行動應用程式開發書面報告

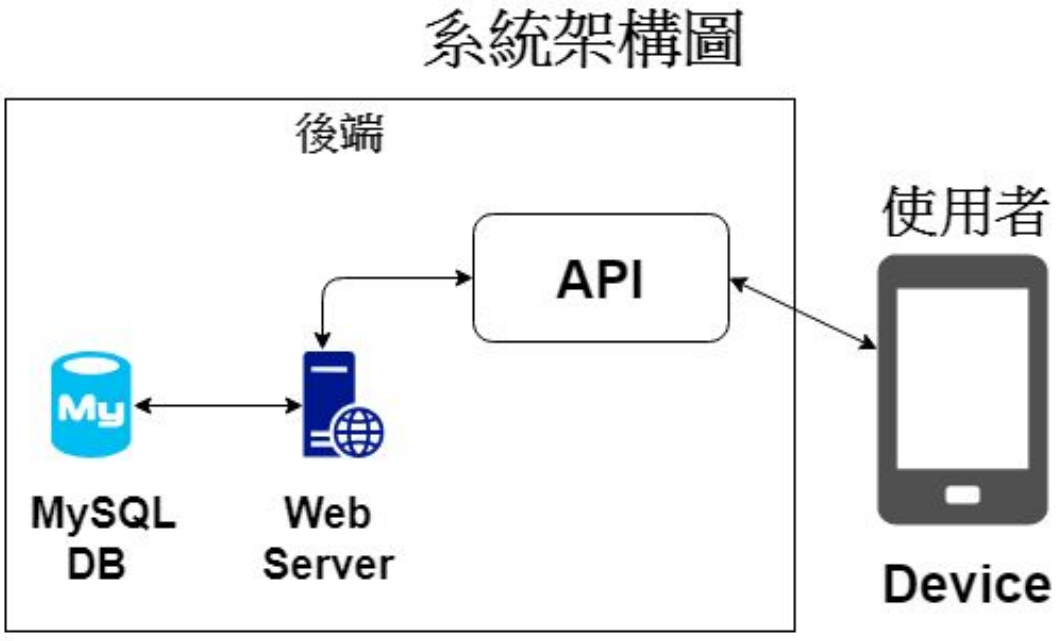
1. 系統簡介

本專題小組做的一個商業性質的行動應用程式--微銷售系統，客戶群定位是中小型企業，有一定的客戶量，但目前所應用的公開銷售平台（如：Facebook）已無法滿足他們的需求（如：買家瀏覽商品的歷史記錄、商品被瀏覽的次數等）。因此應用程式將具備了追踪客戶瀏覽商品的歷史記錄/過程、商品的點擊次數，以讓中小企業可以透過資料進行銷售分析，並針對不同的客戶推薦商品。

