



*Petagram*

寵生活



# 簡介

概念：

這是專為寵物創立的寵物社群媒體平台，統合了許多社群媒體功能，讓毛小孩也能有一個友善的交流平台~

開發動機：

近年來因為經濟壓力，選擇不生育的年輕人大幅增加，有許多人轉往養育毛小孩，導致台灣養育寵物的人口實現大幅度上升的情況，也衍生出許多商機，包括寵物友善餐廳、寵物友善景點、寵物友善旅遊、寵物保姆、寵物保健....等



# 首頁畫面

UI/UX設計概念：

APP的起始頁面是推薦內容是為了讓用戶能快速看到想要的內容，由後端Server對資料庫進行比對分析，統合出觀看次數最多的貼文內容，讓用戶打開APP的同時目光就能馬上被吸引過去！

將發布內容作為大ICON置頂，鼓勵用戶分享内容！

將信箱置頂，提醒用戶是否有新訊息！

將主要主題項目放置於上方黃色導覽欄，讓用戶能聚焦於APP的重點項目！

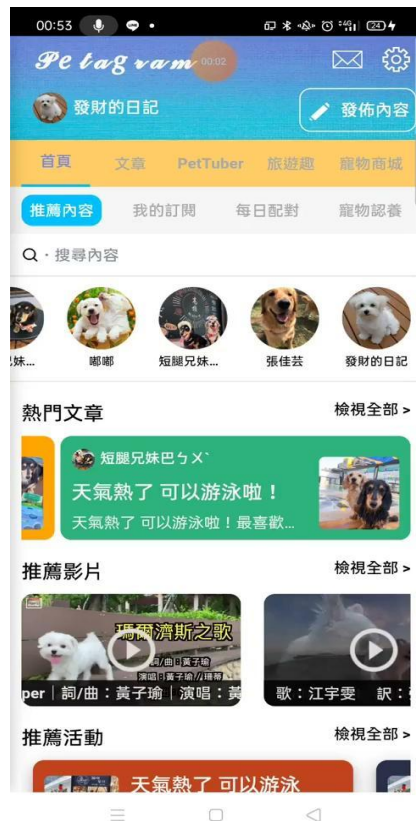
將次要主題作為頁面選擇項目，每個主要主題皆會涵蓋某些次要主題內容！



## 首頁畫面-推薦內容

集合知名用戶，熱門文章，推薦影片，推薦活動，讓用戶快速看到想要查看的內容，因為各類項目版型不同，所以對單一版型進行橫向排版，讓畫面看起來直觀且有規律！

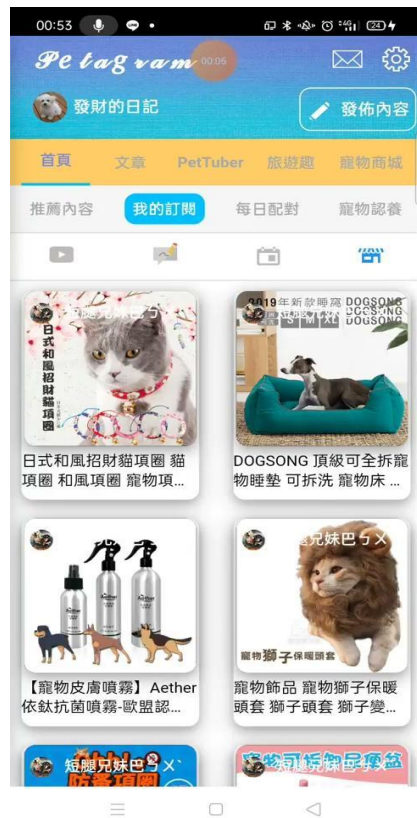
影像預覽(點擊播放):



## 首頁畫面-我的訂閱

快速查看你訂閱用戶的最新動態！

影像預覽(點擊播放):



## 首頁畫面-每日配對

讓你每天充滿著邂逅的驚喜！

影像預覽(點擊播放):



## 首頁畫面-寵物認養

鼓勵以領養代替認養，讓浪浪也能找到屬於自己的家！！

影像預覽(點擊播放):



## 文章畫面

這邊將文章做了個版分類，包括友善餐廳、友善景點、寵物旅遊、寵物保姆、寵物美容、寵物保健，讓用戶能自己選擇想要查看的文章類型!!

