軟體生命週期：

流程順序要依照軟體生命週期

 **需求收集與分析（Requirements Gathering and Analysis）**：

* 在這個階段，開發團隊與利益相關者（客戶、用戶、管理層等）交流，理解他們的需求和期望。這些需求會被文檔化，成為開發的基礎。
* 目標：確定並詳細描述系統功能需求、性能需求等。

 **系統設計（System Design）**：

* 在這個階段，根據需求分析的結果，設計系統的架構、數據結構、界面、用戶交互等。設計可以是高層次的（如架構設計）或者低層次的（如數據庫設計）。
* 目標：定義如何實現需求，選擇合適的技術和工具。

 **開發（Development）**：

* 開發階段是編碼的過程，根據設計文檔編寫實際的代碼，實現功能。
* 目標：將設計轉換為可執行的軟件，並編寫高質量的代碼。

 **測試（Testing）**：

* 在這個階段，對軟件進行全面測試，檢查是否符合需求，是否有錯誤和漏洞，並進行修復。測試可以是單元測試、集成測試、系統測試等。
* 目標：確保軟件的質量，發現並修復 bug。

 **部署（Deployment）**：

* 部署是將已經開發和測試好的軟件應用程序部署到生產環境或客戶使用的環境中。
* 目標：將軟件交付給用戶，並確保系統運行穩定。

 **維護（Maintenance）**：

* 一旦軟件交付使用，進入維護階段。這包括修復運行中的錯誤、更新功能以適應用戶需求的變化，並對軟件進行優化和改進。
* 目標：確保系統穩定運行，並根據需要進行升級。

前後端處理：

如果有安全性不高的驗證可以在前端先驗證就不需要移到後端

盡量把資源留給後端

版本控制：

sourcetree;tortle;

SQL Injection（注入攻擊）：

資料表規範

每一個細項都需要規定清楚

Login登入按鈕的細項：

### 事件啟動順序的原因和控制：

* **Login\_LoggingIn**：此事件在登錄過程開始前觸發，是用來檢查登錄邏輯是否能夠繼續進行的地方。例如，你可以在這裡檢查用戶是否被禁用、是否達到登錄嘗試上限等。如果這裡設定了 e.Cancel = true，登錄過程就會被停止，後續的身份驗證不會進行。
* **Login\_Authenticate**：此事件是最核心的身份驗證邏輯所在。如果 Login\_LoggingIn 沒有取消登錄流程，那麼會進行用戶名和密碼的驗證，並根據結果設置 e.Authenticated。如果身份驗證成功，會繼續觸發 Login\_LoggedIn 事件，否則會顯示錯誤或處理登錄失敗。
* **Login\_LoggedIn**：這是登錄過程的“成功”階段，通常在這裡做一些如 Session 設置、登錄日誌記錄或用戶重定向等操作。這個事件只有在身份驗證成功後觸發。
* **Login\_Error**：這個事件並不是每次都會觸發，它主要是用來處理登錄過程中的異常。比如，如果在 Login\_Authenticate 或 Login\_LoggingIn 中發生了錯誤，Login\_Error 會捕獲到，並讓你可以處理錯誤或提供錯誤訊息。