

1.	프로젝트 개요
----	---------

- 프로젝트명 : 가맹점 관리 프로젝트
- 개발목적
  - 목적 1 효율적인 가맹점 관리 및 재고관리(EDI)를 목표
  - 목적 2 가맹점간의 커뮤니티 활성화
  - 목적 3 실시간 채팅을 통한 본사와의 실시간 업무 처리 극대화
- 개발기간  
2023-02-28 ~ 2023-07-28
- 개발범위 및 기능
  - 개발 범위
    - DB를 이용한 가맹점 총괄 관리
    - 본사와 가맹점간의 채팅을 이용한 통신망 형성

- 프로그램 기능
  - SQL Server를 활용한 DB
  - ASP.NET Core를 활용한 웹페이지
  - .NET Framework를 활용한 WinForm

- 개발환경

Windows 10

- SQL Server 2022
- Visual Studio 2017

2.	프로젝트 세부 내용
----	------------

기본 준수사항 : 각 가맹점,회원 등을 구분 가능하게끔 설계

Database를 활용해 정보를 공유

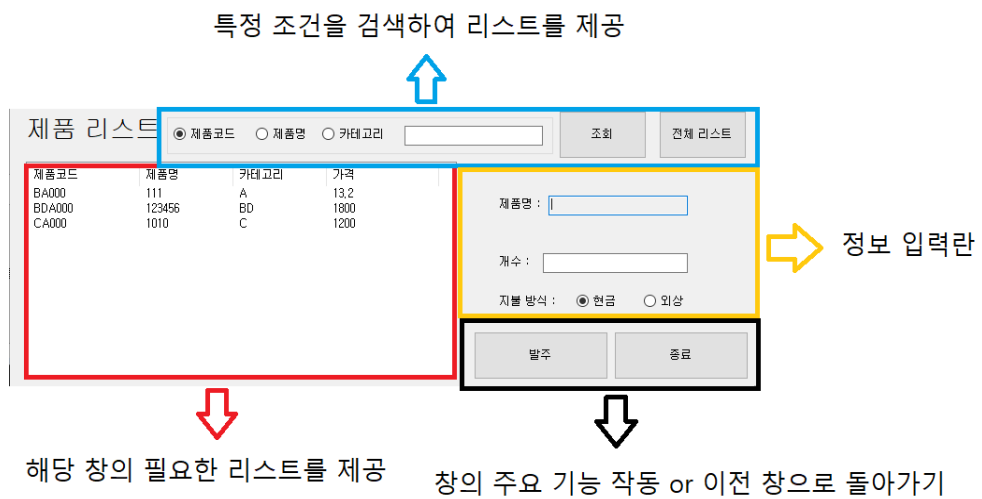
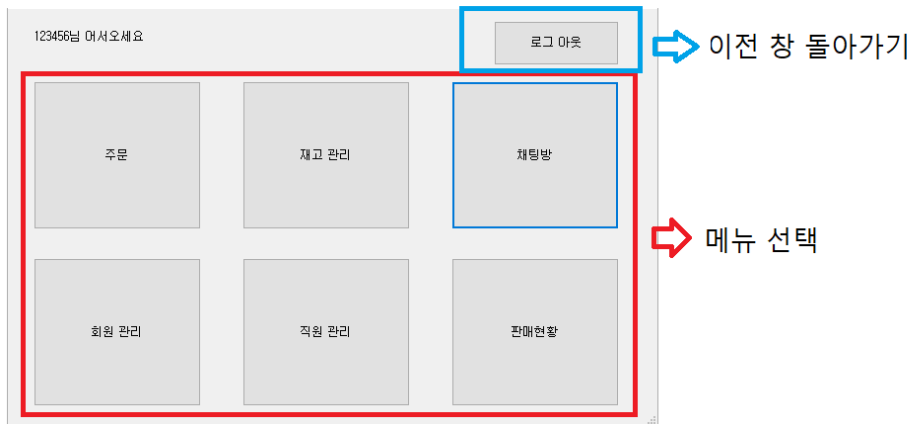
## 별도 서버를 기반한 멀티 채팅

- 개발환경

구 분	시 스템 환 경				비 고
	OS	Language	FrameWork	Database	
Franchise	Windows	C#	.NET 4.6	SQL Server 2022	
Franchise.Web	Windows	C#	ASP.NET Core 1.0	SQL Server 2022	
FranchisePOS	Windows	C#	.NET 4.6	SQL Server 2022	
Server	Windows	C#	.NET 4.6	SQL Server 2022	

