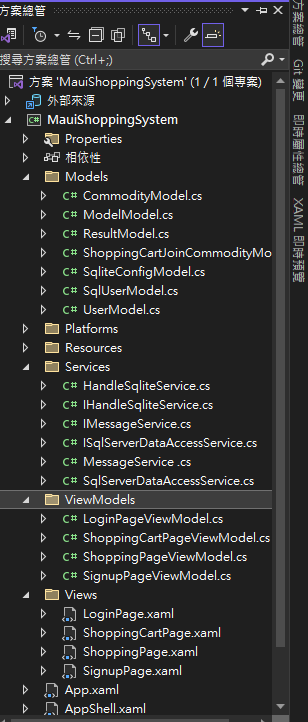
架構介紹  
這次使用了mvvm架構，一個view(介面)就對應一個viewModel，每個View使用一個ViewModel去處理，然後也建立了Service層，分別去處理SQLite、MSSQL、Msg，在viewModel即可使用依賴注入方式去調用這些方法



介面流程介紹  
在開啟應用程式進入首頁前：  
1-1.檢查Sqlite查看UserId的欄位是否有值  
1-2.沒有值時會到Login頁面，否則會到Shopping頁面  
(＊＊正確應該是要存token的，但目前還沒研究出怎麼取出token，所以先以UserId來代替)

Login頁面：  
1.登入btn  
1-1.輸入的帳號密碼會從MSSQL去查詢是否正確  
1-2.不正確則彈出錯誤訊息，正確則將UserId存入SQLite，並且跳轉至Shopping頁面

2.註冊btn  
2-1.跳轉至Signup頁面

Signup頁面:  
1.確認btn  
1-1.至MSSQL檢查UserId是否重複，重複則彈出錯誤訊息  
1-2.檢查姓名、帳號、密碼不可為空，其一為空則彈出錯誤訊息  
1-3.若無前兩點錯誤，則會彈出"註冊成功"訊息，則表示使用者資料已寫入MSSQL，並且跳轉至Shopping頁面

2.取消btn  
2-1.跳轉至Login頁面

Shopping頁面:  
1.初始動作  
1-1.至MSSQL將商品表裡數據顯示出畫面

2.加入購物車(未完成，目前無法取得商品列點選該欄位的值)  
2-1.點選任一商品，取得該商品值  
2-2.點選加入購物車btn  
2-3.將該新商品存入購物車表裡count欄位為1，舊商品則count欄位+1，並且記錄UserId

3.登出  
3-1.至SQLite清空UserId欄位  
3-2.跳轉至Login頁面

4.查看購物車btn  
4-1.跳轉至ShoppingCart頁面

ShoppingCart頁面  
1.初始動作(目前數據都是手動Key進去的)  
1-1.至MSSQL針對UserId及status = 'open' 將購物車表裡數據顯示出畫面清單

2.增加(未完成，目前無法取得清單點選該欄位的值)  
2-1.點選任一商品，取得該商品值  
2-2.點選增加btn  
2-3.MSSQL該筆數據count+1

3.減少(未完成，目前無法取得清單點選該欄位的值)  
3-1.點選任一商品，取得該商品值  
3-2.點選增加btn  
3-3.MSSQL該筆數據count-1

4.返回btn  
4-1.跳轉至Shopping頁面

5.下單btn(未完成，目前無法取得商品列點選該欄位的值)  
5-1.將畫面清單數據送出，並將MSSQL status改為close表示商品購入不需在購物車顯示