

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:

壹. 動機-----	4
貳. 摘要-----	4
參. 使用工具-----	5
肆. 測試方式 -----	5
伍. 測試案例 -----	6
陸. 系統架構圖-----	7
甲、 系統互動模組-----	7
乙、 RMI 服務介面模型-----	8
丙、 時序圖-----	8
i. 註冊-----	8
ii. 登入-----	9
iii. Server 與 MySQL 連線-----	9
iv. 聊天傳遞模型-----	10
丁、 使用流程圖-----	10
柒. 功能架構圖-----	12
捌. 功能說明-----	13
甲、 會員註冊-----	13
乙、 會員登入-----	13

丙、	延續留言板-----	13
丁、	聊天室-----	13
玖.	GUI 介面-----	14
甲、	開始介面-----	14
乙、	登入介面-----	15
丙、	聊天室-----	16
丁、	新增主題-----	16
戊、	新增回覆-----	17
壹拾.	Cmd 介面-----	18
甲、	會員登入-----	18
乙、	聊天室_1-----	18
丙、	聊天室_2-----	18
壹拾壹.	jUNIT 測試(程式片段)-----	19
甲、	登入模組-----	19
乙、	註冊模組-----	20
丙、	主題模組-----	21
丁、	回覆模組-----	22
壹拾貳.	遇到的問題與解決方法-----	23
壹拾參.	心得-----	24

壹. 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。

貳. 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。

參. 使用工具:

甲、 程式語言: JAVA

乙、 其他使用工具: MySQL

丙、 紀錄工具: Github、JUnit 測試

(https://github.com/D038806/software_test.git)

丁、 呈現介面:Terminal(黑白介面)、部分 GUI

肆. 測試方式:

甲、 利用 JUnit 做每個 method 做單元測試、整合測試。

乙、 利用 `assert()` 做程式碼的檢測。

丙、 利用 MySQL 本身的規則做資料的檢測。

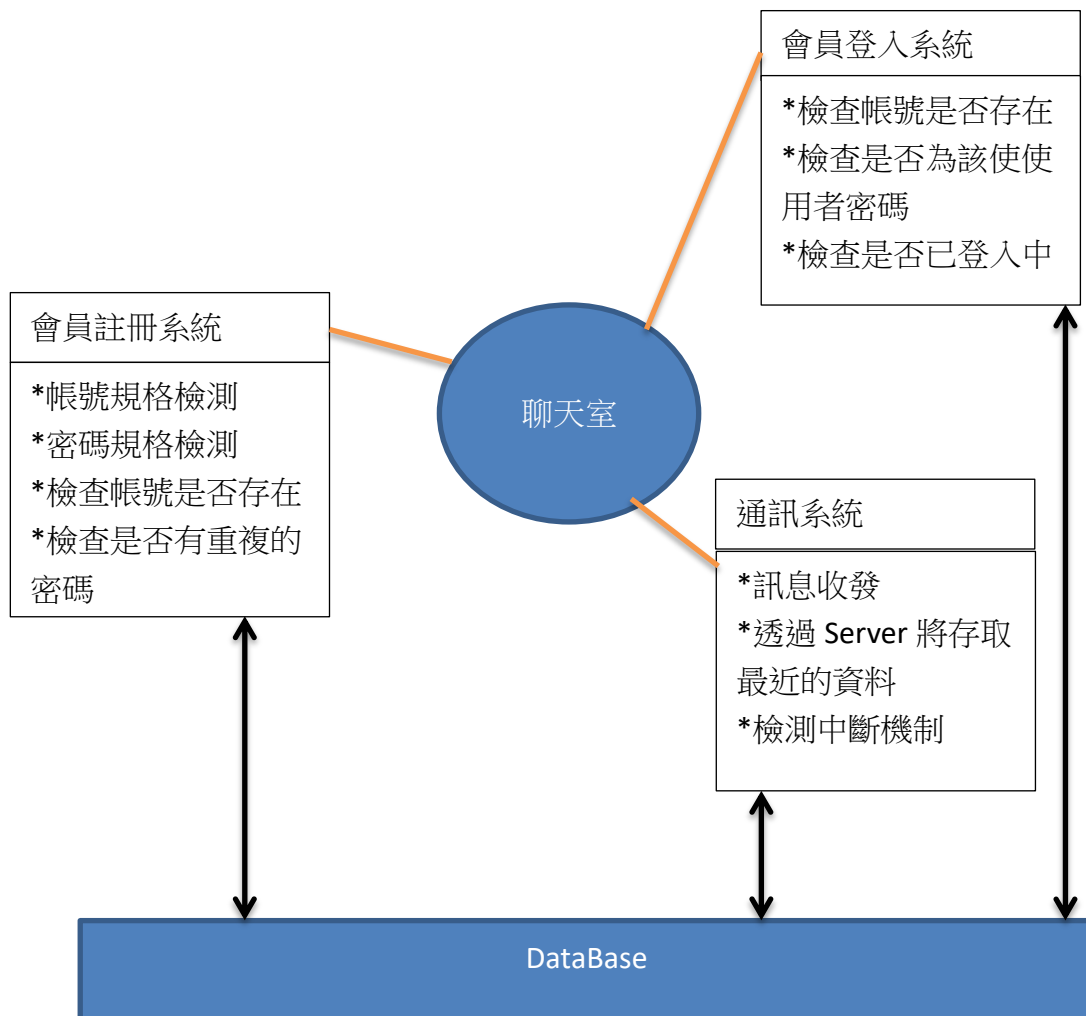
伍. 測試案例:

- 甲、 測試是否為會員
- 乙、 測試該使用者是否有登入
- 丙、 針對特殊的輸入法做測試
- 丁、 測試缺少欄位的回應

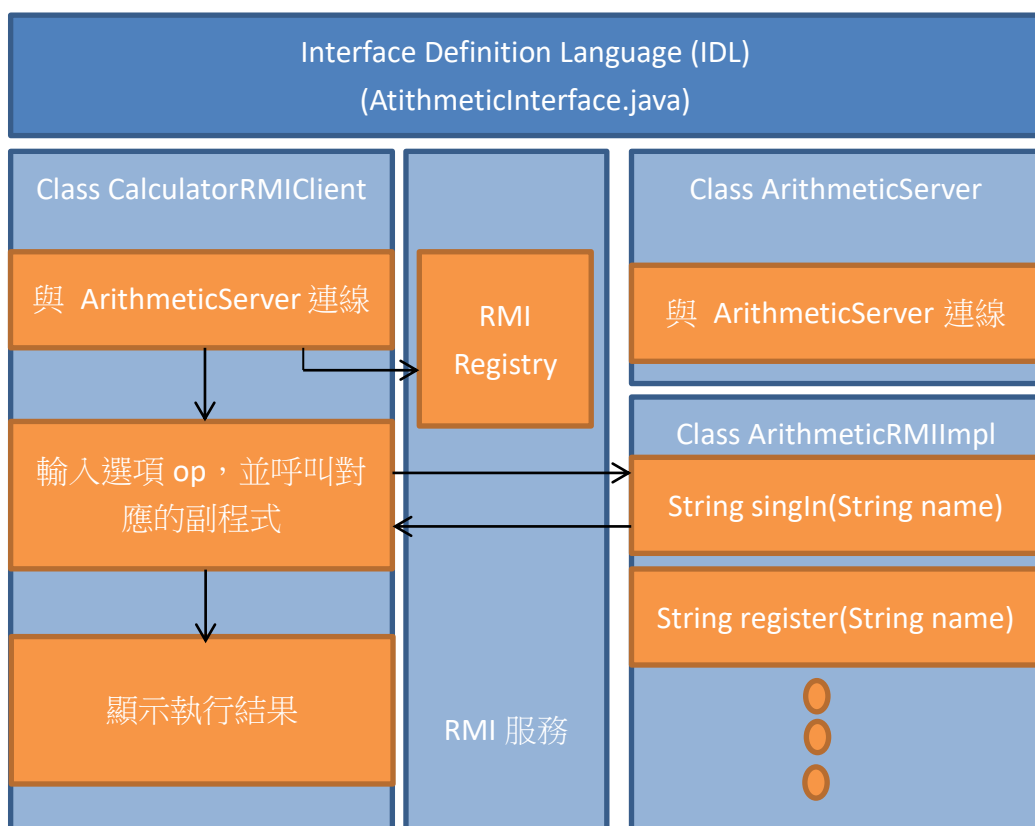
	甲	乙	丙	丁
1	T	F	X	X
2	T	T	T	X
3	T	T	X	T
4	T	T	T	T
5	T	T	X	X
6	F	F	X	X

陸. 系統結構圖:

甲、 系統互動模組:

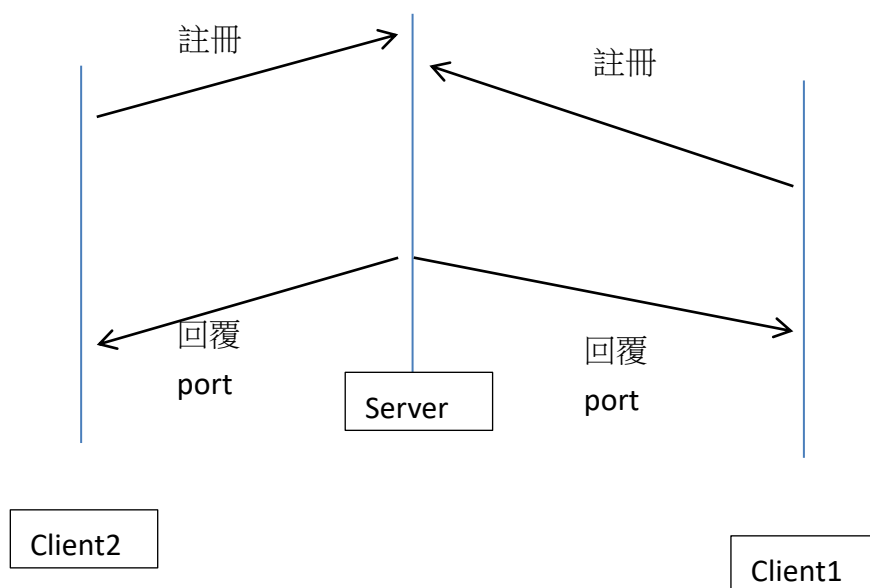


乙、 RMI 服務介面模型:

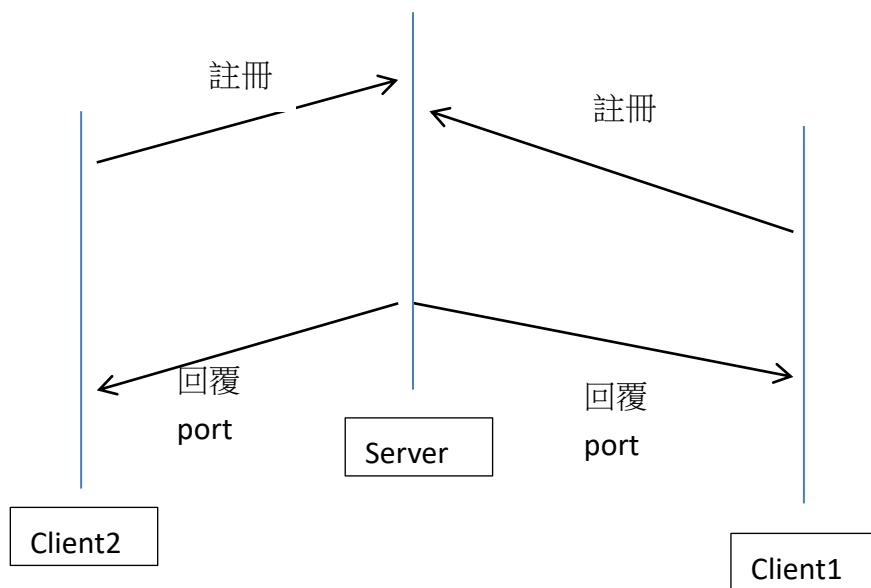


丙、 時序圖:

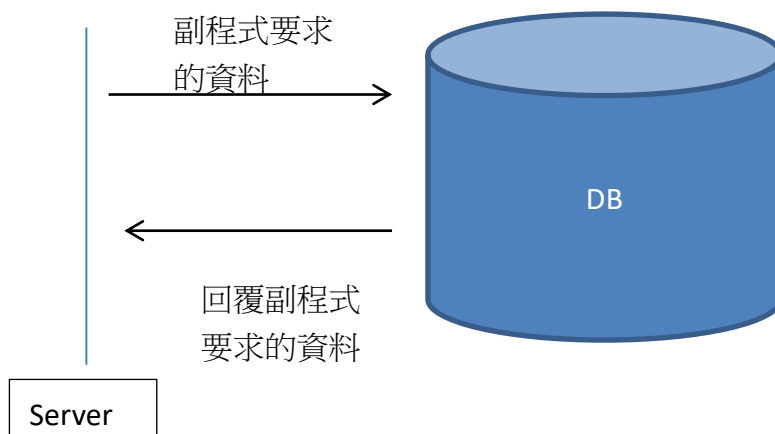
i. 註冊:

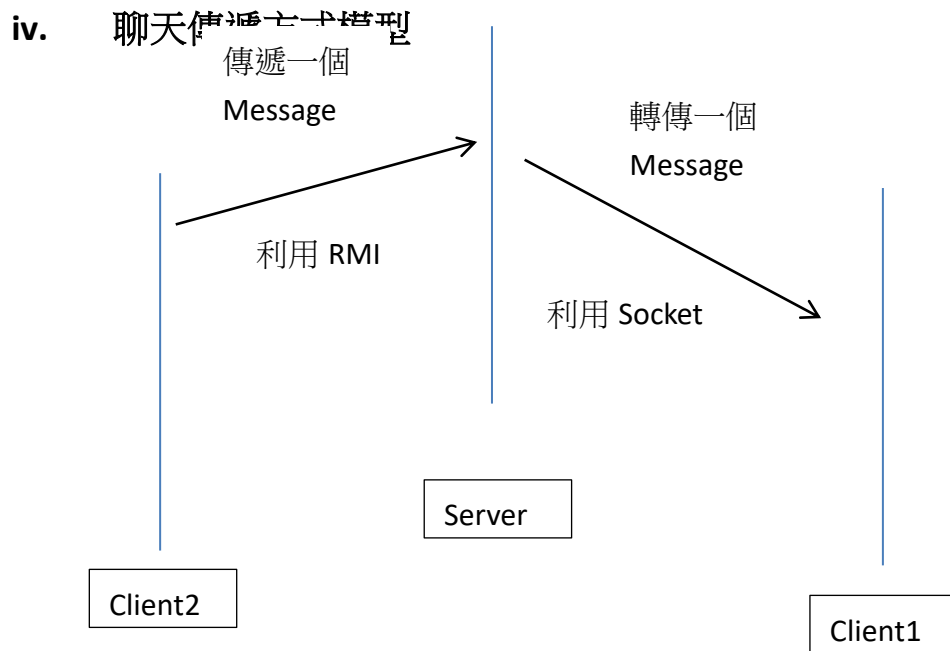


ii. 登入:



iii. Server 與 My SQL 連接:

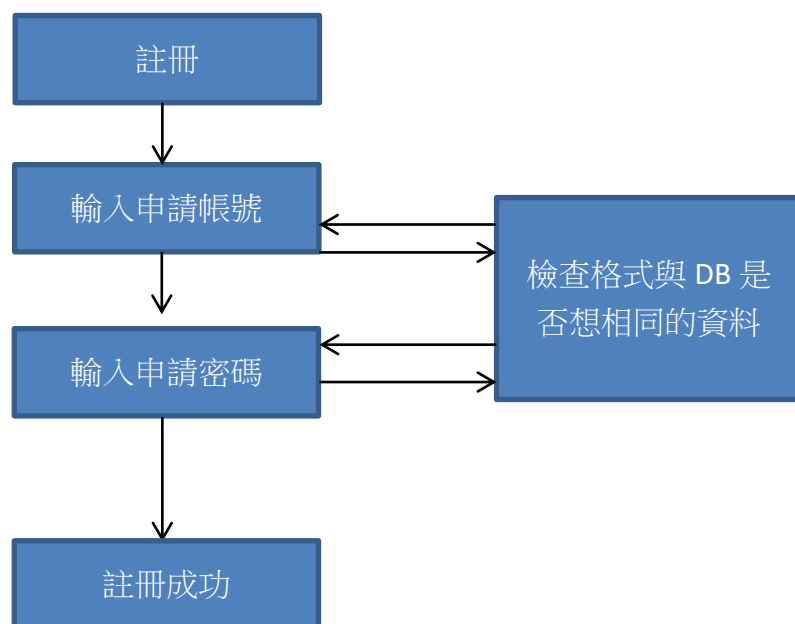




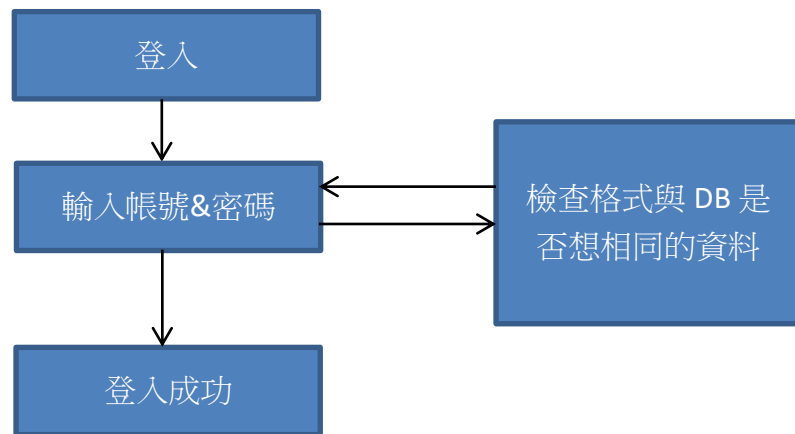
**另用 Socket 需要對方的 port，所以在用 RMI 傳 message 時，必須傳出對方名稱，再由 Server 到 My SQL 中，找出目標 Client 的 port。