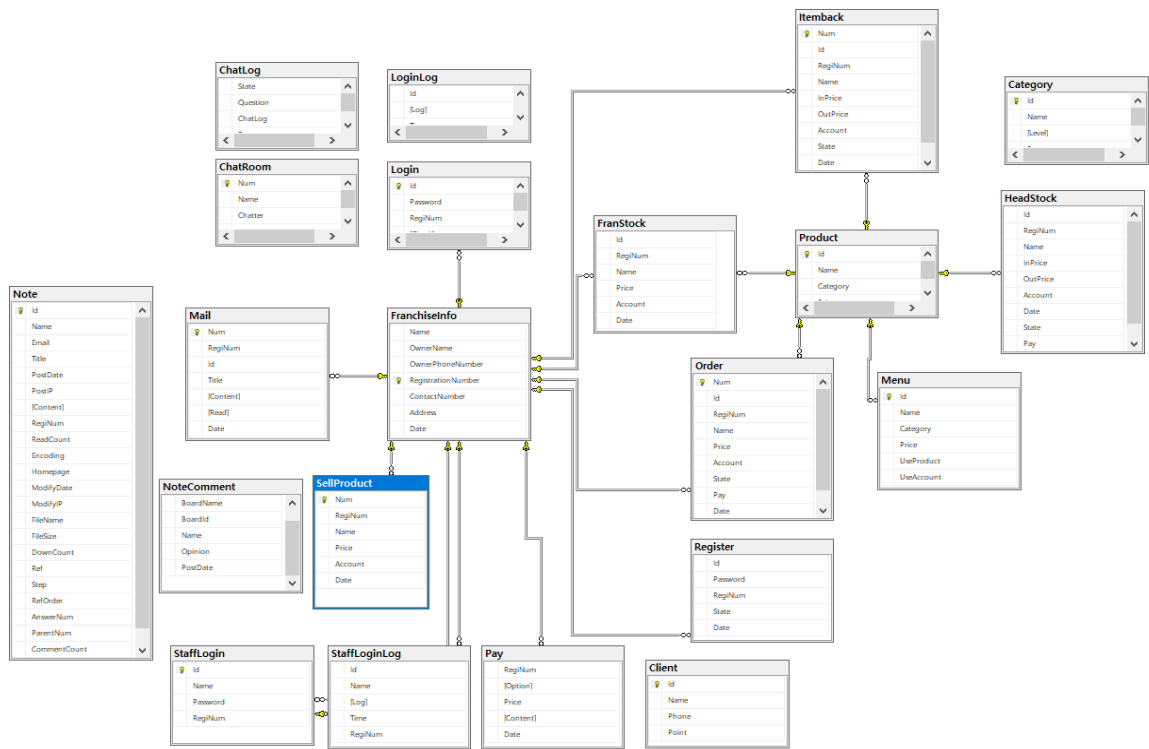
- 구축 세부 내용

- 프로젝트 구성 및 UI 디자인



- 데이터베이스 스키마 구성

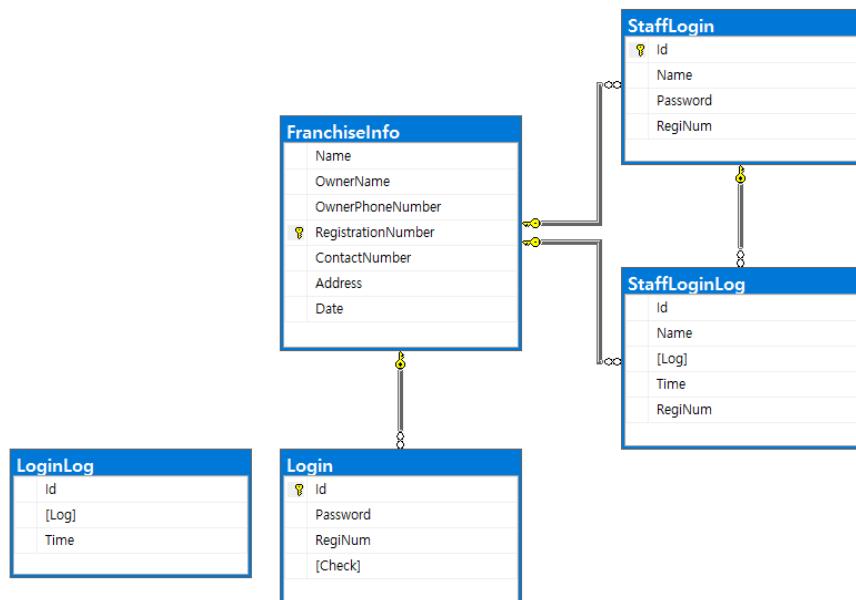
-전체적인 데이터베이스 스키마



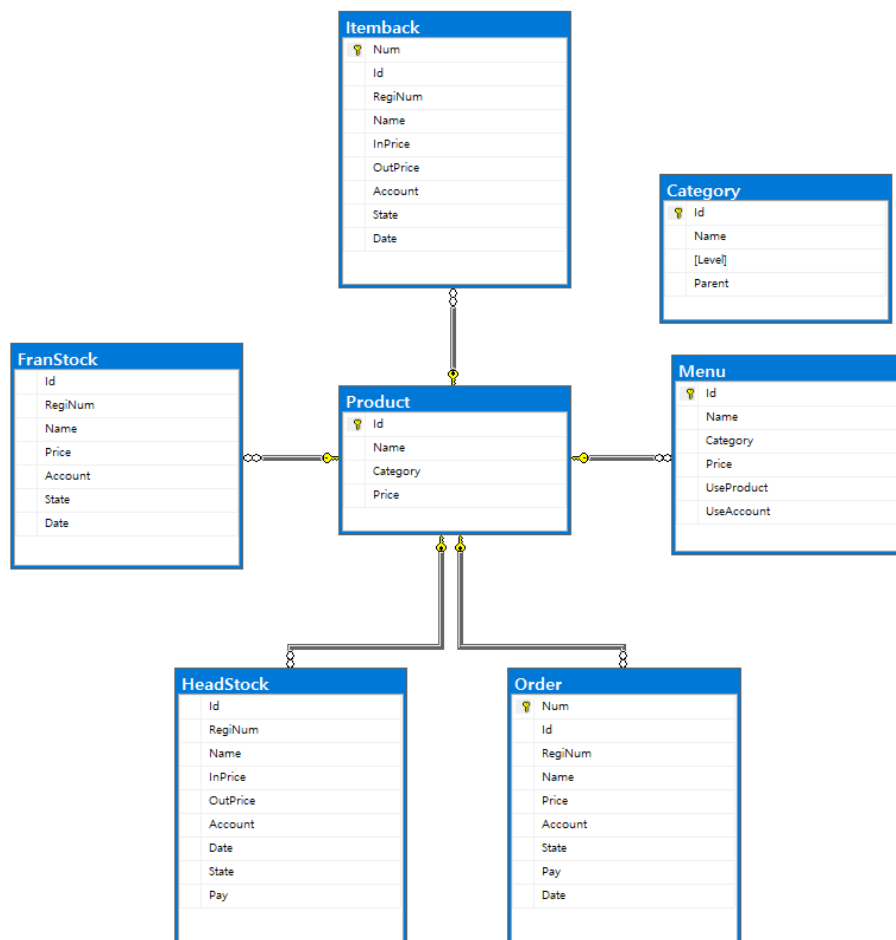
## -가맹점 관련 스키마



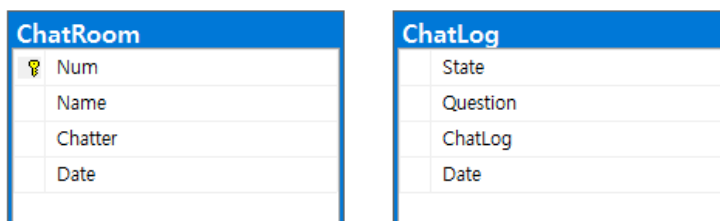
- 로그인 관련



- 상품 관련

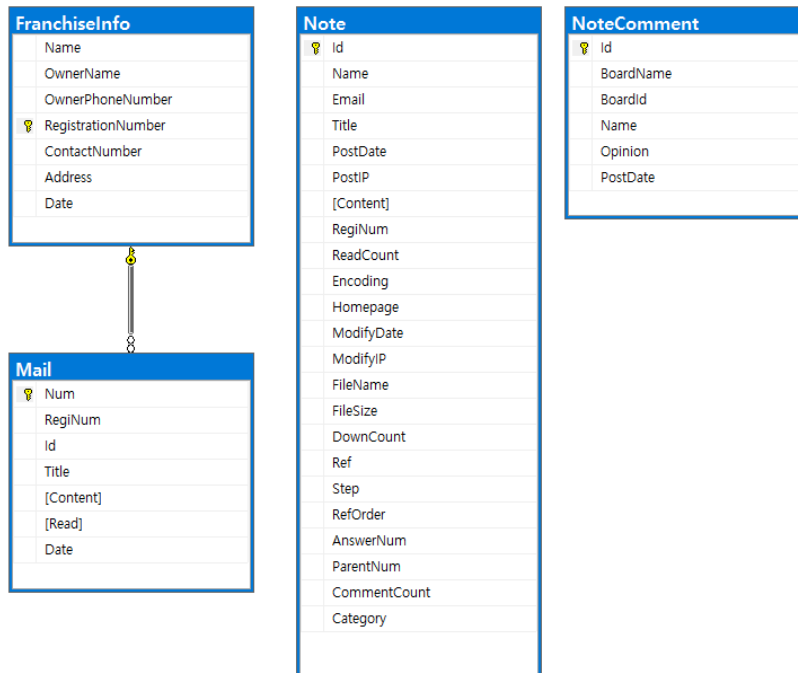


## - 채팅 관련



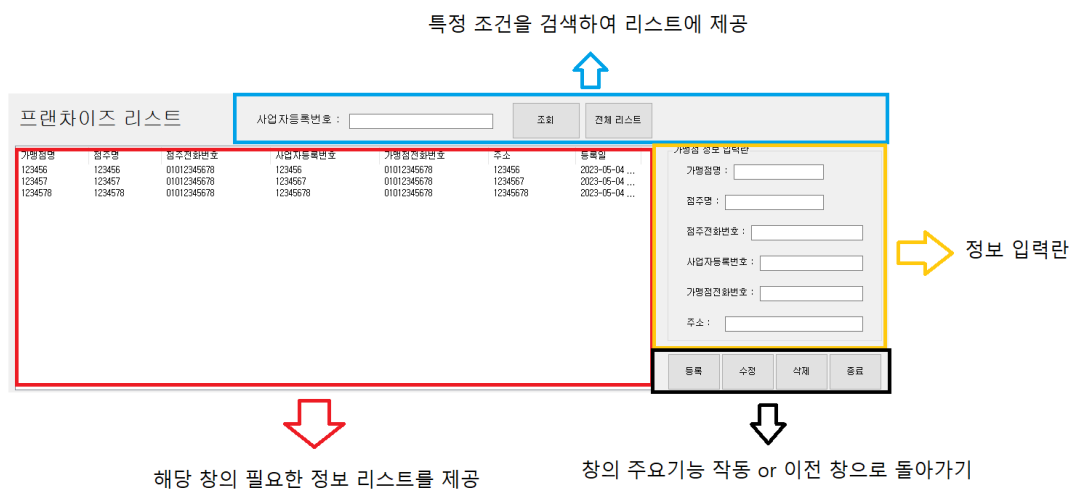


## - 웹 관련



## 관리자

## 기능



구분	내용	비고
접근방법	<ul style="list-style-type: none"> <li>- WinForm 형식을 기반으로 창을 설계</li> <li>- Button은 창을 이동하거나 각종 기능을 작동하는데 활용한다.</li> <li>- Radio Button 및 Check Box를 활용하여 필요한 특정 조건을 구분하게끔 한다.</li> <li>- ListView를 활용해 DB에서 가져온 필요한 내용을 보여준다.</li> <li>- TextBox로 필요한 정보를 입력해 반영한다.</li> </ul>	
관리내용	<ul style="list-style-type: none"> <li>- 가맹점의 사업자등록번호는 중복이 불가능하다.</li> <li>- 회원의 ID는 중복이 불가능하다.</li> </ul>	
사용자관리	<ul style="list-style-type: none"> <li>- 사용자의 사업자등록번호를 연동해 사용자의 접근을 특정할 수 있다.</li> </ul>	

• 취약점 점검항목

구분	내용	비고
제목	내용	날짜

[illegible]

• 기능적 요구사항

업무명		전체적인 프로세스	작성일	2023-06-29	
No	요구사항	요구사항설명	제약조건	우선 순위	수용 여부
1	로그인	ID/PW 입력 시 로그인 가능하게끔 한다.	ID 중복 불가		O
2	가맹점 등록	가맹점에 관련된 정보를 입력하여 등록 임의 로그인을 사업자등록번호(10자리 )와 전화번호로 지정	사업자등록번호 중복 불가		O
3	제품 생성	가맹점에게 보낼 제품의 정보를 입력하 여 등록 제품 코드는 카테고리Id + 위치 + 일련 번호로 자동 생성 하게끔 한다.	제품 코드 중복 불가		O
4	회원 인증	웹페이지에서 가맹점이 인증 요청을 보 내고 본사에서 인증 허용한다면 설정한 ID와 임의 비밀번호를 해당 가 맹점 메일에 전송			O
5	카테고리 생성	카테고리를 대중소로 분류함	카테고리 ID 중복 불가		O
6	반품 생성	가맹점에서 반품을 신청하고 본사가 허 용한다면 반품을 한다. 반품 사유에는 크게 반품/불량반품으로 나뉘 불량반품 시 제품을 폐기한다.	반품번호 임의 생성		O

7	채팅 시스템	1:1 문의 이외의 채팅은 기록되지 않으며 본인(가맹점)이 만든 채팅방은 본인이 해당 방을 나간다면 방이 사라지게끔 한다.	채팅방 식별 번호 부여		○
8	매출 조회	전체 매출 뿐만 아니라 가맹점별 매출 또한 반품률 등의 기능으로 매출 외의 경우도 보여주게끔 한다.			○
9	주문	가맹점이 제품들을 요청하면 그걸 본사에서 허용하면 가맹점의 재고에 들어오게끔 한다.	주문 번호		○
10	상품 구매	고객이 가맹점에게 상품을 구매할 때 카드/현금 결제 방법을 선택 후 회원의 핸드폰번호로 포인트 적립 및 사용을 이용하게끔 한다.			○


• 프로세스 정의서

작 성 일	2023-06-15		수 정 일		
기 능 명	재고량을 실시간으로 갱신 하는 기능		메서드명	DataLoad() : MainMenu.cs	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도

작 성 일	2023-05-09		수 정 일		
기 능 명	Id 자동 생성 기능		메서드명	AutoCreateId(string RegiNum) WebRegist.cs	
프로세스ID	프로세스명	프로세스설명	관련엔티티	수행 빈도	
string	RegiNum	가맹점을 특정하는 정보			

작 성 일	2023-05-10		수 정 일		
기 능 명	카테고리에 등록된 상품이 있는지 파악하는 기능		메서드명	DeleteCheck() Category.cs	
프로세스ID	프로세스명	프로세스설명	관련엔티티	수행 빈도	

작 성 일	2023-04-04		수 정 일		
기 능 명	제품코드를 자동생성 하는 기능 (ex.카테고리Id+위치+ 일련번호)		메서드명	AutoCode(string CategoryCode) Product.cs	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도
string	CategoryCode	카테고리를 특정하는 코드			

