**國立臺北商業大學**

**資 訊 管 理 系**

**107資訊系統專案設計**

**系統手冊**



**組 別：第107202組**

**題 目：一家順順**

**指導老師：許晉龍老師**

**組 長：10636008 游期順**

**組 員：10636011 劉家驤 10636017 陳易陞**

**10636028 李順玉**

**中華民國107年11月28日**

**目錄**

[第1章 背景與動機 1](#_Toc531117213)

[1-1 簡介 1](#_Toc531117214)

[1-2 問題與機會 2](#_Toc531117215)

[1-3 相關系統探討 5](#_Toc531117216)

[第2章 系統目標與預期成果 6](#_Toc531117217)

[2-1 系統目標 6](#_Toc531117218)

[2-2 預期成果 7](#_Toc531117219)

[第3章 系統規格 8](#_Toc531117220)

[3-1 系統架構 8](#_Toc531117221)

[3-2 系統軟、硬體需求與技術平台 9](#_Toc531117222)

[3-3 使用標準與工具 9](#_Toc531117223)

[第4章 專案時程與組織分工 10](#_Toc531117224)

[4-1 專案時程 10](#_Toc531117225)

[4-2 專案組織與分工 11](#_Toc531117226)

[第5章 需求模型 13](#_Toc531117227)

[5-1 使用者需求 13](#_Toc531117228)

[5-2 使用個案圖(Use case diagram) 14](#_Toc531117229)

[5-3 使用個案描述：活動圖（Activity diagram） 15](#_Toc531117230)

[5-4 分析類別圖(Analysis class diagram) 20](#_Toc531117231)

[第6章 設計模型 21](#_Toc531117232)

[6-1 循序圖(Sequential diagram) 21](#_Toc531117233)

[6-2 設計類別圖(Design class diagram) 27](#_Toc531117234)

[第7章 實作模型 28](#_Toc531117235)

[7-1 佈署圖(Deployment diagram) 28](#_Toc531117236)

[7-2 套件圖(Package diagram) 28](#_Toc531117237)

[7-3 元件圖(Component diagram) 29](#_Toc531117238)

[7-4 狀態機(State machine) 30](#_Toc531117239)

[第8章 資料庫設計 31](#_Toc531117240)

[8-1 資料庫關聯表 31](#_Toc531117241)

[8-2 表格及其Meta data 32](#_Toc531117242)

[第9章 程式 37](#_Toc531117243)

[9-1 程式清單 37](#_Toc531117244)

[9-2 程式規格描述 38](#_Toc531117245)

[第10章 測試模型 144](#_Toc531117246)

[10-1 測試計畫 144](#_Toc531117247)

[10-2 測試個案與測試結果資料 144](#_Toc531117248)

[第11章 操作手冊 149](#_Toc531117249)

[第12章 使用手冊 151](#_Toc531117250)

[第13章 感想 161](#_Toc531117251)

[10636008 游期順 161](#_Toc531117252)

[10636011 劉家驤 161](#_Toc531117253)

[10636017 陳易陞 161](#_Toc531117254)

[10636028 李順玉 161](#_Toc531117255)

[第14章 參考資料 162](#_Toc531117256)

[附錄 163](#_Toc531117257)

[附錄1-評審建議之修正情形 164](#_Toc531117258)

[附錄2-系統手冊修正情形 166](#_Toc531117259)

**表目錄**

[表 1-2-1 SWOT分析 3](#_Toc531180371)

[表 1-3-1 相關系統探討表 5](#_Toc531180372)

[表 3-2-1 系統需求表 9](#_Toc531180373)

[表 3-3-1 使用標準與工具表 9](#_Toc531180374)

[表 4-1-1 甘特圖 10](#_Toc531180375)

[表 4-2-1 專案組織與分工 11](#_Toc531180376)

[表 5-1-1 功能需求表 13](#_Toc531180377)

[表 5-1-2 非功能需求表 13](#_Toc531180378)

[表 8-2-1 資料表描述-學生帳號 32](#_Toc531180379)

[表 8-2-2 資料表描述-民宿主帳號 33](#_Toc531180380)

[表 8-2-3 資料表描述-民宿資訊 33](#_Toc531180381)

[表 8-2-4 資料表描述-工作清單 34](#_Toc531180382)

[表 8-2-5 資料表描述-職缺 34](#_Toc531180383)

[表 8-2-6 資料表描述-心得 35](#_Toc531180384)

[表 8-2-7 資料表描述-訊息清單 35](#_Toc531180385)

[表 8-2-8 資料表描述-雇用紀錄 35](#_Toc531180386)

[表 8-2-9 資料表描述-回報內容\_民宿主 36](#_Toc531180387)

[表 8-2-10 資料表描述-回報內容\_學生 36](#_Toc531180388)

[表 8-2-11 資料表描述-Open\_Data 36](#_Toc531180389)

[表 8-2-12 資料表描述-審核清單 36](#_Toc531180390)

[表 9-1-1 雇主端程式清單 37](#_Toc531180391)

[表 9-1-2 學生端程式清單 37](#_Toc531180392)

[表 9-2-1 雇主端程式規格描述 38](#_Toc531180393)

[表 9-2-2 學生端程式規格描述 106](#_Toc531180394)

[表 10-2-1 雇主端-註冊會員測試結果 144](#_Toc531180395)

[表 10-2-2 雇主端-登入會員測試結果 145](#_Toc531180396)

[表 10-2-3 雇主端-新增職缺測試結果 145](#_Toc531180397)

[表 10-2-4 雇主端-新增工作測試結果 146](#_Toc531180398)

[表 10-2-5 雇主端-新增民宿測試結果 146](#_Toc531180399)

[表 10-2-6 雇主端-回報問題測試結果 147](#_Toc531180400)

[表 10-2-7 學生端-註冊會員測試結果 147](#_Toc531180401)

[表 10-2-8 學生端-登入會員測試結果 148](#_Toc531180402)

[表 10-2-9 學生端-新增換宿心得測試結果 148](#_Toc531180403)

[表 11-1-1 雇主端操作手冊 149](#_Toc531180404)

[表 12-1-1 雇主端使用步驟說明 151](#_Toc531180405)

[表 12-1-2學生端使用步驟說明 157](#_Toc531180406)

**圖目錄**

[圖 1-2-1 打工換宿流程 2](#_Toc530765061)

[圖 2-1-1 系統介面圖 6](#_Toc530765062)

[圖 2-1-2 商業模式圖 7](#_Toc530765063)

[圖 3-1-1 系統架構圖 8](#_Toc530765064)

[圖 5-2-1 使用個案圖 14](#_Toc530765065)

[圖 5-3-1 活動圖：登入 15](#_Toc530765066)

[圖 5-3-2 活動圖：註冊 15](#_Toc530765067)

[圖 5-3-3 活動圖：搜尋人才 16](#_Toc530765068)

[圖 5-3-4 活動圖：尋找換宿 16](#_Toc530765069)

[圖 5-3-5活動圖：刊登履歷 17](#_Toc530765070)

[圖 5-3-6 活動圖：刊登職缺 17](#_Toc530765071)

[圖 5-3-7 活動圖：新增民宿資訊 18](#_Toc530765072)

[圖 5-3-8 活動圖：瀏覽工作安排 18](#_Toc530765073)

[圖 5-3-9 活動圖：瀏覽心得 19](#_Toc530765074)

[圖 5-3-10 活動圖：瀏覽職缺 19](#_Toc530765075)

[圖 5-4-1 分析類別圖 20](#_Toc530765076)

[圖 6-1-1 循序圖：登入 21](#_Toc530765077)

[圖 6-1-2 循序圖：註冊 21](#_Toc530765078)

[圖 6-1-3 循序圖：搜尋人才 22](#_Toc530765079)

[圖 6-1-4 循序圖：尋找換宿 22](#_Toc530765080)

[圖 6-1-5循序圖：瀏覽職缺 23](#_Toc530765081)

[圖 6-1-6 循序圖：刊登職缺 23](#_Toc530765082)

[圖 6-1-7 循序圖：新增民宿資訊 24](#_Toc530765083)

[圖 6-1-8 循序圖：瀏覽工作安排 24](#_Toc530765084)

[圖 6-1-9 循序圖：瀏覽心得 25](#_Toc530765085)

[圖 6-1-10 循序圖：發佈心得 25](#_Toc530765086)

[圖 6-1-11 循序圖：刊登履歷 26](#_Toc530765087)

[圖 6-2-1 設計類別圖 27](#_Toc530765088)

[圖 7-1-1佈署圖 28](#_Toc530765089)

[圖 7-2-1 套件圖 28](#_Toc530765090)

[圖 7-3-1元件圖（雇主端） 29](#_Toc530765091)

[圖 7-3-2元件圖（學生端） 29](#_Toc530765092)

[圖 7-4-1 狀態機（雇主端） 30](#_Toc530765093)

[圖 7-4-2 狀態機（學生端） 30](#_Toc530765094)

[圖 8-1-1 資料關聯圖 31](#_Toc530765095)

1. 背景與動機
   1. 簡介

根據世新大學民調中心調查超過一千名年滿20歲以上的年輕族群，結果發現有60%左右的年輕人覺得自己活得並不快樂，只有23%認為自己是快樂的，原因有很多，現代人生活步調緊湊，經濟、課業壓力等現實問題常逼得他們喘不過氣，有人面對這樣日復一日、年復一年的生活感到厭倦，甚至開始思考他們是為了什麼而努力、為了誰而如此拼命。德國詩人里爾克曾經說過：「旅行只有一種，即是走入你自己的內在之旅。」，越來越多年輕人選擇透過旅行來找回他們心中最原始的自己並在其中尋找生命的意義。對於只有滿心熱血、精力充沛卻缺乏經濟基礎的他們而言，打工換宿是一種既能節省開銷，又能發揮所長同時換取快樂的旅遊方式。

現代人生活步調緊湊，各式壓力讓人喘不過氣，面對日復一日的生活心生厭倦，越來越多的人選擇透過旅行找回他們初心，更甚者是追尋生命的意義。但有許多人對於旅行卻是心有餘而力不足，對於缺乏經濟基礎的他們，打工換宿是既能節省開銷，又能發揮所長換取充實體驗的旅遊方式。

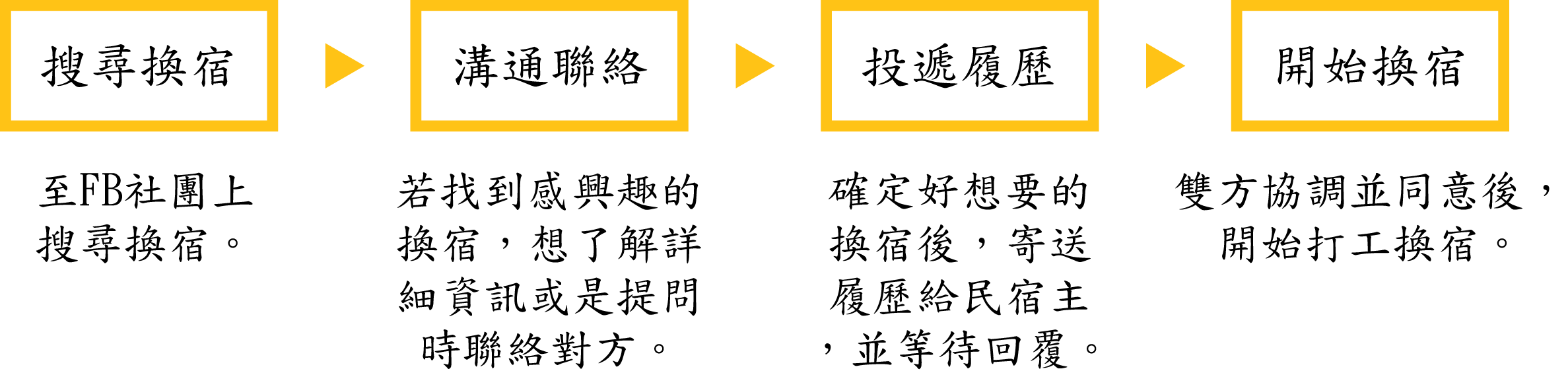
打工換宿是以勞動、專業技能和特殊才藝來換取免費食宿或金錢，時間短至兩周，長至兩個月都有，相較於一般旅遊，這種方式更能讓旅人深入體會各地的風俗民情和特色景點。這種風氣從國外興起，目的是希望旅人通過在當地工作和深度旅遊的經驗，真正融入在地生活並與當地人及其他旅人成為朋友，除此之外，學生們之間互相交流，分享換宿經驗與私房景點，並結伴出遊、培養深厚友誼。

台灣和四周離島擁有豐富多樣的天然景觀，在寒暑假的旅遊旺季，各地的知名旅遊景點都會充滿人潮，而提供旅客休憩的民宿常因此應接不暇，打工換宿成了民宿主的最佳解決方案，相反的，學生在空閒時間可以到當地或附近景點深入旅遊，放鬆心情，兩者各取所需，互惠互利。

如今打工換宿已漸漸在台灣年輕人間掀起一股風潮，除了學生之外，也有許多期望利用長假或工作轉換期間沉澱心情的社會人士加入，但目前能提供民宿主與學生溝通的管道卻寥寥無幾，為此我們期望開發一個手機應用程式作為媒合兩者的工具，並提供雙方各自所需的所有資訊，為學生找到一個更好的換宿環境，而民宿主也可以雇用到一個符合需求的學生。

* 1. 問題與機會

我們團隊觀察了打工換宿的流程，如圖1-2-1，同時訪問了有打工換宿經驗的同學，並且做了一連串的調查，在過程中我們發現了一些問題如下：



▲ 圖 1-2-1 打工換宿流程

* 搜尋換宿

學生在Facebook社團上搜尋換宿時，只能一則一則貼文慢慢查看，且每個人發文提供資訊的格式都不同。冗長且雜亂的資訊讓人失去耐心瀏覽，可能造成雙方資訊不對稱，同時也無法有效率地進行搜尋。

* 溝通聯絡

學生與民宿主聯絡的管道太多樣導致操作相當複雜。

* 投遞履歷

每個打工換宿投遞履歷的管道都不同，且沒有統一的格式。

* 開始換宿

1. 到了換宿的地點發現跟當初在Facebook社團上看到的不同。
2. 無法得知民宿是否合法。
3. 學生在剛開始可能無法快速的適應工作。

這些問題乍看之下不起眼，卻是導致一整個打工換宿流程的繁瑣與不便，若是改善了這些問題，就能夠大大提升使用者的方便性，進而讓彼此都對打工換宿留下了美好的記憶，因此我們希望可以解決這些問題，讓一家順順在打工換宿的市場中打造出無法被取代的價值。

為了了解「一家順順」的市場定位及相關打工換宿的發展戰略，我們對此APP進行深入全面的分析（SWOT）以及競爭優勢的定位，如表1-2-1。

▼ 表 1-2-1 SWOT分析

|  |  |
| --- | --- |
| 優勢(Strengths) | 劣勢(Weaknesses) |
| 1. 快速篩選工作和人才 2. 線上與民宿主溝通 3. 合法民宿驗證 4. 即時掌控工作進度 5. 分享換宿心得 6. 收藏換宿 | 1. 使用人數較少 2. 部分功能和FB換宿社團衝突 3. 無法辨別民宿主照片真偽 |
| 機會(Opportunities) | 威脅(Threat) |
| 1. 打工換宿人數眾多 2. 目前市面上無此類型APP 3. 換宿心得可分享至社群平台 4. 可以鎖定換宿區域 | 1. 學生較熟悉FB換宿社團 |

在SWOT分析後，我們進而使用USED技巧來產出解決方案，方案如下：

U、如何善用每個優勢：

1. 盡量簡化篩選流程，並將最常被選用的條件置頂，節省使用者的選取時間，進而篩選效率。
2. 確保民宿主與學生間的連線穩定，並增加常用問候語標籤供雙方點選，節省打字時間。
3. 除了合法民宿外，也提供評分優良的民宿較高的被搜尋優先權。
4. 將工作進度畫面設計為簡而易懂的進度條，讓學生和民宿主在工作時方便即時確認。
5. 設立獎勵機制來鼓勵學生積極分享換宿心得，以提供其他學生豐富的換宿心得文。
6. 學生可以將有興趣的換宿收藏起來，自行比對選出最適合的換宿工作。

S、如何停止每個劣勢：

1. 在FB各換宿社團宣傳亦或是至全台各大專院校發放問卷，並藉機推廣此APP，增加使用人數。
2. 凸顯並加強FB社團所不能提供的功能，增加使用黏著度。
3. 透過評論功能，我們能集結大眾的意見，讓使用者了解該民宿照片和資訊的真實性。

E、如何成就每個機會：

1. 已此APP的可靠性和功能性，盡可能地吸引最多的打工換宿者。
2. 此APP可成為打工換宿APP的領頭羊，吸引手機平台的使用者。
3. 透過分享社群機制，可以增加APP的曝光度，讓更多年輕族群認識。
4. 鎖定特定換宿區域，進而達到促進當地地區觀光的效果。

D、如何抵禦每個威脅：

1. 盡可能地強化APP的功能並新增FB換宿社團所無法提供的功能。
   1. 相關系統探討

經過本組多方搜尋和比對，發現與打工換宿相關的網站和FB社團寥寥可數，而目前更是沒有相關的APP，以下是針對相關的社團和網站所做的功能特色說明和比較。

1. KiTaiwan-打工換宿遊台灣

KiTaiwan是一個換宿媒合網站，整合全台換宿工作，縮短旅友尋找工作的時間，並具有不透漏隱私的交談平台，民宿主和學生們可透過網站，他們還與全台各協會和機構合作，提供平台更多的換宿機會。

1. FB換宿社團（打工換宿．背包環島）

此為台灣最大的FB打工換宿社團，學生們可在社團貼文，提供個人資訊，而雇主可以張貼換宿機會，雙方皆透過FB聯絡相關事宜，而學生也會在社團分享換宿經驗。

1. Couch Surfing 沙發衝浪

是一個旨在幫助旅行者與當地人建立聯繫的國際性非營利網路，旅人在異地旅遊時，可透過網站來尋找當地會員提供的沙發或床作為免費住宿，而旅人可透過分享旅行故事、幫忙料理、繪畫、演奏樂器、甚至是提供勞力來當作回饋，藉此也達到了文化交流的目的。

▼ 表 1-3-1 相關系統探討表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 相關網站  特色 | 一家順順 | FB換宿社團 | KiTaiwan | Couch Surfing |
| 快速篩選工作和人才 | **🗸** |  | ● |  |
| 線上溝通 | **🗸** | ● | ● | ● |
| 即時掌控工作進度 | **🗸** |  |  |  |
| 換宿心得分享 | **🗸** | ● | ● | ● |
| 免費刊登 | **🗸** | ● | ● | ● |
| 安心審核民宿 | **🗸** |  |  | ● |
| 快速履歷 | **🗸** |  | ● |  |
| 合法民宿驗證 | **🗸** |  |  |  |

1. 系統目標與預期成果
   1. 系統目標

目前臺灣打工換宿在年輕族群中算是相當熱門的，但由於提供換宿的民宿通常都位於鄉下或是離島地區，換宿社團或網站所刊登的資訊也未必與實際相符，對於學生們而言，決定到某間民宿打工換宿其實算是一件賭注，所有事情要去了才可以確定。然而，對於民宿主而言，學生所投的履歷，或是其過去的換宿經驗，也是充滿了未知數。

為此，本專題的目標是希望『一家順順』一個媒合學生及民宿主即時溝通的平台，能使雙方得到最正確和即時的資訊，讓就算是沒有經驗的學生們，也能夠透過平台瞭解打工換宿的精神，以及相關資訊，除此之外學生可以在此平台獲取關於民宿的所有資訊與其他學生的評論和評價，更可以透過閱覽其他學生所撰寫的心得文章，來對該民宿和環境有更深的了解，不會再對民宿一無所知，而我們也要求民宿主上傳其營業登記證明，由後端利用政府提供合法民宿的OpenData來加以審核，以確保學生們的安全與保障。

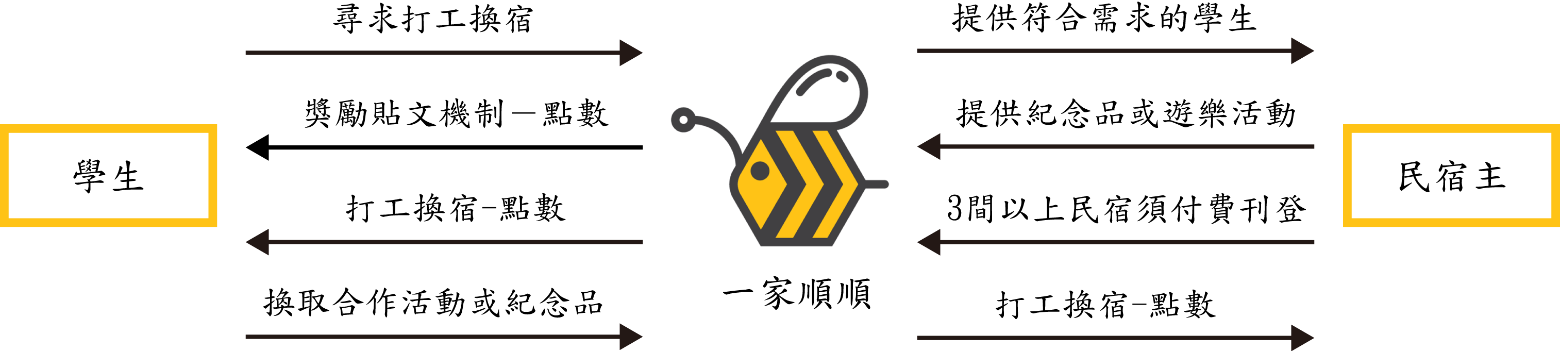
而民宿主可以透過手機APP即時刊登換宿缺額，並設立專業技能篩選您要的人才，即時閱覽學生的履歷和個人評價，隨時隨地找到合適人才。透過工作排程功能，讓『一家順順』就像是個虛擬秘書，隨時為民宿主掌握工作進度。

|  |  |  |  |
| --- | --- | --- | --- |
| 尋找換宿 | 所有評論 | 上傳營業登記證明 | 工作排程 |
|  |  |  |  |

▲ 圖 2-1-1 系統介面圖

* 1. 預期成果
* 能與台灣所有合法民宿合作
* 加強打工換宿的風氣
* 間接發展在地觀光
* 成為打工換宿主流應用程式
* 人性化使用者界面，增加使用意願
* 提供學生們與民宿主雙方最正確的資訊
* 民宿主能及時掌控學生工作進度

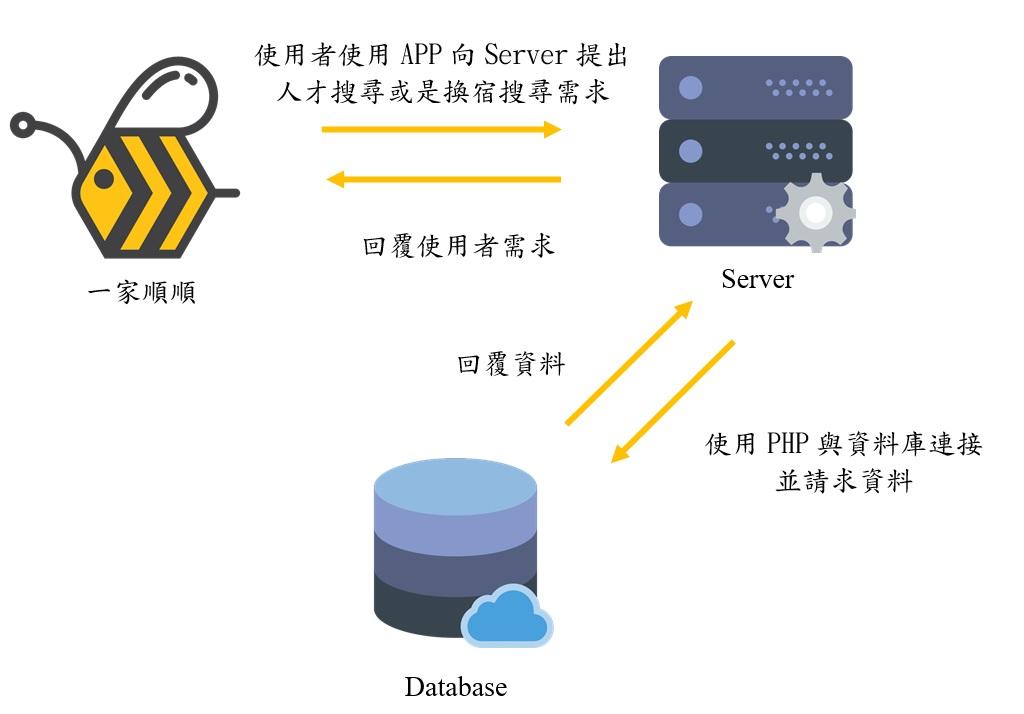
為了提供更多優良合法的民宿，『一家順順』將加入完善的商業制度及會員機制，促進商業流動，並延長『一家順順』的永續發展。如圖2-2-1



▲ 圖 2-1-2 商業模式圖

* 當民宿主需要換宿學生時，可以透過刊登資訊功能，列出民宿資訊和工作內容，供學生搜尋。若是民宿主有3間以上的民宿需要刊登，必須付出些許金額作為刊登費用。
* 學生在打工換宿的過程中，同時也可以賺取打工換宿的點數，換取紀念品或特定遊樂活動，例如：水上遊樂設施等…。
* 若學生在交換心得中，踴躍分享自身打工換宿的經驗，可以獲得獎勵的點數。
* 民宿主在學生換取紀念品或合作遊樂活動後，獲得的點數可以用來換取限時刊登多間民宿的服務，或是獲得的點數達到一定的數量會將此民宿列為優良民宿。

1. 系統規格
   1. 系統架構



▲ 圖 3-1-1 系統架構圖

* 1. 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 | 1. 4G行動上網、Wi-Fi無線網路 | |

* 1. 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、Windows Server 2012 |
| 資料庫 | MYSQL |
| 伺服器 | AppServ |
| 程式開發技術 | |
| 前端技術 | Android Studio、HTML5、CSS3、JavaScript |
| 後段技術 | JAVA、C#、PHP |
| 編輯器 | Android studio、Microsoft Visual Studio、Atom |
| 管理程式平台 | |
| 版本控制 | Git、GitKraken |
| 專案管理 | Google Drive |
| 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 |

1. 專案時程與組織分工
   1. 專案時程

▼ 表 4-1-1 甘特圖

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 月份  工作項目 | 1月 | 2月 | 3月 | 4月 | 5月 | 6月 | 7月 | 8月 | 9月 | 10月 | 11月 | 12月 |
| 主題構思 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 開發工具學習 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 系統功能分析 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 前端設計 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 前端開發 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 資料庫設計 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 後端開發 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 資料庫建置 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 系統模型 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 系統測試 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 系統整合 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 系統需求分析 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 系統手冊製作 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
|  | 預期進度 |
|  | 實際進度 |

* 1. 專案組織與分工

▼表 4-2-1 專案組織與分工

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 學號姓名  項目 | | 10636008 | 10636011 | 10636017 | 10636028 |
| 游期順 | 劉家驤 | 陳易陞 | 李順玉 |
| 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 | ● | ○ | ○ | ○ |
| 資料庫效能調校 | ● |  |  |  |
| 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 | 手冊撰寫 | ● | ○ | ● | ○ |
| 手冊編排 |  | ● |  | ● |
| 簡介撰寫 |  | ● |  |  |
| 內容校正 | ● | ● | ● | ● |
| U  M  L  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 | ● | ○ | ● |  |
| 部署圖 |  |  | ● |  |
| 元件圖 |  |  | ● |  |
| 套件圖 |  |  | ● |  |
| 狀態機 |  |  | ● |  |

※●代表主要負責人 ○代表協助負責人

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 學號姓名  項目 | | | 10636008 | 10636011 | 10636017 | 10636028 |
| 游期順 | 劉家驤 | 陳易陞 | 李順玉 |
| 程式  撰寫 | 前端 | 前端介面 | ○ | ● | ○ | ● |
| 前端設計 | ○ | ● | ○ | ● |
| 後端 | 程式碼撰寫 | ● | ○ | ● | ○ |
| 程式碼整合 | ● |  |  |  |
| 程式碼偵錯 | ● | ○ | ○ | ○ |
| P  H  P | 雇主端 | 登入註冊 |  |  | ● |  |
| 人才搜尋 | ● |  | ● |  |
| 刊登資訊 | ● |  |  |  |
| 履歷收件夾 | ● |  |  |  |
| 工作安排 | ● |  |  |  |
| 雇主頁面 | ● |  |  |  |
| 學生端 | 登入註冊 |  |  | ● |  |
| 找換宿 | ● | ○ | ● |  |
| 交換心得 |  | ○ | ● |  |
| 收件匣 | ● | ○ | ○ | ○ |
| 工作排程 | ● | ○ |  |  |
| 個人頁面 | ● | ○ |  |  |
| 系  統  功  能 | 雇主端 | 登入註冊 |  | ● | ● |  |
| 人才搜尋 | ● |  |  | ● |
| 刊登資訊 | ● |  |  | ● |
| 履歷收件夾 | ● |  |  | ● |
| 工作安排 | ● |  |  | ● |
| 雇主頁面 | ● |  |  | ● |
| 學生端 | 登入註冊 |  | ● | ● |  |
| 找換宿 | ● | ● | ● |  |
| 交換心得 | ● | ● | ● |  |
| 收件匣 | ● | ○ | ○ | ● |
| 工作排程 | ● | ● |  |  |
| 個人頁面 | ● | ● | ● | ● |

※●代表主要負責人 ○代表協助負責人

1. 需求模型
   1. 使用者需求
2. 功能需求

訪客登入可以直接尋找人才和尋找換宿，但僅限瀏覽而無法使用其他功能，註冊後即可以學生或民宿主身分進入系統，學生可以尋找換宿、上傳心得、新增工作安排，而民宿主可以尋找人才、刊登民宿資訊並新增工作安排來掌握學生工作進度。

▼ 表 5-1-1 功能需求表

|  |  |
| --- | --- |
| 功能需求 | 使用案例 |
| 訪客註冊即可以學生或民宿主身分登入系統 | 註冊 |
| 依身分登入使用特定功能 | 登入 |
| 民宿主可篩選符合需求的人才並進行雇用與聯繫 | 尋找人才 |
| 學生可以尋找適合自身能力的換宿工作 | 尋找換宿 |
| 民宿主和學生可透過此功能排定工作時程和查看工作內容 | 新增工作安排 |
| 民宿主可刊登民宿資訊和合法證照供學生瀏覽 | 刊登民宿資訊 |
| 學生可分享自身換宿心得也可以瀏覽其他學生的心得 | 發佈心得 |

1. 非功能需求

使用者能直覺的操作此APP各個功能，在過程中能流暢有效率地達成使用者的目的，並加強資料庫安全，保護使用者個資。

▼ 表 5-1-2 非功能需求表

|  |
| --- |
| 非功能需求 |
| 人性化的使用者介面 |
| 運行順暢的系統 |
| 隱私保護 |

使用者須具備Android 4.0.3以上系統版本的智慧型手機、記憶體至少2G，上傳檔案需要3G/4G/Wi-Fi無線網路。

* 1. 使用個案圖(Use case diagram)

