

개발 완료 보고서

제출일 : 2024년 5월 18일

팀명	참여인원		
뜨거운 2팀	이동준, 김민아, 박선후, 이준호		
개발프로젝트 소개			
프로젝트 명	“도서 관리 프로그램”		
활동일시	24. 05. 13. ~ 24. 05. 18	장소	광주인력개발원 공학1관 2층 드론융합실
주요주제	c++ 언어의 제어문, 반복문, 난수, 선택문, 배열, 포인터, 구조체, 함수, 동적할당, DB와 다중접속 서버를 연동하여 광주광역시 광산구 장덕도서관의 회원/도서별 조회, 대여, 반납을 관리하는 프로그램 개발		
개발목적	C++언어의 다양한 개념과 기능을 활용 *제어문, 반복문, 선택문, 배열, 함수, 포인터, 클래스, 동적할당하고 DB,다중 접속 서버를 활용하여 실전 프로그래밍 경험을 쌓고, 디버깅을 통한 코드 작성법을 익히고, 코드의 가독성과 유지보수성을 높이는 경험을 키운다.		
개발환경	실습실 / 개인pc / Ubuntu 16.04 Linux os / MariaDB / Visual Studio Code		
구현기능	<div>- 로그인 ID와 PW를 입력하여 회원정보와 일치하면 로그인 성공</div>		
	<div>- 회원가입 회원가입을 통한 DB 테이블 등록</div>		
	<div>- 회원등급관리 우수회원, 일반회원, 블랙회원으로 나뉨 1)우수회원 - 가입기간 6개월 이상 AND 10권이상 연체 없이 반납했을 시/책 대여기간 7일 2)일반회원 - 회원가입 시 조건 없이 / 책 대여기간 4일 3)블랙회원 - 연체 이력 3회 이상 OR 연체일자 14일 이상 / 책 반납 후 30일간 대여 불가</div>		
	<div>- 추천도서 목록 랜덤으로 추천도서 출력</div>		
	<div>- 도서 조회 제목, 작가, 장르 등 기본 정보 출력 1) 대여를 위한 도서 조회 - 로그인 필요 2) 책 목록을 확인하기 위한 도서 조회 - 로그인 미 필요</div>		
	<div>- 대여 및 반납 1) 로그인 필요 - ID, PW로 로그인 2) 도서 기본 정보 조회 후 현재 보유하고 있으면 대여 가능</div>		

	3) 현재 대여 중인 도서는 대여 불가능 4) 반납 시, 도서 명 OR 등록 번호로 반납 가능 5) 반납 완료 시 대여 상태 변경
--	--

사용파일	LIB_SER.cpp	서버
	LIB_CNT.cpp	클라이언트
사용헤더	iostream stdlib.h unistd.h string.h arpa/inet.h sys/socket.h netinet/in.h pthread.h mariadb/conncpp.hpp	
사용함수 (서버)	void *handle_clnt(void *arg);	클라이언트와 연결하는 함수
	void error_handling(const char *msg);	오류발생시 사용되는 함수
	void addUser(std::unique_ptr<sql::Connection> &conn, std::string name, std::string state, std::string ID, std::string PW, std::string phone, std::string times);	DB에 회원정보를 등록해 주는 함수
	int Userlogin(std::unique_ptr<sql::Connection> &conn, string user_id, string user_pw);	로그인
	void recomm_book(int clnt_sock, std::unique_ptr<sql::Connection> &conn);	추천도서 출력
	void showBook(std::unique_ptr<sql::Connection> &conn);	도서 검색
	void book_rent(std::unique_ptr<sql::Connection> &conn);	도서 대여
	void Userjoin(std::unique_ptr<sql::Connection> &conn, string id, string pw, string name, string address, string phonenumm, int clnt_sock);	회원가입
	void book_return(std::unique_ptr<sql::Connection> &conn, string user_id, int clnt_sock);	도서 반납
	void GradeUpdate(std::unique_ptr<sql::Connection> &conn, string user_id)	회원 등급 관리
사용함수	void *send_msg(void *arg);	서버와 연결해주는 함수
	void *recv_msg(void *arg);	서버에서 특정 메시지를 가져오는 함수
	void error_handling(const char *msg);	오류 발생시 실행되는 함수.

