

第五十屆全國技能競賽分區初賽 資訊技術（軟體設計）職類 競賽試題暨評分彙總表



競賽試題暨評分項目

<input type="checkbox"/> 系統分析與資料庫設計 SA&DB Design	<input type="checkbox"/> 商用軟體設計 Software Design
<input type="checkbox"/> 資料處理分析與簡報設計 DA&PPT Design	<input checked="" type="checkbox"/> 行動 APP 設計 App Design

選 手 編 號：_____

崗 位 編 號：_____

選 手 姓 名：_____

總 分：_____

評 分 老 師：_____

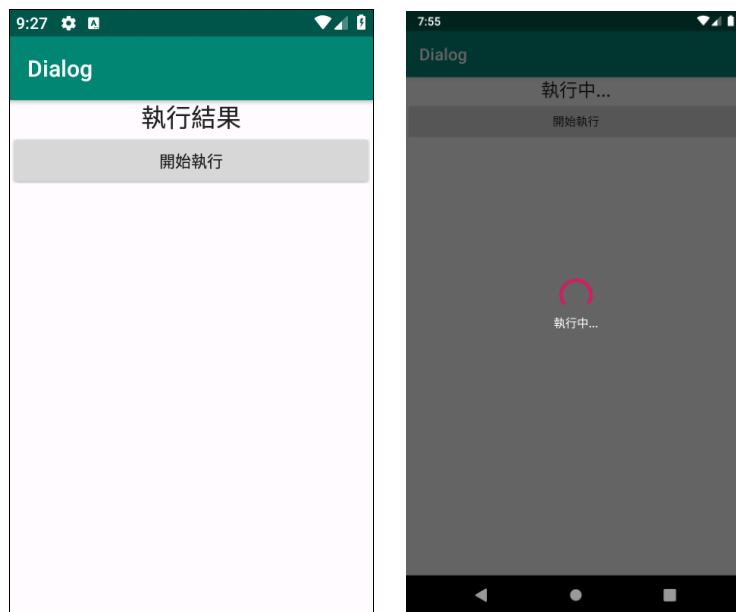
一、全螢幕對話方塊

Android Studio 開發環境如因版本不同，可能需要手動調整相關設定及參數，請自行調整後再接續作答。

1. 題目說明：

請開啟 **COM.TQC.GDD01** 專案，設計「全螢幕對話方塊」應用程式。由 **DialogFragment** 所組成，當按鈕按下後顯示，並利用執行緒暫停 2 秒後，取消此載入中的對話視窗，請依下列題意進行作答，作答完成請儲存所有檔案。

* 本題使用 Kotlin 程式語言。



<參考圖>

2. 設計說明：

- (1) 專案中已預先設計好 **main.xml** 與 **progress_dialog.xml** 兩個 Layout，其中 **main.xml** 為 **MainActivity** 的 Layout，**progress_dialog.xml** 則為 **MyProgressDialog** 的 Layout。開啟 **main.xml**，在 **root** 為 **LinearLayout** 的中間配置一個 **TextView** 與 **Button**。為 **TextView** 的 id 名稱命名為 **main_textView1**，**TextView** 之初始文字設定為在 **strings.xml** 中已定義好的字串，為 **Button** 的 id 命名為 **main_button1**，按鈕的初始文字使用設定在 **strings.xml** 中定義好的字串。
- (2) 開啟 **progress_dialog.xml**，使 Layout 的 **root** 為 **LinearLayout**，設定 id 為 **myProgressBarLinearLayout**，設定此 Layout 為垂直排列，並指定其寬高為 **match_parent**，與螢幕一樣寬一樣高(全螢幕)，並將 **ProgressBar** 置於其中，其樣式為圓圈型態；另外，加上一個 **TextView**，id 名稱為 **myProgressBarTextView**，此 **TextView** 文字放在 **ProgressBar** 下方，顯示文字內容為「執行中...」，文字顏色為白色。