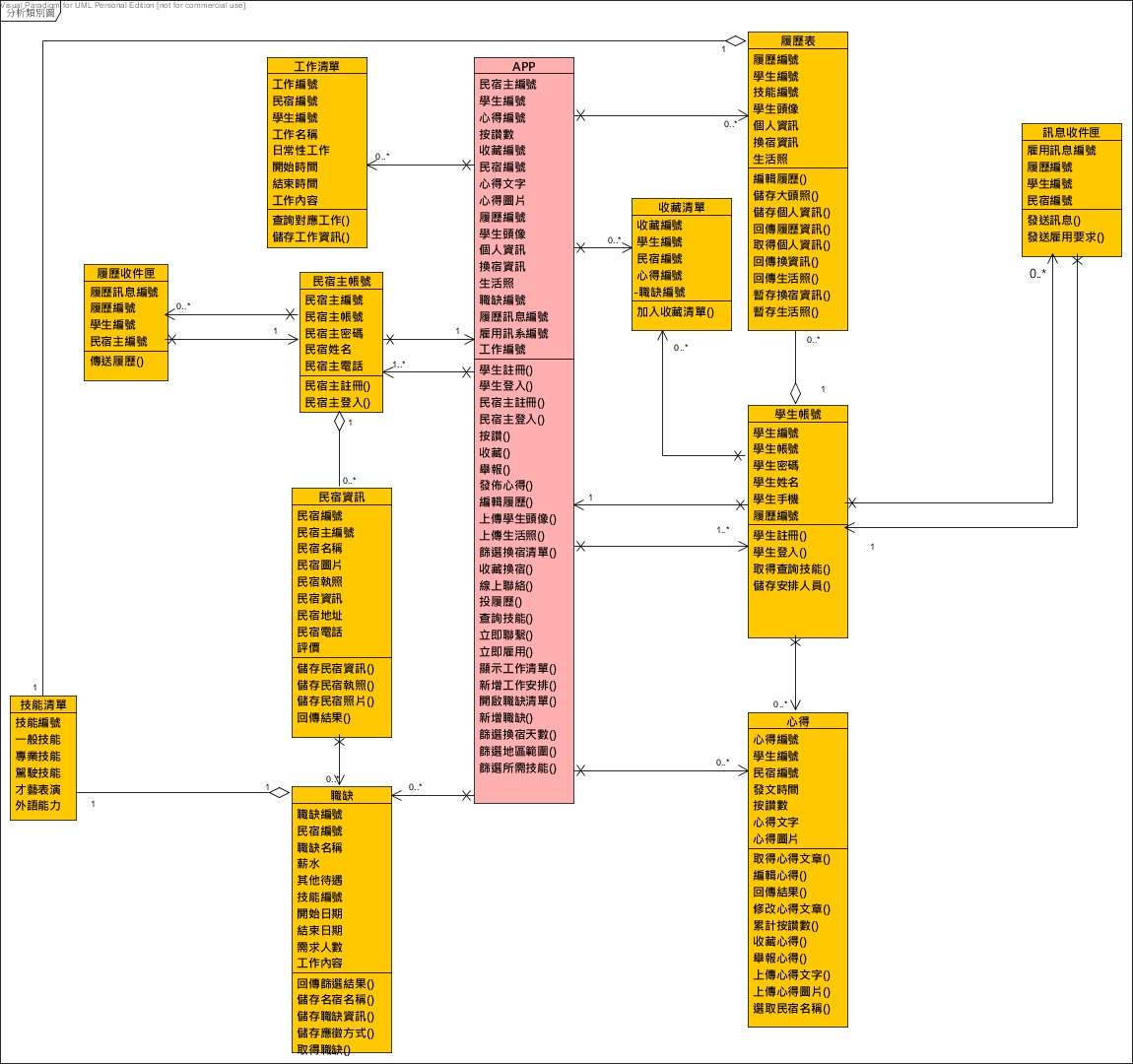


▲ 圖 5-2-1 使用個案圖

* 1. 使用個案描述：活動圖（Activity diagram）

|  |  |
| --- | --- |
| Use Case | 登入 |
| 介面功能 | 讓已註冊的學生和民宿主登入會員，並得以使用後續更多功能。 |
| 登入 | |
| ▲ 圖 5-3-1 活動圖：登入 | |
| Use Case | 註冊 |
| 介面功能 | 當學生和民宿主需要使用更多功能時，便能申請為一家順順的會員來進行。在申請成為會員時，必須填寫基本會員資料以供資料庫進行存取及管理。 |
|  | |
| ▲ 圖 5-3-2 活動圖：註冊 | |
| Use Case | 搜尋人才 |
| 介面功能 | 民宿主可以透過人才搜尋來找到符合的學生來換宿，點擊人才清單後可以瀏覽學生的詳細履歷和立即聯繫、立即雇用。 |
|  | |
| ▲ 圖 5-3-3 活動圖：搜尋人才 | |
| Use Case | 尋找換宿 |
| 介面功能 | 學生可以透過篩選來找到理想的換宿，點擊換宿後可以瀏覽換宿的詳細資訊和進行收藏、立即聯絡、投履歷。 |
|  | |
| ▲ 圖 5-3-4 活動圖：尋找換宿 | |
| Use Case | 刊登履歷 |
| 介面功能 | 學生在編輯履歷中能夠上傳大頭貼、個人資訊、換宿資訊、生活照，完成後可刊登自己的履歷至雇主端的人才搜尋，在日後有發現錯誤都可以隨時在編輯。 |
|  | |
| ▲ 圖 5-3-5活動圖：刊登履歷 | |
| Use Case | 刊登職缺 |
| 介面功能 | 民宿主可以替自己的民宿刊登不同換宿性質且一個以上的職缺。 |
|  | |
| ▲ 圖 5-3-6 活動圖：刊登職缺 | |
| Use Case | 新增民宿資訊 |
| 介面功能 | 雇主能在這新增民宿的相關資訊、民宿外觀圖片及合法營業執照，以供學生們搜尋瀏覽。 |
|  | |
| ▲ 圖 5-3-7 活動圖：新增民宿資訊 | |
| Use Case | 瀏覽工作安排 |
| 介面功能 | 民宿主能夠替學生安排每日的工作行程，並可指派特定學生工作 |
|  | |
| ▲ 圖 5-3-8 活動圖：瀏覽工作安排 | |
| Use Case | 瀏覽心得 |
| 介面功能 | 訪客可以觀看其他會員所發佈的心得進行交流，可以更加瞭解當地的換宿風氣及情況，而有登入的學生可以進行發佈心得。 |
|  | |
| ▲ 圖 5-3-9 活動圖：瀏覽心得 | |
| Use Case | 瀏覽職缺 |
| 介面功能 | 民宿主可以透過瀏覽職缺來即時管理職缺狀況。 |
|  | |
| ▲ 圖 5-3-10 活動圖：瀏覽職缺 | |

* 1. 分析類別圖(Analysis class diagram)

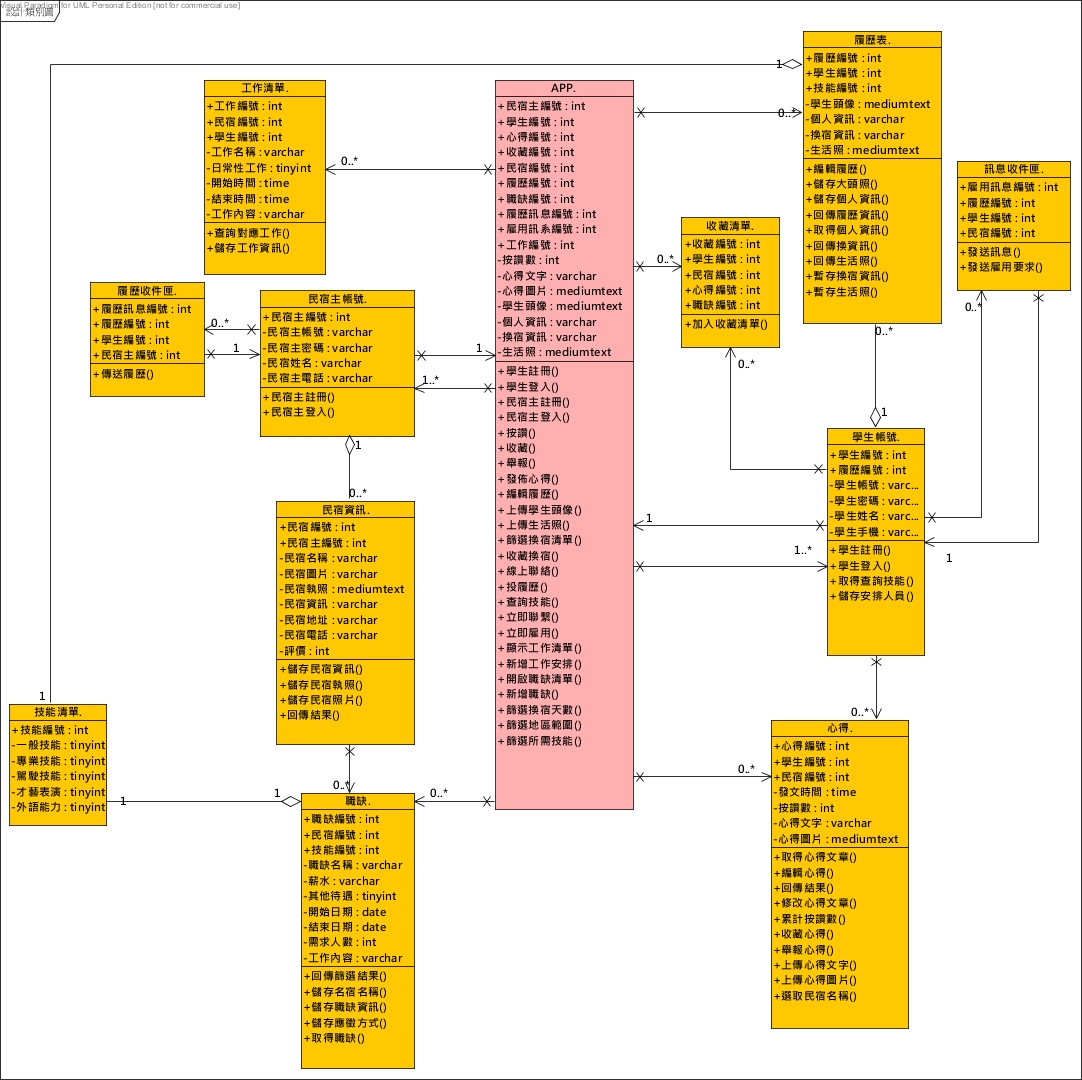


▲ 圖 5-4-1 分析類別圖

1. 設計模型
   1. 循序圖(Sequential diagram)

|  |
| --- |
| 登入 |
| ▲ 圖 6-1-1 循序圖：登入 |
|  |
| ▲ 圖 6-1-2 循序圖：註冊 |
|  |
| ▲ 圖 6-1-3 循序圖：搜尋人才 |
|  |
| ▲ 圖 6-1-4 循序圖：尋找換宿 |
|  |
| ▲ 圖 6-1-5循序圖：瀏覽職缺 |
|  |
| ▲ 圖 6-1-6 循序圖：刊登職缺 |
|  |
| ▲ 圖 6-1-7 循序圖：新增民宿資訊 |
|  |
| ▲ 圖 6-1-8 循序圖：瀏覽工作安排 |
|  |
| ▲ 圖 6-1-9 循序圖：瀏覽心得 |
|  |
| ▲ 圖 6-1-10 循序圖：發佈心得 |
|  |
| ▲ 圖 6-1-11 循序圖：刊登履歷 |

* 1. 設計類別圖(Design class diagram)



▲ 圖 6-2-1 設計類別圖

1. 實作模型
   1. 佈署圖(Deployment diagram)

|  |
| --- |
|  |
| ▲ 圖 7-1-1佈署圖 |

* 1. 套件圖(Package diagram)

|  |
| --- |
|  |
| ▲ 圖 7-2-1 套件圖 |

* 1. 元件圖(Component diagram)

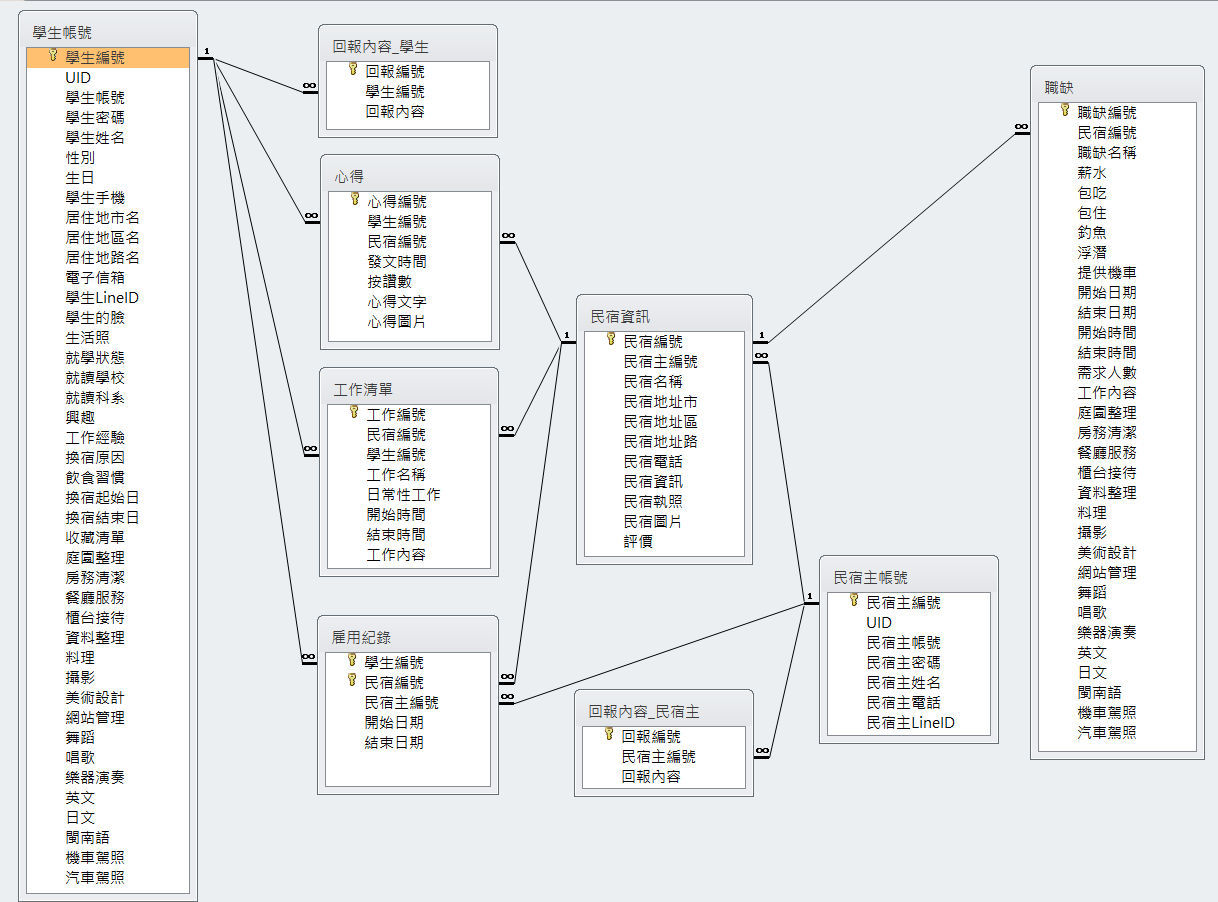
|  |
| --- |
|  |
| ▲ 圖 7-3-1元件圖（雇主端） |

|  |
| --- |
|  |
| ▲ 圖 7-3-2元件圖（學生端） |

* 1. 狀態機(State machine)

|  |
| --- |
|  |
| ▲ 圖 7-4-1 狀態機（雇主端） |
|  |
| ▲ 圖 7-4-2 狀態機（學生端） |

1. 資料庫設計
   1. 資料庫關聯表



▲ 圖 8-1-1 資料關聯圖

* 1. 表格及其Meta data

▼表 8-2-1 資料表描述-學生帳號

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| 學生編號 | int | 5 | 流水編號 | 🗸 |  |
| uid | varchar | 64 | Sha256雜湊 |  |  |
| 學生帳號 | varchar | 50 |  |  |  |
| 學生密碼 | varchar | 25 |  |  |  |
| 學生姓名 | varchar | 10 |  |  |  |
| 性別 | tinyint | 1 | 0:false;1:true |  |  |
| 生日 | date |  |  |  |  |
| 學生手機 | varchar | 10 |  |  |  |
| 居住地市名 | varchar | 10 |  |  |  |
| 居住地區名 | varchar | 10 |  |  |  |
| 居住地路名 | varchar | 10 |  |  |  |
| 電子信箱 | varchar | 50 |  |  |  |
| 學生LineID | varchar | 50 |  |  |  |
| 學生的臉 | mediumtext |  |  |  |  |
| 生活照 | mediumtext |  |  |  |  |
| 就學狀態 | varchar | 3 |  |  |  |
| 就讀學校 | varchar | 20 |  |  |  |
| 就讀科系 | varchar | 20 |  |  |  |
| 興趣 | varchar | 20 |  |  |  |
| 工作經驗 | varchar | 20 |  |  |  |
| 換宿原因 | varchar | 255 |  |  |  |
| 飲食習慣 | tinyint | 1 |  |  |  |
| 換宿起始日 | date |  |  |  |  |
| 換宿結束日 | date |  |  |  |  |
| 收藏清單 | varchar | 255 |  |  |  |
| 庭園整理 | tinyint | 1 | 0:false;1:true |  |  |
| 房務清潔 | tinyint | 1 | 0:false;1:true |  |  |
| 餐廳服務 | tinyint | 1 | 0:false;1:true |  |  |
| 櫃台接待 | tinyint | 1 | 0:false;1:true |  |  |
| 資料整理 | tinyint | 1 | 0:false;1:true |  |  |
| 料理 | tinyint | 1 | 0:false;1:true |  |  |
| 攝影 | tinyint | 1 | 0:false;1:true |  |  |
| 園藝 | tinyint | 1 | 0:false;1:true |  |  |
| 美術設計 | tinyint | 1 | 0:false;1:true |  |  |
| 網站管理 | tinyint | 1 | 0:false;1:true |  |  |
| 舞蹈 | tinyint | 1 | 0:false;1:true |  |  |
| 唱歌 | tinyint | 1 | 0:false;1:true |  |  |
| 樂器演奏 | tinyint | 1 | 0:false;1:true |  |  |
| 英文 | tinyint | 1 | 0:false;1:true |  |  |
| 日文 | tinyint | 1 | 0:false;1:true |  |  |
| 閩南語 | tinyint | 1 | 0:false;1:true |  |  |
| 機車駕照 | tinyint | 1 | 0:false;1:true |  |  |
| 汽車駕照 | tinyint | 1 | 0:false;1:true |  |  |

▼表 8-2-2 資料表描述-民宿主帳號

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| 民宿主編號 | int | 5 | 流水編號 | 🗸 |  |
| uid | varchar | 64 | Sha256雜湊 |  |  |
| 民宿主帳號 | varchar | 50 |  |  |  |
| 民宿主密碼 | varchar | 25 |  |  |  |
| 民宿主姓名 | varchar | 10 |  |  |  |
| 民宿主電話 | tinyint | 10 |  |  |  |
| 民宿主LineID | varchar | 50 |  |  |  |

▼表 8-2-3 資料表描述-民宿資訊

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| 民宿編號 | int | 5 | 流水編號 | 🗸 |  |
| 民宿主編號 | int | 5 | 依來源外鍵 |  | 🗸 |
| 民宿名稱 | varchar | 20 |  |  |  |
| 民宿地址市 | varchar | 3 |  |  |  |
| 民宿地址區 | varchar | 5 |  |  |  |
| 民宿地址路 | varchar | 20 |  |  |  |
| 民宿電話 | varchar | 10 |  |  |  |
| 民宿資訊 | varchar | 50 |  |  |  |
| 民宿執照 | mediumtext |  |  |  |  |
| 民宿圖片 | mediumtext |  |  |  |  |
| 評價 | int | 1 |  |  |  |

▼表 8-2-4 資料表描述-工作清單

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| 工作編號 | int | 5 | 流水編號 | 🗸 |  |
| 民宿編號 | int | 5 | 依來源外鍵 |  | 🗸 |
| 工作名稱 | varchar | 20 |  |  |  |
| 日常性工作 | tinyint | 1 | 0:false;1:true |  |  |
| 開始時間 | time |  |  |  |  |
| 結束時間 | time |  |  |  |  |
| 工作內容 | varchar | 255 |  |  |  |

▼表 8-2-5 資料表描述-職缺

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| 職缺編號 | int | 5 | 流水編號 | 🗸 |  |
| 民宿編號 | int | 5 | 依來源外鍵 |  | 🗸 |
| 職缺名稱 | varchar | 50 |  |  |  |
| 薪水 | tinyint | 20 | 0:false;1:true |  |  |
| 包吃 | tinyint | 1 | 0:false;1:true |  |  |
| 包含住宿 | tinyint | 1 | 0:false;1:true |  |  |
| 包含吃食 | tinyint | 1 | 0:false;1:true |  |  |
| 提供機車 | tinyint | 1 | 0:false;1:true |  |  |
| 浮潛 | tinyint | 1 | 0:false;1:true |  |  |
| 釣魚 | tinyint | 1 | 0:false;1:true |  |  |
| 庭園整理 | tinyint | 1 | 0:false;1:true |  |  |
| 房務清潔 | tinyint | 1 | 0:false;1:true |  |  |
| 餐廳服務 | tinyint | 1 | 0:false;1:true |  |  |
| 櫃台接待 | tinyint | 1 | 0:false;1:true |  |  |
| 資料整理 | tinyint | 1 | 0:false;1:true |  |  |
| 料理 | tinyint | 1 | 0:false;1:true |  |  |
| 攝影 | tinyint | 1 | 0:false;1:true |  |  |
| 園藝 | tinyint | 1 | 0:false;1:true |  |  |
| 美術設計 | tinyint | 1 | 0:false;1:true |  |  |
| 網站管理 | tinyint | 1 | 0:false;1:true |  |  |
| 舞蹈 | tinyint | 1 | 0:false;1:true |  |  |
| 唱歌 | tinyint | 1 | 0:false;1:true |  |  |
| 樂器演奏 | tinyint | 1 | 0:false;1:true |  |  |
| 英文 | tinyint | 1 | 0:false;1:true |  |  |
| 日文 | tinyint | 1 | 0:false;1:true |  |  |
| 閩南語 | tinyint | 1 | 0:false;1:true |  |  |
| 機車駕照 | tinyint | 1 | 0:false;1:true |  |  |
| 汽車駕照 | tinyint | 1 | 0:false;1:true |  |  |
| 開始日期 | date |  |  |  |  |
| 開始時間 | time |  |  |  |  |
| 結束日期 | date |  |  |  |  |
| 結束時間 | time |  |  |  |  |
| 需求人數 | int | 20 |  |  |  |
| 工作內容 | varchar | 1000 |  |  |  |

▼表 8-2-6 資料表描述-心得

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| 心得編號 | int | 5 | 流水編號 | 🗸 |  |
| 學生編號 | int | 5 | 依來源外鍵 |  | 🗸 |
| 民宿編號 | int | 5 | 依來源外鍵 |  | 🗸 |
| 發文時間 | date |  |  |  |  |
| 按讚數 | int | 3 |  |  |  |
| 心得文字 | varchar | 2000 |  |  |  |
| 心得圖片 | mediumtext |  |  |  |  |

▼表 8-2-7 資料表描述-訊息清單

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| 訊息編號 | int | 5 | 流水編號 | 🗸 |  |
| 學生編號 | int | 5 | 依來源外鍵 |  | 🗸 |
| 民宿主編號 | int | 5 | 依來源外鍵 |  | 🗸 |
| 訊息內容 | varchar | 1000 |  |  |  |

▼表 8-2-8 資料表描述-雇用紀錄

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| 學生編號 | int | 5 | 依來源外鍵 | 🗸 | 🗸 |
| 民宿主編號 | int | 5 | 依來源外鍵 | 🗸 | 🗸 |
| 民宿編號 | int | 5 | 依來源外鍵 |  | 🗸 |
| 開始日期 | date |  |  |  |  |
| 結束日期 | date |  |  |  |  |

▼表 8-2-9 資料表描述-回報內容\_民宿主

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| 回報編號 | int | 5 | 流水編號 | 🗸 |  |
| 民宿主編號 | int | 5 | 依來源外鍵 |  | 🗸 |
| 回報內容 | varchar | 1000 | 依來源外鍵 |  |  |

▼表 8-2-10 資料表描述-回報內容\_學生

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| 回報編號 | int | 5 | 流水編號 | 🗸 |  |
| 學生編號 | int | 5 | 依來源外鍵 |  | 🗸 |
| 回報內容 | varchar | 1000 | 依來源外鍵 |  |  |

▼表 8-2-11 資料表描述-Open\_Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| No | int | 11 | 流水編號 | 🗸 |  |
| Updatetime | varchar | 255 |  |  |  |
| ID | varchar | 255 |  |  |  |
| Name | varchar | 255 |  |  |  |
| Address | varchar | 255 |  |  |  |
| Tel | varchar | 255 |  |  |  |

▼表 8-2-12 資料表描述-審核清單

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 欄位名稱 | 資料類型 | 欄位大小 | 欄位格式 | Primary key | Foreign key |
| 認證編號 | int | 5 | 流水編號 | 🗸 |  |
| 民宿名稱 | varchar | 50 | 依來源外鍵 |  | 🗸 |
| 是否合格 | tinyint | 1 |  |  |  |

1. 程式
   1. 程式清單

▼表 9-1-1 雇主端程式清單

|  |  |  |
| --- | --- | --- |
| 編號 | 檔案名稱 | 功能說明 |
| 1 | ApplicationController | 註冊登入的API |
| 2 | Common | 儲存PHP的URL |
| 3 | Calenderview\_Addcalender | 新增工作安排 |
| 4 | Fragment\_Calenderview | 顯示工作安排 |
| 5 | Fragment\_Humansearch | 顯示人才清單 |
| 6 | Fragment\_Jobview | 顯示職缺清單 |
| 7 | getWorksheet | 取得伺服器上JSON資料 |
| 8 | Hotelview\_Addhotel | 新增民宿資訊 |
| 9 | Humansearch\_Info | 顯示詳細人才資訊 |
| 10 | Jobview\_Addjob | 新增刊登職缺 |
| 11 | MainActivityLogin | 登入功能 |
| 12 | MainActivityRegister | 註冊功能 |
| 13 | NavigationActivity | 顯示Fragment各頁面 |
| 14 | Setting\_Hotelinfo | 跳轉至新增民宿資訊 |

▼表 9-1-2 學生端程式清單

|  |  |  |
| --- | --- | --- |
| 編號 | 檔案名稱 | 功能說明 |
| 1 | getAsyncTask | 非同步取得資料 |
| 2 | postUpdateAsyncTask | 非同步上傳資料 |
| 3 | Fragment\_Findhostel | 顯示換宿清單 |
| 4 | Findhostel\_hostelinfo | 顯示民宿詳細資料 |
| 5 | Fragment\_Forum | 顯示換宿心得 |
| 6 | Fragment\_Schedule | 顯示工作排程 |
| 7 | Forum\_Add | 新增心得 |
| 8 | Forum\_Edit | 編輯心得 |
| 9 | Personal\_Resume | 編輯個人履歷 |

* 1. 程式規格描述

▼表 9-2-1 雇主端程式規格描述

|  |  |
| --- | --- |
| 程式名稱 | ApplicationController |
| 目的 | 註冊登入的API |
| 程式碼 | |
| **public class** ApplicationController **extends** Application {  */\*\*  \* Log or request TAG  \*/* **public static final** String ***TAG*** = **"VolleyPatterns"**;  */\*\*  \* Global request queue for Volley  \*/* **private** RequestQueue **mRequestQueue**;  */\*\*  \* A singleton instance of the application class for easy access in other places  \* Singleton物件，簡單的說就是整個app只會有這一個ApplicationController物件，理想上不會產生第二份ApplicationController物件  \*/* **private static** ApplicationController *sInstance*;   @Override  **protected void** attachBaseContext(Context base)  {  **super**.attachBaseContext(base);  MultiDex.*install*(**this**);  }   @Override  **public void** onCreate()  {  **super**.onCreate();   *// initialize the singleton  sInstance* = **this**;  }   */\*\*  \** ***@return*** *ApplicationController singleton instance  \* 要使用ApplicationController一定要先透過此method取得ApplicationController物件  \*/* **public static synchronized** ApplicationController getInstance()  {  **return** *sInstance*;  }   */\*\*  \** ***@return*** *The Volley Request queue, the queue will be created if it is null  \*/* **public** RequestQueue getRequestQueue()  {  *// lazy initialize the request queue, the queue instance will be  // created when it is accessed for the first time* **if** (**mRequestQueue** == **null**)  {  **mRequestQueue** = Volley.*newRequestQueue*(getApplicationContext());  }  **return mRequestQueue**;  } **public void** addToRequestQueue(Request req, String tag)  {  *// set the default tag if tag is empty* req.setTag(TextUtils.*isEmpty*(tag) ? ***TAG*** : tag);   VolleyLog.*d*(**"Adding request to queue: %s"**, req.getUrl());   getRequestQueue().add(req);  } **public void** addToRequestQueue(Request req)  {  *// set the default tag if tag is empty* req.setTag(***TAG***);   getRequestQueue().add(req);  }  **public void** cancelPendingRequests(Object tag)  {  **if** (**mRequestQueue** != **null**)  {  **mRequestQueue**.cancelAll(tag);  }  } } | |

|  |  |
| --- | --- |
| 程式名稱 | Common |
| 目的 | 儲存PHP的URL |
| 程式碼 | |
| **package** ntub107202.hostel.Common;  **public class** Common {  *//--------------------------  //(1) 若連結自己的主機  //--------------------------  // 確認並修改主機的ip位址  // 若為192.168開頭的虛擬IP, 執行時模擬器與主機應使用同一分享器內之網路  //public static String url="http://192.168.56.1:3000";   //--------------------------  //(2) 若連結現有測試主機  //--------------------------* **public static** String *getUrl*=**"http://140.131.114.153/getMysqlJSON"**;  **public static** String *postUrl*=**"http://140.131.114.153/getAndroid"**;  **public static** String *postJob*=**"http://140.131.114.153/postJob"**;  **public static** String *postCalendar*=**"http://140.131.114.153/postCalendar"**;  **public static** String *postQuestionH*=**"http://140.131.114.153/postQuestionH"**;  **public static** String *getHumanSearch*=**"http://140.131.114.153/getHumanSearch"**;  **public static** String *getjob*=**"http://140.131.114.153/getjob"**;  **public static** String *getCalendar*=**"http://140.131.114.153/getSchedule00"**;  **public static** String *postHotel*=**"http://140.131.114.153/postHotel"**;  **public static** String *getHostelinfo*=**"http://140.131.114.153/getHostelinfo"**;  **public static** String *getuid*=**"http://140.131.114.153/getuid"**;  **public static** String *getHostelname*=**"http://140.131.114.153/getHostelname"**;  **public static** String *postResumeHostel2Student*=**"http://140.131.114.153/postResumeHostel2Student"**;  **public static** String *getResumeH*=**"http://140.131.114.153/getResumeH"**;  **public static** String *getCheck*=**"http://140.131.114.153/getCheck"**;  **public static** String *postCheck*=**"http://140.131.114.153/postCheck"**;   } | |

|  |  |
| --- | --- |
| 程式名稱 | Calenderview\_Addcalender |
| 目的 | 新增工作安排 |
| 程式碼 | |
| **public class** Calenderview\_Addcalender **extends** AppCompatActivity {   **private** EditText **work\_start\_time**, **work\_end\_time**,**start\_date**;  **private int mYear**, **mMonth**, **mDay**, **mHour**, **mMinute**;   EditText **edit\_work\_title**;  EditText **edit\_work**;  RadioButton **rad\_work\_type\_always**;  String **rad\_work\_type\_alwaysS**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***calenderview\_addcalender***);  **start\_date** = (EditText) findViewById(R.id.***start\_date***);  **work\_start\_time** = (EditText) findViewById(R.id.***work\_start\_time***);  **work\_end\_time** = (EditText) findViewById(R.id.***work\_end\_time***);   **start\_date**.setOnClickListener(**new** Button.OnClickListener() {  **public void** onClick(View v) {  showDatePickerDialogStart();  }  });   **work\_start\_time**.setOnClickListener(**new** Button.OnClickListener() {  **public void** onClick(View v) {  showTimePickerDialogStart();  }  });  **work\_end\_time**.setOnClickListener(**new** Button.OnClickListener() {  **public void** onClick(View v) {  showTimePickerDialogEnd();  }  });    **edit\_work\_title** = (EditText) findViewById(R.id.***edit\_work\_title***);  **edit\_work** = (EditText) findViewById(R.id.***edit\_work***);  **rad\_work\_type\_always** = (RadioButton)findViewById(R.id.***rad\_work\_type\_always***);  **work\_start\_time** = (EditText) findViewById(R.id.***work\_start\_time***);  **work\_end\_time** = (EditText) findViewById(R.id.***work\_end\_time***);   Button button01 = (Button) findViewById(R.id.***btn\_add\_calendar***);   **work\_start\_time**.addTextChangedListener(**textWatcher1**);  **work\_end\_time**.addTextChangedListener(**textWatcher2**);   **rad\_work\_type\_always**.setChecked(**true**);   **start\_date**.addTextChangedListener(**textWatcher6**); *// Log.v("1123",rad\_work\_type\_alwaysS);  //--------------------------  //取得伺服器上JSON資料  //--------------------------* button01.setOnClickListener(**new** Button.OnClickListener() {  @Override  **public void** onClick(View v) {  **if** (**rad\_work\_type\_always**.isChecked()){  **rad\_work\_type\_alwaysS**=**"1"**;  }**else**{  **rad\_work\_type\_alwaysS**=**"0"**;  }  Log.*v*(**"edit\_work\_title"**, **edit\_work\_title**.getText().toString());  Log.*v*(**"rad\_work\_type\_alwaysS"**,**rad\_work\_type\_alwaysS**);  Log.*v*(**"work\_start\_time"**,**work\_start\_time**.getText().toString());  Log.*v*(**"work\_end\_time"**,**work\_end\_time**.getText().toString());  Log.*v*(**"edit\_work"**,**edit\_work**.getText().toString());  getWorksheet.*postToCalendar*(**edit\_work\_title**.getText().toString(), **rad\_work\_type\_alwaysS**, **work\_start\_time**.getText().toString(), **work\_end\_time**.getText().toString(), **edit\_work**.getText().toString(), **start\_date**.getText().toString());  getWorksheet.*getJSON*();  getWorksheet.*getjobJSON*();  getWorksheet.*getcalendarJSON*();  **try** {  Thread.*sleep*(500); *//1000為1秒* } **catch** (InterruptedException e) { *//* ***TODO Auto-generated catch block*** e.printStackTrace();  }  Intent intent = **new** Intent(Calenderview\_Addcalender.**this**,NavigationActivity.**class**);  intent.putExtra(**"id"**,4);  startActivity(intent);  }  });  }   **public void** showDatePickerDialogStart() {  *// 設定初始日期* **final** Calendar c = Calendar.*getInstance*();  **mYear** = c.get(Calendar.***YEAR***);  **mMonth** = c.get(Calendar.***MONTH***);  **mDay** = c.get(Calendar.***DAY\_OF\_MONTH***);   *// 跳出日期選擇器* DatePickerDialog dpd = **new** DatePickerDialog(**this**,  **new** DatePickerDialog.OnDateSetListener() {  **public void** onDateSet(DatePicker view, **int** year,  **int** monthOfYear, **int** dayOfMonth) {  *// 完成選擇，顯示日期* **start\_date**.setText(year + **"-"** + (monthOfYear + 1) + **"-"** + dayOfMonth);   }  }, **mYear**, **mMonth**, **mDay**);  dpd.show();  }  **public void** showTimePickerDialogStart() {  *// 設定初始時間* **final** Calendar c = Calendar.*getInstance*();  **mHour** = c.get(Calendar.***HOUR\_OF\_DAY***);  **mMinute** = c.get(Calendar.***MINUTE***);   *// 跳出時間選擇器* TimePickerDialog tpd = **new** TimePickerDialog(**this**,  **new** TimePickerDialog.OnTimeSetListener() {  **public void** onTimeSet(TimePicker view, **int** hourOfDay,  **int** minute) {  *// 完成選擇，顯示時間* **if** (minute<10){  **work\_start\_time**.setText(hourOfDay + **":"** + **"0"**+minute);  }**else** {  **work\_start\_time**.setText(hourOfDay + **":"** + minute);  }  }  }, **mHour**, **mMinute**, **false**);  tpd.show();  }  **public void** showTimePickerDialogEnd() {  *// 設定初始時間* **final** Calendar c = Calendar.*getInstance*();  **mHour** = c.get(Calendar.***HOUR\_OF\_DAY***);  **mMinute** = c.get(Calendar.***MINUTE***);   *// 跳出時間選擇器* TimePickerDialog tpd = **new** TimePickerDialog(**this**,  **new** TimePickerDialog.OnTimeSetListener() {  **public void** onTimeSet(TimePicker view, **int** hourOfDay,  **int** minute) {  *// 完成選擇，顯示時間* **if** (minute<10){  **work\_end\_time**.setText(hourOfDay + **":"** + **"0"**+minute);  }**else** {  **work\_end\_time**.setText(hourOfDay + **":"** + minute);  }  }  }, **mHour**, **mMinute**, **false**);  tpd.show();  }  **private** TextWatcher **textWatcher1** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);  Is\_Valid\_Email(**work\_start\_time**);  }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);   }  **public void** Is\_Valid\_Email(EditText edt) {  **try** {  SimpleDateFormat sdf = **new** SimpleDateFormat(**"HH:mm"**);  Date d1 = sdf.parse(**work\_start\_time**.getText().toString());  Date d2 = sdf.parse(**work\_end\_time**.getText().toString());  **long** result = d1.getTime()-d2.getTime();  Log.*d*(**"timeeee"**, String.*valueOf*(d1.getTime()-d2.getTime()));  **if** (result<0){  **work\_end\_time**.setError(**null**);  }**else if** (result == 0) {  **work\_end\_time**.setError(**"結束時間等於開始時間"**);  }**else if**(result>0){  **work\_end\_time**.setError(**"結束時間小於開始時間"**);  }   } **catch** (Exception e) {   }  }  };   **private** TextWatcher **textWatcher2** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);  Is\_Valid\_Email(**work\_end\_time**);  }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);   }  **public void** Is\_Valid\_Email(EditText edt) {  **try** {  SimpleDateFormat sdf = **new** SimpleDateFormat(**"HH:mm"**);  Date d1 = sdf.parse(**work\_start\_time**.getText().toString());  Date d2 = sdf.parse(**work\_end\_time**.getText().toString());  **long** result = d1.getTime()-d2.getTime();  Log.*d*(**"timeeee"**, String.*valueOf*(d1.getTime()-d2.getTime()));  **if** (result<0){  **work\_end\_time**.setError(**null**);  }**else if** (result == 0) {  **work\_end\_time**.setError(**"結束時間等於開始時間"**);  }**else if**(result>0){  **work\_end\_time**.setError(**"結束時間小於開始時間"**);  }   } **catch** (Exception e) {   }  }  };  **private** TextWatcher **textWatcher6** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);  Is\_Valid\_Email(**start\_date**);  }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);   }  **public void** Is\_Valid\_Email(EditText edt) {   }  }; }  co | |

