

# 채팅 클라이언트 프로그램 개발 완료 보고서

작성일: 2024년 11월 19일

참여인원	곽 현 민		
프로젝트 소개			
프로젝트명	C#, Winform 으로 채팅 프로그램 만들기		
활동일시	2024.08.07 ~ 2024.11.20	장소	네이버시스템(주) 연구소
주요주제	C# Winform 기반 TCP/IP를 활용하여 채팅 프로그램 만들기		
개발목적	C#, Winform, DB 활용 부서 내 채팅 프로그램을 개발하며 소켓 통신 숙달		
개발환경	Windows 10 / Visual Studio 2022 community / Winform / SQLite		
구현기능	1. TCP/IP 소켓 통신 네트워크 구현		
	2. 로그인하기		
	3. 로그아웃하기		
	4. 친구목록 불러오기		
	5. 채팅목록 불러오기		
	6. 1대1 채팅하기		
	7. DB에 메시지 내용 저장하기		
	채팅 프로그램 개발		
	화면	분류	요구분석 내용
	로그인	기능	아이디, 비밀번호를 입력하면 로그인을 할 수 있도록 구현한다.
			아이디, 비밀번호를 잘못 입력한 경우 메시지를 출력하도록 구현한다.
			체크박스 체크시 로그인 정보를 저장하여 다음 프로그램 실행할 때 로그인 정보를 불러올 수 있도록 구현한다.
	친구목록	불러오기	친구목록 창이 열릴 때 친구목록을 불러오게 구현한다.
		기능	친구목록에서 항목을 더블클릭하면 해당 상대와의 채팅창이 열리게 구현한다.
			채팅목록버튼을 클릭하면 채팅목록이 나오게 구현한다.
로그아웃 버튼을 클릭하면 로그아웃이 되게 구현한다.			
채팅목록	불러오기	채팅목록이 열릴 때 채팅목록을 불러오게 구현한다.	
	기능	가장 최근 메시지가 보이고 최신순으로 정렬하게 구현한다.	
		채팅목록에 항목을 클릭하면 해당 채팅창이 열리게 구현한다.	
		친구목록버튼을 클릭하면 친구목록이 나오게 구현한다.	
채팅창	불러오기	최근 채팅내역 20건을 불러올 수 있도록 구현한다.	
	기능	메시지 입력창에 메시지를 입력 후 전송 버튼을 클릭하면 서버에 메시지가 전송되도록 구현한다.	

			엔터키를 통해 메시지 전송 버튼을 클릭할 수 있게 구현한다. 이전 메시지 버튼을 클릭하면 이전 메시지 내용을 채팅창에 띄울 수 있도록 구현한다.
	네트워크	연결	C++ 서버와 소켓 통신을 할 수 있게 구현한다. TCP/IP 통신을 통해 서버와 소켓 연결을 할 수 있게 구현한다. 서버와 연결이 끊기면 프로그램을 종료하도록 구현한다.

코드 설명	[클라이언트]
	클래스 및 변수:
	<LoginForm.cs>
	: 로그인 창
	LoginForm_Load()
	: 로그인폼이 열릴 때 실행되는 부분
	LoadConfig()
	: 서버 설정을 위해 config.json파일에서 서버 정보를 불러오는 부분
	ConnectToServer()
	: 서버와 소켓 연결 부분
LoadLoginInfo()	
: 자동 로그인을 위해 이전 로그인 정보 불러오는 부분	
BtnLogin_Click()	
: 로그인 버튼을 클릭하면 비밀번호를 해싱한 다음 로그인 시도 요청을 서버에 전송하는 부분	
Begin_Recv()	
: 로그인 성공여부 응답을 서버로부터 받기 위한 수신 부분	
SaveLoginInfo()	
: 체크박스 체크시 로그인 정보를 config.json파일에 저장하는 부분	
SaveLastLoggedInUser()	
: 최근 로그인 아이디 저장하는 부분	
GetLastLoggedInUser()	
: 가장 최근에 로그인한 아이디 불러오는 부분	
CheckAndDeleteDatabaseIfUserChanged()	
: 현재 로그인 하는 아이디와 가장 최근에 로그인했던 아이디 비교하는 부분	
DeleteChartData()	
: DB 테이블 모든 데이터 삭제	
BtnTogglePassword_Click()	
: 비밀번호를 특수문자로 가리거나 가리기를 해제하는 부분	
Focusing_Textbox()	
: 폼이 열릴 때 텍스트박스에 포커싱을 하는 부분	
LoginForm_FormClosed()	