### 影像預覽(點擊播放):

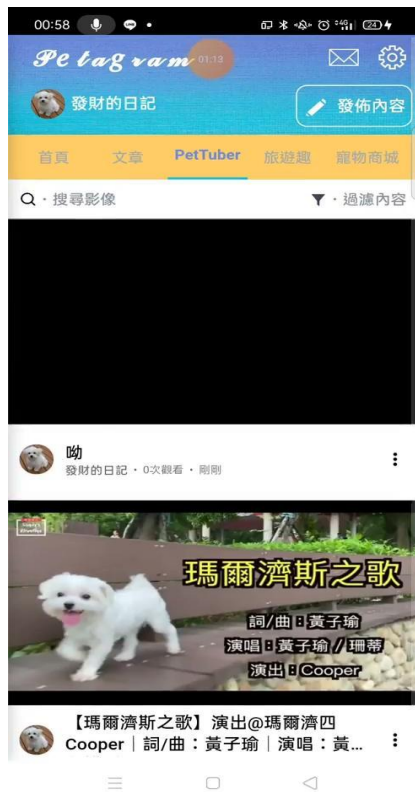




# PetTuber

這邊主要是做影片分享，讓用戶可以互相分享影片，影片串流採用 HLS 技術，使用 (Google Cloud Storage) 作為檔案伺服器，自動化的區域負載平衡，讓用戶能體驗最流暢的播放體驗！

影像預覽(點擊播放):



## 活動頁面

這邊讓用戶可以揪團出遊，並且可以報名參加，查看活動網頁，收藏活動內容！！

影像預覽(點擊播放):



## 商城內容

用戶可以在商城發佈商品，目前是採用連結導向的方式，後續則會嘗試串接綠界金流!!

影像預覽(點擊播放):



## 使用到的技術

前端網頁採用React，設計模式為MVVM，將前後端完全進行分割，而所有資料的產生都是透過ajax技術的方式去達成，後端僅負責建立API與開立網頁路徑，並同時結合自有開發框架Glitter進行(Android/IOS/WEB)三方的跨平台開發，採用單一網頁路徑，多個Iframe子頁的設計模式，概念結合WPA與安卓One Activity Multiple fragment的設計優點，能讓網頁顯示起來更加流暢，讓用戶操作網頁跟APP感受不到任何區別!!

後端Server使用kotlin(Ktor)，開發初期原先使用Java Spring，但是後續接觸到了Kotlin，而轉往使用Ktor，相較於純java的開發模式，Kotlin的語法糖眾多，能使語法大幅度精簡，且能與Java互相進行調用，大幅縮短了開發時程!!

影像處理使用FFMPEG，將影片壓縮至特定大小，在將每10秒分割成一個片段ts，然後產生m3u8目錄，與預覽圖片，之後上傳至Google Cloud Storage進行儲存，資料庫則做路徑的儲存

資料庫使用MYSQL，部分重要資料採用DES加密儲存，駭客想要破解資料庫資料的話，必須同時猜到加密模式以及加密金鑰！

# 自有框架Glitter介紹

Glitter是一個H5混合開發函式庫，與React native概念相同，皆是透過Javascript調用window.fun()和window.webkit.messageHandlers，去執行對應的Kotlin和Swift代碼，而開發者也可以依照需求去對功能函式進行拓展，同時保留了原生代碼的彈性，也能強調Android/iOS/Web三方畫面的一致性!!

右方代碼則示例了Glitter如何判斷裝置類型去呼叫原生代碼!!

代碼示例:

```
//取紀錄
this.getPro = function (tag, fun) {
  var id = glitter.callBackId += 1
  glitter.callBackList.set(id, fun)
  switch (glitter.deviceType) {
    case appearType.Web: {
      glitter.callBackList.get(id)(glitter.webPro[tag])
      glitter.callBackList.delete(id)
      return
    }
    case appearType.Android: {
      glitter.callBackList.get(id)(window.GL.getPro(tag))
      glitter.callBackList.delete(id)
      return
    }
    case appearType.Ios: {
      var map = {
        tag: tag,
        callback: id
      }
      window.webkit.messageHandlers.getPro.postMessage(JSON.stringify(map));
      break
    }
  }
}
```

# 程式載點

Android:

<https://play.google.com/store/apps/details?id=com.orange.petagram>

IOS:

<https://apps.apple.com/us/app/petagram%E5%AF%B5%E7%94%9F%E6%B4%BB/id1559642119>

Web(目前僅針對手機畫面進行排版):

<https://www.squarestudio.tw/Petagram/home>