軟體品質測試設計

會員階級式聊天室(部分GUI)

D0381806

邱皇旗

目錄:

1. 動機-----------------------------------------------------------------------4
2. 摘要-----------------------------------------------------------------------4
3. 使用工具-----------------------------------------------------------------5
4. 測試方式 ---------------------------------------------------------------5
5. 測試案例 ---------------------------------------------------------------6
6. 系統架構圖-------------------------------------------------------------7
   1. 系統互動模組------------------------------------------------7
   2. RMI服務介面模型------------------------------------------8
   3. 時序圖-----------------------------------------------------------8
      1. 註冊---------------------------------------------------------------8
      2. 登入---------------------------------------------------------------9
      3. Server 與 MySQL連線------------------------------------9
      4. 聊天傳遞模型-----------------------------------------------10
   4. 使用流程圖--------------------------------------------------10
7. 功能架構圖------------------------------------------------------------12
8. 功能說明---------------------------------------------------------------13
   1. 會員註冊-----------------------------------------------------13
   2. 會員登入-----------------------------------------------------13
   3. 延續留言板--------------------------------------------------13
   4. 聊天室--------------------------------------------------------13
9. GUI介面----------------------------------------------------------------14
   1. 開始介面-----------------------------------------------------14
   2. 登入介面-----------------------------------------------------15
   3. 聊天室--------------------------------------------------------16
   4. 新增主題-----------------------------------------------------16
   5. 新增回覆-----------------------------------------------------17
10. Cmd 介面---------------------------------------------------------18
    1. 會員登入-----------------------------------------------------18
    2. 聊天室\_1-----------------------------------------------------18
    3. 聊天室\_2-----------------------------------------------------18
11. jUNIT測試(程式片段)-------------------------------------19
    1. 登入模組-----------------------------------------------------19
    2. 註冊模組-----------------------------------------------------20
    3. 主題模組-----------------------------------------------------21
    4. 回覆模組-----------------------------------------------------22
12. 遇到的問題與解決方法----------------------------------23
13. 心得------------------------------------------------------------24
14. **動機:**

利用網路通訊，實作出一個具備留言板的聊天室，其中具備了會員機制，能夠登入、註冊、建立主題，回覆主題，以及聊天功能。雖然這是一個很普通的系統，但在一個學生的角度來看，我想知道如何使一個普通的系統，能有更好的可能性，對於未來在撰寫程式時，也能用這樣的方式去檢驗系統。

1. **摘要:**

關於我們這個聊天室，設置了許多防呆機制，讓每個使用者都可以有自己管理自己聊天室的權利，即便是最底層的會員，至於基本註冊系統，我們也建立 My SQL來做檢查與存取的機制，在通訊方面，我們使用三方交握的方式，讓每個Client都透過Server端，再傳送到目標Client，這樣能確保傳出去的訊息，都能被存取，也能被記錄下來。

1. **使用工具:**
   1. 程式語言: JAVA
   2. 其他使用工具: MySQL
   3. 紀錄工具: Github、Junit測試( [**https://github.com/D038806/software\_test.git**](https://github.com/D038806/software_test.git) **)**
   4. 呈現介面:Terminal(黑白介面)、部分GUI
2. **測試方式:**
   1. 利用JUnit做每個method做單元測試、整合測試。
   2. 利用assert()做程式碼的檢測。
   3. 利用MySQL本身的規則做資料的檢測。
3. **測試案例:**
   1. 測試是否為會員
   2. 測試該使用者是否有登入
   3. 針對特殊的輸入法做測試
   4. 測試缺少欄位的回應

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 甲 | 乙 | 丙 | 丁 |
| 1 | T | F | X | X |
| 2 | T | T | T | X |
| 3 | T | T | X | T |
| 4 | T | T | T | T |
| 5 | T | T | X | X |
| 6 | F | F | X | X |

1. **系統結構圖:** 
   1. **系統互動模組:**

\*訊息收發

\*透過Server將存取最近的資料

\*檢測中斷機制

通訊系統

DataBase

\*帳號規格檢測

\*密碼規格檢測

\*檢查帳號是否存在

\*檢查是否有重複的密碼

會員註冊系統

\*檢查帳號是否存在

\*檢查是否為該使使用者密碼

\*檢查是否已登入中

會員登入系統

* 1. **RMI服務介面模型:**

Interface Definition Language (IDL)

(AtithmeticInterface.java)

Class ArithmeticServer

Class CalculatorRMIClient

RMI服務

RMI

Registry

與 ArithmeticServer連線

與 ArithmeticServer連線

Class ArithmeticRMIImpl

輸入選項op，並呼叫對應的副程式

String singIn(String name)

