Lab4
Activity

本節目的：

- 了解什麼是 Activity，並產生一個 Activity。
- 透過 Intent 切換 Activity。
- 透過 Bundle 攜帶資料。
- 透過 onActivityResult()方法返回資料。

4.1 活動（Activity）

Android 應用程式元件包含 Activity、Service 與 BroadcastReceiver 這類的類別元件，而活動（Activity）是其中最基本的應用程式元件，每個 App 至少有一個 Activity，負責提供應用程式在顯示畫面上的相關工作，大部分的 APP 所顯示的畫面都是寫在 Activity 之上，不論是列表、圖片或是地圖的畫面，都是基於 Activity 來呈現。

如圖 4-1 的捷運地圖、景點收藏與更多畫面，就是基於 Activity 實現出來的。

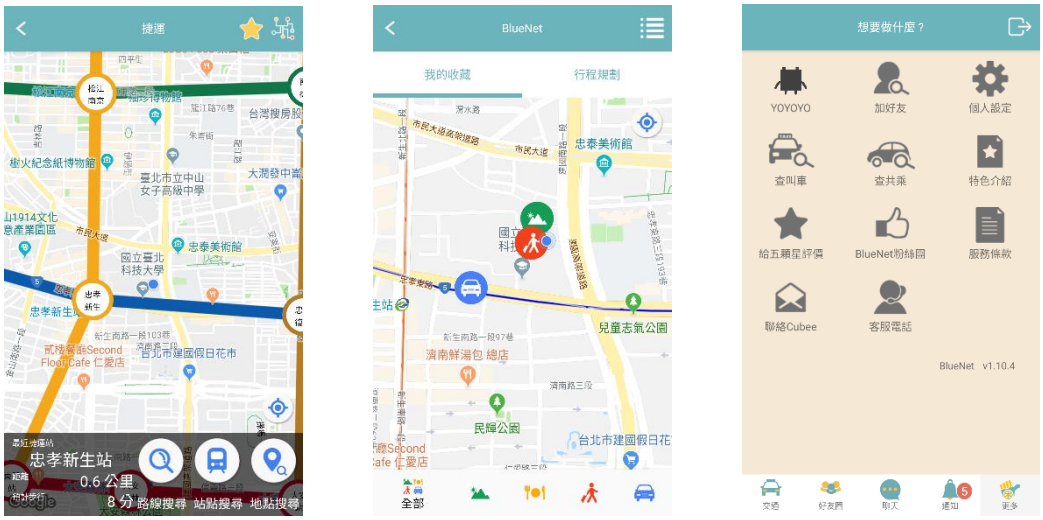

圖 4-1 捷運 Activity（左）、景點收藏（中）、更多 Activity（右）

要在螢幕上顯示畫面需要透過畫面配置元件（*.xml）與產生控制的應用程式元件（*.kt）。前面我們學到畫面配置元件即為 Layout，用於決定了每個元件的擺放位置，而 Activity 賦予這個畫面配置能與使用者互動的功能。

我們能透過 Activity 來顯示出某些資訊（圖片、文字或是地圖）給使用者，或是將使用者的操作傳送給程式來做控制(監聽器)，因此 Activity 扮演著 Android 使用者介面的角色。

4.1.1 產生 Activity

要產生出一個新的 Activity，首先選擇「File→New→Activity→Empty Activity」，來產生出空白的 Activity，如下圖 4-2 所示。

