Sqlite登入程式

**Sqlite登入畫面設計**

1. 建立新專案，以及如圖1所示對應的class與xml檔。

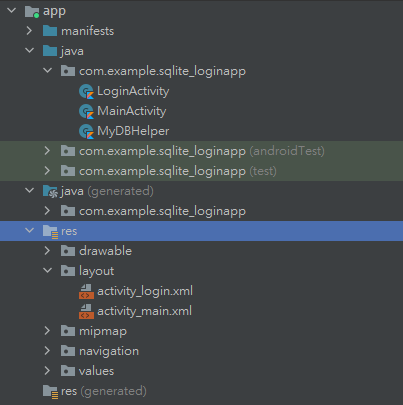


圖 1 專案架構

1. 繪製activity\_login.xml檔，如圖2所示

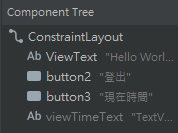
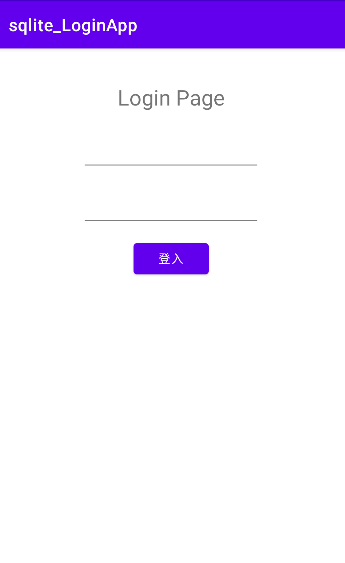


圖 2-1 登入預覽畫面(左) 與布局元件樹(右)

對應的 xml如下 :

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.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/ViewText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 android:textSize="25dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.187" />  
  
<Button  
 android:id="@+id/button2"  
 android:layout\_width="85dp"  
 android:layout\_height="45dp"  
 android:layout\_marginBottom="128dp"  
 android:text="登出"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 tools:ignore="TouchTargetSizeCheck" />  
  
<Button  
 android:id="@+id/button3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="現在時間"  
 app:layout\_constraintBottom\_toTopOf="@+id/button2"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/viewTimeText" />  
  
<TextView  
 android:id="@+id/viewTimeText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:textSize="20dp"  
 android:visibility="invisible"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/ViewText"  
 app:layout\_constraintVertical\_bias="0.215" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. 繪製activity\_main.xml檔，如圖3所示

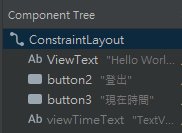
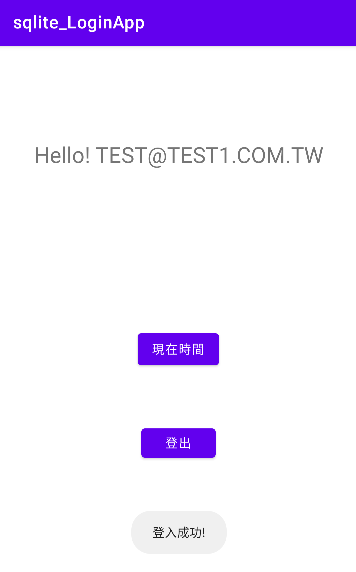


圖 3 登入成功畫面預覽(左) 與布局元件樹(右)

對應的 xml如下

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.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/ViewText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 android:textSize="25dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.187" />  
  
 <Button  
 android:id="@+id/button2"  
 android:layout\_width="85dp"  
 android:layout\_height="45dp"  
 android:layout\_marginBottom="128dp"  
 android:text="登出"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 tools:ignore="TouchTargetSizeCheck" />

<Button  
 android:id="@+id/button3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="現在時間"  
 app:layout\_constraintBottom\_toTopOf="@+id/button2"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/viewTimeText" />  
  
 <TextView  
 android:id="@+id/viewTimeText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:textSize="20dp"

//預設隱藏  
 android:visibility="invisible"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/ViewText"  
 app:layout\_constraintVertical\_bias="0.215" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. 變更AndroidManifest.xml檔，讓LoginActivity登入進AndroidManifest.xml，並(登入頁面)處於預設啟動的Activity 。

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.sqlite\_loginapp">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Sqlite\_LoginApp">  
 <!--註解 :  
 將LoginActivity 設為主要的Activity，並註冊AndroidManifest.xml  
 -->  
 <activity  
 android:name=".LoginActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <!--註解 :  
 MainActivity 改為非主要Activity  
 -->  
 <activity android:name=".MainActivity" />  
 </application>  
  
