國立臺北商業大學

資訊管理系

109’資訊系統專案設計

系統手冊

****

組 別：第109404組

題 目：iPets

指導老師：林俊杰老師

組 長：10646004 李怡蓁

組 員：10646043 蔡欣恬 N1066405 黃琳恩

N1066414 王偉霖 N1066423 劉芳汝

中華民國109年11月25日

**目錄**

[第一章 背景與動機 21](#_Toc56781887)

[1-1 簡介 21](#_Toc56781888)

[1-2 問題與機會 22](#_Toc56781889)

[1-3 相關系統探討 23](#_Toc56781890)

[第二章 系統目標與預期成果 26](#_Toc56781891)

[2-1 系統目標 26](#_Toc56781892)

[2-2 預期成果 28](#_Toc56781893)

[第三章 系統規格 29](#_Toc56781894)

[3-1 系統架構 29](#_Toc56781895)

[3-2 系統軟硬體需求與技術平台 32](#_Toc56781896)

[3-3 使用標準與工具 33](#_Toc56781897)

[第四章 專案時程與組織分工 34](#_Toc56781898)

[4-1 專案時程 34](#_Toc56781899)

[4-2 專案組織與分工 35](#_Toc56781900)

[第五章 需求模型 36](#_Toc56781901)

[5-1 使用者需求 36](#_Toc56781902)

[5-2 使用個案圖（Use case diagram） 38](#_Toc56781903)

[5-3 使用個案描述 39](#_Toc56781904)

[5-4 分析類別圖 44](#_Toc56781905)

[第六章 設計模型 47](#_Toc56781906)

[6-1 循序圖（Sequence diagram） 47](#_Toc56781907)

[6-2 設計類別圖（Design class diagram） 57](#_Toc56781908)

[第七章 實作模型 61](#_Toc56781909)

[7-1 佈署圖（Deployment diagram） 61](#_Toc56781910)

[7-2 套件圖（Package diagram） 62](#_Toc56781911)

[7-3 元件圖（Component diagram） 64](#_Toc56781912)

[7-4 狀態機（State machine） 65](#_Toc56781913)

[第八章 資料庫設計 66](#_Toc56781914)

[8-1 資料庫關聯表 66](#_Toc56781915)

[8-2 表格及其 Meta data 67](#_Toc56781916)

[第九章 程式 70](#_Toc56781917)

[9-1 元件清單及其規格描述 70](#_Toc56781918)

[9-2 其他附屬之各種元件 127](#_Toc56781919)

[第十章 測試模型 203](#_Toc56781920)

[10-1 測試計畫 203](#_Toc56781921)

[10-2 測試個案與測試結果資料 208](#_Toc56781922)

[第十一章 操作手冊 229](#_Toc56781923)

[第十二章 使用手冊 231](#_Toc56781924)

[第十三章 感想 253](#_Toc56781925)

[第十四章 參考資料 259](#_Toc56781926)

**表目錄**

[表1-3-1 相關系統比較 24](#_Toc57151548)

[表3-2-1 環境測試表 32](#_Toc57151549)

[表3-2-2 軟硬體需求表 32](#_Toc57151550)

[表3-3-1 使用標準與工具表 33](#_Toc57151551)

[表4-1-1 專案時程表 34](#_Toc57151552)

[表4-1-2 分工表 35](#_Toc57151553)

[表5-1-1 功能需求表 36](#_Toc57151554)

[表5-1-2 非功能性需求表 37](#_Toc57151555)

[表8-2-1 資料表描述：用戶資料 67](#_Toc57151556)

[表8-2-2 資料表描述：行事曆行程 68](#_Toc57151557)

[表8-2-3 資料表描述：問題反饋 68](#_Toc57151558)

[表8-2-4 資料表描述：寵物資料 69](#_Toc57151559)

[表8-2-5 資料表描述：倒數計時器 69](#_Toc57151560)

[表9-1-1 首頁功能列表 70](#_Toc57151561)

[表9-1-2 關於我們功能列表 70](#_Toc57151562)

[表9-1-3 狗狗科普功能列表 70](#_Toc57151563)

[表9-1-4 聯繫我們功能列表 71](#_Toc57151564)

[表9-1-5 登入功能列表 71](#_Toc57151565)

[表9-1-6 註冊功能列表 71](#_Toc57151566)

[表9-1-7 寵物資訊功能列表 72](#_Toc57151567)

[表9-1-8 行事曆功能列表 73](#_Toc57151568)

[表9-1-9 相簿功能列表 74](#_Toc57151569)

[表9-1-10 相簿瀏覽功能列表 74](#_Toc57151570)

[表9-1-11 寵物品種辨識功能列表 74](#_Toc57151571)

[表9-1-12 風格轉換功能列表 75](#_Toc57151572)

[表9-1-13 新增寵物功能列表 75](#_Toc57151573)

[表9-1-14 文章功能列表 75](#_Toc57151574)

[表9-1-15 管理文章功能列表 76](#_Toc57151575)

[表9-1-16 管理領養機構功能列表 77](#_Toc57151576)

[表9-1-17 管理回饋功能列表 77](#_Toc57151577)

[表9-1-18 Navbar功能列表 77](#_Toc57151578)

[表9-1-19 會員資料功能列表 78](#_Toc57151579)

[表9-1-20 寵物資料功能列表 78](#_Toc57151580)

[表9-1-21 編輯個人資料功能列表 78](#_Toc57151581)

[表9-1-22 更改密碼功能列表 79](#_Toc57151582)

[表9-1-23 管理者資料功能列表 79](#_Toc57151583)

[表9-1-24 取得經緯度 80](#_Toc57151584)

[表9-1-25 轉換經緯度 80](#_Toc57151585)

[表9-1-26 取得領養機構資訊 81](#_Toc57151586)

[表9-1-27 取得組員資訊 81](#_Toc57151587)

[表9-1-28 取得文章資訊 81](#_Toc57151588)

[表9-1-29 排序1-3-1結果 82](#_Toc57151589)

[表9-1-30 新增使用者回饋 82](#_Toc57151590)

[表9-1-31 檢查註冊資訊 83](#_Toc57151591)

[表9-1-32 隱藏密碼 83](#_Toc57151592)

[表9-1-33 顯示密碼 84](#_Toc57151593)

[表9-1-34 檢查註冊資訊 84](#_Toc57151594)

[表9-1-35 隱藏密碼 85](#_Toc57151595)

[表9-1-36 顯示密碼 85](#_Toc57151596)

[表9-1-37 隱藏確認密碼 86](#_Toc57151597)

[表9-1-38 顯示確認密碼 86](#_Toc57151598)

[表9-1-39 取得檔案資料 86](#_Toc57151599)

[表9-1-40 建立相簿並新增頭貼 87](#_Toc57151600)

[表9-1-41 建立寵物資訊 87](#_Toc57151601)

[表9-1-42 下載頭貼路徑 88](#_Toc57151602)

[表9-1-43 轉換時間 89](#_Toc57151603)

[表9-1-44 判斷行程時間 89](#_Toc57151604)

[表9-1-45 判斷行程是否逾期 90](#_Toc57151605)

[表9-1-46 年齡計算 90](#_Toc57151606)

[表9-1-47 更新寵物年齡 91](#_Toc57151607)

[表9-1-48 更新寵物資訊 91](#_Toc57151608)

[表9-1-49 取得寵物頭貼 92](#_Toc57151609)

[表9-1-50 新增事件 93](#_Toc57151610)

[表9-1-51 取得事件 93](#_Toc57151611)

[表9-1-52 點取日期新增事件 94](#_Toc57151612)

[表9-1-53 關閉新增事件 94](#_Toc57151613)

[表9-1-54 行事曆顯示類型 94](#_Toc57151614)

[表9-1-55 取得事件顏色 95](#_Toc57151615)

[表9-1-56 返回當日 95](#_Toc57151616)

[表9-1-57 顯示上個月/週/日/四天 95](#_Toc57151617)

[表9-1-58 顯示下個月/週/日/四天 95](#_Toc57151618)

[表9-1-59 判斷頻率並新增事件 96](#_Toc57151619)

[表9-1-60 取得編輯事件 98](#_Toc57151620)

[表9-1-61 編輯事件 98](#_Toc57151621)

[表9-1-62 刪除事件 99](#_Toc57151622)

[表9-1-63 查看事件 100](#_Toc57151623)

[表9-1-64 取得事件起訖時間 100](#_Toc57151624)

[表9-1-65 行事曆顯示時間範圍 101](#_Toc57151625)

[表9-1-66 取得檔案資料 102](#_Toc57151626)

[表9-1-67 建立相簿 102](#_Toc57151627)

[表9-1-68 取得相簿資訊 102](#_Toc57151628)

[表9-1-69 取得檔案資料 103](#_Toc57151629)

[表9-1-70 新增相片 104](#_Toc57151630)

[表9-1-71 刪除相片 104](#_Toc57151631)

[表9-1-72 取得單一相簿圖片 105](#_Toc57151632)

[表9-1-73 下載圖片 106](#_Toc57151633)

[表9-1-74 取得圖片資訊 106](#_Toc57151634)

[表9-1-75 取得圖片資訊 107](#_Toc57151635)

[表9-1-76 取得上傳圖片 108](#_Toc57151636)

[表9-1-77 建立寵物資訊 108](#_Toc57151637)

[表9-1-78 下載頭貼路徑 110](#_Toc57151638)

[表9-1-79 監聽網址 110](#_Toc57151639)

[表9-1-80 取得單一文章資料 110](#_Toc57151640)

[表9-1-81 刪除文章 111](#_Toc57151641)

[表9-1-82 新增內容列 111](#_Toc57151642)

[表9-1-83 刪除內容列 112](#_Toc57151643)

[表9-1-84 清除文章暫存資料 112](#_Toc57151644)

[表9-1-85 顯示文章內容 112](#_Toc57151645)

[表9-1-86 新增文章 113](#_Toc57151646)

[表9-1-87 編輯文章 114](#_Toc57151647)

[表9-1-88 取得文章 114](#_Toc57151648)

[表9-1-89 刪除領養機構資訊 115](#_Toc57151649)

[表9-1-90 清除領養機構暫存資料 115](#_Toc57151650)

[表9-1-91 新增領養機構 116](#_Toc57151651)

[表9-1-92 編輯領養機構 117](#_Toc57151652)

[表9-1-93 取得領養機構資訊 117](#_Toc57151653)

[表9-1-94 取得回饋資訊 118](#_Toc57151654)

[表9-1-95 確認用戶登入狀態 118](#_Toc57151655)

[表9-1-96 取得會員資料 119](#_Toc57151656)

[表9-1-97 取得寵物清單 119](#_Toc57151657)

[表9-1-98 更新行程列表 120](#_Toc57151658)

[表9-1-99 取得寵物資料 121](#_Toc57151659)

[表9-1-100 取得個人資訊 122](#_Toc57151660)

[表9-1-101 更新個人資訊 122](#_Toc57151661)

[表9-1-102 更新密碼 123](#_Toc57151662)

[表9-1-103 確認重複密碼 124](#_Toc57151663)

[表9-1-104 隱藏目前密碼 124](#_Toc57151664)

[表9-1-105 顯示目前密碼 124](#_Toc57151665)

[表9-1-106 隱藏新密碼 125](#_Toc57151666)

[表9-1-107 顯示新密碼 125](#_Toc57151667)

[表9-1-108 隱藏確認密碼 125](#_Toc57151668)

[表9-1-109 顯示確認密碼 126](#_Toc57151669)

[表9-1-110 取得個人資訊 126](#_Toc57151670)

[表9-2-1 首頁功能列表 127](#_Toc57151671)

[表9-2-2 登入功能列表 127](#_Toc57151672)

[表9-2-3 註冊功能列表 128](#_Toc57151673)

[表9-2-4 新增使用者資訊功能列表 128](#_Toc57151674)

[表9-2-5 行事曆功能列表 128](#_Toc57151675)

[表9-2-6 新增寵物功能列表 129](#_Toc57151676)

[表9-2-7 新增行程功能列表 129](#_Toc57151677)

[表9-2-8 詳細行程顯示功能列表 129](#_Toc57151678)

[表9-2-9 設定功能列表 130](#_Toc57151679)

[表9-2-10 更新會員資訊功能列表 130](#_Toc57151680)

[表9-2-11 更新會員帳號功能列表 130](#_Toc57151681)

[表9-2-12 更新會員密碼功能列表 131](#_Toc57151682)

[表9-2-13 更新寵物資訊功能列表 131](#_Toc57151683)

[表9-2-14 常見問題功能列表 132](#_Toc57151684)

[表9-2-15 聯絡我們功能列表 132](#_Toc57151685)

[表9-2-16 通知功能列表 132](#_Toc57151686)

[表9-2-17 定位功能列表 133](#_Toc57151687)

[表9-2-18 定位功能列表 133](#_Toc57151688)

[表9-2-19 定位功能列表 134](#_Toc57151689)

[表9-2-20 定位功能列表 134](#_Toc57151690)

[表9-2-21 定位功能列表 134](#_Toc57151691)

[表9-2-22 設定通知日期 135](#_Toc57151692)

[表9-2-23 計算倒數天數 136](#_Toc57151693)

[表9-2-24 取得寵物資訊 136](#_Toc57151694)

[表9-2-25 取得寵物照片 138](#_Toc57151695)

[表9-2-26 設定倒數計時器日期 139](#_Toc57151696)

[表9-2-27 計算到期天數 140](#_Toc57151697)

[表9-2-28 取得文件名稱 140](#_Toc57151698)

[表9-2-29 自動登入 141](#_Toc57151699)

[表9-2-30 使用者登入 141](#_Toc57151700)

[表9-2-31 檢查密碼 142](#_Toc57151701)

[表9-2-32 使用者註冊 143](#_Toc57151702)

[表9-2-33 新增使用者資訊 145](#_Toc57151703)

[表9-2-34 函式導向、顯示行程 146](#_Toc57151704)

[表9-2-35 導頁至新增行程 150](#_Toc57151705)

[表9-2-36 標記行程 150](#_Toc57151706)

[表9-2-37 函式導向 153](#_Toc57151707)

[表9-2-38 取得相機、相簿權限 154](#_Toc57151708)

[表9-2-39 讀取照片 155](#_Toc57151709)

[表9-2-40 調整照片大小 156](#_Toc57151710)

[表9-2-41 新增寵物資訊 156](#_Toc57151711)

[表9-2-42 上傳圖片 158](#_Toc57151712)

[表9-2-43 新增行程 159](#_Toc57151713)

[表9-2-44 刪除行程 162](#_Toc57151714)

[表9-2-45 更新行程 163](#_Toc57151715)

[表9-2-46 登出 164](#_Toc57151716)

[表9-2-47 傳值至更新會員帳號 165](#_Toc57151717)

[表9-2-48 更新會員資料 165](#_Toc57151718)

[表9-2-49 將EditText轉換成日期選擇器 167](#_Toc57151719)

[表9-2-50 取得會員資料 168](#_Toc57151720)

[表9-2-51 更新會員帳號 169](#_Toc57151721)

[表9-2-52 取得會員帳號 170](#_Toc57151722)

[表9-2-53 更新會員密碼 171](#_Toc57151723)

[表9-2-54 函式導向 172](#_Toc57151724)

[表9-2-55 取得寵物照片 173](#_Toc57151725)

[表9-2-56 取得寵物資訊 173](#_Toc57151726)

[表9-2-57 上傳圖片 175](#_Toc57151727)

[表9-2-58 更新寵物資訊 175](#_Toc57151728)

[表9-2-59 將EditText轉換成日期選擇器 177](#_Toc57151729)

[表9-2-60 讀取照片 178](#_Toc57151730)

[表9-2-61 調整照片大小 179](#_Toc57151731)

[表9-2-62 顯示常見問題 179](#_Toc57151732)

[表9-2-63 新增聯絡我們資訊 183](#_Toc57151733)

[表9-2-64 發送通知 183](#_Toc57151734)

[表9-2-65 確認有無安裝地圖 185](#_Toc57151735)

[表9-2-66 確認playService版本 186](#_Toc57151736)

[表9-2-67 確認權限 187](#_Toc57151737)

[表9-2-68 初始化地圖 187](#_Toc57151738)

[表9-2-69 執行初始化動作 188](#_Toc57151739)

[表9-2-70 取得現在定位點 189](#_Toc57151740)

[表9-2-71 取得查詢地址定位點 190](#_Toc57151741)

[表9-2-72 附近位置查詢 191](#_Toc57151742)

[表9-2-73 數據解析 196](#_Toc57151743)

[表9-2-74 顯示數據列表的結果 196](#_Toc57151744)

[表9-2-75 顯示附近位置URL 196](#_Toc57151745)

[表9-2-76 下載附近位置URL 197](#_Toc57151746)

[表9-2-77 解析數據存於列表中 198](#_Toc57151747)

[表9-2-78 解析附近地點詳細位置 198](#_Toc57151748)

[表9-2-79 解析附近地點詳細資訊 199](#_Toc57151749)

[表9-2-80 函式導向 200](#_Toc57151750)

[表9-2-81 上傳照片至資料庫 200](#_Toc57151751)

[表9-2-82 資料庫連結設定 202](#_Toc57151752)

[表9-2-83 讀取照片 202](#_Toc57151753)

[表10-2-1 註冊測試結果 208](#_Toc57151754)

[表10-2-2 登入測試結果 208](#_Toc57151755)

[表10-2-3 會員測試結果 209](#_Toc57151756)

[表10-2-4 管理者測試結果 215](#_Toc57151757)

[表10-2-5 關於我們測試結果 219](#_Toc57151758)

[表10-2-6 狗狗科普測試結果 219](#_Toc57151759)

[表10-2-7 人與狗的關係測試結果 220](#_Toc57151760)

[表10-2-8 聯繫我們測試結果 220](#_Toc57151761)

[表10-2-9 註冊測試結果 221](#_Toc57151762)

[表10-2-10 登入測試結果 221](#_Toc57151763)

[表10-2-11 首頁測試結果 222](#_Toc57151764)

[表10-2-12 行事曆測試結果 223](#_Toc57151765)

[表10-2-13 相簿測試結果 225](#_Toc57151766)

[表10-2-14 定位測試結果 225](#_Toc57151767)

[表10-2-15 定位測試結果 227](#_Toc57151768)

[表11-1-1 網頁操作手冊 229](#_Toc57151769)

[表11-1-2 APP操作手冊 230](#_Toc57151770)

[表12-1-1 使用手冊之輪播資訊圖 231](#_Toc57151771)

[表12-1-2 使用手冊之領養機構資訊 232](#_Toc57151772)

[表12-1-3 使用手冊之領養資訊 232](#_Toc57151773)

[表12-1-4 使用手冊之關於我們 233](#_Toc57151774)

[表12-1-5 使用手冊之狗狗科普 233](#_Toc57151775)

[表12-1-6 使用手冊之狗狗科普文章 234](#_Toc57151776)

[表12-1-7 使用手冊之人與狗的關係 234](#_Toc57151777)

[表12-1-8 使用手冊之登入及註冊 235](#_Toc57151778)

[表12-1-9 使用手冊之行事曆 235](#_Toc57151779)

[表12-1-10 使用手冊之新增事件 236](#_Toc57151780)

[表12-1-11 使用手冊之相簿 237](#_Toc57151781)

[表12-1-12 使用手冊之新增相簿 237](#_Toc57151782)

[表12-1-13 使用手冊之查看相片 238](#_Toc57151783)

[表12-1-14 使用手冊之刪除相片 238](#_Toc57151784)

[表12-1-15 使用手冊之寵物品種辨識 239](#_Toc57151785)

[表12-1-16 使用手冊之圖片風格轉換 239](#_Toc57151786)

[表12-1-17 使用手冊之編輯個人資訊 240](#_Toc57151787)

[表12-1-18 使用手冊之新增毛小孩 240](#_Toc57151788)

[表12-1-19 使用手冊之更改密碼 241](#_Toc57151789)

[表12-1-20 使用手冊之常見問題 241](#_Toc57151790)

[表12-1-21 使用手冊之聯絡我們 242](#_Toc57151791)

[表12-1-22 使用手冊之新增文章 242](#_Toc57151792)

[表12-1-23 使用手冊之刪除文章 243](#_Toc57151793)

[表12-1-24 使用手冊之編輯文章 243](#_Toc57151794)

[表12-1-25 使用手冊之新增領養機構 244](#_Toc57151795)

[表12-1-26 使用手冊之刪除領養機構 244](#_Toc57151796)

[表12-1-27 使用手冊之編輯領養機構 245](#_Toc57151797)

[表12-1-28 使用手冊之管理使用者回饋 245](#_Toc57151798)

[表12-2-1 使用手冊之登入及註冊 246](#_Toc57151799)

[表12-2-2 使用手冊之新增寵物 247](#_Toc57151800)

[表12-2-3 使用手冊之編輯寵物 248](#_Toc57151801)

[表12-2-4 使用手冊之倒數計時器 249](#_Toc57151802)

[表12-2-5 使用手冊之行事曆 249](#_Toc57151803)

[表12-2-6 使用手冊之新增行程 250](#_Toc57151804)

[表12-2-7 使用手冊之編輯/刪除行程 250](#_Toc57151805)

[表12-2-8 使用手冊之相簿 251](#_Toc57151806)

[表12-2-9 使用手冊之導航 252](#_Toc57151807)

[表12-2-10 使用手冊之附近店家/醫院/公園 252](#_Toc57151808)

**圖目錄**

[圖3-1-1 系統架構圖 29](#_Toc57151809)

[圖3-1-2 網頁架構 30](#_Toc57151810)

[圖3-1-3 APP架構 31](#_Toc57151811)

[圖5-2-1 使用者個案圖 38](#_Toc57151812)

[圖5-3-1 註冊會員活動圖 39](#_Toc57151813)

[圖5-3-2 使用者登入活動圖 39](#_Toc57151814)

[圖5-3-3 修改會員資料活動圖 39](#_Toc57151815)

[圖5-3-4 行事曆管理活動圖 40](#_Toc57151816)

[圖5-3-5 相簿管理活動圖 40](#_Toc57151817)

[圖5-3-6 寵物資訊管理活動圖 41](#_Toc57151818)

[圖5-3-7 寵物品種辨識活動圖 41](#_Toc57151819)

[圖5-3-8 圖片風格轉換活動圖 42](#_Toc57151820)

[圖5-3-9 地圖活動圖 42](#_Toc57151821)

[圖5-3-10 新增領養機構活動圖 43](#_Toc57151822)

[圖5-3-11 新增狗狗科普活動圖 43](#_Toc57151823)

[圖5-3-12 回饋資訊活動圖 43](#_Toc57151824)

[圖5-4-1 （Web）會員分析類別圖 44](#_Toc57151825)

[圖5-4-2 （Web）行事曆分析類別圖 44](#_Toc57151826)

[圖5-4-3 （Web）深度學習分析類別圖 45](#_Toc57151827)

[圖5-4-4 （APP）會員分析類別圖 45](#_Toc57151828)

[圖5-4-5 （APP）行事曆分析類別圖 46](#_Toc57151829)

[圖5-4-6 （APP）定位分析類別圖 46](#_Toc57151830)

[圖6-1-1 註冊循序圖 47](#_Toc57151831)

[圖6-1-2 登入循序圖 48](#_Toc57151832)

[圖6-1-3 修改會員資料循序圖 48](#_Toc57151833)

[圖6-1-4 相簿管理循序圖 49](#_Toc57151834)

[圖6-1-5 行事曆管理循序圖 49](#_Toc57151835)

[圖6-1-6 寵物資訊管理循序圖 50](#_Toc57151836)

[圖6-1-7 寵物品種辨識循序圖 50](#_Toc57151837)

[圖6-1-8 圖片風格轉換循序圖 51](#_Toc57151838)

[圖6-1-9 查詢寵物醫院循序圖 51](#_Toc57151839)

[圖6-1-10 查詢附近店家循序圖 52](#_Toc57151840)

[圖6-1-11 查詢公園循序圖 53](#_Toc57151841)

[圖6-1-12 領養機構循序圖 53](#_Toc57151842)

[圖6-1-13 狗狗科普循序圖 54](#_Toc57151843)

[圖6-1-14 回饋管理循序圖 54](#_Toc57151844)

[圖6-1-15 回饋循序圖 55](#_Toc57151845)

[圖6-1-16 狗狗科普管理循序圖 55](#_Toc57151846)

[圖6-1-17 領養機構管理循序圖 56](#_Toc57151847)

[圖6-2-1 （Web）會員設計類別圖 57](#_Toc57151848)

[圖6-2-2 （Web）行事曆設計類別圖 58](#_Toc57151849)

[圖6-2-3 （Web）深度學習設計類別圖 58](#_Toc57151850)

[圖6-2-4 （APP）會員設計類別圖 59](#_Toc57151851)

[圖6-2-5 （APP）行事曆設計類別圖 59](file:///C:\Users\YiChen\Desktop\iPets\document\109-1%20iPets_系統手冊資料.docx#_Toc57151852)

[圖6-2-6 （APP）定位設計類別圖 60](#_Toc57151853)

[圖7-1-1 （Web）佈署圖 61](#_Toc57151854)

[圖7-1-2 （APP）佈署圖 61](#_Toc57151855)

[圖7-2-1 （Web）套件圖 62](#_Toc57151856)

[圖7-2-2 （APP）套件圖 63](#_Toc57151857)

[圖7-3-1 （Web）元件圖 64](#_Toc57151858)

[圖7-3-2 （APP）元件圖 64](#_Toc57151859)

[圖7-4-1 狀態機 65](#_Toc57151860)

[圖8-1-1 資料庫關聯表 66](file:///C:\Users\YiChen\Desktop\iPets\document\109-1%20iPets_系統手冊資料.docx#_Toc57151861)

# 第一章 背景與動機

## 1-1 簡介

**一、寵物飼養背景與現況**

近年來，由於「臺灣的人口結構性變化」和「個人經濟壓力因素」，讓寵物陪伴變成主流，現代的年輕族群結婚後未必會生育孩子，反而會選擇飼養寵物，因為飼養寵物的花費及精力遠遠小於養育孩子，導致現在社會上部分家庭都會飼養寵物，將寵物當成孩子寵愛。

現在的社會給年輕族群帶來許多壓力及負擔，當人們感到身心俱疲的時候總會要有一個抒壓、療癒的管道，而寵物的存在便會是最佳首選，當人在極度疲憊的情況下，看到一個可愛的寵物跟你撒嬌，其實疲憊感瞬間可以降低許多，你的喜怒哀樂，都會有一個可愛的寵物與你一起面對。雖然無法和牠言語，但再多的話也無法表達出內心的情緒，陪伴才是最佳消化情緒的辦法。根據農委會調查統計，目前台灣家庭仍以飼養狗類居多，故此次寵物居家管理專題的對象主要針對狗設計。

**二、寵物居家管理相關系統背景與現況**

目前在手機商店的寵物居家管理程式只有少數幾個，經過本組的觀察後，發現功能幾乎大同小異，而且UI設計風格差不多，整體較無差異性。所以本專題目標是做出不同於市面上且更加舒適的UI，同時具備較便利的管理功能，以及寵物品種辨識技術、店家定位等特殊功能，增加系統差異性及使用者的黏著度。

不同於其他寵物居家管理系統只能在手機上使用，本專題開發了手機版及網頁版的管理系統，兩邊系統能夠資料同步與功能相容。除此之外，兩邊系統也會有一些各自獨立的功能，手機版本有定位功能；網頁版本則會有寵物品種辨識以及寵物圖片風格轉換，結合現今主流趨勢人工智慧，讓本次專題的管理系統更加有趣且豐富。

## 1-2 問題與機會

**問題一：飼養寵物的家庭數逐年升高**

情況：

隨著愈來愈多人以養寵物取代結婚生子，許多青壯年到了適婚年齡，都尚未步入婚姻，就算結婚也沒有計畫生育孩子，在這種情況下，這些家庭便會增加飼養貓或狗的想法。

機會：

「iPets」為這些頂客族、不婚主義者以及正在飼養寵物的飼主，打造了方便且實用的網站及APP，輔助飼主管理寵物的生活，輕鬆養寵物不用擔心忙碌而忘記打理自己的寵物；新手或有想養寵物的人們更是適合，iPets系統有多項功能幫助想養寵物的新手學會如何打理自己的寵物，不用擔心會手忙腳亂。有「iPets」在，讓使用者養寵物不費力、開心輕鬆養寵物。

**問題二：現代人習慣使用手機紀錄事情**

情況：

在這網路發達、高科技的時代，愈來愈少人用手札的方式紀錄行程在行事曆，現代社會人手一機，習慣透過手機行事曆紀錄行程，任何與寵物有關的資訊也會上網查詢而並非閱讀書籍。另外，我們也發現許多飼主會忘記定期打理毛孩子，像是剃毛、驅蟲、打疫苗、洗澡等，這會產生一些影響導致寵物無法健康成長。

機會：

「iPets」有多項提醒功能，像是為寵物剃毛、驅蟲、打疫苗等定期性的事項，使用者可以設定一個週期性於行事曆並開啟提醒功能，讓使用者不必擔心會忘記行程。另外還包括一些用戶專屬的功能，像是定位系統，可以協助使用者快速找到寵物店家、寵物醫院、寵物公園。

## 1-3 相關系統探討

本次專案，小組成員蒐集網路資料，發現兩款寵物資訊系統有較高的人氣，分別是「11Pets」及「Afurkid毛小孩寵物資訊」，其中「11Pets」擁有網頁版及手機版，其功能結合寵物醫院資訊、疫苗資訊等多項專業功能，同時也有提供領養的管道，但目前並未有台灣地區的選項；「Afurkid毛小孩寵物資訊」只有提供網頁版，此資訊網會提供各項與寵物有關的活動，像是狗狗講座、狗友聚會等，其中最特別的功能就是寵物保險，協助使用者替寵物辦理保險，同時也提供許多寵物收容所的資料，方便使用者查詢；「iPets」則會結合以上兩個系統的優勢，開發出更方便使用者使用及增加使用者樂趣的寵物資訊系統，像是寵物資訊管理、寵物行事曆、寵物品種辨識等功能，並且提供寵物領養管道，方便使用者查詢領養機構。

### 表1-3-1 相關系統比較

**●：**具備該功能

|  |  |  |  |
| --- | --- | --- | --- |
| 平台  項目 | iPets | 11Pets：呵護寵物 | Afurkid  毛小孩寵物資訊 |
| Web | ● | ● | ● |
| APP （iOS） |  | ● |  |
| APP （Android） | ● | ● |  |
| 主題類別 | 寵物管理、寵物品種辨識、圖片風格轉換、定位服務 | 寵物管理、定位服務、收養專區 | 資訊分享 |
| 通知 | ● | ● | ● |
| 會員系統 | ● | ● | ● |
| 文章推薦 | ● |  | ● |
| 領養資訊 | ● | ● | ● |
| 關鍵字搜尋 |  | ● | ● |
| 意見回饋 | ● | ● | ● |
| 網站導覽 | ● |  |  |
| 寵物管理 | ● | ● |  |
| 倒數計時器 | ● | ● |  |
| 平台  項目 | iPets | 11Pets：呵護寵物 | Afurkid  毛小孩寵物資訊 |
| 行事曆 | ● | ● |  |
| 寵物相簿 | ● | ● |  |
| 寵物品種辨識 | ● |  |  |
| 圖片風格轉換 | ● |  |  |
| 定位功能 | ● | ● |  |
| 附近店家資訊 | ● | ● |  |

# 第二章 系統目標與預期成果

## 2-1 系統目標

「iPets」之系統目標可分為以下四個項目:

1. 簡潔的介面設定

（一）人性化管理介面：操作介面簡潔有力，讓使用者可以一目了然。

1. 方便的寵物管理

（一）多寵物管理：假如飼養多隻寵物，只需要註冊該寵物，便可以輕鬆的切換介面查看寵物的行程。

（二）行事曆：讓用戶可以記錄寵物的相關行程（如：剃毛、美容等）。

（三）倒數計時器：利用簡潔的介面，進行時間倒數，方便用戶快速安排當日行程。

（四）自動排程：用戶只需設定重複頻率及次數，系統便會自動排程。

（五）顏色區別：將即將到期的行程以不同顏色表示，讓用戶快速區分活動的先後順序。

（六）提醒功能：活動當天會跳出通知提醒用戶當日活動。

1. 豐富使用者體驗

（一）寵物品種辨識：運用人工智慧技術，讓用戶體驗辨識功能，只需上傳照片，經由系統分析，便可以得知該寵物品種，提升用戶體驗。

（二）圖片風格轉換：運用人工智慧技術，將上傳的圖片轉換成兩種不同風格的照片，讓用戶對於每次圖片轉換都充滿著期待感。

（三）好文推薦：過濾內容農場，提供寵物相關文章，給予用戶獲得正確知識的優質平台。

（四）網站導覽：可以快速帶領用戶了解網站特色。

1. 提高用戶黏著度

（一）相簿空間：提供用戶存放寵物相片的空間，讓用戶可以將寵物每個瞬間記錄下來，提升用戶使用意願。

（二）提供店家資訊：讓用戶可以快速搜尋周遭的寵物店家、寵物醫院、寵物公園資訊，並且可以做關鍵字查詢，方便用戶快速找到需要的店家。

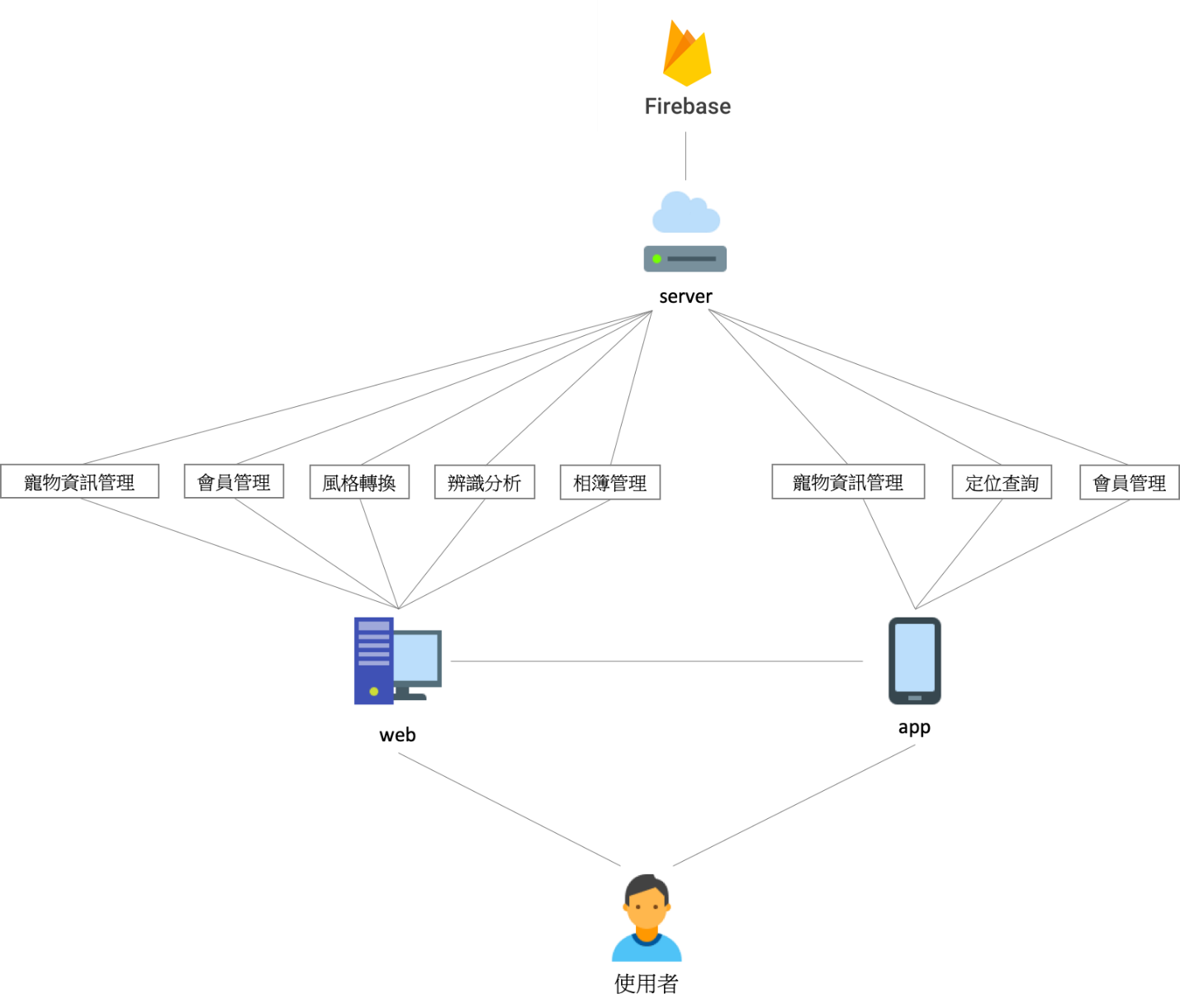
## 2-2 預期成果

1. 使用者使用率提升
2. 本平台定期更新寵物資訊的相關報導，讓使用者可以接收最新寵物資訊，豐富閒暇時間，提高使用者使用忠誠度。
3. 用戶可以有效的運用行事曆，替寵物規劃行程，完整獨立出專屬於寵物的行程。
4. 用戶可以有效運用倒數計時器，提醒用戶寵物的某項行程即將到來。
5. 若不想要手動新增行程，也可以運用內建的行程活動，只要將其時間頻率設定完成，便可以快速替寵物規劃好活動。
6. 用戶可以透過搜尋店家功能，快速找出附近所有的寵物店家、寵物醫院、寵物公園，方便用戶找尋店家。
7. 用戶專屬的寵物品種辨識/圖片風格轉換，讓用戶對於每次使用都充滿期待，增加使用的驚喜感。
8. 聚集更多寵物愛好者/飼主
9. 建立良好的用戶機制，將各種等級的用戶給予不同功能（如：白金用戶可以相互定位狗友的位置）。
10. 降低系統使用難度，將內容圖像化/條列化，方便各年齡層使用者使用。
11. 保障用戶的資訊安全性，讓用戶的個人資訊（如：定位）可以不被第三方監視。

# 第三章 系統規格

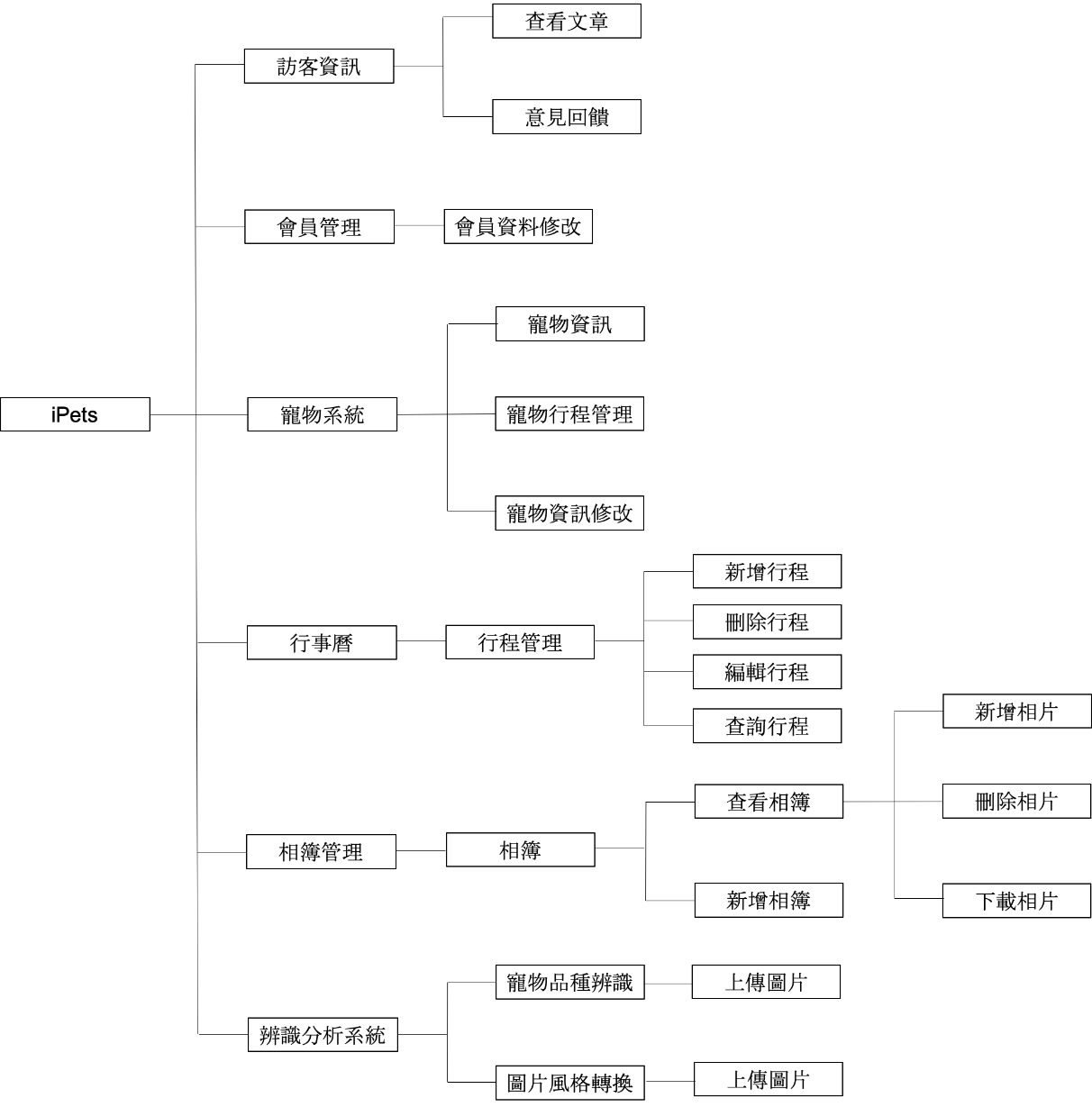
## 3-1 系統架構

「iPets」系統架構如下圖3-1-1所示，網站主要功能有「寵物資訊管理」、「會員管理」、「風格轉換」、「辨識分析」、「相簿管理」；APP主要功能為提供使用者更加方便寵物管理介面，包括「寵物資訊管理」、「定位查詢」、「會員管理」。



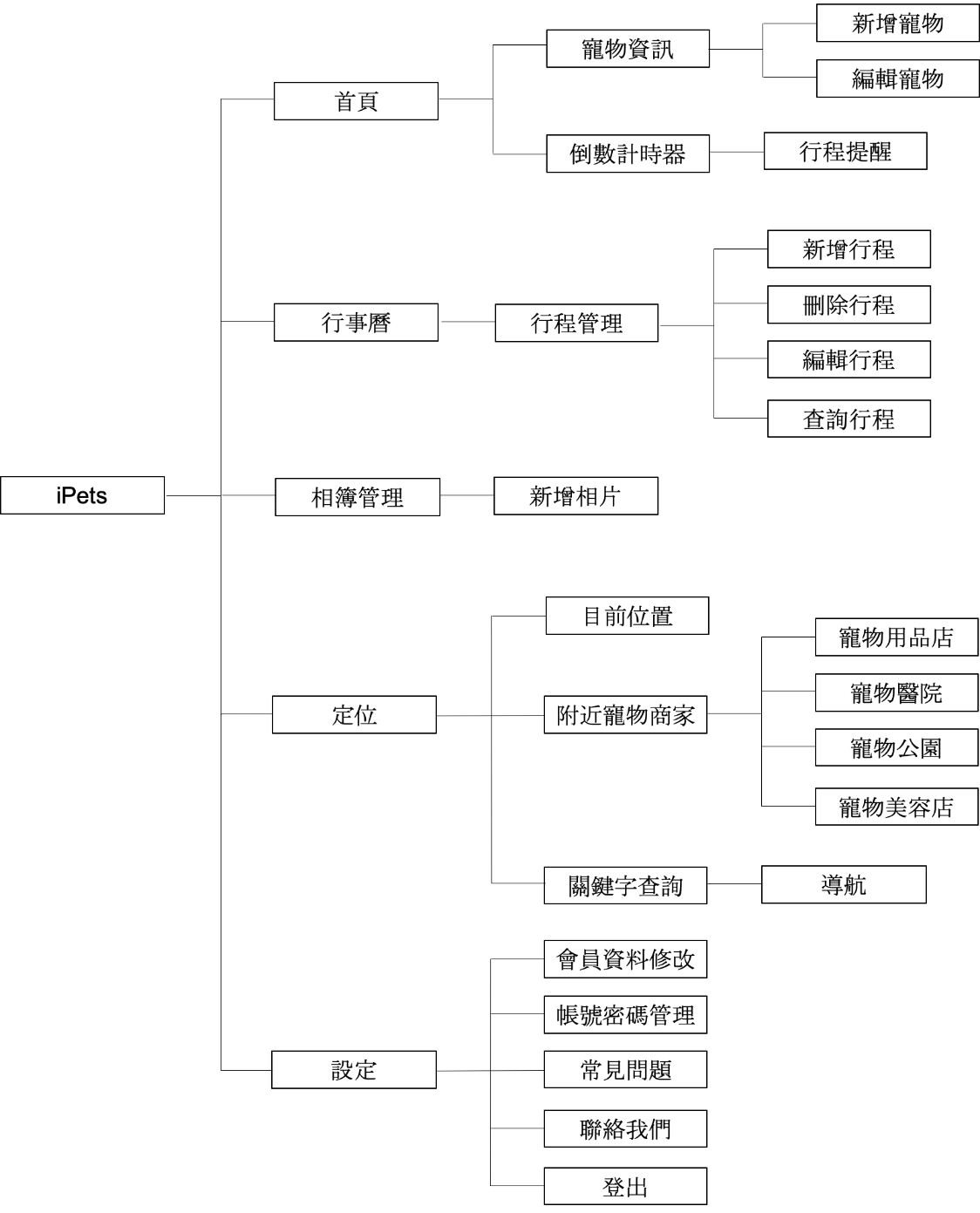
#### 圖3-1-1 系統架構圖

「iPets」網頁架構如下圖3-1-2所示，系統拆成六大部分，用戶可透過「訪客資訊」、「會員管理」、「寵物系統」、「行事曆」、「相簿管理」、「辨識分析系統」得到完善的使用者體驗；「iPets」用戶使用「寵物系統」功能，進行人性化的寵物管理，其中「寵物資訊」、「寵物行程管理」提供簡潔的使用者介面，方便使用者快速且完善的規劃行程；另外還提供「寵物品種辨識」、「圖片風格轉換」兩項功能讓用戶體驗人工智慧技術，增加會員的使用體驗。



#### 圖3-1-2 網頁架構

「iPets」APP架構如下圖3-1-3所示，用戶進入首頁可看到「寵物資訊」、「倒數計時器」，方便用戶快速規劃活動，下方選單列分別為「首頁」、「行事曆」、「相簿管理」、「定位」及「設定」；其中「定位」可顯示周遭五公里內與寵物有關的相關店家，方便用戶快速查詢附近店家。用戶透過登入後所有資訊便會儲存於後端資料庫。



#### 圖3-1-3 APP架構

## 3-2 系統軟硬體需求與技術平台

### 表3-2-1 環境測試表

|  |  |
| --- | --- |
| 測試環境 | |
| 處理器 | IntelCore i5 CPU 1.6GHZ |
| 瀏覽器測試 | Google Chrome、Firefox、Safari |
| 操作介面 | 非觸控、電阻式觸控面板 |
| 網路 | 有線網路、無線網路WIFI/3G或4G網路 |

### 表3-2-2 軟硬體需求表

|  |  |  |
| --- | --- | --- |
| 軟硬體需求 | | |
| 作業系統 | Windows、Mac OS | Android、iOS |
| 瀏覽器需求 | Google Chrome、Firefox、Safari | |
| 操作介面需求 | 非觸控、電阻式觸控面板 | |
| 網路需求 | 有線網路、無線網路WIFI/3G或4G網路 | |

## 3-3 使用標準與工具

### 表3-3-1 使用標準與工具表

|  |  |
| --- | --- |
| 開發工具 | Visual Studio Code、Android studio |
| 網頁前端 | 開發框架：Vue.js  開發語言：HTML5、CSS3、JavaScript  函式庫、套件：jQuery、Bootstrap4 |
| 網頁後端 | Python、Firebase |
| APP前端 | XML |
| APP後端 | Java |
| 資料庫 | Cloud Firestore |
| 美工 | Adobe Illustrator、Adobe PhotoShop |
| 文件 | Microsoft Word |
| 簡報 | Microsoft PowerPoint |
| 專案管理 | GitHub |
| 版本控制 | GitHub |
| 系統架構 | Visual Paradigm |
| 系統分析、設計 | Visual Paradigm |

# 第四章 專案時程與組織分工

## 4-1 專案時程

### 表4-1-1 專案時程表

：預期進度

：實際進度

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 時間  事項 | 2020年 | | | | | | | | | | | | | | | | | | | | | |
| 1月 | | 2月 | | 3月 | | 4月 | | 5月 | | 6月 | | 7月 | | 8月 | | 9月 | | 10月 | | 11月 | |
| 題目構思 |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 技術學習 |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 系統功能分析 |  | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prototype設計 |  | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UI/UX設計 |  |  |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 資料庫設計 |  | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 資料庫建置 |  |  |  |  |  | | | | | |  |  |  |  |  |  |  |  | | | |  |
|  |  |  |  |  | | | | |  |  |  |  |  |  |  |  |  | | | |  |
| 前端程式碼撰寫 |  |  |  |  |  | | | | | | | | | | | | | | | | | |
|  |  |  |  |  | | | | | | | | | | | | | | | | | |
| 後端程式碼撰寫 |  |  |  |  |  | | | | | | | | | | | | | | | | | |
|  |  |  |  |  | | | | | | | | | | | | | | | | | |
| APP程式碼撰寫 |  |  |  |  |  | | | | | | | | | | | | | | | | | |
|  |  |  |  |  | | | | | | | | | | | | | | | | | |
| 文件製作 |  |  |  |  |  |  |  | | | |  |  |  |  |  |  |  |  |  |  | | |
|  |  |  |  |  |  |  | | | |  |  |  |  |  |  |  |  |  |  | | |
| 簡報製作 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| 影片製作 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| 海報製作 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |

## 4-2 專案組織與分工

### 表4-1-2 分工表

* ：主要負責 ○：次要負責

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 時間  事項 | 李怡蓁 | 蔡欣恬 | 黃琳恩 | 王偉霖 | 劉芳汝 |
| 題目構思 | ● | ● |  |  |  |
| 技術學習 | ● | ● | ● | ● | ● |
| 系統功能分析 | ● | ● |  |  |  |
| Prototype設計 | ● | ● |  |  |  |
| UI/UX設計 | ● | ○ |  |  | ● |
| 資料庫設計 | ● | ● |  | ● |  |
| 資料庫建置 | ● | ● | ○ | ● |  |
| Web前端程式碼撰寫 | ● | ○ |  |  |  |
| Web後端程式碼撰寫 | ● | ● |  |  |  |
| APP前端程式碼撰寫 |  |  | ○ | ○ | ● |
| APP後端程式碼撰寫 |  |  | ● | ● |  |
| 文件製作 | ● | ● | ○ | ○ | ○ |
| 簡報製作 | ● | ● |  |  |  |
| 海報製作 |  |  |  |  | ● |
| 影片製作 |  |  | ● |  |  |

# 第五章 需求模型

## 5-1 使用者需求

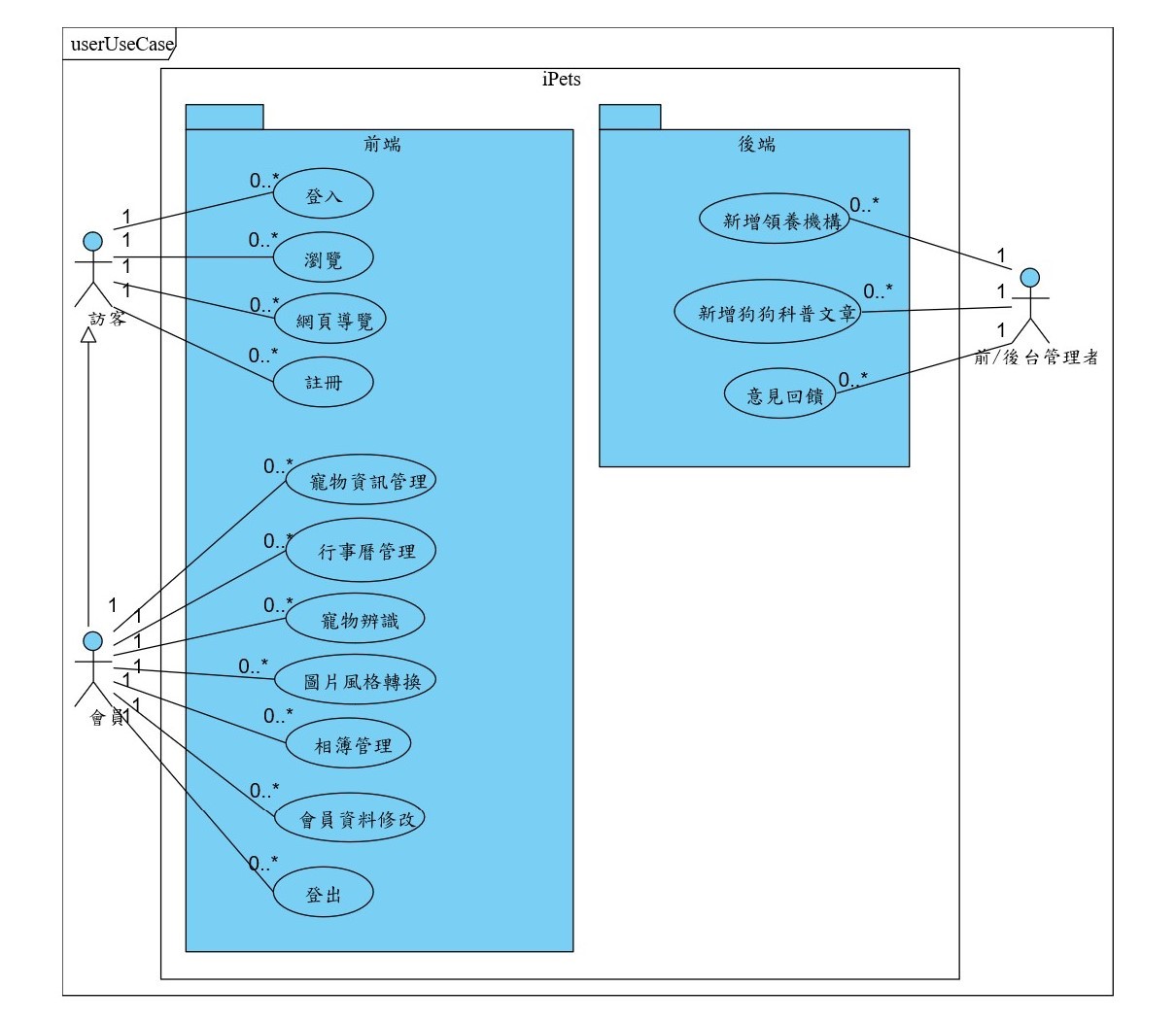
### 表5-1-1 功能需求表

|  |  |
| --- | --- |
| 項目 | 說明 |
| 註冊會員 | 使用者透過系統註冊成為會員。 |
| 登入 | 原有用戶透過系統登入。 |
| 用戶資料修改 | 方便用戶自行修改個人資料。 |
| 寵物領養機構 | 顯示公/私立的領養機構資訊，並提供使用者、用戶領養管道。 |
| 提醒 | 用戶可以開啟提醒功能，提醒用戶某項行程即將到來。 |
| 網站導覽 | 使用者可以快速了解網站特色。 |
| 查看文章 | 讓使用者可以了解寵物的最新文章，提升註冊意願。 |
| 寵物資料管理 | 方便用戶自行管理寵物資料。 |
| 倒數計時器 | 利用簡潔的介面，進行時間倒數，方便用戶快速安排當日行程。 |
| 行事曆排程 | 方便用戶規劃寵物的所有行程。 |
| 寵物品種辨識 | 用戶可以上傳圖片進行寵物品種辨識，增加使用者體驗。 |
| 圖片風格轉換 | 用戶可以上傳圖片進行圖片風格轉換，增加使用者體驗。 |
| 定位 | 開啟定位功能可查看周遭近五公里的寵物店家、寵物醫院、寵物公園。 |
| 搜尋店家 | 搜尋附近與寵物有關的店家，使用戶快速找到相對應的地點。 |
| 建立相簿 | 用戶可以建立專屬自己的相簿存放寵物的照片。 |
| 常見問題 | 將用戶常見問題設計成QA，降低使用者使用難度。 |
| 意見回饋 | 當使用者使用本平台遇到任何問題或者需要改善處，可以透過意見回饋提供給系統管理者。 |

### 表5-1-2 非功能性需求表

|  |  |
| --- | --- |
| 項目 | 說明 |
| 維護性 | 收到修改建議，對於普通修改會在1-2天完成，對於評估後為重大需求或設計需求修改在兩週內完成。 |
| 相容性 | 支援各種螢幕尺寸，且在任何瀏覽器皆可使用。 |
| 易使用性 | 系統操作介面簡單且容易使用。 |
| 安全性 | 註冊用戶時，用戶密碼透過Firebase加密。 |
| 兼容性 | 本系統支持多種作業系統使用。 |

## 5-2 使用個案圖（Use case diagram）

****

#### 圖5-2-1 使用者個案圖

## 5-3 使用個案描述

|  |  |
| --- | --- |
|  |  |
| 圖5-3-1 註冊會員活動圖 | 圖5-3-2 使用者登入活動圖 |
|  |  |
|  |  |
| 圖5-3-3 修改會員資料活動圖 |  |

|  |
| --- |
|  |
| 圖5-3-4 行事曆管理活動圖 |
|  |
| 圖5-3-5 相簿管理活動圖 |

|  |
| --- |
|  |
| 圖5-3-6 寵物資訊管理活動圖 |
|  |
| 圖5-3-7 寵物品種辨識活動圖 |

|  |
| --- |
|  |
| 圖5-3-8 圖片風格轉換活動圖 |
|  |
| 圖5-3-9 地圖活動圖 |

|  |  |
| --- | --- |
|  |  |
| 圖5-3-10 新增領養機構活動圖 | 圖5-3-11 新增狗狗科普活動圖 |
|  |  | |
|  |  | |
| 圖5-3-12 回饋資訊活動圖 |  | |

## 5-4 分析類別圖

|  |
| --- |
|  |
| 圖5-4-1 （Web）會員分析類別圖 |
|  |
| 圖5-4-2 （Web）行事曆分析類別圖 |

|  |
| --- |
|  |
| 圖5-4-3 （Web）深度學習分析類別圖 |
|  |
| 圖5-4-4 （APP）會員分析類別圖 |

|  |
| --- |
|  |
| 圖5-4-5 （APP）行事曆分析類別圖 |
|  |
| 圖5-4-6 （APP）定位分析類別圖 |

# 第六章 設計模型

## 6-1 循序圖（Sequence diagram）

|  |
| --- |
|  |
| 圖6-1-1 註冊循序圖 |

|  |
| --- |
|  |
| 圖6-1-2 登入循序圖 |
|  |
| 圖6-1-3 修改會員資料循序圖 |

|  |
| --- |
|  |
| 圖6-1-4 相簿管理循序圖 |
|  |
| 圖6-1-5 行事曆管理循序圖 |

|  |
| --- |
|  |
| 圖6-1-6 寵物資訊管理循序圖 |
|  |
| 圖6-1-7 寵物品種辨識循序圖 |

|  |
| --- |
|  |
| 圖6-1-8 圖片風格轉換循序圖 |
|  |
| 圖6-1-9 查詢寵物醫院循序圖 |

|  |
| --- |
|  |
| 圖6-1-10 查詢附近店家循序圖 |

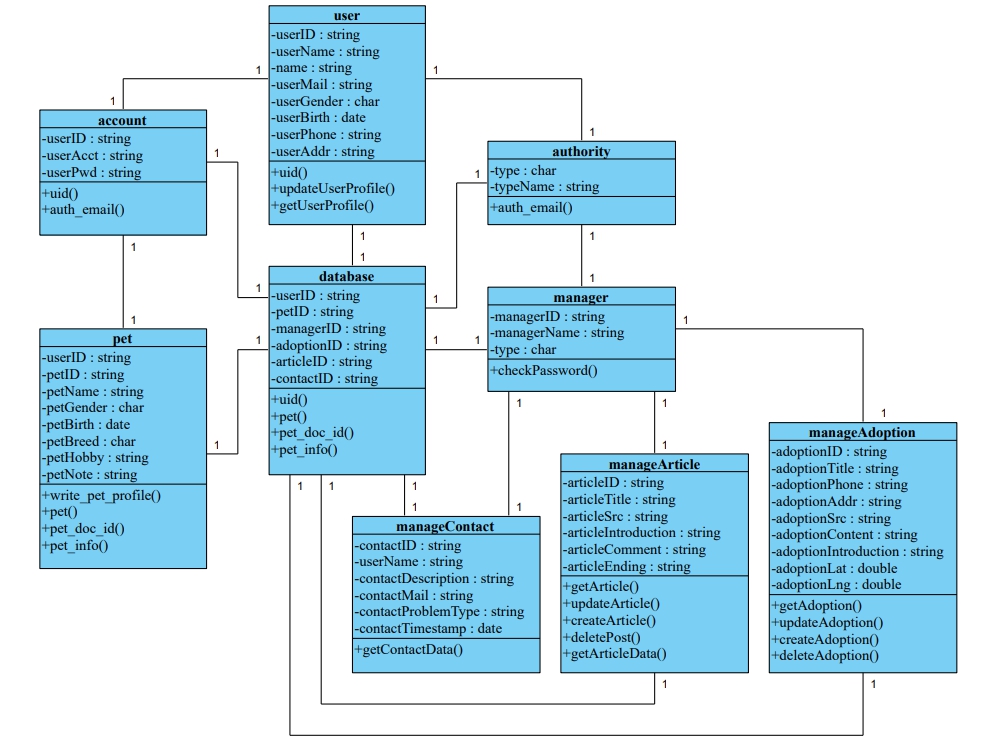
|  |
| --- |
|  |
| 圖6-1-11 查詢公園循序圖 |
|  |
| 圖6-1-12 領養機構循序圖 |

|  |
| --- |
|  |
| 圖6-1-13 狗狗科普循序圖 |
|  |
| 圖6-1-14 回饋管理循序圖 |

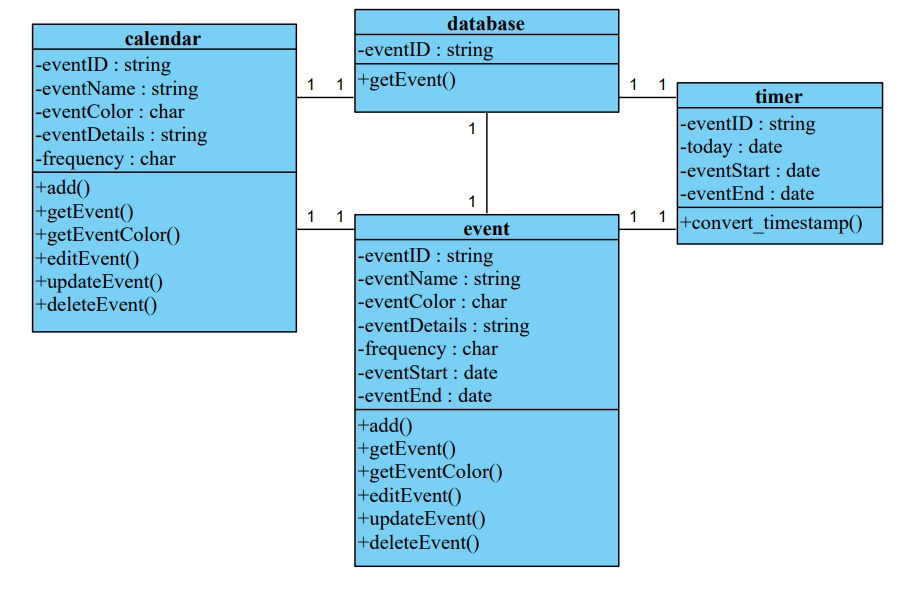
|  |
| --- |
|  |
| 圖6-1-15 回饋循序圖 |
|  |
| 圖6-1-16 狗狗科普管理循序圖 |

