

[우수 분석서 참조용] 요구사항 분석서

[클라이언트 동적 관리로 메모리의 효율을 높인  
GUI기반의 멀티 프로세스 채팅 프로그램]

Version 1.1	2019. 11. 05
Version 1.2	2019. 11. 07
Version 1.3	2019. 11. 08
Version 1.4	2019. 11. 09

학번: xxxxxxxx

작성자: 홍xx

## 1. 요구사항 분석

### 가. 서버 개발용

요구사항 아이디	S00
요구사항 명	DB연결 기능
상세 요구사항 내용	<ul style="list-style-type: none"><li>서버에서 채팅서비스 사용자 DB로 접근하기 위한 연결 기능 구현</li></ul>
이슈사항	<ul style="list-style-type: none"><li>채팅서비스 사용자 DBMS로의 연결을 위한 IP 및 Port 확인 필요</li></ul>
해결방안 (변경내역)	<ul style="list-style-type: none"><li>JDBC를 이용한 서버프로그램 내에서의 접근</li><li>서버 프로그램이 실행된 PC의 로컬 호스트와 DBMS기본 포트를 사용(localhost, Port : 3306)한 DBMS에 연결</li></ul>

요구사항 아이디	S01
요구사항 명	클라이언트에 대한 서버 접속 허용
상세 요구사항 내용	<ul style="list-style-type: none"><li>클라이언트들이 서버에 접속하기 위한 접근 허가 기능 구현</li></ul>
이슈사항	<ul style="list-style-type: none"><li>클라이언트의 수가 정해져 있지 않음</li><li>서버는 클라이언트에서 전송하는 문자열을 읽어 들여야 함</li></ul>
해결방안 (변경내역)	<ul style="list-style-type: none"><li>서버에서 ArrayList클래스를 이용해 클라이언트 객체들을 동적으로 관리</li><li>클라이언트에서 전송하는 문자열을 DataInputStream으로 읽어 들이는 클래스를 Thread로 관리하여 항상 대기하도록 설정</li><li>ServerSocket의 accept()메소드를 사용하여 클라이언트 객체의 서버 접속을 허가 함</li></ul>

요구사항 아이디	S02
요구사항 명	회원가입 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"><li>클라이언트 측에서 전송하는 회원가입 정보의 파싱 기능 구현</li><li>클라이언트 측에서 전송하는 회원가입 정보를 DB에 저장</li></ul>
이슈사항	<ul style="list-style-type: none"><li>클라이언트 측에서 회원가입용 문자열을 전송함을 인지</li><li>DB의 각 테이블에 회원 정보 저장</li></ul>
해결방안 (변경내역)	<ul style="list-style-type: none"><li>클라이언트로부터 전달받는 메시지 타입에 대한 정의 -&gt; 회원가입용 : SU(Signup)</li><li>StringTokenizer클래스를 이용해 문자열 형태의 회원 정보를 분리하여 관리</li><li>서버에서 미리 만들어 둔 DB용 insert문자열을 이용해 각 테이블에 회원 정보 저장</li></ul>

요구사항 아이디	S03
요구사항 명	아이디 중복확인 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>클라이언트 측에서 전송하는 아이디의 파싱 기능 구현</li> <li>클라이언트 측에서 전송하는 아이디가 DB에 존재하는지에 대한 질의문 전송 후 결과 도출</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측에서 중복확인용 문자열을 전송함을 인지</li> <li>DBMS에 정확한 질의문을 사용</li> <li>질의 결과를 한 줄씩 읽어 아이디 중복을 확인</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>클라이언트로부터 전달받는 메시지 타입에 대한 정의 -&gt; 중복확인용 : OL(Overlap)</li> <li>StringTokenizer클래스의 nextToken()메소드를 이용해 메시지 타입과 아이디를 분리</li> <li>만들어 둔 DB용 select문자열을 이용해 id가 존재하는 테이블에 대하여 질의를 함</li> <li>SELECT * 테이블명으로 전체가 나오도록 질의 후 ResultSet클래스의 getString() 메소드를 이용해 아이디 에트리뷰트를 차례로 읽은 후 중복을 확인</li> </ul>