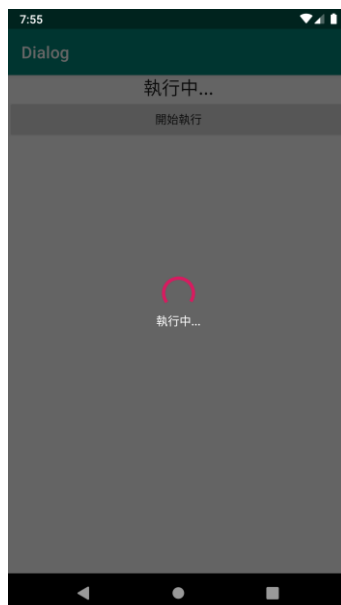
- (3) 開啟 `MyProgressDialog.kt`，於 `onCreateDialog` 中利用 `AlertDialog.Builder` 建立對話視窗物件，利用 `LayoutInflater` 將 `progress_dialog.xml` 轉為 `View` 物件，最後回傳 `AlertDialog.Builder` 的物件為 `Dialog`。於 `onStart()` 方法中，初始化 `mProgressBar` 使之不為 `null`。
- (4) 開啟 `MyProgressDialog.kt`，於 `showDialogFragmentAfterDelay` 中進行修改，利用 `Fragment` 的交易控制來顯示進度條對話框。
- (5) 開啟 `MainActivity.kt`，於 `init()` 方法中，加入當按鈕按下後，顯示的 `MyProgressDialog` 對話視窗的事件程式。

3. 執行結果參考畫面：

- (1) 程式起始時，畫面上方的 `TextView` 中顯示【執行結果】。



- (2) 按下「開始執行」按鈕，顯示一 `ProgressBar` 佔據全螢幕對話視窗，對話視窗為不可取消之對話視窗，顯示 2 秒後自動消失。



(3) 消失後的對話視窗，於上方 **TextView** 顯示【執行完畢】。



4. 自行測試程式是否達成下列結果：

- (1) 主畫面文字設定正確，其字串來自於已定義之字串。
- (2) 確認 `progress_dialog.xml` 之 `Layout` 的 `root` 為 `ProgressBar`。
- (3) 開啟 `MyProgressDialog.kt`，建立 `onCreateDialog` 中利用 `AlertDialog.Builder` 建立對話視窗物件 `progress_dialog.xml`，且使對話視窗為不可取消之對話視窗。
- (4) 開啟 `MyProgressDialog.kt`，於 `showDialogFragmentAfterDelay` 中進行修改，利用 `Fragment` 的交易控制來顯示進度條對話框，並在其下方顯示「執行中...」白色文字。
- (5) 開啟 `MainActivity.kt`，於 `init()`方法中，加入當按鈕按下後，顯示的 `MyProgressDialog` 對話視窗的事件程式。

5. 評分項目：

項	目	配 分	得 分
(1)	主畫面文字設定正確，其字串來自於已定義之字串	4	
(2)	確認 <code>progress_dialog.xml</code> 之 <code>Layout</code> 的 <code>root</code> 為 <code>ProgressBar</code>	7	
(3)	開啟 <code>MyProgressDialog.kt</code> ，建立 <code>onCreateDialog</code> 中利用 <code>AlertDialog.Builder</code> 建立對話視窗物件 <code>progress_dialog.xml</code> ，且使對話視窗為不可取消之對話視窗	5	
(4)	開啟 <code>MyProgressDialog.kt</code> ，於 <code>showDialogFragmentAfterDelay</code> 中進行修改，利用 <code>Fragment</code> 的交易控制來顯示進度條對話框，並在其下方顯示「執行中...」白色文字	7	
(5)	開啟 <code>MainActivity.kt</code> ，於 <code>init()</code> 方法中，加入當按鈕按下後，顯示的 <code>MyProgressDialog</code> 對話視窗的事件程式	7	
總 分		30	