|  |
| --- |
|  |
| 圖6-1-17 領養機構管理循序圖 |

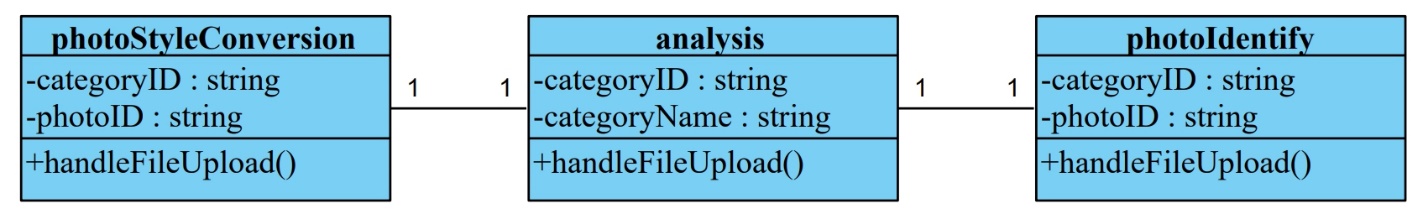
## 6-2 設計類別圖（Design class diagram）



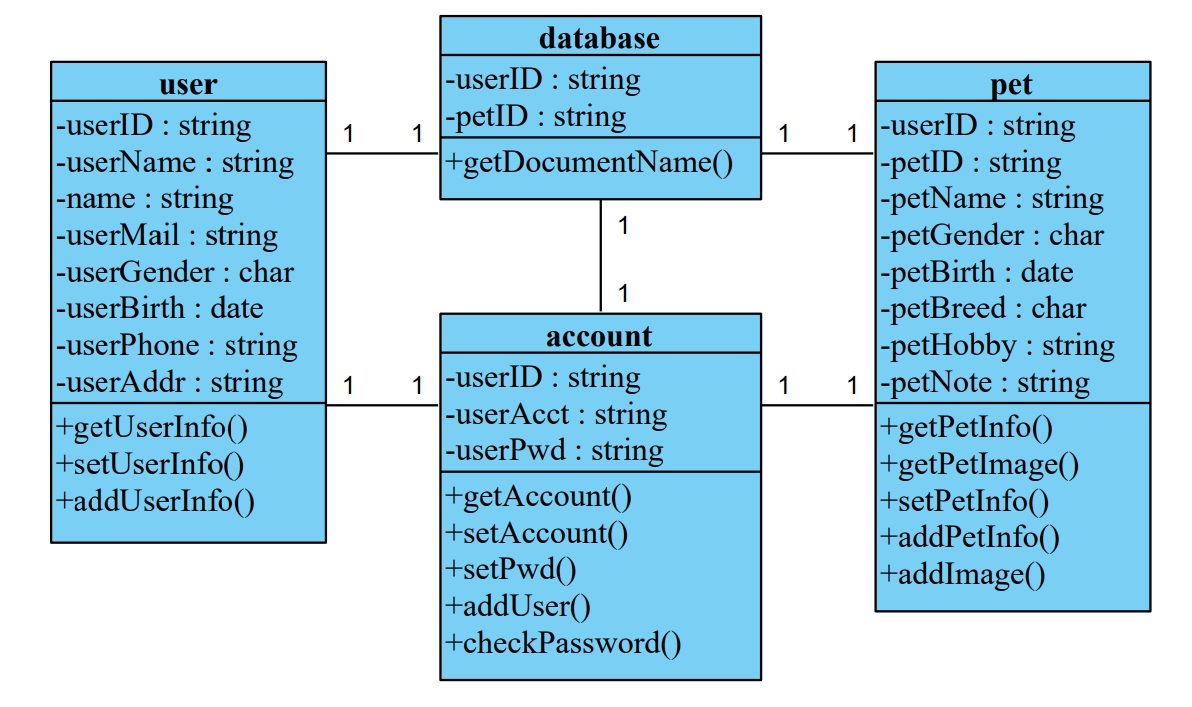
#### 圖6-2-1 （Web）會員設計類別圖



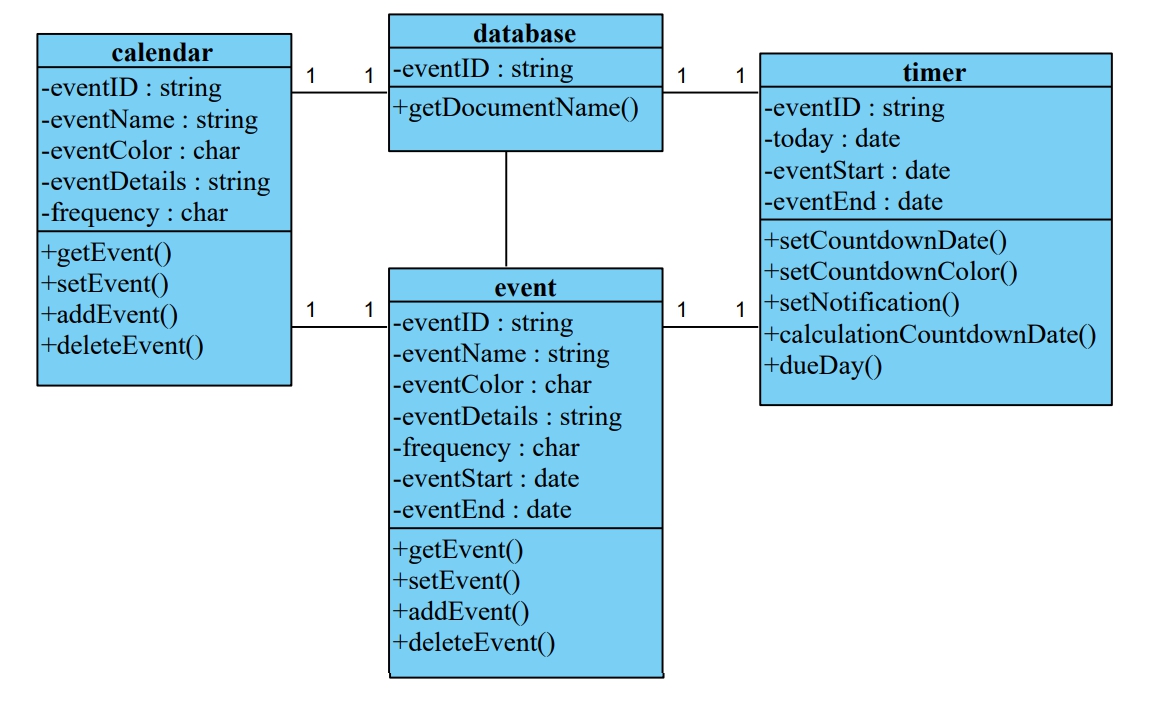
#### 圖6-2-2 （Web）行事曆設計類別圖



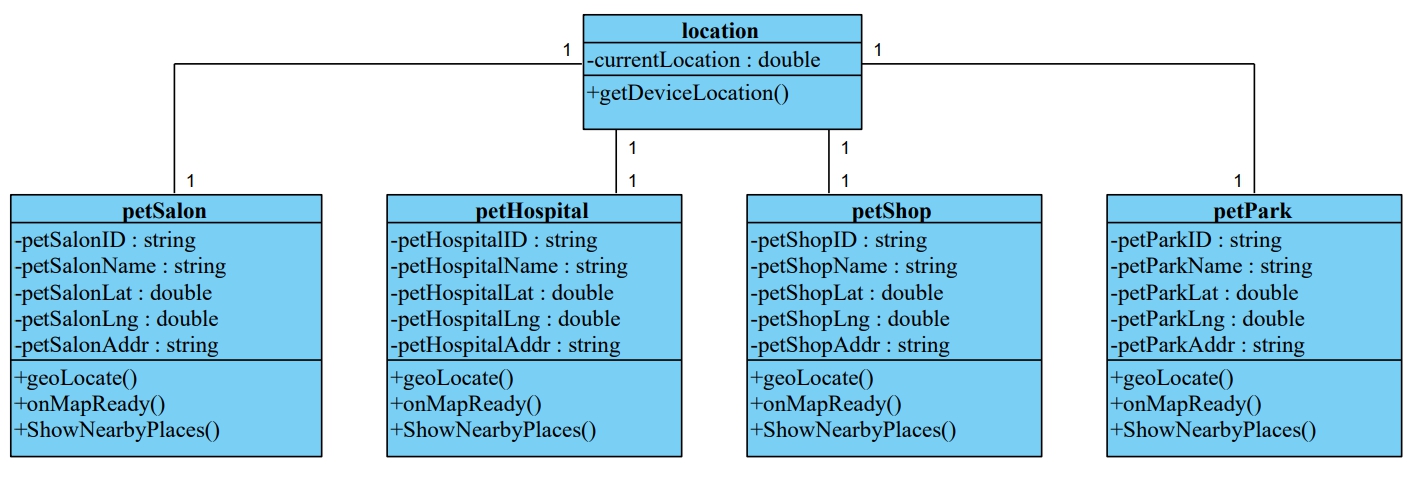
#### 圖6-2-3 （Web）深度學習設計類別圖



#### 圖6-2-4 （APP）會員設計類別圖

****

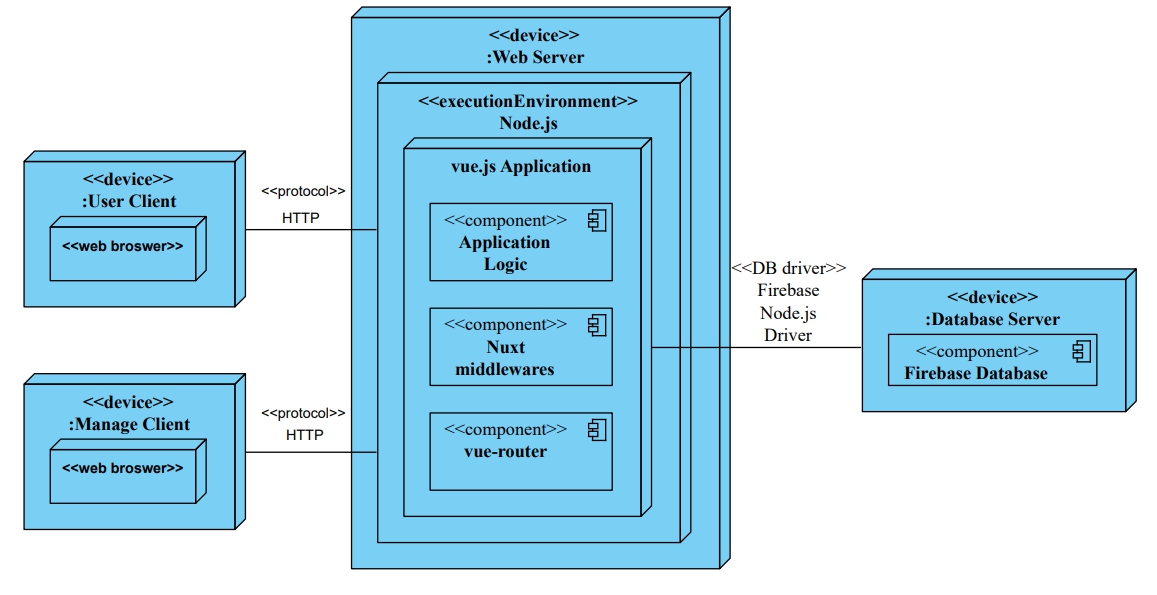
#### 圖6-2-5 （APP）行事曆設計類別圖

****

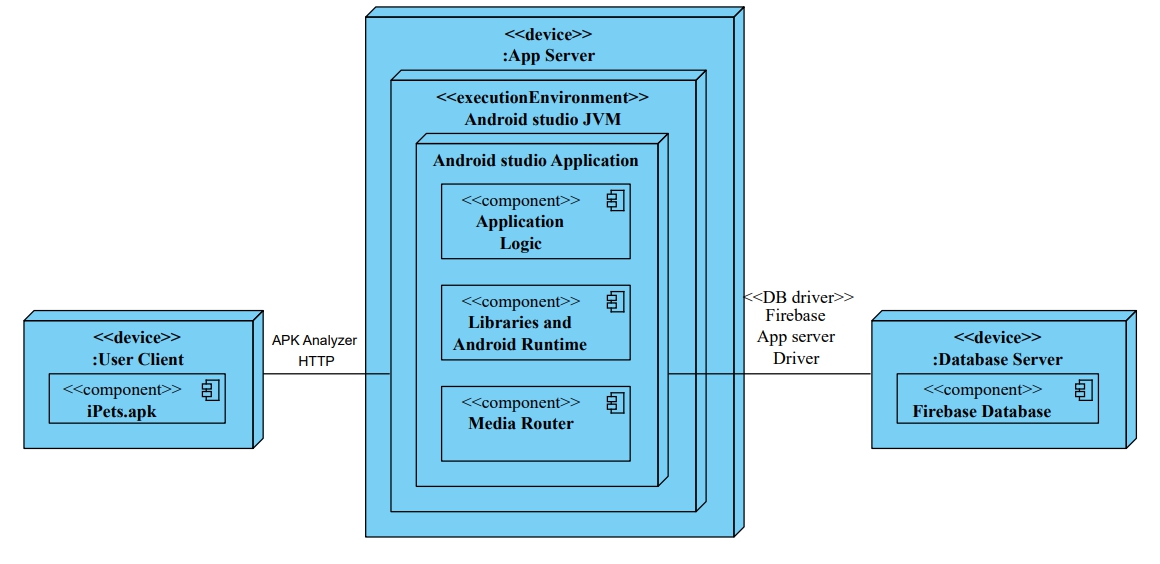
#### 圖6-2-6 （APP）定位設計類別圖

# 第七章 實作模型

## 7-1 佈署圖（Deployment diagram）

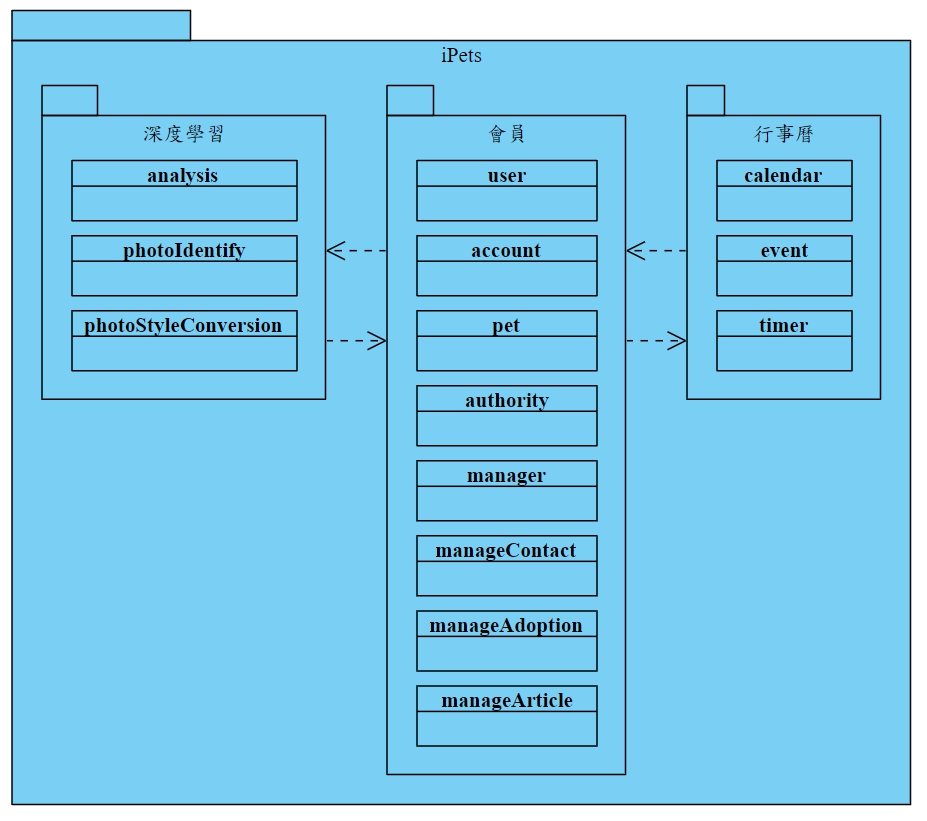


#### 圖7-1-1 （Web）佈署圖

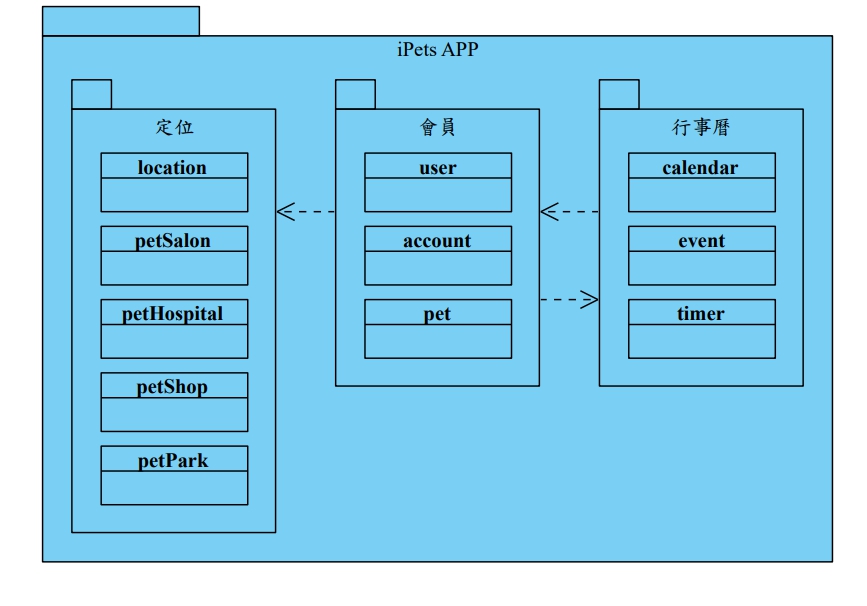


#### 圖7-1-2 （APP）佈署圖

## 7-2 套件圖（Package diagram）

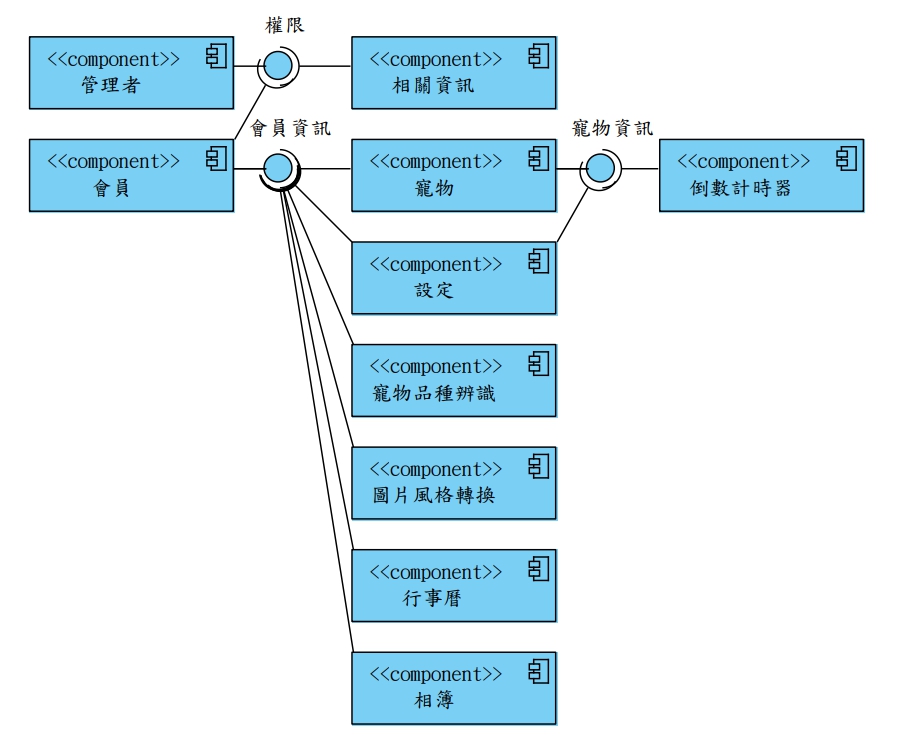


#### 圖7-2-1 （Web）套件圖



#### 圖7-2-2 （APP）套件圖

## 7-3 元件圖（Component diagram）

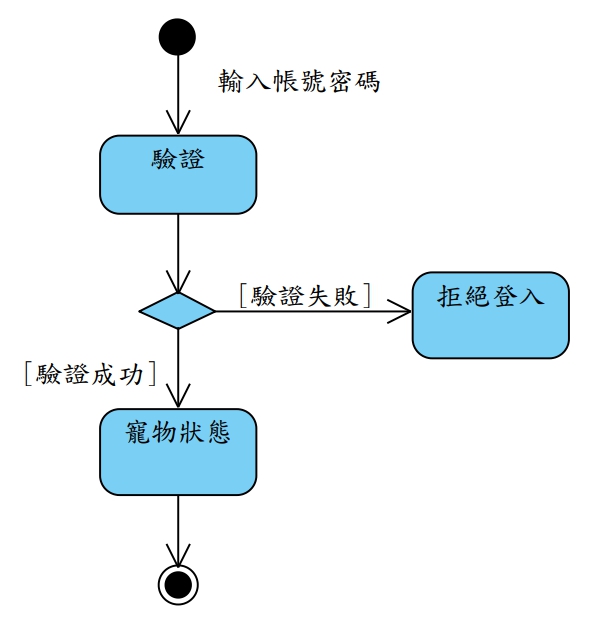


#### 圖7-3-1 （Web）元件圖



#### 圖7-3-2 （APP）元件圖

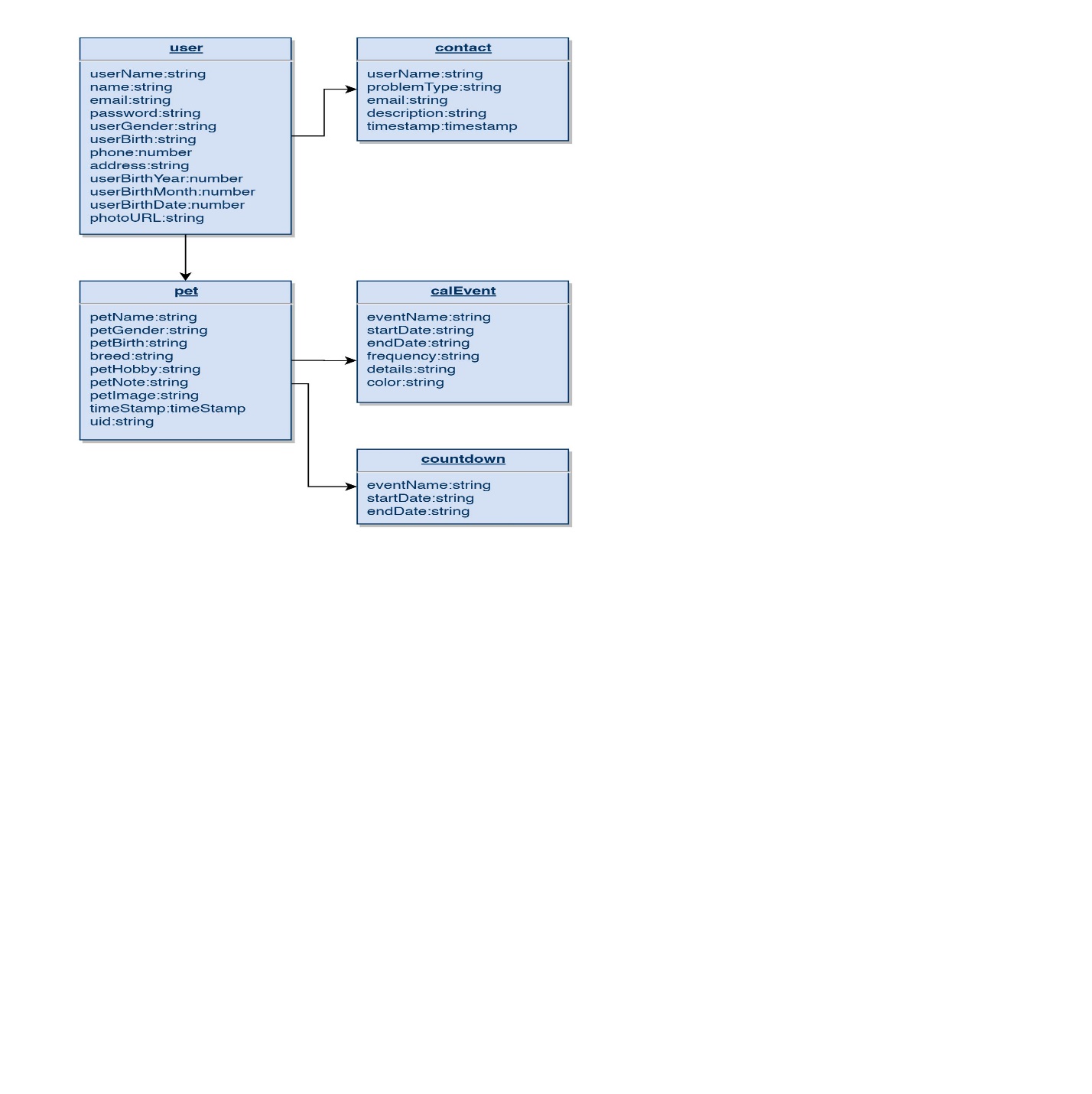
## 7-4 狀態機（State machine）



#### 圖7-4-1 狀態機

# 第八章 資料庫設計

## 8-1 資料庫關聯表



#### 圖8-1-1 資料庫關聯表

## 8-2 表格及其 Meta data

### 表8-2-1 資料表描述：用戶資料

|  |  |  |
| --- | --- | --- |
| 文件名稱：user | | |
| 欄位名稱 | 中文名稱 | 資料類型 |
| userName | 用戶名稱 | string |
| name | 姓名 | string |
| email | 電子郵件 | string |
| password | 密碼 | string |
| userGender | 用戶性別 | string |
| userBirth | 用戶生日 | string |
| phone | 電話 | number |
| address | 住址 | string |
| userBirthYear | 用戶生日年份 | number |
| userBirthMonth | 用戶生日月份 | number |
| userBirthDate | 用戶生日日期 | number |
| photoURL | 用戶相片位置 | string |

### 表8-2-2 資料表描述：行事曆行程

|  |  |  |
| --- | --- | --- |
| 文件名稱：calEvent | | |
| 欄位名稱 | 中文名稱 | 資料類型 |
| eventName | 事件名稱 | string |
| startDate | 事件開始日期 | string |
| endDate | 事件結束日期 | string |
| frequency | 循環 | string |
| details | 細節說明 | string |
| color | 顏色 | string |

### 表8-2-3 資料表描述：問題反饋

|  |  |  |
| --- | --- | --- |
| 文件名稱：contact | | |
| 欄位名稱 | 中文名稱 | 資料類型 |
| userName | 使用者名稱 | string |
| problemType | 問題型態 | string |
| email | 信箱 | string |
| description | 描述 | string |
| timestamp | 時間戳記 | timestamp |

### 表8-2-4 資料表描述：寵物資料

|  |  |  |
| --- | --- | --- |
| 文件名稱：pet | | |
| 欄位名稱 | 中文名稱 | 資料類型 |
| petName | 寵物名稱 | string |
| petGender | 寵物性別 | string |
| petBirth | 寵物生日 | string |
| breed | 寵物品種 | string |
| petHobby | 寵物喜好 | string |
| petNote | 寵物備註 | string |
| petImage | 寵物相片位置 | string |
| timeStamp | 時間戳記 | timestamp |
| uid | 用戶ID | string |

### 表8-2-5 資料表描述：倒數計時器

|  |  |  |
| --- | --- | --- |
| 文件名稱：countdown | | |
| 欄位名稱 | 中文名稱 | 資料類型 |
| eventName | 事件名稱 | string |
| startDate | 事件開始日期 | string |
| endDate | 事件結束日期 | string |

# 第九章 程式

## 9-1 元件清單及其規格描述

### 表9-1-1 首頁功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/home.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-1-1 | geolocate | 取得經緯度 |
| 1-1-2 | convertCenter | 轉換經緯度 |
| 1-1-3 | getAdoption | 取得領養機構資訊 |

### 表9-1-2 關於我們功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/about.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-2-1 | getMember | 取得組員資訊 |

### 表9-1-3 狗狗科普功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/dogScience.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-3-1 | getArticle | 取得文章資訊 |
| 1-3-2 | reverse | 排序1-3-1結果 |

### 表9-1-4 聯繫我們功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/contact.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-4-1 | addContact | 新增使用者回饋 |

### 表9-1-5 登入功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/login.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-5-1 | auth\_email | 檢查註冊資訊 |
| 1-5-2 | hidePassword | 隱藏密碼 |
| 1-5-3 | showPassword | 顯示密碼 |

### 表9-1-6 註冊功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/register.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-6-1 | auth\_email | 檢查註冊資訊 |
| 1-6-2 | hidePassword | 隱藏密碼 |
| 1-6-3 | showPassword | 顯示密碼 |
| 1-6-4 | hideCheckPassword | 隱藏確認密碼 |
| 1-6-5 | showCheckPassword | 顯示確認密碼 |

### 表9-1-7 寵物資訊功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/petProfile.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-7-1 | handleFileUpload | 取得檔案資料 |
| 1-7-2 | createAlbum | 建立相簿並新增頭貼 |
| 1-7-3 | save\_pet\_profile | 建立寵物資訊 |
| 1-7-4 | onSubmit | 下載頭貼路徑 |
| 1-7-5 | convert\_timestamp | 轉換時間 |
| 1-7-6 | badge\_today | 判斷行程時間 |
| 1-7-7 | badge\_overdue | 判斷行程是否逾期 |
| 1-7-8 | calculateAge | 年齡計算 |
| 1-7-9 | edit\_profile\_birth | 更新寵物年齡 |
| 1-7-10 | edit\_pet\_ok | 更新寵物資訊 |
| 1-7-11 | getProfilePicture | 取得寵物頭貼 |

### 表9-1-8 行事曆功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/calendar.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-8-1 | add | 新增事件 |
| 1-8-2 | getEvents | 取得事件 |
| 1-8-3 | setDialogDate | 點取日期新增事件 |
| 1-8-4 | closeDialog | 關閉新增事件 |
| 1-8-5 | viewDay | 行事曆顯示類型 |
| 1-8-6 | getEventColor | 取得事件顏色 |
| 1-8-7 | setToday | 返回當日 |
| 1-8-8 | prev | 顯示上個月/週/日/四天 |
| 1-8-9 | next | 顯示下個月/週/日/四天 |
| 1-8-10 | addEvent | 判斷頻率並新增事件 |
| 1-8-11 | editEvent | 取得編輯事件 |
| 1-8-12 | updateEvent | 編輯事件 |
| 1-8-13 | deleteEvent | 刪除事件 |
| 1-8-14 | showEvent | 查看事件 |
| 1-8-15 | updateRange | 取得事件起訖時間 |
| 1-8-16 | title | 行事曆顯示時間範圍 |

### 表9-1-9 相簿功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/album.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-9-1 | handleFileUpload | 取得檔案資料 |
| 1-9-2 | createAlbum | 建立相簿 |
| 1-9-3 | getAlbumData | 取得相簿資訊 |

### 表9-1-10 相簿瀏覽功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/albumView.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-10-1 | handleFileUpload | 取得檔案資料 |
| 1-10-2 | createPhoto | 新增相片 |
| 1-10-3 | deleteImage | 刪除相片 |
| 1-10-4 | getPhoto | 取得單一相簿圖片 |
| 1-10-5 | downloadImage | 下載圖片 |

### 表9-1-11 寵物品種辨識功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/identify.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-11-1 | handleFileUpload | 取得圖片資訊 |

### 表9-1-12 風格轉換功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/tunit.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-12-1 | handleFileUpload | 取得圖片資訊 |

### 表9-1-13 新增寵物功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/addPetProfile.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-13-1 | handleFileUpload | 取得上傳圖片 |
| 1-13-2 | save\_pet\_profile | 建立寵物資訊 |
| 1-13-3 | onSubmit | 下載頭貼路徑 |

### 表9-1-14 文章功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/post.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-14-1 | $route | 監聽網址 |
| 1-14-2 | getArticleData | 取得單一文章資料 |

### 表9-1-15 管理文章功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/manageArticle.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-15-1 | deletePost | 刪除文章 |
| 1-15-2 | addRow | 新增內容列 |
| 1-15-3 | deleteRow | 刪除內容列 |
| 1-15-4 | cleanData | 清除文章暫存資料 |
| 1-15-5 | putData | 顯示文章內容 |
| 1-15-6 | createArticle | 新增文章 |
| 1-15-7 | updateArticle | 編輯文章 |
| 1-15-8 | getArticle | 取得文章 |

### 表9-1-16 管理領養機構功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/manageAdoption.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-16-1 | deleteAdoption | 刪除領養機構資訊 |
| 1-16-2 | cleanData | 清除領養機構暫存資料 |
| 1-16-3 | createAdoption | 新增領養機構 |
| 1-16-4 | updateAdoption | 編輯領養機構 |
| 1-16-5 | getAdoption | 取得領養機構資訊 |

### 表9-1-17 管理回饋功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/manageContact.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-17-1 | getContactData | 取得回饋資訊 |

### 表9-1-18 Navbar功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/navbar.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-18-1 | logout | 確認用戶登入狀態 |

### 表9-1-19 會員資料功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/firebase/user.js | | |
| 編號 | 函式名稱 | 功能 |
| 1-19-1 | updateUserProfile | 取得會員資料 |
| 1-19-2 | getUserPetsList | 取得寵物清單 |
| 1-19-3 | updateMonthCalendar | 更新行程列表 |

### 表9-1-20 寵物資料功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/pet.js | | |
| 編號 | 函式名稱 | 功能 |
| 1-20-1 | updatePetInfo | 取得寵物資料 |

### 表9-1-21 編輯個人資料功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/editProfile.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-21-1 | getUserProfile | 取得個人資訊 |
| 1-21-2 | updateProfile | 更新個人資訊 |

### 表9-1-22 更改密碼功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/password.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-22-1 | updatePassword | 更新密碼 |
| 1-22-2 | checkPassword | 確認重複密碼 |
| 1-22-3 | hideCurrentPassword | 隱藏目前密碼 |
| 1-22-4 | showCurrentPassword | 顯示目前密碼 |
| 1-22-5 | hideNewPassword | 隱藏新密碼 |
| 1-22-6 | showNewPassword | 顯示新密碼 |
| 1-22-7 | hideConfirmNewPassword | 隱藏確認密碼 |
| 1-22-8 | showConfirmNewPassword | 顯示確認密碼 |

### 表9-1-23 管理者資料功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱：src/components/Manage.vue | | |
| 編號 | 函式名稱 | 功能 |
| 1-23-1 | checkPassword | 取得個人資訊 |

### 表9-1-24 取得經緯度

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-1 | 函式名稱 | geolocate |
| 功能 | 取得經緯度 | | |
| geolocate: function() {  navigator.geolocation.getCurrentPosition(position => {  this.center = {  lat: position.coords.latitude,  lng: position.coords.longitude  };  });  } | | | |

### 表9-1-25 轉換經緯度

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-2 | 函式名稱 | convertCenter |
| 功能 | 轉換經緯度 | | |
| convertCenter(i) {  let target = this.comments[i];  return {  lat: parseFloat(target.center.lat),  lng: parseFloat(target.center.lng)  };  } | | | |

### 表9-1-26 取得領養機構資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-3 | 函式名稱 | getAdoption |
| 功能 | 取得領養機構資訊 | | |
| getAdoption() {  axios.get("http://localhost:4000/comments").then(res => {  console.log(res);  this.comments = res.data;  });  } | | | |

### 表9-1-27 取得組員資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-2-1 | 函式名稱 | getMember |
| 功能 | 取得組員資訊 | | |
| getMember() {  axios.get("/static/member.json").then(response => {  this.post = response.data;  });  } | | | |

### 表9-1-28 取得文章資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3-1 | 函式名稱 | getArticle |
| 功能 | 取得文章資訊 | | |
| getAricle() {  axios.get("http://localhost:3000/comments").then(res => {  console.log(res);  this.comments = res.data;  });  } | | | |

### 表9-1-29 排序1-3-1結果

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3-2 | 函式名稱 | reverse |
| 功能 | 排序1-3-1結果 | | |
| reverse() {  this.comments.reverse();  } | | | |

### 表9-1-30 新增使用者回饋

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-4-1 | 函式名稱 | addContact |
| 功能 | 新增使用者回饋 | | |
| addContact(e) {  e.preventDefault();  var docRef = fStore.collection("contact").doc();  docRef  .set({  userName: this.userName,  email: this.email,  problemType: this.problemType,  description: this.description,  timestamp: new Date()  })  .then(() => {  this.$router.go({ path: this.$router.path });  })  .finally(() => {  alert("您的建議已經送出囉~!");  });  } | | | |

### 表9-1-31 檢查註冊資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-1 | 函式名稱 | auth\_email |
| 功能 | 檢查註冊資訊 | | |
| auth\_email: function(e) {  fAuth  .signInWithEmailAndPassword(this.email, this.password)  .then(userCredential => {  this.$router.go({ path: this.$router.path });  })  .catch(error => {  alert(error.code);  alert(error.message);  })  .finally(() => {  console.log("登入成功");  });  e.preventDefault();  } | | | |

### 表9-1-32 隱藏密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-2 | 函式名稱 | hidePassword |
| 功能 | 隱藏密碼 | | |
| hidePassword() {  this.passwordHidden = false;  this.passwordFieldType =  this.passwordFieldType === "password" ? "text" : "password";  } | | | |

### 表9-1-33 顯示密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-5-3 | 函式名稱 | showPassword |
| 功能 | 顯示密碼 | | |
| showPassword() {  this.passwordHidden = true;  this.passwordFieldType = this.passwordFieldType;  } | | | |

### 表9-1-34 檢查註冊資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-6-1 | 函式名稱 | auth\_email |
| 功能 | 檢查註冊資訊 | | |
| auth\_email: function(e) {  fAuth  .createUserWithEmailAndPassword(this.Email, this.password)  .then(userCredential => {  var user = userCredential.user;  fStore  .collection("users")  .doc(user.uid)  .set({  userName: this.userName,  name: this.name,  Email: this.Email,  password: this.password,  userGender: this.userGender,  userBirth:  this.userBirth.getUTCFullYear() +  "-" +  (this.userBirth.getMonth() + 1) +  "-" +  this.userBirth.getDate(),  userBirth\_year: this.userBirth.getUTCFullYear(),  userBirth\_month: this.userBirth.getMonth() + 1,  userBirth\_date: this.userBirth.getDate(),  address: this.address,  phone: this.phone,  photoURL: ""  })  .then(() => {  this.$router.go({ path: this.$router.path });  });  })  .catch(error => {  alert(error.message);  })  .finally(() => {  alert("歡迎加入iPets的大家庭~!");  });  e.preventDefault();  } | | | |

