

□ 프로젝트 제목 : 장난감 재고 관리 시스템

□ 제출자 :

학번	이름
212210056	팀원 A
213210046	팀원 B
213210048	김문선
213210049	팀원 C
213210063	팀원 D
213210067	팀원 E

이 보고서를 2015학년도 졸업작품 완료보고서로  
제출합니다.

지도교수 : 교수님 (인)

교수님 (인)

교수님 (인)

## - 차 례 -

<b>1. 프로젝트 개요</b>	<b>003</b>
1.1 개발 목적 및 필요성	
1.2 개발 동향	
1.3 개발 환경 및 도구	
1.4 기대효과	
<b>2. 프로젝트 추진체계</b>	<b>004</b>
2.1 추진조직(팀원)	
2.2 추진일정	
2.3 업무분장(팀원별 역할)	
<b>3. 프로젝트 내용</b>	<b>005</b>
3.1 전체 블록 다이어그램(기능 또는 모듈 명시)	005
3.2 기능(모듈)별 순서도(클래스 이름과 함수 이름 명시)	006
3.3 기능(모듈)별 화면 및 주요내용	027
3.4 기능(모듈)별 클래스 목록 및 설명	036
3.5 기능(모듈)별 함수 목록 및 설명	038
3.6 기능(모듈)별 주요 소스코드 및 설명	047
3.7 전체 프로그램의 사용법(메뉴, 컨트롤 등)	126
3.8 업그레이드 요구 사항	128
• 현재 미해결 기능	
• 향후 개선이 요구되는 기능	
<b>4. 프로젝트 소감</b>	<b>129</b>

# 1. 프로젝트 개요

## 1.1 개발 목적 및 필요성

장난감과 아이들은 떼어놓을 수 없는 관계이다. 허나 아이들은 유행에 민감하고 수요 예측이 힘들기 때문에 대부분의 장난감 회사들이 만성적인 재고문제로 불가피한 영업 손실을 초래해왔다. 재고 분야는 기업의 생산성과 직결되는 가장 중요한 요소이기 때문에 이를 해결하기 위해 별도의 장부를 작성할 필요 없이 데이터화함으로써 효율적으로 관리할 수 있게끔 하는 프로그램이 필요할 것이라고 느꼈다.

이것을 기반으로 장난감 재고를 관리하는 관리자는 한꺼번에 한 프로그램에서 거래처 관리, 상품관리, 판매관리, 일정관리가 한꺼번에 가능할 수 있도록 할 수 있는 프로그램을 만드는 것이 가능하게 하는 것을 프로젝트의 목적으로 정했다.

기존의 관리 프로그램보다 구체적이고 편리한 관리 기능을 추가 및 수정, 간단한 사용 환경/쉬운 사용방법. 그로 하여금 관리의 효율성을 높이하고자 하며, 궁극적으로 이익 향상을 야기할 수 있는 프로그램을 만들고자 한다.

## 1.2 개발 동향

현재 시종에는 관리 시스템은 많이 개발되어 있는 상황이다. 그러나 보통 온라인쇼핑몰을 운영하는 회사는 재고관리를 하는 업체가 따로 있다. 쇼핑몰 상에서 판매된 데이터를 엑셀로 변환 받아서 재고관리프로그램에 업로드 하는 형태로 관리하고 있는데 우리는 프로그램 자체에서 관리자가 사전에 등록만 해놨다면 바로 이용 가능해 따로 복잡해지지 않고 쉽게 이용가능하다는 것이 다른 프로그램들과 차이를 가지며 매우 효율이 높을 것이다.

## 1.3 개발 환경 및 도구

Visual Studio, Microsoft Access, Photoshop, Illustrator

## 1.4 기대효과

기존 재고 관리 프로그램에 대하여 조사를 해보니 다음과 같은 점이 부족한 것을 알 수 있었다. 매출과 같이 회사의 직접적인 이익과 관련되어 보안이 중요시 되어야 할 데이터가 누구나 접근이 가능하게끔 되어있었다. 또한 한 화면에 굳이 연관성이 없는 기능들까지도 복잡하게 배치되어 있어서 한 번에 원하는 기능을 바로 찾아내는데 어려움이 따랐다. 이러한 결과를 바탕으로 보안기능과 편한 레이아웃을 차별성으로 하여 하나의 프로그램에서 모두 관리할 수 있을 것이다.

## 2. 프로젝트 추진체계

### 2.1 추진조직(팀원)

팀원A(212210056)  
 팀원B(213210045)  
 김문선(213210048)  
 팀원C(213210049)  
 팀원D(213210063)  
 팀원E(213210067)

### 2.2 추진일정

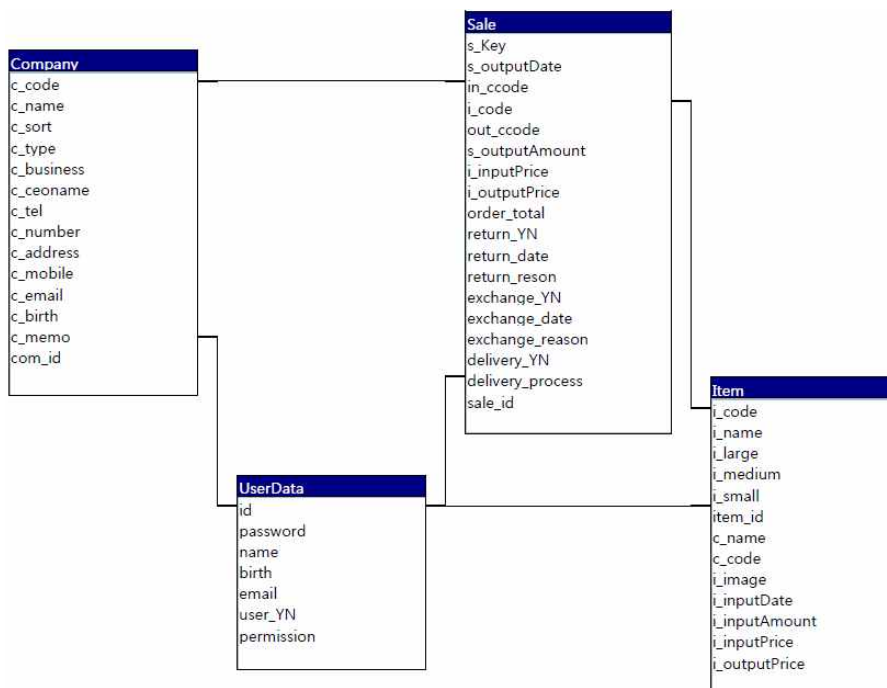
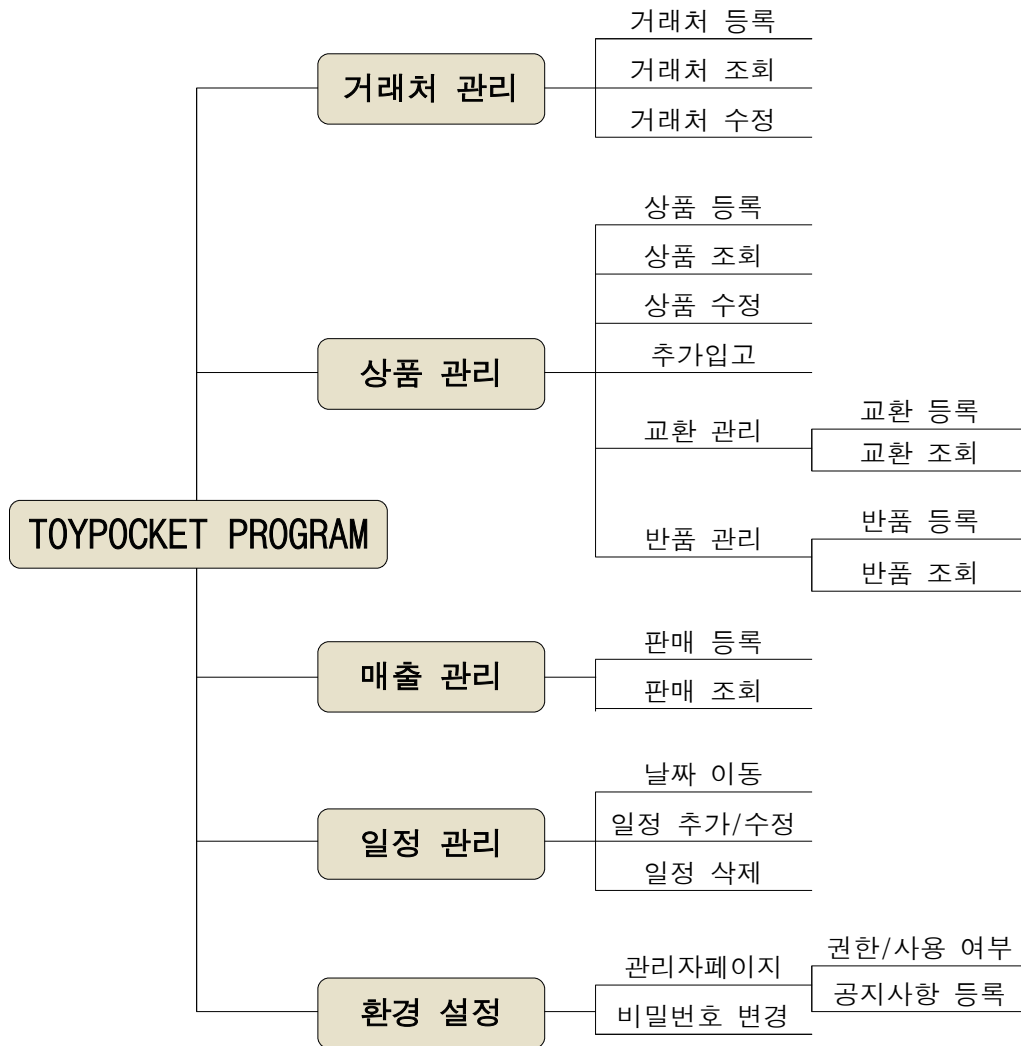
	5월	6월	7월	8월	9월	10월	11월	12월
주제선정								
작년 코드분석, 벤치마킹								
DB구축								
프로그램 사전 조사								
거래처관리 기능 구현								
상품관리 기능 구현								
판매관리 기능 구현								
교환/반품관리 기능 구현								
일정관리 기능 구현								
환경설정 및 기타 기능 구현								
세부 기능 추가/수정								
보고서 작성								

### 2.3 업무분장(팀원별 역할)

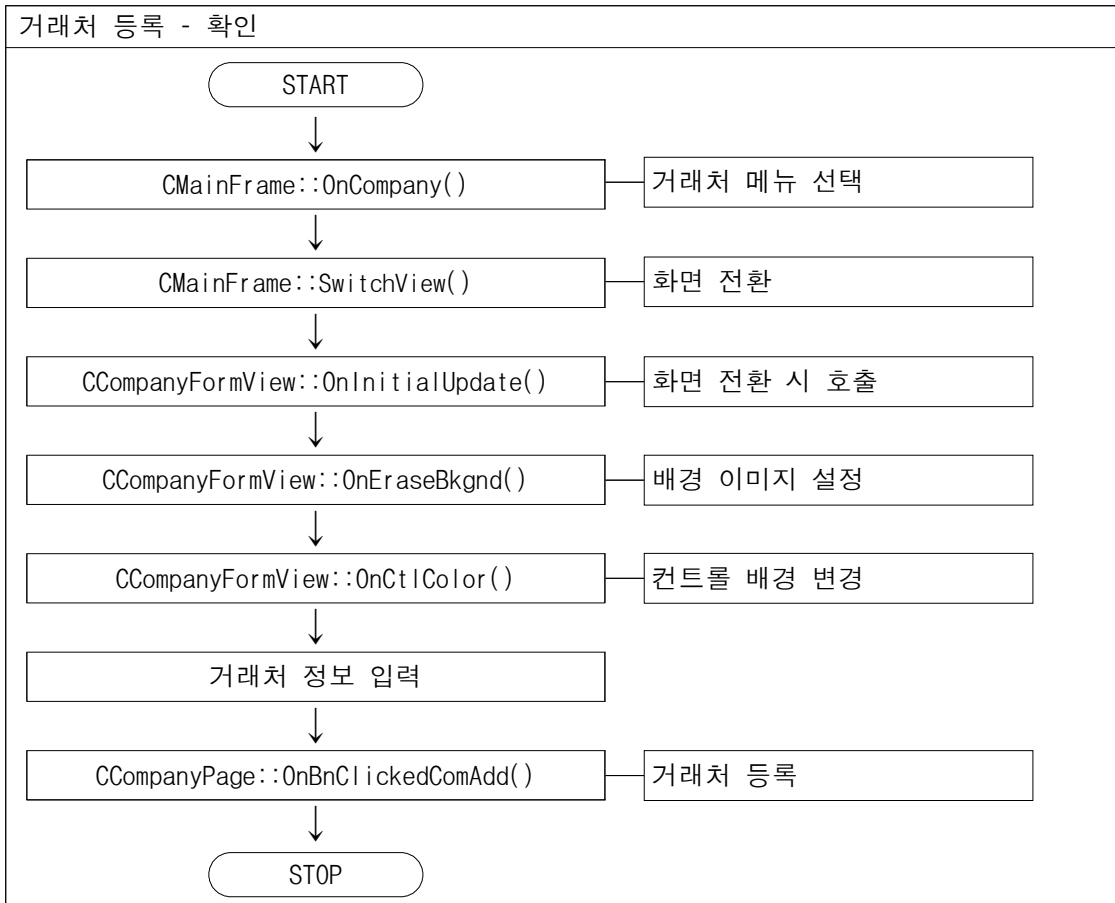
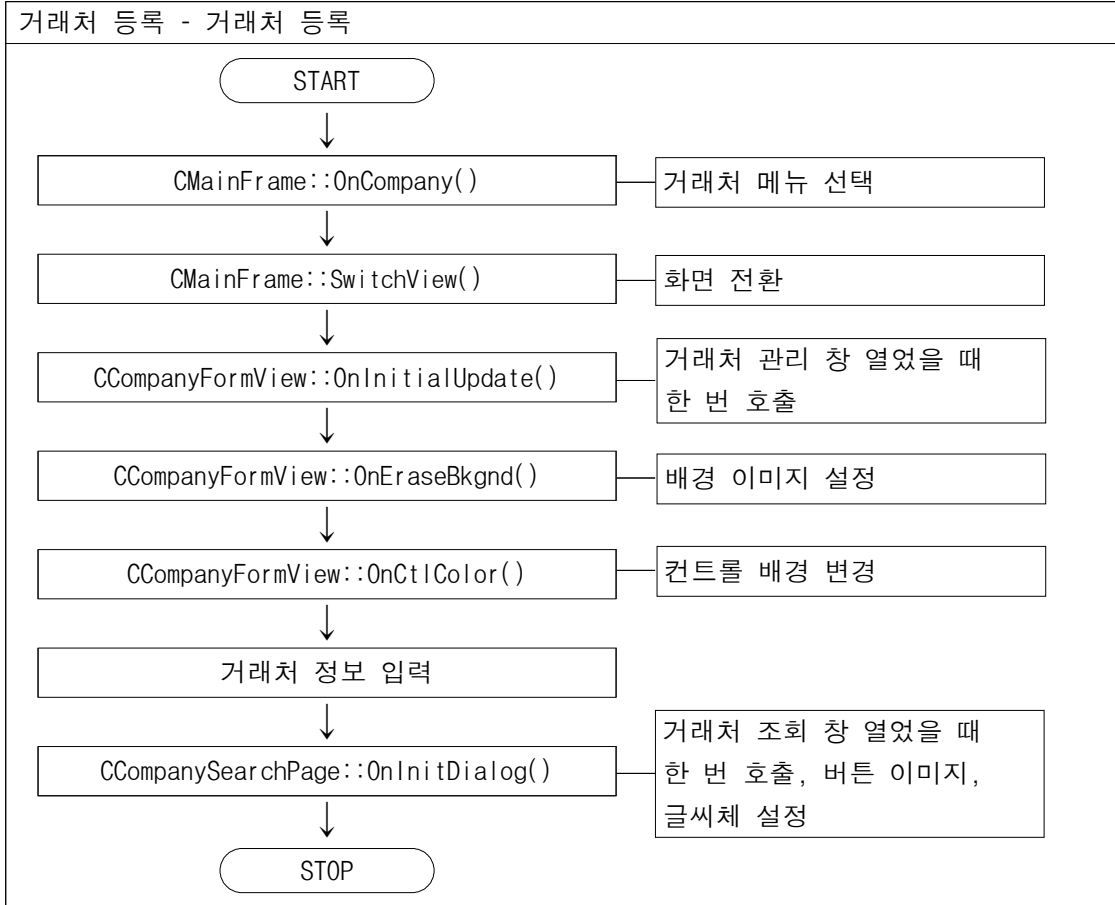
팀원	역할
팀원A	프로젝트 관리, 프로그램 사전조사, 데이터베이스 구축, 코딩
팀원B	프로젝트 관리, 프로그램 사전조사, 데이터베이스 구축, 코딩
김문선	프로젝트 관리, 프로그램 사전조사, 데이터베이스 구축, 코딩
팀원C	프로그램 사전조사, 데이터베이스 구축, 코딩
팀원D	프로그램 사전조사, 코딩, 화면/로고 디자인, 판넬
팀원E	프로그램 사전조사, 화면/로고 디자인

### 3. 프로젝트 내용

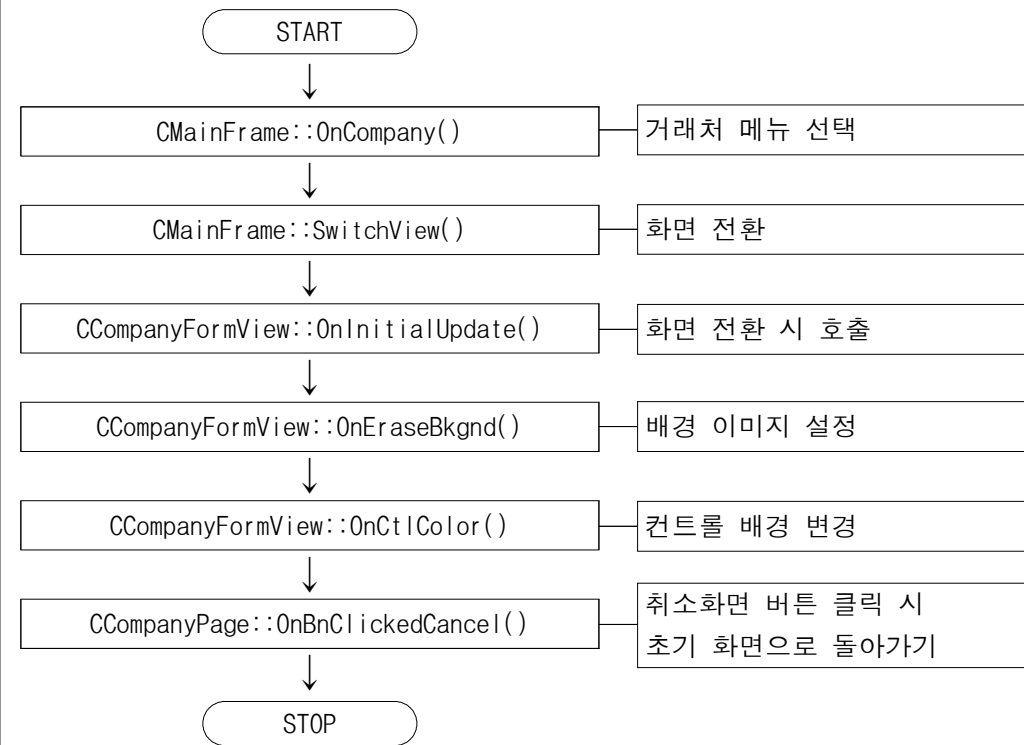
#### 3.1 전체 블록 다이어그램(기능 또는 모듈 명시)



