

## [우수 계획서 참조용]기초 프로젝트2 계획서

프로젝트 명	국문	클라이언트의 동적 관리로 메모리의 효율을 높인 GUI 기반의 멀티 프로세스 채팅 프로그램			
	영문	GUI-based multiprocessor chat program that increases memory efficiency with dynamic management of clients			
수 행 기 간	2019 . 11 . 02 ~ 2019 . 11 . 11				
참 여 자	소속	참여자성명	학년	학번	이메일
	컴퓨터공학부	홍XX	2	20XXXXXXXXXX	-

### 1. 개발내용 요약

#### 가. 개발 목표

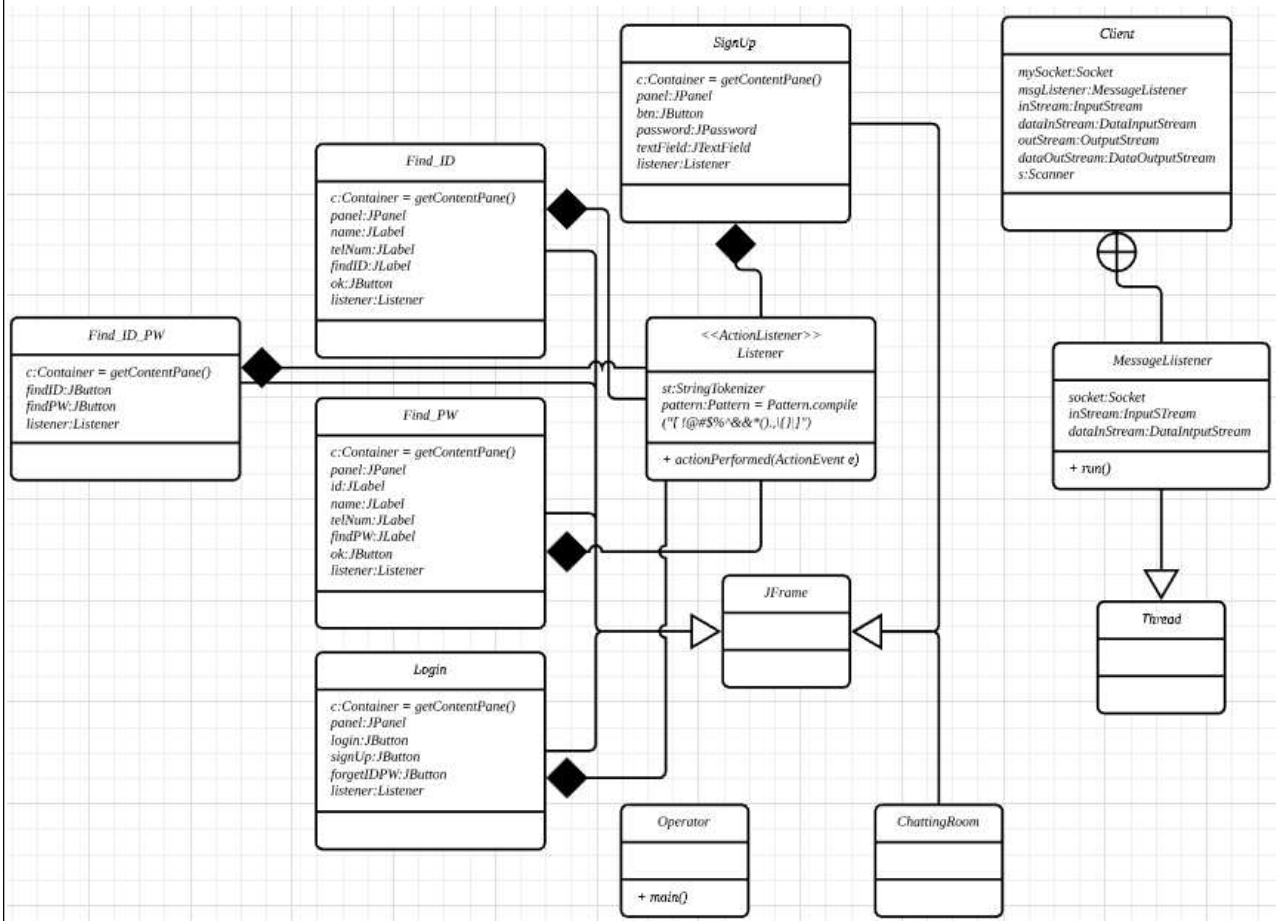
- 여러 기업들의 메신저 애플리케이션이 많이 이용되고 있지만 상업적인 요소가 많이 들어있어 불편함을 느낌. 상업적인 요소를 줄이고 친절한 인터페이스를 통해 사용자에게 좀 더 쾌적한 프로그램이 되도록 나만의 알고리즘을 사용하여 경쟁력 있는 채팅 프로그램을 만들어 보려 함