: 폼을 닫는 이벤트 부분  
 ClearUserListJson()  
 : userList.json을 초기화 하는 부분  
  
**<MainForm.cs>**  
 : 친구목록, 채팅목록 창  
  
 StartReceiveThread()  
 : 수신 스레드를 생성하고 활성화 하는 부분  
 StartSendThread()  
 : 전송 스레드를 생성하고 활성화 하는 부분  
 LoadFriendList()  
 : 친구목록을 불러오는 부분  
 LbMyself\_DoubleClick()  
 : 본인 채팅방 생성 요청 부분  
 LbFriends\_DoubleClick()  
 : 다른 사용자와 채팅방 생성 요청 부분  
 SendMessagesFromQueue()  
 : 메시지 전송을 위해 대기하는 부분  
 QueueMessage()  
 : 메시지가 큐에 추가되면 신호 전송하는 부분  
 SendMessage()  
 : 메시지 전송하는 부분  
 SendMessageToServer()  
 : 메시지를 서버에 전송하는 부분  
 Begin\_rcv()  
 : 데이터를 서버로부터 수신하는 부분  
 ReceiveLargeMessage()  
 : 대용량 데이터를 처리하는 부분  
 ProcessPacket()  
 : 수신한 패킷을 패킷코드에 따라 처리하는 부분  
 OpenChatRoom()  
 : 채팅창을 여는 부분  
 SaveChatRoom()  
 : 채팅방 데이터를 roomtbl에 저장하는 부분  
 GetChatRoomId()  
 : roomtbl에서 채팅방 ID 찾는 부분  
 SendMessageToRoom()  
 : 수신한 메시지를 채팅방에 전달하는 부분  
 RemoveChatForm()  
 : 채팅방 종료시 디렉토리에서 제거하는 부분

	<p>SavePreviousMessageToData() : 이전 채팅 내용을 저장하는 부분</p> <p>SaveMessageToData() : 수신한 메시지를 저장하는 부분</p> <p>SaveRoomToData() : 수신메시지에서 채팅방 정보를 roomtbl에 저장하는 부분</p> <p>RoomExists() : roomtbl에 해당 room_id의 채팅방 정보가 이미 존재하는지 확인하는 부분</p> <p>PbFriends_Click() : 친구목록을 띄우는 클릭 이벤트 부분</p> <p>PbChats_Click() : 채팅목록을 띄우는 클릭 이벤트 부분</p> <p>LvChatList_ColumnWidthChanging() : 채팅목록 리스트뷰의 크기를 고정하는 부분</p> <p>GetOtherUserIdFromRoomId() : 상대방 아이디로 채팅방 아이디를 가져오는 부분</p> <p>GetFriendNameFromUserList() : 상대방 아이디로 상대방에 이름,닉네임을 가져오는 부분</p> <p>LoadChatList() : 채팅목록 불러와서 띄우는 부분</p> <p>UpdateChatList() : 채팅목록 업데이트하는 부분</p> <p>lvChatList_MouseClick() : 채팅목록에서 항목을 클릭하면 해당 채팅창이 열리는 부분</p> <p>BtnLogout_Click() : 로그아웃 버튼 클릭 이벤트 부분</p> <p>ShowConnectionClosedMessage() : 소켓 연결이 끊기면 메시지를 호출하는 부분</p> <p>MainForm_FormClosed : 메인폼 종료 이벤트 부분</p> <p><b>&lt;ChatForm.cs&gt;</b> : 채팅창</p> <p>LoadFriendInfo() : 채팅 상대방의 정보를 userList.json에서 가져오는 부분</p> <p>LoadEvent() : 채팅창이 처음 생길 때 필요한 이벤트들을 호출하는 부분</p> <p>LoadMessage() : 채팅방을 처음 열 때 이전 메시지를 불러오는 부분</p> <p>AddMessage()</p>
--	---