### 3.2 기능(모듈)별 순서도(클래스 이름과 함수 이름 명시)



## 거래처 등록 - 취소

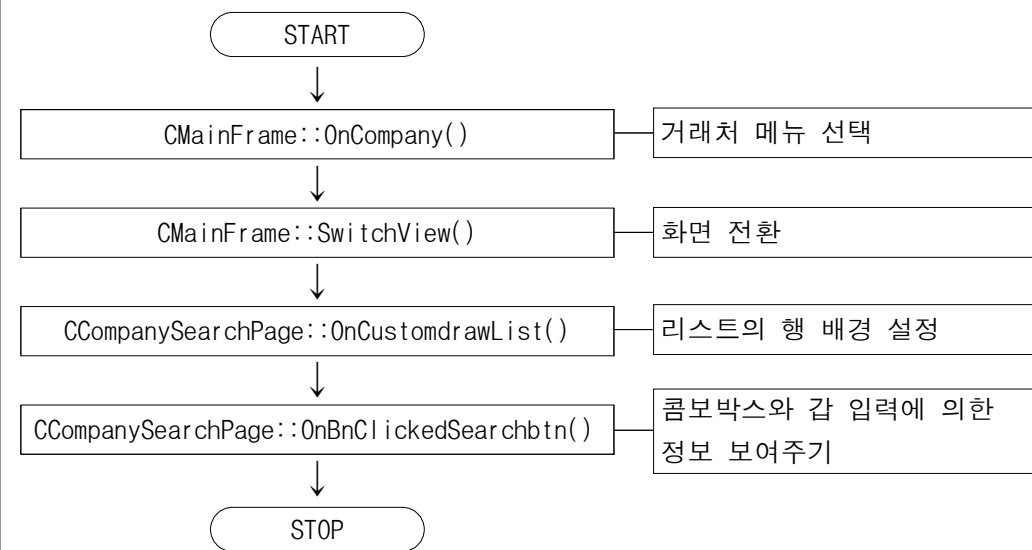


거래처 조회 - 거래처 조회

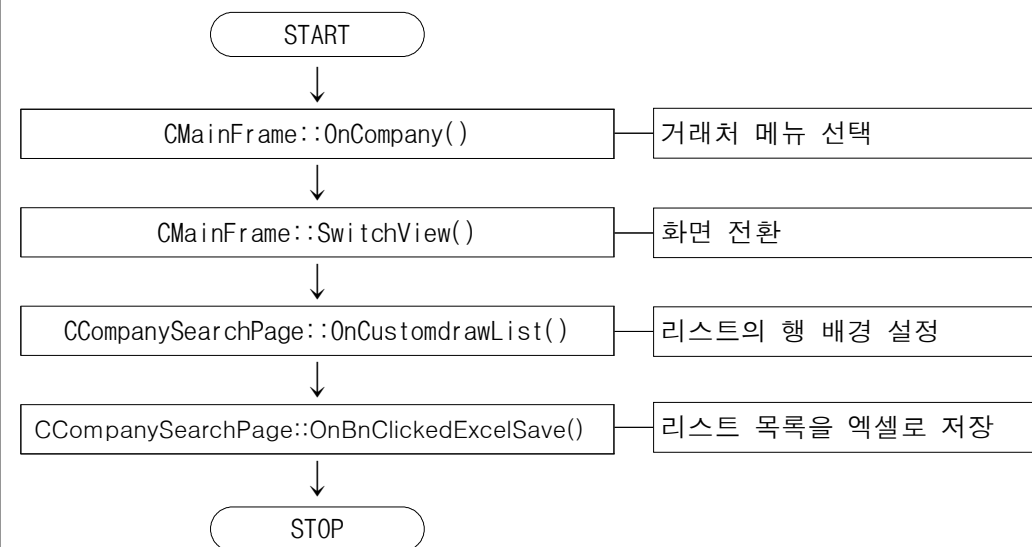




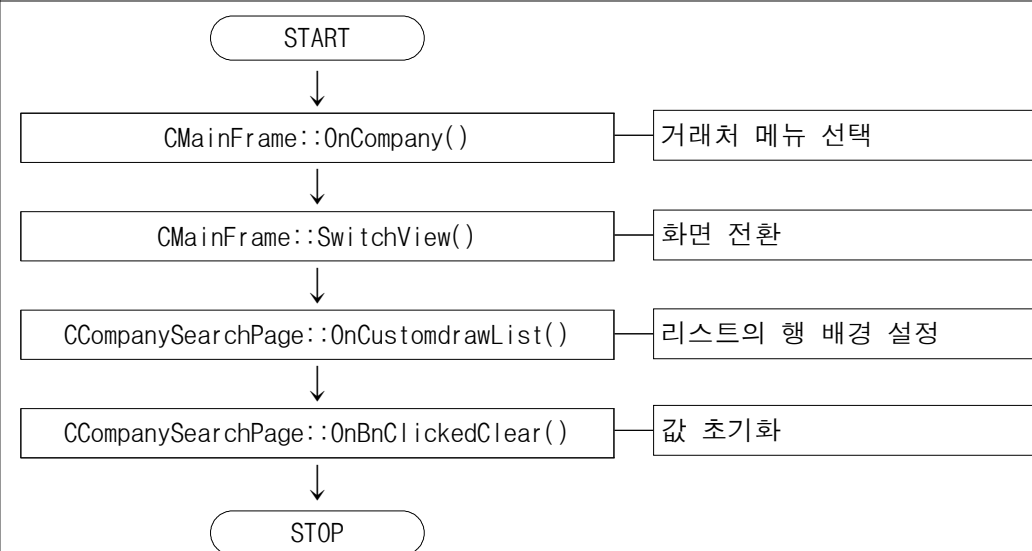
### 거래처 조회 - 조회



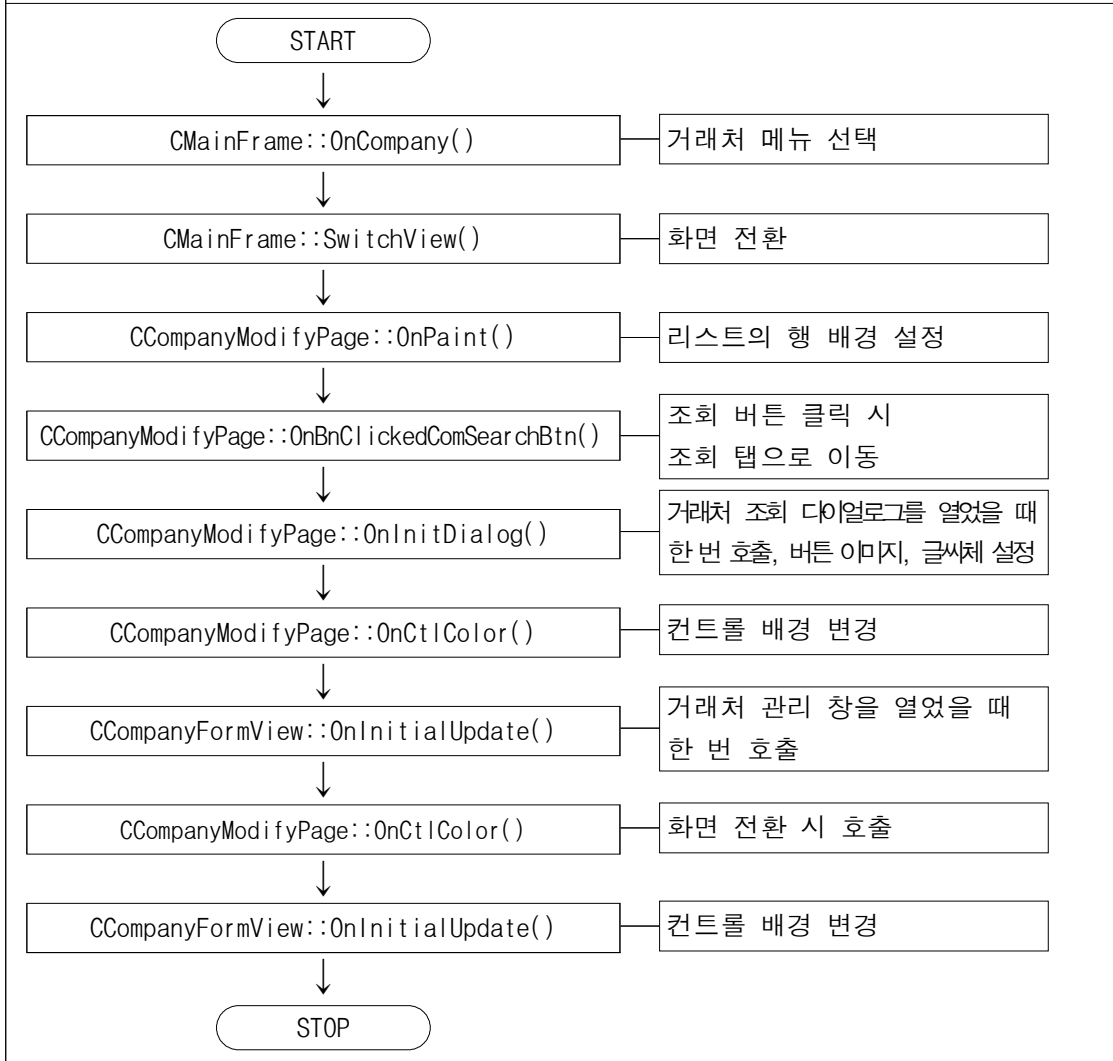
### 거래처 조회 - 엑셀 저장



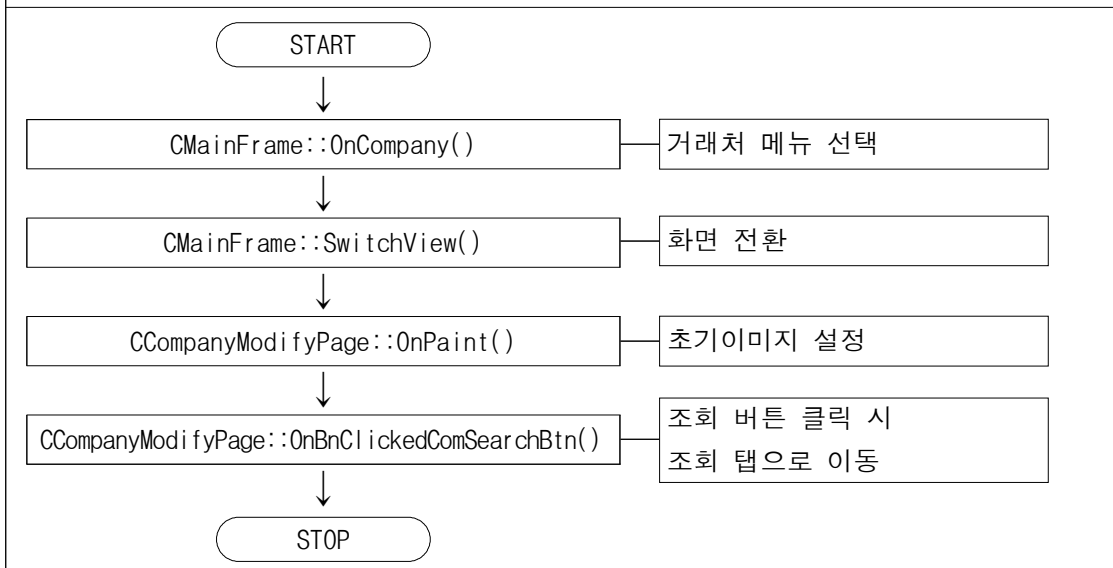
### 거래처 조회 - 초기화



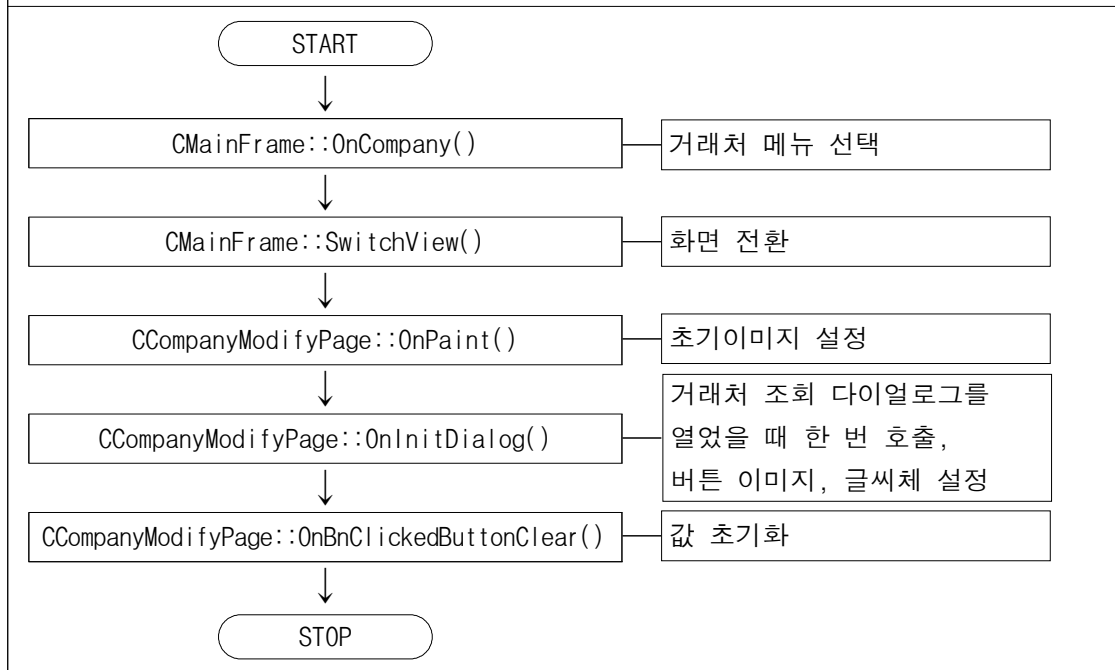
## 거래처 수정 - 거래처 수정



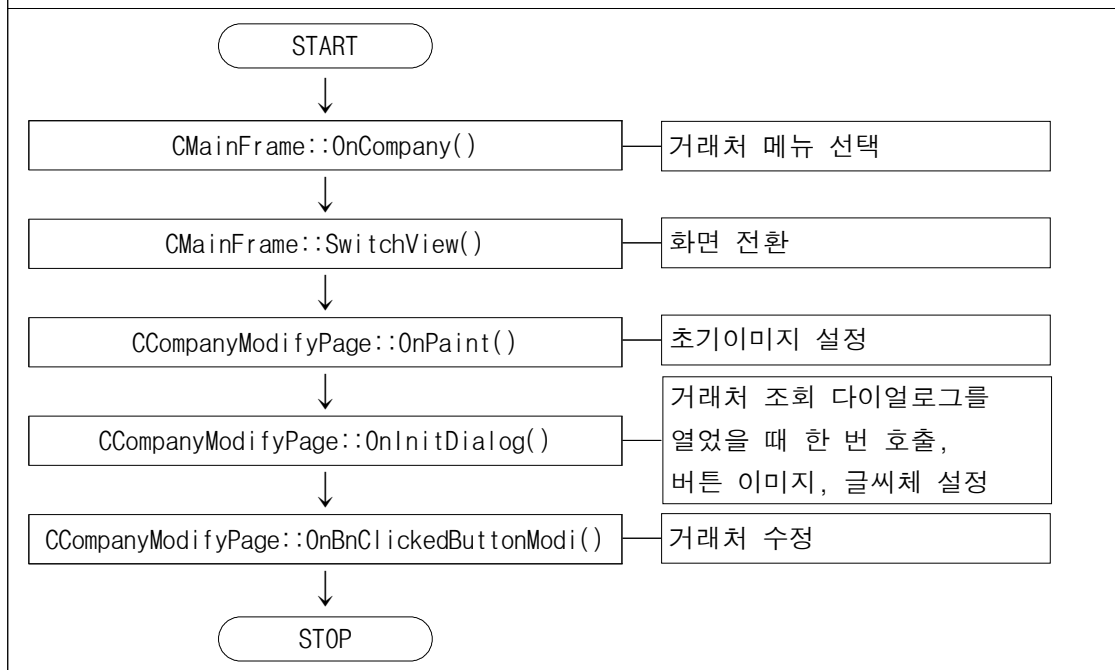
## 거래처 수정 - 검색



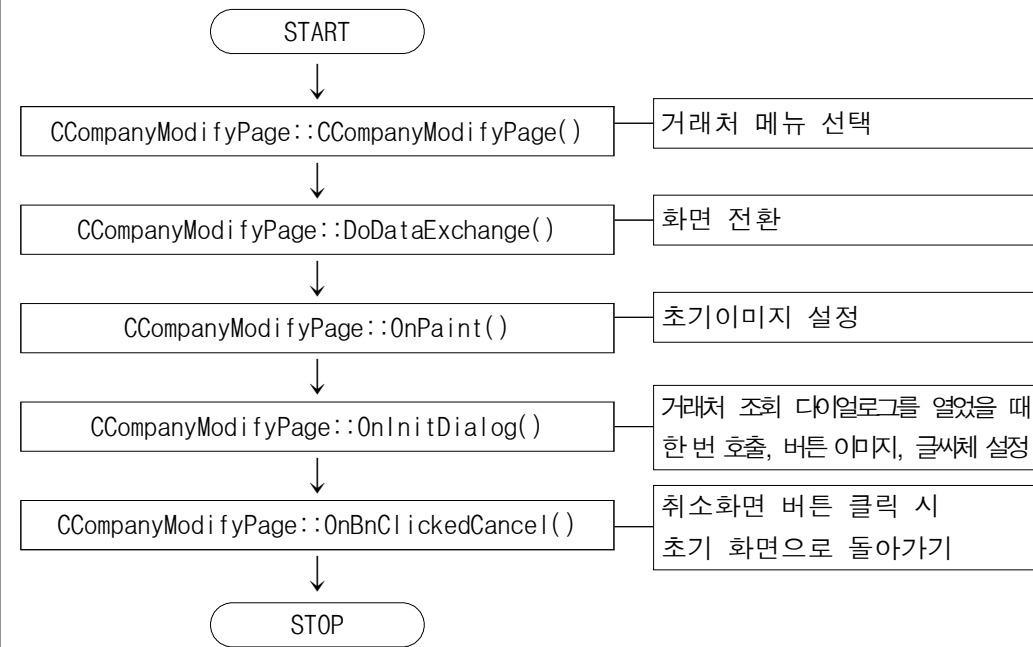
### 거래처 수정 - 초기화



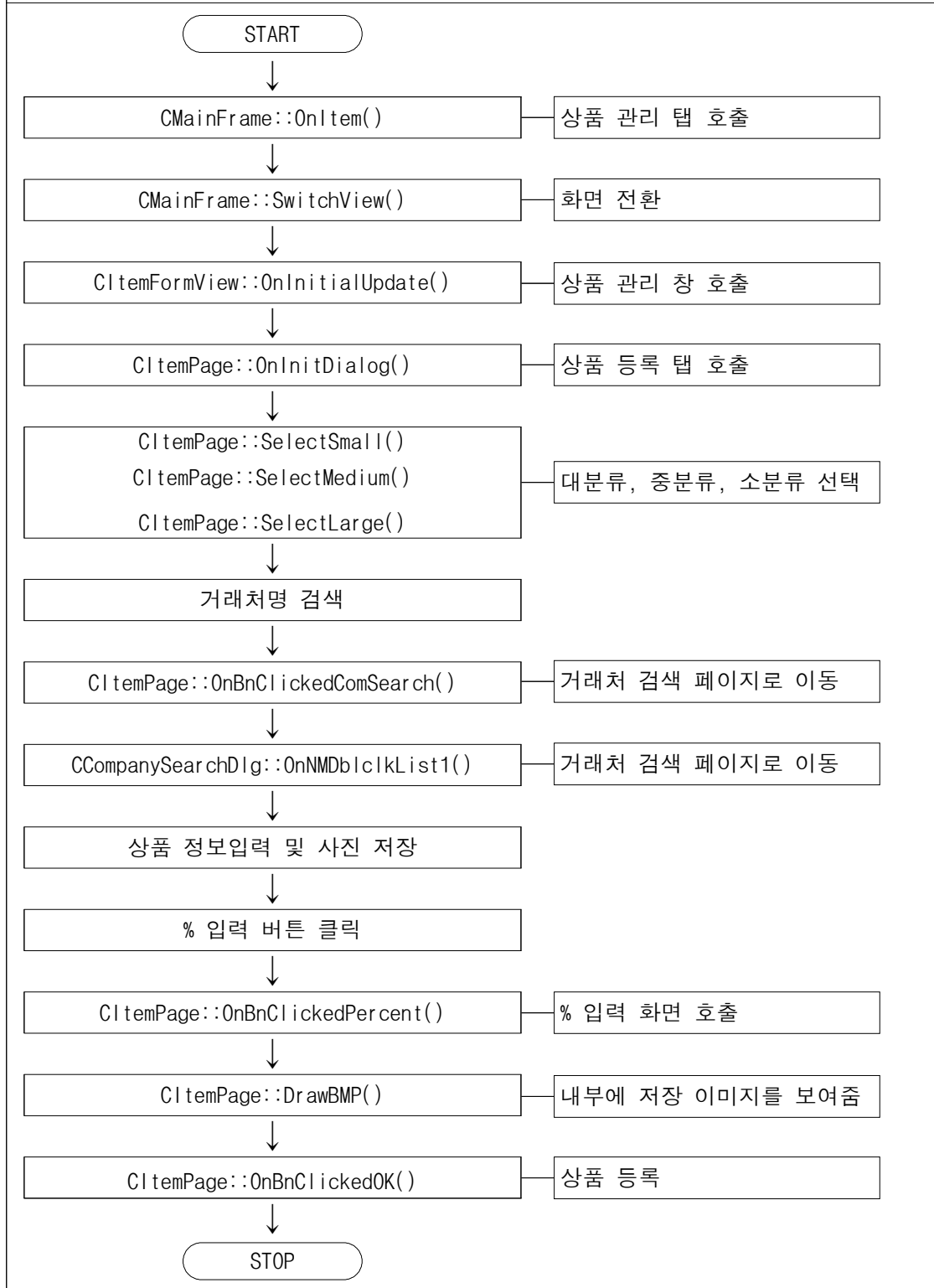
### 거래처 수정 - 수정



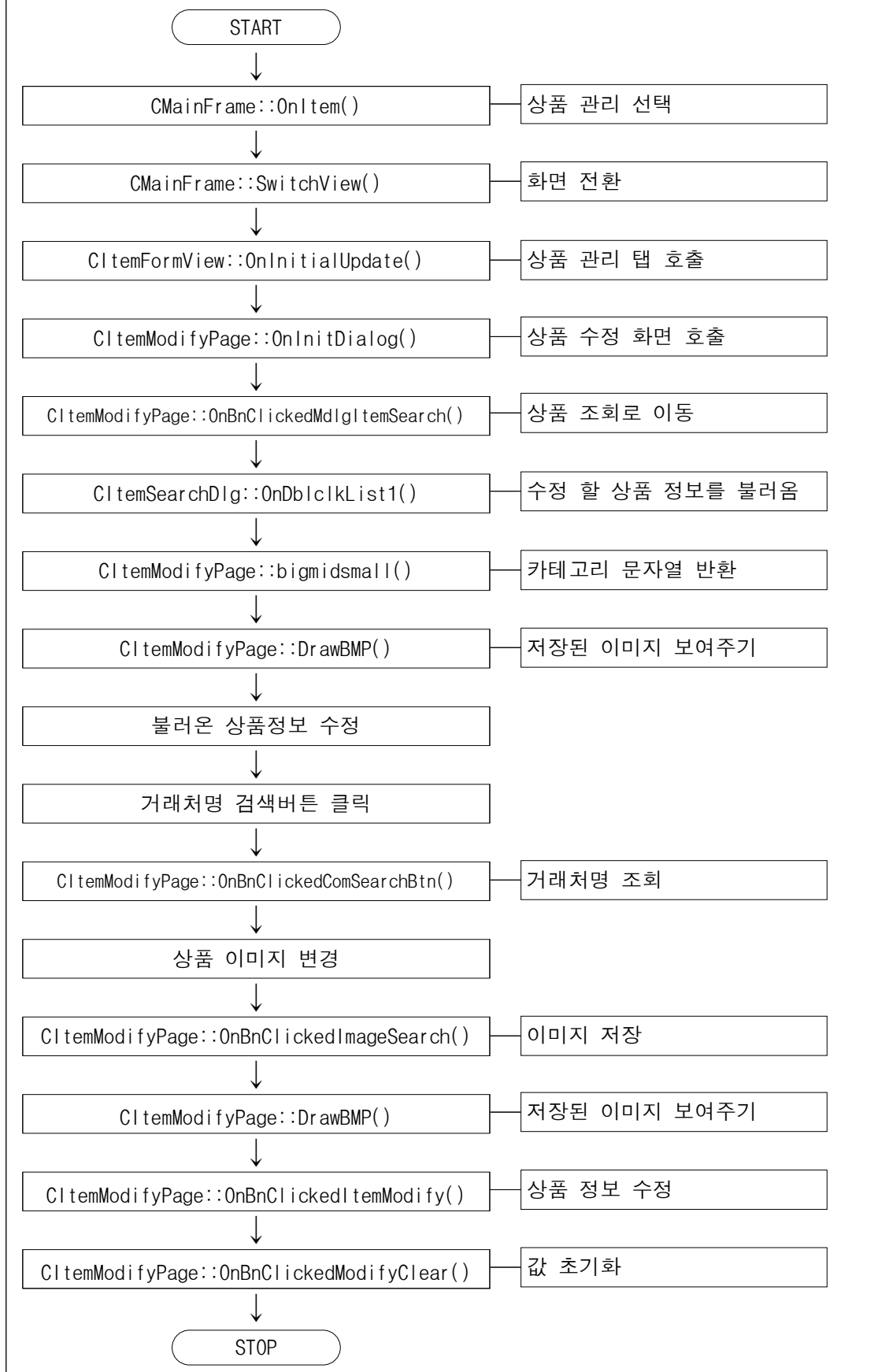
거래처 수정 - 취소



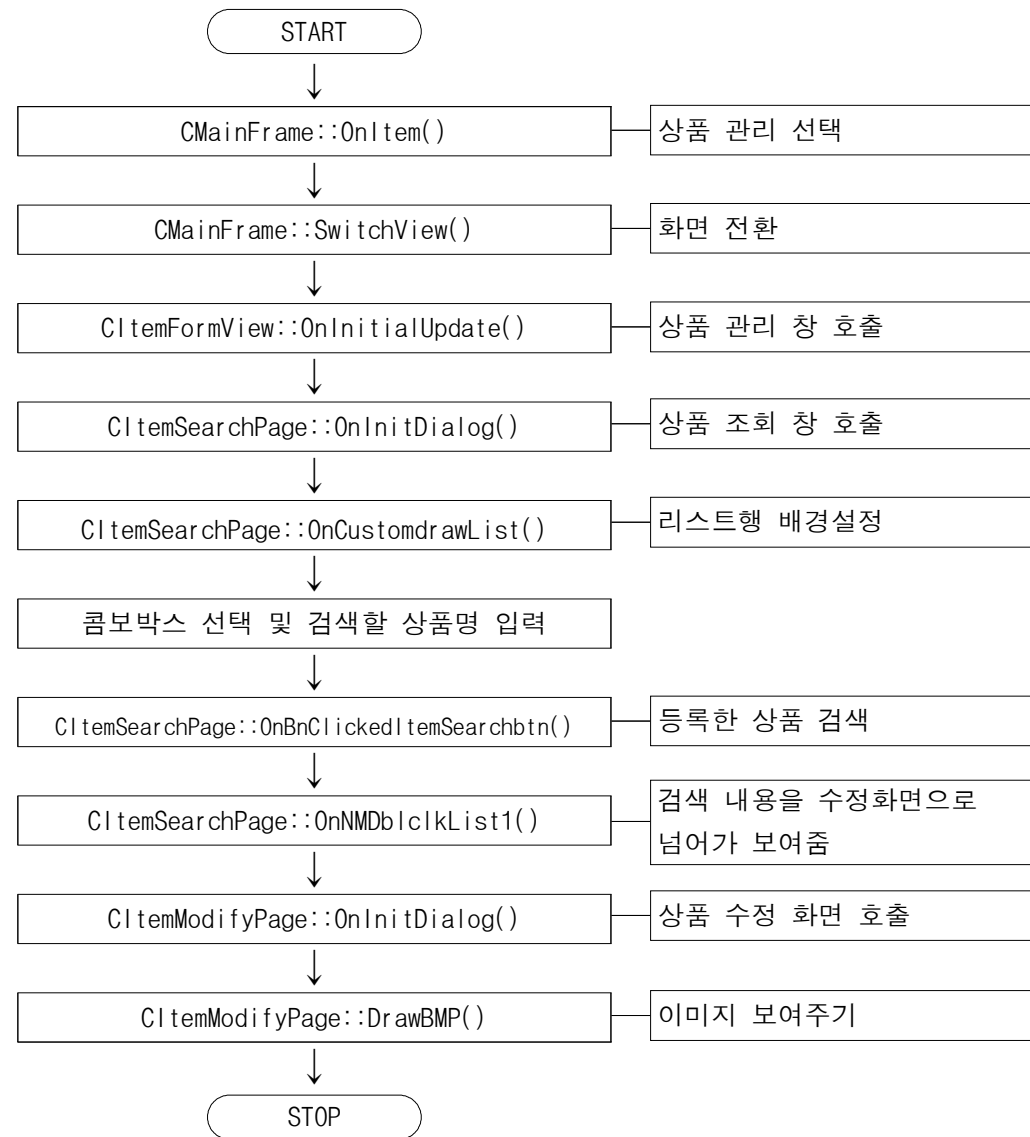
상품 관리 - 상품 등록



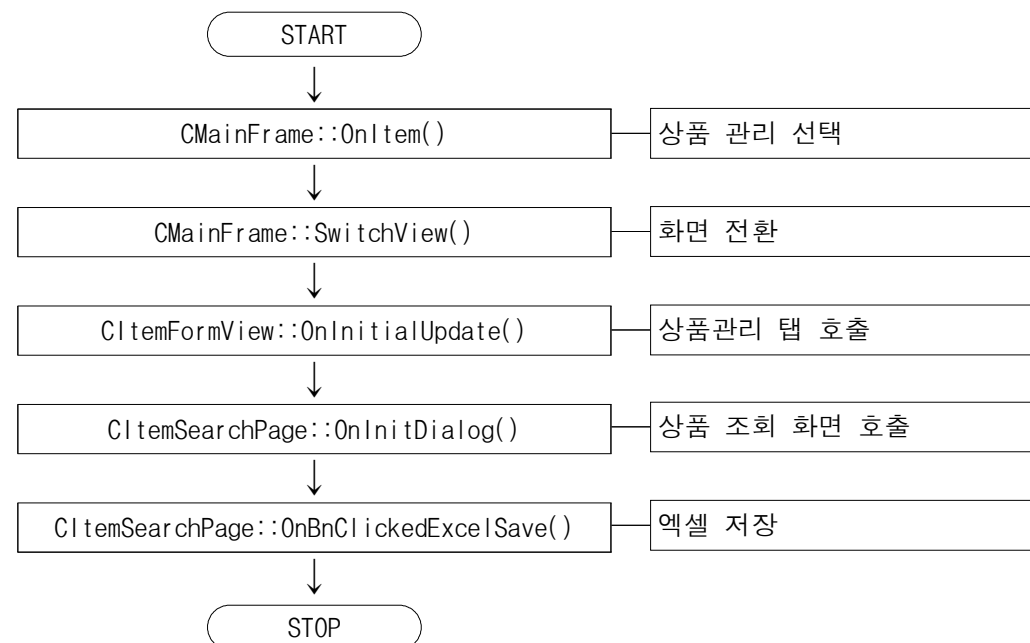
상품 관리 - 상품 수정



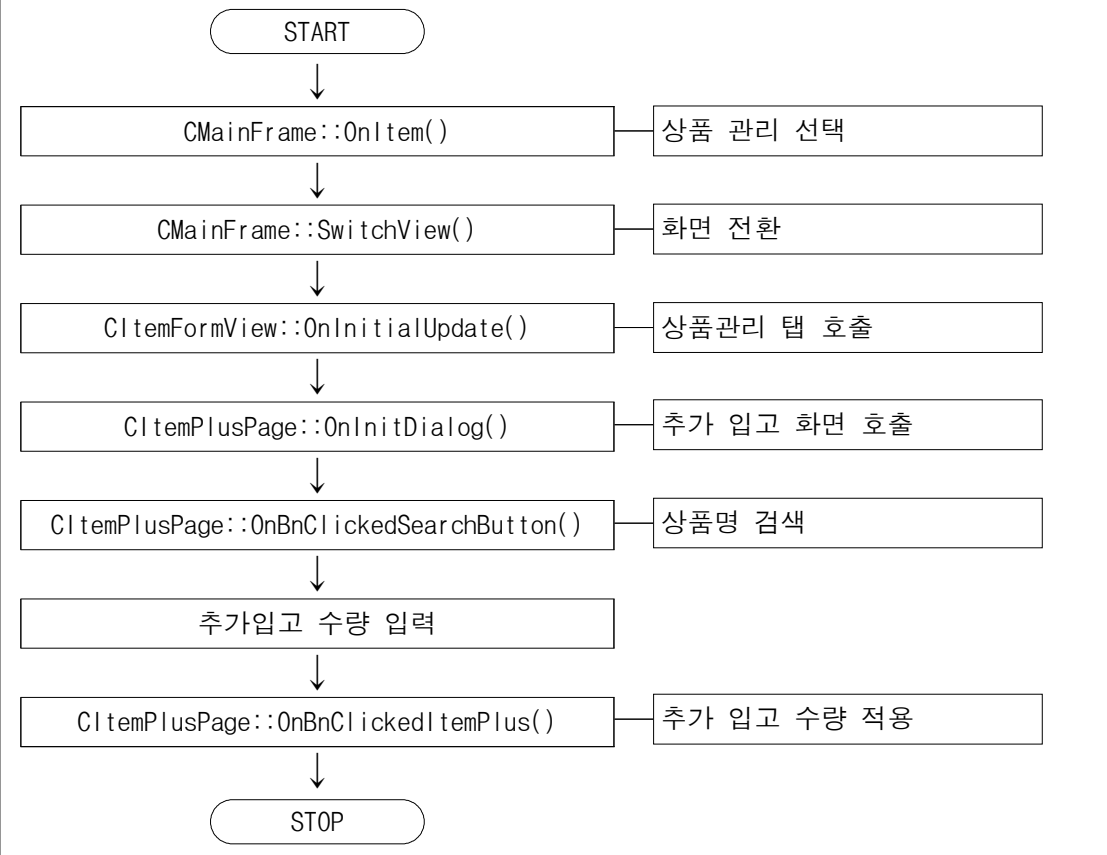
상품 관리 - 상품 조회



상품관리 - 상품조회\_엑셀 저장

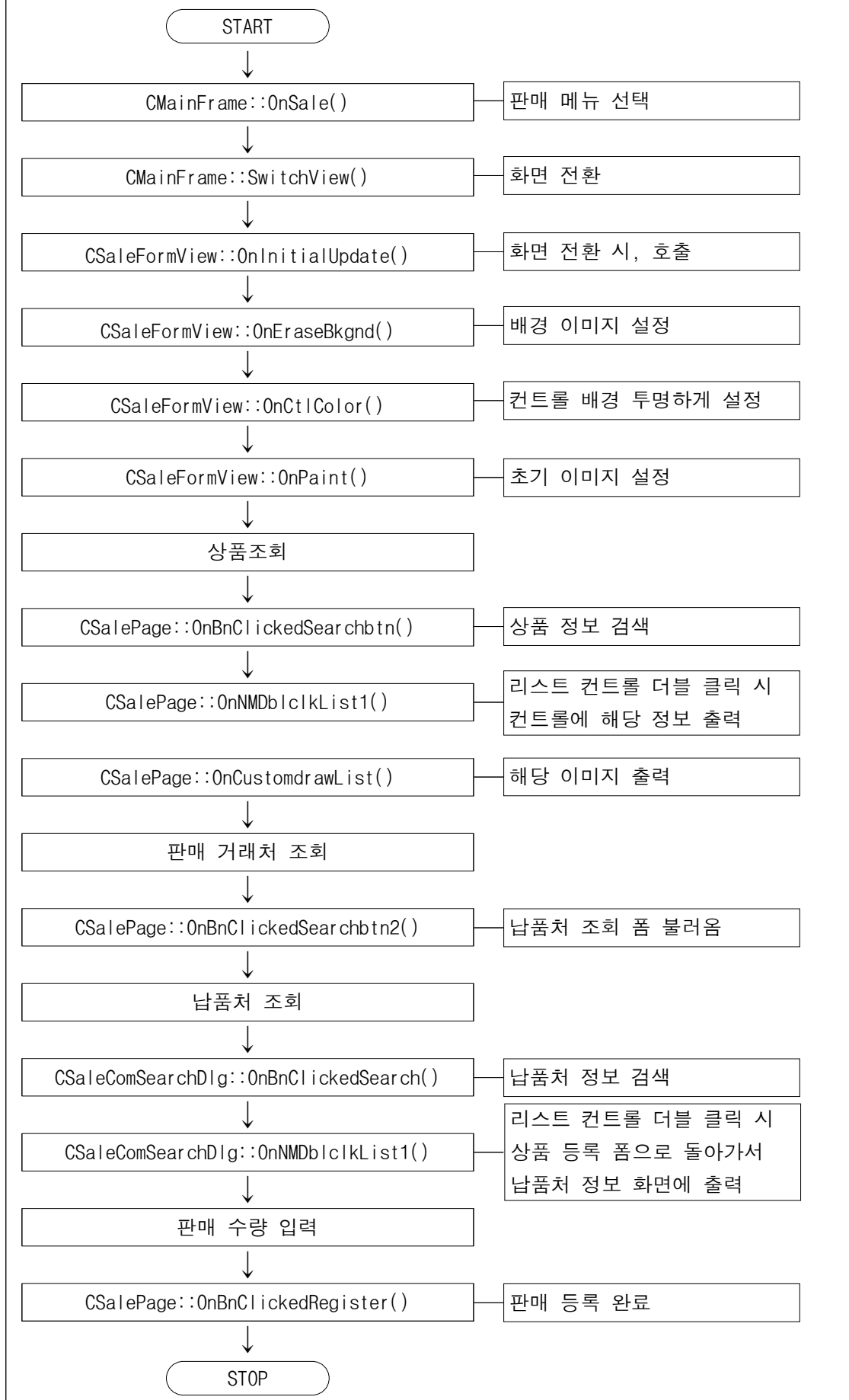


상품 관리 - 추가입고

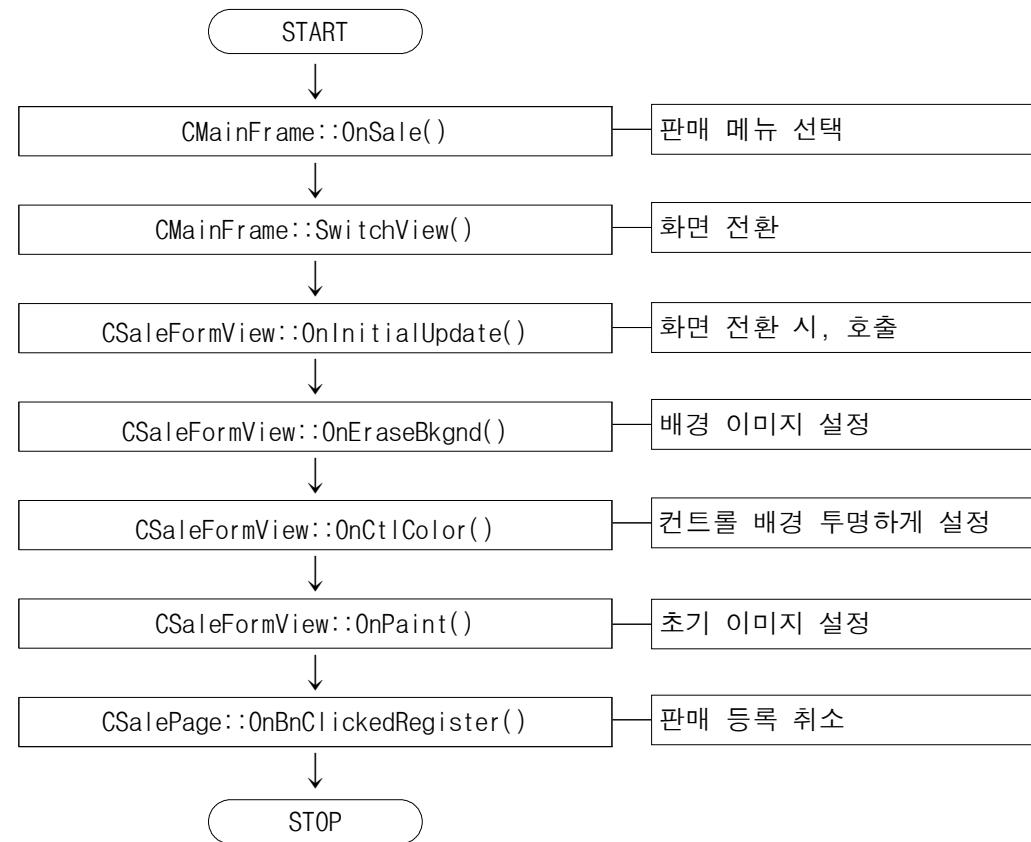




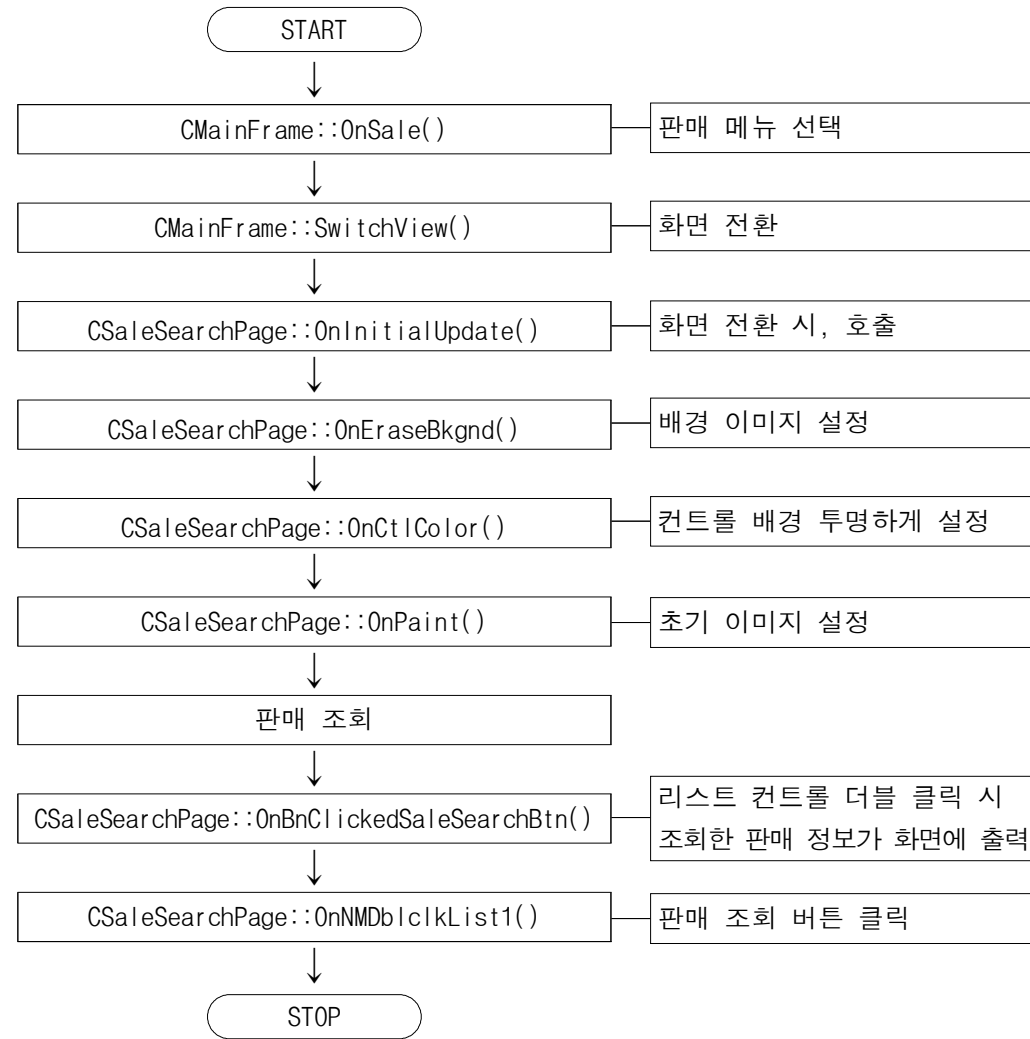
판매 - 판매 등록



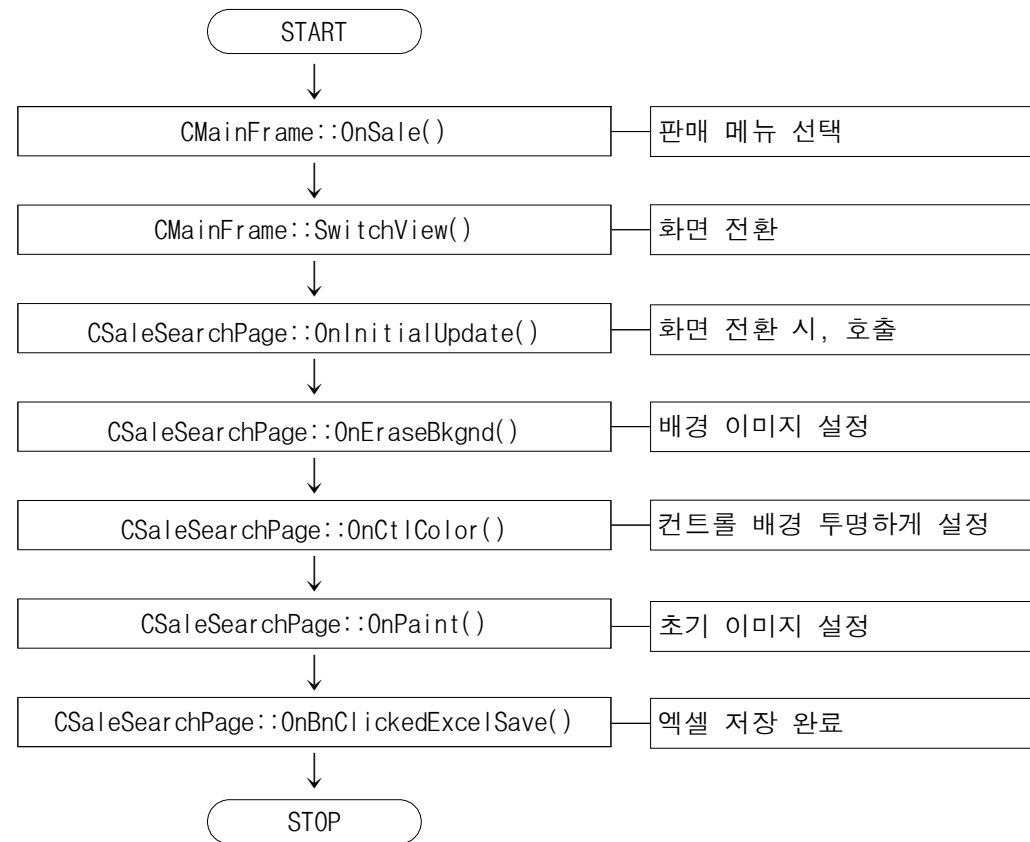
판매 - 판매 등록 취소



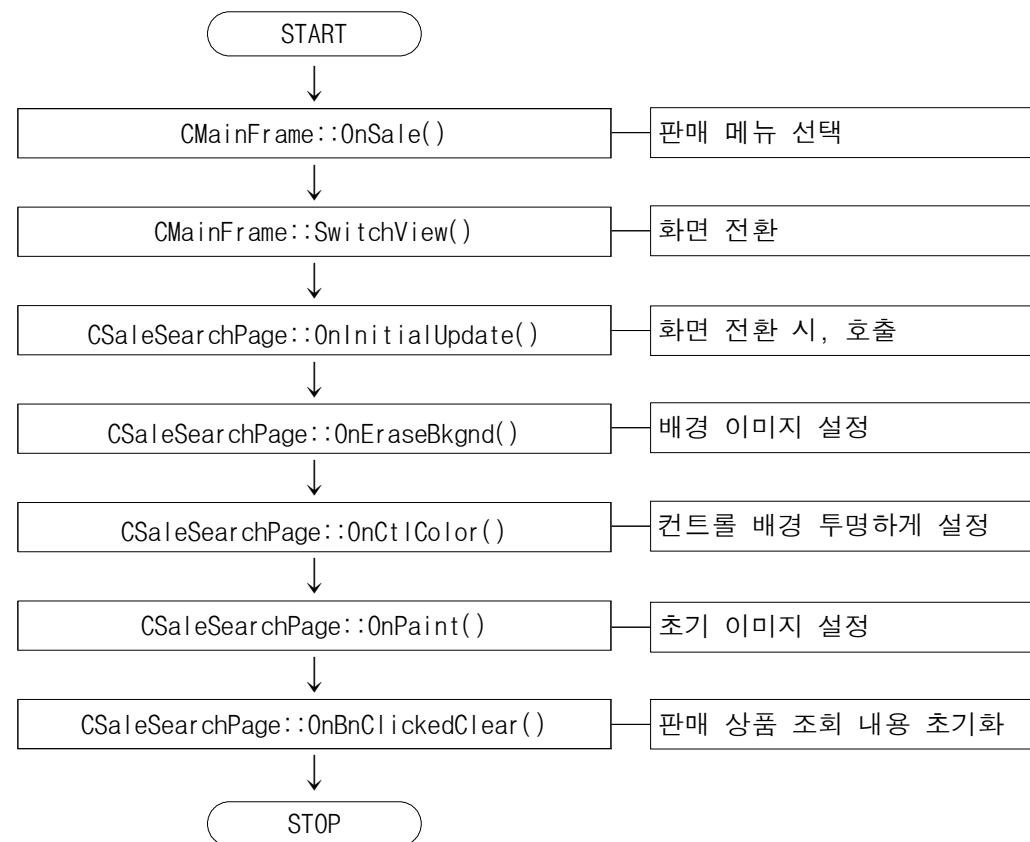
판매 - 판매 조회



판매 - 판매 조회\_엑셀 저장



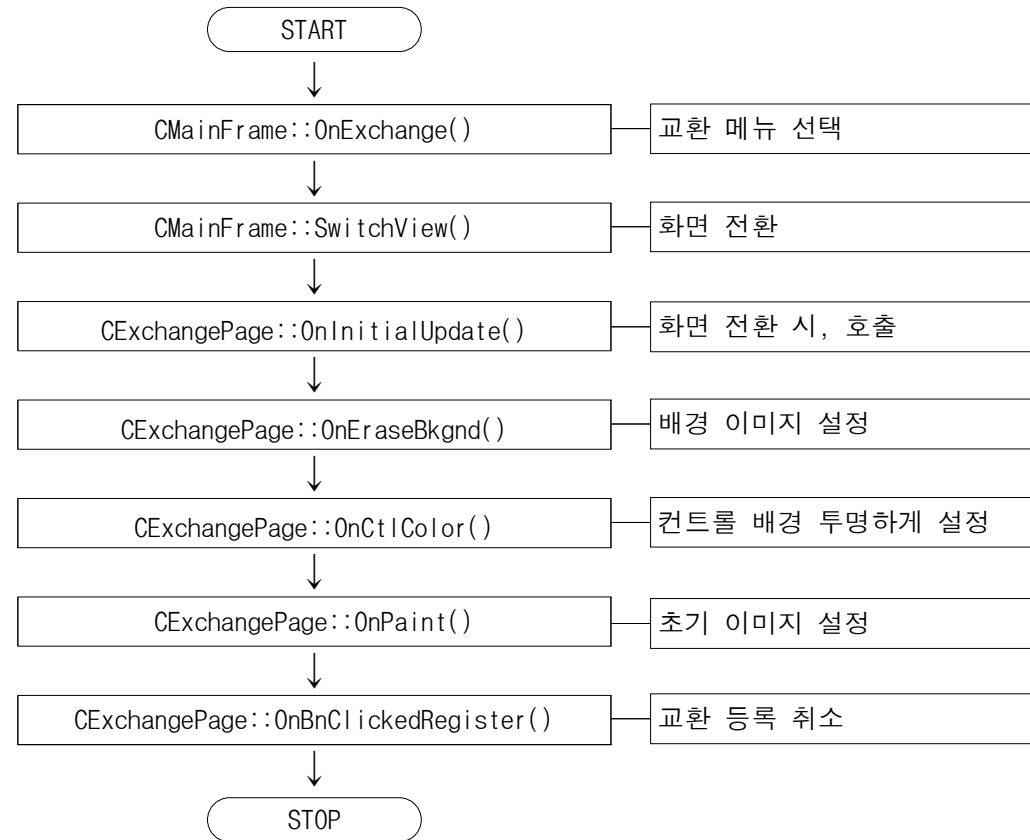
판매 - 판매 조회\_초기화



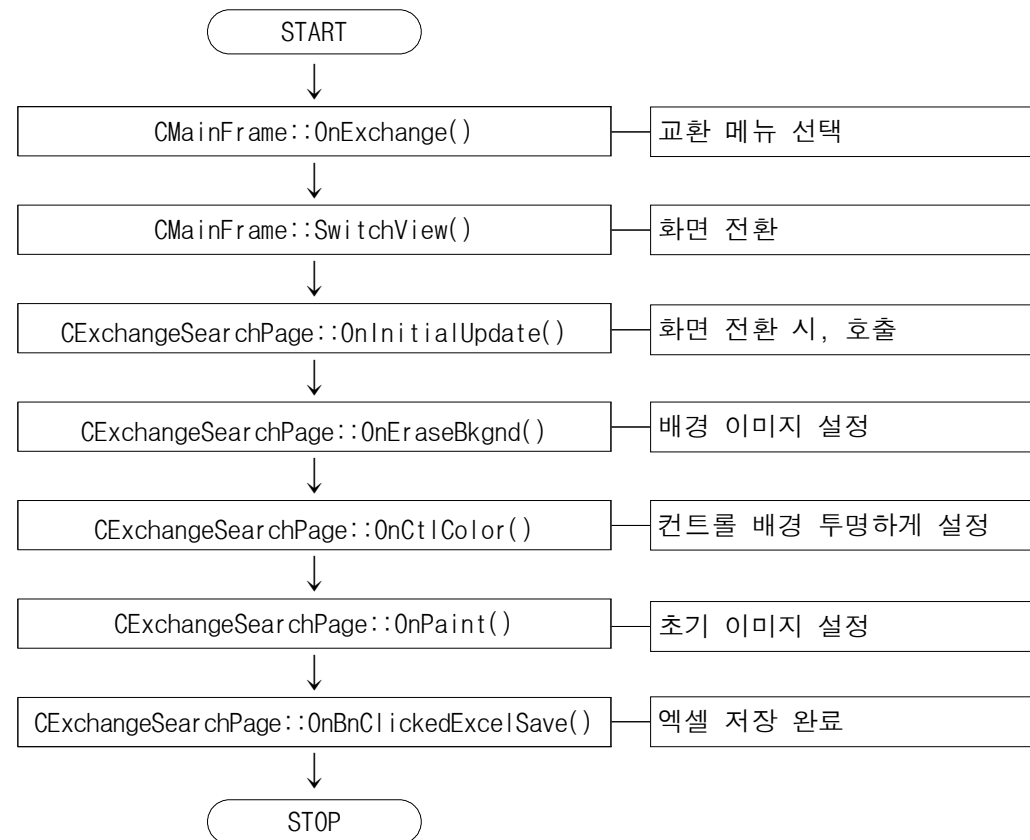
교환 - 교환 등록



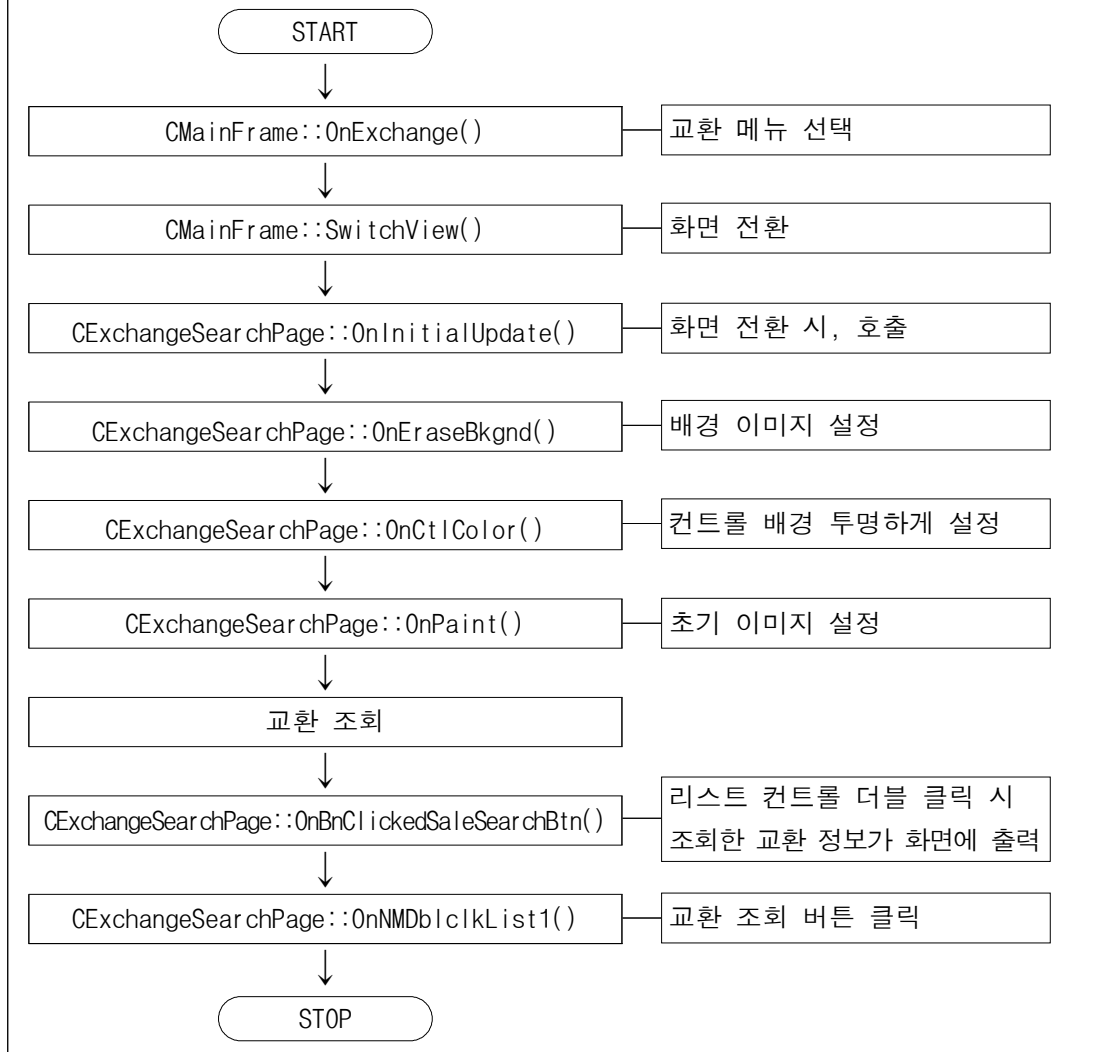
교환 - 교환등록\_취소



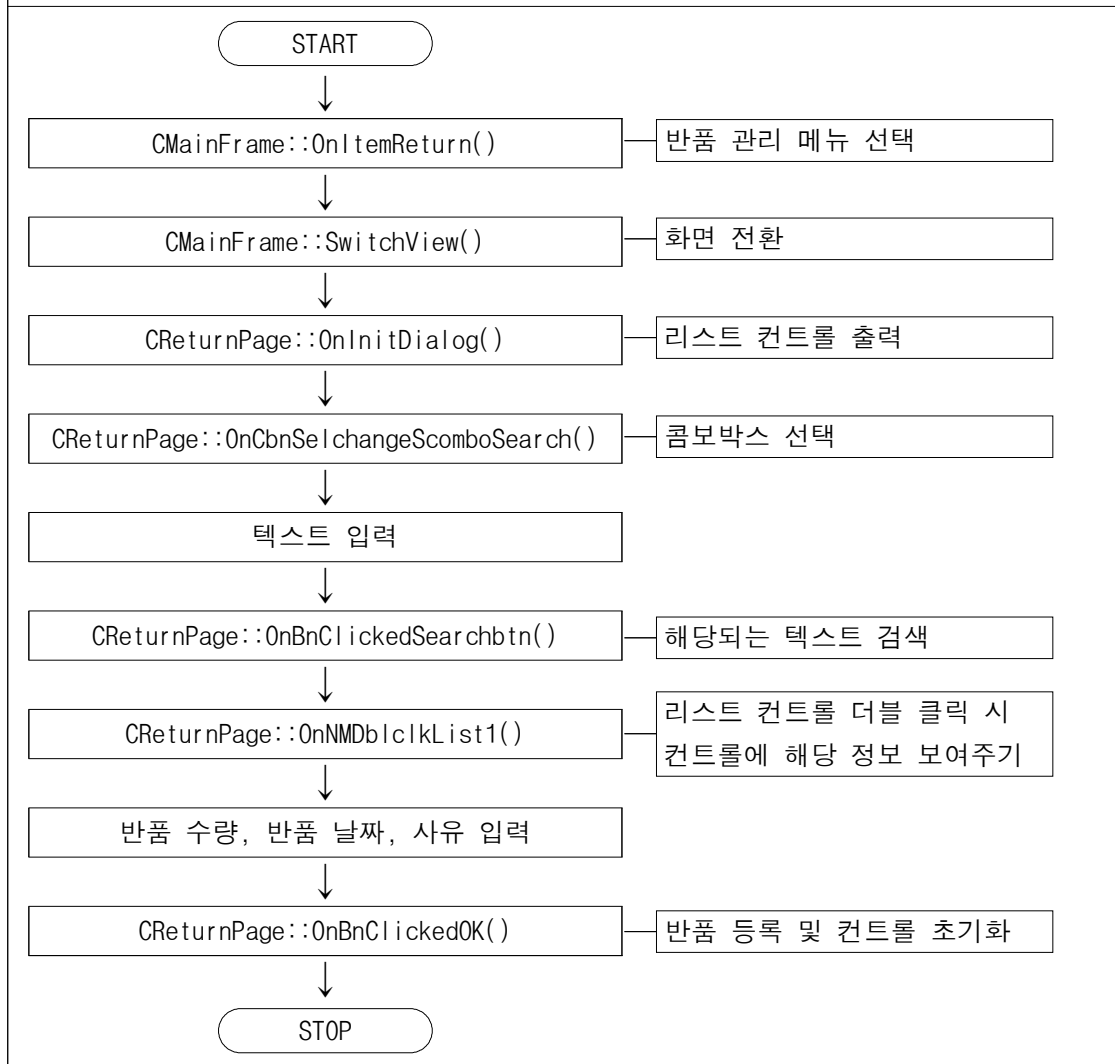
교환 - 교환 조회\_엑셀 저장



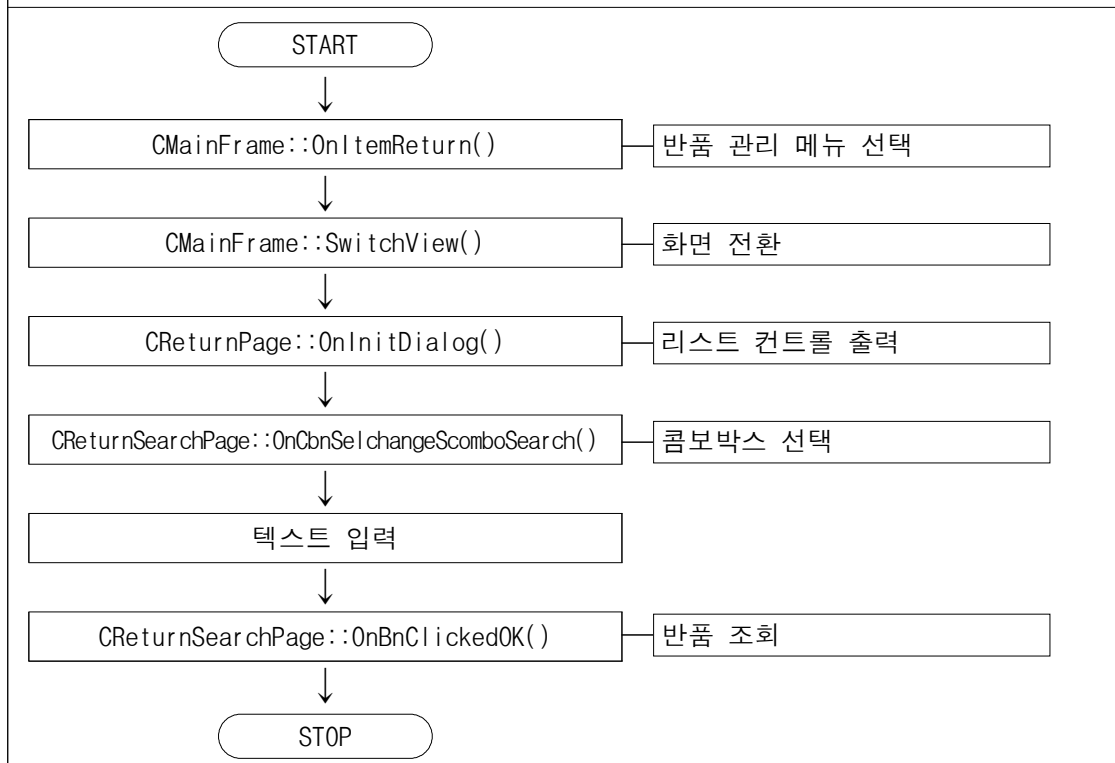
교환 - 교환 조회



반품 등록 - 등록

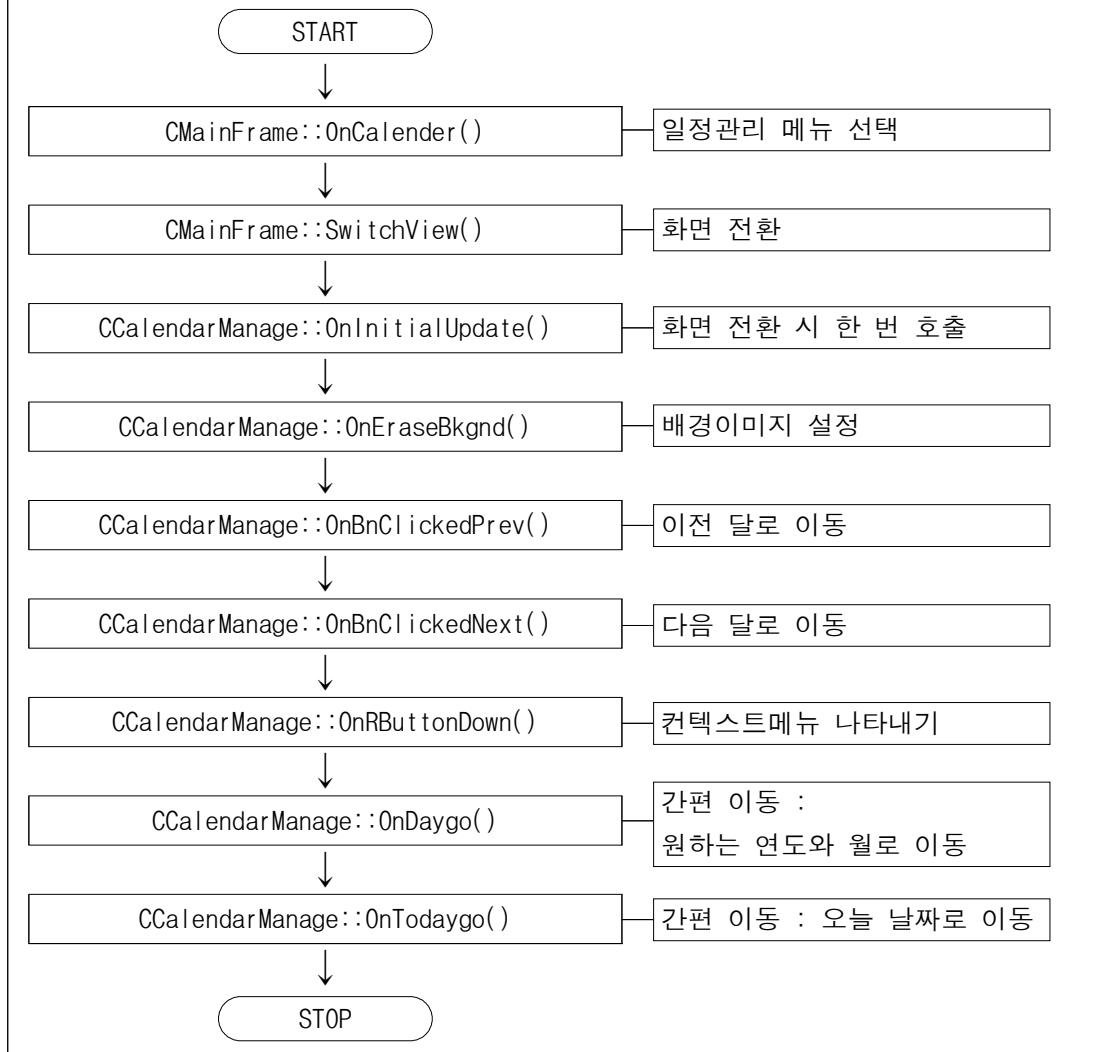


반품 조회 - 조회

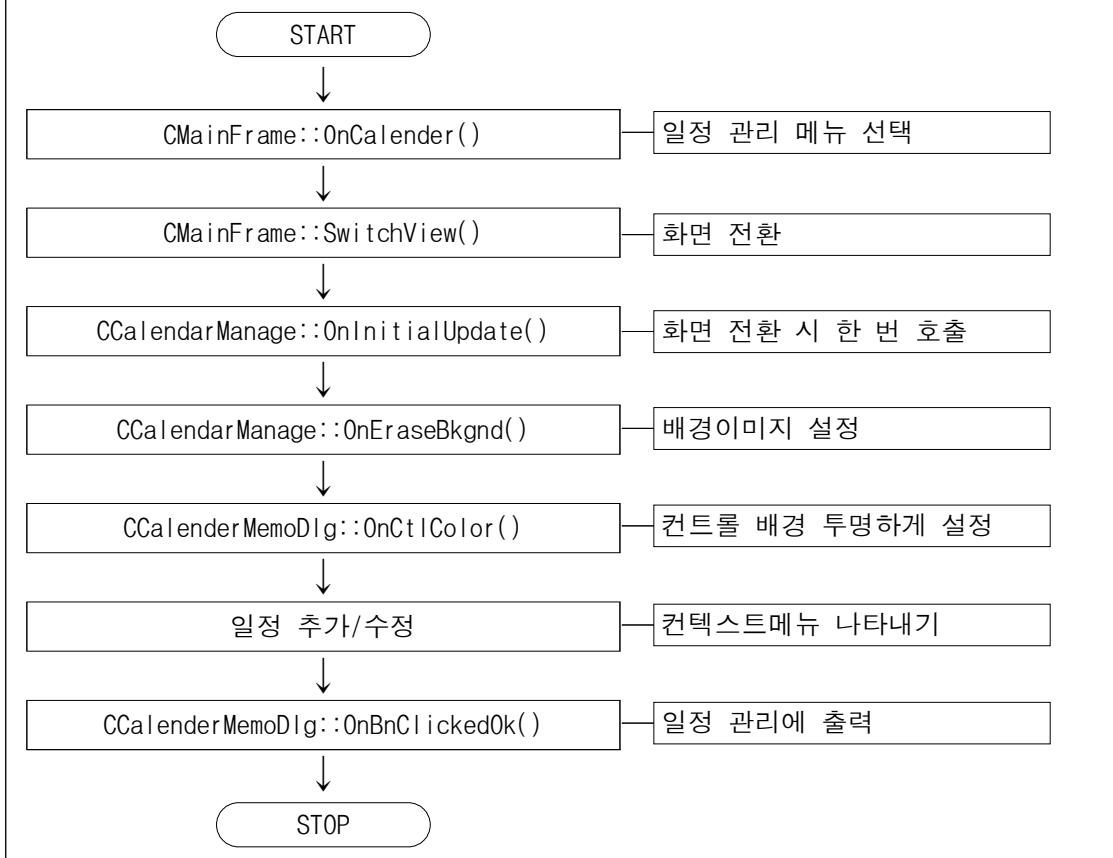




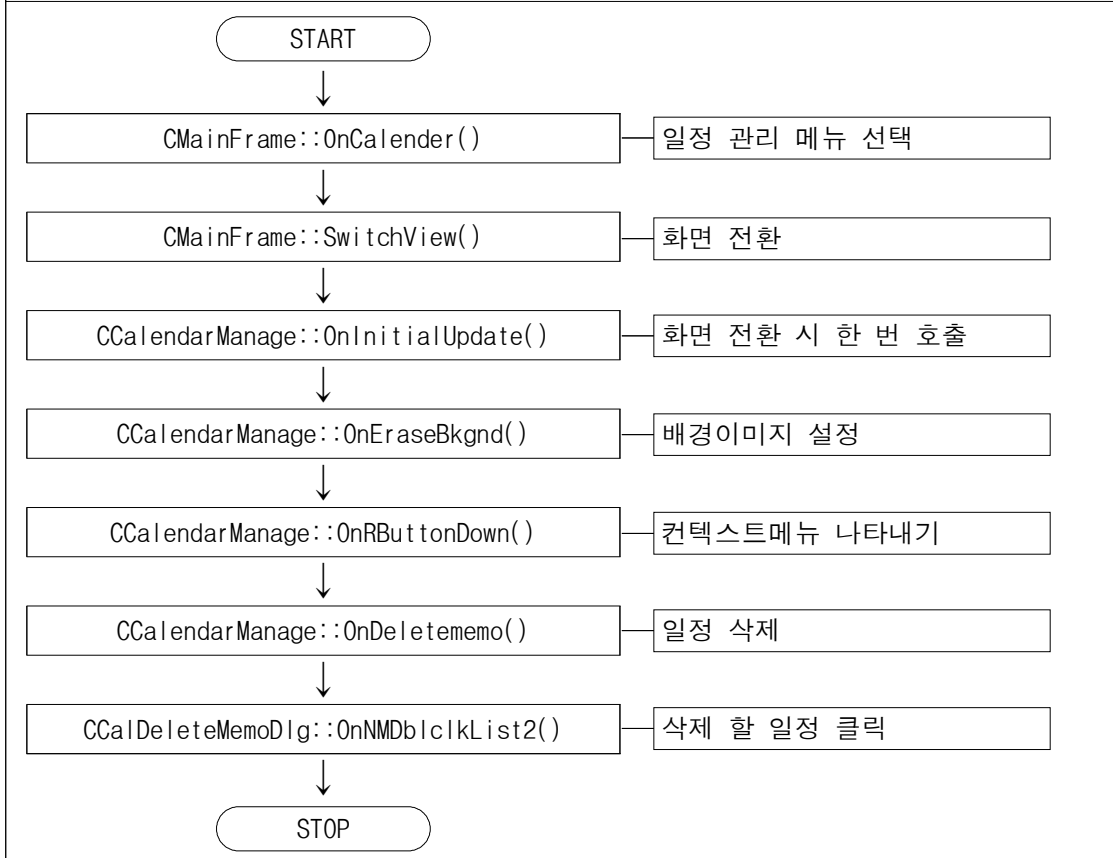
일정 관리 - 조회



일정 관리 - 추가/수정



일정 관리 - 삭제



### 3.3 기능(모듈)별 화면 및 주요내용



초기화면 : 프로그램의 로고와 로그인, 회원가입이 가능하다.



홈 화면 : 로그인 후 보이는 화면, 공지사항 확인이 가능하며, 일정관리, 자주 사용하는 메뉴는 버튼으로 배치하였다.

**ToyPocket**

홈 거래처관리 상품관리 판매관리 교환관리 반품관리 일정관리 환경설정

거래처 등록 거래처 조회 거래처 수정

**거래처**

사업자번호

거래처명

분류

업종

업태

거래처번호 02 -  -

주소

**대표**

대표자명

이메일

생일 2015-11-24

전화번호 010 -  -

**메모**

거래처 관리 - 거래처 등록 : 거래처를 등록할 수 있다.

**ToyPocket**

홈 거래처관리 상품관리 판매관리 교환관리 반품관리 일정관리 환경설정

거래처 등록 거래처 조회 거래처 수정

전체검색

담당자	거래처코드	거래처명	업체분류	업종	업태	대표자명	거래처번호	사업자번호	주소
seungki	OC_0010	로보카폴리	납품처	로봇	소매	김명관	02-224...	117-48...	경기도 의정부시...
kimji	OC_0009	Disney	납품처	강난감	도매	방운경	02-126...	112-15...	경기도 남양주시...
Jeongwoo	OC_0008	이마트	납품처	자동차	도매	김해성	02-158...	122-15...	서울시 중계 퇴계로
yuno	OC_0006	LOTTE	납품처	강난감	소매	신격호	02-157...	115-99...	서울시 중구 소공동
yoojoo0922	OC_0005	쿠광	납품처	감화	도매	김범석	02-157...	115-15...	서울 강남구 테헤란
minjungeya	OC_0004	투바니	납품처	강난감	도매	김관용	02-512...	110-48...	서울시 강남구 논현
hyejin0614	OC_0003	SAMSU...	납품처	감화	도매	이건의	02-158...	110-10...	경기도 수원시 영통
yeonjoo	OC_0002	아이코닉스	납품처	강난감	도매	최종일	02-156...	126-15...	경기도 성남시 분당
joonji12	IC_0010	LG	납입처	강난감	도매	구분무	02-157...	124-10...	서울시 영등포구...
kk762	IC_0009	쿠광	납입처	봉제	소매	김동희	02-154...	122-29...	서울 성동구 성수대
duslfqk	IC_0008	다음	납입처	보드	도매	김삼수	02-157...	177-77...	서울시 금천구 가산
jenny2103	IC_0007	타몬	납입처	인형	도매	신형성	02-154...	120-13...	서울 강남구 테헤란
onseun0908	IC_0006	G마켓	납입처	비디오	소매	김남수	02-156...	124-99...	서울시 강남구 테헤
tkseh88	IC_0005	GS리테일	납입처	퍼즐	소매	허중수	02-168...	170-29...	서울 강남구 논현로
in_jeong	IC_0004	옥션	납입처	강난감	도매	이성용	02-154...	102-10...	경기 부천시 원미구
mj_cho	IC_0003	HELLO ...	납입처	강난감	소매	조민재	02-117...	123-12...	서울시 강서구
super1234	IC_0002	다이소	납입처	감화	도매	박정부	02-124...	107-19...	서울 강남구 도곡동

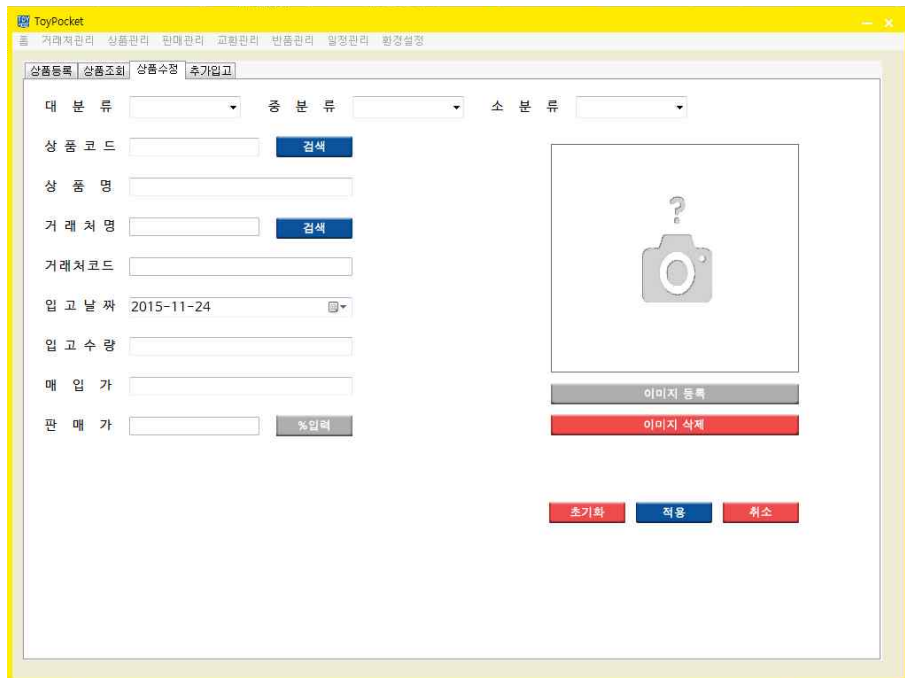