### 表9-1-35 隱藏密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-6-2 | 函式名稱 | hidePassword |
| 功能 | 隱藏密碼 | | |
| hidePassword() {  this.passwordHidden = false;  this.passwordFieldType =  this.passwordFieldType === "password" ? "text" : "password";  } | | | |

### 表9-1-36 顯示密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-6-3 | 函式名稱 | showPassword |
| 功能 | 顯示密碼 | | |
| showPassword() {  this.passwordHidden = true;  this.passwordFieldType = this.passwordFieldType;  } | | | |

### 表9-1-37 隱藏確認密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-6-4 | 函式名稱 | hideCheckPassword |
| 功能 | 隱藏確認密碼 | | |
| hideCheckPassword() {  this.passwordHidden2 = false;  this.passwordFieldType =  this.passwordFieldType === "password" ? "text" : "password";  } | | | |

### 表9-1-38 顯示確認密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-6-5 | 函式名稱 | showCheckPassword |
| 功能 | 顯示確認密碼 | | |
| showCheckPassword() {  this.passwordHidden2 = true;  this.passwordFieldType = this.passwordFieldType;  } | | | |

### 表9-1-39 取得檔案資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-1 | 函式名稱 | handleFileUpload |
| 功能 | 取得檔案資料 | | |
| handleFileUpload(e) {  this.petImage = e.target.files[0];  console.log(this.petImage);  } | | | |

### 表9-1-40 建立相簿並新增頭貼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-2 | 函式名稱 | createAlbum |
| 功能 | 建立相簿並新增頭貼 | | |
| createAlbum() {  var storageRef = firebase  .storage()  .ref(this.uid + "/" + this.pet.petName);  storageRef.put(this.petImage).then(function(snapshot) {  console.log("Uploaded files!");  });  } | | | |

### 表9-1-41 建立寵物資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-3 | 函式名稱 | save\_pet\_profile |
| 功能 | 建立寵物資訊 | | |
| async save\_pet\_profile(file) {  await fStore  .collection("users")  .doc(this.uid)  .collection("pets")  .add({  petName: this.petName,  petGender: this.petGender,  petBirth:  this.petBirth.getUTCFullYear() +  "-" +  (this.petBirth.getMonth() + 1) +  "-" +  this.petBirth.getDate(),  petBirth\_year: this.petBirth.getUTCFullYear(),  petBirth\_month: this.petBirth.getMonth() + 1,  petBirth\_date: this.petBirth.getDate(),  breed: this.breed,  petHobby: this.petHobby,  petNote: this.petNote,  uid: this.uid,  timestamp: new Date(),  profile\_picture: ""  })  .then(function(ref) {  return firebase  .storage()  .ref(firebase.auth().currentUser.uid + "/" + ref.id)  .put(file)  .then(function(fileSnapshot) {  return fileSnapshot.ref.getDownloadURL().then(function(url) {  return ref.update({  profile\_picture: url  });  });  });  })  .catch(function(error) {  console.error(  "There was an error uploading a file to Cloud Storage:",  error  );  });  this.$router.go({ path: this.$router.path });  } | | | |

### 表9-1-42 下載頭貼路徑

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-4 | 函式名稱 | onSubmit |
| 功能 | 下載頭貼路徑 | | |
| onSubmit(e) {  e.preventDefault();  this.save\_pet\_profile(this.petImage);  } | | | |

### 表9-1-43 轉換時間

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-5 | 函式名稱 | convert\_timestamp |
| 功能 | 轉換時間 | | |
| convert\_timestamp(unixTimestamp) {  var date = unixTimestamp.toDate();  var year = date.getFullYear();  var month = date.getMonth() + 1;  var day = date.getDate();  var hour = date.getHours();  var min = date.getMinutes();  var sec = date.getSeconds();  if (month < 10) month = "0" + month;  if (day < 10) day = "0" + day;  if (hour < 10) hour = "0" + hour;  if (min < 10) min = "0" + min;  if (sec < 10) sec = "0" + sec;  var formattedTime =  [year, month, day].join("-") + " " + hour + ":" + min + ":" + sec;  return formattedTime;  } | | | |

### 表9-1-44 判斷行程時間

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-6 | 函式名稱 | badge\_today |
| 功能 | 判斷行程時間 | | |
| badge\_today(unixTimestamp) {  var date = unixTimestamp.toDate();  var today = new Date();  if (date.toDateString() === today.toDateString()) {  return true;  } else {  return false;  } | | | |

### 表9-1-45 判斷行程是否逾期

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-7 | 函式名稱 | badge\_overdue |
| 功能 | 判斷行程是否逾期 | | |
| badge\_overdue(unixTimestamp) {  var date = unixTimestamp.toDate();  var today = new Date();  date = new Date(date.toDateString());  today = new Date(today.toDateString());  if (today > date) {  return true;  } else {  return false;  }  } | | | |

### 表9-1-46 年齡計算

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-8 | 函式名稱 | calculateAge |
| 功能 | 年齡計算 | | |
| calculateAge(birth) {  var ageDiff = new Date(Date.now() - new Date(birth).getTime());  return Math.abs(ageDiff.getUTCFullYear() - 1970);  } | | | |

### 表9-1-47 更新寵物年齡

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-9 | 函式名稱 | edit\_profile\_birth |
| 功能 | 更新寵物年齡 | | |
| edit\_profile\_birth: {  get() {  return new Date(this.pet.petBirth);  },  set(newDate) {  newDate.setDate(newDate.getDate() + 1);  this.pet.petBirth = newDate.toISOString().slice(0, 10);  }  } | | | |

### 表9-1-48 更新寵物資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-10 | 函式名稱 | edit\_pet\_ok |
| 功能 | 更新寵物資訊 | | |
| edit\_pet\_ok(e) {  e.preventDefault();  fStore  .collection("users")  .doc(this.uid)  .collection("pets")  .doc(this.pet\_doc\_id)  .update({  petName: this.pet.petName,  petGender: this.pet.petGender,  petBirth: this.pet.petBirth,  petHobby: this.pet.petHobby,  petNote: this.pet.petNote,  timestamp: new Date()  });  this.$nextTick(() => {  this.$bvModal.hide("modal-prevent-closing");  });  if (this.petImage) {  this.createAlbum();  this.getProfilePicture();  }  setTimeout(function() {  location.reload();  }, 2000);  } | | | |

### 表9-1-49 取得寵物頭貼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-7-11 | 函式名稱 | getProfilePicture |
| 功能 | 取得寵物頭貼 | | |
| getProfilePicture() {  let uid = firebase.auth().currentUser.uid;  var storageRef = firebase.storage().ref(uid + "/" + this.pet.petName);  console.log(storageRef);  storageRef  .listAll()  .then(res => {  console.log(res);  res.items.forEach(itemRef => {  console.log(itemRef);  itemRef.getDownloadURL().then(url => {  this.getUrl.push(url);  console.log(url);  });  });  })  .catch(function(error) {  console.log(error);  });  } | | | |

### 表9-1-50 新增事件

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-1 | 函式名稱 | add |
| 功能 | 新增事件 | | |
| add: function(newdate, endDate) {  let uid = firebase.auth().currentUser.uid;  fStore  .collection("users")  .doc(uid).collection("calEvent").add({  eventName: this.eventName,  details: this.details,  frequency: this.frequency,  start: newdate,  end: endDate,  color: this.color  });  this.getEvents();  } | | | |

### 表9-1-51 取得事件

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-2 | 函式名稱 | getEvents |
| 功能 | 取得事件 | | |
| async getEvents() {  let uid = firebase.auth().currentUser.uid;  let snapshot = await fStore  .collection("users").doc(uid).collection("calEvent").get();  const events = [];  snapshot.forEach(doc => {  let APPData = doc.data();  APPData.id = doc.id;  events.push(APPData);  });  this.events = events;  } | | | |

### 表9-1-52 點取日期新增事件

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-3 | 函式名稱 | setDialogDate |
| 功能 | 點取日期新增事件 | | |
| setDialogDate({ date }) {  this.dialogDate = true;  this.focus = date;  this.startTitle = date;  this.endTitle = date;  } | | | |

### 表9-1-53 關閉新增事件

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-4 | 函式名稱 | closeDialog |
| 功能 | 關閉新增事件 | | |
| closeDialog() {  this.dialogDate = false;  } | | | |

### 表9-1-54 行事曆顯示類型

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-5 | 函式名稱 | viewDay |
| 功能 | 行事曆顯示類型 | | |
| viewDay({ date }) {  this.focus = date;  this.type = "day";  } | | | |

### 表9-1-55 取得事件顏色

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-6 | 函式名稱 | getEventColor |
| 功能 | 取得事件顏色 | | |
| getEventColor(event) {  return event.color;  } | | | |

### 表9-1-56 返回當日

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-7 | 函式名稱 | setToday |
| 功能 | 返回當日 | | |
| setToday() {  this.focus = this.today;  } | | | |

### 表9-1-57 顯示上個月/週/日/四天

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-8 | 函式名稱 | prev |
| 功能 | 顯示上個月/週/日/四天 | | |
| prev() {  this.$refs.calendar.prev();  } | | | |

### 表9-1-58 顯示下個月/週/日/四天

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-9 | 函式名稱 | next |
| 功能 | 顯示下個月/週/日/四天 | | |
| next() {  this.$refs.calendar.next();  } | | | |

### 表9-1-59 判斷頻率並新增事件

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-10 | 函式名稱 | addEvent |
| 功能 | 判斷頻率並新增事件 | | |
| async addEvent() {  if (this.eventName && this.startTitle && this.endTitle) {  if (this.frequency === 1) {  this.startTitle = new Date(this.startTitle);  this.endTitle = new Date(this.endTitle);  for (let i = 0; i < 5; i++) {  let newdate = new Date(this.startTitle);  let endDate = new Date(this.endTitle);  newdate.setDate(this.startTitle.getDate() + i);  endDate.setDate(this.endTitle.getDate() + i);  newdate = newdate.toISOString().slice(0, 10);  endDate = endDate.toISOString().slice(0, 10);  this.add(newdate, endDate);  }  } else if (this.frequency === 2) {  this.startTitle = new Date(this.startTitle);  this.endTitle = new Date(this.endTitle);  for (let i = 0; i < 5; i++) {  let newdate = new Date(this.startTitle);  let endDate = new Date(this.endTitle);  newdate.setDate(this.startTitle.getDate() + i \* 7);  endDate.setDate(this.endTitle.getDate() + i \* 7);  newdate = newdate.toISOString().slice(0, 10);  endDate = endDate.toISOString().slice(0, 10);  this.add(newdate, endDate);  }  } else if (this.frequency === 3) {  this.startTitle = new Date(this.startTitle);  this.endTitle = new Date(this.endTitle);  for (let i = 0; i < 5; i++) {  let newdate = new Date(this.startTitle);  let endDate = new Date(this.endTitle);  newdate.setMonth(this.startTitle.getMonth() + i);  endDate.setMonth(this.endTitle.getMonth() + i);  newdate = newdate.toISOString().slice(0, 10);  endDate = endDate.toISOString().slice(0, 10);  this.add(newdate, endDate);  }  } else if (this.frequency === 4) {  this.startTitle = new Date(this.startTitle);  this.endTitle = new Date(this.endTitle);  for (let i = 0; i < 2; i++) {  let newdate = new Date(this.startTitle);  let endDate = new Date(this.endTitle);  newdate.setFullYear(this.startTitle.getFullYear() + i);  endDate.setFullYear(this.endTitle.getFullYear() + i);  newdate = newdate.toISOString().slice(0, 10);  endDate = endDate.toISOString().slice(0, 10);  this.add(newdate, endDate);  }  } else if (this.frequency === 0) {  let newdate = new Date(this.startTitle);  let endDate = new Date(this.endTitle);  newdate = newdate.toISOString().slice(0, 10);  endDate = endDate.toISOString().slice(0, 10);  this.add(newdate, endDate);  }  }  this.eventName = "";  this.details = "";  this.startTitle = "";  this.endTitle = "";  } | | | |

### 表9-1-60 取得編輯事件

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-11 | 函式名稱 | editEvent |
| 功能 | 取得編輯事件 | | |
| editEvent(ev) {  this.currentlyEditing = ev.id;  this.eventName = ev.eventName;  this.start = ev.start;  this.frequency = ev.frequency;  this.end = ev.end;  this.color = ev.color;  } | | | |

### 表9-1-61 編輯事件

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-12 | 函式名稱 | updateEvent |
| 功能 | 編輯事件 | | |
| async updateEvent(ev) {  let uid = firebase.auth().currentUser.uid;  let selectedEvent = this.selectedEvent;  await fStore  .collection("users")  .doc(uid)  .collection("calEvent")  .doc(ev)  .update({  details: selectedEvent.details,  eventName: selectedEvent.eventName,  color: this.color,  frequency: this.frequency,  start: this.start,  end: this.end  })  .then(res => {  console.log("update completed");  })  .catch(err => {  console.log(err);  });  this.selectedOpen = false;  this.currentlyEditing = null;  console.log(ev);  this.$router.go({ path: this.$router.path });  } | | | |

### 表9-1-62 刪除事件

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-13 | 函式名稱 | deleteEvent |
| 功能 | 刪除事件 | | |
| async deleteEvent(ev) {  let selectedEvent = this.selectedEvent.eventName;  let uid = firebase.auth().currentUser.uid;  if (confirm(`確定是否刪除 ${selectedEvent} ?`)) {  await fStore  .collection("users")  .doc(uid)  .collection("calEvent")  .doc(ev)  .delete()  .then(res => {  console.log(res);  })  .catch(err => {  console.log(err);  });  this.selectedOpen = false;  this.getEvents();  }  } | | | |

### 表9-1-63 查看事件

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-14 | 函式名稱 | showEvent |
| 功能 | 查看事件 | | |
| showEvent({ nativeEvent, event }) {  const open = () => {  this.selectedEvent = event;  this.selectedElement = nativeEvent.target;  setTimeout(() => (this.selectedOpen = true), 10);  };  if (this.selectedOpen) {  this.selectedOpen = false;  setTimeout(open, 10);  } else {  open();  }  nativeEvent.stopPropagation();  } | | | |

### 表9-1-64 取得事件起訖時間

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-15 | 函式名稱 | updateRange |
| 功能 | 取得事件起訖時間 | | |
| updateRange({ start, end }) {  this.start = start;  this.end = end;  } | | | |

### 表9-1-65 行事曆顯示時間範圍

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-8-16 | 函式名稱 | title |
| 功能 | 行事曆顯示時間範圍 | | |
| title() {  const { start, end } = this;  if (!start || !end) {  return "";  }  const startMonth = typeof start === "string" ? start.slice(5, 7) : start.month;  const endMonth = end.month;  const suffixMonth = startMonth === endMonth ? startMonth : endMonth;  const startYear = typeof start === "string" ? start.slice(0, 4) : start.year;  const endYear = end.year;  const suffixYear = startYear === endYear ? startYear : endYear;  const startDay = start.day;  const endDay = end.day;  switch (this.type) {  case "month":  return `${startYear} 年 ${startMonth} 月`;  case "week":  return `${startYear} 年 ${startMonth} 月 ${startDay} 日 - ${suffixYear} 年 ${suffixMonth} 月 ${endDay} 日`;  case "4day":  return `${startYear} 年 ${startMonth} 月 ${startDay} 日 - ${suffixYear} 年 ${suffixMonth} 月 ${endDay - 1} 日`;  case "day":  return `${startYear} 年 ${startMonth} 月 ${startDay} 日`;  }  return "";  } | | | |

### 表9-1-66 取得檔案資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-9-1 | 函式名稱 | handleFileUpload |
| 功能 | 取得檔案資料 | | |
| handleFileUpload(e) {  this.imageData = e.target.files[0];  } | | | |

### 表9-1-67 建立相簿

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-9-2 | 函式名稱 | createAlbum |
| 功能 | 建立相簿 | | |
| createAlbum() {  let uid = firebase.auth().currentUser.uid;  console.log(uid);  var storageRef = firebase  .storage()  .ref(uid + "/" + this.albumName + "/" + this.imageData.name);  storageRef.put(this.imageData).then(function(snapshot) {  console.log("Uploaded files!");  location.reload();  });  } | | | |

### 表9-1-68 取得相簿資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-9-3 | 函式名稱 | getAlbumData |
| 功能 | 取得相簿資訊 | | |
| getAlbumData() {  let uid = firebase.auth().currentUser.uid;  let album = this.album;  let imageUrl = this.url;  var storageRef = firebase.storage().ref(uid + "/");  let folderName = [];  storageRef  .listAll()  .then(function(res) {  res.prefixes.forEach(function(folderRef) {  folderName.push(folderRef.name);  album.push(folderRef);  var imageRef = firebase.storage().ref(uid + "/" + folderRef.name);  imageRef.listAll().then(function(res) {  res.items.forEach(function(itemRef) {  itemRef.getDownloadURL().then(function(url) {  imageUrl.push(url);  });  });  });  });  })  .catch(function(error) {  console.log(error);  });  } | | | |

### 表9-1-69 取得檔案資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-10-1 | 函式名稱 | handleFileUpload |
| 功能 | 取得檔案資料 | | |
| handleFileUpload(e) {  this.imageData = e.target.files[0];  } | | | |

### 表9-1-70 新增相片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-10-2 | 函式名稱 | createPhoto |
| 功能 | 新增相片 | | |
| createPhoto() {  let uid = firebase.auth().currentUser.uid;  var storageRef = firebase  .storage()  .ref(uid + "/" + this.name + "/" + this.imageData.name);  storageRef.put(this.imageData).then(function(snapshot) {  console.log("Uploaded files!");  this.$router.go({ path: this.$router.path });  });  } | | | |

### 表9-1-71 刪除相片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-10-3 | 函式名稱 | deleteImage |
| 功能 | 刪除相片 | | |
| deleteImage(index) {  let uid = firebase.auth().currentUser.uid;  let target = this.album[index];  if (confirm(`是否刪除 ${target.name} ?`)) {  var storageRef = firebase  .storage().ref(uid + "/" + this.name + "/" + target.name);  storageRef.delete()  .then(function() {  alert("刪除成功!");  })  .catch(function(error) {  });  this.$router.go({ path: this.$router.path });  }  } | | | |

### 表9-1-72 取得單一相簿圖片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-10-4 | 函式名稱 | getPhoto |
| 功能 | 取得單一相簿圖片 | | |
| getPhoto() {  let uid = firebase.auth().currentUser.uid;  let name = this.$route.params.name;  this.name = name;  let album = this.album;  let imageUrl = this.url;  var storageRef = firebase.storage().ref(uid + "/" + this.name);  storageRef  .listAll()  .then(function(res) {  res.prefixes.forEach(function(folderRef) {  alert(folderRef);  console.log(folderRef);  });  res.items.forEach(function(itemRef) {  album.push(itemRef);  itemRef.getDownloadURL().then(function(url) {  imageUrl.push(url);  });  });  })  .catch(function(error) {  console.log(error);  });  } | | | |

### 表9-1-73 下載圖片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-10-5 | 函式名稱 | downloadImage |
| 功能 | 下載圖片 | | |
| downloadImage(item) {  let url = item;  let aLink = document.createElement("a");  aLink.setAttribute("target", "\_blank");  aLink.href = url;  aLink.download = "picture.png";  aLink.click();  } | | | |

### 表9-1-74 取得圖片資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-11-1 | 函式名稱 | handleFileUpload |
| 功能 | 取得圖片資訊 | | |
| handleFileUpload(e) {  this.leaderboard = [];  var file = e.target.files[0];  var reader = new FileReader();  reader.onload = e => {  this.preview = e.target.result;  };  reader.readAsDataURL(file);  let formData = new FormData();  formData.APPend("photo", file);  axios  .post("/upload", formData, {  baseURL: "http://localhost:3000",  timeout: 30000,  headers: {}  })  .then(response => {  var i = 1;  for (const [key, value] of Object.entries(response.data)) {  this.leaderboard.push({  top: i,  breed: key,  score: value  });  i = i + 1;  }  });  } | | | |

### 表9-1-75 取得圖片資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-12-1 | 函式名稱 | handleFileUpload |
| 功能 | 取得圖片資訊 | | |
| handleFileUpload(e) {  this.processing = true;  var file1 = e.target.files[0];  var file2 = e.target.files[1];  var reader1 = new FileReader();  var reader2 = new FileReader();  reader1.onload = e => {  this.imgA = e.target.result;  };  reader2.onload = e => {  this.imgB = e.target.result;  };  reader1.readAsDataURL(file1);  reader2.readAsDataURL(file2);  let formData = new FormData();  formData.APPend("photo", file1);  formData.APPend("photo", file2);  axios  .post("/tunit", formData, {  baseURL: "http://140.124.44.8:3000",  timeout: 10000,  headers: {  "Content-Type": "multipart/form-data"  }  })  .then(response => {  this.imgA = response.data["A"];  this.imgB = response.data["B"];  this.imgC = response.data["C"];  this.imgD = response.data["D"];  this.processing = false;  });  } | | | |

### 表9-1-76 取得上傳圖片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-1 | 函式名稱 | handleFileUpload |
| 功能 | 取得上傳圖片 | | |
| handleFileUpload(e) {  this.petImage = e.target.files[0];  } | | | |

### 表9-1-77 建立寵物資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-2 | 函式名稱 | save\_pet\_profile |
| 功能 | 建立寵物資訊 | | |
| async save\_pet\_profile(file) {  await fStore  .collection("users")  .doc(this.uid)  .collection("pets")  .add({  petName: this.petName,  petGender: this.petGender,  petBirth: this.petBirth,  breed: this.breed,  petHobby: this.petHobby,  petNote: this.petNote,  timestamp: new Date(),  profile\_picture: "",  // useless  petBirth\_year: 999999,  petBirth\_month: 999999,  petBirth\_date: 999999,  uid: this.uid  })  .then(function(ref) {  return firebase  .storage()  .ref(firebase.auth().currentUser.uid + "/" + ref.id)  .put(file)  .then(function(fileSnapshot) {  return fileSnapshot.ref.getDownloadURL().then(function(url) {  return ref.update({  profile\_picture: url  });  });  });  })  .then(() => {  alert("新增成功!");  })  .catch(function(error) {  console.error(  "There was an error uploading a file to Cloud Storage:",  error  );  });  this.$router.go({ path: this.$router.path });  } | | | |

### 表9-1-78 下載頭貼路徑

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-13-3 | 函式名稱 | onSubmit |
| 功能 | 下載頭貼路徑 | | |
| onSubmit(e) {  e.preventDefault();  this.save\_pet\_profile(this.petImage);  } | | | |

### 表9-1-79 監聽網址

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-1 | 函式名稱 | $route |
| 功能 | 監聽網址 | | |
| $route(now) {  let id = this.$route.params.id;  this.id = id;  axios.get("http://localhost:3000/comments/" + this.id).then(res => {  console.log(now);  this.comments = res.data;  }); | | | |

### 表9-1-80 取得單一文章資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-14-2 | 函式名稱 | getArticleData |
| 功能 | 取得單一文章資料 | | |
| getArticleData() {  let id = this.$route.params.id;  this.id = id;  axios.get("http://localhost:3000/comments/" + this.id).then(res => {  console.log(res.data);  this.comments = res.data;  });  } | | | |

### 表9-1-81 刪除文章

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-15-1 | 函式名稱 | deletePost |
| 功能 | 刪除文章 | | |
| deletePost(index) {  let target = this.comments[index];  if (confirm(`是否刪除 ${target.title} ?`)) {  axios  .delete(`http://localhost:3000/comments/${target.id}`)  .then(res => {  console.log(res);  this.comments.splice(index, 1);  })  .catch(err => {  console.log(err);  });  }  } | | | |

### 表9-1-82 新增內容列

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-15-2 | 函式名稱 | addRow |
| 功能 | 新增內容列 | | |
| addRow() {  console.log("Add new row");  this.contents.push({ content: "", title: "", img: "" });  } | | | |

### 表9-1-83 刪除內容列

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-15-3 | 函式名稱 | deleteRow |
| 功能 | 刪除內容列 | | |
| deleteRow(index) {  console.log("Delete row");  this.contents.splice(this.contents.length - 1, 1);  } | | | |

### 表9-1-84 清除文章暫存資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-15-4 | 函式名稱 | cleanData |
| 功能 | 清除文章暫存資料 | | |
| cleanData() {  this.title = "";  this.image = "";  this.introduction = "";  this.ending = "";  this.contents = [{ content: "", title: "", img: "" }];  } | | | |

### 表9-1-85 顯示文章內容

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-15-5 | 函式名稱 | putData |
| 功能 | 顯示文章內容 | | |
| putData(index) {  let target = this.comments[index];  for (var i = 0; i < target.contents.length; i++) {  this.contents[i].content = `${target.contents[i].content}`;  this.contents[i].img = `${target.contents[i].img}`;  this.contents[i].title = `${target.contents[i].title}`;  if (this.contents.length >= target.contents.length) {  continue;  }  this.contents.push({ content: "", title: "", img: "" });  }  } | | | |

### 表9-1-86 新增文章

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-15-6 | 函式名稱 | createArticle |
| 功能 | 新增文章 | | |
| createArticle() {  var article = {  title: this.title,  image: this.image,  introduction: this.introduction,  ending: this.ending  };  var content = {};  article["contents"] = [];  for (var i = 1; i < this.contents.length; i++) {  content["content"] = this.contents[i].content;  content["img"] = this.contents[i].img;  content["title"] = this.contents[i].title;  article["contents"].push(content[i]);  }  article["contents"] = this.contents;  axios.post("http://localhost:3000/comments", article).then(res => {  console.log(res);  this.title = "";  this.image = "";  this.introduction = "";  this.ending = "";  for (var i = 0; i < this.contents.length; i++) {  this.contents[i].content = "";  this.contents[i].img = "";  this.contents[i].title = "";  }  this.comments.push(res.data);  });  } | | | |

### 表9-1-87 編輯文章

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-15-7 | 函式名稱 | updateArticle |
| 功能 | 編輯文章 | | |
| updateArticle(index) {  let target = this.comments[index];  var article = {  title: target.title,  image: target.image,  introduction: target.introduction,  ending: target.ending  };  var content = {};  article["contents"] = [];  for (var i = 1; i < this.contents.length; i++) {  content["content"] = this.contents[i].content;  content["img"] = this.contents[i].img;  content["title"] = this.contents[i].title;  article["contents"].push(content[i]);  }  article["contents"] = this.contents;  axios  .patch(`http://localhost:3000/comments/${target.id}`, article)  .then(res => {  });  } | | | |

### 表9-1-88 取得文章

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-15-8 | 函式名稱 | getArticle |
| 功能 | 取得文章 | | |
| getArticle() {  axios.get("http://localhost:3000/comments").then(res => {  this.comments = res.data;  });  } | | | |

### 表9-1-89 刪除領養機構資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-16-1 | 函式名稱 | deleteAdoption |
| 功能 | 刪除領養機構資訊 | | |
| deleteAdoption(index) {  let target = this.comments[index];  if (confirm(`是否刪除 ${target.name} ?`)) {  axios  .delete(`http://localhost:4000/comments/${target.id}`)  .then(res => {  console.log(res);  this.comments.splice(index, 1);  })  .catch(err => {  console.log(err);  });  }  } | | | |

### 表9-1-90 清除領養機構暫存資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-16-2 | 函式名稱 | cleanData |
| 功能 | 清除領養機構暫存資料 | | |
| cleanData() {  this.name = "";  this.src = "";  this.introduction = "";  this.address = "";  this.phone = "";  this.content = "";  this.center = { lat: "", lng: "" };  } | | | |

### 表9-1-91 新增領養機構

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-16-3 | 函式名稱 | createAdoption |
| 功能 | 新增領養機構 | | |
| createAdoption() {  var adoption = {  name: this.name,  src: this.src,  introduction: this.introduction,  address: this.address,  phone: this.phone,  content: this.content  };  adoption["center"] = {};  this.center = { lat: this.center.lat, lng: this.center.lng };  adoption["center"] = this.center;  axios.post("http://localhost:4000/comments", adoption).then(res => {  console.log(res);  this.name = "";  this.src = "";  this.introduction = "";  this.address = "";  this.phone = "";  this.content = "";  this.center.lat = "";  this.center.lng = "";  this.comments.push(res.data);  });  } | | | |

### 表9-1-92 編輯領養機構

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-16-4 | 函式名稱 | updateAdoption |
| 功能 | 編輯領養機構 | | |
| updateAdoption(index) {  let target = this.comments[index];  var adoption = {  name: target.name,  src: target.src,  introduction: target.introduction,  address: target.address,  phone: target.phone,  content: target.content  };  adoption["center"] = {};  this.center = { lat: target.center.lat, lng: target.center.lng };  adoption["center"] = this.center;  axios  .patch(`http://localhost:4000/comments/${target.id}`, adoption)  .then(res => {  console.log(res);  });  } | | | |

### 表9-1-93 取得領養機構資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-16-5 | 函式名稱 | getAdoption |
| 功能 | 取得領養機構資訊 | | |
| getAdoption() {  axios.get("http://localhost:4000/comments").then(res => {  console.log(res);  this.comments = res.data;  });  } | | | |

### 表9-1-94 取得回饋資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-17-1 | 函式名稱 | getContactData |
| 功能 | 取得回饋資訊 | | |
| getContactData() {  fStore.collection("contact").get();  fStore  .collection("contact")  .get()  .then(querySnapshot => {  querySnapshot.forEach(doc => {  this.question.push(doc.data());  console.log(doc.id, doc.data());  });  console.log(this.question);  });  } | | | |

### 表9-1-95 確認用戶登入狀態

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-18-1 | 函式名稱 | logout |
| 功能 | 確認用戶登入狀態 | | |
| logout: function(e) {  firebase  .auth()  .signOut()  .then(() => {  this.$router.go({ path: this.$router.path });  });  } | | | |

### 表9-1-96 取得會員資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-19-1 | 函式名稱 | updateUserProfile |
| 功能 | 取得會員資料 | | |
| async function updateUserProfile(store, uid) {  await fStore  .collection("users")  .doc(uid)  .get()  .then(doc => {  store.commit("updateUserProfile", {  uid: uid,  name: doc.data().name,  email: doc.data().email,  photoURL: doc.data().photoURL  });  });  } | | | |

### 表9-1-97 取得寵物清單

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-19-2 | 函式名稱 | getUserPetsList |
| 功能 | 取得寵物清單 | | |
| async function getUserPetsList(store, uid) {  const petsList = [];  await fStore  .collection("users").doc(uid).collection("pets").get()  .then(function(snapshot) {  snapshot.forEach(function(doc) {  petsList.push({ id: doc.id, name: doc.data().petName });  });  store.commit("getUserPetsList", {  pets\_list: petsList  });  });  } | | | |

### 表9-1-98 更新行程列表

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-19-3 | 函式名稱 | updateMonthCalendar |
| 功能 | 更新行程列表 | | |
| async function updateMonthCalendar(store, uid) {  await fStore  .collection("users")  .doc(uid)  .collection("calEvent")  .get()  .then(function(querySnapshot) {  const events = [];  querySnapshot.forEach(function(doc) {  var name = doc.data().name;  var start = doc.data().start;  var end = doc.data().end;  var dateSplit = start.split("-");  var now = new Date();  // 過濾出當月事件  if (  Number(dateSplit[0]) === now.getFullYear() &&  Number(dateSplit[1]) === now.getMonth() + 1  ) {  if (  // 非連續性活動  new Date(start).setHours(0, 0, 0, 0) ===  new Date(end).setHours(0, 0, 0, 0)  ) {  events.push({  name: name,  start: start,  end: end  });  } else {  // 連續性活動  console.log(name);  }  }  // Sort events by start  });  store.commit("updateMonthCalendar", {  month\_calendar: events  });  });  } | | | |

### 表9-1-99 取得寵物資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-20-1 | 函式名稱 | updatePetInfo |
| 功能 | 取得寵物資料 | | |
| async function updatePetInfo(store, uid) {  var petsRef = fStore  .collection("users")  .doc(uid)  .collection("pets");  await petsRef.get().then(function(querySnapshot) {  querySnapshot.forEach(function(doc) {  store.commit("updatePetInfo", {  pet: doc.data(),  pet\_doc\_id: doc.id  });  });  });  } | | | |

### 表9-1-100 取得個人資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-21-1 | 函式名稱 | getUserProfile |
| 功能 | 取得個人資訊 | | |
| getUserProfile() {  let uid = firebase.auth().currentUser.uid;  fStore  .collection("users")  .doc(uid)  .get()  .then(doc => {  let user = doc.data();  this.name = user.name;  this.email = user.email;  this.userName = user.userName;  this.address = user.address;  this.phone = user.phone;  this.birth = user.userBirth;  this.gender = user.userGender;  });  } | | | |

### 表9-1-101 更新個人資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-21-2 | 函式名稱 | updateProfile |
| 功能 | 更新個人資訊 | | |
| updateProfile() {  let uid = firebase.auth().currentUser.uid;  fStore  .collection("users")  .doc(uid)  .update({  name: this.name,  email: this.email,  address: this.address,  userName: this.userName,  userGender: this.gender,  phone: this.phone,  userBirth: this.birth  });  alert("更新成功！");  } | | | |

### 表9-1-102 更新密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-22-1 | 函式名稱 | updatePassword |
| 功能 | 更新密碼 | | |
| updatePassword(currentPassword, newPassword) {  var user = firebase.auth().currentUser;  const credential = firebase.auth.EmailAuthProvider.credential(  user.email,  currentPassword  );  if (this.check) {  user.reauthenticateWithCredential(credential)  .then(function() {  user.updatePassword(newPassword)  .then(function() {  alert("更改成功！");  })  .catch(function(error) {  });  })  .catch(function(error) {  alert("密碼輸入錯誤");  });  this.currentPassword = "";  this.newPassword = "";  this.confirmNewPassword = "";  }  } | | | |

### 表9-1-103 確認重複密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-22-2 | 函式名稱 | checkPassword |
| 功能 | 確認重複密碼 | | |
| checkPassword(confirmNewPassword, newPassword) {  if (confirmNewPassword === newPassword) {  this.check = true;  } else {  alert("兩次密碼不相等，請再次確認密碼");  this.check = false;  }  } | | | |

### 表9-1-104 隱藏目前密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-22-3 | 函式名稱 | hideCurrentPassword |
| 功能 | 隱藏目前密碼 | | |
| hideCurrentPassword() {  this.passwordCurrentPasswordHidden = false;  this.passwordCurrentPasswordFieldType =  this.passwordCurrentPasswordFieldType === "password"  ? "text" : "password";  } | | | |