요구사항 아이디	S04
요구사항 명	아이디 사용 허가 여부
상세 요구사항 내용	<ul style="list-style-type: none"> <li>아이디 중복확인이 필요한 클라이언트 측에 아이디 사용을 허가 여부를 전송</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측으로 결과와 메시지 타입을 함께 전달함</li> <li>중복확인을 요구한 클라이언트 측으로 정확히 전송해야 함</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>클라이언트로 보낼 메시지 타입에 대한 정의 (문자열의 앞부분에 메시지 타입을 붙여 전송) -&gt; 중복확인 : OV(Overlap)</li> <li>중복확인을 요구한 소켓으로 결과와 메시지 타입을 전송</li> </ul>

요구사항 아이디	S05
요구사항 명	아이디 찾기 정보 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>클라이언트 측에서 전송하는 정보에 대한 파싱 기능 구현</li> <li>클라이언트 측에서 전송하는 정보에 해당하는 아이디가 DB에 존재하는지에 대한 질의문 전송 후 결과 도출</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측에서 아이디 찾기용 문자열을 전송함을 인지</li> <li>DBMS에 정확한 질의문을 사용</li> <li>질의 결과를 한 줄씩 읽어 아이디 중복을 확인</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>클라이언트로부터 전달받는 메시지 타입에 대한 정의 -&gt; 아이디 찾기용 : FID(FindID)</li> <li>StringTokenizer클래스의 nextToken()메소드를 이용해 아이디를 찾으려는 회원 정보 문자열을 분리 -&gt; 이름, 전화번호 등</li> <li>만들어 둔 DB용 select문자열을 이용해 정보가 존재하는 테이블에 대하여 질의</li> <li>질의 후 ResultSet클래스의 getString()메소드를 이용해 아이디 에트리뷰트를 차례로 읽어서 동일한 문자열 확인</li> </ul>

요구사항 아이디	S06
요구사항 명	찾은 아이디에 대한 정보 전송
상세 요구사항 내용	<ul style="list-style-type: none"> <li>아이디를 찾으려는 클라이언트 측으로 DBMS에서 질의한 결과를 전송</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측에게 넘기는 정보가 아이디 찾기라는 것을 명시해야 함</li> <li>아이디 찾기를 요구한 클라이언트로 정확히 전송해야 함</li> <li>클라이언트 측이 넘겨준 정보가 DB에 존재하지 않을 시에 대한 대책 강구</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>클라이언트에 FID 메시지타입과 함께 결과 문자열을 전송</li> <li>회원 정보에 대한 질의 결과가 없다면 회원 등록이 되어있지 않은 것이므로 클라이언트 측으로 NM(NotExistMember) 메시지 전송</li> <li>아이디 찾기를 요구한 소켓으로 문자열 전송</li> </ul>

요구사항 아이디	S07
요구사항 명	비밀번호 찾기 정보 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>클라이언트 측에서 전송하는 정보에 대한 파싱 기능 구현</li> <li>클라이언트 측에서 전송하는 정보에 해당하는 비밀번호가 DB에 존재하는지에 대한 질의문 전송 후 결과 도출</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측에서 비밀번호 찾기용 문자열을 전송함을 인지</li> <li>정확한 질의문을 사용해야 함</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>클라이언트로부터 전달받는 메시지 타입에 대한 정의 -&gt; 비밀번호 찾기용 : FPW(FindPW)</li> <li>StringTokenizer클래스의 nextToken()메소드를 이용하여 회원 정보를 분리</li> <li>만들어 둔 DB용 select문자열을 이용해 정보가 존재하는 테이블에 대하여 질의</li> <li>질의 후 ResultSet클래스의 getString()메소드를 이용해 비밀번호 에트리뷰트를 차례로 읽어서 동일한 문자열 확인</li> </ul>

요구사항 아이디	S08
요구사항 명	찾은 비밀번호에 대한 정보 전송
상세 요구사항 내용	<ul style="list-style-type: none"> <li>비밀번호를 찾으려는 클라이언트 측에 DB에 질의한 결과를 전송</li> <li>클라이언트 측에서 전송하는 정보에 해당하는 비밀번호가 DB에 존재하는지에 대한 질의문 전송 후 결과 도출</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측에 전송하는 메시지가 비밀번호 찾기라는 것임을 명시</li> <li>비밀번호를 찾으려는 회원의 정보가 DB에 존재하지 않을 수 있음</li> <li>비밀번호 찾기를 요구한 클라이언트로 정확히 전송해야 함</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>클라이언트에 FPW 메시지타입과 함께 결과 문자열을 전송</li> <li>회원 정보에 대한 질의 결과가 없다면 회원 등록이 되어있지 않은 것이므로 클라이언트 측으로 NM(NotExistMember) 메시지 전송</li> <li>비밀번호 찾기를 요구한 소켓으로 문자열 전송</li> </ul>