丁、 使用流程:

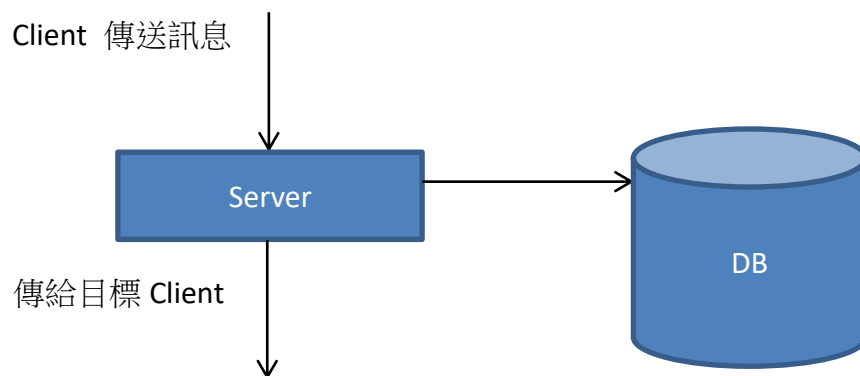
i. Client(初次使用):



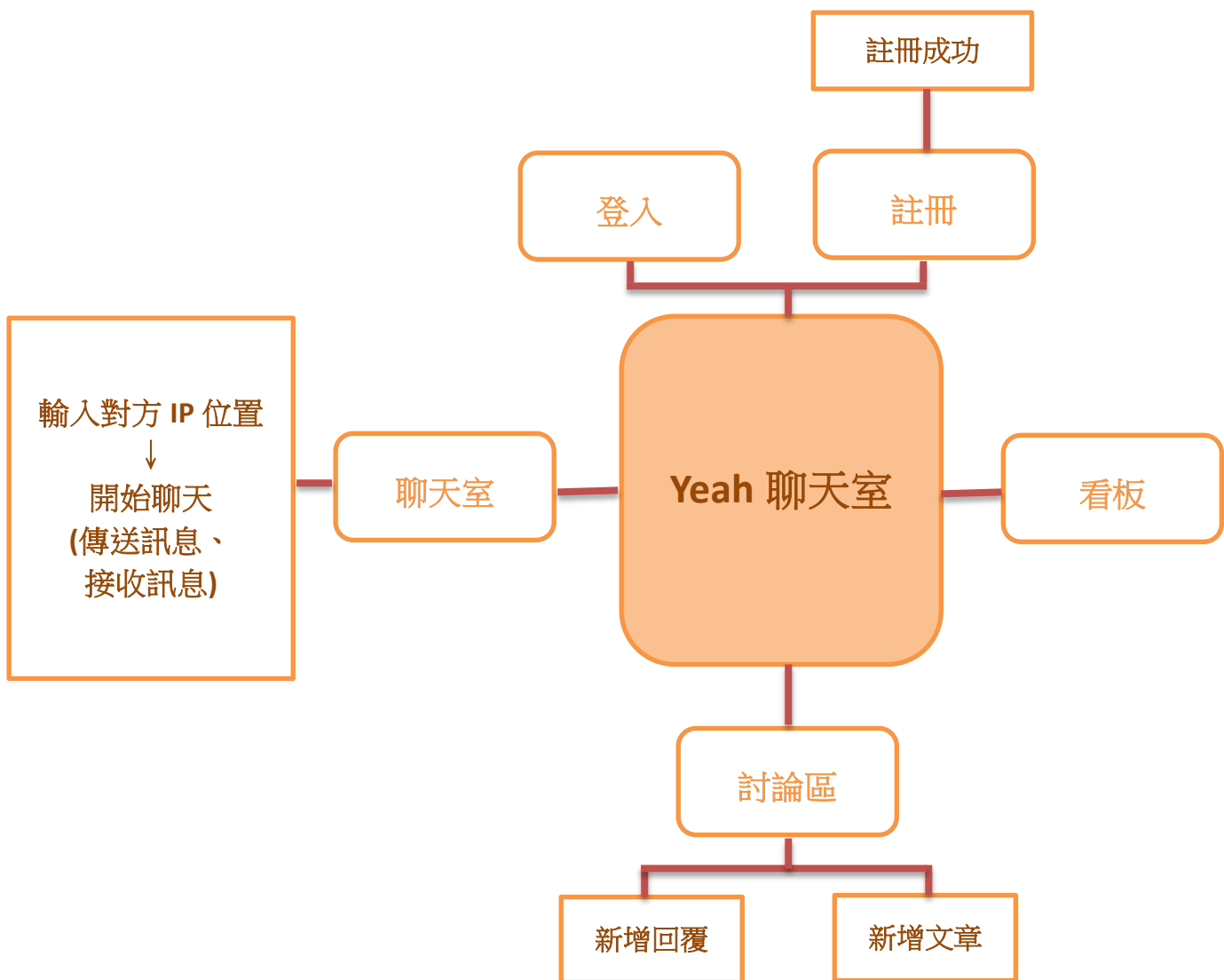
ii. Client:



iii. Server:



柒. 功能架構圖:



捌. 功能說明:

(1) 會員註冊:

- 甲、會進入到 My SQL 裡搜尋是否相同名稱的使用者。
- 乙、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。

(2) 會員登入:

- 甲、會進入到 My SQL 裡搜尋是否有此使用者名稱。
- 乙、將 insideSystem 改成 true ，才能執行主題的建立

(3) 延續留言板:

- 甲、新增主題
 - i. 可加入主題內容
- 乙、對討論區進行回覆
- 丙、顯示所有主題及回覆內容

(4) 聊天室_1

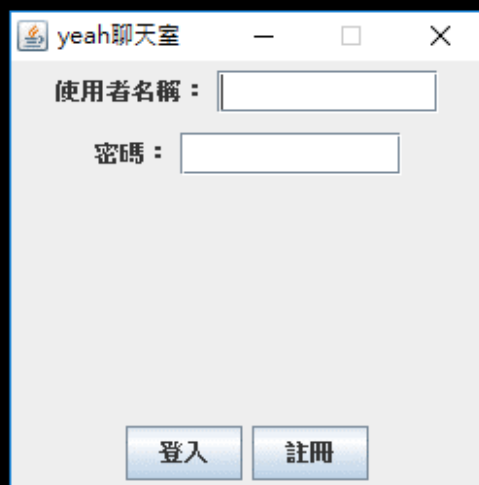
- 甲、使用者在使用此功能前，必須先登入，否則會跳出提醒。
- 乙、於使用過程中，必須輸入對方的 IP 位址及名稱。
- 丙、作法: 傳遞訊息
 - i. 當加入一名使用者(Client)要傳遞訊息時，會先將訊息及自己 IP 以及對方 IP，記錄到 My SQL
 - ii. 再藉由 Server 幫助 Client1 傳給目標 Client2
- 丁、作法: 監聽
 - i. 每個 Client 都有自己的 port 專門負責監聽。
 - ii. 並且每個 Client 都有兩個 Thread，一個正常做其他功能，另一個使用自己的特有的 port 負責監聽。

玖. GUI 介面設計:

甲、 開始介面:

```
C:\Users\root12-31\Documents\software_test\DB_connect>java -cp mysql.jar;. ArithmeticServer
Registering ...
Register success
```

```
C:\Users\root12-31\Documents\software_test\DB_connect>java UI_Demo
RMI server connected
```



The image shows a Java Swing window titled "yeah聊天室" (yeah chat room). The window has a standard Windows-style title bar with minimize, maximize, and close buttons. Inside the window, there are two text input fields. The first field is labeled "使用者名稱:" (Username:) and the second field is labeled "密碼:" (Password:). Below these fields, there are two buttons: "登入" (Login) and "註冊" (Register). The window has a light gray background and a blue border.

乙、 登入介面:

```
Connect to MySQLToJava
Wed May 24 22:20:30 CST 2017 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCertificate property is set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificate verification.
Connect to MySQL
Sign in success.6666
```

