PC網路聊天室(TCP/IP)

## 第一章 緒論

## 1.1研究背景與動機

隨著資訊科技的飛速發展與網際網路的普及，即時通訊（Instant Messaging, IM）已成為現代社會不可或缺的溝通方式。從個人社交、團隊協作到客戶服務，即時通訊軟體憑藉其高效、便捷的特性，深刻地改變了人們的交流習慣和工作模式。PC平台的網路聊天室作為即時通訊的一種早期且經典的形式，其底層的網路通訊協定和程式設計邏輯，對於理解更複雜的分散式系統和網路應用具有重要的學習價值。

本研究動機是對於對網路通訊技術， TCP/IP協定在實際應用中的運作方式。TCP/IP作為網際網路的基石，支援了絕大多數的網路資料傳輸。透過親手設計並實作一個基於C#語言和Windows Forms介面的PC網路聊天室，能夠更深入理解客戶端/伺服器（Client/Server）架構、多執行緒處理等核心技術，同時將理論知識與實際操作相結合，提升軟體開發的實踐能力。

C#語言以其強大的.NET框架支援、物件導向特性以及在Windows平台開發上的便利性，成為了實現此類桌面應用程式的理想選擇。Windows Forms則提供了一套成熟的圖形化使用者介面（GUI）開發工具，有助於快速建構互動友善的聊天室介面。因此，本專題旨在透過C#和Windows Forms技術，結合TCP/IP協定，建構一個功能性的PC網路聊天室，藉此深化對相關技術的理解與應用。

## 1.2研究目的

本研究旨在設計與實作一個基於TCP/IP協定的PC網路聊天室。

具體研究目的如下：

1. 掌握TCP/IP通訊原理：深入理解TCP協定的連線導向、可靠傳輸特性，進行網路資料的傳送與接收。
2. 實現客戶端/伺服器架構：設計並開發聊天室的伺服器端應用程式，負責管理客戶端連線、訊息轉發與使用者狀態維護；同時開發客戶端應用程式，提供使用者登入、訊息收發與顯示等功能。
3. 應用多執行緒技術：在伺服器端採用多執行緒或非同步處理機制，以有效處理多個客戶端的併發連線與請求，確保聊天室的即時性與穩定性。
4. 開發圖形化使用者介面：利用C# Windows Forms技術，為客戶端設計一個直觀、易用的圖形化使用者介面，提升使用者體驗。
5. 實現基本聊天功能：包括但不限於使用者登入/登出、使用者列表顯示等核心聊天功能。

## 1.3問題陳述

在開發PC網路聊天室的過程中，預期將面臨以下主要技術問題與挑戰：

1. 可靠的連線建立與維護：如何在客戶端與伺服器之間建立穩定可靠的TCP連線，並在網路不穩定或意外斷線時進行有效處理？
2. 併發使用者處理：伺服器端如何高效地處理來自多個客戶端的同時連線請求和訊息傳輸，避免阻塞和資源競爭？
3. 訊息同步與即時性：如何確保訊息能夠即時、準確地從傳送端傳遞到所有指定的接收端，並在客戶端介面即時更新？
4. 資料格式與序列化：如何定義客戶端與伺服器之間交換訊息的資料格式，並進行有效的序列化與反序列化，以確保資料的正確解析？
5. 使用者介面回應與互動：如何在進行網路通訊的同時，保持客戶端GUI的流暢回應，避免因網路操作阻塞UI執行緒？
6. 錯誤處理與異常管理：如何設計健壯的錯誤處理機制，以應對網路錯誤、資料格式錯誤、使用者操作不當等各種預期及非預期的異常情況？

