컴퓨터정보공학부 201921725 안성현

CUK Chat Community

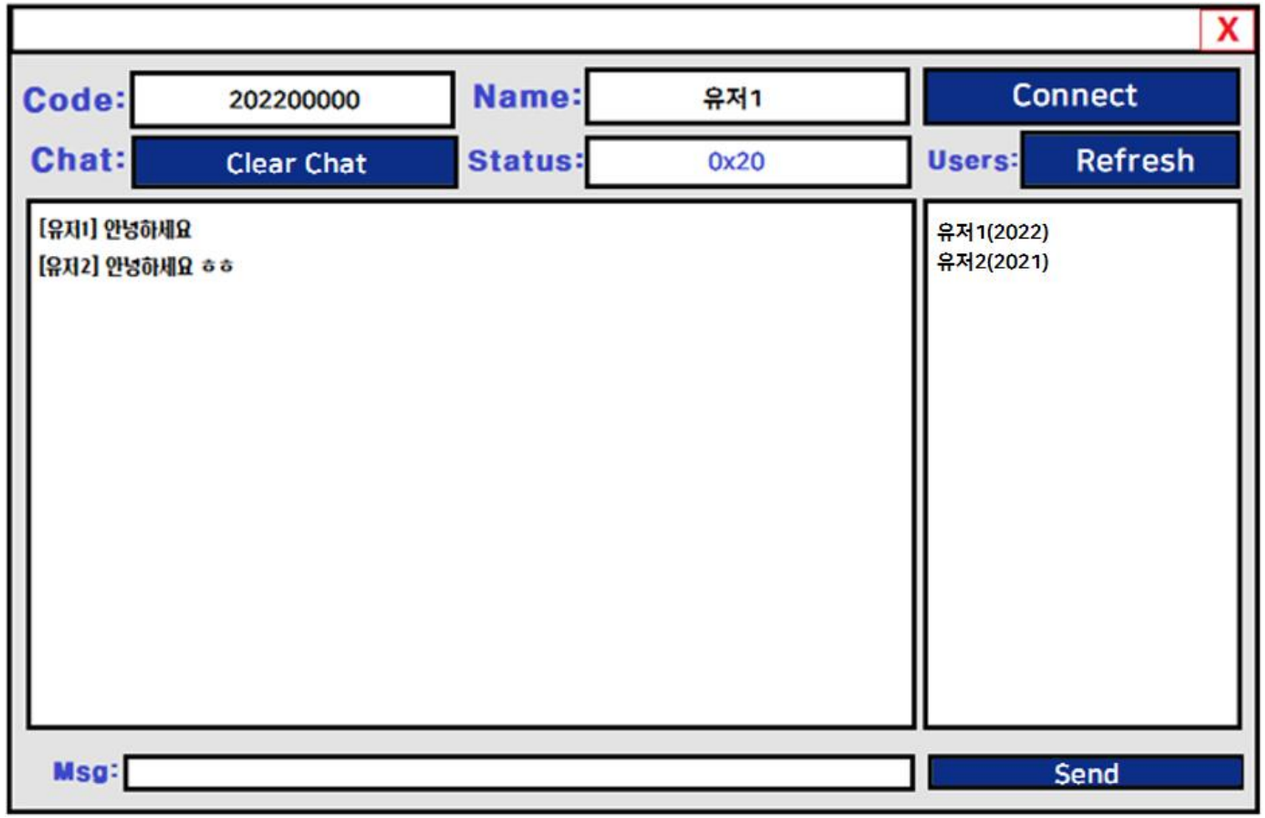
<2022/11/22>

PT

1. 설계
2. 구현
3. 결과
4. 데모

설계

- CUK Chat Community
  - 가톨릭대 학생들이 자유롭게 이야기할 수 있는 커뮤니티 (채팅 프로그램)
  - 예상 GUI는 아래와 같으며, 프로그래밍 언어는 JAVA를 사용한다.



설계

- Use-case List
  - 액터 목록과 유즈케이스 목록

액터	설명
Student	프로그램을 사용하는 사람
Kakao API	키워드로 장소를 검색하는 API 시스템

유즈케이스명	설명
Connect	채팅 서버에 접속한다
Send	채팅창에 메시지를 전송한다
Clear Chat	채팅 내용을 삭제한다
Check Member	채팅에 참여하는 현재 인원을 확인한다
Search	가톨릭대 주변 음식점 목록을 검색한다
Exit	채팅 서버 접속을 해제한다

설계

- Use-case Scenario
  - 처음 접속

유즈케이스명	Connect (처음 접속)		
설명	채팅 서버에 접속을 시도한다		
기본 흐름	사용자 (Sender)	시스템	다른 사용자들 (Receivers)
	1. code란에 학번을 입력한다.		
	2. name란에 닉네임을 입력한다.		
	3. Connect버튼을 클릭한다.		
		4. 올바른 학번인지 검사한다.	
		5. 학번이 올바르면 접속을 허가한다.	
		6. 모든 유저에게 입장 메시지를 보여준다.	
	7. 채팅창에서 입장 메시지를 확인할 수 있다.		7. 채팅창에서 입장 메시지를 확인할 수 있다.
	8. 유즈케이스를 종료한다.		

설계

- Use-case Scenario
  - 메시지 전송

유즈케이스명	Send (메시지 전송)		
설명	채팅창에 메시지를 전송한다		
기본 흐름	사용자 (Sender)	시스템	다른 사용자들 (Receivers)
	1. Msg란에 메시지를 입력한다.		
	2. Send 버튼을 클릭한다.		
		3. 메시지를 모든 사용자에게 보여준다.	
	4. 채팅창에서 메시지를 확인할 수 있다.		4. 채팅창에서 메시지를 확인할 수 있다.
	5. 유즈케이스를 종료한다.		

설계

- Use-case Scenario
  - 채팅 내용 삭제

유즈케이스명	Clear Chat (채팅 내용 삭제)	
설명	채팅창의 내용을 삭제한다	
기본 흐름	사용자	시스템
	1. Clear Chat 버튼을 클릭한다.	
		2. 채팅창의 모든 내용을 삭제한다.
	3. 지워진 채팅창을 확인할 수 있다.	
	4. 유즈케이스를 종료한다.	

설계

- Use-case Scenario
  - 채팅 인원 확인

유즈케이스명	Check Member (채팅 인원 확인)	
설명	채팅에 참가하는 현재 인원을 확인한다	
기본 흐름	사용자	시스템
	1. Refresh 버튼을 클릭한다.	
		2. 채팅에 참가하는 인원 정보를 보여준다.
	3. 유저 정보 창에서 현재 인원 정보를 확인할 수 있다.	
	4. 유즈케이스를 종료한다.	

설계

- Use-case Scenario
  - 음식점 검색

유즈케이스명	Search (음식점 검색)		
설명	카카오 API를 이용해서 역곡 주변 음식점 검색을 한다		
기본 흐름	사용자	시스템	카카오 API
	1. Msg란에 !과 음식 키워드를 입력하고 Send 버튼을 클릭한다.		
		2. 음식 키워드에 '역곡'을 추가한다.	
		3. 카카오 API에게 음식점 리스트를 요청한다.	
			4. 역곡 주변 음식점을 검색한다.
			5. 음식점 리스트를 반환한다.
		6. 사용자에게 음식점 리스트를 보여준다.	
	7. 채팅창에서 음식점 리스트를 확인할 수 있다.		
	8. 유즈케이스를 종료한다.		

설계

- Use-case Scenario
  - 채팅 종료

유즈케이스명	Exit (채팅 종료)		
설명	채팅 서버와 연결을 해제한다		
기본 흐름	사용자 (Sender)	시스템	다른 사용자들 (Receivers)
	1. 종료 버튼을 클릭한다.		
		2. 사용자와 연결을 해제한다.	
		3. 다른 사용자들에게 퇴장 메시지를 전송한다.	
			4. 채팅창에서 퇴장 메시지를 확인할 수 있다.
	5. 유즈케이스를 종료한다.		

설계

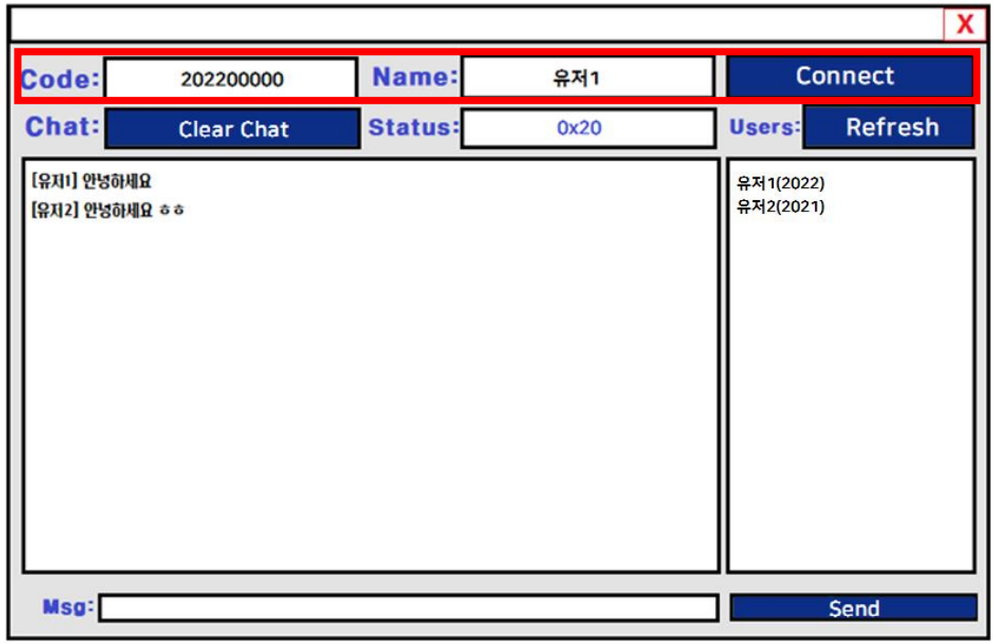
- Protocol
  - 통신에 사용되는 패킷(Packet)의 구조

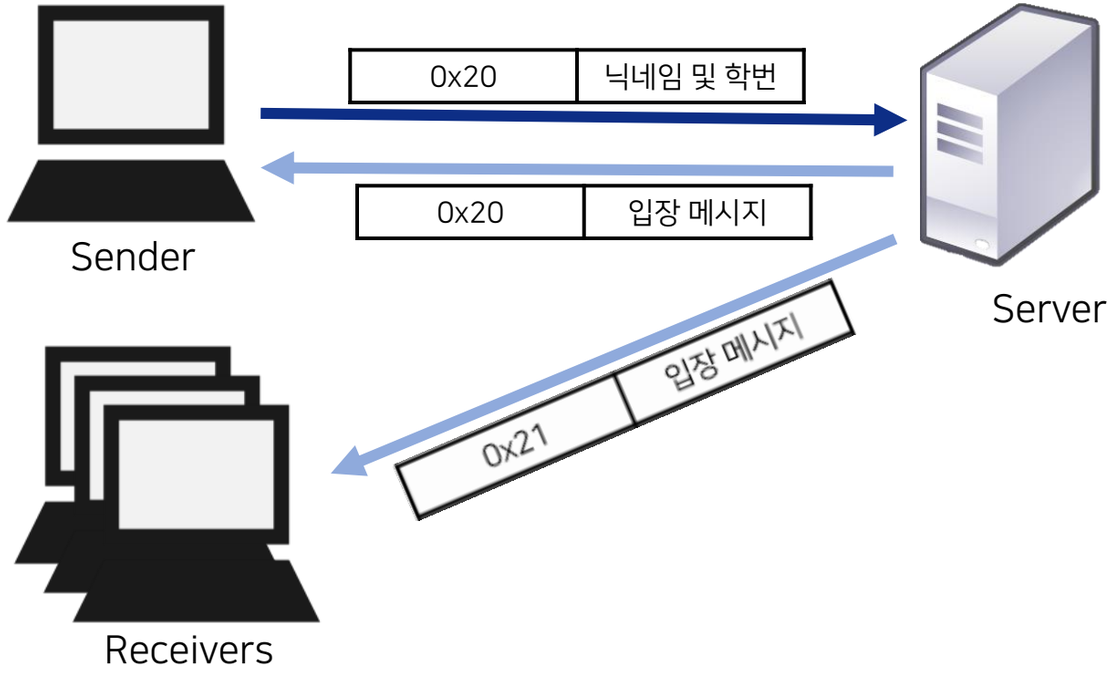
명령어 (command)	메시지 (message)