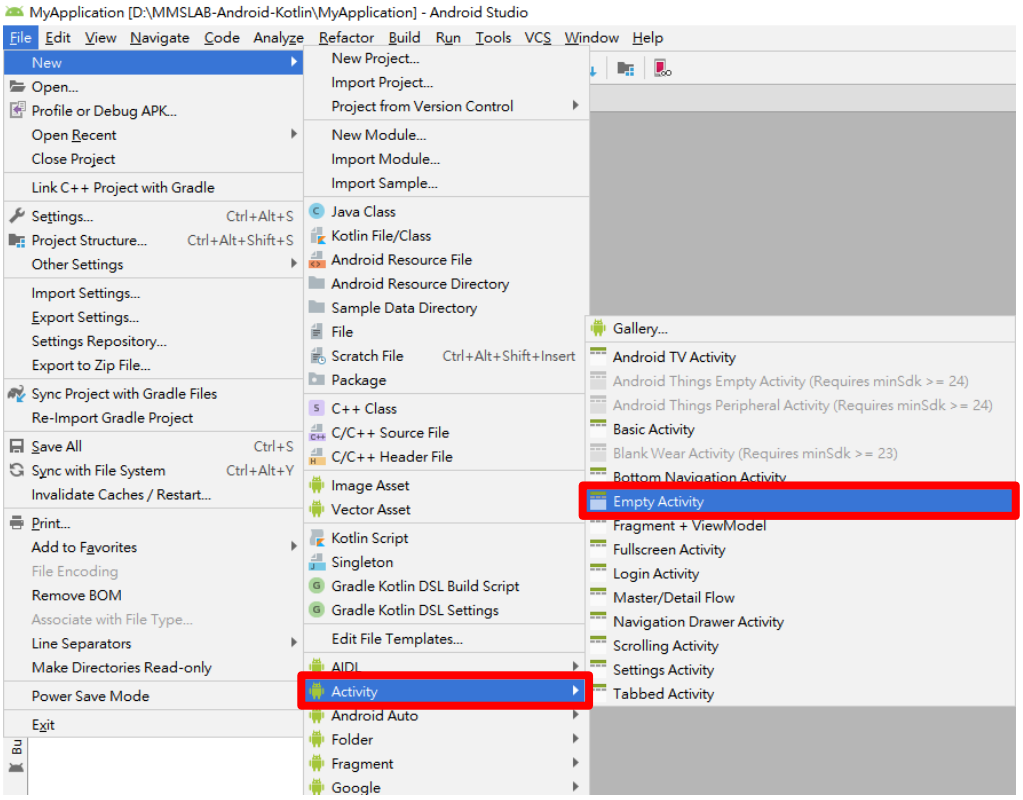

圖 4-2 點擊 File 建立新的 Activity

在視窗中修改 Activity 的名稱與對應 Layout 的名稱，並點選「Finish」按鈕，如圖 4-3 所示。如果只有更改 Activity 的名稱，Android 會自動幫你修改 Layout 的名稱。