### 表9-1-105 顯示目前密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-22-4 | 函式名稱 | showCurrentPassword |
| 功能 | 顯示目前密碼 | | |
| showCurrentPassword() {  this.passwordCurrentPasswordHidden = true;  this.passwordCurrentPasswordFieldType = this.passwordCurrentPasswordFieldType;  } | | | |

### 表9-1-106 隱藏新密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-22-5 | 函式名稱 | hideNewPassword |
| 功能 | 隱藏新密碼 | | |
| hideNewPassword() {  this.passwordNewPasswordHidden = false;  this.passwordNewPasswordFieldType =  this.passwordNewPasswordFieldType === "password" ? "text" : "password";  } | | | |

### 表9-1-107 顯示新密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-22-6 | 函式名稱 | showNewPassword |
| 功能 | 顯示新密碼 | | |
| showNewPassword() {  this.passwordNewPasswordHidden = true;  this.passwordNewPasswordFieldType = this.passwordNewPasswordFieldType;  } | | | |

### 表9-1-108 隱藏確認密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-22-7 | 函式名稱 | hideConfirmNewPassword |
| 功能 | 隱藏確認密碼 | | |
| hideConfirmNewPassword() {  this.passwordConfirmNewPasswordHidden = false;  this.passwordConfirmNewPasswordFieldType =  this.passwordNewPasswordFieldType === "password" ? "text" : "password";  } | | | |

### 表9-1-109 顯示確認密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-22-8 | 函式名稱 | showConfirmNewPassword |
| 功能 | 顯示確認密碼 | | |
| showConfirmNewPassword() {  this.passwordConfirmNewPasswordHidden = true;  this.passwordConfirmNewPasswordFieldType = this.passwordConfirmNewPasswordFieldType;  } | | | |

### 表9-1-110 取得個人資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-23-1 | 函式名稱 | checkPassword |
| 功能 | 取得個人資訊 | | |
| checkPassword() {  if (this.password === "ntub10646043") {  this.auth = false;  } else {  this.auth = true;  }  } | | | |

## 9-2 其他附屬之各種元件

### 表9-2-1 首頁功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/homeFragment.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-1-1 | setNotification | 設定通知日期 |
| 2-1-2 | calculationCountdownDate | 計算倒數天數 |
| 2-1-3 | getPetInfo | 取得寵物資訊 |
| 2-1-4 | getPetImage | 取得寵物照片 |
| 2-1-5 | setCountdownDate | 設定倒數計時器日期 |
| 2-1-6 | dueDay | 計算到期天數 |
| 2-1-7 | getDocumentName | 取得文件名稱 |

### 表9-2-2 登入功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/loginFragment.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-2-1 | onActivityCreated | 自動登入 |
| 2-2-2 | login | 使用者登入 |

### 表9-2-3 註冊功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/signUpFragment.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-3-1 | checkPassword | 檢查密碼 |
| 2-3-2 | addUser | 使用者註冊 |

### 表9-2-4 新增使用者資訊功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/userInfoFragment.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-4-1 | addUserInfo | 新增使用者資訊 |

### 表9-2-5 行事曆功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/calendarActivity.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-5-1 | onCreate | 函式導向、顯示行程 |
| 2-5-2 | addEvent | 導頁至新增行程 |
| 2-5-3 | onStart | 標記行程 |

### 表9-2-6 新增寵物功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/petInfoFragment.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-6-1 | onCreate | 函式導向 |
| 2-6-2 | getPermission | 取得相機、相簿權限 |
| 2-6-3 | onActivityResult | 讀取照片 |
| 2-6-4 | scaleImage | 調整照片大小 |
| 2-6-5 | addPetInfo | 新增寵物資訊 |
| 2-6-6 | addImage | 上傳圖片 |

### 表9-2-7 新增行程功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/addEventActivity.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-7-1 | addEvent | 新增行程 |

### 表9-2-8 詳細行程顯示功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/bottomSheetEventPreview.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-8-1 | deleteEvent | 刪除行程 |
| 2-8-2 | setEvent | 更新行程 |

### 表9-2-9 設定功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/settingActivity.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-9-1 | onCreate | 登出 |

### 表9-2-10 更新會員資訊功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/editUserInfoActivity.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-10-1 | onCreate | 傳值至更新會員帳號 |
| 2-10-2 | setUserInfo | 更新會員資料 |
| 2-10-3 | setEditText | 將EditText轉換成日期選擇器 |
| 2-10-4 | getUserInfo | 取得會員資料 |

### 表9-2-11 更新會員帳號功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/editAccountActivity.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-11-1 | setAccount | 更新會員帳號 |
| 2-11-2 | getAccount | 取得會員帳號 |

### 表9-2-12 更新會員密碼功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/editPwdActivity.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-12-1 | setPwd | 更新會員密碼 |

### 表9-2-13 更新寵物資訊功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/editPetsInfoActivity.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-13-1 | onCreate | 函式導向 |
| 2-13-2 | getPetImage | 取得寵物照片 |
| 2-13-3 | getPetInfo | 取得寵物資訊 |
| 2-13-4 | addImage | 上傳圖片 |
| 2-13-5 | setPetInfo | 更新寵物資訊 |
| 2-13-6 | setEditText | 將EditText轉換成日期選擇器 |
| 2-13-7 | onActivityResult | 讀取照片 |
| 2-13-8 | scaleImage | 調整照片大小 |

### 表9-2-14 常見問題功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/editQuestionActivity.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-14-1 | onCreate | 顯示常見問題 |

### 表9-2-15 聯絡我們功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/editContactusActivity.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-15-1 | addContactUs | 新增聯絡我們資訊 |

### 表9-2-16 通知功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/alarmReceiver.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-16-1 | onReceive | 發送通知 |

### 表9-2-17 定位功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/MapsActivity.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-17-1 | displayTrack | 確認有無安裝地圖 |
| 2-17-2 | chkPlayService | 確認playService版本 |
| 2-17-3 | onRequestPermissionsResult | 確認權限 |
| 2-17-4 | initialMap | 初始化地圖 |
| 2-17-5 | init | 執行初始化動作 |
| 2-17-6 | getDeviceLocation | 取得現在定位點 |
| 2-17-7 | geoLocate | 取得查詢地址定位點 |
| 2-17-8 | onMapReady | 附近位置查詢 |

### 表9-2-18 定位功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/GetNearbyPlacesData.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-18-1 | doInBackground | 數據解析 |
| 2-18-2 | onPostExecute | 顯示數據列表的結果 |
| 2-18-3 | ShowNearbyPlaces | 顯示附近位置URL |

### 表9-2-19 定位功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/DownloadUrl.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-19-1 | readUrl | 下載附近位置URL |

### 表9-2-20 定位功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/DataParser.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-20-1 | parse | 解析數據存於列表中 |
| 2-20-2 | getPlaces | 解析附近地點詳細位置 |
| 2-20-3 | getPlace | 解析附近地點詳細資訊 |

### 表9-2-21 定位功能列表

|  |  |  |
| --- | --- | --- |
| 檔案路徑與名稱: src/main/java/DownloadUrl.java | | |
| 編號 | 函式名稱 | 功能 |
| 2-21-1 | onCreate | 函式導向 |
| 2-21-2 | uploadImage | 上傳照片至資料庫 |
| 2-21-3 | StoreLink | 資料庫連結設定 |
| 2-21-4 | onActivityResult | 讀取照片 |

### 表9-2-22 設定通知日期

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-1-1 | 函式名稱 | setNotification |
| 功能 | 設定通知日期 | | |
| private void setNotification() {  final FirebaseFirestore db = FirebaseFirestore.getInstance();  db.collection("users").document(userUID).collection("pets").document(petdocumentname).collection("countdown")  .get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  @Override  public void onComplete(@NonNull Task<QuerySnapshot> task) {  if (task.isSuccessful()) {  for (QueryDocumentSnapshot document : task.getResult()) {  StringBuilder field1 = new StringBuilder("");  String endDay = field1.APPend(document.get("endDay")).toString();  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  if(!endDay.equals("")) {  date = sdf.parse(endDay);  Calendar calendar = Calendar.getInstance();  calendar.setTime(date);  calendar.set(Calendar.HOUR\_OF\_DAY, 00);  calendar.set(Calendar.MINUTE, 00);  calendar.set(Calendar.SECOND, 00);  int id = calendar.get(Calendar.MONTH)+calendar.get(Calendar.DAY\_OF\_MONTH);  Intent intent = new Intent(getActivity(), alarmReceiver.class);  PendingIntent pendingIntent = PendingIntent.getBroadcast(getActivity(), id, intent, 0);  AlarmManager alarmManager = (AlarmManager) getActivity().getSystemService(getContext().ALARM\_SERVICE);  alarmManager.set(AlarmManager.RTC\_WAKEUP, calendar.getTimeInMillis(), pendingIntent);  }  }  }  }  });  } | | | |

### 表9-2-23 計算倒數天數

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-1-2 | 函式名稱 | calculationCountdownDate |
| 功能 | 計算倒數天數 | | |
| private int calculationCountdownDate(String count) throws ParseException {  Calendar optiondate = Calendar.getInstance();  Calendar now = Calendar.getInstance();  SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");  Date specialdate = df.parse(count);  optiondate.setTime(specialdate);  int showercountdownday = (int) Math.ceil(((optiondate.getTimeInMillis() - now.getTimeInMillis()) / (24 \* 60 \* 60 \* 1000.0)));  return showercountdownday;  } | | | |

### 表9-2-24 取得寵物資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-1-3 | 函式名稱 | getPetInfo |
| 功能 | 取得寵物資訊 | | |
| public void getPetInfo(String query){  final FirebaseFirestore db = FirebaseFirestore.getInstance();  final TextView text\_petsSex = getView().findViewById(R.id.text\_petsSex);  final TextView text\_petsBirth = getView().findViewById(R.id.text\_petsBirth);  final TextView text\_petsAge = getView().findViewById(R.id.text\_petsAge);  final TextView text\_route = getView().findViewById(R.id.text\_route);  db.collection("users").document(userUID).collection("pets").whereEqualTo("petName", query)  .get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  @Override  public void onComplete(@NonNull Task<QuerySnapshot> task) {  if (task.isSuccessful()) {  for (QueryDocumentSnapshot document : task.getResult()) {  DocumentSnapshot doc = document;  StringBuilder fields = new StringBuilder("");  StringBuilder fields2 = new StringBuilder("");  StringBuilder fields3 = new StringBuilder("");  fields.APPend(doc.get("petGender")).toString();  fields2.APPend(doc.get("petBirth")).toString();  fields3.APPend(doc.get("breed")).toString();  text\_petsSex.setText(fields);  text\_petsBirth.setText(fields2);  text\_route.setText(fields3);  SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");  String petsBirth = fields2.toString();  try {  Date Birth = df.parse(petsBirth);  Date nowdate = new Date();  Calendar ca1 = Calendar.getInstance();  Calendar ca2 = Calendar.getInstance();  ca1.setTime(Birth);  ca2.setTime(nowdate);  int year = ca2.get(Calendar.YEAR) - ca1.get(Calendar.YEAR);  int month = ca2.get(Calendar.MONTH) - ca1.get(Calendar.MONTH);  text\_petsAge.setText(year + "歲" + month + "月");  } catch (ParseException e) {  e.printStackTrace();  }  }  }  }  });  } | | | |

### 表9-2-25 取得寵物照片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-1-4 | 函式名稱 | getPetImage |
| 功能 | 取得寵物照片 | | |
| public void getPetImage(String query){  FirebaseStorage storage = FirebaseStorage.getInstance("gs://ipets-APP.APPspot.com");  StorageReference mStorageRef = storage.getReference();  StorageReference islandRef = mStorageRef.child(userUID +'/'+ query+".jpg");  final ImageView head = getView().findViewById(R.id.head);  final long ONE\_MEGABYTE = 3000 \* 3000;  islandRef.getBytes(ONE\_MEGABYTE).addOnSuccessListener(new OnSuccessListener<byte[]>() {  @Override  public void onSuccess(byte[] bytes) {  Glide.with(getContext())  .load(bytes)  .into(head);  }  }).addOnFailureListener(new OnFailureListener() {  @Override  public void onFailure(@NonNull Exception exception) {  // Handle any errors'  }  });  } | | | |

### 表9-2-26 設定倒數計時器日期

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-1-5 | 函式名稱 | setCountdownDate |
| 功能 | 設定倒數計時器日期 | | |
| public void setCountdownDate(String countdownEvent){  final FirebaseFirestore db = FirebaseFirestore.getInstance();  db.collection("users").document(userUID).collection("pets").document(petdocumentname).collection("countdown").whereEqualTo("countdownEvent", countdownEvent)  .get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  @Override  public void onComplete(@NonNull Task<QuerySnapshot> task) {  if (task.isSuccessful()) {  for (QueryDocumentSnapshot document : task.getResult()) {  String deleteid = document.getId();  db.collection("users").document(userUID).collection("pets").document(petdocumentname).collection("countdown").document(deleteid).delete();  }  }  }  });  Calendar optiondate = Calendar.getInstance();  Calendar now = Calendar.getInstance();  optiondate.set(year, monthOfYear, dayOfMonth);  monthOfYear = monthOfYear+1;  SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");  Date curDate = new Date(System.currentTimeMillis()) ;  String showercountdownday = formatter.format(curDate);  String showerday = year+"-"+monthOfYear+"-"+dayOfMonth;  Map<String, Object> countdowndate = new HashMap<>();  countdowndate.put("startDay", showercountdownday);  countdowndate.put("endDay", showerday);  countdowndate.put("countdownEvent",countdownEvent);  db.collection("users").document(userUID).collection("pets").document(petdocumentname).collection("countdown").document().set(countdowndate);  setCountdownColor();  } | | | |

### 表9-2-27 計算到期天數

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-1-6 | 函式名稱 | dueDay |
| 功能 | 計算到期天數 | | |
| private int dueDay(String start,String end) throws ParseException {  SimpleDateFormat sdf= new SimpleDateFormat("yyyy-MM-dd");  Date date =sdf.parse(start);  Calendar calendar = Calendar.getInstance();  calendar.setTime(date);  Date date1 =sdf.parse(end);  Calendar calendar1 = Calendar.getInstance();  calendar1.setTime(date1);  int dueday = (int) Math.ceil(((calendar1.getTimeInMillis() - calendar.getTimeInMillis()) / (24 \* 60 \* 60 \* 1000.0)));  return dueday;  } | | | |

### 表9-2-28 取得文件名稱

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-1-7 | 函式名稱 | getDocumentName |
| 功能 | 取得文件名稱 | | |
| public void getDocumentName(){  final FirebaseFirestore db = FirebaseFirestore.getInstance();  db.collection("users").document(userUID).collection("pets").whereEqualTo("petName", pet\_query)  .get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  @Override  public void onComplete(@NonNull Task<QuerySnapshot> task) {  if (task.isSuccessful()) {  for (QueryDocumentSnapshot document : task.getResult()) {  petdocumentname = document.getId();  }  }  }  });  } | | | |

### 表9-2-29 自動登入

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-2-1 | 函式名稱 | onActivityCreated |
| 功能 | 自動登入 | | |
| @Override  public void onActivityCreated(@Nullable Bundle savedInstanceState) {  super.onActivityCreated(savedInstanceState);  auth = FirebaseAuth.getInstance();  FirebaseUser currentUser = auth.getCurrentUser();  if(currentUser!=null) {  Intent intent=new Intent();  intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK|Intent.FLAG\_ACTIVITY\_NEW\_TASK);  intent.setClass(getActivity(), homeActivity.class);  startActivity(intent);  }  } | | | |

### 表9-2-30 使用者登入

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-2-2 | 函式名稱 | login |
| 功能 | 使用者登入 | | |
| public void login(final View v){  final EditText edpassword = getView().findViewById(R.id.password);  final EditText edaccount = getView().findViewById(R.id.account);  String account = edaccount.getText().toString();  final String password = edpassword.getText().toString();  auth.signInWithEmailAndPassword(account, password)  .addOnCompleteListener(new OnCompleteListener<AuthResult>() {  @Override  public void onComplete(@NonNull Task<AuthResult> task) {  if (!task.isSuccessful()){  AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());  builder.setTitle("auth/user-not-found iPets");  builder.setMessage("帳號或密碼錯誤,大小寫需相符");  builder.setNegativeButton("重新輸入", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  edaccount.setText("");  edpassword.setText("");  }  });  builder.setCancelable(true);  builder.show();  }else{  Intent intent=new Intent(); intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK|Intent.FLAG\_ACTIVITY\_NEW\_TASK);  intent.setClass(getActivity(), homeActivity.class);  startActivity(intent);  }  }  });  } | | | |

### 表9-2-31 檢查密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-3-1 | 函式名稱 | checkPassword |
| 功能 | 檢查密碼 | | |
| private void checkPassword() {  EditText edemail = getView().findViewById(R.id.email);  final EditText edpassword = getView().findViewById(R.id.addpw);  final EditText edconfirmpw = getView().findViewById(R.id.confirmpw);  String email = edemail.getText().toString();  String password = edpassword.getText().toString();  String confirmpw = edconfirmpw.getText().toString();  if(password.length() < 6){  AlertDialog.Builder passworderror = new AlertDialog.Builder(getActivity());  passworderror.setMessage("密碼長度不可<6個字元");  passworderror.setNegativeButton("重新輸入", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  edpassword.setText("");  }  });  passworderror.show();  }else if(password.equals(confirmpw)!= true){  AlertDialog.Builder confirmpwerror = new AlertDialog.Builder(getActivity());  confirmpwerror.setMessage("密碼不符");  confirmpwerror.setNegativeButton("重新輸入", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  edpassword.setText("");  edconfirmpw.setText("");  }  });  confirmpwerror.show();  }else {  addUser(email, password);  }  } | | | |

### 表9-2-32 使用者註冊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-3-2 | 函式名稱 | addUser |
| 功能 | 使用者註冊 | | |
| private void addUser(final String email, final String password) {  auth.createUserWithEmailAndPassword(email, password)  .addOnCompleteListener(  new OnCompleteListener<AuthResult>() {  @Override  public void onComplete(@NonNull Task<AuthResult> task) {  if(task.isSuccessful()) {  final EditText edusername = getView().findViewById(R.id.username);  String username = edusername.getText().toString();  userUID = auth.getUid();  FirebaseFirestore db;  db = FirebaseFirestore.getInstance();  Map<String, Object> userInfo = new HashMap<>();  userInfo.put("userName", username);  userInfo.put("email", email);  userInfo.put("password", password);  db.collection("users").document(userUID).set(userInfo);  NavController controller = Navigation.findNavController(getView());  controller.navigate(R.id.action\_signupFragment\_to\_userinfoFragment);  }else{  AlertDialog.Builder emailerror = new AlertDialog.Builder(getActivity());  emailerror.setMessage("此信箱已使用過或信箱格式錯誤");  emailerror.setNegativeButton("重新輸入", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  EditText edemail = getView().findViewById(R.id.email);  edemail.setText("");  }  });  emailerror.show();  }  }  }  );  } | | | |

### 表9-2-33 新增使用者資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-4-1 | 函式名稱 | addUserInfo |
| 功能 | 新增使用者資訊 | | |
| private void addUserInfo(View v) {  EditText edmyname = getView().findViewById(R.id.myname);  final EditText edmybirth = getView().findViewById(R.id.mybirth);  final EditText edmyaddress = getView().findViewById(R.id.myaddress);  final EditText edmyNum = getView().findViewById(R.id.myNum);  String myname = edmyname.getText().toString();  String mybirth = edmybirth.getText().toString();  String myaddress = edmyaddress.getText().toString();  String myNum = edmyNum.getText().toString();  String mygender = null;  RadioGroup genderselect = getView().findViewById(R.id.genderselect);  switch(genderselect.getCheckedRadioButtonId()){  case R.id.male:  mygender = "男";  break;  case R.id.female:  mygender = "女";  break;  case R.id.unwilling:  mygender = "不願透漏";  break;  }  FirebaseAuth auth = FirebaseAuth.getInstance();  FirebaseUser currentUser = auth.getCurrentUser();  String userUID = currentUser.getUid();  FirebaseFirestore db;  db = FirebaseFirestore.getInstance();  Map<String, Object> userInfo = new HashMap<>();  userInfo.put("name", myname);  userInfo.put("userBirth", mybirth);  userInfo.put("userGender", mygender);  userInfo.put("address", myaddress);  userInfo.put("phone",myNum);  userInfo.put("userBirth\_year", userBirth\_year);  userInfo.put("userBirth\_month", userBirth\_month);  userInfo.put("userBirth\_date",userBirth\_date);  db.collection("users").document(userUID).update(userInfo);  AlertDialog.Builder finishsignup = new AlertDialog.Builder(getActivity());  finishsignup.setMessage("歡迎加入iPets的大家庭~!");  finishsignup.setNegativeButton("確認", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  Intent intent=new Intent();  intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK|Intent.FLAG\_ACTIVITY\_NEW\_TASK);  intent.setClass(getActivity(), homeActivity.class);  startActivity(intent);  }  });  finishsignup.setCancelable(false);  finishsignup.show();  } | | | |

### 表9-2-34 函式導向、顯示行程

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-5-1 | 函式名稱 | onCreate |
| 功能 | 函式導向、顯示行程 | | |
| @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  mCalendarView.setOnDayClickListener(new OnDayClickListener() {  @Override  public void onDayClick(EventDay eventDay) {  final FirebaseFirestore db = FirebaseFirestore.getInstance();  List details = new ArrayList<>();  List eventnames = new ArrayList<>();  List ids = new ArrayList<>();  TextView noteTitle = findViewById(R.id.noteTitle);  TextView noteContent = findViewById(R.id.noteContent);  TextView noteTitle2 = findViewById(R.id.noteTitle2);  TextView noteContent2 = findViewById(R.id.noteContent2);  TextView noteTitle3 = findViewById(R.id.noteTitle3);  TextView noteContent3 = findViewById(R.id.noteContent3);  TextView noteTitle4 = findViewById(R.id.noteTitle4);  TextView noteContent4 = findViewById(R.id.noteContent4);  noteTitle.setText("");  noteContent.setText("");  noteTitle2.setText("");  noteContent2.setText("");  noteTitle3.setText("");  noteContent3.setText("");  noteTitle4.setText("");  noteContent4.setText("");  Calendar clickedDayCalendar = eventDay.getCalendar();  Date date = clickedDayCalendar.getTime();  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  String dateStr = sdf.format(date);  db.collection("users").document(userUID).collection("calEvent").get()  .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  @Override  public void onComplete(@NonNull Task<QuerySnapshot> task) {  if (task.isSuccessful()) {  for (QueryDocumentSnapshot document : task.getResult()) {  String id = document.getId();  db.collection("users").document(userUID).collection("calEvent").document(id)  .get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {  @Override  public void onComplete(@NonNull Task<DocumentSnapshot> task) {  if (task.isSuccessful()) {  DocumentSnapshot doc = task.getResult();  StringBuilder fields = new StringBuilder("");  StringBuilder fields1 = new StringBuilder("");  StringBuilder fields2 = new StringBuilder("");  StringBuilder fields4 = new StringBuilder("");  StringBuilder fields5= new StringBuilder("");  String start = fields.APPend(doc.get("startDate")).toString();  String end = fields1.APPend(doc.get("endDate")).toString();  String detail = fields2.APPend(doc.get("details")).toString();  String eventname = fields4.APPend(doc.get("eventName")).toString();  try {  Date eventday = sdf.parse(dateStr);  Date strday = sdf.parse(start);  Calendar startcal = Calendar.getInstance();  startcal.setTime(strday);  Date endday = sdf.parse(end);  Calendar endcal = Calendar.getInstance();  endcal.setTime(endday);  endcal.add(Calendar.DATE, 1);  endday = endcal.getTime();  do {  if (strday.equals(eventday)) {  String eventid = document.getId();  ids.add(eventid);  details.add(detail);  eventnames.add(eventname);  for (int i = 0; i < ids.size(); i++) {  if (i == 0) {  id1 = (String) ids.get(i);  }else if (i == 1){  id2 = (String) ids.get(i);  }else if (i == 2) {  id3 = (String) ids.get(i);  }else {  id4 = (String) ids.get(i);  }  }  for (int i = 0; i < eventnames.size(); i++) {  if (i == 0) {  noteTitle.setText((String) eventnames.get(i));  }else if (i == 1) {  noteTitle2.setText((String) eventnames.get(i));  }else if (i == 2) {  noteTitle3.setText((String) eventnames.get(i));  }else {  noteTitle4.setText((String) eventnames.get(i));  }  }  for (int i = 0; i < details.size(); i++) {  if (i == 0) {  noteContent.setText((String) details.get(i));  }else if (i == 1) {  noteContent2.setText((String) details.get(i));  }else if (i == 2) {  noteContent3.setText((String) details.get(i));  }else {  noteContent4.setText((String) details.get(i));  }  }  break;  }  startcal.add(Calendar.DATE, 1);  strday = startcal.getTime();  }while (strday.compareTo(endday) < 0);  } catch (ParseException e) {  }  }  }  });  }  }  }  });  }  });  } | | | |

### 表9-2-35 導頁至新增行程

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-5-2 | 函式名稱 | addEvent |
| 功能 | 導頁至新增行程 | | |
| private void addEvent() {  Intent intent = new Intent(this, addEventActivity.class);  startActivityForResult(intent, ADD\_NOTE);  } | | | |

### 表9-2-36 標記行程

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-5-3 | 函式名稱 | onStart |
| 功能 | 標記行程 | | |
| @Override  protected void onStart() {  super.onStart();  mCalendarView = findViewById(R.id.calendarView);  final FirebaseFirestore db = FirebaseFirestore.getInstance();  db.collection("users").document(userUID).collection("calEvent").get()  .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  @Override  public void onComplete(@NonNull Task<QuerySnapshot> task) {  if (task.isSuccessful()) {  for (QueryDocumentSnapshot document : task.getResult()) {  String id = document.getId();  db.collection("users").document(userUID).collection("calEvent").document(id)  .get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {  @Override  public void onComplete(@NonNull Task<DocumentSnapshot> task) {  if (task.isSuccessful()) {  DocumentSnapshot doc = task.getResult();  StringBuilder fields = new StringBuilder("");  StringBuilder fields3 = new StringBuilder("");  StringBuilder fields5 = new StringBuilder("");  String color = fields.APPend(doc.get("color")).toString();  String enddate = fields3.APPend(doc.get("endDate")).toString();  String startdate = fields5.APPend(doc.get("startDate")).toString();  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  Date strday = sdf.parse(startdate);  Calendar startcal = Calendar.getInstance();  startcal.setTime(strday);  Date endday = sdf.parse(enddate);  Calendar endcal = Calendar.getInstance();  endcal.setTime(endday);  if (color.equals("red")) {  for(;startcal.compareTo(endcal) <= 0;startcal.add(Calendar.DATE, 1)){  Calendar c\_temp = Calendar.getInstance();  c\_temp.setTime(startcal.getTime());  events.add(new EventDay(c\_temp, R.drawable.ic\_red));  }  mCalendarView.setEvents(events);  }else if (color.equals("green")) {  for(;startcal.compareTo(endcal) <= 0;startcal.add(Calendar.DATE, 1)){  Calendar c\_temp = Calendar.getInstance();  c\_temp.setTime(startcal.getTime());  events.add(new EventDay(c\_temp, R.drawable.ic\_green));  }  mCalendarView.setEvents(events);  }else if (color.equals("orange")) {  for(;startcal.compareTo(endcal) <= 0;startcal.add(Calendar.DATE, 1)){  Calendar c\_temp = Calendar.getInstance();  c\_temp.setTime(startcal.getTime());  events.add(new EventDay(c\_temp, R.drawable.ic\_orange));  }  mCalendarView.setEvents(events);  }else {  for(;startcal.compareTo(endcal) <= 0;startcal.add(Calendar.DATE, 1)){  Calendar c\_temp = Calendar.getInstance();  c\_temp.setTime(startcal.getTime());  events.add(new EventDay(c\_temp, R.drawable.ic\_yellow));  }  mCalendarView.setEvents(events);  }  }  mCalendarView.setEvents(events);  }  });  }  }  }  });  } | | | |

### 表9-2-37 函式導向

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-6-1 | 函式名稱 | onCreate |
| 功能 | 函式導向 | | |
| @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_pets\_info);  ImageButton camera = findViewById(R.id.camera1);  ImageButton photo = findViewById(R.id.album);  camera.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent intent = new Intent();  intent.setAction("android.media.action.IMAGE\_CAPTURE");  File file = new File(FILE\_PATH);  Intent takePhotoIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);  Uri uri = FileProvider.getUriForFile(  v.getContext(),  petInfoActivity.this.getPackageName() + ".provider",  file);  takePhotoIntent.putExtra(MediaStore.EXTRA\_OUTPUT, uri);  startActivityForResult(intent, CAMERA);  }  });  photo.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent intent = new Intent();  intent.setType("image/\*");  intent.setAction(Intent.ACTION\_GET\_CONTENT);  startActivityForResult(intent, PHOTO);  }  });  } | | | |

### 表9-2-38 取得相機、相簿權限

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-6-2 | 函式名稱 | getPermission |
| 功能 | 取得相機、相簿權限 | | |
| private void getPermission() {  List<String> permissionList = new ArrayList<>();  if (ContextCompat.checkSelfPermission(petInfoActivity.this, Manifest.permission.CAMERA)  != PackageManager.PERMISSION\_GRANTED) {  permissionList.add(Manifest.permission.CAMERA);  }  if (ContextCompat.checkSelfPermission(petInfoActivity.this, Manifest.permission.WRITE\_EXTERNAL\_STORAGE)  != PackageManager.PERMISSION\_GRANTED) {  permissionList.add(Manifest.permission.WRITE\_EXTERNAL\_STORAGE);  }  if (ContextCompat.checkSelfPermission(petInfoActivity.this, Manifest.permission.READ\_EXTERNAL\_STORAGE)  != PackageManager.PERMISSION\_GRANTED) {  permissionList.add(Manifest.permission.READ\_EXTERNAL\_STORAGE);  }  if (!permissionList.isEmpty()) {  ActivityCompat.requestPermissions(petInfoActivity.this,  permissionList.toArray(new String[permissionList.size()]), 1002);  }  } | | | |

### 表9-2-39 讀取照片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-6-3 | 函式名稱 | onActivityResult |
| 功能 | 讀取照片 | | |
| @Override  public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  mPhone = new DisplayMetrics();  petInfoActivity.this.getWindowManager().getDefaultDisplay().getMetrics(mPhone);  if (requestCode == PHOTO && data != null) {  Uri uri = data.getData();  ContentResolver cr = petInfoActivity.this.getContentResolver();  Bitmap bitmap = BitmapFactory.decodeStream(cr.openInputStream(uri));  if (bitmap.getWidth() > bitmap.getHeight()) scaleImage(bitmap,  mPhone.heightPixels);  else scaleImage(bitmap, mPhone.widthPixels);  }  if (requestCode == CAMERA && data != null) {  Bitmap photo = null;  if (data.getData() != null || data.getExtras() != null) {  Uri camerauri = data.getData();  if (camerauri != null) {  photo = BitmapFactory.decodeFile(camerauri.getPath());  }  if (photo == null) {  Bundle bundle = data.getExtras();  if (bundle != null) {  photo = (Bitmap) bundle.get("data");  }  }  if (photo.getWidth() > photo.getHeight()) scaleImage(photo, mPhone.heightPixels);  else scaleImage(photo, mPhone.widthPixels);  }  }  super.onActivityResult(requestCode, resultCode, data);  } | | | |

### 表9-2-40 調整照片大小

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-6-4 | 函式名稱 | scaleImage |
| 功能 | 調整照片大小 | | |
| private void scaleImage(Bitmap bitmap, int phone) {  float mScale = 1;  img = findViewById(R.id.img);  if (bitmap.getWidth() > phone) {  mScale = (float) phone / (float) bitmap.getWidth();  Matrix mMat = new Matrix();  mMat.setScale(mScale, mScale);  Bitmap mScaleBitmap = Bitmap.createBitmap(bitmap,  0,  0,  bitmap.getWidth(),  bitmap.getHeight(),  mMat,  false);  img.setImageBitmap(mScaleBitmap);  } else img.setImageBitmap(bitmap);  } | | | |

### 表9-2-41 新增寵物資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-6-5 | 函式名稱 | addPetInfo |
| 功能 | 新增寵物資訊 | | |
| private void addPetInfo(String petsimage) {  EditText edpetsname = findViewById(R.id.petsname);  final EditText edpetsbirth = findViewById(R.id.petsbirth);  Spinner variety = findViewById(R.id.variety);  int idsex = variety.getSelectedItemPosition();  String[] Spinner\_repeat = getResources().getStringArray(R.array.Spinner\_variety);  String breed = Spinner\_repeat[idsex];  final EditText edlikes = findViewById(R.id.likes);  final EditText ednotes = findViewById(R.id.notes);  String petsbirth = edpetsbirth.getText().toString();  final String petsname = edpetsname.getText().toString();  String petslikes = edlikes.getText().toString();  String petsnotes = ednotes.getText().toString();  String petsgender = null;  RadioGroup sexselect = findViewById(R.id.sexselect);  switch(sexselect.getCheckedRadioButtonId()){  case R.id.malepet:  petsgender = "男孩";  break;  case R.id.femalepet:  petsgender = "女孩";  break;  }  Calendar now = Calendar.getInstance();  Date nowdate = now.getTime();  FirebaseFirestore db = FirebaseFirestore.getInstance();  Map<String, Object> userInfo = new HashMap<>();  userInfo.put("petName", petsname);  userInfo.put("petGender", petsgender);  userInfo.put("petBirth", petsbirth);  userInfo.put("breed", breed);  userInfo.put("petHobby", petslikes);  userInfo.put("petNote", petsnotes);  userInfo.put("petImage", petsimage);  userInfo.put("uid", userUID);  userInfo.put("timestamp", nowdate);  userInfo.put("petBirth\_year", petBirth\_year);  userInfo.put("petBirth\_month", petBirth\_month);  userInfo.put("petBirth\_date",petBirth\_date);  db.collection("users").document(userUID).collection("pets").document(id).set(userInfo);  AlertDialog.Builder finishsignup = new AlertDialog.Builder(petInfoActivity.this);  finishsignup.setMessage("新增成功");  finishsignup.setNegativeButton("確認", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  Intent intent=new Intent();  intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK|Intent.FLAG\_ACTIVITY\_NEW\_TASK);  intent.setClass(petInfoActivity.this, homeActivity.class);  startActivity(intent);  }  });  finishsignup.setCancelable(false);  finishsignup.show();  } | | | |

