

Computer Network HW 1

SOCKET PROGRAMMING

MIS 2 袁子晉

104403520

Overview about the Software

倉儲管理程式

在一間營運的企業體中，總是有很多的企業資源需要管理，如果分配得宜，將可以為公司省下許多成本。

而透過這個軟體，你將可以簡單增減各項目的物品個數。

自由上傳、下載最新資料

透過我的軟體，你將可以利用Socket Programming的概念，將資料透過Socket自由上傳至伺服器或是將資料從伺服器同步



開始囉!

接下來將會介紹程式主題、語言、支援系統、以及如操作這隻程式

1.

程式主題 & 操作流程

一窺程式本體

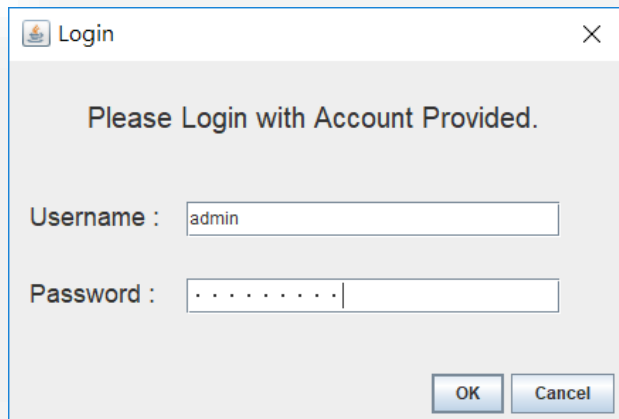


由 Java 撰寫的程式

- ▶ 支持多作業系統，將可在 Mac OSX, Ubuntu等常見的視窗作業系統上使用。

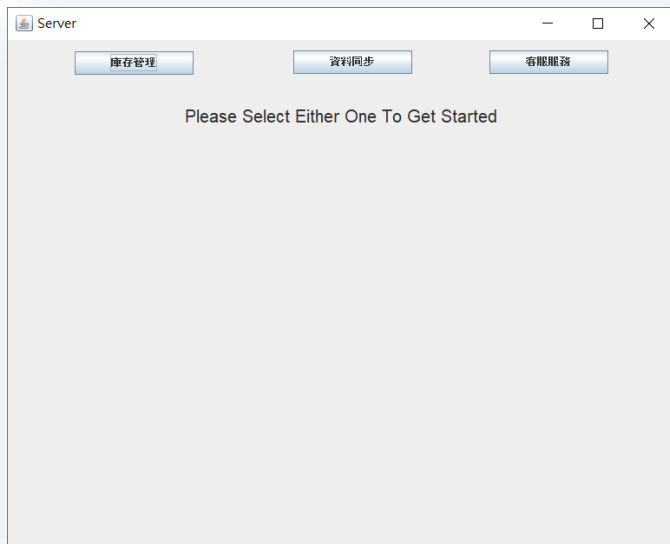
程式分為兩部分 – 以 Server 端為例

▶ 登入畫面



A Windows-style dialog box titled "Login" with a close button (X) in the top right corner. The text "Please Login with Account Provided." is centered. Below it, there are two input fields: "Username :" with the text "admin" entered, and "Password :" with masked characters ".....". At the bottom right, there are two buttons: "OK" and "Cancel".

▶ 程式主畫面



A Windows-style application window titled "Server" with standard minimize, maximize, and close buttons in the title bar. At the top, there are three buttons: "庫存管理", "資料同步", and "容能服務". Below these buttons, the text "Please Select Either One To Get Started" is displayed.