二、餐廳管理

Android Studio 開發環境如因版本不同，可能需要手動調整相關設定及參數，請自行調整後再接續作答。

1. 題目說明：

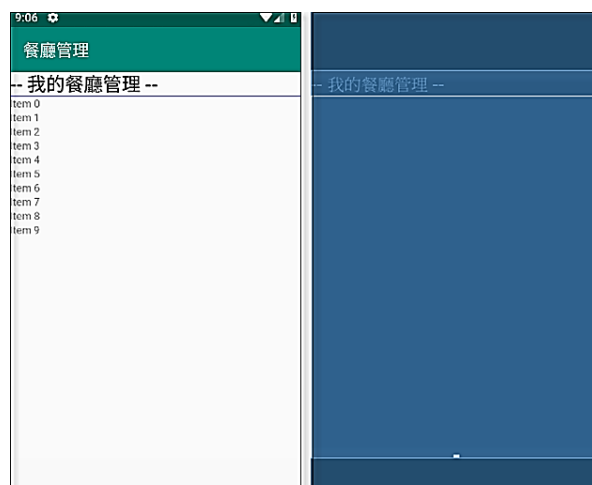
請開啟 **COM.TQC.GDD02** 專案，設計「餐廳管理」。利用 **Room** 資料庫，允許使用者利用 **ActionBar** 上的下拉選單中，新增餐廳資料。主畫面為 **RecyclerView** 來顯示，並允許往左右滑動之後直接刪除資料庫的該筆資料，請依下列題意進行作答，作答完成請儲存所有檔案。



<參考圖>

2. 設計說明：

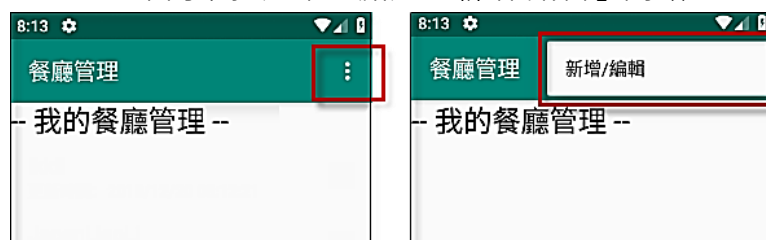
(1) UI 介面上方為 **TextView** 顯示【-- 我的餐廳管理 --】，下方為 **RecyclerView**。



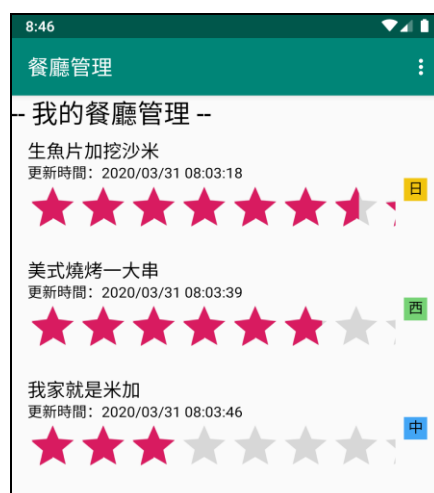
- (2) 開啟 MainActivity.java，修改 onOptionsItemSelected()中，當使用者點選 ActionBar 上方下拉選單的「新增/編輯」，前往 EditActivity.java 頁面編輯。接著，開啟 PlaceDataAccessObject.java，修改 PlaceDataAccessObject 介面中@Query 查詢、@Insert 插入、@Update 更新、@Delete 刪除等 Room 查詢敘述。
- (3) 開啟 DavidDataBase.java，於 synchronized (LOCK){}中呼叫 Room.databaseBuilder 建立資料庫名為 restaurant。
- (4) 修改 MainViewModel.java，新增 DavidDataBase 資料庫物件，呼叫 loadAllPlaces()方法，於主 UI 介面中載入 Room 資料庫所有資料，並利用 getPlaceList()回傳。開啟 MainActivity.java，在 getPlaces()方法中建立 MainViewModel 物件，並利用 MainViewModel 所提供的方法 getPlaceList()來取得 LiveData 物件參考，再以 observe(...)設定觀察者並覆寫相關方法 onChanged();並在 onChanged()方法內將最新的 Places 資料傳入 restaurantListAdapter，欄位包含餐廳名稱、更新時間、料理圖示以及評價星等。
- (5) 開啟編輯 EditActivity，修改 buttonAdd.setOnClickListener()事件，判斷為更新資料或是新增一筆資料，並實作 Room 資料庫的新增與更新。
- (6) 開啟編輯 MainActivity，於覆寫 recyclerView 的 ItemTouchHelper 中 onSwiped()方法，當使用者向左或向右滑動時，直接刪除該筆資料，當中刪除 Room 資料庫一筆的方法為 PlaceDataAccessObject 抽象類別中的 placeDao 方法。

