-6BB38BEE-559E-42AB-9AAD-698C12D12E22&downloadType=WEB&format=
JSON" );
        if (!GPSIsOpen()) {
            return;
        }
```

```java
        //使用 GPS 定位

        commadStr = LocationManager.GPS_PROVIDER;

        // LocationManager 可以用來獲取當前的位置，追蹤設備的移動路線

        locationManager                  =                  (LocationManager)
getSystemService(LOCATION_SERVICE);
        if                  (ActivityCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION)                                    !=
PackageManager.PERMISSION_GRANTED                                          &&
ActivityCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.ACCESS_COARSE_LOCATION)                               !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(MainActivity.this,          new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},
MY_PERMISSION_ACCESS_COARSE_LOCATION);
            return;
        }
        locationManager.requestLocationUpdates(commadStr,       1000,       0,
locationListener);
        android.location.Location                    location                    =
locationManager.getLastKnownLocation(commadStr);
        if (location != null)
            CityName.setText(getAddressByLocation(location));

        // Location.setText("經度：" + location.getLongitude() + "\n 緯度：" +
location.getLatitude());
        else

            CityName.setText("定位中");

    }

    // 判斷當前是否開啟 GPS

    private boolean GPSIsOpen() {
        boolean GPS = true;
        LocationManager                        alm                        =
(LocationManager)this.getSystemService(Context.LOCATION_SERVICE);
        if(!alm.isProviderEnabled(LocationManager.GPS_PROVIDER)) {

            Toast.makeText(this, "未開啟 GPS", Toast.LENGTH_SHORT).show();

            GPS = false;
```

```java
                    Intent                    intent                    =                    new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            startActivityForResult(intent, GPS_REQUEST_CODE);
        } else {

            Toast.makeText(this, "GPS 已開啟", Toast.LENGTH_SHORT).show();

        }
        return GPS;
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == GPS_REQUEST_CODE) {

            // 做需要做的事情，比如再次检测是否打开 GPS 了 或者定位

            GPSIsOpen();

        }
    }
    public LocationListener locationListener = new LocationListener() {

        //當地點改變時

        @Override
        public void onLocationChanged(Location location) {
        }
        // 定位狀態改變

        @Override
        public void onStatusChanged(String provider, int status, Bundle extras) { }
        // 當 GPS 或網路定位功能開啟

        @Override
        public void onProviderEnabled(String provider) { }
        // 當 GPS 或網路定位功能關閉時

        @Override
        public void onProviderDisabled(String provider) { }
    };
    // 轉成地址

    public String getAddressByLocation(Location location) {
        String returnAddress = "";
```

```java
        try {
            if (location != null) {
                Double longitude = location.getLongitude();        //取得經度

                Double latitude = location.getLatitude();         //取得緯度

                Geocoder        gc        =        new        Geocoder(this,
Locale.TRADITIONAL_CHINESE);        //地區:台灣

                //自經緯度取得地址

                List<Address>    lstAddress    =    gc.getFromLocation(latitude,
longitude, 1);
                if (!Geocoder.isPresent()){ //Since: API Level 9
                    returnAddress = "Sorry! Geocoder service not Present.";
                }
                returnAddress = lstAddress.get(0).getAddressLine(0);
                returnAddress = returnAddress.substring(5,8);
            }
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        return returnAddress;
    }
    //  取得伺服端傳來回應

    private class TodayTask extends AsyncTask<String, Void, String> {
        @Override
        protected String doInBackground(String... params) {
            StringBuilder today_sb = new StringBuilder ();
            try {
                URL url = new URL ( params[0] );
                BufferedReader    in    =    new    BufferedReader    (new
InputStreamReader( url.openStream () ) );
                String line = in.readLine ();
                while (line != null) {
                    today_sb.append ( line );
                    line = in.readLine ();
                }
            } catch (MalformedURLException e) {
```

```java
                e.printStackTrace ();
            } catch (IOException e) {
                e.printStackTrace ();
            }
            return today_sb.toString ();
        }
        protected void onPostExecute(String today_data) {
            super.onPostExecute ( today_data );
            parseJSON ( today_data );
            if(Today_Temperature.getText().equals("°C"))

Today_Temperature.setText(dao.getWDweather(city).get(0).getTemperature() + " °C
");
        }
        private void parseJSON(String week_data)    {
            getWeather getWeather = new getWeather();
            JSONObject Ob;
            try{
                Ob = new JSONObject(week_data);
                JSONObject                      locations                   =
Ob.getJSONObject("cwbopendata").getJSONObject("dataset").getJSONObject
("locations");
                JSONArray location_array = locations.getJSONArray("location");
                for (int i = 0; i < location_array .length (); i++) {
                    JSONObject JsonObject = location_array.getJSONObject(i);
                    String locationName = JsonObject.getString("locationName");
                    CityName_list.add(locationName);
                    city = CityName.getText().toString();
//                      city = "台北市";

                    switch (city){
                        case "台北市":

                            city = "臺北市";

                            break;
                        case "台中市":

                            city = "臺中市";

                            break;
```

```java
                        case "台東市":

                            city = "臺東市";

                            break;

                        case "台南市":

                            city = "臺南市";

                            break;
                    }
                    CityName.setText(city);
                }
            for (int b = 0; b < location_array .length (); b++) {
                    JSONObject JsonObject = location_array.getJSONObject(b);
                    String locationName = JsonObject.getString("locationName");
                    for (int citycount = 0; citycount < CityName_list.size();
citycount++) {

                        if (locationName.equals(CityName_list.get(citycount))) {
                            JSONArray           weatherElement           =
JsonObject.getJSONArray("weatherElement");
                            for (int j = 0; j < weatherElement.length(); j++) {
                                JSONObject           jsonObject2           =
weatherElement.getJSONObject(j);
                                String           elementName           =
jsonObject2.getString("elementName");
                                JSONArray               time               =
jsonObject2.getJSONArray("time");
                                switch (elementName) {
                                    case "T":
                                        for (int k = 0; k < time.length(); k++)
{
                                            JSONObject    jsonObject3    =
time.getJSONObject(k);
                                            String        T_date        =
jsonObject3.getString("dataTime");
                                            String      current_date      =
T_date.substring(0, 10);
                                            String        T_time        =
T_date.substring(11, 13);
                                            T_day.setText(current_date);
```

```java
                                        T_hour.setText(T_time);
T_day_list.add(T_day.getText().toString());
T_hour_list.add(T_hour.getText().toString());
                                        JSONObject TelementValue =
jsonObject3.getJSONObject("elementValue");
                                        String value =
TelementValue.getString("value");
Today_Temperature_list.add(value);
                                    }
                                    break;
                            // 天氣現象
                            case "WeatherDescription":
                                for (int k = 0; k < time.length(); k++)
{
                                        JSONObject jsonObject3 =
time.getJSONObject(k);
                                        String startTime =
jsonObject3.getString("startTime");
                                        String WD_startDate =
startTime.substring(0, 10);
WD_Day.setText(WD_startDate);
WD_Hour.setText(startTime.substring(11, 13));
WD_Day_list.add(WD_Day.getText().toString());
WD_Hour_list.add(WD_Hour.getText().toString());
                                        JSONObject WDelementValue =
jsonObject3.getJSONObject("elementValue");
                                        String value =
WDelementValue.getString("value");
threehour_Description.setText(value);
threehour_Description_list.add(threehour_Description.getText().toString());
                                        Integer WD_time =
```

```java
Integer.valueOf(startTime.substring(11, 13)).intValue();
                                          Integer        current_Time        =
Integer.valueOf(Today_Time.getText().toString()).intValue();
                                          int            quotients            =
Math.round(current_Time / 3);
                                          switch (quotients) {
                                              case 0:
                                                  current_Time = 0;
                                                  break;
                                              case 1:
                                                  current_Time = 3;
                                                  break;
                                              case 2:
                                                  current_Time = 6;
                                                  break;
                                              case 3:
                                                  current_Time = 9;
                                                  break;
                                              case 4:
                                                  current_Time = 12;
                                                  break;
                                              case 5:
                                                  current_Time = 15;
                                                  break;
                                              case 6:
                                                  current_Time = 18;
                                                  break;
                                              case 7:
                                                  current_Time = 21;
                                                  break;
                                              case 8:
                                                  current_Time = 0;
                                                  break;
                                          }
                                          Description_list.add(value);
                                          if
(WD_startDate.equals(Today_date.getText().toString())    &&    WD_time    ==
current_Time) {
                                              Description.setText(value);
                                          }
                                      }
```

```java
                                        break;
                                // 降雨機率
                                case "PoP12h":
                                        for (int k = 0; k < time.length(); k++)
{
                                                JSONObject jsonObject3 = time.getJSONObject(k);
                                                String startTime = jsonObject3.getString("startTime");
                                                String PoP_startTime = startTime.substring(0, 10);

PoP_Day.setText(PoP_startTime);

PoP_Day_list.add(PoP_Day.getText().toString());
                                                JSONObject elementValue = jsonObject3.getJSONObject("elementValue");
                                                String value = elementValue.getString("value");
                                                PoP_list.add(value);
                                                if
(PoP_startTime.equals(Today_date.getText().toString())) {
                                                        PoP.setText(value + " % ");
                                                }
                                        }
                                        break;
                                }
                            }
                        }
                        CompleteAdd(citycount);
                    }
                }
                dbcount = 1;
                for(int y = 0 ; y < dao.getWDweather(CityName.getText().toString()).size() ; y++){

if(dao.getWDweather(CityName.getText().toString()).get(0).getDay().equals(dao.getWDweather(CityName.getText().toString()).get(y).getT_Day()) &&

dao.getWDweather(CityName.getText().toString()).get(0).getHour().equals(dao.getWD
```

```java
weather(CityName.getText().toString()).get(y).getT_Hour())){

Today_Temperature.setText(dao.getWDweather(CityName.getText().toString()).get(y).
getTemperature() + " °C ");
                            }
                        }
                } catch(JSONException e) {
                        e.printStackTrace();
                }
        }
    }

    // 管理衣服

    public void Manageclothes(View view) {
        Intent intent = new Intent();
        intent.setClass( MainActivity.this    , Manageclothes.class);
        startActivity(intent);
    }

    // 預覽穿衣

    public void View_wearing(View view) {
        if(clothesDAO.getUPCount() == 0 || clothesDAO.getDOWNCount() == 0){

            Toast.makeText(  MainActivity.this, "請先新增上衣及下衣至少各一

件，才能進行預覽喔！" , Toast.LENGTH_LONG).show();

        }else {
            clothesDAO.close();
            Intent intent = new Intent();
            intent.setClass(MainActivity.this, viewclothes.class);
            startActivity(intent);
            finish();
        }
    }

    // 去天氣

    public void Weekweather(View view) {
        Intent intent = new Intent();
        intent.setClass( MainActivity.this    , weather.class);
        startActivity(intent);
    }
```