요구사항 아이디	S09
요구사항 명	로그인 정보 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>클라이언트 측에서 전송하는 로그인 정보의 파싱 기능 구현</li> <li>클라이언트 측에서 전송하는 아이디와 비밀번호가 DB에 존재하는지에 대한 질의문 전송 후 결과 도출</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측에서 로그인용 문자열을 전송함을 인지</li> <li>질의의 결과 문자열(아이디, 비밀번호)을 분리</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>클라이언트로부터 전달받은 메시지 타입을 정의 -&gt; 로그인용 : LI(Login)</li> <li>비밀번호 찾기를 요구한 소켓으로 문자열 전송</li> </ul>

요구사항 아이디	S10
요구사항 명	로그인 정보에 대한 결과 전송
상세 요구사항 내용	<ul style="list-style-type: none"> <li>로그인 하려는 클라이언트 측에 DB에 질의한 결과를 전송</li> <li>클라이언트 측에서 전송하는 정보에 해당하는 비밀번호가 DB에 존재하는지에 대한 질의문 전송 후 결과 도출</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측으로 전송 시 로그인 정보의 결과를 전송하는 것임을 명시</li> <li>로그인 정보에 해당하는 정보가 DB에 존재하지 않을 수 있음</li> <li>비밀번호 찾기를 요구한 클라이언트로 정확히 전송해야 함</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>클라이언트에 LI 메시지타입과 함께 결과 문자열을 전송</li> <li>회원 정보에 대한 질의 결과가 없다면 회원 등록이 되어있지 않은 것이므로 클라이언트 측으로 NM(NotExistMember) 메시지 전송</li> <li>로그인을 하려고 한 소켓으로 문자열 전송</li> </ul>

요구사항 아이디	S11
요구사항 명	채팅방 생성 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅방 생성 처리 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측에서 채팅방 생성 전용 문자열을 전송함을 인지</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>클라이언트로부터 전달받은 메시지 타입을 정의 -&gt; 채팅방 생성용 : MC(MakeChatting)</li> <li>StringTokenizer클래스를 이용하여 메시지 타입과 문자열을 분리</li> <li>미리 정의해 둔 INSERT 문자열을 사용해 DB에 해당 정보 저장</li> </ul>

요구사항 아이디	S12
요구사항 명	채팅방 검색 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅방 검색 처리 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측에서 채팅방 검색 전용 문자열을 전송함을 인지</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>클라이언트가 전송한 문자열 앞에 "SC:"가 붙어 오기 때문에 StringTokenizer클래스를 이용해 문자열을 분리</li> <li>SELECT 문자열을 사용해 분리한 문자열 정보에 대하여 DB에 질의 후 결과 도출</li> </ul>

요구사항 아이디	S13
요구사항 명	채팅방 검색 결과 전송 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅방 검색 결과 전송 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측으로 채팅방 검색 전용 문자열을 전송함을 명시</li> <li>채팅방 검색 정보에 해당하는 정보가 DB에 존재하지 않을 수 있음</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>결과 문자열 앞에 채팅방 검색 전용 문자열인 "SC:"를 붙여 전송</li> <li>채팅방 검색 정보에 관한 질의 결과가 없다면 채팅방이 생성되지 않은 것이므로 클라이언트 측으로 NE(NotExist) 메시지 전송</li> </ul>

