第9章　程式

9-1 元件清單及其規格描述

▼表9-1-1 元件清單及其規格描述表(前端)

|  |  |  |
| --- | --- | --- |
| Dart檔案(前端)-App介面編寫 | | |
| 編號 | 檔案名稱 | 功能 |
| 1-1-1 | register\_page.dart | 供使用者進行會員註冊 |
| 1-1-2 | login\_page.dart | 供使用者進行會員登入 |
| 1-1-3 | forgetpw\_page.dart | 會員在忘記密碼時驗證其身分並供會員重置密碼 |
| 1-1-4 | control\_page.dart | 供使用者透過底部導覽列切換頁面 |
| 1-1-5 | class.dart | 設立共用元件在需要時調用 |
| 1-1-6 | housecard.dart | 設立共用的房屋卡片元件在需要時調用 |
| 1-1-7 | home\_page.dart | 供使用者查看首頁 |
| 1-1-8 | areasearch\_page.dart | 供使用者進行地區搜尋 |
| 1-1-9 | searchresult\_page.dart | 供使用者查看搜尋結果 |
| 1-1-10 | housedetail\_page.dart | 供使用者查看房屋詳細資料 |
| 1-1-11 | addhouse\_page.dart edithouse\_page.dart | 供會員刊登及編輯房屋 |
| 1-1-12 | createhouse\_page.dart | 載入會員已刊登之房屋並供會員可以進行刪除 |
| 1-1-13 | addsubscription\_page.dart | 供會員新增訂閱條件 |
| 1-1-14 | subscription\_page.dart  propertydetails\_page.dart | 載入會員已訂閱的條件以及查看符合條件的房屋 |
| 1-1-15 | personal\_page.dart | 供會員查看個人頁面 |
| 1-1-16 | modifypersonal\_page.dart | 供會員修改會員資料 |
| 1-1-17 | modifypassword\_page.dart | 供會員修改密碼 |
| 1-1-18 | collection\_page.dart | 供會員查看收藏紀錄 |
| 1-1-19 | clickrecord\_page.dart | 供會員查看瀏覽紀錄 |

9-2 其他附屬之各種元件

▼表9-2-1 部分程式碼－register\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | register \_page.dart |
| 目的 | 供使用者進行會員註冊 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart：用於構建 UI 元素。   + http 和 dart:convert：用於發送 HTTP 請求和處理 JSON 數據。   + shared\_preferences：用於儲存會員資料。 * **初始化屬性**：   + usernameController：控制使用者姓名的輸入框。   + emailController：控制電子郵件的輸入框。   + phoneController：控制電話號碼的輸入框。   + passwordController 和 confirmPasswordController：分別控制密碼和確認密碼的輸入框。   + \_radiogroupA：記錄性別選項的數值，1 為男性，2 為女性。   + \_isPasswordVisible 和 \_isConfirmPasswordVisible：控制密碼和確認密碼是否可見。   **2. 函數與功能**   * **註冊功能**：   + registerUser：     - 將註冊資料發送至後端伺服器。     - 若密碼與確認密碼不一致，彈出對話框提示使用者重新輸入。     - 註冊成功後，顯示成功訊息並導向登入頁面。     - 註冊失敗時，根據伺服器返回的錯誤訊息顯示錯誤對話框。 * **性別選擇功能**：   + 使用 Radio 控讓使用者選擇性別，並通過 \_handleRadioValuChanged 函數更新選中的性別。 * **顯示密碼功能**：   + 使用 IconButton 控制密碼和確認密碼是否可見，點擊Icon後會觸發 \_isPasswordVisible 和 \_isConfirmPasswordVisible 的狀態變更。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 作為頁面容器。   + 頁面由表單元素組成，通過 ListView 顯示使用者輸入欄位。 * **輸入欄位**：   + 包含姓名、性別、電子郵件、電話號碼、密碼和確認密碼的輸入欄位。   + 每個輸入欄位都有對應的 TextField 控制器來儲存使用者輸入的資料，並且為每個輸入欄位設置了提示文本和樣式。 * **註冊按鈕**：   + 點擊按鈕後調用 registerUser 函數，發送註冊請求。 | | | |