Login as Admin or User

- ▶ 開啟Server，代表這是伺服器端。

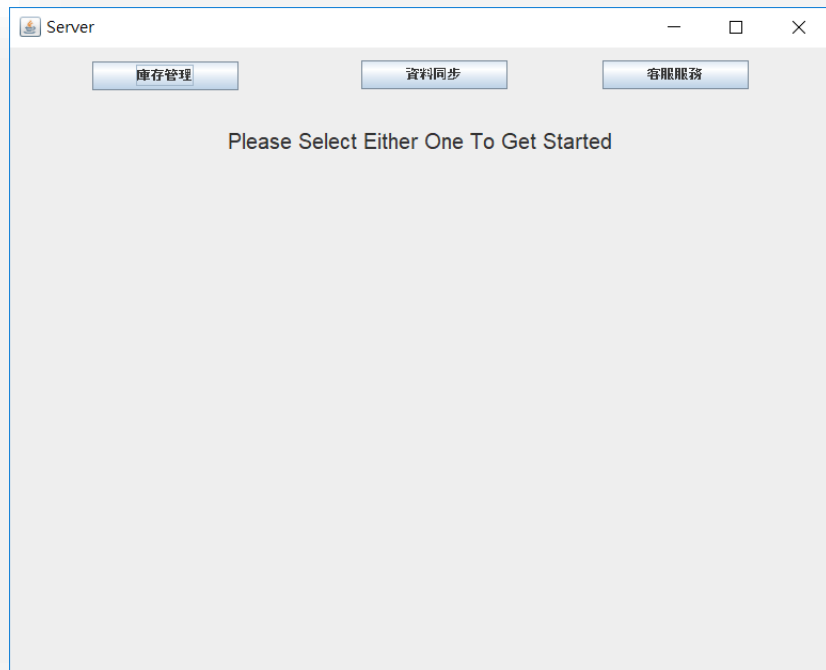
- ▶ 帳號 Admin

- ▶ 密碼 104403520

- ▶ 開啟Client，則為客戶端

- ▶ 帳號 UserA , UserB

- ▶ 密碼 104403520



程式初始畫面 等待使用者選擇頁面

1-1.

庫存管理頁面



Server 端 庫存管理 頁面

name	quantity
pencil	16
pen	16
eraser	16
ruler	16
pencil case	16

► 上牌可選擇分店A或B，依照分店選擇不同，將會改變左下角表格的結果。

► inc及dec按鈕會改變選定項目的quantity。inc為+1 dec為-1



Server 端 庫存管理 頁面 – inc dec

name	quantity
pencil	14
pen	16
eraser	18
ruler	16
pencil case	16

► 如圖為利用按鈕修改後的程式



Feature:

可在右側新增文字輸入框設定物件的quantity。
亦可新增及刪除品項



Server 端 庫存管理 頁面 – 改變分店

The screenshot shows a web application window titled "Server". It features three buttons at the top: "庫存管理" (Inventory Management), "資料同步" (Data Sync), and "查看服務" (View Service). Below these is a slider control with "分店A" (Branch A) on the left and "分店B" (Branch B) on the right. The slider is currently positioned towards Branch A. To the right of the slider are two buttons labeled "inc" and "dec". On the left side of the window is a table with two columns: "name" and "quantity".

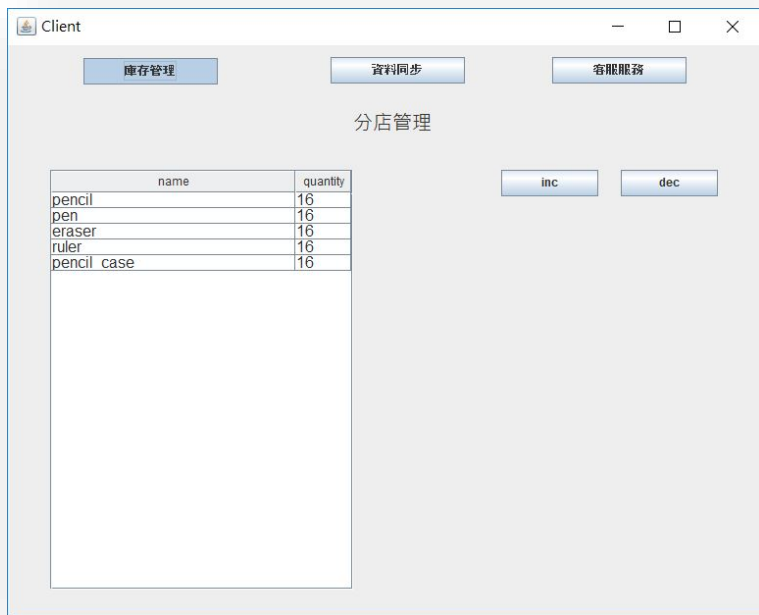
name	quantity
pencil	2
pen	2
eraser	2
ruler	2
pencil case	2