요구사항 아이디	S14
요구사항 명	채팅방 초대 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅방 초대를 보낸 클라이언트가 전송한 문자열 처리 기능 개발</li> <li>채팅방 초대를 받을 클라이언트에게 전송할 문자열 처리 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>초대를 보낸 클라이언트가 채팅방 초대용 문자열을 전송함을 인지</li> <li>초대를 받을 클라이언트에게 채팅방 초대용 문자열을 전송함을 명시</li> <li>초대를 받아야 할 클라이언트가 로그인 하지 않았을 수 있음 or 존재하지 않을 수 있음</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>채팅방 초대에 관한 메시지 타입 정의 -&gt; "CI:"</li> <li>클라이언트들이 보낸 문자열 형태의 정보를 StringTokenizer클래스로 분리</li> <li>분리한 문자열(초대 받을 ID)에 관해 SELECT 문자열을 사용해 DB의 로그인 여부가 저장된 테이블에 질의</li> </ul>

요구사항 아이디	S15
요구사항 명	채팅방 참여 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅방 참여 처리 기능 구현</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>비밀번호가 설정된 채팅방에 관한 예외처리 필요</li> <li>초대를 승낙한 사용자에게 관한 예외처리 필요</li> <li>초대를 받은 클라이언트가 채팅방 초대 승낙용 문자열을 전송함을 인지</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>채팅방 초대 승낙에 관한 메시지 타입 정의 -&gt; "JCI:"</li> <li>채팅방 참여에 관한 메시지 타입 정의 -&gt; "JC:"</li> <li>클라이언트가 전송한 문자열 앞이 "JC:"이면 StringTokenizer클래스로 문자열 분리 후 해당 사용자의 ID정보 문자열을 DB 채팅방 테이블에 삽입 -&gt; INSERT 문자열 사용</li> <li>참여하려는 사용자가 채팅방의 비밀번호까지 보낸다면 SELECT문으로 DB에 해당 비밀번호가 맞는지 질의 후 결과 도출 -&gt; 일치하면 채팅방 접속 허용</li> <li>"JCI:"가 들어왔을 경우 해당 채팅방이 비밀번호가 설정되어 있어도 해당 사용자를 채팅방에 들어가도록 함 -&gt; DB 채팅방 테이블에 INSERT 문자열을 사용해 초대 받은 사용자 정보를 삽입함</li> </ul>

요구사항 아이디	S16
요구사항 명	채팅방 채팅 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅방 채팅 처리 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측에서 채팅용 문자열을 전송함을 인지</li> <li>채팅방에 참여한 모든 클라이언트에게 해당 정보(채팅)가 전달 되어야 함</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>DB에서 해당 채팅방의 전체 인원을 검색하여 IP정보로 채팅을 전송(모든 참여 인원에게 채팅 표시)</li> <li>채팅용 메시지 타입 정의 -&gt; 채팅용 : "C:"</li> <li>해당 채팅의 내용을 StringTokenizer로 분리 후 메시지 부분만 DB에 INSERT 문자열을 사용해 DB에 저장</li> </ul>

요구사항 아이디	S17
요구사항 명	채팅방 나가기 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅방 나가기 처리 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>클라이언트 측에서 채팅방 나가기 전용 문자열을 전송함을 인지</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>채팅방 나가기 전용 메시지 타입 정의 -&gt; "EC:"</li> <li>메시지 타입과 문자열을 구분하기 위해 StringTokenizer 클래스를 사용</li> <li>DELETE 문자열을 사용해 구분된 문자열과 같은 정보를 DB 테이블에서 삭제함</li> </ul>

요구사항 아이디	S18
요구사항 명	닉네임 설정 처리 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 닉네임 설정 처리 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 클라이언트로부터 전송한 정보가 닉네임 설정의 정보라는 것을 인지</li> <li>• 해당 정보를 전송한 클라이언트의 ID정보가 필요</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• 클라이언트가 전송한 문자열에 "CN:"이 붙어있으므로 StringTokenizer클래스로 문자열 분리 후 처리 진행</li> <li>• 해당 클라이언트의 ID정보를 이용하여 어느 사용자의 닉네임을 변경해야 하는지 인지</li> <li>• 미리 선언해 둔 UPDATE 구문의 문자열을 사용하여 사용자 DB 정보를 변경</li> </ul>