#### 나. 개발 내용 개요

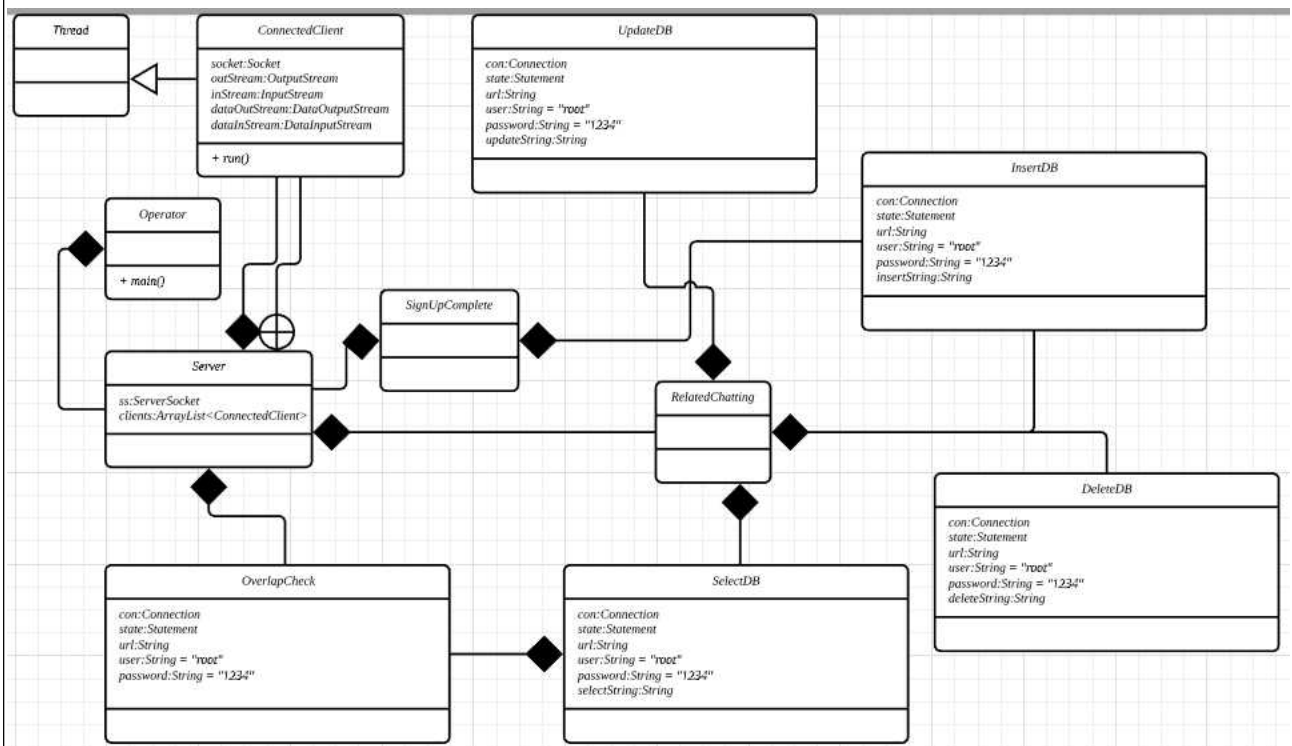
- 개발물에 대한 개요 설명 -
  - > 클라이언트와 서버간의 소켓을 이용한 채팅 프로그램
  - > DB를 연동하여 회원 정보 등을 저장
  - > 클라이언트를 ArrayList클래스를 이용하여 동적으로 관리
  - > 휘발성 채팅방으로 메모리의 부담을 줄임

