

개발 완료 보고서

제출일 : 2025-03-05

참여인원	김도엽		
프로젝트 명	[LMS5 22/30] Project09 C# WPF 프로젝트, 채팅 프로그램		
활동일시	2025.02.24 ~ 2025.03.05	장소	드론융합실
주요주제	C# WPF에 비동기 서버 클라이언트 작업, SQLite 데이터베이스 연동, 여러 클라이언트 간의 통신		
개발목적	서버를 통해 클라이언트끼리 통신할 수 있는 채팅 프로그램 개발		
개발환경	Windows 11, Microsoft Visual Studio C#, DBeaver, SQLite		
구현기능	<ul style="list-style-type: none">- 로그인 / 회원가입, DB 연동- 접속한 클라이언트들과 채팅- 접속한 클라이언트 리스트 확인- 접속한 유저 중 선택해서 채팅방 개설 후 채팅- 서버에서 유저, 채팅방 등 확인		
코드설명	<p><서버></p> <ul style="list-style-type: none">- openServerButton_Click : 서버 오픈- closeServerButton_Click : 서버 종료- AcceptClient : 서버 소켓에 클라이언트 받아들일 준비- ClientAccepted : 클라이언트가 연결되면 준비된 소켓이랑 연결		

	<ul style="list-style-type: none"> - ReceiveInfo : 클라이언트에서 데이터 받아옴 - DataReceived : 받아온 각종 데이터를 조건에 따라 역직렬화 해서 지역변수에 담고 서버 변수에 저장, 클라이언트에 데이터 다시 뿌림 - SendOnlineUserList : 유저가 서버에 접속할 때 정보 받아서 모든 클라이언트에 전송, 클라이언트는 실시간으로 접속 중 유저 갱신 - HandlingMessage : 클라이언트에서 온 데이터가 메시지일 경우, 조건에 따라 메시지를 클라이언트에 뿌리는 메소드 - CreateChatRoom : 클라이언트에서 채팅방 생성한다는 요청 받을 경우 서버에 채팅방 정보 저장하고 클라이언트에 되돌려줌 - ChatRoomList_MouseDoubleClick : 서버에 출력 중인 채팅방 목록을 클릭하면 해당하는 채팅방에 있는 유저 확인 가능 - AddLog : 로그창에 비동기로 텍스트를 추가 <p><클라이언트></p> <p>[MainWindow]</p> <ul style="list-style-type: none"> - loginButton_Click : 입력된 텍스트와 DB 정보를 대조해 로그인 시도 - joinButton_Click : 회원가입 페이지로 이동
--	--

	<p>[newAccount]</p> <ul style="list-style-type: none"> - new_joinButton_Click : 입력한 정보 토대로 회원가입, DB에 저장 - new_cancelButton_Click : 회원가입 창 닫음 - new_idDupCheck_Click : DB 정보 토대로 이미 있는 아이디인지 중복검사 - new_contactBox_PreviewTextInput : 연락처 입력칸에 숫자만 입력되게, 길이 제한 - new_idBox_TextChanged : ID 입력칸 텍스트가 바뀌면 중복검사 다시 하도록 - new_idBox_TextChanged : 연락처 입력칸 텍스트가 바뀌면 중복검사 다시 하도록 <p>[ChatRoom]</p> <ul style="list-style-type: none"> - chatWindow_Loaded : 창이 로드되면 서버와 연결 - ServerConnected : 서버와 연결되면 접속한 유저 정보 보내기 - ReceiveControl : 서버로 제어 요청 넘기고, 데이터 받아오기 - ControlReceived : 서버에서 받은 데이터에 따라 클라이언트가 대화, 온라인 유저 목록 등을 관리하는 메소드 - SendChatInfo : 보낼 메시지 정보를 서버로 보내는 메소드 - sendButton_Click : 버튼 클릭하면 입력칸에 있는 텍스트 서버에 보냄
--	---

