**國 立 臺 灣 科 技 大 學**

**資 訊 管 理 系**

**實務專題成果報告**

**學號：B10909022、B10909058、B10909059**

**智能化寵物飼養系統**

**學 生：李家瑜、鄭佳雯、林佳蓁**

**指導教授：羅乃維　教授**

**中華民國 112 年 6 月 8 日**

**目 錄**

第一章　系統開發計畫 3

**第一節　發展系統的動機** 3

**第二節　系統目標** 4

**第三節　系統需求** 4

**1.3.1 　系統簡介** 4

**1.3.2　系統架構** 5

**第四節　系統預期效益** 6

第二章　系統分析與設計 7

**第一節　系統分析** 7

**第二節　資料庫設計** 10

**第三節　系統開發時程** 13

**第四節　其他系統相關設計或考量** 14

第三章 系統實作成果 15

**第一節　系統作業環境** 15

**3.1.1　硬體規格** 15

**3.1.2　軟體規格** 16

**第二節** **系統功能與介面說明** 17

**3.2.1 系統流程說明** 17

**3.2.2 系統功能說明** 17

**3.2.3 系統操作與介面說明** 19

**第三節　其他系統實測或實驗結果** 24

第四章 結論 25

**第一節　系統目標與特色** 25

**第二節　未來展望** 25

第五章 專題心得與檢討 26

第六章 參考文獻 27

# **第一章　系統開發計畫**

## **第一節　發展系統的動機**

現代科技發展迅速，我們日常生活中出現了許多智能產品，如掃地機器人、智慧燈泡和智慧插座等。透過這些智能產品，我們可以將設備、物品和裝置連接在一起，並添加自動控制的功能，從而提高生活的便利性。

現代社會也有越來越多的人選擇不結婚，而是選擇養寵物來陪伴自己度過單調的日常生活。但是，寵物不同於物品，當沒有互動時，不能隨意放置而不損壞。相反地，寵物需要飼主的照顧才能健康存活下去，而不同的寵物需要不同的生活環境。如果環境不適合寵物生活，它們容易生病甚至死亡。然而，現代人的生活節奏忙碌，很難有足夠的時間照顧寵物的日常需求。

因此，我們的專題計畫旨在設計一個能夠通過遠端控制應用程式和按鈕的系統，使寵物可以自己進食，從而實現自動餵食的目的。如果寵物沒有按時進食，飼主也能及時發現，並利用應用程式主動補充寵物的飼料，也可透過即時影像了解寵物的狀況，假如深夜察覺寵物狀況不佳，也可透過應用程式了解有無寵物醫院提供夜間急診以迅速得到醫治。這樣一來，我們可以確保寵物的生存需求得到滿足，並更好地了解寵物的進食時間。這樣飼主就可以放心地工作，同時通過手機保持與寵物的聯繫，不再受地點和空間的限制。

透過我們的系統，我們希望能夠提供一個解決方案，讓現代人能夠更輕鬆地飼養寵物，並確保寵物的福祉。同時，我們還希望提供一個方便的工具，使飼主能夠與寵物保持連結，並隨時了解牠們的狀態。如此一來，飼主就能夠更好地平衡工作和寵物照顧的需求，同時提供寵物所需的關愛和關注。

## **第二節　系統目標**

本系統期望達到以下目標：

1、讓寵物能夠自主控制進食時間

透過設置按鈕，寵物可以按壓按鈕以觸發裝置投餵飼料，同時飼主也能透過應用程式收到通知，了解寵物的進食時間。此外，飼主還可以通過手機遠端餵食寵物，以應對寵物還未熟悉按鈕使用或未準時進食的情況。為了避免寵物意外按壓按鈕或進食過量，本系統設定了適當的間隔時間。

2、提供監控功能

透過攝影機，飼主可以即時觀察寵物的狀況，並提供清楚的影像呈現。攝影機可以擺放在寵物所在的區域，例如餵食器旁或寵物活動區域，以捕捉寵物的行為和狀態。

3、提供寵物醫院資訊。

在寵物出現突發狀況時，飼主可以即時獲得附近寵物醫院的資訊，包括地址、聯絡電話，還有目前位置到醫院的距離、路線，以便飼主能夠迅速就醫。

透過以上功能，本系統旨在提供一個方便的解決方案，讓飼主能夠更好地照顧寵物的生活需求，同時與寵物保持連結和互動，並在需要時提供即時的醫療資訊。

## **第三節　系統需求**

### **1.3.1 　系統簡介**

我們的系統目的是提供一個方便的解決方案，讓忙碌的飼主能夠照顧和關注家中飼養的寵物。透過我們的系統，飼主可以遠端控制寵物的餵食，監控寵物的狀況。同時，系統還提供寵物醫院資訊，讓飼主在寵物出現緊急情況時能夠迅速尋求幫助。

本系統可以訓練寵物利用控制按鈕，自己去控制他想要在什麼時間點去吃飼料，以防寵物剛開始還不懂得操作，飼主也可直接透過手機操作應用程式直接投放飼料；此外，也能透過監控系統讓不在家中的飼主，也能即時看到寵物的狀況，隨時透過通知觀察自己寵物進食的情況，如果突然遇到寵物有突發狀況的話，也能透過應用程式及時找到營業中的寵物醫院的地址、電話，還有目前位置到醫院的距離、路線。