### 表9-2-42 上傳圖片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-6-6 | 函式名稱 | addImage |
| 功能 | 上傳圖片 | | |
| private void addImage() {  EditText edpetsname = findViewById(R.id.petsname);  final String petsname = edpetsname.getText().toString();  FirebaseStorage storage = FirebaseStorage.getInstance("gs://ipets-APP.APPspot.com");  StorageReference mStorageRef = storage.getReference();  StorageReference mountainsRef = mStorageRef.child(userUID +'/'+ petsname+".jpg");  Bitmap bitmap = ((BitmapDrawable) img.getDrawable()).getBitmap();  ByteArrayOutputStream baos = new ByteArrayOutputStream();  bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos);  byte[] data = baos.toByteArray();  UploadTask uploadTask = mountainsRef.putBytes(data);  String petsimage = String.valueOf(mountainsRef.getDownloadUrl());  addPetInfo(petsimage);  } | | | |

### 表9-2-43 新增行程

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-7-1 | 函式名稱 | addEvent |
| 功能 | 新增行程 | | |
| private void addEvent() {  public void add(String start, String end){  db = FirebaseFirestore.getInstance();  userInfo.put("eventName", note);  userInfo.put("startDate", start);  userInfo.put("endDate", end);  userInfo.put("frequency", repeat);  userInfo.put("details", detail);  userInfo.put("color", colorselect);  db.collection("users").document(userUID).collection("calEvent").document().set(userInfo);  };  public void alert(){  AlertDialog.Builder finishsignup = new AlertDialog.Builder(addNoteActivity.this);  finishsignup.setMessage("新增成功");  finishsignup.setNegativeButton("確認", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  Intent intent = new Intent();  intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);  intent.setClass(addNoteActivity.this, calendarActivity.class);  startActivity(intent);  }  });  finishsignup.setCancelable(false);  finishsignup.show();  };  FirebaseAuth auth = FirebaseAuth.getInstance();  FirebaseUser currentUser = auth.getCurrentUser();  String userUID = currentUser.getUid();  TextInputEditText ednote = findViewById(R.id.noteEditText);  String note = ednote.getText().toString();  TextInputEditText detailEditText = findViewById(R.id.detailEditText);  String detail = detailEditText.getText().toString();  String colorselect = null;  RadioGroup eventColor = findViewById(R.id.eventColor);  switch (eventColor.getCheckedRadioButtonId()) {  case R.id.redCircle1:  colorselect = "red";  break;  case R.id.orangeCircle1:  colorselect = "orange";  break;  case R.id.yellowCircle1:  colorselect = "yellow";  break;  case R.id.greenCircle1:  colorselect = "green";  break;  }  Spinner repeatSpinner = findViewById(R.id.repeatSpinner);  int idsex = repeatSpinner.getSelectedItemPosition();  String[] Spinner\_repeat = getResources().getStringArray(R.array.Spinner\_repeat);  String repeat = Spinner\_repeat[idsex];  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  String startday = sdf.format(startCalendar.getTime());  String endday = sdf.format(endCalendar.getTime());  add(start, end);  if(repeat.equals("不重複")){  alert();  }else if (repeat.equals("每日")) {  for (int i = 0; i < 4; i++) {  startCalendar.add(Calendar.DAY\_OF\_MONTH,1);  endCalendar.add(Calendar.DAY\_OF\_MONTH,1);  String start = sdf.format(startCalendar.getTime());  String end = sdf.format(endCalendar.getTime());  add(start, end);  }  alert();  }else if (repeat.equals("每週")) {  for (int i = 0; i < 4; i++) {  startCalendar.add(Calendar.DAY\_OF\_MONTH,7);  endCalendar.add(Calendar.DAY\_OF\_MONTH,7);  String start = sdf.format(startCalendar.getTime());  String end = sdf.format(endCalendar.getTime());  add(start, end);  }  alert();  }else if (repeat.equals("每月")) {  for (int i = 0; i < 4; i++) {  startCalendar.add(Calendar.MONTH,1);  endCalendar.add(Calendar.MONTH,1);  String start = sdf.format(startCalendar.getTime());  String end = sdf.format(endCalendar.getTime());  add(start, end);  }  alert();  }else {  for (int i = 0; i < 4; i++) {  startCalendar.add(Calendar.YEAR,1);  endCalendar.add(Calendar.YEAR,1);  String start = sdf.format(startCalendar.getTime());  String end = sdf.format(endCalendar.getTime());  add(start, end);  }  alert();  }  } | | | |

### 表9-2-44 刪除行程

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-8-1 | 函式名稱 | deleteEvent |
| 功能 | 刪除行程 | | |
| private void deleteEvent() {  FirebaseAuth auth = FirebaseAuth.getInstance();  FirebaseUser currentUser = auth.getCurrentUser();  String userUID = currentUser.getUid();  FirebaseFirestore db;  db = FirebaseFirestore.getInstance();  db.collection("users").document(userUID).collection("calEvent").document(id).delete();  AlertDialog.Builder finishdelete = new AlertDialog.Builder(getActivity());  finishdelete.setMessage("刪除成功");  finishdelete.setNegativeButton("確認", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  Intent intent=new Intent();  intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK|Intent.FLAG\_ACTIVITY\_NEW\_TASK);  intent.setClass(getActivity(),calendarActivity.class);  startActivity(intent);  }  });  finishdelete.setCancelable(false);  finishdelete.show();  } | | | |

### 表9-2-45 更新行程

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-8-2 | 函式名稱 | setEvent |
| 功能 | 更新行程 | | |
| private void setEvent() {  EditText editNoteText = getView().findViewById(R.id.editNoteText);  EditText noteContentText = getView().findViewById(R.id.noteContentText);  FirebaseAuth auth = FirebaseAuth.getInstance();  FirebaseUser currentUser = auth.getCurrentUser();  String userUID = currentUser.getUid();  FirebaseFirestore db = FirebaseFirestore.getInstance();  String Note = editNoteText.getText().toString();  String Content = noteContentText.getText().toString();  RadioGroup eventColor2 = getView().findViewById(R.id.eventColor2);  switch (eventColor2.getCheckedRadioButtonId()) {  case R.id.redCircle2:  colorselect = "red";  break;  case R.id.orangeCircle2:  colorselect = "orange";  break;  case R.id.yellowCircle2:  colorselect = "yellow";  break;  case R.id.greenCircle2:  colorselect = "green";  break;  }  Map<String, Object> userInfo = new HashMap<>();  userInfo.put("eventName",Note);  userInfo.put("details",Content);  userInfo.put("color", colorselect);  db.collection("users").document(userUID).collection("calEvent").document(id).update(userInfo);  AlertDialog.Builder finishsignup = new AlertDialog.Builder(getActivity());  finishsignup.setMessage("修改成功");  finishsignup.setNegativeButton("確認", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  Intent intent=new Intent();  intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK|Intent.FLAG\_ACTIVITY\_NEW\_TASK);  intent.setClass(getActivity(),calendarActivity.class);  startActivity(intent);  }  });  finishsignup.setCancelable(false);  finishsignup.show();  } | | | |

### 表9-2-46 登出

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-9-1 | 函式名稱 | onCreate |
| 功能 | 登出 | | |
| @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_setting);  Button signout = findViewById(R.id.signout);  signout.setOnClickListener(new Button.OnClickListener(){  @Override  public void onClick(View v) {  FirebaseAuth auth = FirebaseAuth.getInstance();  auth.signOut();  Intent intent=new Intent();  intent.setClass(settingActivity.this,MainActivity.class);  startActivity(intent);  }  });  } | | | |

### 表9-2-47 傳值至更新會員帳號

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-10-1 | 函式名稱 | onCreate |
| 功能 | 傳值至更新會員帳號 | | |
| @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_edit\_masterinfo);  btn\_editAcct = (LinearLayout) findViewById(R.id.btnAccount);  btn\_editAcct.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent intent = new Intent(editUserInfoActivity.this, editAccountActivity.class);  intent.putExtra("email",email);  intent.putExtra("password",password);  startActivity(intent);  }  });  } | | | |

### 表9-2-48 更新會員資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-10-2 | 函式名稱 | setUserInfo |
| 功能 | 更新會員資料 | | |
| public void setUserInfo(){  FirebaseAuth auth = FirebaseAuth.getInstance();  FirebaseUser currentUser = auth.getCurrentUser();  String userUID = currentUser.getUid();  FirebaseFirestore db = FirebaseFirestore.getInstance();  String[] Spinner\_sex =getResources().getStringArray(R.array.Spinner\_sex);  TextInputEditText edmasterName = findViewById(R.id.masterName);  TextInputEditText eduserName = findViewById(R.id.userName);  EditText edmasterBirth = findViewById(R.id.masterBirth);  TextInputEditText edphoneNum = findViewById(R.id.phoneNum);  TextInputEditText edmasterAddress = findViewById(R.id.masterAddress);  String masterName = edmasterName.getText().toString();  String masterBirth = edmasterBirth.getText().toString();  String userName = eduserName.getText().toString();  String phoneNum = edphoneNum .getText().toString();  String masterAddress = edmasterAddress.getText().toString();  Spinner sexSpinner = findViewById(R.id.sexSpinner);  int idsex=sexSpinner.getSelectedItemPosition();  String sex = Spinner\_sex[idsex];  Map<String, Object> userInfo = new HashMap<>();  userInfo.put("name",masterName);  userInfo.put("userName",userName);  userInfo.put("userBirth", masterBirth);  userInfo.put("phone", phoneNum);  userInfo.put("address", masterAddress);  userInfo.put("userGender", sex);  userInfo.put("userBirth\_year", userBirth\_year);  userInfo.put("userBirth\_month", userBirth\_month);  userInfo.put("userBirth\_date",userBirth\_date);  db.collection("users").document(userUID).update(userInfo);  AlertDialog.Builder finishsignup = new AlertDialog.Builder(editUserInfoActivity.this);  finishsignup.setMessage("修改成功");  finishsignup.setNegativeButton("確認", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  Intent intent=new Intent();  intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK|Intent.FLAG\_ACTIVITY\_NEW\_TASK);  intent.setClass(editUserInfoActivity.this, homeActivity.class);  startActivity(intent);  }  });  finishsignup.setCancelable(false);  finishsignup.show();  } | | | |

### 表9-2-49 將EditText轉換成日期選擇器

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-10-3 | 函式名稱 | setEditText |
| 功能 | 將EditText轉換成日期選擇器 | | |
| public void setEditText(){  EditText edmasterBirth = findViewById(R.id.masterBirth);  edmasterBirth.setInputType(InputType.TYPE\_NULL);  edmasterBirth.setOnFocusChangeListener(new View.OnFocusChangeListener() {  @Override  public void onFocusChange(View v, boolean hasFocus) {  if(hasFocus){  Calendar c = Calendar.getInstance();  DatePickerDialog datePickerDialog = new DatePickerDialog(v.getContext(),R.style.MyDatePickerDialogTheme, new DatePickerDialog.OnDateSetListener() {  @Override  public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {  edmasterBirth.setText(year+"-"+(monthOfYear+1)+"-"+dayOfMonth);  userBirth\_year = year;  userBirth\_month = monthOfYear+1;  userBirth\_date = dayOfMonth;  }  }, c.get(Calendar.YEAR), c.get(Calendar.MONTH), c.get(Calendar.DAY\_OF\_MONTH));  DatePicker datePicker = datePickerDialog.getDatePicker();  datePicker.setMaxDate(System.currentTimeMillis());  datePickerDialog.show();  }  }  });  } | | | |

### 表9-2-50 取得會員資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-10-4 | 函式名稱 | getUserInfo |
| 功能 | 取得會員資料 | | |
| public void getUserInfo(){  EditText edmasterName = findViewById(R.id.masterName);  final EditText eduserName = findViewById(R.id.userName);  final EditText edmasterBirth = findViewById(R.id.masterBirth);  final EditText edphoneNum = findViewById(R.id.phoneNum);  final EditText edmasterAddress = findViewById(R.id.masterAddress);  Spinner sexSpinner = findViewById(R.id.sexSpinner);  FirebaseAuth auth = FirebaseAuth.getInstance();  FirebaseUser currentUser = auth.getCurrentUser();  String userUID = currentUser.getUid();  FirebaseFirestore db;  db = FirebaseFirestore.getInstance();  db.collection("users").document(userUID)  .get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {  @Override  public void onComplete(@NonNull Task<DocumentSnapshot> task) {  if (task.isSuccessful()) {  DocumentSnapshot doc = task.getResult();  StringBuilder fields = new StringBuilder("");  StringBuilder fields2 = new StringBuilder("");  StringBuilder fields3 = new StringBuilder("");  StringBuilder fields4 = new StringBuilder("");  StringBuilder fields5 = new StringBuilder("");  StringBuilder fields6 = new StringBuilder("");  StringBuilder fields7 = new StringBuilder("");  StringBuilder fields8= new StringBuilder("");  fields.APPend(doc.get("name")).toString();  fields2.APPend(doc.get("userName")).toString();  fields3.APPend(doc.get("userBirth")).toString();  fields4.APPend(doc.get("phone")).toString();  fields5.APPend(doc.get("address")).toString();  String gender = fields6.APPend(doc.get("userGender")).toString();  email = fields7.APPend(doc.get("Email")).toString();  password = fields8.APPend(doc.get("password")).toString();  edmasterName.setText(fields);  eduserName.setText(fields2);  edmasterBirth.setText(fields3);  edphoneNum.setText(fields4);  edmasterAddress.setText(fields5);  if(gender.equals("女")) {  sexSpinner.setSelection(1);  }  if(gender.equals("不願透露")) {  sexSpinner.setSelection(2);  }  }  }  })  } | | | |

### 表9-2-51 更新會員帳號

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-11-1 | 函式名稱 | setAccount |
| 功能 | 更新會員帳號 | | |
| public void setAccount(){  FirebaseAuth auth = FirebaseAuth.getInstance();  FirebaseUser currentUser = auth.getCurrentUser();  String userUID = currentUser.getUid();  FirebaseFirestore db = FirebaseFirestore.getInstance();  Intent intent = this.getIntent();  String oldemail = intent.getStringExtra("email");  String oldpassword = intent.getStringExtra("password");  TextInputEditText editAcct = findViewById(R.id.editAcct);  String email = editAcct.getText().toString();  AuthCredential credential = EmailAuthProvider  .getCredential(oldemail, oldpassword);  currentUser.reauthenticate(credential)  .addOnCompleteListener(new OnCompleteListener<Void>() {  @Override  public void onComplete(@NonNull Task<Void> task) {  currentUser.updateEmail(email);  }  });  Map<String, Object> userInfo = new HashMap<>();  userInfo.put("email",email);  db.collection("users").document(userUID).update(userInfo);  AlertDialog.Builder finishsignup = new AlertDialog.Builder(editAccountActivity.this);  finishsignup.setMessage("修改成功");  finishsignup.setNegativeButton("確認", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  Intent intent=new Intent();  intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK|Intent.FLAG\_ACTIVITY\_NEW\_TASK);  intent.setClass(editAccountActivity.this, homeActivity.class);  startActivity(intent);  }  });  finishsignup.setCancelable(false);  finishsignup.show();  } | | | |

### 表9-2-52 取得會員帳號

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-11-2 | 函式名稱 | getAccount |
| 功能 | 取得會員帳號 | | |
| public void getAccount(){  Intent intent = this.getIntent();  String email = intent.getStringExtra("email");  TextInputEditText editAcct = findViewById(R.id.editAcct);  editAcct.setText(email);  } | | | |

### 表9-2-53 更新會員密碼

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-12-1 | 函式名稱 | setPwd |
| 功能 | 更新會員密碼 | | |
| public void setPwd(){  FirebaseAuth auth = FirebaseAuth.getInstance();  FirebaseUser currentUser = auth.getCurrentUser();  String userUID = currentUser.getUid();  FirebaseFirestore db = FirebaseFirestore.getInstance();  EditText edoldPWD = findViewById(R.id.oldPWD);  EditText ednewPWD = findViewById(R.id.newPWD);  EditText edcheckPWD = findViewById(R.id.checkPWD);  String oldPWD = edoldPWD.getText().toString();  String newPWD = ednewPWD.getText().toString();  String checkPWD = edcheckPWD.getText().toString();  String email = currentUser.getEmail();  if(newPWD.equals(checkPWD)) {  AuthCredential credential = EmailAuthProvider  .getCredential( email, oldPWD);  currentUser.reauthenticate(credential)  .addOnCompleteListener(new OnCompleteListener<Void>() {  @Override  public void onComplete(@NonNull Task<Void> task) {  currentUser.updatePassword(checkPWD);  }  });  Map<String, Object> userInfo = new HashMap<>();  userInfo.put("password", checkPWD);  db.collection("users").document(userUID).update(userInfo);  AlertDialog.Builder finishsignup = new AlertDialog.Builder(editPwdActivity.this);  finishsignup.setMessage("修改成功");  finishsignup.setNegativeButton("確認", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  Intent intent = new Intent();  intent.setClass(editPwdActivity.this, homeActivity.class);  startActivity(intent);  }  });  finishsignup.setCancelable(false);  finishsignup.show();  }  } | | | |

### 表9-2-54 函式導向

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-13-1 | 函式名稱 | onCreate |
| 功能 | 函式導向 | | |
| @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_edit\_petsinfo);  ImageButton album = findViewById(R.id.album);  album.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent intent = new Intent();  intent.setType("image/\*");  intent.setAction(Intent.ACTION\_GET\_CONTENT);  startActivityForResult(intent, PHOTO);  }  });  ImageButton camera1 = findViewById(R.id.camera1);  camera1.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent intent = new Intent();  intent.setAction("android.media.action.IMAGE\_CAPTURE");  startActivityForResult(intent, CAMERA);  }  });  } | | | |

### 表9-2-55 取得寵物照片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-13-2 | 函式名稱 | getPetImage |
| 功能 | 取得寵物照片 | | |
| public void getPetImage(){  Intent intent = this.getIntent();  String pet = intent.getStringExtra("date");  FirebaseStorage storage = FirebaseStorage.getInstance("gs://ipets-APP.APPspot.com");  StorageReference mStorageRef = storage.getReference();  StorageReference islandRef = mStorageRef.child(userUID +'/'+ pet+".jpg");  img = findViewById(R.id.petsHead);  final long ONE\_MEGABYTE = 3000 \* 3000;  islandRef.getBytes(ONE\_MEGABYTE).addOnSuccessListener(new OnSuccessListener<byte[]>() {  @Override  public void onSuccess(byte[] bytes) {  Glide.with(getAPPlicationContext())  .load(bytes)  .into(img);  }  })  } | | | |

### 表9-2-56 取得寵物資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-13-3 | 函式名稱 | getPetInfo |
| 功能 | 取得寵物資訊 | | |
| private void getPetInfo() {  EditText edmypetName = findViewById(R.id.mypetName);  final EditText edpetsbirth = findViewById(R.id.mypetBirth);  Spinner petSexSpinner = findViewById(R.id.petSexSpinner);  EditText edhobbyText = findViewById(R.id.hobbyText);  final EditText edremarkText = findViewById(R.id.remarkText);  Intent intent = this.getIntent();  String pet = intent.getStringExtra("date");  FirebaseFirestore db;  db = FirebaseFirestore.getInstance();  db.collection("users").document(userUID).collection("pets").whereEqualTo("petName", pet)  .get()  .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  @Override  public void onComplete(@NonNull Task<QuerySnapshot> task) {  if (task.isSuccessful()) {  for (QueryDocumentSnapshot document : task.getResult()) {  documentname = document.getId();  DocumentSnapshot doc = document;  StringBuilder fields = new StringBuilder("");  StringBuilder fields2 = new StringBuilder("");  StringBuilder fields3 = new StringBuilder("");  StringBuilder fields4 = new StringBuilder("");  StringBuilder fields5 = new StringBuilder("");  oldpet = fields.APPend(doc.get("petName")).toString();  String Petsgender = fields2.APPend(doc.get("petGender")).toString();  fields3.APPend(doc.get("petBirth")).toString();  fields4.APPend(doc.get("petHobby")).toString();  fields5.APPend(doc.get("petNote")).toString();  edmypetName.setText(fields);  edpetsbirth.setText(fields3);  edhobbyText.setText(fields4);  edremarkText.setText(fields5);  if(Petsgender.equals("女孩")) {  petSexSpinner.setSelection(1);  }  }  }  }  });  } | | | |

### 表9-2-57 上傳圖片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-13-4 | 函式名稱 | addImage |
| 功能 | 上傳圖片 | | |
| private void addImage() {  EditText edmypetName = findViewById(R.id.mypetName);  final String mypetName = edmypetName.getText().toString();  FirebaseStorage storage = FirebaseStorage.getInstance("gs://ipets-APP.APPspot.com");  StorageReference mStorageRef = storage.getReference();  StorageReference mountainsRef = mStorageRef.child(userUID +'/'+ mypetName+".jpg");  Bitmap bitmap = ((BitmapDrawable) img.getDrawable()).getBitmap();  ByteArrayOutputStream baos = new ByteArrayOutputStream();  bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos);  byte[] data = baos.toByteArray();  UploadTask uploadTask = mountainsRef.putBytes(data);  String petsimage = String.valueOf(mountainsRef.getDownloadUrl());  setPetInfo(petsimage,mypetName);  } | | | |

### 表9-2-58 更新寵物資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-13-5 | 函式名稱 | setPetInfo |
| 功能 | 更新寵物資訊 | | |
| private void setPetInfo(String petsimage, String mypetName) {  final EditText edpetsbirth = findViewById(R.id.mypetBirth);  String petsbirth = edpetsbirth.getText().toString();  SimpleDateFormat sdf= new SimpleDateFormat("yyyy-MM-dd");  date = sdf.parse(petsbirth);  Calendar calendar = Calendar.getInstance();  calendar.setTime(date);  EditText edhobbyText = findViewById(R.id.hobbyText);  String hobbyText = edhobbyText.getText().toString();  final EditText edremarkText = findViewById(R.id.remarkText);  String remarkText = edremarkText.getText().toString();  String[] petSex =getResources().getStringArray(R.array.Spinner\_petSex);  Spinner petSexSpinner = findViewById(R.id.petSexSpinner);  int idsex = petSexSpinner.getSelectedItemPosition();  String sex = petSex[idsex];  FirebaseFirestore db = FirebaseFirestore.getInstance();  Map<String, Object> userInfo = new HashMap<>();  userInfo.put("petImage", petsimage);  userInfo.put("petName", mypetName);  userInfo.put("petBirth", petsbirth);  userInfo.put("petGender", sex);  userInfo.put("petHobby", hobbyText);  userInfo.put("petNote", remarkText);  userInfo.put("petBirth\_year", petBirth\_year);  userInfo.put("petBirth\_month", petBirth\_month);  userInfo.put("petBirth\_date",petBirth\_date);  db.collection("users").document(userUID).collection("pets").document(documentname).update(userInfo);  AlertDialog.Builder finishsignup = new AlertDialog.Builder(editPetInfoActivity.this);  finishsignup.setMessage("修改成功");  finishsignup.setNegativeButton("確認", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface arg0, int arg1) {  Intent intent=new Intent();  intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK|Intent.FLAG\_ACTIVITY\_NEW\_TASK);  intent.setClass(editPetInfoActivity.this, homeActivity.class);  startActivity(intent);  }  });  finishsignup.setCancelable(false);  finishsignup.show();  } | | | |

### 表9-2-59 將EditText轉換成日期選擇器

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-13-6 | 函式名稱 | setEditText |
| 功能 | 將EditText轉換成日期選擇器 | | |
| public void setEditText(){  EditText edmypetBirth = findViewById(R.id.mypetBirth);  edmypetBirth.setInputType(InputType.TYPE\_NULL);  edmypetBirth.setOnFocusChangeListener(new View.OnFocusChangeListener() {  @Override  public void onFocusChange(View v, boolean hasFocus) {  if(hasFocus){  Calendar c = Calendar.getInstance();  DatePickerDialog datePickerDialog = new DatePickerDialog(v.getContext(),R.style.MyDatePickerDialogTheme, new DatePickerDialog.OnDateSetListener() {  @Override  public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {  edmypetBirth.setText(year+"-"+(monthOfYear+1)+"-"+dayOfMonth);  petBirth\_year = year;  petBirth\_month = monthOfYear+1;  petBirth\_date = dayOfMonth;  }  }, c.get(Calendar.YEAR), c.get(Calendar.MONTH), c.get(Calendar.DAY\_OF\_MONTH));  DatePicker datePicker = datePickerDialog.getDatePicker();  datePicker.setMaxDate(System.currentTimeMillis());  datePickerDialog.show();  }  }  });  } | | | |

### 表9-2-60 讀取照片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-13-7 | 函式名稱 | onActivityResult |
| 功能 | 讀取照片 | | |
| @Override  public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  mPhone = new DisplayMetrics();  editPetInfoActivity.this.getWindowManager().getDefaultDisplay().getMetrics(mPhone);  if (requestCode == PHOTO && data != null) {  Uri uri = data.getData();  ContentResolver cr = editPetInfoActivity.this.getContentResolver();  Bitmap bitmap = BitmapFactory.decodeStream(cr.openInputStream(uri));  if (bitmap.getWidth() > bitmap.getHeight()) scaleImage(bitmap,  mPhone.heightPixels);  else scaleImage(bitmap, mPhone.widthPixels);  }  if (requestCode == CAMERA && data != null) {  Bitmap photo = null;  if (data.getData() != null || data.getExtras() != null) {  Uri camerauri = data.getData();  if (camerauri != null) {  photo = BitmapFactory.decodeFile(camerauri.getPath());  }  if (photo == null) {  if (data.getExtras() != null) {  photo = (Bitmap) data.getExtras().get("data");  }  }  if (photo.getWidth() > photo.getHeight()) scaleImage(photo,  mPhone.heightPixels);  else scaleImage(photo, mPhone.widthPixels);  }  }  super.onActivityResult(requestCode, resultCode, data);  } | | | |

### 表9-2-61 調整照片大小

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-13-8 | 函式名稱 | scaleImage |
| 功能 | 調整照片大小 | | |
| private void scaleImage(Bitmap bitmap, int phone) {  float mScale = 1;  img = findViewById(R.id.petsHead);  if (bitmap.getWidth() > phone) {  mScale = (float) phone / (float) bitmap.getWidth();  Matrix mMat = new Matrix();  mMat.setScale(mScale, mScale);  Bitmap mScaleBitmap = Bitmap.createBitmap(bitmap,  0,  0,  bitmap.getWidth(),  bitmap.getHeight(),  mMat,  false);  img.setImageBitmap(mScaleBitmap);  } else img.setImageBitmap(bitmap);  } | | | |

### 表9-2-62 顯示常見問題

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-14-1 | 函式名稱 | onCreate |
| 功能 | 顯示常見問題 | | |
| @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_edit\_question);  btn\_q1 = findViewById(R.id.button\_q1);  btn\_q1.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  new AlertDialog.Builder(EditQuestionActivity.this)  .setTitle("BUG通知")  .setMessage("非常抱歉造成各位使用上的不便，iPets對於已確認狀況的問題和對策進行了更新，在APP版本無法讀取的狀態下，請試著使用網頁版本。")  .setPositiveButton("了解", new DialogInterface.OnClickListener() {  }).show();  }  });  btn\_q2 = findViewById(R.id.button\_q2);  btn\_q2.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  new AlertDialog.Builder(EditQuestionActivity.this)  .setTitle("錯誤403")  .setMessage("錯誤403表示未能連接至iPets伺服器。要解決這個問題，您可以重新登入或者重新安裝iPets應用程式。")  .setPositiveButton("了解", new DialogInterface.OnClickListener() {  }).show();  }  });  btn\_q3 = findViewById(R.id.button\_q3);  btn\_q3.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  new AlertDialog.Builder(EditQuestionActivity.this)  .setTitle("註冊帳號")  .setMessage("以電子郵件信箱註冊")  .setPositiveButton("了解", new DialogInterface.OnClickListener() {  }).show();  }  });  btn\_q4 = findViewById(R.id.button\_q4);  btn\_q4.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  new AlertDialog.Builder(EditQuestionActivity.this)  .setTitle("如何重新登入?")  .setMessage("請謹記登出前沒有建立一個iPets帳號，所有資料將會移師且不能復原。" +  "請按照以下步驟重新登入:" +  "1.選擇主目錄" + "2.點按您的頭像" + "3.點按登出" + "4.輸入您的登入資料和密碼，然後登入")  .setPositiveButton("了解", new DialogInterface.OnClickListener() {  }).show();  }  });  btn\_q5 = findViewById(R.id.button\_q5);  btn\_q5.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  new AlertDialog.Builder(EditQuestionActivity.this)  .setTitle("定位無法啟用?")  .setMessage("請跳出定位畫面，再點按一次定位鈕即可使用。")  .setPositiveButton("了解", new DialogInterface.OnClickListener() {  }).show();  }  });  btn\_q6 = findViewById(R.id.button\_q6);  btn\_q6.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  new AlertDialog.Builder(EditQuestionActivity.this)  .setTitle("能在無網路環境使用嗎?")  .setMessage("無網路環境下可以做的事" +  "1.在行事曆裡新增/編輯行程" + "2.使用相機拍照" + "3.瀏覽相簿照片" )  .setPositiveButton("了解", new DialogInterface.OnClickListener() {  }).show();  }  });  btn\_q7 = findViewById(R.id.button\_q7);  btn\_q7.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  new AlertDialog.Builder(EditQuestionActivity.this)  .setTitle("iPets行事曆能和其他行事曆同步嗎?")  .setMessage("iPets行事曆無法跟Google/APPle/Outlook行事曆共用。")  .setPositiveButton("了解", new DialogInterface.OnClickListener() {  }).show();  }  });  }  } | | | |

### 表9-2-63 新增聯絡我們資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-15-1 | 函式名稱 | addContactUs |
| 功能 | 新增聯絡我們資訊 | | |
| public void addContactUs() {  edmasterName = findViewById(R.id.masterName);  String masterName = edmasterName.getText().toString();  edemail = findViewById(R.id.email);  String Contactemail = edemail.getText().toString();  eddescription = findViewById(R.id.description);  String description = eddescription.getText().toString();  problemSpinner = findViewById(R.id.problemSpinner);  String problemtype = problemSpinner.getSelectedItem().toString();  FirebaseFirestore db = FirebaseFirestore.getInstance();  Map<String, Object> newContact = new HashMap<>();  newContact.put("MasterName", masterName);  newContact.put("Contactemail", Contactemail);  newContact.put("Description", description);  newContact.put("Problemtype", problemtype);  db.collection("userInformation").document(userUID).collection("ContactUs").document("Contactinfo").set(newContact);  } | | | |

### 表9-2-64 發送通知

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-16-1 | 函式名稱 | onReceive |
| 功能 | 發送通知 | | |
| @Override  public void onReceive(Context context, Intent intent) {  final FirebaseFirestore db = FirebaseFirestore.getInstance();  FirebaseAuth auth = FirebaseAuth.getInstance();  FirebaseUser currentUser = auth.getCurrentUser();  String userUID = currentUser.getUid();  db.collection("users").document(userUID).collection("pets").get()  .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  @Override  public void onComplete(@NonNull Task<QuerySnapshot> task) {  if (task.isSuccessful()) {  for (QueryDocumentSnapshot document : task.getResult()) {  db.collection("users").document(userUID).collection("pets").  document(document.getId()).collection("countdown").get()  .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  @Override  public void onComplete(@NonNull Task<QuerySnapshot> task) {  if (task.isSuccessful()) {  for (QueryDocumentSnapshot document : task.getResult()) {  StringBuilder field1 = new StringBuilder("");  String endDay = field1.APPend(document.get("endDay")).toString();  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  if(!endDay.equals("")) {  date = sdf.parse(endDay);  Calendar calendar = Calendar.getInstance();  calendar.setTime(date);  int pendingIntentid = calendar.get(Calendar.MONTH)+calendar.get(Calendar.DAY\_OF\_MONTH);  Intent notifyIntent = new Intent(context, homeActivity.class);  PendingIntent pendingIntent=PendingIntent.getActivity(context,pendingIntentid,notifyIntent,PendingIntent.FLAG\_ONE\_SHOT);  notificationManager=  (NotificationManager)context.getSystemService(Context.NOTIFICATION\_SERVICE);  int importance = NotificationManager.IMPORTANCE\_LOW;  NotificationChannel channel = new NotificationChannel("channel\_1", "123", importance);  channel.enableVibration(true);  channel.enableLights(true);  notificationManager.createNotificationChannel(channel);  Notification notification = new Notification.Builder(context, "channel\_1")  .setCategory(Notification.CATEGORY\_MESSAGE)  .setSmallIcon(R.drawable.APP\_logo1)  .setLargeIcon(BitmapFactory.decodeResource(context.getResources(),R.drawable.APP\_logo1))  .setContentTitle("倒數計時器")  .setContentText("某項行程該做囉!點擊確認")  .setContentIntent(pendingIntent)  .setAutoCancel(true).build();  notificationManager.notify(1,notification);  }  }  }  }  });  }  }  }  });  } | | | |

### 表9-2-65 確認有無安裝地圖

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-17-1 | 函式名稱 | displayTrack |
| 功能 | 確認有無安裝地圖 | | |
| private void displayTrack(String sSource,String sDestination){  try {  Uri uri = Uri.parse("https://www.google.co.in/maps/dir/" + sSource + "/" + sDestination);  Intent intent = new Intent(Intent.ACTION\_VIEW, uri);  intent.setPackage("com.google.android.APPs.maps");  intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);  startActivity(intent);  }catch (ActivityNotFoundException e){  Uri uri = Uri.parse("https://play.google.com/store/APPs/details?id=com.google.android.APPs.maps");  Intent intent = new Intent(Intent.ACTION\_VIEW,uri);  intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);  startActivity(intent);  }  } | | | |

### 表9-2-66 確認playService版本

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-17-2 | 函式名稱 | chkPlayService |
| 功能 | 確認playService版本 | | |
| private boolean chkPlayService() {  int avai = GoogleApiAvailability.getInstance().isGooglePlayServicesAvailable(MapsActivity.this);  if (avai == ConnectionResult.SUCCESS) {  return true;  } else {  Toast.makeText(this, "版本不符合，無法執行MAP", Toast.LENGTH\_LONG).show();  return false;  }  } | | | |

### 表9-2-67 確認權限

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-17-3 | 函式名稱 | onRequestPermissionsResult |
| 功能 | 確認權限 | | |
| @Override  public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  rLocationGranted = false;  switch (requestCode) {  case 1001: {  if (grantResults.length > 0) {  for (int i = 0; i < grantResults.length; i++) {  if (grantResults[i] != PackageManager.PERMISSION\_GRANTED) {  rLocationGranted = false;  return;  }  }  rLocationGranted = true;  }  }  }  } | | | |