► 上排可選擇分店A或B，依照分店選擇不同，將會改變左下角表格的結果。


► 附圖為改變後的畫面，仍然可以做 inc 以及 dec



Client 端 庫存管理 頁面



- ▶ 附圖為 Client 端的畫面，僅會顯示該分店的畫面
- ▶ 依照使用者名稱不同 (userA及userB) 會顯示不同的表格

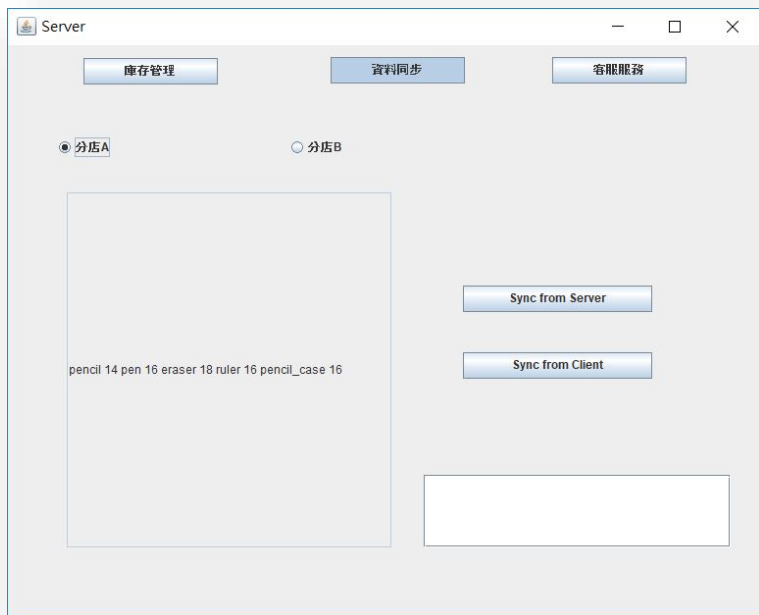
A large green diagonal shape that starts from the top right and extends towards the bottom left, covering the right half of the slide.

1-2.