거래처 관리 - 거래처 조회 : 등록된 거래처들을 볼 수 있으며, 콤보박스의 선택 항목에 따라 원하는 거래처들의 정보를 확인 할 수 있으며 컨텍스트 메뉴를 통해 거래처 삭제가 가능하다. 또한 리스트 목록은 엑셀로 저장이 가능하며 수정하고 싶은 거래처 정보는 더블 클릭하면 거래처 수정 탭으로 이동된다.

**거래처 관리 - 거래처 수정** : 거래처 정보를 수정할 수 있다. 또한 거래처명 검색 버튼을 통해 원하고자 하는 해당 거래처의 정보를 가져올 수 있다.

**상품 관리 - 상품 등록** : 입고된 상품을 등록할 수 있으며 상품의 이미지 또한 저장 가능 하다. 그리고 거래처명으로 입고한 거래처 검색이 가능하다.



**상품 관리 - 상품 조회** : 입고된 상품의 정보를 볼 수 있으며, 콤보박스의 선택 항목에 따라 원하는 상품의 정보를 확인 할 수 있으며 컨텍스트 메뉴를 통해 상품 삭제가 가능하다. 또한 리스트 목록은 엑셀로 저장이 가능하며 수정하고 싶은 상품 정보는 더블클릭하면 상품 수정 탭으로 이동된다.



**상품 관리 - 상품 수정** : 상품 정보를 수정할 수 있다. 또한 상품코드, 거래처명 검색 버튼을 통해 원하고자 하는 해당 상품 코드, 거래처의 정보를 가져올 수 있다.

ToyPocket

거래처관리 상품관리 판매관리 교환관리 반품관리 일정관리 환경설정

상품등록 상품조회 상품수정 **추가입고**

상품명  **검색**
 매입가

상품코드 
 판매가

거래처명 
 입고날짜

거래처코드 
 입고수량

대분류  중분류  소분류

추가입고수량

**적용** **취소**

**상품 관리 - 추가 입고** : 이미 입고된 상품이 추가로 수량이 입고되는 경우 상품명 검색을 통해 정보를 불러오고 추가입고수량만 작성하면 추가입고가 가능하다.

ToyPocket

거래처관리 상품관리 판매관리 교환관리 반품관리 일정관리 환경설정

판매등록 **판매조회**

< 상품조회 >

전체검색  **조회**

담당자	상품코드	상품명	대분류	중분류	소분류	거래처명	거래처코드	입고일	입고수량
hyejin0614	b0052	오르골 부...	신생아/...	신생아	치발기/...	티몬	IC_0007	2015-1...	101
hyejin0614	b0051	오르골 육...	신생아/...	신생아	치발기/...	티몬	IC_0007	2015-1...	134
hyejin0614	b0050	플레이 글...	신생아/...	신생아	치발기/...	티몬	IC_0007	2015-1...	140
hyejin0614	b0049	아기사자...	신생아/...	신생아	치발기/...	티몬	IC_0007	2015-1...	140
hyejin0614	b0048	팔스네아...	신생아/...	신생아	치발기/...	티몬	IC_0007	2015-1...	210
hyejin0614	b0047	사파리 피...	신생아/...	신생아	치발기/...	다움	IC_0008	2015-1...	400

대분류  중분류  소분류

상품코드

상품명

현재재고수량

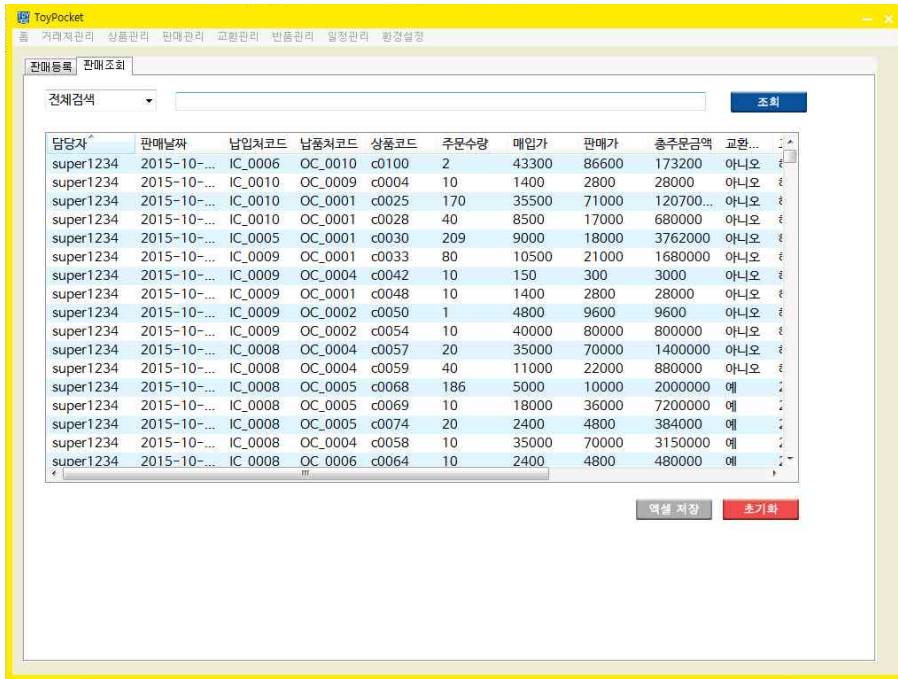
판매거래처  **조회**

판매날짜 2015-11-24

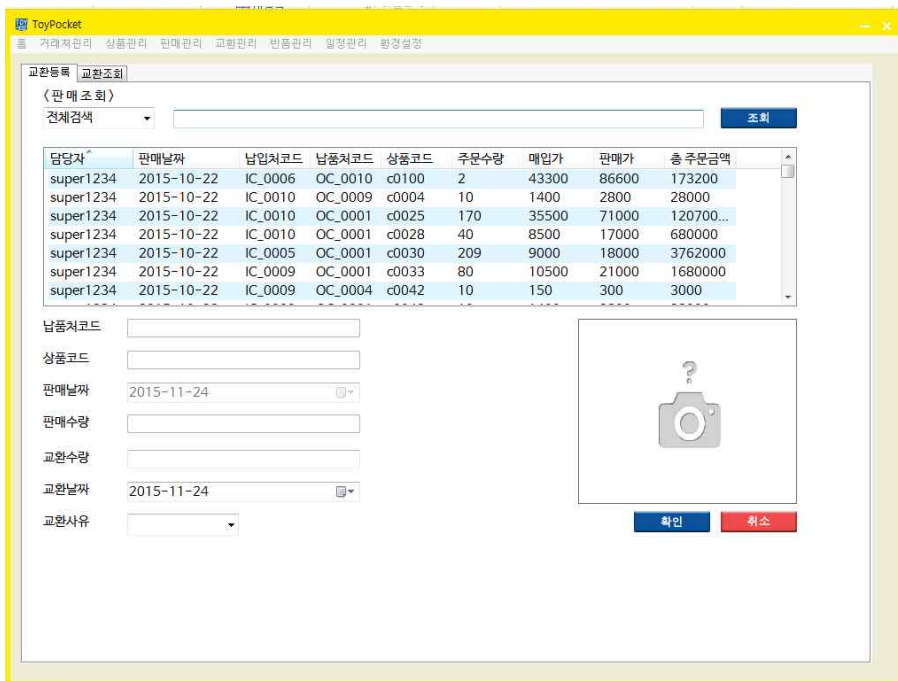
판매수량

**확인** **취소**

**판매 관리 - 판매 등록** : 입고된 상품을 검색을 통해 원하는 정보를 불러오고 판매 거래처, 판매 날짜, 판매 수량만 입력하면 판매 등록이 가능하다.



**판매 관리 - 판매 조회** : 판매된 상품의 정보를 볼 수 있으며, 콤보박스의 선택 항목에 따라 원하는 판매 상품의 정보를 확인 할 수 있으며 컨텍스트 메뉴를 통해 판매 정보 삭제가 가능하다. 또한 리스트 목록은 엑셀로 저장이 가능하다.



**교환 관리 - 교환 등록** : 판매한 상품을 검색을 통해 원하는 정보를 불러오고 교환 수량, 교환 날짜, 교환 사유만 입력하면 교환 등록이 가능하다.



ToyPocket

거래처관리 상품관리 판매관리 교환관리 반품관리 일정관리 환경설정

교환등록 교환조회

전체검색

조회

담당자	판매날짜	납품처코드	상품코드	교환수량	교환날짜	교환이유	매입가	판매가	총주문가
super1234	2015-10-22	OC_0005	c0068	186	2015-1...	단순반품	5000	10000	20000
super1234	2015-10-22	OC_0005	c0069	10	2015-1...	단순반품	18000	36000	72000
super1234	2015-10-22	OC_0005	c0074	20	2015-1...	불량	2400	4800	38400
super1234	2015-10-22	OC_0004	c0058	10	2015-1...	불량	35000	70000	31500
super1234	2015-10-22	OC_0006	c0064	10	2015-1...	불량	2400	4800	48000
super1234	2015-10-22	OC_0006	c0067	1	2015-1...	불량	3000	6000	18000
super1234	2015-10-22	OC_0005	c0072	10	2015-1...	부로로와...	3000	6000	48000
super1234	2015-10-22	OC_0005	c0075	40	2015-1...	불량	2900	5800	46400
super1234	2015-10-22	OC_0004	c0076	20	2015-1...	불량	32000	64000	76800
super1234	2015-10-22	OC_0004	c0079	5	2015-1...	단순반품	34000	68000	71400
super1234	2015-10-22	OC_0006	c0080	2	2015-1...	불량	3400	6800	13600
super1234	2015-10-22	OC_0006	c0081	40	2015-1...	불량	2600	5200	10400
super1234	2015-10-22	OC_0002	c0133	1	2015-1...	불량	8400	16800	84000
super1234	2015-10-22	OC_0002	c0130	40	2015-1...	부로로와...	8400	16800	33600
super1234	2015-10-22	OC_0002	c0129	1	2015-1...	불량	8400	16800	67200
super1234	2015-10-22	OC_0002	c0126	10	2015-1...	불량	8400	16800	20160
super1234	2015-10-22	OC_0002	c0123	139	2015-1...	단순반품	8400	16800	25200

엑셀 저장 초기화

**교환 관리 - 교환 조회** : 교환한 상품의 정보를 볼 수 있으며, 콤보박스의 선택 항목에 따라 원하는 교환 상품의 정보를 확인 할 수 있으며 컨텍스트 메뉴를 통해 교환 정보 삭제가 가능하다. 또한 리스트 목록은 엑셀로 저장이 가능하다.