```java
    //整理進入 DB

    public void CompleteAdd(int citycount){
        getWeather.setDay(Today_date.getText().toString());
        getWeather.setHour(Today_Time.getText().toString());
        getWeather.setCityName(CityName_list.get(citycount));
        getWeather.setNowCity(city);
        for (int a = 0; a < T_day_list.size(); a++) {
            getWeather.setT_Day(T_day_list.get(a));
            getWeather.setT_Hour(T_hour_list.get(a));
            getWeather.setTemperature(Today_Temperature_list.get(a));
            getWeather.setWD_Day(WD_Day_list.get(a));
            getWeather.setWD_Hour(WD_Hour_list.get(a));
            getWeather.setWeatherDescription(Description_list.get(a));

getWeather.setThreehour_Description(threehour_Description_list.get(a));
            switch (a){
                case 0:
                case 1:
                case 2:
                case 3:
                        getWeather.setPoP_Day(PoP_Day_list.get(0));
                        getWeather.setPoPh(PoP_list.get(0));
                        break;
                case 4:
                case 5:
                case 6:
                case 7:
                        getWeather.setPoP_Day(PoP_Day_list.get(1));
                        getWeather.setPoPh(PoP_list.get(1));
                        break;
                case 8:
                case 9:
                case 10:
                case 11:
                        getWeather.setPoP_Day(PoP_Day_list.get(2));
                        getWeather.setPoPh(PoP_list.get(2));
                        break;
                case 12:
                case 13:
                case 14:
```

```java
                case 15:
                    getWeather.setPoP_Day(PoP_Day_list.get(3));
                    getWeather.setPoPh(PoP_list.get(3));
                    break;
                case 16:
                case 17:
                case 18:
                case 19:
                    getWeather.setPoP_Day(PoP_Day_list.get(4));
                    getWeather.setPoPh(PoP_list.get(4));
                    break;
                case 20:
                case 21:
                case 22:
                case 23:
                    getWeather.setPoP_Day(PoP_Day_list.get(5));
                    getWeather.setPoPh(PoP_list.get(5));
                    break;
            }
            String[] Tdaylist_array,Today_date_array;
            Integer
Tdaylist_Y ,Tdaylist_MM ,Tdaylist_DD ,Today_date_Y ,Today_date_MM ,Today_dat
e_DD;
            Tdaylist_array = T_day_list.get(0).split("-");
            Today_date_array =Today_date.getText().toString().split("-");
            Tdaylist_Y = Integer.valueOf(Tdaylist_array[0]);
            Tdaylist_MM = Integer.valueOf(Tdaylist_array[1]);
            Tdaylist_DD = Integer.valueOf(Tdaylist_array[2]);
            Today_date_Y = Integer.valueOf(Today_date_array[0]);
            Today_date_MM = Integer.valueOf(Today_date_array[1]);
            Today_date_DD = Integer.valueOf(Today_date_array[2]);
            if(dao.getCount() != 528){
                dao.insert(getWeather);
            }else {
                if(Today_date_Y  >=  Tdaylist_Y  &&  Today_date_MM  >=
Tdaylist_MM && Today_date_DD >= Tdaylist_DD){
                    if(Integer.valueOf(Hour_three)                      >=
Integer.valueOf(T_hour_list.get(0))){
                        getWeather.setId(dbcount);
                        dao.update(getWeather);
                        dbcount++;
```

```
                }
            }
        }
    }
    T_day_list = new ArrayList<String>();
    T_hour_list = new ArrayList<String>();
    Today_Temperature_list = new ArrayList<String>();
    WD_Day_list = new ArrayList<String>();
    WD_Hour_list = new ArrayList<String>();
    Description_list = new ArrayList<String>();
    threehour_Description_list = new ArrayList<String>();
    PoP_Day_list = new ArrayList<String>();
    PoP_list = new ArrayList<String>();
}

// 回到主頁按鈕

public void toHome(View view) {
    Intent intent = new Intent();
    intent.setClass( MainActivity.this    , MainActivity.class);
    startActivity(intent);
    MainActivity.this.finish();
}

// 重新整理按鈕

public void reLoad(View view) {
    Intent intent=new Intent(this, MainActivity.class);
    startActivity(intent);

    finish(); // 關閉此檔案

    overridePendingTransition(0, 0);
}
}
```

程式名稱	Manageclothes
目的	管理衣服，可執行新增、修改衣服

程式碼

```java
package com.example.clothes;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.TabLayout;
import android.support.v4.app.Fragment;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.view.KeyEvent;
import android.view.View;
import android.widget.ImageButton;
import com.example.clothes.fragment.clothes1Fragment;
import com.example.clothes.fragment.clothes2Fragment;
import com.example.clothes.fragment.clothes3Fragment;
import com.example.clothes.fragment.clothes4Fragment;
import com.example.clothes.fragment.clothes5Fragment;
import com.example.clothes.fragment.clothes6Fragment;
import com.example.clothes.fragment.clothes7Fragment;
import com.example.clothes.fragment.clothes8Fragment;
import java.util.ArrayList;
import java.util.List;
public class Manageclothes extends AppCompatActivity {
    private ViewPager myViewPager;
    private TabLayout tabLayout;
    private int[] IconResID = {R.drawable.selector_one, R.drawable.selector_two,
R.drawable.selector_three,
            R.drawable.selector_four,                     R.drawable.selector_five,
R.drawable.selector_six,
            R.drawable.selector_seven, R.drawable.selector_eight};

    //在別的 activity 中關閉自己的方法

    public static Manageclothes finishself = null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView( R.layout.activity_manageclothes);
        finishself = this;
```

```java
        //去新增衣服

        ImageButton addclothes = (ImageButton)findViewById( R.id.Addclothes);
        addclothes.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent ();
                intent.setClass( Manageclothes.this    , newclothes.class);
                startActivity(intent);
            }
        });
        processView();
    }
    public void processView(){
        myViewPager = (ViewPager) findViewById( R.id.myViewPager);
        tabLayout = (TabLayout) findViewById( R.id.TabLayout);
        setViewPager();
        tabLayout.setupWithViewPager(myViewPager);
        setTabLayoutIcon();
    }
    public void setTabLayoutIcon(){

        //新增 tab&icon

        for(int i =0; i < IconResID.length;i++){
            tabLayout.getTabAt(i).setIcon(IconResID[i]);
        }
    }
    private void setViewPager(){

        //新增各 Fragment 到 ViewPager----需同步更動 PagerAdapter.java

        clothes1Fragment myFragment1 = new clothes1Fragment();
        clothes2Fragment myFragment2 = new clothes2Fragment();
        clothes3Fragment myFragment3 = new clothes3Fragment();
        clothes4Fragment myFragment4 = new clothes4Fragment();
        clothes5Fragment myFragment5 = new clothes5Fragment();
        clothes6Fragment myFragment6 = new clothes6Fragment();
        clothes7Fragment myFragment7 = new clothes7Fragment();
        clothes8Fragment myFragment8 = new clothes8Fragment();
        List<Fragment> fragmentList = new ArrayList<Fragment> ();
        fragmentList.add(myFragment1);
        fragmentList.add(myFragment2);
```

```java
            fragmentList.add(myFragment3);
            fragmentList.add(myFragment4);
            fragmentList.add(myFragment5);
            fragmentList.add(myFragment6);
            fragmentList.add(myFragment7);
            fragmentList.add(myFragment8);
            PagerAdapter          myFragmentAdapter          =          new
    PagerAdapter(getSupportFragmentManager(), fragmentList);
            myViewPager.setAdapter(myFragmentAdapter);
        }

        // 回到主頁按鈕

        public void toHome(View view) {
            Intent intent = new Intent();
            intent.setClass( Manageclothes.this    , MainActivity.class);
            startActivity(intent);
            Manageclothes.this.finish();
        }

        // 重新整理按鈕

        public void reLoad(View view) {
            Intent intent=new Intent(this, Manageclothes.class);
            startActivity(intent);

            finish(); // 關閉此檔案

            overridePendingTransition(0, 0);
        }

        //禁用原生返回鍵

        public boolean onKeyDown(int keyCode, KeyEvent event) {
            if (keyCode == KeyEvent.KEYCODE_BACK) {
                return true;
            }
            return false;
        }
    }
```

程式名稱	Clothes_MemberAdapter
目的	衣服管理中類別細項適配器
程式碼	

```java
package com.example.clothes;
import android.content.Context;
import android.graphics.BitmapFactory;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import com.example.clothes.database.clothesDAO;
import com.example.clothes.database.getClothesMember;
import java.util.ArrayList;
public class MemberAdapter extends RecyclerView.Adapter<MemberAdapter.ViewHolder>{
    private Context context;
    private ArrayList<getClothesMember> ClothesList;
    private LayoutInflater inflater;

    // 宣告資料庫功能類別欄位變數

    private clothesDAO dao;
    public MemberAdapter(Context context, ArrayList<getClothesMember> ClothesList) {
        this.context = context;
        this.ClothesList = ClothesList;
        dao = new clothesDAO(context);
        inflater = LayoutInflater.from(context);
    }
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemView = inflater.inflate( R.layout.cardviewlayout,parent,false);
        return new ViewHolder(itemView);
    }
    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        getClothesMember member = ClothesList.get(position);

holder.imageId.setImageBitmap( BitmapFactory.decodeFile(member.getImgPath()));
        holder.textName.setText(member.getName());
    }
    public void getMultipleSelect(int position){
        getClothesMember member = ClothesList.get(position);
```

```
                dao.delete(member.getId());
                notifyItemRemoved(position);
        }
        @Override
        public int getItemCount() {
                return ClothesList.size();
        }

        //Adapter 需要一個 ViewHolder，只要實作它的 constructor 就好，保存起

來的 view 會放在 itemView 裡面

        class ViewHolder extends RecyclerView.ViewHolder{
                private ImageView imageId;
                private TextView textName;
                public ViewHolder(View itemView) {
                        super(itemView);
                        imageId = (ImageView) itemView.findViewById(R.id.imageId);
                        textName = (TextView) itemView.findViewById(R.id.textName);

                        //在 adapter 中設置点击事件(長按)

                        itemView.setOnLongClickListener(new  View.OnLongClickListener()
{

                                @Override
                                public boolean onLongClick(View v) {
                                        if(onItemLongClickListener!=null){
                                                onItemLongClickListener.OnItemLongClick(v,
ClothesList.get(getLayoutPosition())));
                                        }
                                        return false;
                                }
                        });

                        //在 adapter 中設置点击事件(短按)

                        itemView.setOnClickListener(new View.OnClickListener() {
                                @Override
                                public void onClick(View v) {
                                        if(onItemClickListener!=null){
                                                onItemClickListener.OnItemClick(v,
ClothesList.get(getLayoutPosition())));
                                        }
                                        return;
```