▼表9-2-2 部分程式碼－login\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | login\_page.dart |
| 目的 | 供使用者進行會員登入 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart：用於構建 UI 元素。   + http 和 dart:convert：用於發送 HTTP 請求與處理 JSON 數據。   + shared\_preferences：用於儲存會員本地資料。   + class.dart：定義了相關類別。 * **初始化屬性**：   + gmailController：控制電子郵件輸入框的數據。   + passwordController：控制密碼輸入框的數據。   + \_isPasswordVisible：控制密碼是否可見。   **2. 函數與功能**   * **登入功能**：   + loginUser：     - 向後端伺服器發送電子郵件和密碼進行驗證，並接收會員數據。     - 若登入成功，將會員資料（如姓名、電子郵件、電話、性別等）儲存於 SharedPreferences 中。     - 初始化avoriteManager()，並顯示成功的彈窗，導向到控制頁面。     - 若登入失敗，顯示錯誤訊息。 * **顯示密碼功能**：   + 點擊密碼框右側的Icon，調用 \_isPasswordVisible 切換密碼可見性。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 作為頁面容器。   + 頁面由上下兩部分組成：     1. **上半部分**：顯示背景圖像與返回Icon。     2. **下半部分**：包括電子郵件、密碼輸入框，和登入按鈕。 * **電子郵件與密碼輸入框**：   + 兩個輸入框分別控制 gmailController 和 passwordController，並且在密碼框有顯示/隱藏功能。 * **登入按鈕**：   + 點擊後調用 loginUser 函數，發送登入請求。 * **額外功能**：   + 在頁面底部提供 "註冊帳號" 和 "忘記密碼?" 兩個可點擊文字，分別跳轉至註冊和忘記密碼頁面。 | | | |

▼表9-2-3 部分程式碼－forgetpw\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | forgetpw\_page.dart |
| 目的 | 會員在忘記密碼時驗證其身分並供會員重置密碼 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart：用於構建 UI 元素。   + http 和 dart:convert：用於發送 HTTP 請求和處理 JSON 數據。 * **初始化屬性**：   + gmailController：控制輸入電子郵件的輸入框。   + verifyController：控制驗證碼輸入的輸入框。   + newPasswordController 和 confirmPasswordController：分別控制新密碼和確認新密碼的輸入框。   + showverifyfield：是否顯示驗證碼輸入框。   + isVerifyButtonEnable：驗證碼按鈕是否可用。   **2. 函數與功能**   * **發送驗證碼功能**：   + sendemail：     - 向伺服器發送電子郵件以請求發送驗證碼。     - 成功後顯示驗證碼輸入框，並在對話框中提示已發送驗證碼。 * **驗證驗證碼功能**：   + verifycode：     - 發送電子郵件和驗證碼至伺服器進行驗證。     - 驗證成功調用 showChangePasswordDialog 允許會員修改密碼。     - 若驗證失敗，顯示錯誤訊息。 * **重置密碼功能**：   + resetpassword：     - 驗證新密碼和確認密碼是否一致。     - 發送新的密碼到伺服器進行密碼重置。     - 成功後跳轉到登入頁面並提示會員密碼修改成功。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 作為頁面容器。 * **輸入欄位**：   + 含電子郵件和驗證碼欄位。當驗證碼成功發送後，驗證碼欄位會顯示。 * **按鈕功能**：   + 「傳送驗證碼」按鈕：點擊後調用 sendemail 函數發送驗證碼。   + 「驗證」按鈕：當輸入的驗證碼長度達到 6 位時，按鈕變為可用，點擊後調用 verifycode。 * **密碼重置對話框**：   + showChangePasswordDialog：彈出對話框，允許會員輸入新密碼和確認新密碼，點擊「確認」後調用 resetpassword 函數進行密碼重置。 | | | |

▼表9-2-4 部分程式碼－controll\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | controll\_page.dart |
| 目的 | 供使用者透過底部導覽列切換頁面 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和頁面**：   + flutter/material.dart：用於構建 UI 元素。   + createhouse\_page.dart, home\_page.dart, search\_page.dart, personal\_page.dart, 和 subscription\_page.dart：不同頁面文件的引用。 * **初始化屬性**：   + currentTab：當前選中的頁面索引，默認值為 0，即HomePage()。   + screens：包含不同頁面的 Widget 列表，包括 HomePage, SearchPage, CreateHousePage, SubscriptionPage, AccountPage 等。   + bucket：用於保存頁面的狀態。   + currentScreen：當前顯示的頁面，默認為 HomePage()。   **2. 函數與功能**   * **頁面切換功能**：   + \_buildTabItem：構建底部導航欄按鈕。點擊不同按鈕會改變 currentScreen 和 currentTab，從而更新顯示的頁面。 * **返回首頁功能**：   + 點擊FloatingActionButton 將 currentScreen 設置為 HomePage() 並將 currentTab 重置為 0。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 作為頁面容器，包含主要頁面顯示區域和底部導航欄。   + PageStorage 用於保存頁面狀態。 * **浮動按鈕**：   + FloatingActionButton：位於底部中央，點擊後跳轉至首頁 HomePage()。 * **底部導航欄**：   + 使用 BottomAppBar 作為底部導航列的容器，包含四個按鈕，分別對應不同頁面。   + 當前選中的頁面按鈕將改變顏色。 | | | |