ToyPocket

거래처관리 상품관리 판매관리 교환관리 반품관리 일정관리 환경설정

반품등록 반품조회

< 판매 조회 >

전체검색

조회

담당자	판매날짜	납품처코드	상품코드	주문수량	매입가	판매가	총주문금액	입고수량
tkseh88	2015-10-22	IC_0003	OC_0003	c0200	10	13000	260000	10
tkseh88	2015-10-22	IC_0003	OC_0003	c0199	140	14900	298000	190
tkseh88	2015-10-22	IC_0003	OC_0003	c0198	290	15600	312000	20
tkseh88	2015-10-22	IC_0005	OC_0003	c0196	120	6800	136000	40
tkseh88	2015-10-22	IC_0005	OC_0003	c0195	60	9000	180000	60
tkseh88	2015-10-22	IC_0005	OC_0003	c0194	140	6800	136000	10

거래처코드

상품코드

판매날짜 2015-11-24

판매수량

반품수량

반품날짜 2015-11-24

반품사유

확인 취소

**반품 관리 - 반품 등록** : 판매한 상품을 검색을 통해 원하는 정보를 불러오고 반품 수량, 반품 날짜, 반품 사유만 입력하면 반품 등록이 가능하다.

ToyPocket

홈 | 거래처관리 | 상품관리 | 판매관리 | 교환관리 | 반품관리 | 일정관리 | 환경설정

반품등록 | 반품조회

전체검색

조회

담당자	판매날짜	납입처코드	납품처코드	상품코드	반품수량	반품날짜	반품이유	매입가	판매가
super1234	2015-10-22	IC_0006	OC_0010	c0100	2	2015-1...	불량	43300	86600
super1234	2015-10-22	IC_0010	OC_0009	c0004	10	2015-1...	불량	1400	2800
super1234	2015-10-22	IC_0010	OC_0001	c0025	170	2015-1...	단순변심	35500	71000
super1234	2015-10-22	IC_0010	OC_0001	c0028	40	2015-1...	불량	8500	17000
super1234	2015-10-22	IC_0005	OC_0001	c0030	209	2015-1...	단순변심	9000	18000
super1234	2015-10-22	IC_0009	OC_0001	c0033	80	2015-1...	단순변심	10500	21000
super1234	2015-10-22	IC_0009	OC_0004	c0042	10	2015-1...	불량	150	300
super1234	2015-10-22	IC_0009	OC_0001	c0048	10	2015-1...	불량	1400	2800
super1234	2015-10-22	IC_0009	OC_0002	c0050	1	2015-1...	불량	4800	9600
super1234	2015-10-22	IC_0009	OC_0002	c0054	10	2015-1...	단순변심	40000	80000
super1234	2015-10-22	IC_0008	OC_0004	c0057	20	2015-1...	불량	35000	70000
super1234	2015-10-22	IC_0008	OC_0004	c0059	40	2015-1...	불량	11000	22000
super1234	2015-10-22	IC_0008	OC_0009	c0063	20	2015-1...	불량	1000	2000
super1234	2015-10-22	IC_0008	OC_0006	c0066	10	2015-1...	단순변심	2000	4000
super1234	2015-10-22	IC_0008	OC_0005	c0070	10	2015-1...	단순변심	13500	27000
super1234	2015-10-22	IC_0008	OC_0005	c0068	4	2015-1...	불량	5000	10000
super1234	2015-10-22	IC_0006	OC_0002	c0110	2	2015-1...	불량	1400	2800

엑셀 저장 | 초기화

**반품 관리 - 반품 조회** : 반품한 상품의 정보를 볼 수 있으며, 콤보박스의 선택 항목에 따라 원하는 반품 상품의 정보를 확인 할 수 있으며 컨텍스트 메뉴를 통해 반품 정보 삭제가 가능하다. 또한 리스트 목록은 엑셀로 저장이 가능하다.

ToyPocket

홈 | 거래처관리 | 상품관리 | 판매관리 | 교환관리 | 반품관리 | 일정관리 | 환경설정

2015 년도 11 월

일	월	화	수	목	금	토
1	2 ● 정기휴무	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18 ● 야근 없...	19	20	21
22	23	24	25	26	27	28 ● 체육대회...
29	30					

**일정 관리** : 일정 등록이 가능하며 컨텍스트 메뉴를 통해 원하는 날짜로 이동, 일정 삭제, 오늘 날짜로 이동이 가능하다.

**환경설정 - 관리자 페이지 - 권한/사용여부** : 최상위 관리자에 의해 등록되어 있는 직원들의 권한, 사용여부 변경이 가능하다.

날 짜	공지내용
2015-10-22	[10/29] 신입사원 OT입니다. 모두 참석 바랍니다.
2015-10-01	10월 첫째주 금요일은 정기 휴무입니다.

**환경설정 - 관리자 페이지 - 공지사항 등록** : 최상위 관리자에 의해 공지사항을 등록, 수정, 삭제가 가능하다.

**환경설정 - 비밀번호 변경** : 현재 로그인 되어 있는 직원의 비밀번호 변경을 할 수 있는 화면이다.

### 3.4 기능(모듈)별 클래스 목록 및 설명

	클래스	기능
거래처관리	CCompanyFormView	프로퍼티 시트를 넣는 클래스
	CCompanyModifyPage	거래처 정보를 수정하는 클래스
	CCompanyPage	거래처 정보를 등록하는 클래스
	CCompanySet	Company 테이블을 연결시켜주는 클래스
	CCompanySheet	거래처 등록, 조회, 수정을 하는 프로퍼티 페이지가 들어있는 클래스
	CCompanySearchPage	거래처 정보를 조회하는 클래스
상품관리	CItemPage	상품 정보를 등록하는 클래스
	CItemFormView	프로퍼티 시트를 넣는 클래스
	CItemModifyPage	상품 정보를 수정하는 클래스
	CItemPlusPage	상품 추가 입고 정보를 등록하는 클래스
	CItemSearchPage	상품 정보를 조회하는 클래스
	CItemSet	Item 테이블을 연결시켜주는 클래스
	CItemSheetDlg	상품 등록, 조회, 수정을 하는 프로퍼티 페이지가 들어있는 클래스
	CCompanySearchDlg	거래처 정보를 조회하는 다이얼로그 클래스
	CItemSearchDlg	상품 정보를 조회하는 다이얼로그 클래스
판매관리	CSaleFormView	프로퍼티 시트를 넣는 클래스
	CSaleModifyPage	판매 정보를 수정하는 클래스
	CSalePage	판매 정보를 등록하는 클래스
	CSaleSearchPage	판매 정보를 조회하는 클래스
	CSaleSet	Sale 테이블을 연결시켜주는 클래스
	CSaleSheet	판매 등록, 조회를 하는 프로퍼티 페이지가 들어있는 클래스
	CSaleComSearchDlg	거래처 정보를 조회하는 다이얼로그 클래스
교환관리	CExchangeFormView	프로퍼티 시트를 넣는 클래스
	CExchangePage	교환 정보를 등록하는 클래스
	CExchangeSearchPage	교환 정보를 조회하는 클래스
	CExchangeSheet	교환 등록, 조회를 하는 프로퍼티 페이지가 들어있는 클래스
반품관리	CReturnFormView	프로퍼티 시트를 넣는 클래스
	CReturnPage	반품 정보를 등록하는 클래스
	CReturnSearchPage	반품 정보를 조회하는 클래스
	CReturnSheet	반품 등록, 조회를 하는 프로퍼티 페이지가 들어있는 클래스

	클래스	기능
일정관리	CCalDeleteMemoDlg	일정 삭제를 하는 다이얼로그 클래스
	CCalendarManage	달력에 일정 등록, 삭제, 이동을 하는 클래스
	CCalendarSet	Calendar 테이블을 연결시켜주는 클래스
	CCalenderMemoDlg	일정 등록을 하는 다이얼로그 클래스
	CCalTimeGo	일정 이동을 하는 다이얼로그 클래스
환경설정	CPasswordChangeDlg	비밀번호 변경을 하는 다이얼로그 클래스
	CNoticeDlg	공지사항을 등록/수정/삭제를 하는 다이얼로그 클래스
	CManagerPermitDlg	사용자에게 권한을 부여/변경하는 다이얼로그 클래스
	CRegisterIdSearchDlg	사용자에 대한 정보를 조회하는 다이얼로그 클래스
기타기능	CHomepageFormView	로그인 후 보이는 메인화면을 나타내는 클래스
	CRegisterFormView	회원가입을 하는 클래스
	CToyPocketView	로그인하는 클래스
	CToyPocketSet	User 테이블을 연결시켜주는 클래스
	CXLAutomation	엑셀 자동화 클래스
	CXLEzAutomation	Wrapper 클래스
	CCustomStatusBar	프로그램의 상태표시줄을 사용자 임의로 그리고 설정하는 클래스
	CMfcSkinSDI	프로그램의 창 전체와 세부부분을 사용자 임의로 그리고 설정하는 클래스

### 3.5 기능(모듈)별 함수 목록 및 설명

클래스	함수	설명
CCompanyFormView	OnInitialUpdate()	거래처 관리 창 열었을 때 한 번 호출
	OnEraseBkgnd()	배경 이미지 설정
	OnCtlColor()	컨트롤 배경 변경
CCompanyPage	OnInitDialog()	거래처 등록 창 열었을 때 한번 호출, 버튼 이미지, 글씨체 설정
	OnBnClickedComAdd()	거래처 등록
	OnBnClickedCancel()	취소 화면 버튼 클릭 시 초기 화면으로 돌아가기
CCompanySearchPage	OnInitDialog()	거래처 조회 창 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnBnClickedSearchbtn()	콤보박스와 값 입력에 의한 해당 정보 보여주기
	OnNMDblclkList1()	수정하고자 하는 정보 더블클릭 시 거래처 수정 탭으로 이동
	OnNMRClickList1()	삭제하고자 하는 정보 오른쪽클릭 시 거래처 삭제 메뉴 보이기
	OnCompanyDelete()	컨텍스트 메뉴 거래처 삭제 메뉴를 클릭했을 때 해당하는 거래처 정보 삭제
	OnBnClickedExcelSave()	리스트 목록을 엑셀로 저장
	OnCustomdrawList()	리스트의 행 배경 설정
	OnBnClickedClear()	값 초기화
	ReInitDialog()	CompanyPage.cpp에 의해 갱신되어 보임
	OnHdnItemclickList1()	정렬하고자 하는 필드 선택
CCompanyModifyPage	OnInitDialog()	거래처 수정 창 열었을 때 한번 호출, 버튼 이미지, 글씨체 설정
	OnBnClickedComSearchBtn()	조회 버튼 클릭 시 조회 탭으로 이동
	OnBnClickedButtonClear()	값 초기화
	OnBnClickedButtonModi()	거래처 수정
CCompanySearchDlg	OnInitDialog()	거래처 조회 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnBnClickedOk()	조회 버튼 클릭 시 해당하는 정보를 보여줌
	OnHdnItemclickList1()	정렬하고자 하는 필드 선택
	OnNMDblclkList1()	수정하고자 하는 정보 더블클릭 시 거래처 수정 탭으로 이동
	OnCustomdrawList()	리스트의 행 배경 설정
CItemFormView	OnInitialUpdate()	상품 관리 창 열었을 때 한번 호출
	OnEraseBkgnd()	배경색 설정

클래스	함수	설명
CItemPage	OnInitDialog()	상품 등록 창 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnBnClickedOk()	상품 등록
	SelectSmall()	대분류 문자열 반환
	SelectMedium()	중분류 문자열 반환
	SelectLarge()	소분류 문자열 반환
	OnCbnSelChangeCombo2()	대분류 콤보박스 선택 시 중분류, 소분류 항목 설정
	OnCbnSelChangeCombo4()	중분류 콤보박스 선택 시 소분류 항목 설정
	DrawBMP()	이미지 보여주기
	OnPaint()	상품 초기 이미지 설정
	OnBnClickedPercent()	매입가의 %값을 판매가로 등록
	bigmidsmall()	카테고리 문자열 반환
	OnBnClickedCancel2()	이미지 등록
	OnBnClickedImageDelete()	이미지 삭제
CItemSearchPage	OnInitDialog()	상품 조회 창 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnBnClickedItemSearchbtn()	조회 버튼 클릭 시 해당하는 정보를 보여줌
	OnHdnItemClickList1()	정렬하고자 하는 필드 선택
	OnNMDBlckList1()	수정하고자 하는 정보 더블클릭 시 거래처 수정 탭으로 이동
	OnBnClickedExcelSave()	리스트 목록을 엑셀로 저장
	OnCustomdrawList()	리스트의 행 배경 설정
	OnBnClickedClear()	값 초기화
CItemModifyPage	OnInitDialog()	상품 수정 창 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnBnClickedMdIItemSearch()	클릭 시 아이템 조회 탭으로 이동
	OnBnClickedModifyClear()	값 초기화
	OnCbnSelChangeCombo2()	대분류 콤보박스 선택 시 중분류, 소분류 항목 설정
	OnCbnSelChangeCombo4()	중분류 콤보박스 선택 시 소분류 항목 설정
	OnBnClickedImageSearch()	이미지 등록 이벤트 처리기
	OnBnClickedItemModify()	아이템 수정 이벤트 처리기
	SelectSmall()	대분류 문자열 반환
	SelectMedium()	중분류 문자열 반환
	SelectLarge()	소분류 문자열 반환
	DrawBMP()	이미지 보여주기
	bigmidsmall()	카테고리 문자열 반환
	gbigmidsmall()	카테고리 문자열 반환 (전역변수 이용 함수)
	OnBnClickedPercent()	매입가의 %값을 판매가로 등록
	OnBnClickedComSearchBtn()	거래처명 조회 버튼 이벤트 처리기
	OnBnClickedImageCancel()	이미지 삭제
	OnPaint()	상품 초기 이미지 설정

클래스	함수	설명
CItemPlusPage	OnInitDialog()	추가 입고 창 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnBnClickedItemPlus()	적용 버튼 이벤트처리기
	OnBnClickedSearchButton()	상품명 검색 이벤트처리기
CItemPercentDlg	OnInitDialog()	%입력 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnBnClickedPercent()	적용버튼 이벤트 처리기
	OnBnClickedCancel()	취소 버튼 이벤트 처리기
CItemSearchDlg	OnInitDialog()	상품조회 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnBnClickedSearchButton()	조회 버튼 이벤트 처리기
	OnHdnItemclickList1()	정렬하고자 하는 필드 선택
	OnNMDBlckList1()	불러올 상품을 더블클릭하면 창이 사라지고 값이 입력되어 들어옴
	OnCustomdrawList()	리스트의 행 배경 설정
CToyPocketView	OnBnClickedLogin()	사용자가 입력한 아이디와 비밀번호가 DB에 저장되어 있는 사용자의 정보와 일치한지 비교하고 일치하면 로그인, 아니면 로그인 실패 창을 띄움
	OnPaint()	로고 비트맵 그리기
	OnEraseBkgnd()	배경색 설정
	OnCtlColor()	각 컨트롤들의 배경 색 설정
	PreTranslateMessage()	아이디와 비밀번호를 입력 후 엔터키를 클릭하면 로그인 여부를 알 수 있음
	OnBnClickedRegisterBtn()	회원가입 창으로 변경
CNoticeDlg	OnInitDialog()	공지사항 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnHdnItemclickList1()	정렬하고자 하는 필드 선택
	OnNMDBlckList1()	수정하고자 하는 정보 더블클릭 시 에디트 박스에 더블클릭 한 공지 사항 내용이 불러 들어옴
	OnNMRClickList1()	수정하고자 하는 공지 사항 내용을 오른쪽 클릭 시 내용 수정 메뉴 보이기
	OnBnClickedAddBtn()	플러스 버튼 이벤트 처리기
	OnBnClickedDeleteBtn()	마이너스 버튼 이벤트 처리기 (공지사항 삭제)
	OnBnClickedBModiBtn()	수정 버튼 이벤트 처리기
	OnNoticeModi()	내용 수정 메뉴 클릭 시 이벤트처리기
	OnEraseBkgnd()	배경색 설정
	OnCustomdrawList()	리스트의 행 배경 설정
	OnNMClickList1()	현재 클릭한 값을 알 수 있음



클래스	함수	설명
CManagePermitDlg	OnInitDialog()	권한/사용여부 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnEraseBkgnd()	배경색 설정
	OnCtlColor()	각 컨트롤들의 배경 색 설정
	OnBnClickedOk()	확인 버튼 이벤트 처리기
	OnRadioButton()	체크한 라디오 값을 불러와 임의의 변수(strRadio)에 저장
	OnBnClickedButton1()	아이디 검색 버튼 클릭 시 아이디 찾기 다이얼로그를 띄움
CPassowrdChangeDlg	OnInitDialog()	비밀번호 변경 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnEraseBkgnd()	배경색 설정
	OnCtlColor()	각 컨트롤들의 배경 색 설정
	OnBnClickedOk()	현재비밀번호와 입력한 현재 비밀번호 값이 일치한지 확인 후 변경 할 비밀번호로 사용자의 비밀번호를 변경시킴
CCalendarManage	OnInitUpdate()	일정 관리 창을 열었을 때 글씨체, 비트맵, rect배열 값 설정
	OnPaint()	달력 그리기, DB에 저장되어 있는 값 불러와서 그림
	OnBnClickedPrev()	이전 달 보여줌
	OnBnClickedNext()	다음 달 보여줌
	OnLButtonDown()	CCalenderMemoDlg 다이얼로그 띄우고 일정 그림
	OnRButtonDown()	컨텍스트 메뉴 띄움
	OnDaygo()	원하는 일정으로 이동함
	OnDeletememo()	CCalDeleteMemoDlg 다이얼로그 띄움
	OnTodaygo()	오늘 날짜로 이동
	OnEraseBkgnd()	property page 다이얼로그 배경색 설정
	OnCtlColor()	static text 배경 투명으로 설정
CCalDeleteMemoDlg	OnInitDialog()	리스트 열 이름 설정
	OnPaint()	클릭 한 곳의 년도와 날짜가 일치 하는 값을 DB에서 불러와 띄움
	OnNMDBlckList2()	컨텍스트 메뉴 일정 삭제 메뉴를 클릭했을 때 해당하는 일정 삭제
	OnEraseBkgnd()	property page 다이얼로그 배경색 설정
CCalenderMemoDlg	OnInitDialog()	DB에 저장되어 있는 일정정보를 띄움
	OnPaint()	글씨체 설정
	OnEraseBkgnd(CDC* pDC)	property page 다이얼로그 배경색 설정
	OnCtlColor()	static text 배경 투명으로 설정

클래스	함수	설명
CCalTimeGo	OnStnClickedLayear()	클릭시 현재 년도보다 -1됨
	OnStnClickedLamonth()	클릭시 현재 월보다 -1됨
	OnStnClickedNeyear()	클릭시 현재 년도보다 +1됨
	OnStnClickedNemonth()	클릭시 현재 월보다 +1됨
	OnInitDialog()	글씨체 설정
	OnPaint()	년도, 월 텍스트 설정
	OnCtlColor()	글씨체 설정
	OnEraseBkgnd()	property page 다이얼로그 배경색 설정
CExchangeFormView	OnInitDialog()	교환 관리 창 열었을 때 한번 호출
	OnEraseBkgnd()	property page 다이얼로그 배경색 설정
CExchangeSearchPage	OnCustomdrawList()	리스트의 행 배경 설정
	OnInitDialog()	교환조회 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	CompareItem()	리스트 행 정렬 비교
	OnHdnItemclickList1()	정렬 함수 호출
	OnCbnSelchangeComboSearch()	조건에 맞을 때 텍스트 보여줌
	OnBnClickedSearchBtn()	조회 버튼 클릭 시 해당하는 정보를 보여줌
	OnNMRClickList1()	해당 행 클릭시 컨텍스트 메뉴 띄움
	OnExchangeDelete()	컨텍스트 메뉴 삭제 이벤트 처리
	OnBnClickedExcelSave()	리스트 목록을 엑셀로 저장
	OnBnClickedClear()	값 초기화
CExchangeModifyPage	OnInitDialog()	버튼 이미지 설정
CExchangePage	CompareItem()	리스트 행 정렬 비교
	OnInitDialog()	교환 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	SelectReason()	대분류 문자열 반환
	OnBnClickedOk()	교환된 정보가 DB에 등록
	OnBnClickedSearchbtn()	조회 버튼 클릭 시 해당하는 정보를 보여줌
	OnCustomdrawList()	리스트의 행 배경 설정
	OnCbnSelchangeScomboSearch()	조건에 맞을 때 텍스트 보여줌
	OnPaint()	기본이미지 설정
	OnNMDblclkList1()	판매정보 클릭 시 교환정보 Edit box에 설정됨
	OnCbnSelchangeCombo2()	조건에 맞을 때 텍스트 보여줌
	DrawBMP()	상품 이미지 보여주기
CReturnFormView	OnInitDialog()	반품 관리 창 열었을 때 한번 호출
	OnEraseBkgnd()	property page 다이얼로그 배경색 설정
CReturnModifyPage	OnInitDialog()	비트맵 설정

클래스	함수	설명
CReturnPage	CompareItem()	리스트 행 정렬 비교
	OnInitDialog()	반품 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnBnClickedSearchbtn()	조회 버튼 클릭 시 해당하는 정보를 보여줌
	OnCustomdrawList()	리스트의 행 배경 설정
	OnPaint()	기본 이미지 설정
	OnNMDBlckList1()	반품정보 클릭 시 교환정보 Edit box에 설정됨
	OnBnClickedOk()	반품된 정보가 DB에 등록
	SelectReason()	대분류 문자열 반환
	OnCbnSelChangeCombo2()	조건에 맞을 때 텍스트 보여줌
	DrawBMP()	상품 이미지 보여주기
CReturnSearchPage	OnCustomdrawList()	리스트의 행 배경 설정
	OnInitDialog()	반품조회 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnCbnSelChangeComboSearch()	조건에 맞을 때 텍스트 보여줌
	OnBnClickedOk()	불러올 상품을 더블클릭하면 창이 사라지고 값이 입력되어 들어옴
	OnReturnDelete()	컨텍스트 메뉴 삭제 이벤트 처리
	OnBnClickedExcelSave()	리스트 목록을 엑셀로 저장
	OnBnClickedClear()	값 초기화
CSaleFormView	OnInitInitialUpdate()	판매 관리 창 열었을 때 한번 호출
	OnEraseBkgnd()	property page 다이얼로그 배경색 설정
CSaleModifyPage	OnInitDialog()	판매수정 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnNMDBlckList1()	상품정보 클릭 시 교환정보 Edit box에 설정됨
	OnPaint()	조회탭의 정보를 Edit Box에 띄움
	OnBnClickedSalemodify()	판매 수정된 정보가 DB에 저장
CSalePage	OnInitDialog()	판매 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	CompareItem()	배경색 설정
	OnCbnSelChangeScomboSearch()	조건에 맞을 때 텍스트 보여줌
	OnBnClickedSearchbtn()	조회 버튼 클릭 시 해당하는 정보를 보여줌
	OnPaint()	기본 이미지 저장
	OnNMDBlckList1()	상품정보 클릭 시 교환정보 Edit box에 설정됨
	OnBnClickedRegister()	판매 정보가 DB에 저장
	DrawBMP()	상품 이미지 보여주기

클래스	함수	설명
CSaleSearchPage	OnCustomdrawList()	리스트의 행 배경 설정
	OnBnClickedSaleSearchBtn()	조회 버튼 클릭 시 해당하는 정보를 보여줌
	OnInitDialog()	판매 조회 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	CompareItem()	리스트 행 정렬 비교
	OnCbnSelChangeCombo1()	조건에 맞을 때 텍스트 보여줌
	OnNMRClickList1()	해당 행 클릭시 컨텍스트 메뉴 띄움
	OnSaleDelete()	컨텍스트 메뉴 삭제 이벤트 처리
	OnNMDBlckList1()	판매 조회탭에서 정보 클릭 시 판매 수정탭 Edit box에 설정됨
	OnBnClickedExcelSave()	리스트 목록을 엑셀로 저장
	OnBnClickedClear()	값 초기화
CSaleComSearchDlg	OnCustomdrawList()	리스트의 행 배경 설정
	OnBnClickedSearch()	조회 버튼 클릭 시 해당하는 정보를 보여줌
	OnNMDBlckList1()	해당 행 클릭시 Edit Box에 텍스트 띄움
	OnInitDialog()	판매관리에서 거래처 조회 다이얼로그를 열었을 때 한 번 호출, 버튼 이미지, 글씨체 설정
	OnPaint()	DB에 있는 정보 가져오기
	CompareItem()	리스트 행 정렬 비교
	OnEraseBkgrnd()	property page 다이얼로그 배경색 설정
CXLAutomation	StartExcel()	엑셀을 실행
	ExlInvoke()	엑셀 불러오기
	ClearVariant()	변형구조를 초기화
	ClearAllArgs()	인수배열초기화
	ReleaseVariant()	변형구조 클리어
	AddArgumentCommon()	인수 설정
	AddArgumentDispatch()	인수 설정
	AddArgumentOLEString()	OLEString타입의 argument 추가
	AddArgumentCString()	CString 타입의 argument추가
	SetCellsValueToString()	CELL값설정
	ReleaseExcel()	엑셀닫기
	CreateWorkSheet()	빈워크시트 생성
	CreateXYChart()	XY차트 생성
	SetRangeValueDouble()	범위지정
	AddArgumentCStringArray()	변형 문자열을 포함하는 1차원 배열을 생성
	UpdatePlotRange()	플롯에 새로운 포인트가 추가되면 플롯업데이트
	PasteStringToWorksheet()	클립보드에 문자열을 복사 후 워크시트에 붙여넣기
	ReleaseDispatch()	릴리스대스패치
	ShowException()	예외처리
	DeleteRow()	현재 워크시트의 열삭제
	SaveAs()	엑셀 파일 저장
	GetCellValueCString()	cell값 얻어옴
	InsertPictureToWorksheet()	그림 파일삽입
	PlaceImageToClipboard()	DIB그림삽입
	OpenExcelFile()	읽기를 위한 엑셀 파일 열기

클래스	함수	설명
CXLEzAutomation	CreateXYChart()	XY차트 생성
	UpdateXYChart()	XY차트 업데이트
	ExportCString()	클립보드 사용
	SetCellValue()	cell값 설정
	ReleaseExcel()	엑셀 닫기
	DeleteRow()	현재 워크시트의 열 닫기
	SaveFileAs()	엑셀 파일 저장
	GetCellValue()	cell값얻음
	InsertPictureFromFile()	그림 파일삽입
	InsertPictureToClipboard()	이미지를 클립보드에 복사
	PlacePictureToClipboard()	이미지를 클립보드에 복사
	OpenExcelFile()	읽기를 위한 엑셀 파일 열기
	InsertPictureFromBuffer()	버퍼에서 이미지 가져옴
CCustomStatusBar	OnPaint()	상태 표시줄 스타일 지정
	OnDrawItem()	사용자 지정의 컨트롤을 그리기위해 호출
	SetBackgroundColor()	상태 표시줄의 배경 색상 설정
	SetTextColor()	상태 표시줄의 글자 색상 설정
	GetBackgroundColor()	상태 표시줄의 현재 배경색을 가져옴
	CCustomStatusBar()	상태 표시줄의 배경색을 RGB()로 지정
	DrawPaneContent()	Toggle 키의 현재 상태를 얻어옴
CMfcSkinSDI	CMfcSkinSDI()	전체 스킨 및 부분 스킨의 색상을 RGB()로 지정
	~CMfcSkinSDI()	스킨 설정의 메모리를 설정함
	OnCreate()	기본 메뉴를 해제하거나 설정
	PreCreateWindow()	윈도우 창을 생성함
	OnNotify()	윈도우 창을 그리는 함수
	RecalcLayout()	레이아웃을 불러오는 함수
	tbSetBackColor()	툴바 배경 색상을 설정
	tbSetGripperColor()	툴바 그리퍼 색상을 설정
	tbSetGripperWidth()	툴바 그리퍼의 가로 길이를 설정
	tbSetStyle()	툴바의 테두리를 설정
	DrawBorder()	프레임 테두리를 그려줌
	borderSetDimension()	윈도우 테두리의 굵기를 설정
	borderSetColor()	테두리의 색상을 설정
	borderSetOutlineColor()	메인 프레임 윈도우의 아웃라인 테두리의 색상 설정
	borderEnableOutline()	테두리 아웃라인의 사용여부를 설정
	borderBlendStatusBar()	메인 프레임과 상태바 색상을 조합
	DrawCaptionBar()	캡션바를 그려줌
	DrawSystemButtons()	시스템 버튼 (닫기, 최대화, 최소화)을 생성
	HighlightSystemMenuButton()	시스템 버튼에 마우스를 올렸을 시 호출함
	SetCaptionTextColor()	캡션바의 텍스트 색상을 설정
	SetCaptionBackColor()	캡션바의 배경 설정

클래스	함수	설명
CMfcSkinSDI	SetCaptionText()	캡션바의 스타일 지정
	SetSystemBtnColor()	시스템 버튼 색상 설정
	SetSystemBtnHighlightColor()	시스템 버튼에 마우스를 올렸을 시 색상 변경 설정
	SetSystemBtnHighlightBkColor()	시스템 버튼에 마우스를 올렸을 시 배경 변경 설정
	ChangeMenuToCustomDraw()	사용자가 만든 메뉴를 적용하기 위해서 부름
	OnDrawItem()	메뉴 전체를 그림
	OnMeasureItem()	메뉴의 높이와 너비를 배치함
	DrawCustomMenuBar()	메뉴 상세를 그려줌
	SetMenuBarBottomMargin()	툴바와의 간격을 설정
	SetMenuBarBkColor()	메뉴의 배경색을 설정
	SetMenuBarHoverColor()	메뉴에 마우스를 올렸을 시 배경색 설정
	SetMenuBarSelectColor()	메뉴 선택 시 배경색 설정
	SetMenuTextColor()	메뉴의 텍스트 색상 설정
	CapMenuBarItems()	메뉴 아이템의 텍스트 대문자 사용여부 설정
	OnSysMenuMove()	시스템메뉴를 이동하는 함수
	OnSysMenuRestore()	시스템메뉴를 저장하는 함수
	OnSysMenuSize()	시스템메뉴의 사이즈를 지정하는 함수
	OnSysMenuMinimize()	시스템 메뉴 중 최소화 버튼 관련된 함수
	OnSysMenuMaximize()	시스템 메뉴 중 최대화 버튼 관련된 함수
	OnSysMenuClose()	시스템 메뉴 중 닫기 버튼 관련된 함수
	InitilizeCustomDraw()	사용자 설정을 초기화
	SetMainFrmResourceID()	메인 프레임 리소스 ID를 얻어옴
	SetCommonBkColor()	캡션바의 공통 배경색을 설정
	AdjustWindowRgn()	윈도우 크기를 조정하는 함수
	OnNcPaint()	윈도우 그리기 한 것을 받아와 그리는 함수
	OnNcActivate()	윈도우를 활성화하는 함수
	OnSize()	윈도우 전체 및 세부 사항 크기를 설정함
	LoadFrame()	프레임을 가져옴
	DefWindowProc()	최대화, 최소화 버튼을 클릭 시 크기를 변경
	OnEraseBkgnnd()	배경을 일시적으로 제거하는 함수
	OnTimer()	상태 바에 타이머를 설정
	OnNcLButtonUp()	마우스 포인터가 클릭 시 설정한 행동을 결정
	OnNcHitTest()	시스템 버튼에 사용자 설정을 더함
	OnNcMouseMove()	마우스를 놓았을 시 행동을 결정함
	OnNcRButtonUp()	마우스 오른쪽 버튼을 눌렀을 시 행동을 결정함

### 3.6 기능(모듈)별 주요 소스코드 및 설명

#### [로그인]

```
void CToyPocketView::OnBnClickedLogin() //로그인 성공/실패 판별
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(TRUE);

    CToyPocketSet join;
    join.Open();

    CString id = m_loginid;
    CString pass = m_loginpwd;

    while(!join.IsEOF()){

        CString m_id = join.m_id;
        CString m_pass = join.m_password;

        if(m_id == id && m_pass == pass)
        {
            AfxMessageBox(_T("로그인 성공"));
            g_setid = id;
            g_pwd = pass;
            m_bIsCopyTime = TRUE;
            m_loginid_Edit.SetWindowTextW(_T(""));
            m_loginpwd_Edit.SetWindowTextW(_T(""));
            CMainFrame* pFrame = (CMainFrame*)AfxGetMainWnd();
            pFrame->SwitchView(VIEWID_HOMEPAGE);

            pFrame->m_nFirst = 1;

            CRect rect;
            pFrame->GetWindowRect(&rect);
            pFrame->MoveWindow(250, 0, 100, 100);

            CDatabase db; // 데이터베이스 객체 생성

            CRecordset rs (&db); // 레코드셋 객체 생성
            try
            {
                CString strConnection(L"DRIVER={Microsoft Access Driver (*.mdb)};
                                         DBQ=C:\\WWWDownload\\WWWToyPocketDB.mdb;");

                if (db.OpenEx (strConnection) == FALSE)
                    throw _T("데이터베이스 접속실패!");
                CString strQuery;
                strQuery.Format(_T("SELECT permission FROM UserData
                                         WHERE id='%s' AND password='%s'"), id, pass);

                if (rs.Open (CRecordset::snapshot, strQuery) == FALSE)
                    throw _T("쿼리전송 실패!");
            }
            catch (CException * e)
            {
                e->ReportError();
                e->Delete();
            }
            catch (LPCTSTR msg)
            {
                AfxMessageBox (msg);
            }
        }
    }
}
```

```

//쿼리문을 이용해 검색한 내용 리스트 컨트롤에 출력
short col = rs.GetODBFieldCount();
for(int p=0 ; p<2 ; p++)
{
    while( ! rs.IsEOF())    //행 단위
    {
        //속성단위
        rs.GetFieldValue((short)p, g_per);
        //id값이 기준이 된다.
        for(short i=1; i<col; i++ )
        {
            rs.GetFieldValue(i, g_per);
            // 첫번째 인자는 InsertValue를 통해서 가지고 온 기준이 되는정보,
            // 두번째 인자는 열단위로 하나씩
        }
        rs.MoveNext();    //다음행으로 이동
    }
}

rs.Close();    //레코드셋 객체 소멸
db.Close();    //데이터베이스 객체 소멸
return;

{
    join.MoveNext();
}

join.MoveLast();
AfxMessageBox(_T("로그인 실패"));
}

```

## [회원가입]

```

void CRegisterFormView::OnBnClickedRegister()    //회원 등록
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(TRUE);
    if(m_strName!="&&m_loginpwd!="&&m_loginpwd2!="&&m_userName!="&&m_userEmail){

        CString s;
        CString sdate;
        sdate.Format(_T("%d-%02d-%02d"),m_userBirth.GetYear(),
            m_userBirth.GetMonth(),m_userBirth.GetDay());

        CDatabase db;    // 데이터베이스 객체 생성
        CRecordset rs (&db);    // 레코드셋 객체 생성
        try
        {
            CString strConnection(L"DRIVER={Microsoft Access Driver (*.mdb)};
                DBQ=C:\\WWWDownload\\WWWToyPocketDB.mdb;"); //DB오픈

            if (db.OpenEx (strConnection) == FALSE)
                throw _T("데이터베이스 접속실패!");

            CString strQuery;    //데이터베이스에 전송할 쿼리문
            strQuery.Format(_T("INSERT INTO UserData VALUES
                ('%s', '%s', '%s', '%s', '%s', '%s', '%s')"),
                m_strName, m_loginpwd, m_userName, sdate,
                m_userEmail, _T("예"), _T("manager") );

            db.ExecuteSQL(strQuery);    //데이터베이스에 쿼리문 전송
        }
    }
}

```



```

        catch (CException * e)
        {
            e->ReportError();
            e->Delete();
        }
        catch (LPCTSTR msg)
        {
            AfxMessageBox (msg);
        }
        MessageBox(_T("성공적으로 추가되었습니다. "));
        //OnBnClickedButtonComClear(); //컨트롤 초기화

        rs.Close(); //레코드셋 객체 소멸
        db.Close(); //데이터베이스 객체 소멸
    }else MessageBox(_T("입력되지 않은 값이 있습니다.\n모든 값을 입력(선택)해주세요. "));
        UpdateData(false);
        AfxMessageBox(_T("가입완료!"));
        CMainFrame* pFrame = (CMainFrame*)AfxGetMainWnd();
        pFrame->SwitchView(VIEWID_HOME);
    }
}