```java
            }
        });
    }
}

//设置 item 的监听事件的接口(長)

public interface OnItemLongClickListener {
    public void OnItemLongClick(View view, getClothesMember data);
}
private OnItemLongClickListener onItemLongClickListener;
public        void        setOnItemLongClickListener(OnItemLongClickListener
onItemLongClickListener) {
    this.onItemLongClickListener = onItemLongClickListener;
}

//设置 item 的监听事件的接口(短)

public interface OnItemClickListener {
    public void OnItemClick(View view, getClothesMember data);
}
private OnItemClickListener onItemClickListener;
public  void  setOnItemClickListener(OnItemClickListener  onItemClickListener)
{
    this.onItemClickListener = onItemClickListener;
}
}
```

程式名稱	Clothes_PagerAdapter
目的	衣服管理中類別適配器

程式碼

```java
package com.example.clothes;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import com.example.clothes.fragment.clothes1Fragment;
import com.example.clothes.fragment.clothes2Fragment;
import com.example.clothes.fragment.clothes3Fragment;
import com.example.clothes.fragment.clothes4Fragment;
import com.example.clothes.fragment.clothes5Fragment;
import com.example.clothes.fragment.clothes6Fragment;
import com.example.clothes.fragment.clothes7Fragment;
import com.example.clothes.fragment.clothes8Fragment;
import java.util.List;
public class PagerAdapter extends FragmentPagerAdapter {
    private List<Fragment> fragmentList;
    public PagerAdapter(FragmentManager fm, List<Fragment> fragmentList) {
        super(fm);
        this.fragmentList = fragmentList;
    }
    @Override
    public Fragment getItem(int position) {
        switch(position)
        {
            case 0:
                clothes1Fragment tab1=new clothes1Fragment();
                return tab1;
            case 1:
                clothes2Fragment tab2=new clothes2Fragment();
                return tab2;
            case 2:
                clothes3Fragment tab3=new clothes3Fragment();
                return tab3;
            case 3:
                clothes4Fragment tab4=new clothes4Fragment();
                return tab4;
            case 4:
```

```java
                clothes5Fragment tab5=new clothes5Fragment();
                return tab5;
            case 5:
                clothes6Fragment tab6=new clothes6Fragment();
                return tab6;
            case 6:
                clothes7Fragment tab7=new clothes7Fragment();
                return tab7;
            case 7:
                clothes8Fragment tab8=new clothes8Fragment();
                return tab8;
        }
        return null;
    }
    @Override
    public int getCount() {
        return fragmentList.size();
    }
}
}
```

程式名稱	WelcomeActivity
目的	一打開 APP 顯示的初始畫面，顯示 APP 的 LOGO 和名稱
程式碼	

```java
package com.example.clothes;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
public class WelcomeActivity extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);
        mHandler.sendEmptyMessageDelayed(GOTO_MAIN_ACTIVITY, 2000);

//2 秒跳轉

    }
    private static final int GOTO_MAIN_ACTIVITY = 0;
    private Handler mHandler = new Handler() {
        public void handleMessage(android.os.Message msg) {
            switch (msg.what) {
                case GOTO_MAIN_ACTIVITY:
                    Intent intent = new Intent();

                    //將原本 Activity 的換成 MainActivity

                    intent.setClass(WelcomeActivity.this, MainActivity.class);
                    startActivity(intent);
                    finish();
                    break;
                default:
                    break;
            }
        }
    };
}
```

程式名稱	editclothes
目的	修改衣服

程式碼

```java
package com.example.clothes;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.Paint;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import com.bigkoo.pickerview.OptionsPickerView;
import com.example.clothes.database.clothesDAO;
import com.example.clothes.database.getClothesMember;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.Locale;
public class editclothes extends AppCompatActivity {

    //傳入衣服 id

    public static Long clothesID ;

    //傳入照片

    protected static String PicPath ;

    // 宣告資料庫功能類別欄位變數
```

```java
    private clothesDAO dao;
    public static boolean load = true;
    String tempDBpic;

    //溫度區間_宣告

    private ArrayList<String> options1Items = new ArrayList<> ();
    private OptionsPickerView pvOptions;
    int flag ;
    String clothesType;
    String clothesStyle;
    String clothesSeason = "";

    String[] season = new String[] {"春", "夏", "秋", "冬"};

    boolean spring, summer, autumn, winter = false;
    String[] mType;
    String[] mStyle;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_editclothes);
        dao = new clothesDAO(getApplicationContext());

        //衣服種類選單

        final Spinner spinnertype = (Spinner)findViewById( R.id.spinnertype);
        final            ArrayAdapter<CharSequence>            clothesList            =
ArrayAdapter.createFromResource( editclothes.this,
                    R.array.clothesclass,
android.R.layout.simple_spinner_dropdown_item);
        spinnertype.setAdapter(clothesList);

        //衣服類型監聽事件

        spinnertype.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener(){
            public void onItemSelected(AdapterView adapterView, View view, int
position, long id){
                    clothesType = adapterView.getSelectedItem().toString();
            }
            public void onNothingSelected(AdapterView arg0) {

                    Toast.makeText( editclothes.this, "您沒有選擇任何項目",
Toast.LENGTH_LONG).show();
```

```java
            }
        });


        //衣服樣式選單

        final Spinner spinnerstyle = (Spinner)findViewById( R.id.spinnerstyle);
        final            ArrayAdapter<CharSequence>            styleList            =
ArrayAdapter.createFromResource( editclothes.this,
                R.array.styleclass,
android.R.layout.simple_spinner_dropdown_item);
        spinnerstyle.setAdapter(styleList);

        //衣服類型監聽事件

        spinnerstyle.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener(){
            public void onItemSelected(AdapterView adapterView, View view, int
position, long id){
                clothesStyle = adapterView.getSelectedItem().toString();
            }
            public void onNothingSelected(AdapterView arg0) {

                Toast.makeText(   editclothes.this, "您沒有選擇任何項目",
Toast.LENGTH_LONG).show();
            }
        });


        initData(); //定義資料

        //完成後選項

        final Button editok = (Button)findViewById( R.id.editok);
        final Button delclothes = (Button)findViewById( R.id.delclothes);
        editok.setOnClickListener(EditOrDel);
        delclothes.setOnClickListener(EditOrDel);

        //新增圖片(addclothes)

        ImageView clothesPic = (ImageView)findViewById( R.id.clothesPic);
        clothesPic.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AddClothes.clothesID = clothesID;
```

```
                    Intent intent = new Intent ();
                    intent.setClass( editclothes.this   , AddClothes.class);
                    startActivity(intent);
                    finish();
                }
            });
        showPic(PicPath);
        if(PicPath != tempDBpic) AddClothes.finishself.finish();

        //溫度區間--定義資料

        final                TextView            temp_range_upper            =
(TextView)findViewById( R.id.TempRange_Upper);
        final                TextView            temp_range_lower            =
(TextView)findViewById( R.id.TempRange_Lower);


temp_range_upper.getPaint().setFlags( Paint.UNDERLINE_TEXT_FLAG); //下底線


        temp_range_upper.getPaint().setAntiAlias(true);//抗鋸齒


temp_range_lower.getPaint().setFlags( Paint.UNDERLINE_TEXT_FLAG); //下底線


        temp_range_lower.getPaint().setAntiAlias(true);//抗鋸齒

        temp_range_upper.setOnClickListener(tempChangeListener);
        temp_range_lower.setOnClickListener(tempChangeListener);

        //穿衣季節

        CheckBox Spring = (CheckBox) findViewById( R.id.spring);
        CheckBox Summer = (CheckBox) findViewById( R.id.summer);
        CheckBox Autumn = (CheckBox) findViewById( R.id.autumn);
        CheckBox Winter = (CheckBox) findViewById( R.id.winter);
        Spring.setOnCheckedChangeListener(seasonCheckedChangeListener);
        Summer.setOnCheckedChangeListener(seasonCheckedChangeListener);
        Autumn.setOnCheckedChangeListener(seasonCheckedChangeListener);
        Winter.setOnCheckedChangeListener(seasonCheckedChangeListener);

        //建立選擇器資料(溫度區間)

        initOptionData();

        //建立選擇器
```

```java
        pvOptions      =      new      OptionsPickerView.Builder(this,      new
OptionsPickerView.OnOptionsSelectListener() {
            @Override
            public void onOptionsSelect(int options1, int options2, int options3,
View v) {
                String tx = options1Items.get(options1) ;
                if (flag == 1)
                    temp_range_lower.setText(tx);
                else if(flag == 2)
                    temp_range_upper.setText(tx);
                flag = 3;
            }
        }).setTitleText("請選擇溫度區間") // 選擇器標題

            .setContentTextSize(20)//設定滾輪文字大小

            .setDividerColor( Color.GREEN)//設定分割線顏色

            .setSelectOptions(120)//默認選中值

            .setBgColor( Color.BLACK)
            .setTitleBgColor( Color.DKGRAY)
            .setTitleColor( Color.LTGRAY)
            .setCancelColor( Color.YELLOW)
            .setSubmitColor( Color.YELLOW)

            .setCancelText("取消")

            .setSubmitText("確定")

            .setTextColorCenter( Color.LTGRAY)

            .setBackgroundId(0x66000000) //設定外部遮罩顏色

            .build();
        pvOptions.setPicker(options1Items);
    }

    //顯示圖片

    public void showPic(String picpath){
        ImageView imv = (ImageView)findViewById( R.id.clothesPic);
        Bitmap bitmap = BitmapFactory.decodeFile(picpath);
        imv.setImageBitmap(bitmap);
```

```java
        }
    //溫度區間監聽事件

    private        View.OnClickListener        tempChangeListener    =    new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                switch (v.getId()){
                    case R.id.TempRange_Lower:
                        if (pvOptions != null){
                            flag = 1;

                            pvOptions.show(); // 從下方彈出選擇器

                        }
                        break;
                    case R.id.TempRange_Upper:
                        if (pvOptions != null){
                            flag = 2;

                            pvOptions.show(); // 從下方彈出選擇器

                        }
                        break;
                }
            }
    };
    private void initOptionData() {

        //穿衣溫度-選項

        for (int i = -100; i < 100; i++) {
            options1Items.add( String.valueOf(i) + "°C");
        }
    }

    //穿衣季節監聽事件

    private                            CompoundButton.OnCheckedChangeListener
seasonCheckedChangeListener                            =                    new
CompoundButton.OnCheckedChangeListener(){
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
            switch (buttonView.getId()){
```

```java
                        case R.id.spring:
                            if(isChecked)
                                spring = true;
                            else
                                spring = false;
                            break;
                        case R.id.summer:
                            if(isChecked)
                                summer = true;
                            else
                                summer = false;
                            break;
                        case R.id.autumn:
                            if(isChecked)
                                autumn = true;
                            else
                                autumn = false;
                            break;
                        case R.id.winter:
                            if(isChecked)
                                winter = true;
                            else
                                winter = false;
                            break;
                }
            }
    };

    private void initData(){
        getClothesMember member = dao.getoneID(clothesID);