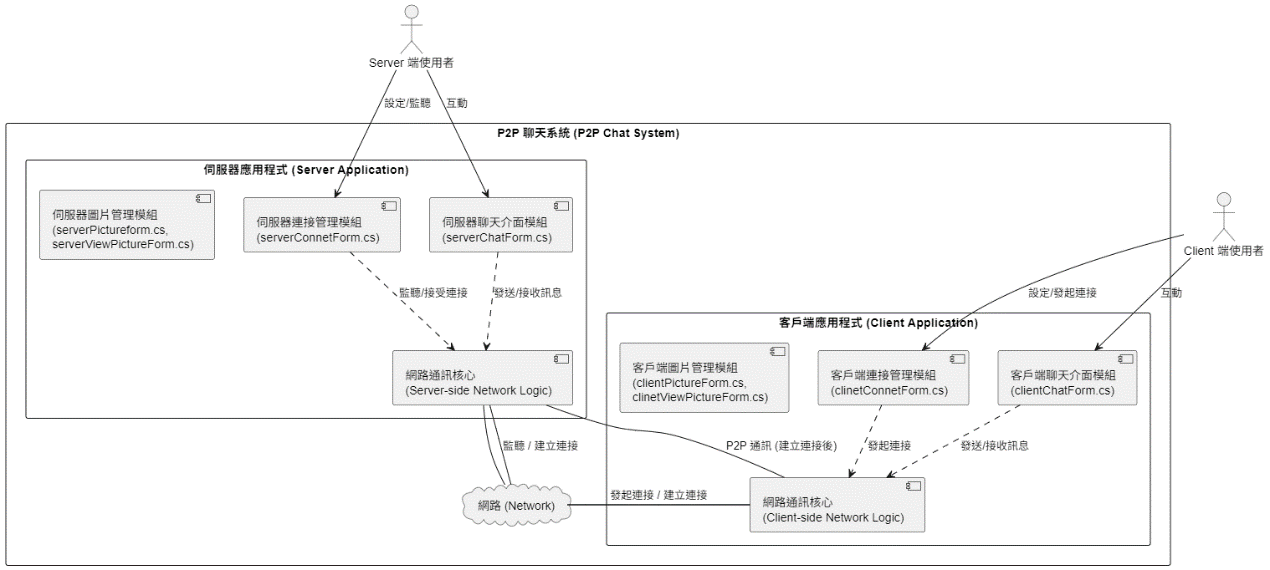
## 1.4預期貢獻

本研究的完成，預期將達成以下幾方面的貢獻：

1. 技術實踐能力的提升：對於研究者本身，將顯著提升在C#程式設計、Windows Forms介面開發、TCP/IP網路編程以及多執行緒應用等方面的綜合實踐能力。
2. 一個功能性的聊天室應用程式：產出一個可在PC環境下運行的、具備基本聊天功能的網路聊天室原型系統，可作為學習和演示TCP/IP通訊的實例。
3. 網路編程知識的深化：透過專案的完整開發週期，加深對客戶端/伺服器模型、資料封包處理等網路核心概念的理解。
4. 解決方案的探索與驗證：針對在開發過程中遇到的技術挑戰（如併發處理、UI回應等），所採用的解決方案可為類似專案提供參考。

# 第二章 系統設計

## 2.1系統架構圖



### 2.1.1架構概述

本系統由兩個獨立但協同運作的應用程式組成：

1. Server 應用程式：負責監聽指定連接埠，接受來自 Client 應用程式的連接請求，並在連接建立後作為通訊的另一端。
2. Client 應用程式：負責發起連接到 Server 應用程式，並在連接建立後作為通訊的另一端。

兩者在建立連接後，均包含使用者介面、應用邏輯和網路通訊層，以實現聊天室的核心功能。

## 2.2功能模組說明

本系統的功能模組可分為兩大類：

兩端共通的模組，以及 Server 端和 Client 端各自特有的模組。

### A.通用功能模組:

這些模組存在於 Server 和 Client 兩端應用程式中，提供核心的聊天室功能。

1. 聊天介面模組 (serverChatForm.cs, clientChatForm.cs)：

職責：提供使用者進行文字聊天和顯示聊天歷史的介面。

功能：顯示即時聊天訊息、提供訊息輸入框、顯示參與者列表（如果適用）、處理訊息的顯示格式。

1. 圖片顯示/管理模組 (serverPictureform.cs, serverViewPictureForm.cs, clientPictureForm.cs, clinetViewPictureForm.cs)：

