

# 개발 완료 보고서

제출일 : 24.05.20

팀명	8팀	참여자	김성근, 김재신, 이준호
개발 프로젝트 소개			
프로젝트명	채팅 프로그램		
활동일시	24.05.20 ~ 24.05.26	장소	광주인력개발원 공학1관 2층 드론융합실
주제	DB와 소켓 서버, 클라이언트를 기반으로 멀티쓰레드를 활용한 채팅 프로그램		
개발목적	C++과 DB, TCP/IP의 소켓(서버, 클라이언트)과(와) 쓰레드를 활용하여 회원가입과 로그인 기능이 있는 채팅 프로그램을 제작한다. 쓰레드 기능을 사용하면서 쓰레드의 이해와 학습을 중점으로 둔다.		
프로젝트 세부 내용	<ul style="list-style-type: none"><li>- 순서도, ERD 작성 (하단 첨부)</li><li>- 요구사항 분석서 작성 (하단 첨부)</li><li>- 일정표 작성 (하단 첨부)</li><li>- 데이터 베이스 제작</li><li>- TCP/IP를 사용하여 서버 제작</li><li>- TCP/IP를 사용하여 클라이언트 제작</li><li>- 데이터 베이스 연동</li><li>- 완료 보고서 작성</li></ul>		
구현 예정 기능	<ul style="list-style-type: none"><li>※ 필수 기능<ul style="list-style-type: none"><li>- 로그인, 회원 가입 기능</li><li>- 친구 추가와 접속 상태 확인 기능</li><li>- 친구와 1:1 채팅 기능</li></ul></li><li>※ 추가 기능<ul style="list-style-type: none"><li>- 단체 채팅 기능</li><li>- 랜덤 채팅 기능</li></ul></li></ul>		
담당	<ul style="list-style-type: none"><li>- 김성근 : 순서도 및 코드 작성</li><li>- 김재신 : 문서 작성 및 코드 작성</li><li>- 이준호 : DB와 서버 및 코드 작성</li></ul>		
개발환경	Ubuntu 22.04 / Visual Studio Code / MariaDB		