- Class Diagram -

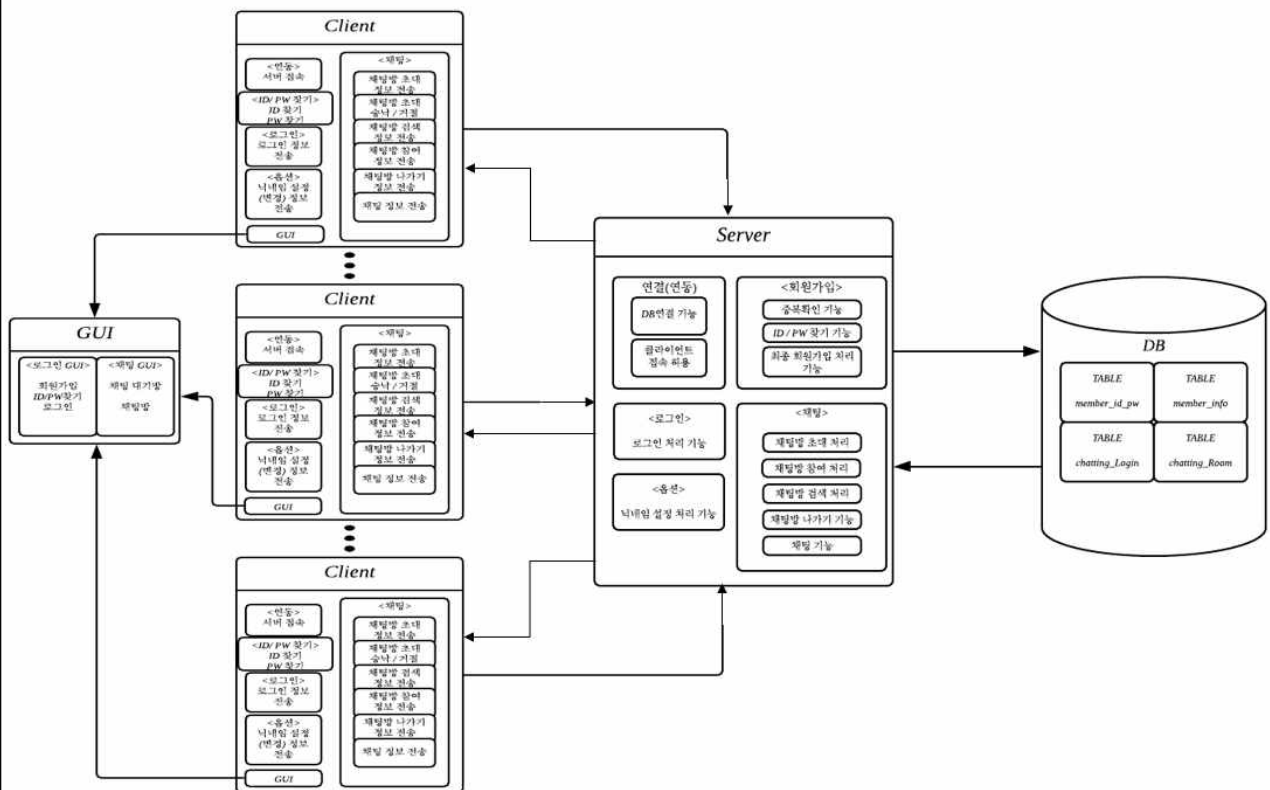
<클라이언트 클래스 다이어그램>



<서버 클래스 다이어그램>



- 시스템 구성도 -



## 2. 상세 개발 사항

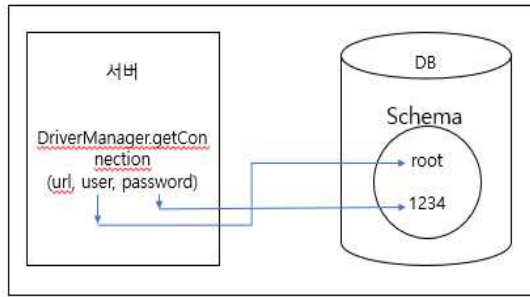
### 가. 상세 개발 내용

#### <서버 측 상세 개발 내용>

##### 1) DB 연결 기능

: 서버에서 사용자들의 여러 정보를 저장하는 DB에 접근하기 위한 연결 기능 개발

- DBMS로 MYSQL을 사용할 것임
- JDBC를 프로젝트 라이브러리에 추가하여 사용



- DB에서는 root, 1234(사용자, 비밀번호)의 스키마를 만들고 ip로 localhost를 port로 기본포트(3306)를 사용

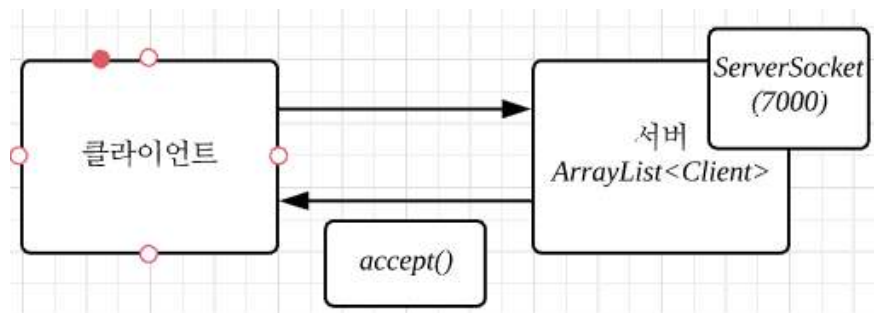
- 이후 Connection클래스 타입의 변수를 만들어 DB의 URL(ip, port를 포함), 사용자, 비밀번호를 매개변수로 하는 DriverManager.getConnection()메소드로 DB에 연결

\* 클라이언트는 DB에 직접 접근할 수 없고 오직 서버를 통해서만 가능

## 2) 클라이언트 서버 접속

: 클라이언트가 서버에 접속할 수 있도록 접근 허가 기능을 개발

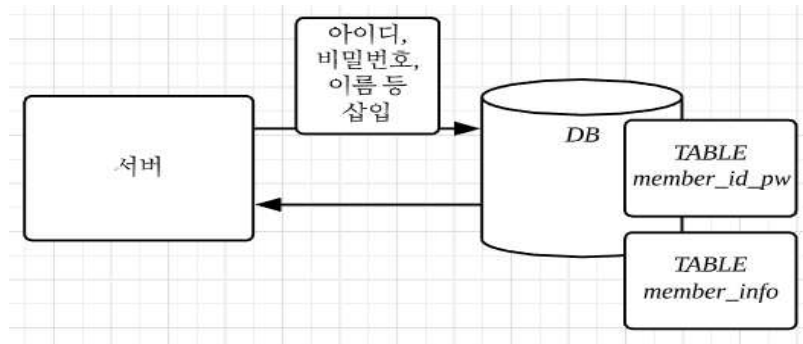
- 클라이언트와 서버는 Socket을 통해 송수신함
- 서버는 7000번 포트를 사용하는 ServerSocket 객체 생성, 클라이언트는 서버의 IP와 7000번 포트를 매개변수로 하는 Socket 객체 생성
- 클라이언트가 접근을 요청할 때, 서버는 accept()메소드를 사용해 접근을 허가하고 해당 클라이언트를 스레드로 동작시킴
- 클라이언트가 얼마나 서버에 접속할지 모르기 때문에 서버는 ArrayList클래스를 이용해 클라이언트 객체들의 메모리를 동적으로 관리할 수 있도록 함



## 3) 회원가입 처리 기능

: 클라이언트에서 전송하는 회원가입 정보를 DB에 저장

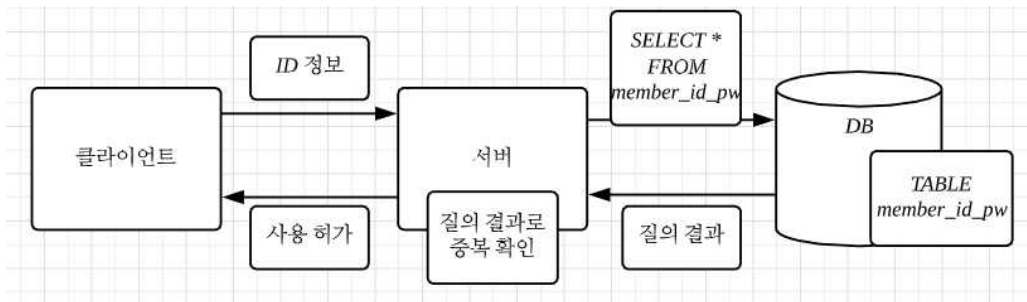
- 클라이언트에서 전송할 때 회원정보라는 것을 알려주기 위해 문자열 앞에 "SU:"을 붙인다는 것을 인지해야함
- 클라이언트에서 전송한 문자열 형식의 회원 정보를 StringTokenizer를 이용해 분리 후 DB 테이블에 알맞게 저장
- 서버에서는 데이터삽입 시 항상 중복되는 INSERT문의 앞부분(INSERT INTO 테이블명 VALUES)을 문자열 변수로 선언하고, 클라이언트에서 회원가입 요청이 들어올 때마다 사용할 수 있도록 함



#### 4) 아이디 중복확인 처리 기능

: 클라이언트에서 전송하는 아이디에 대하여 DB에 질의 후 아이디 사용 허가 여부를 클라이언트로 전송

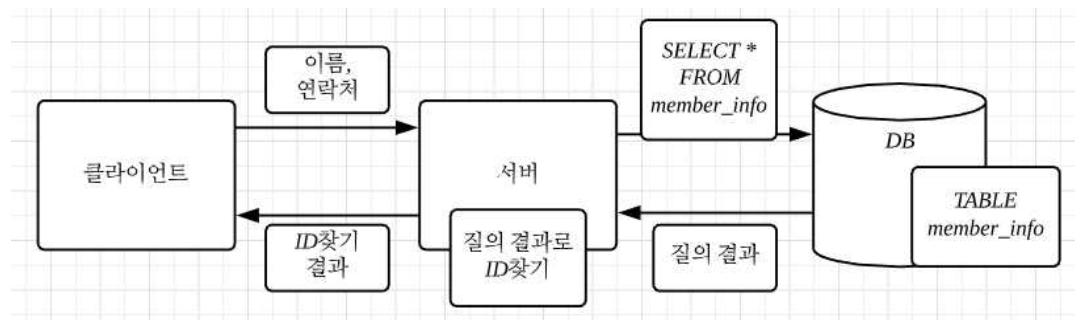
- 클라이언트에서 전송한 문자열의 앞글자가 "OL(Overlap):"이면 ID의 중복 여부를 DB에 질의할 수 있도록 함
- 데이터검색 시 항상 중복되는 SELECT문의 앞부분(SELECT \* FROM 테이블명)을 문자열로 선언해 놓고, 클라이언트에서 중복확인을 요청할 때마다 사용할 수 있도록 함
- 질의의 결과 릴레이션을 한 줄씩 읽어 아이디 중복을 확인
- 아이디 중복 여부(사용 허가 or 사용 불허)를 클라이언트로 전송
- 전송할 때 문자열의 앞에 "OL:"을 붙여 클라이언트가 아이디 중복확인의 결과라는 것을 인지할 수 있도록 함