#### 나. 클라이언트 개발용

요구사항 아이디	C00
요구사항 명	클라이언트의 서버 접속 요청 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 클라이언트에서 서버로 접속을 요청하는 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 클라이언트는 접속하려는 서버의 IP와 Port에 대한 정보 필요</li> <li>• 클라이언트는 서버에게 항상 메시지를 전송할 수 있어야 함</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• DataOutputStream클래스의 writeUTF()메소드와 Scanner의 nextLine()을 사용하여 항상 메시지를 전송할 수 있도록 함</li> <li>• 클라이언트에서 서버의 IP와 Port를 갖는 소켓 생성 new Socket(IP, Port) -&gt; Port는 7000으로 고정</li> </ul>

요구사항 아이디	C01
요구사항 명	회원가입 인터페이스
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 회원가입용 인터페이스 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 사용자의 아이디가 중복될 우려가 있음</li> <li>• 사용자가 무엇을 입력해야 하는지 표시</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• 텍스트필드로 ID, PW, 이름, 연락처를 입력받음</li> <li>• ID를 입력한 후에는 중복확인 버튼으로 중복 여부를 확인할 수 있도록 함</li> <li>• 확인한 중복 여부는 boolean타입으로 정보를 저장함</li> </ul>



요구사항 아이디	C02
요구사항 명	회원가입 정보 전송기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>회원가입 정보 전송기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>사용자가 각각의 정보에 올바르게 값을 입력할 수 있음</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>확인버튼을 누를 시 사용자가 입력한 모든 정보를 getText()로 읽어 오류를 확인</li> <li>중복 여부의 값이 저장된 boolean타입 변수를 확인</li> <li>비밀번호는 재입력과 비교하여 boolean타입의 변수에 결과를 저장</li> <li>이름에는 특수문자와 숫자가 입력되어선 안 됨</li> <li>연락처 TextField를 getText()로 읽은 후 parseInt()메소드를 이용해 int타입으로 변환, 그 후 ArithmeticException을 사용해 숫자로만 이루어질 수 있도록 예외처리</li> <li>모든 조건을 만족하는 정보는 문자열 형태로 합치고 문자열 앞에 "SU:"을 붙여 서버로 전송</li> </ul>

요구사항 아이디	C03
요구사항 명	로그인 인터페이스
상세 요구사항 내용	<ul style="list-style-type: none"> <li>로그인 인터페이스 개발</li> </ul>
이슈사항	X
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>사용자가 입력할 수 있는 ID와 PW TextField를 만들어 둠</li> <li>로그인 버튼을 입력 시 ID와 PW를 문자열 형태로 합치고 문자열 앞에 "LI:"을 붙여 서버로 전송</li> <li>"ID/PW찾기" 버튼과 "회원가입" 버튼을 만들어 두고 ActionListener의 actionPerformed()메소드를 사용해 각각에 알맞은 인터페이스 생성</li> </ul>

요구사항 아이디	C04
요구사항 명	아이디 중복확인 정보 전송
상세 요구사항 내용	<ul style="list-style-type: none"> <li>회원가입에서 아이디 중복확인 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>사용자가 아이디 형식에 맞지 않게 입력할 수 있음 ex) 특수문자</li> <li>중복확인이 되었다는 것을 표시하기 위한 기능 필요</li> <li>서버로 전송하는 정보가 중복확인이 필요한 문자열이라는 것을 명시</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>getText()메소드로 TextField에 있는 문자열을 읽어 옴</li> <li>특수문자로 이루어진 패턴을 만들어 "중복확인" 버튼이 눌릴 때 읽어 온 아이디에 특수문자가 있는지 확인</li> <li>특수문자가 있다면 사용자에게 JOptionPane으로 오류메시지 출력</li> <li>특수문자가 없다면 읽어 온 문자열의 앞에 "OL:"을 붙여 DataOutputStream클래스의 writeUTF()메소드를 통해 소켓이 연결된 서버로 전송</li> </ul>

요구사항 아이디	C05
요구사항 명	아이디 중복확인 결과 확인
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 서버로 넘겨 준 아이디 중복확인의 결과 확인 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 서버로부터 전송한 정보가 중복확인의 결과라는 것을 인지</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• 서버로부터 전송한 문자열 앞에는 “OL:”이 붙어서 옴</li> <li>• StringTokenizer클래스의 new StringTokenizer(“:”) 사용해 “OL”과 중복확인의 결과를 나눔</li> <li>• 중복확인의 결과 메시지에 따라 중복확인을 표시하기 위한 boolean값을 바꿈</li> </ul>