작 성 일	2023-05-15		수 정 일		
기 능 명	탭이름과 같은 곳에 제품들 을 담는 기능		메서드명	AddList(string regiNum) Release.cs	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도
string	regiNum				



작 성 일	2023-05-15		수 정 일		
기 능 명	탭을 만들고 해당 탭에 설정한 상품을 담는 기능		메서드명	AddTabPage(ListViewItem listview) Release.cs	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도
ListViewItem	listview				

작 성 일	2023-06-16		수 정 일		
기 능 명	라디오 버튼 체크박스 버튼 클릭에 따라 알맞은 지리어를 담는 기능		메서드명	QuerySet()	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도

--	--	--	--	--

작 성 일	2023-06-17		수 정 일		
기 능 명	판매 TOP 5 상품을 실시간 으로 보여주는 기능		메서드명	Top5Load()	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도

작 성 일	2023-04-26		수 정 일	2023-05-22 ~ 2023-06-05	
기 능 명	서버에게 메시지 혹은 특정 정보를 받는 기능		메서드명	Receive(TcpClient client) ClientChat.cs	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도
TcpClient	client				


작 성 일	2023-04-27		수 정 일		
기 능 명	입력한 메시지와 특정 정보를 서버에 보내는 기능		메서드명	MsgSend() ClientChat.cs	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도

작 성 일	2023-05-22		수 정 일		
기 능 명	서버에 접속하는 기능		메서드명	ServerConnect() ClientChat.cs	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도


작 성 일	2023-05-22		수 정 일		
기 능 명	채팅방 DB에 저장되어 있는 해당 클라이언트가 속한 방을 불러와 화면에 나타내는 기능		메서드명	DataLoad() ClientChat.cs	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도

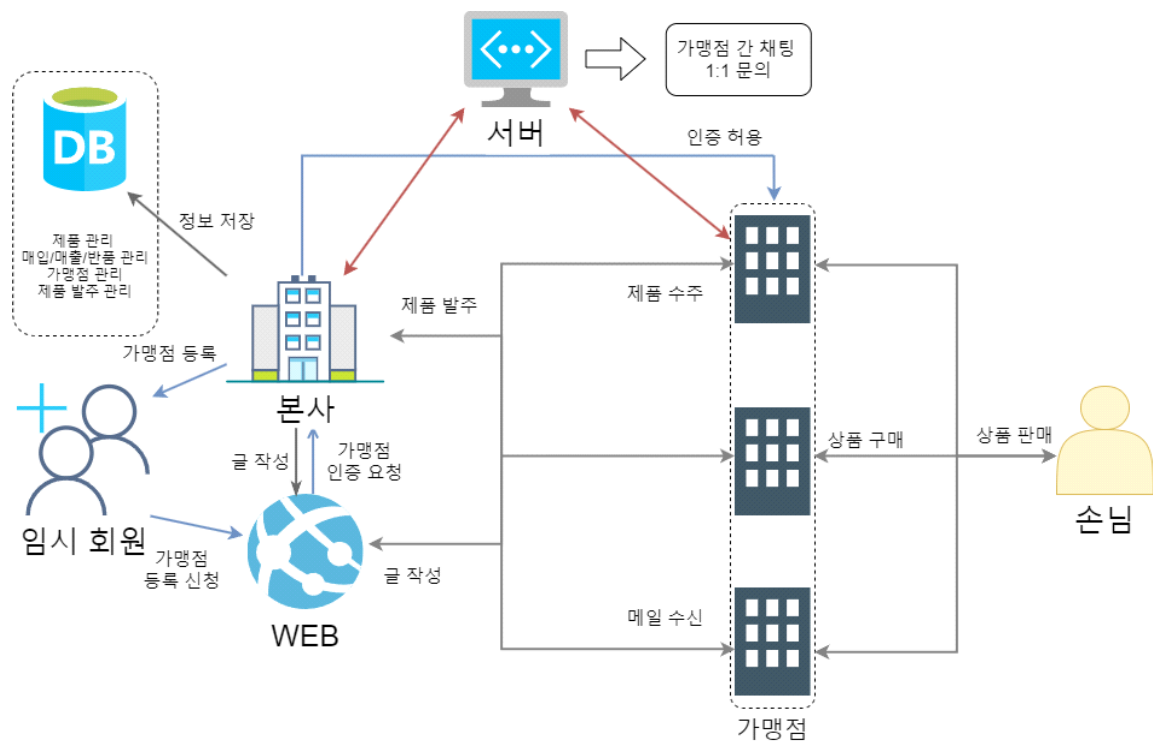
작 성 일	2023-04-26		수 정 일	2023-05-22 ~ 2023-06-05	
기 능 명	서버를 실행하여 클라이언트의 접속을 받는 기능		메서드명	ServerStart() ServerConnect.cs	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도


작 성 일	2023-04-27		수 정 일	2023-05-22 ~ 2023-06-05	
기 능 명	클라이언트의 메시지를 받아 관련 기능을 작동시키는 함수		메서드명	Run(Client client) ServerConnect.cs	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도
Client	client				

작 성 일	2023-05-28		수 정 일	2023-05-28 ~ 2023-06-05	
기 능 명	클라이언트가 만든 채팅방에 들어왔을때 작동하는 스레드 함수		메서드명	RoomChat(Client client)  ServerConnect.cs	
프로세스ID	프로세스명	프로세스설명		관련엔티티	수행 빈도
Client	client				


- 시스템구성도

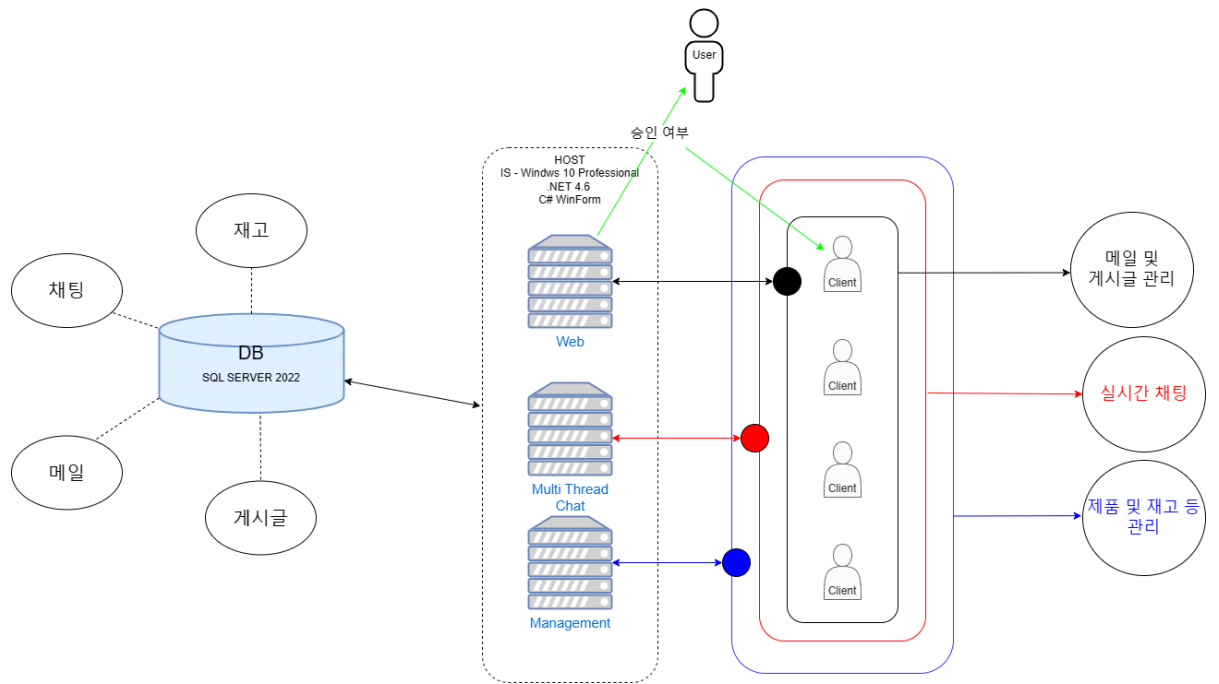
- 시스템 개념도



가맹점 화면 / 본사 화면 주요 업무 흐름(다이어그램)

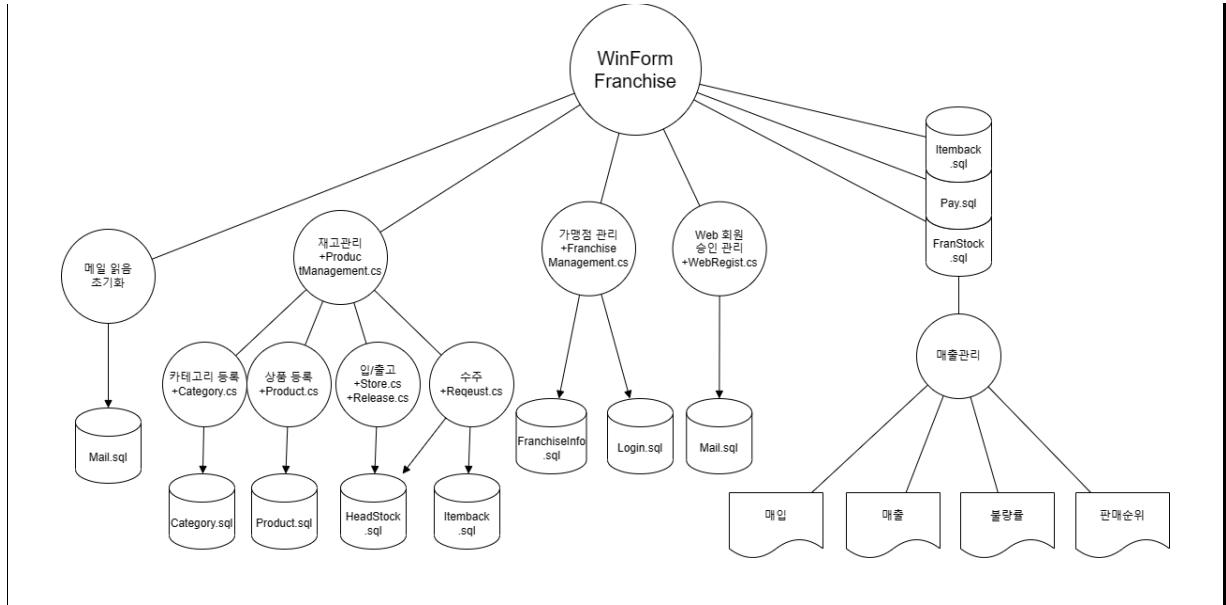
발주 -> Web,Offline -> 작동순서 -> DB 테이블 어디에 저장되는지? -> 사용  
되는 무언가?