#### 5) 아이디 찾기 기능

: 클라이언트에서 전송하는 사용자 정보에 대하여 DB에 질의 후 결과 도출

- 클라이언트에서 전송한 문자열의 앞 두 글자가 "FI(FindID):"이면 아이디 찾기에 대해 DB에 질의할 수 있도록 함
- 아이디 찾기 요청이 들어오면 받은 정보와 선언해 놓은 SELECT문을 더해서 DB에 질의 함
- 클라이언트에서 전송한 정보와 질의 결과를 비교 후, 그 결과를 클라이언트로 전송
- 결과를 전송할 때에 문자열의 앞에 "FI:"를 붙여 클라이언트가 아이디 찾기의 결과라는 것을 인지할 수 있도록 함

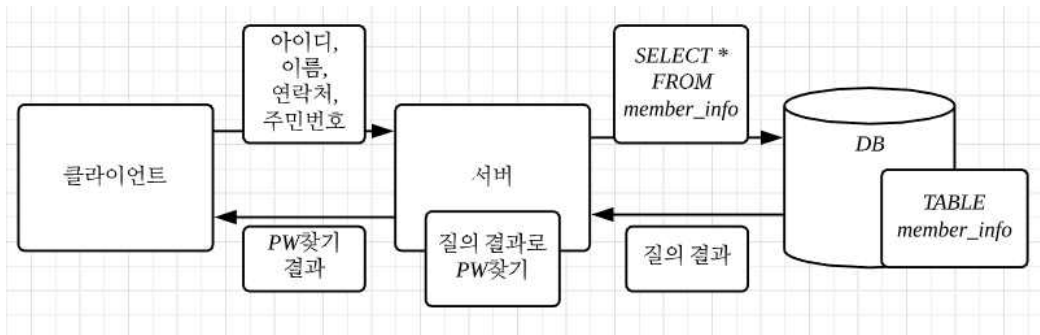


#### 6) 비밀번호 찾기 기능

: 클라이언트에서 전송하는 사용자 정보에 대하여 DB에 질의 후 결과 도출

- 클라이언트에서 전송한 문자열의 앞 두 글자가 "FP(FindPassword):"이면 비밀번호 찾기에 대해 DB에 질의할 수 있도록 함
- 비밀번호 찾기 요청이 들어오면 받은 정보와 선언해 놓은 SELECT문을 더해서 DB에 질의 함
- 클라이언트에서 전송한 정보와 질의 결과를 비교 후, 그 결과를 클라이언트로 전송
- 결과를 전송할 때에 문자열의 앞에 "FP:"를 붙여 클라이언트가 비밀번호 찾기의 결과라는 것을

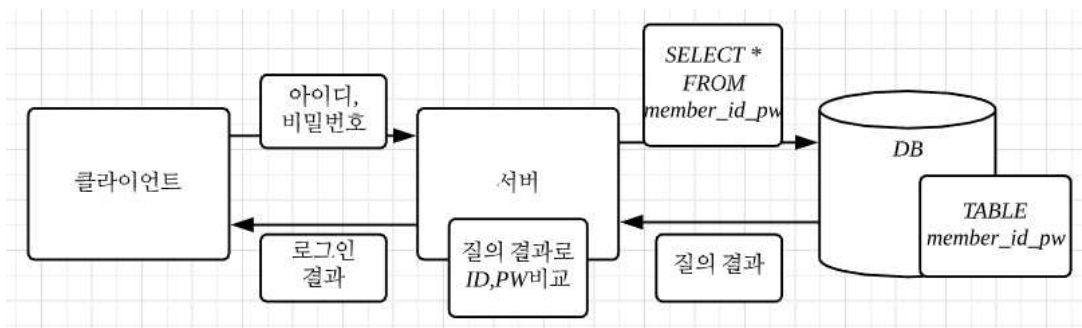
인지할 수 있도록 함



## 7) 로그인 정보 처리 기능

: 클라이언트에서 전송하는 로그인 정보에 대하여 DB에 질의 후 로그인 허용 여부를 클라이언트로 전송

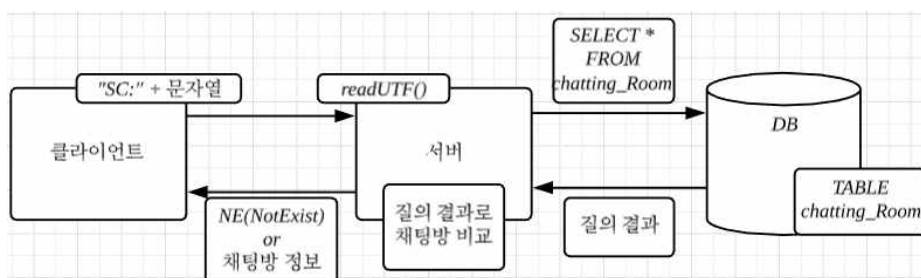
- 클라이언트에서 전송한 문자열의 앞 두 글자가 "LI(Login):"이면 로그인에 대해 DB에 질의할 수 있도록 함
- 클라이언트에서 전송한 정보와 선언해 놓은 SELECT문을 더해서 DB에 질의 함
- 클라이언트에서 전송한 정보와 질의 결과를 비교 후, 그 결과를 클라이언트로 전송
- 결과를 전송할 때에 문자열의 앞에 "LI:"을 붙여 클라이언트가 로그인의 결과라는 것을 인지할 수 있도록 함