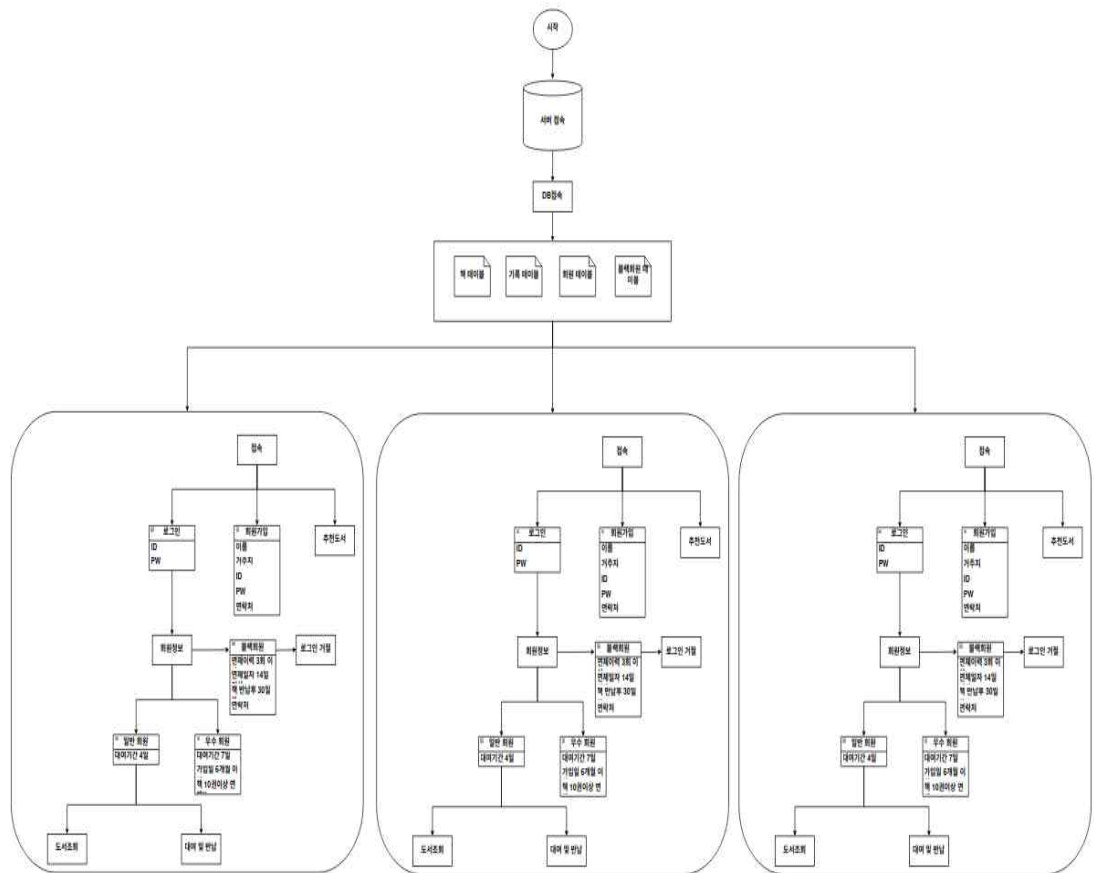
프로젝트 일정표

항 목		05/13		05/14		05/15		05/16		05/17		05/18	
개발계획서 및 일정표 작성													
요구분석서 작성													
순서도 작성													
DB제작	블랙 회원 테이블 제작												
	회원 정보 테이블 제작												
	대여 정보 테이블 제작												
	도서 정보 테이블 제작												
로직작성	회원가입												
	회원등급												
	추천도서												
	서버 설계												
	클라이언트 설계												
	대여 및 반납												
검토	중간점검												
	디버깅 및 오류수정												
제출	완료보고서 및 문서 작성												