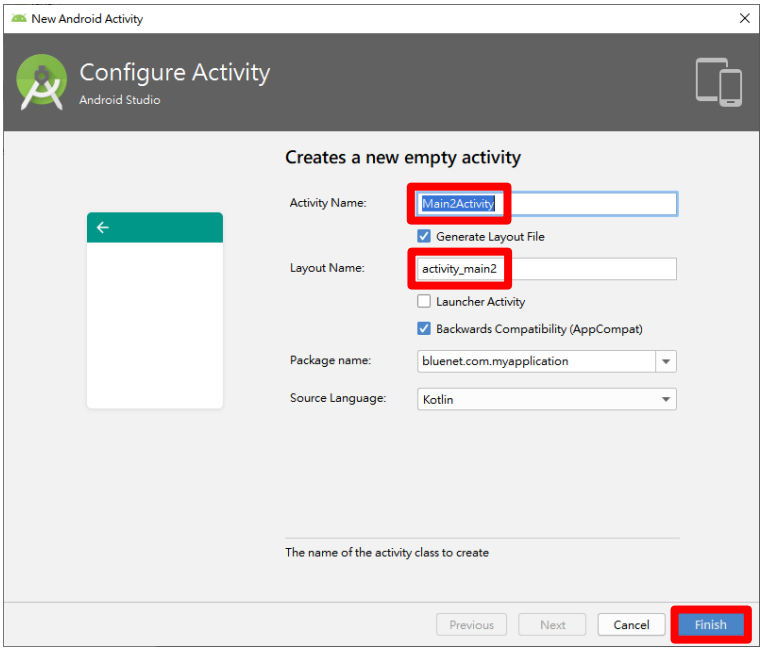

圖 4-3 設定 Activity 名稱與 Layout 名稱

點選「Finish」按鈕後，可以在目錄中看到系統幫你產生出 Main2Activity 以及 activity_main2.xml，如圖 4-4 所示。

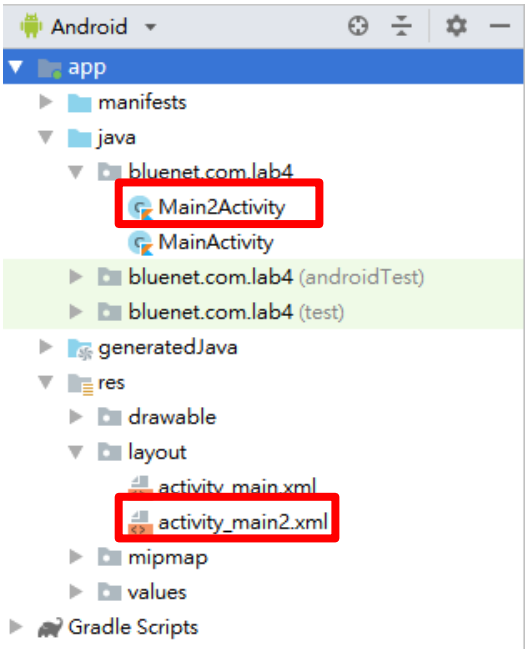

圖 4-4 目錄產生新的 Activity 與 Layout

而 AndroidManifest.xml 也會自動增加 Main2Activity 的資訊。

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="bluenet.com.myapplication">

    ...

                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity android:name=".Main2Activity"/>
    </application>
</manifest>
```

說明

觀察布局與畫面之間的關聯性，可以注意到元件的擺放是受到 layout 影響的。因此需要去理解 layout 的用法。

4.1.2 使用 Intent 切換 Activity

Android 應用程式元件（Activity、Service、BroadcastReceiver）之間的切換會需要透過「Intent」。Intent 是可用來向另一個應用程式元件（Activity、Service、BroadcastReceiver）要求動作的傳遞物件。最基本的 Intent 用途是來啟動其他的應用程式元件。如果啟動的對象是 Activity，則可以在畫面上顯示一個新的 Activity，可以說是 Activity 的切換動作。

Intent 字義上是指「意圖」，以 Activity 切換的目的上來解釋我們可以口語描述成「A 元件意圖啟動 B 元件」。下面圖 4-5 我們以 MainActivity 切換至 Main2Activity 為例，MainActivity 就表示 A 元件，Main2Activity 就表示 B 元件，兩者透過意圖傳遞把顯示畫面由 MainActivity 改為 Main2Activity。

圖 4-5 透過 Intent 從 MainActivity（左）切換到 Main2Activity（右）

從目前畫面（MainActivity）切換到 Main2Activity 的程式碼如下：

```
startActivity(Intent(this, Main2Activity::class.java))
```

Intent 有兩個參數，第一個參數我們要描述由哪個元件發起這個意圖，如從 MainActivity 發起則要填入 MainActivity 或是 this（表示 MainActivity 本身），第二個參數則要描述要接受意圖（被啟動）的對象是哪個元件，對象如果為 Main2Activity 則要描述成「Main2Activity::class.java」。

而要將這個 Intent 的發出，我們需要用到 startActivity()方法來送出 Intent，Main2Activity 獲得通知後便會被啟動，並覆蓋在 MainActivity 之上。

4.1.3 傳遞資料

Intent 除了可以做到基本的切換之外，Intent 也可夾帶一些資料到接收意圖方，例如某個使用者在 MainActivity 填寫了一個表單，並希望在 Main2Activity 看到結果。這時我們就要使用 Intent 傳遞資料的方法，以下則是最簡單的傳遞資料語法，主要是描述將 MainActivity 透過 Intent 切換到 Main2Activity 的意圖，透過 Intent 傳送 123 的整數資料，並從 MainActivity 切換到 Main2Activity。

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        //宣告意圖，透過 Intent 從 MainActivity 切換到 Main2Activity
        val intent = Intent(this, Main2Activity::class.java)
        //藉由 Intent 夾帶資料到新頁面
        intent.putExtra("key", 123)
        //開始動作
        startActivity(intent)
    }
}
```