## 8) 채팅방 검색 기능

: 클라이언트에서 전송하는 채팅방 정보에 대하여 DB에 질의 후 해당하는 채팅방 정보를 클라이언트로 전송

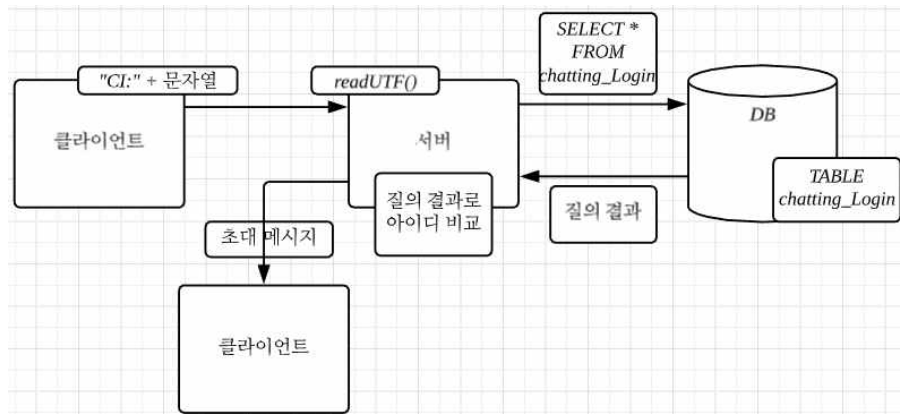
- 클라이언트에서 전송할 채팅방 검색용 메시지 타입을 정의 -> "SC:"
- StringTokenizer클래스를 사용해 메시지 타입과 채팅방 정보를 분리
- 모든 채팅방 정보를 DB에서 불러옴 -> 미리 지정해둔 SELECT 문자열을 사용함
- 질의의 결과와 클라이언트에서 전송한 채팅방 정보를 비교
- 비교한 결과 채팅방의 정보를 클라이언트 측으로 전송 -> DB에 해당 채팅방의 정보가 없다면 "NE(NotExist)" 문자열 전송



## 9) 채팅방 초대 기능

: 클라이언트에서 전송한 초대 사용자 정보에 대하여 DB에 질의 후 해당하는 사용자가 로그인 상태이면 해당 사용자에게 초대 메시지 전송

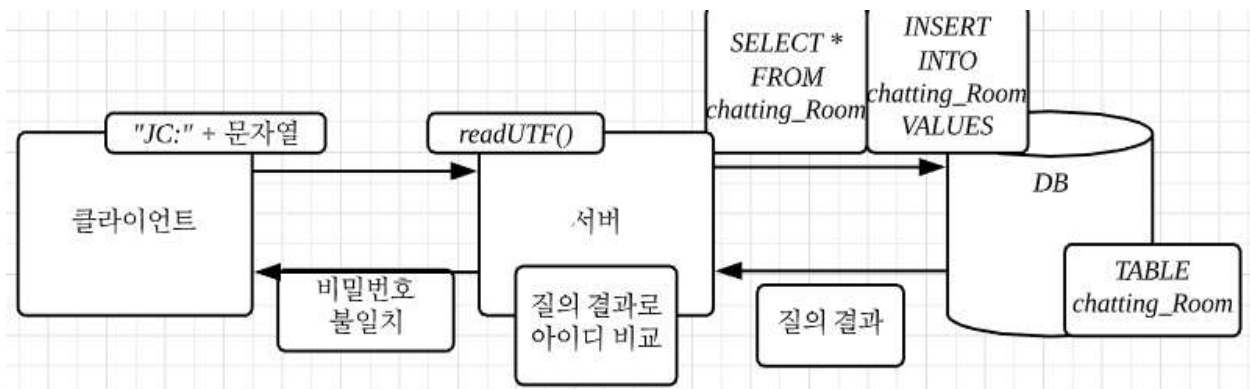
- 클라이언트에서 전송할 채팅방 초대 전용 메시지 타입을 정의 -> "CI:"
- StringTokenizer클래스를 사용해 메시지 타입과 초대할 사용자 정보를 분리
- 전체 로그인 사용자의 정보를 DB에서 불러옴 -> 미리 지정해둔 SELECT 문자열 사용
- 질의 결과와 초대를 받을 사용자의 정보를 비교
- 질의 결과 중에 초대를 받을 사용자의 정보가 존재한다면 해당 사용자의 IP 정보를 통해 초대 메시지 전송



## 10) 채팅방 참여 기능

: 클라이언트가 전송한 사용자 정보와 채팅방 정보에 대해 DB에 SELECT와 INSERT를 할 수 있도록 함

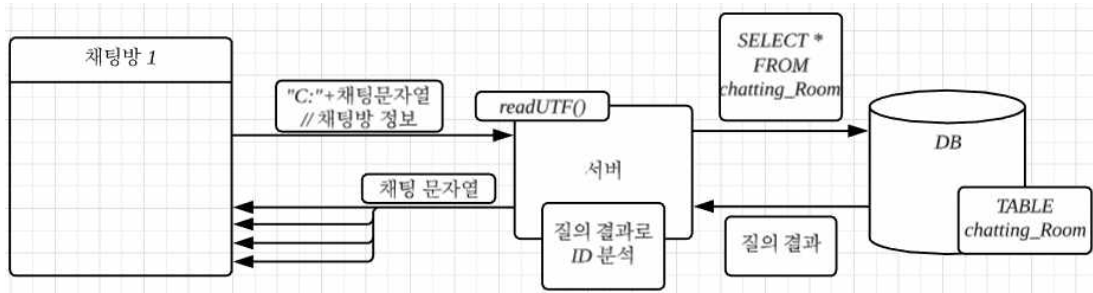
- 클라이언트에서 전송할 채팅방 참여 전용 메시지 타입을 정의 -> "JC:"
- StringTokenizer클래스를 사용해 메시지 타입과 초대할 사용자 정보를 분리
- 클라이언트에서 전송한 정보 중에 비밀번호가 있다면 SELECT 문자열을 사용해 해당 채팅방의 비밀번호가 맞는지 질의
- 사용자가 비밀번호가 설정되어 있지 않은 채팅방에 참여하거나 비밀번호가 설정된 채팅방의 비밀번호와 일치하면 해당 사용자의 정보를 채팅방 DB에 삽입 -> 미리 지정해둔 INSERT 문자열 사용
- 비밀번호가 일치하지 않다면 사용자에게 틀렸다는 메시지를 전송
- DB에 사용자의 정보가 삽입 됐다면 사용자에게 채팅방 참여를 허용하는 문자열 전송