資料同步頁面



Server 端 資料同步 頁面 – 取得資料



► 上排可選擇分店A或B，依照分店選擇不同，將會改變左下角文字的結果。



Server 端 資料同步 頁面 – 從伺服器同步資料

► Sync From Server

將伺服器上的資料同步
回客戶端

首先先將表格資料轉存
成字串，再利用socket
傳送回客戶端

► Sync From Client

將客戶端上的資料同步
回伺服器

將接入的data利用空格
分開讀入，將數字由字
串轉回數字加回表格中



Server 端 資料同步 頁面 – Sync From Server

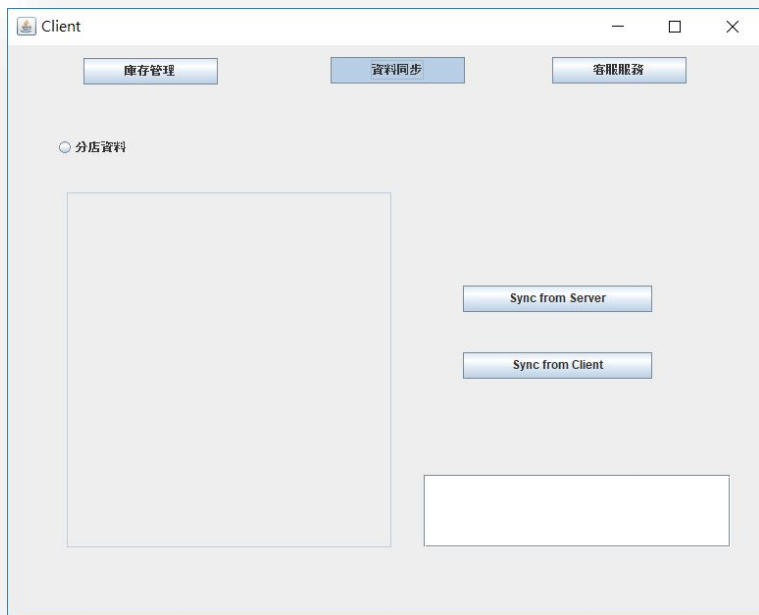


▶點選 Sync From Server 後，
將會開啟一個 Socker
Server 等待接入連線。

▶此時已經準備好資料，等
待client一連線就將資料傳
送至client端。



Client 端 資料同步 頁面 – Sync From Server



▶ 當按下 Sync From Server時，將會將伺服器的資料接收進左方的 textfield。

▶ 而當偵測到內容改變時，就會將資料寫回分店的 table中



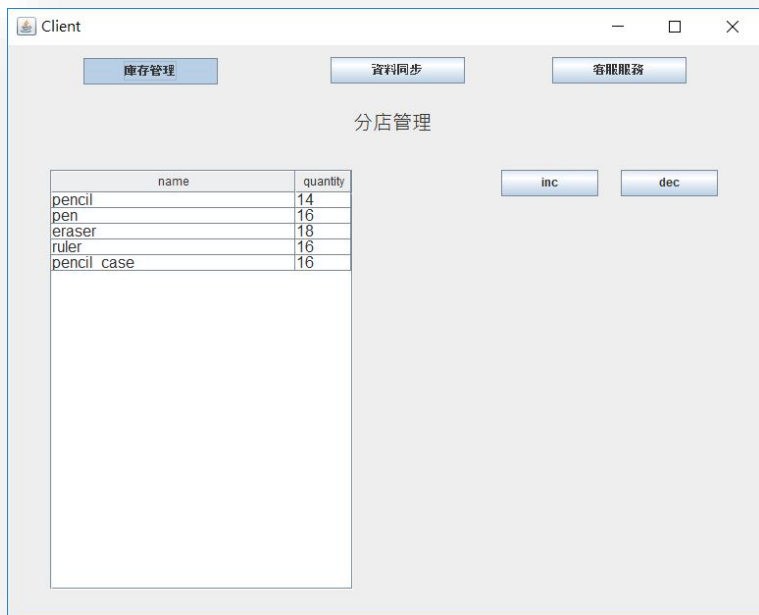
Client 端 資料同步 頁面 – Sync From Server



►如圖，本來資料為皆為16，而當收到資料後將資料同步為伺服器傳過來的data。



Client 端 資料同步 頁面 – Sync From Server



►接收完資料後，當回到庫存管理介面查看時，則會發現 Client 端的資訊被同步為 Server 上的資料了。



Client 端 資料同步 頁面 – Sync From Client

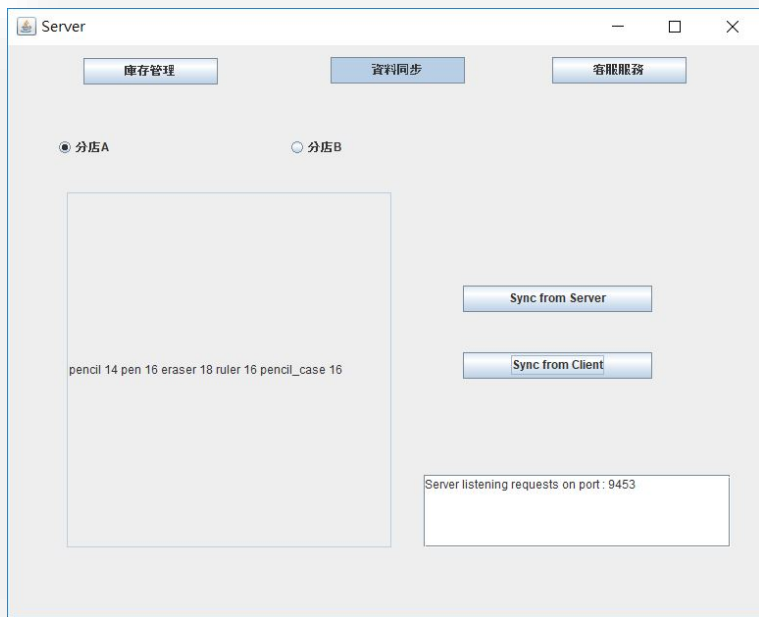


▶點選 Sync From Client 後，Client 將會把資料顯示在左方文字框，呈現等待傳送的状态。

▶此時若伺服器開啟，就可按下 Sync from Client 按鈕將資料送回Server.