職責：處理圖片的選取、在介面中顯示，以及提供圖片預覽功能。

功能：允許使用者選擇本機圖片、在聊天介面中顯示圖片縮圖、在新視窗中顯示圖片原圖（ViewPictureForm）、處理圖片的載入與傳輸。

1. 訊息處理模組 (nternal Logic)：

職責：處理聊天訊息的結構化、序列化/反序列化，以及在應用邏輯層的處理。

功能：建立標準化的訊息物件（例如，包含發送者、內容、時間戳記、訊息類型），處理訊息的編碼與解碼，以供網路傳輸。

### B. Client 特定功能模組：

1. 連接管理模組 (clinetConnetForm.cs)：

職責：提供使用者發起連接到 Server 端應用程式的介面與邏輯。

功能：讓使用者輸入 Server 的 IP 位址和連接埠、發起 TCP/IP 連接、顯示連接狀態、處理連接錯誤。

1. 客戶端網路通訊模組 (Internal Logic)：

職責：處理 Client 端應用程式所有的網路數據發送與接收。

功能：建立與 Server 的 Socket 連接、發送序列化的訊息資料、接收來自 Server 的訊息資料、處理網路異常斷開。

### C. Server 特定功能模組 (Server-Specific Functional Modules):

1. 連接監聽模組 (serverConnetForm.cs)：

職責：提供使用者設定監聽連接埠，並啟動監聽以接受 Client 連接的介面與邏輯。

功能：允許使用者指定監聽的 IP 位址和連接埠、啟動/停止監聽服務、顯示已連接的 Client 數量、處理新連接的接受。

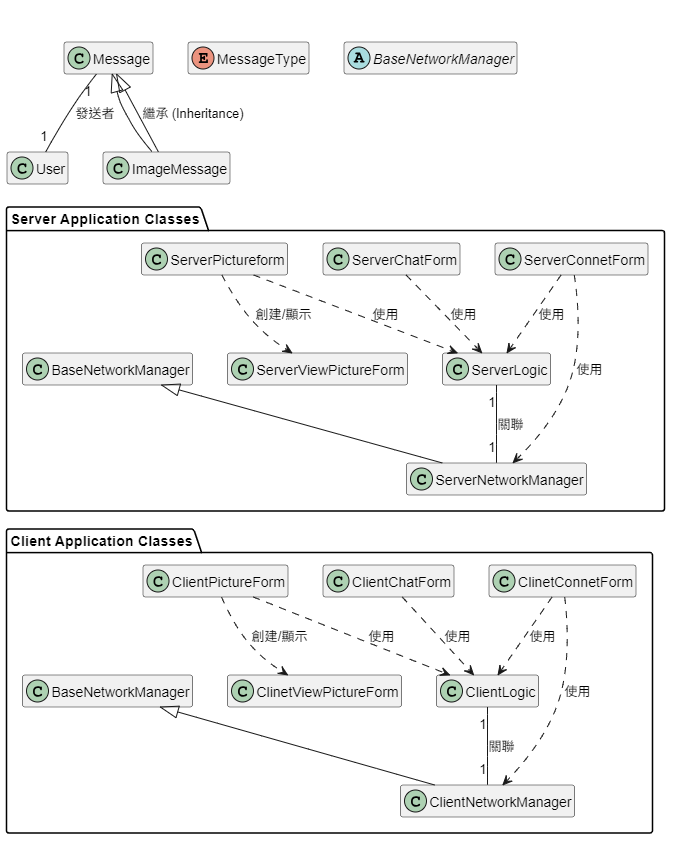
1. 伺服器網路通訊模組 (Internal Logic)：

職責：處理 Server 端應用程式所有的網路數據監聽、連接管理、數據發送與接收。

功能：建立監聽 Socket、接受新的 Client 連接、管理多個已連接的 Client Socket、發送數據給特定 Client、接收來自 Client 的數據。