intent.putExtra()可以想像是把想傳遞的資料（value）貼上一個標籤（key），接收的對象可以透過標籤去得到他要的資料。

接收到 intent 而被喚起的 Main2Activity 如果要取得傳過來的資料，可以用以下語法：

```
class Main2Activity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2)
        //判斷 Intent 不為空，並檢查是否夾帶資料
        intent?.extras? let {
            val data = it.getInt("key")
        }               //以 key 找到對應的資料並取出
    }
}
```

intent 可以取得從 MainActivity 傳過來的 Intent，而 **extras** 則可以取得底下夾帶的資料，我們可以使用 getInt(key)方法去找到你傳遞的資料，他的返回值就是 MainActivity 夾帶的資料。此外，由於傳遞的資料是 int 型態，因此使用 getInt()，如果是 float 型態則要用 getFloat()，以此類推。

然而 intent.putExtra()每一次傳遞資料我們都必須要知道傳遞的資料的型態是什麼，而且資料只能單獨的傳入、單獨的讀出，這樣資料的完整性並不高，有時候我們希望某些資料能被視為整體一次傳遞，例如一份菜單資料，我們不要每一項餐點資料都單獨傳過去，而是能以訂單為單位傳送。因此我們就會需要用到打包成包裹的概念，而在 android 中這就是 Bundle。

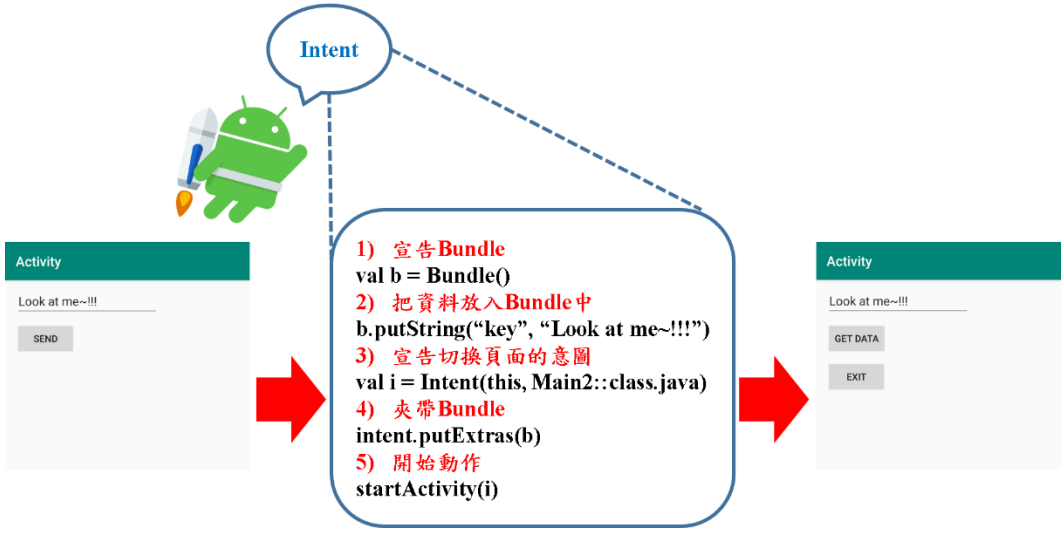

圖 4-6 MainActivity（左）藉由 Bundle 傳遞資料到 Main2Activity（右）

透過 Bundle 可以一次打包不同的資料型別，例如 Integer 或 String，如圖 4-6 所示。打包時，需要依據型態透過，putInt()或 putString()來儲存資料。

舉個例子，我們希望從 MainActivity 中夾帶一筆整數資料以及一筆字串資料到 Main2Activity 去，我們編寫的程式如下：

```
val bundle = Bundle() //宣告 Bundle
bundle.putInt("key1", 123) //把 123 放入 Bundle
bundle.putString("key2", "ABC") //把 “ABC” 放入 Bundle
val i = Intent(this, Main2Activity::class.java) //宣告意圖
i.putExtras(bundle) //夾帶 Bundle
startActivity(i) //開始動作
```