利用我們的系統，飼主可以更輕鬆地照顧寵物的生活需求，同時保持與寵物的聯繫和互動。

### **1.3.2　系統架構**

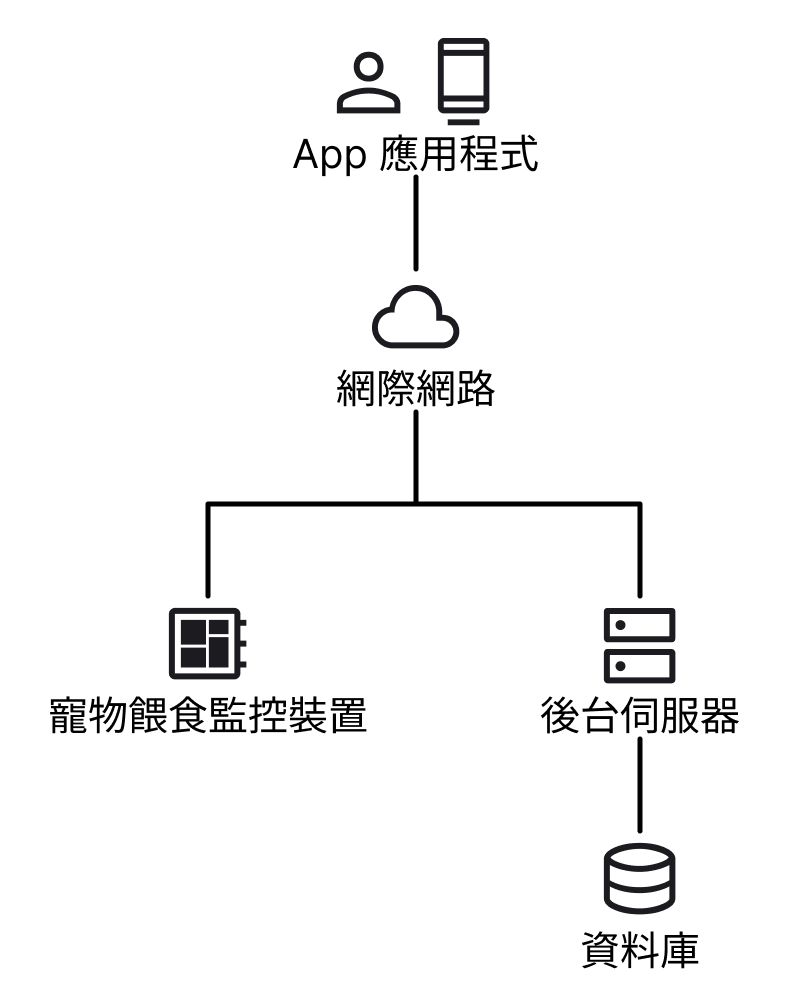


圖1- 1　系統架構圖

1、資料庫：

儲存飼主資料，包括個人資料、寵物資訊、設備資訊等。

2、伺服器：

透過伺服器進行數據資料的交換。

3、App 應用程式：

提供應用程式，給多個飼主使用手機下載，主要用來操控、查詢、捕捉最即時的寵物動態及相關資訊的介面。

4、寵物餵食監控裝置：

設置硬體在飼主家中，能提供監控、餵食寵物等功能，並透過物聯網技術，連接網際網路，進行遠端連線。

## **第四節　系統預期效益**

希望能藉由此專題，實踐寵物智能家電控制系統，讓繁忙的工作時光中，也能抽出些許時間觀察寵物動態，而不是放任寵物獨自在家中，連發生了什麼意外了都不知道，提供飼主更便利和靈活的方式來照顧寵物，減輕日常生活的壓力。

利用此應用程式，讓飼主也能安心上班、同時兼任一個稱職的好飼主，隨時接收寵物的最新動態，即使寵物臨時有突發狀況，也能冷靜下來，利用此應用程式思考該將寵物送去何處醫治傷勢。

