네트워크 프로그래밍

HUSH Chat

컴퓨터공학과 201414137 홍성현

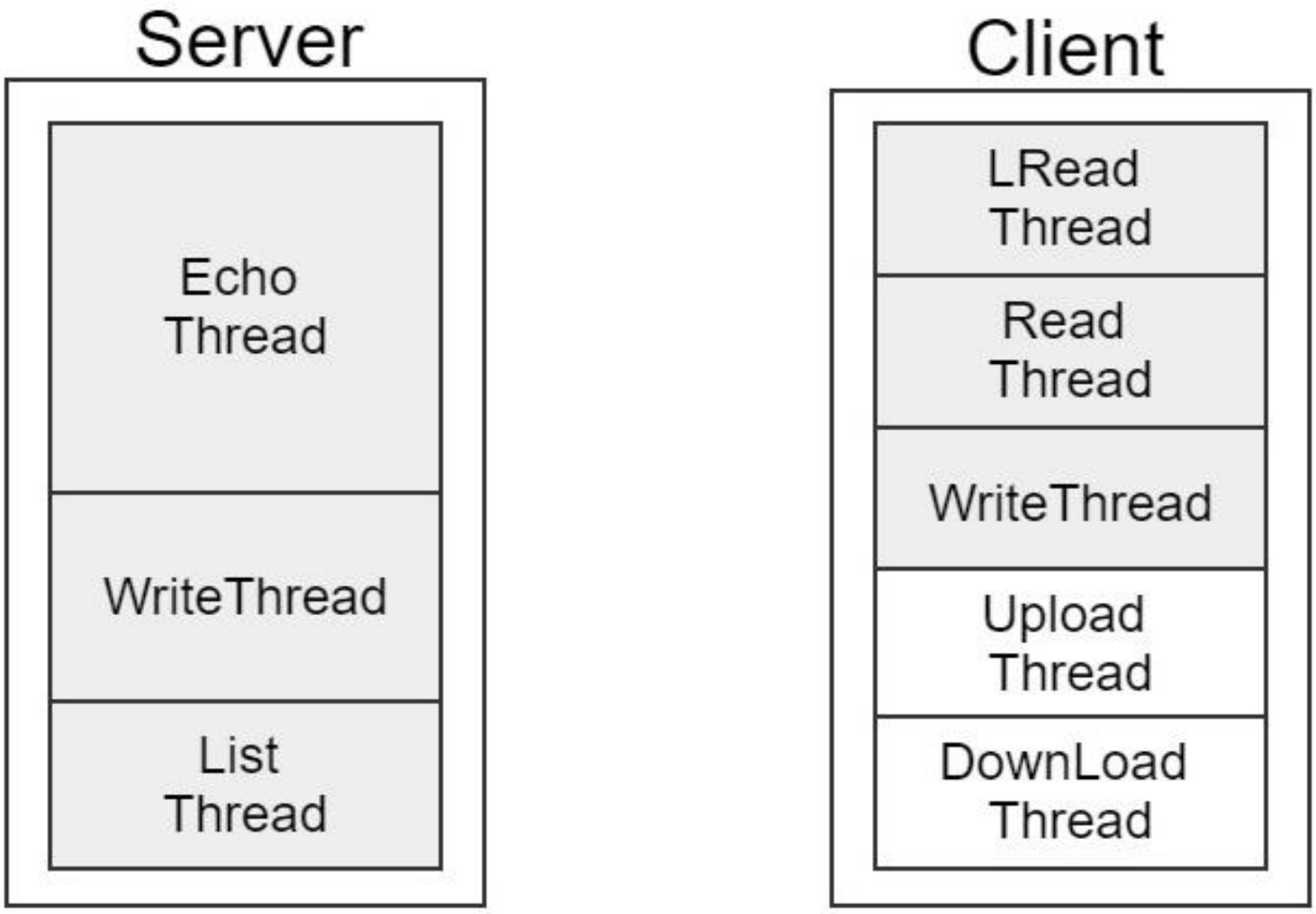
[Contents]

01 프로젝트 목표 프로젝트 목표

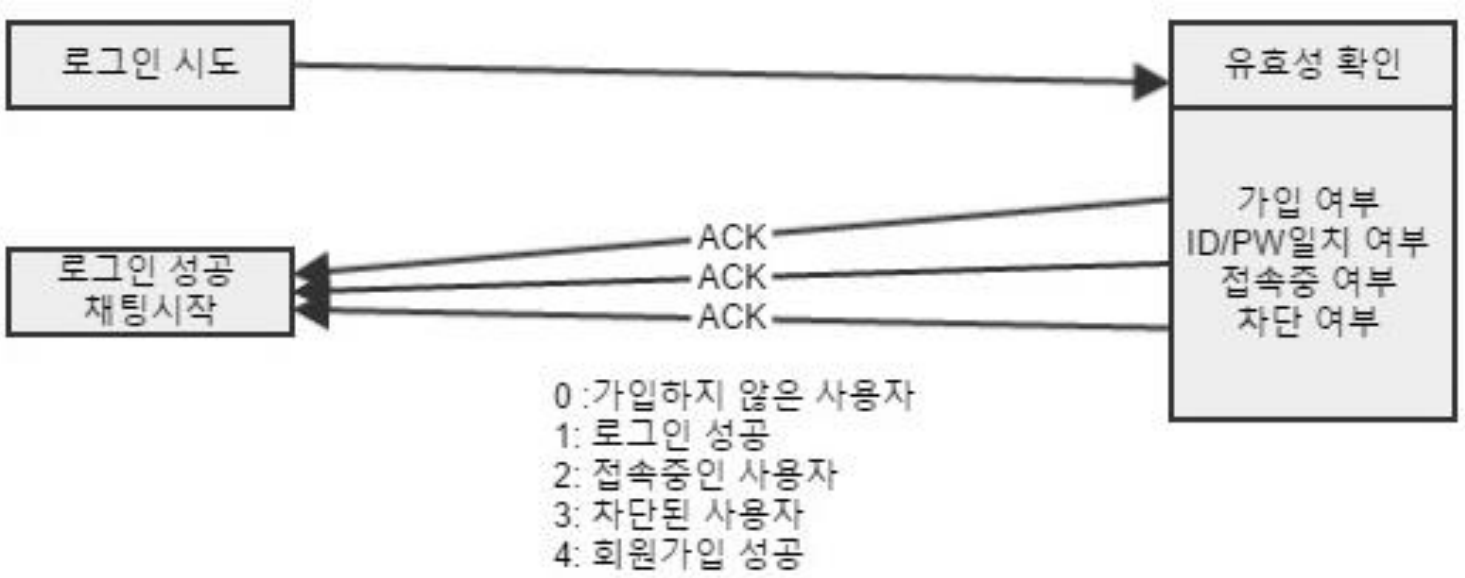
- TCP/IP 소켓 프로그램의 기본 기능 구현
  - 자체적인 통신규약 설정
  - 1:1, 1:N의 기본 채팅 기능 구현
- 클라이언트 다중 접속이 가능한 서버 구축
  - Vector 자료구조를 이용
  - 다중 Thread 방식으로 동시에 여러 기능 사용
- Peer-to-Peer 방식의 파일전송
  - 서버의 오버헤드 및 보안 유지
- 기존 메신저와 차별되는 기능
  - 수신 측 '메시지읽음' 확인
  - 메시지 내용 은닉 기능