# 요구 사항 분석서

'24.05.20

요구사항  
분석서

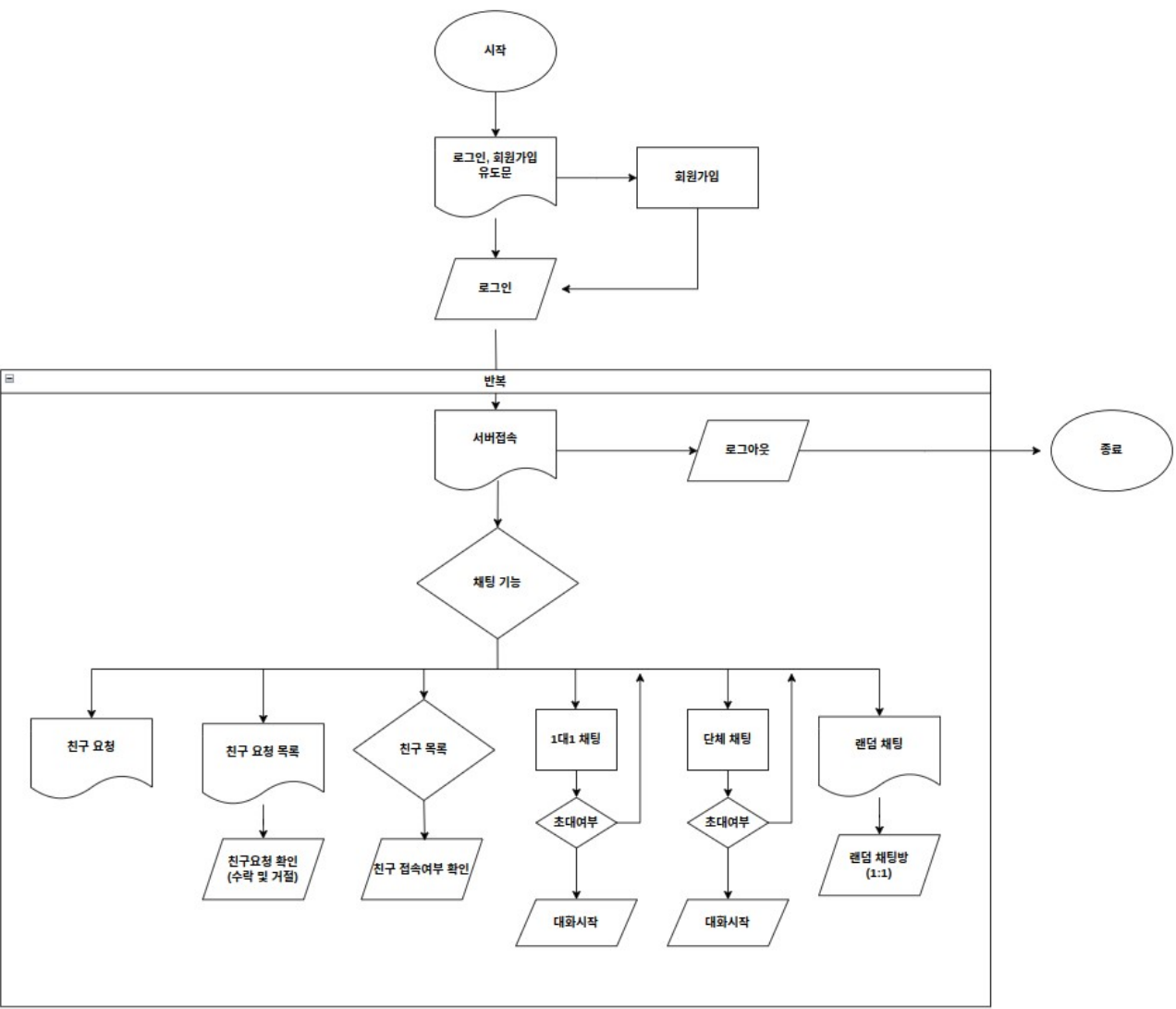
유형	분류	설명	
		세부 분류/기능	세부 내용
필수 요구 기능	회원 가입 및 로그인	회원 등록	사용자는 ID와 비밀번호, 닉네임으로 회원 등록 할 수 있어야 한다
		로그인	등록한 ID와 비밀번호로 인증해야 하고 5번 틀릴 시 인증 방식에 고유 번호가 필요 해야 한다.
		회원 정보 관리	사용자의 고유 번호, ID, 비밀번호, 닉네임을 개별적으로 중복되지 않게 해야 한다.
	친구 신청 및 접속 여부	친구 맺기	사용자는 고유 번호로 상대방에게 친구 신청을 보낼 수 있어야 한다.
		친구 수락/거절	사용자는 신청을 수락하거나 거절 할 수 있어야 한다.
		접속 여부	사용자는 친구창에서 접속 여부를 확인 할 수 있어야 한다.
	친구와의 채팅	친구와 채팅	사용자는 친구창에서 1:1 채팅을 할 사람을 선택 할 수 있어야 한다.
		이름 표시	대화방에서의 대화에는 채팅자의 닉네임이 표시되어야 한다.
	시스템 관리	데이터 베이스 연동	시스템은 MariaDB의 데이터 베이스와 연동되어 회원의 정보와 친구 목록의 관리를 해야한다.
	단체 채팅 기능	인원 선택	사용자는 친구창에서 다수를 선택할 수 있어야 한다.
		초대 결정	사용자는 채팅 초대 수락 여부를 결정할 수 있어야 한다.
		이름 표시	각각의 사용자의 채팅 앞에 닉네임이 표시되어야 한다.
		안내 메시지	접속자가 접속 종료 시 안내 메시지로 알려줘야 한다.
	랜덤 채팅 기능	랜덤 연결	사용자는 랜덤하게 다른 고유번호의 사람과 채팅이 연결되어야 한다.
		다음 대화 상대 찾기	사용자는 얼마든 지 채팅이 끝나면 다른 상대를 찾을 수 있어야 한다.
		대화 종료 기능	사용자는 대화가 진행중이어도 끝낼 수 있어야 한다.