- 시스템구성도



- 클래스 설계서(기본)

클래스 다이어그램(기본)			
서브시스템명	Franchise	작성일	



# • 클래스 명세(기본)

클래스 명세					
서브시스템  명  클래스명	Franchise		작성일		2023-07-06
	Category.cs				
	메소드 설명				
메소드명	DataLoad	리턴  타입	void	파라  미터	
메소드명		설명	DB의 Category 정보를 가져와 리스트뷰에 담는 기능		
메소드명	btnCreate_Click	리턴  타입	void	파라  미터	object sender , EventArgs e
		설명	생성 버튼 클릭시 카테고리를 생성하는 기능		



메소드명	btnModify_Click	리턴 타입 설명	void  수정 버튼 클릭시 해당하는 카테고리 정보를 수정하는 기능	파라 미터	object sender, EventArgs e
메소드명	btnDelete_Click	리턴 타입	void	파라 미터	object sender, EventArgs e
메소드명	RemovePlaceholder	설명	삭제 버튼 클릭시 해당하는 카테고리를 삭제하는 기능		
메소드명	SetPlaceholder	리턴 타입	void	파라 미터	object sender, EventArgs e
메소드명	SetPlaceholder	설명	텍스트박스 내용이 사용자가 입력한 값이 아닌 Placeholder일 경우에만, 커서 포커스일때 빈칸으로 만들기		
메소드명	CategoryTree_AfterSelect	리턴 타입	void	파라 미터	object sender, TreeViewEventArgs e
메소드명	CategoryTree_AfterSelect	설명	사용자 입력값이 하나도 없는 경우에 포커스 잃으면 Placeholder 적용해주기		
메소드명	DeleteCheck	리턴 타입	bool	파라 미터	
메소드명	DeleteCheck	설명	카테고리 삭제 시 해당 카테고리에 포함된 상품이 있는지 파악하는 기능		

클래스 명세

서비스시스템	Franchise		작성일		2023-07-06
명					
클래스명	Product.cs				
메소드 설명					
메소드명	CategoryLoad	리턴 타입	void	파라 미터	
메소드명	DataLoad	설명 리턴 타입 설명	카테 고리 정보를 가져와 텍스트박스에 적용하는 기능  void  파라 미터  제 품 정보를 DB에서 가져와서 리스트뷰에 담는 기능		
메소드명	SetCategoryName	리턴 타입	void	파라 미터	
메소드명	btnCreate_Click	설명 리턴 타입 설명	카테 고리 코드에 맞는 카테고리 이름을 리스트 뷰에 보여주는 기능  void  파라 미터  object sender , EventArgs		
메소드명	AutoCode	리턴 타입	void	파라 미터	string CategoryCode
메소드명	btnModify_Click	설명 리턴	void	파라	제 품코드를 자동생성 하는 기능 (ex. 제 품코드 =카테 고리 Id+위치+일련번호)  object sender , EventArgs

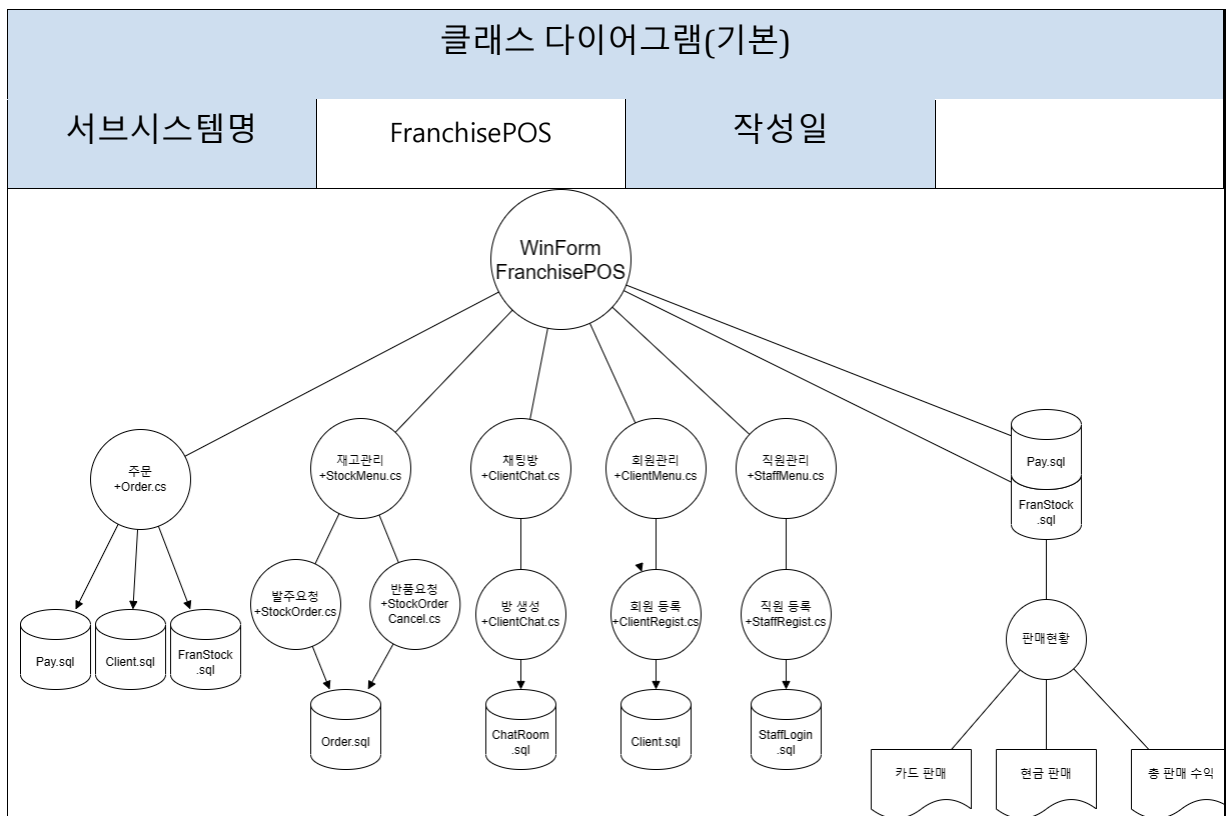
메소드명	btnDelete_Click	타입	미터	
		설명	수정버튼 클릭 시 제품 정보를 수정하는 기능	
		리턴	void	파라미터
		타입		object sender , EventArgs e
		설명	삭제버튼 클릭 시 제품 정보를 삭제하는 기능	
메소드명	CategoryTree_AfterSelect_1	리턴	void	파라미터
		타입		object sender , TreeViewEventArgs e
		설명	트리뷰에서 선택한 노드의 값을 가져와 이름과 코드명을 분리하는 기능	

클래스 명세				
서브시스템명	Franchise	작성일		2023-07-06
명				
클래스명	Release.cs			
메소드 설명				
메소드명	DataLoad	리턴	void	파라미터
		타입		미터
메소드명	FranDataLoad	설명	DB의 재고 정보를 가져와 리스트 뷰에 담는 기능	
		리턴	void	메소드명
		타입		
		설명	DB에서 가맹점 정보를 리스트뷰에 담는 기능	

메소드명	SetCategoryName	리턴 타입 설명	void  카테고리 코드에 맞는 카테고리 이름을 리스트 뷰에 보여주는 기능	파라 미터	
메소드명	btnOut_Click_Click	리턴 타입 설명	void  출고 버튼 클릭 시 출고할 상품을 보낼 가맹점 을 선택하는 창으로 이동하는 기능	파라 미터	object sender, EventArgs e
메소드명	getAccount	리턴 타입 설명	int  각 상품코드의 재고량을 가져오는 기능	파라 미터	string code
메소드명	FranListView _ColumnClick	리턴 타입 설명	void  리스트뷰 머리글 클릭 시 체크박스 전체 선택/ 해제 기능	파라 미터	object sender, ColumnClickEventArgs e
메소드명	FranListView _DrawColumnHeader	리턴 타입 설명	void  리스트뷰 머리글을 그리는 기능(체크박스)	파라 미터	object sender, DrawListViewColumnHeaderEventArgs e
메소드명	FranListView _DrawItem	리턴 타입 설명	void  리스트뷰 항목 그리는 기능	파라 미터	object sender, DrawListViewItemEventArgs e
메소드명	FranListView _DrawSubItem	리턴 타입	void	파라 미터	Object sender, DrawListViewSubItemEventArgs e

		설명	리스트뷰 하위항목 그리는 기능		
메소드명	ProductListView_MouseClick	리턴	void	파라미터	object sender , MouseEventArgs e
		타입			
		설명	수량의 하위항목 클릭 시 개수 조절이 가능하게 하는 기능		
메소드명	AccountSpin_KeyDown	리턴	void	파라미터	object sender , KeyEventArgs e
		타입			
		설명	특정 키 클릭 시 해당 수량 적용하는 기능		
메소드명	AccountSpin_KeyDown	리턴	void	파라미터	object sender , KeyEventArgs e
		타입			
		설명	특정 키 클릭 시 해당 수량 적용하는 기능		
메소드명	AccountSpin_Leave	리턴	void	파라미터	object sender , EventArgs e
		타입			
		설명	AccountSpin을 벗어날때 발생하는 기능		
메소드명	AddList	리턴	void	파라미터	string regiNum
		타입			
		설명	탭이름과 같은 곳에 제품들을 담는 기능		
메소드명	pageCheck	리턴	bool	파라미터	string regiNum
		타입			
		설명	탭 이름 중복을 체크하는 기능		
메소드명	AddTapPage	리턴	void	파라	ListViewItem listview