### 11) 채팅 기능

: 클라이언트가 전송한 문자열에 대하여 채팅방에 참여하는 모든 클라이언트에게 보여지도록 재전송

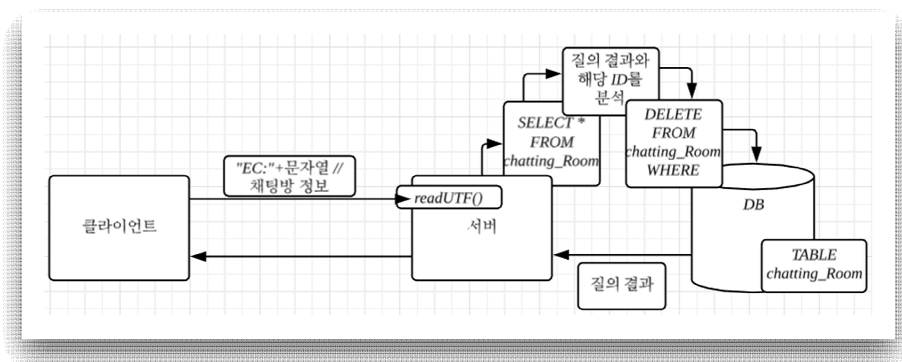
- 클라이언트에서 전송할 채팅 전용 메시지 타입을 정의 -> "C:"
- StringTokenizer클래스를 사용해 메시지 타입과 채팅 문자열, 채팅방 정보를 분리
- 해당 채팅방에 존재하는 모든 사용자에게 DB에 질의 -> SELECT 문자열 사용
- 모든 사용자의 정보를 이용해 IP를 도출하고 채팅방에 참여한 모든 IP 정보에게 채팅 문자열을 전송
- 전송 시 "C:"를 붙여서 보냄 -> 클라이언트에게 채팅 전용 문자열을 보낼을 명시



### 12) 채팅방 나가기 기능

: 클라이언트가 전송한 채팅방 정보와 사용자 정보에 대해 DB에 DELETE를 할 수 있도록 함

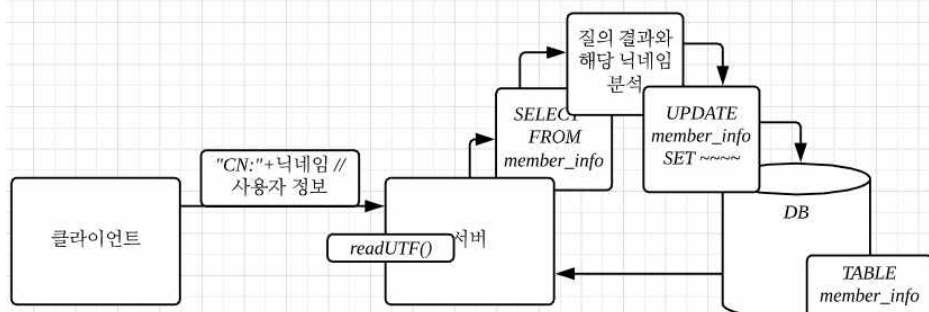
- 클라이언트에서 전송할 채팅방 나가기 전용 메시지 타입을 정의 -> "EC:"
- StringTokenizer클래스를 사용해 메시지 타입과 채팅방 정보, 사용자 정보를 분리
- 해당 채팅방 정보와 사용자 정보를 이용해 DELETE 문자열을 사용하여 채팅방을 나가려는 사용자를 채팅방 DB에서 삭제



### 13) 닉네임 설정 기능

: 클라이언트가 전송한 사용자 정보에 대해 DB에 UPDATE를 할 수 있도록 함

- 클라이언트에서 전송할 닉네임 설정 전용 메시지 타입을 정의 -> "CN:"
- 메시지 타입과, 사용자 정보를 분리
- 해당 사용자 정보를 이용해 UPDATE 문자열을 사용하여 닉네임을 설정(변경)하려는 사용자 DB에서 닉네임 필드를 재설정
- 변경이 끝난 후에는 클라이언트 측으로 정보를 전송해 인터페이스 부분이 업데이트될 수 있도록 함



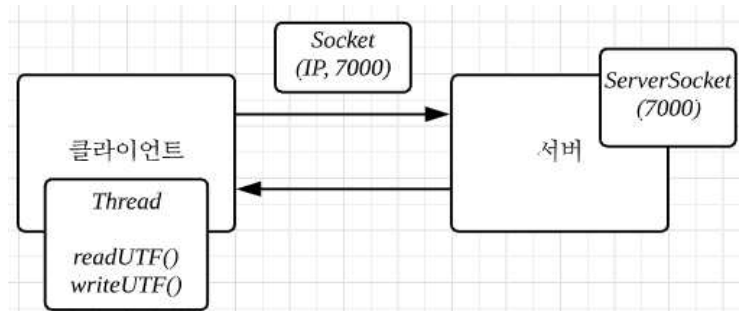


### <클라이언트 측 상세 개발 내용>

#### 1) 서버 접속

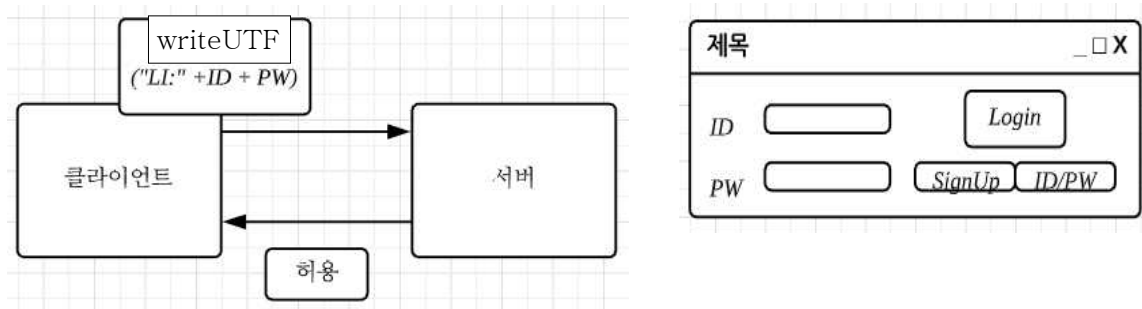
: 서버는 들어온 모든 클라이언트에 대해 접속을 허용하기에 클라이언트는 소켓을 통해 접속 요청만 하면 됨

- 클라이언트는 `DataInputStream`과 `DataOutputStream`을 사용해 서버와 데이터를 주고 받을 상시 대기 상태의 스레드 객체 생성
- 클라이언트는 서버의 IP와 서버의 Port번호를 매개변수로 갖는 소켓 생성자를 통해 서버에 접속을 요청



#### 2) 로그인 인터페이스와 로그인 기능

- 로그인 버튼을 누르면 ID와 PW 텍스트필드에 입력된 정보를 `getText()`메소드로 얻음
- 얻은 문자열을 합치고 문자열 앞에 "LI:"을 붙여서 `DataOutputStream`의 `writeUTF()`메소드를 통해 서버로 전송
- 서버가 로그인을 허가하면 로그인 창을 끄고 채팅 대기 화면을 보여줌