而新的 Activity 只需要取出 Bundle 就可以還原資料。

```
class Main2Activity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2)
        //判斷 Intent 不為空，並檢查是否夾帶資料
        intent?.extras?.let {
            //透過 key 從 Bundle 找到對應的 Data
            val value1 = it.getInt("key1")
            val value2 = it.getString("key2")
        }
    }
}
```

4.1.4 返回資料

透過 Intent 方法啟動的 Activity，除了之前介紹的 startActivity()方法之外，某些情況我們希望新的 Activity 在接收到資料後，能再夾帶資料返回到前一個 Activity，實現兩個 Activity 資料往來的目的，這時我們就會使用到 startActivityForResult()，實現的步驟流程如下圖 4-7：

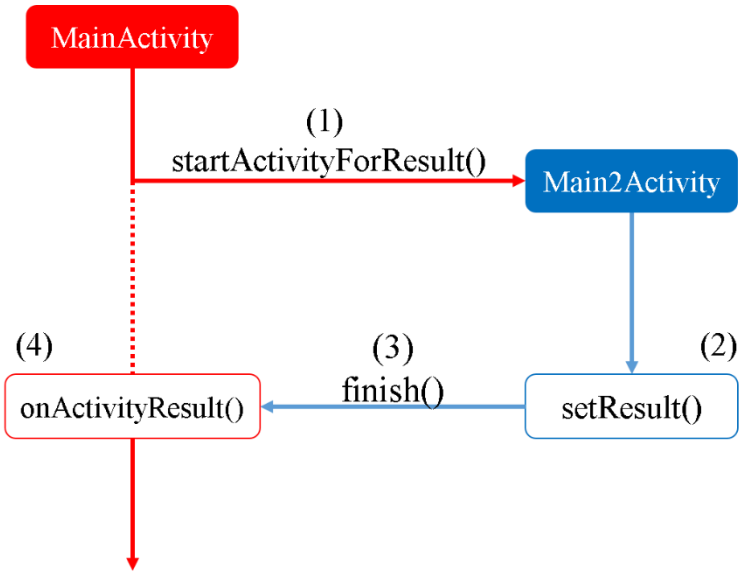

圖 4-7 Main2Activity 返回 MainActivity 傳遞資料流程

Step1 MainActivity 使用 startActivityForResult()方法，啟動 Main2Activity。

Step2 Main2Activity 使用 setResult()方法，儲存要返回的資料。

Step3 Main2Activity 使用 finish()方法結束 Main2Activity，並返回 MainActivity。

Step4 MainActivity 使用 onActivityResult()方法，取得返回資料。

此例中，我們要從 MainActivity 傳送夾帶一筆整數資料以及一筆字串資料到 Main2Activity 去，並且再接收 Main2Activity 回傳的資料。依據上述四個步驟編寫後的程式如下：

```
class MainActivity : AppCompatActivity() {
    //4) 使用 onActivityResult()方法，取得返回資料
    override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        //判斷 Intent 不為空，並檢查是否夾帶資料
        data?.extras?.let {
            //驗證發出對象
            if(requestCode==1 && resultCode== Activity.RESULT_OK){
                ... //取得返回資料
            }
        }
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        //宣告 Bundle
        val bundle = Bundle()
        //把 123 放入 Bundle
        bundle.putInt("key1", 123)
        //把 “ABC” 放入 Bundle
        bundle.putString("key2", "ABC")
        //宣告你的意圖
        val i = Intent(this, Main2Activity::class.java)
        //夾帶 Bundle
        i.putExtras(bundle)
        //1) 使用 startActivityForResult()方法，啟動 Main2Activity
        startActivityForResult(i, 1)
    }
}
```

- requestCode 的目的就是在於我們向新 Activity 提出要求並得到回覆時，用來判斷發出需求者（某個功能或頁面）為誰，以及要如何應對。requestCode 扮