명령어 (16진수)	메시지	설명
0x10	None	채팅 서버에 접속을 해제
0x20	닉네임 및 학번	채팅 서버에 접속을 처음 시도
0x21	메시지	메시지 전송
0x30	!음식 키워드	키워드를 이용해서 음식점 검색
0x40	None	현재 채팅을 참여하는 인원 확인

설계

- Protocol
  - **처음 접속 시도 (0x20)**: 올바른 학번을 입력하고 채팅할 닉네임을 입력 후 Connect버튼을 클릭

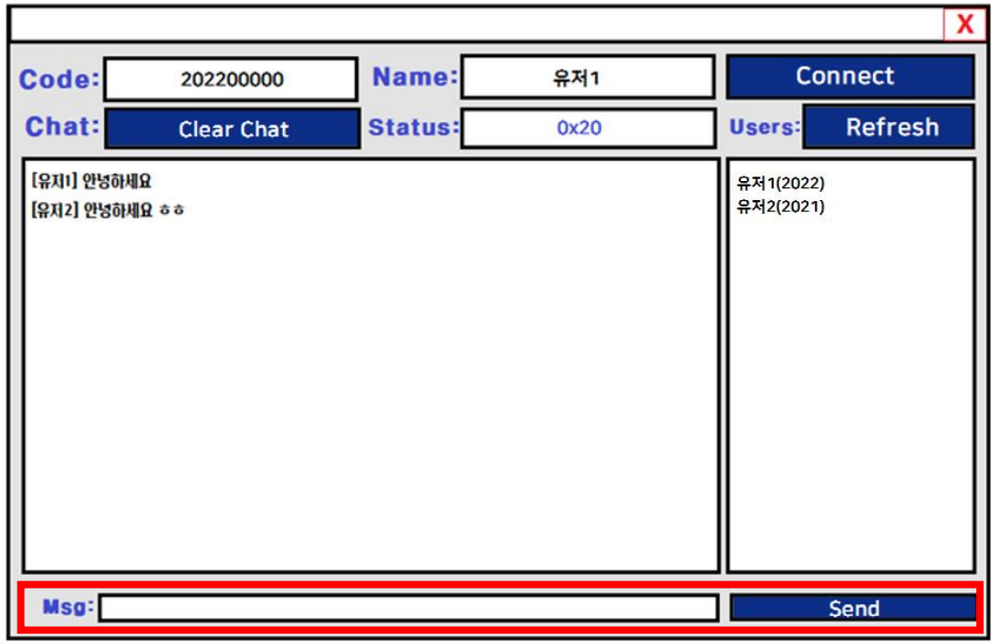


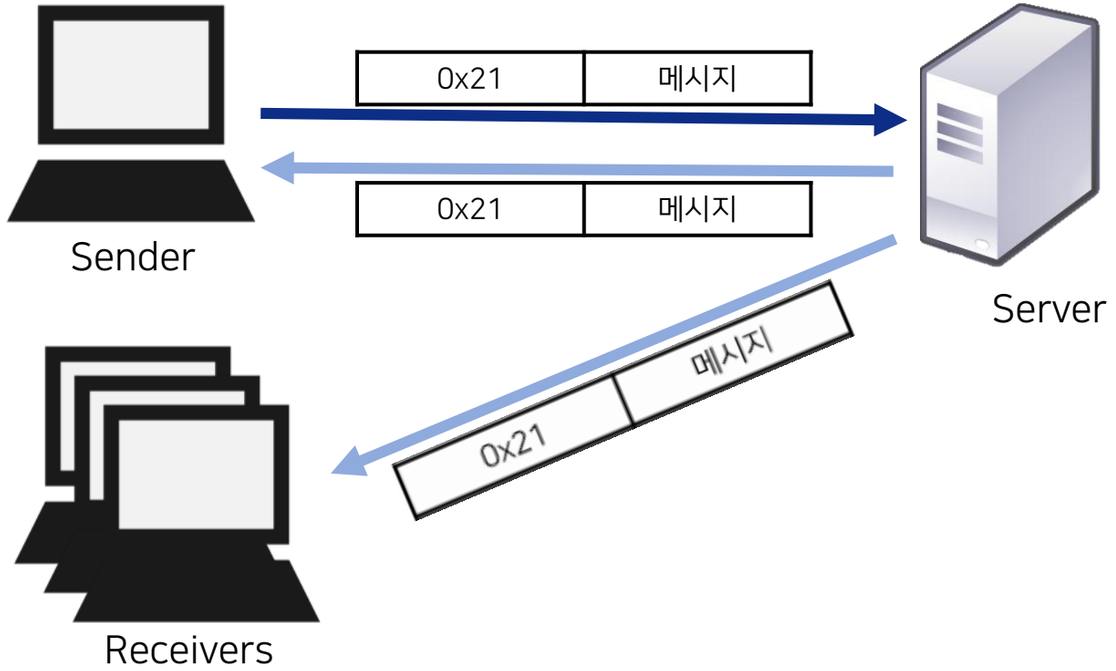


- 입장 메시지: “#(닉네임)님이 입장하셨습니다.”

설계

- Protocol
  - **메시지 전송 (0x21)**: 채팅할 메시지를 입력 후 Send 버튼을 클릭

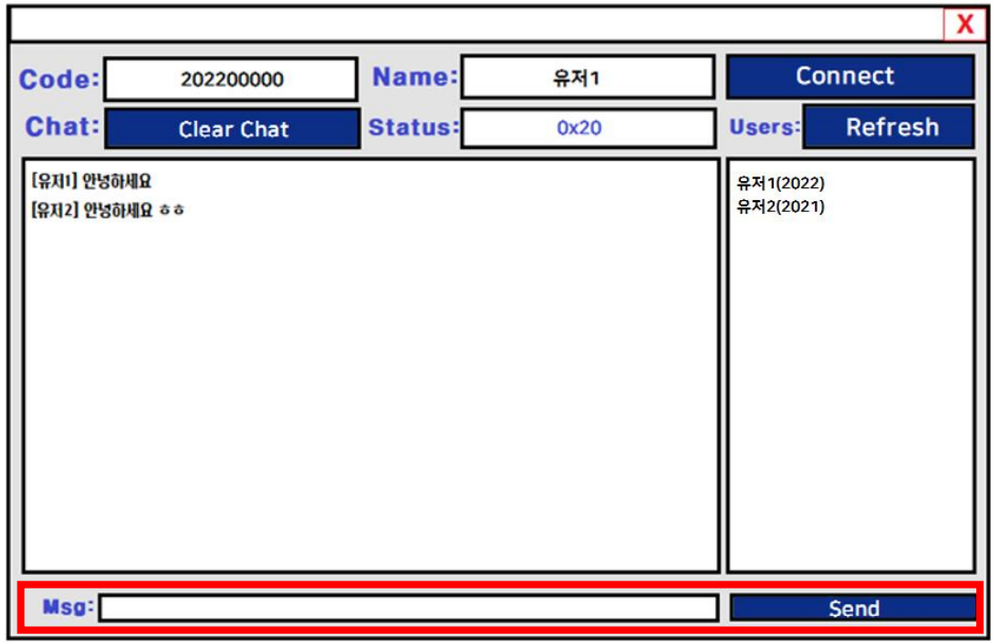


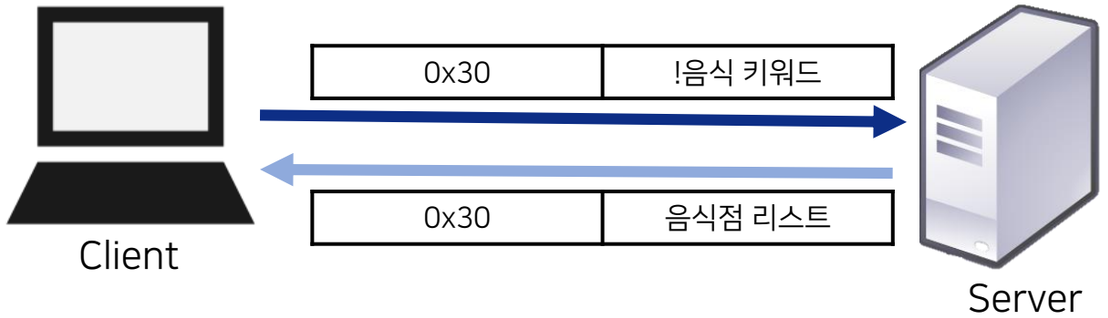


- 메시지 예시: “[닉네임] 안녕하세요”

설계

- Protocol
  - **음식점 검색 (0x30)**: !와 음식 키워드를 입력 후 Send 버튼을 클릭



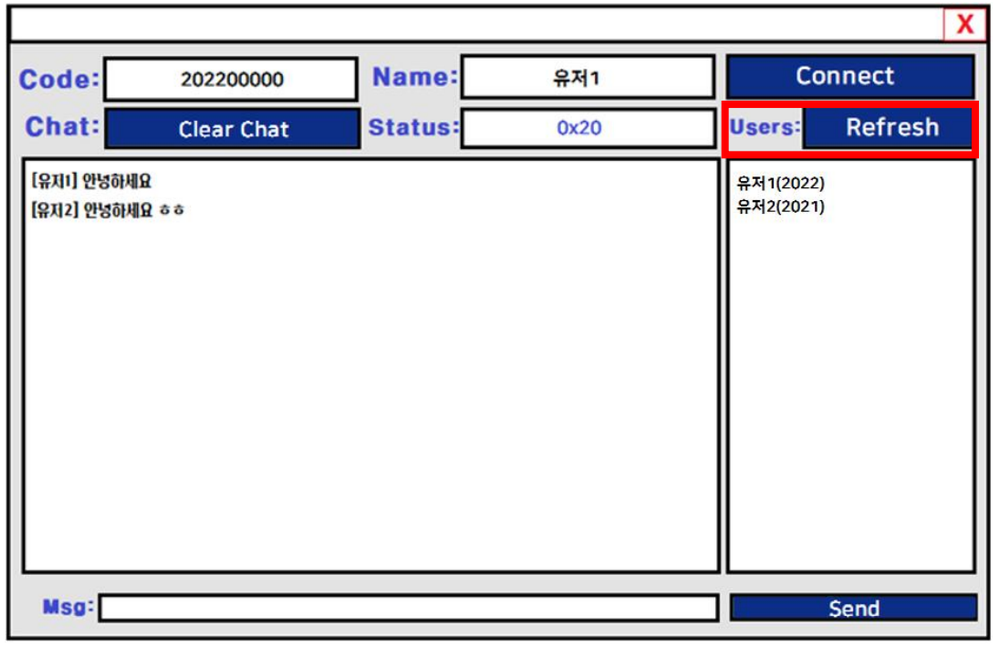


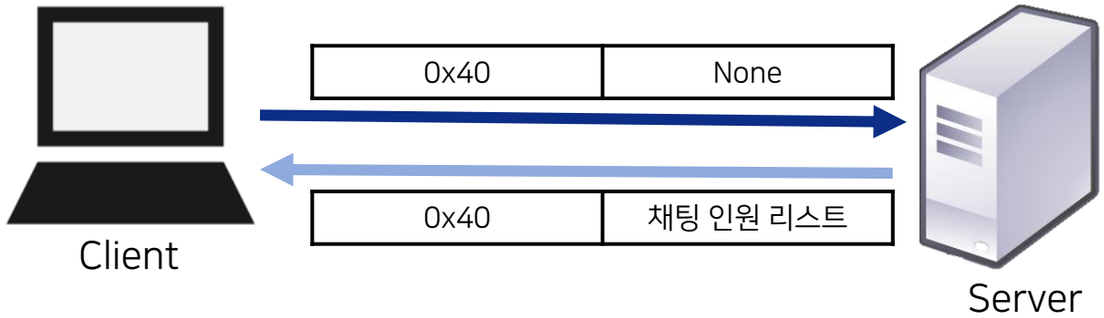
- 음식 키워드 예시: 치킨, 햄버거, 피자 등

(음식점 리스트에는 음식점 이름과 주소가 저장됨)

