

# CPS System

## (Company Publish Subscribe)

- 최종 보고서 -

제출일 : 2019년 7월 26일

팀명	Altiall Juniors
팀원	1. 정 진구 (현장 실습생)
	2. 박 혜은 (현장 실습생)
	3. 이 승원 (현장 실습생)

## **결과 보고서**

- 분야 : TCP 통신 윈도우 응용프로그램
- 프로젝트 명 : CPS(Company Publish Subscribe) system
- 프로젝트 팀 : Alltiall Juniors
- 프로젝트 참여 팀원 : 현장 실습생(정 진 구, 박 혜 은, 이 승 원)

**2019 년도 Alltiall Technologies 현장실습 프로젝트 결과물로  
이 보고서를 제출합니다.**

**프로젝트 공동 참여 : 현장 실습생 정 진 구 (인)**

**프로젝트 공동 참여 : 현장 실습생 박 혜 은 (인)**

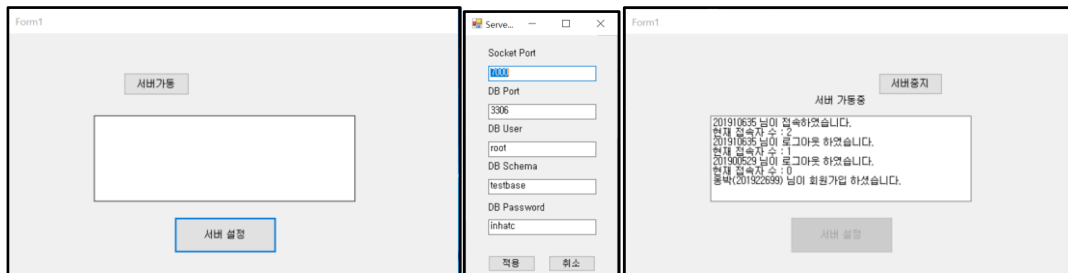
**프로젝트 공동 참여 : 현장 실습생 이 승 원 (인)**

**2019. 07. 26 (금)**

## - 프로젝트 요약문

프로젝트 팀 명	Altiall Juniors
프로젝트 명	CPS system
프로젝트 참여 팀원	현장 실습생 정 친구 현장 실습생 박 혜은 현장 실습생 이 승원

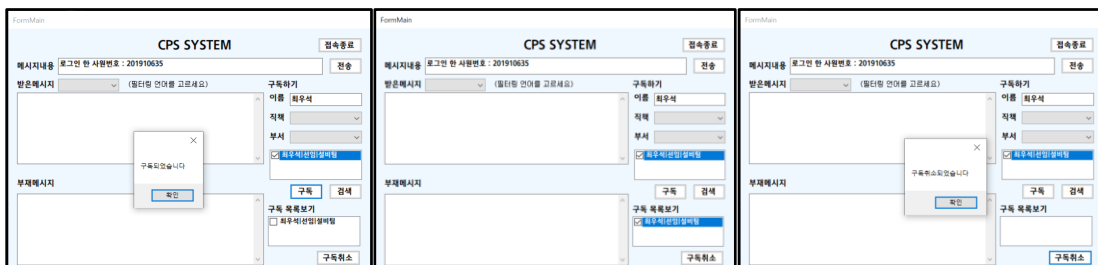
### 1. TCP 통신을 활용한 다중 클라이언트 서버 구현



### 2. 로그인, 회원가입 기능 구현



### 3. 클라이언트 검색 및 구독, 구독취소 기능 구현



#### 4. 메시지 전송 및 sorting 기능 구현

FormMain

CPS SYSTEM

메시지내용: 메시지 전송 테스트

받은메시지: (필터링 언어를 고르세요)

부재메시지:

구독하기

구독 목록보기

☐ 정관구사원분부팀

구독취소

전송

이름

직책

부서

구독

검색

전송 성공하였습니다.

확인

FormMain

CPS SYSTEM

메시지내용: 로그인 한 회원번호 : 1234

받은메시지: (필터링 언어를 고르세요)

부재메시지:

구독하기

구독 목록보기

☐ 송원사원/명업직

구독취소

전송

이름

직책

부서

구독

검색

FormMain

CPS SYSTEM

메시지내용: 로그인 한 회원번호 : 1234

받은메시지: (필터링 언어를 고르세요)

부재메시지:

구독하기

구독 목록보기

☐ 송원사원/명업직

구독취소

전송

이름

직책

부서

구독

검색

#### 5. 메시지 수신 및 부재메시지 기능 구현

CPS SYSTEM

메시지내용: 부재 메시지 테스트 1

받은메시지: (필터링 언어를 고르세요)

구독하기

전송

이름

직책

부서

구독

검색

CPS SYSTEM

메시지내용: 부재 메시지 테스트 2

받은메시지: (필터링 언어를 고르세요)

구독하기

전송

이름

직책

부서

구독

검색

CPS SYSTEM

메시지내용: 부재 메시지 테스트 3

받은메시지: (필터링 언어를 고르세요)

구독하기

전송

이름

직책

부서

구독

검색

#### 6. My-SQL 데이터베이스 사용



	Field	Type	Null	Key
▶	mem_name	varchar(45)	NO	
	mem_password	varchar(45)	NO	
	mem_department	varchar(45)	NO	
	mem_position	varchar(45)	NO	
	mem_jobyear	int(11)	NO	
	mem_number	varchar(45)	NO	PRI
	mem_session	int(11)	NO	

- 참여자 인적사항

1. 프로젝트 팀원

<b>성명</b>		정 진 구
<b>소속</b>		인하공업 전문대학
<b>직책</b>		현장 실습생
<b>연락처</b>	<b>C.P</b>	010 - 5845 - 4080
	<b>E-mail</b>	sbwls4210@naver.com

<b>성명</b>		박 혜 은
<b>소속</b>		인하공업 전문대학
<b>직책</b>		현장 실습생
<b>연락처</b>	<b>C.P</b>	010 - 3340 - 9673
	<b>E-mail</b>	haliey931220@gmail.com

<b>성명</b>		이 승 원
<b>소속</b>		인하공업 전문대학
<b>직책</b>		현장 실습생
<b>연락처</b>	<b>C.P</b>	010 - 7667 - 6873
	<b>E-mail</b>	tmddnjs7667@naver.com

## 목차

### 1. 프로젝트 착수

- 1.1 프로젝트 개요
- 1.2 프로젝트 목표
- 1.3 프로젝트 목적 및 기대효과
- 1.4 프로젝트 진행 방법
- 1.5 개발환경
- 1.6 프로젝트 일정

### 2. 요구사항 명세서

- 2.1 요구사항 명세서
- 2.2 기능 명세서

### 3. 설계 명세서(서버)

- 3.1 DataBase Table
- 3.2 Global Variable
- 3.3 DataBase Mapper Method
- 3.4 슈도코드

### 4. 설계 명세서(클라이언트)

- 4.1 GUI 설계
- 4.2 슈도코드

### 5. Server Install Manual

- 5.1 DataBase 환경 조성
- 5.2 ServerForm.exe 환경 조성

### 6. Client User Manual

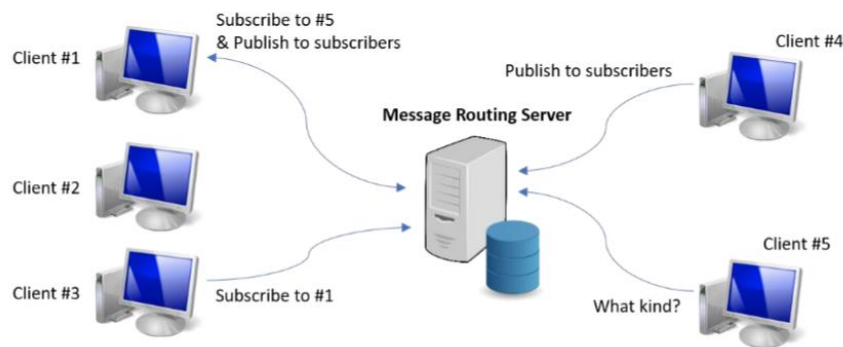
- 6.1 서버 화면
- 6.2 로그인 화면
- 6.3 회원가입 화면
- 6.4 메인 화면

### 7. 부록

- 7.1 최종 테스트 결과 및 검토
- 7.2 참고문헌
- 7.3 프로젝트 소감
- 7.4 프로젝트 구현 소스코드 링크(Git)

## 1. 프로젝트 착수

### 1.1 프로젝트 개요



‘다중 Client가 Publish와 Subscribe 모두 가능하게 하는 메시지 라우팅 서버’ 라는 프로젝트 주제를 들고, 회사 내에서 내가 받고 싶은 사람의 메시지만 받고, 나의 메시지를 원하는 사람들에게만 메시지를 주는 프로그램을 만들면 좋겠다는 생각에 만들게 되었다.



이 프로그램은 직원들이 자신이 원하는 사람을 구독하고, 그 사람의 메시지를 실시간으로 전달받을 수 있게 해준다. 만약, 직원이 접속 중이 아니라면, 부재메시지에 메시지가 쌓이게 될 것이다. 또한, 직원들은 자신을 구독하는 사람들에게 메시지를 전달할 수 있다.