# **第二章　系統分析與設計**

## **第一節　系統分析**

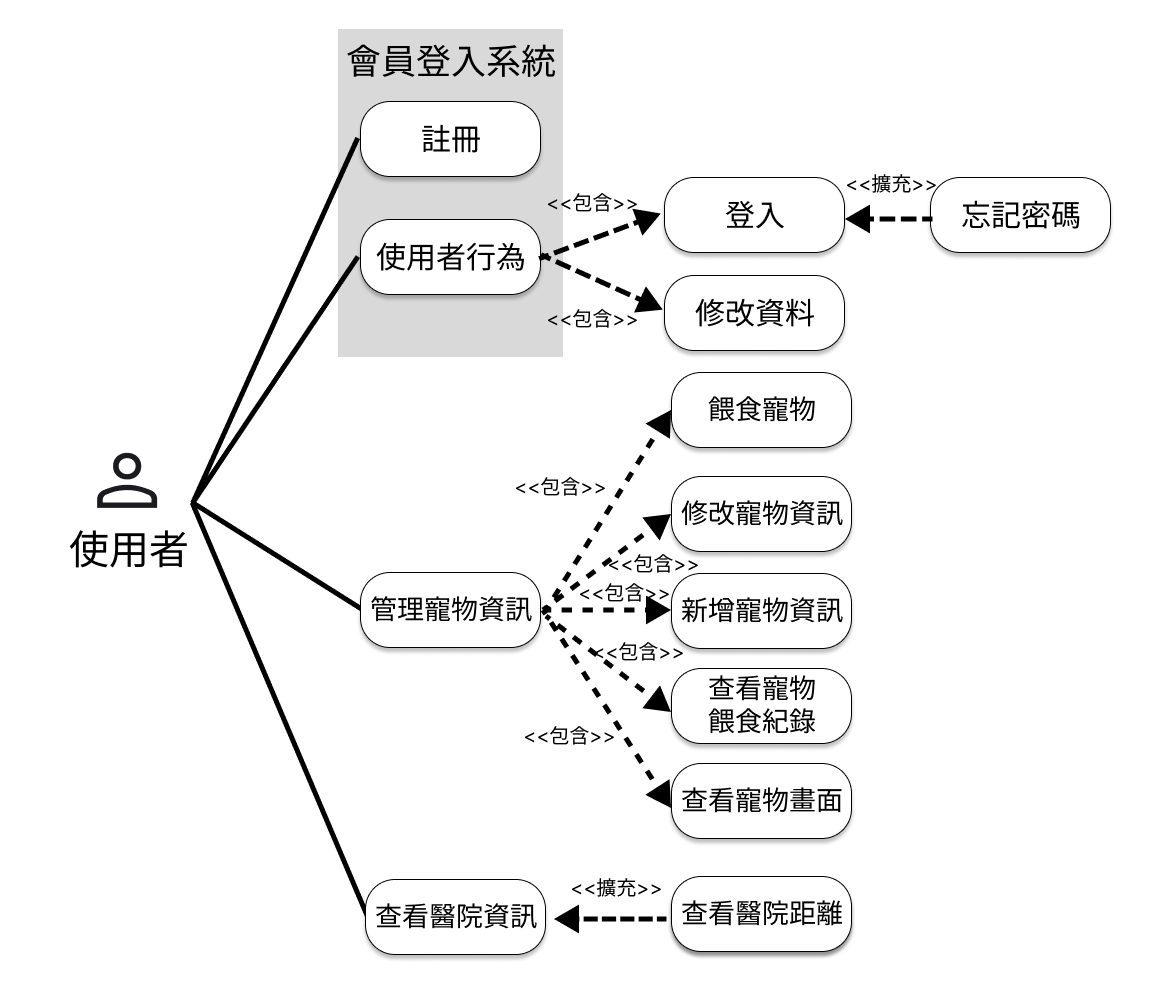


圖2- 1　用例圖（Use Case Diagram）

使用者必須先註冊後再登入做使用。已經註冊過的使用者也就是飼主本人，為其提供的基本功能可以分成三個部分。

第一部分是如果登入前忘記飼主的密碼，系統提供忘記密碼的功能，透過寄送信件來幫助修改新密碼，重新登入以使用此應用程式。

第二個部分是關於寵物的管理，在這部分可以新增多個寵物，畢竟飼主飼養的寵物可能不僅僅只有一個，也可以針對個別寵物資料進行修改，當想要觀察寵物的時候，也可以透過已架設好的攝影鏡頭來觀察寵物即時的動態；由於本系統是讓寵物自主按壓按鈕進行餵食，為了預防剛開始寵物還不懂得如何去操作，這邊有訊息中心能夠查看寵物甚麼時候按下按鈕進食過了，如果寵物還是遲遲沒有按下按鈕，也能讓飼主透過此應用程式手動投放飼料，也能依需求去調整寵物飼料份量、餵食間隔時間。

第三個部分則是提供飼主如果突然有需要查看寵物醫院資訊的時候，可以讓飼主查詢、篩選，選取要查看的醫院後，也可獲取現在所在位置與該醫院的距離及路線。

|  |
| --- |
| 圖2- 2　活動圖（Activity Diagram） |
|  |

圖2-2 表示用戶在使用應用程式的操作流程。

首先，飼主需要選擇註冊或登入會員。在註冊部分，如果已經註冊過或輸入內容錯誤，系統會顯示錯誤訊息；若成功註冊，則可以輸入會員資料並開始使用應用程式的功能。

在登入部分，系統會檢查使用者是否已註冊會員，並驗證密碼是否正確。此外，系統也提供了忘記密碼的功能，透過寄送郵件，讓使用者能夠重新設定密碼。

登入完成後，飼主將進入主選單，其中包含以下選項：瀏覽寵物醫院資訊、寵物資料、監視畫面、餵食通知以及全部功能。

在主頁面，飼主可以查看寵物的相關資訊和餵食設定。飼主可以透過點擊餵食按鈕進行餵食操作。此外，飼主也可以選擇觀看寵物的監控畫面，即時捕捉寵物的現況。同時，系統還提供寵物詳細的餵食紀錄，讓飼主能夠了解寵物的進食情況。