설계

- Protocol
  - **채팅 인원 확인 (0x40)**: Refresh 버튼을 클릭

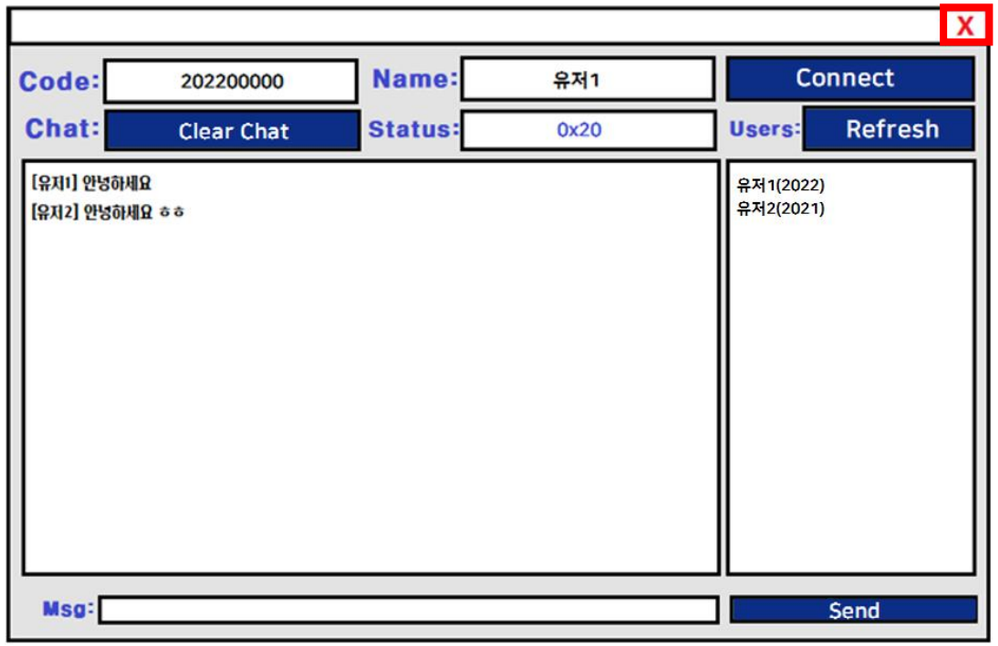


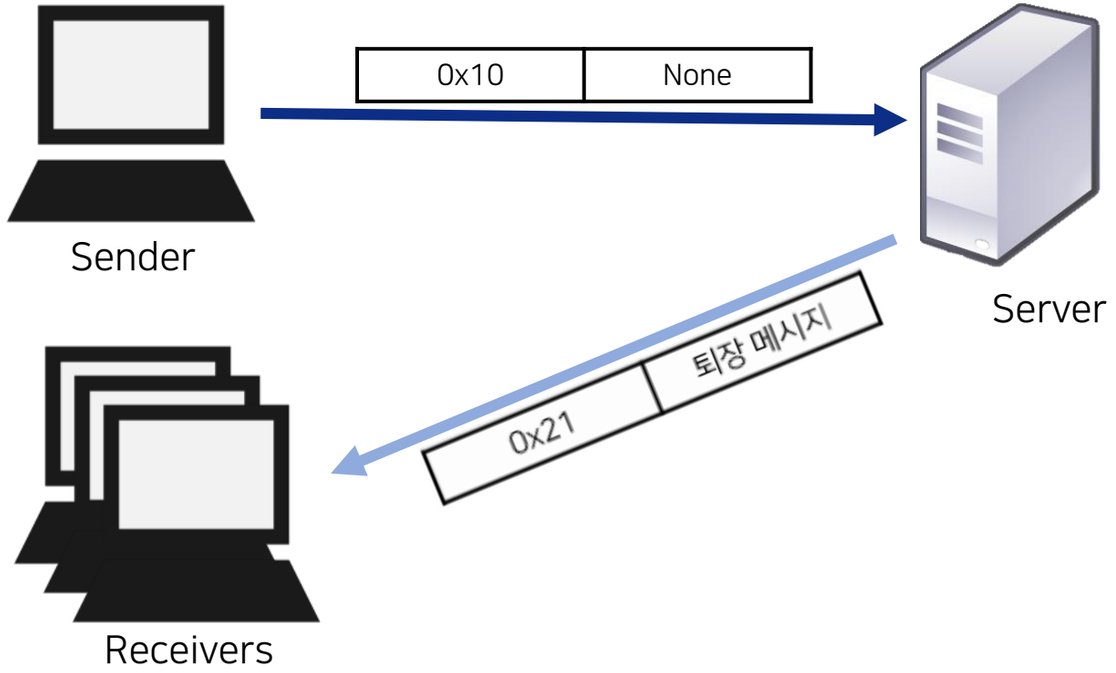


(채팅 인원 리스트에는 현재 온라인 상태인 유저의 닉네임과 학번 일부가 저장됨)

설계

- Protocol
  - 접속 해제 (0x10): 프로그램을 종료

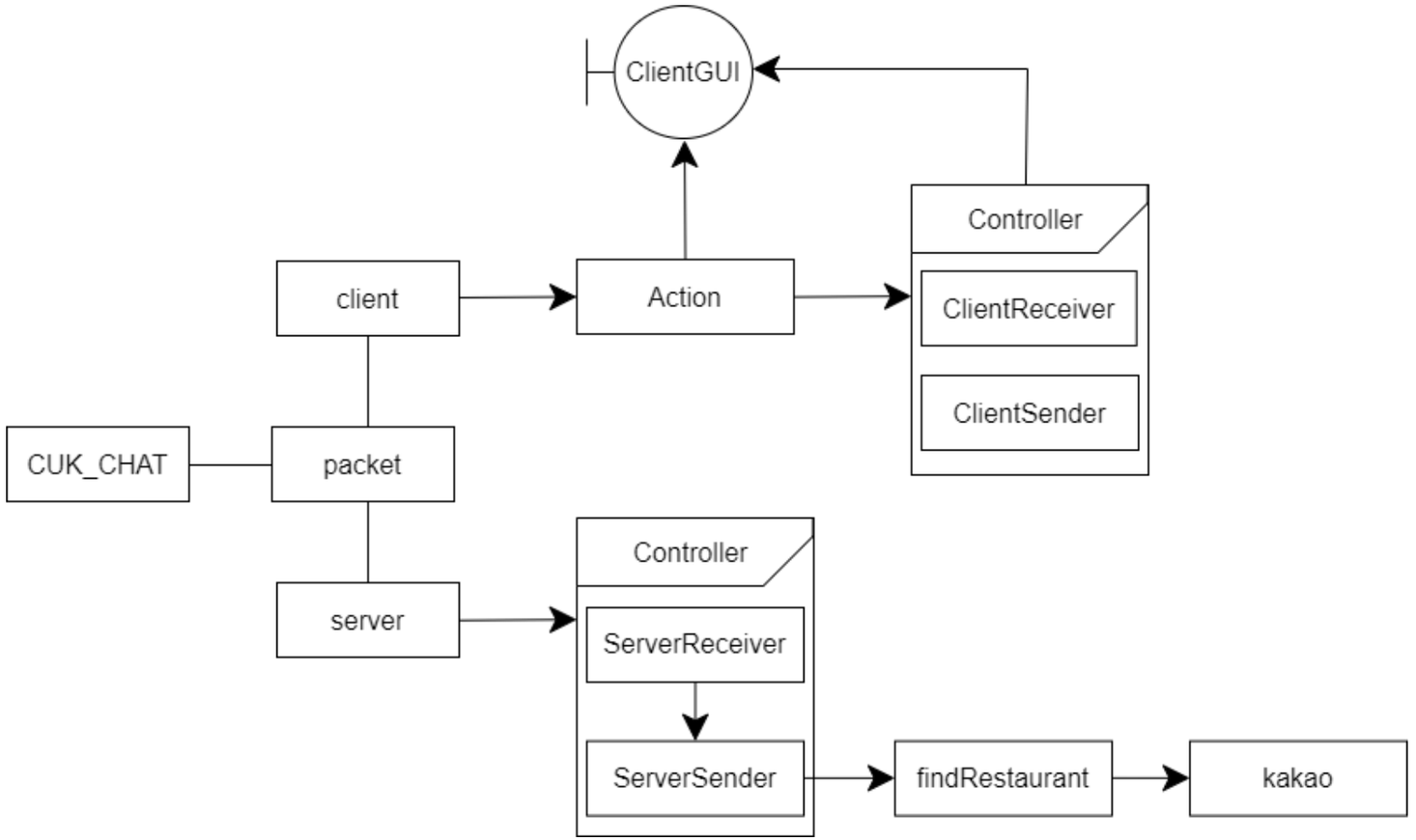




- 퇴장 메시지: “#(닉네임)님이 퇴장하셨습니다.”

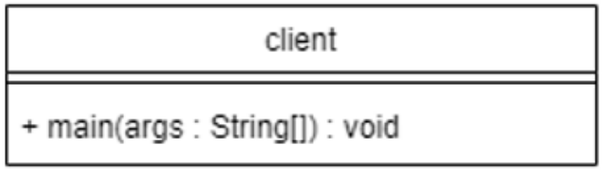
설계

- Class Diagram
  - 실제 구현할 때는 client, packet, server 단위로 제작함



설계

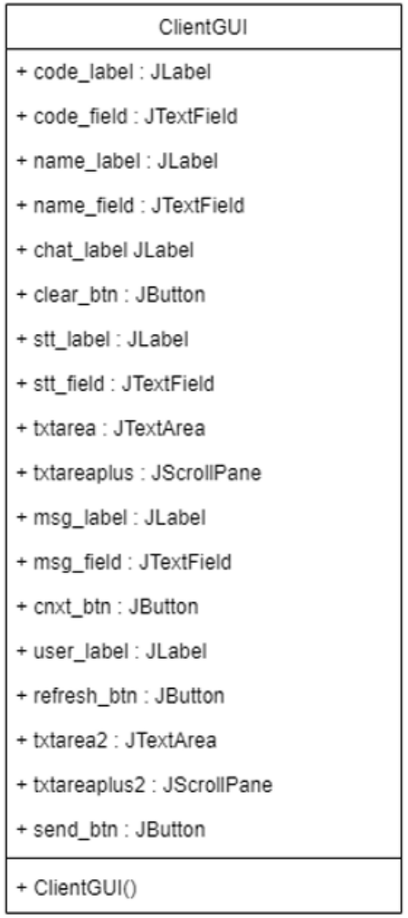
- Class Diagram (client)
  - 클라이언트 프로그램을 실행하는 클래스



클래스명	client	
메소드	main(args)	ClientGUI() 호출

설계

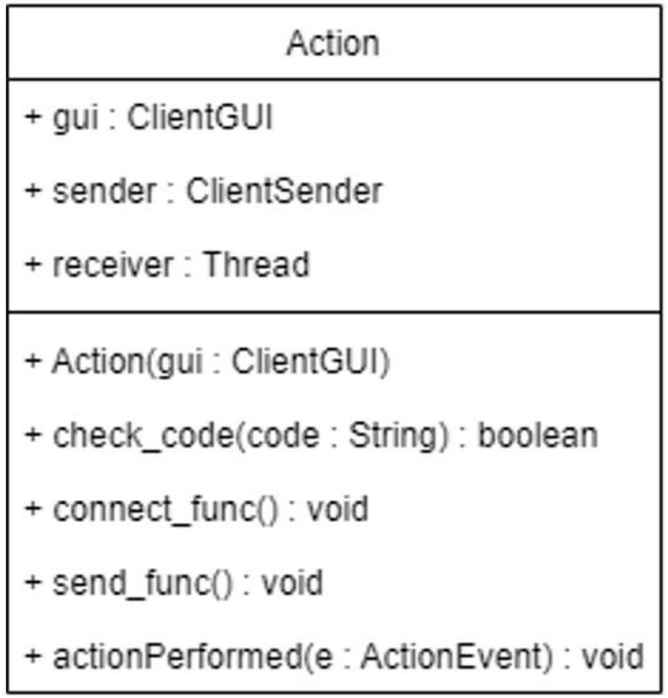
- Class Diagram (ClientGUI)
  - 클라이언트 프로그램을 GUI로 나타내는 클래스