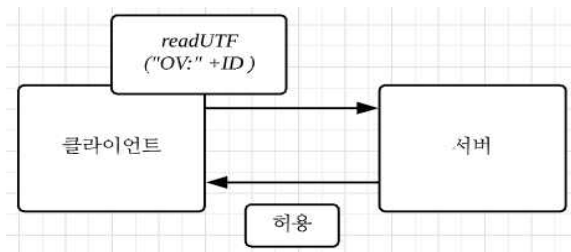


#### 3) 회원가입 인터페이스

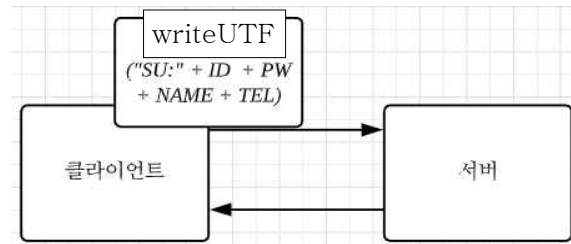
: 아이디, 비밀번호, 이름 등을 사용자에게 입력받음 / ID의 중복확인, 비밀번호 재입력 등으로 사용자의 실수와 중복 제거

- 사용자는 텍스트필드에 회원가입 정보를 입력함

(중복확인)



(회원가입 확인)



제목

ID

Overlap

PW

재입력

이름

전화번호

확인

(중복확인)

- ID에는 특수문자가 들어갈 수 없게 특수문자가 있는지 확인 기능이 필요
- 중복확인(Overlap)버튼을 누를 시 ID 텍스트필드에 입력된 정보를 getText()메소드로 받아오고 문자열의 앞부분에 "OV:"를 붙여 서버로 전송
- 서버에서 아이디 사용을 허용하면 필드로 갖고 있던 boolean값을 변경하고 회원가입 진행

(회원가입 확인)

- 1) 확인 버튼을 누를 시 ID 중복확인의 boolean값을 먼저 확인
  - 2) PW와 PW 재입력란의 정보를 equals()메소드를 사용해서 같은 문자열인지 확인
  - 3) 이름은 특수문자나 숫자를 가질 수 없도록 확인
  - 4) 전화번호는 숫자 이외의 값을 가질 수 없도록 확인 - 예외처리 필요
- 위 4가지 사항을 모두 만족하는 정보는 문자열의 앞에 "SU:"을 붙여 서버로 전송

#### 4) 채팅 대기방 인터페이스

제목

채팅방 이름

방장 // 인원 수

채팅방 이름

방장 // 인원 수

채팅방 이름

방장 // 인원 수

닉네임 : 000

ID : -----

프로필 변경

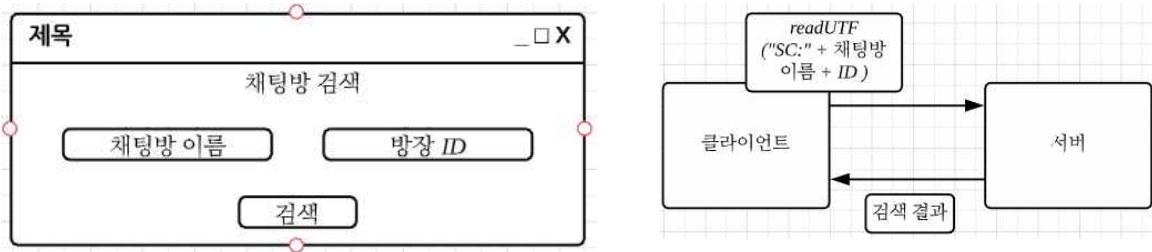
로그아웃

Search chatting

Create chatting

#### 4-1) 채팅방 검색 인터페이스

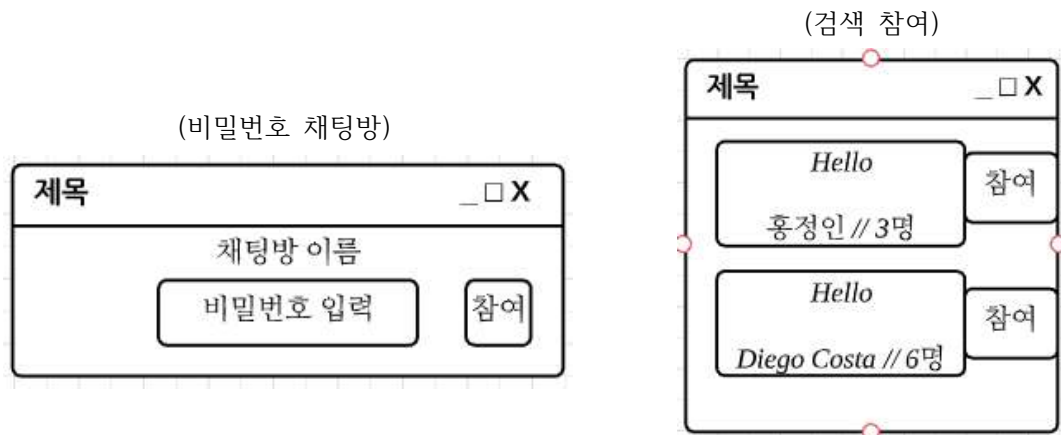
: 채팅방 검색 인터페이스에는 채팅방 이름과 방장 ID를 입력할 수 있음



- 채팅방 이름이나 방장 ID 둘 중 하나는 빈칸으로 둘 수 있음
- "검색" 버튼을 눌렀을 때 둘 다 빈칸일 시에는 JOptionPane으로 경고 메시지를 출력
- 그 이외에는 버튼을 눌렀을 때 채팅방 이름과 방장 ID 문자열을 getText()로 얻은 후 앞에 "SC:"를 붙여 서버로 전송

#### 4-2) 채팅방 참여 인터페이스

: 비밀번호가 존재하는 채팅방에 참여하려 하거나 검색을 통해 채팅방 참여를 할 때 참여 인터페이스 생성



- 비밀번호 채팅방 인터페이스에서 "참여" 버튼을 누르면 입력한 비밀번호 정보를 getText()로 읽은 후 "JC:"를 붙여 읽어들이는 비밀번호 정보와 사용자 정보 채팅방 정보를 서버로 전송
- 비밀번호가 일치하지 않을 경우는 서버에서 "JC:"가 붙은 경고 문자열이 전송됨
- 일반 채팅방에 참여시 "참여" 버튼을 누르면 "JC:"를 붙여 참여하려는 사용자 정보와 채팅방 정보를 서버로 전송