        //layout 的物件宣告

        TextView clothesname = (TextView)findViewById( R.id.clothesname);
        TextView                    temp_range_upper                    =
(TextView)findViewById( R.id.TempRange_Upper);
        TextView                    temp_range_lower                    =
(TextView)findViewById( R.id.TempRange_Lower);
        Spinner spinnertype = (Spinner)findViewById( R.id.spinnertype);
        Spinner spinnerstyle = (Spinner)findViewById( R.id.spinnerstyle);
        CheckBox Spring = (CheckBox) findViewById( R.id.spring);
        CheckBox Summer = (CheckBox) findViewById( R.id.summer);
```

```java
        CheckBox Autumn = (CheckBox) findViewById( R.id.autumn);
        CheckBox Winter = (CheckBox) findViewById( R.id.winter);
        //showPic(member.getImgPath());
        if(load) PicPath = member.getImgPath();
        tempDBpic = member.getImgPath();
        clothesname.setText(member.getName());
        temp_range_lower.setText(member.getTempLower().toString() + "°C");
        temp_range_upper.setText(member.getTempUpper().toString() + "°C");

        //Spinner 修改預設值成 db 內容

        mType =   getResources().getStringArray(R.array.clothesclass);

spinnertype.setSelection(Arrays.asList(mType).indexOf(member.getType()), true);
        mStyle =   getResources().getStringArray(R.array.styleclass);

spinnerstyle.setSelection(Arrays.asList(mStyle).indexOf(member.getStyle()), true);

        //season 預設內容

        clothesSeason = member.getSeason();
        for (int i = 1; i<=clothesSeason.length() ; i++){
            for (int j = 0; j<4 ; j++){
                if(season[j].equals(String.valueOf(clothesSeason.charAt(i-1)))){
                    switch (j){
                        case 0:
                            spring = true;
                            Spring.setChecked(true);
                            break;
                        case 1:
                            summer = true;
                            Summer.setChecked(true);
                            break;
                        case 2:
                            autumn = true;
                            Autumn.setChecked(true);
                            break;
                        case 3:
                            winter = true;
                            Winter.setChecked(true);
                            break;
                    }
                }
```

```java
            }
        }
        load = false;
    }
    //完成選項

    private View.OnClickListener EditOrDel = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            switch (v.getId()) {
                case R.id.delclothes:
                    LeaveEdit();
                    dao.delete(clothesID);
                    dao.close();
                    break;
                case R.id.editok:
                    CompleteEDIT();
                    if(clothesSeason != "") LeaveEdit();
                    break;
            }
        }
    };
    //離開編輯畫面

    public void LeaveEdit(){
        Intent intent = new Intent ();
        //回到管理頁面

        Manageclothes.finishself.finish();
        intent.setClass(editclothes.this, Manageclothes.class);
        startActivity(intent);
        finish();
    }
    //整理進入 DB

    public void CompleteEDIT(){
        getClothesMember getclothesmember = new getClothesMember();
        //1 衣服編號

        getclothesmember.setId(clothesID);
```

```java
//2 衣服照片(路徑)<<PicPath>>

getclothesmember.setImgPath(PicPath);

//3 衣服名稱<<clothesName>>

String                          clothesName                          =
((EditText)findViewById( R.id.clothesname)).getText().toString();
getclothesmember.setName(clothesName);

//4 衣服類型<<clothesType>>

getclothesmember.setType(clothesType);

//5 衣服類型<<clothesStyle>>

getclothesmember.setStyle(clothesStyle);

//6 溫度下限<<tempLower>>

String                          tempL                          =
((TextView)findViewById( R.id.TempRange_Lower)).getText().toString();
Long                          tempLower                          =
Long.parseLong(tempL.substring(0,tempL.indexOf("°C")));
getclothesmember.setTempLower(tempLower);

//7 溫度上限<<tempUpper>>

String                          tempU                          =
((TextView)findViewById( R.id.TempRange_Upper)).getText().toString();
Long                          tempUpper                          =
Long.parseLong(tempU.substring(0,tempU.indexOf("°C")));
getclothesmember.setTempUpper(tempUpper);

//8 衣服季節<<clothesSeason>>

clothesSeason = "";

if (spring) clothesSeason = clothesSeason + "春";

if (summer) clothesSeason = clothesSeason + "夏";

if (autumn) clothesSeason = clothesSeason + "秋";

if (winter) clothesSeason = clothesSeason + "冬";

getclothesmember.setSeason(clothesSeason);
```

```java
        //9 建立日期<<timeStamp>>

        String timeStamp = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss",
Locale.getDefault()).format(new Date());
        getclothesmember.setUpdateTime(timeStamp);
        if(clothesSeason == ""){

            Toast.makeText( editclothes.this, "請至少選擇一個季節",
Toast.LENGTH_LONG).show();
        }else{
            dao.update(getclothesmember);
            PicPath = null;
            load = true;
            dao.close();
        }
    }
}
```

程式名稱	newclothes
目的	新增衣服，新增一件新的服飾至衣櫃與資料庫
程式碼	

```java
package com.example.clothes;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import com.bigkoo.pickerview.OptionsPickerView;
import com.example.clothes.database.clothesDAO;
import com.example.clothes.database.getClothesMember;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Locale;
public class newclothes extends AppCompatActivity {

    //傳入照片

    protected static String PicPath ;

    // 宣告資料庫功能類別欄位變數

    private clothesDAO dao;

    //溫度區間_宣告
```

```java
private ArrayList<String> options1Items = new ArrayList<> ();
private OptionsPickerView pvOptions;
int flag ;
String clothesType;
String clothesStyle;
String clothesSeason = "";
boolean spring, summer, autumn, winter = false;
@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView( R.layout.activity_newclothes);

        //完成後選項

        final Button Leave = (Button)findViewById( R.id.leave);
        final Button backAdd = (Button)findViewById( R.id.keepadd);
        Leave.setOnClickListener(LeaveOrAdd);
        backAdd.setOnClickListener(LeaveOrAdd);

        //溫度區間--定義資料

        final          TextView          temp_range_upper          =
(TextView)findViewById( R.id.TempRange_Upper);
        final          TextView          temp_range_lower          =
(TextView)findViewById( R.id.TempRange_Lower);


temp_range_upper.getPaint().setFlags( Paint.UNDERLINE_TEXT_FLAG); //下底線

        temp_range_upper.getPaint().setAntiAlias(true);//抗鋸齒


temp_range_lower.getPaint().setFlags( Paint.UNDERLINE_TEXT_FLAG); //下底線

        temp_range_lower.getPaint().setAntiAlias(true);//抗鋸齒

        temp_range_upper.setOnClickListener(tempChangeListener);
        temp_range_lower.setOnClickListener(tempChangeListener);

        //穿衣季節

        CheckBox Spring = (CheckBox) findViewById( R.id.spring);
        CheckBox Summer = (CheckBox) findViewById( R.id.summer);
        CheckBox Autumn = (CheckBox) findViewById( R.id.autumn);
        CheckBox Winter = (CheckBox) findViewById( R.id.winter);
```

```
Spring.setOnCheckedChangeListener(seasonCheckedChangeListener);
Summer.setOnCheckedChangeListener(seasonCheckedChangeListener);
Autumn.setOnCheckedChangeListener(seasonCheckedChangeListener);
Winter.setOnCheckedChangeListener(seasonCheckedChangeListener);

//顯示照片

if(PicPath !=null){

    //如果有則顯示圖片

    showPic(PicPath);
    AddClothes.finishself.finish();
}

//新增圖片(addclothes)

ImageView clothesPic = (ImageView)findViewById( R.id.clothesPic);
clothesPic.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        AddClothes.clothesID = null;
        Intent intent = new Intent ();
        intent.setClass( newclothes.this    , AddClothes.class);
        startActivity(intent);
        finish();
    }
});

//衣服種類選單

final Spinner spinnertype = (Spinner)findViewById( R.id.spinnertype);
final            ArrayAdapter<CharSequence>            clothesList            =
ArrayAdapter.createFromResource( newclothes.this,
            R.array.clothesclass,
android.R.layout.simple_spinner_dropdown_item);
spinnertype.setAdapter(clothesList);

//衣服類型監聽事件

spinnertype.setOnItemSelectedListener(new OnItemSelectedListener (){
    public void onItemSelected(AdapterView adapterView, View view, int
position, long id){
            clothesType = adapterView.getSelectedItem().toString();
    }
    public void onNothingSelected(AdapterView arg0) {
```

```java
                    Toast.makeText( newclothes.this, "您沒有選擇任何項目",
Toast.LENGTH_LONG).show();
            }
        });
        //衣服樣式選單

        final Spinner spinnerstyle = (Spinner)findViewById( R.id.spinnerstyle);
        final          ArrayAdapter<CharSequence>          styleList          =
ArrayAdapter.createFromResource( newclothes.this,
                R.array.styleclass,
android.R.layout.simple_spinner_dropdown_item);
        spinnerstyle.setAdapter(styleList);

        //衣服類型監聽事件

        spinnerstyle.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener(){
            public void onItemSelected(AdapterView adapterView, View view, int
position, long id){
                clothesStyle = adapterView.getSelectedItem().toString();
            }
            public void onNothingSelected(AdapterView arg0) {

                Toast.makeText( newclothes.this, "您沒有選擇任何項目",
Toast.LENGTH_LONG).show();
            }
        });
        //建立選擇器資料(溫度區間)

        initOptionData();
        //建立選擇器

        pvOptions      =      new      OptionsPickerView.Builder(this,      new
OptionsPickerView.OnOptionsSelectListener() {
            @Override
            public void onOptionsSelect(int options1, int options2, int options3,
View v) {
                String tx = options1Items.get(options1) ;
                if (flag == 1)
                    temp_range_lower.setText(tx);
                else if(flag == 2)
```

```java
                temp_range_upper.setText(tx);
                flag = 3;
            }

        }).setTitleText("請選擇溫度區間") // 選擇器標題

                .setContentTextSize(20)//設定滾輪文字大小

                .setDividerColor( Color.GREEN)//設定分割線顏色

                .setSelectOptions(120)//默認選中值

                .setBgColor( Color.BLACK)
                .setTitleBgColor( Color.DKGRAY)
                .setTitleColor( Color.LTGRAY)
                .setCancelColor( Color.YELLOW)
                .setSubmitColor( Color.YELLOW)

                .setCancelText("取消")

                .setSubmitText("確定")

                .setTextColorCenter( Color.LTGRAY)