요구사항 아이디	C06
요구사항 명	비밀번호 재입력
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 회원가입에서 사용자에게 비밀번호 재입력 기능 표시</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 사용자가 입력한 비밀번호와, 비밀번호 재입력 란의 내용이 다를 수 있음</li> <li>• 회원가입에서 “확인” 버튼을 눌렀을 때 비밀번호와 재입력의 내용이 같다는 것을 표시할 수 있어야 함</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• 비밀번호 텍스트필드와 비밀번호 재입력 텍스트필드의 문자열을 getText()메소드로 읽어 들여 equals()메소드로 문자열을 비교</li> <li>• boolean타입의 변수를 선언해 두 문자열이 같다면 true값으로 변경</li> </ul>

요구사항 아이디	C07
요구사항 명	아이디 찾기 인터페이스
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 아이디 찾기 인터페이스 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 사용자가 올바른 정보를 입력할 수 있도록 해야 함 ex) 연락처 : 000 0000 0000, 이름 : 홍정인 등</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• 사용자로부터 정보를 TextField로 입력받을 수 있도록 함</li> <li>• 특수문자로 이루어진 패턴을 사용하여 입력받는 정보에 특수문자가 있는지 확인</li> <li>• 연락처는 띄어쓰기 없는 11자의 숫자여야 함</li> </ul>

요구사항 아이디	C08
요구사항 명	아이디 찾기 정보 전송
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 사용자가 입력한 정보를 서버로 전송</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 서버로 전송하는 정보가 아이디 찾기의 정보라는 것을 명시</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• TextField들의 문자열을 합친 후 아이디 찾기라는 것을 명시하기 위해 앞에 "FID:"를 붙임</li> <li>• "확인" 버튼을 누를 시 DataOutputStream클래스를 통해 문자열을 소켓으로 연결된 서버로 전송</li> </ul>

요구사항 아이디	C09
요구사항 명	비밀번호 찾기 인터페이스
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 비밀번호 찾기 인터페이스 구현</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 사용자가 올바른 정보를 입력할 수 있어야 함 ex) 주민번호 010101-4111411 등</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• 사용자로부터 정보를 TextField로 입력받을 수 있도록 함</li> <li>• 특수문자로 이루어진 패턴을 사용하여 입력받는 정보에 특수문자가 있는지 확인</li> <li>• 연락처는 띄어쓰기 없는 11자의 숫자여야 함</li> <li>• 주민등록번호는 각각 6자와 7자의 숫자 형태의 문자열이어야 함</li> </ul>

요구사항 아이디	C10
요구사항 명	비밀번호 찾기 정보 전송
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 사용자가 입력한 정보를 서버로 전송</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 서버로 전송하는 정보가 비밀번호 찾기의 정보라는 것을 명시</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• TextField들의 문자열을 합친 후 아이디 찾기라는 것을 명시하기 위해 앞에 "FPW:"를 붙임</li> <li>• "확인" 버튼을 누를 시 DataOutputStream클래스를 통해 문자열을 소켓으로 연결된 서버로 전송</li> </ul>

요구사항 아이디	C11
요구사항 명	아이디/비밀번호 찾기 결과 인터페이스
상세 요구사항 내용	<ul style="list-style-type: none"> <li>입력한 정보에 해당하는 결과를 출력</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>서버로부터 넘어온 정보가 아이디 찾기의 결과인지 비밀번호 찾기의 결과인지 구분해서 넘어온다는 것을 인지</li> <li>해당 결과를 보여줄 때 보안을 위한 표시기능 필요</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>서버에서 넘어온 문자열 형태의 정보 앞부분이 "FID:"이면 아이디 찾기의 결과이고 "FPW:"이면 비밀번호 찾기의 결과임</li> <li>아이디나 비밀번호를 사용자에게 보여줄 때 뒤의 4자리는 "*"로 표시</li> </ul>

요구사항 아이디	C12
요구사항 명	로그인 정보 전송
상세 요구사항 내용	<ul style="list-style-type: none"> <li>사용자가 입력한 아이디와 비밀번호를 서버로 전송</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>서버로 전송할 정보가 로그인 정보라는 것임을 명시</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>"로그인" 버튼을 누를 시 사용자가 입력한 아이디와 비밀번호를 getText()로 읽어 들여 합친 후 앞에 "LI:"을 붙인 문자열 생성</li> <li>해당 문자열을 DataOutputStream클래스를 통해 소켓으로 연결된 서버로 전송</li> </ul>