## 1.2 프로젝트 목표

### Plan A(최대)

- 로그인 및 회원가입 기능 구현 (표준 정규 식 활용 유효성 검사 포함)
- 회원가입 및 로그인 시 비밀번호 암호화 기능 적용
- TCP 소켓 방식 멀티 쓰레드를 활용한 다중 클라이언트 서버와 메시지 라우팅 서버 구축
- DB를 활용한 서버 기반의 메시지 전송 시스템 구현
- DB를 활용한 서버 기반의 메시지 sorting 시스템 구현
- DB를 활용한 클라이언트 to 클라이언트 구독 시스템 구현
- Session flag를 사용한 중복 로그인 방지 및 부재 메시지 sort 기능 구현
- TCP 소켓을 활용한 Ping 방식의 클라이언트 접속 유무 확인 기능
- 클라이언트 메시지 구독 및 전송기능의 UI 시스템 구현
- 수신된 메시지를 클라이언트가 원하는 언어로 번역(필터링)하는 기능 구현

### Plan B(최소)

- 로그인 및 회원가입 기능 구현 (유효성 검사, 비밀번호 암호화 제외)
  - TCP 소켓 방식 멀티 쓰레드를 활용한 다중 클라이언트 서버와 메시지 라우팅 서버 구축
  - DB를 활용한 서버 기반의 메시지 전송 시스템 구현
  - DB를 활용한 서버 기반의 메시지 sorting 시스템 구현
  - DB를 활용한 클라이언트 to 클라이언트 구독 시스템 구현
  - TCP 소켓을 활용한 Ping 방식의 클라이언트 접속 유무 확인 기능
  - 클라이언트 메시지 구독 및 전송기능의 UI 시스템 구현
-



### 1.3 프로젝트 목적 및 기대효과

#### 목적

- TCP 소켓 통신에 기반하여 모든 클라이언트가 publish/subscribe 할 수 있는 실시간 사내 통신 시스템을 구현한다.
- 기존의 메신저 혹은 메시지 통신 시스템(ex. 카카오톡)과 달리 사내 통신망을 활용하여 인터넷이 연결되지 않은 상태에서도 메시지를 전송 및 수신할 수 있는 통신 시스템을 구현한다.

#### 기대효과

- 구현된 시스템을 사내에 적용하여 분산적이며 유동적인 업무 지시체계를 확립하여 상하계층간 자율적인 소통이 활성화될 것이다.
- 구독 방식의 메시지 sorting을 활용하여 선택적이며 각 부서간 독립적인 사내 통신 체계 확립하며 정확한 업무 지시가 가능해질 것이다.
- 구독 및 sorting기능을 적극 활용 및 응용하여 사내 알림 서비스, 업무 지시 서비스, 업무 보고 서비스 등 다양한 기능으로 발전할 것이다.

### 1.4 프로젝트 진행 방법

- 공동 개발 방식 : Git Hub를 활용하여 형상관리 및 공동 개발을 한다.
  - 이슈 관리 : 기본적으로 폭포수 모델(SDLC)를 따르며 설계 및 구현 시 프로토타이핑 방식을 채택한다.
  - 개발 구분
    - 요구사항 수집 : 팀원 전원이 함께 진행
    - 요구사항 분석 : 팀원 전원이 함께 진행
    - 시스템 설계 : 각자 기능을 분할하여 정해진 틀안에서 설계 진행
    - 시스템 구현 : 각자 담당한 기능을 정해진 형식으로 구현 후 조합 진행
    - 시스템 테스트 : 팀원 전원이 함께 진행
    - 시스템 유지 보수 : 각자 구현한 기능에 대한 유지 보수 진행
-

## 1.5 개발 환경



- OS 환경 : Window 10 pro
- 개발 Tools : visual studio 2019 (C#, NET, Windows Form),
- 개발 언어 : C#
- DataBase : MySQL 5.7 community server, MySQL Workbench 8.0

## 1.6 프로젝트 일정

개발구분	세부항목	1 주차	2 주차	3 주차	4 주차
계획 및 분석	주제 선정				
	요구사항 수집				
	요구사항 분석				
	프로젝트 세부 계획				
설계	시스템 설계				
	인터페이스 설계				
구현	구현				
테스트	단위테스트				
	통합테스트				
완료	프로젝트 완료				

## 2. 요구사항 명세서

### 2.1 요구사항 명세서

요구사항 명세서					프로젝트 명 : CPS System	
					작성 : 2019-07-22	
No	ID	Depth1	Depth2	상세내용	개발분류	비고
01	R-01	로그인	사번	사원번호를 입력 받을 창이 필요하다.	화면	
02	R-02		비밀번호확인	비밀번호를 입력 받을 창이 필요하다.	화면	
03	R-03		로그인	로그인 기능을 수행할 버튼이 필요하다.	화면	
04	R-04		로그인	로그인 버튼 클릭 시 로그인 기능 수행	기능	
05	R-05	회원가입	회원가입	회원가입 기능을 수행할 버튼이 필요하다.	화면	
06	R-06		회원가입	회원가입 버튼 클릭 시 회원가입 기능 수행	기능	
07	R-07		이름	이름을 입력 받을 창이 필요하다.	화면	
08	R-08		이름	한글 표준 사용 준수	기능	
09	R-09		비밀번호	비밀번호를 입력 받을 창이 필요하다.	화면	
10	R-10		비밀번호	*로 표시	기능	
11	R-11		비밀번호	숫자, 영문만 입력 가능	기능	
12	R-12		비밀번호확인	비밀번호를 입력 받을 창이 필요하다.	화면	
13	R-13		직책	직책을 입력 받을 창이 필요하다.	화면	
14	R-14		직책	사원, 선임, 책임, 사장 순으로 선택 가능하다.	기능	
15	R-15		직책	기본값 사원	기능	
16	R-16		부서	영업 팀, 개발 팀, 본부 팀, 설비 팀 순으로 선택 가능하다.	기능	
17	R-17		부서	기본값 0	기능	
18	R-18		부서	부서를 입력 받을 창이 필요하다.	화면	
19	R-19		입사연도	입사연도를 입력 받을 창이 필요하다.	화면	
20	R-20		입사연도	숫자만 입력 가능	화면	
21	R-21		가입취소	가입취소 기능을 수행할 버튼이 필요하다.	화면	
22	R-22		가입취소	가입취소 버튼 클릭 시 현재 품을 종료하는 기능 수행	기능	
23	R-23		회원가입	회원가입 기능을 수행할 버튼이 필요하다.	화면	
24	R-24	메인	메시지 전송	메시지를 입력 받을 창이 필요하다.	화면	
25	R-25		메시지 전송	글자수 제한(100 자)	기능	
26	R-26		전송	메시지 전송 기능을 수행할 버튼이 필요하다.	화면	
27	R-27		전송	전송 버튼 클릭 시 메시지 전송 기능 수행	기능	
28	R-28		수신 메시지	수신 메시지를 띄워줄 창이 필요하다.	화면	

29	R-29	수신 메시지	받은 메시지가 없을 때 '받은 메시지가 없습니다.'라는 라벨을 띄운다.	기능	
30	R-30	수신 메시지	받은 메시지가 있을 때 받은 메시지를 띄어 준다.	기능	
31	R-31	수신 메시지 - 번역	받은 메시지를 어떤 언어로 번역할지 선택할 창이 필요하다.	화면	
32	R-32	수신 메시지 - 번역	언어는 일본어, 영어, 중국어 중 선택할 수 있다.	기능	
33	R-33	수신 메시지 - 번역	언어를 선택하면 받은 메시지를 번역해주는 기능 수행	기능	
34	R-34	부재메시지	부재 메시지를 띄워줄 창이 필요하다.	화면	
35	R-35	부재메시지	부재 메시지가 없을 때 '부재 메시지가 없습니다.'라는 라벨을 띄운다.	기능	
36	R-36	부재메시지	부재 메시지가 있을 때 받은 메시지를 띄어 준다.	기능	
37	R-37	구독하기 - 이름	이름을 입력 받을 창이 필요하다.	화면	
38	R-38	구독하기 - 직책	직책을 입력 받을 창이 필요하다.	화면	
39	R-39	구독하기 - 부서	부서를 입력 받을 창이 필요하다.	화면	
40	R-40	구독하기 - 목록	검색한 정보를 띄어 줄 창이 필요하다.	화면	
41	R-41	구독하기 - 검색	검색 기능을 수행할 버튼이 필요하다.	화면	
42	R-42	구독하기 - 검색	검색 버튼 클릭 시 회원 검색 기능 수행	기능	
43	R-43	구독하기 - 구독	구독 기능을 수행할 버튼이 필요하다.	화면	
44	R-44	구독하기 - 구독	구독 버튼 클릭 시 회원 구독 기능 수행	기능	
45	R-45	구독 목록 보기	구독중인 정보를 띄어 줄 창이 필요하다.	화면	
46	R-46	구독취소	구독 취소 기능을 수행할 버튼이 필요하다.	화면	
47	R-47	구독취소	구독 취소 버튼 클릭 시 구독중인 회원 취소 기능 수행	기능	
48	R-48	연결종료	연결종료 기능을 수행할 버튼이 필요하다.	화면	
49	R-49	연결종료	연결종료 버튼 클릭 시 서버 연결 종료 기능 수행	기능	

## 2.2 기능 명세서

요구사항 ID	R-0003	요구사항명	로그인 기능을 수행할 버튼이 필요하다.	구분	기능
요구사항내역	1. 입력 받은 회원번호와 비밀번호가 데이터베이스에 존재하는 데이터와 일치할 시 메인 화면을 띄어 준다. 2. 서버가 비 가동 시 서버연결에 실패했다는 메시지화면을 띄어 준다. 3. 사번이 없다면 로그인이 실패했다는 메시지화면을 띄어 준다. 4. 비밀번호가 틀렸다면 로그인이 실패했다는 메시지화면을 띄어 준다.				

요구사항 ID	R-0006	요구사항명	회원가입 버튼 클릭 시 회원가입 기능 수행	구분	기능
요구사항내역	1. 모든 입력창이 입력되어진 상태로 회원가입 버튼을 클릭하면 데이터베이스에 입력된 정보를 넣는다. 2. 입력 받은 입사연도와 직책, 부서를 바탕으로 회원번호를 생성하여 띄어 준다. 3. 현재 화면은 종료되고 로그인 화면으로 돌아간다.				