3. 執行結果參考畫面：

- (1) 點開上方 ActionBar 中的下拉選單，點選「新增/編輯」開啟 EditActivity 編輯頁面。



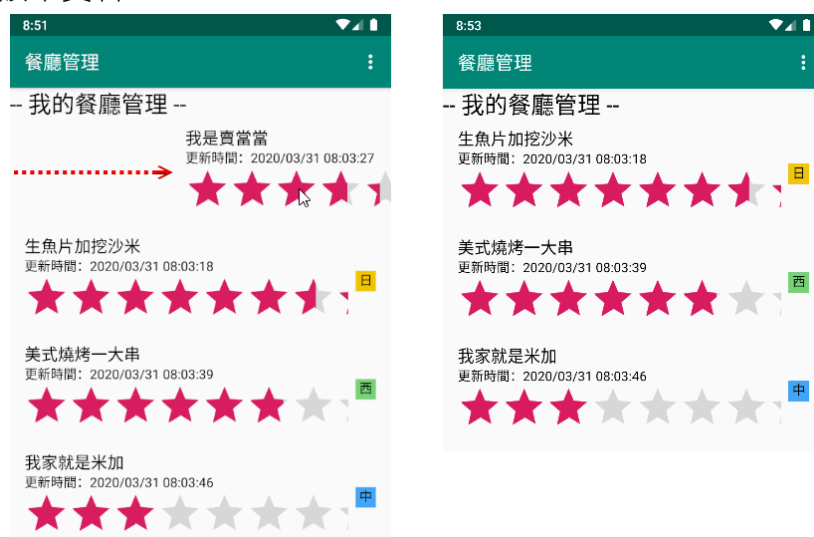
- (2) 輸入餐廳名稱、選擇餐廳型態，按下「新增」，則返回前主頁面 MainActivity，顯示於 RecyclerView 當中。



- (3) 於 MainActivity 主畫面，點擊餐廳名稱，會開啟 EditActivity，並將餐廳於 Room 資料庫的 id 帶入，此時 EditActivity 的按鈕為「更新」，修改餐廳名稱後，按下「更新」按鈕，立即更新 Room 資料庫。



- (4) MainActivity 中，對任何一筆餐廳資料，按住餐廳名稱後，左滑動或右滑動出螢幕時，直接刪除該筆資料。



4. 自行測試程式是否達成下列結果：

- (1) 點開上方 ActionBar 中的下拉選單，點選「新增/編輯」開啟 EditActivity 編輯頁面。
- (2) 輸入餐廳名稱、選擇餐廳型態、選擇 RatingBar 設定評價星等，按下「新增」，則返回前主頁面 MainActivity，顯示於 RecyclerView 當中。
- (3) 退出程式，再次開啟應用程式，主畫面 MainActivity 會載入先前紀錄下的餐廳資料，包含餐廳名稱、餐點型態縮圖以及資料更新時間。
- (4) 於 MainActivity 主畫面，點擊餐廳名稱，會開啟 EditActivity，並將餐廳於 Room 資料庫的 id 帶入，此時 EditActivity 的按鈕為「更新」，修改餐廳名稱後，按下「更新」按鈕，立即更新 Room 資料庫。
- (5) MainActivity 中，對任何一筆餐廳資料，按住餐廳名稱後，左滑動或右滑動出螢幕時，直接刪除該筆資料。