|  |  |
| --- | --- |
| 程式名稱 | Fragment\_Calenderview |
| 目的 | 顯示工作安排 |
| 程式碼 | |
| **public class** Fragment\_Calenderview **extends** Fragment {  **static** TextView *textView30*;  **static** TextView *textView31*;  **static** TextView *textView32*;  **static** TextView *textView33*;  **static** TextView *textView34*;  **static** String[] *row* = **new** String[100];  **static** String *row9*;  **static** String *row10*;  **static** String *row11*;  **static** String *row12*;  RecyclerView **mList**;  ArrayList<String> **myDataset**;  ArrayList<String> **myDataset2**;  ArrayList<String> **myDataset3**;  ArrayList<String> **myDataset4**;  MyAdapter **myAdapter**;  **static** LinearLayoutManager *layoutManager*;  @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {  View view=inflater.inflate(R.layout.***fragment\_calender\_view***,container,**false**);  *textView30* = (TextView)view.findViewById(R.id.***text001***); **mList** = (RecyclerView)view.findViewById(R.id.***list\_view***);  **if**(getWorksheet.*getRow9*(0) != **null**){  *textView30*.setVisibility(View.***INVISIBLE***);  }  FloatingActionButton fab = (FloatingActionButton)view.findViewById(R.id.***fab\_go\_to\_add\_calendar***) ;  fab.setOnClickListener(**new** Button.OnClickListener(){  @Override  **public void** onClick(View v) {  Intent intent = **new** Intent(getActivity(),Calenderview\_Addcalender.**class**);  startActivity(intent);  }  });  **return** view;  }  @Override  **public void** onHiddenChanged(**boolean** hidden) {  **super**.onHiddenChanged(hidden);  **if**(hidden){  *//pause* }**else**{  **myDataset** = **new** ArrayList<>();  **myDataset2** = **new** ArrayList<>();  **myDataset3** = **new** ArrayList<>();  **myDataset4** = **new** ArrayList<>();  **myAdapter** = **new** MyAdapter(**myDataset**);  **for**(**int** i = 0; i < getWorksheet.*calendarLength*; i++){ *// myDataset.add(i + "");* **myDataset**.add(getWorksheet.*getRow9*(i));  **myDataset2**.add(getWorksheet.*getRow10*(i));  **myDataset3**.add(getWorksheet.*getRow11*(i));  **myDataset4**.add(getWorksheet.*getRow12*(i));  } *// mList = (RecyclerView)view.findViewById(R.id.list\_view);  layoutManager* = **new** LinearLayoutManager(getActivity());  *layoutManager*.setOrientation(LinearLayoutManager.***VERTICAL***);  **mList**.setLayoutManager(*layoutManager*);  **mList**.setAdapter(**myAdapter**);  *//resume* }  }  **public class** MyAdapter **extends** RecyclerView.Adapter<MyAdapter.ViewHolder> {  **private** List<String> **mData**;  **public class** ViewHolder **extends** RecyclerView.ViewHolder {  **public** TextView **TextView0001**, **TextView0002** ,**TextView0003**, **TextView0004**,**TextView33**;  **public** ViewHolder(View v) {  **super**(v);  **TextView0001** = (TextView) v.findViewById(R.id.***textView0001***);  **TextView0002** = (TextView) v.findViewById(R.id.***textView0002***);  **TextView0003** = (TextView) v.findViewById(R.id.***textView0003***);  **TextView0004** = (TextView) v.findViewById(R.id.***textView0004***);  **TextView33** = (TextView) v.findViewById(R.id.***textView33***);  **TextView33**.setOnClickListener(**new** View.OnClickListener() {  Boolean **flag** = **true**;  @Override  **public void** onClick(View v) {  **if** (**flag**) {  **flag** = **false**;  **TextView0002**.setMaxLines(5);  **TextView0002**.setEllipsize(**null**); *// 展開* **TextView33**.setText(**"... 摺疊內容"**);  } **else** {  **flag** = **true**;  **TextView0002**.setLines(1);  **TextView0002**.setEllipsize(TextUtils.TruncateAt.***END***); *// 收縮* **TextView33**.setText(**"... 查看更多"**);  }  }  });  }  }  **public** MyAdapter(List<String> data) {  **mData** = data;  }  @Override  **public** MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType)   View v = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.***calenderview\_item***, parent, **false**);  ViewHolder vh = **new** ViewHolder(v);  **return** vh;  }  @Override  **public void** onBindViewHolder(ViewHolder holder, **int** position) {  holder.**TextView0001**.setText(**mData**.get(position));  holder.**TextView0002**.setText(**myDataset2**.get(position));  holder.**TextView0003**.setText(**myDataset3**.get(position));  holder.**TextView0004**.setText(**myDataset4**.get(position));  }   @Override  **public int** getItemCount() {  **return mData**.size();  }  } } | |

|  |  |
| --- | --- |
| 程式名稱 | Fragment\_Humansearch |
| 目的 | 顯示人才清單 |
| 程式碼 | |
| **public class** Fragment\_Humansearch **extends** Fragment {  RecyclerView **mList**;  ArrayList<String> **myDataset**;  ArrayList<String> **myDataset19**;  MyAdapter **myAdapter**;  **static** LinearLayoutManager *layoutManager*;  @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {  View view=inflater.inflate(R.layout.***fragment\_humansearch***,container,**false**);  **mList** = (RecyclerView)view.findViewById(R.id.***list\_view***);  setJobview();  **return** view;  }  @Override  **public void** onHiddenChanged(**boolean** hidden) {  **super**.onHiddenChanged(hidden);  **if**(hidden){  *//pause* }**else**{  setJobview();  *//resume* }  }  **public void** setJobview(){  **myDataset** = **new** ArrayList<>();  ～  **myDataset19** = **new** ArrayList<>();  **myAdapter** = **new** MyAdapter(**myDataset**);  **for**(**int** i = 0; i < getWorksheet.*humanLength*; i++){ *// myDataset.add(i + "");* **myDataset**.add(getWorksheet.*getRow13*(i));  ～  **myDataset19**.add(getWorksheet.*getRow31*(i));   Log.*d*(**"get0000"**, **myDataset5**.get(i)+**"resume"**);  } *// mList = (RecyclerView)view.findViewById(R.id.list\_view);  layoutManager* = **new** LinearLayoutManager(getActivity());  *layoutManager*.setOrientation(LinearLayoutManager.***VERTICAL***);  **mList**.setLayoutManager(*layoutManager*);  **mList**.setAdapter(**myAdapter**);  }  **public class** MyAdapter **extends** RecyclerView.Adapter<MyAdapter.ViewHolder> {  **private** List<String> **mData**;  **private** Context **mContext**;   **public class** ViewHolder **extends** RecyclerView.ViewHolder {  **public** TextView **TextView0001**, **TextView0002** ,**TextView0003**, **TextView0004**;  **public** ImageView **Img01**;  **public** CardView **cardView**;   **public** ViewHolder(View v) {  **super**(v);  **TextView0001** = (TextView) v.findViewById(R.id.***textView0001***);  **TextView0002** = (TextView) v.findViewById(R.id.***textView0002***);  **TextView0003** = (TextView) v.findViewById(R.id.***textView0003***);  **TextView0004** = (TextView) v.findViewById(R.id.***textView0004***);  **Img01** = (ImageView) v.findViewById(R.id.***img01***);  **cardView** = (CardView) v.findViewById(R.id.***card\_view***);  }  }   **public** MyAdapter(List<String> data) {  **mData** = data;  }   @Override  **public** MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {  **mContext** = parent.getContext();  View v = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.***humansearch\_item***, parent, **false**);  MyAdapter.ViewHolder vh = **new** MyAdapter.ViewHolder(v);   vh.**cardView**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Intent i = **new** Intent(**mContext**, Humansearch\_Info.**class**);  i.putExtra(**"row1"**,**myDataset**.get(vh.getAdapterPosition()));  ～  i.putExtra(**"row19"**,**myDataset19**.get(vh.getAdapterPosition()));  **mContext**.startActivity(i);  }  });  **return** vh;  }   @Override  **public void** onBindViewHolder(ViewHolder holder, **int** position) {  holder.**TextView0001**.setText(**mData**.get(position));  holder.**TextView0002**.setText(**myDataset2**.get(position));  holder.**TextView0003**.setText(**myDataset3**.get(position));  holder.**TextView0004**.setText(**myDataset4**.get(position));  holder.**Img01**.setImageBitmap(stringToBitmap(**myDataset5**.get(position)));  Log.*d*(**"get0000"**, **"position"** + **myDataset5**.get(position));  }   @Override  **public int** getItemCount() {  **return mData**.size();  }  }  **public** Bitmap stringToBitmap(String string) {  Bitmap bitmap = **null**;  **try** {  **byte**[] bitmapArray = Base64.*decode*(string, Base64.***DEFAULT***);  bitmap = BitmapFactory.*decodeByteArray*(bitmapArray, 0, bitmapArray.**length**);  **return** bitmap;  } **catch** (NullPointerException e) {  e.getMessage();  **return null**;  } **catch** (OutOfMemoryError e) {  **return null**;  }  } } | |

|  |  |
| --- | --- |
| 程式名稱 | Fragment\_Jobview |
| 目的 | 顯示職缺清單 |
| 程式碼 | |
| **public class** Fragment\_Jobview **extends** Fragment {  **private** Fragment\_Jobview **Fragment2JobView**;  **static** TextView *textView0*;  **static** TextView *textView1*;  **static** TextView *textView2*;  **static** TextView *textView3*;  **static** TextView *textView4*;  **static** TextView *textView30*;  **static** String[] *row* = **new** String[100];  **static** String *row5*;  **static** String *row6*;  **static** String *row7*;  **static** String *row8*;  **static** String *row00*;  RecyclerView **mList**;  ArrayList<String> **myDataset**;  ArrayList<String> **myDataset2**;  ArrayList<String> **myDataset3**;  ArrayList<String> **myDataset4**;  MyAdapter **myAdapter**;  **static** LinearLayoutManager *layoutManager*;  FragmentTransaction **beginTransaction**;  @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {  View view=inflater.inflate(R.layout.***fragment\_job\_view***,container,**false**);  FragmentManager fragmentMgr = getFragmentManager();  **beginTransaction** = getFragmentManager().beginTransaction();  *textView30* = (TextView)view.findViewById(R.id.***text01***); **mList** = (RecyclerView)view.findViewById(R.id.***list\_view***);  **if**(getWorksheet.*getRow5*(0) != **null**){  *textView30*.setVisibility(View.***INVISIBLE***);} FloatingActionButton fab = (FloatingActionButton)view.findViewById(R.id.***fab\_go\_to\_add\_job***) ;  fab.setOnClickListener(**new** Button.OnClickListener(){  @Override  **public void** onClick(View v) {  Intent intent = **new** Intent(getActivity(),Jobview\_Addjob.**class**);  startActivity(intent);  }});  **return** view;}  @Override  **public void** onHiddenChanged(**boolean** hidden) {  **super**.onHiddenChanged(hidden);  **if**(hidden){  *//pause* }**else**{  setJobview();  *//resume* }  }  **public void** setJobview(){  **myDataset** = **new** ArrayList<>();  **myDataset2** = **new** ArrayList<>();  **myDataset3** = **new** ArrayList<>();  **myDataset4** = **new** ArrayList<>();  **myAdapter** = **new** MyAdapter(**myDataset**);  **for**(**int** i = 0; i < getWorksheet.*jobLength*; i++){**myDataset**.add(getWorksheet.*getRow5*(i));  **myDataset2**.add(getWorksheet.*getRow6*(i));  **myDataset3**.add(getWorksheet.*getRow7*(i));  **myDataset4**.add(getWorksheet.*getRow8*(i));}  *layoutManager* = **new** LinearLayoutManager(getActivity());  *layoutManager*.setOrientation(LinearLayoutManager.***VERTICAL***);  **mList**.setLayoutManager(*layoutManager*);  **mList**.setAdapter(**myAdapter**);  }  **public class** MyAdapter **extends** RecyclerView.Adapter<MyAdapter.ViewHolder> {  **private** List<String> **mData**;  **public class** ViewHolder **extends** RecyclerView.ViewHolder {  **public** TextView **TextView0001**, **TextView0002** ,**TextView0003**, **TextView0004**;  **public** ViewHolder(View v) {  **super**(v);  **TextView0001** = (TextView) v.findViewById(R.id.***textView0001***);  **TextView0002** = (TextView) v.findViewById(R.id.***textView0002***);  **TextView0003** = (TextView) v.findViewById(R.id.***textView0003***);  **TextView0004** = (TextView) v.findViewById(R.id.***textView0004***);  } }  **public** MyAdapter(List<String> data) {  **mData** = data;  }  @Override  **public** MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {  View v = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.***jobview\_item***, parent,**false**);  ViewHolder vh = **new** ViewHolder(v);  **return** vh;}  @Override  **public void** onBindViewHolder(ViewHolder holder, **int** position) {  holder.**TextView0001**.setText(**mData**.get(position));  holder.**TextView0002**.setText(**myDataset2**.get(position));  holder.**TextView0003**.setText(**myDataset3**.get(position));  holder.**TextView0004**.setText(**myDataset4**.get(position));}  @Override  **public int** getItemCount() {  **return mData**.size();  }}} | |

|  |  |
| --- | --- |
| 程式名稱 | getWorksheet |
| 目的 | 取得伺服器上JSON資料 |
| 程式碼 | |
| **public class** getWorksheet {   **private static** String[] *row1*= **new** String[100];  ～  **private static** String[] *row49*= **new** String[100];  **private static** String[] *row111*= **new** String[100];  ～  **private static** String[] *row115*= **new** String[100];  **private static** String[] *row17487*= **new** String[100];  **private static** String[] *row510*= **new** String[100];  **private static** String[] *row999*= **new** String[100];  **private static int** *worksheetLength* ;  **public static int** *humanLength*;  **public static int** *jobLength*;  **public static int** *calendarLength*;  **public static int** *hostelinfoLength*;   **public static void** getJSON() {   getAsyncTask myAsyncTask = **new** getAsyncTask(**new** getAsyncTask.TaskListener() {  @Override  **public void** onFinished(String result) {  **try** {  JSONObject object = **new** JSONObject(result);   JSONArray jsonArray = object.getJSONArray(**"result"**);  *worksheetLength* = jsonArray.length();   **int** j=0;   **for** (**int** i = 0 ; i<*getWorksheetLength*() ; i++){  *row1*[i] = jsonArray.getJSONObject(i).getString(**"row1"**);  *row2*[i] = jsonArray.getJSONObject(i).getString(**"row2"**);  *row3*[i] = jsonArray.getJSONObject(i).getString(**"row3"**);  *row4*[i] = jsonArray.getJSONObject(i).getString(**"row4"**);  Log.*v*(**"Row1"**, *row1*[i]);  }  } **catch** (Exception e) {  Log.*v*(**"ABC"**, Log.*getStackTraceString*(e));  }  }  });  myAsyncTask.execute(Common.*getUrl*);  }   **public static void** getuidJSON() {   getAsyncTask myAsyncTask = **new** getAsyncTask(**new** getAsyncTask.TaskListener() {  @Override  **public void** onFinished(String result) {  **try** {  JSONObject object = **new** JSONObject(result);   JSONArray jsonArray = object.getJSONArray(**"result"**);  **int** j=0;  Log.*v*(**"get0000"**,*jobLength* + **"ergegje"**);  **for** (**int** i = 0 ; i<*jobLength* ; i++){  *row999*[i] = jsonArray.getJSONObject(i).getString(**"row1"**);  Log.*v*(**"Row5"**, *row5*[i]);   }  } **catch** (Exception e) {  Log.*v*(**"ABC"**, Log.*getStackTraceString*(e));  }  }  });  myAsyncTask.execute(Common.*getuid*);  }   **public static void** gethumansearchJSON() {   getAsyncTask myAsyncTask = **new** getAsyncTask(**new** getAsyncTask.TaskListener() {  @Override  **public void** onFinished(String result) {  **try** {  JSONObject object = **new** JSONObject(result);   JSONArray jsonArray = object.getJSONArray(**"result"**);  *worksheetLength* = jsonArray.length();  *humanLength* = *worksheetLength*;  **int** j=0;  Log.*v*(**"get0000"**,*humanLength* + **"human"**);  **for** (**int** i = 0 ; i<*humanLength* ; i++){  *row13*[i] = jsonArray.getJSONObject(i).getString(**"name"**);  *row14*[i] = jsonArray.getJSONObject(i).getString(**"school"**);  *row15*[i] = jsonArray.getJSONObject(i).getString(**"jobExp"**);  *row16*[i] = jsonArray.getJSONObject(i).getString(**"exchangeReason"**);  *row17*[i] = jsonArray.getJSONObject(i).getString(**"face"**);  *row18*[i] = jsonArray.getJSONObject(i).getString(**"gender"**);  *row19*[i] = jsonArray.getJSONObject(i).getString(**"birth"**);  *row20*[i] = jsonArray.getJSONObject(i).getString(**"cellphone"**);  *row21*[i] = jsonArray.getJSONObject(i).getString(**"address"**);  *row22*[i] = jsonArray.getJSONObject(i).getString(**"email"**);  *row23*[i] = jsonArray.getJSONObject(i).getString(**"lifePhoto"**);  *row24*[i] = jsonArray.getJSONObject(i).getString(**"studyState"**);  *row25*[i] = jsonArray.getJSONObject(i).getString(**"interest"**);  *row26*[i] = jsonArray.getJSONObject(i).getString(**"eatingHabit"**);  *row27*[i] = jsonArray.getJSONObject(i).getString(**"startingDate"**);  *row28*[i] = jsonArray.getJSONObject(i).getString(**"endingDate"**);  *row29*[i] = jsonArray.getJSONObject(i).getString(**"collectionList"**);  *row30*[i] = jsonArray.getJSONObject(i).getString(**"department"**);  *row31*[i] = jsonArray.getJSONObject(i).getString(**"gardening"**);  *row40*[i] = jsonArray.getJSONObject(i).getString(**"studentAccount"**);  *row49*[i] = jsonArray.getJSONObject(i).getString(**"lineId"**);  *row510*[i] = jsonArray.getJSONObject(i).getString(**"url"**);   Log.*v*(**"Row17"**, *row17*[i]);  Log.*v*(**"Rowline"**, *row49*[i]);   }  } **catch** (Exception e) {  Log.*v*(**"ABC"**, Log.*getStackTraceString*(e));  }  }  });  myAsyncTask.execute(Common.*getHumanSearch*);  }   **public static void** getjobJSON() {   NavigationAsyncTask myNavigationAsyncTask = **new** NavigationAsyncTask(**new** NavigationAsyncTask.TaskListener() {  @Override  **public void** onFinished(String result) {  **try**{  JSONObject object = **new** JSONObject(result);  JSONArray jsonArray = object.getJSONArray(**"result"**);   *worksheetLength* = jsonArray.length();  *jobLength* = *worksheetLength*;   **for** (**int** i = 0 ; i < jsonArray.length(); i++){  *row111*[i] = jsonArray.getJSONObject(i).getString(**"row1"**);  *row112*[i] = jsonArray.getJSONObject(i).getString(**"row2"**);  *row113*[i] = jsonArray.getJSONObject(i).getString(**"row3"**);  *row114*[i] = jsonArray.getJSONObject(i).getString(**"row4"**);  }  JSONArray jsonArray2 = object.getJSONArray(**"result2"**);   **for**(**int** i = 0 ; i< jsonArray2.length();i++){  *row115*[i] = jsonArray2.getJSONObject(i).getString(**"row6"**);  Log.*v*(**"uuuu"**,*row115*[i]);  }   *//-----------------------------------------取得目前版本----------------------------------* }**catch**(Exception e){   }  }  });  myNavigationAsyncTask.execute(Common.*getjob*, MainActivityLogin.*getUser*()); *//第一個參數是Common的網址,第二個是要上傳的值  /\*  getAsyncTask myAsyncTask = new getAsyncTask(new getAsyncTask.TaskListener() {  @Override  public void onFinished(String result) {  try {  JSONObject object = new JSONObject(result);   JSONArray jsonArray = object.getJSONArray("result");  worksheetLength = jsonArray.length();  jobLength = worksheetLength;  int j=0;  Log.v("get0000",jobLength + "ergegje");  for (int i = 0 ; i<jobLength ; i++){  row5[i] = jsonArray.getJSONObject(i).getString("row1");  row6[i] = jsonArray.getJSONObject(i).getString("row2");  row7[i] = jsonArray.getJSONObject(i).getString("row3");  row8[i] = jsonArray.getJSONObject(i).getString("row4");  Log.v("Row5", row5[i]);   }  } catch (Exception e) {  Log.v("ABC", Log.getStackTraceString(e));  }  }  });  myAsyncTask.execute(Common.getjob);  \*/* }  **public static void** gethostelnameJSON() {   getAsyncTask myAsyncTask = **new** getAsyncTask(**new** getAsyncTask.TaskListener() {  @Override  **public void** onFinished(String result) {  **try** {  JSONObject object = **new** JSONObject(result);   JSONArray jsonArray = object.getJSONArray(**"result"**);  *worksheetLength* = jsonArray.length();  *hostelnameLength* = *worksheetLength*;  **int** j=0;   **for** (**int** i = 0 ; i<*getWorksheetLength*() ; i++){  *row38*[i] = jsonArray.getJSONObject(i).getString(**"row1"**);   Log.*v*(**"66668"**, getWorksheet.*row38*[i]); *// Log.v("66668", getWorksheet.row35[i]);* }  } **catch** (Exception e) {  Log.*v*(**"ABC"**, Log.*getStackTraceString*(e));  }  }  });  myAsyncTask.execute(Common.*getHostelname*);  }  **public static void** getcalendarJSON() {   getAsyncTask myAsyncTask = **new** getAsyncTask(**new** getAsyncTask.TaskListener() {  @Override  **public void** onFinished(String result) {  **try** {  JSONObject object = **new** JSONObject(result);   JSONArray jsonArray = object.getJSONArray(**"result"**);  *worksheetLength* = jsonArray.length();  *calendarLength* = *worksheetLength*;  **int** j=0;   **for** (**int** i = 0 ; i<*getWorksheetLength*() ; i++){  *row9*[i] = jsonArray.getJSONObject(i).getString(**"row1"**);  *row10*[i] = jsonArray.getJSONObject(i).getString(**"row2"**);  *row11*[i] = jsonArray.getJSONObject(i).getString(**"row3"**);  *row12*[i] = jsonArray.getJSONObject(i).getString(**"row4"**);  *row36*[i] = jsonArray.getJSONObject(i).getString(**"row5"**);  Log.*v*(**"Row5"**, *row36*[i]);  }  } **catch** (Exception e) {  Log.*v*(**"ABC"**, Log.*getStackTraceString*(e));  }  }  });  myAsyncTask.execute(Common.*getCalendar*);  }  **public static void** gethostelinfoJSON() {   NavigationAsyncTask myNavigationAsyncTask = **new** NavigationAsyncTask(**new** NavigationAsyncTask.TaskListener() {  @Override  **public void** onFinished(String result) {  **try** {  JSONObject object = **new** JSONObject(result);  JSONArray jsonArray = object.getJSONArray(**"result"**);  *worksheetLength* = jsonArray.length();  *hostelinfoLength* = *worksheetLength*;  **int** j=0;  Log.*v*(**"0000000"**,*hostelinfoLength* + **"66666666"**);  **for** (**int** i = 0 ; i<*hostelinfoLength* ; i++){  *row32*[i] = jsonArray.getJSONObject(i).getString(**"row1"**);  *row33*[i] = jsonArray.getJSONObject(i).getString(**"row2"**);  *row34*[i] = jsonArray.getJSONObject(i).getString(**"row3"**);  *row35*[i] = jsonArray.getJSONObject(i).getString(**"row4"**);  *row39*[i] = jsonArray.getJSONObject(i).getString(**"row5"**);  Log.*v*(**"row39"**, *row33*[i]);  Log.*v*(**"row39"**, *row39*[i]);   }  Setting\_Hotelinfo.*start\_sethostelname* = **true**;  } **catch** (Exception e) { *// Log.v("ABC", Log.getStackTraceString(e));* }  }  });  myNavigationAsyncTask.execute(Common.*getHostelinfo*, MainActivityLogin.*getUser*());  }  **public static void** getResumeJSON() {   NavigationAsyncTask myNavigationAsyncTask = **new** NavigationAsyncTask(**new** NavigationAsyncTask.TaskListener() {  @Override  **public void** onFinished(String result) {  **try** {  JSONObject object = **new** JSONObject(result);  JSONArray jsonArray = object.getJSONArray(**"result"**);  *worksheetLength* = jsonArray.length();  *resumeLength* = *worksheetLength*;  **int** j=0;  Log.*v*(**"0000000"**,*resumeLength* + **"66666666"**);  **for** (**int** i = 0 ; i<*resumeLength* ; i++){  *row41*[i] = jsonArray.getJSONObject(i).getString(**"row1"**);  *row42*[i] = jsonArray.getJSONObject(i).getString(**"row2"**);  *row43*[i] = jsonArray.getJSONObject(i).getString(**"row3"**);  *row44*[i] = jsonArray.getJSONObject(i).getString(**"row4"**);  *row45*[i] = jsonArray.getJSONObject(i).getString(**"row5"**);  *row46*[i] = jsonArray.getJSONObject(i).getString(**"row6"**);  *row47*[i] = jsonArray.getJSONObject(i).getString(**"row7"**);  *row48*[i] = jsonArray.getJSONObject(i).getString(**"row8"**);  Log.*v*(**"row41"**, *row41*[i]);*//履歷編號* Log.*v*(**"row42"**, *row42*[i]);*//學生帳號* Log.*v*(**"row43"**, *row43*[i]);*//民宿編號* Log.*v*(**"row44"**, *row44*[i]);*//學生姓名* Log.*v*(**"row45"**, *row45*[i]);*//就讀學校* Log.*v*(**"row46"**, *row46*[i]);*//工作經驗* Log.*v*(**"row47"**, *row47*[i]);*//換宿原因* Log.*v*(**"row48"**, *row48*[i]);*//學生的臉* }  *//Setting\_Hotelinfo.start\_sethostelname = true;* } **catch** (Exception e) { *// Log.v("ABC", Log.getStackTraceString(e));* }  }  });  myNavigationAsyncTask.execute(Common.*getResumeH*, MainActivityLogin.*getUser*());  }  **public static void** getcheckJSON() {   NavigationAsyncTask myNavigationAsyncTask = **new** NavigationAsyncTask(**new** NavigationAsyncTask.TaskListener() {  @Override  **public void** onFinished(String result) {  **try** {  JSONObject object = **new** JSONObject(result);  JSONArray jsonArray = object.getJSONArray(**"result"**);  *worksheetLength* = jsonArray.length();  *checkLength* = *worksheetLength*;  **int** j=0;  Log.*v*(**"0000000"**,*checkLength* + **"66666666"**);  **for** (**int** i = 0 ; i<*checkLength* ; i++){  *row17487*[i] = jsonArray.getJSONObject(i).getString(**"row1"**);   Log.*v*(**"row17487"**, *row17487*[i]);  Log.*v*(**"row17487"**, Hotelview\_Addhotel.*getID*());   }  **if** (*row17487*[0] != **null**){  *postTocheck*(Hotelview\_Addhotel.*getID*(),**"1"**);  Log.*v*(**"row17487"**, **"111111111111111111"**);  }**else** {  *postTocheck*(Hotelview\_Addhotel.*getID*(),**"0"**);  Log.*v*(**"row17487"**, **"000000000000000000"**);  }  } **catch** (Exception e) { *// Log.v("ABC", Log.getStackTraceString(e));* }  }  });  myNavigationAsyncTask.execute(Common.*getCheck*, Hotelview\_Addhotel.*getID*());  }  **public static void** postToPHP(String row1, String row2, String row3) {  postUpdateAsyncTask myAsyncTask = **new** postUpdateAsyncTask(**new** postUpdateAsyncTask.TaskListener() {  @Override  **public void** onFinished(String result) {  }  });  **if**(!myAsyncTask.isCancelled()) {  *//執行上傳動作* myAsyncTask.execute(Common.*postUrl* ,row1,row2,row3);  }  }  **public static void** postToJob(String row1, String row2, String row3, String row4, String row5, String row6, String row7, String row8, String row9, String row10, String row11, String row12, String row13) {  postUpdateAsyncTask myAsyncTask = **new** postUpdateAsyncTask(**new** postUpdateAsyncTask.TaskListener() {  @Override  **public void** onFinished(String result) {  }  });  **if**(!myAsyncTask.isCancelled()) {  *//執行上傳動作* myAsyncTask.execute(Common.*postJob* ,row1,row2,row3,row4,row5,row6,row7,row8,row9,row10,row11,row12,row13);  }  }  **public static void** postTocheck(String row1, String row2) {  postUpdateAsyncTask6 myAsyncTask = **new** postUpdateAsyncTask6(**new** postUpdateAsyncTask6.TaskListener() {  @Override  **public void** onFinished(String result) {  }  });  **if**(!myAsyncTask.isCancelled()) {  *//執行上傳動作* myAsyncTask.execute(Common.*postCheck* ,row1,row2);  }  }   **public static void** postToCalendar(String row1, String row2, String row3, String row4, String row5,String row6) {  postUpdateAsyncTask2 myAsyncTask = **new** postUpdateAsyncTask2(**new** postUpdateAsyncTask2.TaskListener() {  @Override  **public void** onFinished(String result) {  }  });  **if**(!myAsyncTask.isCancelled()) {  *//執行上傳動作* myAsyncTask.execute(Common.*postCalendar* ,row1,row2,row3,row4,row5,row6);  }  }  **public static void** postToQuestion(String row1 , String row2) {  postUpdateAsyncTask3 myAsyncTask = **new** postUpdateAsyncTask3(**new** postUpdateAsyncTask3.TaskListener() {  @Override  **public void** onFinished(String result) {  }  });  **if**(!myAsyncTask.isCancelled()) {  *//執行上傳動作* myAsyncTask.execute(Common.*postQuestionH* ,row1,row2);  }  }  **public static void** postToResume(String row1 , String row2, String row3) {  postUpdateAsyncTask5 myAsyncTask = **new** postUpdateAsyncTask5(**new** postUpdateAsyncTask5.TaskListener() {  @Override  **public void** onFinished(String result) {  }  });  **if**(!myAsyncTask.isCancelled()) {  *//執行上傳動作* myAsyncTask.execute(Common.*postResumeHostel2Student* ,row1,row2,row3);  }  }  **public static void** postToHotel(String row1, String row2, String row3, String row4, String row5, String row6, String row7, String row8 , String row9 , String row10, String row11) {  postUpdateAsyncTask4 myAsyncTask = **new** postUpdateAsyncTask4(**new** postUpdateAsyncTask4.TaskListener() {  @Override  **public void** onFinished(String result) {  }  });  **if**(!myAsyncTask.isCancelled()) {  *//執行上傳動作* myAsyncTask.execute(Common.*postHotel* ,row1,row2,row3,row4,row5,row6,row7,row8,row9,row10,row11);  }  }  **public static int** getWorksheetLength(){  **return** *worksheetLength*;  }  **public static** String getRow5(**int** i){**return** *row5*[i];}  ～  **public static** String getRow35(**int** i){**return** *row49*[i];}  **hopublic static** String getRow5(**int** i){**return** *row111*[i];}  ～  **public static** String getRow35(**int** i){**return** *row115*[i];}  **public static** String getRow510(**int** i){**return** *row510*[i];}  } | |