요구사항 ID	R-0027	요구사항명	전송 버튼 클릭 시 메시지 전송 기능 수행	구분	기능
요구사항내역	1. 메시지내용이 null 이면 '메시지내용을 입력하세요'라는 메시지창을 띄어 준다. 2. 메시지내용 글자수가 초과하면 '글자수가 초과했습니다'라는 메시지창을 띄어 준다. 3. 서버와 통신하여 나를 구독하는 사람에게 메시지를 보낸다.				

요구사항 ID	R-0033	요구사항명	언어를 선택하면 받은 메시지를 번역해주는 기능 수행	구분	기능
요구사항내역	1. 받은 메시지를 기본적으로 그냥 보여준다. 2. 영어, 일본어, 중국어 중 언어 선택 시 받은 메시지를 번역해서 보여준다.				

요구사항 ID	R-0042	요구사항명	검색 버튼 클릭 시 회원 검색 기능 수행	구분	기능
요구사항내역	1. 이름, 직책, 부서 중 한가지 이상을 입력 받는다. 2. 입력 받은 값을 가진 회원을 모두 띄어 준다.				

요구사항 ID	R-0044	요구사항명	구독 버튼 클릭 시 회원 구독 기능 수행	구분	기능
요구사항내역	1. 리스트에 체크된 아이템이 있는지 확인한다. 2. 이미 구독 된 회원인지 혹은 나 자신인지 중복확인을 한다. 3. 2 번이 아니면 데이터를 저장한다. 4. 구독중인 회원의 정보를 데이터베이스에서 찾아, 구독 목록보기-목록에 띄어 준다.				

요구사항 ID	R-0047	요구사항명	구독 취소 버튼 클릭 시 구독중인 회원 취소 기능 수행	구분	기능
요구사항내역	1. 리스트에 체크된 아이템이 있는지 확인한다. 2. 체크된 회원을 데이터베이스에서 지운다. 3. 현재 구독중인 회원만 검색하여 구독 목록보기- 목록에 띄어 준다.				

### 3. 설계 명세서(서버)

#### 3.1 DataBase Tables

Member					
PK	FK	Null	Name	Type	
			mem_name	VARCHAR(45)	
			mem_password	VARCHAR(45)	
			mem_position	VARCHAR(45)	
			mem_department	VARCHAR(45)	
			mem_jobyear	INT	
✓			mem_number	VARCHAR(45)	
			mem_session	INT	

Repository					
PK	FK	Null	Name	Type	
✓	✓		post_number	VARCHAR(45)	
			data_time	VARCHAR(45)	
			data_context	VARCHAR(200)	
		✓	send_number	VARCHAR(45)	

subscribe					
PK	FK	Null	Name	Type	
✓	✓		mem_number	VARCHAR(45)	
			mem_subscribe	VARCHAR(1000)	

#### 3.2 Global Variable

Class Name	Method Name	Method 설명	Parameter	return
messageDB	readSub()	메시지를 전송한 사원을 구독하는 사람을 불러오는 메서드이다.	string	string 성공: 구독자 명단 실패: null
	searchMessage()	클라이언트 자신에게 수신될 메시지가 repository에 있는지 확인하는 메서드이다.	void	bool 성공: true 실패: false
	insertMessage()	전송한 메시지를 각 구독자 별로 repository에 저장하는 메서드이다.	string	void
	readMessage()	repository에 있는 메시지를 읽어오는 메서드이다.	void	List<string> 성공: 메시지 리스트 실패: null
	deleteMessage()	전송이 완료된 repository의 메시지를 삭제하는 메서드이다.	void	void
Make Member_num	mknum()	회원가입 시 입사연도, 직책, 부서를 가지고 일정한 방식에 따라 사원번호(id 로 활용)를 자동으로 생성하는 메서드이다.	string , string , string	string 성공: 사원번호 실패: null

### 3.3 DataBase Mapper Method

Class Name	Method Name	Method 설명	Parameter	return
<b>mapperDB</b>	selectLogin()	클라이언트가 로그인 시 해당 아이디의 유무를 체크하고 그에 따른 비밀번호 체크를 하는 메서드이다.	memDB	bool 성공: true 실패: false
	insertSub()	한 클라이언트가 다른 클라이언트를 구독할 때 사용하는 메서드이다.	string , string	List<string> 성공: 구독자정보 실패: null
	LoadFormPutData()	클라이언트가 로그인 시 해당 아이디의 부재 메시지를 검색하고 부재메시지가 있다면 반환해주는 메서드이다.	string	List<string> 성공: 메시지정보 실패: null
	FindThenDelete()	구독 취소 시 취소할 클라이언트에 해당하는 사원번호를 검색 후 삭제하는 메서드이다.	string , string	List<string> 성공: 삭제할 구독자 정보 실패: null
	selectSearchName()	구독하기 기능에서 이름, 직책, 부서를 사용하여 해당 사원의 사원번호를 찾아 사원번호를 반환해주는 메서드이다.	string , string , string	List<string> 성공: 사원번호 리스트 실패: null
	insertJoin()	회원가입 시 member DB 에 저장하는 메서드이다.	memDB	void
	selectNumber()	사원번호 랜덤 생성 시 사원번호 중복 확인에 사용되는 메서드이다.	string	bool 성공: true 실패: false
	connectDB()	Database 연결을 담당하는 메서드이다.	void	void
	closeDB()	Database 연결종료를 담당하는 메서드이다.	void	void



### 3.4 슈도코드

<b>Class Name</b>	ServerForm
<b>Method Name</b>	TCPServer()
<b>Method 설명</b>	서버 가동버튼 클릭 시 TCP 소켓을 Bind 하고 클라이언트 소켓의 Accept 를 대기하는 기능을 한다.
<b>슈도 코드</b>	<pre> public void TCPServer() {     new Tread(count_mem).start();     new Tread(logout_mem).start();     //count_mem, logout_mem 메서드를 백 스레드로 돌린다.      while(1){         if(연결종료)             break;          clientSocket &lt;- TcpListener.Accept();         sortSocket &lt;- TcpListener.Accept();         pingSocket &lt;- TcpListener.Accept();         //클라이언트에서 요청하는 소켓을 받아들임          ClientListen cHandler &lt;- new ClientListen(clientSocket, sortSocket, pingSocket);         //ClientListen 클래스 변수 cHandler 생성          new Tread(cHandler.chat).start();         new Tread(cHandler.sorter).start();         new Tread(cHandler.pingging).start();         //ClientListen 클래스의 chat, sorter, pingging 기능을         각각 백 스레드로 돌림.     } } </pre>

<b>Class Name</b>	ServerForm
<b>Method Name</b>	Count_mem()
<b>Method 설명</b>	현재 접속중인 클라이언트의 수를 체크하고 로그인 클라이언트를 체크하는 기능을 한다.
<b>슈도 코드</b>	<pre> public void count_mem(){     while(1){         if(접속 신호){             mem_count ++;             print(mem_number+"님이 로그인 하셨습니다.");             print("현재 접속자 수 : "+mem_count);         }elseif(회원가입 신호){             print(mem_number+"님이 회원가입 하셨습니다.");         }elseif(연결종료){             break;         }     } } </pre>

<b>Class Name</b>	ServerForm
<b>Method Name</b>	Logout_mem()
<b>Method 설명</b>	현재 접속중인 클라이언트의 수를 체크하고 로그아웃 클라이언트를 체크하는 기능을 한다.
<b>슈도 코드</b>	<pre> public void logout_mem(){     while(1){         if(접속종료 신호){             mem_count--;             print(mem_number+"님이 로그아웃 하였습니다.");             print("현재 접속자 수 : "+mem_count);         }elseif(연결종료){             break;         }     } } </pre>

<b>Class Name</b>	ClientListen
<b>Method Name</b>	ClientListen()
<b>Method 설명</b>	ClientListen Class 메서드이다.
<b>슈도 코드</b>	<pre> public ClientListen(Socket a, Socket b, Socket c){     this.clientSocket &lt;- a;     this.sortSocket &lt;- b;     this.pintSocket &lt;- c; } </pre>

<b>Class Name</b>	ClientListen
<b>Method Name</b>	Chat()
<b>Method 설명</b>	클라이언트의 로그인, 회원가입, 메시지 전송, 구독, 구독취소의 기능을 담당한다.
<b>슈도 코드</b>	<pre> public void chat(){     while(1){         firstQ &lt;- clientReader.ReadLine();         //최초 신호 받아옴.         if(firstQ=="회원가입"){             joinQ &lt;- true;             //다음 while 문 기능 중지 플레그.             while(1){                 streamStr &lt;- clientReader.ReadLine();                 //회원가입 정보 받아옴.                 string[] split &lt;- Split(streamStr, "/");                 for(i=0;i&lt;split.size();i++)                     split[i] 의 유효성 검사;                 if(검사 통과)                     break;                 clientWriter.WriteLine(실패신호);             }             mapperDB.insertJoin();//DB 에 저장 쿼리 실행             clientWriter.WriteLine(성공신호);             break; //chat 스레드 종료         }elseif(firstQ=="로그인"){ </pre>

```

streamStr <- clientReader.ReadLine();
string[] split <- Split(streamStr, "/");
bool log_check <- mapperDB.selectlogin(split);
//id , pw 확인
if(log_check){
    clientWriter.WriteLine("로그인 성공신호");
    joinQ <- false;
    break;
}
clientWriter.WriteLine("로그인 실패신호");
}elseif(연결종료)
    break;
}
while(!joinQ){
    if(연결종료)
        break;
    streamStr <- clientReader.ReadLine();
    string[] split <- Split(streamStr, "/");
    if(split[0]=="전송")
        mapperDB.insertMessage(split[1]);
    elseif(split[0]=="검색"){
        getsublist
            <- mapperDB.selectSearchName(split[1], split[2], split[3]);
        //해상사원(1:이름,2:직책,3:부서) 검색후 존재한다면 반환.
        clientWriter.WriteLine(getsublist);
    }elseif(split[0]=="구독취소"){
        mapperDB.findThenDelete(split[1], mem_number);
        //해당사원(1:사원번호)검색 후 구독삭제.
    }elseif(split[0]=="구독"){
        mapperDB.insertSub(split[1], mem_number);
        //해당사원(1:사원번호) 구독 추가.
    }elseif(split[0]=="로그인구독목록"){
        mapperDB.LoadFormPutData(split[1]);
        //로그인한 사원(1:사원번호)의 구독목록을 불러옴.
    }
}
}