Server 端 資料同步 頁面 – Sync From Client

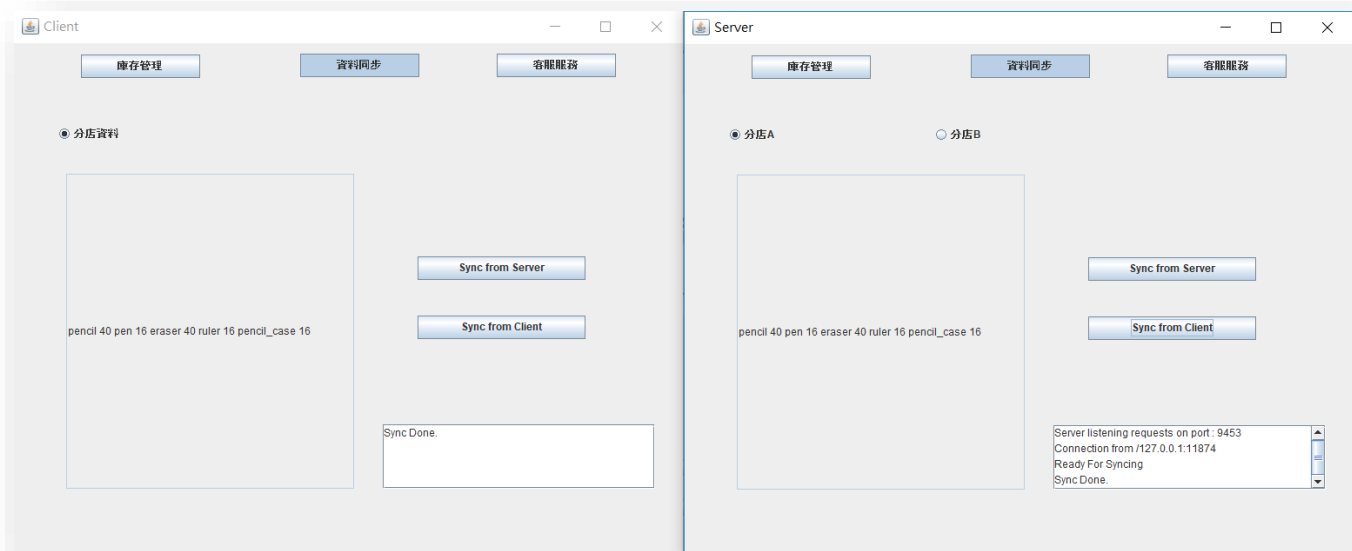


► Server端先選擇分店A或B為要同步的目標，資料會顯示在左方欄位供使用者參考。

► 按下 Sync From Client 後，就會打開伺服器等待使用者連線。



兩端 資料同步 頁面 – Sync From Client



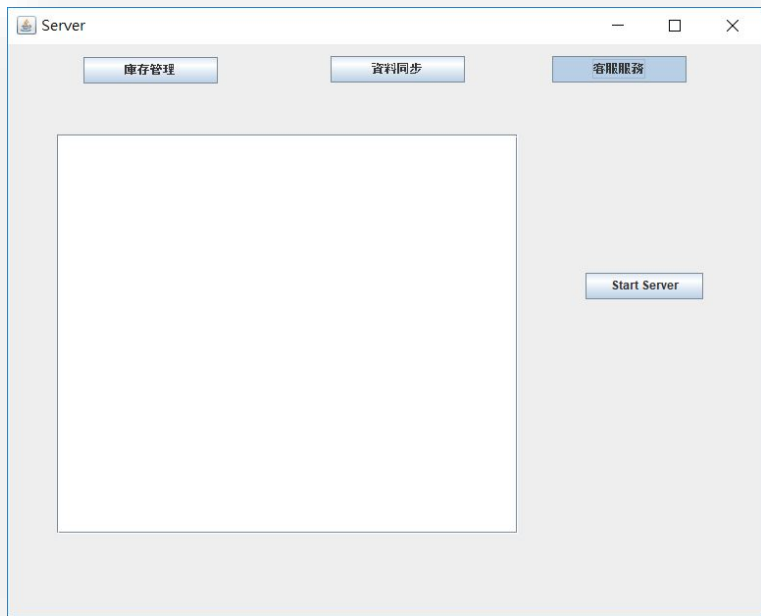
- ▶ 這邊我們可以看到，Client的資料已經被成功傳回到伺服器上，也顯示出 Sync Done 的訊息給使用者。

1-3.