同時必定會提供基本功能（瀏覽商品，加入購物車，加入喜愛商品，訂購商品，瀏覽購物歷史，管理用戶資料等）

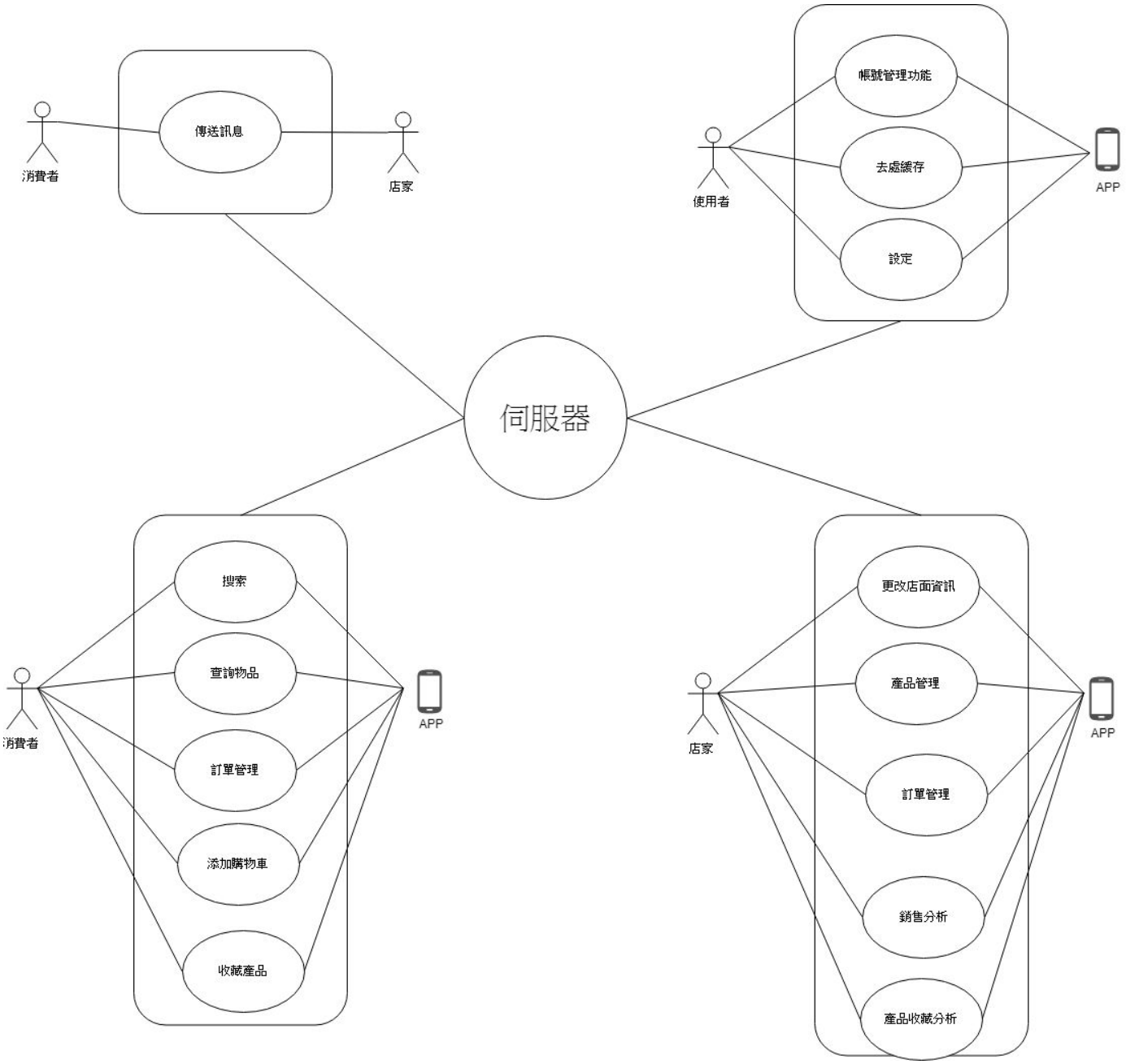
2. 系統開發

a. 系統架構圖

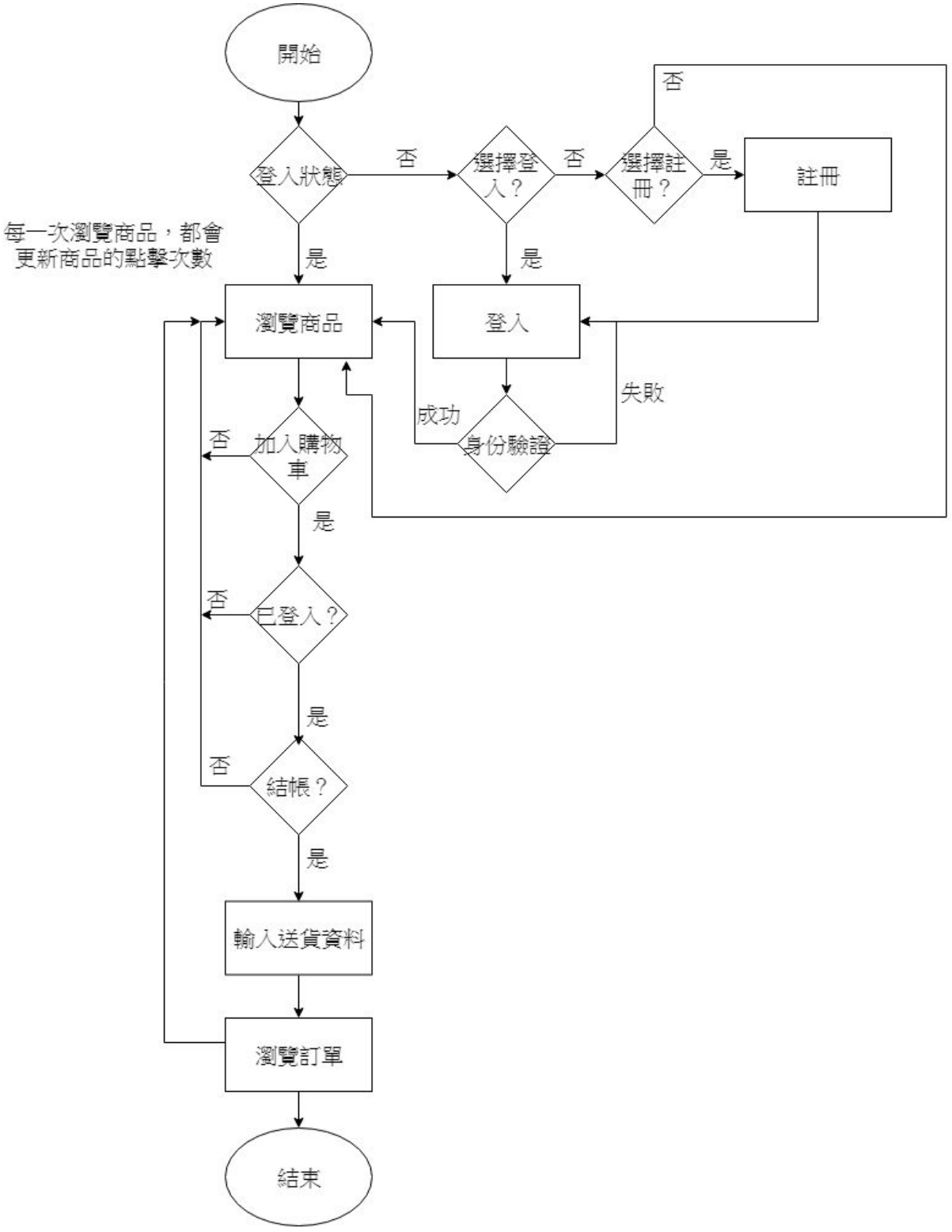


b. 使用案例圖

本系統目前的實作尚未完整，使用案例圖為先前之規劃。



c. 流程圖



d. 版本控管

本專案運用GitHub做版本控管，網址如下：
https://github.com/fcu-d0515211/Mobile-Application-Software

e. 程式模組說明

Abstract:

CustomActivity：於AppCompatActivity中加入靜態Toast方法，使每個於其上的Fragment使用。

Adapter:

BrandGridViewAdapter
CategoryAdapter
FavoriteAdapter
HistoryAdapter
ImagePagerAdapter
ItemGridViewAdapter
ProductListViewAdapter
RecycleAdapter
TypeGridViewAdapter
其皆為客製化的Adapter，因Android所提供的無法符合我們需求。

Model:

Order(Package)
Product(Package)
Users
Favorite
History
OrderCollection:
上述為自製Class並且運用建構子來存放從伺服器要求的資料。
DataManagement
將上述資料集合起來做成一個統一取值和改值的Class

Services:

Order(Package)
Product(Package)
FavoriteManagerment
HistoryManagerment
RequestManager
UserManagement
以上為和伺服器交互的Class

View:

Fragment(Package)
AccountManageActivity
CategoryActivity
FavoriteyActivity
HistoryActivity
LoginActivity
OrderDetailActivity
ProductCategoryActivity

ProductDetailActivity
SignupActivity
MainActivity:最底層的Activity，大多數Fragment建構於其上。

f. 程式碼解說(重點行數已標紅)

```
1.public static void requestProductBrand(final Context mCtx) {
2. final String url = host + "/product_brand";
3. StringRequest request = new StringRequest(
4.         url,
5.         new Response.Listener<String>() {
6.           @Override
7.           public void onResponse(String response) {