```

<b>Class Name</b>	ClientListen
<b>Method Name</b>	Sorter()
<b>Method 설명</b>	전송되어 DB 에 저장된 메시지를 구독하는 클라이언트들에게 sorting 해주는 기능을 하는 메서드이다.
<b>슈도 코드</b>	<pre> public void sorter() {      sortQ &lt;- sortReader.ReadLine();      while(1){         if(접속종료) {             break;         }         if(sortQ=="회원가입") {             break;         }         elseif(sortQ=="로그인"){             //부재메시지를 불러온다.             List(string) dataMs &lt;- messageDB.readMessage();             sortWriter.WriteLine(dataMs);             sortQ &lt;- "솔팅기능하렴";         }else{             bool check &lt;- messageDB.serchMessage();             if(check){                 //수신된 메시지가 있는지 확인.                 List(string) dataMs &lt;- messageDB.readMessage();                 sortWriter.WriteLine(dataMs);             }             messageDB.deleteMessage();//수신완료후 삭제              Tread.Sleep(1000);         }     } } </pre>

<b>Class Name</b>	ClientListen
<b>Method Name</b>	Pinging()
<b>Method 설명</b>	TCP 방식으로 연결된 클라이언트에게 일정간격으로 신호를 발송해 클라이언트가 살아있는지 확인하는 기능을 하는 메서드이다.
<b>슈도 코드</b>	<pre> public void pingging(){     try{         firstQ &lt;- pingReader.ReadLine();          if(firstQ != "회원가입"){             while(1){                 pingWriter.WriteLine("ping");                 Tread.Sleep(1000);                  if(접속종료){                     break;                 }             }         }     }catch{         logout_mem &lt;- 접속 종료 신호 발송;     } } </pre>

## 4. 설계 명세서(클라이언트)

### 4.1 GUI 설계

FormJoin			
Control	Variable Name	Text	Role
TextBox	txtName	null	회원가입 시 이름 입력 창
TextBox	txtPasswdj	null	회원가입 시 비밀번호 입력 창
TextBox	txtPasswdck	null	회원가입 시 비밀번호 확인 입력 창
TextBox	txtJobyear	null	회원가입 시 입사연도 입력 창
ComboBox	cmbPosition	null	회원가입 시 직책 선택창
ComboBox	cmbDepartment	null	회원가입 시 부서 선택창
Button	btnJoinDB	"회원가입"	회원가입 시도 이벤트 버튼
Button	btnCandle	"취소"	회원가입 취소 이벤트 버튼
FormMain			
Control	Variable Name	Text	Role
TextBox	txtsendbox	null	전송 메시지 입력 창
TextBox	txtpostbox	null	수신 메시지 출력 창
TextBox	txtleavemessage	null	부재 메시지 출력 창
TextBox	txtNames	null	검색 조건(이름) 입력 창
ComboBox	cmbPositions	null	검색 조건(직책) 선택창
ComboBox	cmbDepartments	null	검색 조건(부서) 선택창
CheckListBox	subListBox	null	구독 검색 결과 출력 및 선택창
CheckListBox	SubedListBox	null	구독 리스트 출력 및 선택창
Button	btnsearch	"검색"	회원을 검색하는 이벤트 버튼
Button	btnsub	"구독"	구독기능 이벤트 버튼
Button	btnsubcandle	"구독취소"	구독취소기능 이벤트 버튼
Button	btnserver	"접속종료"	접속종료기능 이벤트 버튼
Button	btncancel	"전송"	메시지 전송기능 이벤트 버튼
FormLogin			
Control	Variable Name	Text	Role
TextBox	txtId	null	로그인 시 사원번호 입력 창
TextBox	txtPassword	null	로그인 시 비밀번호 입력 창
Button	btnLogin	"로그인"	로그인 기능 이벤트 버튼
Button	btnJoin	"회원가입"	회원가입 폼 띄우는 이벤트 버튼

## 4.2 슈도코드

<b>Class Name</b>	FormLogin
<b>Method Name</b>	BtnLogin_Click(), BtnJoin_Click()
<b>Method 설명</b>	로그인 버튼과 회원가입 버튼을 클릭했을 때 반응하는 메서드이다.
<b>슈도 코드</b>	<pre> Public static string id; //로그인을 할 클래스변수 선언 Public static string pw; Public static string ServerIPv4; //통신 설정을 위한 클래스변수 선언 Public static string ServerPort;  private void BtnLogin_Click(){     id &lt;- txtid.text;     pw &lt;- txtpasswd.text;     //text 창에 입력한 id 와 pw 를 클래스 변수에 넣는다.      frm &lt;- new FormMain();     frm.show();     //FormMain 창을 띄운다. }  private void BtnLogin_Click(){     frm &lt;- new FormJoin();     frm.show();     //FormJoin 창을 띄운다. }  private void BtnEdit_Click(){     frm &lt;- new SocketEdit();     frm.show();     // SocketEdit 창을 띄운다. } </pre>



<b>Class Name</b>	FormMain
<b>Method Name</b>	FormMain_Load()
<b>Method 설명</b>	FormMain 창을 띄울 때 최초 로드 하는 메서드이다.
<b>슈도 코드</b>	<pre> private void FormMain_Load(){     clientSocket.Connect(FormLogin.ServerIPv4, FormLogin.ServerPort);     //클라이언트 소켓통신을 서버로 연결 요청함.     sortSocket.Connect(FormLogin.ServerIPv4, FormLogin.ServerPort);     //sorting 소켓통신을 서버로 연결 요청함.     pingSocket.Connect(FormLogin.ServerIPv4, FormLogin.ServerPort);     //pingging 소켓통신을 서버로 연결 요청함.      clientWriter.WriteLine("로그인");     string sendLogin &lt;- FormLogin.id, FormLogin.pw     clientWriter.WriteLine(sendLogin);//로그인 정보를 서버로보냄.      recvLogin &lt;- clientReader.ReadLine();     //로그인에 대한 응답이 오기를 기다림.     if(recvLogin=="로그인 성공"){         clientWriter.WriteLine("로그인구독목록");         string recvSearch &lt;- clientReader.ReadLine();         if(recvSearch != null){             print(recvSearch);             //불러온 구독목록을 구독창에 띄움.         }         sortWriter.WriteLine(FormLogin.id);         //로그인한 사원번호를 sorter 로 보냄.         pingWriter.WriteLine(로그인 신호);         //서버의 pinggin()에 로그인 신호를 보냄.          new Tread(recvMessage).start();         //메세지 수신 메서드를 백 스레드로 보냄.     } } </pre>

<b>Class Name</b>	FormMain
<b>Method Name</b>	recvMessage()
<b>Method 설명</b>	실시간으로 수신이 되는 메시지를 창에 띄우는 메서드이다.
<b>슈도 코드</b>	<pre> public void recvMessage(){     try{         sortWriter.WriteLine("부재메시지");         //부재메시지를 호출하는 신호를 sorter 로 보낸다.          string lastMs &lt;- sortReader.ReadLine();         if(lastMs != null){             print(lastMs); //부재메시지가 있다면 부재메시지창에 띄움.         }         while(1){             if(접속종료) break;              string postMs &lt;- sortReader.ReadLine();             //메시지가 수신될때 까지 기다림.             //선택한 언어에 맞게 필터링 작업.(번역기 API 사용)             if (FormMain.cmbStrCheck == "영어")             {                 postMs &lt;- transStr(postMs,"en");             }else if (FormMain.cmbStrCheck == "일본어")             {                 postMs &lt;- transStr(postMs, "ja");             }else if (FormMain.cmbStrCheck == "중국어(간체)")             {                 postMs &lt;- transStr(postMs, "zh-CN");             }else if (FormMain.cmbStrCheck == "중국어(번체)")             {                 postMs &lt;- transStr(postMs, "zh-TW");             }             //필터링 후 수신창에 메시지 띄움.             print(postMs);         }     }catch{         print("수신 error");     } } </pre>

<b>Class Name</b>	FormMain
<b>Method Name</b>	Btnsub_Click(), Btnsubcancle_Click()
<b>Method 설명</b>	구독하고 싶은 멤버를 구독하는 메서드와 구독 취소하고 싶은 멤버를 구독 취소 하는 메서드이다.
<b>슈도 코드</b>	<pre> private void Btnsub_Click(){     try{        //구독 신호와 구독할 사람 정보를 보냄.         clientWriter.WriteLine("구독");         checklistData &lt;- sublistbox.checkedItem;         clientWriter.WriteLine(checklistData);         string rcvSubscribe &lt;- clientReader.ReadLine();         //구독 되었는지 신호와 현재 구독중인 정보를 받음.         if(rcvSubscribe){             subedlistbox.Items.Clear();             subedlistbox.print(checklistData);             //구독한 회원정보를 구독창에 띄움.         }else{             print("구독 실패");         }     }catch{         print("서버연결 error");    } }  private void Btnsubcancle_Click(){     try{        //구독취소 신호와 구독취소할 사람 정보를 보냄.         clientWriter.WriteLine("구독취소");         checklistData &lt;- subedlistbox.checkedItem;         clientWriter.WriteLine(checklistData);         string rcvSubscribe &lt;- clientReader.ReadLine();         //구독취소 되었는지 신호와 현재 구독중인 정보를 받음.         if(rcvSubscribe){             subedlistbox.Items.Clear();             subedlistbox.print(checklistData);             //구독한 회원정보를 구독창에 띄움.         }else{             print("구독 취소 실패");         }     }catch{         print("서버연결 error");    } } </pre>