	타입	미터
	설명	탭을 만들고 해당 탭에 설정한 상품을 담는 기능



## 클래스 명세

서브시스템  명  클래스명	FranchisePOS	작성일
	StockOrder.cs	
메소드 설명		
메소드명	DataLoad	리턴 타입
		void
		파라 미터
메소드명	ProductListView _SelectedIndexChanged	설명 리턴 타입 설명
		DB에서 제품 정보를 가져와 리스트뷰에 담는 기능
		void
		파라 미터
		object sender , EventArgs e
		리스트뷰의 인덱스 선택 시 해당 인덱스의 정 보들을 가져오는 기능
메소드명	btnOrder_Click	리턴 타입
		void
		파라 미터
		object sender , EventArgs e
메소드명	btnSearch_Click	설명 리턴 타입 설명
		발주 버튼 클릭시 입력된 수량과 제품정보를 바탕으로 본사에 주문하는 기능
		void
		파라 미터
		object sender , EventArgs e
		조회 버튼 클릭 시 조건에 맞는 제품정보를 DB에서 가져와 리스트뷰에서 보여주는 기능
메소드명	StockOrder_FormClosing	리턴 타입
		void
		파라 미터
		object sender , FormClosing EventArgs e
		설명
		현재 창 닫을 때 DB연결을 끊는 기능
메소드명	rbCash_CheckedChanged	리턴
		void
		파라
		object sender , EventArgs e

메소드명	btnReset_Click	타입	미터
		설명	라디오 버튼 변경 때마다 지불방식 변경
		리턴	void
		타입	object sender, EventArgs e
메소드명	getRegiNum	리턴	string
		타입	string
		설명	DB에서 해당 가맹점의 사업자등록번호를 가져 오는 기능
		리턴	string

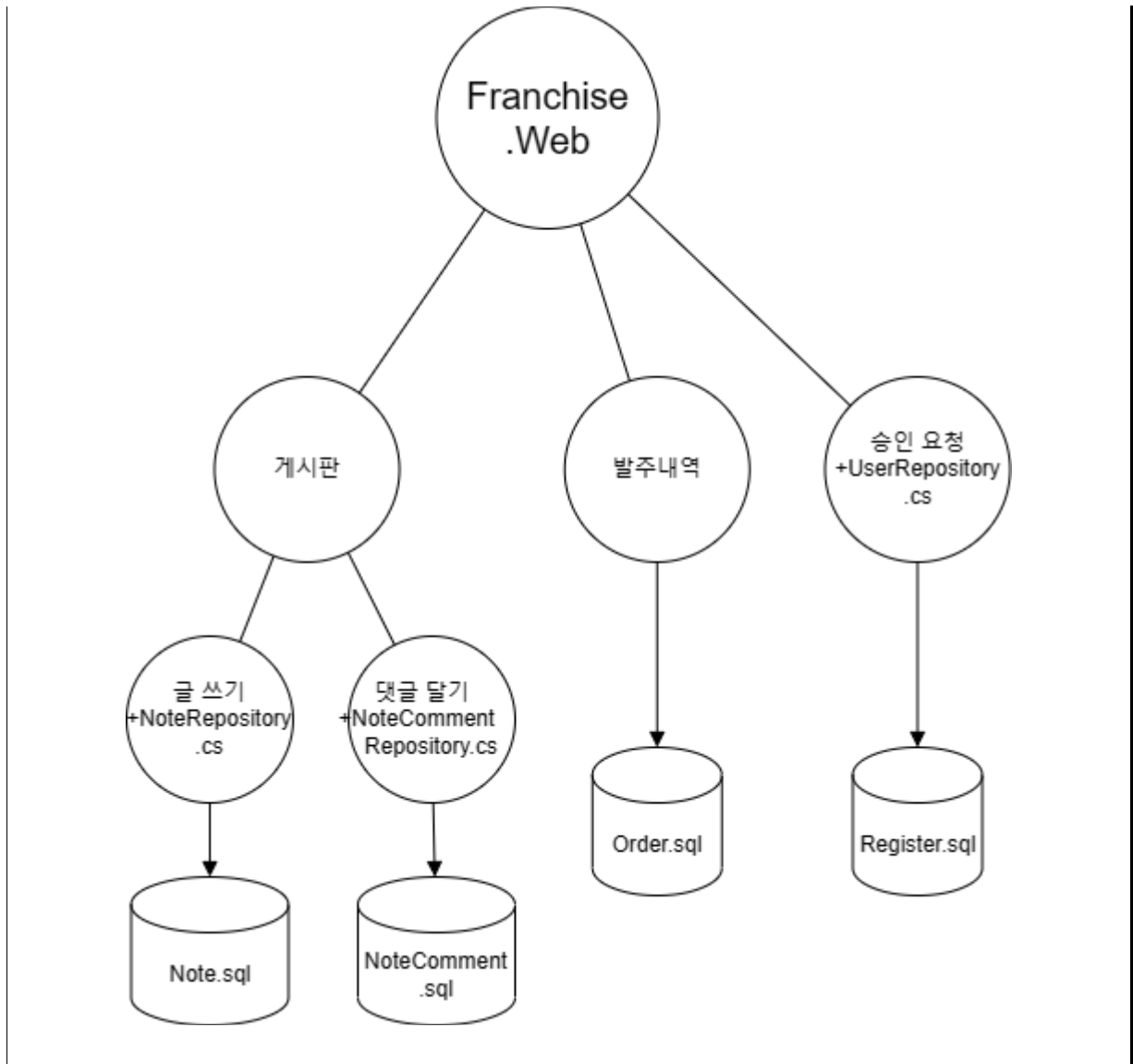
클래스 명세					
서브시스템  명	FranchisePOS	작성일			
클래스명	ClientChat.cs				
메소드 설명					
메소드명	Receive	리턴	void	파라	TcpClient client
		타입		미터	
메소드명	MessageView	설명	서버에게 메시지를 받는 기능		
		리턴	void	파라	string strText
		타입		미터	
		설명	채팅한 기록을 띄워주는 기능		



메소드명	MsgSend	리턴 타입 설명	void  입력한 메시지를 서버에 보내는 기능	파라 미터	
메소드명	ServerStop	리턴 타입 설명	void  Tcp 관련 기능 및 스레드 중지 기능	파라 미터	
메소드명	txtMessage_KeyPress	리턴 타입 설명	void  엔터키 클릭시 메시지 보내는 기능을 작동	파라 미터	object sender , KeyPressEventArgs e
메소드명	btnBack_Click	리턴 타입 설명	void  방을 나가는 기능	파라 미터	object sender , EventArgs e
메소드명	btn_ChatClick	리턴 타입 설명	void  리스트에 있는 채팅방을 클릭 시 방에 입장하는 기능	파라 미터	object sender , EventArgs e
메소드명	ServerConnect	리턴 타입 설명	void  서버에 접속하는 기능	파라 미터	
메소드명	DataLoad	리턴 타입	void	파라 미터	

		설명	채팅방 DB에 저장되어 있는 방을 불러와 화면에 나타내는 기능		
메소드명	btnCorS_Click	리턴 타입	void	파라미터	<code>object</code> sender , <code>EventArgs</code>
		설명	특정 상황에 맞게 방을 생성하는 버튼 혹은 방을 찾는 버튼으로 작동하는 기능		
메소드명	SearchRoom	리턴 타입	void	파라미터	
		설명	검색한 방을 보여주는 기능		
메소드명	CreateRoom	리턴 타입	void	파라미터	
		설명	채팅방 생성 시 채팅방(버튼)을 만들고 보이게 하는 기능		
메소드명	GetNameByRegiNum	리턴 타입	<code>string</code>	파라미터	<code>string</code> regiNum
		설명	가맹점명을 받아오는 기능		

클래스 다이어그램(기본)			
서비스시스템명	Franchise.Web	작성일	



클래스 명세				
서브시스템명	Franchise.Web	작성일		
클래스명	NoteRepository.cs			
메소드 설명				
메소드명	SaveOrUpdate	리턴	int	파라 NoteBoardWriteFormType formType n,

메소드명	Add	타입	미터			
		설명	데이터저장, 수정, 답변 공통 메서드			
		리턴	void	파라	Note vm	
		타입		미터		
		설명		게시판 글쓰기		
메소드명	UpdateNote	리턴	int	파라	Note vm	
		타입		미터		
		설명	수정 하기			
메소드명	ReplyNote	리턴	void	파라	Note vm	
		타입		미터		
		설명	답변 글쓰기			
메소드명	GetAll	리턴	List<Note>	파라	int page	
		타입		미터		
		설명	게시판 리스트			
메소드명	GetCountBySearch	리턴	int	파라	String searchField, string searchQuery	
		타입		미터		
		설명	검색 카운트			
메소드명	GetCountAll	리턴	int	파라		
		타입		미터		
		설명	Notes 테이블의 모든 레코드 수			

메소드명	GetFileNameById	리턴 타입 설명	<code>string</code>  Id에 해당하는 파일명 반환	파라 미터	<code>int</code> id
메소드명	GetSearchAll	리턴 타입 설명	<code>List&lt;Note&gt;</code>  검색 결과 리스트	파라 미터	<code>int</code> page, <code>string</code> searchField, <code>string</code> searchQuery
메소드명	UpdateDownCount	리턴 타입 설명	<code>void</code>  다운 카운트 1 증가	파라 미터	<code>string</code> fileName
메소드명	UpdateDownCountById	리턴 타입 설명	<code>void</code>  다운 카운트 1 증가	파라 미터	<code>int</code> id
메소드명	GetNoteById	리턴 타입 설명	<code>Note</code>  게시판 상세보기	파라 미터	<code>int</code> id
메소드명	DeleteNote	리턴 타입 설명	<code>int</code>  게시판 삭제	파라 미터	<code>int</code> id
메소드명	GetNewPhotos	리턴 타입	<code>List&lt;Note&gt;</code>	파라 미터	

