



C# 팀 명: LAL  
(lost One leg)

최광현  
이대건  
박재홍

# 목차

## Introduction

선정 배경



## Body

구현

UI & 동작도



## Conclusion

소감

질문



# Loading

Loading...



Loding.. 33%

# login



ID

Password

Log in

Sign in

# Sign in

URL Sign\_In



*ID*

*Name*

*PW*

*Sign in*

Cancel

# Mentor selection

Main

×

(↵ · \_ ·) legendary mentor

select 9(① · 3 - ①) ♡ Up key



reject (。 · ~ ~ · 。) Down key

# JSON

```
Var Json = {  
    "Key1" : "Value1",  
    "Key2" : "Value2",  
    "Key3" : "Value3"  
}
```

# ChatRoom


Cover

Chat Room

## 빌 게이츠방


참여자 2명

Mentor



빌 게이츠

Mentee



박재홍

—> [박재홍]제가 파일 하나 보냈어요  
<— [최광현]잘 됩니다.  
—> [박재홍]저거로 수정해주세요  
—> [박재홍]카톡이 메아리되는게 아니라  
—> [박재홍]그냥 음원 파일이 카톡 두번이예요



# Client server communication

*Client*

<Login>

<Register>

<SelectMento>

<Tab>

<Chat>

<Exit>

Request



*Server*

<Login>

<Register>

<SelectMento>

<Tab>

<Chat>

<Exit>

Answer



# Client server - Login

*Client*

request:Login,  
ID:{ID\_txt.Text},  
PW:{Pw\_txt.Text}

Request



Answer

*Server*

로그인 정보가 Server에  
저장되어 있는지 확인

<성공>

Answer:Login,  
Username:username,  
Data:접속되었습니다.

<실패>

Answer:Login,  
Data:(등록되지 않은  
사용자입니다. ||  
잘못된 비밀번호입니다.)

# Client server - Register

*Client*

회원가입 정보를 입력

request:Register,

ID:id,

Name:name,

PW:pw

Request



*Server*

Client에게 받은  
회원가입 정보를  
파일에 출력

Answer



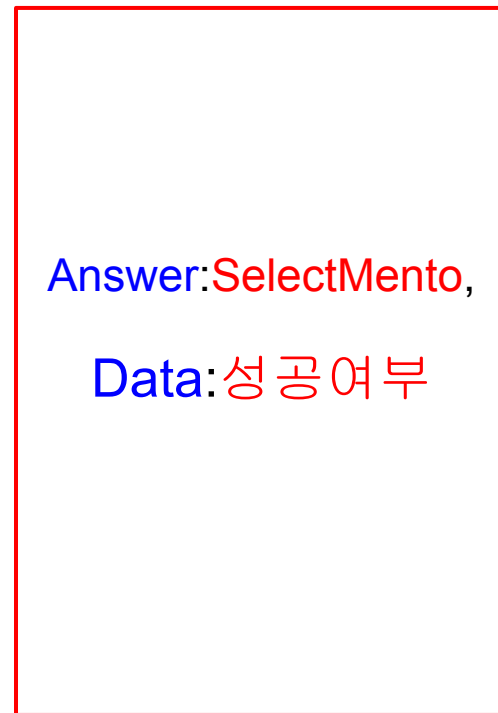
# Client server - SelectMento

*Client*

*Server*



Request



Answer



# Client server - Tab

*Client*

User가 탭을 변경

request:Tab{PageIndex}

Server로부터  
해당 Tab에 출력할  
정보들을 받아  
화면에 그려줌

Request



Answer

*Server*

Answer:Tab{PageIndex},

ChatRoomName:roomname,

Lines:linenum,  
[ChatLog:log,]

TotalMentee:menteeCnt,  
[MenteeName:name,  
MenteeImgPath:path,]