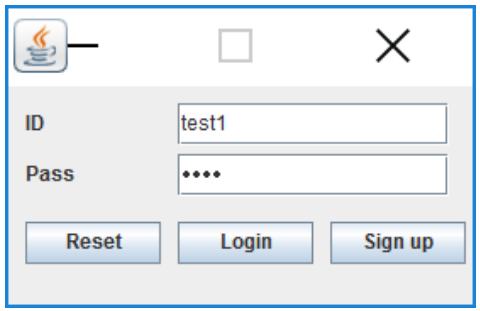
01 프로젝트 목표 프로젝트 개발환경

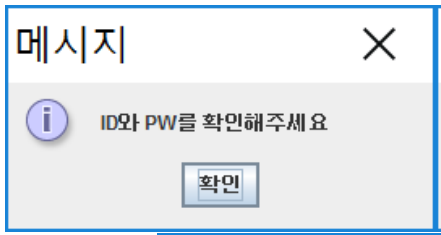
서버, 클라이언트 Thread 구성

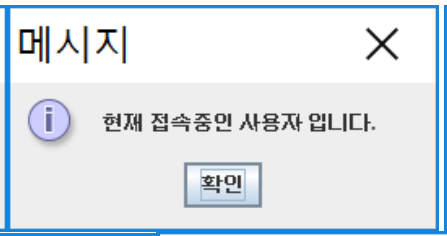


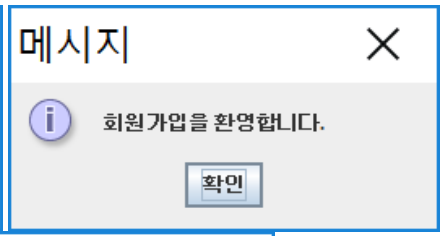
01) 로그인, 회원가입

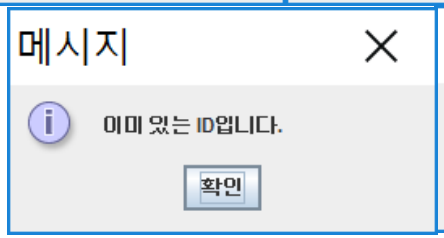

로그인 시도
유효성 확인
가입 여부
ID/PW일치 여부
접속중 여부
차단 여부
ACK
ACK
ACK
로그인 성공
채팅시작
0 :가입하지 않은 사용자
1: 로그인 성공
2: 접속중인 사용자
3: 차단된 사용자
4: 회원가입 성공

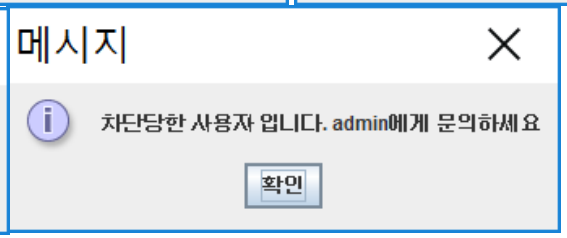

ID
test1
Pass
Reset
Login
Sign up


메시지
ID와 PW를 확인해주세요
확인


메시지
현재 접속중인 사용자 입니다.
확인


메시지
회원가입을 환영합니다.
확인

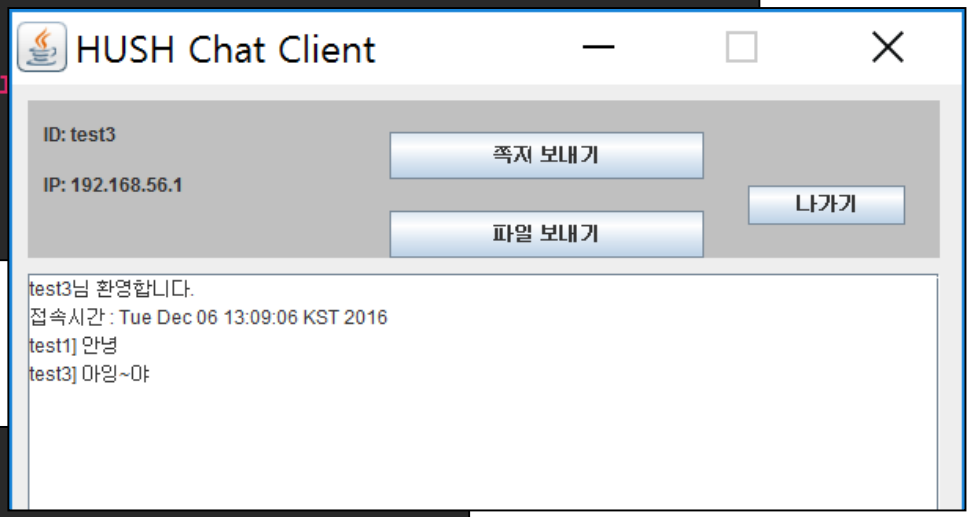

메시지
이미 있는 ID입니다.
확인


메시지
차단당한 사용자 입니다. admin에게 문의하세요
확인

02) 채팅 – 1:N 채팅, 비속어 필터링

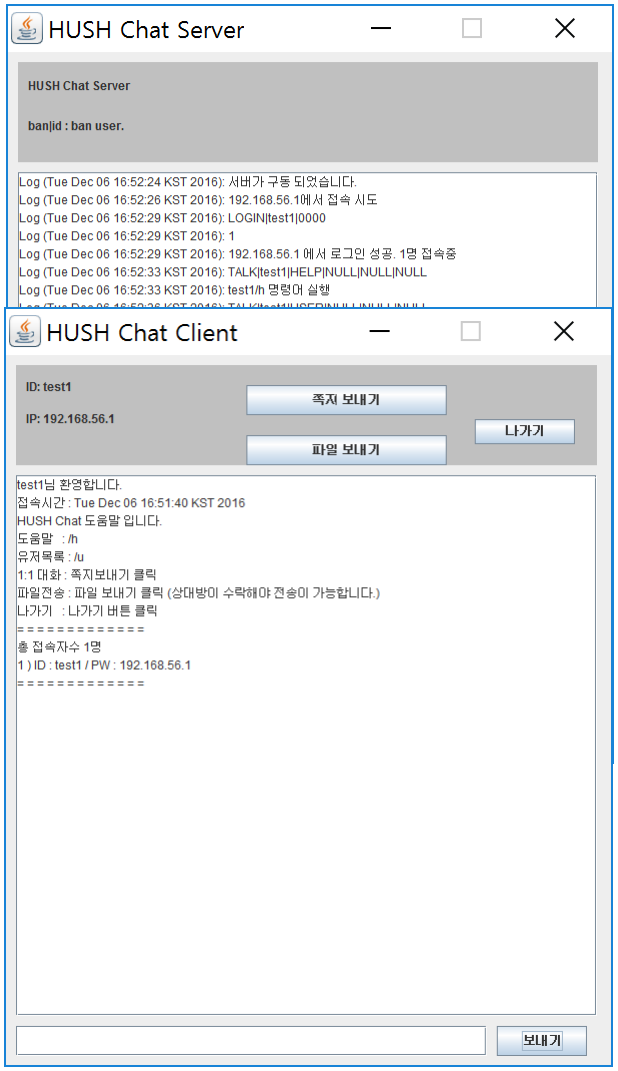
```
public String filter(String msg) throws IOException {
    BufferedReader br = new BufferedReader(new FileReader("C:\\CHAT\\filter.txt"));
    String line = null;
    String result = null;

    while ((line = br.readLine()) != nul
        msg = msg.replace(line, "아잉~");
    }
    return msg;
}
```