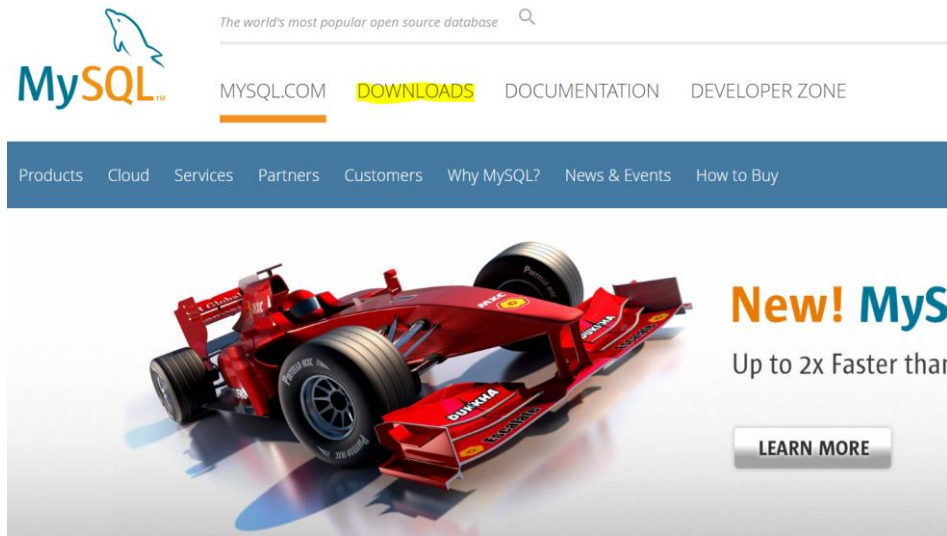
<b>Class Name</b>	FormJoin
<b>Method Name</b>	FormJoin_Load(), BtnJoinDB_Click()
<b>Method 설명</b>	FormJoin 창을 띄울 때 최초 로드 하는 메서드이다. 회원가입 버튼을 눌렀을 때 발생하는 이벤트 메서드이다.
<b>슈도 코드</b>	<pre> private void FormJoin_Load(){     clientSocket.Connect(FormLogin.ServerIPv4, FormLogin.ServerPort);     //클라이언트 소켓통신을 서버로 연결 요청함.     sortSocket.Connect(FormLogin.ServerIPv4, FormLogin.ServerPort);     //sorting 소켓통신을 서버로 연결 요청함.     pingSocket.Connect(FormLogin.ServerIPv4, FormLogin.ServerPort);     //pingging 소켓통신을 서버로 연결 요청함.      //회원가입 신호를 서버의 각 스레드로 보냄.     clientWriter.WriteLine("회원가입");     sortWriter.WriteLine("회원가입");     pingWriter.WriteLine("회원가입"); } }  private void BtnJoinDB_Click(){     sendJoin &lt;- null;     //회원가입 정보를 넣을 변수 선언     if(각 text 창이 null 값이 아닌가){         sendJoin             &lt;- txtNaem, txtPasswd, cmbPosition, cmbDepartment, txtJobyear;         clientWriter.WriteLine(sendJoin);     }else{         print(경고메시지);         return;     }     //회원가입 성공에대한 신호와 사원번호를 기다림.     recvJoin &lt;- clientReader.ReadLine();     if(recvJoin){ //성공했다면.         print("가입성공"+사원번호);     }else{         print(가입 실패에 대한 이유);     } } </pre>

## 5. Server Install Manual

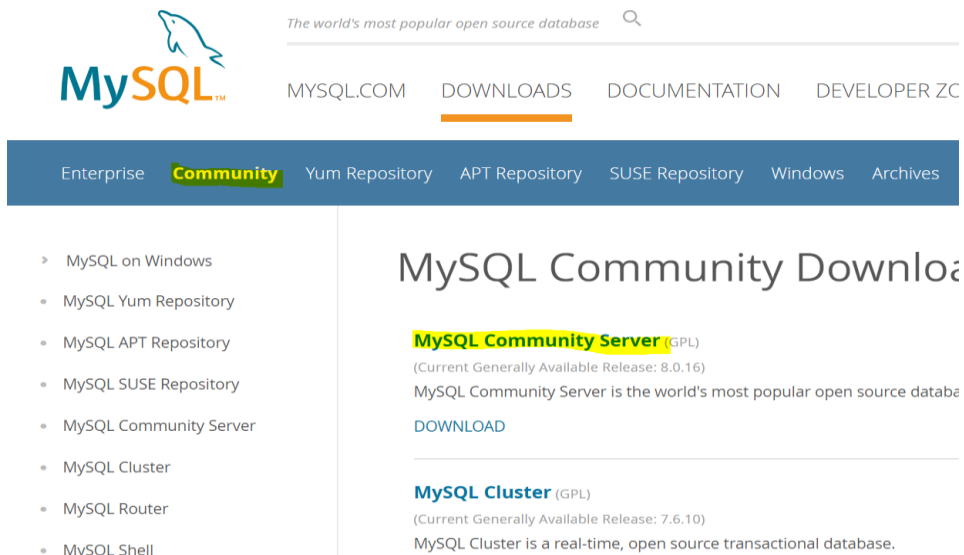
### 5.1 DataBase 환경 조성

#### 5.1.1 my-SQL 설치

(<https://www.mysql.com/>) -> DOWNLOADS 클릭



->Community 클릭 -> MySQL Community Server 클릭



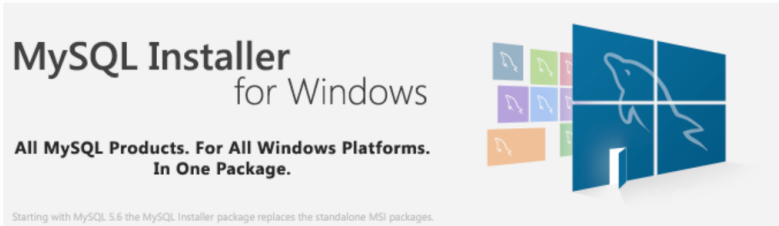
->Looking for previous GA versions? 클릭

**MySQL Community Server 8.0.16**

Select Operating System:

[Looking for previous GA versions?](#)

**Recommended Download:**



**MySQL Installer for Windows**

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

**Windows (x86, 32 & 64-bit), MySQL Installer MSI**

[Go to Download Page >](#)

**Other Downloads:**

->MySQL 5.7 version 선택 -> Go to Download Page 클릭

**MySQL Community Server 5.7.26**

Select Version:

Select Operating System:

Select OS Version:

**Recommended Download:**



**MySQL Installer for Windows**

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

**Windows (x86, 32 & 64-bit), MySQL Installer MSI**

[Go to Download Page >](#)

->Windows 버전 installer 클릭 후 다운로드 및 설치

Windows (x86, 32-bit), MSI Installer	5.7.26	414.8M	<a href="#">Download</a>
(mysql-installer-community-5.7.26.0.msi)		MD5: cc304f3a21518be14c0be780e743b0fc   <a href="#">Signature</a>	

## 5.1.2 my-SQL 스키마, 테이블 생성 (My-SQL Workbench 사용)

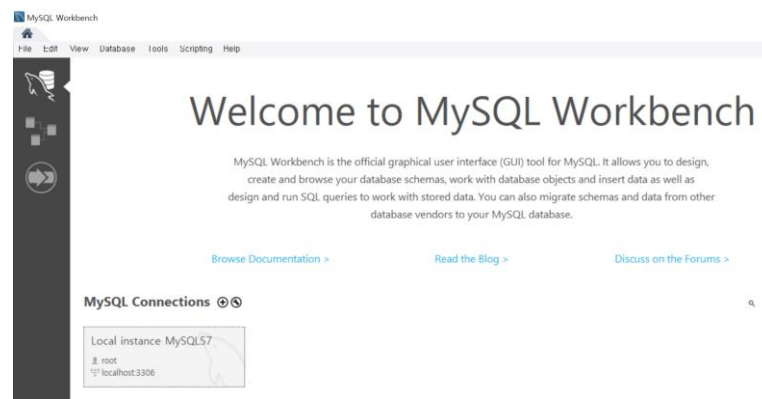
//Workbench 다운로드 : <https://dev.mysql.com/downloads/workbench/>

### - 스키마 생성

Workbench의 첫 화면 root로 connection 합니다.

(설치 과정에서 root 비밀번호 분실 주의)

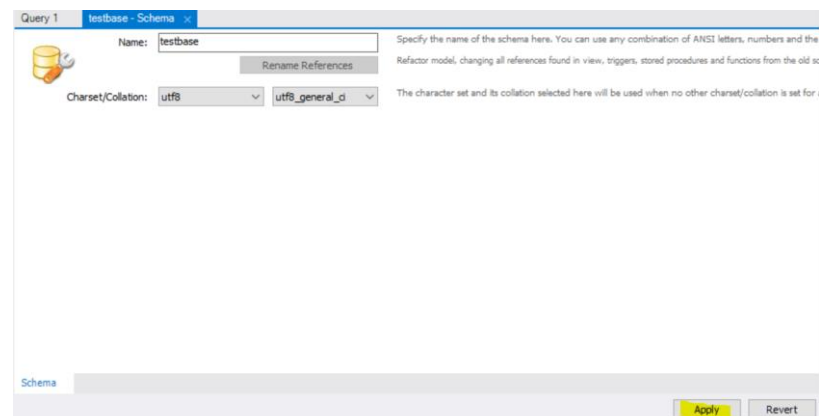
//이후 사용자를 생성하여도 되고 root로 진행해도 무방함.