MentoName:name,  
MentoImgPath:path

# Client server - Chat

*Client*

User 채팅

`request:Chat,`  
`Data:ChatData`

Server의 답변을 받는다

User가 본인이면  
“`--> Data`”를 화면에 출력

User가 다른 유저이면  
“`<-- Data`”를 화면에 출력

Request



Answer

*Server*

채팅정보를 받고  
request를 보낸 Client가  
포함된 채팅방의  
Client들에게 전송  
`Answer:Chat,`  
`User:user,`  
`Data:chatdata`

# Client server - Exit

## Client

프로그램을 종료할 때  
Server에게

request:Exit 전송

Answer:Tab을 받은  
Client들은  
Tab화면의 내용을 갱신

Request



Answer

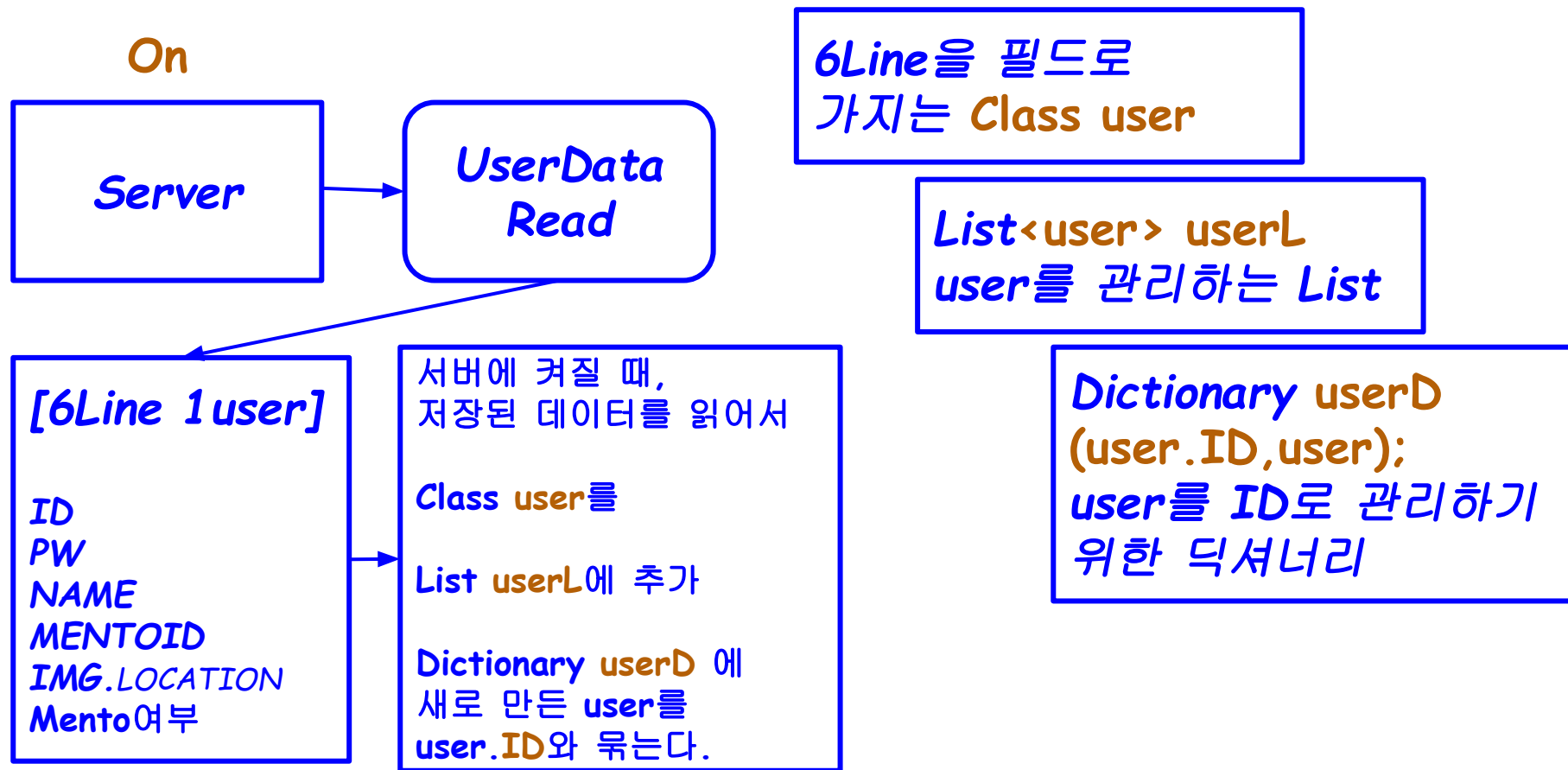
## Server

관리하는 Client 대상에서 제외  
Exit을 전송한 User의 멘토에  
포함된 Client에게

Answer:Tab

형 답변을 전송하여  
같은 채팅방 유저의 화면을  
갱신하도록 함

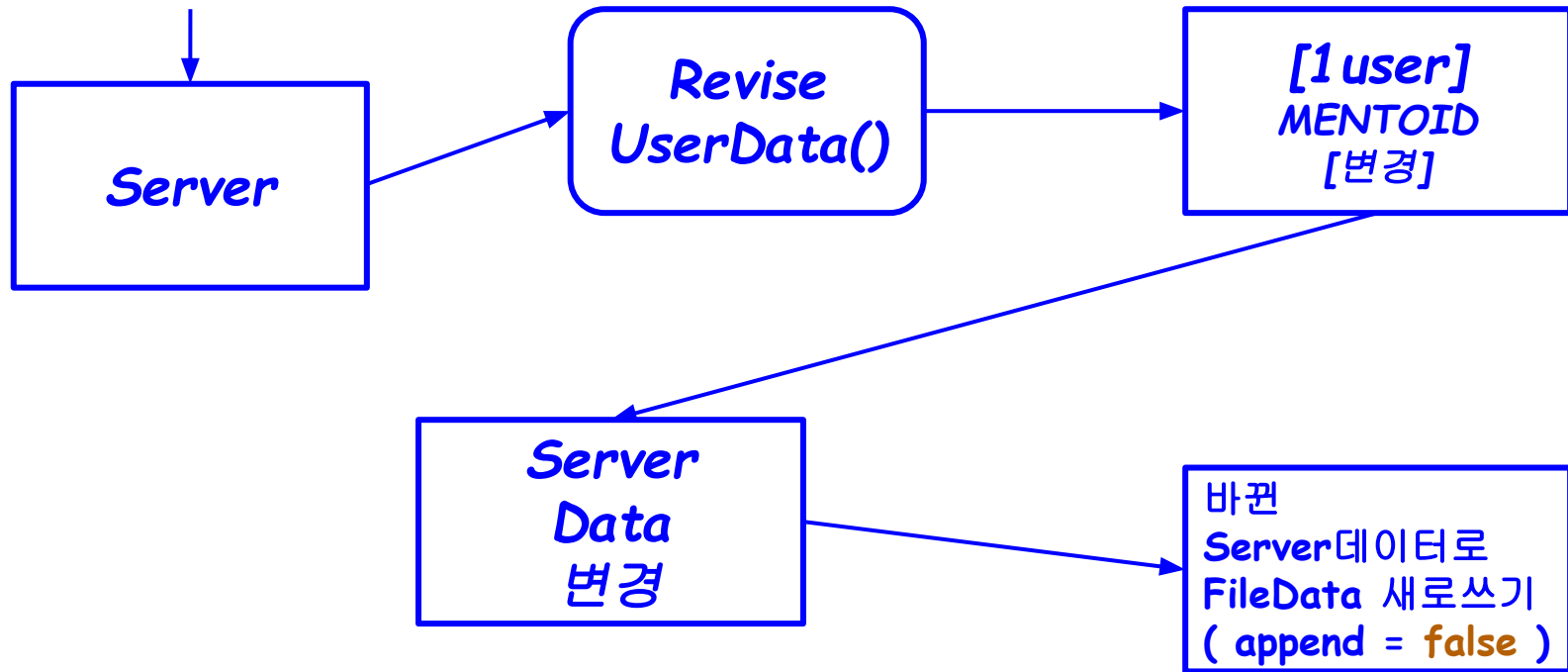
# File database - On,Revise



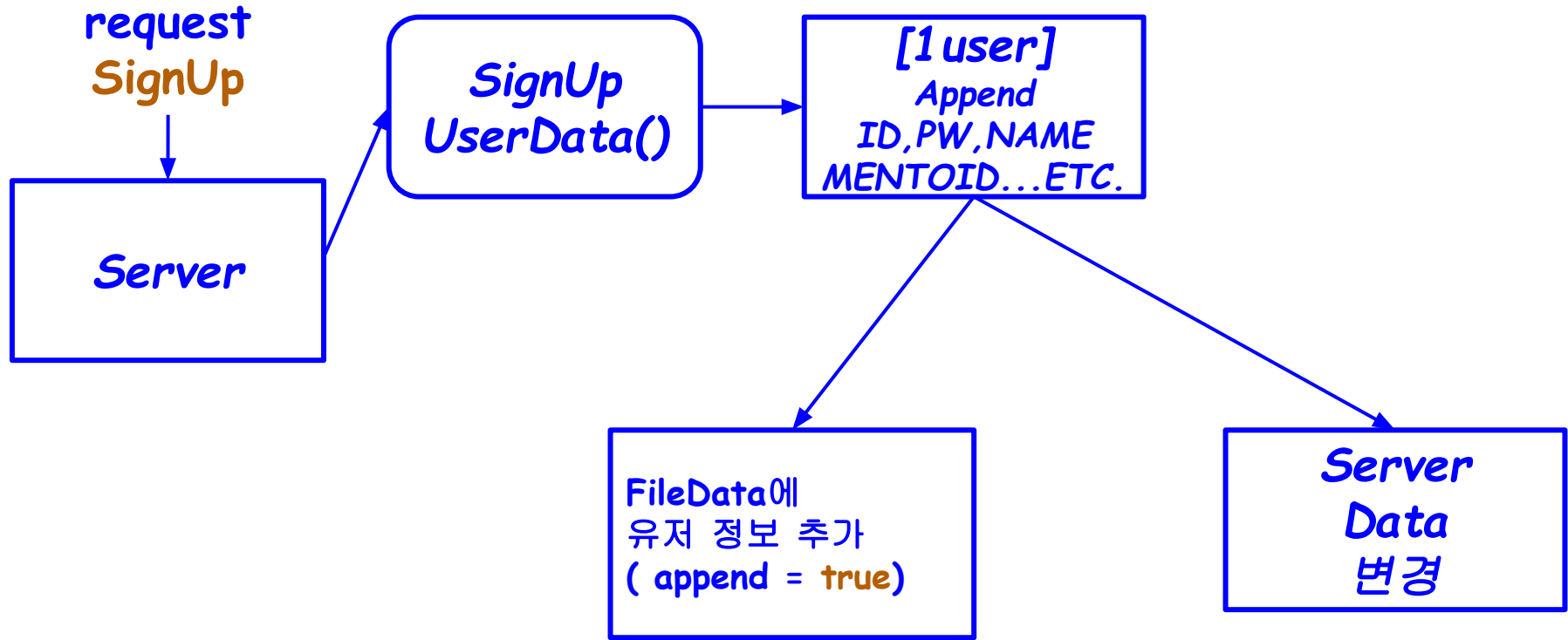