客服服務頁面



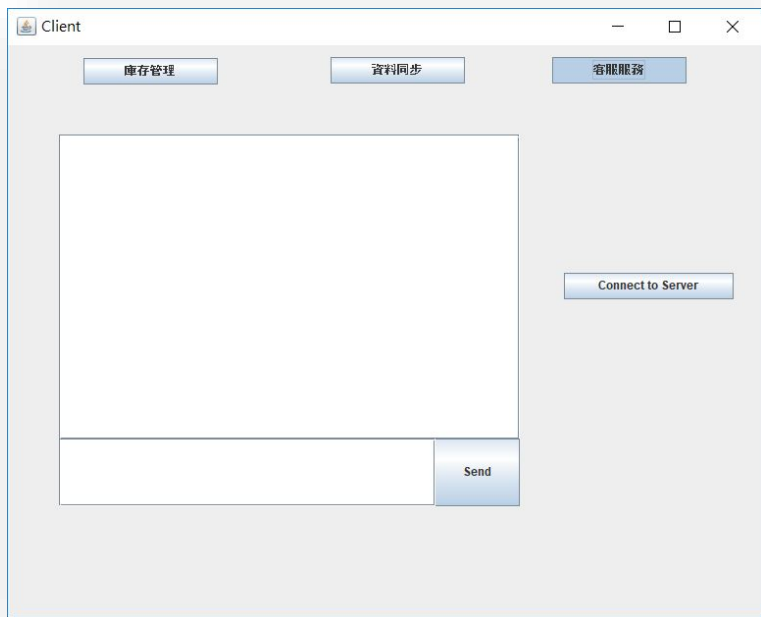
Server 端 客服服務頁



►預設的想法是因為伺服器端並不會一直有人在顧，所以這個功能比較像是一個表單填寫功能，主要是讓客戶在客戶端打上資料後傳送回伺服器端存起來，要看的時候可以點閱，但這邊只做顯示出來的功能。



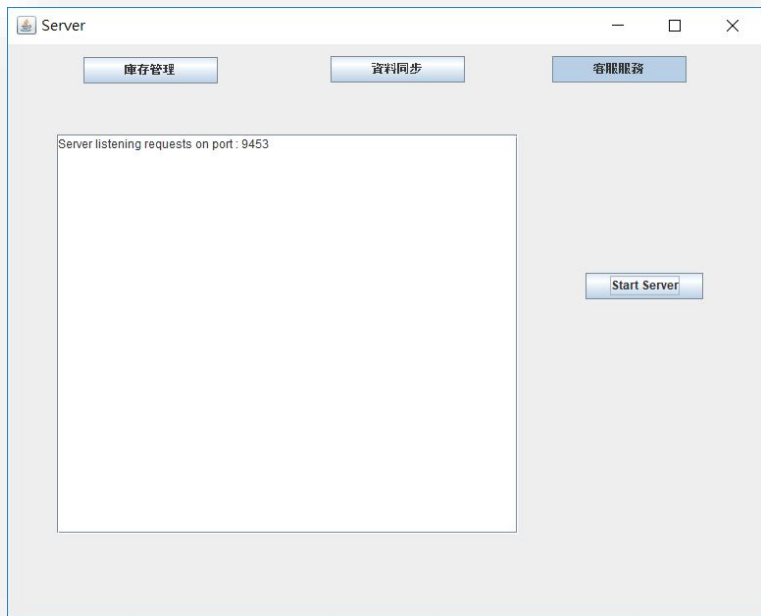
Client 端 客服服務頁面



► Client端可以將資料打在下方文字框後按下send，則可將資料放在上方文字框，等到按下Connect To Server就會將資料丟到server上顯示出來。