|  |  |
| --- | --- |
| 程式名稱 | Hotelview\_Addhotel |
| 目的 | 新增民宿資訊 |
| 程式碼 | |
| **public class** Hotelview\_Addhotel **extends** AppCompatActivity {  EditText **edit\_hotel\_name**;  Spinner **spin\_address**;  Spinner **spin\_area**;  EditText **edit\_hotel\_address**;  EditText **edit\_hotel\_phone**;  EditText **edit\_hotel\_info**;  EditText **edit\_url**;  Button **btn\_add\_hotel**;   **public** String **qrcode**;   **private static** String *pic1*;  **private static** String *pic2*;  **private static** String *pic3*;   **private** ImageView **startCameraButton** = **null**;  **private** ImageView **choiceFromAlbumButton** = **null**;  **private** ImageView **choiceFromAlbumButton2** = **null**;  **private** ImageView **choiceFromAlbumButton3** = **null**;  **private** ImageView **pictureImageView** = **null**;  **private static** String *ID*;   **private static final int *TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE*** = 0; *// 拍照的权限处理返回码* **private static final int *WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE*** = 1; *// 读储存卡内容的权限处理返回码* **private static final int *TAKE\_PHOTO\_REQUEST\_CODE*** = 3; *// 拍照返回的 requestCode* **private static final int *CHOICE\_FROM\_ALBUM\_REQUEST\_CODE*** = 4; *// 相册选取返回的 requestCode* **private static final int *CROP\_PHOTO\_REQUEST\_CODE*** = 5; *// 裁剪图片返回的 requestCode* **private static final int *CROP\_PHOTO\_REQUEST\_CODE2*** = 6; *// 裁剪图片返回的 requestCode* **private** Uri **photoUri** = **null**;  **private** Uri **photoOutputUri** = **null**; *// 图片最终的输出文件的 Uri* @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***hotelview\_addhotel***);  **edit\_hotel\_name** = (EditText) findViewById(R.id.***edit\_hotel\_name***);  **spin\_address** = (Spinner) findViewById(R.id.***spin\_address***);  **spin\_area** = (Spinner) findViewById(R.id.***spin\_area***);  **edit\_hotel\_address** = (EditText) findViewById(R.id.***edit\_hotel\_address***);  **edit\_hotel\_phone**=(EditText) findViewById(R.id.***edit\_hotel\_phone***);  **edit\_hotel\_info**=(EditText) findViewById(R.id.***edit\_hotel\_info***);  **edit\_url**=(EditText) findViewById(R.id.***edit\_url***);  *// startCameraButton = (ImageView) findViewById(null); // startCameraButton.setOnClickListener(clickListener);* **choiceFromAlbumButton** = (ImageView) findViewById(R.id.***img\_license***);  **choiceFromAlbumButton**.setOnClickListener(**clickListener**);  **choiceFromAlbumButton2** = (ImageView) findViewById(R.id.***img\_pic***);  **choiceFromAlbumButton2**.setOnClickListener(**clickListener**);  **choiceFromAlbumButton3** = (ImageView) findViewById(R.id.***img\_line***);  **choiceFromAlbumButton3**.setOnClickListener(**clickListener**);  *//计算图片左右间距之和* **int** padding = 15;  **int** spacePx = (**int**) (UIUtil.*dp2px*(**this**, padding) \* 2);  *//计算图片宽度* **int** imageWidth = UIUtil.*getScreenWidth*(**this**) - spacePx;  **int** imageWidth2 = UIUtil.*getScreenWidth*(**this**) - spacePx;  **int** imageWidth3 = UIUtil.*getScreenWidth*(**this**) - spacePx;  *//计算宽高比，注意数字后面要加上f表示浮点型数字* **float** scale = 16f / 9f;  **float** scale2 = 16f / 9f;  **float** scale3 = 16f / 9f;  *//根据图片宽度和比例计算图片高度* **int** imageHeight = (**int**) (imageWidth / scale);  LinearLayout.LayoutParams params = **new** LinearLayout.LayoutParams( imageWidth,imageHeight);   **int** imageHeight2 = (**int**) (imageWidth2 / scale2);  LinearLayout.LayoutParams params2 = **new** LinearLayout.LayoutParams( imageWidth2,imageHeight2);  **int** imageHeight3 = (**int**) (imageWidth3 / scale3);  LinearLayout.LayoutParams params3 = **new** LinearLayout.LayoutParams( imageWidth3,imageHeight3);  *//设置左右边距* params.**leftMargin** = (**int**) UIUtil.*dp2px*(**this**, padding);  params.**rightMargin** = (**int**) UIUtil.*dp2px*(**this**, padding);  params2.**leftMargin** = (**int**) UIUtil.*dp2px*(**this**, padding);  params2.**rightMargin** = (**int**) UIUtil.*dp2px*(**this**, padding);  params3.**leftMargin** = (**int**) UIUtil.*dp2px*(**this**, padding);  params3.**rightMargin** = (**int**) UIUtil.*dp2px*(**this**, padding);  **choiceFromAlbumButton**.setLayoutParams(params2);  **choiceFromAlbumButton2**.setLayoutParams(params);  **choiceFromAlbumButton3**.setLayoutParams(params3);   Button button01 = (Button) findViewById(R.id.***btn\_add\_hotel***);   *//--------------------------  //取得伺服器上JSON資料  //--------------------------* button01.setOnClickListener(**new** Button.OnClickListener() {  @Override  **public void** onClick(View v) {  *ID*=**edit\_hotel\_name**.getText().toString();  Log.*v*(**"8888888"**, **spin\_address**.getSelectedItem().toString());  Log.*v*(**"8888888"**, **spin\_area**.getSelectedItem().toString());  Log.*v*(**"8888888"**, *pic1*);  Log.*v*(**"8888888"**, *pic2*);  String user = getSharedPreferences(**"userpw"**, ***MODE\_PRIVATE***).getString(**"USER"**, **""**);  Log.*v*(**"8888888"**, user);  getWorksheet.*postToHotel*(**edit\_hotel\_name**.getText().toString(), **spin\_address**.getSelectedItem().toString(), **spin\_area**.getSelectedItem().toString() , **edit\_hotel\_address**.getText().toString() , **edit\_hotel\_phone**.getText().toString(), **edit\_hotel\_info**.getText().toString(),*pic1*,*pic2*,user,**qrcode**,**edit\_url**.getText().toString());  Log.*d*(**"000000"**, String.*valueOf*(getWorksheet.*hostelinfoLength*) + **"post"**);  getWorksheet.*gethostelinfoJSON*();  getWorksheet.*getcheckJSON*();  **try** {  Thread.*sleep*(3000); *//1000為1秒* } **catch** (InterruptedException e) { *//* ***TODO Auto-generated catch block*** e.printStackTrace();  }  getWorksheet.*getJSON*();  getWorksheet.*getjobJSON*();  Log.*d*(**"000000"**, String.*valueOf*(getWorksheet.*hostelinfoLength*) + **"get"**);  getWorksheet.*getcalendarJSON*();  Setting\_Hotelinfo.*start\_sethostelname* = **false**;  getWorksheet.*gethostelinfoJSON*();  Intent intent = **new** Intent(Hotelview\_Addhotel.**this**,Hostelinfo\_Jump.**class**);  intent.putExtra(**"id"**,2);  finish();  }  });   **if**(ContextCompat.*checkSelfPermission*(Hotelview\_Addhotel.**this**, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***)  != PackageManager.***PERMISSION\_GRANTED***) {  *// 申请读写内存卡内容的权限* ActivityCompat.*requestPermissions*(Hotelview\_Addhotel.**this**,  **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***}, ***WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE***);  }  **final** Spinner spin\_address = (Spinner)findViewById(R.id.***spin\_address***);  **final** Spinner spin\_area = (Spinner)findViewById(R.id.***spin\_area***);  **final** String[] Keelung = {**"仁愛區"**,**"中正區"**,**"信義區"**,**"中山區"**,**"安樂區"**,**"暖暖區"**,**"七堵區"**};  **final** String[] Taipei = {**"中正區"**,**"大同區"**,**"中山區"**,**"松山區"**,**"大安區"**,**"萬華區"**,**"信義區"**,**"士林區"**,**"北投區"**,**"內湖區"**,**"南港區"**,**"文山區"**};  **final** String[] NewTaipei = {**"板橋區"**,**"新莊區"**,**"中和區"**,**"永和區"**,**"土城區"**,**"樹林區"**,**"三峽區"**,**"鶯歌區"**,**"三重區"**,**"蘆洲區"**,**"五股區"**,**"泰山區"**,**"林口區"**,**"八里區"**,**"淡水區"**,**"三芝區"**,**"石門區"**,**"金山區"**,**"萬里區"**,**"汐止區"**,**"瑞芳區"**,**"貢寮區"**,**"平溪區"**,**"雙溪區"**,**"新店區"**,**"深坑區"**,**"石碇區"**,**"坪林"**,**"烏來區"**};  **final** String[] Taoyuan = {**"桃園區"**,**"中壢區"**,**"平鎮區"**,**"八德區"**,**"楊梅區"**,**"蘆竹區"**,**"大溪區"**,**"龍潭區"**,**"龜山區"**,**"大園區"**,**"觀音區"**,**"新屋區"**,**"復興區"**};  **final** String[] Hsinchu = {**"東區"**,**"北區"**,**"香山區"**};  **final** String[] Hsinchu2 = {**"竹北市"**,**"竹東鎮"**,**"新埔鎮"**,**"關西鎮"**,**"湖口鄉"**,**"新豐鄉"**,**"峨眉鄉"**,**"寶山鄉"**,**"北埔鄉"**,**"芎林鄉"**,**"橫山鄉"**,**"尖石鄉"**,**"五峰鄉"**};  **final** String[] Miaoli = {**"苗栗市"**,**"頭份市"**,**"竹南鎮"**,**"後龍鎮"**,**"通霄鎮"**,**"苑裡鎮"**,**"卓蘭鎮"**,**"造橋鄉"**,**"西湖鄉"**,**"頭屋鄉"**,**"公館鄉"**,**"銅鑼鄉"**,**"三義鄉"**,**"大湖鄉"**,**"獅潭鄉"**,**"三灣鄉"**,**"南庄鄉"**,**"泰安鄉"**};  **final** String[] Taichung = {**"中區"**,**"東區"**,**"南區"**,**"西區"**,**"北區"**,**"北屯區"**,**"西屯區"**,**"南屯區"**,**"太平區"**,**"大里區"**,**"霧峰區"**,**"烏日區"**,**"豐原區"**,**"后里區"**,**"石岡區"**,**"東勢區"**,**"新社區"**,**"潭子區"**,**"大雅區"**,**"神岡區"**,**"大肚區"**,**"沙鹿區"**,**"龍井區"**,**"梧棲區"**,**"清水區"**,**"大甲區"**,**"外埔區"**,**"大安區"**,**"和平區"**};  **final** String[] Nantou = {**"南投市"**,**"埔里鎮"**,**"草屯鎮"**,**"竹山鎮"**,**"集集鎮"**,**"名間鄉"**,**"鹿谷鄉"**,**"中寮鄉"**,**"魚池鄉"**,**"國姓鄉"**,**"水里鄉"**,**"信義鄉"**,**"仁愛鄉"**};  **final** String[] Changhua = {**"彰化市"**,**"員林市"**,**"和美鎮"**,**"鹿港鎮"**,**"溪湖鎮"**,**"二林鎮"**,**"田中鎮"**,**"北斗鎮"**,**"花壇鄉"**,**"芬園鄉"**,**"大村鄉"**,**"永靖鄉"**,**"伸港鄉"**,**"線西鄉"**,**"福興鄉"**,**"秀水鄉"**,**"埔心鄉"**,**"埔鹽鄉"**,**"大城鄉"**,**"芳苑鄉"**,**"竹塘鄉"**,**"社頭鄉"**,**"二水鄉"**,**"田尾鄉"**,**"埤頭鄉"**,**"溪州鄉"**};  **final** String[] Yunlin = {**"斗六市"**,**"斗南鎮"**,**"虎尾鎮"**,**"西螺鎮"**,**"土庫鎮"**,**"北港鎮"**,**"林內鄉"**,**"古坑鄉"**,**"大埤鄉"**,**"莿桐鄉"**,**"褒忠鄉"**,**"二崙鄉"**,**"崙背鄉"**,**"麥寮鄉"**,**"臺西鄉"**,**"東勢鄉"**,**"元長鄉"**,**"四湖鄉"**,**"口湖鄉"**,**"水林鄉"**};  **final** String[] Chiayi = {**"東區"**,**"西區"**};  **final** String[] Chiayi2 = {**"太保市"**,**"朴子市"**,**"布袋鎮"**,**"大林鎮"**,**"民雄鄉"**,**"溪口鄉"**,**"新港鄉"**,**"六腳鄉"**,**"東石鄉"**,**"義竹鄉"**,**"鹿草鄉"**,**"水上鄉"**,**"中埔鄉"**,**"竹崎鄉"**,**"梅山鄉"**,**"番路鄉"**,**"大埔鄉"**,**"阿里山鄉"**};  **final** String[] Tainan = {**"中西區"**,**"東區"**,**"南區"**,**"北區"**,**"安平區"**,**"安南區"**,**"永康區"**,**"歸仁區"**,**"新化區"**,**"左鎮區"**,**"玉井區"**,**"楠西區"**,**"南化區"**,**"仁德區"**,**"關廟區"**,**"龍崎區"**,**"官田區"**,**"麻豆區"**,**"佳里區"**,**"西港區"**,**"七股區"**,**"將軍區"**,**"學甲區"**,**"北門區"**,**"新營區"**,**"後壁區"**,**"白河區"**,**"東山區"**,**"六甲區"**,**"下營區"**,**"柳營區"**,**"鹽水區"**,**"善化區"**,**"大內區"**,**"山上區"**,**"新市區"**,**"安定區"**};  **final** String[] Kaohsiung = {**"楠梓區"**,**"左營區"**,**"鼓山區"**,**"三民區"**,**"鹽埕區"**,**"前金區"**,**"新興區"**,**"苓雅區"**,**"前鎮區"**,**"旗津區"**,**"小港區"**,**"鳳山區"**,**"大寮區"**,**"鳥松區"**,**"林園區"**,**"仁武區"**,**"大樹區"**,**"大社區"**,**"岡山區"**,**"路竹區"**,**"橋頭區"**,**"梓官區"**,**"彌陀區"**,**"永安區"**,**"燕巢區"**,**"田寮區"**,**"阿蓮區"**,**"茄萣區"**,**"湖內區"**,**"旗山區"**,**"美濃區"**,**"內門區"**,**"杉林區"**,**"甲仙區"**,**"六龜區"**,**"茂林區"**,**"桃源區"**,**"那瑪夏區"**};  **final** String[] Pingtung = {**"屏東市"**,**"潮州鎮"**,**"東港鎮"**,**"恆春鎮"**,**"萬丹鄉"**,**"長治鄉"**,**"麟洛鄉"**,**"九如鄉"**,**"里港鄉"**,**"鹽埔鄉"**,**"高樹鄉"**,**"萬巒鄉"**,**"內埔鄉"**,**"竹田鄉"**,**"新埤鄉"**,**"枋寮鄉"**,**"新園鄉"**,**"崁頂鄉"**,**"林邊鄉"**,**"南州鄉"**,**"佳冬鄉"**,**"琉球鄉"**,**"車城鄉"**,**"滿州鄉"**,**"枋山鄉"**,**"霧臺鄉"**,**"瑪家鄉"**,**"泰武鄉"**,**"來義鄉"**,**"春日鄉"**,**"獅子鄉"**,**"牡丹鄉"**,**"三地門鄉"**};  **final** String[] Yilan = {**"宜蘭市"**,**"頭城鎮"**,**"羅東鎮"**,**"蘇澳鎮"**,**"礁溪鄉"**,**"壯圍鄉"**,**"員山鄉"**,**"冬山鄉"**,**"五結鄉"**,**"三星鄉"**,**"大同鄉"**,**"南澳鄉"**};  **final** String[] Hualien = {**"花蓮市"**,**"鳳林鎮"**,**"玉里鎮"**,**"新城鄉"**,**"吉安鄉"**,**"壽豐鄉"**,**"光復鄉"**,**"豐濱鄉"**,**"瑞穗鄉"**,**"富里鄉"**,**"秀林鄉"**,**"萬榮鄉"**,**"卓溪鄉"**};  **final** String[] Taitung = {**"臺東市"**,**"成功鎮"**,**"關山鎮"**,**"長濱鄉"**,**"池上鄉"**,**"東河鄉"**,**"鹿野鄉"**,**"卑南鄉"**,**"大武鄉"**,**"綠島鄉"**,**"太麻里鄉"**,**"海端鄉"**,**"延平鄉"**,**"金峰鄉"**,**"達仁鄉"**,**"蘭嶼鄉"**};  **final** String[] Penghu = {**"馬公市"**,**"湖西鄉"**,**"白沙鄉"**,**"西嶼鄉"**,**"望安鄉"**,**"七美鄉"**};  **final** String[] Kinmen = {**"金城鎮"**,**"金湖鎮"**,**"金沙鎮"**,**"金寧鄉"**,**"烈嶼鄉"**,**"烏坵鄉"**};  **final** String[] Lienchiang = {**"南竿鄉"**,**"北竿鄉"**,**"莒光鄉"**,**"東引鄉"**};   spin\_address.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  @Override  **public void** onItemSelected(AdapterView<?> parent, View view, **int** position, **long** id) {  Toast.*makeText*(Hotelview\_Addhotel.**this**, **"你選的是"** + spin\_address.getSelectedItem().toString(), Toast.***LENGTH\_SHORT***).show();  **switch** (spin\_address.getSelectedItem().toString()) {  **case "基隆市"**:  ArrayAdapter<String> KeelungList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Keelung);  spin\_area.setAdapter(KeelungList);  **break**;  **case "台北市"**:  ArrayAdapter<String> TaipeiList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Taipei);  spin\_area.setAdapter(TaipeiList);  **break**;  **case "新北市"**:  ArrayAdapter<String> NewTaipeiList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  NewTaipei);  spin\_area.setAdapter(NewTaipeiList);  **break**;  **case "桃園市"**:  ArrayAdapter<String> TaoyuanList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Taoyuan);  spin\_area.setAdapter(TaoyuanList);  **break**;  **case "新竹市"**:  ArrayAdapter<String> HsinchuList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Hsinchu);  spin\_area.setAdapter(HsinchuList);  **break**;  **case "新竹縣"**:  ArrayAdapter<String> Hsinchu2List = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Hsinchu2);  spin\_area.setAdapter(Hsinchu2List);  **break**;  **case "苗栗縣"**:  ArrayAdapter<String> MiaoliList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Miaoli);  spin\_area.setAdapter(MiaoliList);  **break**;  **case "台中市"**:  ArrayAdapter<String> TaichungList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Taichung);  spin\_area.setAdapter(TaichungList);  **break**;  **case "南投縣"**:  ArrayAdapter<String> NantouList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Nantou);  spin\_area.setAdapter(NantouList);  **break**;  **case "彰化縣"**:  ArrayAdapter<String> ChanghuaList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Changhua);  spin\_area.setAdapter(ChanghuaList);  **break**;  **case "雲林縣"**:  ArrayAdapter<String> YunlinList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Yunlin);  spin\_area.setAdapter(YunlinList);  **break**;  **case "嘉義市"**:  ArrayAdapter<String> ChiayiList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Chiayi);  spin\_area.setAdapter(ChiayiList);  **break**;  **case "嘉義縣"**:  ArrayAdapter<String> Chiayi2List = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Chiayi2);  spin\_area.setAdapter(Chiayi2List);  **break**;  **case "台南市"**:  ArrayAdapter<String> TainanList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Tainan);  spin\_area.setAdapter(TainanList);  **break**;  **case "高雄市"**:  ArrayAdapter<String> KaohsiungList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Kaohsiung);  spin\_area.setAdapter(KaohsiungList);  **break**;  **case "屏東縣"**:  ArrayAdapter<String> PingtungList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Pingtung);  spin\_area.setAdapter(PingtungList);  **break**;  **case "宜蘭縣"**:  ArrayAdapter<String> YilanList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Yilan);  spin\_area.setAdapter(YilanList);  **break**;  **case "花蓮縣"**:  ArrayAdapter<String> HualienList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Hualien);  spin\_area.setAdapter(HualienList);  **break**;  **case "台東縣"**:  ArrayAdapter<String> TaitungList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Taitung);  spin\_area.setAdapter(TaitungList);  **break**;  **case "澎湖縣"**:  ArrayAdapter<String> PenghuList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Penghu);  spin\_area.setAdapter(PenghuList);  **break**;  **case "金門縣"**:  ArrayAdapter<String> KinmenList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Kinmen);  spin\_area.setAdapter(KinmenList);  **break**;  **case "連江縣"**:  ArrayAdapter<String> LienchiangList = **new** ArrayAdapter<>(Hotelview\_Addhotel.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  Lienchiang);  spin\_area.setAdapter(LienchiangList);  **break**;  }  }   @Override  **public void** onNothingSelected(AdapterView<?> parent) {   }  });  }  **private** View.OnClickListener **clickListener** = **new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *// 调用相机拍照* **if**(v == **startCameraButton**) {  *// 同上面的权限申请逻辑* **if**(ContextCompat.*checkSelfPermission*(Hotelview\_Addhotel.**this**, Manifest.permission.***CAMERA***)  != PackageManager.***PERMISSION\_GRANTED***) {  */\*  \* 下面是对调用相机拍照权限进行申请  \*/* ActivityCompat.*requestPermissions*(Hotelview\_Addhotel.**this**,  **new** String[]{Manifest.permission.***CAMERA***,}, ***TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE***);  } **else** { *// startCamera();* }  *// 从相册获取* } **else if**(v == **choiceFromAlbumButton**) {  choiceFromAlbum();  }  **else if**(v == **choiceFromAlbumButton2**) {  choiceFromAlbum2();  }  **else if**(v == **choiceFromAlbumButton3**) {  choiceFromAlbum3();  }  }  };   */\*\*  \* 拍照  \*/ // private void startCamera() { // /\*\* // \* 设置拍照得到的照片的储存目录，因为我们访问应用的缓存路径并不需要读写内存卡的申请权限， // \* 因此，这里为了方便，将拍照得到的照片存在这个缓存目录中 // \*/ // File file = new File(getExternalCacheDir(), "image.jpg"); // try { // if(file.exists()) { // file.delete(); // } // file.createNewFile(); // } catch (IOException e) { // e.printStackTrace(); // } // /\*\* // \* 因 Android 7.0 开始，不能使用 file:// 类型的 Uri 访问跨应用文件，否则报异常， // \* 因此我们这里需要使用内容提供器，FileProvider 是 ContentProvider 的一个子类， // \* 我们可以轻松的使用 FileProvider 来在不同程序之间分享数据(相对于 ContentProvider 来说) // \*/ // if(Build.VERSION.SDK\_INT >= 24) { // photoUri = FileProvider.getUriForFile(this, "com.zhi\_dian.provider", file); // } else { // photoUri = Uri.fromFile(file); // Android 7.0 以前使用原来的方法来获取文件的 Uri // } // // 打开系统相机的 Action，等同于："android.media.action.IMAGE\_CAPTURE" // Intent takePhotoIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE); // // 设置拍照所得照片的输出目录 // takePhotoIntent.putExtra(MediaStore.EXTRA\_OUTPUT, photoUri); // startActivityForResult(takePhotoIntent, TAKE\_PHOTO\_REQUEST\_CODE); // }   /\*\*  \* 从相册选取  \*/* **private void** choiceFromAlbum() {  *// 打开系统图库的 Action，等同于: "android.intent.action.GET\_CONTENT"* Intent choiceFromAlbumIntent = **new** Intent(Intent.***ACTION\_GET\_CONTENT***);  *// 设置数据类型为图片类型* choiceFromAlbumIntent.setType(**"image/\*"**);  startActivityForResult(choiceFromAlbumIntent, ***CHOICE\_FROM\_ALBUM\_REQUEST\_CODE***);  }  **private void** choiceFromAlbum2() {  *// 打开系统图库的 Action，等同于: "android.intent.action.GET\_CONTENT"* Intent choiceFromAlbumIntent = **new** Intent(Intent.***ACTION\_GET\_CONTENT***);  *// 设置数据类型为图片类型* choiceFromAlbumIntent.setType(**"image/\*"**);  startActivityForResult(choiceFromAlbumIntent, 6);  }  **private void** choiceFromAlbum3() {  *// 打开系统图库的 Action，等同于: "android.intent.action.GET\_CONTENT"* Intent choiceFromAlbumIntent = **new** Intent(Intent.***ACTION\_GET\_CONTENT***);  *// 设置数据类型为图片类型* choiceFromAlbumIntent.setType(**"image/\*"**);  startActivityForResult(choiceFromAlbumIntent, 8);  }    */\*\*  \* 裁剪图片  \*/* **private void** cropPhoto(Uri inputUri) {  *// 调用系统裁剪图片的 Action* Intent cropPhotoIntent = **new** Intent(**"com.android.camera.action.CROP"**);  *// 设置数据Uri 和类型* cropPhotoIntent.setDataAndType(inputUri, **"image/\*"**);  *// 授权应用读取 Uri，这一步要有，不然裁剪程序会崩溃* cropPhotoIntent.addFlags(Intent.***FLAG\_GRANT\_READ\_URI\_PERMISSION***);  *// 设置图片的最终输出目录* cropPhotoIntent.putExtra(MediaStore.***EXTRA\_OUTPUT***,  **photoOutputUri** = Uri.*parse*(**"file:////sdcard/image\_output.jpg"**));  startActivityForResult(cropPhotoIntent, ***CROP\_PHOTO\_REQUEST\_CODE***);  }  **private void** cropPhoto2(Uri inputUri) {  *// 调用系统裁剪图片的 Action* Intent cropPhotoIntent = **new** Intent(**"com.android.camera.action.CROP"**);  *// 设置数据Uri 和类型* cropPhotoIntent.setDataAndType(inputUri, **"image/\*"**);  *// 授权应用读取 Uri，这一步要有，不然裁剪程序会崩溃* cropPhotoIntent.addFlags(Intent.***FLAG\_GRANT\_READ\_URI\_PERMISSION***);  *// 设置图片的最终输出目录* cropPhotoIntent.putExtra(MediaStore.***EXTRA\_OUTPUT***,  **photoOutputUri** = Uri.*parse*(**"file:////sdcard/image\_output.jpg"**));  startActivityForResult(cropPhotoIntent, 7);  }  **private void** cropPhoto3(Uri inputUri) {  *// 调用系统裁剪图片的 Action* Intent cropPhotoIntent = **new** Intent(**"com.android.camera.action.CROP"**);  *// 设置数据Uri 和类型* cropPhotoIntent.setDataAndType(inputUri, **"image/\*"**);  *// 授权应用读取 Uri，这一步要有，不然裁剪程序会崩溃* cropPhotoIntent.addFlags(Intent.***FLAG\_GRANT\_READ\_URI\_PERMISSION***);  *// 设置图片的最终输出目录* cropPhotoIntent.putExtra(MediaStore.***EXTRA\_OUTPUT***,  **photoOutputUri** = Uri.*parse*(**"file:////sdcard/image\_output.jpg"**));  startActivityForResult(cropPhotoIntent, 9);  }   */\*\*  \* 在这里进行用户权限授予结果处理  \** ***@param requestCode*** *权限要求码，即我们申请权限时传入的常量  \** ***@param permissions*** *保存权限名称的 String 数组，可以同时申请一个以上的权限  \** ***@param grantResults*** *每一个申请的权限的用户处理结果数组(是否授权)  \*/* @Override  **public void** onRequestPermissionsResult(**int** requestCode, @NonNull String[] permissions, @NonNull **int**[] grantResults) {  **switch** (requestCode) {  *// 调用相机拍照：* **case *TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE***:  *// 如果用户授予权限，那么打开相机拍照* **if**(grantResults.**length** > 0 && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) { *// startCamera();* } **else** {  Toast.*makeText*(**this**, **"拍照权限被拒绝"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  *// 打开相册选取：* **case *WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE***:  **if**(grantResults.**length** > 0 && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {  } **else** {  Toast.*makeText*(**this**, **"读写内存卡内容权限被拒绝"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  }  }   */\*\*  \* 通过这个 activity 启动的其他 Activity 返回的结果在这个方法进行处理  \* 我们在这里对拍照、相册选择图片、裁剪图片的返回结果进行处理  \** ***@param requestCode*** *返回码，用于确定是哪个 Activity 返回的数据  \** ***@param resultCode*** *返回结果，一般如果操作成功返回的是 RESULT\_OK  \** ***@param data*** *返回对应 activity 返回的数据  \*/* @Override  **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  **if**(resultCode == ***RESULT\_OK***) {  *// 通过返回码判断是哪个应用返回的数据* **switch** (requestCode) {  *// 拍照* **case *TAKE\_PHOTO\_REQUEST\_CODE***:  cropPhoto(**photoUri**);  **break**;  *// 相册选择* **case *CHOICE\_FROM\_ALBUM\_REQUEST\_CODE***:  cropPhoto(data.getData());  **break**;  *// 裁剪图片* **case *CROP\_PHOTO\_REQUEST\_CODE***:  File file = **new** File(**photoOutputUri**.getPath());  **if**(file.exists()) {  Bitmap bitmap = BitmapFactory.*decodeFile*(**photoOutputUri**.getPath());  *BitmapToString*(bitmap);  **choiceFromAlbumButton**.setImageBitmap(bitmap); *// file.delete(); // 选取完后删除照片* } **else** {  Toast.*makeText*(**this**, **"找不到照片"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  **case** 6:  cropPhoto2(data.getData());  **break**;  *// 裁剪图片* **case** 7:  File file2 = **new** File(**photoOutputUri**.getPath());  **if**(file2.exists()) {  Bitmap bitmap = BitmapFactory.*decodeFile*(**photoOutputUri**.getPath());  *BitmapToString2*(bitmap);  **choiceFromAlbumButton2**.setImageBitmap(bitmap); *// file.delete(); // 选取完后删除照片* } **else** {  Toast.*makeText*(**this**, **"找不到照片"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  **case** 8:  cropPhoto3(data.getData());  **break**;  *// 裁剪图片* **case** 9:  File file3 = **new** File(**photoOutputUri**.getPath());  **if**(file3.exists()) {  Bitmap bitmap = BitmapFactory.*decodeFile*(**photoOutputUri**.getPath());  *BitmapToString3*(bitmap);  **choiceFromAlbumButton3**.setImageBitmap(bitmap);  BarcodeDetector detector = **new** BarcodeDetector.Builder(getApplicationContext())  .setBarcodeFormats(Barcode.***QR\_CODE***)  .build();  Frame frame = **new** Frame.Builder()  .setBitmap(bitmap).build();  SparseArray<Barcode> barsCode = detector.detect(frame);   Barcode result = barsCode.valueAt(0);  **qrcode** = result.**rawValue**;  Log.*v*(**"dick"**,result.**rawValue**); *// file.delete(); // 选取完后删除照片* } **else** {  Toast.*makeText*(**this**, **"找不到照片"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;}}}  @Override  **protected void** onDestroy() {  getWorksheet.*gethostelinfoJSON*();  **super**.onDestroy();  getWorksheet.*gethostelinfoJSON*();  }  **public static** String BitmapToString(Bitmap bitmap) {  String des = **null**;  **try** {  ByteArrayOutputStream out = **new** ByteArrayOutputStream();  bitmap.compress(Bitmap.CompressFormat.***JPEG***, 100, out);  out.flush();  out.close();  **byte**[] buffer = out.toByteArray();  **byte**[] encode = Base64.*encode*(buffer, Base64.***DEFAULT***);  des = **new** String(encode);  } **catch** (Exception e) {  e.printStackTrace();  }  *pic1*=des;  **return** des;  }  **public static** String BitmapToString2(Bitmap bitmap) {  String des = **null**;  **try** {  ByteArrayOutputStream out = **new** ByteArrayOutputStream();  bitmap.compress(Bitmap.CompressFormat.***JPEG***, 100, out);  out.flush();  out.close();  **byte**[] buffer = out.toByteArray();  **byte**[] encode = Base64.*encode*(buffer, Base64.***DEFAULT***);  des = **new** String(encode);  } **catch** (Exception e) {  e.printStackTrace();  }  *pic2*=des;  **return** des;  }  **public static** String BitmapToString3(Bitmap bitmap) {  String des = **null**;  **try** {  ByteArrayOutputStream out = **new** ByteArrayOutputStream();  bitmap.compress(Bitmap.CompressFormat.***JPEG***, 100, out);  out.flush();  out.close();  **byte**[] buffer = out.toByteArray();  **byte**[] encode = Base64.*encode*(buffer, Base64.***DEFAULT***);  des = **new** String(encode);  } **catch** (Exception e) {  e.printStackTrace();  }  *pic3*=des;  **return** des;  }  **public static** String getID(){  **return** *ID*;  }} | |