일정표

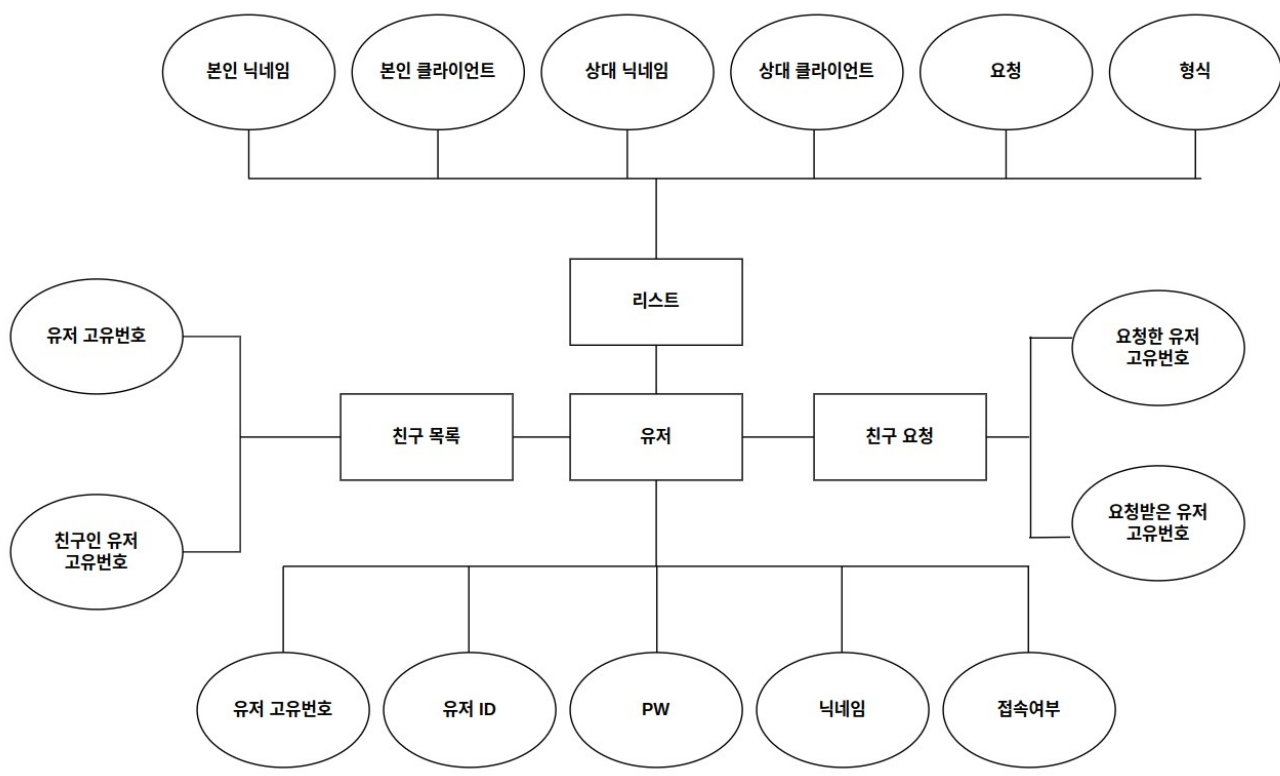
프로젝트 일정표

항목		05 / 20		05 / 21		05 / 22		05 / 23		05 / 24		05 / 25		05 / 26	
개발 계획서 및 일정표 작성															
구현	DB 설계														
	기능 구현														
	검토 및 수정														
제출	완료보고서 및 문서 작성														