Server 端 客服服務頁



- ▶ 顯示出伺服器有在監聽。
- ▶ 等待客戶端連線



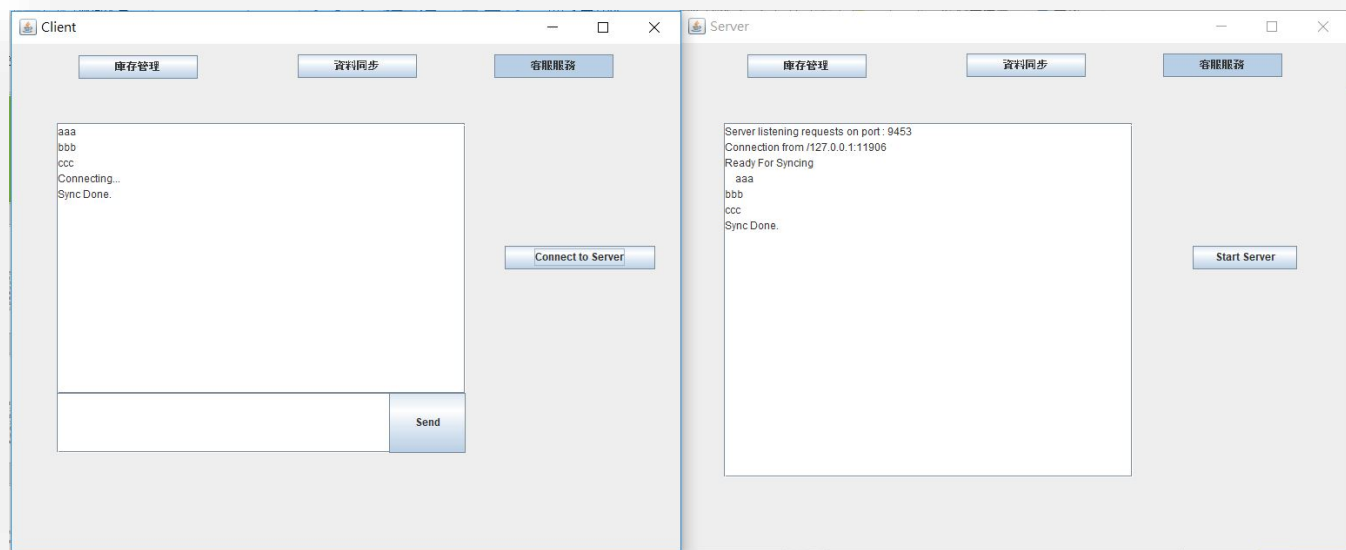
Client 端 客服服務頁面



► 資料打好後，server 也準備好了，就可以按下connect。



兩端 客戶服務頁面



- ▶ 這邊我們可以看到，Client的資料已經被成功傳到伺服器上。
- ▶ 若有需要可以存起來供日後參考。

A thick, solid green diagonal stripe runs from the top right corner towards the bottom left, separating the white background on the left from the solid green background on the right.

2.

防呆設計



防呆設計

A screenshot of a Windows-style login dialog box titled "Login". The dialog has a light gray background and a blue border. At the top left is a small icon of a person, and at the top right is a close button (X). The main content area displays the error message "Incorrect Input. Please re-enter." in red text. Below this message are two input fields: "Username :" followed by a text box, and "Password :" followed by a password box (indicated by a small vertical line on the left). At the bottom right of the dialog are two buttons: "OK" and "Cancel".

Login

Incorrect Input. Please re-enter.

Username :

Password :