</manifest>

SQL存取資料庫

1. 撰寫MyDBHelper需要建立USERS資料表，並設立2組帳號密碼資料。

import android.content.Context  
import android.database.sqlite.SQLiteDatabase  
import android.database.sqlite.SQLiteOpenHelper  
//自訂建構子，並通過SQLiteOpenHelper類別來對SQLite資料庫進行操作  
class MyDBHelper (context: Context) : SQLiteOpenHelper(context,"USERDB",null,1){  
 override fun onCreate(db: SQLiteDatabase?) {  
 //建立資料表USERS，並包含一個帳號、一個密碼的字串欄位  
 db?.execSQL("CREATE TABLE USERS(USERID INTEGER PRIMARY KEY AUTOINCREMENT, UNAME TEXT, PWD TEXT)")  
 //新增2筆帳號密碼  
 db?.execSQL("INSERT INTO USERS(UNAME,PWD) VALUES('TEST@TEST1.COM.TW', '123')")  
 db?.execSQL("INSERT INTO USERS(UNAME,PWD) VALUES('123@456.789','132465')")  
 }  
  
 override fun onUpgrade(p0: SQLiteDatabase?, p1: Int, p2: Int) {  
  
 }  
}

1. 撰寫LoginActivity，建立MyDBHelper實體，並通過readableDatabase來讀取資料

class LoginActivity : AppCompatActivity() {  
 //延後建立物件  
 lateinit var button : Button  
 lateinit var username: EditText  
 lateinit var password: EditText  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 //綁定物件實體  
 setContentView(R.layout.*activity\_login*)  
 username = findViewById(R.id.*editTextTextEmailAddress3*)  
 password = findViewById(R.id.*editTextTextPassword4*)  
 button = findViewById(R.id.*button*)  
 //建立MyDBHelper實體  
 var helper = MyDBHelper(*applicationContext*)  
 var db = helper.*readableDatabase* //按鈕觸發事件  
 button.setOnClickListener **{** //取得登入者的帳號字串  
 var unameText = username.*text*.toString()  
 //將輸入的帳號密碼與資料庫聯繫，確認是否有這筆資料  
 var arge = *listOf*<String>(username.*text*.toString(), password.*text*.toString()).*toTypedArray*()  
 var rs = db.rawQuery("SELECT \* FROM USERS WHERE UNAME = ? AND PWD = ?",arge)

if (rs.moveToNext()) {  
 //建立 intent，設定啟動的Activity ，並建立key為 Text 來儲存帳號字串  
 val intent = Intent(this, MainActivity::class.*java*)  
 intent.putExtra("Text",unameText)  
 //重製密碼輸入  
 password.setText("")  
 //發送Toast，登入成功  
 Toast.makeText(*applicationContext*, "登入成功!",Toast.*LENGTH\_LONG*).show()  
 //起動 MainActivity  
 startActivity(intent)  
} else{  
 //發送登入失敗訊息  
 Toast.makeText(*applicationContext*, "帳號或密碼錯誤",Toast.*LENGTH\_LONG*).show()  
}

1. 撰寫MainActivity，並顯示使用者帳號名子，建立現在時間查詢按鈕與登出按鈕

import android.os.Bundle  
import android.view.View  
import android.widget.Button  
import android.widget.TextView  
import androidx.appcompat.app.AppCompatActivity  
import java.time.ZoneId  
import java.time.ZonedDateTime  
  
  
class MainActivity : AppCompatActivity() {  
 //延後建立物件  
 lateinit var UnameText: TextView  
 lateinit var viewTimeText: TextView  
 lateinit var backButton : Button  
 lateinit var timeButton : Button

override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
 //讀取傳送資料，並使用key定位  
 val textE = *intent*.getStringExtra("Text")  
 //綁定物件實體  
 UnameText = findViewById(R.id.*ViewText*)  
 viewTimeText = findViewById(R.id.*viewTimeText*)  
 backButton = findViewById(R.id.*button2*)  
 timeButton = findViewById(R.id.*button3*)  
 //將使用者帳號顯示出來  
 UnameText.*text* = "Hello! "+textE  
  
 //建立按鈕事件  
 backButton.setOnClickListener **{** //登出  
 finish()  
 **}** timeButton.setOnClickListener **{** //取得現在時間  
 getTime()  
 //取消 viewTimeText 內容隱藏，並顯示出現在時間  
 viewTimeText.*visibility* = View.*VISIBLE* **}**}

private fun getTime(){  
 //此處ZoneId.of 、 ZonedDateTime.now 需要調整API級別到26，但不影響使用  
  
 //設定時區  
 val londonZone = ZoneId.of("GMT+8")  
 //取得現在時間(預設GMT+0)，並帶入londonZone 設定時區  
 val londonCurrentDateTime = ZonedDateTime.now(londonZone)  
 viewTimeText.*text* = londonCurrentDateTime.toString()  
}