String register(String name)

顯示執行結果

* 1. **時序圖:**
     1. **註冊:**

註冊

註冊

回覆 port

回覆 port

Server

Client2

Client1

* + 1. **登入:**

註冊

註冊

回覆 port

回覆 port

Server

Client2

Client1

* + 1. **Server與My SQL 連接:**

副程式要求的資料

DB

回覆副程式要求的資料

Server

* + 1. **聊天傳遞方式模型**

傳遞一個

Message

轉傳一個

Message

利用RMI

利用Socket

Server

Client2

Client1

\*\*另用Socket 需要對方的port，所以在用RMI傳message 時，必須傳出對方名稱，再由Server到My SQL中，找出目標Client的port。

* 1. **使用流程:**
     1. **Client(初次使用):**

註冊

輸入申請帳號

檢查格式與DB是否想相同的資料

輸入申請密碼

註冊成功

* + 1. **Client:**

登入

檢查格式與DB是否想相同的資料

輸入帳號&密碼

登入成功

* + 1. **Server:**

Client 傳送訊息

DB

Server

傳給目標Client

1. **功能架構圖:**

**輸入對方IP位置**

**↓**

**開始聊天**

**(傳送訊息、  
接收訊息)**

**討論區**

**Yeah聊天室**

**看板**

**註冊成功**

**新增文章**

**新增回覆**

**聊天室**

**註冊**

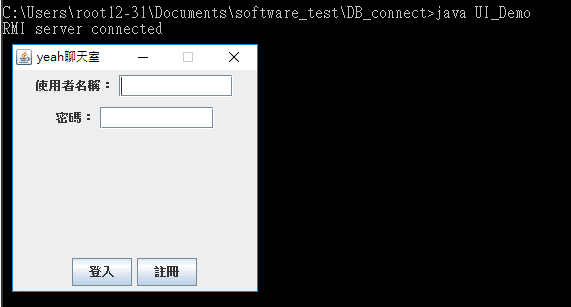
**登入**

1. **功能說明:**
2. **會員註冊:**
   1. 會進入到 My SQL裡搜尋是否相同名稱的使用者。
   2. 註冊成功時會給予一個獨特的port

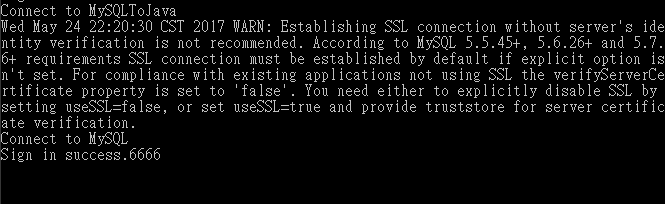
**\*\*備註: 註冊完後，仍需要進行登入的動作。**

1. **會員登入:**
   1. 會進入到My SQL 裡搜尋是否有此使用者名稱。
   2. 將 insideSystem 改成 true ，才能執行主題的建立
2. **延續留言板:**
   1. 新增主題
      1. 可加入主題內容
   2. 對討論區進行回覆
   3. 顯示所有主題及回覆內容
3. **聊天室\_1** 
   1. 使用者在使用此功能前，必須先登入，否則會跳出提醒。
   2. 於使用過程中，必須輸入對方的IP位址及名稱。
   3. 作法: 傳遞訊息
      1. 當加入一名使用者( Client )要傳遞訊息時，會先將訊息及自己IP以及對方IP，記錄到 My SQL
      2. 再藉由Server幫助Client1傳給目標Client2
   4. 作法: 監聽
      1. 每個Client都有自己的port專門負責監聽。
      2. 並且每個Client都有兩個Thread，一個正常做其他功能，另一個使用自己的特有的port負責監聽。
4. **GUI介面設計:**
   1. **開始介面:**