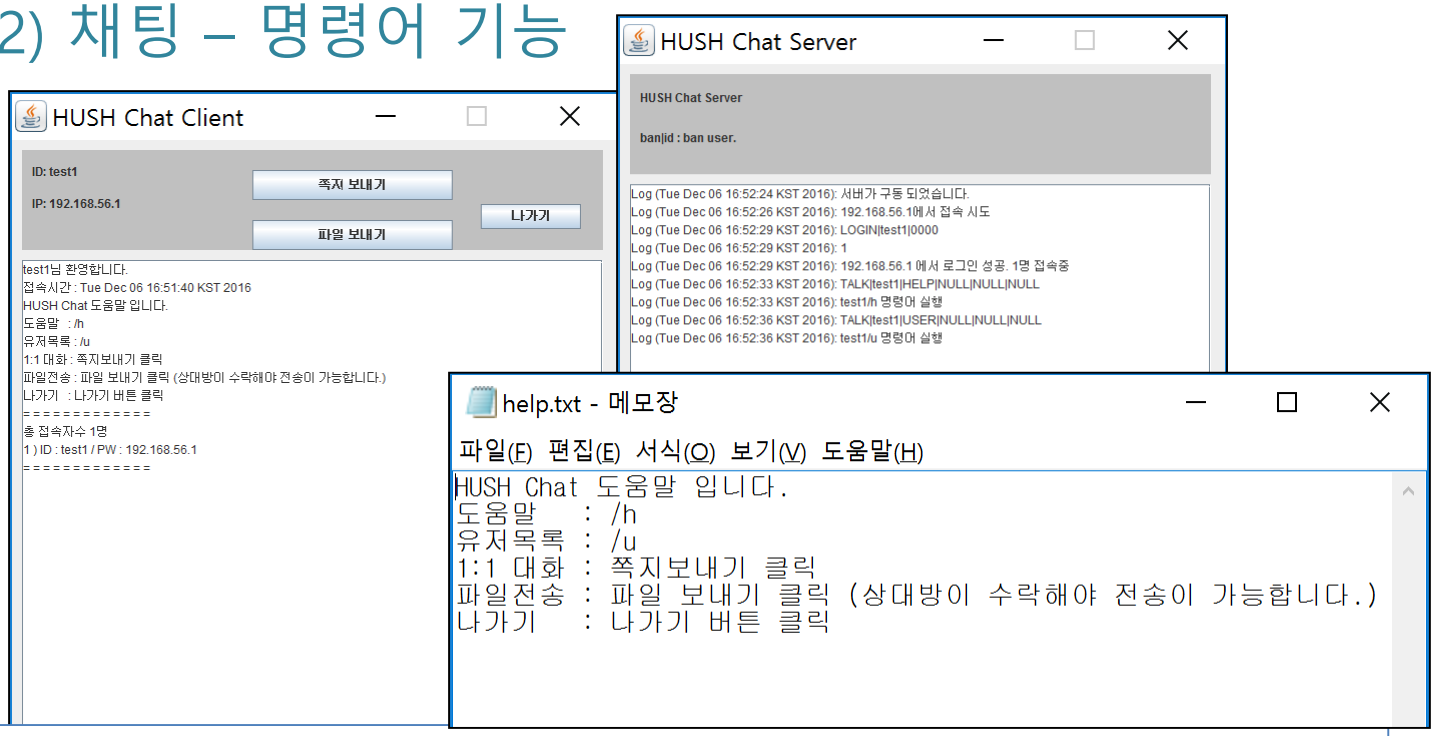

```
public void sendMsg(String str) {
    PrintWriter pw = null;
    try {
        str = filter(str);
        pw = new PrintWriter(socket.getOutputStream(), true);
        StringTokenizer st = new StringTokenizer(str, "|");
        String protocol = st.nextToken();
        if (protocol.equals("WHSP")) {
            pw.println(str);
```