會員的 port

yeah聊天室

使用者名稱: AAAA

密碼: 0000

登入 註冊

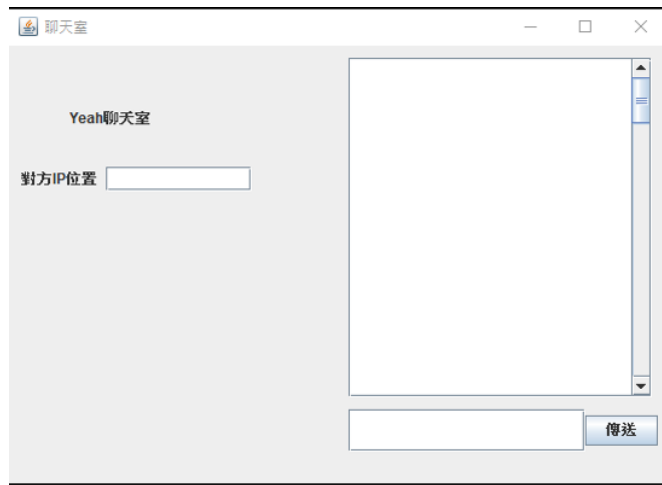
yeah聊天室

看板

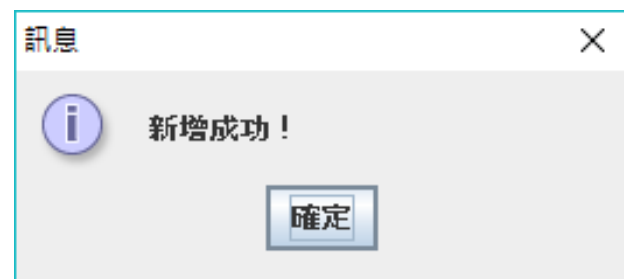
聊天室

討論區

丙、 聊天室:

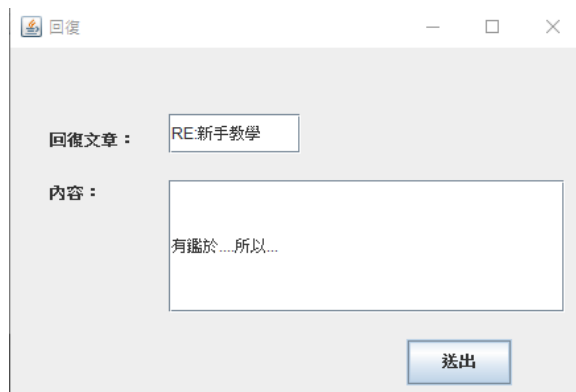


丁、 新增主題:



subject_id	user	subject	date	content
1	ADSD	QQQQ	2017-05-23 23:39:09	WWWWW
2	ASD		2017-05-24 21:26:55	ASDFGHJ
3	ASD		2017-05-24 21:26:55	ASDFGHJ
4		QQQQasd	2017-05-24 22:24:18	SSSasdsd

戊、 新增回覆:

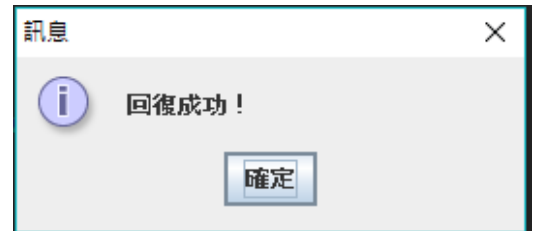


回復

回復文章： RE 新手教學

內容： 有鑑於....所以...

送出



壹拾. CMD 介面:

甲、 會員登入:

```
(0)sing in (1)register (2)create (3)subject (4)reply (5)discussion (6)
delete (7)QUIT (8)chatroom
-----
請輸入選樣 :
0
Input your UserName:
AAAA
6666
6666
```

→ Client1

```
(0)sing in (1)register (2)create (3)subject (4)reply (5)discussion (6)
delete (7)QUIT (8)chatroom
-----
請輸入選樣 :
0
Input your UserName:
QQQQ
6667
```

→ Client2

	username	portNumber
刪除	AAAA	6666
刪除	QQQQ	6667
刪除	WWWW	6668

乙、 聊天室_1:

```
(0)sing in (1)register (2)create (3)subject (4)reply (5)discussion (6)
delete (7)QUIT (8)chatroom
-----
請輸入選樣 :
8
Waiting .....
toAddress :127.0.0.1
message :AAQQ
toName :QQQQ
The message have been deliver to 127.0.0.1
```

Client1 傳訊息給 Client2

```
請輸入選樣 :
6667
Waiting .....
AAQQ 127.0.0.1
```

丙、 聊天室_2:

```
Waiting .....
AAQQ 127.0.0.1
8
toAddress :127.0.0.1
message :WWWW
toName :AAAA
The message have been deliver to 127.0.0.1
```

Client2 馬上就可以直接回覆 Client1

```
(0)sing in (1)register (2)create (3)subject (4)reply (5)discussion (6)
delete (7)QUIT (8)chatroom
-----
請輸入選樣 :
WWWW 127.0.0.1
```

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。

壹拾壹. JUNIT 測試(程式片段):