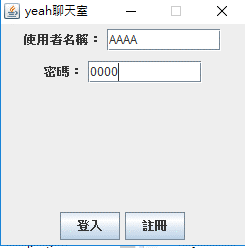


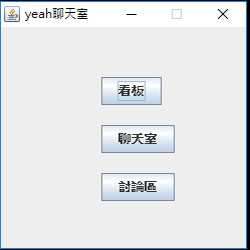


* 1. **登入介面:**

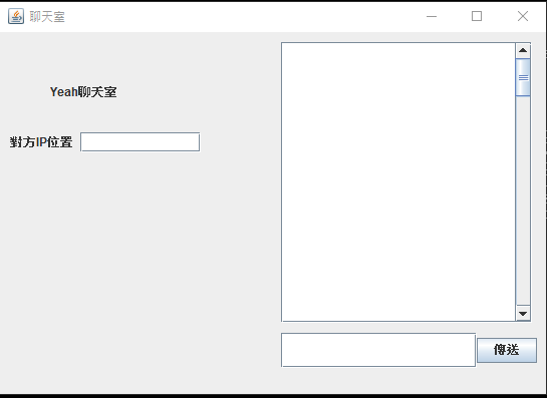


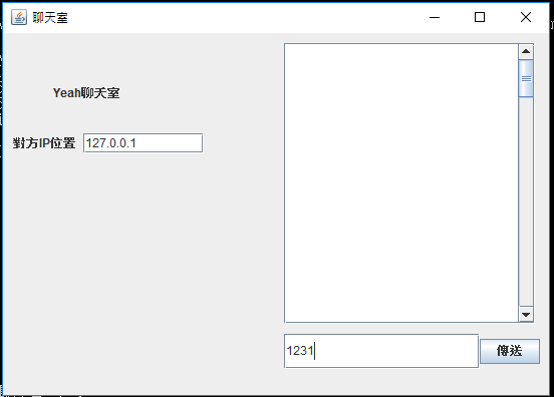
會員的port



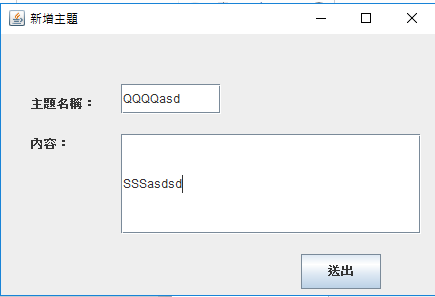


* 1. **聊天室:**

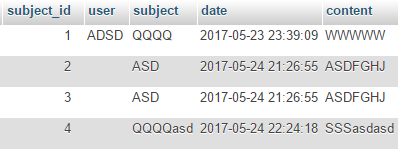




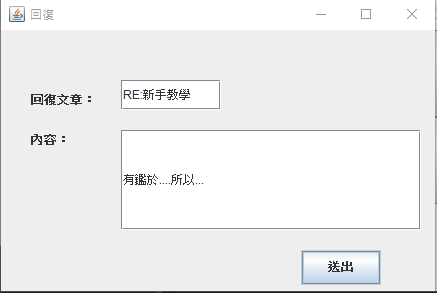
* 1. **新增主題:**





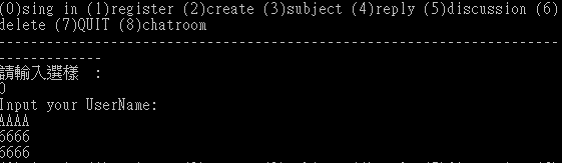


* 1. **新增回覆:**





1. **CMD介面:**
   1. **會員登入:**



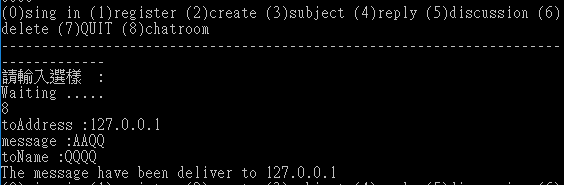
🡪 Client1



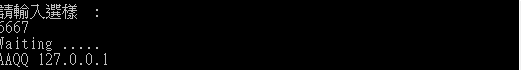


🡪 Client2

* 1. **聊天室\_1:**



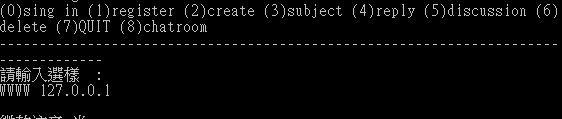
Client1傳訊息給Client2



* 1. **聊天室\_2:**



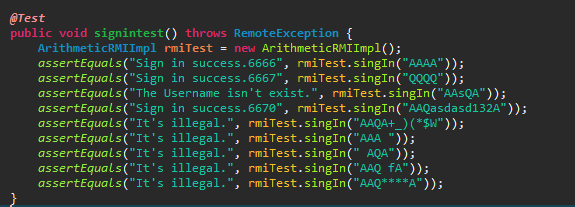
Client2 馬上就可以直接回覆Client1



Client1 可以做自己的事，直到Client2

傳送訊息，而會跳出訊息，但並不會終止Client1的動作。

1. **JUNIT測試(程式片段):**
   1. **登入模組:**



其中測試案例針對:

* + 1. 輸入已註冊的會員名稱

-回覆成功訊息並回傳專屬的port

* + 1. 輸入未註冊的會員名稱

-回覆為存在的訊息

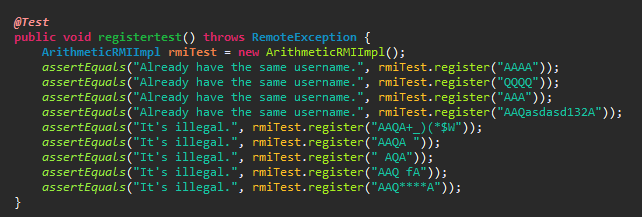
* + 1. 輸入名稱中包含特殊字元

-回覆錯誤訊息

* + 1. 輸入名稱中包含空白

-回覆錯誤訊息

* 1. **註測模組:**



其中測試案例針對:

1. 輸入已經註冊的會員名稱

-回覆已存在的訊息

1. 輸入未註冊的名稱

-回覆成功訊息並回傳專屬的port

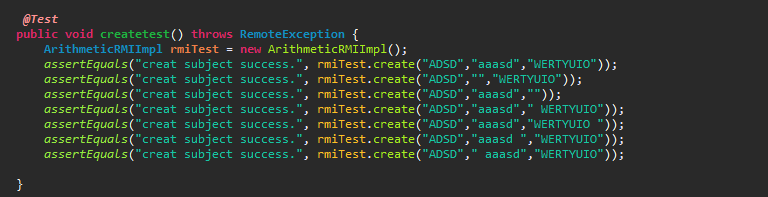
1. 輸入具有特殊字元的名稱

-回覆錯誤訊息

1. 輸入具有空白字元的名稱(包含:名稱前、名稱後、名稱中)

-回覆錯誤訊息

* 1. **主題模組:**



1. 輸入正常新增格式

-回覆新增成功的訊息

1. 輸入缺少主題名稱的案例
2. -回覆新增成功的訊息
3. 輸入缺少主題內容的測試案例

-回覆新增成功的訊息

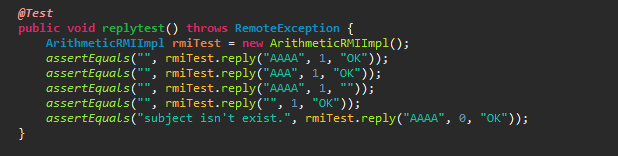
1. 輸入主題名稱包含特殊字元

--回覆新增成功的訊息

1. 輸入主題內容包含特殊字元

-回覆新增成功的訊息

* 1. **回覆模組:**



1. 輸入正常回覆
2. 輸入缺少回覆人的名稱
3. 輸入缺少回覆內容
4. 輸入為存在編號的主題

-回覆錯誤

1. **遇到的困難:**
   1. 在負責UI的介面設計上，由於已有一段時間沒有寫UI，所以剛開始著手時並不是非常熟悉，後來是到圖書館借了工具書一邊溫習一別撰寫，比較特別的困難是，當初要做事件處裡的視窗功能，沒有注意到一個frame視窗一次只能賦予一個button一個事件處理的功能，若在一個frame視窗中有n個button皆要做事件處理，則需再另外獨立寫出n個frame去做n個button的事件。版面配置上也有一些小阻礙，一開始在做button的配置時沒有注意到給予的座標位置而被前面設定的Panel擋住。基本上遇到的困難不少，但都是在細心觀察下，另外再做其他測試才得以解決。
   2. 在監聽部分，我多設置一個Thread，使他可以一直處於監聽的部分，因為我們無法預測對方何時會傳訊息過來，所以要一直處於預備的狀態，其中我在撰寫時遇到的問題是，他不能一直占用相同的port，這會使其他Client執行錯誤，所以後來我直接在註冊部分，將每個會員都在註冊時，加入一個專用的port，使每個傳訊息的client能知道對方的port方便傳遞，也同時解決監聽的部分。
   3. Swing在結合功能方面，因為我們是分開撰寫UI與功能，所以在銜接時，有時會有溝通上的困難，也會有程式碼上的不符合的問題，但這些都是我們學習的重點，未來在一個團隊裡，不可能會是一個人負責全部，反而需要互相溝通，所以這次的專題，真的學到很多。
2. **心得**

這次作業是我這個學期的集大成，也是我在大學生活中，第一次將所學的做結合，過去只是單單針對某一個領域加以應用，卻從未將其結合，之前在學習資料庫系統時，也只是建檔，正規化處理，但這是將其運用到了網路通訊上，是一大收穫，其中有許多需要了解的部分。在我們的系統中，運用了RMI這個服務介面，雖然他只支援java這個語言，但是他也幫助我們節省許多細節，這次對於我們的系統，最大的特點莫過於監聽這個部分，雖說這是一個很基本的功能，但卻是使用作多技術的地方。