                .setBackgroundId(0x66000000) //設定外部遮罩顏色

                .build();
        pvOptions.setPicker(options1Items);
        dao = new clothesDAO(getApplicationContext());
    }
    public void showPic(String picpath){
        ImageView imv = (ImageView)findViewById( R.id.clothesPic);
        Bitmap bitmap = BitmapFactory.decodeFile(picpath);
        imv.setImageBitmap(bitmap);
    }
    private View.OnClickListener LeaveOrAdd = new View.OnClickListener() {
        Intent intent = new Intent ();
        @Override
        public void onClick(View v) {
            if(PicPath !=null) {
                CompleteAdd();
                if(clothesSeason != ""){
                    switch (v.getId()) {
                        case R.id.leave:
```

```java
                        //回到管理頁面

                        Manageclothes.finishself.finish();
                        intent.setClass(newclothes.this,
Manageclothes.class);

                        startActivity(intent);
                        newclothes.this.finish();
                        break;
                    case R.id.keepadd:

                        //回到新增衣服頁面

                        intent.setClass(newclothes.this, newclothes.class);
                        startActivity(intent);
                        newclothes.this.finish();
                        break;
                }
            }
        }else {

            Toast.makeText(   newclothes.this, "請拍照或選擇圖片",
Toast.LENGTH_LONG).show();
        }
    }
};

//溫度區間監聽事件

private        View.OnClickListener    tempChangeListener    =        new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.TempRange_Lower:
                if (pvOptions != null){
                    flag = 1;

                    pvOptions.show(); // 從下方彈出選擇器

                }
                break;
            case R.id.TempRange_Upper:
                if (pvOptions != null){
                    flag = 2;
```

```java
                        pvOptions.show(); // 從下方彈出選擇器
                    }
                    break;
            }
        }
    };

    private void initOptionData() {//穿衣溫度-選項
        for (int i = -100; i < 100; i++) {
            options1Items.add( String.valueOf(i) + "°C");
        }
    }

    //穿衣季節監聽事件

    private                                  CompoundButton.OnCheckedChangeListener
seasonCheckedChangeListener                        =                        new
CompoundButton.OnCheckedChangeListener(){
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
            switch (buttonView.getId()){
                case R.id.spring:
                    if(isChecked)
                        spring = true;
                    else
                        spring = false;
                    break;
                case R.id.summer:
                    if(isChecked)
                        summer = true;
                    else
                        summer = false;
                    break;
                case R.id.autumn:
                    if(isChecked)
                        autumn = true;
                    else
                        autumn = false;
                    break;
                case R.id.winter:
```

```
                    if(isChecked)
                        winter = true;
                    else
                        winter = false;
                    break;
        }
    }
};

//整理進入 DB

public void CompleteAdd(){
    getClothesMember getclothesmember = new getClothesMember();
    //1 衣服編號

    //2 衣服照片(路徑)<<PicPath>>

    getclothesmember.setImgPath(PicPath);
    //3 衣服名稱<<clothesName>>

    String                          clothesName                          =
((EditText)findViewById( R.id.clothesname)).getText().toString();
    getclothesmember.setName(clothesName);

    //4 衣服類型<<clothesType>>

    getclothesmember.setType(clothesType);

    //5 衣服類型<<clothesStyle>>

    getclothesmember.setStyle(clothesStyle);

    //6 溫度下限<<tempLower>>

    String                          tempL                          =
((TextView)findViewById( R.id.TempRange_Lower)).getText().toString();
    Long                          tempLower                          =
Long.parseLong(tempL.substring(0,tempL.indexOf("°C")));
    getclothesmember.setTempLower(tempLower);

    //7 溫度上限<<tempUpper>>

    String                          tempU                          =
((TextView)findViewById( R.id.TempRange_Upper)).getText().toString();
    Long                          tempUpper                          =
Long.parseLong(tempU.substring(0,tempU.indexOf("°C")));
```

```java
        getclothesmember.setTempUpper(tempUpper);
        //8 衣服季節<<clothesSeason>>
        clothesSeason = "";
        if (spring) clothesSeason = clothesSeason + "春";

        if (summer) clothesSeason = clothesSeason + "夏";

        if (autumn) clothesSeason = clothesSeason + "秋";

        if (winter) clothesSeason = clothesSeason + "冬";

        getclothesmember.setSeason(clothesSeason);
        //9 建立日期<<timeStamp>>

        String timeStamp = new SimpleDateFormat ("yyyy/MM/dd HH:mm:ss",
Locale.getDefault()).format(new Date ());
        getclothesmember.setUpdateTime(timeStamp);
        if(clothesSeason == ""){

            Toast.makeText(  newclothes.this, "請至少選擇一個季節",
Toast.LENGTH_LONG).show();
        }else{
            dao.insert(getclothesmember);
            PicPath = null;
            dao.close();
        }
    }
}
```

9-2 其他附屬之各種元件

從 Database 與 Fragment 元件中各取一個規格描述

<p align="center">▼表 9-2-1Database 規格描述</p>

程式名稱	DBhelper
目的	建立與修改應用程式需要的資料庫
程式碼	

```
package com.example.clothes.database
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DBhelper extends SQLiteOpenHelper {

    // 資料庫名稱

    public static final String DATABASE_NAME = "clothes.db";

    // 資料庫版本，資料結構改變的時候要更改這個數字，通常是加一

    public static final int VERSION = 6;

    // 資料庫物件，固定的欄位變數

    private static SQLiteDatabase database;


    // 建構子，在一般的應用都不需要修改

    public DBhelper(Context context, String name,
                        SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }


    // 需要資料庫的元件呼叫這個方法，這個方法在一般的應用都不需要修改

    public static SQLiteDatabase getDatabase(Context context) {
        if (database == null || !database.isOpen()) {
            database = new DBhelper(context, DATABASE_NAME,
                        null, VERSION).getWritableDatabase();
```

```java
        }
        return database;
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        // 建立應用程式需要的表格

        db.execSQL(clothesDAO.CREATE_TABLE);
        db.execSQL(weatherDAO.CREATE_TABLE);
        db.execSQL(WeekWeatherDAO.CREATE_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

        // 刪除原有的表格

        db.execSQL("DROP TABLE IF EXISTS " +
clothesDAO.TABLE_NAME);
        db.execSQL("DROP TABLE IF EXISTS " +
weatherDAO.TABLE_NAME);
        db.execSQL("DROP TABLE IF EXISTS " +
WeekWeatherDAO.TABLE_NAME);

        // 呼叫 onCreate 建立新版的表格

        onCreate(db);
    }
}
```

程式名稱	clothes1Fragment
目的	從資料庫中抓取圖片跟名稱並顯示
程式碼	

```java
package com.example.clothes.fragment;
import android.content.Intent;
import android.graphics.Rect;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import com.example.clothes.MemberAdapter;
import com.example.clothes.R;
import com.example.clothes.database.clothesDAO;
import com.example.clothes.database.getClothesMember;
import com.example.clothes.editclothes;
import com.goyourfly.multiple.adapter.MultipleAdapter;
import com.goyourfly.multiple.adapter.MultipleSelect;
import com.goyourfly.multiple.adapter.StateChangeListener;
import com.goyourfly.multiple.adapter.menu.SimpleDeleteMenuBar;

import org.jetbrains.annotations.NotNull;

import java.util.ArrayList;


public class clothes1Fragment extends Fragment {

    // 宣告資料庫功能類別欄位變數

    private clothesDAO dao;

    private View view;//定義 view 用來設置 fragment 的 layout
```

```java
    public RecyclerView mCollectRecyclerView;//定義 RecyclerView

    private ArrayList<getClothesMember> clothesList;
    private MemberAdapter memberAdapter;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                                Bundle savedInstanceState) {
        view = inflater.inflate( R.layout.fragment_clothes, container, false);
        return view;
    }

    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        dao = new clothesDAO(getContext());
        if (clothesList != null)
            clothesList = null;
        clothesList = new ArrayList<>();
        initData();
        initRecyclerView();
    }

    private void initData(){

        clothesList = dao.getoneType("\"短袖上衣\"");

    }
    private void initRecyclerView(){
        mCollectRecyclerView =
(RecyclerView)view.findViewById(R.id.collect_recyclerView);
        memberAdapter = new MemberAdapter(getActivity(), clothesList);


        //長按進入選擇

        MultipleAdapter adapter = MultipleSelect
                .with(getActivity())
                .adapter(memberAdapter)
                .stateChangeListener(new StateChangeListener() {
                    public void onCancel() {}
                    public void onSelectMode() {}
                    public void onSelect(int i, int i1) {}
```

```java
                    public void onUnSelect(int i, int i1) {}
                    public void onDone(@NotNull ArrayList<Integer> arrayList)
{}

                    @Override
                    public void onDelete(@NotNull ArrayList<Integer> arrayList)
{

                        for(int i=arrayList.size() ; i>0 ;i--) {

memberAdapter.getMultipleSelect(arrayList.get(i-1));
                            clothesList.remove((int)arrayList.get(i-1));
                        }
                    }
                })
                .customMenu(new
SimpleDeleteMenuBar(getActivity(),getResources().getColor(R.color.colorAccent),Gra
vity.TOP))
                .build();
        mCollectRecyclerView.setAdapter(adapter);
        mCollectRecyclerView.setLayoutManager(new GridLayoutManager
(getActivity() ,2, GridLayoutManager.VERTICAL,false));
        mCollectRecyclerView.addItemDecoration(new MyPaddingDecoration());


        //點擊(短按)進入修改

        memberAdapter.setOnItemClickListener(new
MemberAdapter.OnItemClickListener() {
            @Override
            public void OnItemClick(View view , getClothesMember data) {
                editclothes.clothesID = data.getId();
                editclothes.load = true;
                Intent intent = new Intent();
                intent.setClass( getActivity() , editclothes.class);
                startActivity(intent);
            }
        });
    }

    public class MyPaddingDecoration extends RecyclerView.ItemDecoration {
        private int divider;
        public MyPaddingDecoration() {
```