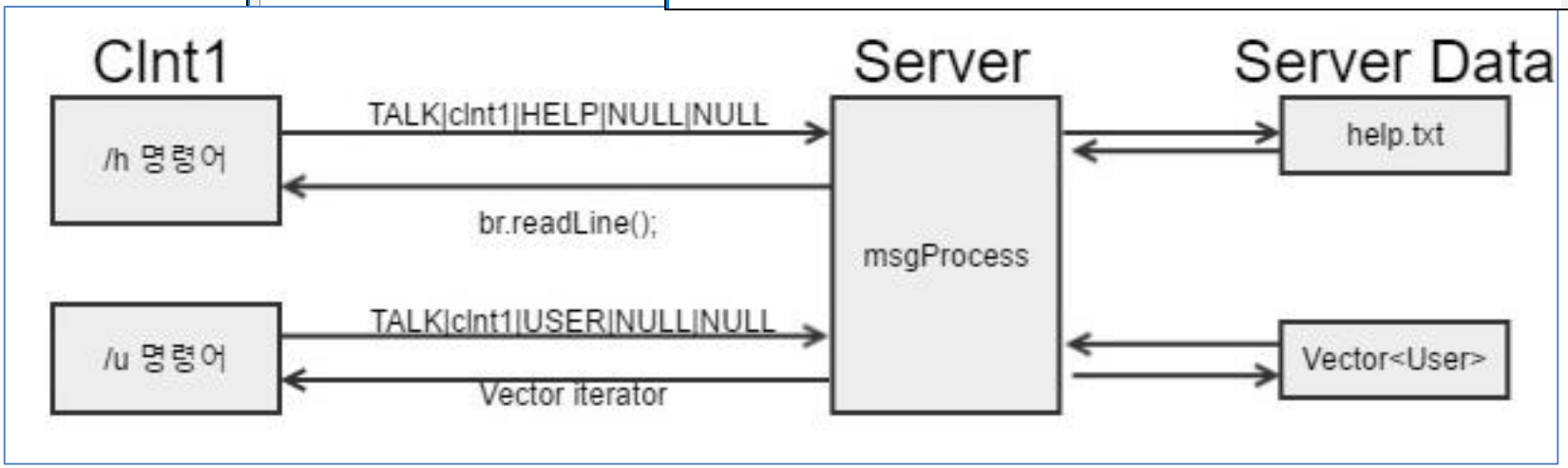
02) 채팅 – 명령어 기능



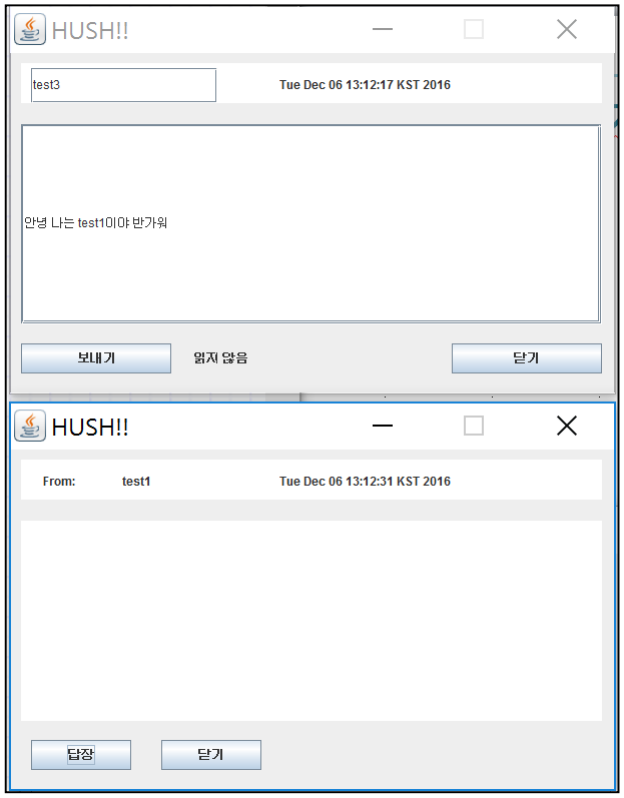

```
public void actionPerformed(ActionEvent e) {
    String btn = e.getActionCommand();
    String txt = txtField.getText();
    if (btn.equals("보내기")) {
        if (txt.equals("")) {
            txtField.setText("");
        } else {
            if (txt.charAt(0) == '/') {
                if (txt.length() == 2) {
                    if (txt.charAt(1) == 'h') {
                        wt.sendMsg("HELP|NULL|NULL");
                        txtField.setText("");
                    } else if (txt.charAt(1) == 'u') {
                        wt.sendMsg("USER|NULL|NULL");
                        txtField.setText("");
                    }
                }
            } else {
                wt.sendMsg(this.txtField.getText());
                txtField.setText("");
```

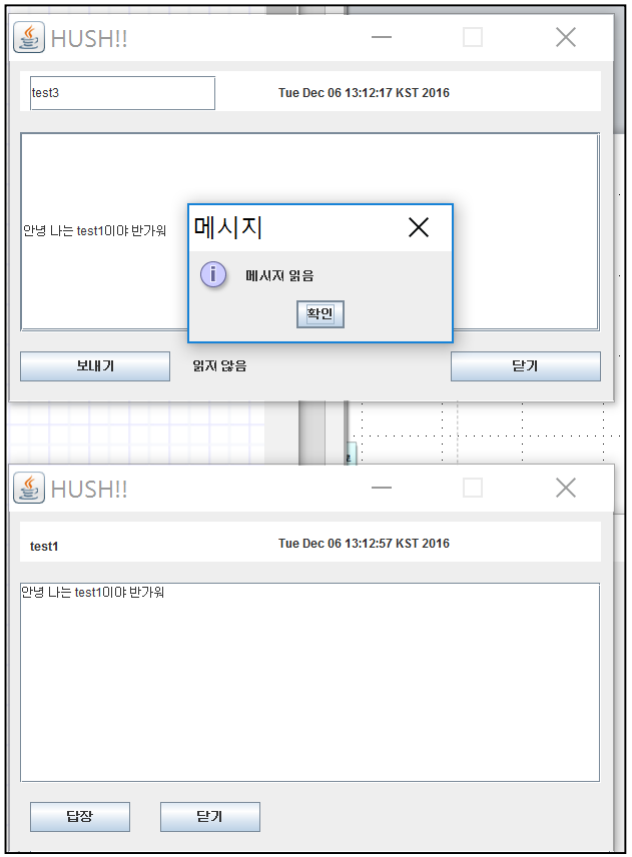
02) 채팅 – 명령어 기능


HUSH Chat Server
HUSH Chat Server
ban|id : ban user.
Log (Tue Dec 06 16:52:24 KST 2016): 서버가 구동 되었습니다.
Log (Tue Dec 06 16:52:26 KST 2016): 192.168.56.1에서 접속 시도
Log (Tue Dec 06 16:52:29 KST 2016): LOGIN|test1|0000
Log (Tue Dec 06 16:52:29 KST 2016): 1
Log (Tue Dec 06 16:52:29 KST 2016): 192.168.56.1 에서 로그인 성공. 1명 접속중
Log (Tue Dec 06 16:52:33 KST 2016): TALK|test1|HELP|NULL|NULL|NULL
Log (Tue Dec 06 16:52:33 KST 2016): test1/h 명령어 실행
Log (Tue Dec 06 16:52:36 KST 2016): TALK|test1|USER|NULL|NULL|NULL
Log (Tue Dec 06 16:52:36 KST 2016): test1/u 명령어 실행
HUSH Chat Client
ID: test1
IP: 192.168.56.1
쪽지 보내기
파일 보내기
나가기
test1님 환영합니다.
접속시간 : Tue Dec 06 16:51:40 KST 2016
HUSH Chat 도움말 입니다.
도움말 : /h
유저목록 : /u
1:1 대화 : 쪽지보내기 클릭
파일전송 : 파일 보내기 클릭 (상대방이 수락해야 전송이 가능합니다.)
나가기 : 나가기 버튼 클릭
==========
총 접속자수 1명
1) ID : test1 / PW : 192.168.56.1
==========
help.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
HUSH Chat 도움말 입니다.
도움말 : /h
유저목록 : /u
1:1 대화 : 쪽지보내기 클릭
파일전송 : 파일 보내기 클릭 (상대방이 수락해야 전송이 가능합니다.)
나가기 : 나가기 버튼 클릭

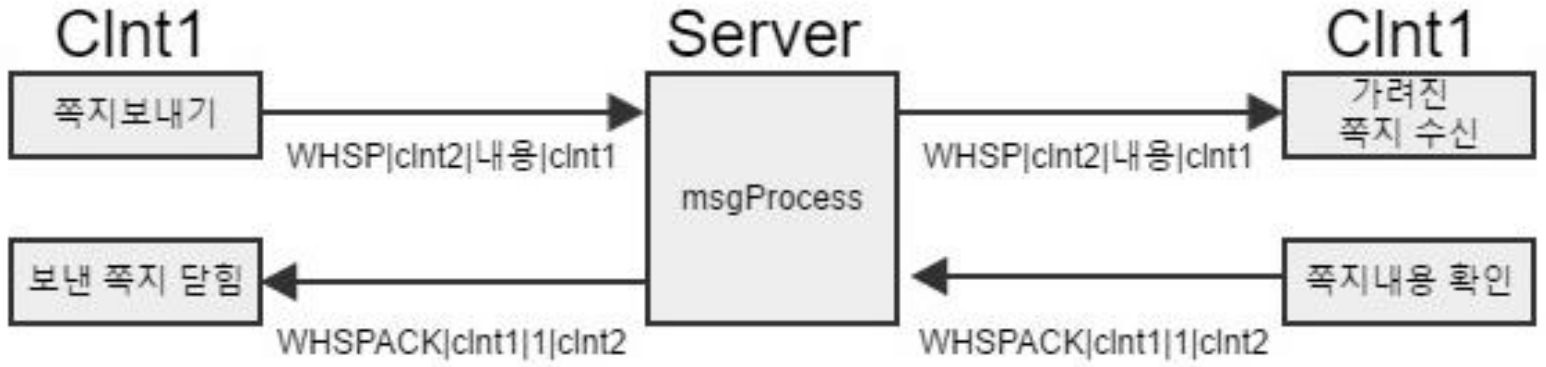

Clnt1
Server
Server Data
/h 명령어
TALK|clnt1|HELP|NULL|NULL
br.readLine();
msgProcess
help.txt
/u 명령어
TALK|clnt1|USER|NULL|NULL
Vector iterator
Vector<User>