▼表9-2-5 部分程式碼－class.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | class.dart |
| 目的 | 設立共用元件供調取 | | |
| 部分程式碼 | | | |
| **FavoriteManager**  **1. 初始化與設置**   * **載入需要的模組和頁面**：   + shared\_preferences: 用於本地存儲會員收藏的房屋數據。   + http 和 dart:convert: 用於與後端進行 API 請求，獲取會員收藏的房屋列表。 * **初始化屬性**：   + \_favoriteHids: 用於存儲會員收藏房屋的 hid。   + FavoriteManager:確保在整個應用中只有一個 FavoriteManager 實例。通過 \_instance 和 \_internal() 的結合實現。   **2. 函數與功能**   * **initializeFavorites**：   + 初始化會員收藏列表，首先檢查 \_favoriteHids 是否為空，如果是空的，則通過 SharedPreferences 獲取 member\_id，然後調用後端 API獲取用戶收藏的房屋。   + 成功獲取後，將房屋的 hid 轉換成 Set<String>，存入 \_favoriteHids。 * **getter 方法**：   + favoriteHids: 用來返回當前收藏的房屋 hid 集合。   **Icons 與房屋設備**  **1. 初始化與設置**   * **allservices1**：   + 包含了所有可能的房屋設備名稱的列表。這些設備名稱用於展示 UI 元素，並與相應的Icon進行匹配。 * **servicesIcons1**：   + 設備名稱到Icon的映射表，包含了設備名稱以及它們對應的Icon。 | | | |

▼表9-2-6 部分程式碼－housecard.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | housecard.dart |
| 目的 | 設立共用的房屋卡片元件在需要時調用 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart 用於構建 Flutter UI 元素。 * **初始化屬性**：   + houseData：保存該房屋的詳細資料   + isFavorite：布林值，顯示該房屋是否被收藏。   + showFavoriteIcon：布林值，決定是否顯示愛心Icon，默認值為 true。   + onFavoriteToggle：點擊愛心Icon時調用的回調函數，切換收藏狀態。   + onTap：會員點擊房屋卡片時調用的回調函數，打開房屋詳細資料頁面。   **2. 函數與功能**   * **build 函數**：   + 構建卡片的佈局，呈現房屋的圖片、基本訊息。   + GestureDetector 用來監聽點擊事件，當使用者點擊卡片時，觸發 onTap 回調函數。 * **圖片顯示邏輯**：   + 使用 Image.network 加載房屋圖片，若加載失敗，則顯示本地圖片。 * **愛心Icon的顯示邏輯**：   + 通過 showFavoriteIcon 判斷是否顯示愛心Icon。   + 愛心Icon根據 isFavorite 的狀態，顯示實心或空心的愛心Icon。   + 當會員點擊愛心Icon時，觸發 onFavoriteToggle 回調函數來切換收藏狀態。   **3. 頁面佈局與構建**   * **卡片佈局**：   + 使用 Container 作為最外層容器，設置卡片的邊距、大小、邊框和陰影效果。   + 卡片的內容由 Row 組成，分別顯示房屋圖片和文字描述：     - 左側顯示房屋的圖片，並通過 ClipRRect 設置圖片的圓角效果。     - 右側顯示房屋的基本訊息，包括房屋標題、大小、地區等。 | | | |

▼表9-2-7 部分程式碼－home\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | home\_page.dart |
| 目的 | 供使用者查看首頁 | | |
| 部分程式碼 | | | |
|  | | | |

▼表9-2-8 部分程式碼－areasearch\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | areasearch\_page.dart |
| 目的 | 供使用者進行地區查詢 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和頁面**：   + flutter/material.dart：用於構建 UI 元素。   + SearchResultPage：用於顯示搜尋結果的頁面引用。 * **初始化屬性**：   + \_selectedCity：選擇的城市。   + \_selectedDistrict：選擇的區域。   + \_selectedRoomType：選擇的房屋類型。   + \_selectedRentalRange：選擇的租金範圍。   + \_selectedRoomCount：選擇的房間數量。   + \_selectedHouseSize：選擇的坪數。   + \_selectedHouseType：選擇的房屋型態。   + \_selectedOtherOptions：選擇的其他篩選條件。   **2. 函數與功能**   * **搜尋功能**：   + \_search：當按下搜尋按鈕時，會調用此函數。負責處理所有的選擇條件，將它們轉換為適合的格式，並將它們作為搜尋參數發送到後端 API。 * **彈出選擇列表**：   + \_showSelectionBottomSheet：顯示底部彈窗，允許使用者選擇篩選條件。 * **篩選條件選擇**：   + \_roomTypeBottomSheet：選擇房屋類型的彈窗。   + \_rentalBottomSheet：選擇租金範圍的彈窗，支持自定義租金範圍。   + \_roomCountBottomSheet：選擇房間數量的彈窗。   + \_houseSizeBottomSheet：選擇坪數範圍的彈窗。   + \_houseTypeBottomSheet：選擇房屋型態的彈窗。   + \_otherSelectionBottomSheet：選擇其他篩選條件的彈窗。   **3. 頁面佈局與構建**   * **搜尋頁面佈局**：   + 使用 ListView 作為頁面的主容器，顯示篩選條件的按鈕列表。   + 每個按鈕使用 \_buildListTile 函數生成，點擊後彈出相應的篩選條件選擇彈窗。   + 在最下方有搜尋按鈕，點擊後調用\_search 函數 | | | |