클래스명	ClientGUI				
필드	code_label	학번 입력란 라벨	필드	txtareaplus	채팅창(스크롤)
	code_field	학번 입력란		msg_label	메시지 입력란 라벨
	name_label	닉네임 입력란 라벨		msg_field	메시지 입력란
	name_field	닉네임 입력란		cnxt_btn	접속 버튼
	chat_label	채팅 삭제 라벨		user_label	유저 확인 라벨
	clear_btn	채팅 삭제 버튼		refresh_btn	유저 확인 버튼
	stt_label	상태 확인창 라벨		txtarea2	유저 확인창
	stt_field	상태 확인창		txtareaplus2	유저 확인창(스크롤)
	txtarea	채팅창		send_btn	메시지 전송 버튼
			메소드	ClientGUI()	GUI 화면 표시

설계

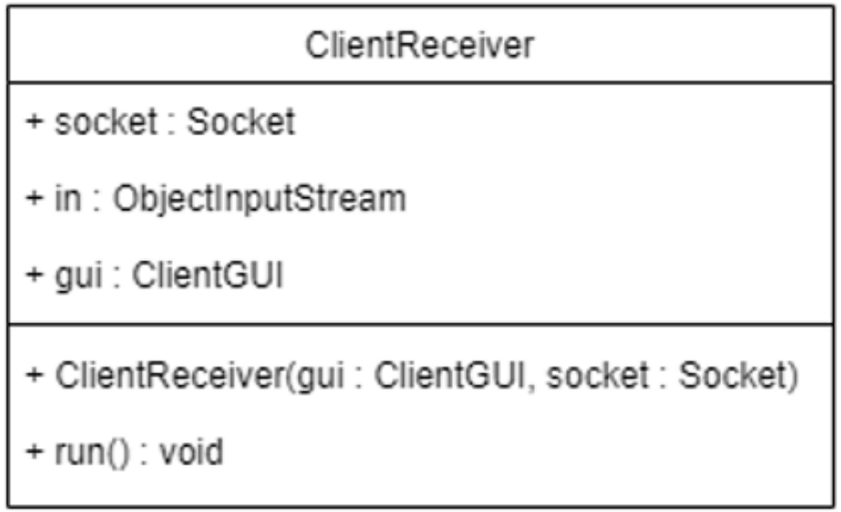
- Class Diagram (Action)
  - GUI의 컴포넌트에 기능을 부여하는 클래스



클래스명	Action	
필드	gui	GUI 화면
	sender	패킷 송신 객체
	receiver	패킷 수신 객체
메소드	Action(gui)	버튼에 따라 이벤트를 지정
	check_code(code)	올바른 학번인지 확인
	connect_func()	채팅 서버에 연결
	send_func()	메시지 전송
	actionPerformed(e)	버튼에 따라 이벤트를 지정

설계

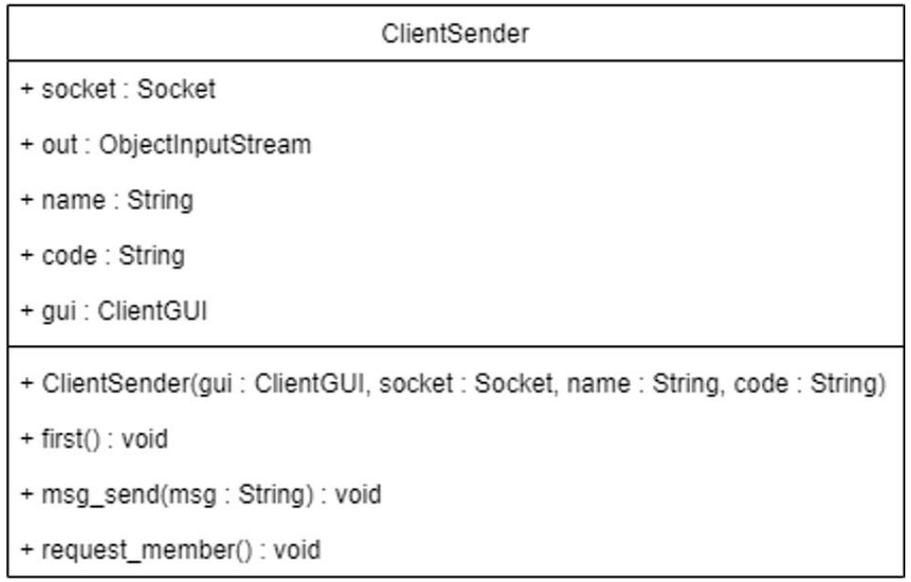
- Class Diagram (ClientReceiver)
  - 서버로부터 수신한 패킷을 분석해서 GUI 화면에 출력하는 클래스



클래스명	ClientReceiver	
필드	socket	통신을 위한 소켓
	in	서버로부터 패킷을 수신하는 객체
	gui	GUI 화면
메소드	ClientReceiver(gui, socket)	객체 필드 초기화 (수신 준비)
	Run()	패킷 수신 및 분석 및 GUI 출력

설계

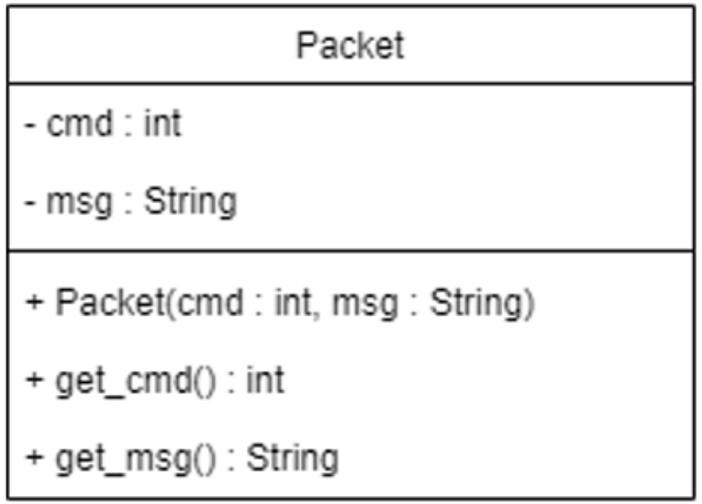
- Class Diagram (ClientSender)
  - 패킷을 생성한 후 서버에게 송신하는 클래스



클래스명	ClientSender	
필드	socket	통신을 위한 소켓
	out	서버에게 패킷을 송신하는 객체
	name	사용자 닉네임
	Code	사용자 학번
	gui	GUI 화면
메소드	ClientSender(gui, socket,name,code)	객체 필드 초기화 (송신 준비)
	first()	서버와 처음 통신을 위한 패킷 송신
	msg_send(msg)	메시지 전송을 위한 패킷 송신
	request_member()	채팅 인원 확인을 위한 패킷 송신

설계

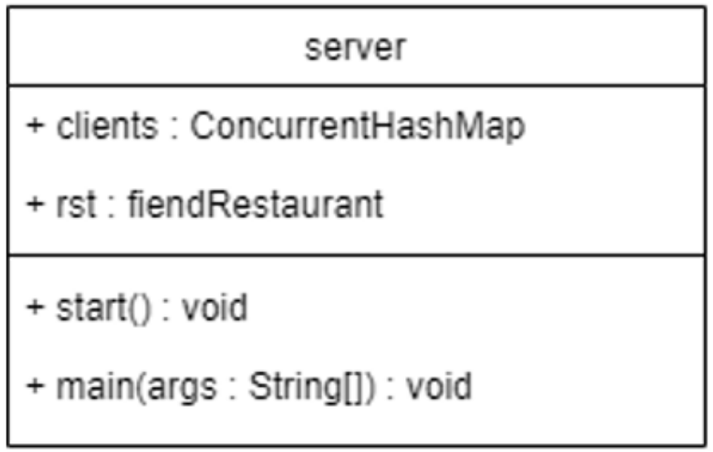
- Class Diagram (Packet)
  - 패킷에 관한 클래스



클래스명	Packet	
필드	cmd	명령어
	msg	메시지
메소드	Packet(cmd,msg)	패킷(Packet) 생성
	get_cmd()	패킷의 명령어 반환
	get_msg()	패킷의 메시지 반환

설계

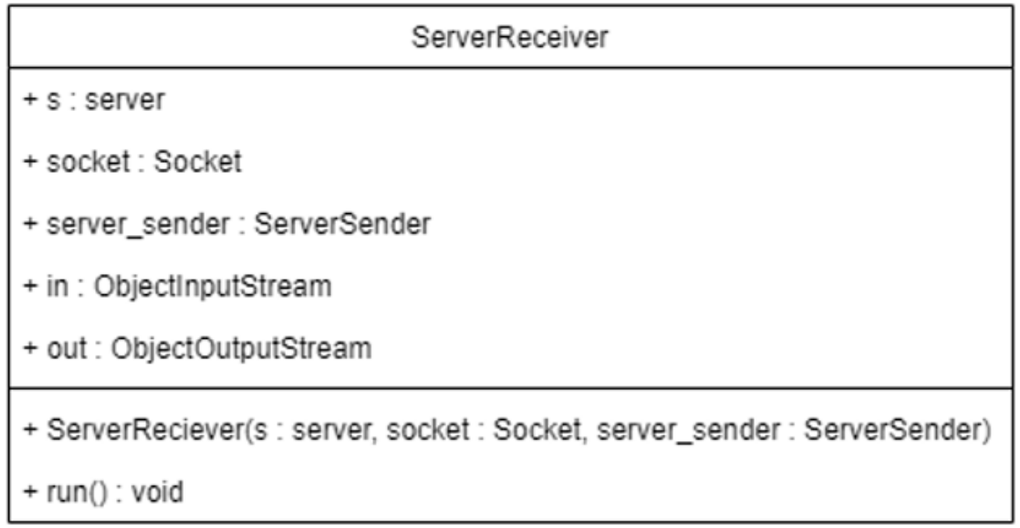
- Class Diagram (server)
  - 서버 프로그램을 실행하는 클래스



클래스명	server	
필드	clients	접속한 유저 목록
	rst	검색한 음식점 목록
메소드	start()	서버 프로그램 실행
	main(args)	start() 호출

설계

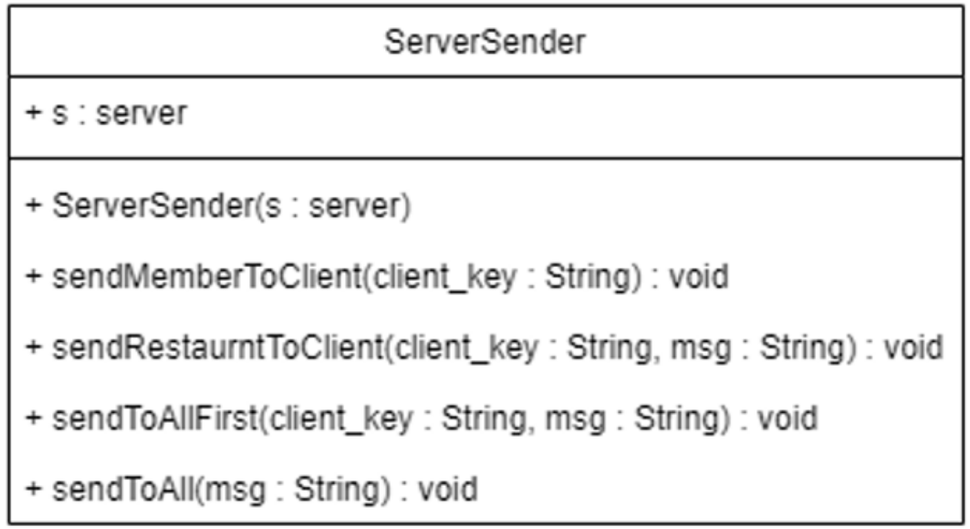
- Class Diagram (ServerReceiver)
  - 클라이언트로부터 수신한 패킷을 분석해서 Sender를 호출하는 클래스



클래스명	ServerReceiver	
필드	s	server 객체
	socket	통신을 위한 소켓
	server_sender	ServerSender 객체
	in	클라이언트로부터 패킷을 수신하는 객체
	out	클라이언트에게 패킷을 송신하는 객체
메소드	ServerReceiver(s,socket, server_sender)	객체 멤버 초기화 (수신 준비)
	run()	패킷 수신 및 분석 및 Sender 호출