요구사항 아이디	C13
요구사항 명	채팅 대기방 인터페이스
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅 대기방 인터페이스 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>대기방에 존재하는 사용자에게 채팅방 초대 메시지가 올 수 있음</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>서버로부터 전송한 메시지의 앞 문자열이 "CI:"(ChattingInvite)이면 채팅방 초대 인터페이스 생성</li> <li>채팅 대기방은 채팅방 목록, 채팅방 생성, 채팅방 참여, 옵션 등으로 이루어짐</li> <li>채팅방 목록은 채팅방 제목, 방장 그리고 참여 인원수가 나타내어짐</li> </ul>

요구사항 아이디	C14
요구사항 명	채팅방 생성
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 채팅방 생성 기능 개발</li> <li>• 채팅방 생성은 채팅방 이름, 비밀번호로 이루어짐</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 채팅방 이름은 중복이 될 수 있고, 비밀번호는 생략 가능 p.s)채팅방 이름은 생략되어선 안 됨</li> <li>• 서버로 정보를 전송할 때 채팅방 생성의 정보라는 것을 명시</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• 채팅방 목록에서 "채팅방 생성" 버튼을 누를 시 채팅방 생성 인터페이스 생성</li> <li>• 생성시 채팅방 이름을 getText()로 읽어와 equals("")를 사용해 빈칸인지 확인 (빈칸일 경우 생성할 수 없도록 함)</li> <li>• 채팅방 이름과 비밀번호 정보는 서버를 통해 DB에 저장</li> <li>• 서버로 정보를 전송할 때 넘어가는 문자열 앞에 "MC:"(MakeChatting)을 붙임</li> </ul>

요구사항 아이디	C15
요구사항 명	채팅방 검색
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 채팅방 검색 기능 개발</li> <li>• 채팅방 검색은 채팅방 이름, 방장(닉네임)으로 이루어짐</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 서버로 정보를 전송할 때 채팅방 검색의 정보라는 것을 명시</li> <li>• 검색 시에 사용자가 정보를 입력하지 않았을 수 있음</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• 채팅방 목록에서 "채팅방 검색" 버튼을 누를 시 채팅방 검색 인터페이스 생성</li> <li>• "검색" 버튼을 누를 시 채팅방 이름, 방장의 문자열을 getText()메소드로 읽어 온 후 둘 중 하나의 문자열이 equals("")일 경우는 나머지 하나의 값(문자열)만 가지고 서버로 전송</li> <li>• 둘 다 빈칸일 경우는 검색기능이 동작하지 않도록 함</li> <li>• 전송 시에 문자열 앞에 "SC:"(SearchChatting)을 붙여서 전송함</li> </ul>

요구사항 아이디	C16
요구사항 명	채팅방 초대
상세 요구사항 내용	<ul style="list-style-type: none"> <li>• 채팅방 초대 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>• 서버로 정보를 전송할 때 채팅방 초대라는 것을 명시</li> <li>• 초대를 받을 사람에 관한 정보 필요</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>• "채팅방 초대" 버튼을 누를 시 사용자는 초대할 사람의 ID정보를 입력함</li> <li>• 입력한 ID정보의 앞에 채팅방 초대라는 것을 명시하기 위해 "CI:"를 붙여서 서버로 전송</li> </ul>

요구사항 아이디	C17
요구사항 명	채팅방 참여
상세 요구사항 내용	<ul style="list-style-type: none"> <li>검색의 결과 목록을 사용자에게 보여줌</li> <li>채팅방 참여 기능 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>서버로 정보를 전송할 때 채팅방 참여의 정보라는 것을 명시</li> <li>참여하려는 채팅방에 비밀번호가 설정되어 있을 수 있음</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>검색 결과 목록 중에서 “채팅방 참여” 버튼을 누를 시 해당 채팅방의 인터페이스 생성</li> <li>사용자가 비밀번호가 설정된 방에 참여하려면 해당하는 비밀번호를 입력해야 함 → 이 경우엔 비밀번호가 맞는지 DB에 질의해야 하므로 입력한 정보를 서버에 전송해야 함</li> <li>사용자가 채팅방에 참여하면 해당 사용자의 정보(아이디, 닉네임 등)를 서버로 전송하여 서버는 DB에 저장</li> <li>전송 시에 문자열 앞에 “JC:”(JoinChatting)을 붙여서 전송함</li> </ul>