▼表9-2-9 部分程式碼－searchresult\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | searchresult\_page.dart |
| 目的 | 供使用者查看搜尋結果 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和頁面**：   + flutter/material.dart：用於構建 UI 元素。   + http 和 dart:convert：用於處理 API 請求與解析結果。   + shared\_preferences：用於儲存與檢索本地數據。 * **初始化屬性**：   + \_displayResults：用來存放當前顯示的搜尋結果，初始為空列表。   + \_currentMax：定義當前顯示的結果數量，初始為 8，使用於滾動加載。   + \_scrollController：控制滾動的控制器，用來監聽滾動事件並加載更多結果。   + \_isLoading：表示是否處於加載狀態，初始為 false。   **2. 函數與功能**   * **點擊記錄功能**：   + clickrecord：當點擊房屋卡片時，調用此函數記錄點擊操作。它發送一個 POST 請求到伺服器，記錄會員對該房屋的點擊。 * **初始化搜尋結果**：   + 在 initState 方法中，將搜尋結果設置到 \_displayResults 中，並初始顯示前 8 個結果。同時設置滾動控制器來監聽滾動事件。 * **滾動加載功能**：   + \_loadMore：當滾動到底部時，會加載更多結果。每次加載 8 個結果，並更新 \_displayResults 列表。 * **獲取房屋詳情功能**：   + fetchHouseDetails：當點擊房屋卡片時，調用此函數來獲取房屋的詳細訊息，然後導航到 HouseDetailPage 顯示詳情。 * **收藏功能**：   + \_toggleFavorite：當會員點擊收藏按鈕時，調用此函數來切換收藏狀態。它向伺服器發送 POST 或 DELETE 請求來更新收藏狀態，並更新本地的收藏列表。   **3. 頁面佈局與構建**   * **搜尋結果頁面佈局**：   + 使用 CustomScrollView 來顯示搜尋結果列表，並且支援滾動加載。   + 每個房屋結果由 HouseCard 構建，展示房屋基本訊息、收藏按鈕和點擊卡片以查看詳情。 * **按下房屋卡片後調用**：   調用 clickrecord 記錄點擊操作。  調用 fetchHouseDetails 獲取房屋詳細訊息，並導航到HouseDetailPage。   * **滾動加載更多結果**：   + 當使用者滾動到頁面的底部時，觸發 \_loadMore 函數來加載更多搜尋結果，並更新 \_displayResults 列表。 * **收藏操作**：   + 當點擊收藏按鈕時，觸發 \_toggleFavorite 函數來更新收藏狀態，並發送請求到伺服器更新遠端收藏訊息。 | | | |

▼表9-2-10 部分程式碼－housedetail\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | housedetail\_page.dart |
| 目的 | 供使用者查看房屋詳細資料 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和頁面**：   + flutter/material.dart：用於構建 UI 元素。   + http 和 dart:convert 用於發送 HTTP 請求與處理 JSON 數據。   + shared\_preferences：用於存儲和檢索本地用戶數據。 * **初始化屬性**：   + \_current：表示目前在房屋圖片輪播中的位置，初始值為 0。   + \_getMemberId：用於從 SharedPreferences 中獲取 member\_id 來標識用戶。   **2. 函數與功能**   * **底部按鈕功能**：   + \_onBottomAppBarPressed：當用戶點擊底部按鈕時，系統會先通過 \_getMemberId 函數獲取用戶的 member\_id，然後發送 POST 請求到伺服器記錄點擊行為。成功後，將帶有房屋hid的訊息發送到 LINE 智能助手。 * **展示房屋介紹功能**：   + \_buildIntroduce：顯示房東資訊和房屋簡介。通過 TextButton 提供一個按鈕，可以查看更多詳細介紹，觸發 \_showBottomSheet。   **3. 頁面佈局與構建**   * **圖片輪播功能**：   + CarouselSlider：用來顯示房屋圖片，支持自動輪播功能。當輪播的頁面改變時，通過 setState 更新當前顯示的圖片索引 \_current。 * **房屋基本訊息展示**：   + **房屋標題**：使用 Text 顯示房屋的標題。   + **房屋屬性**：使用 \_buildInfoChip 來顯示房屋訊息會根據後端的 houseDetails 數據動態生成。   + **租金和押金**：顯示房屋的租金和押金。 * **房屋設備展示**：   + **設備列表**：根據 houseDetails 中的 furniture 數據，動態生成每個設備的Icon和名稱。Icon的顏色根據設備是否可用來決定，如果設備不可用，會使用刪除線來表示。 * **房屋簡介**：   + 通過 \_buildIntroduce 方法構建簡短的房屋介紹，並提供按鈕供用戶點擊以查看更多詳細內容。 | | | |