甲、 登入模組:

```
@Test
public void signintest() throws RemoteException {
    ArithmeticRMIIImpl rmiTest = new ArithmeticRMIIImpl();
    assertEquals("Sign in success.6666", rmiTest.signIn("AAAA"));
    assertEquals("Sign in success.6667", rmiTest.signIn("QQQQ"));
    assertEquals("The Username isn't exist.", rmiTest.signIn("AAsQA"));
    assertEquals("Sign in success.6670", rmiTest.signIn("AAQasdasd132A"));
    assertEquals("It's illegal.", rmiTest.signIn("AAQA+)(*$W"));
    assertEquals("It's illegal.", rmiTest.signIn("AAA "));
    assertEquals("It's illegal.", rmiTest.signIn(" AQA"));
    assertEquals("It's illegal.", rmiTest.signIn("AAQ fA"));
    assertEquals("It's illegal.", rmiTest.signIn("AAQ***A"));
}
```

其中測試案例針對:

- a. 輸入已註冊的會員名稱
 - 回覆成功訊息並回傳專屬的 port
- b. 輸入未註冊的會員名稱
 - 回覆為存在的訊息
- c. 輸入名稱中包含特殊字元
 - 回覆錯誤訊息
- d. 輸入名稱中包含空白
 - 回覆錯誤訊息

乙、 註測模組:

```
@Test
public void registertest() throws RemoteException {
    ArithmeticRMIImpl rmiTest = new ArithmeticRMIImpl();
    assertEquals("Already have the same username.", rmiTest.register("AAAA"));
    assertEquals("Already have the same username.", rmiTest.register("QQQQ"));
    assertEquals("Already have the same username.", rmiTest.register("AAA"));
    assertEquals("Already have the same username.", rmiTest.register("AAQasdasd132A"));
    assertEquals("It's illegal.", rmiTest.register("AAQA+")( "$w"));
    assertEquals("It's illegal.", rmiTest.register("AAQA "));
    assertEquals("It's illegal.", rmiTest.register(" AQA"));
    assertEquals("It's illegal.", rmiTest.register("AAQ fA"));
    assertEquals("It's illegal.", rmiTest.register("AAQ***A"));
}
```

其中測試案例針對:

- a. 輸入已經註冊的會員名稱
 - 回覆已存在的訊息
- b. 輸入未註冊的名稱
 - 回覆成功訊息並回傳專屬的 port
- c. 輸入具有特殊字元的名稱
 - 回覆錯誤訊息
- d. 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

丙、 主題模組:

```
@Test
public void createtest() throws RemoteException {
    ArithmeticRMImpl rmiTest = new ArithmeticRMImpl();
    assertEquals("creat subject success.", rmiTest.create("ADSD", "aaasd", "WERTYUIO"));
    assertEquals("creat subject success.", rmiTest.create("ADSD", "", "WERTYUIO"));
    assertEquals("creat subject success.", rmiTest.create("ADSD", "aaasd", ""));
    assertEquals("creat subject success.", rmiTest.create("ADSD", "aaasd", " WERTYUIO"));
    assertEquals("creat subject success.", rmiTest.create("ADSD", "aaasd", "WERTYUIO "));
    assertEquals("creat subject success.", rmiTest.create("ADSD", "aaasd ", "WERTYUIO"));
    assertEquals("creat subject success.", rmiTest.create("ADSD", " aaasd", "WERTYUIO"));
}
```

- a. 輸入正常新增格式
 - 回覆新增成功的訊息
- b. 輸入缺少主題名稱的案例
- c. -回覆新增成功的訊息
- d. 輸入缺少主題內容的測試案例
 - 回覆新增成功的訊息
- e. 輸入主題名稱包含特殊字元
 - 回覆新增成功的訊息
- f. 輸入主題內容包含特殊字元
 - 回覆新增成功的訊息

丁、 回覆模組:

```
@Test
public void replytest() throws RemoteException {
    ArithmeticRMIImpl rmiTest = new ArithmeticRMIImpl();
    assertEquals("", rmiTest.reply("AAAA", 1, "OK"));
    assertEquals("", rmiTest.reply("AAA", 1, "OK"));
    assertEquals("", rmiTest.reply("AAAA", 1, ""));
    assertEquals("", rmiTest.reply("", 1, "OK"));
    assertEquals("subject isn't exist.", rmiTest.reply("AAAA", 0, "OK"));
}
```

- a. 輸入正常回覆
- b. 輸入缺少回覆人的名稱
- c. 輸入缺少回覆內容
- d. 輸入為存在編號的主題

-回覆錯誤

壹拾貳. 遇到的困難:

- 甲、 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，當初要做事件處理的視窗功能，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。
- 乙、 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。
- 丙、 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。

壹拾參. 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。