|  |  |
| --- | --- |
| 程式名稱 | Humansearch\_Info |
| 目的 | 顯示詳細人才資訊 |
| 程式碼 | |
| **public class** Humansearch\_Info **extends** AppCompatActivity {   **public** TextView **name**, **gender**,**birth**,**home**,**school**,**department**,**study\_state**,**interest**,**work\_exp**,**exchange\_reason**,**eatting\_habit**,**start\_date**,**end\_date**,**phone**,**email**;  **public** ImageView **img\_photo**,**life\_photo**;  **public** Button **btn\_contact**,**btn\_hire**;  **public** String **user**, **str\_hostelNo**,**stud\_account**;  **public** ArrayList<String> **ary\_hostel**, **ary\_hostelNo**;  **public** Spinner **chooseHostel**;  **static final int *REQUEST\_ACTION\_PICK*** = 1;  **public static final** String ***PACKAGE\_NAME*** = **"jp.naver.line.android"**;  **public static final** String ***CLASS\_NAME*** = **"jp.naver.line.android.activity.selectchat.SelectChatActivity"**;  **private** List<ApplicationInfo> **m\_appList**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {   **super**.onCreate(savedInstanceState);  setContentView(R.layout.***humansearch\_info***);  **chooseHostel** = findViewById(R.id.***choose\_hostel***);   **ary\_hostel** = **new** ArrayList<>();  **ary\_hostelNo** = **new** ArrayList<>();  **for**(**int** i = 0 ; i < getWorksheet.*hostelinfoLength*; i++){  **ary\_hostel**.add(getWorksheet.*getRow33*(i));  **ary\_hostelNo**.add(getWorksheet.*getRow39*(i));  }  ArrayAdapter<String> adptHstChoose = **new** ArrayAdapter<>(**this**.getBaseContext(),  android.R.layout.***simple\_spinner\_dropdown\_item***,  **ary\_hostel**);  **chooseHostel**.setAdapter(adptHstChoose);   **chooseHostel**.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  @Override  **public void** onItemSelected(AdapterView<?> parent, View view, **int** position, **long** id) {  **int** indexofHos = **chooseHostel**.getSelectedItemPosition();  **str\_hostelNo** = **ary\_hostelNo**.get(indexofHos);  Log.*d*(**"Log"**, **"str\_hostelNo是沙小???"** + **str\_hostelNo**);  *// Toast.makeText(getActivity().getBaseContext(), "你選的是" + city + district, Toast.LENGTH\_SHORT).show();* }   @Override  **public void** onNothingSelected(AdapterView<?> parent) {   }  });  Bundle bundle=getIntent().getExtras();  **int** position=bundle.getInt(**"position"**);  *//* **btn\_contact** = (Button) findViewById(R.id.***btn\_contact***);  **btn\_contact**.setOnClickListener(**new** Button.OnClickListener() {  @Override  **public void** onClick(View v) { *// Intent intent = new Intent(Intent.ACTION\_SEND); // intent.setPackage(PACKAGE\_NAME); //// intent.setClassName(PACKAGE\_NAME, CLASS\_NAME); // Uri number = Uri.parse("line://nv/recommendOA/@s920613a");  // intent.setType("text/plain"); // intent.putExtra(Intent.EXTRA\_TEXT,"//ti/p/%40linedevelopers"); // startActivity(intent);* Log.*v*(**"話都你在說"**,getWorksheet.*getRow49*(position));  Uri uri = Uri.*parse*(getWorksheet.*getRow49*(position));  Intent intent = **new** Intent(Intent.***ACTION\_VIEW***, uri);  startActivity(intent);  }  });  *// btn\_hire = (Button) findViewById(R.id.btn\_contact); // btn\_hire.setOnClickListener(new Button.OnClickListener() { // @Override // public void onClick(View v) { // Log.d("Log", "我要傳???" + user); // Log.d("Log", "我要傳???" + str\_hostelNo); // Log.d("Log", "我要傳???" + stud\_account); // getWorksheet.postToResume(user,str\_hostelNo,stud\_account); // } // });* **btn\_hire** = (Button) findViewById(R.id.***btn\_hire***);  **btn\_hire**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Log.*d*(**"Log"**, **"我要傳!!!"** + **user**);  Log.*d*(**"Log"**, **"我要傳!!!"** + **str\_hostelNo**);  Log.*d*(**"Log"**, **"我要傳!!!"** + **stud\_account**);  getWorksheet.*postToResume*(**user**,**str\_hostelNo**,**stud\_account**);  }  });    **name** = (TextView) findViewById(R.id.***name***);  **gender** = (TextView) findViewById(R.id.***gender***);  **birth** = (TextView) findViewById(R.id.***birth***);  **home** = (TextView) findViewById(R.id.***home***);  **school** = (TextView) findViewById(R.id.***school***);  **department** = (TextView) findViewById(R.id.***department***);  **study\_state** = (TextView) findViewById(R.id.***study\_state***);  **interest** = (TextView) findViewById(R.id.***interest***);  **work\_exp** = (TextView) findViewById(R.id.***work\_exp***);  **exchange\_reason** = (TextView) findViewById(R.id.***exchange\_reason***);  **eatting\_habit** = (TextView) findViewById(R.id.***eatting\_habit***);  **start\_date** = (TextView) findViewById(R.id.***start\_date***);  **end\_date** = (TextView) findViewById(R.id.***end\_date***);  **interest** = (TextView) findViewById(R.id.***interest***);  **phone** = (TextView) findViewById(R.id.***phone***);  **email** = (TextView) findViewById(R.id.***email***);   **img\_photo** = (ImageView) findViewById(R.id.***img\_photo***);  **life\_photo** = (ImageView) findViewById(R.id.***life\_photo***);    **name**.setText(getWorksheet.*getRow13*(position));  **school**.setText(getWorksheet.*getRow14*(position));  **work\_exp**.setText(getWorksheet.*getRow15*(position));  **exchange\_reason**.setText(getWorksheet.*getRow16*(position));  **img\_photo**.setImageBitmap(stringToBitmap(getWorksheet.*getRow17*(position)));  **gender**.setText(getWorksheet.*getRow18*(position));  **birth**.setText(getWorksheet.*getRow19*(position));  **phone**.setText(getWorksheet.*getRow20*(position));  **home**.setText(getWorksheet.*getRow21*(position));  **email**.setText(getWorksheet.*getRow22*(position));  **life\_photo**.setImageBitmap(stringToBitmap(getWorksheet.*getRow23*(position)));  **study\_state**.setText(getWorksheet.*getRow24*(position));  **interest**.setText(getWorksheet.*getRow25*(position));  **eatting\_habit**.setText(getWorksheet.*getRow26*(position));  **start\_date**.setText(getWorksheet.*getRow27*(position));  **end\_date**.setText(getWorksheet.*getRow28*(position));  **department**.setText(getWorksheet.*getRow30*(position));  **stud\_account** = getWorksheet.*getRow40*(position);  **user** = getSharedPreferences(**"userpw"**, ***MODE\_PRIVATE***).getString(**"USER"**, **""**);   *// Log.d("Log", "position是沙小???" + ary\_hostel); // Log.d("Log", "position是沙小???" + user); // Log.d("Log", "position是沙小???" + stud\_account);* }   **public** Bitmap stringToBitmap(String string) {  Bitmap bitmap = **null**;  **try** {  **byte**[] bitmapArray = Base64.*decode*(string, Base64.***DEFAULT***);  bitmap = BitmapFactory.*decodeByteArray*(bitmapArray, 0, bitmapArray.**length**);  **return** bitmap;  } **catch** (NullPointerException e) {  e.getMessage();  **return null**;  } **catch** (OutOfMemoryError e) {  **return null**;  }  }   **public void** sendTextHandler(View view) {  String sendText = **"s920613a"**;  **if**(checkLineInstalled()){  Intent intent = **new** Intent(Intent.***ACTION\_SEND***);  intent.setClassName(***PACKAGE\_NAME***, ***CLASS\_NAME***);  intent.setType(**"text/plain"**);  intent.putExtra(Intent.***EXTRA\_TEXT***, sendText);  startActivity(intent);  }**else**{  Toast toast = Toast.*makeText*(**this**, **"LINEがインストールされていません"**, Toast.***LENGTH\_SHORT***);  toast.show();  }  }  **private boolean** checkLineInstalled(){  PackageManager pm = getPackageManager();  **m\_appList** = pm.getInstalledApplications(0);  **boolean** lineInstallFlag = **false**;  **for** (ApplicationInfo ai : **m\_appList**) {  **if**(ai.**packageName**.equals(***PACKAGE\_NAME***)){  lineInstallFlag = **true**;  **break**;  }  }  **return** lineInstallFlag;  } } | |

|  |  |
| --- | --- |
| 程式名稱 | Jobview\_Addjob |
| 目的 | 新增刊登職缺 |
| 程式碼 | |
| **public class** Jobview\_Addjob **extends** AppCompatActivity {   **private** EditText **start\_date**, **start\_time**, **end\_date**, **end\_time**;  **private int mYear**, **mMonth**, **mDay**, **mHour**, **mMinute**;   EditText **edit\_job\_title**;  EditText **edit\_salary**;  CheckBox **chk\_stay**;  String **chk\_stayS**;  CheckBox **chk\_food**;  String **chk\_foodS**;  CheckBox **chk\_fish**;  String **chk\_fishS**;  CheckBox **chk\_snorkel**;  String **chk\_snorkelS**;  CheckBox **chk\_moto**;  String **chk\_motoS**;  EditText **edit\_number\_people**;  EditText **edit\_work**;  CheckBox **chk\_salary**;  EditText **edit\_contact**;  EditText **edit\_email**;  EditText **edit\_phone**;  Spinner **spinner**;    **private** String **valid\_edit\_salary**;  **private** String **valid\_email**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***jobview\_addjob***);  **start\_date** = (EditText) findViewById(R.id.***start\_date***);  **start\_time** = (EditText) findViewById(R.id.***start\_time***);  **end\_date** = (EditText) findViewById(R.id.***end\_date***);  **end\_time** = (EditText) findViewById(R.id.***end\_time***);  **chk\_salary**=(CheckBox) findViewById(R.id.***chk\_salary***);  **final** Spinner choose\_hostel = (Spinner)findViewById(R.id.***spin\_hotel***);  **final** ArrayList<String> myList2 = **new** ArrayList<String>();  **for**(**int** i = 0; i < getWorksheet.*hostelnameLength*; i++) {  myList2.add(getWorksheet.*getRow38*(i)); *// Log.v("88989898",getWorksheet.getRow36(i)); // Log.v("88989898",myList2.get(i));* }  ArrayAdapter<String> myList = **new** ArrayAdapter<String>(Jobview\_Addjob.**this**,  android.R.layout.***simple\_spinner\_dropdown\_item***,  myList2);   choose\_hostel.setAdapter(myList);    **chk\_salary**.setOnClickListener(**new** Button.OnClickListener(){  @Override  **public void** onClick(View v) {  **if**(**chk\_salary**.isChecked()){  **edit\_salary**.setFocusable(**false**);  **edit\_salary**.setText(**"0"**);  }**else** {  **edit\_salary**.setFocusable(**true**);  **edit\_salary**.setFocusableInTouchMode(**true**);  **edit\_salary**.requestFocus();  **edit\_salary**.requestFocusFromTouch();  **edit\_salary**.setText(**""**);  }  **return**;  }  });   **start\_date**.setOnClickListener(**new** Button.OnClickListener() {  **public void** onClick(View v) {  showDatePickerDialogStart();  }  });  **start\_time**.setOnClickListener(**new** Button.OnClickListener() {  **public void** onClick(View v) {  showTimePickerDialogStart();  }  });  **end\_date**.setOnClickListener(**new** Button.OnClickListener() {  **public void** onClick(View v) {  showDatePickerDialogEnd();  }  });  **end\_time**.setOnClickListener(**new** Button.OnClickListener() {  **public void** onClick(View v) {  showTimePickerDialogEnd();  }  });    **edit\_job\_title** = (EditText) findViewById(R.id.***edit\_job\_title***);  **edit\_salary** = (EditText) findViewById(R.id.***edit\_salary***);  **chk\_stay** = (CheckBox) findViewById(R.id.***chk\_stay***);  **chk\_food** = (CheckBox) findViewById(R.id.***chk\_food***);  **chk\_fish** = (CheckBox) findViewById(R.id.***chk\_fish***);  **chk\_snorkel** = (CheckBox) findViewById(R.id.***chk\_snorkel***);  **chk\_moto** = (CheckBox) findViewById(R.id.***chk\_moto***);  **edit\_number\_people** = (EditText) findViewById(R.id.***edit\_number\_people***);  **edit\_work** = (EditText) findViewById(R.id.***edit\_work***);  **edit\_contact**= (EditText) findViewById(R.id.***edit\_contact***);  **edit\_email**= (EditText) findViewById(R.id.***edit\_email***);  **edit\_phone**= (EditText) findViewById(R.id.***edit\_phone***);   Button button01 = (Button) findViewById(R.id.***btn\_add\_job***);   **edit\_salary**.addTextChangedListener(**textWatcher**);  **edit\_number\_people**.addTextChangedListener(**textWatcher1**);  **edit\_contact**.addTextChangedListener(**textWatcher2**);  **edit\_email**.addTextChangedListener(**textWatcher3**);  **edit\_phone**.addTextChangedListener(**textWatcher4**);  **end\_date**.addTextChangedListener(**textWatcher5**);  **start\_date**.addTextChangedListener(**textWatcher6**);  **end\_time**.addTextChangedListener(**textWatcher7**);  **start\_time**.addTextChangedListener(**textWatcher8**);      *//--------------------------  //取得伺服器上JSON資料  //--------------------------* button01.setOnClickListener(**new** Button.OnClickListener() {  @Override  **public void** onClick(View v) {  **if** (**chk\_stay**.isChecked()){  **chk\_stayS** = **"1"**;  }**else** {  **chk\_stayS** = **"0"**;  }  **if** (**chk\_food**.isChecked()){  **chk\_foodS** = **"1"**;  }**else** {  **chk\_foodS** = **"0"**;  }  **if** (**chk\_fish**.isChecked()){  **chk\_fishS** = **"1"**;  }**else** {  **chk\_fishS** = **"0"**;  }  **if** (**chk\_snorkel**.isChecked()){  **chk\_snorkelS** = **"1"**;  }**else** {  **chk\_snorkelS** = **"0"**;  }  **if** (**chk\_moto**.isChecked()){  **chk\_motoS** = **"1"**;  }**else** {  **chk\_motoS** = **"0"**;  }  Log.*v*(**"chk\_stay"**, **start\_date**.getText().toString());  getWorksheet.*postToJob*(**edit\_job\_title**.getText().toString(), **edit\_salary**.getText().toString(), **chk\_stayS**, **chk\_foodS**, **chk\_fishS**, **chk\_snorkelS**, **chk\_motoS**, **start\_date**.getText().toString(), **start\_time**.getText().toString(), **end\_date**.getText().toString(), **end\_time**.getText().toString(), **edit\_number\_people**.getText().toString(), **edit\_work**.getText().toString());  Log.*d*(**"get0000"**, String.*valueOf*(getWorksheet.*jobLength*) + **"post"**);  getWorksheet.*getJSON*();  getWorksheet.*getjobJSON*();  Log.*d*(**"get0000"**, String.*valueOf*(getWorksheet.*jobLength*) + **"get"**);  getWorksheet.*getcalendarJSON*();  Intent intent = **new** Intent(Jobview\_Addjob.**this**,NavigationActivity.**class**);  intent.putExtra(**"id"**,2);  startActivity(intent);  }  });  }   **public void** showDatePickerDialogStart() {  *// 設定初始日期* **final** Calendar c = Calendar.*getInstance*();  **mYear** = c.get(Calendar.***YEAR***);  **mMonth** = c.get(Calendar.***MONTH***);  **mDay** = c.get(Calendar.***DAY\_OF\_MONTH***);   *// 跳出日期選擇器* DatePickerDialog dpd = **new** DatePickerDialog(**this**,  **new** DatePickerDialog.OnDateSetListener() {  **public void** onDateSet(DatePicker view, **int** year,  **int** monthOfYear, **int** dayOfMonth) {  *// 完成選擇，顯示日期* **start\_date**.setText(year + **"-"** + (monthOfYear + 1) + **"-"** + dayOfMonth);   }  }, **mYear**, **mMonth**, **mDay**);  dpd.show();  }   **public void** showTimePickerDialogStart() {  *// 設定初始時間* **final** Calendar c = Calendar.*getInstance*();  **mHour** = c.get(Calendar.***HOUR\_OF\_DAY***);  **mMinute** = c.get(Calendar.***MINUTE***);   *// 跳出時間選擇器* TimePickerDialog tpd = **new** TimePickerDialog(**this**,  **new** TimePickerDialog.OnTimeSetListener() {  **public void** onTimeSet(TimePicker view, **int** hourOfDay,  **int** minute) {  *// 完成選擇，顯示時間* **if** (minute<10){  **start\_time**.setText(hourOfDay + **":"** + **"0"**+minute);  }**else** {  **start\_time**.setText(hourOfDay + **":"** + minute);  }  }  }, **mHour**, **mMinute**, **false**);  tpd.show();  }  **public void** showDatePickerDialogEnd() {  *// 設定初始日期* **final** Calendar c = Calendar.*getInstance*();  **mYear** = c.get(Calendar.***YEAR***);  **mMonth** = c.get(Calendar.***MONTH***);  **mDay** = c.get(Calendar.***DAY\_OF\_MONTH***);   *// 跳出日期選擇器* DatePickerDialog dpd = **new** DatePickerDialog(**this**,  **new** DatePickerDialog.OnDateSetListener() {  **public void** onDateSet(DatePicker view, **int** year,  **int** monthOfYear, **int** dayOfMonth) {  *// 完成選擇，顯示日期* **end\_date**.setText(year + **"-"** + (monthOfYear + 1) + **"-"** + dayOfMonth);   }  }, **mYear**, **mMonth**, **mDay**);  dpd.show();  }   **public void** showTimePickerDialogEnd() {  *// 設定初始時間* **final** Calendar c = Calendar.*getInstance*();  **mHour** = c.get(Calendar.***HOUR\_OF\_DAY***);  **mMinute** = c.get(Calendar.***MINUTE***);   *// 跳出時間選擇器* TimePickerDialog tpd = **new** TimePickerDialog(**this**,  **new** TimePickerDialog.OnTimeSetListener() {  **public void** onTimeSet(TimePicker view, **int** hourOfDay,  **int** minute) {  *// 完成選擇，顯示時間* **if** (minute<10){  **end\_time**.setText(hourOfDay + **":"** + **"0"**+minute);  }**else** {  **end\_time**.setText(hourOfDay + **":"** + minute);  }  }  }, **mHour**, **mMinute**, **false**);  tpd.show();  }   **private** TextWatcher **textWatcher** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);   }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);  Is\_Valid\_Email(**edit\_salary**);   }  **public void** Is\_Valid\_Email(EditText edt) {  **if**(**edit\_salary**.isFocusable()) {  **if** (edt.getText().toString().matches(**"0"**)) {  edt.setError(**"數字不得為0開頭!"**);  edt.setText(**""**);  }  }**else** {   }  }    **boolean** isEmailValid(CharSequence email) {  **return** android.util.Patterns.***EMAIL\_ADDRESS***.matcher(email)  .matches();  } *// end of TextWatcher (email)* };   **private** TextWatcher **textWatcher1** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);   }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);  Is\_Valid\_Email(**edit\_number\_people**);  }  **public void** Is\_Valid\_Email(EditText edt) {  **if** (edt.getText().toString().matches(**"0"**)) {  edt.setError(**"數字不得為0開頭!"**);  edt.setText(**""**);  }  }  };   **private** TextWatcher **textWatcher3** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);  Is\_Valid\_Email(**edit\_email**);  }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);   }  **public void** Is\_Valid\_Email(EditText edt) {  **if** (edt.getText().toString().equals(**""**)) {  edt.setError(**"電子郵件不得為空!"**);  **valid\_email** = **null**;  } **else if** (isEmailValid(edt.getText().toString()) == **false**) {  edt.setError(**"無效的電子郵件格式!"**);  **valid\_email** = **null**;  } **else** {  **valid\_email** = edt.getText().toString();  }  }   **boolean** isEmailValid(CharSequence email) {  **return** android.util.Patterns.***EMAIL\_ADDRESS***.matcher(email)  .matches();  } *// end of TextWatcher (email)* };   **private** TextWatcher **textWatcher2** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);  Is\_Valid\_Email(**edit\_contact**);  }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);   }  **public void** Is\_Valid\_Email(EditText edt) {  **try** {  *//if ((heighText.getText().toString())!=null)* **if**(isInteger(edt.getText().toString()) ==**true**)  edt.setError(**"不能為數字"**);  } **catch** (Exception e) {   }  }  **boolean** isNumber(String value) {  **return** isInteger(value) || isDouble(value);  } *// end of TextWatcher (email)  /\*\*  \* 判断字符串是否是整数  \*/* **boolean** isInteger(String value) {  **try** {  Integer.*parseInt*(value);  **return true**;  } **catch** (NumberFormatException e) {  **return false**;  }  }   */\*\*  \* 判断字符串是否是浮点数  \*/* **boolean** isDouble(String value) {  **try** {  Double.*parseDouble*(value);  **if** (value.contains(**"."**))  **return true**;  **return false**;  } **catch** (NumberFormatException e) {  **return false**;  }  }  };   **private** TextWatcher **textWatcher4** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);  **if** (**edit\_phone**.length()<2){  **edit\_phone**.setError(**"長度不為10碼"**);  }**else if** (s.length() != 10) {  **edit\_phone**.setError(**"長度不為10碼"**);  }**else if**(!(**edit\_phone**.getText().toString().substring(0,2).equals(**"09"**))) {  **edit\_phone**.setError(**"手機格式不符合"**);  Log.*d*(**"TAGGGGGGGGGGG"**, **edit\_phone**.getText().toString().substring(0,2) );  }**else**{   } *//* }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);    }  };   **private** TextWatcher **textWatcher5** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);  Is\_Valid\_Email(**end\_date**);  }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);   }  **public void** Is\_Valid\_Email(EditText edt) {  **try** {  SimpleDateFormat sdf = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  Date d1 = sdf.parse(**start\_date**.getText().toString());  Date d2 = sdf.parse(**end\_date**.getText().toString());  **long** result = d1.getTime()-d2.getTime();  Log.*d*(**"dateeee"**, String.*valueOf*(d1.getTime()-d2.getTime()));  **if** (result<=0){  **end\_date**.setError(**null**); *// }else if (result == 0) { // end\_date.setError("結束日期等於初始日期");* }**else if**(result>0){  **end\_date**.setError(**"結束日期小於初始日期"**);  }  } **catch** (Exception e) {   }  }  };   **private** TextWatcher **textWatcher6** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);  Is\_Valid\_Email(**start\_date**);  }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);   }  **public void** Is\_Valid\_Email(EditText edt) {  **try** {  SimpleDateFormat sdf = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  Date d1 = sdf.parse(**start\_date**.getText().toString());  Date d2 = sdf.parse(**end\_date**.getText().toString());  **long** result = d1.getTime()-d2.getTime();  Log.*d*(**"dateeee"**, String.*valueOf*(d1.getTime()-d2.getTime()));  **if** (result<=0){  **end\_date**.setError(**null**); *// }else if (result == 0) { // end\_date.setError("結束日期等於初始日期");* }**else if**(result>0){  **end\_date**.setError(**"結束日期小於初始日期"**);  }  } **catch** (Exception e) {   }  }  };   **private** TextWatcher **textWatcher7** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);  Is\_Valid\_Email(**end\_time**);  }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);   }  **public void** Is\_Valid\_Email(EditText edt) {  **try** {  SimpleDateFormat sdf = **new** SimpleDateFormat(**"HH:mm"**);  Date d1 = sdf.parse(**start\_time**.getText().toString());  Date d2 = sdf.parse(**end\_time**.getText().toString());  **long** result = d1.getTime()-d2.getTime();  Log.*d*(**"timeeee"**, String.*valueOf*(d1.getTime()-d2.getTime()));  **if** (result<0){  **end\_time**.setError(**null**);  }**else if** (result == 0) {  **end\_time**.setError(**"結束時間等於開始時間"**);  }**else if**(result>0){  **end\_time**.setError(**"結束時間小於開始時間"**);  }   } **catch** (Exception e) {   }  }  };   **private** TextWatcher **textWatcher8** = **new** TextWatcher() {  @Override  **public void** afterTextChanged(Editable s) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"afterTextChanged--------------->"**);  Is\_Valid\_Email(**start\_time**);  }   @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count,  **int** after) {  *//* ***TODO Auto-generated method stub*** Log.*d*(**"TAG"**, **"beforeTextChanged--------------->"**);  }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before,  **int** count) {  Log.*d*(**"TAG"**, **"onTextChanged--------------->"**);   }  **public void** Is\_Valid\_Email(EditText edt) {  **try** {  SimpleDateFormat sdf = **new** SimpleDateFormat(**"HH:mm"**);  Date d1 = sdf.parse(**start\_time**.getText().toString());  Date d2 = sdf.parse(**end\_time**.getText().toString());  **long** result = d1.getTime()-d2.getTime();  Log.*d*(**"timeeee"**, String.*valueOf*(d1.getTime()-d2.getTime()));  **if** (result<0){  **end\_time**.setError(**null**);  }**else if** (result == 0) {  **end\_time**.setError(**"結束時間等於開始時間"**);  }**else if**(result>0){  **end\_time**.setError(**"結束時間小於開始時間"**);  }   } **catch** (Exception e) {   }  }  }; } | |

|  |  |
| --- | --- |
| 程式名稱 | MainActivityLogin |
| 目的 | 登入功能 |
| 程式碼 | |
| **public class** MainActivityLogin **extends** AppCompatActivity {  **private** Button **buttonByPass**, **buttonRegister**;  **private** EditText **email**;  **private** EditText **password**;  **private** String **valid\_email**;  **private** String **valid\_password**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_login***);  getWorksheet.*gethumansearchJSON*();  getWorksheet.*getjobJSON*();  getWorksheet.*getcalendarJSON*();  **email** = (EditText) findViewById(R.id.***email***);  **password** = (EditText) findViewById(R.id.***password***);  **email**.addTextChangedListener(**textWatcher**);  **password**.addTextChangedListener(**textWatcher1**);  **buttonByPass** = findViewById(R.id.***buttonByPass***);  **buttonByPass**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  openBypass();}  });  **buttonRegister** = findViewById(R.id.***buttonRegister***);  **buttonRegister**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  openRegister();  }  });  findViewById(R.id.***button***).setOnClickListener((v)->  { postLogin();  });  }  **private void** postLogin()  {  *// Service的URL* String url = **"http://140.131.114.153/postLogin.php"**;  *//對應Postman StringRequest -> content-type=不設置* StringRequest request = **new** StringRequest(Request.Method.***POST***,url,  **new** Response.Listener<String>()  {  @Override  *//respone = php回應的結果* **public void** onResponse(String response)  {  Log.*i*(**"Log"**,**"result:"** + response);  **try** {  String ret = parsePostLoginJSon(response);  **if**(ret.equals(**"成功"**))  {  openHtl();}  **else** {  Log.*i*(**"Log"**,**"帳密錯誤"**);  }  } **catch** (JSONException e) {  e.printStackTrace();}  }},  **new** Response.ErrorListener()  {  *// Request失敗處理* @Override  **public void** onErrorResponse(VolleyError error)  {  Toast.*makeText*(getApplicationContext(), error.getMessage(), Toast.***LENGTH\_LONG***).show();  }  }  {  @Override  *//=postman keyvalue (formdata)* **protected** Map<String, String> getParams() {  Map<String, String> param = **new** HashMap<>();  param.put(**"row1"**, **email**.getText().toString());  param.put(**"row2"**, **password**.getText().toString());  **return** param;  }  };  *// 將request放到RequestQueue，Volley會以非同步方式送出request* ApplicationController.*getInstance*().addToRequestQueue(request);  }  *//string 轉json* **private** String parsePostLoginJSon(String response) **throws** JSONException  {  **if** (response == **null**)  **return null**;  JSONObject jsonResponse = **new** JSONObject(response);  JSONArray jsonStars = jsonResponse.getJSONArray(**"result"**);  **if**(jsonStars.length() > 0)  {  **return "成功"**;  }  **return "失敗"**;  } } | |

|  |  |
| --- | --- |
| 程式名稱 | MainActivityRegister |
| 目的 | 註冊功能 |
| 程式碼 | |
| **public class** MainActivityRegister **extends** AppCompatActivity {  **private** EditText **email**;  **private** EditText **password**;  **private** EditText **password1**;  **private** EditText **name\_hostel**;  **private** EditText **phone\_hostel**;  **private** String **valid\_email**;  **private** String **valid\_password**;  **private** String **valid\_password1**;  **private** String **valid\_name\_hostel**;  **private** String **valid\_phone\_hostel**;  @RequiresApi(api = Build.VERSION\_CODES.***JELLY\_BEAN***)  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_register***);  **email** = (EditText) findViewById(R.id.***email***);  **password** = (EditText) findViewById(R.id.***password***);  **password1** = (EditText)findViewById(R.id.***password1***);  **name\_hostel**=(EditText)findViewById(R.id.***name\_hostel***);  **phone\_hostel**=(EditText)findViewById(R.id.***phone\_hostel***);  **email**.addTextChangedListener(**textWatcher**);  **password**.addTextChangedListener(**textWatcher1**);  **password1**.addTextChangedListener(**textWatcher2**);  **name\_hostel**.addTextChangedListener(**textWatcher3**);  **phone\_hostel**.addTextChangedListener(**textWatcher4**);   findViewById(R.id.***button13***).setOnClickListener((v)->  {  postRegister();  openLogin();  });  **public class** Register  {  **public** String **row1**;  **public** String **row2**;  **public** String **row3**;  **public** String **row4**;  }   **private** List<MainActivityRegister.Register> parseGetJSon(String data) **throws** JSONException  {  **if** (data == **null**)  **return null**;   List<MainActivityRegister.Register> list = **new** ArrayList<MainActivityRegister.Register>();  JSONObject jsonResponse = **new** JSONObject(data); *// String ret = jsonResponse.getString("result");* JSONArray jsonStars = jsonResponse.getJSONArray(**"result"**);  **for** (**int** i = 0; i < jsonStars.length(); i++)  {  JSONObject jsonStar = jsonStars.getJSONObject(i);   MainActivityRegister.Register r = **new** MainActivityRegister.Register();  r.**row1** = jsonStar.getString(**"row1"**);  r.**row2** = jsonStar.getString(**"row2"**);  r.**row3** = jsonStar.getString(**"row3"**);  r.**row4** = jsonStar.getString(**"row4"**);  list.add(r);  }  **return** list;  }    **private void** generateAllStars()  {  *// Service的URL* String url = **"http://140.131.114.153/getRegister"**;   *// 建立期望收到JsonObject的request  // 若new JsonObjectRequest的第二個參數為null則代表是GET，反之若為JSONObject則為POST* JsonObjectRequest request = **new** JsonObjectRequest(url, **null**,  **new** Response.Listener<JSONObject>()  {  *// 若server回傳HTTP status 200或204則會進入onResponse method  // 註:根據官網Issue 57527的討論，最新版應該是2xx都算request success* @Override  **public void** onResponse(JSONObject response)  {  Log.*i*(**"Log"**,**"result:"** + response);  List<MainActivityRegister.Register> list = **new** ArrayList<MainActivityRegister.Register>();  **try** {  list = parseGetJSon(response.toString());  **for**(MainActivityRegister.Register r : list)  {  Log.*i*(**"Log"**,**"Register:"** + r.**row1**);  }  }  **catch**(Exception e)  {  Log.*i*(**"Log"**, **"Exception: "** + e.toString());  }  }  },  **new** Response.ErrorListener()  {  *// Request失敗處理* @Override  **public void** onErrorResponse(VolleyError error)  {  Toast.*makeText*(getApplicationContext(), error.getMessage(), Toast.***LENGTH\_LONG***).show();  }  }  );  *// 將request放到RequestQueue，Volley會以非同步方式送出request* ApplicationController.*getInstance*().addToRequestQueue(request);  }   **public void** openLogin(){  Intent intent =**new** Intent(**this**, MainActivityLogin.**class**);  startActivity(intent);  }   **private void** postRegister()  {  *// Service的URL* String url = **"http://140.131.114.153/postRegister.php"**;   *// 建立期望收到JsonObject的request  // 若new JsonObjectRequest的第二個參數為null則代表是GET，反之若為JSONObject則為POST* StringRequest request = **new** StringRequest(Request.Method.***POST***,url,  **new** Response.Listener<String>()  {  @Override  **public void** onResponse(String response)  {  Log.*i*(**"Log"**,**"result:"** + response);  }  },  **new** Response.ErrorListener()  {  *// Request失敗處理* @Override  **public void** onErrorResponse(VolleyError error)  {  Toast.*makeText*(getApplicationContext(), error.getMessage(), Toast.***LENGTH\_LONG***).show();  }  }  )  {  @Override  **protected** Map<String, String> getParams() {  Map<String, String> param = **new** HashMap<>();   param.put(**"row1"**, **email**.getText().toString());  param.put(**"row2"**, **password**.getText().toString());  param.put(**"row3"**, **name\_hostel**.getText().toString());  param.put(**"row4"**, **phone\_hostel**.getText().toString());  **return** param;  }  };  *// 將request放到RequestQueue，Volley會以非同步方式送出request* ApplicationController.*getInstance*().addToRequestQueue(request);  } } | |