	<ul style="list-style-type: none"> - sendMessage : 메시지 서버로 보낸 후, 입력칸 비우기 - chatBox_PreviewKeyDown : 엔터키 눌러도 입력칸에 있는 채팅 보내지도록 구현 - createChatRoom_Click : 접속 중인 유저를 초대해서 채팅방을 만들 수 있는 버튼 이벤트 - LogoutButton_Click : 클릭하면 접속 종료하고 로그인 페이지로 돌아가는 버튼 이벤트 <p>[CreateRoom]</p> <ul style="list-style-type: none"> - onlineUserList_create_PreviewMouseDoubleClick : 온라인 유저를 선택해서 초대할 유저를 추가할 수 있는 ListView 더블 클릭 이벤트 - CreateRoom_Button_Click : 설정한 정보 토대로 채팅방 개설하는 버튼 이벤트 - Cancel_Button_Click : 채팅방 개설창 닫기 - chatTitle_PreviewTextInput : 채팅방 제목 길이 수 제한하는 이벤트
후기	<p>더 많은 기능을 넣어보고 싶었는데, 이번에도 사소한 디테일에 집착하다가 시간을 많이 소비하고 말았다.</p> <p>서버 소켓 하나에 클라이언트 소켓 여러개를 받아서 하는 부분은 의외로 어렵지 않게 구현할 수 있었다.</p> <p>하지만 Dispatcher.Invoke를 활용한 비동기 작업, 데이터 관리를 lock으로 통제해주는 부분, SendAsync로 데이터 주고 받는 구조 이해에 대한 어려움 등 고난이 많았다.</p> <p>많은 자료를 탐방한 뒤 채팅을 전체 클라이언트에 뿌리는 것은 일찌감치 구현했는데, 접속중인 유저 목록을 띄우는 것이 쉽지 않았다.</p>

	<p>왜냐하면 예전에 C++로 만들었을 때 문자열 파싱을 이용했던 것 토대로 C#에서도 이 방법을 채용했는데, 이러면 접속중인 유저를 볼 때 이후에 들어온 유저가 이미 들어온 유저를 유저 목록에서 볼 수 없는 문제 때문이었다.</p> <p>이걸 해결하기 위해선 유저 목록을 관리하는 리스트를 서버에서 만들고, 클라이언트에 보내주는 방법이 좋았는데 문자열 파싱을 리스트에 하면 오류가 발생해서, 로직을 갈아엎어야 했다.</p> <p>다만 클래스 자체를 넘기면서 이 안에 구별할 수 있는 변수 이름을 편하게 지정하고 지정한 변수에 따라 조건문을 통과하게 해두니 오히려 다른 기능을 추가하긴 편했다.</p> <p>이번에 프로젝트를 하면서 통신이나 클래스, 함수 등 많은 기능을 숙달하게 된 것 같다. 또 다른 프로젝트를 하게 된다면 더 수월하게 할 수 있을 것 같다.</p>
--	--

요구사항 분석서

유형	요구분석 내용	
	분류	세부내용
환경	개발환경	Windows 11 / Microsoft Visual Studio 2022 / DBeaver / SQLite
채팅 계정	회원가입	ID, 비밀번호, 이름 등을 입력 받아서 신규 계정 생성 (중복 검사 거침)
	로그인	생성된 계정을 이용해 채팅 서버에 입장 가능
	회원 탈퇴	로그인 한 상태에서, 회원 탈퇴 가능
서버 통신	1:1 채팅	접속한 유저 중, 선택한 유저와 1:1 채팅 개설해서 대화 가능
	단체 채팅	접속한 유저 중 선택한 유저들과 단체 채팅방 개설해서 대화 가능
	쪽지함	접속 유무 상관없이, 존재하는 계정이라면 쪽지 주고받기 가능
DB	데이터	회원가입한 정보들 저장
		각 계정별로, 쪽지함 데이터 저장
		회원탈퇴시 해당 계정 정보 DB에서 삭제
		쪽지함 데이터 삭제 가능

일정표							
유형	요구분석 내용		2월				
	분류	세부내용	24	25	26	27	28
계정	로그인	로그인 UI 구현, DB에 들어있는 정보 토대로 계정에 접근					
	회원가입	회원가입 UI 구현, DB에 데이터 저장					
서버 통신	채팅	접속중인 유저 선택, 1:1 채팅 개설해서 대화 가능					
		접속중인 유저들 선택, 단체 채팅방 개설해서 대화 가능					
	쪽지	접속 유무 상관없이, 존재하는 계정에 쪽지 보내기					
DB	데이터	회원가입한 정보들 저장					
		각 계정별로, 쪽지함 데이터 저장					
		회원탈퇴시 해당 계정 정보 DB에서 삭제					
		쪽지함 데이터 삭제 가능					