순서도



개념적 설계



ERD 순서도



후기	김성근	이번 프로젝트는 tcp/ip를 통해서 채팅 프로그램을 만들어봤습니다. 배웠던 내용으로 구현이 가능할 것 같았지만 생각보다 로직을 복잡하게 하고 많이 어려웠습니다. DB는 친구목록이나 로그인 여부 등 저장이나 관계성이 복잡해서 구성하는데 시간이 걸렸습니다. 채팅 로직은 아직 쓰레드의 이해가 부족해 생각한대로 구현하지 못하고 팀원에게 많이 도움을 받았습니다. 좀 더 공부해보고 이해해보도록 하겠습니다.
	김재신	기존의 TCP/IP와 DB 연동에 더해 멀티쓰레드, 큐, 벡터 등을 사용해 가며 친구 기능을 포함한 멀티 채팅 프로그램을 만드는 동안 각각의 기능을 별개로 구현만 해봤지 하나로 합쳐서 작동 시키는게 많이 힘들었습니다. 쓰레드가 많이 생소했기에 그 부분은 아직 많이 부족 한 것 같습니다.
	이준호	8번째 팀 프로젝트인 채팅 프로그램을 마무리하였습니다. 7번째 팀 프로젝트였던 도서관리 프로그램으로 인해 TCP/IP에 어느 정도 익숙해져 있어, 통신에는 큰 어려움이 없었습니다. 하지만 채팅으로 인한 멀티 쓰레드 구현이 어려웠습니다. 계속된 고뇌 끝에 로그인하자마자 유저들의 클라이언트 번호를 구조체에 저장하고, 유저가 다른 유저에게 채팅을 신청하게 되면 DB에 데이터를 입력하여 저장하게끔 하는 로직을 구현하였습니다. 1 대 1 채팅이 완료되니, 단체 채팅과 랜덤 채팅은 약간의 로직 변경만으로 로직을 구현할 수 있었습니다. 짧은 기간으로 인해 팀원인 성근형님과 기숙사에서 밤을 새가며 프로젝트를 진행하였습니다. 시간이 지나면 좋은 추억으로 기억될 것 같습니다. 또한 이번 프로젝트로 인해 통신에 대해 저번 프로젝트보다 더 굵직하게 배울 수 있었고, 멀티 쓰레드 구현으로 인해 쓰레드에 자신감이 생겼습니다.

Chat_class	class DB DB 연결 클래스	string m_id; string m_pw;	DB ID, 비밀번호
		sql::Connection *m_connection;	DB 연결
		void account(const string &id, const string &pw)	ID, 비밀번호 입력
		void connect()	DB 연결
	class chat : public DB 채팅과 관련된 클래스	string LOGIN(std::string id, std::string pw, int U_id = 0)	채팅 로그인
		void login_connect(std::string id)	로그인시 연결 여부 변경
		void logout_connect(std::string id)	로그아웃 시 연결 여부 변경
		void logout_chatlist(std::string nickname)	로그아웃 시 채팅 요구 상태 변경
		void user_info(std::string id, std::string pw, std::string nickname)	회원 가입
		string check_ID(std::string ID)	아이디 중복 검사
		string check_nickname(std::string nickname)	닉네임 중복 검사
		string get_uid(string id)	유저 아이디를 매개 변수로, 고유 번호 받기
		string request_friend(string u_id, string f_id)	친구 요청 테이블에 친구 요청 입력