演著一種需求者的編號，讓原 Activity 可以根據這編號來判斷發出需求的對象。

- onActivityResult()會等待新的 Activity 返回結果，我們可以根據傳過去的 requestCode 去驗證發出對象，之後透過 resultCode 確認在 Main2Activity 中執行的情況如何。

```
class Main2Activity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2)
        //判斷 Intent 不為空，並檢查是否夾帶資料
        intent?.extras?.let {
            //透過 key 從 Bundle 找到對應的 Data
            val value1 = it.getInt("key1")
            val value2 = it.getString("key2")
            //宣告 Bundle
            val bundle = Bundle()
            //把資料放入 Bundle
            bundle.putString("key", "value")
            //夾帶 Bundle
            val intent = Intent().putExtras(bundle)
            //2) 使用 setResult()方法，儲存要返回的資料
            setResult(Activity.RESULT_OK, intent)
            //3) 使用 finish()方法結束 Main2Activity，並返回 MainActivity
            finish()
        }
    }
}
```

- resultCode 可以用於回報執行結果給 MainActivity，例如成功時我們可以考慮用 resultCode=0 來表示，失敗時 resultCode=-1，用這方法來告知原 Activity 該如何處理。

4.2 點餐系統設計

- 本次範例實作一個點餐系統，透過設計兩個不同的 Activity，分別帶有不同的布局來完成點餐作業。
- MainActivity 按下選擇按鈕後會切換到 Main2Activity。
- Main2Activity 中可以輸入飲料與選擇甜度、冰塊，設定完點餐資訊後再回傳 MainActivity 點餐資訊。
- MainActivity 接收點餐資訊後可以顯示訂單資訊。