▼表9-2-11 部分程式碼－addhouse\_page.dart、edithouse\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | addhouse\_page.dart edithouse\_page.dart |
| 目的 | 供會員刊登及編輯房屋 | | |
| 部分程式碼 | | | |
| **AddHousePage**  **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart 用於構建 Flutter UI 元素。   + http 和 dart:convert 用於發送 HTTP 請求與處理 JSON 數據。   + image\_picker 用於圖片選擇功能。   + shared\_preferences 用於讀取已登入會員的 ID。 * **初始化屬性**：   + \_picker：負責圖片選擇的控制器。   + 各類 TextEditingController 控制各個輸入框的內容。   + \_imageFileList：保存已選擇的圖片列表。   **2. 函數與功能**   * **圖片選擇功能**：   + \_pickImages：使用 ImagePicker 選擇多張圖片並將其加入圖片列表中。   + \_deleteImage：從圖片列表中刪除指定索引的圖片。 * **表單提交功能**：   + \_submitData：     - 檢查使用者是否已登入，未登入時顯示對話框提示。     - 組合表單數據，並將其發送至後端 API。     - 若請求成功，回傳新增的房屋數據，並返回上一頁。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold，並使用 ListView 展示輸入框與選項。 * **輸入框與選擇功能**：   + 包含地址、標題、描述、租金等 TextFormField 輸入框。   + 使用 showModalBottomSheet 彈出對話框選擇縣市、區域、家具等選項。 * **圖片展示與操作**：   + 可以通過 GridView 查看已選擇的圖片，並可以刪除不需要的圖片。   **EditHousePage**  **1. 初始化與設置**   * **載入需要的模組和庫**：   + 與 AddPage 相同，載入 flutter/material.dart、http、dart:convert、image\_picker 和 shared\_preferences。 * **初始化屬性**：   + 將傳遞過來的houseData 初始化到對應的 TextEditingController 和選項中。   + 圖片列表 \_imageFileList 會根據 houseData['images'] 的內容初始化，並確保圖片來源正確（本地或 URL）。   **2. 函數與功能**   * **圖片選擇功能**：   + 與 AddPage 類似，允許新增與刪除房屋圖片，並將變更的圖片保存到 \_changedFields 中，確保更新時只提交更改部分。 * **表單提交功能**：   + \_submitData：檢查更改的字段並將它們與房屋 hid 提交後端 API。   **3. 頁面佈局與構建**   * **頁面佈局**：   + 與 AddPage 類似，使用 ListView 展示已填寫的房屋訊息並允許編輯。 * **修改邏輯**：   + 會員可以修改房屋的字段，修改後的數據將被儲存在 \_changedFields 中，只提交有變動的數據以減少負擔。 | | | |

▼表9-2-12 部分程式碼－createhouse\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | createhouse\_page.dart |
| 目的 | 載入會員已刊登之房屋並供會員可以進行刪除 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart 用於構建 UI。   + http 和 dart:convert 用於與後端 API 通信，處理 HTTP 請求和 JSON 數據。   + shared\_preferences 用於儲存和讀取會員資料。 * **初始化屬性**：   + createhouses：用於儲存從服務器獲取的房屋列表。   + selectedHouseIndex：用來追蹤當前選中的房屋索引，以顯示覆蓋層。   + selected：記錄房屋是否被選中，用於多選操作（顯示刪除選項）。   + showcheckbox：控制是否顯示刪除Icon。 * **初始設置**：   + initState：在頁面初始化時，通過 \_fetchHousesFromServer() 從服務器獲取房屋數據，並將數據儲存到 SharedPreferences 中。   **2. 函數與功能**   * **獲取房屋數據**：   + \_fetchHousesFromServer：通過會員 ID 從後端獲取房屋數據，並將其儲存到 createhouses 和 SharedPreferences 中。 * **新增房屋**：   + \_addHouse：在會員添加新房屋後，將房屋數據加入 createhouses，並更新 SharedPreferences 中的數據。 * **刪除房屋**：   + \_deleteHouse：刪除指定 hid 的房屋，並同步更新服務器和本地儲存（SharedPreferences）。 * **跳轉到新增房屋頁面**：   + \_navigateToAddHousePage：跳轉到 AddPage 頁面，並在會員新增房屋後，重新獲取房屋列表。 * **獲取房屋詳細資料**：   + fetchHouseDetails：通過房屋 ID 獲取房屋詳細資料，並跳轉到 CreateHouseDetailPage 頁面顯示該房屋訊息。 * **顯示與隱藏覆蓋層**：   + \_showOverlay：顯示當前選中的房屋覆蓋層，用於顯示房屋的編輯和瀏覽選項。   + \_hideOverlay：隱藏覆蓋層。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold建立頁面結構，並展示房屋列表。 * **房屋列表顯示**：   + 使用 ListView.builder 來動態生成房屋卡片。   + 如果房屋列表為空，則顯示提示文字 "尚未有刊登物件"。 * **操作按鈕與功能**：   + ElevatedButton：用於跳轉到新增房屋的頁面。   + 當 showcheckbox 為 true 時，房屋卡片顯示刪除Icon，點擊Icon可以刪除房屋。   + 當選中房屋卡片時，顯示瀏覽和編輯房屋的按鈕。 | | | |