설계

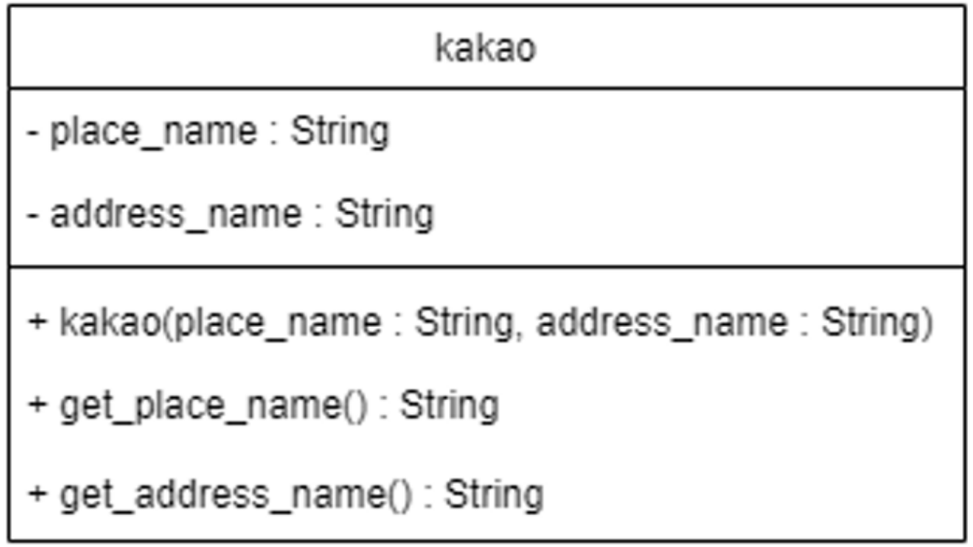
- Class Diagram (ServerSender)
  - ServerReceiver의 요청에 맞게 패킷을 생성 후 클라이언트에게 송신하는 클래스



클래스명	ServerSender	
필드	s	server 객체
메소드	ServerSender(s)	객체 멤버 초기화 (송신 준비)
	sendMemberToClient (client_key)	요청한 클라이언트에게 현재 유저 목록이 담긴 패킷 송신
	sendRestaurantToClient(cli ent_key,msg)	요청한 클라이언트에게 음식점 목록이 담긴 패킷 송신
	sendToAllFirst (client_key,msg)	모든 클라이언트에게 입장 메시지가 담긴 패킷 송신
	sendToAll(msg)	모든 클라이언트에게 메시지가 담긴 패킷 송신

설계

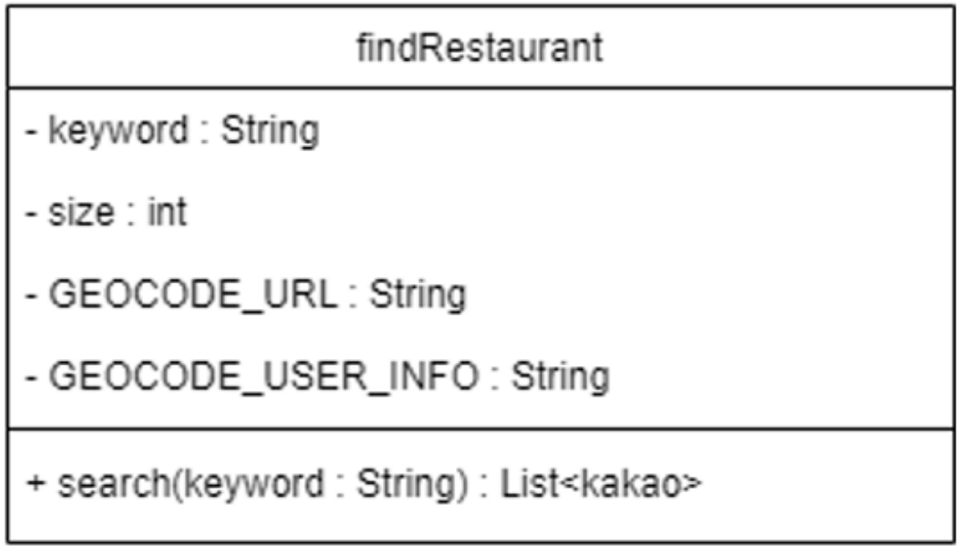
- Class Diagram (kakao)
  - 카카오 API로부터 얻은 음식점 정보에 관한 클래스



클래스명	kakao	
필드	place_name	음식점 이름
	address_name	음식점 주소
메소드	kakao(place_name, address_name)	음식점 정보 객체(kakao) 생성
	get_place_name()	음식점 이름 반환
	get_address_name()	음식점 주소 반환

설계

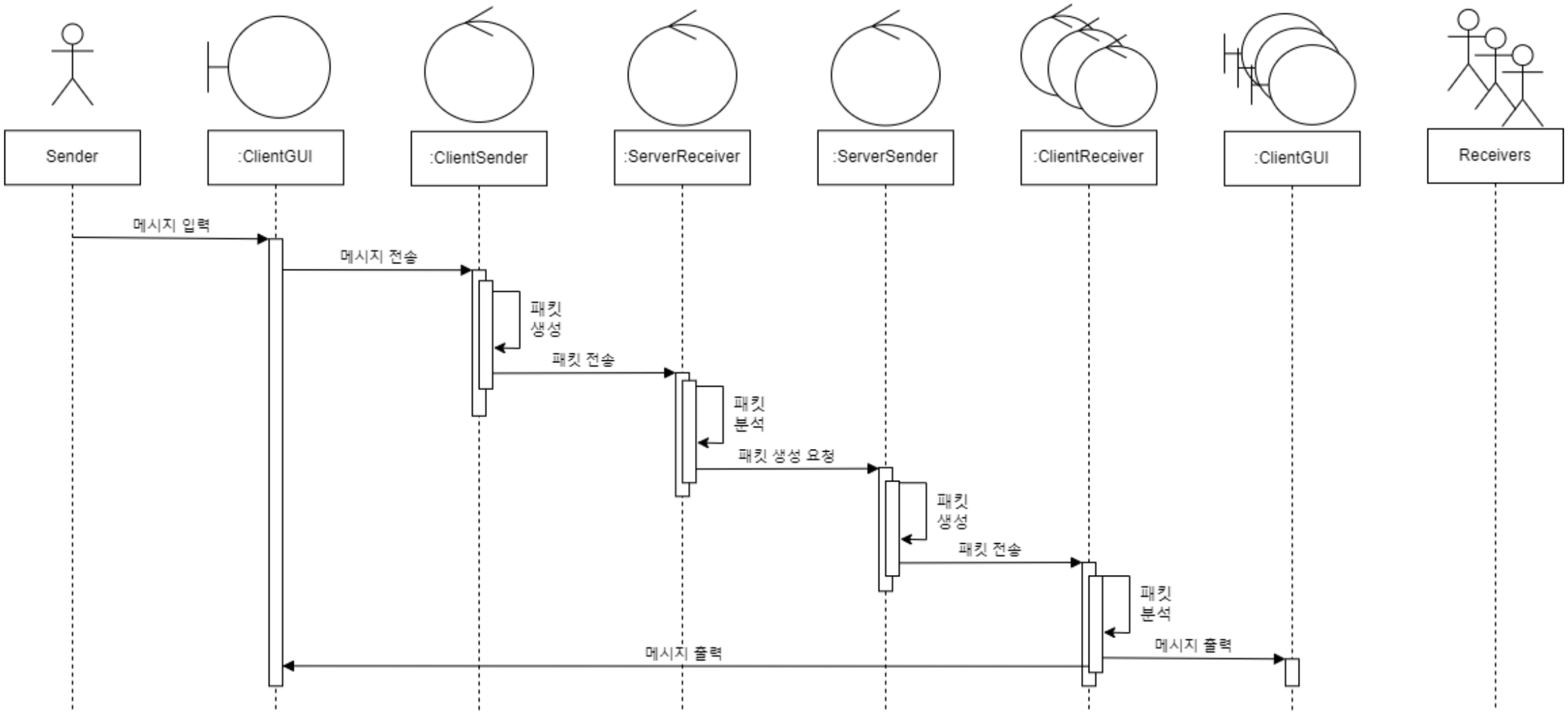
- Class Diagram (findRestaurant)
  - 카카오 API를 이용해서 음식점을 검색하는 클래스



클래스명	findRestaurant	
필드	keyword	음식 키워드
	size	키워드 사이즈
	GEOCODE_URL	카카오 API 링크
	GEOCODE_USER_INFO	카카오 API 개발자 코드
메소드	Search(keyword)	키워드로 음식점 목록 검색

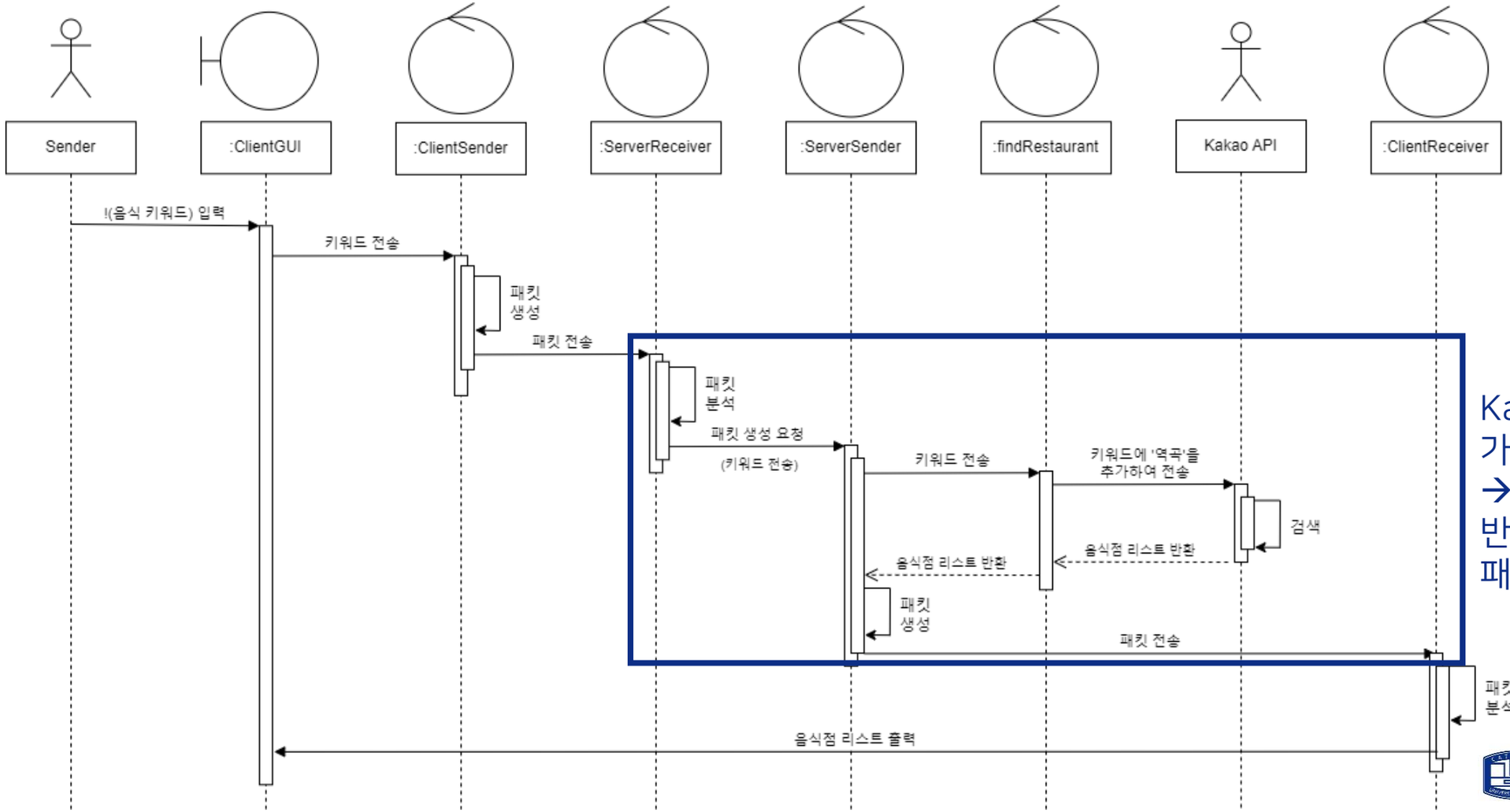
설계

- Sequence Diagram (1:N 채팅)