		설명	최근 올라온 사진 리스트 4개 출력		
메소드명	GetNoteSummary ByCategory	리턴 타입	List<Note>	파라 미터	string category
		설명	최근 글 리스트		
메소드명	GetRecentPosts	리턴 타입	List<Note>	파라 미터	int id
		설명	최근 글 리스트 전체(최근 글 5개 리스트)		
메소드명	GetRecentPosts	리턴 타입	List<Note>	파라 미터	int numberOfNotes
		설명	최근 글 리스트 n개		
메소드명	Pinned	리턴 타입	void	파라 미터	int id
		설명	특정 게시물을 공지사항(NOTICE)으로 올리기		

클래스 명세					
서비스 템명	Franchise.Web	작성일			
클래스명	NoteCommentRepository.cs				
메소드 설명					
메소드명	AddNoteComment	리턴	void	파	NoteComment model

메소드명	GetNoteComments	타입	라미터
		설명	특정 게시물에 댓글 추가
메소드명	GetCountBy	리턴 타입	파라미터
		설명	특정 게시물에 해당하는 댓글 리스트
메소드명	DeleteNoteComment	리턴 타입	파라미터
		설명	특정 게시물의 특정 Id에 해당하는 댓글 카운트
메소드명	GetRecentComments	리턴 타입	파라미터
		설명	댓글 삭제

	타입	라미터
	설명	최근 댓글 리스트 전체

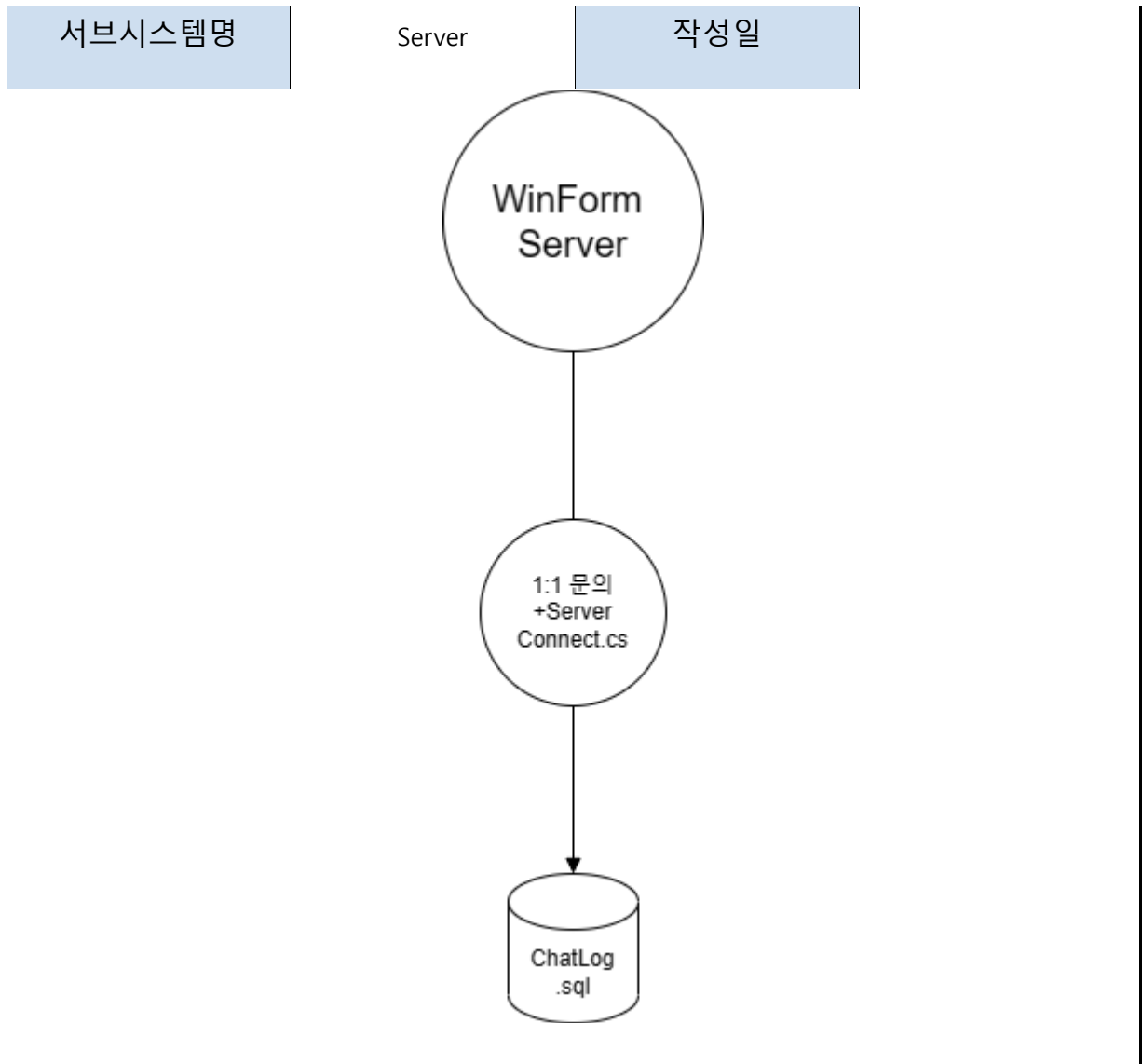
클래스 명세					
서비스 템명 클래스명	Franchise.Web		작성일		
	UserRepository.cs				
	메소드 설명				
메소드명	GetUserByRegiNum	리턴 타입	UserViewModel	파 라 미 터	string RegiNum
		설명	사업자등록번호를 이용한 특정 가맹점의 회원 정보를 얻는 기능		
메소드명	ModifyUser	리턴 타입	void	파 라 미 터	string password, string RegiNum
		설명	회원 정보를 수정하는 기능		



메소드명	IsCorrectUser	리턴 타입	bool	파라미터	string Id, string password
		설명	아이디와 암호가 해당 회원의 정보와 동일한지 아닌지를 확인하는 기능		
메소드명	Register	리턴 타입	void	파라미터	string Id, string password, string RegiNum
		설명	회원가입 요청 시 Register DB에 그 정보를 담는 기능		
메소드명	IsCheck	리턴 타입	bool	파라미터	string Id
		설명	임시 로그인인지 아닌지를 판별하는 기능		
	GetRegiNum	리턴 타입	string	파라미터	string Id
		설명	Id에 해당하는 사업자등록번호 정보를 가져오는 기능		

메소드명	Regist	리턴 타입	void	파라미터	string RegiNum
		설명	회원승인 요청한 ID와 PW로 로그인정보를 적용하는 기능		
메소드명	Log	리턴 타입	void	파라미터	string Id,string Log
		설명	로그인 기록을 저장하거나 수정하는 기능		
메소드명	MailNotice	리턴 타입	bool	파라미터	string Id
		설명	메일을 읽었는지 확인하는 기능		

클래스 다이어그램(기본)



클래스 명세					
서브시스템  명  클래스명	Server		작성일		
	ServerConnect.cs				
메소드 설명					
메소드명	MessageView	리턴	void	파라	string strText,string color

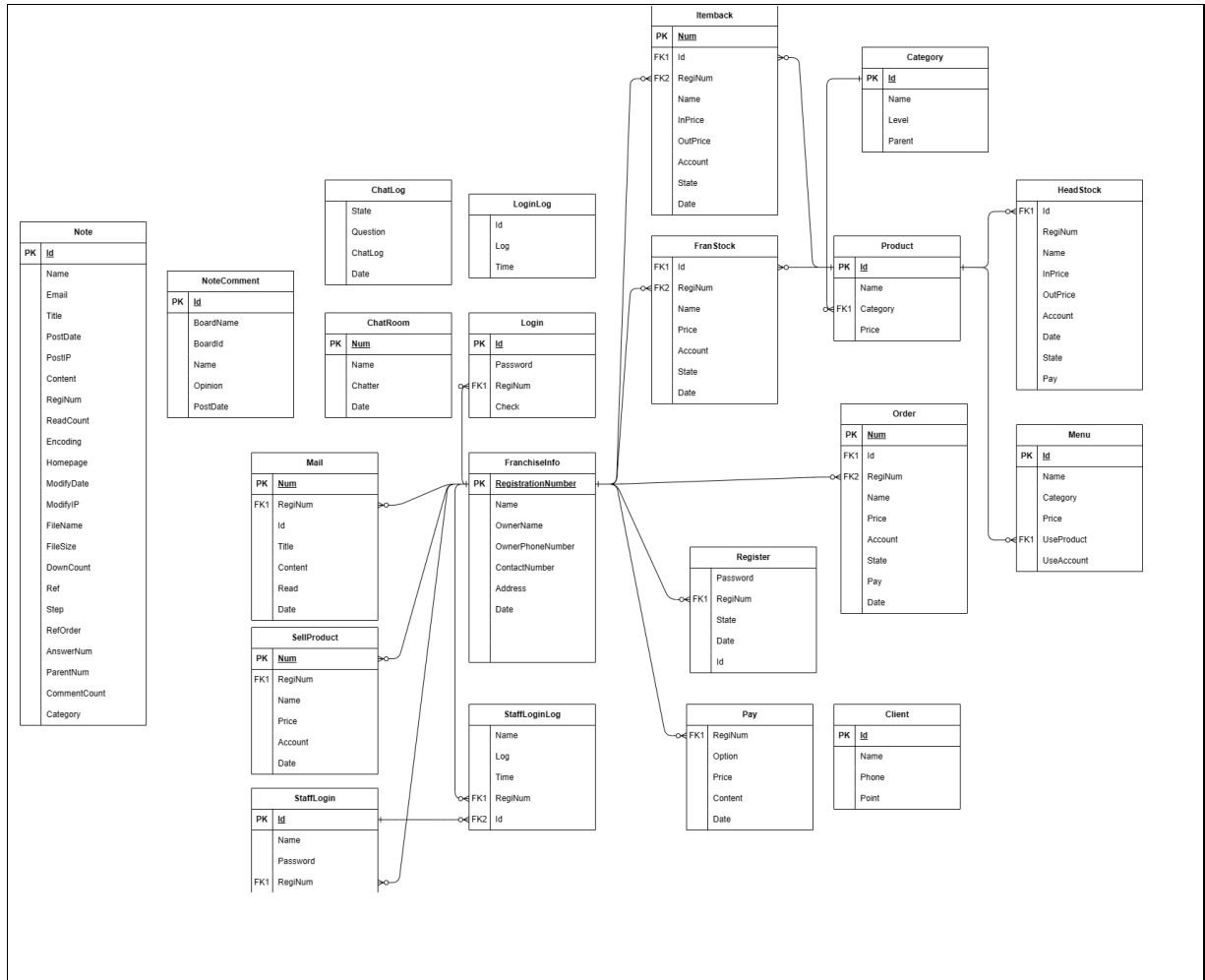
메소드명	ServerStart	타입	미터		
		설명	채팅창에 채팅 기록을 띄워주는 기능		
		리턴	void	파라	
		타입		미터	
		설명		서버를 실행하여 클라이언트의 접속을 받는 기능	
메소드명	Run	리턴	void	파라	Client client
		타입		미터	
메소드명	RoomChat	설명	클라이언트의 메시지를 받아 관련 기능을 작동시키는 함수		
		리턴	void	파라	Client client
		타입		미터	
		설명		클라이언트가 만든 채팅방에 들어왔을때 작동하는 스레드함수	
메소드명	ServerStop	리턴	void	파라	
		타입		미터	
		설명		채팅창 종료 시 서버 및 스레드 기능을 중단	
메소드명	ChatterAdd	리턴	void	파라	string Chatter , string Num
		타입		미터	
메소드명	GetNameByRegiNum	설명	채팅방 정보를 DB에 담는 기능		
		리턴	string	파라	string RegiNum
		타입		미터	
		설명		가맹점 이름을 받아오는 기능	