```

### [거래처관리 - 거래처등록]

```

#include "CompanySheet.h"
void CCompanyPage::OnBnClickedComAdd() //회사등록 이벤트 처리기
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(TRUE);
    if(m_cnumber!="&&m_cname!="&&m_cbusiness!="&&m_ctype!="&&m_ctel1!=
        "&&m_ctel2!="&&m_caddress!="&&m_cceoname!="&&m_cemail!="&&m_cmtel1!=
        "&&m_cmtel2!="&&m_cmemo!="")
    {
        int nIndex;
        CString s;
        CString strSort;
        CString strCTel;
        CString strCEOTel;
        CString sdate;
        CString tmp;
        sdate.Format(_T("%d-%02d-%02d"),m_cbirth.GetYear(),m_cbirth.GetMonth(),
            m_cbirth.GetDay());

        nIndex = m_csor.t.GetCurSel();
        if(nIndex == -1)
            return;
        m_csor.t.GetLBText(nIndex, strSort); //strSort에 콤보박스 값 읽어오기

        nIndex = m_ctel.GetCurSel();
        if(nIndex == -1)
            return;
        m_ctel.GetLBText(nIndex, s);
        strCTel.Format(_T("%s-%s-%s"),s, m_ctel1, m_ctel2);

        nIndex = m_cmtel.GetCurSel();
        if(nIndex == -1)
            return;
        m_cmtel.GetLBText(nIndex, s);
        strCEOTel.Format(_T("%s-%s-%s"),s, m_cmtel1, m_cmtel2);

        //거래처 코드 자동 추가
        CDatabase db0; // 데이터베이스 객체 생성
    }
}

```

```

CRecordset rs0 (&db0); // 레코드셋 객체 생성
try
{
    CString strConnection(L"DRIVER={Microsoft Access Driver (*.mdb)};
                           DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

    if (db0.OpenEx (strConnection) == FALSE)
        throw _T("데이터베이스 접속실패!");

    CString strQuery;
    nIndex = m_csort.GetCurSel();
    if(nIndex == 0){
        strQuery.Format(_T("SELECT c_code FROM Company WHERE
                           c_code LIKE '%IC%' ORDER BY c_code ASC"));
        //db에서 납입처에 해당하는 거래처 코드 중 가장 마지막 값 읽어오기
    }
    if(nIndex == 1){
        strQuery.Format(_T("SELECT c_code FROM Company WHERE
                           c_code LIKE '%OC%' ORDER BY c_code ASC"));
        //db에서 납품처에 해당하는 거래처 코드 중 가장 마지막 값 읽어오기
    }

    if (rs0.Open (CRecordset::snapshot, strQuery) == FALSE)
        throw _T("쿼리전송 실패!");
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}

short col = rs0.GetODBCFieldCount(); //쿼리문을 이용해 검색한 내용을
                                     //리스트 컨트롤에 출력
for(int p=0 ; p<2 ; p++)
{
    while( ! rs0.IsEOF())
    {
        rs0.GetFieldValue((short)p, tmp);

        for(short i=1; i<col; i++ )
        {
            rs0.GetFieldValue(i, tmp); //받은 값을 tmp에 저장
        }
        rs0.MoveNext();
    }
}
rs0.Close();
db0.Close();

lastint=tmp.Right(2); //tmp에 저장된 거래처 코드의 오른쪽 2글자만 읽어오기
length = lastint.GetLength(); //글자 수 읽어오기
nIndex = m_csort.GetCurSel();
if(nIndex == 0){ //CSORT인덱스 0-납입처 / 1-납품처
    result.Format(_T("IC_%04d"),_ttoi(lastint)+1);
}
else if(nIndex == 1){
    result.Format(_T("OC_%04d"),_ttoi(lastint)+1);
}
}

```

```

CDatabase db;           // 데이터베이스 객체 생성
CRecordset rs (&db);    // 레코드셋 객체 생성
try
{
    CString strConnection(L"DRIVER={Microsoft Access Driver (*.mdb)};
                           DBQ=C:\\WWWDownload\\WWWToyPocketDB.mdb;"); //DB오픈

    if (db.OpenEx (strConnection) == FALSE)
        throw _T("데이터베이스 접속실패!");

    CString strQuery;     //데이터베이스에 전송할 쿼리문
    strQuery.Format(_T("INSERT INTO Company VALUES ('%s', '%s', '%s',
'%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s')"),
    result, m_cname, strSort, m_ctype, m_cbusiness, m_cceoname,
    strCTel, m_cnumber, m_caddress, strCEOTel, m_cemail, sdate,
    m_cmemo, g_setid);

    db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}
MessageBox(_T("성공적으로 추가되었습니다. "));

rs.Close(); //레코드셋 객체 소멸
db.Close(); //데이터베이스 객체 소멸
}else MessageBox(_T("입력되지 않은 값이 있습니다.\n모든 값을 입력(선택)해주세요. "));

//다 등록 후 값들 초기화시키기
m_cnumber = _T("");
m_cname = _T("");
m_csor t.SetCurSel(0);
m_ctype = _T("");
m_cbusiness = _T("");
m_ctel.SetCurSel(0);
m_ctel1 = _T("");
m_ctel2 = _T("");
m_caddress = _T("");
m_cceoname = _T("");
m_cemail = _T("");
m_cmtel.SetCurSel(0);
m_cmtel1 = _T("");
m_cmtel2 = _T("");
m_cmemo = _T("");
//생일 초기화
CDateTimeCtrl *pCtrl = (CDateTimeCtrl *) GetDlgItem(IDC_DATETIMEPICKER1);
CTime tm = CTime::GetCurrentTime();
pCtrl->SetTime(&tm);
UpdateData(false);

//등록
CCompanySheet* pSheet = (CCompanySheet*)GetParent();
CCompanySearchPage* pPage = (CCompanySearchPage*)(pSheet->GetPage(1));
pPage->ReInitDialog();
}

```

## [거래처관리 - 거래처 조회]

```
void CCompanySearchPage::OnBnClickedSearchbtn() //거래처 정보 조회 이벤트 처리기
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(true);

    if(m_cs_list.GetItemCount() > 0) m_cs_list.DeleteAllItems();

    CDatabase db; // 데이터베이스 객체 생성
    CRecordset rs(&db); // 레코드셋 객체 생성
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access Driver (*.mdb)};
                               DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

        if (db.OpenEx(strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery;

        int nIndex = m_combosearch.GetCurSel();
        if(nIndex == 0){//전체검색 클릭시
            strQuery.Format(_T("SELECT com_id, c_code, c_name, c_sort, c_type,
                                c_business, c_ceoname, c_tel, c_number, c_address, c_mobile,
                                c_email, c_birth, c_memo FROM Company WHERE c_code LIKE '%%s%%' OR
                                c_name LIKE '%%s%%' OR c_sort LIKE '%%s%%' OR c_type LIKE
                                '%%s%%' OR c_business LIKE '%%s%%' OR c_ceoname LIKE '%%s%%' OR
                                c_tel LIKE '%%s%%'"), m_csearch, m_csearch, m_csearch, m_csearch,
                                m_csearch, m_csearch, m_csearch);
            result.Format(_T("전체검색-( '%s' )검색시_"), m_csearch);
        }
        if(nIndex == 1){//거래처코드 클릭시
            strQuery.Format(_T("SELECT com_id, c_code, c_name, c_sort, c_type,
                                c_business, c_ceoname, c_tel, c_number, c_address, c_mobile,
                                c_email, c_birth, c_memo FROM Company WHERE c_code LIKE
                                '%%s%%'"), m_csearch);
            result.Format(_T("거래처코드-( '%s' )검색시_"), m_csearch);
        }
        if(nIndex == 2){//거래처명 클릭시
            strQuery.Format(_T("SELECT com_id, c_code, c_name, c_sort, c_type,
                                c_business, c_ceoname, c_tel, c_number, c_address, c_mobile,
                                c_email, c_birth, c_memo FROM Company WHERE c_name LIKE
                                '%%s%%'"), m_csearch);
            result.Format(_T("거래처명-( '%s' )검색시_"), m_csearch);
        }
        if(nIndex == 3){//업종 클릭시
            strQuery.Format(_T("SELECT com_id, c_code, c_name, c_sort, c_type,
                                c_business, c_ceoname, c_tel, c_number, c_address, c_mobile,
                                c_email, c_birth, c_memo FROM Company WHERE c_type LIKE
                                '%%s%%'"), m_csearch);
            result.Format(_T("업종-( '%s' )검색시_"), m_csearch);
        }
        if(nIndex == 4){//업태 클릭시
            strQuery.Format(_T("SELECT com_id, c_code, c_name, c_sort, c_type,
                                c_business, c_ceoname, c_tel, c_number, c_address, c_mobile,
                                c_email, c_birth, c_memo FROM Company WHERE c_business LIKE
                                '%%s%%'"), m_csearch);
            result.Format(_T("업태-( '%s' )검색시_"), m_csearch);
        }

        if (rs.Open(CRecordset::snapshot, strQuery) == FALSE)
```

```

        throw _T("쿼리전송 실패!");
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox (msg);
    }

    short col = rs.GetODBCFieldCount(); //쿼리문을 이용해 검색한 내용 리스트 컨트롤에 출력
    for(int p=0 ; p<2 ; p++)
    {
        while( ! rs.IsEOF())
        {
            CString tmp;
            rs.GetFieldValue((short)p, tmp);
            m_cs_list.InsertItem(p,tmp,0);
            for(short i=1; i<col; i++ )
            {
                rs.GetFieldValue(i, tmp);
                m_cs_list.SetItemText(p, i ,tmp);
            }
            rs.MoveNext();
        }
    }
    rs.Close();
    db.Close();
}

```

//더블클릭 시 이벤트 처리기 (바로 수정탭으로 이동해서 값 setting)

#include "CompanyModifyPage.h"

void CCompanySearchPage::OnNMDBlclclList1(NMHDR \*pNMHDR, LRESULT \*pResult)

```

{
    LPNMITEMACTIVATE pNMItemActivate = reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    nListIndex = pNMListView->iItem;

    int formindex;
    CString num1;

    g_set_cnumber = m_cs_list.GetItemText(nListIndex,8);
    g_set_cname = m_cs_list.GetItemText(nListIndex,2);

    if(m_cs_list.GetItemText(nListIndex,3)==_T("납입처"))
        g_set_csortindex = 0;
    else
        g_set_csortindex = 1;

    g_set_ctype = m_cs_list.GetItemText(nListIndex,4);
    g_set_cbusiness = m_cs_list.GetItemText(nListIndex,5);
    g_set_caddress = m_cs_list.GetItemText(nListIndex,9);
    g_set_cceoname = m_cs_list.GetItemText(nListIndex,6);
    g_set_cemail = m_cs_list.GetItemText(nListIndex,11);
    g_set_cmemo = m_cs_list.GetItemText(nListIndex,13);

    AfxExtractSubString(g_set_cbirth_Year,m_cs_list.GetItemText(nListIndex,12), 0, '-');
    AfxExtractSubString(g_set_cbirth_Month,m_cs_list.GetItemText(nListIndex,12), 1, '-');
    AfxExtractSubString(g_set_cbirth_Day,m_cs_list.GetItemText(nListIndex,12), 2, '-');
}

```

```

CString ctel;
CString cmTel;

AfxExtractSubString(ctel,m_cs_list.GetItemText(nListIndex,7), 0, '-');
if(ctel==_T("02"))
    g_set_ctelindex = 0;
else
    g_set_ctelindex = 1;

AfxExtractSubString(g_set_ctel1,m_cs_list.GetItemText(nListIndex,7), 1, '-');
AfxExtractSubString(g_set_ctel2,m_cs_list.GetItemText(nListIndex,7), 2, '-');

AfxExtractSubString(cmTel,m_cs_list.GetItemText(nListIndex,10), 0, '-');
if(cmTel==_T("010"))
    g_set_cmtelindex = 0;
else
    g_set_cmtelindex = 1;
AfxExtractSubString(g_set_cmtel1,m_cs_list.GetItemText(nListIndex,10), 1, '-');
AfxExtractSubString(g_set_cmtel2,m_cs_list.GetItemText(nListIndex,10), 2, '-');

UpdateData(FALSE);

//더블클릭 시 수정탭으로 이동
CCompanySheet* pSheet = (CCompanySheet*)GetParent();
pSheet->SetActivePage(2);
*getResult = 0;
//modifypage의 에디트 컨트롤의 위에 변수들 넣어줘야함
}

//거래처 정보 삭제 컨텍스트 메뉴
void CCompanySearchPage::OnNMRClickList1(NMHDR *pNMHDR, LRESULT *pResult)
{
LPNMITEMACTIVATE pNMItemActivate = reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);
// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
nListIndex = pNMListView->iItem;

CPoint ptInScreen;
GetCursorPos(&ptInScreen); //R버튼 눌린좌표얻음

if(nListIndex==--1){
}
else{
CMenu muTemp, *pContextMenu;
muTemp.LoadMenuW(IDR_PopupMenu);
pContextMenu = muTemp.GetSubMenu(2);
pContextMenu->TrackPopupMenu(TPM_LEFTALIGN|TPM_RIGHTBUTTON, ptInScreen.x,
ptInScreen.y, this);
}
ccode = m_cs_list.GetItemText(nListIndex,1);
*getResult = 0;
}

void CCompanySearchPage::OnCompanyDelete() //삭제 메뉴 명령 처리기
{
// TODO: 여기에 명령 처리기 코드를 추가합니다.
int ret;
ret = AfxMessageBox(_T("삭제하시겠습니까?"), MB_YESNO|MB_ICONSTOP);

if(ret == IDYES){//삭제 버튼 클릭시
UpdateData(TRUE);

```

```

CDatabase db;           // 데이터베이스 객체 생성
CRecordset rs (&db);    // 레코드셋 객체 생성

try
{
    CString strConnection(L"DRIVER={Microsoft Access Driver (*.mdb)};
                           DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

    if (db.OpenEx (strConnection) == FALSE)
        throw _T("데이터베이스 접속실패!");

    CString strQuery;           //데이터베이스에 전송할 쿼리문
    strQuery.Format(_T("DELETE * FROM Company WHERE c_code = '%s'"), ccode);
    //선택한 거래처 코드에 해당하는 db값 삭제
    db.ExecuteSQL(strQuery);     //데이터베이스에 쿼리문 전송

}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}
rs.Close();    //레코드셋 객체 소멸
db.Close();    //데이터베이스 객체 소멸
}

if(m_cs_list.GetItemCount() > 0) m_cs_list.DeleteAllItems();

CDatabase db1;           // 데이터베이스 객체 생성
CRecordset rs1 (&db1);  // 레코드셋 객체 생성
try
{
    CString strConnection(L"DRIVER={Microsoft Access Driver (*.mdb)};
                           DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

    if (db1.OpenEx (strConnection) == FALSE)
        throw _T("데이터베이스 접속실패!");

    CString strQuery;
    strQuery.Format(_T("SELECT com_id, c_code, c_name, c_sort, c_type,
c_business, c_ceoname, c_tel, c_number, c_address, c_mobile, c_email,
c_birth, c_memo FROM Company"));

    if (rs1.Open (CRecordset::snapshot, strQuery) == FALSE)
        throw _T("쿼리전송 실패!");
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}

//쿼리문을 이용해 검색한 내용 리스트 컨트롤에 출력

```

```

short col1 = rs1.GetODBCFieldCount();
for(int p=0 ; p<2 ; p++)
{
    while( ! rs1.IsEOF())          //행단위
    {
        //속성단위
        CString tmp;
        rs1.GetFieldValue((short)p, tmp);
        //id값이 기준이 된다.
        m_cs_list.InsertItem(p,tmp,0);
        for(short i=1; i<col1; i++ )
        {
            rs1.GetFieldValue(i, tmp);
            // 첫 번째 인자는 InsertValue를 통해서 가지고 온 기준이 되는 정보,
            // 두 번째 인자는 열 단위로 하나씩
            m_cs_list.SetItemText(p, i ,tmp);
        }
        rs1.MoveNext();          //다음행으로 이동
    }
}
rs1.Close();                    //레코드셋 객체 소멸
db1.Close();                    //데이터베이스 객체 소멸
}

void CCompanySearchPage::OnBnClickedExcelSave()    //엑셀 저장
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    //리스트컨트롤 값 엑셀로 저장
    if (m_cs_list.GetItemCount() <= 0)
    {
        AfxMessageBox(_T("조회된 데이터가 0건이므로 엑셀 저장 할 수 없습니다.));
        return;
    }

    CTime timeNow = CTime::GetCurrentTime();
    CString strDateTime;
    strDateTime = timeNow.Format(_T("%Y-%m-%d_%H시%M분%S초"));
    CString strFileName;
    strFileName.Format(_T("%s거래처 조회_%s"), result, strDateTime);

    CFileDialog fdlg(FALSE, _T("xlsx"), strFileName+_T(".xlsx"), OFN_HIDEREADONLY |
    OFN_OVERWRITEPROMPT | OFN_NOCHANGEDIR, _T("xlsx 파일 (*.xlsx)|*.xlsx"), NULL);
    if (fdlg.DoModal() != IDOK) return;

    CXLEzAutomation xls(FALSE); // FALSE : 처리 과정을 화면에 보이지 않는다.

    int rowOffset = 2;
    int colOffset = 1;

    int rowCount = m_cs_list.GetItemCount();
    CHeaderCtrl* pHeaderCtrl = m_cs_list.GetHeaderCtrl();
    int colCount = pHeaderCtrl->GetItemCount();
    int rowIdx;
    int colIdx;

    CString strItemText;
    for (colIdx = 0; colIdx < colCount; colIdx++){
        HDITEM hdi;
        hdi.mask = HDI_TEXT;
        hdi.pszText = strItemText.GetBuffer( 256 );
        hdi.cchTextMax = 256;
        pHeaderCtrl->GetItem( colIdx, &hdi );
    }
}

```



```

        strItemText.ReleaseBuffer();
        xls.SetCellValue(colIdx + colOffset, 0 + rowOffset, strItemText);
    }

    for (rowIdx = 0; rowIdx < rowCount; rowIdx++)
    {
        for (colIdx = 0; colIdx < colCount; colIdx++)
        {
            strItemText = m_cs_list.GetItemText(rowIdx, colIdx);

            // 컬럼 데이터
            xls.SetCellValue(colIdx + colOffset, rowIdx + 1 + rowOffset, strItemText);
        }
    }
    xls.SaveFileAs(fdlg.GetPathName());
    xls.ReleaseExcel();
}

```

### [거래처관리 - 거래처수정]

```

void CCompanyModifyPage::OnBnClickedButtonModi()    //수정 버튼 이벤트 처리기
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(TRUE);
    if(m_cnumber!=""&&m_cname!=""&&m_cbusiness!=""&&m_ctype!=""&&m_ctel1!=
        ""&&m_ctel2!=""&&m_caddress!=""&&m_cceoname!=""&&m_cemail!=""&&m_cmte11!=
        ""&&m_cmte12!=""&&m_cmemo!="") {
        int nIndex;
        CString s;
        CString strSort;
        CString strCTel;
        CString strCEOTel;
        COleDateTime selectDate;
        m_cbirth.GetTime(selectDate);    //datetimepicker 컨트롤 변수
        CString sdate = selectDate.Format(_T("%Y-%m-%d"));

        nIndex = m_csor.sort.GetCurSel();
        if(nIndex == -1)
            return;
        m_csor.sort.GetLBText(nIndex, strSort);

        nIndex = m_ctel.GetCurSel();
        if(nIndex == -1)
            return;
        m_ctel.GetLBText(nIndex, s);
        strCTel.Format(_T("%S-%S-%S"),s, m_ctel1, m_ctel2);

        nIndex = m_cmte1.GetCurSel();
        if(nIndex == -1)
            return;
        m_cmte1.GetLBText(nIndex, s);
        strCEOTel.Format(_T("%S-%S-%S"),s, m_cmte11, m_cmte12);

        CDatabase db;    // 데이터베이스 객체 생성
        CRecordset rs(&db);    // 레코드셋 객체 생성
        try
        {
            CString strConnection(L"DRIVER={Microsoft Access Driver (*.mdb)};
                                DBQ=C:\\WWWDownload\\WWWToyPocketDB.mdb;");    //DB오픈

            if (db.OpenEx (strConnection) == FALSE)

```

```

        throw _T("데이터베이스 접속실패!");

        CString strQuery;          //데이터베이스에 전송할 쿼리문
        strQuery.Format(_T("UPDATE Company SET c_name='%s', c_sort='%s',
        c_type='%s', c_business='%s', c_ceoname='%s', c_tel='%s',
        c_number='%s', c_address='%s', c_mobile='%s', c_email='%s',
        c_birth='%s', c_memo='%s', com_id='%s' WHERE c_name='%s'"),
        m_cname, strSort, m_ctype, m_cbusiness, m_cceoname, strCTel,
        m_cnumber, m_caddress, strCEOTel, m_cemail, sdate, m_cmemo,
        g_setid, m_cname);

        db.ExecuteSQL(strQuery);      //데이터베이스에 쿼리문 전송
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox (msg);
    }
    MessageBox(_T("성공적으로 수정되었습니다. "));

    rs.Close();          //레코드셋 객체 소멸
    db.Close();          //데이터베이스 객체 소멸
} else MessageBox(_T("입력되지 않은 값이 있습니다. 모든 값을 입력(선택)해주세요. "));

m_cnumber = _T("");
m_cname = _T("");
m_csor t.SetCurSel(0);
m_ctype = _T("");
m_cbusiness = _T("");
m_ctel.SetCurSel(0);
m_ctel1 = _T("");
m_ctel2 = _T("");
m_caddress = _T("");
m_cceoname = _T("");
m_cemail = _T("");
m_cmtel.SetCurSel(0);
m_cmtel1 = _T("");
m_cmtel2 = _T("");
m_cmemo = _T("");
//생일 초기화
CDateTimeCtrl *pCtrl = (CDateTimeCtrl *) GetDlgItem(IDC_DATETIMEPICKER1);
CTime tm = CTime::GetCurrentTime();
pCtrl->SetTime(&tm);
UpdateData(false);
}

void CCompanyModifyPage::OnPaint() //조희탭에서 더블클릭시 바로 수정탭으로 이동해서 값
setting
{
    CPaintDC dc(this); // device context for painting
    // TODO: 여기에 메시지 처리기 코드를 추가합니다.
    // 그리기 메시지에 대해서는 CPropertyPage::OnPaint()을(를) 호출하지 마십시오.

    if(m_cnumber != g_set_cnumber)
    {
        m_ccnuber_Edit.SetWindowTextW(g_set_cnumber);
        m_cname_Edit.SetWindowTextW(g_set_cname);
        m_csor t.SetCurSel(g_set_csor t index);
        m_ctype_Edit.SetWindowTextW(g_set_ctype);
    }
}

```

```

m_cbusiness_Edit.SetWindowTextW(g_set_cbusiness);
m_ctel.SetCurSel(g_set_ctelindex);
m_ctel1_Edit.SetWindowTextW(g_set_ctel1);
m_ctel2_Edit.SetWindowTextW(g_set_ctel2);
m_caddress_Edit.SetWindowTextW(g_set_caddress);
m_cceoname_Edit.SetWindowTextW(g_set_cceoname);
m_cemail_Edit.SetWindowTextW(g_set_cemail);
//생일
CDateTime Date; //사용자가 지정한 날짜를 저장할 CDateTimeDate 변수

CString wsYear = g_set_cbirth_Year; //년
CString wsMonth = g_set_cbirth_Month; //월
CString wsDay = g_set_cbirth_Day; //일
Date.SetDate(_tstoi(wsYear), _tstoi(wsMonth), _tstoi(wsDay));
//CDateTime타입은 UpdateData해줘야함 그럼 나머지만먹힘
//그래서 변수 지우고 다시 DateTime형식의 컨트롤로 다시만들
m_cbirth.SetTime(Date);

m_cmtel.SetCurSel(g_set_cmtelindex);
m_cmtel1_Edit.SetWindowTextW(g_set_cmtel1);
m_cmtel2_Edit.SetWindowTextW(g_set_cmtel2);
m_cmemo_Edit.SetWindowTextW(g_set_cmemo);
}
}

//조회 버튼 클릭시 조회탭으로 이동
void CCompanyModifyPage::OnBnClickedComSearchBtn()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    CCompanySheet* pSheet = (CCompanySheet*)GetParent();
    pSheet->SetActivePage(1);
}

```

### [상품관리 - 상품등록]

```

void CItemPage::OnBnClickedOk() //상품 등록 이벤트 처리기
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(TRUE);
    if(m_icode!=""&&m_iname!=""&&m_icname!=""&&m_icode!=""&&m_icount!=" "
        &&m_iinput!=""&&m_ioutput!="") {

        CString s;
        CString sdate;
        sdate.Format(_T("%d-%02d-%02d"), m_idate.GetYear(), m_idate.GetMonth(),
            m_idate.GetDay());

        CString strlarge; //대분류
        strlarge = SelectLarge(m_ilarge.GetCurSel());
        CString strmid;
        strmid = SelectMedium(m_imedium.GetCurSel()); //중분류
        CString strsmall;
        strsmall = SelectSmall(m_ilarge.GetCurSel(), m_imedium.GetCurSel(),
            m_ismall.GetCurSel()); //소분류

        CDatabase db; // 데이터베이스 객체 생성
        CRecordset rs(&db); // 레코드셋 객체 생성
        try
        {
            CString strConnection(L"DRIVER={Microsoft Access Driver

```



```

{
    switch( index)
    {
    case 0 :
        temp = _T( "신생아/유아");
        break;
    case 1 :
        temp = _T( "여아");
        break;
    case 2 :
        temp = _T( "남아");
        break;
    case 3 :
        temp = _T( "캐릭터별");
        break;
    }
    return temp;
}

```

```

CString CItemPage::SelectMedium( int index)           //중분류 문자열 반환
{
    int select1 = m_iLarge.GetCurSel();
    if(select1 == 0)
    switch( index){
        case 0:
            temp = _T( "신생아");
            break;
        case 1:
            temp = _T( "유아");
            break;
    }
    else if(select1 == 1)
    switch( index){
        case 0:
            temp = _T( "역할놀이");
            break;
        case 1:
            temp = _T( "인형");
            break;
        case 2:
            temp = _T( "꾸미기");
            break;
    }
    else if(select1 == 2)
    switch( index){
        case 0:
            temp = _T( "역할놀이");
            break;
        case 1:
            temp = _T( "액션");
            break;
        case 2:
            temp = _T( "조종");
            break;
        case 3:
            temp = _T( "자동차/기차");
            break;
    }
    else if(select1 == 3)
    switch( index){
        case 0:
            temp = _T( "유아캐릭터");

```

```

        break;
    case 1:
        temp = _T("여아캐릭터");
        break;
    case 2:
        temp = _T("남아캐릭터");
        break;
    }
    return temp;
}

CString CItemPage::SelectSmall(int bigc, int midc, int smallc) //소분류 문자열 반환
{
    if(bigc == 0)
    switch(midc){
        case 0: {
            switch(smallc){
                case 0: temp = _T("치발기/딸랑이"); break;
                case 1: temp = _T("모빌"); break;
                case 2: temp = _T("촉각놀이/오뎅이"); break;
                case 3: temp = _T("아기체육관/러닝홈"); break;
                case 4: temp = _T("보행기/바운서/쏘서"); break;
            }break; }
        case 1: {
            switch(smallc){
                case 0: temp = _T("공놀이"); break;
                case 1: temp = _T("목욕놀이"); break;
                case 3: temp = _T("병원/역할놀이"); break;
                case 4: temp = _T("악기놀이"); break;
                case 5: temp = _T("자동차놀이"); break;
            }break; }
        }
    else if(bigc == 1)
    switch(midc){
        case 0: {
            switch(smallc){
                case 0: temp = _T("소꿉/주방놀이"); break;
                case 1: temp = _T("마트놀이"); break;
                case 2: temp = _T("엄마/청소놀이"); break;
                case 3: temp = _T("병원/기타놀이"); break;
            }break; }
        case 1: {
            switch(smallc){
                case 0: temp = _T("공주인형"); break;
                case 1: temp = _T("인형놀이"); break;
                case 3: temp = _T("아기인형"); break;
                case 4: temp = _T("캐릭터봉제인형"); break;
                case 5: temp = _T("동물봉제인형"); break;
                case 6: temp = _T("작동봉제인형"); break;
            }break; }
        case 2: {
            switch(smallc){
                case 0: temp = _T("드레스 및 소품"); break;
                case 1: temp = _T("헤어/화장놀이"); break;
                case 3: temp = _T("핸드폰/디카/기타"); break;
            }break; }
        }
    else if(bigc == 2)
    switch(midc){
        case 0: {
            switch(smallc){
                case 0: temp = _T("공구놀이"); break;

```

```

        case 1: temp = _T("카드놀이"); break;
        case 2: temp = _T("폼건놀이"); break;
        case 3: temp = _T("코스튬플레이"); break;
        case 4: temp = _T("요요/팽이놀이"); break;
        case 5: temp = _T("필거/기타놀이"); break;
    }break; }
case 1: {
    switch(smallc){
        case 0: temp = _T("변신로봇/탈것"); break;
        case 1: temp = _T("액션무기"); break;
        case 3: temp = _T("액션피규어"); break;
    }break; }
case 2: {
    switch(smallc){
        case 0: temp = _T("명품자동차RC"); break;
        case 1: temp = _T("일반자동차RC"); break;
        case 3: temp = _T("캐릭터RC"); break;
        case 4: temp = _T("드론/헬기/보트/기타RC"); break;
    }break; }
case 3: {
    switch(smallc){
        case 0: temp = _T("빅카/트럭/중장비"); break;
        case 1: temp = _T("일반자동차"); break;
        case 2: temp = _T("수집자동차"); break;
        case 3: temp = _T("자동차놀이세트"); break;
        case 4: temp = _T("비행기/기차/기타"); break;
    }break; }
}
else if(bigc == 3)
switch(midc){
    case 0: {
        switch(smallc){
            case 0: temp = _T("그라미"); break;
            case 1: temp = _T("그린토이즈"); break;
            case 2: temp = _T("라바"); break;
            case 3: temp = _T("로보카폴리"); break;
            case 4: temp = _T("뽀로로"); break;
            case 5: temp = _T("라마즈"); break;
        }break; }
    case 1: {
        switch(smallc){
            case 0: temp = _T("겨울왕국"); break;
            case 1: temp = _T("디즈니 프린세스"); break;
            case 2: temp = _T("똘똘이"); break;
            case 3: temp = _T("미미"); break;
            case 4: temp = _T("바비"); break;
            case 5: temp = _T("인사이드아웃"); break;
            case 6: temp = _T("쥬쥬"); break;
            case 7: temp = _T("콩순이"); break;
            case 8: temp = _T("헬로키티"); break;
        }break; }
    case 2: {
        switch(smallc){
            case 0: temp = _T("건담"); break;
            case 1: temp = _T("또봇"); break;
            case 3: temp = _T("마인크래프트"); break;
            case 4: temp = _T("미니언즈"); break;
            case 5: temp = _T("어벤저스"); break;
            case 6: temp = _T("원피스"); break;
        }break; }
}
return temp;

```

```

}
void CItemPage::DrawBMP(LPCTSTR lpszFileName)    //이미지 보여주기
{
    CDC* pDC = GetDC(); //화면 DC를 얻어옴
    CDC MemDC; //메모리 DC
    CBitmap bmp, *pOldBmp;
    BITMAP bm; //비트맵 정보를 저장하는 변수
    CRect rect;
    CPoint cp;

    //이미지 파일 로딩
    bmp.m_hObject = (HBITMAP)LoadImage(NULL, lpszFileName, IMAGE_BITMAP, 0, 0,
        LR_LOADFROMFILE | LR_CREATEDIBSECTION | LR_DEFAULTSIZE);
    if(bmp.m_hObject == NULL) return;
    bmp.GetBitmap(&bm); //로딩된 이미지 파일의 형태를 얻음
    MemDC.CreateCompatibleDC(pDC); //메모리 DC와 화면 DC를 호환성있도록 연결
    pOldBmp = MemDC.SelectObject(&bmp); //메모리에 적재된 이미지 선택

    m_image.GetWindowRect(&rect); //다이얼로그에 배치된 컨트롤의 크기를 얻어옴
    cp.x = rect.left; //이미지 출력을 위한 시작 x 좌표
    cp.y = rect.top; //이미지 출력을 위한 시작 y 좌표
    ScreenToClient(&cp); //cp가 가지고 있는 좌표로 화면을 지정함
    pDC->StretchBlt(cp.x, cp.y, rect.Width(), rect.Height(), &MemDC, 0, 0,
        bm.bmWidth, bm.bmHeight, SRCCOPY); //이미지를 해당 컨트롤로 전송
    MemDC.SelectObject(pOldBmp);
    ReleaseDC(pDC); //DC 반환
}

void CItemPage::OnBnClickedCancel2()    //이미지 불러오기
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(true);

    CFileDialog sDlg(false, NULL, NULL, OFN_HIDEREADONLY, 0);

    if(sDlg.DoModal() != IDOK)
        return;

    CString m_i;

    m_i = sDlg.GetFileName();

    imgname.Format(_T("res\\item\\%s.bmp"), m_i);
    CopyFile(sDlg.GetPathName(), imgname, FALSE);
    AfxMessageBox(_T("이미지가 저장되었습니다. "));
    DrawBMP(imgname);
}

void CItemPage::bigmidsmall(void)    //카테고리 문자열 반환
{
    if(Big == "신생아/유아"){
        m_iLarge.SetCurSel(0);
        OnCbnSelchangeCombo2();
        if(Middle == "신생아"){
            m_iMedium.SetCurSel(0);
            OnCbnSelchangeCombo4();
            if(Small == "치발기/딸랑이")
                m_iSmall.SetCurSel(0);
            else if(Small == "모빌")
                m_iSmall.SetCurSel(1);
            else if(Small == "촉각놀이/오뎅이")
                m_iSmall.SetCurSel(2);
            else if(Small == "아기체육관/러닝홈")
                m_iSmall.SetCurSel(3);
        }
    }
}

```



```

        else if(Small == "보행기/바운서/쏘서")
            m_ismall.SetCurSel(4);
    }
    else if(Middle == "유아"){
        m_imedium.SetCurSel(1);
        OnCbnSelchangeCombo4();
        if(Small == "공놀이")
            m_ismall.SetCurSel(0);
        else if(Small == "목욕놀이")
            m_ismall.SetCurSel(1);
        else if(Small == "병원/역할놀이")
            m_ismall.SetCurSel(2);
        else if(Small == "악기놀이")
            m_ismall.SetCurSel(3);
        else if(Small == "자동차놀이")
            m_ismall.SetCurSel(4);
    }
}

else if(Big == "여아"){
    m_ilarge.SetCurSel(1);
    OnCbnSelchangeCombo2();
    if(Middle == "역할놀이"){
        m_imedium.SetCurSel(0);
        OnCbnSelchangeCombo4();
        if(Small == "소꿉/주방놀이")
            m_ismall.SetCurSel(0);
        else if(Small == "마트놀이")
            m_ismall.SetCurSel(1);
        else if(Small == "엄마/청소놀이")
            m_ismall.SetCurSel(2);
        else if(Small == "병원/기타놀이")
            m_ismall.SetCurSel(3);
    }
    else if(Middle == "인형"){
        m_imedium.SetCurSel(1);
        OnCbnSelchangeCombo4();
        if(Small == "공주인형")
            m_ismall.SetCurSel(0);
        else if(Small == "인형놀이")
            m_ismall.SetCurSel(1);
        else if(Small == "아기인형")
            m_ismall.SetCurSel(2);
        else if(Small == "캐릭터봉제인형")
            m_ismall.SetCurSel(3);
        else if(Small == "동물봉제인형")
            m_ismall.SetCurSel(4);
        else if(Small == "작동봉제인형")
            m_ismall.SetCurSel(5);
    }
    else if(Middle == "꾸미기"){
        m_imedium.SetCurSel(2);
        OnCbnSelchangeCombo4();
        if(Small == "드레스 및 소품")
            m_ismall.SetCurSel(0);
        else if(Small == "헤어/화장놀이")
            m_ismall.SetCurSel(1);
        else if(Small == "핸드폰/디카/기타")
            m_ismall.SetCurSel(2);
    }
}
else if(Big == "남아"){