透過瀏覽寵物醫院資訊，飼主可以查看全台各地的寵物醫院資訊。飼主可以選擇特定縣市，以篩選出該地區的寵物醫院，並查看相關詳細資訊，包括醫院地址、聯絡方式以及距離飼主目前位置的距離。

在設定功能頁面，可以讓飼主更改個人基本資料、寵物資料、硬體設置以及通知中心。在寵物資料修改部分，可以選擇新增寵物或修改現有寵物資料；在硬體設置的部分，可以修改餵食份量、餵食間隔時間。

## **第二節　資料庫設計**

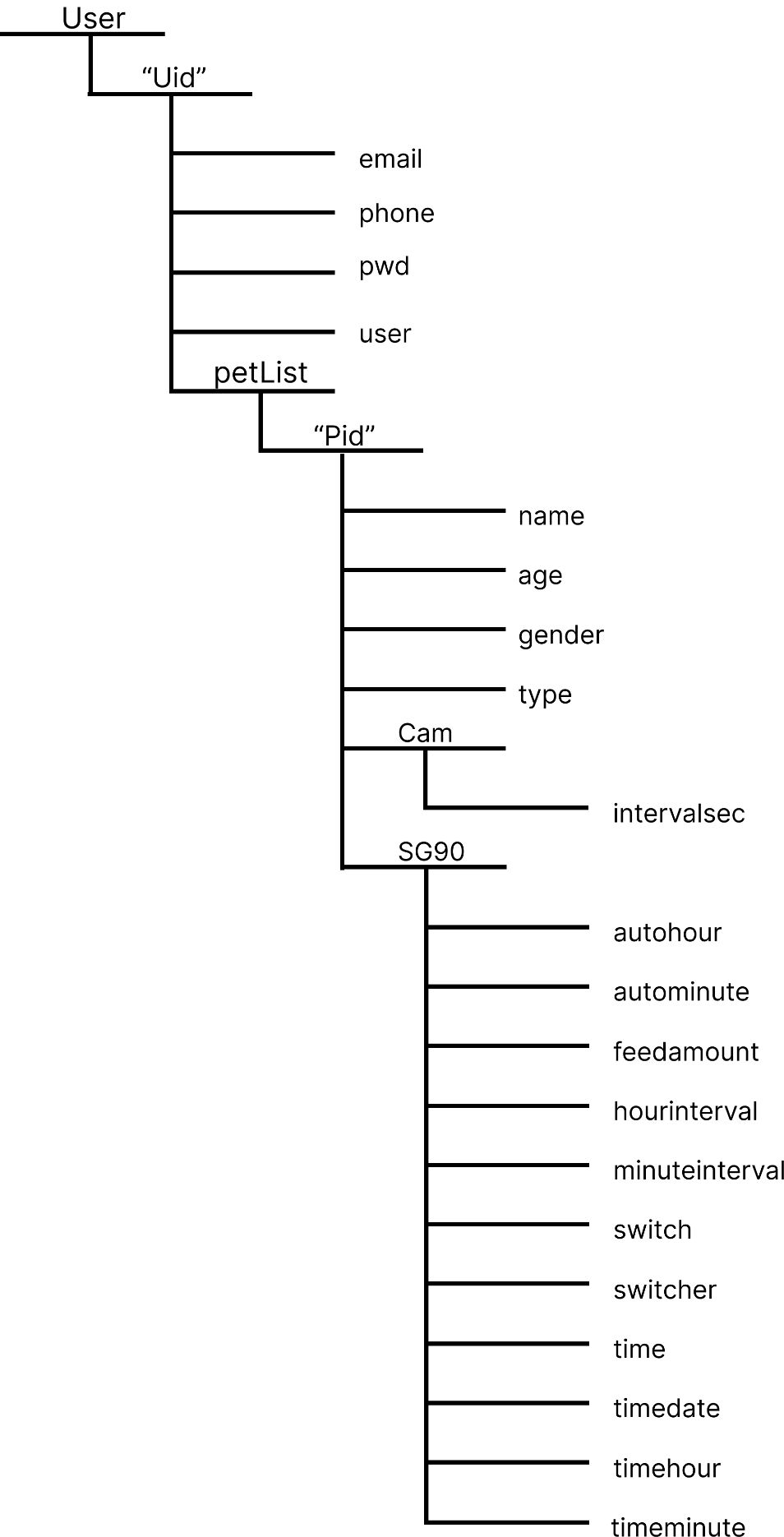


圖2- 3　鍵-值使用者資料庫（Key-value database）

1. 飼主資料 （User）：儲存所有飼主資料
   * 飼主ID （“Uid”）：儲存單一飼主資料
     1. 電子郵件 （email）
     2. 電話（phone）
     3. 密碼（pwd）
     4. 名稱（name）
     5. 寵物列表（petList）：儲存該飼主所有寵物資料
        1. 寵物ID（“Pid”）：儲存該飼主單一寵物資料
           1. 寵物名稱（name）
           2. 寵物年齡（age）
           3. 寵物性別（gender）
           4. 寵物品種（type）
           5. 攝影機（Cam）：儲存攝影機參數