|  |  |
| --- | --- |
| 程式名稱 | NavigationActivity |
| 目的 | 顯示Fragement各頁面 |
| 程式碼 | |
| **public class** NavigationActivity **extends** AppCompatActivity {   **private** TextView **mTextMessage**;  **private** Fragment\_Humansearch **Fragment1HumanSearch**;  **private** Fragment\_Jobview **Fragment2JobView**;  **private** Fragment\_Inbox **Fragment3Inbox**;  **private** Fragment\_Calenderview **Fragment4CalenderView**;  **private** Fragment\_Bosssetting **Fragment5BossSetting**;  BottomNavigationView **navigation**;  **private** BottomNavigationView.OnNavigationItemSelectedListener **mOnNavigationItemSelectedListener** = **new** BottomNavigationView.OnNavigationItemSelectedListener() {   @Override  **public boolean** onNavigationItemSelected(@NonNull MenuItem item) {  Log.*v*(**"text"**, **"abc"**);   **switch** (item.getItemId()) {  **case** R.id.***navigation\_people***:  getWorksheet.*gethumansearchJSON*();  **mTextMessage**.setText(**""**);  showNav(R.id.***navigation\_people***);  **return true**;  **case** R.id.***navigation\_publish***:  getWorksheet.*getjobJSON*();  **mTextMessage**.setText(**""**);  showNav(R.id.***navigation\_publish***);  **return true**;  **case** R.id.***navigation\_inbox***:  getWorksheet.*getjobJSON*();  **mTextMessage**.setText(**""**);  showNav(R.id.***navigation\_inbox***);  **return true**;  **case** R.id.***navigation\_work***:  getWorksheet.*getjobJSON*();  **mTextMessage**.setText(**""**);  showNav(R.id.***navigation\_work***);  **return true**;  **case** R.id.***navigation\_page***:  **mTextMessage**.setText(**""**);  showNav(R.id.***navigation\_page***);  **return true**;  }  **return false**;  }   };   @Override  **public void** onBackPressed() {  }   @RequiresApi(api = Build.VERSION\_CODES.***JELLY\_BEAN***)  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_navigation***);  init();  **mTextMessage** = (TextView) findViewById(R.id.***message***);  **navigation** = (BottomNavigationView) findViewById(R.id.***navigation***);  **navigation**.setOnNavigationItemSelectedListener(**mOnNavigationItemSelectedListener**);  String user = getSharedPreferences(**"userpw"**, ***MODE\_PRIVATE***).getString(**"USER"**, **""**);  String pw = getSharedPreferences(**"userpw"**, ***MODE\_PRIVATE***).getString(**"PW"**, **""**);  Log.*v*(**"useraa"**, user);  Log.*v*(**"useraa"**, pw); *// final int notifyID = 1; // 通知的識別號碼 // final NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE); // 取得系統的通知服務 // final Notification notification = new Notification.Builder(getApplicationContext()).setSmallIcon(R.drawable.logo1).setContentTitle("我是凱偉").setContentText("我想吃小寶屌").build(); // 建立通知 // notificationManager.notify(notifyID, notification); // 發送通知 // Intent notifyIntent = new Intent(NavigationActivity.this, NavigationActivity.class); // notifyIntent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK); // PendingIntent appIntent // = PendingIntent.getActivity(NavigationActivity.this, 0, notifyIntent, 0); // NotificationManager mNotificationManager = (NotificationManager)getSystemService(NOTIFICATION\_SERVICE); // // //Step2. 設定當按下這個通知之後要執行的activity // Intent notifyIntent = new Intent(NavigationActivity.this, NavigationActivity.class); // notifyIntent.setFlags( Intent.FLAG\_ACTIVITY\_NEW\_TASK); // PendingIntent appIntent = PendingIntent.getActivity(NavigationActivity.this, 0, notifyIntent, 0); // Notification notification // = new Notification.Builder(NavigationActivity.this) // .setContentIntent(appIntent) // .setSmallIcon(R.drawable.logo1) // 設置狀態列裡面的圖示（小圖示）　　 // .setLargeIcon(BitmapFactory.decodeResource(NavigationActivity.this.getResources(), R.drawable.logo2)) // 下拉下拉清單裡面的圖示（大圖示） // .setTicker("notification on status bar.") // 設置狀態列的顯示的資訊 // .setWhen(System.currentTimeMillis())// 設置時間發生時間 // .setAutoCancel(false) // 設置通知被使用者點擊後是否清除 //notification.flags = Notification.FLAG\_AUTO\_CANCEL; // .setContentTitle("Notification Title") // 設置下拉清單裡的標題 // .setContentText("Notification Content")// 設置上下文內容 // .setOngoing(true) //true使notification变为ongoing，用户不能手动清除 // notification.flags = Notification.FLAG\_ONGOING\_EVENT; notification.flags = Notification.FLAG\_NO\_CLEAR; // // .setDefaults(Notification.DEFAULT\_ALL) //使用所有默認值，比如聲音，震動，閃屏等等 // .setDefaults(Notification.DEFAULT\_VIBRATE) //使用默認手機震動提示 // .setDefaults(Notification.DEFAULT\_SOUND) //使用默認聲音提示 // .setDefaults(Notification.DEFAULT\_LIGHTS) //使用默認閃光提示 // .setDefaults(Notification.DEFAULT\_LIGHTS | Notification.DEFAULT\_SOUND) //使用默認閃光提示 與 默認聲音提示 // //// .setVibrate(vibrate) //自訂震動長度 //// .setSound(uri) //自訂鈴聲 // .setLights(0xff00ff00, 300, 1000) //自訂燈光閃爍 (ledARGB, ledOnMS, ledOffMS) // .build(); // ////把指定ID的通知持久的發送到狀態條上 // notification.flags = Notification.FLAG\_ONGOING\_EVENT; // // // 表明在點擊了通知欄中的"清除通知"後，此通知不清除， // // 經常與FLAG\_ONGOING\_EVENT一起使用 // notification.flags = Notification.FLAG\_NO\_CLEAR; // // //閃爍燈光 // notification.flags = Notification.FLAG\_SHOW\_LIGHTS; // // // 重複的聲響,直到用戶響應。 // notification.flags = Notification.FLAG\_INSISTENT; // // // // 把指定ID的通知持久的發送到狀態條上. // mNotificationManager.notify(0, notification); // // // 取消以前顯示的一個指定ID的通知.假如是一個短暫的通知， // // 試圖將之隱藏，假如是一個持久的通知，將之從狀態列中移走. //// mNotificationManager.cancel(0); // // //取消以前顯示的所有通知. //// mNotificationManager.cancelAll();* }   @Override  **public void** onResume() {  getWorksheet.*gethumansearchJSON*();  getWorksheet.*getjobJSON*();  getWorksheet.*getcalendarJSON*();  getWorksheet.*gethostelinfoJSON*();  getWorksheet.*getResumeJSON*();  getWorksheet.*gethostelnameJSON*();  **int** id = getIntent().getIntExtra(**"id"**, 0);  **navigation** = (BottomNavigationView) findViewById(R.id.***navigation***);  FragmentTransaction beginTransaction = getFragmentManager().beginTransaction();  **switch** (id) {  **case** 1:  beginTransaction.hide(**Fragment2JobView**).hide(**Fragment3Inbox**).hide(**Fragment4CalenderView**).hide(**Fragment5BossSetting**);  beginTransaction.show(**Fragment1HumanSearch**);  beginTransaction.addToBackStack(**null**);  beginTransaction.commit();  **navigation**.getMenu().getItem(0).setChecked(**true**);  **break**;  **case** 2:  beginTransaction.hide(**Fragment1HumanSearch**).hide(**Fragment3Inbox**).hide(**Fragment4CalenderView**).hide(**Fragment5BossSetting**);  beginTransaction.show(**Fragment2JobView**);  beginTransaction.addToBackStack(**null**);  beginTransaction.commit();  **navigation**.getMenu().getItem(1).setChecked(**true**);  **break**;  **case** 3:  beginTransaction.hide(**Fragment2JobView**).hide(**Fragment1HumanSearch**).hide(**Fragment5BossSetting**).hide(**Fragment4CalenderView**);  beginTransaction.show(**Fragment3Inbox**);  beginTransaction.addToBackStack(**null**);  beginTransaction.commit();  **navigation**.getMenu().getItem(2).setChecked(**true**);  **break**;  **case** 4:  beginTransaction.hide(**Fragment2JobView**).hide(**Fragment3Inbox**).hide(**Fragment1HumanSearch**).hide(**Fragment5BossSetting**);  beginTransaction.show(**Fragment4CalenderView**);  beginTransaction.addToBackStack(**null**);  beginTransaction.commit();  **navigation**.getMenu().getItem(2).setChecked(**true**);  **break**;  **case** 5:  beginTransaction.hide(**Fragment1HumanSearch**).hide(**Fragment2JobView**).hide(**Fragment3Inbox**).hide(**Fragment4CalenderView**);  beginTransaction.show(**Fragment5BossSetting**);  beginTransaction.addToBackStack(**null**);  beginTransaction.commit();  **mTextMessage**.setText(**""**);  showNav(R.id.***navigation\_page***);  **navigation**.getMenu().getItem(3).setChecked(**true**);  **break**;  }   **super**.onResume();  }   *// @Override // protected void onStart() { // super.onStart(); // getWorksheet.getjobJSON(); // // The activity is about to become visible. // } // @Override // protected void onResume() { // super.onResume(); // getWorksheet.getjobJSON(); // // The activity has become visible (it is now "resumed"). // }  //init（）用来初始化组件* **private void** init() {  **Fragment1HumanSearch** = **new** Fragment\_Humansearch();  **Fragment2JobView** = **new** Fragment\_Jobview();  **Fragment3Inbox** = **new** Fragment\_Inbox();  **Fragment4CalenderView** = **new** Fragment\_Calenderview();  **Fragment5BossSetting** = **new** Fragment\_Bosssetting();  FragmentTransaction beginTransaction = getFragmentManager().beginTransaction();  beginTransaction.add(R.id.***content***, **Fragment1HumanSearch**).add(R.id.***content***, **Fragment2JobView**).add(R.id.***content***, **Fragment3Inbox**).add(R.id.***content***, **Fragment4CalenderView**).add(R.id.***content***, **Fragment5BossSetting**);*//开启一个事务将fragment动态加载到组件* beginTransaction.hide(**Fragment1HumanSearch**).hide(**Fragment2JobView**).hide(**Fragment3Inbox**).hide(**Fragment4CalenderView**).hide(**Fragment5BossSetting**);*//隐藏fragment* beginTransaction.addToBackStack(**null**);*//返回到上一个显示的fragment* beginTransaction.commit();*//每一个事务最后操作必须是commit（），否则看不见效果* showNav(R.id.***navigation\_people***);   }   **private void** showNav(**int** navid) {  FragmentTransaction beginTransaction = getFragmentManager().beginTransaction();  **switch** (navid) {  **case** R.id.***navigation\_people***:  beginTransaction.hide(**Fragment2JobView**).hide(**Fragment3Inbox**).hide(**Fragment4CalenderView**).hide(**Fragment5BossSetting**);  beginTransaction.show(**Fragment1HumanSearch**);  beginTransaction.addToBackStack(**null**);  beginTransaction.commit();  **break**;  **case** R.id.***navigation\_publish***:  beginTransaction.hide(**Fragment1HumanSearch**).hide(**Fragment3Inbox**).hide(**Fragment4CalenderView**).hide(**Fragment5BossSetting**);  beginTransaction.show(**Fragment2JobView**);  beginTransaction.addToBackStack(**null**);  beginTransaction.commit();  **break**;  **case** R.id.***navigation\_inbox***:  beginTransaction.hide(**Fragment1HumanSearch**).hide(**Fragment2JobView**).hide(**Fragment4CalenderView**).hide(**Fragment5BossSetting**);  beginTransaction.show(**Fragment3Inbox**);  beginTransaction.addToBackStack(**null**);  beginTransaction.commit();  **break**;  **case** R.id.***navigation\_work***:  beginTransaction.hide(**Fragment2JobView**).hide(**Fragment3Inbox**).hide(**Fragment1HumanSearch**).hide(**Fragment5BossSetting**);  beginTransaction.show(**Fragment4CalenderView**);  beginTransaction.addToBackStack(**null**);  beginTransaction.commit();  **break**;  **case** R.id.***navigation\_page***:  beginTransaction.hide(**Fragment1HumanSearch**).hide(**Fragment2JobView**).hide(**Fragment3Inbox**).hide(**Fragment4CalenderView**);  beginTransaction.show(**Fragment5BossSetting**);  beginTransaction.addToBackStack(**null**);  beginTransaction.commit();  **break**;  }  } } | |

|  |  |
| --- | --- |
| 程式名稱 | Setting\_Hotelinfo |
| 目的 | 跳轉至新增民宿資訊 |
| 程式碼 | |
| **public class** Setting\_Hotelinfo **extends** AppCompatActivity {   RecyclerView **mList**;  ArrayList<String> **myDataset**;  ArrayList<String> **myDataset2**;  ArrayList<String> **myDataset3**;  ArrayList<String> **myDataset4**;  MyAdapter **myAdapter**;  **static** LinearLayoutManager *layoutManager*;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***setting\_hotelinfo***);  getWorksheet.*gethostelinfoJSON*();  FloatingActionButton fab = (FloatingActionButton)findViewById(R.id.***fab\_go\_to\_add\_hotel***) ;  **mList** = (RecyclerView)findViewById(R.id.***list\_view***);  setJobview();  fab.setOnClickListener(**new** Button.OnClickListener(){  @Override  **public void** onClick(View v) {   Intent intent = **new** Intent(Setting\_Hotelinfo.**this**,Hotelview\_Addhotel.**class**);  startActivity(intent);  }  });  }  **public void** setJobview(){  **myDataset** = **new** ArrayList<>();  **myDataset2** = **new** ArrayList<>();  **myDataset3** = **new** ArrayList<>();  **myDataset4** = **new** ArrayList<>();  **myAdapter** = **new** MyAdapter(**myDataset**);  **for**(**int** i = 0; i < getWorksheet.*hostelinfoLength*; i++){ *// myDataset.add(i + "");* **myDataset**.add(getWorksheet.*getRow32*(i));  **myDataset2**.add(getWorksheet.*getRow33*(i));  **myDataset3**.add(getWorksheet.*getRow34*(i));  **myDataset4**.add(getWorksheet.*getRow35*(i));  } *// mList = (RecyclerView)view.findViewById(R.id.list\_view);  layoutManager* = **new** LinearLayoutManager(**this**);  *layoutManager*.setOrientation(LinearLayoutManager.***VERTICAL***);  **mList**.setLayoutManager(*layoutManager*);  **mList**.setAdapter(**myAdapter**);  }  **public class** MyAdapter **extends** RecyclerView.Adapter<MyAdapter.ViewHolder> {  **private** List<String> **mData**;  **public class** ViewHolder **extends** RecyclerView.ViewHolder {  **public** TextView **txt\_hotel\_name** ,**txt\_hotel\_address** ,**txt\_hotel\_phone**;  **public** ImageView **img\_hotel\_photo**;  **public** ViewHolder(View v) {  **super**(v);  **img\_hotel\_photo** = (ImageView) v.findViewById(R.id.***img\_hotel\_photo***);  **txt\_hotel\_name** = (TextView) v.findViewById(R.id.***txt\_hotel\_name***);  **txt\_hotel\_address** = (TextView) v.findViewById(R.id.***txt\_hotel\_address***);  **txt\_hotel\_phone** = (TextView) v.findViewById(R.id.***txt\_hotel\_phone***);  }  }  **public** MyAdapter(List<String> data) {  **mData** = data;  }  @Override  **public** MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {  View v = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.***hotelview\_item***, parent, **false**);  ViewHolder vh = **new** ViewHolder(v);  **return** vh;  }  @Override  **public void** onBindViewHolder(MyAdapter.ViewHolder holder, **int** position) {  holder.**img\_hotel\_photo**.setImageBitmap(stringToBitmap(**mData**.get(position)));  holder.**txt\_hotel\_name**.setText(**myDataset2**.get(position));  holder.**txt\_hotel\_address**.setText(**myDataset3**.get(position));  holder.**txt\_hotel\_phone**.setText(**myDataset4**.get(position));  }  @Override  **public int** getItemCount() {  **return mData**.size();  }  }  **public** Bitmap stringToBitmap(String string) {  Bitmap bitmap = **null**;  **try** {  **byte**[] bitmapArray = Base64.*decode*(string, Base64.***DEFAULT***);  bitmap = BitmapFactory.*decodeByteArray*(bitmapArray, 0, bitmapArray.**length**);  **return** bitmap;  } **catch** (NullPointerException e) {  e.getMessage();  **return null**;  } **catch** (OutOfMemoryError e) {  **return null**;  }  } } | |

▼表 9-2-2 學生端程式規格描述

|  |  |
| --- | --- |
| 程式名稱 | getAsyncTask |
| 目的 | 非同步取得資料 |
| 程式碼 | |
| *//========================================= // 非同步工作, 用來取回網站回傳的資料 //=========================================* **public class** getAsyncTask **extends** AsyncTask<String, Integer, String> {   *//----------------------------------------  // 宣告一個接收回傳結果的程式必須實作的介面  //----------------------------------------* **public interface** TaskListener {  **void** onFinished(String result);  }   **private** TaskListener **taskListener**;   *//-----------------------------------------------------------  // 建構元, 傳入(1)context, (2)取回資料後執行的程式  //-----------------------------------------------------------* **public** getAsyncTask(TaskListener taskListener) {  **this**.**taskListener** = taskListener;  }   @Override  **protected void** onPreExecute() {  **super**.onPreExecute();  }  @Override  **protected void** onProgressUpdate(Integer... values) {  **super**.onProgressUpdate(values);  }  *//========================================  // 由主程式呼叫.execute()方法時啟動,  // 由主程式傳入:(1)主機網址  //========================================* @Override  **protected** String doInBackground(String... params) {  String data=**null**;  InputStream inputStream = **null**;  ByteArrayOutputStream outputStream = **new** ByteArrayOutputStream();  **try** {  URL url = **new** URL(params[0]);  HttpURLConnection conn = (HttpURLConnection) url.openConnection();  conn.setReadTimeout(10000);  conn.setConnectTimeout(15000);  conn.setRequestMethod(**"GET"**);  *//conn.setDoInput(true);  //conn.setDoOutput(true);* **int** statusCode = conn.getResponseCode();  Log.*v*(**"Test2"**,**"statuus:"**+statusCode);  conn.connect();  inputStream = conn.getInputStream();   **if** (statusCode >= 200 && statusCode < 400) {  *// Create an InputStream in order to extract the response object* inputStream = conn.getInputStream();  }  **else** {  inputStream = conn.getErrorStream();  }   BufferedReader bufferedReader=**new** BufferedReader(  **new** InputStreamReader(inputStream, **"utf-8"**));   data=bufferedReader.readLine();  } **catch**(Exception e) {  e.printStackTrace();  } **finally** {  **try** {  inputStream.close();  outputStream.close();  } **catch**(Exception e) {  e.printStackTrace();  }  }  **return** data;  }   *//------------------------------------------------------------------  // 完成資料取回後, 由主程式的taskListener.onFinished()處理取回資料  //------------------------------------------------------------------* @Override  **protected void** onPostExecute(String result) {  **super**.onPostExecute(result);  **taskListener**.onFinished(result);  }   @Override  **protected void** onCancelled(String result) {  **super**.onCancelled(result);  }   @Override  **protected void** onCancelled() {  **super**.onCancelled();  } } | |

|  |  |
| --- | --- |
| 程式名稱 | postUpdateAsyncTask |
| 目的 | 非同步傳送資料 |
| 程式碼 | |
| **public class** postUpdateAsyncTask **extends** AsyncTask<String, Integer, String> {  *//----------------------------------------------------  // 宣告一個TaskListener介面, 接收回傳值的物件必須實作它  //----------------------------------------------------* **public interface** TaskListener {  **void** onFinished(String result);  } *// 接收回傳值的物件參考* **private final** TaskListener **taskListener**;  *//---------------------------------------  // 建構元, 傳入context及接收回傳值的物件  //---------------------------------------* **public** postUpdateAsyncTask(TaskListener taskListener) {  **this**.**taskListener** = taskListener;  }  @Override  **protected void** onProgressUpdate(Integer... values) {  **super**.onProgressUpdate(values);  }  *//=========================================================  // 執行非同步工作, 建立一個HttpURLConnection, 讀取主機的資料.  //=========================================================* @Override  **protected** String doInBackground(String... params) {  String data=**null**;  InputStream inputStream = **null**;  ByteArrayOutputStream outputStream = **new** ByteArrayOutputStream();   **try** {  URL url = **new** URL(params[0]); *//params[0] 是myNavigationAsyncTask.execute(Common.updateUrl, getId);的第一個參數* HttpURLConnection conn = (HttpURLConnection) url.openConnection();  conn.setReadTimeout(100000);  conn.setConnectTimeout(150000);  conn.setRequestMethod(**"POST"**);  conn.setDoInput(**true**);  conn.setDoOutput(**true**);  *//----------------------------------------------  // 傳給主機的參數(name, amount, deliverDate)  //----------------------------------------------  //params[1] 是myNavigationAsyncTask.execute(Common.updateUrl, getId);的第二個參數* String args =  **"row1="** + URLEncoder.*encode*(params[1], **"UTF-8"**)+  **"&row2="** + URLEncoder.*encode*(params[2], **"UTF-8"**)+  **"&row3="** + URLEncoder.*encode*(params[3], **"UTF-8"**)+  **"&row4="** + URLEncoder.*encode*(params[4], **"UTF-8"**)+  **"&row5="** + URLEncoder.*encode*(params[5], **"UTF-8"**)+  **"&row6="** + URLEncoder.*encode*(params[6], **"UTF-8"**)+  **"&row7="** + URLEncoder.*encode*(params[7], **"UTF-8"**)+  **"&row8="** + URLEncoder.*encode*(params[8], **"UTF-8"**)+  **"&row9="** + URLEncoder.*encode*(params[9], **"UTF-8"**)+  **"&row10="** + URLEncoder.*encode*(params[10], **"UTF-8"**)+  **"&row11="** + URLEncoder.*encode*(params[11], **"UTF-8"**)+  **"&row12="** + URLEncoder.*encode*(params[12], **"UTF-8"**)+  **"&row13="** + URLEncoder.*encode*(params[13], **"UTF-8"**);OutputStream os = conn.getOutputStream();  BufferedWriter writer = **new** BufferedWriter(  **new** OutputStreamWriter(os, **"UTF-8"**));  writer.write(args);  writer.flush();  writer.close();  os.close();  conn.connect();  inputStream = conn.getInputStream();  BufferedReader bufferedReader=**new** BufferedReader(  **new** InputStreamReader(inputStream, **"utf-8"**));  data=bufferedReader.readLine();  } **catch**(Exception e) {  e.printStackTrace();  } **finally** {  **try** {  inputStream.close();  outputStream.close();  } **catch**(Exception e) {  e.printStackTrace();  }  }  **return** data;  }  *//===========================  // 執行完非同步工作之後執行  //===========================* @Override  **protected void** onPostExecute(String result) {  **super**.onPostExecute(result);  **taskListener**.onFinished(result);  }  @Override  **protected void** onCancelled(String result) {  **super**.onCancelled(result);  }  @Override  **protected void** onCancelled() {  **super**.onCancelled();  } } | |

|  |  |
| --- | --- |
| 程式名稱 | Fragment\_Findhostel |
| 目的 | 顯示換宿清單 |
| 程式碼 | |
| **public class** Fragment\_Findhostel **extends** Fragment {  **static** TextView *textView30*;  **static** TextView *textView31*;  **static** TextView *textView32*;  **static** TextView *textView33*;  **static** TextView *textView34*;  **static** String[] *row* = **new** String[100];  **static** String *row9*;  **static** String *row10*;  **static** String *row11*;  **static** String *row12*;  RecyclerView **mList**;  ArrayList<String> **myDataset**;  ～  ArrayList<String> **myDataset16**;  MyAdapter **myAdapter**;  **static** LinearLayoutManager *layoutManager*;   @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {  View view = inflater.inflate(R.layout.***fragment\_findhostel***, container, **false**);  **mList** = (RecyclerView) view.findViewById(R.id.***list\_view***);  **myDataset** = **new** ArrayList<>();  ～  **myDataset16** = **new** ArrayList<>();  **myAdapter** = **new** MyAdapter(**myDataset**);  **for** (**int** i = 0; i < getWorksheet.*hostelLength*; i++) { *// myDataset.add(i + "");* **myDataset**.add(getWorksheet.*getRow19*(i));  ～  **myDataset16**.add(getWorksheet.*getRow34*(i));  Log.*d*(**"get0000"**, String.*valueOf*(getWorksheet.*hostelLength*) + **"hostelLength\_resume"**);  }  *layoutManager* = **new** LinearLayoutManager(getActivity());  *layoutManager*.setOrientation(LinearLayoutManager.***VERTICAL***);  **mList**.setLayoutManager(*layoutManager*);  **mList**.setAdapter(**myAdapter**);  *//resume* **return** view;  }  **public void** onHiddenChanged(**boolean** hidden) {  **super**.onHiddenChanged(hidden);  **if** (hidden) {  *//pause* } **else** {  **myDataset** = **new** ArrayList<>();  ～  **myDataset16** = **new** ArrayList<>();  **myAdapter** = **new** MyAdapter(**myDataset**);  **for** (**int** i = 0; i < getWorksheet.*hostelLength*; i++) { *// myDataset.add(i + "");* **myDataset**.add(getWorksheet.*getRow19*(i));  ～  **myDataset16**.add(getWorksheet.*getRow34*(i));   *layoutManager* = **new** LinearLayoutManager(getActivity());  *layoutManager*.setOrientation(LinearLayoutManager.***VERTICAL***);  **mList**.setLayoutManager(*layoutManager*);  **mList**.setAdapter(**myAdapter**);  *//resume* }  }  **public class** MyAdapter **extends** RecyclerView.Adapter<MyAdapter.ViewHolder> {  **private** List<String> **mData**;  **private** Context **mContext**;  **public class** ViewHolder **extends** RecyclerView.ViewHolder {  **public** TextView **TextViewHostelName**, **TextViewHostelAddr**, **TextViewNumPeople**, **TextViewDays**,  **TextViewSkill**,**TextViewHostelStar**;  **public** ImageView **TextViewHostelPic**;  **public** ImageButton **ib\_popup\_menu**;  **public** CardView **cardView**;   **public** ViewHolder(View v) {  **super**(v);  **TextViewHostelName** = (TextView) v.findViewById(R.id.***txt\_hostel\_name***);  **TextViewHostelAddr** = (TextView) v.findViewById(R.id.***txt\_hostel\_addr***);  **TextViewNumPeople** = (TextView) v.findViewById(R.id.***txt\_number\_people***);  **TextViewDays** = (TextView) v.findViewById(R.id.***txt\_days***);  **TextViewSkill** = (TextView) v.findViewById(R.id.***txt\_skill***);  **TextViewHostelStar** = (TextView) v.findViewById(R.id.***txt\_hostel\_star***);  **TextViewHostelPic** = (ImageView) v.findViewById(R.id.***img\_S\_pic***);  **ib\_popup\_menu** = (ImageButton) v.findViewById(R.id.***ib\_popup\_menu***);  **cardView** = (CardView) v.findViewById(R.id.***card\_view\_calender***);  }  }  **public** MyAdapter(List<String> data) {  **mData** = data;  }  @Override  **public** MyAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  **mContext** = parent.getContext();  View v = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.***findhostel\_item***, parent, **false**);  MyAdapter.ViewHolder vh = **new** MyAdapter.ViewHolder(v);  vh.**cardView**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Intent i = **new** Intent(**mContext**, Findhostel\_hostelinfo.**class**);  i.putExtra(**"hostelName"**,**mData**.get(vh.getAdapterPosition()));  i.putExtra(**"hostelAddress"**,**myDataset2**.get(vh.getAdapterPosition()));  i.putExtra(**"hostelPhoto"**,**myDataset3**.get(vh.getAdapterPosition()));  i.putExtra(**"vacancyName"**,**myDataset4**.get(vh.getAdapterPosition()));  i.putExtra(**"vacancySalary"**,**myDataset5**.get(vh.getAdapterPosition()));  i.putExtra(**"vacancyStartDate"**,**myDataset6**.get(vh.getAdapterPosition()));  i.putExtra(**"vacancyEndDate"**,**myDataset7**.get(vh.getAdapterPosition()));  i.putExtra(**"vacancyStartTime"**,**myDataset8**.get(vh.getAdapterPosition()));  i.putExtra(**"vacancEndTime"**,**myDataset9**.get(vh.getAdapterPosition()));  i.putExtra(**"vacancyNumPeople"**,**myDataset10**.get(vh.getAdapterPosition()));  i.putExtra(**"vacancyJob"**,**myDataset11**.get(vh.getAdapterPosition()));  i.putExtra(**"hostelOwnerName"**,**myDataset12**.get(vh.getAdapterPosition()));  i.putExtra(**"hostelOwnerAccount"**,**myDataset13**.get(vh.getAdapterPosition()));  i.putExtra(**"hostelOwnerPhone"**,**myDataset14**.get(vh.getAdapterPosition()));  i.putExtra(**"vacancDays"**,**myDataset15**.get(vh.getAdapterPosition()));  **mContext**.startActivity(i);  }  });  **return** vh;  }  @Override  **public void** onBindViewHolder(**final** MyAdapter.ViewHolder holder, **int** position) {  holder.**TextViewHostelName**.setText(**mData**.get(position));  holder.**TextViewHostelAddr**.setText(**myDataset2**.get(position));  holder.**TextViewNumPeople**.setText(**myDataset10**.get(position));  holder.**TextViewDays**.setText(**myDataset15**.get(position));  holder.**TextViewSkill**.setText(**myDataset11**.get(position));  holder.**TextViewHostelStar**.setText(**myDataset16**.get(position));  holder.**TextViewHostelPic**.setImageBitmap(stringToBitmap(**myDataset3**.get(position)));  holder.**ib\_popup\_menu**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  PopupMenu popupMenu = **new** PopupMenu(**mContext**,holder.**ib\_popup\_menu**);  popupMenu.inflate(R.menu.***findhostel\_popup\_menu***);  popupMenu.setOnMenuItemClickListener(**new** PopupMenu.OnMenuItemClickListener() {  @Override  **public boolean** onMenuItemClick(MenuItem item) {  **switch** (item.getItemId()){  **case** R.id.***menu\_collection***:  Toast.*makeText*(**mContext**,**"已收藏換宿"**,Toast.***LENGTH\_LONG***).show();  **break**;  **default**:  **break**;  }  **return false**;  }  });  popupMenu.show();  }  });  }  @Override  **public int** getItemCount() {  **return mData**.size();  }  }  **public** Bitmap stringToBitmap(String string) {  Bitmap bitmap = **null**;  **try** {  **byte**[] bitmapArray = Base64.*decode*(string, Base64.***DEFAULT***);  bitmap = BitmapFactory.*decodeByteArray*(bitmapArray, 0, bitmapArray.**length**);  **return** bitmap;  } **catch** (NullPointerException e) {  e.getMessage();  **return null**;  } **catch** (OutOfMemoryError e) {  **return null**;  }  } } | |