OK Cancel

► 當輸入錯誤的帳號時，系統將不會接受登入。



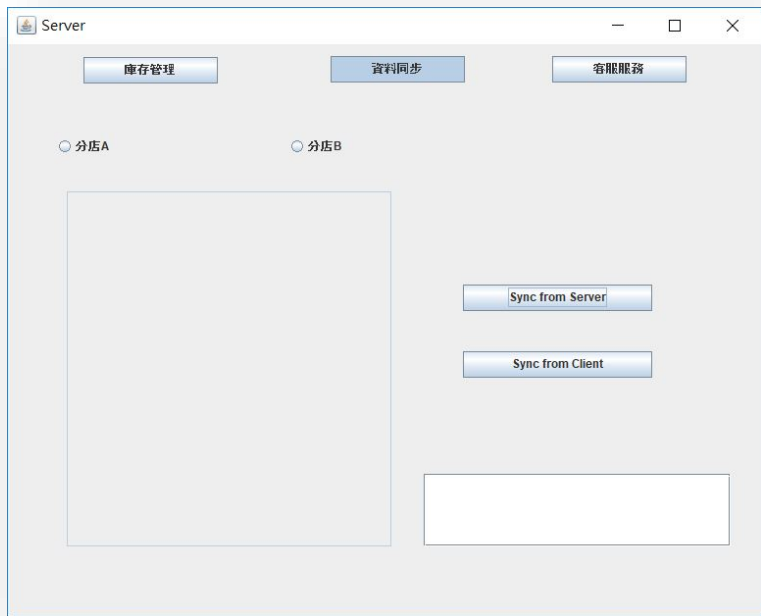
防呆設計



► 當 Server 資料同步頁面還正在等待連線時，將無法切換到別的頁面，需要等到資料傳送完成才可以離開。



防呆設計



► 當未選中分店時，無法進行資料同步，按下按鈕將不會有任何反應

A large, solid green diagonal shape that starts from the top right corner and extends towards the bottom left, creating a split background effect.

3.

支援的系統



多虧java 將可在多系統上執行

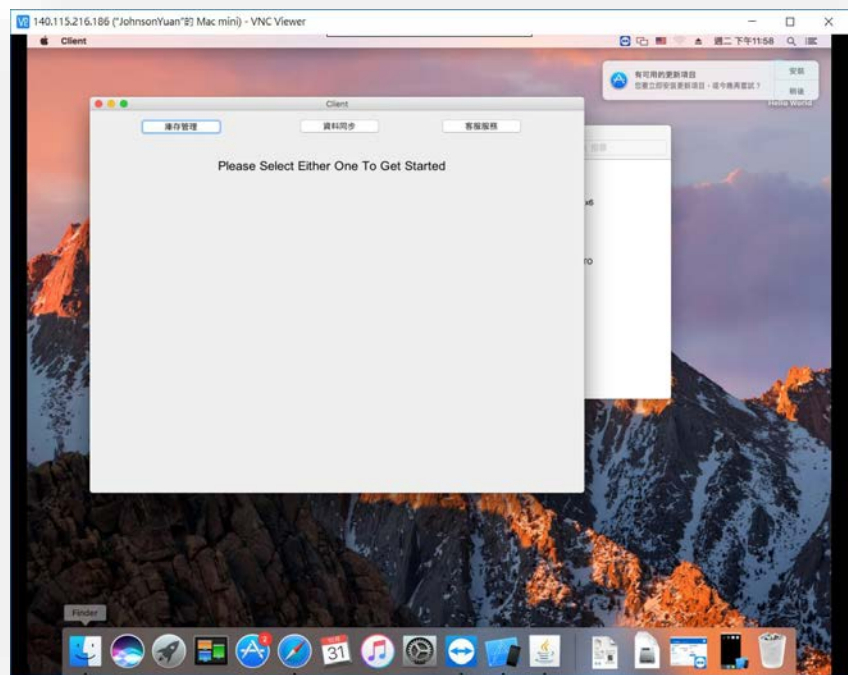
```
johnsonyuan@JohnsonYuanUbuntu: ~  
61 packages can be updated.  
0 updates are security updates.  
  
*** System restart required ***  
Last login: Mon Oct 30 01:21:26 2017 from 101.8.163.231  
johnsonyuan@JohnsonYuanUbuntu:~$ java Server  
Server listening requests on port : 5987  
^Cjohnsonyuan@JohnsonYuanUbuntu:~$ java Server  
Server listening requests on port : 9453  
有 /140.115.216.186:11971 連線進來!  
pencil 16 pen 16 eraser 16 ruler 16 pencil_case 16  
java -cp ./Server.jar  
at java.io.DataInputStream.readUnsignedShort(DataInputStream.java:340)  
at java.io.DataInputStream.readUTF(DataInputStream.java:589)  
at java.io.DataInputStream.readUTF(DataInputStream.java:564)  
at Server$RequestThread.run(Server.java:89)  
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.  
java:1149)  
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.  
.java:624)  
at java.lang.Thread.run(Thread.java:748)
```

►Ubuntu 上的 Server 端可以接收到 windows 上的 Client 資料

►此為未來可能實作功能，這邊做簡單示範而已。



多虧java 將可在多系統上執行



►Mac OSX 上也可使用這個程式