傳送影像間隔秒數（intervalsec）

* + - * 1. 餵食機器（SG90）：儲存餵食機器參數

自動餵食小時（autohour）

自動餵食分鐘（autominute）

餵食份量（feedamount）

寵物取食間隔小時（hourinterval）

寵物取食間隔分鐘（minuteinterval）

餵食開關狀態（switch）

最近餵食方式（switcher）

最近餵食時間的時間戳記（timestamp）

最近餵食時間的日期（timedate）

最近餵食時間的小時（timehour）

最近餵食時間的分鐘（timeminute）

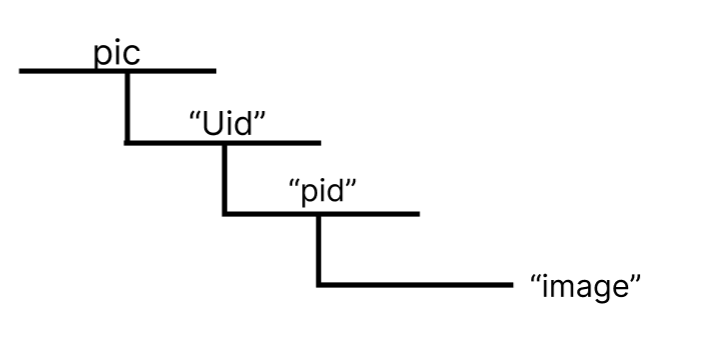


圖2- 4　鍵-值影像資料庫（Key-value database）

1. 影像資料（pic）：儲存所有飼主影像資料
   1. 飼主ID（“Uid”）：儲存單一飼主影像資料
2. 寵物ID（“Pid”）：儲存該飼主單一寵物影像資料
   * + 1. 影像資料(“image”)

## **第三節　系統開發時程**

|  |
| --- |
| 圖2- 5　專題時程甘特圖（Gantt Chart） |
|  |

1. 需求分析和規劃 （111/7/1 - 12/ 31）：
2. 定義系統需求和功能
3. 制定開發計劃和時程安排
4. 收集相關資源和素材
5. 使用者介面設計 （111/12/1 - 12/31）：
6. 設計系統的使用者介面和操作流程
7. 建立草圖和原型
8. 確定使用的設計工具和技術
9. 前端開發 （112/1/1 - 4/15）：
10. 開發首頁、寵物醫院和設定頁面的前端功能
11. 實作使用者登入和註冊功能
12. 整合相關套件和框架
13. 後端開發 （112/3/1 - 5/15）：
14. 建立資料庫和伺服器端環境
15. 實作與前端的資料交換和通訊機制
16. 開發與寵物資料、餵食紀錄等相關的後端功能
17. 硬體整合 （112/4/1 - 5/31）：
18. 連接硬體設備，如觸摸開關按鍵模塊、伺服馬達、開發板和相機
19. 驗證硬體和軟體之間的正確互動
20. 測試和優化硬體控制的相關功能
21. 測試和除錯 （112/2/1 - 5/31）：
22. 進行整體系統測試，確保各項功能正常運作
23. 修復和解決可能出現的錯誤和問題
24. 最終整合和部署 （112/5/1 - 5/31）：
25. 整合所有模塊和功能，確保系統運作良好
26. 準備正式部署所需的環境和資源

## **第四節　其他系統相關設計或考量**

在資料庫設定的部分為了確保資料的保密性和完整性，需要進行適當的資料存取權設定，我們利用以下三點。

1. 使用者身分驗證：確保每個資料庫使用者都有獲取資料的正確權限。這可以通過使用帳號和密碼進行身分驗證。
2. 角色和權限管理：將使用者分配到不同的角色或組，並為每個角色分配適當的資料存取權限。例如，管理員可以擁有完全的讀寫權限，而普通使用者只能擁有部分的讀取權限。
3. 權限控制清單：定義資料庫中各個表格或資源的存取權限清單，並確保只有經過授權的使用者能夠存取或修改相應的資料。

而針對Google API 金鑰設定，在設定金鑰時，確保只授予應用程式所需的最小權限，我們也針對該Android 應用程式才能使用該金鑰；基於安全性考量，我們也會定期更新和輪替 Google API 金鑰，這可以增加系統的安全性，並防止過時的金鑰被濫用。