```
                divider = 40;    //  設置 RecyclerView 分隔線寬度

        }
        @Override
        public void getItemOffsets(Rect outRect, View view, RecyclerView parent,
 RecyclerView.State state) {
                super.getItemOffsets(outRect, view, parent, state);
                outRect.left = divider;
                outRect.top = divider;
                outRect.right = divider;
                outRect.bottom = divider;
        }
    }
 }
```

因 Database 與 Fragment 之元件較多，只各取其中之一，愈看詳細元件規格描述至

Database:

https://github.com/Yeh-Hsiu-Hsiang/Project/tree/master/clothes/app/src/main/java/com/example/clothes/database

Fragment(管理衣服):

https://github.com/Yeh-Hsiu-Hsiang/Project/tree/master/clothes/app/src/main/java/com/example/clothes/fragment

Fragment(預覽衣服):

https://github.com/Yeh-Hsiu-Hsiang/Project/tree/master/clothes/app/src/main/java/com/example/viewclothes/fragment

第 10 章 測試模型

10-1 測試計畫
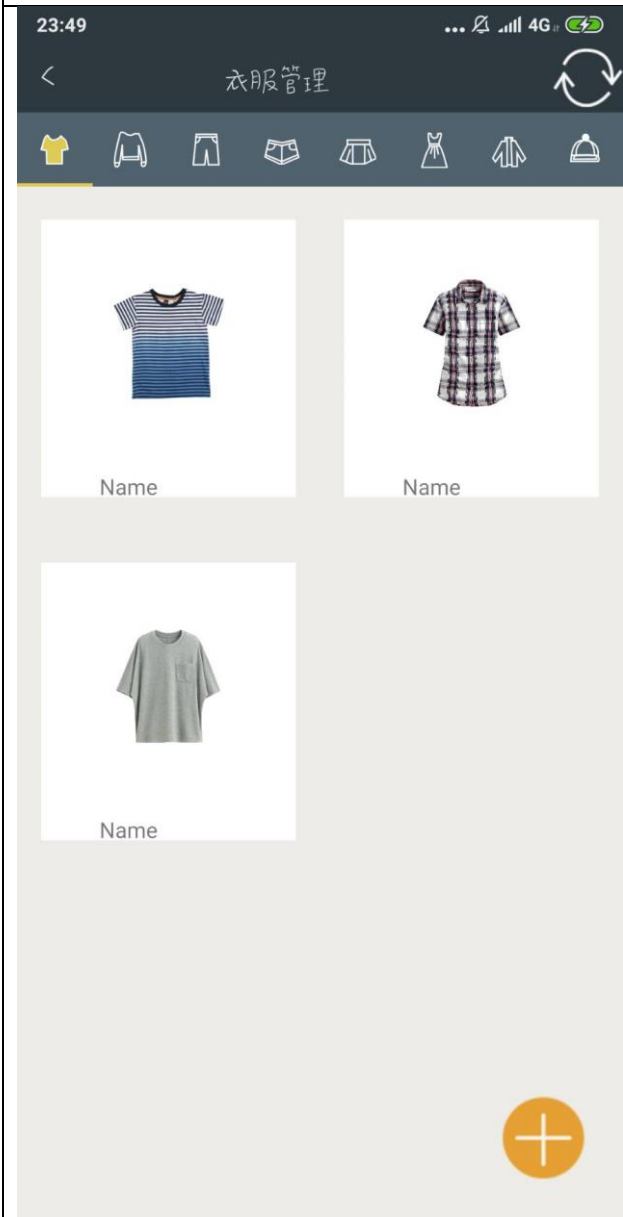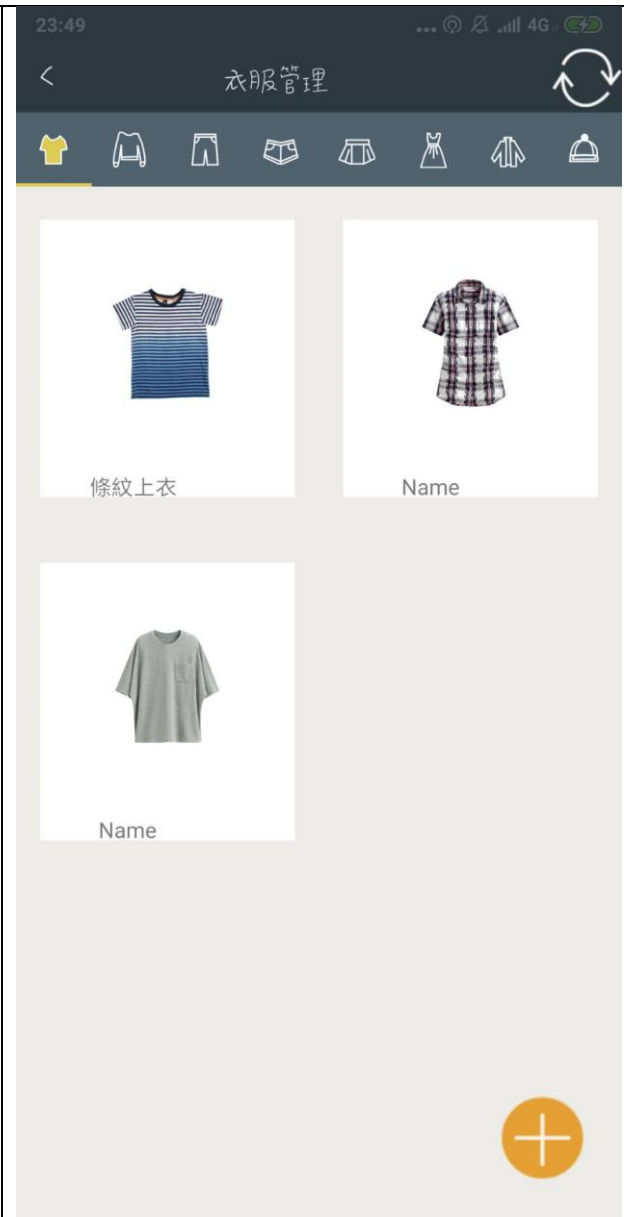
    我們以使用者的角度測試今天穿什麼的各個功能是否能正常運作。

1.新增衣服資訊：確認新增衣服資訊是否正常。

2.修改衣服資訊：確認修改衣服設定是否正常。

3.管理衣服:確認衣服是否能夠正常刪除。

4.預覽穿衣：確認搭配成果是否能完整呈現。

5.氣象顯示：確認使用者是否可以顯示當地的氣象位置資訊及一周氣象資訊。

10-2 測試個案與測試結果資料
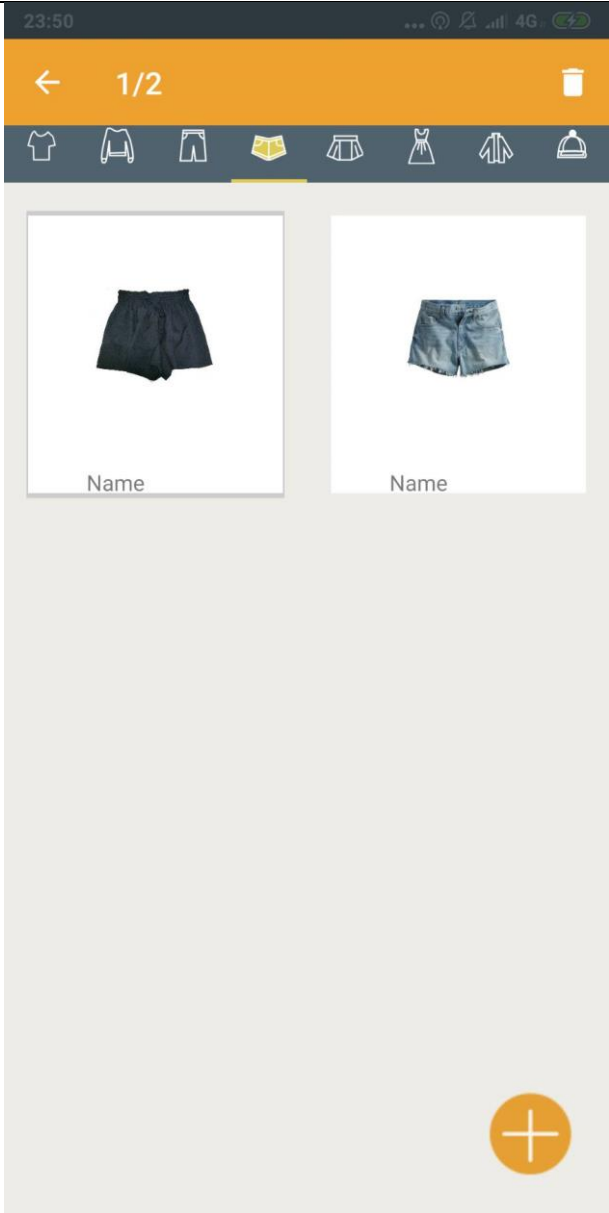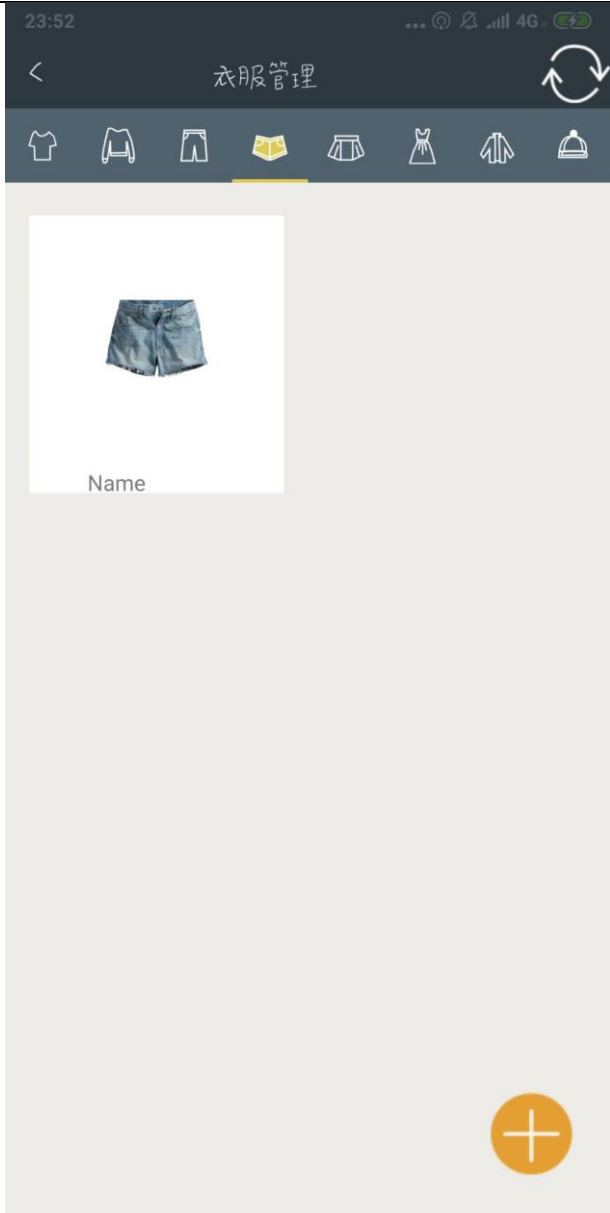
▼表 10-1-1 新增衣服資訊測試結果

功能名稱	新增衣服資訊	測試結果是否成功	是
測試畫面			

▼表 10-1-2 修改衣服資訊測試結果

功能名稱	修改衣服資訊	測試結果是否成功	是
測試畫面			

功能名稱	管理衣服	測試結果是否成功	是
測試畫面			

功能名稱	預覽穿衣	測試結果是否成功	是
測試畫面			

▼表 10-1-5 氣象資訊顯示測試結果

功能名稱	氣象資訊顯示	測試結果是否成功	是
測試畫面			

第十一章 操作手冊

11-1 系統安裝

▼表 11-1-1 安裝步驟圖示說明
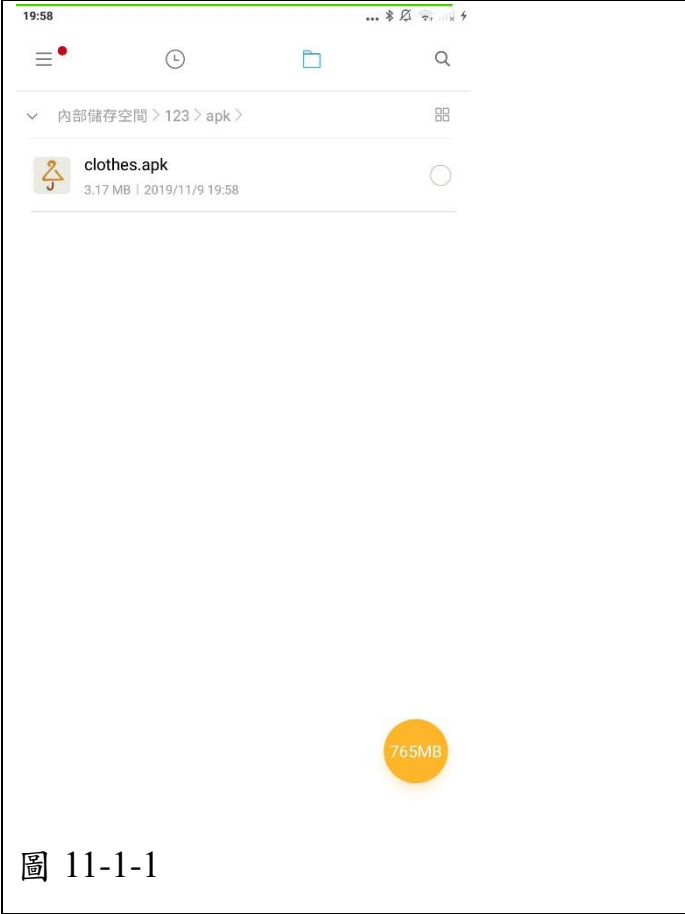
圖 11-1-1	將.APK 檔案放入手機內部儲存空間,並點選開即進入下一步

19:59　　　　　　　　　　 ❊ ⊿ ⬭ 📶 ⚡	點選安裝，進入下一步
👕　今天穿甚麼?	
您要安裝這個應用程式嗎? 應用程式不需任何特殊權限。	
取消　　　　　　　　　安裝	
圖 11-1-2	
19:59　　　　　　　　　　 ❊ ⊿ ⬭ 📶 ⚡	等待安裝，進入下一步
👕　今天穿甚麼?	
安裝中...	
圖 11-1-3	

👕　今天穿甚麼?

您要安裝這個應用程式嗎? 應用程式不需任何特殊權限。
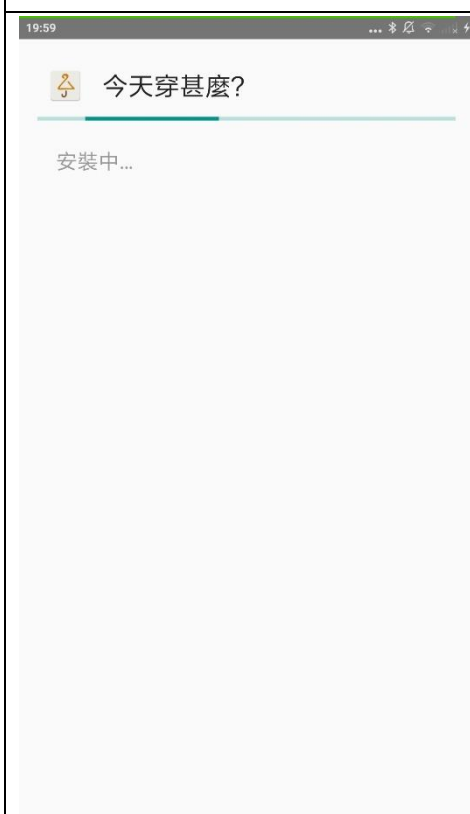
點選安裝，進入下一步
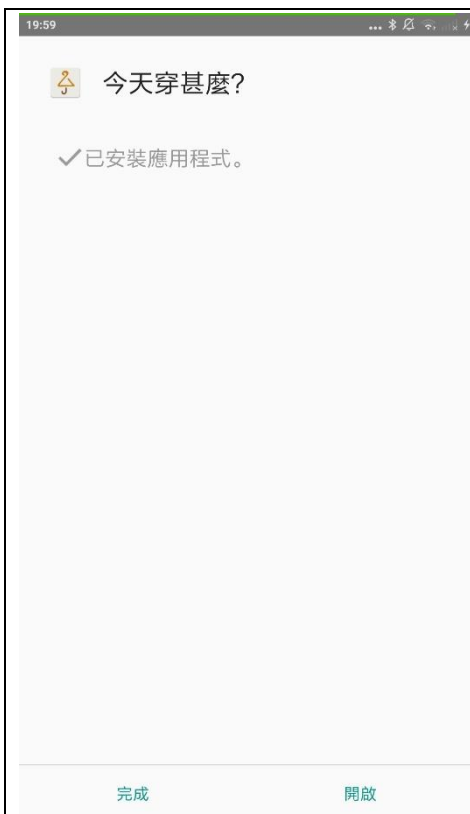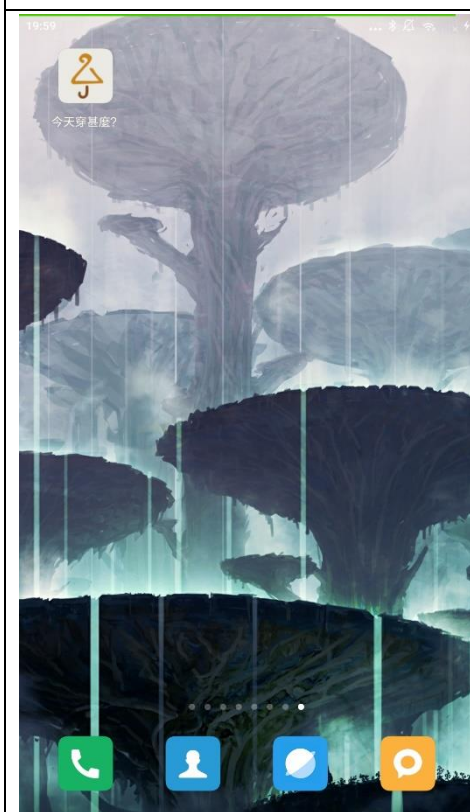
  圖 11-1-4	等待安裝完成後繼完成安裝
  圖 11-1-5	安裝完成後即會在桌面上顯示，並可直接 點選使用。

第十二章 使用手冊
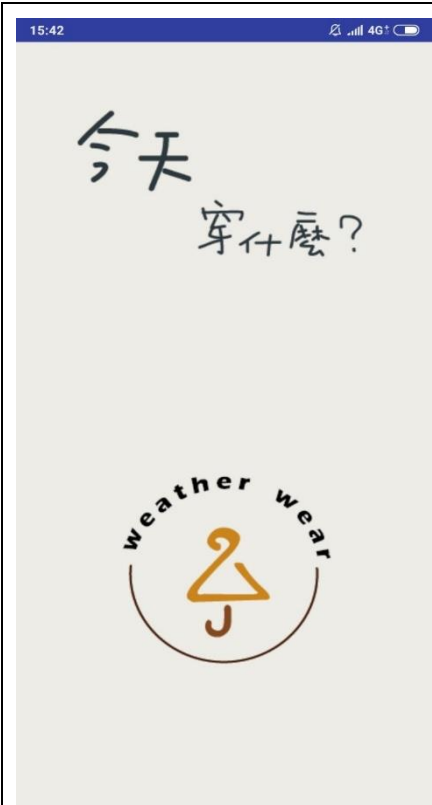
 圖 12-1-1	進入 APP 的顯示畫面 無論是否是第一次使用都無須設定 可以直接借跳入首頁