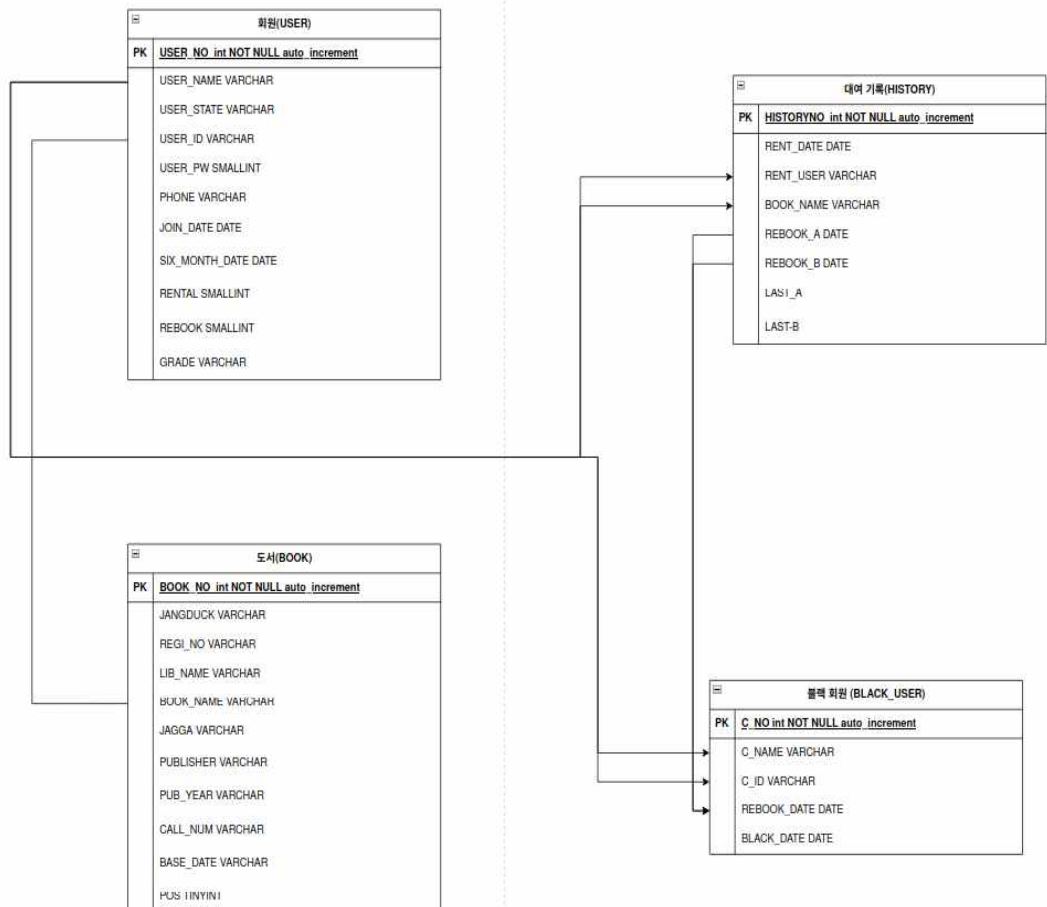
일정표

요구분석서	분류	세부 분류	요구 분석 내용
			세부 내용
	회원관리	로그인 및 회원가입	회원가입 시 회원정보를 등록할 TABLE 생성
			조건문과 쿼리문을 통한 회원정보 파일 일치 유무 확인
		회원 등급 관리	회원 등급 - 우수회원, 일반회원, 블랙회원
			우수회원 - 가입기간 6개월 이상과 10권 이상 연체 없이 반납, 책 대여기간 7일
			일반회원 - 책 대여기간 4일
			블랙회원 - 연체 이력 3회 이상 또는 연체 일자 14일 이상, 책 반납 후 대여 30일 불가
	도서관리	추천 도서 목록	랜덤함수를 통한 추천 도서 선정
		도서 조회	도서 기본 정보(제목, 작가, 장르, 출판사 등) 조회 기능
			대여가능 정보(대여 여부, 반납 예정일자) 조회 기능
		대여 및 반납	대여 서비스 이용 시, ID와 PW를 통한 로그인
			Bool type을 통한 도서의 현재 대여 상태 파악
			현재 대여중인 도서 대여 불가능
			반납 시, 도서명 또는 등록번호로 반납 가능
			반납완료 시, 대여 상태 변경