5. 評分項目：

項	目	配 分	得 分
(1)	點開上方 ActionBar 中的下拉選單，點選「新增/編輯」開啟 EditActivity 編輯頁面	4	
(2)	輸入餐廳名稱、選擇餐廳型態，按下「新增」，則返回前主頁面 MainActivity，顯示於 RecyclerView 當中，欄位包含餐廳名稱、更新時間、料理圖示以及評價星等。	7	
(3)	退出程式，再次開啟應用程式，主畫面 MainActivity 會載入先前紀錄下的餐廳資料，包含餐廳名稱、餐點型態縮圖以及資料更新時間	5	
(4)	於 MainActivity 主畫面，點擊餐廳名稱，會開啟 EditActivity，並將餐廳於 Room 資料庫的 id 帶入，此時 EditActivity 的按鈕為「更新」，修改餐廳名稱後，按下「更新」按鈕，立即更新 Room 資料庫	7	
(5)	MainActivity 中，對任何一筆餐廳資料，按住餐廳名稱後，左滑動或右滑動出螢幕時，直接刪除該筆資料	7	
總 分		30	

三、網路連線 API 與 JSON 解析

Android Studio 開發環境如因版本不同，可能需要手動調整相關設定及參數，請自行調整後再接續作答。

1. 題目說明：

請開啟 **COM.TQC.GDD03** 專案，設計一個可讀取三個 JSON 檔案的應用程式，將選擇的 json 解析後，顯示於 UI 的 ListView widget 當中，點選清單項目，以 Toast 顯示被選中項目的名稱及地址，ListView 呈現的欄位包含名稱與地址，請依下列題意進行作答，作答完成請儲存所有檔案。



<參考圖>

2. 設計說明：

- (1) 請開啟專案主程式，主程式 `onResume()` 時，註冊一 `BroadcastReceiver` (receiver 物件)，捕捉一個 `IntentFilter` 事件：
"android.net.conn.CONNECTIVITY_CHANGE"。當發生網路連線事件或網路連線改變事件時，被自訂的 `NetworkReceiver()` 接收廣播訊息。
- (2) 承設計說明(1)，離開程式呼叫 `onPause()`，反註冊 receiver 物件。
- (3) 修改主程式 `NetworkReceiver` 裡的 `onReceive()` 方法，此為當網路狀態變更時會接收的廣播訊息接收處，變更網路狀態改變時，開啟網路以 `Toast` 顯示【Wi-Fi 已連線】，關閉網路時，以 `Toast` 顯示【失去網路連線】。
- (4) 承設計說明(3)，關閉網路後 (當失去連線)，清除 `ListView` 內容。

- (5) 為配置好的 Spinner 下拉選單，指定三個項目，分別代表要存取的三個 API，三個項目名稱如下：

台北市運動中心	/assets/j01.json
YouBike 臺北市公共自行車即時資訊	/assets/j02.json
臺北市夜市專區攤商資訊	/assets/j03.json