 圖 12-1-2	主畫面 畫面上會顯示時間及氣象資訊並會有三 個主要的工具選擇按鈕 第一個是衣服管理 按進去會跳到圖 12-1-3 第二個是預覽穿衣 按進去會跳到圖 12-1-10 第三個是一周的天氣 按進去會跳到圖 12-1-11 按下右上角的旋轉符號會重新整理畫面
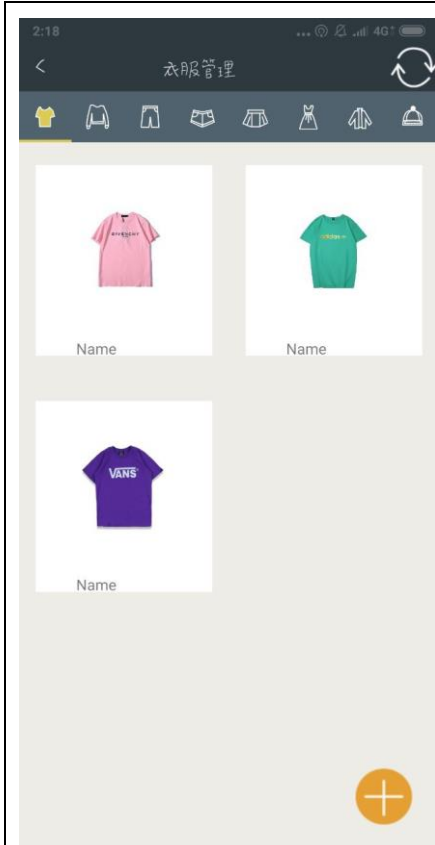
  圖 12-1-3	衣服管理的介面 按 ⊕ 可以新增衣服 會跳到圖 12-1-4 得畫面進行新增 點選上面小圖示可以選擇衣服的分類 由左至右分別為短袖、長袖、長褲、短褲、半身裙、連身裙、外套、配件
  圖 12-1-4	新增衣服的介面 先點選圖片會跳到圖 12-1-5 進行新增的步驟，完成後才回到此畫面繼續新增，填寫衣服的名稱、選擇衣服的類別、選取穿衣適合的溫度及季節、選取衣服的樣式即可完成設定

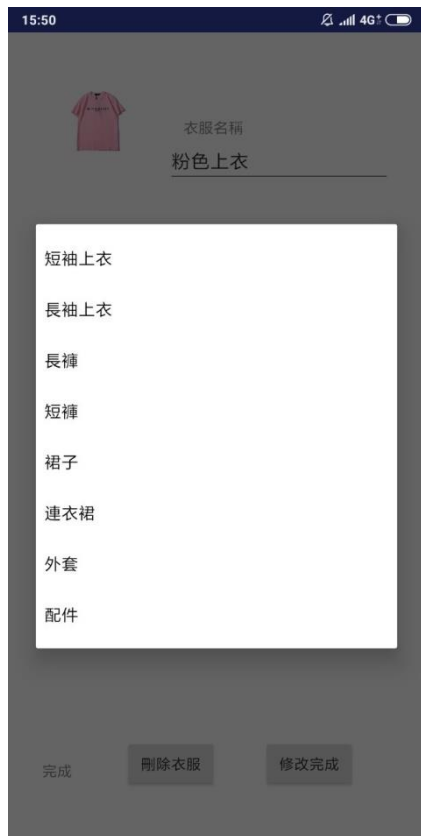|	
圖 12-1-5	點選圖片後會跳入此畫面 先按左側的匯入圖片會跳入圖 12-1-6 的 畫面，選取或拍攝好衣服後點選確定衣 服的按鈕，即可跳至圖 12-1-7 繼續進行 建檔。
圖 12-1-6	選擇需要使用的拍照程式進行拍照並載 入圖片或是點選已經存放在相簿中的圖 片載入，放上圖片後可以提供編修圖片 的功能，例如去背景

圖 12-1-7	選取衣服的類別 會提供選單選項給使用者進行選取
圖 12-1-8	選取衣服適合穿的溫度 會提供一個選單,提供使用者選取
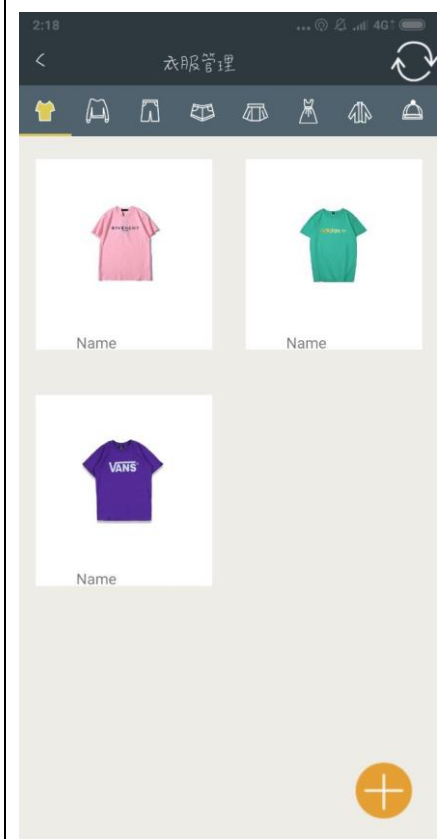
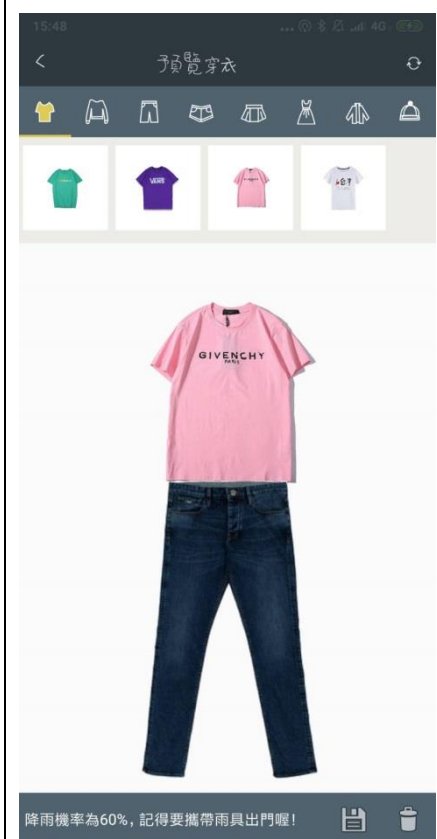|	
圖 12-1-9	新增完畢後會回到衣服管理 可以直接看到新增完的服飾
圖 12-1-10	預覽穿衣的畫面 可以自行選擇上方的各個類別進行服飾 的各項搭配，最下方會依照氣象的降雨 機率提醒使用者攜帶雨具，可以旁邊還 有選擇按鈕，可以將搭配好的衣服進行 存檔

  圖 12-1-11	一周天氣畫面會顯示 所在地的最高/最低溫度及 當時的天氣資訊
  圖 12-1-12	將一周天氣的頁面往下滑能夠得知整周 的天氣資訊

第十三章 感想

10546002 葉秀湘

　　這是第一次自己獨立寫出一個 app 功能。在程式方面，雖然有老師可以請教，但大多最考驗的還是自己的學習能力，畢竟老師不可能時時刻刻盯著你、幫助你，這個時候網路很方便，但也必須自己去判別什麼是真正對你有用的資訊。很感謝我的夥伴們，團隊中因為大家互相幫忙，才能夠順利完成這一次的專題。

10546003 楊婷文

　　從一剛開始思考專題題目，到開始行動，再到現在最後即將發表，經歷一天一天的努力，過程當中與組員之間的合作、相互討論，詢問老師等等，都是相當寶貴的經驗；在分工上跟另一位組員一起負責文件編輯，當中最困難的大概就是 UML 圖表了，但幸好在大家相互的幫助之下還是完成了這個難題。對於這次專題所學習、體認到的，就是團隊合作真的不容易，因為各自可能有事情要忙，所以大家能夠盡可能抽出時間作討論這件事情真的是非常感動，另外也非常感謝指導老師，在遇到有什麼問題或遇到困難時，給予我們許多建議、幫助我們解惑。

N1056407 羅方伶

　　我主要負責整個程式中衣服管理的部分，包括寫進資料庫與呈現，起初以為這個主題可能很容易，但由於這是我第一次寫 Android，對 APP 的撰寫與介面整體的建立是從零開始學習，才深深體會到自己能力有限，跟不上想法，雖然有網路資源和老師的幫助，但更重要的還是自己理解。感謝這次專題夥伴們的分工合作與老師提供的建議，才有了這一次的成果。

N1056414 謝佳錚

　　我主要是負責文件的部分，在製作的過程中，發現文件並沒有我們想像的這麼簡單，在寫文件的過程中會發現光有想法是行不通的，在這段時間學會找到問題並解決問題，也體悟到在將近一年的時間，要作出一個可行的 APP 對我們來說是一件很困難的，證明我們的能力還有待加強，並去學習團隊合作的重要性及如何與人合作。

1054A037 簡培雯

　　專題做到現在已經快一年了，從原本只是構想到現在已經將成品展現。這段過程不只是這一年的專題，還包含了過去幾年在大學時期所學習到的技能整合。雖然在製作專題個過程中常常遇到很多問題，在團隊間的溝通和討論、以及老師的指導下，才能把這些困難解決。因為專題讓我學習到的不僅只是課堂上教的東西，因為實作的過程，也讓我們了解的自己的不足及需要加強的能力。

第十四章 參考資料

1.新增衣服/修改衣服

圖片去背(套件)

https://github.com/GabrielBB/Android-CutOut

滾輪選擇器，選擇溫度(套件)

https://github.com/Bigkoo/Android-PickerView

儲存圖片到本地端