# File database - On,Revise

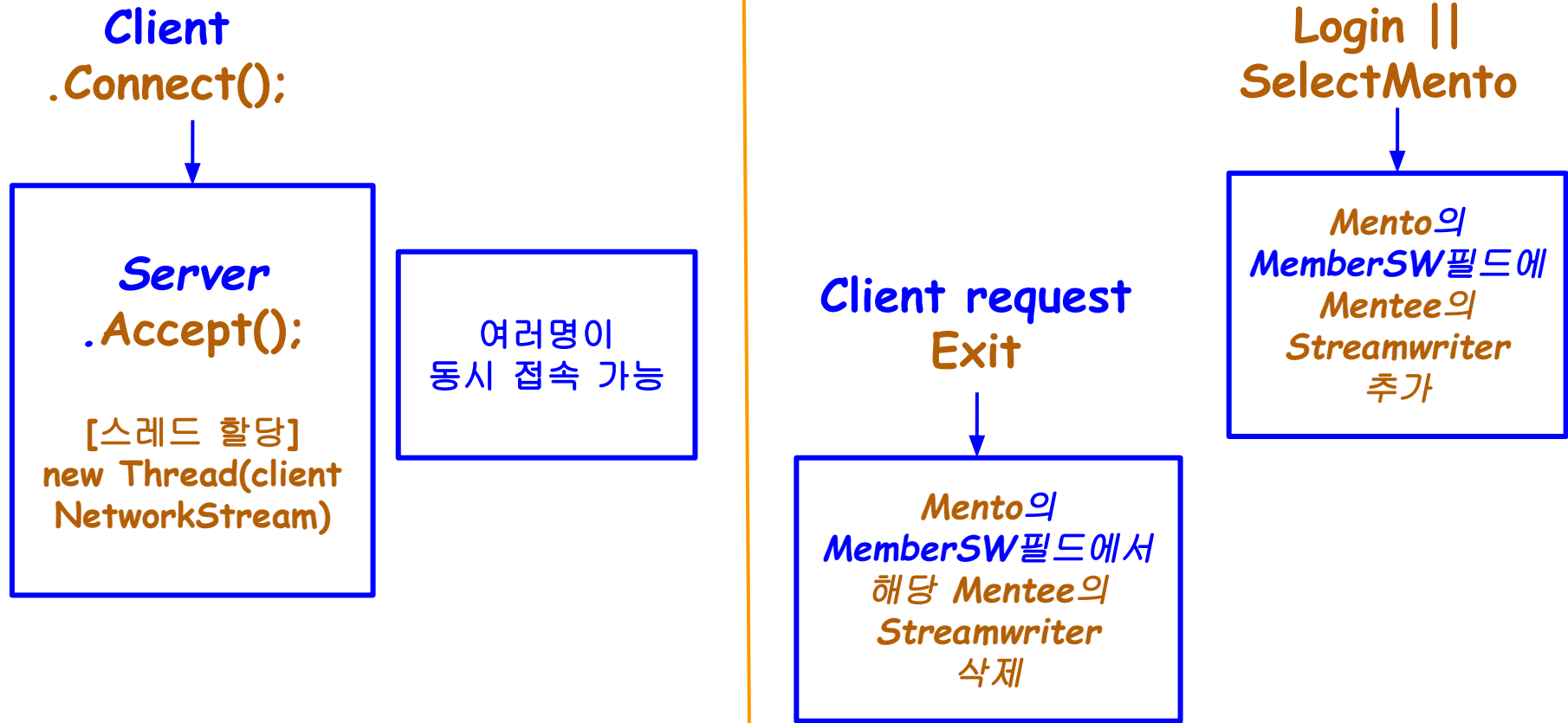
request  
SelectMento



# File database - Sign Up



# Multi client management



# Multi client management

Client request  
Chat...etc

[Mento]  
List<Streamwriter>  
MemberSW

foreach  
(StreamWriter  
in  
MemberSW)

MemberSW필드에서  
Mentee들의  
StreamWriter를  
꺼내서  
데이터를 보낸다.

"빌"을 Mento로 가진  
Mantee "a"가  
행동하면,  
Mento가 "빌"인  
모든 Mentee들은  
"a"의 행동과 관계된  
데이터를 서버로부터  
송신받는다.

즉 Mento가 동일한  
Client들은 같은 그룹  
Mentee들의 행동을  
알 수 있다.

user  
ID  
PW  
NAME  
MENTOID

Dictionary  
userD  
(user.ID,user);

userD  
(userD(user.ID).  
MentoID).Membe  
rSW;

userD에 Mantee"a"의 ID를 사용해 "a"의  
user(Class)를 호출. [userD("aID").MentoID]  
user "a"에 저장된 MentoID"빌"을 불러와  
Mento에 저장된 MemberSW를  
호출한다.userD("빌").MemberSW.

소감



Q & A