순서도



DB 다이어그램



코드설명	<pre> sql::Driver *driver = sql::mysql::get_driver_instance(); sql::SQLString url("jdbc:mysql://10.10.21.227:3306/LIB"); sql::Properties properties({"user", "LIB"}, {"password", "1"}); std::unique_ptr<sql::Connection> conn(driver->connect(url, properties)); </pre>	mariadb 와 vscode를 연동
	<pre> time_t now = time(0); tm *settime = localtime(&now); char times[20]; strftime(times, sizeof(times), "%Y/%m/%d %H:%M:%S", settime); addUser(conn, name, address, id, pw, phonenumber, times); </pre>	가입시기 및 유저 정보 등록
	<pre> if (res->getString(4) == user_id && res->getString(5) == user_pw) { string set; set = res->getString(2); const char *showname = set.c_str(); write(clnt_sock, showname, strlen(showname)); std::cout << set << " 님이 접속하셨습니다." << endl; } </pre>	유저 ID와 PW가 일치하면 로그인
	<pre> randnum = random() % 76509 + 1; randlist[count] = randnum; if (res->getInt(1) == randlist[count]){ string book; book = "도서명: " + res->getString(5) + "\n"; const char *recomm_book = book.c_str(); write(clnt_sock, recomm_book, strlen(recomm_book)); string author; author = "지은이: " + res->getString(6) + "\n\n"; const char *recomm_author = author.c_str(); write(clnt_sock, recomm_author, strlen(recomm_author)); } </pre>	난수 뽑은 후 책 번호와 비교해서 책이름, 작가 출력
	<pre> read(clnt_sock, msg, sizeof(msg)); cout << msg << endl; </pre>	

	<pre> book_name = msg; memset(msg, 0, sizeof(msg)); string save_book = book_name; int k = 0; while (res6->next()) { ++k; } string book_cnt = to_string(k); const char *book_count = book_cnt.c_str(); write(clnt_sock, book_count, strlen(book_count)); int h = 0; while (res7->next()) { cout << "책 번호: " << res7->getInt(1) << endl; book_no = res7->getString(1) + "\n"; const char *showno = book_no.c_str(); write(clnt_sock, showno, strlen(showno)); if (h == k) { cout << "안녕" << endl; msg, sizeof(msg)); // 책 번호 입력받음 cout << "받은 책 번호 " << msg << endl; msg[strcspn(msg, "\n")] = '\0'; string kia(msg); } } </pre>	<p>클라이언트에서 책 정보 받아와서 DB조회 후 클라이언트에게 보내준다</p>
	<pre> while ((str_len = read(clnt_sock, msg, sizeof(msg) - 2)) != 0) { std::cout << msg; if (!strcmp(msg, "1\n")) { int ID_read = read(clnt_sock, msg, sizeof(msg)); std::string user_id(msg); int PW_read = read(clnt_sock, msg, sizeof(msg)); std::string user_pw(msg); memset(msg, 0, BUF_SIZE); Userlogin(conn, user_id, user_pw, clnt_sock); int recomend = read(clnt_sock, msg, sizeof(msg)); } } </pre>	<p>함수 실행하는 선택문 1번 선택 시 로그인 하고 회원 메뉴 진입 (추천 도서, 도서 조회, 대여 및 반납)</p>

	<pre> if (!strcmp(msg, "1Wn")) recomm_book(conn, clnt_sock); else if (!strcmp(msg, "2Wn")) book_rent(conn, user_id, clnt_sock); </pre>	
	<pre> if (!strcmp(msg, "2Wn")) { cout << "2번 선택" << endl; for (int i = 0; i < 5; i++) { str_len = read(clnt_sock, msg, sizeof(msg)); msg[str_len] = '\0'; if (i == 0) id = string(msg); else if (i == 1) pw = string(msg); else if (i == 2) name = string(msg); else if (i == 3) address = string(msg); else if (i == 4) phonenum = string(msg); } } if (!strcmp(msg, "3Wn")) showBook(clnt_sock, conn); </pre>	<p>2번 선택 시 회원 가입,</p> <p>3번 선택 시 도서 검색</p>