https://blog.csdn.net/fang323619/article/details/74909073

建立資料庫

http://www.codedata.com.tw/mobile/android-tutorial-the-3rd-class-3-sqlite/

2.管理衣服

頂部分頁切換(TabLayout+ViewPager+Fragment)

https://blog.csdn.net/TS_learner/article/details/71104511

製作清單(RecyclerView+CardView)

https://spicyboyd.blogspot.com/2018/03/app-recyclerview-cardview.html

建立各分頁中的清單(RecyclerView+Fragment)

https://blog.csdn.net/zhuchenglin830/article/details/82286109

批量選取刪除(RecyclerView)

https://github.com/goyourfly/MultiSelectAdapter

3.預覽衣服

預覽衣服，圖片拖動放大等調整

https://github.com/RoseZy/Picture_view

4.檔案轉換

圖片(uri、file、path)

https://blog.csdn.net/hust_twj/article/details/76665294

5.氣象

中央氣象局氣象資料開放平臺 – 資料擷取使用說明(抓取各類天氣資料)

https://opendata.cwb.gov.tw/opendatadoc/CWB_Opendata_API_V1.2.pdf

預報 XML 產品預報因子欄位中文說明表

https://opendata.cwb.gov.tw/opendatadoc/MFC/D0047.pdf

GPS 設置和 經緯度轉地址

https://blog.csdn.net/ww897532167/article/details/70162678
https://www.moke.tw/wordpress/computer/advanced/299/
https://apk.tw/thread-458127-1-1.html

取得目前座標(定位現在位置)

http://ntougnsslab.pixnet.net/blog/post/160966041

Android App 歡迎畫面（幾秒後進入主頁面）

https://fiend1120.pixnet.net/blog/post/192683310

6.參考書籍

Android App 程式開發剖析, 3/e（適用 Android 8 Oreo 與 Android Studio 3）

https://www.tenlong.com.tw/products/9789572246580

第十五章附錄

15.1 初審評審建議

<p align="center">▼表 15-1-1 初審評審建議</p>

組別	名稱	評審 1	評審 2	評審 3	評審 4
108402	今天穿什麼？	1. 文件的完整度需加強 2. 主題不夠明確，想做的項目與實際可達的狀況不符 3. 系統目標不要使用形容詞來作為衡量依據 4. 系統功能不夠完善 5. 進度宜加快 6. 建議加上 server 端的功能	1. 有人會用嗎？ 2. 加重 AR 3. 專題進度極需加速(已完成的太少)	1. 美工沒有做 2. 核心功能都沒做，看不懂目前完成度 3. 分工有問題 4. 資料(氣象)不用自己手動輸入 5. Github 沒用到 6. 資料庫設計有問題，功能沒串上 7. 太理想化，實務上不實用 8. 請規劃出使用者流程圖 9. VR 還沒研究，這將來不及開發 10.請將範圍調整至自己能力所及範圍 11.TA 不正確。應該錯定成年購衣女性	1. 寫程式就兩位，這樣負擔會不會太大 2. 大綱請修正，為何是範本資料 3. 文字標點符號運用有問題，逗點與句點都亂標。 4. 系統需求表軟硬體需求的建議是如何來的？ 5. 版本控管：Git 是甚麼東西 6. 行事曆，自製或整合 Google/Apple 的行事曆 App 7. 系統功能要再加強

針對初評所做的一些修正

▼表 15-1-2 初評建議修正

建議	修正
文件的完整度及大綱	多次審查修改，並與指導老師討論。
核心功能都沒做，看不懂目前完成度 專題進度極需加速	已完成所有專題所需。
寫程式就兩位，這樣負擔會不會太大	主要為兩位寫程式其他三人作為輔助，並協助提供判斷條件。
Github 沒用到	定期將檔案上傳至 Github
請將範圍調整至自己能力所及範圍	以調整至組員能力所及的範圍，並將原先無法達到的移至未來展望的部分
分工有問題	已進行重新分配