03) 쪽지보내기 – 1:1 채팅





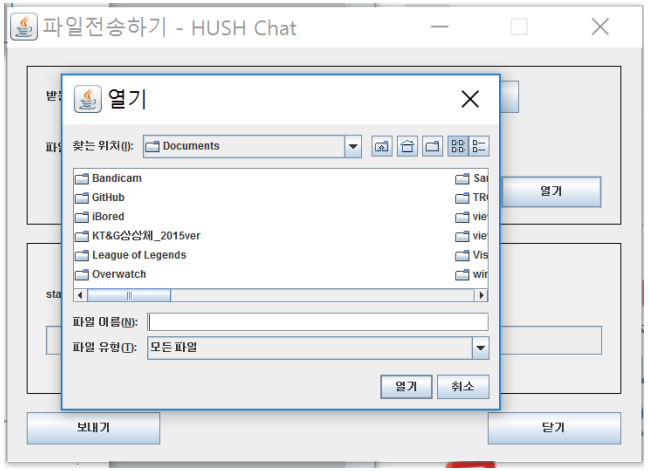
03) 쪽지보내기 – 1:1 채팅

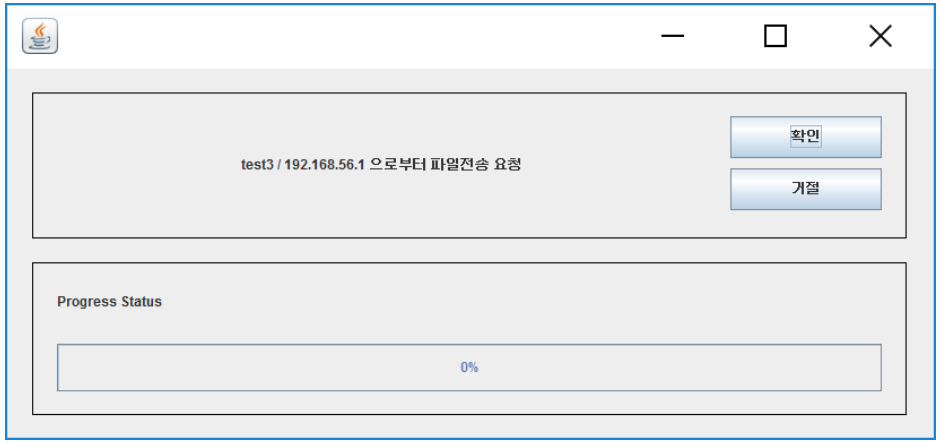


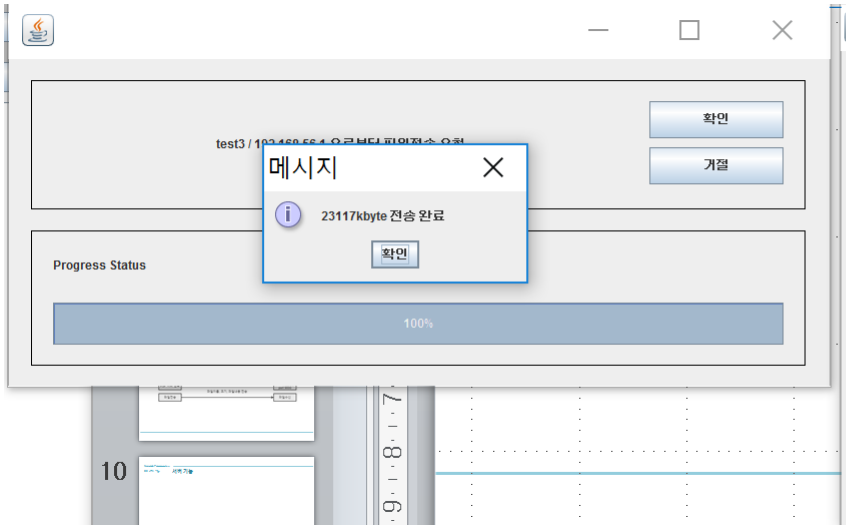

```
Log (Tue Dec 06 13:12:31 KST 2016): test3에게 test1가 :WHSP|test3|안녕 나는 test1이야 반가워|test1
Log (Tue Dec 06 13:12:31 KST 2016): test3에게 test1가 :WHSP|test3|안녕 나는 test1이야 반가워|test1
Log (Tue Dec 06 13:12:57 KST 2016): WHSPACK|test1|1|test1
Log (Tue Dec 06 13:12:57 KST 2016): test1에게 test1가 :WHSPACK|test1|1
Log (Tue Dec 06 13:12:57 KST 2016): test1에게 test1가 :WHSPACK|test1|1
```