설계

- Sequence Diagram (음식점 검색)



Kakao API를 이용해서
가톨릭대 주변 음식점 검색
→
반환된 음식점 리스트로
패킷을 생성

구현

- GUI & Event
  - Swing 프레임워크로 GUI를 구성 & ActionListener로 컴포넌트별 이벤트 생성

```
class ClientGUI extends JFrame{
    // 컴포넌트 선언
    JLabel code_label;
    JTextField code_field;
    JLabel name_label;
    JTextField name_field;
    JLabel chat_label;
    JButton clear_btn;
    JLabel stt_label;
    JTextField stt_field;
    JTextArea txtarea;
    JScrollPane txtareaplus;
    JLabel msg_label;
    JTextField msg_field;
    JButton cnxt_btn;
    JLabel user_label;
    JButton refresh_btn;
    JTextArea txtarea2;
    JScrollPane txtareaplus2;
    JButton send_btn;

    public ClientGUI() {
        // GUI 기본 세팅
        setSize(670,420);
        setLocation(700,300);
        setTitle("CUK Chat Community");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        /*
         * 채팅 앱 디자인
         */

        // contentPane 정의
        JPanel contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
```

```
// 버튼 이벤트 처리
// (서버 연결, 메시지 전송, 인원 수 확인)
class Action implements ActionListener{
    ClientGUI gui;
    ClientSender sender;
    Thread receiver;

    // 생성자
    public Action(ClientGUI gui) {
        this.gui=gui;

        // 키 이벤트 생성 <Connect, Send>
        // <Connect>
        gui.name_field.addKeyListener(new KeyAdapter() {
            public void keyPressed(KeyEvent e) {
                if(e.getKeyCode()==e.VK_ENTER) {
                    connect_func(); // 서버 연결
                    gui.msg_field.requestFocusInWindow(); // 커서 위치 변경
                }
            }
        });
        // <Send>
        gui.msg_field.addKeyListener(new KeyAdapter() {
            public void keyPressed(KeyEvent e) {
                if(e.getKeyCode()==e.VK_ENTER) {
                    send_func(); // 메시지 전송
                }
            }
        });
    }
```

- ClientGUI와 Action 클래스 코드 일부

구현

- ObjectStream
  - 소켓 통신에서 객체 단위로 송/수신을 편하게 하도록 도와주는 객체

```java
//클라이언트에게 메시지를 보내는 역할
class ServerSender{
    server s;

     public ServerSender(server s) {
         this.s=s;
     }

     // 특정 클라이언트에게 송신(멤버 리스트) : 0x40
     void sendMemberToClient(String client_key) {
         String members="";
         Iterator it = s.clients.keySet().iterator();
         while(it.hasNext()) {
             members=members.concat(it.next()+"\n");
         }
         try {
             ObjectOutputStream out = (ObjectOutputStream)s.clients.get(client_key);
             out.writeObject(new Packet(0x40,members));
         }
         catch(Exception e){
              System.out.println("예외 메시지:"+e.getMessage());
         }
     }
```

```java
//수신 클래스
//서버의 메시지를 받는 역할 <스레드>
class ClientReceiver extends Thread{
      Socket socket;
      ObjectInputStream in;
      ClientGUI gui;

      // (서버와 수신하는) 스레드 객체의 멤버 초기화
      ClientReceiver(ClientGUI gui, Socket socket) {
          this.gui=gui;
          this.socket = socket;
          try {
              in = new ObjectInputStream(socket.getInputStream());  // 데이터 수신을 편하게 하도록 도와주는 객체 (서버 -> 클라이언트)
          }
          catch(Exception e) {
              System.out.println("예외 메시지:"+e.getMessage());
          }
      }

      @Override
      public void run() {
          // 반복 (수신 대기하다가 메시지 받으면 저장 -> TextArea에 출력)
          while(in!=null) {
              try {
                  Packet packet = (Packet)in.readObject();
                  String msg=packet.get_msg();

                  // 멤버 리스트
                  if(packet.get_cmd()==0x40) {
                      gui.txtarea2.append(msg);
                      gui.stt_field.setText("0x40");
                  }
```

- ServerSender와 ClientReceiver 클래스 코드 일부

구현

- Kakao API
  - Kakao 지역 검색 API로부터 음식점 정보(json)를 입력받아 파싱

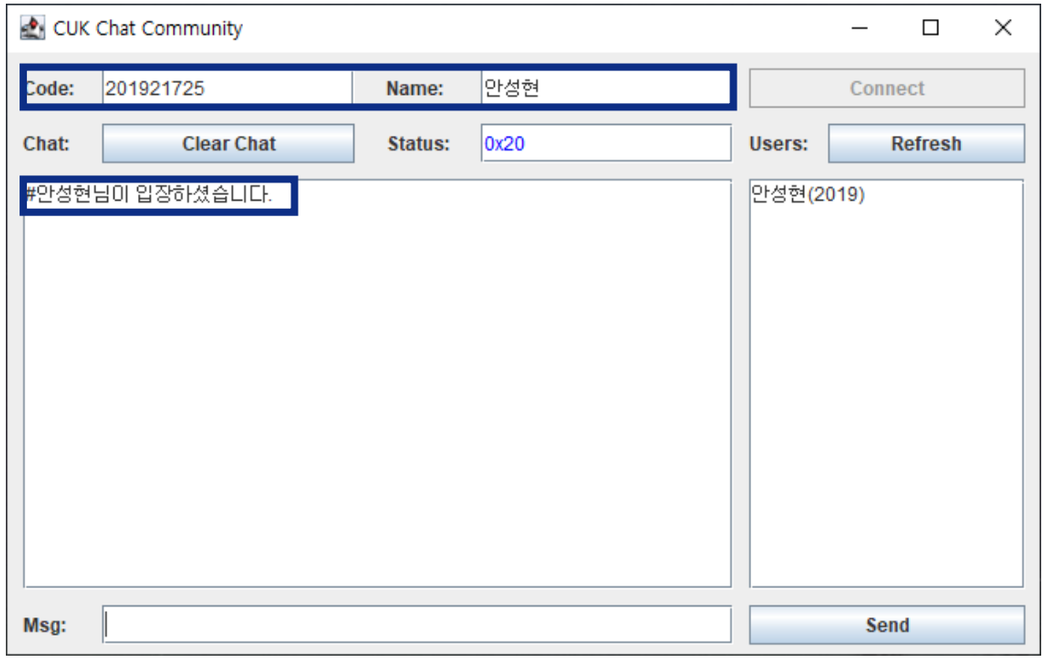
```
// Kakao API - 키워드로 지역 검색
keyword = URLEncoder.encode("역곡"+keyword, "UTF-8");
size=15;
GEOCODE_URL=String.format("http://dapi.kakao.com/v2/local/search/keyword.json?query=%s&size=%d",keyword,size);
System.out.println("검색 URL: "+GEOCODE_URL);

obj = new URL(GEOCODE_URL);
HttpURLConnection con = (HttpURLConnection)obj.openConnection();

con.setRequestMethod("GET");
con.setRequestProperty("Authorization",GEOCODE_USER_INFO);
con.setRequestProperty("content-type", "application/json");
con.setDoOutput(true);
con.setUseCaches(false);
con.setDefaultUseCaches(false);

Charset charset = Charset.forName("UTF-8");
BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream(), charset));

String inputLine;
StringBuffer response = new StringBuffer();

while ((inputLine = in.readLine()) != null) {
    response.append(inputLine);
}
String jsonString=response.toString();
```

- findRestaurant 클래스 코드 일부

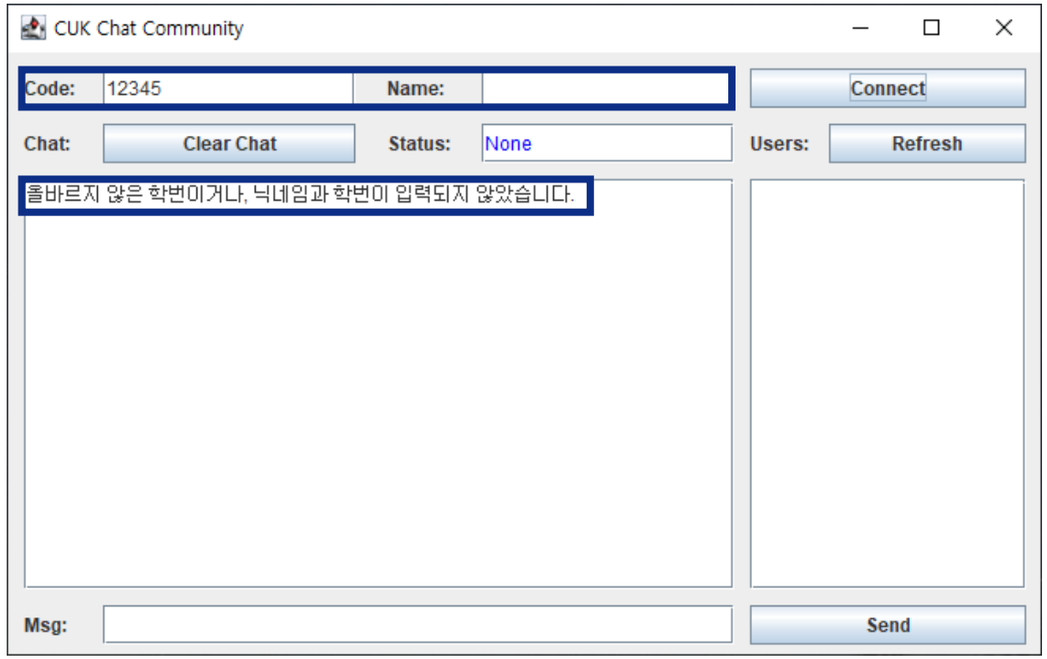
```
// Json 문자열 파싱 -> List<kakao>에 저장
JSONParser jsonParser = new JSONParser();
JSONObject jsonObject = (JSONObject) jsonParser.parse(jsonString);
JSONArray jsonArray = (JSONArray) jsonObject.get("documents");
try {
    for(int i=0; i<jsonArray.size(); i++){
        JSONObject objectInArray = (JSONObject) jsonArray.get(i);
        String place_name=(String) objectInArray.get("place_name");
        String address_name=(String) objectInArray.get("address_name");
        kakao k=new kakao(place_name,address_name);
        list.add(k);
    }
}
catch(Exception e){
    e.printStackTrace();
}
```