->상단 메뉴의 스키마 생성버튼 클릭



->생성할 database name 지정 후 언어 셋팅을 utf8/utf8\_general\_ci로 하여 Apply 버튼을 누른다.



- 테이블 생성 (모든 테이블의 언어 셋팅은 utf8 로 스키마와 동일)

#### Member 테이블 생성

```
create table Member(mem_name varchar(45) not null,  
mem_password varchar(45) not null,  
mem_department varchar(45) not null,  
mem_position varchar(45) not null,  
mem_jobyear int not null,  
mem_number varchar(45) not null,  
mem_session int not null,  
constraint memTable_PK2 primary key(mem_number));
```

#### Repository 테이블 생성

```
create table repository(  
post_number varchar(45) not null,  
data_time varchar(45) not null,  
data_context varchar(200) not null,  
send_number varchar(45) not null  
);
```

#### Subscribe 테이블 생성

```
create table subscribe(  
mem_number varchar(45) not null,  
mem_subscribe varchar(1000) not null);
```

#### DB(work bench) 설정

상단 메뉴 Edit 클릭 -> Preferences...클릭 -> 좌측 SQL Editor 클릭

-> 하단 Safe Update 체크 해제.



## **\*Database Data Insert 과정에서 한글이 깨질 시\***

1. 파일 탐색기 설정 -> 숨김 파일 표시

2. 검색(메모장) -> 관리자 권한으로 실행 -> 파일 열기(디렉토리 검색)  
-> 파일 형태(모든 파일)

2. my-SQL 서버 설치 디렉토리 경로로 이동 -> my.ini 열기

(My-SQL 서버 설치 시 설치했던 경로로 이동한다.)

(일반적인 경로 : C:\Program files\MySQL\MySQL 버전\my.ini)

(위 경로에 없을 시 : C:\Program data\ MySQL\MySQL 버전\my.ini)

4. my.ini 파일 수정하기

- 수정 사항

```
[client]
default-character-set=utf8
```

```
[mysql]
default-character-set=utf8
#no-beep
```

```
[mysqld]
collation-server = utf8_unicode_ci
init-connect='SET NAMES utf8'
character-set-server = utf8
```

## 5.2 Server Install Manual - ServerForm.exe 환경 조성

### 5.2.1 Database 환경 조성

1. CPSSystem.zip 압축 풀기 (압축 비밀번호 : inhatc)
2. ServerForm.exe 파일 실행 -> 서버 설정 버튼 클릭



- Socket Port : 소켓 통신을 하기위한 서버의 port 번호이다. (기본값 = 7000)  
//주의사항 - 클라이언트 시스템의 port 번호와 일치해야 통신가능 함.
- DB Port : 서버 내에서 DB와 연동을 할 때 사용되는 port 번호이다. (기본값 = 3600)  
//주의사항 - My-SQL 설치 시 설정했던 connect port 번호와 일치해야 함.
- DB User : My-SQL 서버 내 설정된 User 명을 넣는다. (기본값 = root)
- DB Schema : My-SQL 서버 내 User 의 DB 명을 넣는다. (기본값 = testbase)
- DB Password : My-SQL 서버 내 User 의 Password 를 넣는다. (기본값 = inhatc)

3. 적용 버튼 클릭 -> 서버가동 버튼 클릭 후 정상가동 확인 -> 클라이언트 사용 가능 함.

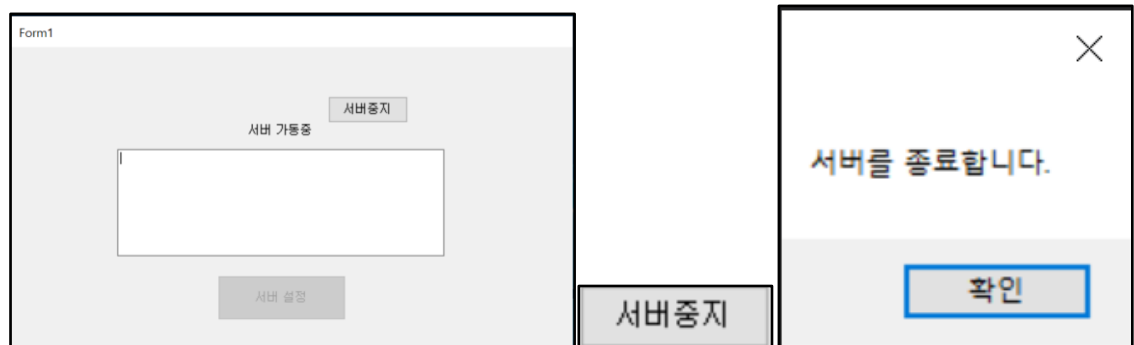
## 6 Client User Manual - 서버 화면

### 6.1 서버 화면

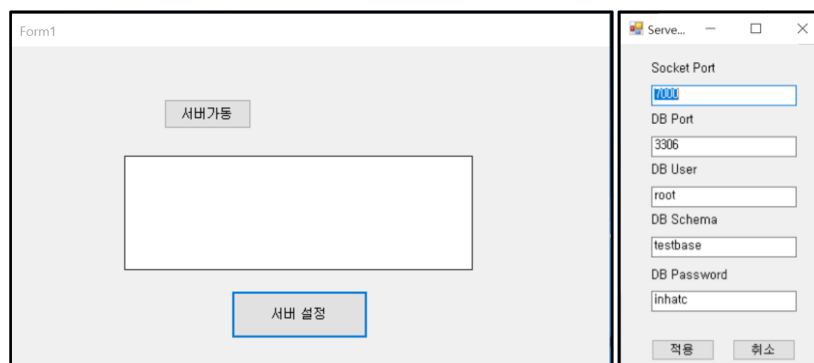
서버가 가동 되어있어야 클라이언트들이 모든 기능들을 사용할 수 있지만, 서버가 중지 되어있는 상태에서는 모든 기능들을 사용할 수 없습니다.



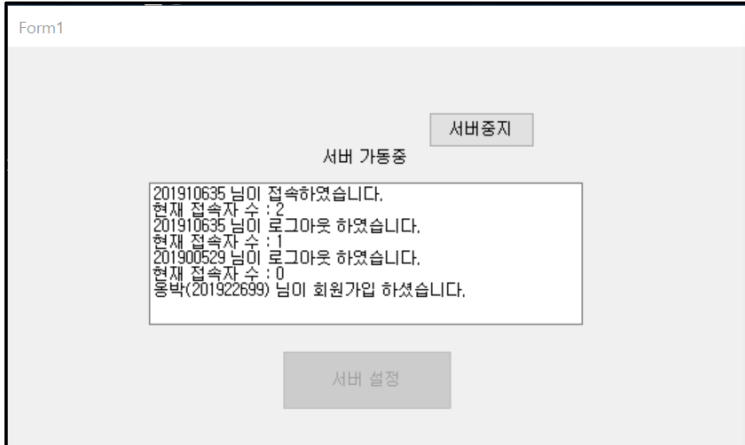
(서버화면 1) 서버가동 버튼을 클릭 시 -> 서버 가동버튼이 사라지며 서버 중지 버튼이 생깁니다.



(서버화면 2) 서버중지 버튼을 클릭 시 -> 창은 닫히며 서버가 종료됩니다.



(서버화면 3) 서버 설정 클릭 시 -> DB 와 socket Port의 설정이 가능합니다.



Form1

서버 중지

서버 가동중

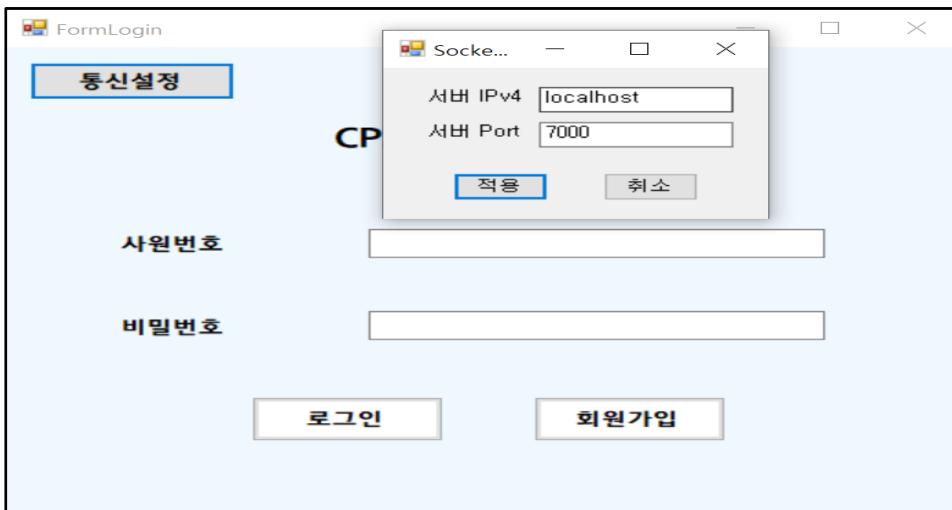
201910635 님이 접속하였습니다.  
 현재 접속자 수 : 2  
 201910635 님이 로그아웃 하였습니다.  
 현재 접속자 수 : 1  
 201900529 님이 로그아웃 하였습니다.  
 현재 접속자 수 : 0  
 동박(201922699) 님이 회원가입 하셨습니다.