이동준

TCP/IP는 이전 공부들과 달리 잘 이해가 되지 않아 프로젝트 초반에 감을 잡지 못했습니다. 모르는 상황에서 코드를 이리저리 바꿔가며 몸으로 학습한 결과 TCP의 READ와 WRITE를 조금 이해했지만 제가 올바르게 이해한 것인지 확신이 들지 않습니다. 프로젝트를 끝낼 때까지 문자열의 변환과 싸웠는데 여전히 이해가 되지 않는 부분이 있어 좀 더 공부할 계획입니다. 일반적인 코드와 달리 병렬적으로 사고하며 코드의 흐름 역시 생각하지 못한 방향으로 진행돼서 많은 고민 끝에 해결한 문제들도 다시 돌아봐야겠습니다.

어려웠던 프로젝트임에도 힘내서 함께 공부하고 모르는 부분을 서로 물어가며 같이 공부해 준 팀원들 덕에 기간 내에 프로젝트를 마칠 수 있었습니다.

김민아

7번째 팀 프로젝트 “도서 관리 프로그램”을 완료하였습니다.

이번에 회원 등급 관리 로직을 맡아서 진행했는데, DB에서 세 개의 테이블을 모두 활용해야 하는 부분이라서 시간이 많이 소요되었다.

쿼리문도 많이 작성하고 여러 가지 방법을 시도하면서 배운 것이 많은 프로젝트였다. 작성한 코드가 팀원과 겹치는 부분이 있어 로직을 작성하면서도 의견 교환이 필요한 부분이 있었다.

서버와 클라이언트를 연결하는 것보다 읽고 쓰는 것이 어려웠고 이 부분은 더 공부를 할 예정이다.

팀원들 모두 새벽까지 공부하며 프로젝트를 진행하여 기간 내에 끝낸 것 같다.

박선후

C로 된 서버, 클라이언트와 C++로 된 MariaDB를 연결하는 부분에서 어려움을 겪었으나, 팀원들의 집단 지성으로 빠르게 해결했다.

클라이언트와 서버에서는 개행문자가 정말 미워졌으나 없애기에는 너무 많은 불편함을 감수해야 하기 때문에 내가 좀 더 공부해야겠다.

이번 프로젝트를 진행하면서 TCP/IP에 대해서 많이 공부했다. 서버와 클라이언트의 관계, 실행 원리, 동기화 비동기의 차이점, 쓰레드와 뮤텍스에 대해서 많이 이해하게 되었다.

팀원들이 열정적으로 진행하여 의견 교환도 많이 하고, 그에 따른 로직을 짜면서 서로 발전하는 시간이었다.

이준호

7번째 팀 프로젝트 '도서 관리 프로그램'을 완료하였습니다.

TCP/IP 배움에 어려움이 있어, 지금까지 했던 프로젝트에 비해 시간도 촉박하고 계속된 오류를 맞보았던 프로젝트였습니다. 다행히 저번 "DBZA" 프로젝트에서 DB와 C++ 연동 로직을 짜는데 익숙해져 이번 프로젝트에서는 금방 로직을 구현할 수 있었습니다. 하지만 최대 복병인 서버와 클라이언트 통신에 어려움이 있었습니다. 그로 인해 책에 있는 로직을 하나하나 해석하고 실행해 본 뒤 쓰레드의 기능과 read, write 함수의 쓰임을 알게 되었습니다. 그 결과 랜덤 함수를 통해 랜덤 숫자 30개를 입력받고 30개의 추천 도서들을 서버와 클라이언트 간에 통신할 수 있었고, 반납과 대여 로직을 병합한 후 서버와 클라이언트 간에 통신까지 완료할 수 있었습니다.

	아직은 부족하지만 이번 프로젝트를 통해서 서버와 클라이언트의 통신을 조금이나마 이해하고 이용해 볼 수 있었습니다.
--	---

	또한, 열정이 가득한 팀원들 덕에 힘을 내며 프로젝트를 진행할 수 있었습니다. 2팀 팀원분들 고생하셨습니다.
--	--