- (6) 修改主程式 getAssetJsonData()方法，以處理三個 API 檔案之 JSON 格式，其 API 回傳內容皆已存入檔案中。以 Taipei Open Data 台北市運動中心 API 回傳的 JSON 字串為例，如下標記格式：

```
{
  "result": {
    "offset": 0,
    "limit": 10000,
    "count": 12,
    "sort": "",
    "results": [
      {
        "_id": "1",
        "url": "http://www.jjsports.com.tw",
        "zipcode": "10048",
        "name": "中正運動中心",
        "fax": "3322-5019",
        "Y": null,
        "X": null,
        "addr": "臺北市中正區信義路1段1號",
        "tel": "3322-5016"
      },
      {
        "_id": "2",
        "url": "http://whsc.cyc.org.tw",
        "zipcode": "10841",
        "name": "萬華運動中心",
        "fax": "2381-8667",
        "Y": null,
        "X": null,
        "addr": "臺北市萬華區西寧南路6之1號",
        "tel": "2381-8878"
      },
      {
        "_id": "3",
        "url": "http://www.taipeidt.com.tw",
        "zipcode": "10361",
        "name": "大同運動中心",
        "fax": "2572-0045",
        "Y": null,
        "X": null,
        "addr": "臺北市大龍街51號",
        "tel": "2592-0055"
      },
      {
        "_id": "4",
        "url": "http://www.btsport.com.tw/",
        "zipcode": "11271",
        "name": "北投運動中心",
        "fax": "2820-6070",
        "Y": null,
        "X": null,
        "addr": "臺北市北投區石牌路1段39巷100號",
        "tel": "2820-2880"
      },
      {
        "_id": "5",
        "url": "http://dasc.cyc.org.tw",
        "zipcode": "10671",
        "name": "大安運動中心",
        "fax": "2377-0331",
        "Y": null,
        "X": null,
        "addr": "臺北市辛亥路3段55號",
        "tel": "2377-0300"
      },
      {
        "_id": "6",
        "url": "http://cssc.cyc.org.tw",
        "zipcode": "10412",
        "name": "中山運動中心",
        "fax": "2560-4894",
        "Y": null,
        "X": null,
        "addr": "臺北市中山區中山北路2段44巷2號",
        "tel": "2581-1060"
      },
      {
        "_id": "7",
        "url": "http://www.nhsports.com.tw/",
        "zipcode": "11493",
        "name": "內湖運動中心",
        "fax": "2657-9782",
        "Y": null,
        "X": null,
        "addr": "臺北市內湖區洲子街12號",
        "tel": "2627-7776"
      },
      {
        "_id": "8",
        "url": "http://xyxc.cyc.org.tw",
        "zipcode": "11047",
        "name": "信義運動中心",
        "fax": "8786-1911",
        "Y": null,
        "X": null,
        "addr": "臺北市信義區松勤街100號",
        "tel": "8786-1711"
      },
      {
        "_id": "9",
        "url": "http://ngsc.cyc.org.tw",
        "zipcode": "11577",
        "name": "南港運動中心",
        "fax": "2653-8311",
        "Y": null,
        "X": null,
        "addr": "臺北市南港區玉成街69號",
        "tel": "2653-2279"
      },
      {
        "_id": "10",
        "url": "http://www.slsc-taipei.org/big5/",
        "zipcode": "11168",
        "name": "士林運動中心",
        "fax": "2880-5966",
        "Y": null,
        "X": null,
        "addr": "臺北市士林區士南路一號",
        "tel": "2880-6066"
      },
      {
        "_id": "11",
        "url": "https://wssc.cyc.org.tw/IIISystem/Portal/wssc/",
        "zipcode": "11698",
        "name": "文山運動中心",
        "fax": "2230-8676",
        "Y": null,
        "X": null,
        "addr": "臺北市文山區興隆路3段222號",
        "tel": "2230-8268"
      },
      {
        "_id": "12",
        "url": "http://www.sssc.com.tw",
        "zipcode": "10549",
        "name": "松山運動中心",
        "fax": "2579-"
      }
    ]
  }
}
```

- (7) 將上述 JSON 字串轉以 JSONObject 處理，解析傳回 JSON 內名稱 (欄位：name) 與地址 (欄位：addr) 後，顯示於 UI 中的 ListView widget 當中，條列清單內容如下示意：

```
--
中正運動中心
臺北市中正區信義路 1 段 1 號
--
萬華運動中心
臺北市萬華區西寧南路 6 之 1 號
--
大同運動中心
臺北市大龍街 51 號
```

- (8) **ListView** 呈現的欄位包含運動中心名稱與地址，點選運動中心項目，以 **Toast** 顯示被選中的運動中心名稱及地址。
- (9) 不同的 **API** 回傳不同的資料結果，需要成功解析並呈現在 **ListView** 中。如 **YouBike** 臺北市公共自行車即時資訊的欄位為 **sns** 與 **ar** 欄位，而臺北市夜市專區攤商資訊 **API** 則是取出夜市名稱與所在位置兩個欄位。
- (10) 按下「重設」按鈕，清除 **ListView** 與上方 **TextView** 顯示為【台北市運動中心】文字。
- (11) 於 **AndroidManifest.xml** 中新增上網權限，讓程式具有網際網路連線能力。

3. 執行結果參考畫面：

- (1) 執行程式後，開啟或關閉網路，網路連線時顯示【Wi-Fi 已連線】的 **Toast** 文字訊息，當網路斷線時，顯示【失去網路連線】的 **Toast** 文字訊息。