: 채팅창에 채팅메시지 내용을 띄우는 부분  
 BtnSend\_Click()  
 : 메시지 전송 버튼 이벤트 부분  
 BtnBeforeChat\_Click()  
 : 이전 메시지 불러오기 버튼 이벤트 부분  
 LoadPreviousMessages()  
 : 이전 메시지를 messagetbl에서 불러오는 부분  
 AddPreviousMessage()  
 : 불러온 이전 메시지를 채팅창에 띄우는 부분  
 AddLastMessagePanel()  
 : 더 이상 불러올 메시지가 없을 경우 마지막 메시지임을 띄우는 부분  
 AddDateBoundaryPanel()  
 : 메시지 수신시 날짜 경계선 추가 부분  
 AddBeforeDateBoundaryPanel()  
 : 이전 메시지 불러올 때 날짜 경계선 추가 부분  
 UpdateReadStatus()  
 : 읽은 메시지 상태 변경 부분  
 PassMessage()  
 : 수신 받은 메시지 채팅창에 가져오는 부분  
 tbChat\_TextChanged()  
 : 입력한 문자 수 채팅창에 출력하는 이벤트 부분  
 ChatForm\_FormClosed()  
 : 채팅창 닫기 이벤트 부분

#### <Packet.cs>

: 패킷구조체 클래스

packet\_data()

: 패킷구조를 만드는 부분

#### <PacketCodes.cs>

: 패킷 코드를 저장한 클래스

#### <PasswordHashing.cs>

: 비밀번호 해싱을 위한 클래스

HashPassword()

: 비밀번호를 해싱하는 부분

#### <UserList.cs>

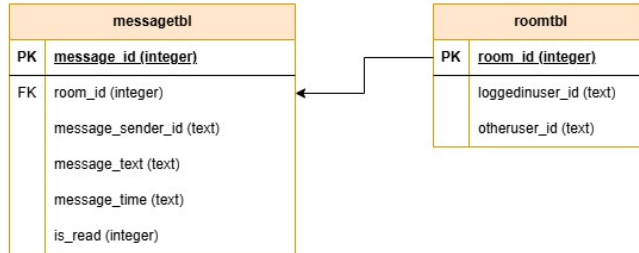
: userList.json파일을 활용하는 클래스

	<p>SaveUserList() : userList.json에 사용자 정보를 저장하는 부분</p> <p>LoadUserList() : userList.json에서 사용자 정보를 불러오는 부분</p> <p>Profile.cs: 채팅창 프로필 사용자 컨트롤</p> <p>Bubble.cs: 채팅 말풍선 사용자 컨트롤</p>
데이터베이스	<p>[SQLite]</p> <p>1) roomtbl</p> <ul style="list-style-type: none"> <li>• room_id (integer) : 채팅방 고유번호(PK)</li> <li>• loggedinuser_id (text) : 현재 사용자 ID</li> <li>• otheruser_id (text) : 다른사용자 ID</li> </ul> <p>2) messagetbl</p> <ul style="list-style-type: none"> <li>• <b>message_id (integer) : 메시지고유번호(PK)</b></li> <li>• room_id (integer) : 채팅방 고유번호[FK, references 'roomtbl(room_id)']</li> <li>• message_sender_id (text) : 발신자 ID</li> <li>• message_text (text) : 메시지 내용</li> <li>• message_time (text) : 메시지 전송 시간</li> <li>• is_read (integer) : 메시지 읽음 상태</li> </ul> <p>[JSON]</p> <p>1) config.json</p> <ul style="list-style-type: none"> <li>• IPAddress : 서버 IP 정보</li> <li>• Port : 서버 Port 정보</li> <li>• UserId : 저장한 로그인 ID 정보</li> <li>• UserPassword : 저장한 로그인 비밀번호 정보</li> <li>• last_logged_in_user : 최근 로그인한 아이디 정보</li> </ul> <p>2) userList.json</p> <ul style="list-style-type: none"> <li>• user_id : 사용자 ID 정보</li> <li>• user_name : 사용자 이름 정보</li> <li>• user_nickname : 사용자 별칭 정보</li> </ul>