8.             Log.i(TAG, response);
9.             try {
10.              JSONArray array = new JSONArray(response);
11.              for (int i = 0; i < array.length(); i++) {
12.                JSONObject object = array.getJSONObject(i);
13.                int id = object.getInt("id");
14.                DataManagement.getProductBrands().add(
15.                        new ProductBrand(
16.                                id,
17.                                object.getInt("grade"),
18.                                object.getString("name"),
19.                                object.getString("created_at"),
20.                                object.getString("updated_at"),
21.                                BitmapFactory.decodeResource(mCtx.getResources()
,R.drawable.default_image)
22.                        ));
23.                MainShopFragment.reloadBrandGV();
24.                getImage(id, i);
25.              }
26.
27.            } catch (JSONException e) {
28.              e.printStackTrace();
29.            }
30.          }
31.        },
32.        new Response.ErrorListener() {
33.          @Override
34.          public void onErrorResponse(VolleyError error) {
35.                Log.i(TAG, responseerror);
36.          }
37.        }
38.);
39. MainActivity.volleyQueue.add(request);
40.}
```

伺服器交互部分程式碼（使用Volley）

本程式於第3行新增了字串請求，第7行為有正確響應時要的Listener，第32行為無正確響應時要的Listener，在正確響應時我們將資料整理成自製Class形態並運用