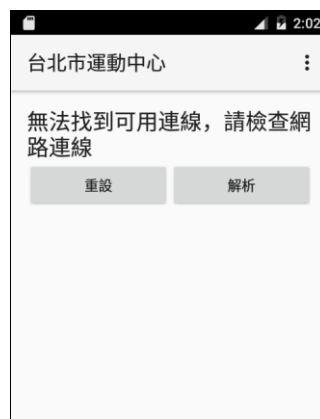
- (2) 點擊 **Spinner** 下拉選單，按下「解析」按鈕，判斷已連線，存取 **API** 將上方的 **TextView** 顯示【查詢 **API** 中，請稍候】變更為【解析成功】，並依據所選擇的 **Spinner** 項目更新下方 **ListView** 內容，顯示正確的名稱與地址。



(3) 點選 ListView 項目，以 Toast 顯示被選中的名稱及地址。



(4) 按下「解析」按鈕，若判斷無 Wi-Fi 或行動網路，則上方 TextView 顯示【無法找到可用連線，請檢查網路連線】，並同時清除下方的 ListView。



(5) 離開程式，變更模擬器網路，不再顯示 Toast 訊息。



4. 自行測試程式是否達成下列結果：

- (1) 執行程式後，開啟或關閉網路，網路連線時顯示【Wi-Fi 已連線】的 Toast 文字訊息，當網路斷線時，顯示【失去網路連線】的 Toast 文字訊息。
- (2) 按下「解析」按鈕，若判斷無 Wi-Fi 或行動網路，則上方 TextView 顯示【無法找到可用連線，請檢查網路連線】，並同時清除下方的 ListView。
- (3) 點擊 Spinner 下拉選單，顯示三個 API 的名稱：台北市運動中心、YouBike 臺北市公共自行車即時資訊、臺北市夜市專區攤商資訊。依據所點選名稱，按下「解析」按鈕，判斷已連線，存取 API 將上方的 TextView 顯示【查詢 API 中，請稍候】變更為【解析成功】。
- (4) 承(3)，按下「解析」按鈕，解析三種不同格式的 JSON 檔案，順利將正確的名稱與地址顯示於 ListView 當中。
- (5) 點選 ListView 清單項目，以 Toast 顯示被選中的名稱及地址。
- (6) 按下「重設」按鈕，會清空下方的 ListView，並以台北市運動中心為第一個預設的 API。
- (7) 離開程式，變更模擬器網路，不再顯示 Toast 訊息。

5. 評分項目：

項 目	配 分	得 分
(1) 執行程式後，開啟或關閉網路，網路連線時顯示【Wi-Fi 已連線】的 Toast 文字訊息，當網路斷線時，顯示【失去網路連線】的 Toast 文字訊息	5	
(2) 按下「解析」按鈕，若判斷無 Wi-Fi 或行動網路，則上方 TextView 顯示【無法找到可用連線，請檢查網路連線】，並同時清除下方的 ListView	5	
(3) 點擊 Spinner 下拉選單，顯示三個 API 的名稱：台北市運動中心、YouBike 臺北市公共自行車即時資訊、臺北市夜市專區攤商資訊。依據所點選名稱，按下「解析」按鈕，判斷已連線，存取 API 將上方的 TextView 顯示【查詢 API 中，請稍候】變更為【解析成功】。	4	
(4) 承(3)，選擇「台北市運動中心」，按下「解析」按鈕，順利解析 JSON 檔案，將正確的名稱與地址顯示於 ListView 當中	4	
(5) 承(3)，選擇「YouBike 臺北市公共自行車即時資訊」，按下「解析」按鈕，順利解析 JSON 檔案，將正確的名稱與地址顯示於 ListView 當中	4	
(6) 承(3)，選擇「臺北市夜市專區攤商資訊」，按下「解析」按鈕，順利解析 JSON 檔案，將正確的名稱與地址顯示於 ListView 當中	4	
(7) 點選 ListView 清單項目，以 Toast 顯示被選中的名稱及地址	5	
(8) 按下「重設」按鈕，會清空下方的 ListView，並以台北市運動中心為第一個預設的 API	4	
(9) 離開程式，變更模擬器網路，不再顯示 Toast 訊息	5	
總 分	40	