```

```

m_iLarge.SetCurSel(1);
OnCbnSelchangeCombo2();
if(Middle == "역할놀이"){
    m_imedium.SetCurSel(0);
    OnCbnSelchangeCombo4();
    if(Small == "공구놀이")
        m_ismall.SetCurSel(0);
    else if(Small == "카드놀이")
        m_ismall.SetCurSel(1);
    else if(Small == "품건놀이")
        m_ismall.SetCurSel(2);
    else if(Small == "코스튬플레이")
        m_ismall.SetCurSel(3);
    else if(Small == "요요/팽이놀이")
        m_ismall.SetCurSel(4);
    else if(Small == "핑거/기타놀이")
        m_ismall.SetCurSel(5);
}
else if(Middle == "액션"){
    m_imedium.SetCurSel(1);
    OnCbnSelchangeCombo4();
    if(Small == "변신로봇/탈것")
        m_ismall.SetCurSel(0);
    else if(Small == "액션무기")
        m_ismall.SetCurSel(1);
    else if(Small == "액션피규어")
        m_ismall.SetCurSel(2);
}
else if(Middle == "조종"){
    m_imedium.SetCurSel(2);
    OnCbnSelchangeCombo4();
    if(Small == "명품자동차RC")
        m_ismall.SetCurSel(0);
    else if(Small == "일반자동차RC")
        m_ismall.SetCurSel(1);
    else if(Small == "캐릭터RC")
        m_ismall.SetCurSel(2);
    else if(Small == "드론/헬기/보트/기타RC")
        m_ismall.SetCurSel(3);
}
}
else if(Big == "캐릭터별"){
    m_iLarge.SetCurSel(1);
    OnCbnSelchangeCombo2();
    if(Middle == "유아캐릭터"){
        m_imedium.SetCurSel(0);
        OnCbnSelchangeCombo4();
        if(Small == "그라미")
            m_ismall.SetCurSel(0);
        else if(Small == "그린토이즈")
            m_ismall.SetCurSel(1);
        else if(Small == "라바")
            m_ismall.SetCurSel(2);
        else if(Small == "로보카폴리")
            m_ismall.SetCurSel(3);
        else if(Small == "뽀로로")
            m_ismall.SetCurSel(4);
        else if(Small == "라마즈")
            m_ismall.SetCurSel(5);
    }
    else if(Middle == "여아캐릭터"){
        m_imedium.SetCurSel(1);
    }
}

```

```

        OnCbnSelchangeCombo4();
        if(Small == "겨울왕국")
            m_ismall.SetCurSel(0);
        else if(Small == "디즈니 프린세스")
            m_ismall.SetCurSel(1);
        else if(Small == "뽀통령")
            m_ismall.SetCurSel(2);
        else if(Small == "미미")
            m_ismall.SetCurSel(3);
        else if(Small == "바비")
            m_ismall.SetCurSel(4);
        else if(Small == "인사이드아웃")
            m_ismall.SetCurSel(5);
        else if(Small == "쥬쥬")
            m_ismall.SetCurSel(6);
        else if(Small == "콩순이")
            m_ismall.SetCurSel(7);
        else if(Small == "헬로키티")
            m_ismall.SetCurSel(8);
    }
    else if(Middle == "남아캐릭터"){
        m_imedium.SetCurSel(2);
        OnCbnSelchangeCombo4();
        if(Small == "건담")
            m_ismall.SetCurSel(0);
        else if(Small == "또봇")
            m_ismall.SetCurSel(1);
        else if(Small == "마인크래프트")
            m_ismall.SetCurSel(2);
        else if(Small == "미니언즈")
            m_ismall.SetCurSel(3);
        else if(Small == "어벤져스")
            m_ismall.SetCurSel(4);
        else if(Small == "원피스")
            m_ismall.SetCurSel(5);
    }
}

#include "ItemPercentDlg.h"
void CItemPage::OnBnClickedPercent()  //입력 다이얼로그 띄우기
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

    CItemPercentDlg pDlg;
    pDlg.DoModal();

    CString setper;
    CString setper1;
    CString editresult;

    GetDlgItemText(IDC_EDIT15,setper);
    setper1 = pDlg.m_percent;
    editresult.Format(_T("%.2f"), _ttof(setper)+_ttof(setper1) * _ttof(setper));
    m_outprice_Edit.SetWindowTextW(editresult);
}

void CItemPage::OnPaint()  //기본 이미지 띄우기
{
    CPaintDC dc(this); // device context for painting
    // TODO: 여기에 메시지 처리기 코드를 추가합니다.

    if(bFirst){
        CString img;

```

```

        img.Format(_T("res\\WWNoImage.bmp"));
        DrawBMP(img);
    }

    // 그리기 메시지에 대해서는 CPropertyPage::OnPaint()을(를) 호출하지 마십시오.
}

void CItemPage::OnBnClickedImageDelete() //설정 한 이미지 취소
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    CString img;
    img.Format(_T("res\\WWNoImage.bmp"));
    DrawBMP(img);
}

```

### [상품관리 - 상품조회]

```

void CItemSearchPage::OnBnClickedItemSearchbtn() //상품 정보 조회
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(true);

    if(m_is_list.GetItemCount() > 0) m_is_list.DeleteAllItems();

    CDatabase db; // 데이터베이스 객체 생성
    CRecordset rs(&db); // 레코드셋 객체 생성
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access Driver
            (*.mdb)};DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

        if (db.OpenEx(strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery;

        int nIndex = m_icombosearch.GetCurSel();
        if(nIndex == 0){
            strQuery.Format(_T("SELECT item_id, i_code, i_name, i_large,
                i_medium, i_small, c_name, c_code, i_inputDate,i_inputAmount,
                i_inputPrice, i_outputPrice FROM Item WHERE i_code LIKE
                '%%s%%' OR i_name LIKE '%%s%%' OR i_large LIKE '%%s%%' OR
                i_medium LIKE '%%s%%' OR i_small LIKE '%%s%%' OR item_id
                LIKE '%%s%%' OR i_inputDate LIKE '%%s%%'"), m_isearch,
                m_isearch,m_isearch,m_isearch,m_isearch,m_isearch,m_isearch);
            result.Format(_T("전체검색-( '%s' )검색시_"),m_isearch);
        }
        if(nIndex == 1){
            strQuery.Format(_T("SELECT item_id, i_code, i_name, i_large,
                i_medium, i_small, c_name, c_code, i_inputDate, i_inputAmount,
                i_inputPrice, i_outputPrice FROM Item WHERE i_code LIKE
                '%%s%%'"), m_isearch);
            result.Format(_T("상품 코드-( '%s' )검색시_"),m_isearch);
        }
        if(nIndex == 2){
            strQuery.Format(_T("SELECT item_id, i_code, i_name, i_large,
                i_medium, i_small, c_name, c_code, i_inputDate, i_inputAmount,
                i_inputPrice, i_outputPrice FROM Item WHERE i_name LIKE
                '%%s%%'"), m_isearch);
            result.Format(_T("상품명-( '%s' )검색시_"),m_isearch);
        }
    }
}

```

```

    }
    if(nIndex == 3){
        strQuery.Format(_T("SELECT item_id, i_code, i_name, i_large,
        i_medium, i_small, c_name, c_code, i_inputDate, i_inputAmount,
        i_inputPrice, i_outputPrice FROM Item WHERE c_code LIKE
        '%%%s%%'",m_isearch));
        result.Format(_T("거래처 코드-('%s')검색시_"),m_isearch);
    }
    if (rs.Open (CRecordset::snapshot, strQuery) == FALSE)
        throw _T("쿼리전송 실패!");
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}

short col = rs.GetODBCFieldCount(); //쿼리문을 이용해 검색한 내용 리스트 컨트롤에
                                   출력
for(int p=0 ; p<2 ; p++)
{
    while( ! rs.IsEOF())
    {
        CString tmp;
        rs.GetFieldValue((short)p, tmp);
        m_is_list.InsertItem(p,tmp,0);
        for(short i=1; i<col; i++ )
        {
            rs.GetFieldValue(i, tmp);
            m_is_list.SetItemText(p, i ,tmp);
        }
        rs.MoveNext();
    }
}
rs.Close();
db.Close();
}
void CItemSearchPage::OnItemDelete() //삭제 명령 처리기
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    int ret;
    ret = AfxMessageBox(_T("삭제 하시겠습니까?"), MB_YESNO|MB_ICONSTOP);
    if(ret == IDYES){
        UpdateData(TRUE);

        CDatabase db; // 데이터베이스 객체 생성
        CRecordset rs (&db); // 레코드셋 객체 생성

        try
        {
            CString strConnection(L"DRIVER={Microsoft Access Driver
            (*.mdb)};DBQ=C:\\WWW\\Download\\WWWToyPocketDB.mdb;");

            if (db.OpenEx (strConnection) == FALSE)
                throw _T("데이터베이스 접속실패!");

            CString strQuery; //데이터베이스에 전송할 쿼리문
            strQuery.Format(_T("DELETE * FROM Item WHERE i_code LIKE '%%%s%%'",icode);

```

```

        db.ExecuteSQL(strQuery);           //데이터베이스에 쿼리문 전송
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox (msg);
    }

    rs.Close();           //레코드셋 객체 소멸
    db.Close();           //데이터베이스 객체 소멸
}

if(m_is_list.GetItemCount() > 0) m_is_list.DeleteAllItems();

CDatabase db1;           // 데이터베이스 객체 생성
CRecordset rs1 (&db1); // 레코드셋 객체 생성
try
{
    CString strConnection(L"DRIVER={Microsoft Access Driver
        (*.mdb)};DBQ=C:\\\\Download\\\\ToyPocketDB.mdb;");

    if (db1.OpenEx (strConnection) == FALSE)
        throw _T("데이터베이스 접속실패!");

    CString strQuery;
    strQuery.Format(_T("SELECT item_id, i_code, i_name, i_large,
        i_medium, i_small, c_name, c_code, i_inputDate, i_inputAmount,
        i_inputPrice, i_outputPrice FROM Item"));

    if (rs1.Open (CRecordset::snapshot, strQuery) == FALSE)
        throw _T("쿼리전송 실패!");
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}

//쿼리문을 이용해 검색한 내용 리스트 컨트롤에 출력
short col1 = rs1.GetODBCFieldCount();
for(int p=0 ; p<2 ; p++)
{
    while( ! rs1.IsEOF())           //행단위
    {
        //속성단위
        CString tmp;
        rs1.GetFieldValue((short)p, tmp);
        //id값이 기준이 된다.
        m_is_list.InsertItem(p,tmp,0);
        for(short i=1; i<col1; i++ )
        {
            rs1.GetFieldValue(i, tmp);
            // 첫 번째 인자는 InsertValue를 통해서 가지고 온 기준이 되는 정보,

```

```

        두 번째 인자는 열단위로 하나씩
        m_is_list.SetItemText(p, i, tmp);
    }
    rs1.MoveNext(); //다음행으로 이동
}
}
rs1.Close(); //레코드셋 객체 소멸
db1.Close(); //데이터베이스 객체 소멸
}

void CItemSearchPage::OnNMRClickList1(NMHDR *pNMHDR, LRESULT *pResult)
//상품 정보 삭제 컨텍스트 메뉴
{
    LPNMITEMACTIVATE pNMItemActivate = reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    nListIndex = pNMListView->iItem;

    CPoint ptInScreen;

    GetCursorPos(&ptInScreen); //R버튼 눌린좌표얻음

    if(nListIndex==--1){
    }
    else{
        CMenu muTemp, *pContextMenu;
        muTemp.LoadMenuW( IDR_PopupMenu);
        pContextMenu = muTemp.GetSubMenu(1);
        pContextMenu->TrackPopupMenu(TPM_LEFTALIGN|TPM_RIGHTBUTTON, ptInScreen.x,
                                     ptInScreen.y, this);
    }
    icode = m_is_list.GetItemText(nListIndex,1);

    *pResult = 0;
}
void CItemSearchPage::OnBnClickedExcelSave() //엑셀 저장
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    if (m_is_list.GetItemCount() <= 0)
    {
        AfxMessageBox(_T("조회된 데이터가 0건이므로 엑셀 저장 할 수 없습니다.));
        return;
    }

    CTime timeNow = CTime::GetCurrentTime();
    CString strDateTime;
    strDateTime = timeNow.Format(_T("%Y-%m-%d_%H시%M분%S초"));
    CString strFileName;
    strFileName.Format(_T("%s상품 조회_%s"), result, strDateTime);

    CFileDialog fdlg(FALSE, _T("xlsx"), strFileName+_T(".xlsx"), OFN_HIDEREADONLY
        | OFN_OVERWRITEPROMPT | OFN_NOCHANGEDIR,
        _T("xlsx 파일 (*.xlsx)|*.xlsx"), NULL);
    if (fdlg.DoModal() != IDOK) return;

    CXLEzAutomation xls(FALSE); // FALSE : 처리 과정을 화면에 보이지 않는다.

    int rowOffset = 2;
    int colOffset = 1;

    int rowCount = m_is_list.GetItemCount();
    CHeaderCtrl* pHeaderCtrl = m_is_list.GetHeaderCtrl();

```

```

int colCount = pHeaderCtrl->GetItemCount();
int rowIdx;
int colIdx;

CString strItemText;
for (colIdx = 0; colIdx < colCount; colIdx++){
    HDITEM hdi;
    hdi.mask = HDI_TEXT;
    hdi.pszText = strItemText.GetBuffer( 256 );
    hdi.cchTextMax = 256;
    pHeaderCtrl->GetItem( colIdx, &hdi );
    strItemText.ReleaseBuffer();
    xls.SetCellValue(colIdx + colOffset, 0 + rowOffset, strItemText);
}

for (rowIdx = 0; rowIdx < rowCount; rowIdx++)
{
    for (colIdx = 0; colIdx < colCount; colIdx++)
    {
        strItemText = m_is_list.GetItemText(rowIdx, colIdx);

        // 컬럼 데이터
        xls.SetCellValue(colIdx + colOffset, rowIdx + 1 + rowOffset,
            strItemText);
    }
}

xls.SaveFileAs(fdlg.GetPathName());
xls.ReleaseExcel();
}

```

### [상품관리 - 상품수정]

```

void CItemModifyPage::OnBnClickedItemModify() //수정버튼 이벤트 처리기
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(TRUE);
    if(m_icode!=""&&m_iname!=""&&m_icname!=""&&m_icode!=" "
        &&m_icount!=""&&m_input!=""&&m_output!=""){

        CString s;
        COleDateTime selectDate;
        m_idate.GetTime(selectDate); //datetimepicker 컨트롤 변수
        CString sdate = selectDate.Format(_T( "%Y-%m-%d" ));

        CString strlarge;//대분류
        strlarge = SelectLarge(m_ilarge.GetCurSel());
        CString strmid;
        strmid = SelectMedium(m_imedium.GetCurSel()); //중분류
        CString strsmall;
        strsmall = SelectSmall(m_ilarge.GetCurSel(), m_imedium.GetCurSel(),
            m_ismall.GetCurSel()); //소분류

        CDatabase db; // 데이터베이스 객체 생성
        CRecordset rs (&db); // 레코드셋 객체 생성
        try
        {
            CString strConnection(L"DRIVER={Microsoft Access Driver
                (*.mdb)};DBQ=C:\\\\Download\\\\ToyPocketDB.mdb;"); //DB오픈

            if (db.OpenEx (strConnection) == FALSE)

```



```

        throw _T("데이터베이스 접속실패!");

        CString strQuery;          //데이터베이스에 전송할 쿼리문
        strQuery.Format(_T("UPDATE Item SET i_code='%s', i_name='%s',
        i_large='%s', i_medium='%s', i_small='%s', item_id='%s',
        c_name='%s', c_code='%s', i_image='%s', i_inputDate='%s',
        i_inputAmount='%s', i_inputPrice='%s', i_outputPrice='%s'
        WHERE i_code='%s'"), m_icode, m_iname, strlarge, strmid,
        strsmall, g_setid, m_icname, m_iccode, imgname, sdate,
        m_icount, m_iinput, m_ioutput, m_icode);

        db.ExecuteSQL(strQuery);      //데이터베이스에 쿼리문 전송
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox (msg);
    }
    MessageBox(_T("성공적으로 수정되었습니다. "));

    rs.Close();          //레코드셋 객체 소멸
    db.Close();          //데이터베이스 객체 소멸
} else MessageBox(_T("입력되지 않은 값이 있습니다. \n모든 값을
        입력(선택)해주세요. "));

    m_icode = _T("");
    m_iname = _T("");
    m_iccode = _T("");
    m_icname = _T("");
    m_icount = _T("");
    m_iinput = _T("");
    m_ioutput = _T("");
    //생일초기화
    CDateTimeCtrl *pCtrl = (CDateTimeCtrl *)
        GetDlgItem(IDC_DATETIMEPICKER2);
    CTime tm = CTime::GetCurrentTime();
    pCtrl->SetTime(&tm);

    m_icode.SetCurSel(-1);
    m_imedium.SetCurSel(-1);
    m_ismall.SetCurSel(-1);

    //이미지초기화
    CString img;
    img.Format(_T("res\\NoImage.bmp"));
    DrawBMP( img);

    UpdateData(false);

}

//이미지 등록 이벤트처리기
void CItemModifyPage::OnBnClickedImageSearch()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(true);

    CFileDialog sDlg(false, NULL, NULL, OFN_HIDEREADONLY, 0);

```



```

        m_ismall.SetCurSel(1);
    else if(g_iSmall == "축각놀이/오뎅이")
        m_ismall.SetCurSel(2);
    else if(g_iSmall == "아기체육관/러닝홈")
        m_ismall.SetCurSel(3);
    else if(g_iSmall == "보행기/바운서/쏘서")
        m_ismall.SetCurSel(4);
    }
    else if(g_iMiddle == "유아"){
        m_imedium.SetCurSel(1);
        OnCbnSelchangeCombo4();
        if(g_iSmall == "공놀이")
            m_ismall.SetCurSel(0);
        else if(g_iSmall == "목욕놀이")
            m_ismall.SetCurSel(1);
        else if(g_iSmall == "병원/역할놀이")
            m_ismall.SetCurSel(2);
        else if(g_iSmall == "악기놀이")
            m_ismall.SetCurSel(3);
        else if(g_iSmall == "자동차놀이")
            m_ismall.SetCurSel(4);
    }
}

else if(g_iLarge == "여아"){
    m_ilarge.SetCurSel(1);
    OnCbnSelchangeCombo2();
    if(g_iMiddle == "역할놀이"){
        m_imedium.SetCurSel(0);
        OnCbnSelchangeCombo4();
        if(g_iSmall == "소꿉/주방놀이")
            m_ismall.SetCurSel(0);
        else if(g_iSmall == "마트놀이")
            m_ismall.SetCurSel(1);
        else if(g_iSmall == "엄마/청소놀이")
            m_ismall.SetCurSel(2);
        else if(g_iSmall == "병원/기타놀이")
            m_ismall.SetCurSel(3);
    }
    else if(g_iMiddle == "인형"){
        m_imedium.SetCurSel(1);
        OnCbnSelchangeCombo4();
        if(Small == "공주인형")
            m_ismall.SetCurSel(0);
        else if(Small == "인형놀이")
            m_ismall.SetCurSel(1);
        else if(Small == "아기인형")
            m_ismall.SetCurSel(2);
        else if(Small == "캐릭터봉제인형")
            m_ismall.SetCurSel(3);
        else if(Small == "동물봉제인형")
            m_ismall.SetCurSel(4);
        else if(Small == "작동봉제인형")
            m_ismall.SetCurSel(5);
    }
    else if(g_iMiddle == "꾸미기"){
        m_imedium.SetCurSel(2);
        OnCbnSelchangeCombo4();
        if(g_iSmall == "드레스 및 소품")
            m_ismall.SetCurSel(0);
        else if(g_iSmall == "헤어/화장놀이")
            m_ismall.SetCurSel(1);
    }
}

```

```

        else if(g_iSmall == "핸드폰/디카/기타")
            m_small.SetCurSel(2);
    }
}
else if(g_iLarge == "남아"){
    m_iLarge.SetCurSel(2);
    OnCbnSelchangeCombo2();
    if(g_iMiddle == "역할놀이"){
        m_imedium.SetCurSel(0);
        OnCbnSelchangeCombo4();
        if(g_iSmall == "공구놀이")
            m_small.SetCurSel(0);
        else if(g_iSmall == "카드놀이")
            m_small.SetCurSel(1);
        else if(g_iSmall == "품건놀이")
            m_small.SetCurSel(2);
        else if(g_iSmall == "코스튬플레이")
            m_small.SetCurSel(3);
        else if(g_iSmall == "요요/팽이놀이")
            m_small.SetCurSel(4);
        else if(g_iSmall == "핑거/기타놀이")
            m_small.SetCurSel(5);
    }
    else if(g_iMiddle == "액션"){
        m_imedium.SetCurSel(1);
        OnCbnSelchangeCombo4();
        if(g_iSmall == "변신로봇/탈것")
            m_small.SetCurSel(0);
        else if(g_iSmall == "액션무기")
            m_small.SetCurSel(1);
        else if(g_iSmall == "액션피규어")
            m_small.SetCurSel(2);
    }
    else if(g_iMiddle == "조종"){
        m_imedium.SetCurSel(2);
        OnCbnSelchangeCombo4();
        if(g_iSmall == "명품자동차RC")
            m_small.SetCurSel(0);
        else if(g_iSmall == "일반자동차RC")
            m_small.SetCurSel(1);
        else if(g_iSmall == "캐릭터RC")
            m_small.SetCurSel(2);
        else if(g_iSmall == "드론/헬기/보트/기타RC")
            m_small.SetCurSel(3);
    }
}
else if(g_iLarge == "캐릭터별"){
    m_iLarge.SetCurSel(3);
    OnCbnSelchangeCombo2();
    if(g_iMiddle == "유아캐릭터"){
        m_imedium.SetCurSel(0);
        OnCbnSelchangeCombo4();
        if(g_iSmall == "그라미")
            m_small.SetCurSel(0);
        else if(g_iSmall == "그린토이즈")
            m_small.SetCurSel(1);
        else if(g_iSmall == "라바")
            m_small.SetCurSel(2);
        else if(g_iSmall == "로보카폴리")
            m_small.SetCurSel(3);
        else if(g_iSmall == "뽀로로")
            m_small.SetCurSel(4);
    }
}

```

```

        else if(g_iSmall == "라마즈")
            m_ismall.SetCurSel(5);
    }
    else if(g_iMiddle == "여아캐릭터"){
        m_imedium.SetCurSel(1);
        OnCbnSelchangeCombo4();
        if(g_iSmall == "겨울왕국")
            m_ismall.SetCurSel(0);
        else if(g_iSmall == "디즈니 프린세스")
            m_ismall.SetCurSel(1);
        else if(g_iSmall == "뽕뽕이")
            m_ismall.SetCurSel(2);
        else if(g_iSmall == "미미")
            m_ismall.SetCurSel(3);
        else if(g_iSmall == "바비")
            m_ismall.SetCurSel(4);
        else if(g_iSmall == "인사이드아웃")
            m_ismall.SetCurSel(5);
        else if(g_iSmall == "쥬쥬")
            m_ismall.SetCurSel(6);
        else if(g_iSmall == "콩순이")
            m_ismall.SetCurSel(7);
        else if(g_iSmall == "헬로키티")
            m_ismall.SetCurSel(8);
    }
    else if(g_iMiddle == "남아캐릭터"){
        m_imedium.SetCurSel(2);
        OnCbnSelchangeCombo4();
        if(g_iSmall == "건담")
            m_ismall.SetCurSel(0);
        else if(g_iSmall == "또봇")
            m_ismall.SetCurSel(1);
        else if(g_iSmall == "마인크래프트")
            m_ismall.SetCurSel(2);
        else if(g_iSmall == "미니언즈")
            m_ismall.SetCurSel(3);
        else if(g_iSmall == "어벤저스")
            m_ismall.SetCurSel(4);
        else if(g_iSmall == "원피스")
            m_ismall.SetCurSel(5);
    }
}

CString CItemModifyPage::SelectLarge(int index) //대분류 문자열 반환
{
    switch(index)
    {
    case 0 :
        temp = _T("신생아/유아");
        break;
    case 1 :
        temp = _T("여아");
        break;
    case 2 :
        temp = _T("남아");
        break;
    case 3 :
        temp = _T("캐릭터별");
        break;
    }
    return temp;
}

```

```

CString CItemModifyPage::SelectMedium( int index) //중분류 문자열 반환
{
    int select1 = m_iLarge.GetCurSel();
    if(select1 == 0)
    switch(index){
        case 0:
            temp = _T("신생아");
            break;
        case 1:
            temp = _T("유아");
            break;
    }
    else if(select1 == 1)
    switch(index){
        case 0:
            temp = _T("역할놀이");
            break;
        case 1:
            temp = _T("인형");
            break;
        case 2:
            temp = _T("꾸미기");
            break;
    }
    else if(select1 == 2)
    switch(index){
        case 0:
            temp = _T("역할놀이");
            break;
        case 1:
            temp = _T("액션");
            break;
        case 2:
            temp = _T("조종");
            break;
        case 3:
            temp = _T("자동차/기차");
            break;
    }
    else if(select1 == 3)
    switch(index){
        case 0:
            temp = _T("유아캐릭터");
            break;
        case 1:
            temp = _T("여아캐릭터");
            break;
        case 2:
            temp = _T("남아캐릭터");
            break;
    }
    return temp;
}

```

```

CString CItemModifyPage::SelectSmall( int bigc, int midc, int smallc) //소분류 문자열 반환
{
    if(bigc == 0)
    switch(midc){
        case 0: {
            switch(smallc){
                case 0: temp = _T("치발기/팔랑이"); break;
                case 1: temp = _T("모빌"); break;
            }
        }
    }
}

```

```

        case 2: temp = _T("축각놀이/오뎅이"); break;
        case 3: temp = _T("아기체육관/러닝홈"); break;
        case 4: temp = _T("보행기/바운서/쏘서"); break;
    }break; }
case 1: {
    switch(smallc){
        case 0: temp = _T("공놀이"); break;
        case 1: temp = _T("목욕놀이"); break;
        case 3: temp = _T("병원/역할놀이"); break;
        case 4: temp = _T("악기놀이"); break;
        case 5: temp = _T("자동차놀이"); break;
    }break; }
}
else if(bigc == 1)
switch(midc){
    case 0: {
        switch(smallc){
            case 0: temp = _T("소꿉/주방놀이"); break;
            case 1: temp = _T("마트놀이"); break;
            case 2: temp = _T("엄마/청소놀이"); break;
            case 3: temp = _T("병원/기타놀이"); break;
        }break; }
    case 1: {
        switch(smallc){
            case 0: temp = _T("공주인형"); break;
            case 1: temp = _T("인형놀이"); break;
            case 3: temp = _T("아기인형"); break;
            case 4: temp = _T("캐릭터봉제인형"); break;
            case 5: temp = _T("동물봉제인형"); break;
            case 6: temp = _T("작동봉제인형"); break;
        }break; }
    case 2: {
        switch(smallc){
            case 0: temp = _T("드레스 및 소품"); break;
            case 1: temp = _T("헤어/화장놀이"); break;
            case 3: temp = _T("핸드폰/디카/기타"); break;
        }break; }
    }
else if(bigc == 2)
switch(midc){
    case 0: {
        switch(smallc){
            case 0: temp = _T("공구놀이"); break;
            case 1: temp = _T("카드놀이"); break;
            case 2: temp = _T("폼건놀이"); break;
            case 3: temp = _T("코스튠플레이"); break;
            case 4: temp = _T("요요/팽이놀이"); break;
        }
    case 5: temp = _T("필거/기타놀이"); break;
    }break; }
    case 1: {
        switch(smallc){
            case 0: temp = _T("변신로봇/탈것"); break;
            case 1: temp = _T("액션무기"); break;
            case 3: temp = _T("액션피규어"); break;
        }break; }
    case 2: {
        switch(smallc){
            case 0: temp = _T("명품자동차RC"); break;
            case 1: temp = _T("일반자동차RC"); break;
            case 3: temp = _T("캐릭터RC"); break;
            case 4: temp = _T("드론/헬기/보트/기타RC"); break;
        }break; }
    }
}

```

```

        case 3: {
            switch(smallc){
                case 0: temp = _T("빅카/트럭/중장비"); break;
                case 1: temp = _T("일반자동차"); break;
                case 2: temp = _T("수집자동차"); break;
                case 3: temp = _T("자동차놀이세트"); break;
                case 4: temp = _T("비행기/기차/기타"); break;
            }break; }
    }
    else if(bigc == 3)
    switch(midc){
        case 0: {
            switch(smallc){
                case 0: temp = _T("그라미"); break;
                case 1: temp = _T("그린토이즈"); break;
                case 2: temp = _T("라바"); break;
                case 3: temp = _T("로보카폴리"); break;
                case 4: temp = _T("뽀로로"); break;
                case 5: temp = _T("라마즈"); break;
            }break; }
        case 1: {
            switch(smallc){
                case 0: temp = _T("겨울왕국"); break;
                case 1: temp = _T("디즈니 프린세스"); break;
                case 2: temp = _T("똥똥이"); break;
                case 3: temp = _T("미미"); break;
                case 4: temp = _T("바비"); break;
                case 5: temp = _T("인사이드아웃"); break;
                case 6: temp = _T("쥬쥬"); break;
                case 7: temp = _T("콩순이"); break;
                case 8: temp = _T("헬로키티"); break;
            }break; }
        case 2: {
            switch(smallc){
                case 0: temp = _T("건담"); break;
                case 1: temp = _T("또봇"); break;
                case 3: temp = _T("마인크래프트"); break;
                case 4: temp = _T("미니언즈"); break;
                case 5: temp = _T("어벤져스"); break;
                case 6: temp = _T("원피스"); break;
            }break; }
    }
    return temp;
}

#include "ItemPercentDlg.h"
//%입력 버튼 이벤트 처리기
void CItemModifyPage::OnBnClickedPercent()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    CItemPercentDlg pDlg;
    pDlg.DoModal();

    CString setper;
    CString setper1;
    CString editresult;

    GetDlgItemText(IDC_EDIT15,setper);
    setper1 = pDlg.m_percent;
    editresult.Format(_T("%d"), _ttoi(setper1) * _ttoi(setper));
    m_ioutputEdit.SetWindowTextW(editresult);
    //매입가에 %된 값이 판매가로 들어감

```



```

}

#include "CompanySearchDlg.h"
//거래처명 검색 이벤트 처리기
void CItemModifyPage::OnBnClickedComSearchBtn()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    CCompanySearchDlg pDlg;
    pDlg.DoModal();
    m_icname_Edit.SetWindowTextW(pDlg.m_cnameset);
    m_iccode_Edit.SetWindowTextW(pDlg.m_ccodeset);
}

//이미지 삭제 이벤트처리기
void CItemModifyPage::OnBnClickedImageCancel()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    CString img;
    img.Format(_T("res\\NoImage.bmp"));
    DrawBMP(img);
}

```

### [상품관리 - 추가입고]

```

void CItemPlusPage::OnBnClickedItemPlus() //상품 추가 입고 등록
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(TRUE);
    if(m_iname!=""){

        CString m_icount;
        m_icount.Format(_T("%d"),_ttoi(m_iplus_amount) + _ttoi(m_inamount));

        CDatabase db; // 데이터베이스 객체 생성
        CRecordset rs (&db); // 레코드셋 객체 생성
        try
        {
            CString strConnection(L"DRIVER={Microsoft Access Driver
            (*.mdb)};DBQ=C:\\WWW\\Download\\WWWToyPocketDB.mdb;"); //DB오픈

            if (db.OpenEx (strConnection) == FALSE)
                throw _T("데이터베이스 접속실패!");

            CString strQuery; //데이터베이스에 전송할 쿼리문
            strQuery.Format(_T("UPDATE Item SET i_inputAmount='%s' WHERE
            i_name='%s'"),m_icount, m_iname);
            db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송
        }
        catch (CException * e)
        {
            e->ReportError();
            e->Delete();
        }
        catch (LPCTSTR msg)
        {
            AfxMessageBox (msg);
        }
        MessageBox(_T("성공적으로 추가되었습니다. "));

        rs.Close(); //레코드셋 객체 소멸
    }
}

```

```

db.Close(); //데이터베이스 객체 소멸
}else MessageBox(_T("입력되지 않은 값이 있습니다.\n모든 값을 입력(선택)해주세요.));
m_icode_Edit.SetWindowText(_T(""));
m_cname_Edit.SetWindowText(_T(""));
m_ccode_Edit.SetWindowText(_T(""));
m_first.SetWindowText(_T(""));
m_second.SetWindowText(_T(""));
m_third.SetWindowText(_T(""));
m_inprice_Edit.SetWindowText(_T(""));
m_outprice_Edit.SetWindowText(_T(""));
m_date.SetWindowText(_T(""));
m_iplus_amount = _T("");
m_inamount = _T("");
m_iname = _T("");
UpdateData(false);
}
//상품명 검색 버튼 클릭시
void CItemPlusPage::OnBnClickedSearchButton()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    CItemSearchDlg pDlg;
    pDlg.DoModal();
    m_iname_Edit.SetWindowText(pDlg.m_iname);
    m_icode_Edit.SetWindowText(pDlg.m_icode);
    m_cname_Edit.SetWindowText(pDlg.m_cname);
    m_ccode_Edit.SetWindowText(pDlg.m_ccode);
    m_inprice_Edit.SetWindowText(pDlg.m_inprice);
    m_outprice_Edit.SetWindowText(pDlg.m_outprice);
    m_inamount_Edit.SetWindowText(pDlg.m_inamount);
    m_first.SetWindowText(pDlg.m_combo1);
    m_second.SetWindowText(pDlg.m_combo2);
    m_third.SetWindowText(pDlg.m_combo3);
    m_date.SetWindowText(pDlg.m_inDate);
}

```

### [판매관리 - 판매등록]

```

void CSalePage::OnBnClickedRegister() //판매 등록
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    if(!UpdateData()) return;
    UpdateData(TRUE);

    //판매날짜
    CString sdate;
    sdate.Format(_T("%d-%02d-%02d"),m_sdate.GetYear(),m_sdate.GetMonth(),m_sdate.GetDay());

    CDatabase db; // 데이터베이스 객체 생성
    CRecordset rs (&db); // 레코드셋 객체 생성

    CString amount; //재고수량계산
    amount.Format(_T("%d"), _ttoi(m_icount) - _ttoi(m_samount));

    CString total_price;
    total_price.Format(_T("%d"), _ttoi(m_samount) * _ttoi(m_iOutputprice));
    CString s;
    if(_ttoi(m_samount) < _ttoi(m_icount)){ //판매수량 < 상품수량
        try
        {

```

```

CString strConnection(L"DRIVER={Microsoft Access Driver
(*.mdb)};DBQ=C:\\WWWDownload\\WWToypocketDB.mdb;");

if (db.OpenEx (strConnection) == FALSE)
    throw _T("데이터베이스 접속실패!");

CString strQuery;        //데이터베이스에 전송할 쿼리문
strQuery.Format(_T("INSERT INTO Sale (s_outputDate ,
in_ccode, out_ccode, i_code , s_outputAmount, i_inputPrice,
i_outputPrice, order_total, return_YN, return_date,
return_reson, exchange_YN, exchange_date, exchange_reason,
delivery_YN, delivery_process, sale_id) VALUES('%s', '%s',
'%s', '%s', '%s', '%s', '%s', '아니오', '해당없음',
'해당없음', '아니오', '해당없음', '해당없음', '아니오',
'해당없음', '%s')"), sdate, m_iccode, m_outccode,
m_si_code, m_samount, m_iinputprice, m_iOutputprice,
total_price , g_setid);
db.ExecuteSQL(strQuery);        //데이터베이스에 쿼리문 전송

strQuery.Format(_T("UPDATE Item SET i_inputAmount = '%s'
WHERE i_code = '%s'"), amount, m_icode);
db.ExecuteSQL(strQuery);        //데이터베이스에 쿼리문 전송
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}

rs.Close();        //레코드셋 객체 소멸
db.Close();        //데이터베이스 객체 소멸

AfxMessageBox(_T("판매되었습니다. "));
m_ssearch = _T("");
m_si_code = _T("");
m_si_name = _T("");
m_total_amount = _T("");
m_outname = _T("");
m_samount = _T("");
//생일초기화
CDateTimeCtrl *pCtrl = (CDateTimeCtrl *)
    GetDlgItem(IDC_DATETIMEPICKER3);
CTime tm = CTime::GetCurrentTime();
pCtrl->SetTime(&tm);

m_scombosearch.SetCurSel(0);
m_combo1.SetCurSel(-1);
m_combo2.SetCurSel(-1);
m_combo3.SetCurSel(-1);

//이미지초기화
CString img;
img.Format(_T("res\\NoImage.bmp"));
DrawBMP( img);

}else{        //판매수량이 재고수량보다 많을 때
AfxMessageBox(_T("판매될 수 없습니다. "));
m_samount = _T("");