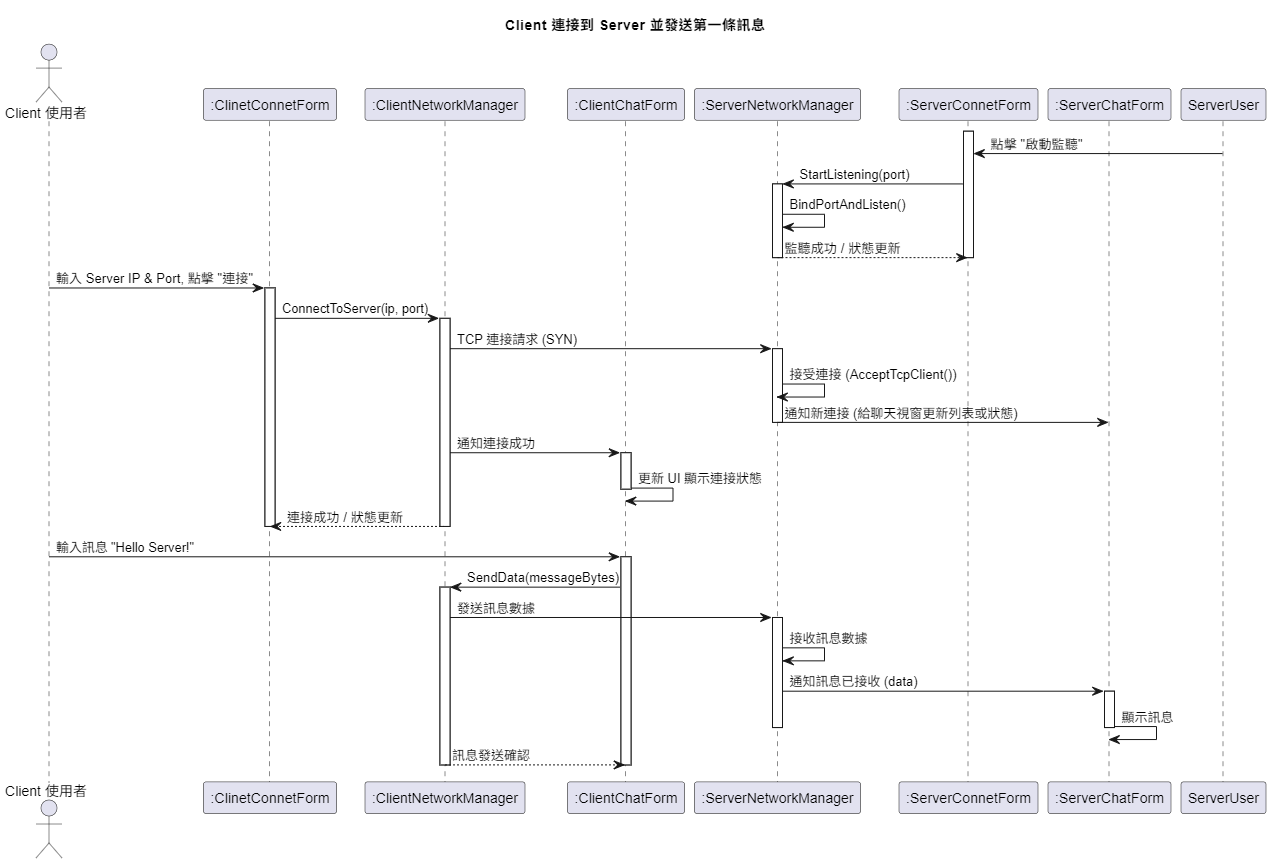
## 2.3類別圖、循序圖等 UML 圖表

### 2.3.1類別圖:



### 2.3.2循序圖:

場景：Client 連接到 Server 並發送第一條訊息



# 第三章 系統實現

## 3.1開發環境（Visual Studio 版本、.NET 版本等）

Microsoft Visual Studio Community 2022 (64 位元) - Current版本 17.12.4

Microsoft .NET Framework版本4.8.09037

## 3.2核心功能代碼解析

## .3.3關鍵技術實現說明

### 3.3.1界面設計與使用者操作流程

# 第四章 測試與驗證

## 4.1單元測試案例

## 4.2系統整合測試

## 4.3效能評估（如有）

# 第五章 結論與未來工作

## 5.1研究成果總結

## 5.2專題限制

## 5.3未來改進方向

# 附錄. 參考文獻

引用書籍、論文、網路資源

[**.NET Framework API 參考 version 4.8**](https://learn.microsoft.com/zh-tw/dotnet/api/?view=netframework-4.8&preserve-view=true)

### 撰寫建議

- \*\*字數分配\*\*：假設報告約 1000-2500 字，每

個大綱約佔 200-500 字，根據需求調整。