▼表9-2-13 部分程式碼－addsubscription\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | addsubscription\_page.dart |
| 目的 | 供會員新增訂閱條件 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart 用於構建 UI 元素。   + shared\_preferences 用於處理會員 ID 的存取。 * **初始化屬性**：   + \_selectedCity：當前選中的縣市。   + \_selectedAreas：選中的地區列表。   + \_selectedtype：選中的房屋類型。   + \_selectedRentalRange：選中的租金範圍。   + \_selectedRoomCount：選中的房間數。   + \_selectedHouseSize：選中的房屋坪數。   + \_selectedTypes：選中的房屋型態。   **2. 函數與功能**   * **表單提交功能**：   + 點擊「新增訂閱」按鈕後，組合所有選擇的條件並提交到後端 API。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 和 AppBar 建立頁面結構。   + 使用 ListTile 和 Divider 組織各個選項的顯示和點擊選擇。 * **表單輸入與確認**：   + 在使用者選擇縣市、地區、房屋類型、租金等條件後，點擊「新增訂閱」按鈕，將條件提交給後端並關閉頁面。 | | | |

▼表9-2-14 部分程式碼－subscription\_page.dart、propertydetails\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | subscription\_page.dart  propertydetails\_page.dart |
| 目的 | 載入會員已訂閱的條件以及查看符合條件的房屋 | | |
| 部分程式碼 | | | |
| **SubscriptionPage**  **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart 用於構建 UI 元素。   + http 和 dart:convert 用於發送 HTTP 請求與處理 JSON 數據。   + shared\_preferences 用於處理本地儲存，獲取會員 ID 與訂閱條件。 * **初始化屬性**：   + subscriptions：儲存所有的訂閱條件與房屋結果的列表。   **2. 函數與功能**   * **訂閱條件處理**：   + \_fetchProperties：從後端獲取符合訂閱條件的房屋列表，並將房屋結果保存至對應的訂閱條件中。   + \_saveSubscriptions：將當前的訂閱條件保存至本地 SharedPreferences。   + \_loadSubscriptions：從後端加載會員的所有訂閱條件，並將其初始化到 subscriptions 中。 * **新增訂閱**：   + \_addSubscription：將新的訂閱條件發送到後端並添加到訂閱列表中，並同步加載符合該訂閱條件的房屋數據。 * **刪除訂閱**：   + \_removeSubscription：根據訂閱 ID 刪除指定的訂閱條件，並同步更新後端與本地儲存。 * **更新最後查看時間**：   + \_updateLastCheckTime：在點擊查看房屋詳細資料後，更新該訂閱的最後查看時間。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 和 AppBar 構建頁面結構，提供標題與新增訂閱按鈕。 * **訂閱條件顯示**：   + 使用 ListView.builder 迭代顯示每個訂閱條件卡片，展示城市、地區、房屋類型、租金、格局等條件，並提供刪除訂閱按鈕。 * **點擊訂閱卡片**：   + 點擊訂閱條件卡片後，會觸發 \_fetchProperties 從後端獲取對應的房屋數據，並跳轉到 PropertyDetailsPage 顯示符合條件的房屋列表。   **PropertyDetailsPage**  **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart 用於構建 UI。   + http 和 dart:convert 用於與後端 API 通信，處理 HTTP 請求和 JSON 數據。   + shared\_preferences 用於儲存和讀取會員資料。 * **初始化屬性**：   + properties：用於儲存傳遞進來的房屋列表。   + subscription：記錄訂閱的條件。   + subscriptionId：訂閱 ID，用於回傳時使用。   + onReturn：在返回上頁時執行的回調函數。   **2. 函數與功能**   * **返回處理**：   + \_handlePop：當返回時觸發 onReturn 函數，並傳遞 subscriptionId。 * **收藏功能**：   + \_toggleFavorite：切換收藏狀態，向後端發送新增或刪除收藏的請求，並同步更新本地狀態。如果操作失敗則恢復之前的狀態。 * **獲取房屋詳細資料**：   + fetchHouseDetails：根據房屋 ID 獲取詳細資料，並跳轉到詳細頁面顯示。 * **記錄點擊操作**：   + clickrecord：將會員的點擊記錄發送至後端。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 和 AppBar 建立頁面結構，顯示房屋結果或提示沒有新的房屋。 * **房屋列表顯示**：   + 使用 ListView.builder 動態生成房屋卡片，展示每個房屋的圖片和詳情，並提供收藏按鈕和點擊查看詳細的功能。 | | | |