```

```

    }

    UpdateData(false);
}

void CSalePage::OnNMDoubleClickList1(NMHDR *pNMHDR, LRESULT *pResult) //상품 정보 리스트
더블클릭 시 이벤트 처리기
{
    LPNMITEMACTIVATE pNMItemActivate =
        reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    nIndex = pNMListView->iItem;

    m_si_code.SetWindowTextW(m_si_List.GetItemText(nListIndex,1));
    m_si_name.SetWindowTextW(m_si_List.GetItemText(nListIndex,2));
    m_totalamount_Edit.SetWindowTextW(m_si_List.GetItemText(nListIndex,9));
    GetDlgItem(IDC_SCOMB01)->SetWindowTextW(m_si_List.GetItemText(nListIndex,3));
    GetDlgItem(IDC_SCOMB02)->SetWindowTextW(m_si_List.GetItemText(nListIndex,4));
    GetDlgItem(IDC_SCOMB03)->SetWindowTextW(m_si_List.GetItemText(nListIndex,5));

    m_i_code= m_si_List.GetItemText(nListIndex,1);
    m_i_count = m_si_List.GetItemText(nListIndex,9);
    m_i_code = m_si_List.GetItemText(nListIndex,7);
    m_i_inputprice = m_si_List.GetItemText(nListIndex,10);
    m_i_outputprice = m_si_List.GetItemText(nListIndex,11);

    imgname.Format(_T("res\\item\\%s.bmp"),m_i_code);
    DrawBMP(imgname);

    *pResult = 0;
}

void CSalePage::OnBnClickedSearchbtn() //상품 정보 리스트 조회
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(true);

    if(m_si_List.GetItemCount() > 0) m_si_List.DeleteAllItems();

    CDatabase db; // 데이터베이스 객체 생성
    CRecordset rs(&db); // 레코드셋 객체 생성
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access Driver
(*.mdb)};DBQ=C:\\WWW\\Download\\WWWToyPocketDB.mdb;");

        if (db.OpenEx(strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery;

        int nIndex = m_scombosearch.GetCurSel();
        if(nIndex == 0){
            strQuery.Format(_T("SELECT item_id, i_code, i_name, i_large,
i_medium, i_small, c_code, i_inputDate,
i_inputAmount, i_inputPrice, i_outputPrice FROM Item WHERE
i_code LIKE '%%s%%' OR i_name LIKE '%%s%%' OR i_large LIKE
'%%s%%' OR i_medium LIKE '%%s%%' OR i_small LIKE '%%s%%'
OR item_id LIKE '%%s%%' OR i_inputDate LIKE '%%s%%'"),
m_ssearch,m_ssearch,m_ssearch,m_ssearch,m_ssearch
,m_ssearch,m_ssearch);
        }
        if(nIndex == 1){

```

```

        strQuery.Format(_T("SELECT item_id, i_code, i_name, i_large,
        i_medium, i_small, c_name, c_code, i_inputDate,
        i_inputAmount, i_inputPrice, i_outputPrice FROM Item WHERE
        i_cname LIKE '%%s%%'"), m_ssearch);
    }
    if(nIndex == 2){
        strQuery.Format(_T("SELECT item_id, i_code, i_name, i_large,
        i_medium, i_small, c_name, c_code, i_inputDate,
        i_inputAmount, i_inputPrice, i_outputPrice FROM Item WHERE
        c_code LIKE '%%s%%'"), m_ssearch);
    }
    if(nIndex == 3){
        strQuery.Format(_T("SELECT item_id, i_code, i_name, i_large,
        i_medium, i_small, c_name, c_code, i_inputDate,
        i_inputAmount, i_inputPrice, i_outputPrice FROM Item WHERE
        i_inputDate LIKE '%%s%%'"), m_ssearch);
    }
    if (rs.Open (CRecordset::snapshot, strQuery) == FALSE)
        throw _T("쿼리전송 실패!");
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}

short col = rs.GetODBCFieldCount();    //쿼리문을 이용해 검색한 내용 리스트
                                        //컨트롤에 출력
for(int p=0 ; p<2 ; p++)
{
    while( ! rs.IsEOF())
    {
        CString tmp;
        rs.GetFieldValue((short)p, tmp);
        m_si_List.InsertItem(p, tmp, 0);
        for(short i=1; i<col; i++ )
        {
            rs.GetFieldValue(i, tmp);
            m_si_List.SetItemText(p, i ,tmp);
        }
        rs.MoveNext();
    }
}
rs.Close();
db.Close();
}
void CSalePage::DrawBMP(LPCTSTR lpszFileName)    //이미지 보여주기
{
    CDC* pDC = GetDC();//화면 DC를 얻어 옴
    CDC MemDC;//메모리 DC
    CBitmap bmp, *pOldBmp;
    BITMAP bm;//비트맵 정보를 저장하는 변수
    CRect rect;
    CPoint cp;

    //이미지 파일 로딩
    bmp.m_hObject = (HBITMAP)LoadImage(NULL, lpszFileName, IMAGE_BITMAP, 0, 0,
    LR_LOADFROMFILE | LR_CREATEDIBSECTION | LR_DEFAULTSIZE);

```

```

    if(bmp.m_hObject == NULL) return;
    bmp.GetBitmap(&bm); //로딩된 이미지 파일의 형태를 얻음
    MemDC.CreateCompatibleDC(pDC); //메모리 DC와 화면 DC를 호환성있도록 연결
    pOldBmp = MemDC.SelectObject(&bmp); //메모리에 적재된 이미지 선택

    m_siimage.GetWindowRect(&rect); //다이얼로그에 배치된 컨트롤의 크기를 얻어 옴
    cp.x = rect.left; //이미지 출력을 위한 시작 x 좌표
    cp.y = rect.top; //이미지 출력을 위한 시작 y 좌표
    ScreenToClient(&cp); //cp가 가지고 있는 좌표로 화면을 지정함
    pDC->StretchBlt(cp.x, cp.y, rect.Width(), rect.Height(), &MemDC, 0, 0,
        bm.bmWidth, bm.bmHeight, SRCCOPY); //이미지를 해당 컨트롤로 전송
    MemDC.SelectObject(pOldBmp);
    ReleaseDC(pDC); //DC 반환
}

#include "SaleComSearchDlg.h"
void CSalePage::OnBnClickedSearchbtn2() //거래처 검색 다이얼로그 띄우기
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    CSaleComSearchDlg pDlg;
    pDlg.DoModal();
    m_cname_D.SetWindowText(pDlg.m_cnameset);
    m_outccode = pDlg.m_ccodeset;
}

```

### [판매관리 - 판매조회]

```

void CSaleSearchPage::OnBnClickedSaleSearchBtn() //판매 정보 조회
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(true);

    if(m_s_List.GetItemCount() > 0) m_s_List.DeleteAllItems();

    CDatabase db; // 데이터베이스 객체 생성
    CRecordset rs(&db); // 레코드셋 객체 생성
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access Driver
            (*.mdb)};DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

        if (db.OpenEx(strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery;

        int nIndex = m_ssearch_combo.GetCurSel();
        if(nIndex == 0){
            strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate,
                in_ccode,out_ccode, i_code, s_outputAmount, i_inputPrice,
                i_outputPrice, order_total, exchange_YN, exchange_date,
                exchange_reason, return_YN, return_date, return_reson FROM
                Sale WHERE s_outputDate LIKE '%%s%%' OR out_ccode LIKE
                '%%s%%' OR i_code LIKE '%%s%%' OR in_ccode LIKE '%%s%%' OR
                s_outputAmount LIKE '%%s%%' OR i_inputPrice LIKE '%%s%%'
                OR i_outputPrice LIKE '%%s%%' OR order_total LIKE '%%s%%'
                OR return_YN LIKE '%%s%%' OR return_date LIKE '%%s%%' OR
                return_reson LIKE '%%s%%' OR exchange_YN LIKE '%%s%%' OR
                exchange_date LIKE '%%s%%' OR exchange_reason LIKE '%%s%%'
                OR sale_id LIKE '%%s%%'"), m_ssearch, m_ssearch, m_ssearch,
                m_ssearch, m_ssearch, m_ssearch, m_ssearch, m_ssearch, m_ssearch, m_ssearch);
        }
    }
}

```

```

        ,m_ssearch,m_ssearch,m_ssearch,m_ssearch,m_ssearch,m_ssearch);
        result.Format(_T("전체검색-('%s')검색시_"),m_ssearch);
    }
    if(nIndex == 1){
        strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate,
            in_ccode,out_ccode, i_code, s_outputAmount, i_inputPrice,
            i_outputPrice, order_total, exchange_YN, exchange_date,
            exchange_reason, return_YN, return_date, return_reson FROM
            Sale WHERE s_outputDate LIKE '%%s%%'"),m_ssearch);
        result.Format(_T("판매날짜-('%s')검색시_"),m_ssearch);
    }
    if(nIndex == 2){
        strQuery.Format(_T("SELECT s_Key, sale_id,
            s_outputDate,in_ccode,out_ccode, i_code, s_outputAmount,
            i_inputPrice, i_outputPrice, order_total, exchange_YN,
            exchange_date, exchange_reason, return_YN, return_date,
            return_reson FROM Sale WHERE i_code LIKE
            '%%s%%'"),m_ssearch);
        result.Format(_T("상품코드-('%s')검색시_"),m_ssearch);
    }
    if(nIndex == 3){
        strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate,
            in_ccode,out_ccode, i_code, s_outputAmount, i_inputPrice,
            i_outputPrice, order_total, exchange_YN, exchange_date,
            exchange_reason, return_YN, return_date, return_reson FROM
            Sale WHERE out_ccode LIKE '%%s%%'"),m_ssearch);
        result.Format(_T("납품처코드-('%s')검색시_"),m_ssearch);
    }
    if (rs.Open (CRecordset::snapshot, strQuery) == FALSE)
        throw _T("쿼리전송 실패!");
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}

short col = rs.GetODBCFieldCount(); //쿼리문을 이용해 검색한 내용 리스트 컨트롤에 출력
for(int p=0 ; p<2 ; p++)
{
    while( ! rs.IsEOF())
    {
        CString tmp;
        rs.GetFieldValue((short)p, tmp);
        m_s_List.InsertItem(p,tmp,0);
        for(short i=1; i<col; i++ )
        {
            rs.GetFieldValue(i, tmp);
            m_s_List.SetItemText(p, i ,tmp);
        }
        rs.MoveNext();
    }
}
rs.Close();
db.Close();
}

void CSaleSearchPage::OnNMClickList1(NMHDR *pNMHDR, LRESULT *pResult)
//판매 정보 삭제 컨텍스트 메뉴 띄움

```

```

{
    LPNMITEMACTIVATE pNMItemActivate =
        reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    nIndex = pNMListView->iItem;

    CPoint ptInScreen;

    GetCursorPos(&ptInScreen); //R버튼 눌린좌표얻음

    if(nListIndex== -1){
    }
    else{
        CMenu muTemp, *pContextMenu;
        muTemp.LoadMenuW( IDR_PopupMenu);
        pContextMenu = muTemp.GetSubMenu(0);
        pContextMenu -> TrackPopupMenu(TPM_LEFTALIGN|TPM_RIGHTBUTTON, ptInScreen.x,
                                         ptInScreen.y, this);
    }

    m_skey = _ttoi(m_s_List.GetItemText(nListIndex,0));

    *pResult = 0;
}

void CSaleSearchPage::OnSaleDelete() //삭제 명령 처리기
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    int ret;
    ret = AfxMessageBox(_T("삭제하시겠습니까?"), MB_YESNO|MB_ICONSTOP);
    if(ret == IDYES){
        UpdateData(TRUE);

        CDatabase db; // 데이터베이스 객체 생성
        CRecordset rs (&db); // 레코드셋 객체 생성

        try
        {
            CString strConnection(L"DRIVER={Microsoft Access Driver
                                   (*.mdb)};DBQ=C:\\\\Download\\\\ToyPocketDB.mdb;");

            if (db.OpenEx (strConnection) == FALSE)
                throw _T("데이터베이스 접속실패!");

            CString strQuery; //데이터베이스에 전송할 쿼리문
            strQuery.Format(_T("DELETE * FROM Sale WHERE Sale.s_Key= %d
                                "),m_skey);
            db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송

        }
        catch (CException * e)
        {
            e->ReportError();
            e->Delete();
        }
        catch (LPCTSTR msg)
        {
            AfxMessageBox (msg);
        }
    }
}

```



```

        rs.Close();        //레코드셋 객체 소멸
        db.Close();        //데이터베이스 객체 소멸
    }
    if(m_s_List.GetItemCount() > 0) m_s_List.DeleteAllItems();

    CDatabase db;                // 데이터베이스 객체 생성
    CRecordset rs (&db);        // 레코드셋 객체 생성
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access Driver
                                (*.mdb)};DBQ=C:\\\\Download\\\\ToyPocketDB.mdb;");

        if (db.OpenEx (strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery;
        strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate, in_ccode,
                                out_ccode, i_code, s_outputAmount, i_inputPrice, i_outputPrice,
                                order_total, exchange_YN, exchange_date, exchange_reason, return_YN,
                                return_date, return_reson FROM Sale"));

        if (rs.Open (CRecordset::snapshot, strQuery) == FALSE)
            throw _T("쿼리전송 실패!");
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox (msg);
    }

    //쿼리문을 이용해 검색한 내용 리스트 컨트롤에 출력
    short col = rs.GetODBCFieldCount();
    for(int p=0 ; p<2 ; p++)
    {
        while( ! rs.IsEOF())                //행단위
        {
            //속성단위
            CString tmp;
            rs.GetFieldValue((short)p, tmp);
            //id값이 기준이 된다.
            m_s_List.InsertItem(p,tmp,0);
            for(short i=1; i<col; i++ )
            {
                rs.GetFieldValue(i, tmp);
                // 첫번째 인자는 InsertValue를 통해서 가지고 온
                // 기준이 되는 정보, 두번째 인자는 열단위로 하나씩
                m_s_List.SetItemText(p, i ,tmp);
            }
            rs.MoveNext(); //다음행으로 이동
        }
    }
    rs.Close();        //레코드셋 객체 소멸
    db.Close();        //데이터베이스 객체 소멸
}

//더블클릭시 이벤트 처리기 (바로 수정탭으로 이동해서 값 setting)
void CSaleSearchPage::OnNMDoubleClickList1(NMHDR *pNMHDR, LRESULT *pResult)

```

```

{
    LPNMITEMACTIVATE pNMItemActivate = reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    nListIndex = pNMListView->iItem;
    AfxExtractSubString(g_soutDate_Year, m_s_List.GetItemText(nListIndex, 2),
                                                                0, '-');
    AfxExtractSubString(g_soutDate_Month, m_s_List.GetItemText(nListIndex, 2),
                                                                1, '-');
    AfxExtractSubString(g_soutDate_Day, m_s_List.GetItemText(nListIndex, 2),
                                                                2, '-');

    g_soutAmount = m_s_List.GetItemText(nListIndex, 5);
    g_icode = m_s_List.GetItemText(nListIndex, 4);

    CSaleSheet* pSheet = (CSaleSheet*)GetParent();
    pSheet->SetActivePage(2);
    *pResult = 0;
}

void CSaleSearchPage::OnBnClickedExcelSave() //엑셀 저장
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    if (m_s_List.GetItemCount() <= 0)
    {
        AfxMessageBox(_T("조회된 데이터가 0건이므로 엑셀 저장 할 수 없습니다. "));

        return;
    }

    CTime timeNow = CTime::GetCurrentTime();
    CString strDateTime;
    strDateTime = timeNow.Format(_T("%Y-%m-%d_%H시%M분%S초"));
    CString strFileName;
    strFileName.Format(_T("판매 조회_%s"), strDateTime);

    CFileDialog fdlg(FALSE, _T("xlsx"), strFileName+_T(".xlsx"), OFN_HIDEREADONLY
|OFN_OVERWRITEPROMPT | OFN_NOCHANGEDIR, _T("xlsx 파일 (*.xlsx)|*.xlsx"),
                                                                NULL);

    if (fdlg.DoModal() != IDOK) return;

    CXLEzAutomation xls(FALSE); // FALSE : 처리 과정을 화면에 보이지 않는다.

    int rowOffset = 2;
    int colOffset = 1;

    int rowCount = m_s_List.GetItemCount();
    CHeaderCtrl* pHeaderCtrl = m_s_List.GetHeaderCtrl();
    int colCount = pHeaderCtrl->GetItemCount();
    int rowIdx;
    int colIdx;

    CString strItemText;
    for (colIdx = 0; colIdx < colCount; colIdx++){
        HDITEM hdi;
        hdi.mask = HDI_TEXT;
        hdi.pszText = strItemText.GetBuffer( 256 );
        hdi.cchTextMax = 256;
        pHeaderCtrl->GetItem( colIdx, &hdi );
        strItemText.ReleaseBuffer();
        xls.SetCellValue(colIdx + colOffset, 0 + rowOffset, strItemText);
    }
}

```

```

for (rowIdx = 0; rowIdx < rowCount; rowIdx++)
{
    for (colIdx = 0; colIdx < colCount; colIdx++)
    {
        strItemText = m_s_List.GetItemText(rowIdx, colIdx);

        // 컬럼 데이터
        xls.SetCellValue(colIdx + colOffset, rowIdx + 1 + rowOffset,
                        strItemText);
    }
}

xls.SaveFileAs(fdlg.GetPathName());

xls.ReleaseExcel();
}

```

### [교환관리 - 교환등록]

```

void CExchangePage::OnBnClickedSearchbtn() //판매 정보 리스트 조회
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(true);

    if(m_exList.GetItemCount() > 0) m_exList.DeleteAllItems();

    CDatabase db; // 데이터베이스 객체 생성
    CRecordset rs(&db); // 레코드셋 객체 생성
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access Driver
            (*.mdb)};DBQ=C:\\WWWDownload\\WWWToyPocketDB.mdb;");

        if (db.OpenEx(strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery;

        int nIndex = m_exCombo.GetCurSel();
        if(nIndex == 0){
            strQuery.Format(_T("SELECT s_outputDate, c_code, i_code,
                s_outputAmount, i_inputPrice, i_outputPrice, order_total,
                sale_id FROM Sale WHERE (s_outputDate LIKE '%%s%%' OR c_code
                LIKE '%%s%%' OR i_code LIKE '%%s%%' OR s_outputAmount LIKE
                '%%s%%' OR i_inputPrice LIKE '%%s%%' OR i_outputPrice LIKE
                '%%s%%' OR order_total LIKE '%%s%%' OR return_YN LIKE
                '%%s%%' OR return_date LIKE '%%s%%' OR return_reson LIKE
                '%%s%%' OR exchange_YN LIKE '%%s%%' OR exchange_date LIKE
                '%%s%%' OR exchange_reason LIKE '%%s%%' OR delivery_YN LIKE
                '%%s%%' OR delivery_process LIKE '%%s%%' OR sale_id LIKE
                '%%s%%') AND Sale.exchange_YN = '아니오'"), m_exSearch
                ,m_exSearch,m_exSearch,m_exSearch,m_exSearch,m_exSearch,
                m_exSearch,m_exSearch,m_exSearch,m_exSearch,m_exSearch,
                m_exSearch,m_exSearch,m_exSearch,m_exSearch);
        }
        if(nIndex == 1){
            strQuery.Format(_T("SELECT s_outputDate, c_code, i_code,
                s_outputAmount, i_inputPrice, i_outputPrice, order_total,
                sale_id FROM Sale WHERE s_outputDate LIKE '%%s%%' AND
                exchange_YN = '아니오'"), m_exSearch);
        }
    }
}

```

```

        if(nIndex == 2){
            strQuery.Format(_T("SELECT s_outputDate, c_code, i_code,
                                s_outputAmount, i_inputPrice, i_outputPrice, order_total,
                                sale_id FROM Sale WHERE c_code LIKE '%%%S%%' AND exchange_YN
                                                                = '아니오'"),m_exSearch);
        }
        if(nIndex == 3){
            strQuery.Format(_T("SELECT s_outputDate, c_code, i_code,
                                s_outputAmount, i_inputPrice, i_outputPrice, order_total,
                                sale_id FROM Sale WHERE i_code LIKE '%%%S%%' AND exchange_YN
                                                                = '아니오'"),m_exSearch);
        }
        if (rs.Open (CRecordset::snapshot, strQuery) == FALSE)
            throw _T("쿼리전송 실패!");
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox (msg);
    }

    short col = rs.GetODBCFieldCount();    //쿼리문을 이용해 검색한 내용 리스트
                                           //컨트롤에 출력

    for(int p=0 ; p<2 ; p++)
    {
        while( ! rs.IsEOF())
        {
            CString tmp;
            rs.GetFieldValue((short)p, tmp);
            m_exList.InsertItem(p,tmp,0);
            for(short i=1; i<col; i++ )
            {
                rs.GetFieldValue(i, tmp);
                m_exList.SetItemText(p, i ,tmp);
            }
            rs.MoveNext();
        }
    }
    rs.Close();
    db.Close();
}

//등록 버튼 클릭시 이벤트 처리기
void CExchangePage::OnBnClickedOk()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    if(!UpdateData()) return;

    UpdateData(TRUE);

    CString samount, eamount;    //판매수량, 교환수량

    samount = m_sAmount.GetString();
    eamount = m_exAmount.GetString();

    CString sdate;    //판매날짜
    sdate.Format(_T("-%d-%02d-%02d"),m_sdate.GetYear(),m_sdate.GetMonth())

```

```

, m_sdate.GetDay());

CString edate;           //교환날짜
edate.Format(_T("%d-%02d-%02d"), m_exDate.GetYear(),
, m_exDate.GetMonth(), m_exDate.GetDay());

int nIndex;
nIndex = m_exReason_Combo.GetCurSel();
reason = SelectReason(nIndex);

CDatabase db;           // 데이터베이스 객체 생성
CRecordset rs (&db);    // 레코드셋 객체 생성

if(_ttoi(samount) == _ttoi(eamount)){ //판매수량 == 교환수량
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access
Driver (*.mdb)};DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

        if (db.OpenEx (strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery; //데이터베이스에 전송할 쿼리문
        strQuery.Format(_T("UPDATE Sale SET exchange_YN =
'예', exchange_date = '%s', exchange_reason = '%s'
WHERE i_code = '%s' AND s_outputDate='%s'"), edate,
reason, m_sicode, sdate);
        db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox (msg);
    }
    MessageBox(_T("성공적으로 교환되었습니다.));
}
else if(_ttoi(samount) > _ttoi(eamount)){ //판매수량 > 교환수량
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access
Driver (*.mdb)};DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

        if (db.OpenEx (strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery; //데이터베이스에 전송할 쿼리문
        CString amount; //판매수량 - 교환수량
        amount.Format(_T("%d"), _ttoi(m_sAmount) -
_ttoi(m_exAmount));

        strQuery.Format(_T("UPDATE Sale SET s_outputAmount =
'%s' WHERE s_outputDate = '%s' AND i_code = '%s' AND
exchange_YN='아니오'"), amount, sdate, m_sicode);
        db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송
        strQuery.Format(_T("INSERT INTO Sale (s_outputDate,
in_ccode, out_ccode, i_code, s_outputAmount,
i_inputPrice, i_outputPrice, order_total,

```

```

        return_YN, return_date, return_reson, exchange_YN,
        exchange_date, exchange_reason, delivery_YN,
        delivery_process, sale_id) VALUES( '%s', '%s', '%s',
        '%s', '%s', '%s', '%s', '%s', '아니오', '해당없음',
        '해당없음', '예', '%s', '%s', '아니오', '해당없음',
        '%s')", sdate, m_incode, m_sscore, m_siccode, eamount
        , inprice, outprice, total , edate, reason, g_setid);
        db.ExecuteSQL(strQuery);
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox (msg);
    }
    MessageBox(_T("성공적으로 교환되었습니다.));
}
else{
    // 판매수량 < 교환수량
    MessageBox(_T("수량을 다시 확인해주세요.));
}
rs.Close(); //레코드셋 객체 소멸
db.Close(); //데이터베이스 객체 소멸
m_exCombo.SetCurSel(0);
m_exReason_Combo.SetCurSel(-1);
GetDlgItem(IDC_STATIC_INPUTDATE)->ShowWindow(SW_HIDE);
GetDlgItem(IDC_EDIT10)->ShowWindow(SW_HIDE);
m_exSearch = _T("");
m_sccode_Edit.SetWindowTextW(_T(""));
m_siccode_Edit.SetWindowTextW(_T(""));
m_sAmount_Edit.SetWindowTextW(_T(""));
m_exAmount_Edit.SetWindowTextW(_T(""));
CDateTimeCtrl *pCtrl = (CDateTimeCtrl *)
    GetDlgItem(IDC_DATETIMEPICKER4);
CTime tm = CTime::GetCurrentTime();
pCtrl->SetTime(&tm);
CDateTimeCtrl *pCtrl2 = (CDateTimeCtrl *)
    GetDlgItem(IDC_DATETIMEPICKER5);
CTime tm2 = CTime::GetCurrentTime();
pCtrl2->SetTime(&tm2);

//이미지초기화
CString img;
img.Format(_T("res\\NoImage.bmp"));
DrawBMP( img);
}

```

```

CString CExchangePage::SelectReason( int index) //대분류 문자열 반환
{
    switch(index)
    {
    case 0 :
        reason = _T("단순변심");
        break;
    case 1 :
        reason = _T("불량");
        break;
    case 2 :
        etcreason = m_exReason.GetString();
        reason = etcreason;
    }
}

```

```

        break;
    }
    return reason;
}
//판매날짜 콤보박스 클릭시 이벤트처리기
void CExchangePage::OnCbnSel changeScomboSearch()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    int nIndex;
    nIndex = m_exCombo.GetCurSel();    // 현재 선택된 항목의 인덱스를 얻음

    if(nIndex == 1)
    {
        GetDlgItem(IDC_STATIC_INPUTDATE)->ShowWindow(SW_SHOW);
    }
    else
        GetDlgItem(IDC_STATIC_INPUTDATE)->ShowWindow(SW_HIDE);
}
//판매 정보 리스트 더블클릭시 밑에 에디트 박스에 값이 setting
void CExchangePage::OnNMdblclkList1(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMITEMACTIVATE pNMItemActivate = reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    nIndex = pNMListView->iItem;

    m_scode_Edit.SetWindowTextW(m_exList.GetItemText(nListIndex,3));
    m_siccode_Edit.SetWindowTextW(m_exList.GetItemText(nListIndex,4));
    m_sAmount_Edit.SetWindowTextW(m_exList.GetItemText(nListIndex,5));
    AfxExtractSubString(g_set_ibirth_Year,m_exList.GetItemText(nListIndex,1),
        0,'-');
    AfxExtractSubString(g_set_ibirth_Month,m_exList.GetItemText(nListIndex,1),
        1,'-');
    AfxExtractSubString(g_set_ibirth_Day,m_exList.GetItemText(nListIndex,1),
        2,'-');

    COleDateTime Date;    //사용자가 지정한 날짜를 저장할 COleDateTimeDate 변수
    CString wsYear = g_set_ibirth_Year;    //년
    CString wsMonth = g_set_ibirth_Month;    //월
    CString wsDay = g_set_ibirth_Day;    //일
    Date.SetDate(_tstoi(wsYear),_tstoi(wsMonth),_tstoi(wsDay));
    //COleDateTime타입은 UpdateData해줘야함 그럼 나머지만먹힘
    //그래서 변수 지우고 다시 DateTime형식의 컨트롤로 다시만들
    m_sDate_ctrl.SetTime(Date);

    m_icode = m_exList.GetItemText(nListIndex,4);//상품코드
    m_incode = m_exList.GetItemText(nListIndex,2);//납입처코드
    inprice = m_exList.GetItemText(nListIndex,6);
    outprice = m_exList.GetItemText(nListIndex,7);
    total = m_exList.GetItemText(nListIndex,8);
    imgname.Format(_T("res\\Item\\%s.bmp"),m_icode);
    DrawBMP(imgname);

    *pResult = 0;
}
//상품 이미지 보여주기
void CExchangePage::DrawBMP(LPCTSTR lpszFileName)
{
    CDC* pDC = GetDC();//화면 DC를 얻어옴
    CDC MemDC;//메모리 DC

```

```

CBitmap bmp, *pOldBmp;
BITMAP bm; //비트맵 정보를 저장하는 변수
CRect rect;
CPoint cp;

//이미지 파일 로딩
bmp.m_hObject = (HBITMAP)LoadImage(NULL, lpszFileName, IMAGE_BITMAP, 0, 0,
    LR_LOADFROMFILE | LR_CREATEDIBSECTION | LR_DEFAULTSIZE);
if(bmp.m_hObject == NULL)
    return;
bmp.GetBitmap(&bm); //로딩된 이미지 파일의 형태를 얻음
MemDC.CreateCompatibleDC(pDC); //메모리 DC와 화면 DC를 호환성있도록 연결
pOldBmp = MemDC.SelectObject(&bmp); //메모리에 적재된 이미지 선택

m_eiImage.GetWindowRect(&rect); //다이얼로그에 배치된 컨트롤의 크기를 얻어 옴
cp.x = rect.left; //이미지 출력을 위한 시작 x 좌표
cp.y = rect.top; //이미지 출력을 위한 시작 y 좌표
ScreenToClient(&cp); //cp가 가지고 있는 좌표로 화면을 지정함
pDC->StretchBlt(cp.x, cp.y, rect.Width(), rect.Height(), &MemDC, 0, 0,
    bm.bmWidth, bm.bmHeight, SRCCOPY); //이미지를 해당 컨트롤로 전송
MemDC.SelectObject(pOldBmp);
ReleaseDC(pDC); //DC 반환
}

```

### [교환관리 - 교환조회]

```

void CExchangeSearchPage::OnBnClickedSearchBtn() //교환 정보 조회
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(true);

    if(m_ex_list.GetItemCount() > 0) m_ex_list.DeleteAllItems();

    CDatabase db; // 데이터베이스 객체 생성
    CRecordset rs(&db); // 레코드셋 객체 생성
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access Driver
            (*.mdb)};DBQ=C:\\WWWDownload\\WWWToyPocketDB.mdb;");

        if (db.OpenEx(strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery;

        int nIndex = m_ex_Combo.GetCurSel();
        if(nIndex == 0){
            strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate,
                out_ccode, i_code, s_outputAmount, exchange_date,
                exchange_reason, i_inputPrice, i_outputPrice, order_total
                FROM Sale WHERE ( s_outputDate LIKE '%%s%%' OR in_ccode LIKE
                '%%s%%' OR out_ccode LIKE '%%s%%' OR i_code LIKE '%%s%%' OR
                s_outputAmount LIKE '%%s%%' OR i_inputPrice LIKE '%%s%%' OR
                i_outputPrice LIKE '%%s%%' OR order_total LIKE '%%s%%' OR
                exchange_date LIKE '%%s%%' OR exchange_reason LIKE '%%s%%'
                OR sale_id LIKE '%%s%%') AND exchange_YN = '예'"),
                m_ex_Search, m_ex_Search, m_ex_Search, m_ex_Search, m_ex_Search,
                m_ex_Search, m_ex_Search, m_ex_Search, m_ex_Search,
                m_ex_Search, m_ex_Search);
            result.Format(_T("전체검색-( '%s' ) 검색시_"), m_ex_Search);
        }
    }
}

```



```

        if(nIndex == 1){
            strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate,
                                out_ccode, i_code, s_outputAmount, exchange_date,
                                exchange_reason, i_inputPrice, i_outputPrice, order_total
                                FROM Sale WHERE exchange_date LIKE '%%s%%' AND exchange_YN =
                                '예'"),m_ex_Search);
            result.Format(_T("교환날짜-('%s')검색시_"),m_ex_Search);
        }
        if(nIndex == 2){
            strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate,
                                out_ccode, i_code, s_outputAmount, exchange_date,
                                exchange_reason, i_inputPrice, i_outputPrice, order_total
                                FROM Sale WHERE i_code LIKE '%%s%%' AND exchange_YN =
                                '예'"),m_ex_Search);
            result.Format(_T("상품코드-('%s')검색시_"),m_ex_Search);
        }
        if(nIndex == 3){
            strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate,
                                out_ccode, i_code, s_outputAmount, exchange_date,
                                exchange_reason, i_inputPrice, i_outputPrice, order_total
                                FROM Sale WHERE out_ccode LIKE '%%s%%' AND exchange_YN =
                                '예'"),m_ex_Search);
            result.Format(_T("거래처코드-('%s')검색시_"),m_ex_Search);
        }
        if (rs.Open (CRecordset::snapshot, strQuery) == FALSE)
            throw _T("쿼리전송 실패!");
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox (msg);
    }

    short col = rs.GetODBCFieldCount(); //쿼리문을 이용해
                                         검색한 내용 리스트 컨트롤에 출력
    for(int p=0 ; p<2 ; p++)
    {
        while( ! rs.IsEOF())
        {
            CString tmp;
            rs.GetFieldValue((short)p, tmp);
            m_ex_list.InsertItem(p,tmp,0);
            for(short i=1; i<col; i++ )
            {
                rs.GetFieldValue(i, tmp);
                m_ex_list.SetItemText(p, i ,tmp);
            }
            rs.MoveNext();
        }
    }
    rs.Close();
    db.Close();
}

//교환 정보 삭제 컨텍스트 메뉴 띄움
void CExchangeSearchPage::OnNMClickList1(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMITEMACTIVATE pNMItemActivate = reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);

```

```

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
nListIndex = pNMListView->iItem;

CPoint ptInScreen;

GetCursorPos(&ptInScreen); //R버튼 눌린좌표얻음

if(nListIndex==--1){
}
else{
CMenu muTemp, *pContextMenu;
muTemp.LoadMenuW( IDR_PopupMenu);
pContextMenu = muTemp.GetSubMenu(5);
pContextMenu -> TrackPopupMenu(TPM_LEFTALIGN|TPM_RIGHTBUTTON, ptInScreen.x,
                                ptInScreen.y, this);
}
m_skey = _ttoi(m_ex_list.GetItemText(nListIndex,0));
*pResult = 0;
}

//컨텍스트 메뉴 이벤트 처리기(삭제 메뉴)
void CExchangeSearchPage::OnExchangeDelete()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    int ret;
    ret = AfxMessageBox(_T("삭제하시겠습니까?"), MB_YESNO|MB_ICONSTOP);
    if(ret == IDYES){
        UpdateData(TRUE);

        CDatabase db; // 데이터베이스 객체 생성
        CRecordset rs (&db); // 레코드셋 객체 생성

        try
        {
            CString strConnection(L"DRIVER={Microsoft Access Driver
            (*.mdb)};DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

            if (db.OpenEx (strConnection) == FALSE)
                throw _T("데이터베이스 접속실패!");

            CString strQuery; //데이터베이스에 전송할 쿼리문
            strQuery.Format(_T("DELETE * FROM Sale WHERE Sale.s_Key= %d
            "),m_skey);
            db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송

        }
        catch (CException * e)
        {
            e->ReportError();
            e->Delete();
        }
        catch (LPCTSTR msg)
        {
            AfxMessageBox (msg);
        }

        rs.Close(); //레코드셋 객체 소멸
        db.Close(); //데이터베이스 객체 소멸
    }
}

```

```

if(m_ex_list.GetItemCount() > 0) m_ex_list.DeleteAllItems();

CDatabase db; // 데이터베이스 객체 생성
CRecordset rs (&db); // 레코드셋 객체 생성
try
{
    CString strConnection(L"DRIVER={Microsoft Access Driver
                          (*.mdb)};DBQ=C:\\WWWDownload\\WWWToyPocketDB.mdb;");

    if (db.OpenEx (strConnection) == FALSE)
        throw _T("데이터베이스 접속실패!");

    CString strQuery;
    strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate, out_ccode,
                       i_code, s_outputAmount, exchange_date, exchange_reason,
                       i_inputPrice, i_outputPrice, order_total FROM Sale WHERE
                           exchange_YN='예'"));

    if (rs.Open (CRecordset::snapshot, strQuery) == FALSE)
        throw _T("쿼리전송 실패!");
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}

//쿼리문을 이용해 검색한 내용 리스트 컨트롤에 출력
short col = rs.GetODBCFieldCount();
for(int p=0 ; p<2 ; p++)
{
    while( ! rs.IsEOF()) //행단위
    {
        //속성단위
        CString tmp;
        rs.GetFieldValue((short)p, tmp);
        //id값이 기준이 된다.
        m_ex_list.InsertItem(p,tmp,0);
        for(short i=1; i<col; i++ )
        {
            rs.GetFieldValue(i, tmp);
            // 첫번째 인자는 InsertValue를 통해서 가지고 온
            // 기준이 되는 정보, 두번째 인자는 열단위로 하나씩
            m_ex_list.SetItemText(p, i ,tmp);
        }
        rs.MoveNext(); //다음행으로 이동
    }
}
rs.Close(); //레코드셋 객체 소멸
db.Close(); //데이터베이스 객체 소멸
}

void CExchangeSearchPage::OnBnClickedExcelSave() //엑셀 저장
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    if (m_ex_list.GetItemCount() <= 0)
    {
        AfxMessageBox(_T("조회된 데이터가 0건이므로 엑셀 저장 할 수