요구사항 아이디	C18
요구사항 명	채팅방 인터페이스
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅방 인터페이스 개발</li> <li>채팅방 인터페이스는 채팅창(입력란, 채팅 표시창, 전송), 참여 인원 목록, 인원 초대, 옵션 등으로 구성됨</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>채팅방에 참여하는 모든 인원에게 같은 인터페이스를 표시</li> </ul>
해결방안 (변경내역)	X

요구사항 아이디	C19
요구사항 명	채팅 기능
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅방 인터페이스의 채팅창 기능(입력란, 채팅 표시창, 전송) 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>서버로 전송하는 정보가 채팅이라는 것을 명시해야 함</li> <li>서버로부터 전송한 정보가 채팅이라는 것을 인지해야 함</li> <li>채팅방에 참여한 모든 사람에게 채팅이 보여야 함</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>사용자가 “전송” 버튼을 누를 시 입력란의 데이터를 getText()메소드로 읽어 들여 서버로 전송 → 전송 시 문자열의 앞에 “C:”(Chatting)을 붙이고 전송</li> <li>채팅의 정보를 전송할 때는 해당 채팅방의 정보(참여 인원의 정보 등)를 같이 전송</li> <li>서버가 DB를 거쳐 다시 채팅방으로 정보를 전송할 때 “C:”를 앞에 붙여서 오기에 StringTokenizer클래스를 이용해 분리하여 정보를 저장</li> </ul>

요구사항 아이디	C20
요구사항 명	채팅방 옵션
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅방의 옵션의 기능 개발</li> <li>옵션에는 채팅방 나가기, 프로필 변경 등이 있음</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>서버로 전송하는 정보가 채팅방 나가기라는 것을 명시해야 함</li> <li>사용자가 채팅을 나갔을 시 DB정보를 갱신해야 함</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>사용자가 채팅방을 나가지 않고 프로그램을 종료할 시에 채팅방이 나가도록 설정</li> <li>사용자가 채팅을 나가면 해당 사용자가 나갔다는 문자열을 서버로 전송</li> <li>전송 시 문자열의 앞에 "EC:" (ExitChatting)을 붙여서 전송</li> <li>프로필 변경을 누를 시 닉네임 설정 등의 행위를 할 수 있음</li> </ul>

요구사항 아이디	C21
요구사항 명	닉네임 설정
상세 요구사항 내용	<ul style="list-style-type: none"> <li>사용자는 닉네임을 설정할 수 있음</li> <li>닉네임은 채팅방에서도, 채팅 대기방에서도 설정이 가능</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>서버로 전송하는 정보가 닉네임 설정이라는 것을 명시해야 함</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>닉네임을 TextField로 입력 받고 "완료" 버튼을 클릭하면 getText()메소드로 읽어온 후 해당 문자열 앞에 "CN:" (ChangeNickName)을 붙이고 서버로 전송</li> <li>모든 사용자의 초기 닉네임은 equals(사용자 ID)로 설정</li> </ul>

요구사항 아이디	C23
요구사항 명	채팅방 초대장 인터페이스
상세 요구사항 내용	<ul style="list-style-type: none"> <li>채팅방 초대장 인터페이스 개발</li> </ul>
이슈사항	<ul style="list-style-type: none"> <li>초대를 받은 사용자는 채팅방에 비밀번호가 설정되어 있어도 자동 입장이 가능</li> </ul>
해결방안 (변경내역)	<ul style="list-style-type: none"> <li>초대한 사용자의 아이디 정보와 "초대 승낙", "초대 거절" 버튼으로 이루어짐</li> <li>초대를 받은 사용자가 비밀번호가 설정된 방에 입장 시 "JCI:"를 붙이고 서버로 정보를 전송 -&gt; 일반 입장과 구분되는 메시지 타입 설정 (일반입장 - JC)</li> </ul>