		string accept_friend(string u_id, string f_id)	요청 수락
		string reject_friend(string u_id, string f_id)	요청 거절
		string check_connect(string f_id)	친구 로그인 상태 확인
		int show_uid()	고유번호 최대값 얻기
		string get_nickname(string id)	유저 아이디를 매개 변수로, 고유번호 얻기
		void insert_chat(string who_nickname, int who_clnt, string whom_nickname, int whom_clnt, string type)	채팅 목록에 신청 사항 추가
		int check_chatcount(string nickname, string type)	1대1 채팅 목록 확인
		int take_whocln(string nickname, string whom, string type)	채팅을 신청한 유저의 고유 번호 얻기
		int take_whomclnt(string nickname, string who)	채팅 신청 받은 유저의 고유번호 얻기
		vector<int> takeg_whomclnt(string who_nick, int who_clnt)	단체 채팅 신청 받은 유저들 고유번호 얻기
		void check_onechat(string who_nick, string whom_nick, int whom_clnt, string type)	채팅 수락 및 거절 시 채팅 신청 상태 변경
Chat_serv	struct Client 클라이언트 입력 구조체	int cl_sock;	클라이언트 소켓
		string cl_nick;	클라이언트 닉네임
	void *handle_cln t(void *arg) 클라이언트를 연결하는 서버 핸들러	if (choice == "q"    choice == "Q")	목록에서 'Q' 입력 시 종료
		else if (choice == "1"    choice == "로그인")	로그인
		else if (choice == "2"    choice == "회원가입")	회원가입
		if (count < 6) / else	5번 이상 로그인 오류 시 로그인, 비밀번호, 고유번호 입력하여 로그인
		if (logreturn == "a") / else	로그인 성공 및 실패
		if (num == "1")	친구 요청 수신
		else if (num == "2")	친구 요청란 확인 후, 친구 요청 수락 및 거절
		else if (num == "3")	친구의 접속여부 확인하기
		else if (num == "4")	1 대 1 채팅 요청하기

		else if (num == "5")	1 대 1 채팅 확인 및 수락, 거절
		else if (num == "6")	단체 채팅 요청
		else if (num == "7")	단체 채팅 확인 및 수락, 거절 (신청 받은 사람들 들어가는 곳)
		else if (num == "8")	랜덤 채팅방
		void send_msg(char *msg, int len, string nick, string want)	1 대 1 채팅 신청 받은 사람 (정보 DB에서 가져오기)
		void send_groupmsg(char *msg, int len, string nick, string want)	그룹 채팅 신청 받은 사람 (정보 DB에서 가져오기)
		void send_msg1(char *msg, int len, int clnt_sock, int cl_sock)	그1대 1 채팅 신청한 사람
		void send_groupmsg1(char *msg, int len, int *group_cl, int p)	그룹 채팅 신청한 사람(구조체에서 가져오기)
		void error_handling(std::string msg)	read함수에서 오류 발생 시 클라이언트에게 에러 메시지를 보여줌
		int requestlist(string u_id, int clnt_sock)	친구 요청 리스트
		int friendlist(string u_id, int clnt_sock)	친구 리스트
		void Client_list(string id, int clnt_sock)	구조체 입력시키는 핸들러
		int clnt(string nick)	구조체에서 클라이언트 번호 가져오는 함수
		void chatlist(string nick, int clnt_sock, string type)	채팅 리스트
		void rand_msg(const char *msg, int len) // send to all	랜덤 채팅
<b>Chat_clnt</b>	void* handle_serv(void *arg, pthread_t snd_thread, pthread_t rcv_thread, void *thread_return)		

	if (choice == "q"    choice == "Q")	채팅 프로그램 종료
	else if (choice == "1"    choice == "로그인")	로그인, 5번 실패시 고유번호와 함께
	if (num == "1")	친구요청 보내기
	else if (num == "2")	친구 요청란 확인 후, 친구 요청 수락 및 거절.
	else if (num == "3")	친구 접속여부 확인 가능
	else if (num == "4")	1:1 채팅 친구에게 요청 , 닉네임으로 구분
	else if (num == "5")	1:1 채팅 확인 및 수락 거절
	else if (num == "6")	단체 채팅 요청하기
	else if (num == "7")	단체 채팅 확인 및 수락, 거절



	else if (num == "8")	랜덤 채팅방 입장
	else if (choice == "2"    choice == "회원가입")	회원 가입, id, 닉네임 중복 체크
	void error_handling(std::string msg);	read함수에서 오류 발생 시 클라이언트에게 에러 메세지를 보여줍니다.
	void *send_msg(void *arg)	서버에 채팅방에 보낼 채팅 메세지를 보냅니다.
	void *recv_msg(void *arg);	서버에서 상대방이 보낸 채팅을 받고 출력합니다.