Notes

- To create entities of the system namely the hub, client, command and message
- Identify the falling/offline tcp connections and mark them inactive
- Same user_id may get assigned to multiple connections, lets avoid that by some mutex
- Have a loop mechanism to hold the users tcp connection to let him enter the 3 commands I am supposed to design
- Dead clients appearing in the active list - gotta do something about it
- A bug possiblity - what if the relay message is sent my myself or sent to a reciever multiple times
- Testing in scope - unittest, component and performance benchmarks - may have to restructure classes to make it testable
- Can we do docker the whole project ?
- The relative paths are ugly and bad practise, plan to remove it

Structure

Client

user_id : This integer id will uniquely identify the user in the network
handler : net.Conn object which will
incoming : channel of RelayMessage which is supposed to go to the user
timestamp : timestamp of creation of the user
active : dunno now
history : maybe have this as all the commands executed by that particular user
SendMessage() -> Input: RelayMessage Output: Should add the message to the channel which the user may be listening to

Network

assignNodeAddress() -> Output: user_id (integer)
getAllCurrentActive() -> Input: user_id/connection (string/list)
addClientToNetwork() -> Input: Conn.Net. Output: Success, create a new client object and add it to the network

RelayMessage

message: Stores the message(json/string/bytes/anything)
from: Identity of the user who sent the message
receiptClients : Contains the list of user_id which will receive the message

```
├── Entities
|   ├── Client.go
|   ├── Client_test.go
|   ├── Network.go
|   ├── Network_test.go
|   ├── RelayMessage.go
|   ├── RelayMessage_test.go
|
├── README.md
|
├── RelayServer
|   └── RelayServer.go
|
├── Settings
|   └── config.go
|
├── Utils
|   ├── Utils.go
|   ├── Utils_test.go
|
├── config.json
├── main.go
|
└── tests
    └── mocks
        ├── client.go
        └── conn.go
```

Hub setup

```
$go run main.go
```

or

```
$go build
$./relay_solution
```

Client connection

```
$nc localhost 6666
```

Client usage

```
>> IDENTIFY
>> LIST
>> RELAY #Message i want to send, its a lame way but lets do this #1,3,4,10
>> EXIT
```

Unittest

```
$ go test $(go list ./...| grep -v test) -coverprofile cover.out; go tool cove
```

or go to the respective folder

```
$ go test -coverprofile cover.out; go tool cover -func cover.out
```

Unittests Output

```
$ sudo go test $(sudo go list ./...| grep -v test) -coverprofile cover.out; su
?       relay_solution  [no test files]
ok      relay_solution/Entities 4.032s  coverage: 100.0% of statements
?       relay_solution/RelayServer  [no test files]
?       relay_solution/Settings [no test files]
ok      relay_solution/Utils    0.017s  coverage: 100.0% of statements
relay_solution/Entities/Client.go:28:       GetUserId           100.0%
relay_solution/Entities/Client.go:32:       GetActive           100.0%
relay_solution/Entities/Client.go:36:       SetActive           100.0%
relay_solution/Entities/Client.go:40:       SendMessage         100.0%
relay_solution/Entities/Client.go:44:       AddToHistory            100.0%
relay_solution/Entities/Client.go:48:       ReceiveMessages         100.0%
relay_solution/Entities/Network.go:20:      assignAddressToNode     100.0%
relay_solution/Entities/Network.go:25:      Register            100.0%
relay_solution/Entities/Network.go:34:      GetUserIdByConnection       100.0%
relay_solution/Entities/Network.go:39:      GetClientById           100.0%
relay_solution/Entities/Network.go:44:      GetActiveClients        100.0%
```

```
relay_solution/Entities/Network.go:65:      SendRelayMessage        100.0%
relay_solution/Entities/Network.go:78:      RemoveClientByConnection    100.0%
relay_solution/Entities/Network.go:92:      NewNetwork          100.0%
relay_solution/Entities/RelayMessage.go:15: ValidateMessageLength       100.0%
relay_solution/Entities/RelayMessage.go:19: ValidateRecieverCount       100.0%
relay_solution/Entities/RelayMessage.go:23: CreateRelayMessage      100.0%
relay_solution/Utils/Utils.go:10:       SendResponse            100.0%
relay_solution/Utils/Utils.go:15:       SendPrompt          100.0%
relay_solution/Utils/Utils.go:19:       SendBroadcast           100.0%
relay_solution/Utils/Utils.go:24:       PrintHelpText           100.0%
total:                      (statements)            100.0%
```