결과

- Connect (접속)
  - 학번 검사를 해서 올바르면 접속 허가, 그렇지 않으면 접속 불가



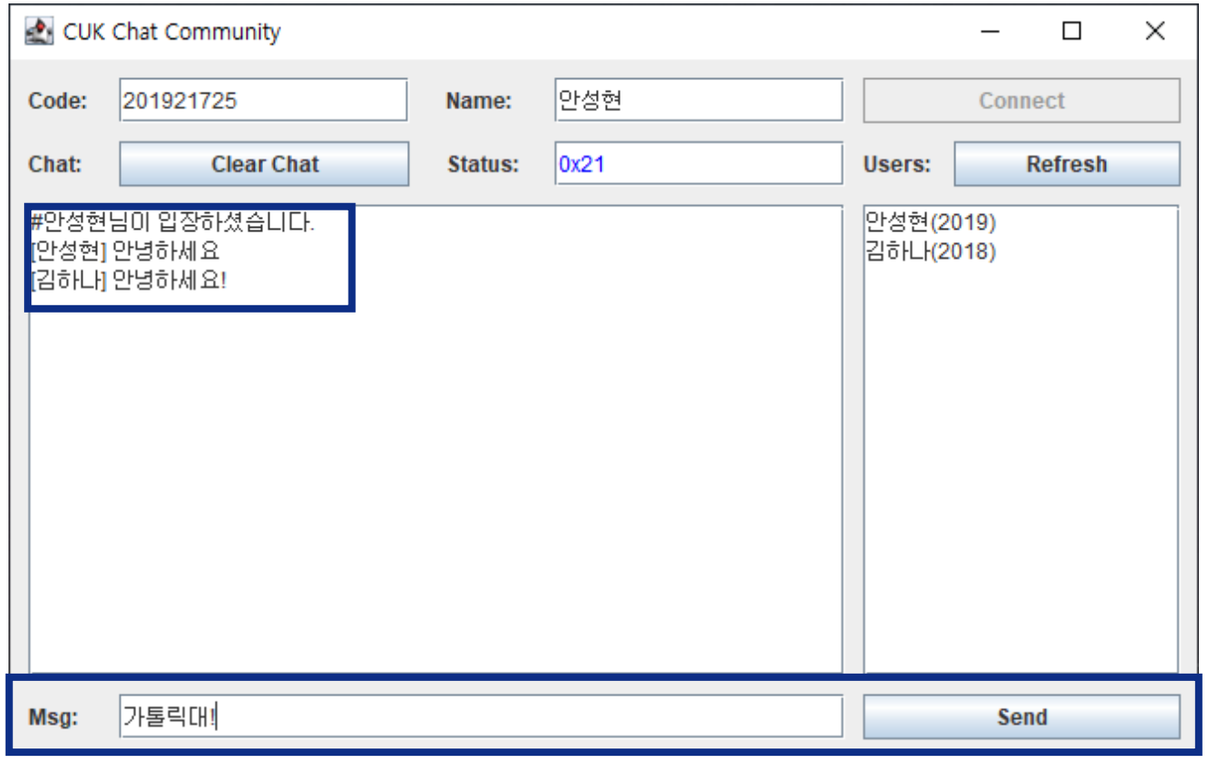
- 학번과 닉네임을 올바르게 입력한 경우



- 학번이 올바르지 않거나, 닉네임을 입력하지 않은 경우

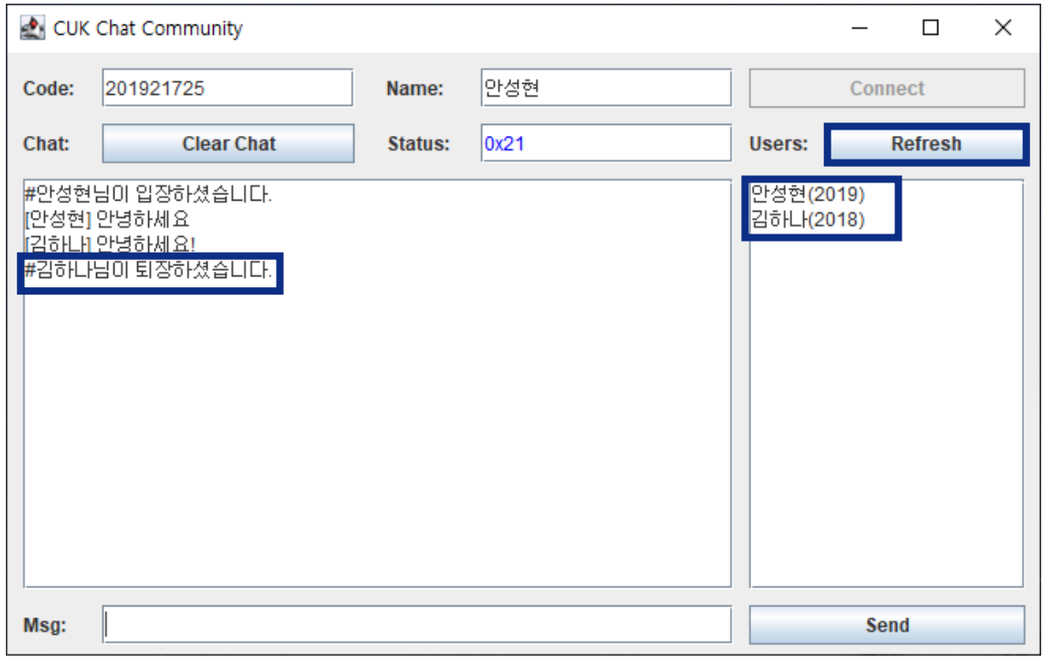
결과

- Chatting (채팅)
  - 채팅창과 메시지 입력란을 통해 채팅이 가능함

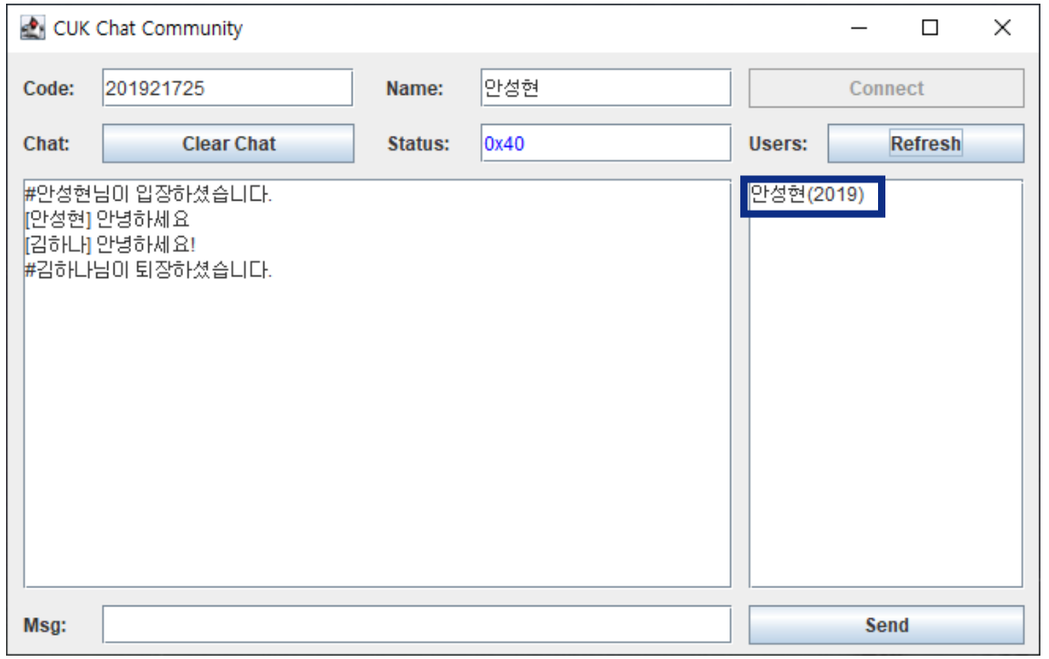


결과

- Refresh (현재 인원 확인)
  - 현재 채팅방에 참여한 인원 정보를 수동적으로 확인할 수 있음



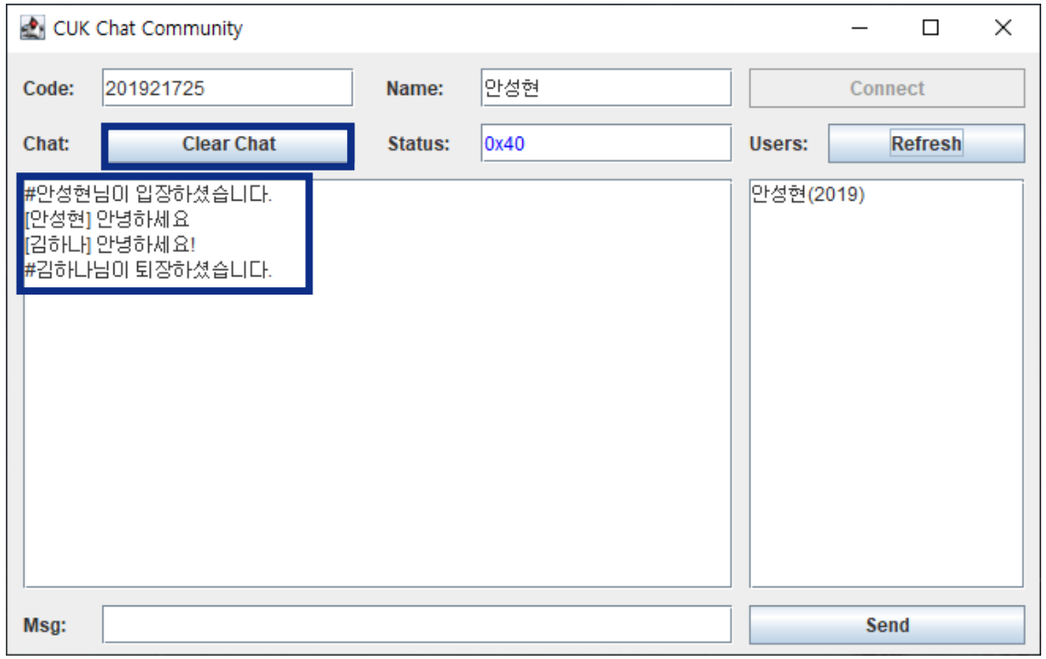
- 한 학우가 퇴장한 상황에서 Refresh 버튼을 클릭



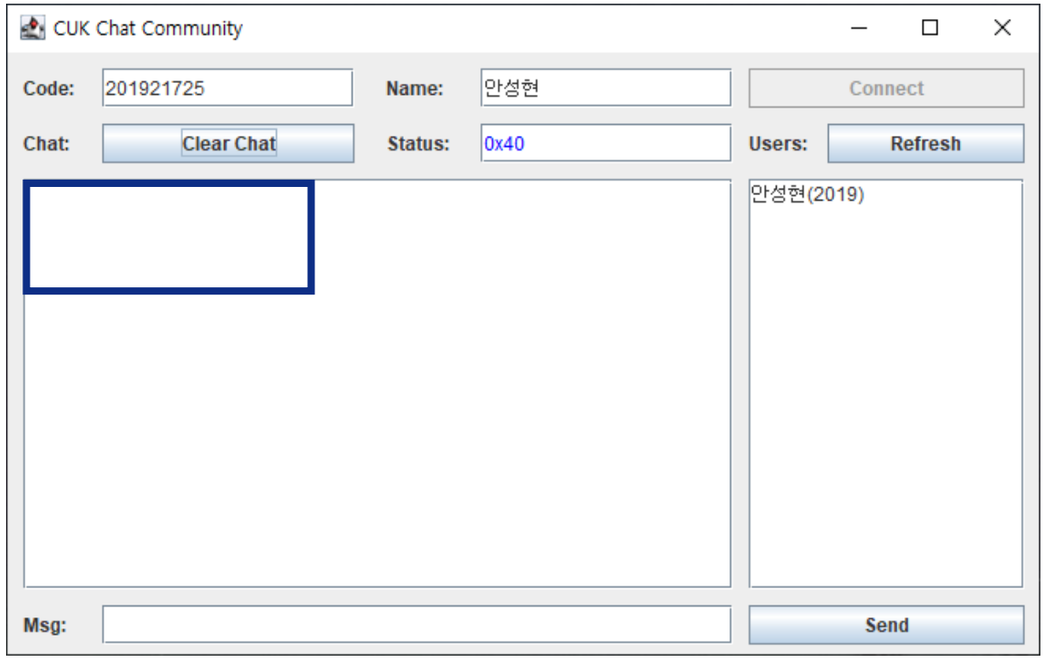
- 채팅방 인원 정보가 갱신됨

결과

- Clear Chat (채팅 내용 삭제)
  - 채팅창에 적힌 내용을 삭제할 수 있음



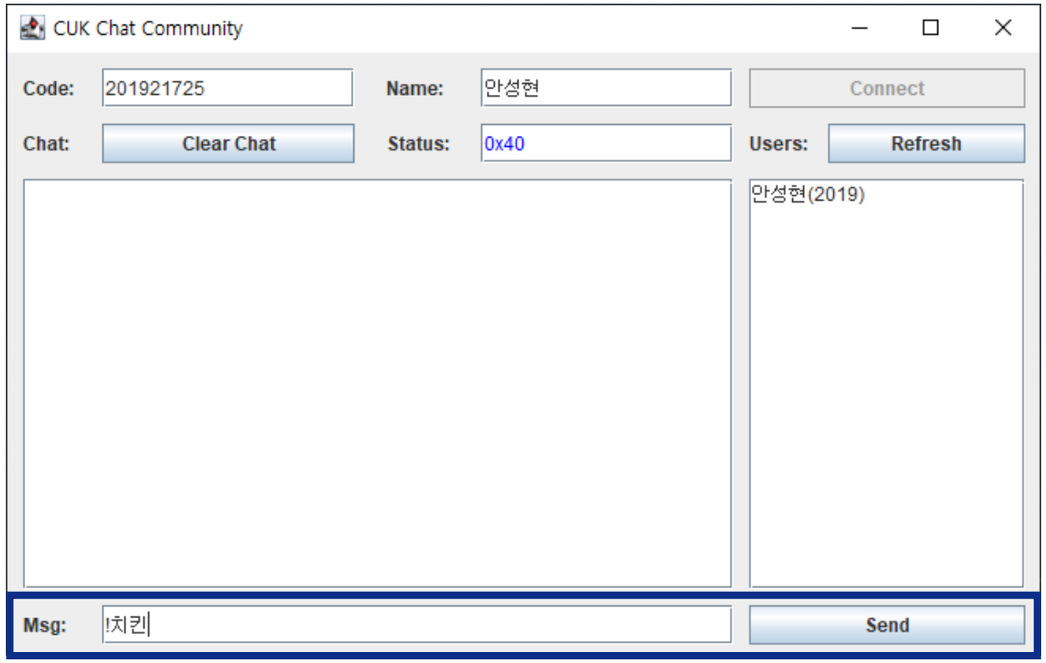
- 채팅 내용이 존재한 상태에서 Clear Chat 버튼을 클릭



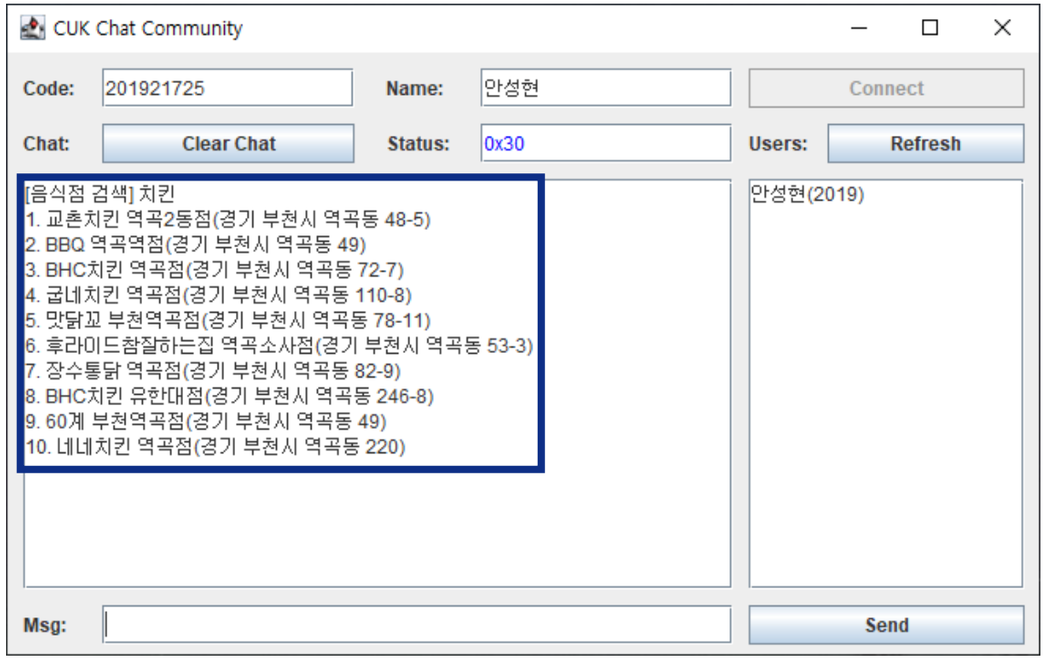
- 채팅 내용이 전부 삭제됨

결과

- Search (음식점 검색)
  - 메시지 입력란에 !(음식 키워드)를 입력해서 역곡 주변 음식점 목록을 얻음



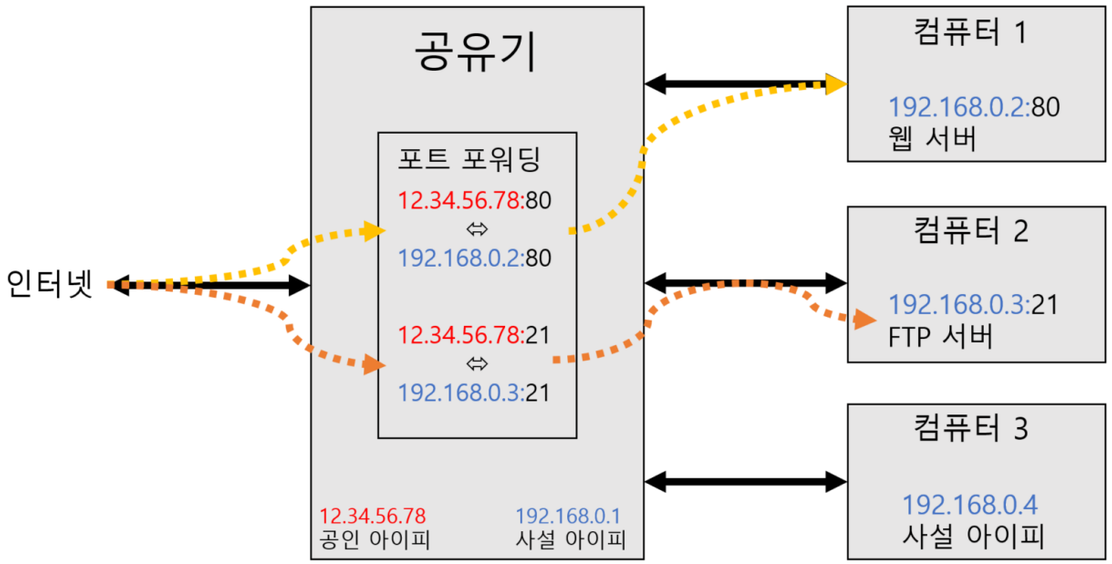