### 表9-2-68 初始化地圖

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-17-4 | 函式名稱 | initialMap |
| 功能 | 初始化地圖 | | |
| private void initialMap() {  String[] permissions = {Manifest.permission.ACCESS\_FINE\_LOCATION, Manifest.permission.ACCESS\_COARSE\_LOCATION};  if ((ContextCompat.checkSelfPermission(this.getAPPlicationContext(), permissions[0])  == PackageManager.PERMISSION\_GRANTED ||  (ContextCompat.checkSelfPermission(this.getAPPlicationContext(), permissions[1])  == PackageManager.PERMISSION\_GRANTED))) {  rLocationGranted = true;  } else {  ActivityCompat.requestPermissions(this, permissions,  LOCATION\_PERMISSION\_REQUEST\_CODE  );  }  } | | | |

### 表9-2-69 執行初始化動作

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-17-5 | 函式名稱 | init |
| 功能 | 執行初始化動作 | | |
| private void init() {  mSearchText.setOnEditorActionListener(new TextView.OnEditorActionListener() {  @Override  public boolean onEditorAction(TextView v, int actionId, KeyEvent keyEvent) {  if (actionId == EditorInfo.IME\_ACTION\_SEARCH  || actionId == EditorInfo.IME\_ACTION\_DONE  || keyEvent.getAction() == KeyEvent.ACTION\_DOWN  || keyEvent.getAction() == KeyEvent.KEYCODE\_ENTER) {  geoLocate();  }  return false;  }  });  btTrack.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  String sSource = mSearchText.getText().toString().trim();  String sDestination = etDestination.getText().toString().trim();  if(!sSource.equals("") && !sDestination.equals("")){  DisplayTrack(sSource,sDestination);  }  }  });  } | | | |

### 表9-2-70 取得現在定位點

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-17-6 | 函式名稱 | getDeviceLocation |
| 功能 | 取得現在定位點 | | |
| private void getDeviceLocation() {  mFusedLocationProvider = LocationServices.getFusedLocationProviderClient(this);  try {  if (rLocationGranted == true) {  final Task location = mFusedLocationProvider.getLastLocation();  location.addOnCompleteListener(new OnCompleteListener() {  @Override  public void onComplete(@NonNull Task task) {  if (task.isSuccessful()) {  Location mLocation = (Location) task.getResult();  LatLng nowlatLng = new LatLng(mLocation.getLatitude(), mLocation.getLongitude());  mMap.animateCamera(CameraUpdateFactory.newLatLng(nowlatLng));  mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(nowlatLng, 17));  marker = mMap.addMarker(new MarkerOptions()  .position(nowlatLng)  .draggable(true)  .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_BLUE)));  }  }  });  }  } catch (Exception ex) {  }  } | | | |

### 表9-2-71 取得查詢地址定位點

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-17-7 | 函式名稱 | geoLocate |
| 功能 | 取得查詢地址定位點 | | |
| private void geoLocate() {  Log.d("geoLocate", "geoLocate : geolocating");  String searchString = mSearchText.getText().toString();  Geocoder geocoder = new Geocoder(MapsActivity.this);  List<Address> list = new ArrayList<>();  try {  list = geocoder.getFromLocationName(searchString, 1);  } catch (IOException e) {  }  if (list.size() > 0) {  Address address = list.get(0);  if (marker != null) {  marker.remove();  }  LatLng searchlatLng = new LatLng(address.getLatitude(), address.getLongitude());  mMap.animateCamera(CameraUpdateFactory.newLatLng(searchlatLng));  mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(searchlatLng, 17));  marker = mMap.addMarker(new MarkerOptions()  .position(searchlatLng)  .draggable(true));  }  } | | | |

### 表9-2-72 附近位置查詢

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-17-8 | 函式名稱 | onMapReady |
| 功能 | 附近位置查詢 | | |
| @Override  public void onMapReady(GoogleMap googleMap) {  mMap = googleMap;  getDeviceLocation();  Button btnPetStore = findViewById(R.id.PetStore);  btnPetStore.setOnClickListener(new View.OnClickListener() {  String PetStore = "PetStore";  @Override  public void onClick(View v) {  mMap.clear();  String url = getUrl(PetStore);  Object[] DataTransfer = new Object[2];  DataTransfer[0] = mMap;  DataTransfer[1] = url;  GetNearbyPlacesData getNearbyPlacesData = new GetNearbyPlacesData();  getNearbyPlacesData.execute(DataTransfer);  }  private String getUrl(String nearbyPlace) {  if (ActivityCompat.checkSelfPermission(  MapsActivity.this,Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(  MapsActivity.this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  ActivityCompat.requestPermissions(MapsActivity.this, new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION}, LOCATION\_PERMISSION\_REQUEST\_CODE);  } else {  LocationManager mgr = (LocationManager) getSystemService(Context.LOCATION\_SERVICE);  Location locationGPS = mgr.getLastKnownLocation(LocationManager.GPS\_PROVIDER);  if (locationGPS != null) {  latitude = locationGPS.getLatitude();  longitude = locationGPS.getLongitude();  Toast.makeText(MapsActivity.this, "latitude:" + latitude + " longitude:" + longitude, Toast.LENGTH\_SHORT).show();  StringBuilder googlePlacesUrl = new StringBuilder("https://maps.googleapis.com/maps/api/place/nearbysearch/json?key=AIzaSyBiU7Qs7b5GjLBZ8kxHrJU-2VOmRXR6XpY&radius=1000&type=pet\_store&sensor=true&");  googlePlacesUrl.APPend("location=" + latitude + "," + longitude);  return (googlePlacesUrl.toString());  }  }  return "";  }  });  Button btnPetSalon = findViewById(R.id.PetSalon);  btnPetSalon.setOnClickListener(new View.OnClickListener() {  String PetSalon = "PetSalon";  @Override  public void onClick(View v) {  mMap.clear();  String url = getUrl(PetSalon);  Object[] DataTransfer = new Object[2];  DataTransfer[0] = mMap;  DataTransfer[1] = url;  GetNearbyPlacesData getNearbyPlacesData = new GetNearbyPlacesData();  getNearbyPlacesData.execute(DataTransfer);  }  private String getUrl(String nearbyPlace) {  if (ActivityCompat.checkSelfPermission(  MapsActivity.this,Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(  MapsActivity.this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  ActivityCompat.requestPermissions(MapsActivity.this, new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION}, LOCATION\_PERMISSION\_REQUEST\_CODE);  } else {  LocationManager mgr = (LocationManager) getSystemService(Context.LOCATION\_SERVICE);  Location locationGPS = mgr.getLastKnownLocation(LocationManager.GPS\_PROVIDER);  if (locationGPS != null) {  latitude = locationGPS.getLatitude();  longitude = locationGPS.getLongitude();  Toast.makeText(MapsActivity.this, "latitude:" + latitude + " longitude:" + longitude, Toast.LENGTH\_SHORT).show();  StringBuilder googlePlacesUrl = new StringBuilder("https://maps.googleapis.com/maps/api/place/nearbysearch/json?key=AIzaSyBiU7Qs7b5GjLBZ8kxHrJU-2VOmRXR6XpY&radius=1000&type=pet\_store&sensor=true&");  googlePlacesUrl.APPend("location=" + latitude + "," + longitude);  return (googlePlacesUrl.toString());  }  }  return "";  }  });  Button btnPetHospital = findViewById(R.id.PetHospital);  btnPetHospital.setOnClickListener(new View.OnClickListener() {  String PetHospital = "PetHospital";  @Override  public void onClick(View v) {  mMap.clear();  String url = getUrl(PetHospital);  Object[] DataTransfer = new Object[2];  DataTransfer[0] = mMap;  DataTransfer[1] = url;  GetNearbyPlacesData getNearbyPlacesData = new GetNearbyPlacesData();  getNearbyPlacesData.execute(DataTransfer);  }  private String getUrl(String nearbyPlace) {  if (ActivityCompat.checkSelfPermission(  MapsActivity.this,Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(  MapsActivity.this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  ActivityCompat.requestPermissions(MapsActivity.this, new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION}, LOCATION\_PERMISSION\_REQUEST\_CODE);  } else {  LocationManager mgr = (LocationManager) getSystemService(Context.LOCATION\_SERVICE);  Location locationGPS = mgr.getLastKnownLocation(LocationManager.GPS\_PROVIDER);  if (locationGPS != null) {  latitude = locationGPS.getLatitude();  longitude = locationGPS.getLongitude();  Toast.makeText(MapsActivity.this, "latitude:" + latitude + " longitude:" + longitude, Toast.LENGTH\_SHORT).show();  StringBuilder googlePlacesUrl = new StringBuilder("https://maps.googleapis.com/maps/api/place/nearbysearch/json?key=AIzaSyBiU7Qs7b5GjLBZ8kxHrJU-2VOmRXR6XpY&radius=1000&type=veterinary\_care&sensor=true&");  googlePlacesUrl.APPend("location=" + latitude + "," + longitude);  return (googlePlacesUrl.toString());  }  }  return "";  }  });  Button btnPark = findViewById(R.id.Park);  btnPark.setOnClickListener(new View.OnClickListener() {  String Park = "Park";  @Override  public void onClick(View v) {  mMap.clear();  String url = getUrl(Park);  Object[] DataTransfer = new Object[2];  DataTransfer[0] = mMap;  DataTransfer[1] = url;  GetNearbyPlacesData getNearbyPlacesData = new GetNearbyPlacesData();  getNearbyPlacesData.execute(DataTransfer);  }  private String getUrl(String nearbyPlace) {  if (ActivityCompat.checkSelfPermission(  MapsActivity.this,Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(  MapsActivity.this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  ActivityCompat.requestPermissions(MapsActivity.this, new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION}, LOCATION\_PERMISSION\_REQUEST\_CODE);  } else {  LocationManager mgr = (LocationManager) getSystemService(Context.LOCATION\_SERVICE);  Location locationGPS = mgr.getLastKnownLocation(LocationManager.GPS\_PROVIDER);  if (locationGPS != null) {  latitude = locationGPS.getLatitude();  longitude = locationGPS.getLongitude();  //Toast.makeText(MapsActivity.this, "latitude:" + latitude + " longitude:" + longitude, Toast.LENGTH\_SHORT).show();  StringBuilder googlePlacesUrl = new StringBuilder("https://maps.googleapis.com/maps/api/place/nearbysearch/json?key=AIzaSyBiU7Qs7b5GjLBZ8kxHrJU-2VOmRXR6XpY&radius=1000&type=park&sensor=true&");  googlePlacesUrl.APPend("location=" + latitude + "," + longitude);  return (googlePlacesUrl.toString());  }  }  return "";  }  });  } | | | |

### 表9-2-73 數據解析

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-18-1 | 函式名稱 | doInBackground |
| 功能 | 數據解析 | | |
| @Override  protected String doInBackground(Object... params) {  mMap = (GoogleMap) params[0];  url = (String) params[1];  DownloadUrl downloadUrl = new DownloadUrl();  googlePlacesData = downloadUrl.readUrl(url);  return googlePlacesData;  } | | | |

### 表9-2-74 顯示數據列表的結果

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-18-2 | 函式名稱 | onPostExecute |
| 功能 | 顯示數據列表的結果 | | |
| @Override  protected void onPostExecute(String result) {  List<HashMap<String, String>> nearbyPlacesList = null;  DataParser dataParser = new DataParser();  nearbyPlacesList = dataParser.parse(result);  ShowNearbyPlaces(nearbyPlacesList);  } | | | |

### 表9-2-75 顯示附近位置URL

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-18-3 | 函式名稱 | ShowNearbyPlaces |
| 功能 | 顯示附近位置URL | | |
| private void ShowNearbyPlaces(List<HashMap<String, String>> nearbyPlacesList) {  for (int i = 0; i < nearbyPlacesList.size(); i++) {  MarkerOptions markerOptions = new MarkerOptions();  HashMap<String, String> googlePlace = nearbyPlacesList.get(i);  double lat = Double.parseDouble(googlePlace.get("lat"));  double lng = Double.parseDouble(googlePlace.get("lng"));  String placeName = googlePlace.get("place\_name");  String vicinity = googlePlace.get("vicinity");  LatLng latLng = new LatLng(lat, lng);  markerOptions.position(latLng);  markerOptions.title(placeName + " : " + vicinity);  mMap.addMarker(markerOptions);  markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_RED));  mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));  mMap.animateCamera(CameraUpdateFactory.zoomTo(14));  }  } | | | |

### 表9-2-76 下載附近位置URL

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-19-1 | 函式名稱 | readUrl |
| 功能 | 下載附近位置URL | | |
| public String readUrl(String strUrl) throws IOException {  String data = "";  InputStream iStream = null;  HttpURLConnection urlConnection = null;  try {  URL url = new URL(strUrl);  urlConnection = (HttpURLConnection) url.openConnection();  urlConnection.connect();  iStream = urlConnection.getInputStream();  BufferedReader br = new BufferedReader(new InputStreamReader(iStream));  StringBuffer sb = new StringBuffer();  String line = "";  while ((line = br.readLine()) != null) {  sb.APPend(line);  }  data = sb.toString();  br.close();  } catch (Exception e) {  } finally {  iStream.close();  urlConnection.disconnect();  }  return data;  } | | | |

### 表9-2-77 解析數據存於列表中

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-20-1 | 函式名稱 | parse |
| 功能 | 解析數據存於列表中 | | |
| public List<HashMap<String, String>> parse(String jsonData) {  JSONArray jsonArray = null;  JSONObject jsonObject;  try {  jsonObject = new JSONObject((String) jsonData);  jsonArray = jsonObject.getJSONArray("results");  } catch (JSONException e) {  }  return getPlaces(jsonArray);  } | | | |

### 表9-2-78 解析附近地點詳細位置

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-20-2 | 函式名稱 | getPlaces |
| 功能 | 解析附近地點詳細位置 | | |
| private List<HashMap<String, String>> getPlaces(JSONArray jsonArray) {  int placesCount = jsonArray.length();  List<HashMap<String, String>> placesList = new ArrayList<>();  HashMap<String, String> placeMap = null;  for (int i = 0; i < placesCount; i++) {  placeMap = getPlace((JSONObject) jsonArray.get(i));  placesList.add(placeMap);  }  return placesList;  } | | | |

### 表9-2-79 解析附近地點詳細資訊

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-20-3 | 函式名稱 | getPlace |
| 功能 | 解析附近地點詳細資訊 | | |
| private HashMap<String, String> getPlace(JSONObject googlePlaceJson) {  HashMap<String, String> googlePlaceMap = new HashMap<String, String>();  String placeName = "-NA-";  String vicinity = "-NA-";  String latitude = "";  String longitude = "";  String reference = "";  if (!googlePlaceJson.isNull("name")) {  placeName = googlePlaceJson.getString("name");  }  if (!googlePlaceJson.isNull("vicinity")) {  vicinity = googlePlaceJson.getString("vicinity");  }  latitude = googlePlaceJson.getJSONObject("geometry").getJSONObject("location").getString("lat");  longitude = googlePlaceJson.getJSONObject("geometry").getJSONObject("location").getString("lng");  reference = googlePlaceJson.getString("reference");  googlePlaceMap.put("place\_name", placeName);  googlePlaceMap.put("vicinity", vicinity);  googlePlaceMap.put("lat", latitude);  googlePlaceMap.put("lng", longitude);  googlePlaceMap.put("reference", reference);  Log.d("getPlace", "Putting Places");  return googlePlaceMap;  } | | | |

### 表9-2-80 函式導向

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-21-1 | 函式名稱 | onCreate |
| 功能 | 函式導向 | | |
| @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  storage = FirebaseStorage.getInstance("gs://ipets-APP.APPspot.com");  storageReference = storage.getReference();  btnChoose = findViewById(R.id.addPhoto);  btnChoose.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent addPhotoIntent = new Intent(Intent.ACTION\_GET\_CONTENT);  addPhotoIntent.putExtra(Intent.EXTRA\_ALLOW\_MULTIPLE,true);  addPhotoIntent.setType("image/\*");  startActivityForResult(Intent.createChooser(addPhotoIntent,"Select Picture"),PICK\_IMAGE\_REQUEST);  }  });  } | | | |

### 表9-2-81 上傳照片至資料庫

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-21-2 | 函式名稱 | uploadImage |
| 功能 | 上傳照片至資料庫 | | |
| private void uploadImage() {  StorageReference ImageFolder = storageReference.child(userUID + '/' + "Gallery/");  for (int upload\_count = 0; upload\_count < ImageList.size(); upload\_count++) {  Uri IndividualImage = ImageList.get(upload\_count);  StorageReference ImageName = ImageFolder.child(IndividualImage.getLastPathSegment());  ImageName.putFile(IndividualImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {  @Override  public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  ImageName.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  @Override  public void onSuccess(Uri uri) {  String url = String.valueOf(uri);  StoreLink(url);  Glide.with(photoGalleryActivity.this).load(uri).into(imageView);  Toast.makeText(photoGalleryActivity.this, "Uploaded", Toast.LENGTH\_SHORT).show();  }  })  }  });  }  if (filepath != null) {  StorageReference ref = storageReference.child(userUID +'/' +"gallery/" + UUID.randomUUID().toString());  ref.putFile(filepath).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {  @Override  public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  progressDialog.dismiss();  Glide.with(photoGalleryActivity.this).load(filepath).into(imageView2);  Toast.makeText(photoGalleryActivity.this, "Uploaded", Toast.LENGTH\_SHORT).show();  }  }).addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {  @Override  public void onProgress(UploadTask.TaskSnapshot taskSnapshot) {  double progress = (100.0 \* taskSnapshot.getBytesTransferred() / taskSnapshot.getTotalByteCount());  progressDialog.setMessage("Uploaded " + (int) progress + "%");  }  });  }  } | | | |

### 表9-2-82 資料庫連結設定

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-21-3 | 函式名稱 | StoreLink |
| 功能 | 資料庫連結設定 | | |
| private void StoreLink (String url){  DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReference().child("UserOne");  HashMap<String, String> hashMap = new HashMap<>();  hashMap.put("Imglink", url);  databaseReference.push().setValue(hashMap);  } | | | |

### 表9-2-83 讀取照片

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-21-4 | 函式名稱 | onActivityResult |
| 功能 | 讀取照片 | | |
| @Override  protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  super.onActivityResult(requestCode, resultCode, data);  if (requestCode == PICK\_IMAGE\_REQUEST && resultCode == RESULT\_OK) {  if (data.getData() != null) {  Bitmap bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), data.getData());  imageView2.setImageBitmap(bitmap);  } else if (data.getClipData() != null) {  while (0 < data.getClipData().getItemCount()) {  ImagesUri = data.getClipData().getItemAt(currentImageSelect).getUri();  ImageList.add(ImagesUri);  imageView.setImageURI(ImagesUri);  currentImageSelect = currentImageSelect + 1;  }  }  }  } | | | |

# 第十章 測試模型

## 10-1 測試計畫

本小組將測試計畫分為網頁及APP兩部分，以下為主要功能測試：

1. **網頁測試計畫**
2. 註冊：於頁面右上方點選「註冊」按鈕並輸入註冊頁面上要求之資料。
3. 登入：至登入頁面輸入帳號、密碼後，點選「登入」按鈕。
4. 會員：註冊後可執行的功能。

3-1. 行事曆：點選「行事曆」按鈕。

3-1-1. 新增事件：至行事曆左上方點選「新增事件」按鈕，或是直接點選要新增事件的「日期」。

3-1-2. 編輯事件：點選欲編輯的事件後，點選「編輯」按鈕，輸入欲修改之事件資料後，點選「儲存」按鈕。

3-1-3. 刪除事件：點選欲刪除的事件後，點選事件左上方「垃圾桶」圖示，確認刪除該事件後，點選頁面上方提示欄「確認」按鈕。

3-1-4. 查看事件：至行事曆頁面點選「事件」按鈕。

3-2. 相簿：點選「相簿」按鈕。

3-2-1. 新增相簿：至相簿頁面右上方點選「新增相簿」按鈕，輸入相簿名稱並選取一張圖片後，點選「確認新增」按鈕。

3-2-2. 查看相簿內相片：點選欲查看之「相簿圖案」。

3-2-3. 刪除相片：點選欲刪除相片後，點選左上方「垃圾桶」圖示。

3-2-4. 下載相片：點選欲下載之相片，點選左上方「下載」圖示。

3-3. 寵物品種辨識：至寵物品種辨識頁面選取欲辨識圖片後，點選「確認」按鈕。

3-4. 風格轉換：至風格轉換頁面選取欲風格轉換之圖片後，點選「確認」按鈕。

3-5. 設定：點選「設定」按鈕。

3-5-1. 會員資料修改：至編輯個人資訊頁面，輸入欲修改之資料後，點選「送出」按鈕。

3-5-2. 新增毛小孩：至新增毛小孩頁面，輸入欲新增之資料後，點選「送出」按鈕。

3-5-3. 更改密碼：至更改密碼頁面，輸入舊密碼與新密碼，確認無誤後點選「送出」按鈕。

3-5-4. 常見問題：至常見問題頁面，點選欲得知的常見問題。

3-5-5. 聯絡我們：至聯絡我們頁面，輸入基本資料及遇到的問題後，點選「送出」按鈕。

1. 管理者：至管理者頁面輸入密碼後，點選「送出」按鈕。

4-1. 文章管理：點選「管理文章」按鈕。

4-1-1. 新增文章：點選長條「+」圖示，輸入欲新增之資料後，點選「確認新增文章」按鈕。

4-1-2. 刪除文章：點選欲刪除之文章後方「X」圖示後，點選「確認」按鈕。

4-1-3. 編輯文章：點選欲編輯之文章後方「✎」圖示，輸入欲修改之資料後，點選「確認」按鈕。

4-1-4. 查看文章：點選欲查看之文章後方「眼睛」圖示。

4-2. 領養機構管理：點選「管理領養機構」按鈕。

4-2-1. 新增領養機構：點選長條「+」圖示，輸入欲新增之資料後點選「確認新增領養機」按鈕。

4-2-2. 刪除領養機構：點選欲刪除之領養機構後方「X」圖示後，點選「確認」按鈕。

4-2-3. 編輯領養機構：點選欲編輯之領養機構後方「✎」圖示，輸入欲修改之資料後點選「確認」按鈕。

4-2-4. 查看領養機構：點選欲查看之領養機構後方「眼睛」圖示。

4-3. 使用者回饋管理：點選「管理使用者回饋」按鈕。

1. 關於我們：點選「關於我們」按鈕。
2. 狗狗科普：點選「狗狗科普」按鈕後，點選欲查看之文章下方「了解更多」按鈕。
3. 人與狗的關係：點選「人與狗的關係」按鈕。
4. 聯繫我們：點選「聯繫我們」按鈕。

8-1. 聯絡我們：輸入基本資料及遇到的問題後，點選「送出」按鈕。

8-2. 加入我們：至頁面下方點選「加入我們」按鈕。

1. **APP測試計畫**
2. 註冊：至登入頁面點選「註冊」按鈕並輸入註冊頁面上要求之資料。
3. 登入：至登入頁面輸入帳號、密碼後點選「登入」按鈕。
4. 首頁：點選「首頁」圖示。

3-1. 寵物切換/新增：點選寵物名稱的下拉式選單，點選要切換的寵物/點選新增寵物。

3-2. 編輯寵物資訊：切換至欲編輯之寵物後，點選「編輯寵物」按鈕，在編輯寵物資訊介面輸入要更改的部份後，點選「✓」按鈕。

3-3. 倒數計時器：點選欲倒數計時之項目圖示，選擇排程的日期。

1. 行事曆：點選「行事曆」圖示。

4-1. 新增事件：至行事曆右上方點選「+」按鈕。

4-2. 刪除事件：點選欲刪除事件的「日期」後，點選欲刪除的事件，再點選「刪除」。

4-3. 編輯事件：點選欲編輯事件的「日期」後，點選欲編輯的事件，再點選「編輯」，輸入欲修改之事件資料後點選「儲存」按鈕。

4-4. 查看事件：至行事曆頁面點選欲查看行程之日期。

1. 相簿：點選「相簿」圖示，點選右下方「+」按鈕，選擇要上傳的照片後點選「選取」按鈕，點選「上傳圖片」按鈕。
2. 定位：點選「定位」圖示。

6-1. 查詢地點：於上方搜尋框輸入要搜尋的地點，點選「搜尋」按鈕。

6-2. 導航：上方搜尋框輸入要搜尋的地點後，點選「導航」圖示。

6-3. 尋找附近寵物用品店：點選「寵物用品」按鈕。

6-4. 尋找附近寵物醫院：點選「寵物醫院」按鈕。

6-5. 尋找附近公園：點選「公園」按鈕。

6-6. 尋找附近寵物美容店：點選「寵物美容」按鈕。

6-7. 返回目前位置：點選「GPS」圖示。

1. 設定：點選「設定」圖示。

7-1. 編輯主人基本資料：至設定頁面點選「主人基本資料」按鈕，在編輯主人基本資料介面輸入要更改的部份後，點選「✓」按鈕。

7-1-1. 變更帳號：至編輯主人基本資料介面點選「帳號（Email）」按鈕，在變更帳號（Email）介面輸入要更改的帳號，點選「✓」按鈕。

7-1-2. 變更密碼：至編輯主人基本資料介面點選「密碼」按鈕，在變更密碼介面輸入新、舊密碼，點選「✓」按鈕。

7-2. 常見問題：至設定頁面點選「常見問題」按鈕，在常見問題介面點選您遇到的問題，便可獲得相關解答。

7-3. 聯絡我們：至設定頁面點選「聯絡我們」按鈕，在聯絡我們介面輸入要求的帳戶資料、問題及描述，點選「確認送出」按鈕。

7-4. 登出：點選「登出」按鈕。

## 10-2 測試個案與測試結果資料

1. **網頁個案與測試結果**

### 表10-2-1 註冊測試結果

|  |  |
| --- | --- |
| 功能名稱 | 1. 註冊 |
| 測試流程 | 1. 於頁面右上方點選「註冊」按鈕。 2. 輸入註冊頁面上要求之資料。 3. 點選「註冊」按鈕，自動跳轉至登入首頁。 |
| 預期成果 | 使用者註冊完後獲得專屬帳號，同時資料庫新增註冊資料。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

### 表10-2-2 登入測試結果

|  |  |  |
| --- | --- | --- |
| 功能名稱 | 1. 登入 | |
| 測試流程 | 1. 於登入頁面輸入已註冊之帳號、密碼。 2. 點選「登入」按鈕。 | |
| 預期成果 | 帳號、密碼輸入正確 | 帳號、密碼輸入錯誤 |
| 自動跳轉至登入後首頁。 | 1. 帳號錯誤顯示「user-not-found」。 2. 密碼錯誤則顯示「wrong-password」。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 | |

### 表10-2-3 會員測試結果

|  |  |  |
| --- | --- | --- |
| 1. 會員 | | |
| 3-1. 行事曆 | | |
| 功能名稱 | 3-1-1. 新增事件 | |
| 測試流程 | 1. 使用者登入。 2. 至行事曆頁面。 3. 至行事曆左上方點選「新增事件」按鈕。 4. 輸入欲新增之事件內容後點選「建立事件」。 | 1. 使用者登入。 2. 至行事曆頁面。 3. 點選欲新增事件之日期。 4. 輸入欲新增之事件內容後點選「建立事件」。 |
| 預期成果 | 建立事件後顯示行程於行事曆頁面上，並新增事件資料於資料庫。 | |
| 執行結果 | 測試成功，執行結果與預期結果一致。 | |
| 功能名稱 | 3-1-2. 編輯事件 | |
| 測試流程 | 1. 使用者登入。 2. 至行事曆頁面。 3. 點選事件開啟行程視窗。 4. 點選下方「編輯」按鈕。 5. 輸入欲修改之行程內容。 6. 點選確認按鈕。 | |
| 預期成果 | 編輯完成事件後顯示行程於行事曆頁面上，並成功更新事件資料於資料庫。 | |
| 執行結果 | 測試成功，執行結果與預期結果一致。 | |
| 功能名稱 | 3-1-3. 刪除事件 | |
| 測試流程 | 1. 使用者登入。 2. 至行事曆頁面。 3. 點選事件開啟行程視窗。 4. 點選左上方「垃圾桶」圖示。 5. 點選確認按鈕。 | |
| 預期成果 | 刪除完成事件後在頁面移除該行程，並更新事件於資料庫。 | |
| 執行結果 | 測試成功，執行結果與預期結果一致。 | |
| 功能名稱 | 3-1-4. 查看事件 | |
| 測試流程 | 1. 使用者登入。 2. 至行事曆頁面。 3. 點選事件開啟行程視窗。 | |
| 預期成果 | 點選後跳出行程視窗。 | |
| 執行結果 | 測試成功，執行結果與預期結果一致。 | |

|  |  |
| --- | --- |
| 3-2. 相簿 | |
| 功能名稱 | 3-2-1. 新增相簿 |
| 測試流程 | 1. 使用者登入。 2. 至相簿頁面。 3. 至行事曆右上方點選「新增相簿」按鈕。 4. 輸入新增相簿名稱並選取圖片。 5. 點選「確認新增」按鈕。 |
| 預期成果 | 建立相簿後顯示於相簿頁面上，並新增相簿資料於資料庫。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 3-2-2. 查看相簿內相片 |
| 測試流程 | 1. 使用者登入。 2. 至相簿頁面。 3. 點選欲查看之相簿。 |
| 預期成果 | 跳轉頁面至相片顯示畫面，並顯示該相簿內所有相片。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 3-2-3. 刪除相片 |
| 測試流程 | 1. 使用者登入。 2. 至相簿頁面。 3. 點選欲查看之相簿。 4. 跳轉至相片顯示頁面。 5. 點選欲刪除之相片。 6. 點選相片左上角「垃圾桶」圖示。 |
| 預期成果 | 刪除相片並更新資料庫相簿。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 3-2-4. 下載相片 |
| 測試流程 | 1. 使用者登入。 2. 至相簿頁面。 3. 點選欲查看之相簿。 4. 跳轉至相片顯示頁面。 5. 點選欲下載之相片。 6. 點選相片左上角「下載」圖示。 |
| 預期成果 | 下載圖片至本地端。 |
| 執行結果 | 執行結果與預期結果不符，沒有下載成功但可以跳出圖片。 |
| 功能名稱 | 3-3. 寵物品種辨識 |
| 測試流程 | 1. 使用者登入。 2. 至寵物品種辨識頁面。 3. 選取上傳相片。 4. 點選「確認」按鈕。 |
| 預期成果 | 上傳圖片至後端伺服器後於網頁顯示辨識結果。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 3-4. 風格轉換 |
| 測試流程 | 1. 使用者登入。 2. 至寵物品種辨識頁面。 3. 選取上傳相片。 4. 點選「確認」按鈕。 |
| 預期成果 | 上傳圖片至後端伺服器後於網頁顯示圖片風格轉換結果。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 3-5. 設定 | |
| 功能名稱 | 3-5-1. 會員資料修改 |
| 測試流程 | 1. 使用者登入。 2. 至設定頁面。 3. 點選「編輯個人資訊」選項。 4. 輸入欲修改之資料。 5. 點選「送出」按鈕。 |
| 預期成果 | 顯示更新後資料並更新資料於資料庫。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 3-5-2. 新增毛小孩 |
| 測試流程 | 1. 使用者登入。 2. 至設定頁面。 3. 點選「新增毛小孩」選項。 4. 輸入欲新增之資料及相片。 5. 點選「送出」按鈕。 |
| 預期成果 | 新增資料於資料庫。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 3-5-3. 更改密碼 |
| 測試流程 | 1. 使用者登入。 2. 至設定頁面。 3. 點選「更改密碼」選項。 4. 輸入目前密碼及欲更新之密碼及確認密碼。 5. 點選「送出」按鈕。 |
| 預期成果 | 成功更新密碼於資料庫並將其他登入裝置強制登出。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 3-5-4. 常見問題 |
| 測試流程 | 1. 使用者登入。 2. 至設定頁面。 3. 點選「常見問題」選項。 |
| 預期成果 | 顯示所有常見問題及點選後便會出現下拉視窗，即可查看解答。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

|  |  |
| --- | --- |
| 功能名稱 | 3-5-5. 聯絡我們 |
| 測試流程 | 1. 使用者登入。 2. 至設定頁面。 3. 點選「聯絡我們」選項。 4. 輸入基本資料及遇到的問題。 5. 點選「送出」按鈕。 |
| 預期成果 | 送出意見回饋並新增資料於資料庫。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

### 表10-2-4 管理者測試結果

|  |  |
| --- | --- |
| 1. 管理者 | |
| 4-1. 文章管理 | |
| 功能名稱 | 4-1-1. 新增文章 |
| 測試流程 | 1. 管理者登入。 2. 至管理文章頁面。 3. 點選上方長條「+」圖示。 4. 輸入欲新增之文章內容及圖片後點選「確認新增文章」按鈕。 |
| 預期成果 | 1. 新增文章於下方文章列表。 2. 新增資料於資料庫。 3. 顯示新增之文章於狗狗科普頁面。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 4-1-2. 刪除文章 |
| 測試流程 | 1. 管理者登入。 2. 至管理文章頁面。 3. 點選欲刪除之文章後方「X」圖示。 4. 確認選取無誤後點選「確認」按鈕。 |
| 預期成果 | 1. 移除文章於下方文章列表。 2. 刪除資料於資料庫。 3. 刪除文章於狗狗科普頁面。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 4-1-3. 編輯文章 |
| 測試流程 | 1. 管理者登入。 2. 至管理文章頁面。 3. 點選欲編輯之文章後方「✎」圖示。 4. 輸入欲編輯之文章內容及圖片後點選「確認儲存文章」按鈕。 |
| 預期成果 | 1. 更新資料於資料庫。 2. 更新文章於狗狗科普頁面。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

|  |  |
| --- | --- |
| 功能名稱 | 4-1-4. 查看文章 |
| 測試流程 | 1. 管理者登入。 2. 至管理文章頁面。 3. 點選欲查看之文章後方「眼睛」圖示。 |
| 預期成果 | 簡單顯示所有文章內容。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 4-2. 領養機構管理 | |
| 功能名稱 | 4-2-1. 新增領養機構 |
| 測試流程 | 1. 管理者登入。 2. 至管理領養機構頁面。 3. 點選上方長條「+」圖示。 4. 輸入欲新增之領養機構內容及圖片後點選「確認新增」按鈕。 |
| 預期成果 | 1. 新增領養機構於下方領養機構列表。 2. 新增資料於資料庫。 3. 顯示新增之領養機構於首頁。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 4-2-2. 刪除領養機構 |
| 測試流程 | 1. 管理者登入。 2. 至管理領養機構頁面。 3. 點選欲刪除之領養機構後方「X」圖示。 4. 確認選取無誤點選「確認」按鈕。 |
| 預期成果 | 1. 移除領養機構於下方領養機構列表。 2. 刪除資料於資料庫。 3. 刪除領養機構於首頁。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 4-2-3. 編輯領養機構 |
| 測試流程 | 1. 管理者登入。 2. 至管理領養機構頁面。 3. 點選欲編輯之領養機構後方「✎」圖示。 4. 輸入欲編輯之領養機構內容及圖片後點選「確認儲存」按鈕。 |
| 預期成果 | 1. 更新資料於資料庫。 2. 更新領養機構內容於首頁。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 4-2-4. 查看領養機構 |
| 測試流程 | 1. 管理者登入。 2. 至管理領養機構頁面。 3. 點選欲查看之領養機構後方「眼睛」圖示。 |
| 預期成果 | 顯示該領養機構之資訊。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 4-3. 使用者回饋管理 |
| 測試流程 | 1. 管理者登入。 2. 至管理使用者回饋頁面。 |
| 預期成果 | 於頁面顯示所有回饋資料。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