▼表9-2-15 部分程式碼－personal\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | personal\_page.dart |
| 目的 | 供會員查看個人頁面 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart：用於構建 UI 元素。   + shared\_preferences：用於處理會員資料的存取，如使用者名稱和 Gmail。   + image\_picker：用於圖片選擇功能。   + dart:io：處理本地文件（圖片）的功能。   + permission\_handler：用於請求相機、相片和儲存權限。 * **初始化屬性**：   + \_image：用於儲存使用者上傳的頭像圖片。   + username 和 gmail：分別用於顯示使用者名稱和 Gmail。 * **權限請求**：   + requestPermissions：請求相機、相片、儲存的權限。   **2. 函數與功能**   * **圖片選擇功能**：   + \_pickImage：用於選擇圖片，可以從相簿或相機中選取，並將選擇的圖片展示在頁面上。 * **會員資料載入**：   + \_loadmembers：從 SharedPreferences 中讀取會員名稱和 Gmail 並更新 UI。 * **會員登出功能**：   + \_logout：清除會員資料並跳轉回控制頁面。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 和 ListView 來構建頁面。   + 頭像、名稱、Gmail 和登出按鈕放置在卡片中，提供點擊頭像修改功能。 * **圖片與按鈕操作**：   + 點擊頭像時，彈出選擇相簿或相機的選項，並顯示選取的圖片。   + 提供「登出」按鈕供使用者退出系統，並清除已儲存的會員資料。 * **功能列表**：   + 提供「個人資料」、「更改密碼」、「我的收藏」、「瀏覽紀錄」等選項，點擊後可跳轉到對應的頁面。 | | | |

▼表9-2-16 部分程式碼－modifypersonal\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | modifypersonal\_page.dart |
| 目的 | 供會員修改會員資料 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **初始化屬性**：   + \_nameController、\_gmailController 和 \_phoneController：控制姓名、Gmail 和電話的輸入。   + \_gender：顯示會員性別。   + \_image：負責顯示並儲存上傳的頭像圖片。   **2. 函數與功能**   * **圖片選擇功能**：   + \_pickImage：可以從相機或相簿中選擇頭像圖片，並更新顯示。 * **會員資料載入與保存**：   + \_loadUserData：從 SharedPreferences 中加載會員的姓名、Gmail、電話號碼和性別等資料。   + \_saveUserData：將會員修改後的訊息提交到伺服器，並根據伺服器回應更新本地資料。若修改成功，將彈出提示框。 * **修改按鈕點擊與調用**：   + 會員點擊 "確認" 按鈕後，會調用 \_saveUserData 函數，將所有修改後的個人資料發送到伺服器進行更新，並在成功後提示修改成功。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 和 ListView 構建頁面，頁面包含頭像圖片、姓名、性別（不可修改）、Gmail 和電話號碼的輸入欄位。 * **圖片與輸入框操作**：   + 點擊頭像允許會員選擇新的圖片。   + 各輸入框允許編輯姓名、Gmail 和電話。點擊 "確認" 按鈕後，會觸發 onPressed 事件，調用 \_saveUserData 來提交修改。 | | | |

▼表9-2-17 部分程式碼－modifypassword\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | modifypassword\_page.dart |
| 目的 | 供會員修改密碼 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart：用於構建 UI 元素。   + shared\_preferences：用於加載會員的 Gmail 資訊。   + http 和 dart:convert：用於發送修改密碼請求。 * **初始化屬性**：   + \_oldPasswordController、\_newPasswordController 和 \_confirmPasswordController：控制舊密碼、新密碼和確認新密碼的輸入。   + \_isPasswordVisible、\_isPasswordVisible1 和 \_isPasswordVisible2：控制密碼輸入框的顯示與隱藏狀態。   **2. 函數與功能**   * **密碼修改功能**：   + changePassword：檢查新密碼與確認密碼是否一致，然後將密碼修改請求發送到伺服器。若密碼修改成功，將顯示提示框，否則顯示錯誤訊息。 * **修改按鈕點擊與調用**：   + 點擊 "確認" 按鈕會調用 changePassword 函數，檢查密碼是否一致並向伺服器發送密碼更新請求。成功後，彈出成功對話框，失敗則顯示錯誤訊息。   **3. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 和 SingleChildScrollView 構建頁面，頁面包含舊密碼、新密碼和確認新密碼的輸入框。 * **密碼輸入與顯示控制**：   + 使用 TextFormField 構建密碼輸入框，通過點擊Icon可以切換密碼的顯示狀態。點擊 "確認" 按鈕後，會調用 changePassword 提交密碼修改。 | | | |