|  |  |
| --- | --- |
| 程式名稱 | Findhostel\_hostelinfo |
| 目的 | 顯示民宿詳細資料 |
| 程式碼 | |
| **package** ntub107202.student;  **public class** Findhostel\_hostelinfo **extends** AppCompatActivity {  **public** TextView **hostelName**, **hostelAddr**, **jobTitle**, **salary**, **startDate**,  **endDate**, **startTime**, **endTime**, **numberOfPeople**, **work**,  **contact**, **email**, **phone**;  **public** ImageView **imgHostelPic**;  **public** String **hostelNum**, **hostelOwnerAccount**;  **public** Button **btnResume**,**btn\_connection**;   **private** Context **mContext**;   *//參數* **private** ViewPager **mViewPager**;   *//假資料* **private** ImageView[] **imageViews** = **null**;  **private** ImageView **imageView** = **null**;  **private** ViewPager **advPager** = **null**;  **private** AtomicInteger **what** = **new** AtomicInteger(0);  **private boolean isContinue** = **true**;    **private** List<YoutubePlayerView> **playerViewList**;*//一个页面可以播放多个视频，将所有的播放控件收集到这里进行维护，主要是控制离开页面时候的暂停  //定位到youtube的某个视频有三种方式* **public static final** String ***VideoUrl\_normal*** = **""**;*//这种是最普通的写在地址栏中的视频地址* **public static final** String ***VideoUrl\_embed*** = **"https://www.youtube.com/embed/0xtcWek2tcM"**;*//这种是分享嵌入式的视频地址* **public static final** String ***VideoUrl\_short*** = **"https://youtu.be/wQ5Gj0UB\_R8"**;*//分享到facebook等社交平台的短url* **private** View **mVideoProgressView**;  **private** View **mCustomView**;*//全屏显示的View* **private** View **mVideoFullScreenBack**;  **private** LinearLayout **ll\_player\_container**;  **private int mOriginalSystemUiVisibility**;  **private int mOriginalOrientation**;  **private** WebChromeClient.CustomViewCallback **mCustomViewCallback**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***findhostel\_hostelinfo***);  initViewPager(); *// mContext = this; // mViewPager = (ViewPager) findViewById(R.id.view\_pager);  // //滑動圖片 // List<String> mUrlList = new ArrayList<>(); // mUrlList.add("http://mh.sfacg.com/Logo/201201/fe5df43e-f699-4d7b-85c0-3876dee217cb.jpg"); // mUrlList.add("http://cdn.makeuseof.com/wp-content/uploads/2017/05/android-apps-eat-battery-670x335.jpg"); // mUrlList.add("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSpSWJ32XK7TcLUT2XpXTrzwHMOiKHJkzD7QodGrhRzfvRn35Z4mg"); // mUrlList.add("http://android.suvenconsultants.com/newimage/android-developer2.png"); // mUrlList.add("http://android.suvenconsultants.com/newimage/android-developer2.png"); // // //呼叫Adapter // BannerAdapter bannerAdapter = new BannerAdapter(this, mUrlList); // mViewPager.setAdapter(bannerAdapter);* Bundle bundle=getIntent().getExtras();  **int** position=bundle.getInt(**"position"**);   String videoUrl = ***VideoUrl\_normal***;  videoUrl=getWorksheet.*getRow510*(position);  String videoID = YoutubePlayerView.*parseIDfromVideoUrl*(videoUrl);  Log.*i*(**"Alex"**,**"视频的ID是=="**+videoID);  View youtubeView = LayoutInflater.*from*(**this**).inflate(R.layout.***layout\_youtube\_player***, **null**);  YoutubePlayerView youtubePlayerView = (YoutubePlayerView) youtubeView.findViewById(R.id.***youtubePlayerView***);  youtubePlayerView.setAutoPlayerHeight(**this**);  youtubePlayerView.initialize(videoID, **new** YoutubePlayerCallBack(youtubePlayerView), **mWebChromeClient**);  **mVideoFullScreenBack** = findViewById(R.id.***detail\_video\_back***);  **if**(**playerViewList** == **null**){  **playerViewList** = **new** ArrayList<>();  }  **ll\_player\_container** = (LinearLayout) findViewById(R.id.***ll\_player\_container***);  **ll\_player\_container**.addView(youtubeView);  **playerViewList**.add(youtubePlayerView);     **btnResume** = findViewById(R.id.***btn\_resume***);  **btn\_connection** = findViewById(R.id.***btn\_connection***);   **hostelName** = (TextView) findViewById(R.id.***txt\_hostel\_name***);*//row19* **hostelAddr** = (TextView) findViewById(R.id.***txt\_hostel\_addr***);*//20* **jobTitle** = (TextView) findViewById(R.id.***txt\_job\_title***);*//22* **salary** = (TextView) findViewById(R.id.***txt\_salary***);*//23* **startDate** = (TextView) findViewById(R.id.***start\_date***);*//24* **endDate** = (TextView) findViewById(R.id.***end\_date***);*//25* **startTime** = (TextView) findViewById(R.id.***startTime***);*//26* **endTime** = (TextView) findViewById(R.id.***endTime***);*//27* **numberOfPeople** = (TextView) findViewById(R.id.***txt\_number\_people***);*//28* **work** = (TextView) findViewById(R.id.***txt\_work***);*//29* **contact** = (TextView) findViewById(R.id.***txt\_contact***);*//30* **email** = (TextView) findViewById(R.id.***txt\_email***);*//31* **phone** = (TextView) findViewById(R.id.***txt\_phone***);*//32 // imgHostelPic = (ImageView) findViewById(R.id.imageView\_hostel\_pic);//21* String user = getSharedPreferences(**"userpw"**, ***MODE\_PRIVATE***).getString(**"USER"**, **""**);   **btnResume**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  getWorksheet.*postToResume*(user, **hostelOwnerAccount**, **hostelNum**);  Log.*d*(**"get5487"**, **"我是User"** + user);  Log.*d*(**"get5487"**, **"fuck123fuck"** + getWorksheet.*getRow35*(position));  Log.*d*(**"get5487"**, **"fuck123fuck"** + getWorksheet.*getRow31*(position));  }  });  **btn\_connection**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Log.*v*(**"話都你在說"**,getWorksheet.*getRow46*(position));  Uri uri = Uri.*parse*(getWorksheet.*getRow46*(position));  Intent intent = **new** Intent(Intent.***ACTION\_VIEW***, uri);  startActivity(intent);  }  });   **hostelName**.setText(getWorksheet.*getRow19*(position));  **hostelAddr**.setText(getWorksheet.*getRow20*(position));  **jobTitle**.setText(getWorksheet.*getRow22*(position));  **salary**.setText(getWorksheet.*getRow23*(position));  **startDate**.setText(getWorksheet.*getRow24*(position));  **endDate**.setText(getWorksheet.*getRow25*(position));  **startTime**.setText(getWorksheet.*getRow26*(position));  **endTime**.setText(getWorksheet.*getRow27*(position));  **numberOfPeople**.setText(getWorksheet.*getRow28*(position));  **work**.setText(getWorksheet.*getRow29*(position));  **contact**.setText(getWorksheet.*getRow30*(position));  **email**.setText(getWorksheet.*getRow31*(position));  **phone**.setText(getWorksheet.*getRow32*(position)); *// imgHostelPic.setImageBitmap(stringToBitmap(getWorksheet.getRow21(position)));* **hostelNum** = getWorksheet.*getRow45*(position);  **hostelOwnerAccount** = getWorksheet.*getRow31*(position);   } *// // class BannerAdapter extends PagerAdapter { // // private LayoutInflater mInflater; // private List<String> mUrlList; // // //建構子 // public BannerAdapter(Context context, List<String> mUrlList) { // //如果把這個寫在instantiateItem的話，每一個Item都會呼叫一次，很吃資源 // mInflater = LayoutInflater.from(context); // this.mUrlList = mUrlList; // } // // //看你這ViewPager要有幾頁，靠著List的大小擴充 // @Override // public int getCount() { // //如果陣列為空，返回0 防呆機制 // return mUrlList == null ? 0 : mUrlList.size(); // } // // //用來判斷目前的畫面是否和instantiateItem創建的為同一個 // @Override // public boolean isViewFromObject(View view, Object o) { // return o == view; // } // // @Override // public Object instantiateItem(ViewGroup container, int position) { // //每一個Item都創建View // View view = mInflater.inflate(R.layout.main\_viewpager, container, false); // ImageView imageView = (ImageView) view.findViewById(R.id.image); // // //用Glide下載圖片 // Glide.with(mContext) // .load(mUrlList.get(position)) // .error(R.drawable.image\_loading) // .placeholder(R.drawable.image\_loading) // .centerCrop() // .fitCenter() // .diskCacheStrategy(DiskCacheStrategy.RESULT) // .into(imageView); // container.addView(view); // return view; // } // // //移除ViewPager內所對應的視圖 // @Override // public void destroyItem(ViewGroup container, int position, Object object) { // container.removeView((View) object); // } // }* **protected void** onStart() {  **super**.onStart();  }  **private class** YoutubePlayerCallBack **implements** YoutubePlayerView.YouTubeListener {   **private** YoutubePlayerView **mYoutubeView**;   YoutubePlayerCallBack(YoutubePlayerView view){  **this**.**mYoutubeView** = view;  }   @Override  **public void** onReady() {  }   @Override  **public void** onStateChange(YoutubePlayerView.STATE state) {  **if**(state == YoutubePlayerView.STATE.***PLAYING*** && **mYoutubeView**!=**null**){  **if**(**playerViewList**!=**null**){  **for**(YoutubePlayerView v : **playerViewList**){  **if** (v != **null** && v != **mYoutubeView** && (v.getPlayerState() == YoutubePlayerView.STATE.***PLAYING*** ||  v.getPlayerState() == YoutubePlayerView.STATE.***PAUSED***)) {  v.stop();  }  }  }  }  }   @Override  **public void** onPlaybackQualityChange(String arg) {  }   @Override  **public void** onPlaybackRateChange(String arg) {   }   @Override  **public void** onError(String arg) {  }   @Override  **public void** onApiChange(String arg) {  }   @Override  **public void** onCurrentSecond(**double** second) {  }   @Override  **public void** onDuration(**double** duration) {  }   @Override  **public void** logs(String log) {  }  }   */\*\*  \* 用于全屏显示的代码  \*/* **private** WebChromeClient **mWebChromeClient** = **new** WebChromeClient(){   @Override  **public** View getVideoLoadingProgressView() {  **if** (**mVideoProgressView** == **null**) {  LayoutInflater inflater = LayoutInflater.*from*(Findhostel\_hostelinfo.**this**);  **mVideoProgressView** = inflater.inflate(R.layout.***video\_layout\_loading***, **null**);  }  **return mVideoProgressView**;  }   @Override  **public void** onShowCustomView(View view, WebChromeClient.CustomViewCallback callback) {  *// if a view already exists then immediately terminate the new one* **if** (**mCustomView** != **null**) {  onHideCustomView();  **return**;  }   *// 1. Stash the current state* **mCustomView** = view;  **mOriginalSystemUiVisibility** = Findhostel\_hostelinfo.**this**.getWindow().getDecorView().getSystemUiVisibility();  **mOriginalOrientation** = Findhostel\_hostelinfo.**this**.getRequestedOrientation();  Log.*i*(**"Alex"**,**"原来的屏幕方向是"**+**mOriginalOrientation**);  *// 2. Stash the custom view callback* **mCustomViewCallback** = callback;   *// 3. Add the custom view to the view hierarchy* FrameLayout decor = (FrameLayout) Findhostel\_hostelinfo.**this**.getWindow().getDecorView();  decor.addView(**mCustomView**, **new** FrameLayout.LayoutParams(  ViewGroup.LayoutParams.***MATCH\_PARENT***,  ViewGroup.LayoutParams.***MATCH\_PARENT***));  **if**(**mVideoFullScreenBack**!=**null**){  **mVideoFullScreenBack**.setVisibility(View.***VISIBLE***);  }   *// 4. Change the state of the window* Findhostel\_hostelinfo.**this**.getWindow().getDecorView().setSystemUiVisibility(  View.***SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE*** |  View.***SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION*** |  View.***SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN*** |  View.***SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION*** |  View.***SYSTEM\_UI\_FLAG\_FULLSCREEN*** |  View.***SYSTEM\_UI\_FLAG\_IMMERSIVE***);  Findhostel\_hostelinfo.**this**.setRequestedOrientation(ActivityInfo.***SCREEN\_ORIENTATION\_LANDSCAPE***);  }   @Override  **public void** onHideCustomView() {  *// 1. Remove the custom view* FrameLayout decor = (FrameLayout) Findhostel\_hostelinfo.**this**.getWindow().getDecorView();  decor.removeView(**mCustomView**);  **mCustomView** = **null**;  **if**(**mVideoFullScreenBack**!=**null**){  **mVideoFullScreenBack**.setVisibility(View.***GONE***);  }   *// 2. Restore the state to it's original form* Findhostel\_hostelinfo.**this**.getWindow().getDecorView().setSystemUiVisibility(**mOriginalSystemUiVisibility**);  Findhostel\_hostelinfo.**this**.setRequestedOrientation(**mOriginalOrientation**);   *// 3. Call the custom view callback* **if**(**mCustomViewCallback**!=**null**){  **mCustomViewCallback**.onCustomViewHidden();  **mCustomViewCallback** = **null**;  }   }  };   @Override  **public void** onPause() {  *//视频播放器当页面停止的时候所有的视频播放全部暂停* **if**(**playerViewList**!=**null**){  **for**(YoutubePlayerView v : **playerViewList**){  **if**(v.getPlayerState() == YoutubePlayerView.STATE.***PLAYING*** ){  v.pause();  }**else if**(v.getPlayerState() == YoutubePlayerView.STATE.***BUFFERING***){  v.stop();  }  }  }  **super**.onPause();  }   @Override  **public void** onDestroy() {  **super**.onDestroy();  **if** (**playerViewList** != **null**) {  **for** (YoutubePlayerView v : **playerViewList**) {  **if** (v != **null**) v.onDestroy();  }  }  }  **public boolean** closeFullScreen(){  **if**(**mCustomView**!=**null** && **mCustomViewCallback**!=**null**){  **mWebChromeClient**.onHideCustomView();  **return true**;  }  **return false**;  }   @Override  **public void** onBackPressed() {  Log.*i*(**"Alex"**, **"进入onBackPressed方法"**);  closeFullScreen();  **super**.onBackPressed();  }   **public** Bitmap stringToBitmap(String string) {  Bitmap bitmap = **null**;  **try** {  **byte**[] bitmapArray = Base64.*decode*(string, Base64.***DEFAULT***);  bitmap = BitmapFactory.*decodeByteArray*(bitmapArray, 0, bitmapArray.**length**);  **return** bitmap;  } **catch** (NullPointerException e) {  e.getMessage();  **return null**;  } **catch** (OutOfMemoryError e) {  **return null**;  }  }   **private void** initViewPager() {  **advPager** = (ViewPager) findViewById(R.id.***adv\_pager***);  ViewGroup group = (ViewGroup) findViewById(R.id.***viewGroup***);  Bundle bundle=getIntent().getExtras();  **int** position=bundle.getInt(**"position"**);  *// 这里存放的是四张广告背景* List<View> advPics = **new** ArrayList<View>();   ImageView img1 = **new** ImageView(**this**);  img1.setImageBitmap(stringToBitmap(getWorksheet.*getRow21*(position)));  advPics.add(img1);   ImageView img2 = **new** ImageView(**this**);  img2.setImageBitmap(stringToBitmap(getWorksheet.*getRow500*(position)));  advPics.add(img2);   ImageView img3 = **new** ImageView(**this**);  img3.setImageBitmap(stringToBitmap(getWorksheet.*getRow47*(position)));  advPics.add(img3);   ImageView img4 = **new** ImageView(**this**);  img4.setImageBitmap(stringToBitmap(getWorksheet.*getRow48*(position)));  advPics.add(img4);   ImageView img5 = **new** ImageView(**this**);  img5.setImageBitmap(stringToBitmap(getWorksheet.*getRow49*(position)));  advPics.add(img5);  *// 对imageviews进行填充* **imageViews** = **new** ImageView[advPics.size()]; *//小图标* **for** (**int** i = 0; i < advPics.size(); i++) {  **imageView** = **new** ImageView(**this**);  **imageView**.setLayoutParams(**new** LayoutParams(20, 20));  **imageView**.setPadding(5, 5, 5, 5);  **imageViews**[i] = **imageView**;  **if** (i == 0) {  **imageViews**[i]  .setBackgroundResource(R.drawable.***logo1***);  } **else** {  **imageViews**[i]  .setBackgroundResource(R.drawable.***logo2***);  }  group.addView(**imageViews**[i]);  }   **advPager**.setAdapter(**new** AdvAdapter(advPics));  **advPager**.setOnPageChangeListener(**new** GuidePageChangeListener());  **advPager**.setOnTouchListener(**new** OnTouchListener() {   @Override  **public boolean** onTouch(View v, MotionEvent event) {  **switch** (event.getAction()) {  **case** MotionEvent.***ACTION\_DOWN***:  **case** MotionEvent.***ACTION\_MOVE***:  **isContinue** = **false**;  **break**;  **case** MotionEvent.***ACTION\_UP***:  **isContinue** = **true**;  **break**;  **default**:  **isContinue** = **true**;  **break**;  }  **return false**;  }  });  **new** Thread(**new** Runnable() {   @Override  **public void** run() {  **while** (**true**) {  **if** (**isContinue**) {  **viewHandler**.sendEmptyMessage(**what**.get());  whatOption();  }  }  }   }).start();  }   **private void** whatOption() {  **what**.incrementAndGet();  **if** (**what**.get() > **imageViews**.**length** - 1) {  **what**.getAndAdd(-4);  }  **try** {  Thread.*sleep*(5000);  } **catch** (InterruptedException e) {   }  }   **private final** Handler **viewHandler** = **new** Handler() {   @Override  **public void** handleMessage(Message msg) {  **advPager**.setCurrentItem(msg.**what**);  **super**.handleMessage(msg);  }   };   **private final class** GuidePageChangeListener **implements** OnPageChangeListener {   @Override  **public void** onPageScrollStateChanged(**int** arg0) {   }   @Override  **public void** onPageScrolled(**int** arg0, **float** arg1, **int** arg2) {   }   @Override  **public void** onPageSelected(**int** arg0) {  **what**.getAndSet(arg0);  **for** (**int** i = 0; i < **imageViews**.**length**; i++) {  **imageViews**[arg0]  .setBackgroundResource(R.drawable.***logo1***);  **if** (arg0 != i) {  **imageViews**[i]  .setBackgroundResource(R.drawable.***logo2***);  }  }   }   }   **private final class** AdvAdapter **extends** PagerAdapter {  **private** List<View> **views** = **null**;   **public** AdvAdapter(List<View> views) {  **this**.**views** = views;  }   @Override  **public void** destroyItem(View arg0, **int** arg1, Object arg2) {  ((ViewPager) arg0).removeView(**views**.get(arg1));  }   @Override  **public void** finishUpdate(View arg0) {   }   @Override  **public int** getCount() {  **return views**.size();  }   @Override  **public** Object instantiateItem(View arg0, **int** arg1) {  ((ViewPager) arg0).addView(**views**.get(arg1), 0);  **return views**.get(arg1);  }   @Override  **public boolean** isViewFromObject(View arg0, Object arg1) {  **return** arg0 == arg1;  }   @Override  **public void** restoreState(Parcelable arg0, ClassLoader arg1) {   }   @Override  **public** Parcelable saveState() {  **return null**;  }   @Override  **public void** startUpdate(View arg0) {   }   } } | |

|  |  |
| --- | --- |
| 程式名稱 | Fragment\_Forum |
| 目的 | 顯示換宿心得 |
| 程式碼 | |
| **public class** Fragment\_Forum **extends** Fragment{  **static** TextView *textView30*;  **static** TextView *textView31*;  **static** TextView *textView32*;  **static** TextView *textView33*;  **static** TextView *textView34*;  **static** String[] *row* = **new** String[100];  **static** String *row9*;  **static** String *row10*;  **static** String *row11*;  **static** String *row12*;  RecyclerView **mList**;  ArrayList<String> **myDataset**;  ArrayList<String> **myDataset2**;  ArrayList<String> **myDataset3**;  ArrayList<String> **myDataset4**;  ArrayList<String> **myDataset5**;  ArrayList<String> **myDataset6**;  MyAdapter **myAdapter**;  **static** LinearLayoutManager *layoutManager*;  @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {  View view=inflater.inflate(R.layout.***fragment\_forum***,container,**false**);  **mList** = (RecyclerView)view.findViewById(R.id.***list\_view***);   FloatingActionButton fab = (FloatingActionButton)view.findViewById(R.id.***fab\_go\_to\_forum\_add***) ;  fab.setOnClickListener(**new** Button.OnClickListener(){  @Override  **public void** onClick(View v) {  Intent intent = **new** Intent(getActivity(),Forum\_Add.**class**);  startActivity(intent);  }  });  **return** view;  }    **public void** onHiddenChanged(**boolean** hidden) {  **super**.onHiddenChanged(hidden);  **if**(hidden){  *//pause* }**else**{  **myDataset** = **new** ArrayList<>();  **myDataset2** = **new** ArrayList<>();  **myDataset3** = **new** ArrayList<>();  **myDataset4** = **new** ArrayList<>();  **myDataset5** = **new** ArrayList<>();  **myDataset6** = **new** ArrayList<>();  **myAdapter** = **new** MyAdapter(**myDataset**);  **for**(**int** i = 0; i < getWorksheet.*forumLength*; i++){ *// myDataset.add(i + "");* **myDataset**.add(getWorksheet.*getRow13*(i));  **myDataset2**.add(getWorksheet.*getRow14*(i));  **myDataset3**.add(getWorksheet.*getRow15*(i));  **myDataset4**.add(getWorksheet.*getRow16*(i));  **myDataset5**.add(getWorksheet.*getRow17*(i));  **myDataset6**.add(getWorksheet.*getRow18*(i));  } *// mList = (RecyclerView)view.findViewById(R.id.list\_view);  layoutManager* = **new** LinearLayoutManager(getActivity());  *layoutManager*.setOrientation(LinearLayoutManager.***VERTICAL***);  **mList**.setLayoutManager(*layoutManager*);  **mList**.setAdapter(**myAdapter**);  *//resume* }  }  **public class** MyAdapter **extends** RecyclerView.Adapter<MyAdapter.ViewHolder> {  **private** List<String> **mData**;  **private** Context **mContext**;   **public class** ViewHolder **extends** RecyclerView.ViewHolder {  **public** TextView **TextView0001**, **TextView0002** ,**TextView0003**, **TextView0004** ,**TextView0005**,**TextView33**;  **public** ImageView **TextView0006**;  **public** ImageButton **ib\_popup\_menu**;  **public** ViewHolder(View v) {  **super**(v);  **TextView0001** = (TextView) v.findViewById(R.id.***textView0001***);  **TextView0002** = (TextView) v.findViewById(R.id.***textView0002***);  **TextView0003** = (TextView) v.findViewById(R.id.***textView0003***);  **TextView0004** = (TextView) v.findViewById(R.id.***textView0004***);  **TextView0005** = (TextView) v.findViewById(R.id.***textView0005***);  **TextView0006** = (ImageView) v.findViewById(R.id.***image000006***);  **ib\_popup\_menu** = (ImageButton) v.findViewById(R.id.***ib\_popup\_menu***);  **TextView33** = (TextView) v.findViewById(R.id.***textView33***);   **TextView33**.setOnClickListener(**new** View.OnClickListener() {  Boolean **flag** = **true**;  @Override  **public void** onClick(View v) {  **if** (**flag**) {  **flag** = **false**;  **TextView0005**.setMaxLines(10);  **TextView0005**.setEllipsize(**null**); *// 展開* **TextView33**.setText(**"... 摺疊內容"**);  } **else** {  **flag** = **true**;  **TextView0005**.setLines(1);  **TextView0005**.setEllipsize(TextUtils.TruncateAt.***END***); *// 收縮* **TextView33**.setText(**"... 查看更多"**);  }  }  });  }  }  **public** MyAdapter(List<String> data) {  **mData** = data;  }  @Override  **public** MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {  **mContext** = parent.getContext();  View v = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.***forum\_item***, parent, **false**);  MyAdapter.ViewHolder vh = **new** MyAdapter.ViewHolder(v);  **return** vh;  }  @Override  **public void** onBindViewHolder(**final** MyAdapter.ViewHolder holder, **int** position) {  holder.**TextView0001**.setText(**mData**.get(position));  holder.**TextView0002**.setText(**myDataset2**.get(position));  holder.**TextView0003**.setText(**myDataset3**.get(position));  holder.**TextView0004**.setText(**myDataset4**.get(position));  holder.**TextView0005**.setText(**myDataset5**.get(position));  holder.**TextView0006**.setImageBitmap(stringToBitmap(**myDataset6**.get(position)));  holder.**ib\_popup\_menu**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  PopupMenu popupMenu = **new** PopupMenu(**mContext**,holder.**ib\_popup\_menu**);  popupMenu.inflate(R.menu.***forum\_popup\_menu***);  popupMenu.setOnMenuItemClickListener(**new** PopupMenu.OnMenuItemClickListener() {  @Override  **public boolean** onMenuItemClick(MenuItem item) {  **switch** (item.getItemId()){  **case** R.id.***menu\_edit***:  Toast.*makeText*(**mContext**,**"開始編輯貼文"**,Toast.***LENGTH\_LONG***).show();  Intent intent = **new** Intent(getActivity(),Forum\_Edit.**class**);  startActivity(intent);  **break**;  **case** R.id.***menu\_collection***:  Toast.*makeText*(**mContext**,**"已收藏貼文"**,Toast.***LENGTH\_LONG***).show();  **break**;  **case** R.id.***menu\_report***:  Toast.*makeText*(**mContext**,**"已檢舉貼文"**,Toast.***LENGTH\_LONG***).show();  intent = **new** Intent(getActivity(),Forum\_Report.**class**);  startActivity(intent);  **break**;  **default**:  **break**;  }  **return false**;  }  });  popupMenu.show();  }  });  }  @Override  **public int** getItemCount() {  **return mData**.size();  }  }  **public** Bitmap stringToBitmap(String string) {  Bitmap bitmap = **null**;  **try** {  **byte**[] bitmapArray = Base64.*decode*(string, Base64.***DEFAULT***);  bitmap = BitmapFactory.*decodeByteArray*(bitmapArray, 0, bitmapArray.**length**);  **return** bitmap;  } **catch** (NullPointerException e) {  e.getMessage();  **return null**;  } **catch** (OutOfMemoryError e) {  **return null**;  }  } } | |

|  |  |
| --- | --- |
| 程式名稱 | Fragment\_Schedule |
| 目的 | 顯示工作排程 |
| 程式碼 | |
| **public class** Fragment\_Schedule **extends** Fragment {  **private** DataView **dataView** ;  **private** TextView **info** ;  **static** TextView *textView30*;  ～  **static** TextView *textView34*;  **static** String[] *row* = **new** String[100];  **static** String *row9*;  ～  **static** String *row12*;  RecyclerView **mList**;  ArrayList<String> **myDataset**;  ～  ArrayList<String> **myDataset4**;  MyAdapter **myAdapter**;  **static** LinearLayoutManager *layoutManager*;  @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {  View view=inflater.inflate(R.layout.***fragment\_schedule***,container,**false**);  *textView30* = (TextView)view.findViewById(R.id.***text001***);  **mList** = (RecyclerView)view.findViewById(R.id.***list\_view***);  **if** (getWorksheet.*getRow10*(0) != **null**) {  *textView30*.setVisibility(View.***INVISIBLE***);  }  **info** = (TextView)view.findViewById(R.id.***info***);  **dataView** = (DataView)view.findViewById(R.id.***week***);  **dataView**.setOnSelectListener(**new** DataView.OnSelectListener() {  @Override  **public void** onSelected(DateEntity date) {  **info**.setText(**"日期："**+ date.**date**+**"\n"**+  **"周幾："**+ date.**weekName**+**"\n"**+  **"今日："**+ date.**isToday**+**"\n"**+  **"時間戳："**+ date.**million**+**"\n"**);  }  });  **dataView**.getData(**"2018-10-1"**);  **return** view;  }   **public void** onHiddenChanged(**boolean** hidden) {  **super**.onHiddenChanged(hidden);  **if**(hidden){}**else**{  **myDataset** = **new** ArrayList<>();  ～  **myDataset4** = **new** ArrayList<>();  **myAdapter** = **new** MyAdapter(**myDataset**);  **for**(**int** i = 0; i < getWorksheet.*scheduleLength*; i++){ **myDataset**.add(getWorksheet.*getRow9*(i));  ～  **myDataset4**.add(getWorksheet.*getRow12*(i));  }  *layoutManager* = **new** LinearLayoutManager(getActivity());  *layoutManager*.setOrientation(LinearLayoutManager.***VERTICAL***);  **mList**.setLayoutManager(*layoutManager*);  **mList**.setAdapter(**myAdapter**);  *//resume* }  }  **public class** MyAdapter **extends** RecyclerView.Adapter<MyAdapter.ViewHolder> {  **private** List<String> **mData**;  **public class** ViewHolder **extends** RecyclerView.ViewHolder {  **public** TextView **TextView0001**, **TextView0002** ,**TextView0003**, **TextView0004**,**TextView33**;  @RequiresApi(api = Build.VERSION\_CODES.***JELLY\_BEAN***)  **public** ViewHolder(View v) {  **super**(v);  **TextView0001** = (TextView) v.findViewById(R.id.***textView0001***);  ～  **TextView0004** = (TextView) v.findViewById(R.id.***textView0004***);  **TextView33** = (TextView) v.findViewById(R.id.***textView33***);  **TextView33**.setOnClickListener(**new** View.OnClickListener() {  Boolean **flag** = **true**;  @Override  **public void** onClick(View v) {  **if** (**flag**) {  **flag** = **false**;  **TextView0002**.setMaxLines(5);  **TextView0002**.setEllipsize(**null**); *// 展開*  } **else** {  **flag** = **true**;  **TextView0002**.setLines(1);  **TextView0002**.setEllipsize(TextUtils.TruncateAt.***END***); *// 收縮*}}});  }  }  **public** MyAdapter(List<String> data) {  **mData** = data;  }  @RequiresApi(api = Build.VERSION\_CODES.***JELLY\_BEAN***)  @Override  **public** MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {  View v = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.***schedule\_item***, parent, **false**);  ViewHolder vh = **new** ViewHolder(v);  **return** vh;  }  @Override  **public void** onBindViewHolder(ViewHolder holder, **int** position) {  holder.**TextView0001**.setText(**mData**.get(position));  ～  holder.**TextView0004**.setText(**myDataset4**.get(position));  }  @Override  **public int** getItemCount() {  **return mData**.size();  }  } } | |