# **第三章 系統實作成果**

## **第一節　系統作業環境**

### **3.1.1　硬體規格**

1. HTTM系列電容式觸摸感測器
2. 工作電壓：+2.7 ～6V
3. 工作溫度：30 ～+70 ℃
4. OUT 引腳輸出電壓：+3.3V ± 0.1V
5. OUT 引腳最大輸出電流：500mA
6. SG90伺服馬達
7. 尺寸：23mmX12.5mmX29.5mm
8. 重量：9克
9. 無負載速度：0.12秒/60度（4.8V）
10. 堵轉扭矩：1.2-1.4公斤/釐米（4.8V）
11. 使用溫度：-30~~+60攝氏度
12. 死區設定：7微秒
13. 工作電壓：4.8V-6V
14. 舵機有三條線定義：
15. 棕色線： GND
16. 紅色線： VCC 4.8-6V
17. 橙黃線： 脈衝輸
18. 窄版ESP32開發板 （Nodemcu-esp32S）
19. 藍芽 4.0+WIFI 模組
20. USB 採用 ch340
21. ESP32-CAM開發板，含OV2460攝像頭
    1. 低功耗雙核32位CPU
    2. 802.11b/g/n Wi-Fi+BT/BLE SoC 模組
    3. 支持OV2460和OV7670攝像頭
    4. 支持圖片WiFi上傳
    5. 支持TF卡

### **3.1.2　軟體規格**

1. 作業系統：Windows 10、Mac OS 13.3.1
2. 程式語言：Kotlin、ARDUINO草稿碼（類C語言）
3. 開發工具：
4. Android Studio 11.0.15
5. ARDUINO IDE
6. Android Studio AVD Manager
7. 使用套件：
8. 網路請求：
9. OkHttp 4.9.0
10. Volley 1.2.1
11. 數據處理：
12. Gson 2.8.6
13. 資料庫：
14. Firebase Realtime Database 20.0.4
15. Firebase Firestore 24.4.4
16. Firebase Storage 20.1.0
17. 飼主驗證：
18. Firebase Authentication
19. 地圖及位置服務：
20. Google Maps 18.1.0
21. Google Location 21.0.1
22. Android Maps Utils 6.2.1
23. 權限管理：
24. Dexter 6.2.1
25. Google Play服務：
26. Google Play Services 4.3.15

## **第二節 系統功能與介面說明**

### **3.2.1 系統流程說明**

飼主開啟應用程式後，可選擇註冊帳號或直接登入。登入需要填寫帳號密碼，並經過驗證即可登入，假如飼主忘記密碼也可以透過發送電子郵件來重設密碼。而註冊帳號需要填寫基本資料，假如註冊過或填寫不完整會顯示錯誤訊息，註冊成功則會跳返到登入頁面進行登入。

登入系統後，下方導覽列有首頁、寵物醫院以及設定三個頁面。可透過下拉式選單選取想要查看的寵物，首頁會顯示飼主的寵物基本資料、實時影像以及餵食紀錄，同時也能點擊按鈕立即餵食寵物、設定拍攝影像間隔時間。

寵物醫院頁面顯示寵物醫院列表，透過下拉式選單選取飼主所在的城市及區域，查看所選地區的寵物醫院列表，點擊特定醫院後可查看所選的寵物醫院地址、電話、急診資訊，以及地圖資訊，讓飼主了解與目的地之間的距離、路線，也可以跳轉到 Google Map 查看詳細路線。

設定頁面可選擇修改個人資料、修改寵物資料、硬體設置，修改個人資料頁面可修改飼主基本資訊，修改寵物資料頁面可選擇新增寵物或修改現有寵物資料，硬體設置可設定自動餵食的間隔時間以及餵食份量。下方有登出按鈕可讓飼主登出系統。

### **3.2.2 系統功能說明**

1. 使用者註冊與登入：

使用者可以進行帳號註冊，填寫基本資料，並設定帳號密碼。

已註冊的使用者，也就是飼主，可以透過填寫帳號密碼進行登入。如果飼主忘記密碼，系統提供密碼重設功能，飼主可以透過發送電子郵件來重設密碼。

1. 首頁功能：
2. 寵物基本資料：

顯示飼主的寵物基本資料，包括名稱、品種、年齡等，以及硬體設置資料，並且可以點擊按鈕即時餵食寵物。

1. 即時影像：

提供寵物的即時影像，飼主可以查看寵物的情況，也能設定回傳影像時間。

1. 餵食紀錄：

顯示寵物的最近餵食紀錄，包括餵食時間、餵食方式、餵食份量等。

1. 寵物醫院列表：
2. 寵物醫院列表：

顯示所選地區的寵物醫院列表，飼主可以透過下拉式選單選擇城市和區域。

1. 醫院詳細資訊：

點擊特定醫院後，顯示所選寵物醫院的地址、電話、急診資訊等詳細資訊。飼主可以透過地圖資訊了解當前位置與目的地之間的距離和路線，也可以跳轉到 Google Map 查看詳細路線。