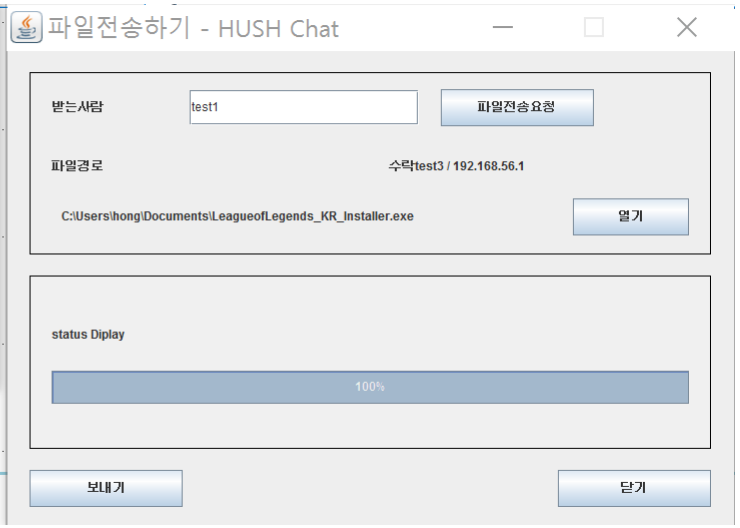
* 서버 측에 기록된 로그

04) P2P 파일전송

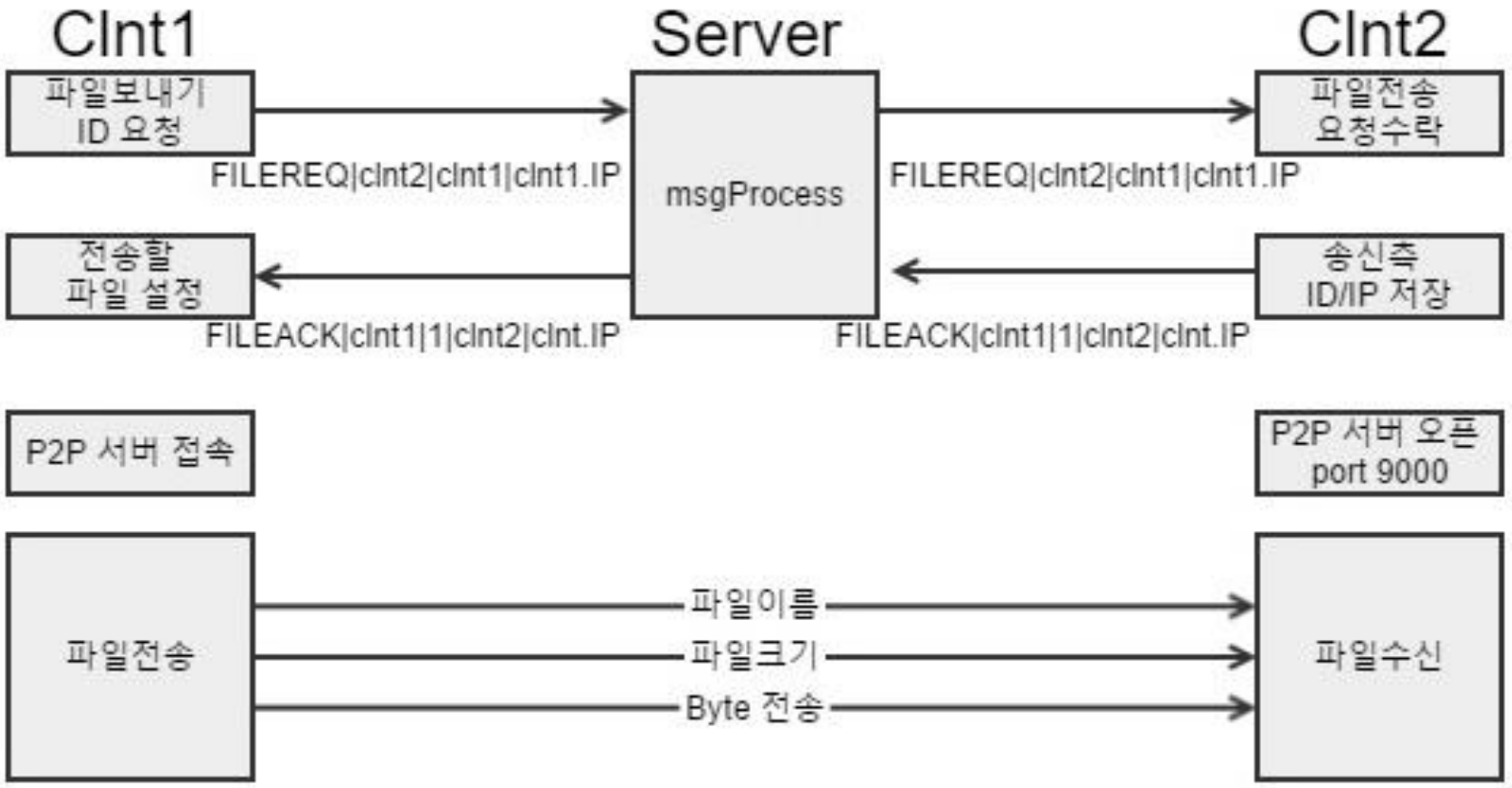

파일전송하기 - HUSH Chat
열기
찾는 위치(I):
Documents
Bandicam
GitHub
iBored
KT&G상상채_2015ver
League of Legends
Overwatch
파일 이름(N):
파일 유형(T):
모든 파일
열기
취소
보내기
닫기


test3 / 192.168.56.1 으로부터 파일전송 요청
확인
거절
Progress Status
0%


확인
거절
메시지
23117kbyte 전송 완료
확인
Progress Status
100%

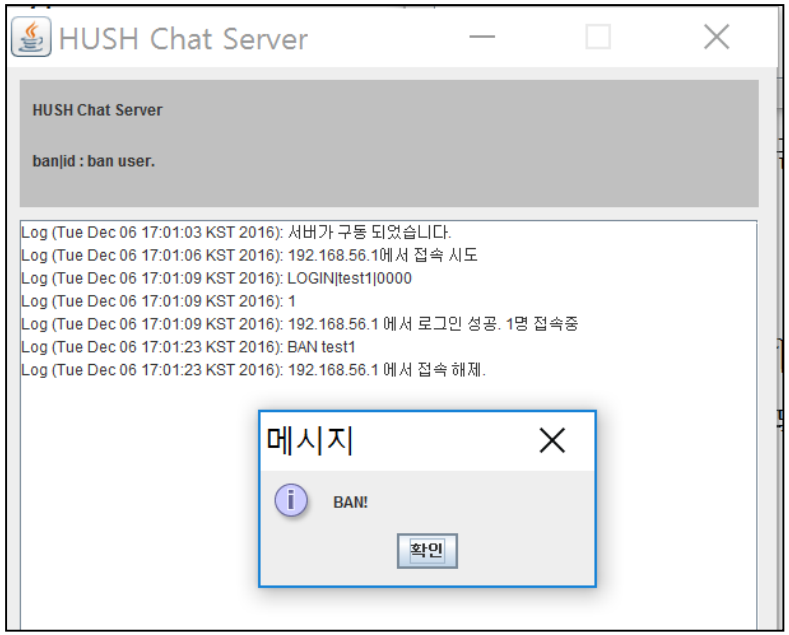

파일전송하기 - HUSH Chat
받는사람
test1
파일전송요청
파일경로
수락test3 / 192.168.56.1
C:\Users\hong\Documents\LeagueofLegends_KR_Installer.exe
열기
status Diplay
100%
보내기
닫기

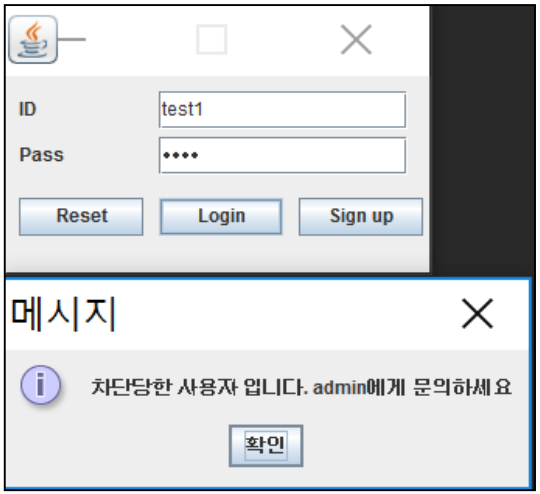
04) P2P 파일전송

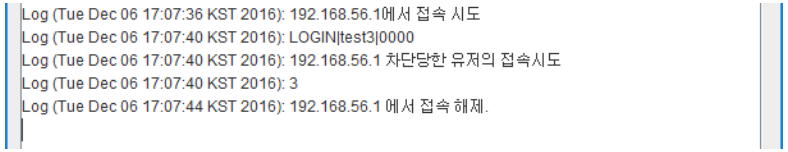


* 수신측의 C:₩Down₩<송신측ID> 경로에 파일이 저장됨.

01) 서버 측 명령어 (ban|ID, unban|ID)

