請到此處觀看markdown

聊天室

目錄結構

```
/
  client/
  server/
```

環境建置

源碼

客戶端採取 ruby ，伺服器端採用 scala

ruby 相關

- 請採用 ruby2.0 以上版本若為 2.1 則請先安裝`gem install bundler`再在/client底下`bundle install`以安裝ncurses庫（ruby 在 2.1 版本從標準庫拿走了 ncurses

scala 相關

- jdk8（沒有在 jdk7 底下測試，也許能動）
- sbt

資料庫

資料庫密碼被寫死進程式碼中，但我並沒有做資料庫遷移的腳本。所以請自行 立table如下。 或者進入 server/src/main/scala/ChatRoomServer.scala 14行進行修改 database名字為 chatRoom user=chatRoom password=xxxxx

並建立table

```
CREATE table file (url varchar(128), filename varchar(128));
CREATE table Members (username varchar(128), password varchar(128), unique (username));
```

運行

1. 伺服器端 在 /server 下執行 sbt run [host] [port] 以運行伺服器
2. 客戶端 在 /client下執行ruby register.rb [host] [port] 以註冊 在 /client下執行ruby client.rb [host] [port] 以進入聊天室

設計

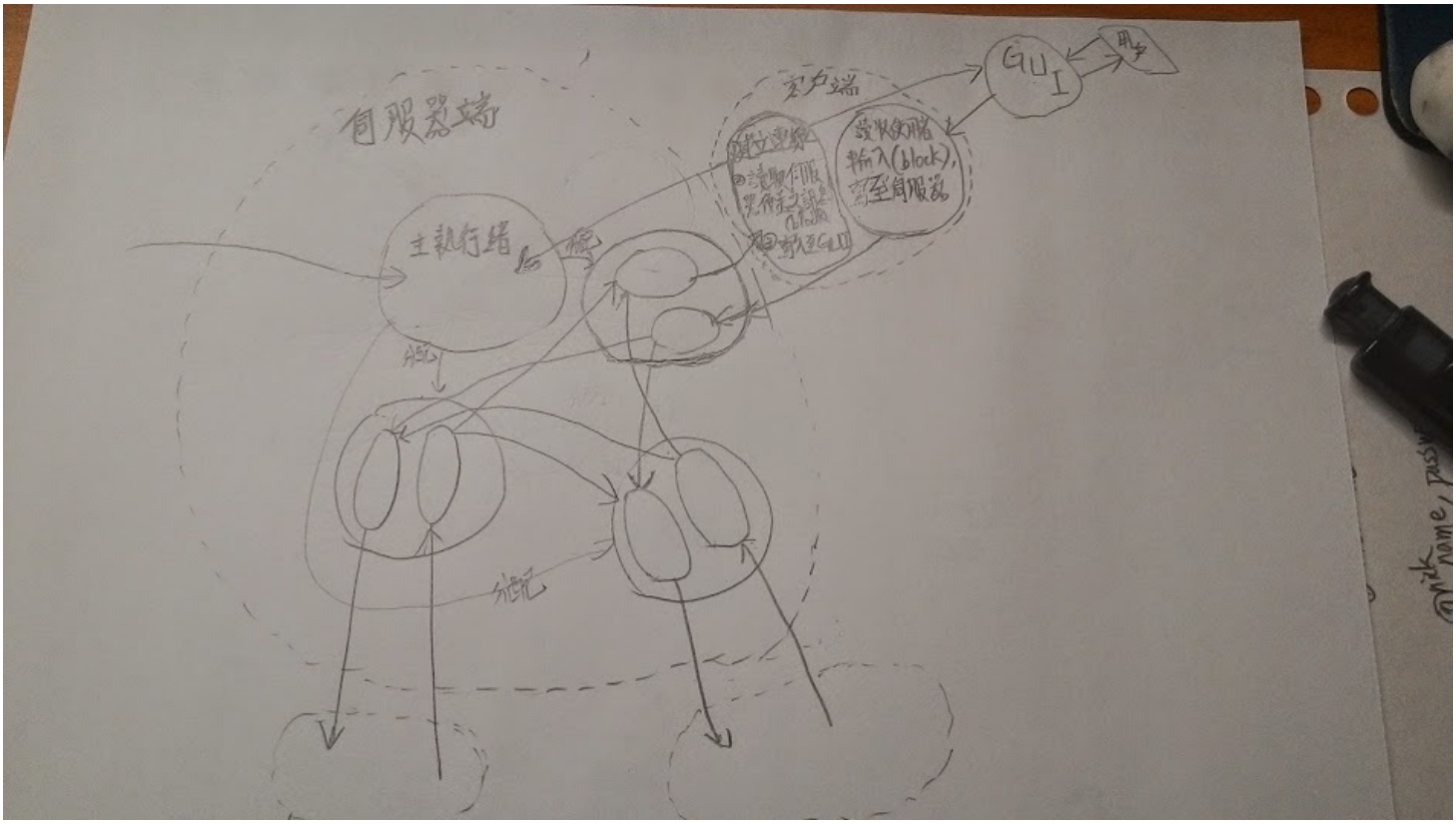
帳密系統

以mysql作為資料庫 如一般網站雷同，僅存hash過的密碼

檔案傳輸

1.上傳 上傳後即存於資料庫，並給定一hash碼（用以識別） 2.下載 檔案傳輸僅傳輸一個識別碼，點擊後再開啟一socket與之傳輸。 類似於dropbox連結

流程圖



協定

請求

註冊

```
{"function": "register", "data": {"username": "XXX", "password": "YYY"}}
### 登入
``` javascript
{"function": "login", "data": {"username": "XXX", "password": "YYY"}}
```

發訊

```
{"function": "message", "data": {"message": "XXXXX"}}
```

傳檔預備

```
{"function": "upload", "data": {"filename": "XXXXX"}}
```

傳檔

```
password            //首行為密碼
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXX
// 送到檔案結束socket關掉為止
```

下載

```
{"function": "download", "data": {"filename": "XXXXX", "url": "XXXX"}}
```

回應

準備傳檔

```
{"function": "upload", "data": {"filename": "XXXXX", "port": XX, "password": "XXXXX"}}
// 在伺服器端產生密碼以避免被攻擊
```

有人發訊

```
{"function": "message", "data": {"message": "XXXXX"}}
```
```

遇到的困難

主要在多執行緒，伺服器端要保持資料同步，是很大費周章的問題。