|  |  |
| --- | --- |
| 程式名稱 | Forum\_Add |
| 目的 | 新增心得 |
| 程式碼 | |
| **public class** Forum\_Add **extends** AppCompatActivity {  **private** Button **startCameraButton** = **null**;  **private** Button **choiceFromAlbumButton** = **null**;  **private** ImageView **pictureImageView** = **null**;  **private static final int *TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE*** = 0; *// 拍照的許可權處理返回碼* **private static final int *WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE*** = 1; *// 讀儲存卡內容的許可處理* **private static final int *TAKE\_PHOTO\_REQUEST\_CODE*** = 3; *// 拍照返回的 requestCode* **private static final int *CHOICE\_FROM\_ALBUM\_REQUEST\_CODE*** = 4; *// 相冊選取返回的 requestCode* **private static final int *CROP\_PHOTO\_REQUEST\_CODE*** = 5; *// 裁剪圖片返回的 requestCode* **private** Uri **photoUri** = **null**;  **private** Uri **photoOutputUri** = **null**; *// 圖片最終的輸出檔的 Uri* @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***forum\_add***);  **startCameraButton** = (Button) findViewById(R.id.***button8***);  **startCameraButton**.setOnClickListener(**clickListener**);  **choiceFromAlbumButton** = (Button) findViewById(R.id.***button7***);  **choiceFromAlbumButton**.setOnClickListener(**clickListener**);  **pictureImageView** = (ImageView) findViewById(R.id.***imageView10***);  Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  setSupportActionBar(toolbar);  getSupportActionBar().setDisplayShowTitleEnabled(**false**);  */\*  \* 先判斷使用者以前有沒有對我們的應用程式允許過讀寫記憶體卡內容的許可權，  \* 用戶處理的結果在 onRequestPermissionResult 中進行處理  \*/* **if**(ContextCompat.*checkSelfPermission*(Forum\_Add.**this**, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***)  != PackageManager.***PERMISSION\_GRANTED***) {  *// 申請讀寫記憶體卡內容的許可權* ActivityCompat.*requestPermissions*(Forum\_Add.**this**,  **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***}, ***WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE***);  }  }  **private** View.OnClickListener **clickListener** = **new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *// 調用相機拍照* **if**(v == **startCameraButton**) {  *// 同上面的許可權申請邏輯* **if**(ContextCompat.*checkSelfPermission*(Forum\_Add.**this**, Manifest.permission.***CAMERA***)  != PackageManager.***PERMISSION\_GRANTED***) {  */\*  \* 下面是對調用相機拍照許可權進行申請  \*/* ActivityCompat.*requestPermissions*(Forum\_Add.**this**,  **new** String[]{Manifest.permission.***CAMERA***,}, ***TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE***);  } **else** {  startCamera();  }  *// 從相冊獲取* } **else if**(v == **choiceFromAlbumButton**) {  choiceFromAlbum();  }  }  };  *\* 拍照* **private void** startCamera() { *//\* 設置拍照得到的照片的儲存目錄，因為我們訪問應用的緩存路徑並不需要讀寫記憶體卡的申請許可權  //\* 因此，這裡為了方便，將拍照得到的照片存在這個緩存目錄中* File file = **new** File(getExternalCacheDir(), **"image.jpg"**);  **try** {  **if**(file.exists()) {  file.delete();  }  file.createNewFile();  } **catch** (IOException e) {  e.printStackTrace();  }  */\*\*  \* 因 Android 7.0 開始，不能使用 file:// 類型的 Uri 訪問跨應用文件，否則報異常，  \* 因此我們這裡需要使用內容提供器，FileProvider 是 ContentProvider 的一個子類，  \* 我們可以輕鬆的使用 FileProvider 來在不同程式之間分享資料(相對於 ContentProvider 來說)  \*/* **if**(Build.VERSION.***SDK\_INT*** >= 24) {  **photoUri** = FileProvider.*getUriForFile*(**this**, **"com.zhi\_dian.provider"**, file);  } **else** {  **photoUri** = Uri.*fromFile*(file); *// Android 7.0 以前使用原來的方法來獲取檔的 Uri* }  *// 打開系統相機的 Action，等同於："android.media.action.IMAGE\_CAPTURE"* Intent takePhotoIntent = **new** Intent(MediaStore.***ACTION\_IMAGE\_CAPTURE***);  *// 設置拍照所得照片的輸出目錄* takePhotoIntent.putExtra(MediaStore.***EXTRA\_OUTPUT***, **photoUri**);  startActivityForResult(takePhotoIntent, ***TAKE\_PHOTO\_REQUEST\_CODE***);  } **private void** choiceFromAlbum() {  *// 打開系統圖庫的 Action，等同於: "android.intent.action.GET\_CONTENT"* Intent choiceFromAlbumIntent = **new** Intent(Intent.***ACTION\_GET\_CONTENT***);  *// 設置資料類型為圖片類型* choiceFromAlbumIntent.setType(**"image/\*"**);  startActivityForResult(choiceFromAlbumIntent, ***CHOICE\_FROM\_ALBUM\_REQUEST\_CODE***);  } **private void** cropPhoto(Uri inputUri) {  *// 調用系統裁剪圖片的 Action* Intent cropPhotoIntent = **new** Intent(**"com.android.camera.action.CROP"**);  *// 設置資料Uri 和類型* cropPhotoIntent.setDataAndType(inputUri, **"image/\*"**);  *// 授權應用讀取 Uri，這一步要有，不然裁剪程式會崩潰* cropPhotoIntent.addFlags(Intent.***FLAG\_GRANT\_READ\_URI\_PERMISSION***);  *// 設置圖片的最終輸出目錄* cropPhotoIntent.putExtra(MediaStore.***EXTRA\_OUTPUT***,  **photoOutputUri** = Uri.*parse*(**"file:////sdcard/image\_output.jpg"**));  startActivityForResult(cropPhotoIntent, ***CROP\_PHOTO\_REQUEST\_CODE***);  } @Override  **public void** onRequestPermissionsResult(**int** requestCode, @NonNull String[] permissions, @NonNull **int**[] grantResults) {  **switch** (requestCode) {  *// 調用相機拍照：* **case *TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE***:  *// 如果用戶授予許可權，那麼打開相機拍照* **if**(grantResults.**length** > 0 && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {  startCamera();  } **else** {  Toast.*makeText*(**this**, **"拍照許可權被拒絕"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  *// 打開相冊選取：* **case *WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE***:  **if**(grantResults.**length** > 0 && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {  } **else** {  Toast.*makeText*(**this**, **"讀寫記憶體卡內容許可權被拒絕"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  }  } @Override  **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  **if**(resultCode == ***RESULT\_OK***) {  *// 通過返回碼判斷是哪個應用返回的資料* **switch** (requestCode) {  *// 拍照* **case *TAKE\_PHOTO\_REQUEST\_CODE***:  cropPhoto(**photoUri**);  **break**;  *// 相冊選擇* **case *CHOICE\_FROM\_ALBUM\_REQUEST\_CODE***:  cropPhoto(data.getData());  **break**;  *// 裁剪圖片* **case *CROP\_PHOTO\_REQUEST\_CODE***:  File file = **new** File(**photoOutputUri**.getPath());  **if**(file.exists()) {  Bitmap bitmap = BitmapFactory.*decodeFile*(**photoOutputUri**.getPath());  **pictureImageView**.setImageBitmap(bitmap); *// file.delete(); // 選取完後刪除照片* } **else** {  Toast.*makeText*(**this**, **"找不到照片"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  }  }  } } | |

|  |  |
| --- | --- |
| 程式名稱 | Forum\_Edit |
| 目的 | 編輯心得 |
| 程式碼 | |
| **public class** Forum\_Edit **extends** AppCompatActivity {  **private** Button **startCameraButton** = **null**;  **private** Button **choiceFromAlbumButton** = **null**;  **private** ImageView **pictureImageView** = **null**;   **private static final int *TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE*** = 0; *// 拍照的許可權處理返回碼* **private static final int *WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE*** = 1; *// 讀儲存卡內容的許可處理* **private static final int *TAKE\_PHOTO\_REQUEST\_CODE*** = 3; *// 拍照返回的 requestCode* **private static final int *CHOICE\_FROM\_ALBUM\_REQUEST\_CODE*** = 4; *// 相冊選取返回的 requestCode* **private static final int *CROP\_PHOTO\_REQUEST\_CODE*** = 5; *// 裁剪圖片返回的 requestCode* **private** Uri **photoUri** = **null**;  **private** Uri **photoOutputUri** = **null**; *// 圖片最終的輸出檔的 Uri* @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***forum\_edit***);  **startCameraButton** = (Button) findViewById(R.id.***button8***);  **startCameraButton**.setOnClickListener(**clickListener**);  **choiceFromAlbumButton** = (Button) findViewById(R.id.***button7***);  **choiceFromAlbumButton**.setOnClickListener(**clickListener**);  **pictureImageView** = (ImageView) findViewById(R.id.***imageView10***);   Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  setSupportActionBar(toolbar);  getSupportActionBar().setDisplayShowTitleEnabled(**false**); **if**(ContextCompat.*checkSelfPermission*(Forum\_Edit.**this**, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***)  != PackageManager.***PERMISSION\_GRANTED***) {  *// 申請讀寫記憶體卡內容的許可權* ActivityCompat.*requestPermissions*(Forum\_Edit.**this**,  **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***}, ***WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE***);  }  }  **private** View.OnClickListener **clickListener** = **new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *// 調用相機拍照* **if**(v == **startCameraButton**) {  *// 同上面的許可權申請邏輯* **if**(ContextCompat.*checkSelfPermission*(Forum\_Edit.**this**, Manifest.permission.***CAMERA***)  != PackageManager.***PERMISSION\_GRANTED***) {  */\*  \* 下面是對調用相機拍照許可權進行申請  \*/* ActivityCompat.*requestPermissions*(Forum\_Edit.**this**,  **new** String[]{Manifest.permission.***CAMERA***,}, ***TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE***);  } **else** {  startCamera();  }  *// 從相冊獲取* } **else if**(v == **choiceFromAlbumButton**) {  choiceFromAlbum();  }  }  }; **private void** startCamera() {File file = **new** File(getExternalCacheDir(), **"image.jpg"**);  **try** {  **if**(file.exists()) {  file.delete();  }  file.createNewFile();  } **catch** (IOException e) {  e.printStackTrace();  } **if**(Build.VERSION.***SDK\_INT*** >= 24) {  **photoUri** = FileProvider.*getUriForFile*(**this**, **"com.zhi\_dian.provider"**, file);  } **else** {  **photoUri** = Uri.*fromFile*(file); *// Android 7.0 以前使用原來的方法來獲取檔的 Uri* }  *// 打開系統相機的 Action，等同於："android.media.action.IMAGE\_CAPTURE"* Intent takePhotoIntent = **new** Intent(MediaStore.***ACTION\_IMAGE\_CAPTURE***);  *// 設置拍照所得照片的輸出目錄* takePhotoIntent.putExtra(MediaStore.***EXTRA\_OUTPUT***, **photoUri**);  startActivityForResult(takePhotoIntent, ***TAKE\_PHOTO\_REQUEST\_CODE***);  }   */\*\*  \* 從相冊選取  \*/* **private void** choiceFromAlbum() {  *// 打開系統圖庫的 Action，等同於: "android.intent.action.GET\_CONTENT"* Intent choiceFromAlbumIntent = **new** Intent(Intent.***ACTION\_GET\_CONTENT***);  *// 設置資料類型為圖片類型* choiceFromAlbumIntent.setType(**"image/\*"**);  startActivityForResult(choiceFromAlbumIntent, ***CHOICE\_FROM\_ALBUM\_REQUEST\_CODE***);  } **private void** cropPhoto(Uri inputUri) {  *// 調用系統裁剪圖片的 Action* Intent cropPhotoIntent = **new** Intent(**"com.android.camera.action.CROP"**);  *// 設置資料Uri 和類型* cropPhotoIntent.setDataAndType(inputUri, **"image/\*"**);  cropPhotoIntent.putExtra(**"crop"**,**"true"**);  *// 裁剪框的比例，1：1* cropPhotoIntent.putExtra(**"aspectX"**,1);  cropPhotoIntent.putExtra(**"aspectY"**,1);  *// 裁剪後輸出圖片的尺寸大小* cropPhotoIntent.putExtra(**"outputX"**,1);  cropPhotoIntent.putExtra(**"outputY"**,1);  *// 授權應用讀取 Uri，這一步要有，不然裁剪程式會崩潰* cropPhotoIntent.addFlags(Intent.***FLAG\_GRANT\_READ\_URI\_PERMISSION***);  *// 設置圖片的最終輸出目錄* cropPhotoIntent.putExtra(MediaStore.***EXTRA\_OUTPUT***,  **photoOutputUri** = Uri.*parse*(**"file:////sdcard/image\_output.jpg"**));  startActivityForResult(cropPhotoIntent, ***CROP\_PHOTO\_REQUEST\_CODE***);  } @Override  **public void** onRequestPermissionsResult(**int** requestCode, @NonNull String[] permissions, @NonNull **int**[] grantResults) {  **switch** (requestCode) {  *// 調用相機拍照：* **case *TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE***:  *// 如果用戶授予許可權，那麼打開相機拍照* **if**(grantResults.**length** > 0 && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {  startCamera();  } **else** {  Toast.*makeText*(**this**, **"拍照許可權被拒絕"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  *// 打開相冊選取：* **case *WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE***:  **if**(grantResults.**length** > 0 && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {  } **else** {  Toast.*makeText*(**this**, **"讀寫記憶體卡內容許可權被拒絕"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  }  }  @Override  **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  **if**(resultCode == ***RESULT\_OK***) {  *// 通過返回碼判斷是哪個應用返回的資料* **switch** (requestCode) {  *// 拍照* **case *TAKE\_PHOTO\_REQUEST\_CODE***:  cropPhoto(**photoUri**);  **break**;  *// 相冊選擇* **case *CHOICE\_FROM\_ALBUM\_REQUEST\_CODE***:  cropPhoto(data.getData());  **break**;  *// 裁剪圖片* **case *CROP\_PHOTO\_REQUEST\_CODE***:  File file = **new** File(**photoOutputUri**.getPath());  **if**(file.exists()) {  Bitmap bitmap = BitmapFactory.*decodeFile*(**photoOutputUri**.getPath());  **pictureImageView**.setImageBitmap(bitmap); *// file.delete(); // 選取完後刪除照片* } **else** {  Toast.*makeText*(**this**, **"找不到照片"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  }  }  } } | |

|  |  |
| --- | --- |
| 程式名稱 | Personal\_Resume |
| 目的 | 編輯履歷功能 |
| 程式碼 | |
| **public class** Personal\_Resume **extends** AppCompatActivity {  **private** Button **startCameraButton** = **null**;  **private** Button **choiceFromAlbumButton** = **null**;  **private** ImageView **pictureImageView** = **null**;  **private static final int *TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE*** = 0; *//拍照的許可權處理返回碼* **private static final int *WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE*** = 1;*//讀儲存卡內容的許可處理* **private static final int *TAKE\_PHOTO\_REQUEST\_CODE*** = 3; *// 拍照返回的 requestCode* **private static final int *CHOICE\_FROM\_ALBUM\_REQUEST\_CODE*** = 4; *// 相冊選取返回的 requestCode* **private static final int *CROP\_PHOTO\_REQUEST\_CODE*** = 5; *// 裁剪圖片返回的 requestCode* **private** Uri **photoUri** = **null**;  **private** Uri **photoOutputUri** = **null**; *// 圖片最終的輸出檔的 Uri* @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***personal\_resume***);  **startCameraButton** = (Button) findViewById(R.id.***button8***);  **startCameraButton**.setOnClickListener(**clickListener**);  **choiceFromAlbumButton** = (Button) findViewById(R.id.***button7***);  **choiceFromAlbumButton**.setOnClickListener(**clickListener**);  **pictureImageView** = (ImageView) findViewById(R.id.***imageView10***);  Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  setSupportActionBar(toolbar);  getSupportActionBar().setDisplayShowTitleEnabled(**false**);**if**(ContextCompat.*checkSelfPermission*(Personal\_Resume.**this**, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***)  != PackageManager.***PERMISSION\_GRANTED***) {  *// 申請讀寫記憶體卡內容的許可權* ActivityCompat.*requestPermissions*(Personal\_Resume.**this**,  **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***}, ***WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE***);  }  }  **private** View.OnClickListener **clickListener** = **new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *// 調用相機拍照* **if**(v == **startCameraButton**) {  *// 同上面的許可權申請邏輯* **if**(ContextCompat.*checkSelfPermission*(Personal\_Resume.**this**, Manifest.permission.***CAMERA***)  != PackageManager.***PERMISSION\_GRANTED***) {  */\*  \* 下面是對調用相機拍照許可權進行申請  \*/* ActivityCompat.*requestPermissions*(Personal\_Resume.**this**,  **new** String[]{Manifest.permission.***CAMERA***,}, ***TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE***);  } **else** {  startCamera();  }  *// 從相冊獲取* } **else if**(v == **choiceFromAlbumButton**) {  choiceFromAlbum();  }  }  };   */\*\*  \* 拍照  \*/* **private void** startCamera() {  */\*\*  \* 設置拍照得到的照片的儲存目錄，因為我們訪問應用的緩存路徑並不需要讀寫記憶體卡的申請許可權，  \* 因此，這裡為了方便，將拍照得到的照片存在這個緩存目錄中  \*/* File file = **new** File(getExternalCacheDir(), **"image.jpg"**);  **try** {  **if**(file.exists()) {  file.delete();  }  file.createNewFile();  } **catch** (IOException e) {  e.printStackTrace();  }**if**(Build.VERSION.***SDK\_INT*** >= 24) {  **photoUri** = FileProvider.*getUriForFile*(**this**, **"com.zhi\_dian.provider"**, file);  } **else** {  **photoUri** = Uri.*fromFile*(file); *// Android 7.0 以前使用原來的方法來獲取檔的 Uri* }  *// 打開系統相機的 Action，等同於："android.media.action.IMAGE\_CAPTURE"* Intent takePhotoIntent = **new** Intent(MediaStore.***ACTION\_IMAGE\_CAPTURE***);  *// 設置拍照所得照片的輸出目錄* takePhotoIntent.putExtra(MediaStore.***EXTRA\_OUTPUT***, **photoUri**);  startActivityForResult(takePhotoIntent, ***TAKE\_PHOTO\_REQUEST\_CODE***);  } **private void** choiceFromAlbum() {  *// 打開系統圖庫的 Action，等同於: "android.intent.action.GET\_CONTENT"* Intent choiceFromAlbumIntent = **new** Intent(Intent.***ACTION\_GET\_CONTENT***);  *// 設置資料類型為圖片類型* choiceFromAlbumIntent.setType(**"image/\*"**);  startActivityForResult(choiceFromAlbumIntent, ***CHOICE\_FROM\_ALBUM\_REQUEST\_CODE***);  }**private void** cropPhoto(Uri inputUri) {  *// 調用系統裁剪圖片的 Action* Intent cropPhotoIntent = **new** Intent(**"com.android.camera.action.CROP"**);  *// 設置資料Uri 和類型* cropPhotoIntent.setDataAndType(inputUri, **"image/\*"**);  cropPhotoIntent.putExtra(**"crop"**,**"true"**);  *// 裁剪框的比例，1：1* cropPhotoIntent.putExtra(**"aspectX"**,1);  cropPhotoIntent.putExtra(**"aspectY"**,1);  *// 裁剪後輸出圖片的尺寸大小* cropPhotoIntent.putExtra(**"outputX"**,500);  cropPhotoIntent.putExtra(**"outputY"**,500);  *// 授權應用讀取 Uri，這一步要有，不然裁剪程式會崩潰* cropPhotoIntent.addFlags(Intent.***FLAG\_GRANT\_READ\_URI\_PERMISSION***);  *// 設置圖片的最終輸出目錄* cropPhotoIntent.putExtra(MediaStore.***EXTRA\_OUTPUT***,  **photoOutputUri** = Uri.*parse*(**"file:////sdcard/image\_output.jpg"**));  startActivityForResult(cropPhotoIntent, ***CROP\_PHOTO\_REQUEST\_CODE***);  } @Override  **public void** onRequestPermissionsResult(**int** requestCode, @NonNull String[] permissions, @NonNull **int**[] grantResults) {  **switch** (requestCode) {  *// 調用相機拍照：* **case *TAKE\_PHOTO\_PERMISSION\_REQUEST\_CODE***:  *// 如果用戶授予許可權，那麼打開相機拍照* **if**(grantResults.**length** > 0 && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {  startCamera();  } **else** {  Toast.*makeText*(**this**, **"拍照許可權被拒絕"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  *// 打開相冊選取：* **case *WRITE\_SDCARD\_PERMISSION\_REQUEST\_CODE***:  **if**(grantResults.**length** > 0 && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {  } **else** {  Toast.*makeText*(**this**, **"讀寫記憶體卡內容許可權被拒絕"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  }  }@Override  **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  **if**(resultCode == ***RESULT\_OK***) {  *// 通過返回碼判斷是哪個應用返回的資料* **switch** (requestCode) {  *// 拍照* **case *TAKE\_PHOTO\_REQUEST\_CODE***:  cropPhoto(**photoUri**);  **break**;  *// 相冊選擇* **case *CHOICE\_FROM\_ALBUM\_REQUEST\_CODE***:  cropPhoto(data.getData());  **break**;  *// 裁剪圖片* **case *CROP\_PHOTO\_REQUEST\_CODE***:  File file = **new** File(**photoOutputUri**.getPath());  **if**(file.exists()) {  Bitmap bitmap = BitmapFactory.*decodeFile*(**photoOutputUri**.getPath());  **pictureImageView**.setImageBitmap(bitmap); *// file.delete(); // 選取完後刪除照片* } **else** {  Toast.*makeText*(**this**, **"找不到照片"**, Toast.***LENGTH\_SHORT***).show();  }  **break**;  }  }  } } | |

後端資料庫相關程式碼的部分礙於手冊頁數太多，我們團隊不列入系統手冊內容中，若想知道詳細程式碼，請至下方網址查看：

<https://github.com/ntub107202/107202_PHP.git>

1. 測試模型
   1. 測試計畫

我們以使用者的角度測試一家順順的各個功能是否能正常運作，雇主端的測試結果如表10-2-1，學生端的測試結果如表10-2-2。

* 雇主端

1. 註冊會員：確認民宿主註冊會員是否成功。
2. 登入會員：確認民宿主登入會員是否成功。
3. 新增職缺：確認民宿主登入會員後是否可以新增職缺。
4. 新增工作：確認民宿主會員後是否可以新增工作。
5. 新增民宿：確認民宿主會員後是否可以新增民宿。
6. 回報問題：確認民宿主在發現問題是否可以回報至後台。

* 學生端

1. 註冊會員：確認學生註冊會員是否成功。
2. 登入會員：確認學生登入會員是否成功。
3. 新增換宿心得：確認學生登入會員後是否可以新增換宿心得。
   1. 測試個案與測試結果資料

▼表 10-2-1 雇主端-註冊會員測試結果

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 註冊會員 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
| C:\Users\carbo\Downloads\44211056_552543738528174_142007346975997952_n.png | | C:\Users\carbo\Downloads\44269260_2040638772700381_3661283088094199808_n.png | |

▼表 10-2-2 雇主端-登入會員測試結果

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 登入會員 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表 10-2-3 雇主端-新增職缺測試結果

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 新增職缺 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表 10-2-4 雇主端-新增工作測試結果

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 新增工作 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表 10-2-5 雇主端-新增民宿測試結果

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 新增民宿 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表 10-2-6 雇主端-回報問題測試結果

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 回報問題 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表 10-2-7 學生端-註冊會員測試結果

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 註冊會員 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表 10-2-8 學生端-登入會員測試結果

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 登入會員 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表 10-2-9 學生端-新增換宿心得測試結果

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 新增換宿心得 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

1. 操作手冊

以下是一家順順之雇主端及學生端系統安裝之步驟，一家順順的操作手冊說明如表 11-1-1。

▼表 11-1-1 雇主端操作手冊

|  |  |
| --- | --- |
| 步驟1. 先點選設定，選擇安全性。 | 步驟2. 將未知來源的選項勾選。 |
|  |  |
| 步驟3. 跳出警告視窗，按下確定。 | 步驟4. 至我的檔案中，選擇安裝APK。 |
|  |  |
| 步驟5. 點選APK檔後，選擇安裝。 | 步驟6. 安裝好後，點擊完成或開啟。 |
|  |  |

1. 使用手冊

一家順順的使用方式十分簡單，以下是雇主端及學生端所有功能的操作步驟介面圖及步驟說明，相信能讓使用者更清楚了解雇主端和學生端的使用方式，雇主端的使用步驟說明如表12-1-1，學生端使用步驟說明如表12-1-2。

▼表 12-1-1 雇主端使用步驟說明

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | | 會員登入、註冊 | |
| 步驟1. 進入後輸入帳號與密碼登入。 | | | 步驟2. 選擇註冊會先看到小提醒。 |
|  | | |  |
| 步驟3. 選擇民宿註冊後，填寫資料。 | | | 步驟4. 若是不符合格式會跳出小提醒。 |
|  | | |  |
| 功能名稱 | | 人才搜尋 | |
| 步驟1. 進入主畫面後，可以查看人才資訊列表。 | | | 步驟2. 選擇有興趣的人才即可了解詳細資訊。 |
|  | | |  |
| 步驟3. 滑至底部，可以傳遞訊息聯絡，或是傳送雇用邀請。 | | | 步驟4. 點擊聯絡他會以Line的方式傳送訊息。 |
|  | | |  |
| 功能名稱 | 刊登職缺 | | |
| 步驟1. 查看刊登過的職缺列表。 | | | 步驟2. 按右下角按鈕，進行新增職缺。 |
|  | | |  |
| 步驟3. 填寫完畢後，即可新增。 | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| 功能名稱 | 工作安排 | |
| 步驟1. 查看已新增的工作安排列表。 | | 步驟2. 點擊項目，即可展開查看詳細內容。 |
|  | |  |
| 步驟3. 點選右下角按鈕即可新增工作。 | | 步驟4. 回到列表即可看到新增的工作。 |
|  | |  |
| 功能名稱 | 雇主頁面 | |
| 步驟1. 選擇最右方的雇主頁面。 | | 步驟2. 點選民宿資訊，可以查看自己的民宿列表。 |
|  | |  |
| 步驟3. 點選右下角按鈕即可新增民宿。 | | 步驟4. 填寫完即可新增。 |
|  | |  |
| 步驟5. 在通知設定，可開關系統通知。 | | 步驟6. 使用者可以在回報問題，將系統的問題傳送給製作團隊。 |
|  | |  |
| 步驟3. 關於我們可查看製作團隊簡介。 | | 步驟4. 相關規範查看系統之使用規範。 |
|  | |  |

▼表 12-1-2學生端使用步驟說明

|  |  |  |
| --- | --- | --- |
| 功能名稱 | 註冊、登入功能 | |
| 步驟1. 點擊註冊並輸入資料 | | 步驟2. 點擊創建學生帳戶 |
|  | |  |
| 步驟3. 進入後輸入帳號與密碼登入 | | 步驟4. 登入後進入搜尋民宿畫面 |
|  | |  |

|  |  |  |
| --- | --- | --- |
| 功能名稱 | 新增交換心得 | |
| 步驟1. 查看交換心得頁面。 | | 步驟2. 點擊查看更多，顯示詳細內容。 |
|  | |  |
| 步驟3. 點擊右下角按鈕即可新增心得。 | |
|  | |

|  |  |  |
| --- | --- | --- |
| 功能名稱 | 查看工作排程 | |
| 步驟1. 查看交換心得頁面。 | | 步驟2. 點擊查看更多，顯示詳細內容。 |
|  | |  |
| 步驟3. 點擊行事曆選取日期，查看當日工作時間與內容。 | |
|  | |

|  |  |  |
| --- | --- | --- |
| 功能名稱 | 編輯個人履歷 | |
| 步驟1. 查看右方個人頁面。 | | 步驟2. 點擊上方編輯履歷 |
|  | |  |
| 步驟3. 勾選技能並輸入資料並且點擊儲存即可上傳履歷。 | |
|  | |

1. 感想

10636008 游期順

加上上次的五專專題，這是我人生中的第二個專題，也因為有了五專的經驗，已經對於流程以及作業方式有所了解，讓我能夠很順利的完成。而專題畢竟是團隊合作，且這次我是扮演組長的角色，說到團隊合作，必定有分工，有意見上的分歧，也難免會吵架。這時組長就要帶領大家來溝通，而剛開始我覺得溝通真的是一件很難的是，但慢慢地一次兩次，對於溝通這件事就漸入佳境了，

而這我覺得才是我在這次專題中學到最多的，也很感謝我的組員願意跟我溝通，讓我們可以一起完成專題。

10636011 劉家驤

從一開始完全不懂怎麼做專題，到現在已經實作了兩次的專題，感觸非常良多，了解到要製作一份專題並不容易，必須要靠團隊合作才能完成，有時在找資料的過程中會出現一點口角，意見上也會有分岐，工作內容分配不當也會產生爭執。但在製作專題的過程中，不僅讓我學習到新的知識，更進一步熟練我的程式技術以及美術設計能力，最重要的是能夠了解組員們的想法，還能培養組員的默契以及感情，也要感謝老師的幫忙，無論什麼時候，再忙也會幫我們看專題，然後告訴我們哪裡要改進、哪裡做得很好。

10636017 陳易陞

因為有了五專做專題的經驗，所以這次對於整個專題流程和作業有了更進一步的了解，從主題構思、繪製介面、設計資料庫、程式設計等等，我們組開了無數次的會議，慢慢將整個系統建構出來，非常感謝組長，總是在我們需要幫助的時候，給予指導，也謝謝組員們願意接納我的提議，並將它擴大並付諸實現，希望最後能做出一個完整的打工換宿APP，滿足使用者需求，也解決自己當初在尋找打工換宿時，所面臨的問題。

10636028 李順玉

專題是我們要將所學的專業能力實現在應用層面上，不再只是照著書本走，而是要多方面的面對不同領域的知識。開始應用系上老師教導給我們的事物，並和各式不同的人溝通，其中包括我們自己團隊的夥伴、指導老師以及不同的評審老師；還有如何調配時間，該如何制訂各項目標完成的時間，怎樣才能讓我們的專案在限制的時間內完成。在技術層面，還好有組長豐富的專業知識與技術，在我遇到問題和困難時，只要虛心求教，組長能立刻給我支援。在團隊合作這方面，面對全新的團隊與指導老師，由於跟五專時期的團隊運作的方式相差甚遠，讓我一開始有些不習慣，在彼此溝通時不是那麼的順利，但是我們依然順利地完成了這個專題。

1. 參考資料

註冊登入<https://medium.com/@parsaniahardik/android-login-and-registration-example-with-php-mysql-session-management-27d42170cdd7>

<https://blog.csdn.net/IAFMAY/article/details/78467026>

跳頁&傳遞資料

<http://tw-hkt.blogspot.com/2017/12/intent.html>

讀取資料

<http://kanchengzxdfgcv.blogspot.com/2018/01/json-volley.html>

美術設計資源

<https://www.freepik.com/>

<https://colorhunt.co/palette/18378>

<https://www.flaticon.com/>

Android ：Adapter

<http://hsingjungchen.blogspot.com/2017/06/android-adapter.html>

Android--獲取系統時間的幾種方式

<https://blog.csdn.net/chaoyu168/article/details/50729689>

為你自己學 Git

<https://gitbook.tw/chapters/environment/gui-tools.html>

Android學習筆記 - 下拉式選單(Spinner)

<http://cooking-java.blogspot.com/2010/02/android-spinner.html>

附錄

|  |
| --- |
| 提案評審建議或意見： |
| 如何審核民宿? 系統特色如何達成? |
| 專題組的回應或遭遇之困難： |
| 我們打算利用交通部觀光局的Open Data 與我們民宿主上傳的合法營業執照和號碼進行比對，來審核民宿。  工作排程功能是我們的特色之一，民宿主可設定工作並排定時間，讓學生可以即時確認工作內容，這是FB和其他網站無法達到的。 |

|  |
| --- |
| 提案評審建議或意見： |
| 題目是組員的姓名組合嗎? |
| 專題組的回應或遭遇之困難： |
| 是的，希望學生在換宿時，可以和民宿主相處的像一家人一樣順心順利。 |

|  |
| --- |
| 提案評審建議或意見： |
| 1. B/C端App可分開做  2. 要考慮人性 |
| 專題組的回應或遭遇之困難： |
| 目前已將程式分割為學生端和民宿主端，且介面已進行優化。 |
| 提案評審建議或意見： |
| 1. 媒合平台的機制與規則可再加強  2. 研究Couchsurfing.com  3. 中英文化 |
| 專題組的回應或遭遇之困難： |
| 針對Couchsurfing我們了解到打工換宿在國外也非常流行，假如有外國旅客想來臺灣打工換宿，我們提供多國語言的機制，讓外國人使用起來更沒有語言障礙，也從Couchsurfing的理念來強化我們的平台機制。 |

附錄1-評審建議之修正情形

|  |  |
| --- | --- |
| 評審建議事項 | 修正情形 |
| 比較缺乏影音之互動，例如線上交談、民宿瀏覽……等。 | 我們會在系統上加上可以瀏覽多張圖片的ViewPager以及讓使用者可以上傳Youtube影片網址的WebView。而線上交談的功能，目前系統已可以使用Line互相交談，且Line也有提供網路通話或是視訊的功能。 |
| 建議住宿學生發揮創意與專長，例如網頁製作，協助宿主推廣在地特產，增加雙方收入。 | 目前系統已有可以讓學生根據自身專長做履歷的填寫，請詳見第12章使用手冊第160頁，同時民宿主也可以根據所需要的技能搜尋人才。 |
| 請增加學生與宿主之影音上傳功能。 | 如上，我們會在系統加入讓使用者上傳Youtube影片。 |
| 請評估是否能提供類似媒合開發平台，供不同族群使用。 | 目前我們系統鎖定的特定族群是APP的使用者，我們將此項目列入未來展望中。 |
| 文件遲交。 | 我們在下一次評分會提早交文件。 |
| 要分出與人力外包網的差異，目前現行的商務模式已經成形，如何改善使用者習慣，要點出來產品的優勢。 | 由於人力外包網和打工換宿的性質以及目標不相同，我們團隊未將此列入作為比較，原因是：   1. 打工換宿其目的是希望可以透過勞力以換取在「當地旅遊」的基本食宿。而人力外包網則是以賺錢為目的。 2. 並未有人將打工換宿作為長期工作。 |
| 沒有串接上政府的OPEN Data，希望能補上這一段（一評的時候評審已經給過建議）。 | 目前系統已加入全台灣的OPEN Data進行驗證。 |
| 網站功能應該要再更友善,能夠使得雇主（民宿主）與想要換宿的人，都更友善且明確知道如何操作。 | 我們系統會加入新手入門教學的功能，讓使用者可以更了解如何操作。 |
| 操作流程過於簡陋，無法肩背完善的滿足目前在換宿需求上的情境，要再多想想。 | 一家順順根據換宿流程出現的問題設計各個功能需求，詳細請至系統手冊1-2問題與機會，已新增了換宿流程與問題的文字說明。 |
| 系統功能有呈現。 | 謝謝評審。 |
| DEMO的方式應加強。 | 我們下次會改進DEMO的方式，謝謝評審的建議。 |

附錄2-系統手冊修正情形

|  |  |  |
| --- | --- | --- |
| 修正前 | 修正後 | 備註 |
| 部分頁圖片標題位於圖片上方。 | 將部分頁圖片標題移至圖片下方。 | 統一圖片標題位置。 |
| 第1章 背景與動機  第2頁 | 新增說明打工換宿流程。  第2頁 | 新增內容與圖片。 |
| 第2章 系統目標與預期成果  第7頁 | 修改商業模式圖片與文字敘述。  第7頁 | 修改圖片與文字。 |
| 第8章 資料庫設計  第36頁 | 新增OpenData相關資料表。  第36頁 | 新增資料表。 |
| 第9章 程式  第40頁 | 更改Common程式碼。  第40頁 | 更改程式碼。 |
| 第9章 程式  第54頁 | 更改getWorksheet程式碼。  第54頁 | 更改程式碼。 |
| 第9章 程式  第63頁 | 更改Hotelview\_Addhotel程式碼。  第63頁 | 更改程式碼。 |
| 第9章 程式  第76頁 | 更改Humansearch\_Info程式碼。  第76頁 | 更改程式碼。 |
| 第9章 程式  第80頁 | 更改Jobview\_Addjob程式碼。  第80頁 | 更改程式碼。 |
| 第9章 程式  第98頁 | 更改NavigationActivity程式碼。  第98頁 | 更改程式碼。 |
| 第9章 程式  第114頁 | 更改Findhostel\_hotelinfo程式碼。  第114頁 | 更改程式碼。 |
| 第9章 程式  第143頁 | 新增後端程式碼之連結。  第143頁 | 新增連結。 |