## 채팅 클라이언트 프로그램 데이터

### 1. 채팅 메시지 데이터베이스 (SQLite)

chat\_Database.db



### 2. 초기설정을 위한 정보 (json)

config.json

config.json
IpAddress
Port
UserId
UserPassword
last_logged_in_user

### 3. 채팅 프로그램 사용자 정보 (json)

userList.json

userList.json
user_id
user_name
user_nickname

## [프로그램 UI]


### 1. 로그인 화면

로그인

아이디

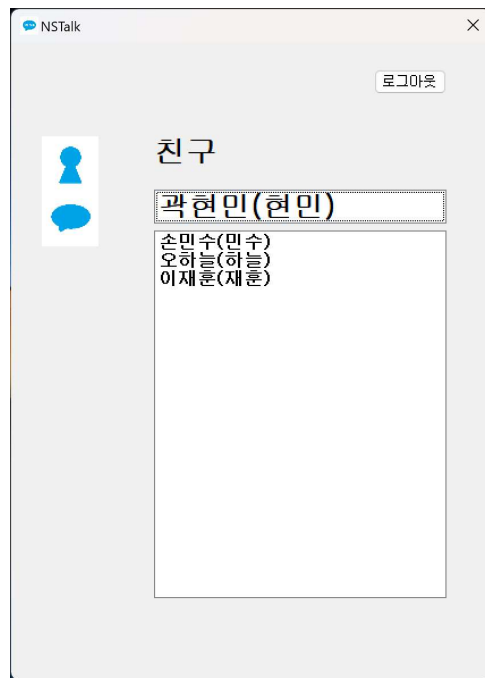
비밀번호

☐ 로그인정보 저장

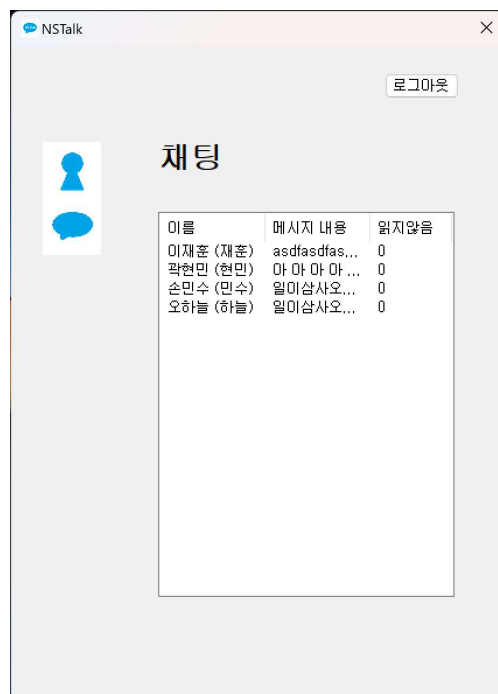
 네이버시스템

사용자  
인터페이스

## 2. 친구 목록 화면



## 3. 채팅 목록 화면





#### 4. 채팅 화면