#### 5) 채팅방 인터페이스

: 사용자가 채팅방을 생성하거나 참여하면 해당 채팅방 인터페이스를 생성

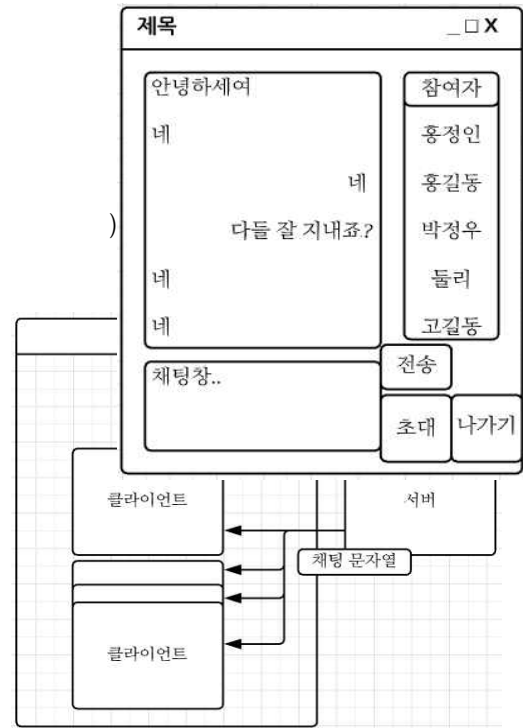
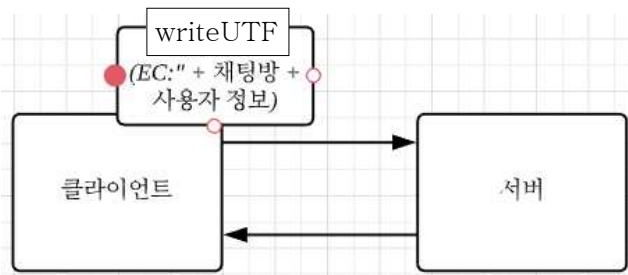
(채팅 전송)

- 사용자가 채팅창에 전하고자 하는 말을 입력하고 "전송" 버튼을 누르면 getText()로 읽은 후 "C:"를 붙여 서버로 전송
- 해당 채팅에 대한 정보를 서버에서 채팅방에

참여한 모든 사용자에게 표시됨

( 나 가 기 )  
(채팅 전송)

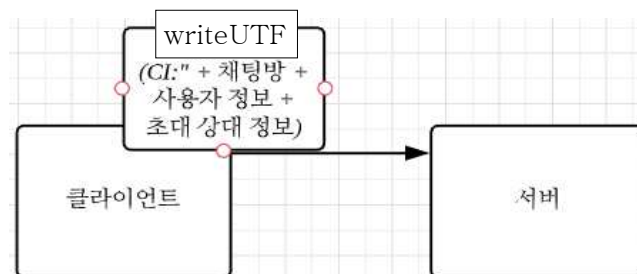
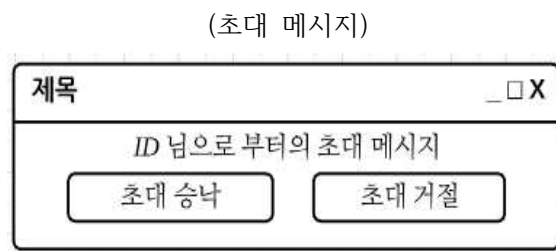
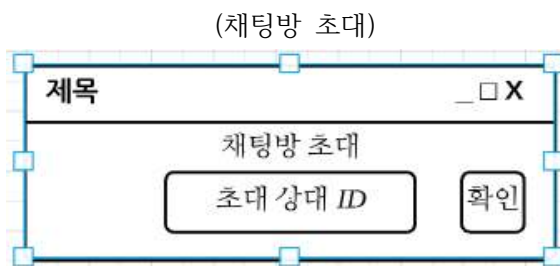
- “나가기” 버튼을 누르면 해당 채팅방 정보와 나가려는 사용자의 정보를 서버로 전송



### 5-1) 채팅방 초대 인터페이스

: “채팅방 초대” 버튼을 누른 사용자에게 채팅방 초대 인터페이스 생성, 초대를 받을 사용자에게 초대 메시지 형태의 인터페이스 생성

- “채팅방 초대”에서 초대할 상대의 ID를 입력하고 “확인” 버튼을 누르면 채팅방 정보와 초대를 할 사용자 정보, 상대방의 정보를 문자열로 합치고 앞에 “CI:”를 붙여 서버로 전송
- 서버에서 넘어온 문자열에 앞 부분이 “CI:”이면 초대 메시지 인터페이스를 생성



### 6) 닉네임 설정 인터페이스

- “확인” 버튼을 누를 시

닉네임 정보와 사용자 정보의 앞에

“CN:”을 붙여 서버로 전송

제목		□X
닉네임 설정		
닉네임	확인	

### 3. 과제 추진 일정

주요내용			추진 일정 (일차)								
			1일	2일	3일	4일	5일	6일	7일	8일	9일
서버 기능	연결 (연동)	DB연결 기능									
		클라이언트 접속 허용									
	회원가입	중복확인 기능									
		아이디 / PW 찾기 기능									
		최종 회원가입 처리 기능									
	로그인	로그인 처리 기능									
	채팅	채팅방 초대 처리									
		채팅방 참여 처리									
		채팅방 검색 처리									
		채팅방 나가기 기능									
		채팅 기능									
	옵션	닉네임 설정 처리 기능									
클라이언트 기능	연결 (연동)	서버 접속 기능									
	회원가입	해당 정보 입력									
		중복확인 정보 전송									
		최종 회원가입 정보 전송									
		회원가입 GUI									
	ID / PW 찾기	ID 찾기									
		PW 찾기									
		ID / PW 찾기 GUI									
	로그인	로그인 정보 전송									
		로그인 GUI									
	채팅	채팅방 초대 정보 전송									
		채팅방 초대 승낙 / 거절									
		채팅방 검색 정보 전송									
		채팅방 참여 정보 전송									
		채팅방 나가기 정보 전송									
		채팅 정보 전송									
		GUI	채팅 대기방								
			채팅방								
		닉네임 설정(변경) 정보 전송									

### 결과활용계획

- 대학 내의 그룹 스터디용 채팅방 프로그램으로 활용 가능
- 대학 내의 작은 커뮤니티 사이트로 활용 가능 ex) 에브리타임, 동아리 모임
- 사회인들의 취미 생활 동호회 등에 활용 가능
- 휘발성 채팅이기에 랜덤채팅 등의 채팅 프로그램으로 발전 가능성이 있음
- 해외로 수출해 외화를 벌어들여 우리나라 경제의 큰 보탬이 됨

### 애로사항