統一接口存入，並且將讀入的圖片加入至介面且重整介面（若不重整頁面會無法顯示於伺服器傳入的圖片），第39為以Queue的方式送出請求。

```
1. void changeView(ViewSwitch viewSwitch) {
2.   switch(viewSwitch) {
3.     case ListView:
4.       getSupportFragmentManager().beginTransaction()
5.               .hide(fragments[0])
6.               .show(fragments[1]).commit();
7.       break;
8.
9.     case GridView:
10.     getSupportFragmentManager().beginTransaction()
11.             .hide(fragments[1])
12.             .show(fragments[0]).commit();
13.      break;
14.  }
15.}
```

切換View程式碼

為了達到切換不同View的效果，我們運用了Switch讓使用者有切換介面的選擇，並且取得FragmentTransaction來控制Fragment的出現和隱藏，而無需切換Activity損耗手機的效能。

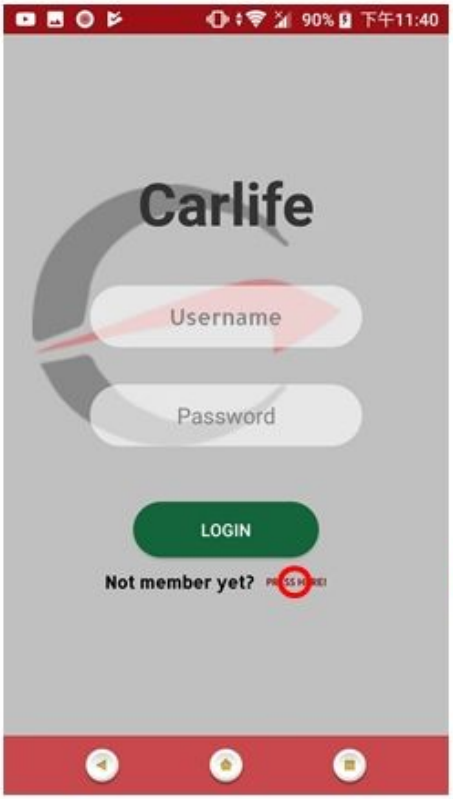
```
1.public class CustomAdapter extends FragmentStatePagerAdapter {
2.
3. public CustomAdapter(FragmentManager fm) {
4.   super(fm);
5. }
6.
7. @Override
8. public Fragment getItem(int position) {
9.   return fragments[position];
10.}
11.
12.@Override
13.public int getCount() {
14.  return 4;
15.}
16.
17.}
```

使用運用Page切換Fragment的Adapter

×××書面報告中的程式碼展示與最終Demo的成品不全然相同，×××
×××Demo前會再進行微調。 ×××

g. 操作介面

使用者登入頁面



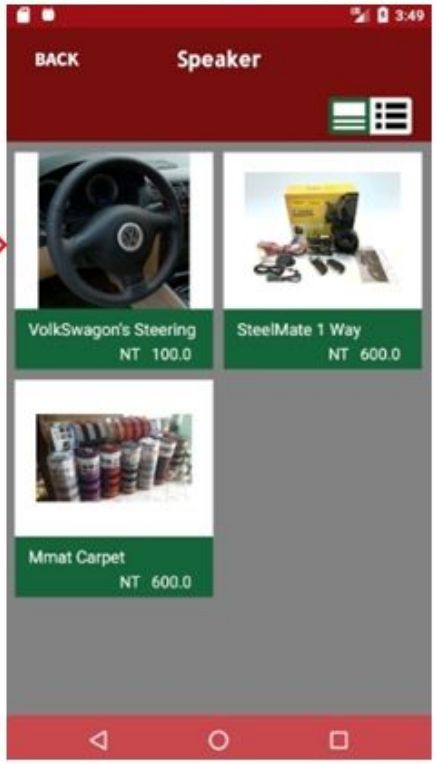
按了“PRESS HERE!”的紅色字體會帶到註冊頁面

使用者註冊頁面

深灰色區的Product Type和Car Brand會出現更多

同類商品頁面

點擊任一Product Type會帶到同類商品頁面



主頁面

購物車

訂單頁面

使用者管理頁面

3. 未來發展性

本專題小組對於未來此應用程式的期許：

a. 應用程式擁有人可在前端進行商品上架。
b. 新增切換身份功能，讓買家不只是買家而能可切換身份成為賣家，用賣家身份上架及管理商品。
c. 實踐付款行為，增加多元付款方式（Google Pay、信用卡、ATM…）。
d. 完善物流系統，串接各物流業者以提供買/賣家能夠隨時查看商品狀態。

4. 成品完成度（自評）

功能	進度
用戶登入	完成
用戶登出	完成
修改用戶資料	半完成
瀏覽商品	完成
搜索商品	待完成
加入購物車	完成
加入我的喜愛	完成
查詢歷史訂單	完成
下訂單	完成
更新商品點擊數量	完成
瀏覽商品銷售記錄	待完成
清除緩存	完成