메소드명	MsgSend	리턴 타입	void	파라미터	
		설명	메시지를 클라이언트에게 보내는 기능		
메소드명	CreateRoom	리턴 타입	void	파라미터	string RegiNum,string name,int RoomNum
		설명	방 정보를 DB에 담는 기능		
메소드명	DeleteChatRoom	리턴 타입	void	파라미터	
		설명	모든 방 정보를 DB에서 제거하는 기능		
메소드명	IsHost	리턴 타입	bool	파라미터	string RegiNum,string RoomNum
		설명	방의 호스트를 찾는 기능		
메소드명	ExitRoom	리턴 타입	void	파라미터	string RoomNum
		설명	특정 방 정보를 DB에서 제거하는 기능		
메소드명	DataLoad	리턴 타입	void	파라미터	
		설명	DB에서 저장된 '답변' 관련 채팅로그를 불러오는 기능		
메소드명	ChattingAdd	리턴 타입	void	파라미터	string state,string Question,string ChatLog

## 설명

채팅로그 DB에 내용을 저장

- 데이터모형설계(ERD) : 테이블 간의 모형설계



- 화면설계정의

- 메인화면

메뉴명(경로)

ClientMenu

화면명

기본 메뉴 화면

화면설명

1. 버튼식으로 이용할 메뉴 리스트를 출력

	2. 종료 및 이전 창으로 돌아가는 버튼 3. 현재 창에 대한 정보를 표기		
관련요구사항	가시성 우선	관련기능	

• 서브화면

메뉴명(경로)	ClientDelete	화면명	세부화면
---------	--------------	-----	------

화면설명	1. 해당 창에 관련된 정보를 리스트에 출력		
	2. 현재 창에 대한 정보를 표기		
관련요구사항	3. 조건에 맞는 정보를 검색		
	4. 리스트 정보를 취합해 정리해서 출력		
	5. 기능 버튼 및 종료 버튼		
	조건 세분화	관련기능	라디오버튼

The screenshot shows a window titled '회원 삭제' (Member Deletion). It contains a search bar with radio buttons for '이름' (Name) and '핸드폰 번호' (Phone Number), a search button, and a '전체 리스트' (Full List) button. Below the search bar is a table with columns '이름', '핸드폰 번호', and '포인트'. A large empty area is below the table. At the bottom right, there are two buttons: '삭제' (Delete) and '종료' (End).

Annotations on the screenshot:

- 1: Red circle around the table area.
- 2: Green circle around the window title bar.
- 3: Purple circle around the search bar.
- 4: Orange circle around the large empty area.
- 5: Blue circle around the '삭제' and '종료' buttons.

• 테이블목록

순번	테이블 ID	테이블 설명
1	Cateogry	상품 분류하는 카테고리 기록
2	ChatLog	채팅 기록을 저장
3	ChatRoom	채팅방 정보를 기록



4	Client	회원(고객) 정보를 기록
5	FranchiseInfo	가맹점 정보를 기록
6	FranStock	가맹점 재고를 기록
7	HeadStock	본사의 재고를 기록
8	Itemback	반품/불량품을 기록
9	Login	로그인 정보를 기록
10	LoginLog	로그인 로그를 기록
11	Mail	메일 정보를 기록
12	Menu	판매 상품 정보를 기록
13	Note	게시판 글 정보를 기록
14	NoteComment	게시판의 댓글 정보를 기록
15	Order	본사-가맹점간 주문(요청)을 기록
16	Pay	고객이 상품을 결제한 정보를 기록
17	Product	본사가 유통할 제품 정보를 기록
18	Register	웹 페이지 회원인증요청 정보를 기록
19	SellProduct	고객에게 판매한 상품 정보를 기록
20	StaffLogin	가맹점 직원 정보를 기록
21	StaffLoginLog	가맹점 직원의 로그인 내역 기록

• 테이블명세서

테이블 ID	테이블 설명	길이
Cateogry	상품 분류하는 카테고리 기록	
DB 테이블 생성 명령문		
<pre>CREATE TABLE [dbo].[Category] (</pre>		

[Id] NvarChar(3) NOT NULL PRIMARY KEY, --카테고리를 분별할 수 있는 코드 [Name] NvarChar(20) NOT NULL, --카테고리명 [Level] Int NOT NULL, --카테고리 상하위 구분 [Parent] NvarChar(20) NULL -- 상위 카테고리 )		
테이블 ID	테이블 설명	길이
ChatRoom	채팅방 정보를 기록	
DB 테이블 생성 명령문		
<pre>CREATE TABLE [dbo].[ChatRoom] (     [Num] int NOT NULL PRIMARY KEY, --채팅방 번호     [Name] NvarChar(20) NOT NULL, --채팅방명     [Chatter] NvarChar(200) NOT NULL, --채팅방에 들어온 가맹점 목록     [Date] DateTime Default(GetDaTe()) --날짜 및 시간 )</pre>		
테이블 ID	테이블 설명	길이
FranchiseInfo	가맹점 정보를 기록	
DB 테이블 생성 명령문		
<pre>CREATE TABLE [dbo].[FranchiseInfo] (     [Name] NvarChar(20) Not Null, -- 가맹점명     [OwnerName] NvarChar(10) Not Null, -- 가맹점주 이름     [OwnerPhoneNumber] NvarChar(25) Not Null, --가맹점주 연락처     [RegistrationNumber] NvarChar(30) Not Null PRIMARY KEY, --사업자등록번호     [ContactNumber] NvarChar(30) Not Null, -- 가맹점 연락처     [Address] NvarChar(30) Not Null, --주소     [Date] Datetime Default(GetDate())--날짜 및 시간 )</pre>		
테이블 ID	테이블 설명	길이
FranStock	가맹점 재고를 기록	
DB 테이블 생성 명령문(DML)		
<pre>CREATE TABLE [dbo].[FranStock] (     [Id] NvarChar(25) NOT NULL References Product(Id), -- 제품코드     [RegiNum] NvarChar(30) NOT NULL References FranchiseInfo(RegistrationNumber),</pre>		

```

--가맹점 구분할 수 있는 사업자등록번호
[Name] NVarChar(30) NOT NULL, -- 제품명
[Price] Int NOT NULL, --가격
[Account] Int NOT NULL, --재고량
[State] NVarChar(25) NOT NULL, --현황(입고/출고/반품) 구분
[Date] DateTime Default(GETDATE())--날짜 및 시간
)

```

테이블 ID	테이블 설명	길이
HeadStock	본사의 재고를 기록	

#### DB 테이블 생성 명령문(DML)

```

CREATE TABLE [dbo].[HeadStock]
(
    [Id] NVarChar(25) NOT NULL References Product(Id), -- 제품코드
    [RegiNum] NVarChar(30) NULL , -- 사업자등록번호
    [Name] NVarChar(30) NOT NULL, -- 제품명
    [InPrice] Int NOT NULL, -- 입고가
    [OutPrice] int NULL, -- 출고가
    [Account] Int NOT NULL, -- 수량
    [Date] DateTime Default(GETDATE()),--날짜 및 시간
    [State] NVarChar(25) NOT NULL, -- 현황(입고/출고/반품) 구분
    [Pay] NVarChar(20) NULL - 지출(현금/외상) 구분
)