1. 設定功能：
2. 修改個人資料：

飼主可以修改自己的個人資料，例如姓名、電子郵件等。

1. 修改寵物資料：

飼主可以新增寵物或修改現有寵物的資料，包括名稱、品種、年齡等。

1. 硬體設置：

飼主可以設定自動餵食的間隔時間和餵食份量。

1. 通知中心：

飼主可以設定餵食份量，並透過餵食按鈕手動餵食寵物。

1. 登出：

提供登出系統的功能，讓飼主安全登出。

### **3.2.3 系統操作與介面說明**

1. 飼主登入、註冊畫面：

當飼主進入應用程式時，會顯示登入畫面（如圖 3-1），如果飼主沒有帳號可以利用註冊按鈕跳轉到註冊畫面（如圖 3-2），輸入完基本資料後，按下註冊按鈕完成註冊，系統將會自動跳回登入畫面（如圖 3-1）；如果飼主已經擁有帳號，卻忘記當初所設定的密碼時，也可按下忘記密碼按鈕，跳轉到忘記密碼畫面（如圖 3-3），只要輸入當初設置的電子郵件，就能讓飼主重新設置新的密碼，並跳轉到登入畫面（如圖 3-1）。

|  |  |  |
| --- | --- | --- |
| 圖3- 1　登入畫面 | 圖3- 2　註冊畫面 | 圖3- 3　忘記密碼 |

1. 應用程式首頁：
2. 寵物基本資料

登入完後在首頁可以看到寵物的基本資料及設定的餵食時間、間隔時間及餵食份量（如圖3-4），點擊餵食按鈕可以讓飼主自行投放飼料。

1. 即時影像

點擊即時影像標籤可以看到監控畫面（如圖3-5）。

1. 餵食紀錄

在餵食紀錄的部分則會顯示上次餵食的時間，以及餵食是由寵物自行按鈕、飼主手動按鈕或是系統自動餵食，方便飼主了解寵物的進食時間（如圖3-6）。

|  |  |  |
| --- | --- | --- |
| 圖3- 4　寵物基本資訊 | 圖3- 5　即時影像 | 圖3- 6　餵食紀錄 |

1. 寵物醫院資訊及功能總覽：

飼主可以進入寵物醫院資訊內查詢關於寵物醫院的相關資料（如圖3-7），起始頁面的設置資料是沒有經過過濾的，會顯示全台寵物醫院資料，飼主可以在下拉式選單篩選所在的縣市和地區，資料將會篩選出飼主想查看的寵物醫院資料。

在列表中，飼主可點選特定醫院進行查看，應用程式會提供該醫院的詳細資料（如圖3-8），飼主可查看該醫院的詳細地址、聯絡資訊、急診資訊，以及地圖畫面，點選地圖可選擇是否跳轉到 Google Map 應用程式，以查看更詳細的路線規劃。

當飼主選擇下方的設定選項後就會進到設定頁面（如圖3-9），在這邊可以修改飼主、寵物、硬體的相關資訊。

|  |  |  |
| --- | --- | --- |
| 圖3- 7　寵物醫院總覽 | 圖3- 8　醫院詳細資訊 | 圖3- 9　功能總覽 |

1. 寵物資料修改及新增：

飼主可以查看目前所有已登錄的寵物列表（如圖3-10），點擊既有的寵物資料後，會進入修改資訊的介面（如圖3-12），預設為既有資料，點選各個欄位可以進行修改，回到寵物列表，點擊新增寵物，會進入預設各個欄位的值為空的介面（如圖3-11），飼主可以填入寵物資料，按下儲存按鈕，會回到寵物列表畫面（如圖3-10），並顯示修改後的資料，如果按下取消按鈕，資料則不會更動，直接返回寵物列表畫面（如圖3-10）。

|  |  |  |
| --- | --- | --- |
| 圖3- 10　寵物總覽 | 圖3- 11　新增寵物 | 圖3- 12　寵物資料變更 |

1. 個人基本資料變更及硬體設置：

點選功能總覽中（如圖3-9）的基本資料變更後，即可更改飼主的基本資料（如圖3-13），預設欄位為舊有資料，可透過點選欄位來更改資料，按下儲存後，會驗證各個欄位是否符合條件，或是否為空，假如驗證不通過則會顯示錯誤訊息，驗證通過後則顯示成功訊息；按下取消按鈕則會跳回功能總覽畫面（如圖3-9）。

1. 硬體設置：

點選功能總覽中（如圖3-9）的硬體設置後，飼主可自行設置寵物的餵食份量以及取食間隔時間（如圖3-14）。

|  |  |
| --- | --- |
| 圖3- 13　個人資料變更 | 圖3- 14　硬體設置 |

## **第三節　其他系統實測或實驗結果**

當飼主選擇按下按鈕（如圖3-16），按壓成功後按鈕就會亮燈，下方藍色的是伺服馬達（如圖3-15），伺服馬達轉動投放飼料的塑膠檔板，讓寵物可以食用飼料。而針對影像實際展示的過程，我們可以利用已架設好的ESP32-CAM將畫面轉成圖片在顯示在該頁面當中（如圖3-17），讓飼主能查看即時的寵物畫面。

|  |  |
| --- | --- |
| 圖3- 15　塑膠檔板及伺服馬達 | 圖3- 16　按鈕 |



圖3- 17　實際影像呈現

# **第四章 結論**

## **第一節　系統目標與特色**

我們系統最大的目標及特色就在於訓練寵物能夠自主餵食。為了避免寵物在剛開始還不曉得該如何控制按鈕，我們設計按鍵的次數限制。飼主也能透過訊息查看寵物的進食狀況，並透過架設好的攝影機，去捕捉即時的寵物畫面。當寵物發生突發狀況時，飼主也能利用本系統所提供的醫院資訊，去尋找適合的醫院，並了解到醫院的距離。運用上述功能，即使飼主無法親自照顧寵物，也能安心地利用此系統觀察及照料寵物。