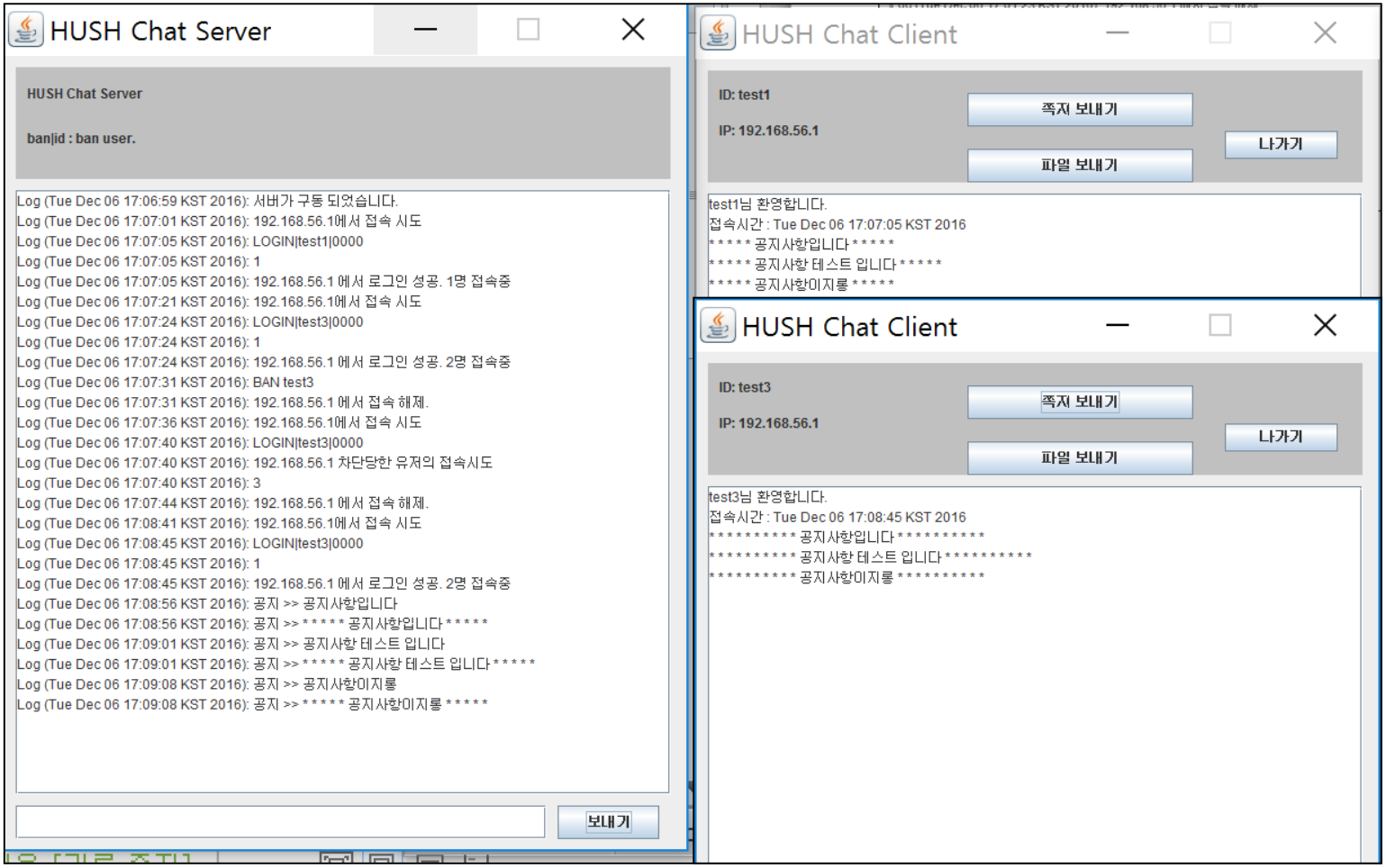




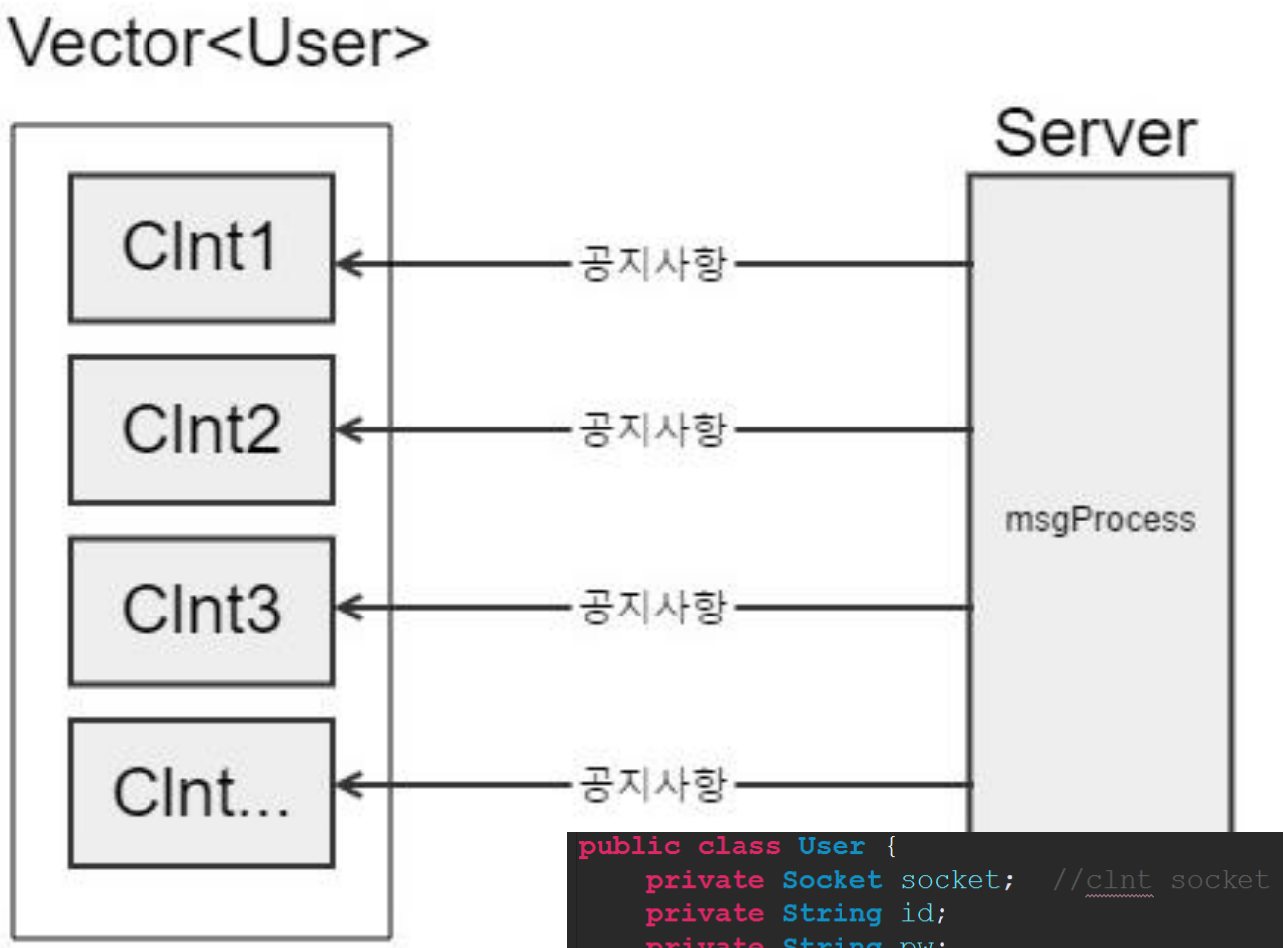


* unban|ID 를 통해 차단을 해제할 수도 있음.