서버 설정

(서버화면 4) 로그인 성공 시 -> 서버에서 어떤 사용자가 로그인, 로그아웃, 회원가입, 현재 몇 명이 접속했는지를 알 수 있습니다.

## 6.2 로그인 화면

로그인은 회원가입을 통해 생성된 사원번호와 비밀번호로 로그인 할 수 있습니다.



FormLogin

통신설정

CP

사원번호

비밀번호

로그인

회원가입

Socket...

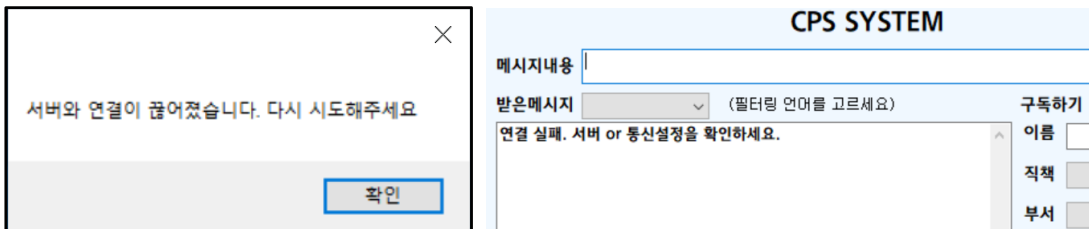
서버 IPv4: localhost

서버 Port: 7000

적용

취소

(로그인 화면 1) 통신 설정 -> 통신 설정으로 서버를 변경할 수 있습니다.



서버와 연결이 끊어졌습니다. 다시 시도해주세요

확인

CPS SYSTEM

메시지내용

받은메시지 (필터링 언어를 고르세요)

구독하기

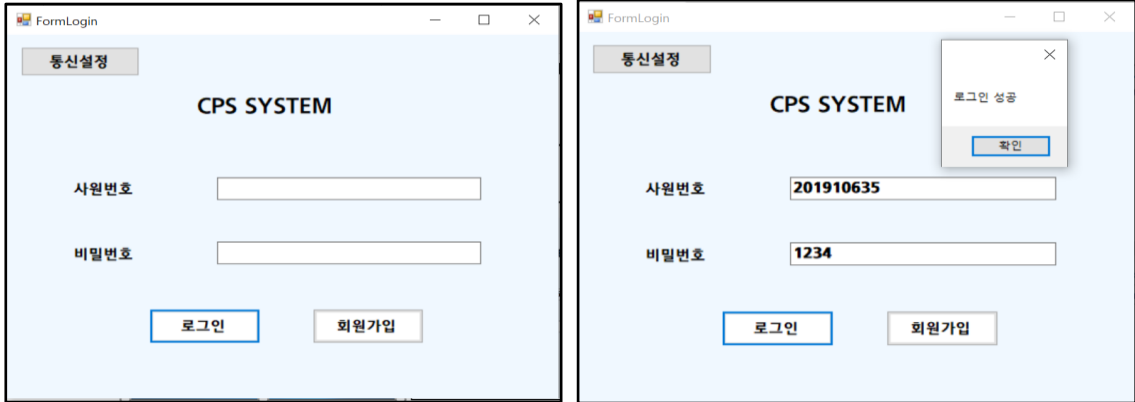
이름

직책

부서

연결 실패. 서버 or 통신설정을 확인하세요.

(로그인 화면 2) 서버가 종료했을 시 -> 모든 기능들을 사용할 수 없습니다.



(로그인 화면 3) 회원가입을 통해 생성된 사원 번호로 로그인성공

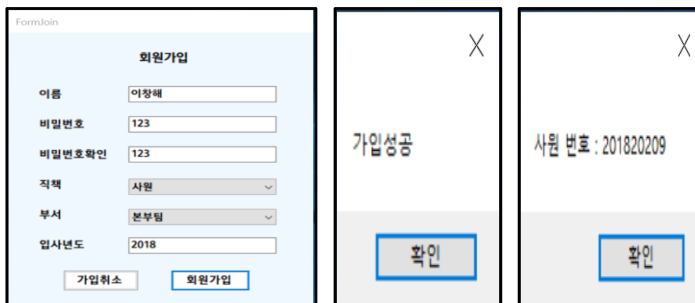
### 6.3 회원가입 화면

로그인 화면에서 회원가입 버튼을 클릭하면 회원가입 화면이 띄워집니다. 회원가입을 통해 자신의 사원번호를 생성합니다.

(회원가입 화면 1)



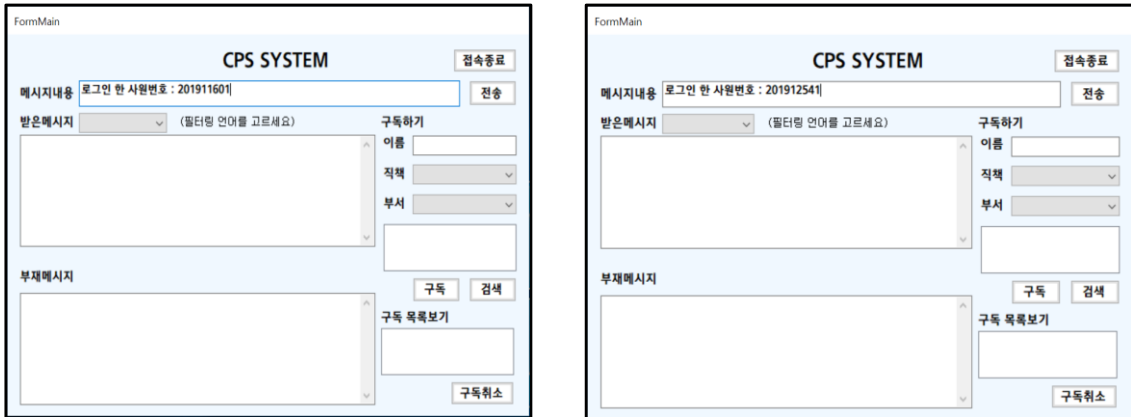
- ✓ 이름은 한글만 입력 가능합니다.
- ✓ 비밀번호, 비밀번호확인은 영어와 숫자 조합으로만 가능합니다.
- ✓ 직책, 부서는 combo박스 안에 있는 내용만 사용 가능합니다.
- ✓ 입사 년도는 숫자만 입력 가능합니다.
- ✓ 빈칸이 있으면 회원가입 불가능합니다.



(회원가입 화면 2) 회원가입 성공 시 -> 사원번호를 생성해줍니다.

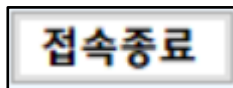
## 6.4 메인 화면

메시지를 주고받는 기능들을 사용할 수 있습니다.

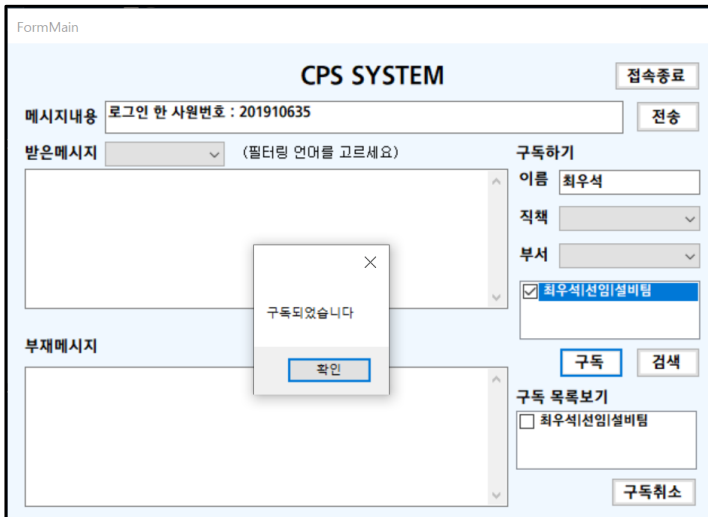


The screenshots show the 'CPS SYSTEM' main interface. The top bar includes 'FormMain' and '접속종료'. The main content area has a '메시지내용' field with a login number, a '전송' button, and a '받은메시지' list. Below this is a '부재메시지' section with a '구독' button and a '구독 목록보기' button. The right sidebar contains '구독하기' fields for '이름', '직책', and '부서', and a '구독 취소' button.

(메인 화면 1) 로그인 성공하여 메인 화면에 접속했을 시 -> 메시지 내용에 자신이 접속한 회원번호가 입력되어있습니다.




(메인 화면 2) 접속 종료 버튼은 클릭 시 -> 로그아웃 됩니다.