▼表9-2-18 部分程式碼－collection\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | collection\_page.dart |
| 目的 | 供會員查看收藏紀錄 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart：用於構建 UI 元素。   + http 和 dart:convert：用於發送 HTTP 請求與處理 JSON 數據。   + shared\_preferences：用於讀取當前登入會員的 ID。   + housecard.dart：用於展示房屋列表的組件。   + housedetail\_page.dart：用於顯示房屋詳細資訊的頁面。 * **初始化屬性**：   + \_favoriteHousesFuture：儲存收藏的房屋數據。   + \_selectedHouses：儲存會員選中的房屋 hid 列表。   + ismodifyclicked：控制是否進入編輯模式。   **2. 函數與功能**   * **收藏房屋載入功能**：   + fetchFavorites：從伺服器加載當前會員收藏的房屋，並返回房屋數據。若會員未登入，則顯示提示。 * **房屋詳情加載功能**：   + fetchHouseDetails：從伺服器加載選中房屋的詳細數據，並跳轉到房屋詳情頁面。 * **刪除收藏功能**：   + deleteSelectedFavorites：刪除選中的收藏房屋，發送請求到伺服器，並更新本地的收藏狀態。成功刪除後，會更新收藏列表，並清空選中項。   **3. 修改按鈕點擊與調用**   * **編輯與刪除模式**：   + 點擊 "編輯" 進入編輯模式，顯示可勾選的收藏項目。   + 若選中收藏房屋，顯示 "清除" 按鈕，點擊後會刪除選中的收藏。   + 若未選中收藏房屋，點擊 "完成" 退出編輯模式。   **4. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 和 FutureBuilder 構建頁面，根據當前的收藏數據展示房屋卡片或提示 "尚未有收藏"。 * **房屋卡片與編輯功能**：   + 展示房屋卡片，若進入編輯模式，會在每個房屋卡片顯示勾選框。點擊勾選框可選中或取消選中房屋。 * **全選功能**：   + 若進入編輯模式，顯示全選框，點擊後可以選擇或取消選擇所有收藏的房屋。 | | | |

▼表9-2-19 部分程式碼－clickrecord\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | clickrecord\_page.dart |
| 目的 | 供會員查看瀏覽紀錄 | | |
| 部分程式碼 | | | |
| **1. 初始化與設置**   * **載入需要的模組和庫**：   + flutter/material.dart：用於構建 UI 元素。   + http 和 dart:convert：用於發送 HTTP 請求與處理 JSON 數據。   + shared\_preferences：用於讀取會員的 ID。   + class.dart 和 housecard.dart：自定義類別與房屋卡片顯示組件。   + housedetail\_page.dart：用於顯示房屋詳細資訊的頁面。 * **初始化屬性**：   + \_clickHistory：儲存會員的點擊紀錄。   + \_isLoading：控制資料是否正在加載。   **2. 函數與功能**   * **瀏覽紀錄加載功能**：   + \_fetchClickHistory：根據會員 ID 從伺服器加載會員的點擊紀錄，並更新 \_clickHistory。 * **收藏功能**：   + \_toggleFavorite：當會員點擊收藏按鈕時，發送請求到伺服器，切換該房屋的收藏狀態。根據是否已收藏，發送 POST 或 DELETE 請求，並更新本地的收藏狀態。 * **房屋詳情加載功能**：   + fetchHouseDetails：從伺服器加載選中房屋的詳細數據，並跳轉到房屋詳情頁面。 * **清除瀏覽紀錄功能**：   + \_clearclickrecord：向伺服器發送清除點擊紀錄的請求，成功後清空本地的 \_clickHistory 列表。   **3. 修改按鈕點擊與調用**   * **收藏按鈕**：   + 點擊房屋卡片上的收藏按鈕調用 \_toggleFavorite 函數，切換收藏狀態。 * **清除紀錄按鈕**：   + 點擊 "清除" 按鈕會調用 \_clearclickrecord 函數，清除當前會員的所有瀏覽紀錄。   **4. 頁面佈局與構建**   * **整體佈局**：   + 使用 Scaffold 構建頁面，並且設置清除按鈕來刪除所有紀錄。 * **瀏覽紀錄列表顯示**：   + 使用 ListView.builder顯示瀏覽紀錄。正在加載，顯示 CircularProgressIndicator；若無紀錄，顯示 "尚無瀏覽紀錄" 提示。 * **房屋卡片**：   + 每條瀏覽紀錄顯示為一個 HouseCard，包含房屋數據和收藏功能，並可點擊進入房屋詳情頁面。 | | | |