圖 4-8 MainActivity（左）接收 Main2Activity（右）傳遞的資料

4.2.1 點餐畫面設計

Step1 新建專案，以及圖 4-9 對應的 class 與 xml 檔：

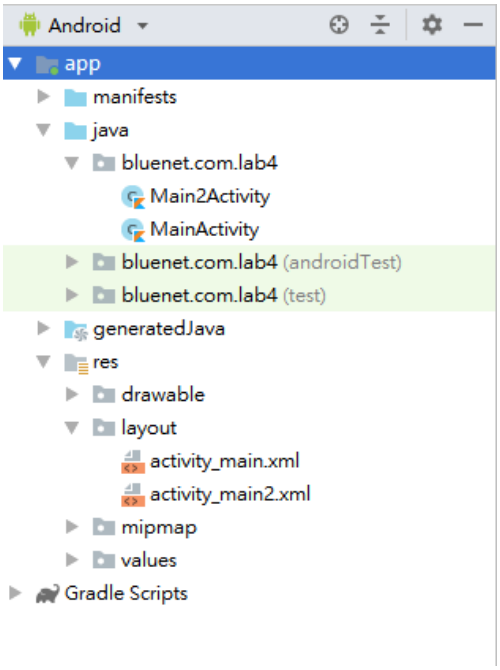

圖 4-9 點餐系統專案架構

Step2 繪製 activity_main.xml 檔，如圖 4-10 所示。

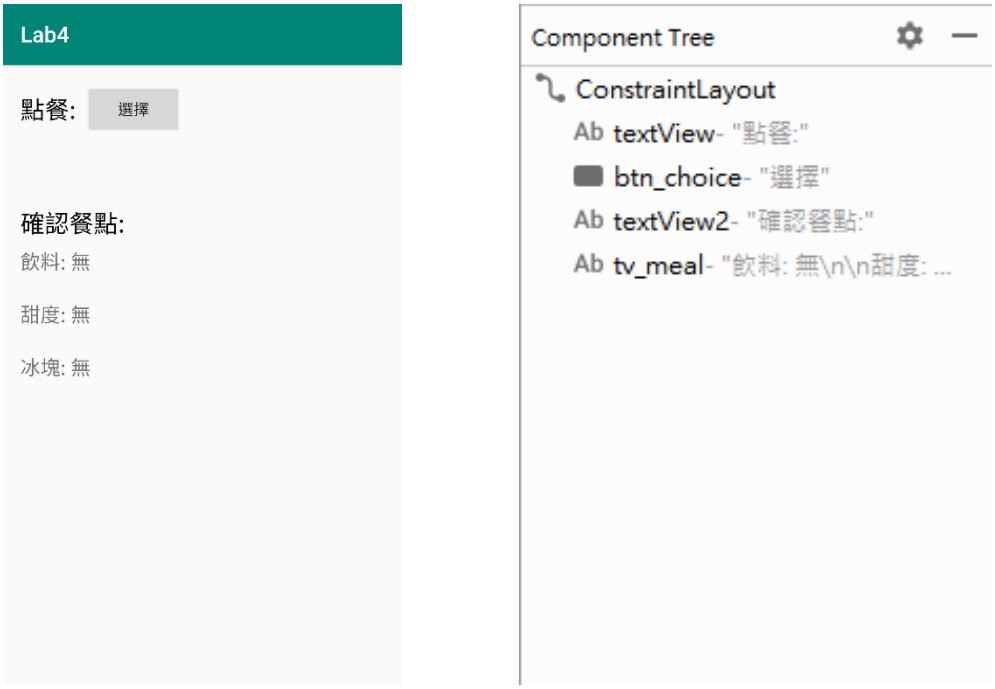

圖 4-10 點餐首頁預覽畫面（左）與布局元件樹（右）

對應的 xml 如下：

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="16dp"
        android:text="點餐:"
        android:textSize="22sp"
        android:textColor="@android:color/black"
        app:layout_constraintBottom_toBottomOf="@+id/btn_choice"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="@+id/btn_choice" />

    <Button
        android:id="@+id/btn_choice"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:text="選擇"
        app:layout_constraintStart_toEndOf="@+id/textView"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="64dp"
        android:text="確認餐點:"
        android:textSize="22sp"
        android:textColor="@android:color/black"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/btn_choice" />

    <TextView
        android:id="@+id/tv_meal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="飲料: 無\n\n 甜度: 無\n\n 冰塊: 無"
        android:textSize="18sp"
        app:layout_constraintStart_toStartOf="@+id/textView2"
        app:layout_constraintTop_toBottomOf="@+id/textView2" />
</android.support.constraint.ConstraintLayout>
```

Step3 繪製 activity_main2.xml 檔，如圖 4-11 所示。

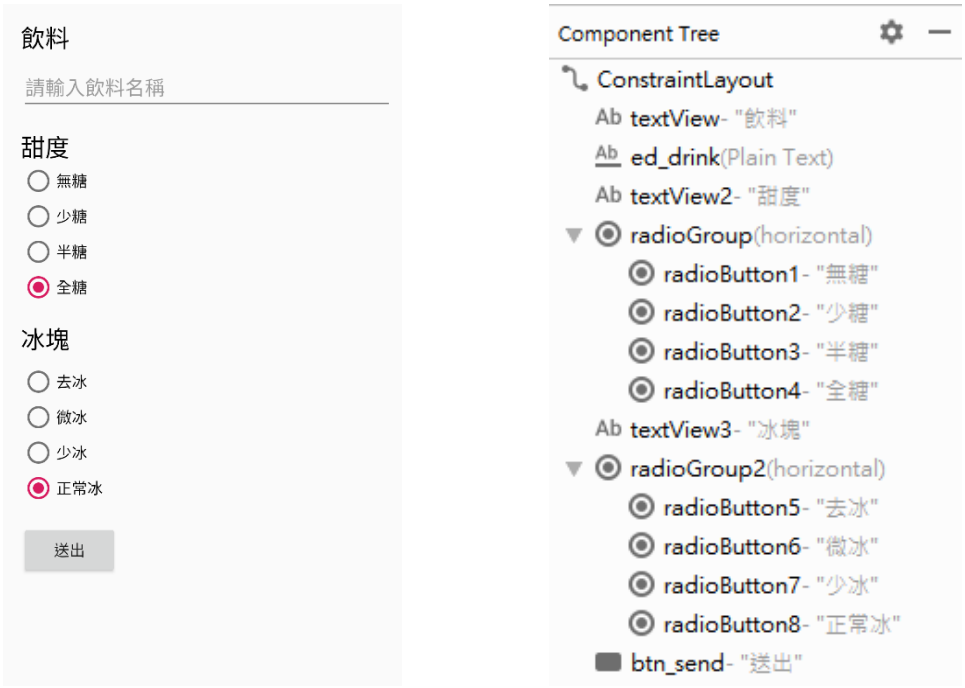

圖 4-11 點餐頁面預覽畫面（左）與點餐布局元件樹（右）

對應的 xml 如下：

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Main2Activity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:text="飲料"
        android:textSize="22sp"
        android:textColor="@android:color/black"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/ed_drink"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="請輸入飲料名稱"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="甜度"
        android:textSize="22sp"
        android:textColor="@android:color/black"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/ed_drink" />

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/textView2">

        <RadioButton
            android:id="@+id/radioButton1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
        android:layout_weight="1"
        android:text="無糖" />

    <RadioButton
        android:id="@+id/radioButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="少糖" />

    <RadioButton
        android:id="@+id/radioButton3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="半糖" />

    <RadioButton
        android:id="@+id/radioButton4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="全糖"
        android:checked="true"/>
</RadioGroup>

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="冰塊"
    android:textSize="22sp"
    android:textColor="@android:color/black"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toBottomOf="@+id/radioGroup" />

<RadioGroup
```

```
        android:id="@+id/radioGroup2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/textView3">

        <RadioButton
            android:id="@+id/radioButton5"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="去冰" />

        <RadioButton
            android:id="@+id/radioButton6"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="微冰" />

        <RadioButton
            android:id="@+id/radioButton7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="少冰" />

        <RadioButton
            android:id="@+id/radioButton8"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="正常冰"
            android:checked="true"/>
    </RadioGroup>

    <Button
```

```
        android:id="@+id/btn_send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="送出"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/radioGroup2" />
</android.support.constraint.ConstraintLayout>
```