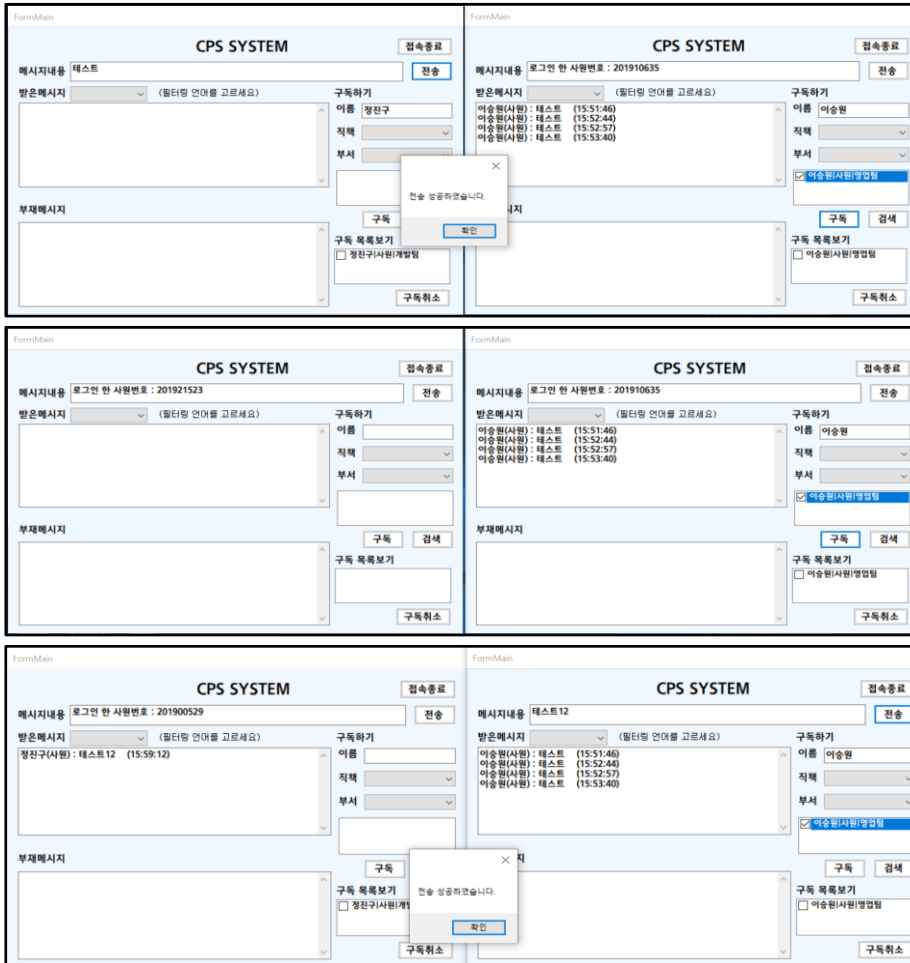


This screenshot shows the 'CPS SYSTEM' main interface with a '구독하기' (Subscribe) dialog box open. The dialog box has a '확인' (Confirm) button. The background shows the same interface as the previous screenshots, but with the '구독하기' dialog box open. The '구독하기' fields are filled with '최우석', '직책', and '부서'. The '구독 목록보기' button is also visible.

(메인 화면 3) 구독하기 - 이름, 직책, 부서 중에 아무거나 검색할 내용을 넣어 검색 버튼을 클릭 시 해당 존재하는 데이터가 뜨고 데이터를 클릭하여 구독 버튼을 클릭 시 구독목록보기에 해당 데이터가 전송됩니다.



(메인 화면 4) 구독 취소 - 구독목록보기에서 해당 데이터를 클릭 후 구독취소 버튼을 클릭하면 구독 취소되었습니다. 메시지와 함께 구독 목록보기에서 사라집니다.



(메인 화면 5) 메시지 전송 - 메시지 전송은 모든 사용자들이 메시지 전송을 할 수 있습니다. 하지만 자신을 구독한 사람이 없으면 그 메시지는 DB 에 저장되지 않고 사라집니다. 자신을 한명이라도 구독한 사람이 있으면 구독한 사람에게 메시지가 전송됩니다.

FormMain

### CPS SYSTEM

접속종료

메시지내용: 테스트123 전송

받은메시지: (필터링 언어를 고르세요)

이승원(사원): 테스트 (15:51:46)

이승원(사원): 테스트 (15:52:44)

이승원(사원): 테스트 (15:52:57)

이승원(사원): 테스트 (15:53:40)

구독하기

이름: 이승원

직책:

부서:

☒ 이승원(사원)명업팀

구독 검색

구독 목록보기

☐ 이승원(사원)명업팀

구독취소

×

전송 성공하였습니다.

확인

부재메시지

FormMain

### CPS SYSTEM

접속종료

메시지내용: 로그인 한 사원번호 : 201900529 전송

받은메시지: (필터링 언어를 고르세요)

부재메시지: 장진구(사원): 테스트123 (15:59:43)

FormMain

### CPS SYSTEM

접속종료

메시지내용: 테스트123 전송

받은메시지: (필터링 언어를 고르세요)

이승원(사원): 테스트 (15:51:46)

이승원(사원): 테스트 (15:52:44)

이승원(사원): 테스트 (15:52:57)

이승원(사원): 테스트 (15:53:40)

구독하기

이름: 이승원

직책:

부서:

☒ 이승원(사원)명업팀

구독 검색

구독 목록보기

☐ 이승원(사원)명업팀

구독취소

FormMain

### CPS SYSTEM

접속종료

메시지내용: 로그인 한 사원번호 : 201900529 전송

받은메시지: (필터링 언어를 고르세요)

부재메시지

(메인 화면 6) 부재 메시지 - 사용자가 구독한 사람이 메시지를 보냈을 때 사용자가 접속하지 않았을 경우 부재메시지에 메시지를 남깁니다. 사용자가 접속 후 부재메시지를 확인한 후 다음 로그인 접속 시 부재메시지는 사라집니다.



## 7 부록

### 7.1 최종 테스트 결과 및 검토

#### 테스트 결과

- 로그인 회원가입 기능 정상 가동됨.
- 메시지 전송 및 sorting 기능, 수신 및 번역 기능 정상 가동됨.
- 클라이언트 구독 및 구독취소, 검색 기능 정상 가동됨.
- Session Flag 기능 미 구현으로 인해 같은 아이디로 중복 로그인을 할 수 있는 오류가 있음. (중복 로그인을 하여도 그 외 기능에는 이상이 없음.)

#### 최종 검토

- 시스템 완성 정도는 초기 계획한 Plan A와 B의 중간정도 완성하였다.  
Why - Plan B는 모두 완성하였으나 A에서의 몇 가지 기능을 구현하지 못함.
- Session flag를 사용한 중복 로그인 방지 기능 미 구현하였다.  
Why - 구현을 위한 기간이 부족 했으며 테스트 작업으로 넘어가기 위하여 구현을 보류하였음. (추가 개발 기간 확립 시 구현가능.)
- 회원가입 및 로그인 시 비밀번호를 암호화하는 기능 미 구현하였다.  
Why - 궁극적으로 시간이 부족했으며 기존의 계획은 SHA-256 방식으로 구현하려고 하였음. (추가 개발 기간 확립 시 구현가능.)

### 7.2 참고서적

- 네이버 검색 및 Googling
- C# 프로그래밍(프로그래밍 기초부터 객체지향 핵심까지) - 한빛 아카데미
- Git 분산버전 관리 시스템 - 인사이트
- UNIX SYSTEM Programming - Kay A. Robbins / 인 포 북
- UNIX Network Programming - W.RICHARD STEVENS

### 7.3 프로젝트 소감

#### - 정 진 구

CPS system 팀 프로젝트를 진행하며 C#에 대해 조금 더 깊이 있게 학습하게 되었습니다. 학교 교육과정 대부분이 웹 프로그래밍에 초점을 두고 있어 익숙하던 언어인 JAVA로 프로젝트를 진행할 수도 있었지만 C#과 JAVA의 공통점과 차이점에 대해 조금이나마 배우고 싶은 생각에 C#을 선정하게 되었습니다. 그 결과 프로그래밍을 하는데 있어 언어의 역할에 대한 시야가 조금 넓어졌고, 궁극적으로 많은 도움이 되는 결과를 낳은 것 같습니다.

또 프로젝트 주제인 TCP 소켓 통신방식에 대해 학교수업에서는 이론적인 부분을 교육받고 간단한 에코 통신을 C 언어로 실습 해본 것이 전부였습니다. 그래서 이번 프로젝트를 진행하기 위하여 개인적으로 많은 공부를 하였습니다. 학습한 내용을 토대로 하나하나 실험 및 테스트 해가며 구현해 나가며 결국 최종 구현을 하게 되었습니다. 그렇게 시스템 구현을 완성하고 프로젝트를 마무리하니 TCP 소켓 통신이 실제로 어떻게 어느 방면에서 활용되는지 조금이나마 알게 되었습니다. 짧은 기간과 개발에 있어 아직 조금 미숙한 부분 때문에 구현하지 못한 기능이 있어서 조금 아쉽지만 팀 프로젝트로 시간을 정하고 계획을 짜서 최종적인 프로젝트를 마무리한 경험이 앞으로 많은 도움이 될 것 같습니다.

#### - 이 승 원

이번 프로젝트를 통해 저는 한 단계 더 성장하였습니다. 처음 이 프로젝트를 접했을 때 어떻게 해야 할지 몰라 많이 당황했는데 인터넷 검색을 통해 하나씩 공부하고 이해했습니다 또한 프로젝트에 대해 모르는 부분이 있을 때 팀원들끼리 서로 도와서 이 프로젝트를 수행할 수 있었습니다. 처음으로 팀 프로젝트를 진행하면서 개인 프로젝트와는 달리 내가 생각하지 못하거나 이해하지 못한 부분을 같이 회의를 통해 해결하는 게 가장 좋았습니다. 4 주 동안 현장실습을 통해 TCP 소켓 통신에 대해 좀 더 자세하게 배워 갈수 있는 좋은 시간이었던 거 같습니다.

#### - 박 해 은

거의 사용하지 않았던 언어인 C#으로 개념만 알고 있는 TCP 소켓 통신을 하는 프로젝트를 해야 한다는 생각에 처음엔 막막했지만 어느덧 개발까지 끝이 났습니다. 짧은 기간이어서 많은 것을 구현하지는 못했지만, 소켓 통신에 대해서 어느 정도 이해할 수 있는 시간이 되었던 것 같습니다.

## 7.5 최종 시스템 구현 소스코드 링크(Git)

[https://github.com/jinkoooooo/CPSSystem/tree/master/Final\\_Code](https://github.com/jinkoooooo/CPSSystem/tree/master/Final_Code)

- 링크를 클릭해 들어간 후 serverForm 소스파일, CPSSystem 소스파일, exe 실행파일 등을 확인하실 수 있습니다.