### 表10-2-5 關於我們測試結果

|  |  |
| --- | --- |
| 功能名稱 | 1. 關於我們 |
| 測試流程 | 至關於我們頁面。 |
| 預期成果 | 顯示關於我們頁面四項主題。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

### 表10-2-6 狗狗科普測試結果

|  |  |
| --- | --- |
| 功能名稱 | 1. 狗狗科普 |
| 測試流程 | 1. 至狗狗科普頁面。 2. 點選欲查看之文章下方「了解更多」按鈕。 3. 跳轉至該文章頁面顯示詳細內容。 |
| 預期成果 | 顯示所有狗狗科普存於資料庫之文章。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

### 表10-2-7 人與狗的關係測試結果

|  |  |
| --- | --- |
| 功能名稱 | 1. 人與狗的關係 |
| 測試流程 | 至人與狗的關係頁面。 |
| 預期成果 | 顯示人與狗的五大關係及改善方法。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

### 表10-2-8 聯繫我們測試結果

|  |  |  |
| --- | --- | --- |
| 1. 聯繫我們 | | |
| 功能名稱 | 8-1. 聯絡我們 | |
| 測試流程 | 1. 至聯繫我們頁面。 2. 輸入基本資料及遇到的問題。 3. 點選「送出」按鈕。 | |
| 預期成果 | 送出意見回饋並新增資料於資料庫。 | |
| 執行結果 | 測試成功，執行結果與預期結果一致。 | |
| 功能名稱 | 8-2. 加入我們 | |
| 測試流程 | 1. 至聯繫我們頁面下方。 2. 點選「加入我們」按鈕。 3. 轉跳至Line行動條碼畫面。 | 1. 至聯繫我們頁面下方。 2. 點選「加入我們」按鈕。 3. 跳轉至手機Line加入畫面。 |
| 預期成果 | 加入iPets Line好友，並送出歡迎加入訊息。 | |
| 執行結果 | 測試成功，執行結果與預期結果一致。 | |

1. **APP個案與測試結果**

### 表10-2-9 註冊測試結果

|  |  |
| --- | --- |
| 功能名稱 | 1. 註冊 |
| 測試流程 | 1. 於登入頁面點選「註冊」按鈕。 2. 輸入註冊頁面上要求之資料。 3. 點選「下一步」按鈕，自動跳轉至首頁。 |
| 預期成果 | 使用者註冊完後獲得專屬帳號，同時資料庫新增註冊資料。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

### 表10-2-10 登入測試結果

|  |  |  |
| --- | --- | --- |
| 功能名稱 | 1. 登入 | |
| 測試流程 | 1. 於登入頁面輸入已註冊之帳號、密碼。 2. 點選「登入」按鈕，自動跳轉至首頁。 | |
| 預期成果 | 帳號、密碼輸入正確 | 帳號、密碼輸入錯誤 |
| 帳號、密碼輸入正確，自動跳轉至首頁。 | 帳號、密碼輸入錯誤，顯示彈跳視窗「帳號或密碼錯誤，大小寫需相符」，並要求重新輸入。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 | |

### 表10-2-11 首頁測試結果

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 首頁 | | | |
| 功能名稱 | 3-1. 寵物切換/新增 | | |
| 測試流程 | 切換 | | 新增 |
| 1. 點選寵物名稱的下拉式選單。 2. 點選要切換的寵物。 | | 1. 點選寵物品種的下拉式選單。 2. 點選「新增寵物」按鈕。 3. 自動跳轉至新增「寵物基本資料」介面。 4. 輸入欲新增之寵物資料。 5. 點選「完成」按鈕。 |
| 預期成果 | 點選寵物名稱即切換至該寵物資訊介面。 | 1. 新增完成後於首頁顯示該寵物資訊。 2. 新增資料於資料庫。 3. 下拉式選單出現現有寵物讓使用者進行切換。 | |
| 執行結果 | 測試成功，執行結果與預期結果一致。 | | |

|  |  |
| --- | --- |
| 功能名稱 | 3-2. 編輯寵物資訊 |
| 測試流程 | 1. 於首頁點選「編輯寵物資訊」按鈕。 2. 在編輯寵物資訊介面輸入欲更改之資料。 3. 點選「✓」按鈕。 |
| 預期成果 | 1. 顯示更新完成之資料於首頁。 2. 更新資料於資料庫。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 3-3. 倒數計時器 |
| 測試流程 | 1. 點選圖示（如：剃毛、洗澡、經期……等）。 2. 選擇排成的日期。 3. 按確定，開始倒數。 |
| 預期成果 | 設定完排成日期後，開始倒數計時，進度條遞增且呈現藍色，過50%呈現橘色，到了到期日時進度條全滿呈現紅色。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

### 表10-2-12 行事曆測試結果

|  |  |
| --- | --- |
| 1. 行事曆 | |
| 功能名稱 | 4-1. 新增事件 |
| 測試流程 | 1. 點選右下方「+」按鈕。 2. 輸入行程後點選「儲存」按鈕。 |
| 預期成果 | 完成後顯示於行事曆，並新增資料於資料庫。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 4-2. 刪除事件 |
| 測試流程 | 1. 點選欲刪除之事件日期，顯示當日事件列表。 2. 點選欲刪除之事件。 3. 點選下方「刪除」按鈕。 4. 確認無誤後，點選「確定」按鈕。 |
| 預期成果 | 完成後於行事曆頁面移除該事件，並更新資料於資料庫。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 4-3. 編輯事件 |
| 測試流程 | 1. 點選欲編輯之事件日期，顯示當日事件列表。 2. 點選欲編輯之事件。 3. 輸入欲修改事件之資料。 4. 點選「儲存」按鈕。 |
| 預期成果 | 完成後顯示於行事曆，並更新資料於資料庫。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 4-4. 查看事件 |
| 測試流程 | 點選欲查看之事件日期。 |
| 預期成果 | 事件顯示於行事曆。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

### 表10-2-13 相簿測試結果

|  |  |
| --- | --- |
| 功能名稱 | 1. 相簿 |
| 測試流程 | 1. 點選右下方「＋」按鈕。 2. 選擇要上傳的照片後點選「上傳」按鈕。 |
| 預期成果 | 照片上傳，於手機上顯示所有上傳之照片。 |
| 執行結果 | 測試成功，但執行結果不如預期，選取多張照片時，只能顯示單一照片。 |

### 表10-2-14 定位測試結果

|  |  |
| --- | --- |
| 1. 定位 | |
| 功能名稱 | 6-1. 查詢地點 |
| 測試流程 | 1. 輸入要搜尋的地點於上方搜尋框。  2. 點選「搜尋」按鈕。 |
| 預期成果 | 搜尋出所查詢的該地點。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 6-2. 導航 |
| 測試流程 | 1. 上方搜尋框輸入要搜尋的地點。  2. 點選「導航」圖示。 |
| 預期成果 | 搜尋出所查詢的該地點路線，並且開始導航。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 6-3. 尋找附近寵物用品店 |
| 測試流程 | 點選頁面上方「寵物用品」按鈕。 |
| 預期成果 | 搜尋出附近的寵物用品店。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 6-4. 尋找附近寵物醫院 |
| 測試流程 | 點選頁面上方「寵物醫院」按鈕。 |
| 預期成果 | 搜尋出附近的寵物醫院。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 6-5. 尋找附近公園 |
| 測試流程 | 點選頁面上方「公園」按鈕。 |
| 預期成果 | 搜尋出附近的公園。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 6-6. 尋找附近寵物美容店 |
| 測試流程 | 點選頁面上方「寵物美容」按鈕。 |
| 預期成果 | 搜尋出附近的寵物美容店。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | 6-7. 返回目前位置 |
| 測試流程 | 點選右上角「寵物美容」按鈕下方之「GPS」圖示。 |
| 預期成果 | 頁面地圖返回目前位置。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

### 表10-2-15 定位測試結果

|  |  |
| --- | --- |
| 7. 設定 | |
| 功能名稱 | * 1. 編輯主人基本資料 |
| 測試流程 | 1. 於設定頁面點選「主人基本資料」按鈕。 2. 在編輯主人基本資料介面輸入要更改的部份。 3. 點選「✓」按鈕。 |
| 預期成果 | 更改主人的名稱、用戶名稱……等基本資料，供使用者隨時查看，同時資料庫內的資料也會更改。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | * + 1. 變更帳號 |
| 測試流程 | 1. 於編輯主人基本資料介面點選「帳號（Email）」按鈕。 2. 在變更帳號（Email）介面輸入要更改的帳號。 3. 點選「✓」按鈕。 |
| 預期成果 | 更改用戶帳號，同時資料庫內的資料也會更改。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | * + 1. 變更密碼 |
| 測試流程 | 1. 於編輯主人基本資料介面點選「密碼」按鈕。 2. 在變更密碼介面輸入新、舊的密碼。 3. 點選「✓」按鈕。 |
| 預期成果 | 更改用戶密碼，同時資料庫內的資料也會更改。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | * 1. 常見問題 |
| 測試流程 | 1. 於設定頁面點選「常見問題」按鈕。 2. 在常見問題介面點選您遇到的問題。 |
| 預期成果 | 以彈跳視窗方式顯示您問題的解答。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | * 1. 聯絡我們 |
| 測試流程 | 1. 於設定頁面點選「聯絡我們」按鈕。 2. 在聯絡我們介面輸入帳戶資料、問題及描述。 3. 點選「確認送出」按鈕。 |
| 預期成果 | 訊息會發送給管理者，由專人親自回覆解決您的問題，並盡快修改系統之BUG。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |
| 功能名稱 | * 1. 登出 |
| 測試流程 | 1. 於設定頁面點選「登出」按鈕。 2. 登出APP，並自動跳轉至登入畫面。 |
| 預期成果 | 登出並跳轉至登入畫面。 |
| 執行結果 | 測試成功，執行結果與預期結果一致。 |

# 第十一章 操作手冊

### 表11-1-1 網頁操作手冊

|  |
| --- |
| 網頁 |
| 1. 開啟任一瀏覽器 |
|  |
| 2. 輸入網址：https://ipets-2020.herokuapp.com/ |
|  |

### 表11-1-2 APP操作手冊

|  |
| --- |
| APP |
| 1. 輸入載點網址：   https://drive.google.com/drive/folders/1r6Kl5-zVZYVWXaA-pU-pOD2vHN\_rBQoL?usp=sharing |
|  |
| 2. 點擊下載，便可進行安裝 |
|  |

# 第十二章 使用手冊

**一、 網頁使用手冊**

### 表12-1-1 使用手冊之輪播資訊圖

|  |
| --- |
| 輪播資訊圖 |
| 提供最新消息給使用者，讓使用者知道我們最新消息。 |
|  |

### 表12-1-2 使用手冊之領養機構資訊

|  |
| --- |
| 領養機構資訊 |
| 開啟「iPets」首頁，下拉頁面，便可以看到所有的領養機構。 |
|  |

### 表12-1-3 使用手冊之領養資訊

|  |
| --- |
| 領養資訊 |
| 開啟「iPets」首頁，下拉頁面，便可以看到所有的領養機構，點選了解更多，便可以獲得該機構的詳細資訊。 |
|  |

### 表12-1-4 使用手冊之關於我們

|  |
| --- |
| 關於我們 |
| 點選導覽列中的「關於我們」按鈕，便可以看到關於我們的相關資訊。 |
|  |

### 表12-1-5 使用手冊之狗狗科普

|  |
| --- |
| 狗狗科普 |
| 點選導覽列中的「狗狗科普」按鈕，便可以看到所有的狗狗科普文章列表。 |
|  |

### 表12-1-6 使用手冊之狗狗科普文章

|  |
| --- |
| 狗狗科普文章 |
| 點選導覽列中的「狗狗科普」按鈕，便可以看到所有的狗狗科普文章列表，點選「了解更多」按鈕，便可以閱讀狗狗科普文章內容。 |
|  |

### 表12-1-7 使用手冊之人與狗的關係

|  |
| --- |
| 人與狗的關係 |
| 點選導覽列中的「人與狗的關係」按鈕，便可獲得與狗狗相處的基本知識。 |
|  |

### 表12-1-8 使用手冊之登入及註冊

|  |  |
| --- | --- |
| 登入及註冊 | |
| 開啟「iPets」進入登入頁面後，非會員可註冊新帳號，會員可直接登入。 | |
|  |  |

### 表12-1-9 使用手冊之行事曆

|  |
| --- |
| 行事曆 |
| 點選導覽列中的「行事曆」按鈕，便可以瀏覽所有的行程。 |
|  |

### 表12-1-10 使用手冊之新增事件

|  |
| --- |
| 新增事件 |
| 於行事曆頁面，點選左上角「新增事件」，便可新增事件於行事曆。 |
|  |

### 表12-1-11 使用手冊之相簿

|  |
| --- |
| 相簿 |
| 點選導覽列中的「相簿」按鈕，便可以瀏覽所有的相簿。 |
|  |

### 表12-1-12 使用手冊之新增相簿

|  |
| --- |
| 新增相簿 |
| 於相簿頁面，點選右上角「新增相簿」按鈕，便可新增相簿於相簿中。 |
|  |

### 表12-1-13 使用手冊之查看相片

|  |
| --- |
| 查看相片 |
| 點選相簿，便可進入該相簿查看相片。 |
|  |

### 表12-1-14 使用手冊之刪除相片

|  |
| --- |
| 刪除相片 |
| 選取欲刪除的相片，便可點選左上的「垃圾桶」圖示刪除相片。 |
|  |

### 表12-1-15 使用手冊之寵物品種辨識

|  |
| --- |
| 寵物品種辨識 |
| 點選導覽列中的「寵物辨識」按鈕，上傳喜歡的照片，系統便會開始圖片辨識，並且將結果顯示於頁面。 |
|  |

### 表12-1-16 使用手冊之圖片風格轉換

|  |
| --- |
| 圖片風格轉換 |
| 點選導覽列中的「風格轉換」按鈕，上傳兩張喜歡的照片，系統便會開始進行風格轉換，並將結果顯示於頁面。 |
|  |

### 表12-1-17 使用手冊之編輯個人資訊

|  |
| --- |
| 編輯個人資訊 |
| 於「設定」頁面點選左方「編輯個人資訊」選項，輸入欲更改之資料，點選「送出」按鈕，成功更新資料於資料庫並顯示於頁面上。 |
|  |

### 表12-1-18 使用手冊之新增毛小孩

|  |
| --- |
| 新增毛小孩 |
| 於「設定」頁面點選左方「新增毛小孩」選項，輸入欲新增毛小孩之資料，點選「送出」按鈕，成功新增資料於資料庫。 |
|  |

### 表12-1-19 使用手冊之更改密碼

|  |
| --- |
| 更改密碼 |
| 於「設定」頁面點選左方「更改密碼」選項，輸入目前密碼及欲更新之密碼及確認密碼，點選「送出」按鈕，成功更新密碼於資料庫。 |
|  |

### 表12-1-20 使用手冊之常見問題

|  |
| --- |
| 常見問題 |
| 於「設定」頁面點選左方「常見問題」選項，點選欲得知問題之選項，將顯示解答於問題下方。 |
|  |

### 表12-1-21 使用手冊之聯絡我們

|  |
| --- |
| 聯絡我們 |
| 於「設定」頁面點選左方「聯絡我們」選項，輸入基本資料及遇到的問題，點選「送出」按鈕，成功新增意見回饋於資料庫。 |
|  |

### 表12-1-22 使用手冊之新增文章

|  |  |
| --- | --- |
| 新增文章 | |
| 步驟1. 於「管理者」頁面點選上方「管理文章」選項，點選「+」圖示。 | 步驟2. 輸入欲新增文章之內容後，點選「確認新增文章」按鈕。 |
|  |  |

### 表12-1-23 使用手冊之刪除文章

|  |  |
| --- | --- |
| 刪除文章 | |
| 步驟1. 點選欲刪除之文章後方「X」圖示。 | 步驟2. 確認無誤後，點選「確定」按鈕。 |
|  |  |

### 表12-1-24 使用手冊之編輯文章

|  |  |
| --- | --- |
| 編輯文章 | |
| 步驟1. 點選欲編輯之文章後方「✎」圖示。 | 步驟2. 輸入欲更改文章之內容後，點選「確認儲存文章」按鈕。 |
|  |  |

### 表12-1-25 使用手冊之新增領養機構

|  |  |
| --- | --- |
| 新增領養機構 | |
| 步驟1. 於「管理者」頁面點選上方「管理領養機構」選項，點選「+」圖示。 | 步驟2. 輸入欲新增領養機構資訊後，點選「確認新增」按鈕。 |
|  |  |

### 表12-1-26 使用手冊之刪除領養機構

|  |  |
| --- | --- |
| 刪除領養機構 | |
| 步驟1. 點選欲刪除之領養機構後方「X」圖示。 | 步驟2. 確認無誤後，點選「確定」按鈕。 |
|  |  |

### 表12-1-27 使用手冊之編輯領養機構

|  |  |
| --- | --- |
| 編輯領養機構 | |
| 步驟1. 點選欲編輯之領養機構後方「✎」圖示。 | 步驟2. 輸入欲更改領養機構之內容後，點選「確認儲存」按鈕。 |
|  |  |

### 表12-1-28 使用手冊之管理使用者回饋

|  |
| --- |
| 管理使用者回饋 |
| 於「管理者」頁面點選上方「管理使用者回饋」選項，即可查看所有使用者回饋。 |
|  |

**二、 APP使用手冊**

### 表12-2-1 使用手冊之登入及註冊

|  |  |
| --- | --- |
| 登入及註冊 | |
| 開啟「iPets」進入登入頁面後，非會員可註冊新帳號，會員可直接登入。 | |
|  |  |

### 表12-2-2 使用手冊之新增寵物

|  |  |
| --- | --- |
| 新增寵物 | |
| 步驟1. 點選寵物名稱。 | 步驟2. 點選新增寵物。 |
|  |  |
| 步驟3. 輸入後點選「完成」按鈕便完成新增。 |  |
|  |

### 表12-2-3 使用手冊之編輯寵物

|  |  |
| --- | --- |
| 編輯寵物 | |
| 步驟1. 點選「編輯寵物」按鈕。 | 步驟2. 編輯完後，點選「✓」圖示便可修改寵物資訊。 |
|  |  |

### 表12-2-4 使用手冊之倒數計時器

|  |  |
| --- | --- |
| 倒數計時器 | |
| 步驟1. 點選欲設定倒數計時之圖示。 | 步驟2. 選擇排程的日期，點選「確定」按鈕，便可開始倒數計時。 |
|  |  |

### 表12-2-5 使用手冊之行事曆

|  |  |
| --- | --- |
| 行事曆 | |
| 步驟1. 點選導覽列中的「行事曆」圖示。 | 步驟2. 點選日曆日期，便可查看當日行程。 |
|  |  |

### 表12-2-6 使用手冊之新增行程

|  |  |
| --- | --- |
| 新增行程 | |
| 步驟1. 點選「+」按鈕。 | 步驟2. 輸入完，點選「儲存」按鈕，便可新增行程。 |
|  |  |

### 表12-2-7 使用手冊之編輯/刪除行程

|  |  |
| --- | --- |
| 編輯/刪除行程 | |
| 步驟1. 點選欲編輯/刪除之行程。 | 步驟2. 輸入完，點選「儲存」按鈕，便可修改行程/點選「刪除」按鈕，便可刪除行程。 |
|  |  |

### 表12-2-8 使用手冊之相簿

|  |  |
| --- | --- |
| 相簿 | |
| 步驟1. 點選導覽列中的「相簿」圖示。 | 步驟2. 點選右下方「+」圖示，即可開啟手機相簿。 |
|  |  |
| 步驟3. 選取欲上傳相簿之圖片，點選右上方「選取」按鈕。 | 步驟4. 相簿即出現圖片，點選下方「上傳圖片」按鈕，即可成功上傳。 |
|  |  |

### 表12-2-9 使用手冊之導航

|  |  |
| --- | --- |
| 導航 | |
| 步驟1. 點選導覽列中的「定位」圖示。 | 步驟2. 輸入地點後，點選「導航」圖示，便可開啟導航。 |
|  |  |

### 表12-2-10 使用手冊之附近店家/醫院/公園

|  |  |
| --- | --- |
| 附近店家/醫院/公園 | |
| 步驟1. 點選導覽列中的「定位」圖示。 | 步驟2. 點選欲搜尋之附近店家/醫院/公園。 |
|  |  |

# 第十三章 感想

**10646004 李怡蓁**

終於可以放下專題的重擔了，從規劃的那一天開始，每天都被專題追著跑，在這一年學習了相當多的技術，一步一步地把構思的專題變成了實體，那種澎湃是不能用言語表達的，在撰寫程式時常常踢到鐵板，好在每次都有順利的解決，覺得自己對於程式的熱忱也越來越高，有些時候也會想到剛開始時，組員對於程式的不熟悉，到現在可以完整的做出一個成果，內心也是十分感動。

專題所遇到的問題不單只有技術面，團隊合作是最重要的關鍵，對於專題的構思還有分工，難免都會有衝突產生，需要組員們一次又一次的耐心溝通、磨合才可以達到共識並解決問題。還有，很謝謝俊杰老師給我們技術上的指導，以及給我們許多非常棒的建議，給了我們很大的發揮空間，帶領我們一步步前進。也十分感謝我的組員，這一路上走得跌跌撞撞，謝謝組員們這一路的堅持，沒有他們就沒有辦法完成這個作品，辛苦大家了！

**10646043 蔡欣恬**

從專題開始到現在一路走來也快一年了。在專題過程中，我們五個人一起經歷了很多，包括剛開始的主題討論，互相提出有趣的想法跟意見；一開始的技術學習，互相扶持解決問題；寫程式趕進度，一起在半夜討論專題；意見不合，激動地表達自己的想法；在專題報告的前幾天，錄影練習報告的樣子，並真誠的給予問題與建議；專題結束開心的去吃好吃的餐廳…。這些回憶深刻的記在我的腦海裡，雖然過程中有快樂有壓力，但回想起來是覺得甜美，也是一種成長。

專題帶給我的東西不僅僅是跟朋友們相處的這段回憶，還有在程式技術、邏輯思考、文件處理、簡報製作、上台報告等，回過頭檢視，才真正發現自己已經進步許多，還會認為以前的自己怎麼那麼的好笑，寫出一些奇奇怪怪的function，或是文件上明顯的錯誤沒有檢查到。這也是為什麼想起這些回憶時，能夠開心的面帶笑容繼續往未來邁進。

謝謝組員們一直以來的支持與陪伴，相信大家都在專題的過程中有所成長，雖然我們都還有很多要進步的空間，希望未來能夠一起學習並更加精進自己的能力。也謝謝俊杰老師的幫助，您給我們的不只是平常私下的專題指導，更像是個朋友般關心我們給予我們寶貴的意見，我們也會更加努力學習，讓老師為我們感到驕傲。

**N1066405 黃琳恩**

時間過得真快，不知不覺的做了專題將近一年了，而如今走到了尾聲，這一路真的很不簡單，不管是程式的困難和燒腦、繳交文件的煩躁和急迫性、團隊溝通的磨合與包容，我們都克服了，真的是充滿感動和感恩。

謝謝組長和每一位夥伴的支持，沒有他們，也不會有成果的展現。組長雖然嚴厲，但是如果沒有她的嚴厲，我們會是一盤散沙，除了她的嚴厲，她也會給予鼓勵。和夥伴一起，從沒處理專題的經驗，到現在已然學習良多和擁有處理過專題的經驗，她們的陪伴，是我巨大的支持。謝謝俊杰老師，在我在程式遇到困難時，給予我提點，使我能持續的努力去做好對我而言的不可能。

透過專題的實作，讓我了解到自主學習的威力，從不會到會的過程，是趟艱辛但收穫滿滿的經歷，專題讓我挑戰了我覺得我絕對不可能完成的任務，而如今我學習良多。感謝每一次的歷練和辛苦，這培養了我的意志，相信透過此次專題的學習，能使我有更多的學術經驗，且能運用於未來的職業發展上。

**N1066414王偉霖**

在團隊中我負責APP後端跟資料庫架接，APP後端語言我們選擇學習過且較穩定的Java 10，降低一些進入門檻，在寒假期間我們各自學習寫Android，從如何傳接值，到現在可以成功寫出功能執行。過程中遇到很多瓶頸，雖然很令我苦惱，但每當克服了那道坎，就會覺得很有成就感，也開始讓我覺得學習寫程式也不是那麼的恐怖，不過這也要感謝我的組員和俊杰老師，當我思考很久卻沒有結果時，他們會幫忙我看出哪裡錯誤、給予我方向、給予我鼓勵，有時候我也會對自己的粗心懊惱，像有一次查很久查不出來哪裡有錯誤，結果是因為沒注意到程式誤包到其他函式裡，真的對自己的粗心感到很生氣，希望下次自己不要再因為類似的錯誤而煩惱很久。

在團隊合作方面，由於剛開始對Git的不熟悉，導致我每次遇到衝突，或是推錯分支都會很緊張，跑去找組長求助，真的很謝謝她總是有耐心的教導我、幫助我，使我現在比較了解Git的功能與使用方式。

很慶幸學校的專題是必修，很多學校是專題或者實習則一，如果是如此，我一定會選擇實習，因為我覺得寫程式對我來說太難了，但實際操作後發現，並沒有想像中的困難，而是看自己有沒有心要去努力學習，因為專題讓我跨出了不敢嘗試的第一步，讓我覺得以後或許可以朝軟體工程師的方面前進，讓我覺得自己要好好運用讀大學所學到的東西，很感謝學校給了我這次的機會。

**N1066414劉芳汝**

在專題中我主要負責UI/UX 設計與 APP前端撰寫，使用的語言為XML，類似寫網頁前端的HTML，也是用一層一層的tag結構化的定義文件。

我們從去年開始到現在，前段至寒假結束皆為訓練階段，前後端的程式撰寫都一併訓練，自己摸索，在這當中我發現從網路上找練習不太難，只要跟著步驟基本上都會成功。

經過幾次討論結果由我來負責此專題的前端設計撰寫，也是因為在訓練階段發現自己對於視覺上的美很要求，有完美主義，開始撰寫後常常碰到瓶頸，尤其是ImageView困惑我最久，看似好像只要把圖匯進去就能完工的卻不盡然。在撰寫的過程中也了解到要刻一個好看的畫面是要花很多時間調整構想的，要以使用者的角度去思考到底怎樣的設計、配色看起來會是舒服且也方便使用的。

今年暑假過後，我也接觸了後端程式撰寫的部分，嘗試撰寫功能，如我所料的是花費比較多心力，後端真的不容易寫，邏輯要好、對語言要很熟悉，遇到的瓶頸遠大於寫前端所遇到的，花很多時間在查找如何寫、嘗試了各種寫法，然而寫出來的多數是錯誤的，非常挫折，從中了解到自己真的不是這方面的料，還是前端這種要有美感設計的比較適合自己。

專題當中學到最多的是服從與團隊合作，組長要你做什麼就做什麼，不能因為是同輩就不想聽從，就好像出社會後會有上司，而不一定上司就比你年長，可能是平輩，也可能是晚輩，但都得盡力完成每一項被交辦的任務。團隊合作方面感謝組長、組員、俊杰老師在我找不到方法時，提點我如何網路上搜尋關鍵字，找不出錯誤時，幫我看到底哪裡有錯。

# 第十四章 參考資料

**線上參考資料：**

網頁行事曆功能套件。

<https://github.com/jsfanatik/vuestacks-calendar-vue-firebase/blob/master/src/components/Calendar.vue>

網頁色碼表。<https://www.ifreesite.com/color/>

bootstrap 前端元件套件。<https://bootstrap-vue.org/>

網頁開發基礎學習網。<https://www.w3schools.com/>

Timetree APP範本。[https://timetreeAPP.com/](https://timetreeapp.com/)

Cloud Firestore 使用說明。<https://firebase.google.com/docs/firestore/quickstart?hl=zh-cn>

將 Firebase 添加到 Android 說明。<https://firebase.google.com/docs/android/setup?hl=zh-cn>

在 Cloud Firestore 中執行簡單及複合查詢說明。<https://firebase.google.com/docs/firestore/query-data/queries?hl=zh-cn>

Android上的Cloud Storage入門說明。<https://firebase.google.com/docs/storage/android/start?authuser=0>

在Android上使用基於密碼的帳戶對Firebase進行身份驗證說明。<https://firebase.google.com/docs/auth/android/password-auth?authuser=0>

Firebase Authentication : how to get current user's Password?。

<https://terryjryeh.blogspot.com/2019/03/uml-package-diagrams-4.html>

UML套件圖型(Package Diagrams)文件說明。

<https://stackoverflow.com/questions/47910898/firebase-authentication-how-to-get-current-users-password>

Google cloud platform。

<https://console.cloud.google.com/home/dashboard?authuser=2&organizationId=137680207813&project=mymaptest-270708>

Android定位初始化、權限。

<https://www.tutorialspoint.com/how-to-get-current-location-latitude-and-longitude-in-android>

Android定位取得經緯度。

<https://dotblogs.com.tw/ttmac/2015/11/28/114143>

Android Toolbar 學習。

<https://www.youtube.com/watch?v=QOADDUvGvD8&ab_channel=SagarRana>

<https://www.youtube.com/watch?v=3x5tsJnuahw&ab_channel=SagarRana>

Android Calendar學習。

<https://thumbb13555.pixnet.net/blog/post/326337258-calendar-view>

<https://www.youtube.com/watch?v=OwYp5rqNGtE&ab_channel=iKh4everStudio>

Android GridView 相簿學習。

<https://blog.csdn.net/xiaoyuan511/article/details/39317313>

文件附錄1\_一次審查評審意見之修正情形

|  |  |
| --- | --- |
| 評審建議事項 | 修正情形 |
| 1. 「品種辨識」功能，透過什麼樣的技術進行辨識？此外，使用者有何動機需要使用此功能？ | 本小組使用Keras所提供之套件進行寵物品種辨識。  本小組的寵物品種辨識主要是希望可以增加使用者趣味性。 |
| 1. 「寵物風格轉換」功能，透過什麼樣的技術完成？是以random？還是以使用者個人喜好進行學習來完成此功能？ | 透過機器大量學習貓的特徵，也就是訓練深度學習模型。當輸入一張貓的圖片（Source Image）時，模型會萃取其重要特徵，例如紋路、瞳孔顏色，將這些特徵轉移到另一張參考圖片（Reference Image），達成風格轉換的功能。 |
| 1. 雖具實用性，但創意價值要再進一步思考。到底和市場上已有的相關功能APP相比較，本系統的特色？應用價值？ | 本專題系統特色是可以自由地進行行程規劃、定位，並且在之後會於APP新增圖片辨識、風格轉換功能，增加使用者的趣味性，並且也會優化相簿。 |
| 1. 建議引用數據來加強應證該專題的必要性。 | 本小組引用農委會等相關文獻進行佐證專題的必要性。 |
| 1. 建議使用簡單化界面，及自然操作介面。 | 本小組已將介面重新設計，符合簡單化介面、自然操作介面。 |
| 1. 建議增加商業模式。 | 本小組計畫未來結合「寵物店家優惠」、「寵物相關活動宣傳」等商業模式。 |
| 1. 可以試試看和IoT設備整合，串接一些裝置，會比較貼近題目的初衷。 | 本小組計畫未來可以結合智慧寵物餵食機，提高實用性。 |
| 1. 若僅是針對寵物的一些知識，認為很難有明確的市場需求，因為現在真的也蠻多的寵物網站。 | 本小組目標在於整合所有寵物相關資訊，並排除內容農場之問題。 |
| 1. 在文件中撰寫的製作動機，個人覺得有點牽強，應有更強而有力的說法。動機太廣，與多數產品發展相同，要找痛點。 | 已針對製作動機進行修改。 |
| 1. 是否能整合智能家庭？寵物毛孩商機蠻大的。 | 本小組計畫未來結合智能家庭，將針對辨識不同毛小孩，給予不同毛小孩「客製化」的服務。（例如：給飼料的數量、毛小孩狀況異常提醒)，提高實用性。 |

|  |  |
| --- | --- |
| 1. 不斷提到核心的AI技術，但在簡報過程中都沒有特別說明到這一整塊的應用與未來的規劃，可著重發展AI技術。 | 將會在12/04末評報告時說明本小組使用之技術。 |
| 1. 圖像辨識是哪種狗已經能夠做到一定的準確度，不確定這個跟AI的關聯性在於何處。 | 人工智慧發展過程包括學習（大量讀取資訊、並判斷何時使用該資訊）、感知、推理（利用已知資訊做出結論）、自我校正，以及如何操縱或移動物品。圖像辨識屬於機器學習中的「歸類問題」。 |
| 1. 晶片資料整合走失與領養。 | 經過本小組與指導老師討論，認為本專題目前進度及規劃尚不適合加入此功能，因此將此功能加入未來規劃。 |
| 1. 視覺UI/UX可再優化。 | 本小組已將視覺UI/UX重新設計。 |
| 1. APP和後端建議可建整合API。 | 本小組後端、APP皆使用Firebase所提供之API進行撰寫。 |
| 1. 進度落後。 | 本小組已將進度規劃重新調整。 |