## **第二節　未來展望**

希望未來針對攝影機及捕捉畫面這點可以做改善，由於我們測試設備並沒有那麼好，只能利用定時拍攝即時影像的方式呈現，無法做到 24 小時同步畫面，有機會的話可以針對這部分做優化及改進。

在觀察寵物影像的部分，若是未來能夠融合影像辨識，去判斷寵物的狀態跟日常行為是否有不同，並且能夠發出訊息通知飼主，藉此完善寵物照護的部分，也能增加實用性。

在實際應用的部分 ，可以為硬體設計成更利於實際使用的設備，例如製作模型，將硬體線路保護起來，例如製作模型，將硬體線路保護起來，以及安裝上飼料，進行以及安裝上飼料罐，實際運行餵食的動作。

在資料庫安全性的部分，可以針對不同系統管理員做不同的權限，避免任何人都能直接登入資料庫做增刪讀寫的功能。也可以改變直接從 Android Studio 直接連線到資料庫的設計，為資料庫多一層保障。

# **第五章 專題心得與檢討**

透過這次專題，我們最大的收穫便是教授所提供的建言。每次跟教授開會時，他總是給予我們許多建議和指導。在我們討論主題的階段，教授會針對每個人的主題提供不同的意見，協助我們更加明確地定義專題方向。而在確定主題後，教授也會著重於細節，像是如何突顯系統的特色，以及如何進一步完善系統。

在實際開發系統的過程中，我們可能會忽略一些細節，但是教授卻能夠從實務的角度給予建議。他會指出系統可能存在的問題，並提供改進的方法和方向。這些建議是我們在實作過程中額外的收穫。

透過不斷跟教授交流、討論，我們能夠不斷改進和提升我們的系統，並從中學習到實務上的經驗和技巧。

除了教授給予的指導外，我們的專題內容涉及到許多我們之前未曾學習或接觸過的領域，例如應該如何結合應用程式與硬體設備、如何導入 GOOGLE API 等等。目前的系統雖然已經大致完成，但仍有許多細節可以改進。

每個人都有各自負責的部分，因此需要一段時間來整合各個部分，學習其他知識和修正錯誤。這對於第一次接觸這些領域的我們來說，確實存在一些挑戰。

這次專題中，教授的建議和指導不僅對我們的專題具有重要影響，也對我們的專業成長有著深遠的影響。這次實作過程中所面臨的挑戰和學習也為我們提供了寶貴的經驗和知識。我們會持續努力，如果未來有更多時間來完善我們的系統，我們肯定會進一步改良系統！

# **第六章 參考文獻**

**中文文獻**

**（一）網站**

Eleg.aces.kc. (2020, December 7). *[Android][Kotlin]Google Map Decode Polyline 預估路徑/畫線 解碼教學*. Medium. <https://elegant-access-kc.medium.com/android-kotlin-google-map-api-decode-polyline-%E9%A0%90%E4%BC%B0%E8%B7%AF%E5%BE%91-%E7%95%AB%E7%B7%9A-%E8%A7%A3%E7%A2%BC%E6%95%99%E5%AD%B8-611d62644549>

Mrfengzh. (2019, October 20). *FragmentTransaction 的 Replace, Add, Hide, Show 的使用和区别.* CSDN博客. <https://blog.csdn.net/fzhhsa/article/details/102652620?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-102652620-blog-57416720.235%5Ev38%5Epc_relevant_sort_base2&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-102652620-blog-57416720.235%5Ev38%5Epc_relevant_sort_base2&utm_relevant_index=2>

HKT 線上教室. (2021, September 22). *Day 22：Spinner 下拉選單結合縣市鄉鎮小工具.* IT邦幫忙. <https://ithelp.ithome.com.tw/articles/102695533>

Wayne chen. (2018, May 4). *Android 基礎的 Fragment 使用方式.* Medium. <https://medium.com/@waynechen323/android-%E5%9F%BA%E7%A4%8E%E7%9A%84-fragment-%E4%BD%BF%E7%94%A8%E6%96%B9%E5%BC%8F-730858c12a43>

夜市小霸王. (2019, March 21). *第一篇ESP32 Arduino開發環境架設*. <https://youyouyou.pixnet.net/blog/post/119410732>

**英文文獻**

**（一）website**

Azhar. (2017, November 5). *How to Show Current Location on a Google Map on Android Using Kotlin?* Tutorialspoint. <https://www.tutorialspoint.com/how-to-show-current-location-on-a-google-map-on-android-using-kotlin>

Admin mindorks. (2019, July 1). *Shared ViewModel in Android: Shared between Fragments*. MindOrks. <https://blog.mindorks.com/shared-viewmodel-in-android-shared-between-fragments/>

Sara Santos*.ESP32-CAM Save Picture in Firebase Storage.*

<https://randomnerdtutorials.com/esp32-cam-save-picture-firebase-storage/>