02) 공지사항


HUSH Chat Server
HUSH Chat Server
ban|id : ban user.
Log (Tue Dec 06 17:06:59 KST 2016): 서버가 구동 되었습니다.
Log (Tue Dec 06 17:07:01 KST 2016): 192.168.56.1에서 접속 시도
Log (Tue Dec 06 17:07:05 KST 2016): LOGIN|test1|0000
Log (Tue Dec 06 17:07:05 KST 2016): 1
Log (Tue Dec 06 17:07:05 KST 2016): 192.168.56.1 에서 로그인 성공. 1명 접속중
Log (Tue Dec 06 17:07:21 KST 2016): 192.168.56.1에서 접속 시도
Log (Tue Dec 06 17:07:24 KST 2016): LOGIN|test3|0000
Log (Tue Dec 06 17:07:24 KST 2016): 1
Log (Tue Dec 06 17:07:24 KST 2016): 192.168.56.1 에서 로그인 성공. 2명 접속중
Log (Tue Dec 06 17:07:31 KST 2016): BAN test3
Log (Tue Dec 06 17:07:31 KST 2016): 192.168.56.1 에서 접속 해제.
Log (Tue Dec 06 17:07:36 KST 2016): 192.168.56.1에서 접속 시도
Log (Tue Dec 06 17:07:40 KST 2016): LOGIN|test3|0000
Log (Tue Dec 06 17:07:40 KST 2016): 192.168.56.1 차단당한 유저의 접속시도
Log (Tue Dec 06 17:07:40 KST 2016): 3
Log (Tue Dec 06 17:07:44 KST 2016): 192.168.56.1 에서 접속 해제.
Log (Tue Dec 06 17:08:41 KST 2016): 192.168.56.1에서 접속 시도
Log (Tue Dec 06 17:08:45 KST 2016): LOGIN|test3|0000
Log (Tue Dec 06 17:08:45 KST 2016): 1
Log (Tue Dec 06 17:08:45 KST 2016): 192.168.56.1 에서 로그인 성공. 2명 접속중
Log (Tue Dec 06 17:08:56 KST 2016): 공지 >> 공지사항입니다
Log (Tue Dec 06 17:08:56 KST 2016): 공지 >> ***** 공지사항입니다 *****
Log (Tue Dec 06 17:09:01 KST 2016): 공지 >> 공지사항 테스트 입니다
Log (Tue Dec 06 17:09:01 KST 2016): 공지 >> ***** 공지사항 테스트 입니다 *****
Log (Tue Dec 06 17:09:08 KST 2016): 공지 >> 공지사항이지롱
Log (Tue Dec 06 17:09:08 KST 2016): 공지 >> ***** 공지사항이지롱 *****
보내기
HUSH Chat Client
ID: test1
IP: 192.168.56.1
쪽지 보내기
파일 보내기
나가기
test1님 환영합니다.
접속시간 : Tue Dec 06 17:07:05 KST 2016
***** 공지사항입니다 *****
***** 공지사항 테스트 입니다 *****
***** 공지사항이지롱 *****
HUSH Chat Client
ID: test3
IP: 192.168.56.1
쪽지 보내기
파일 보내기
나가기
test3님 환영합니다.
접속시간 : Tue Dec 06 17:08:45 KST 2016
********** 공지사항입니다 **********
********** 공지사항 테스트 입니다 **********
********** 공지사항이지롱 **********

02) 공지사항

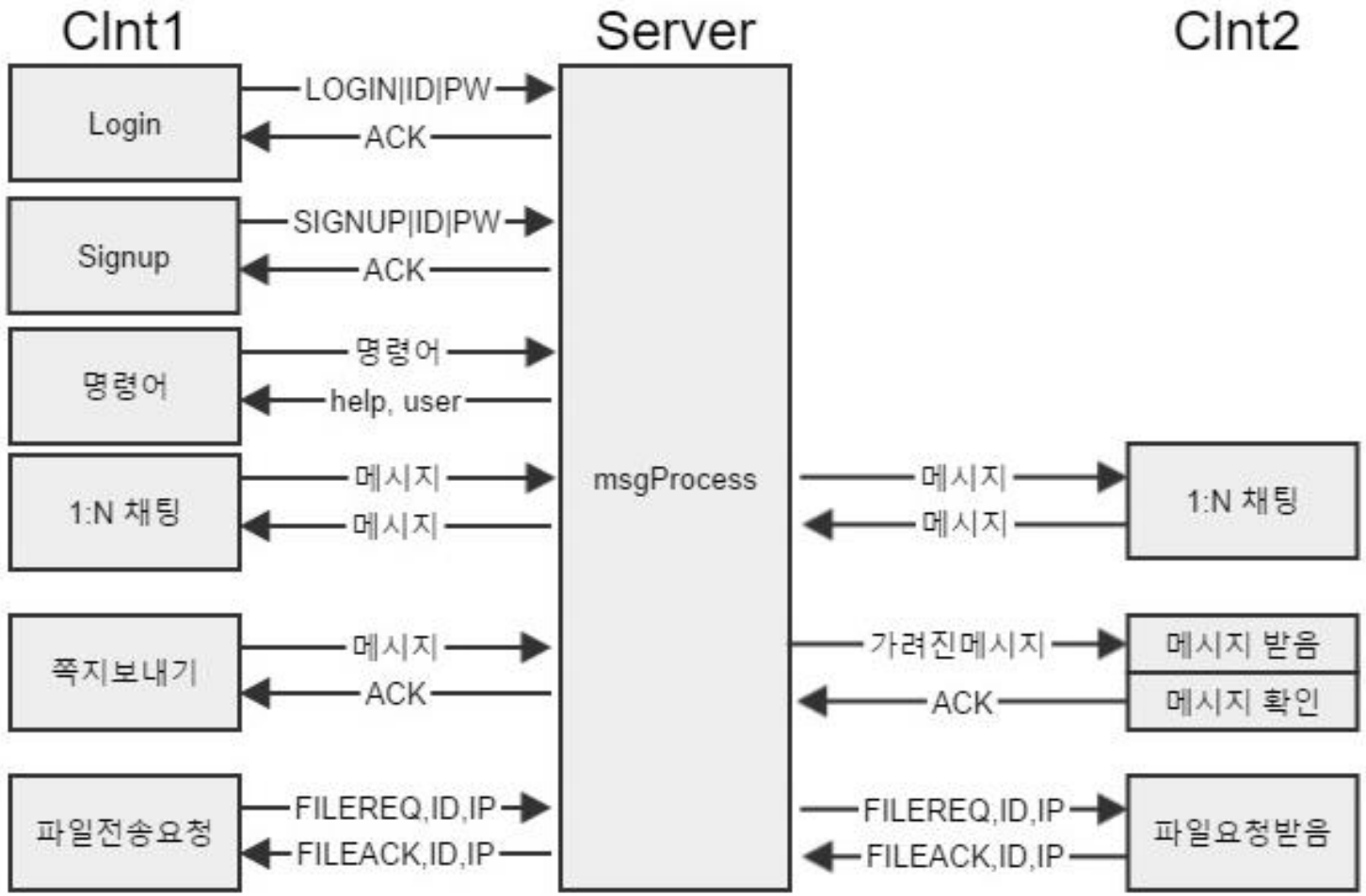



```
public class User {
    private Socket socket;   //clnt socket
    private String id;
    private String pw;
    private String ip;
    private String wb; // whisper back
    private String banned;
```

03) 클라이언트 간 중계

```
public void run() {
    BufferedReader br = null;
    try {
        br = new BufferedReader(new InputStreamReader(user.getSocket().getInp
        while (true) {
            // 클라이언트로 부터 문자열 받기
            String str = br.readLine();
            if (str == null) {
                continue;
            }
            sf.addLog(str);
            msgProcess(str, socket);
        }
    } catch (Exception ie) {
        ie.printStackTrace();
        uv.remove(user);
        sf.addLog(socket.getInetAddress().getHostAddress() + " 에서 접속 해제.");
    } finally {
        try {
            if (br != null)
                br.close();
            if (socket != null)
                socket.close();
        } catch (IOException ie) {
            System.out.println(ie.getMessage() + "run2");
        }
    }
}
```

03) 클라이언트 간 중계



- 메시지 문자열을 StringTokenizer로 Parsing 하여 상황에 맞게 중계한다.
- sendMsg() Overloading을 통해 구현하여 유연한 대처가 가능하다.

04) 서버 모니터링 – Logging, Vector Thread