```

테이블 ID	테이블 설명	길이
Itemback	반품/불량품을 기록	

#### DB 테이블 생성 명령문(DML)

```

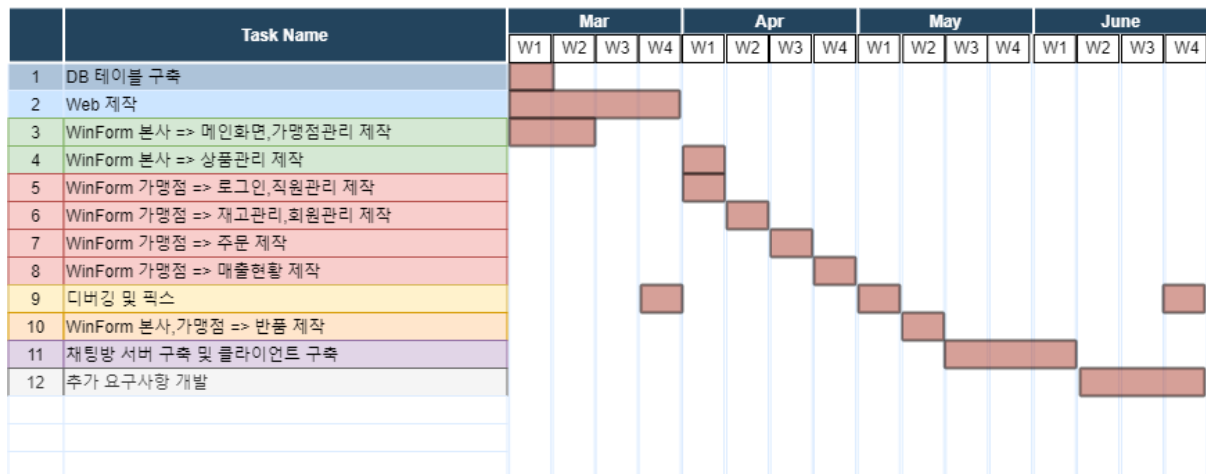
CREATE TABLE [dbo].[Itemback]
(
    [Num] int NOT NULL PRIMARY KEY Identity(0,1), -- 특정 할 수 있는 번호
    [Id] NVarChar(25) NOT NULL References Product(Id), --제품코드
    [RegiNum] NVarChar(30) NOT NULL References FranchiseInfo(RegistrationNumber),
    -- 사업자등록번호
    [Name] NVarChar(30) NOT NULL, --제품명
    [InPrice] int NOT NULL, -- 입고가
    [OutPrice] Int NOT NULL, -- 출고가
    [Account] Int NOT NULL, -- 수량
    [State] NVarChar(20) NOT NULL, -- 반품사유(반품/불량) 구분
    [Date] DateTime Default(GETDATE())--날짜 및 시간
)

```

)		
테이블 ID	테이블 설명	길이
Note	게시판 글 정보를 기록	
DB 테이블 생성 명령문(DML)		
<pre> CREATE TABLE [dbo].[Note] (     [Id] INT IDENTITY (1, 1) NOT NULL, -- 게시판 번호     [Name] NVARCHAR (25) NOT NULL, -- 게시판     [Email] NVARCHAR (100) NULL, -- 이메일     [Title] NVARCHAR (150) NOT NULL, -- 게시판 제목     [PostDate] DATETIME DEFAULT (getdate()) NOT NULL, -- 게시한 날짜 및 시간     [PostIP] NVARCHAR (15) NULL, -- 게시할 때 IP     [Content] NTEXT NOT NULL, -- 글 내용     [RegiNum] NVARCHAR (30) NOT NULL, -- 가맹점 구분을 위한 사업자등록번호     [ReadCount] INT DEFAULT ((0)) NULL, -- 게시글 읽은 수     [Encoding] NVARCHAR (10) NOT NULL, -- 인코딩     [Homepage] NVARCHAR (100) NULL,     [ModifyDate] DATETIME NULL, -- 수정한 날짜     [ModifyIP] NVARCHAR (15) NULL, -- 수정할 때 IP     [FileName] NVARCHAR (255) NULL, -- 파일명     [FileSize] INT DEFAULT ((0)) NULL, -- 파일크기     [DownCount] INT DEFAULT ((0)) NULL, -- 다운로드 횟수     [Ref] INT NOT NULL,     [Step] INT DEFAULT ((0)) NULL,     [RefOrder] INT DEFAULT ((0)) NULL,     [AnswerNum] INT DEFAULT ((0)) NULL,     [ParentNum] INT DEFAULT ((0)) NULL,     [CommentCount] INT DEFAULT ((0)) NULL, -- 댓글 수     [Category] NVARCHAR (10) NULL, -- 카테고리     PRIMARY KEY CLUSTERED ([Id] ASC) ) </pre>		
)		
테이블 ID	테이블 설명	길이
Order	본사-가맹점간 주문(요청)을 기록	
DB 테이블 생성 명령문(DML)		
<pre> CREATE TABLE [dbo].[Order] (     [Num] int NOT NULL PRIMARY KEY Identity(0,1), --특정을 위한 번호     [Id] NVarChar(25) NOT NULL References Product(Id), --제품명     [RegiNum] NVarChar(30) NOT NULL References FranchiseInfo(RegistrationNumber),     --가맹점 구분을 위한 사업자등록번호 </pre>		

)

- 프로젝트 개발 일정



## 3.

## 프로젝트 소스 코드

Server – ServerConnect.cs – ServerStart 함수

```

130 private void ServerStart()
131 {
132     int cnt = 0;
133     Invoke(AddText, "서버 실행", "1");
134     while (Start)
135     {
136         try
137         {
138             lock (obj)
139             {
140                 clients.Add(new Client { SerClient = Server.AcceptTcpClient(), RoomNum = -2, Main = true, Room = false });
141                 cnt = clients.Count - 1;
142
143                 myStream = clients[cnt].SerClient.GetStream();
144                 myRead = new StreamReader(myStream);
145
146                 if (myStream.CanRead)
147                 {
148                     var msg = myRead.ReadLine();
149                     string[] Smsg = msg.Split('&');
150                     if (Smsg.Length > 0)
151                     {
152                         for (int i = 0; i < cnt; i++)
153                         {
154                             if (clients[i].RegiNum == Smsg[0])
155                             {
156                                 foreach (Client cli in clients)
157                                 {
158                                     if (cli.SerClient == clients[i].SerClient)
159                                     {
160                                         NetworkStream stream = cli.SerClient.GetStream();
161                                         StreamWriter mywrite = new StreamWriter(stream);
162                                         mywrite.WriteLine(cli.RegNum + "&&&연결 끊음");
163                                         mywrite.Flush();
164                                     }
165                                 }
166                                 clients.RemoveAt(i);
167                                 cnt = clients.Count - 1;
168                                 break;
169                             }
170                         }
171                         clients[cnt].RegiNum = Smsg[0];
172                     }
173                     Invoke(AddText, GetNameByRegiNum(Smsg[0]) + "님이 입장하셨습니다.", "1");
174                 }
175                 myReader = new Thread(() => Run(clients[cnt]));
176                 myReader.IsBackground = true;
177                 myReader.Start();
178             }
179         }
180         catch
181         {
182         }
183     }
184 }

```

채팅 서버를 실행하여 클라이언트가 접속을 할 때 입장 신호를 받는 함수이다.

lock을 이용해 동시에 클라이언트의 접속할 때 상호 배제를 구현하였다.

## Server - ServerConnect.cs - Run 함수

```

    mywrite.Flush();
}

}

Invoke(AddText, Smsg[0] + "," + Smsg[1] + "," + Smsg[2] + "," + Smsg[3] + "," + Smsg[4] + ",", "1");
}

else if (Smsg[4] == "방 개설")
{
lock (obj3)
{
StreamWriter mywrite = new StreamWriter(Stream);
client.RoomNum = RoomNum;
RoomNum++;
mywrite.WriteLine(Smsg[0] + "&" + "방 개설 완료&" + client.RoomNum + "&" + Smsg[3] + "&" + "방 개설 완료");
mywrite.Flush();
CreateRoom(Smsg[0], Smsg[1], client.RoomNum);

myRoom.Add(new Thread(() => RoomChat(client)));
}
Invoke(AddText, Smsg[0] + "," + Smsg[1] + "," + Smsg[2] + "," + Smsg[3] + "," + Smsg[4] + ",", "1");
}

else if (Smsg[4] == "방 입장")
{
lock (obj3)
{
StreamWriter mywrite = new StreamWriter(Stream);
client.RoomNum = Convert.ToInt32(Smsg[3]);
Invoke(AddText, Smsg[0] + "," + Smsg[1] + "," + Smsg[2] + "," + Smsg[3] + "," + Smsg[4] + ",", "1");
}
if (client.RoomNum != -1)
{
if (client.RoomNum != -1)
{
client.Main = false;
client.Room = true;
myRoom[client.RoomNum].IsBackground = true;
myRoom[client.RoomNum].Start();
myRoom[client.RoomNum].Join();
}
}

else if (Smsg[4] == "연결 끊음")
{
lock (obj3)
{
for (int i = 0; i < clients.Count; i++)
{
if (clients[i].RegiNum == Smsg[0])
{
Invoke(AddText, clients[i].RegiNum + "님이 나가셨습니다.", "1");
clients.RemoveAt(i);
}
}
}

}

Invoke(AddText, Smsg[0] + "," + Smsg[1] + "," + Smsg[2] + "," + Smsg[3] + "," + Smsg[4] + ",", "1");
}

else
{
lock (obj3)
{
if (Smsg[5] == "0")

```

위의 ServerStart 함수에서 클라이언트 접속한 스레드를 실행하는 함수의 일부이다.

채팅방을 개설할 때 리스트로 방의 스레드를 생성하여 채팅방마다 별개의 스레드가 작동하게끔 설계하였다.

4.

## 프로젝트 개선 방안

- WinForm 제작에서 기능별 함수 분리를 명확하게 하여 향후 관리를 용이하게 해야 합니다.
- UI 제작에서 가시성을 조금 더 고려해서 수정해야 할 필요가 있습니다.

5.

## 참고문헌 및 URL

URL .

<https://jujun.tistory.com/435>

<https://hyokim.tistory.com/16>

<https://rainflys.tistory.com/221>

<https://ella-devblog.tistory.com/70>

참고문헌 .

ASP.NET & Core를 다루는 기술

이것이 C#이다

C# 7.0 프로그래밍 실전 프로젝트