- !(음식 키워드)를 입력하고 Send 버튼을 클릭



- 채팅창에 역곡 주변 음식점 목록이 출력됨

기타

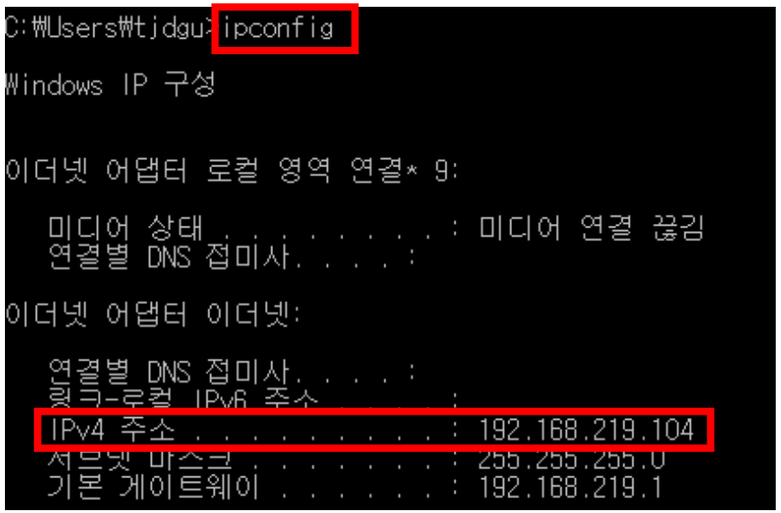
- **포트포워딩**
  - 외부에서 내 공유기와 연결된 컴퓨터의 프로그램으로 접속할 수 있게 하는 기술



ex) 외부에서 (외부 아이피: 12.34.56.78, 외부 포트: 80)로 연결을 시도하면,
해당 공유기의 외부 포트(80)와 연결된 컴퓨터1(내부 아이피:192.168.0.2)의 프로그램(내부 포트: 80)에 접속할 수 있음
[외부 포트와 연결된 PC와 내부 포트는 개발자가 설정]

기타

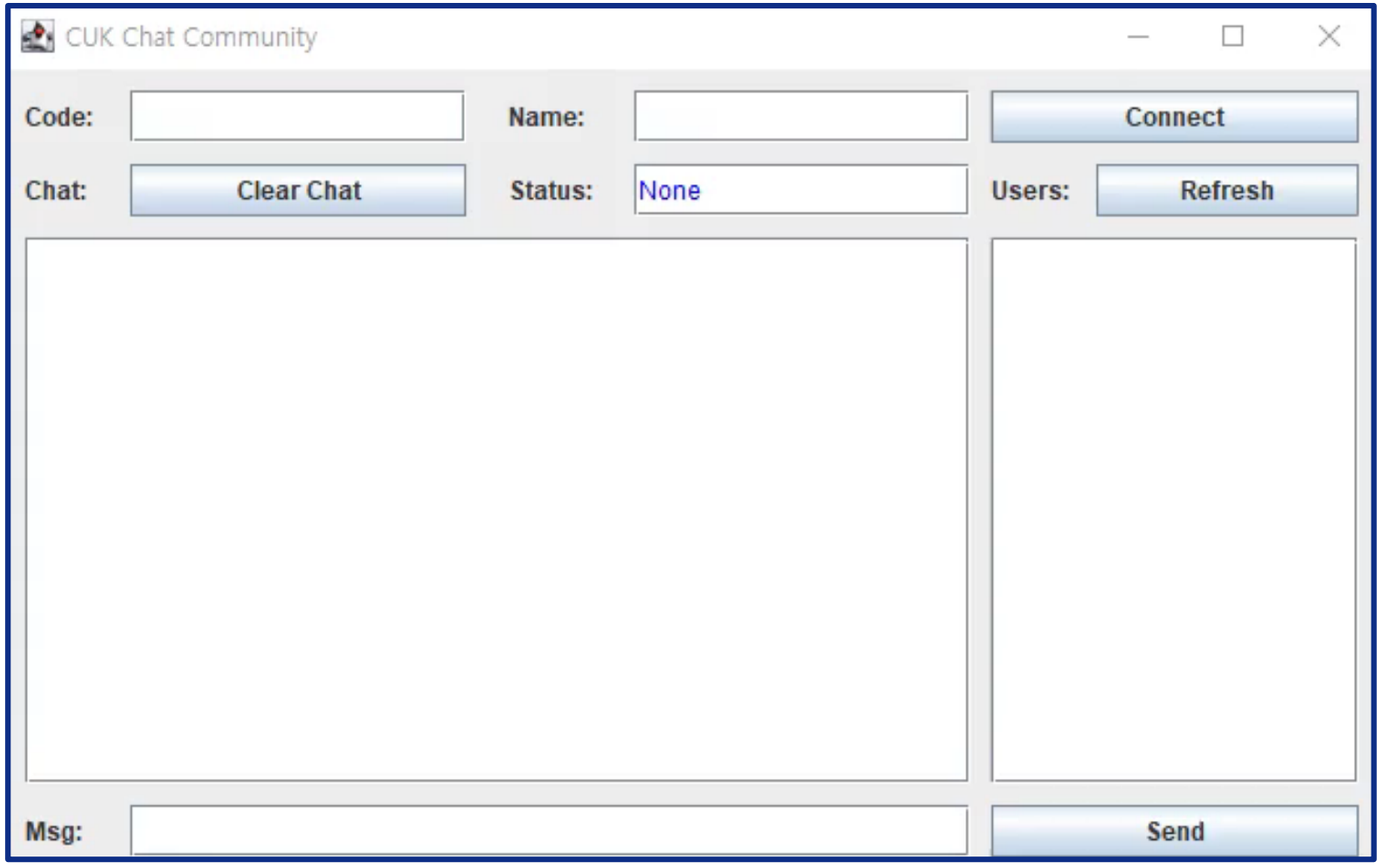
- 포트포워딩
  - LG U+ 공유기를 예시로 한 포트포워딩 설정



1. ipconfig 명령어로 내 PC의 내부 IP 확인
2. LG U+는 'http://192.168.219.1/'로 접속
3. 네트워크 설정 -> NAT 설정 클릭
4. 포트포워딩 추가
   1. 외부 포트: 7777
   2. 내부 포트: 7777
   3. 내부 IP 주소: 192.168.219.104
5. 설정 저장
6. 내 PC에서 7777 포트로 서버 프로그램 운영
   -> 외부에서 서버 프로그램에 접근이 가능!

데모

- 데모 영상
  - 접속 -> 채팅 -> (유저가 들어온 뒤) 인원 확인 -> 음식점 검색 -> (유저가 나간 뒤) 인원 확인



컴퓨터정보공학부 201921725 안성현

감사합니다

<2022/11/22>