4.2.2 按鈕監聽與資料傳遞

Step1 撰寫 MainActivity 程式，按下按鈕後切換至 Main2Activity。

```
package bluenet.com.lab4

import android.app.Activity
import android.content.Intent
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        btn_choice.setOnClickListener {
            startActivityForResult(Intent(this,Main2Activity::class.java),1)
        } //透過 Intent 切換至 Main2Activity 並傳遞 requestCode 來紀錄發出者
    }
}
```

Step2 建立 onActivityResult()接收返回資料後，將 data 的內容讀出以 TextView 作顯示。

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    data?.extras?.let {
        //驗證發出對象，確認 Main2Activity 執行的狀態
        if(requestCode==1 && resultCode== Activity.RESULT_OK){
            //讀取 Bundle 資料
            tv_meal.text = "飲料: ${it.getString("drink")}\n\n" +
                    "甜度: ${it.getString("sugar")}\n\n" +
                    "冰塊: ${it.getString("ice")}"
        }
    }
}
```

Step3 撰寫 Main2Activity 程式，設定 Button 監聽事件，判斷是否輸入飲料名稱，並且讀取 RadioGroup 數值。

```
package bluenet.com.lab4

import android.app.Activity
import android.content.Intent
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.RadioButton
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_main2.*

class Main2Activity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2)

        btn_send.setOnClickListener {
            if(ed_drink.length()<1)
                Toast.makeText(this, "請輸入飲料名稱",
                                        Toast.LENGTH_SHORT).show()
            else{
                //宣告 Bundle
                val b = Bundle()
                //取得 EditText 字串內容，把飲料名稱甜度與冰塊資訊放入 Bundle
                b.putString("drink", ed_drink.text.toString())
                b.putString("sugar", radioGroup.findViewById<RadioButton>
                          (radioGroup.checkedRadioButtonId).text.toString())
                b.putString("ice", radioGroup2.findViewById<RadioButton>
                     (radioGroup2.checkedRadioButtonId).text.toString())
                //用 Activity.RESULT_OK 標記執行狀態並記錄 Intent
                setResult(Activity.RESULT_OK, Intent().putExtras(b))
                finish()
            }
        }
    }
}
```