```

```

        return;
    }

    CTime timeNow = CTime::GetCurrentTime();
    CString strDateTime;
    strDateTime = timeNow.Format(_T("%Y-%m-%d_%H시%M분%S초"));
    CString strFileName;
    strFileName.Format(_T("%s교환 조회_%s"), result, strDateTime);

    CFileDialog fdlg(FALSE, _T("xlsx"), strFileName+_T(".xlsx"), OFN_HIDEREADONLY
    | OFN_OVERWRITEPROMPT | OFN_NOCHANGEDIR, _T("xlsx 파일 (*.xlsx)|*.xlsx"), NULL);

    if (fdlg.DoModal() != IDOK) return;

    CXLEzAutomation xls(FALSE); // FALSE : 처리 과정을 화면에 보이지 않는다.

    int rowOffset = 2;
    int colOffset = 1;

    int rowCount = m_ex_list.GetItemCount();
    CHeaderCtrl* pHeaderCtrl = m_ex_list.GetHeaderCtrl();
    int colCount = pHeaderCtrl->GetItemCount();
    int rowIdx;
    int colIdx;

    CString strItemText;
    for (colIdx = 0; colIdx < colCount; colIdx++){
        HDITEM hdi;
        hdi.mask = HDI_TEXT;
        hdi.pszText = strItemText.GetBuffer( 256 );
        hdi.cchTextMax = 256;
        pHeaderCtrl->GetItem( colIdx, &hdi );
        strItemText.ReleaseBuffer();
        xls.SetCellValue(colIdx + colOffset, 0 + rowOffset, strItemText);
    }

    for (rowIdx = 0; rowIdx < rowCount; rowIdx++)
    {
        for (colIdx = 0; colIdx < colCount; colIdx++)
        {
            strItemText = m_ex_list.GetItemText(rowIdx, colIdx);

            // 컬럼 데이터
            xls.SetCellValue(colIdx + colOffset, rowIdx + 1 + rowOffset,
                            strItemText);
        }
    }

    xls.SaveFileAs(fdlg.GetPathName());

    xls.ReleaseExcel();
}

```

## [반품관리 - 반품등록]

```

void CReturnPage::OnBnClickedOk()    //반품 등록
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

    if(!UpdateData()) return;
    UpdateData(TRUE);

    //판매날짜

    CString sdate;
    sdate.Format(_T("%d-%02d-%02d"),m_sdate.GetYear(),m_sdate.GetMonth(),m_sdate.GetDay());

    //반품날짜

    CString rdate;
    rdate.Format(_T("%d-%02d-%02d"),m_reDate.GetYear(),m_reDate.GetMonth(),m_reDate.GetDay());

    int nIndex;
    nIndex = m_reReason_Combo.GetCurSel();
    reason = SelectReason(nIndex);

    itemAmount.Format(_T("%d"), _ttoi(itemAmount) + _ttoi(m_reAmount));

    CDatabase db;                                // 데이터베이스 객체 생성
    CRecordset rs (&db);    // 레코드셋 객체 생성

    if(_ttoi(m_reAmount) == _ttoi(m_sAmount)){
        try
        {
            CString strConnection(L"DRIVER={Microsoft Access
            Driver (*.mdb)};DBQ=C:\\WWWDownload\\WWWToyPocketDB.mdb;");

            if (db.OpenEx (strConnection) == FALSE)
                throw _T("데이터베이스 접속실패!");

            CString strQuery; //데이터베이스에 전송할 쿼리문
            strQuery.Format(_T("UPDATE Sale SET return_YN='예',
            return_date='%s', return_reson='%s' WHERE
            i_code='%s'AND s_outputDate='%s'"),
            rdate ,reason ,m_sicode ,sdate);
            db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송
            strQuery.Format(_T("UPDATE Item SET
            i_inputAmount='%s' WHERE i_code='%s'"), m_reAmount,
            m_sicode);
            db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송
        }
        catch (CException * e)
        {
            e->ReportError();
            e->Delete();
        }
        catch (LPCTSTR msg)
        {
            AfxMessageBox (msg);
        }
        MessageBox(_T("성공적으로 반품되었습니다.));
    }
    else if(_ttoi(m_reAmount) < _ttoi(m_sAmount)){
        CString AA;
    }
}

```

```

AA.Format(_T("%d"),_ttoi(m_sAmount)-_ttoi(m_reAmount));

CString total;
total.Format(_T("%d"),_ttoi(outputprice)*_ttoi(m_reAmount));

try
{
    CString strConnection(L"DRIVER={Microsoft Access
Driver (*.mdb)};DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

    if (db.OpenEx (strConnection) == FALSE)
        throw _T("데이터베이스 접속실패!");

    CString strQuery; //데이터베이스에 전송할 쿼리문

    strQuery.Format(_T("UPDATE Sale SET
s_outputAmount='%s' WHERE i_code='%s' AND
return_YN='아니오' AND s_outputDate='%s'"),AA
,m_sicode ,sdate);
    db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송

    strQuery.Format(_T("INSERT INTO Sale (s_outputDate,
in_ccode, out_ccode, i_code, s_outputAmount,
i_inputPrice, i_outputPrice, order_total, return_YN,
return_date, return_reson, exchange_YN,
exchange_date, exchange_reason,delivery_YN,
delivery_process, sale_id) VALUES ('%s', '%s', '%s',
'%s', '%s', '%s', '%s', '%s', '예', '%s', '%s',
'아니오', '해당없음', '해당없음', '아니오',
'해당없음', '%s')"), sdate,m_incode, m_sscore,
m_sicode,m_reAmount ,inputprice,outputprice,total
,rdate, reason, g_setid);

    db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송

    strQuery.Format(_T("UPDATE Item SET
i_inputAmount='%s' WHERE i_code='%s'"), itemAmount,
m_sicode);
    db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}

    MessageBox(_T("성공적으로 반품되었습니다.));
}
else {
    MessageBox(_T("수량을 다시 확인해주세요.));
}
rs.Close(); //레코드셋 객체 소멸
db.Close(); //데이터베이스 객체 소멸

m_reCombo.SetCurSel(0);
m_reReason_Combo.SetCurSel(-1);
GetDlgItem(IDC_STATIC_INPUTDATE)->ShowWindow(SW_HIDE);
GetDlgItem(IDC_EDIT10)->ShowWindow(SW_HIDE);

```

```

        m_reSearch = _T("");
        m_sccode_Edit.SetWindowText(_T(""));
        m_sicode_Edit.SetWindowText(_T(""));
        m_sAmount_Edit.SetWindowText(_T(""));
        m_reAmount_Edit.SetWindowText(_T(""));
        CDateTimeCtrl *pCtrl = (CDateTimeCtrl *)
            GetDlgItem(IDC_DATETIMEPICKER4);
        CTime tm = CTime::GetCurrentTime();
        pCtrl->SetTime(&tm);
        CDateTimeCtrl *pCtrl2 = (CDateTimeCtrl *)
            GetDlgItem(IDC_DATETIMEPICKER1);
        CTime tm2 = CTime::GetCurrentTime();
        pCtrl2->SetTime(&tm2);

        //이미지초기화
        CString img;
        img.Format(_T("res\\NoImage.bmp"));
        DrawBMP(img);
    }
    //판매 정보 리스트 더블클릭시 밑에 있는 에디트 박스에 값이 setting
    void CReturnPage::OnNMdblclkList1(NMHDR *pNMHDR, LRESULT *pResult)
    {
        LPNMITEMACTIVATE pNMItemActivate =
            reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);
        // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
        NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
        nListIndex = pNMListView->iItem;

        m_sccode_Edit.SetWindowText(m_reList.GetItemText(nListIndex,3));
        m_sicode_Edit.SetWindowText(m_reList.GetItemText(nListIndex,4));
        m_sAmount_Edit.SetWindowText(m_reList.GetItemText(nListIndex,5));

        AfxExtractSubString(g_set_ibirth_Year,m_reList.GetItemText(nListIndex,1),
            0,'-');
        AfxExtractSubString(g_set_ibirth_Month,m_reList.GetItemText(nListIndex,1),
            1,'-');
        AfxExtractSubString(g_set_ibirth_Day,m_reList.GetItemText(nListIndex,1),
            2,'-');

        COleDateTime Date; //사용자가 지정한 날짜를 저장할 COleDateTimeDate 변수
        CString wsYear = g_set_ibirth_Year; //년
        CString wsMonth = g_set_ibirth_Month; //월
        CString wsDay = g_set_ibirth_Day; //일
        Date.SetDate(_tstoi(wsYear),_tstoi(wsMonth),_tstoi(wsDay));
        //COleDateTime타입은 UpdateData해줘야함 그럼 나머지만먹힘
        //그래서 변수 지우고 다시 DateTime형식의 컨트롤로 다시만듬
        m_sDate_ctrl.SetTime(Date);

        m_icode = m_reList.GetItemText(nListIndex,4);
        m_incode = m_reList.GetItemText(nListIndex,2);
        inputprice = m_reList.GetItemText(nListIndex,6);
        outputprice = m_reList.GetItemText(nListIndex,7);
        imgname.Format(_T("res\\Item\\%s.bmp"),m_icode);
        DrawBMP(imgname);

        *pResult = 0;
    }
    CString CReturnPage::SelectReason(int index) //대분류 문자열 반환
    {
        switch(index)
        {
            case 0 :

```

```

        reason = _T("단순변심");
        break;
    case 1 :
        reason = _T("불량");
        break;
    case 2 :
        etcresaon = m_reReason.GetString();
        reason = etcresaon;
        break;
    }
    return reason;
}
void CReturnPage::DrawBMP(LPCTSTR lpszFileName) //이미지 보여주기
{
    CDC* pDC = GetDC(); //화면 DC를 얻어 옴
    CDC MemDC; //메모리 DC
    CBitmap bmp, *pOldBmp;
    BITMAP bm; //비트맵 정보를 저장하는 변수
    CRect rect;
    CPoint cp;

    //이미지 파일 로딩
    bmp.m_hObject = (HBITMAP)LoadImage(NULL, lpszFileName, IMAGE_BITMAP, 0, 0,
    LR_LOADFROMFILE | LR_CREATEDIBSECTION | LR_DEFAULTSIZE);
    if(bmp.m_hObject == NULL) return;
    bmp.GetBitmap(&bm); //로딩된 이미지 파일의 형태를 얻음
    MemDC.CreateCompatibleDC(pDC); //메모리 DC와 화면 DC를 호환성있도록 연결
    pOldBmp = MemDC.SelectObject(&bmp); //메모리에 적재된 이미지 선택

    m_riimage.GetWindowRect(&rect); //다이얼로그에 배치된 컨트롤의 크기를 얻어 옴
    cp.x = rect.left; //이미지 출력을 위한 시작 x 좌표
    cp.y = rect.top; //이미지 출력을 위한 시작 y 좌표
    ScreenToClient(&cp); //cp가 가지고 있는 좌표로 화면을 지정함
    pDC->StretchBlt(cp.x, cp.y, rect.Width(), rect.Height(), &MemDC, 0, 0,
    bm.bmWidth, bm.bmHeight, SRCCOPY); //이미지를 해당 컨트롤로 전송
    MemDC.SelectObject(pOldBmp);
    ReleaseDC(pDC); //DC 반환
}
void CReturnPage::OnBnClickedSearchbtn() //판매 정보 리스트 조회
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(true);

    if(m_reList.GetItemCount() > 0) m_reList.DeleteAllItems();

    CDatabase db; // 데이터베이스 객체 생성
    CRecordset rs(&db); // 레코드셋 객체 생성
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access Driver
        (*.mdb)};DBQ=C:\\WW\\Download\\WWToyPocketDB.mdb;");

        if (db.OpenEx (strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery;

        int nIndex = m_reCombo.GetCurSel();
        if(nIndex == 0){
            strQuery.Format(_T("SELECT *, lme.i_inputAmount FROM Sale"));
            strQuery.Format(_T("SELECT Sale.sale_id, Sale.s_outputDate,
            Sale.in_ccode, Sale.out_ccode, Sale.i_code ,

```



```

Sale.s_outputAmount, Sale.i_inputPrice, Sale.i_outputPrice,
Sale.order_total, Item.i_inputAmount FROM Item INNER JOIN
Sale ON Item.i_code = Sale.i_code WHERE Sale.s_outputDate
LIKE '%%s%%' OR Sale.in_ccode LIKE '%%s%%' OR
Sale.out_ccode LIKE '%%s%%' OR Sale.i_code LIKE '%%s%%' OR
Sale.s_outputAmount LIKE '%%s%%' OR Sale.i_inputPrice LIKE
'%%s%%' OR Sale.i_outputPrice LIKE '%%s%%' OR
Sale.order_total LIKE '%%s%%' OR Sale.return_date LIKE
'%%s%%' OR Sale.return_reson LIKE '%%s%%' OR Sale.sale_id
LIKE '%%s%%' AND Sale.return_YN = '아니오' AND
Sale.exchange_YN = '아니오'),m_reSearch,m_reSearch,m_reSearch
,m_reSearch,m_reSearch,m_reSearch,m_reSearch,m_reSearch);
}
if(nIndex == 1){
    strQuery.Format(_T("SELECT *,Item.i_inputAmount FROM Sale"));
    strQuery.Format(_T("SELECT Sale.sale_id, Sale.s_outputDate,
Sale.in_ccode, Sale.out_ccode, Sale.i_code ,
Sale.s_outputAmount, Sale.i_inputPrice, Sale.i_outputPrice,
Sale.order_total, Item.i_inputAmount FROM Item INNER JOIN
Sale ON Item.i_code = Sale.i_code WHERE Sale.s_outputDate
LIKE '%%s%%' AND Sale.return_YN = '아니오' AND
Sale.exchange_YN = '아니오'),m_reSearch);
}
if(nIndex == 2){
    strQuery.Format(_T("SELECT *,Item.i_inputAmount FROM Sale"));
    strQuery.Format(_T("SELECT Sale.sale_id, Sale.s_outputDate,
Sale.in_ccode, Sale.out_ccode, Sale.i_code ,
Sale.s_outputAmount, Sale.i_inputPrice, Sale.i_outputPrice,
Sale.order_total, Item.i_inputAmount FROM Item INNER JOIN
Sale ON Item.i_code = Sale.i_code WHERE Sale.out_ccode LIKE
'%%s%%' AND Sale.return_YN = '아니오' AND Sale.exchange_YN =
'아니오'),m_reSearch);
}
if(nIndex == 3){
    strQuery.Format(_T("SELECT *,Item.i_inputAmount FROM Sale"));
    strQuery.Format(_T("SELECT Sale.sale_id, Sale.s_outputDate,
Sale.in_ccode, Sale.out_ccode, Sale.i_code ,
Sale.s_outputAmount, Sale.i_inputPrice, Sale.i_outputPrice,
Sale.order_total, Item.i_inputAmount FROM Item INNER JOIN
Sale ON Item.i_code = Sale.i_code WHERE Sale.i_code LIKE
'%%s%%' AND Sale.return_YN = '아니오' AND Sale.exchange_YN =
'아니오'),m_reSearch);
}
if (rs.Open (CRecordset::snapshot, strQuery) == FALSE)
    throw _T("쿼리전송 실패!");
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}

short col = rs.GetODBCFieldCount(); //쿼리문을 이용해 검색한 내용 리스트 컨트롤에 출력
for(int p=0 ; p<2 ; p++)
{
    while( ! rs.IsEOF())
    {

```

```

        CString tmp;
        rs.GetFieldValue((short)p, tmp);
        m_reList.InsertItem(p,tmp,0);
        for(short i=1; i<col; i++ )
        {
            rs.GetFieldValue(i, tmp);
            m_reList.SetItemText(p, i ,tmp);
        }
        rs.MoveNext();
    }
}
rs.Close();
db.Close();
}

```

### [반품관리 - 반품조회]

```

void CReturnSearchPage::OnBnClickedOk()           // 반품 정보 조회
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

    UpdateData( true);

    if(m_re_list.GetItemCount() > 0) m_re_list.DeleteAllItems();

    CDatabase db;                                // 데이터베이스 객체 생성
    CRecordset rs (&db);                        // 레코드셋 객체 생성
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access Driver (*.mdb)};
                                DBQ=C:\\WWWDownload\\WWWToyPocketDB.mdb;");

        if (db.OpenEx (strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery;

        int nIndex = m_res_Combo.GetCurSel();
        if(nIndex == 0){
            strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate, in_ccode,
                                out_ccode, i_code, s_outputAmount, return_date, return_reson,
                                i_inputPrice, i_outputPrice, order_total FROM Sale W WHERE
                                s_outputDate LIKE '%%s%%' OR in_ccode LIKE '%%s%%' OR out_ccode
                                LIKE '%%s%%' OR i_code LIKE '%%s%%' OR s_outputAmount LIKE
                                '%%s%%' OR i_inputPrice LIKE '%%s%%' W OR i_outputPrice LIKE
                                '%%s%%' OR order_total LIKE '%%s%%' OR return_date LIKE '%%s%%'
                                OR return_reson LIKE '%%s%%' W OR sale_id LIKE '%%s%%' AND
                                Sale.return_YN = '예'"), m_res_Search,m_res_Search,m_res_Search,
                                m_res_Search,m_res_Search, m_res_Search, m_res_Search,
                                m_res_Search, m_res_Search, m_res_Search, m_res_Search);
            result.Format(_T("전체검색-( '%s' )검색시_"),m_res_Search);
        }
        if(nIndex == 1){
            strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate,in_ccode,
                                out_ccode, i_code, s_outputAmount, return_date, return_reson,
                                i_inputPrice, i_outputPrice, order_total FROM Sale WHERE
                                return_date LIKE '%%s%%' AND Sale.return_YN = '예'"),m_res_Search);
            result.Format(_T("반품날짜-( '%s' )검색시_"),m_res_Search);
        }
        if(nIndex == 2){

```

```

        strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate, in_ccode,
out_ccode, i_code, s_outputAmount, return_date, return_reson,
i_inputPrice, i_outputPrice, order_total FROM Sale WHERE i_code
LIKE '%%s%%'AND Sale.return_YN = '예'"),m_res_Search);
        result.Format(_T("상품코드-('%s')검색시_"),m_res_Search);
    }
    if(nIndex == 3){
        strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate, in_ccode,
out_ccode, i_code, s_outputAmount, return_date, return_reson,
i_inputPrice, i_outputPrice, order_total FROM Sale WHERE out_ccode
LIKE '%%s%%'AND Sale.return_YN = '예'"),m_res_Search);
        result.Format(_T("납품처코드-('%s')검색시_"),m_res_Search);
    }
    if (rs.Open (CRecordset::snapshot, strQuery) == FALSE)
        throw _T("쿼리전송 실패!");
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}

short col = rs.GetODBCFieldCount(); //쿼리문을 이용해 검색한 내용 리스트 컨트롤에 출력
for(int p=0 ; p<2 ; p++)
{
    while( ! rs.IsEOF())
    {
        CString tmp;
        rs.GetFieldValue((short)p, tmp);
        m_re_list.InsertItem(p,tmp,0);
        for(short i=1; i<col; i++ )
        {
            rs.GetFieldValue(i, tmp);
            m_re_list.SetItemText(p, i ,tmp);
        }
        rs.MoveNext();
    }
}
rs.Close();
db.Close();
}

void CReturnSearchPage::OnNMRClickList1(NMHDR *pNMHDR, LRESULT *pResult)
//반품 정보 삭제 컨텍스트 메뉴 띄움
{
    LPNMITEMACTIVATE pNMItemActivate = reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    nIndex = pNMListView->iItem;

    CPoint ptInScreen;

    GetCursorPos(&ptInScreen); //R버튼 눌린 좌표 얻음

    if(nListIndex==--1){
    }
    else{
        CMenu muTemp, *pContextMenu;

```

```

muTemp.LoadMenuW( IDR_PopupMenu);
pContextMenu = muTemp.GetSubMenu(4);
pContextMenu -> TrackPopupMenu(TPM_LEFTALIGN|TPM_RIGHTBUTTON, ptInScreen.x, ptInScreen.y, this);
}

m_skey = _ttoi(m_re_list.GetItemText(nListIndex,0));
*pResult = 0;
}

void CReturnSearchPage::OnReturnDelete() //삭제 메뉴 명령 처리기
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    int ret;
    ret = AfxMessageBox(_T("삭제하시겠습니까?"), MB_YESNO|MB_ICONSTOP);
    if(ret == IDYES){
        UpdateData(TRUE);

        CDatabase db;                                // 데이터베이스 객체 생성
        CRecordset rs (&db);    // 레코드셋 객체 생성

        try
        {
            CString strConnection(L"DRIVER={Microsoft Access Driver (*.mdb)};
                                   DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

            if (db.OpenEx (strConnection) == FALSE)
                throw _T("데이터베이스 접속실패!");

            CString strQuery;    //데이터베이스에 전송할 쿼리문
            strQuery.Format(_T("DELETE * FROM Sale WHERE Sale.s_Key= %d "),m_skey);
            db.ExecuteSQL(strQuery);    //데이터베이스에 쿼리문 전송

        }
        catch (CException * e)
        {
            e->ReportError();
            e->Delete();
        }
        catch (LPCTSTR msg)
        {
            AfxMessageBox (msg);
        }

        rs.Close();    //레코드셋 객체 소멸
        db.Close();    //데이터베이스 객체 소멸
    }

    if(m_re_list.GetItemCount() > 0) m_re_list.DeleteAllItems();

    CDatabase db;                                // 데이터베이스 객체 생성
    CRecordset rs (&db);    // 레코드셋 객체 생성
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access Driver
                                (*.mdb)};DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;");

        if (db.OpenEx (strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery;    //데이터베이스에 전송할 쿼리문
        strQuery.Format(_T("SELECT s_Key, sale_id, s_outputDate, in_ccode,
                            out_ccode, i_code, s_outputAmount, return_date, return_reson, i_inputPrice,

```

```

        i_outputPrice, order_total FROM Sale WHERE return_YN= '예'"));
        if (rs.Open (CRecordset::snapshot, strQuery) == FALSE)
            throw _T("쿼리전송 실패!");
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox (msg);
    }

    //쿼리문을 이용해 검색한 내용 리스트 컨트롤에 출력
    short col = rs.GetODBCFieldCount();
    for(int p=0 ; p<2 ; p++)
    {
        while( ! rs.IsEOF())                //행단위
        {
            //속성단위
            CString tmp;
            rs.GetFieldValue((short)p, tmp);
            //id값이 기준이 된다.
            m_re_list.InsertItem(p,tmp,0);
            for(short i=1; i<col; i++ )
            {
                rs.GetFieldValue(i, tmp);
                // 첫 번째 인자는 InsertValue를 통해서 가지고 온 기준이 되는 정보,
                // 두 번째 인자는 열 단위로 하나씩
                m_re_list.SetItemText(p, i ,tmp);
            }
            rs.MoveNext(); //다음행으로 이동
        }
    }
    rs.Close();        //레코드셋 객체 소멸
    db.Close();        //데이터베이스 객체 소멸
}

void CReturnSearchPage::OnBnClickedExcelSave() //엑셀 저장
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    if (m_re_list.GetItemCount() <= 0)
    {
        AfxMessageBox(_T("조회된 데이터가 0건이므로 엑셀 저장 할 수 없습니다.));
        return;
    }

    CTime timeNow = CTime::GetCurrentTime();
    CString strDateTime;
    strDateTime = timeNow.Format(_T("%Y-%m-%d_%H시%M분%S초"));
    CString strFileName;
    strFileName.Format(_T("%s반품 조회_%s"), result, strDateTime);

    CFileDialog fdlg(FALSE, _T("xlsx"), strFileName+_T(".xlsx"), OFN_HIDEREADONLY |
    OFN_OVERWRITEPROMPT | OFN_NOCHANGEDIR,
        _T("xlsx 파일 (*.xlsx)|*.xlsx"), NULL);
    if (fdlg.DoModal() != IDOK) return;

    CXLEzAutomation xls(FALSE); // FALSE : 처리 과정을 화면에 보이지 않는다.

    int rowOffset = 2;

```

```

int colOffset = 1;

int rowCount = m_re_list.GetItemCount();
CHeaderCtrl* pHeaderCtrl = m_re_list.GetHeaderCtrl();
int colCount = pHeaderCtrl->GetItemCount();
int rowIdx;
int colIdx;

CString strItemText;
for (colIdx = 0; colIdx < colCount; colIdx++){
    HDITEM hdi;
    hdi.mask = HDI_TEXT;
    hdi.pszText = strItemText.GetBuffer( 256 );
    hdi.cchTextMax = 256;
    pHeaderCtrl->GetItem( colIdx, &hdi );
    strItemText.ReleaseBuffer();
    xls.SetCellValue(colIdx + colOffset, 0 + rowOffset, strItemText);
}

for (rowIdx = 0; rowIdx < rowCount; rowIdx++)
{
    for (colIdx = 0; colIdx < colCount; colIdx++)
    {
        strItemText = m_re_list.GetItemText(rowIdx, colIdx);

        // 컬럼 데이터
        xls.SetCellValue(colIdx + colOffset, rowIdx + 1 + rowOffset,
            strItemText);
    }
}

xls.SaveFileAs(fdlg.GetPathName());

xls.ReleaseExcel();
}

```

## [일정관리]

```

void CCalendarManage::OnPaint() //달력그리기
{
    //CPaintDC dc(this); // device context for painting
    // TODO: 여기에 메시지 처리기 코드를 추가합니다.

    int i, day_x, day_y, last_day, k;
    int centry, draw_x, draw_y;
    char daynum[5], y[10], m[10];
    int t_day[12]={31,28,31,30,31,30,31,31,30,31,30,31}; // 달별 날수

    CString day[7]={_T("일"), _T("월"),_T("화"),_T("수"),_T("목"),_T("금"),_T("토")};
    // 요일
    CPaintDC dc(this); // device context for painting

    CFont font7;
    font7.CreateFont(20,15,0,0, FW_NORMAL, FALSE, FALSE, 0,  DEFAULT_CHARSET,
        OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY, DEFAULT_PITCH | FF_SWISS,
        _T("나눔바른고딕"));
    CFont *oldFont = dc.SelectObject(&font7);

    dc.SetBkMode(TRANSPARENT);

    // TODO: Add your message handler code here

```

```

dc.Rectangle(x1,y1,x2,y2); // 달력 사각형 그리기
dc.Rectangle(x1,y1-20,x2,y1); //요일 표시사각형 그리기
for ( i=1; i<=(daycount-1); i++)
{
    dc.MoveTo(x1+gap*i,y1-20);
    dc.LineTo(x1+gap*i,y2);
}
for ( i=1; i<=(weekcount-1); i++)
{
    dc.MoveTo(x1,y1+gap*i);
    dc.LineTo(x2,y1+gap*i);
}
for ( i=1; i<=7; i++) { // 요일 표시하기
day_x = x1+gap*(i-1)+gap/2-7;
day_y = y1-18;
if ( i==1) dc.SetTextColor( RGB(255,0,0)); // 토요일이면 파란색, 일요일이면 빨간색
else if ( i==7) dc.SetTextColor( RGB(0,0,255));
else dc.SetTextColor( RGB(0,0,0));
dc.TextOut(day_x,day_y,day[i-1]);
}

CTime timer;
timer = CTime::GetCurrentTime();
if (DrawMode) {
    // 현재 년월일 구하기
year = timer.GetYear();
month = timer.GetMonth();
day_ = timer.GetDay();
}
_itoa(year,y,10);
_itoa(month,m,10);

UpdateData(TRUE); // 이걸해줘야 컨트롤(에디트박스)의 값이 바뀜
m_year = CString(y);
m_month = CString(m);
UpdateData(FALSE);

centry = (year/100+1); // 세기는 년도에 100을 나눠서 1을 빼주면 됨
one_whatday = (year-1)*365+(year-1)/4-(year-1)/100+(year-1)/400; // 년도일수
if ((year%4 == 0 && year%100 != 0) || (year%400==0))
//년도가 400으로 나누어떨어지거나 4의 배수이고 100으로 나누어떨어지지 않으면 윤년
t_day[1]=29;
else t_day[1]=28;

for(i=1; i<month; i++)
    one_whatday += t_day[i-1]; // 달일수

one_whatday += 1; // 그달의 1일의 요일
one_whatday = one_whatday % 7; // 1이면 월요일 7이면 일요일
one_whatday += 1;
draw_x = x1+gap*(one_whatday-1)+gap/2-7; // 1일 표시하기
draw_y = y1+gap/2;

if (one_whatday==1) dc.SetTextColor( RGB(255,0,0)); // 일요일이면 빨간색, 토요일이면 파란색
else if (one_whatday==7) dc.SetTextColor( RGB(0,0,255));
else dc.SetTextColor( RGB(0,0,0));

dc.TextOut(draw_x-33,draw_y-47,_T("1"));

if ((month==1) || (month==3) || (month==5) || (month==7) || (month==8) ||
(month==10) || (month==12)) // 몇 일까지 있는지 계산 1,3,5,7,8,10,12월은 31까지 있음.

```

```

        last_day = 31;
    else if ((month==4) || (month==6) || (month==9) || (month==11))
        last_day=30;

    if (month==2)
    if ((year%4 == 0 && year%100 != 0) || (year%400==0))
        //년도가 400으로 나누어떨어지거나 4의 배수이고 100으로 나누어떨어지지 않으면 윤년
        last_day=29; // last_day 변수는 그달의 일수를 구하는 것입니다. 예를 들면 1월은 31일
    else last_day=28; // 그렇지 않으면 평년 // 윤년계산해서 2월 달에 대입하기

    k = one_whatday+1;
    if(k>7) k=1;
    for(i=2; i<=last_day; i++) { // 날짜 표시하기
        draw_x = draw_x+gap; // 날짜 표시 x좌표구하기
        if (draw_x>x2) {
            draw_x=x1+gap/2-7;
            draw_y=draw_y+gap; // 만약에 칸을 넘어가면 다음줄로 이동
        }
        _itoa(i, daynum, 10);
        if (k==1) dc.SetTextColor(RGB(255,0,0));
        else if (k==7) dc.SetTextColor(RGB(0,0,255));
        else dc.SetTextColor(RGB(0,0,0));

        dc.TextOut(draw_x-33, draw_y-47, CString(daynum));
        dc.SetTextColor(RGB(0,0,0));
        k++;
        if (k>7) k=1;
    }

    dc.SelectObject(oldFont);
    font7.DeleteObject();

    UpdateData(TRUE);

    CCalendarSet cal;
    cal.Open();

    CFont font14;

    font14.CreateFont(18,13,0,0, FW_NORMAL, FALSE, FALSE, 0, DEFAULT_CHARSET,
    OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY, DEFAULT_PITCH | FF_SWISS,
    _T(" 나눔바른고딕"));
    CFont *oldFont1 = dc.SelectObject(&font14);

    while(!cal.IsEOF()){

        int preweek = cal.m_week;
        int preday = cal.m_day;
        int preyear = cal.m_preyear;
        int premonth = cal.m_premonth;
        BOOL ch = cal.m_check;
        BOOL ch2 = cal.m_check2;
        BOOL ch3 = cal.m_check3;
        BOOL ch4 = cal.m_check4;

        dc.SetTextColor(RGB(0,0,0));
        dc.SetBkMode(TRANSPARENT);

        //디비에 저장되어있는 값 나타내기
        if((preyear == _ttoi(m_year)) && (premonth == _ttoi(m_month))){

```



```

if(cal.m_memo != ""){
if(cal.m_memo.GetLength() <= 4){
if(ch == TRUE){
dc.SetTextColor( RGB(29,219,22));
dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday].top+25, _T("●"));
dc.SetTextColor( RGB(0,51,153));
dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+25
,cal.m_memo);
}
else{
dc.SetTextColor( RGB(140,140,140));
dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday].top+25, _T("●"));
dc.SetTextColor( RGB(0,51,153));
dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+25
,cal.m_memo);
}
}
else{
CString lmemo;
lmemo= cal.m_memo.Left(4);
if(ch == TRUE){
dc.SetTextColor( RGB(29,219,22));
dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday].top+25
,T("●"));

dc.SetTextColor( RGB(0,51,153));
dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+25
, lmemo+_T("..."));
}
else{
dc.SetTextColor( RGB(140,140,140));
dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday]
.top+25, _T("●"));

dc.SetTextColor( RGB(0,51,153));
dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+25
, lmemo+_T("..."));
}
}
}
if(cal.m_memo2 != ""){
if(cal.m_memo2.GetLength() <= 4){
if(ch2 == TRUE){
dc.SetTextColor( RGB(29,219,22));
dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday].top+45
,T("●"));

dc.SetTextColor( RGB(0,51,153));
dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+45
,cal.m_memo2);
}
else{
dc.SetTextColor( RGB(140,140,140));
dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday].top+45
,T("●"));

dc.SetTextColor( RGB(0,51,153));
dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+45
,cal.m_memo2);
}
}
}
else{
CString lmemo;
lmemo= cal.m_memo2.Left(4);
if(ch2 == TRUE){
dc.SetTextColor( RGB(29,219,22));
dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday]
.top+45, _T("●"));

dc.SetTextColor( RGB(0,51,153));
dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+45
, lmemo+_T("..."));
}
}
}

```

```

}else{
    dc.SetTextColor( RGB( 140,140,140) );
    dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday].top+45
                                                         ,_T( "●" ));

    dc.SetTextColor( RGB( 0,51,153) );
    dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+45
                                                         , lmemo+_T( "..." ));
}
}
}
if( cal.m_memo3 != "" ){
if( cal.m_memo3.GetLength() <= 4 ){
if( ch3 == TRUE ){
    dc.SetTextColor( RGB( 29,219,22) );
    dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday].top+65
                                                         ,_T( "●" ));

    dc.SetTextColor( RGB( 0,51,153) );
    dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+65
                                                         , cal.m_memo3 );
}else{
    dc.SetTextColor( RGB( 140,140,140) );
    dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday].top+65
                                                         ,_T( "●" ));

    dc.SetTextColor( RGB( 0,51,153) );
    dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+65
                                                         , cal.m_memo3 );
}
}else{
CString lmemo;
lmemo= cal.m_memo3.Left(4);
if( ch3 == TRUE ){
    dc.SetTextColor( RGB( 29,219,22) );
    dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday]
                                                         .top+65,_T( "●" ));

    dc.SetTextColor( RGB( 0,51,153) );
    dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+65
                                                         , lmemo+_T( "..." ));
}else{
    dc.SetTextColor( RGB( 140,140,140) );
    dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday]
                                                         .top+65,_T( "●" ));

    dc.SetTextColor( RGB( 0,51,153) );
    dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+65
                                                         , lmemo+_T( "..." ));
}
}
}
if( cal.m_memo4 != "" ){
if( cal.m_memo4.GetLength() <= 4 ){
if( ch4 == TRUE ){
    dc.SetTextColor( RGB( 29,219,22) );
    dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday]
                                                         .top+85,_T( "●" ));

    dc.SetTextColor( RGB( 0,51,153) );
    dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+85
                                                         , cal.m_memo4 );
}else{
    dc.SetTextColor( RGB( 140,140,140) );
    dc.TextOut( rect[preweek][preday].left+5, rect[preweek][preday]
                                                         .top+85,_T( "●" ));

    dc.SetTextColor( RGB( 0,51,153) );
    dc.TextOut( rect[preweek][preday].left+20, rect[preweek][preday].top+85

```

```

,cal.m_memo4);
}
}else{
    CString lmemo;
    lmemo= cal.m_memo4.Left(4);
    if(ch4 == TRUE){
        dc.SetTextColor( RGB( 29,219,22));
        dc.TextOut( rect[preweek][preday].left+5,rect[preweek][preday]
                    .top+85,_T( "●"));

        dc.SetTextColor( RGB( 0,51,153));
        dc.TextOut( rect[preweek][preday].left+20
                    ,rect[preweek][preday].top+85,lmemo+_T( "..."));
    }else{
        dc.SetTextColor( RGB( 140,140,140));
        dc.TextOut( rect[preweek][preday].left+5,rect[preweek][preday].top+85
                    ,_T( "●"));

        dc.SetTextColor( RGB( 0,51,153));
        dc.TextOut( rect[preweek][preday].left+20,rect[preweek][preday].top+85
                    ,lmemo+_T( "..."));
    }
}
}
}
}
cal.MoveNext();
}
cal.MoveLast();

dc.SelectObject(oldFont1);
font14.DeleteObject();

// 그리기 메시지에 대해서는 CFormView::OnPaint()을(를) 호출하지 마십시오.
}
void CCalendarManage::OnBnClickedPrev() //이전달로 이동
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    month=month-1;
    day_=1;
    DrawMode = FALSE;
    if (month<1) {
        year = year-1;
        month=12;
        day_=1;
    }
    Invalidate();
}

void CCalendarManage::OnBnClickedNext() //다음달로 이동
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    month=month+1; // 달은 한달 증가
    day_=1;
    DrawMode = FALSE;
    if (month>12) {
        year = year+1;
        month=1;
        day_=1;
    }
    Invalidate();
}

#include "CalendarMemoDlg.h"
//CalendarMemo다이얼로그 나타내기

```

```

void CCalendarManage::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: 여기에 메시지 처리기 코드를 추가 및/또는 기본값을 호출합니다.

    CClientDC dc(this);

    CFont font16;
    font16.CreateFont(18,13,0,0, FW_NORMAL, FALSE, FALSE, 0,  DEFAULT_CHARSET,
    OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY, DEFAULT_PITCH
    | FF_SWISS, _T("나눔바른고딕"));
    CFont *oldFont1 = dc.SelectObject(&font16);

    int i, x, y, j,k,r;

    for (i=0;i<weekcount;i++)
    {
        for (j=0;j<daycount;j++)
        {
            boolpoint = rect[i][j].PtInRect(point);
            if(boolpoint == TRUE){
                week = i;
                day = j;
                daypre = (7- one_whatday+1) + (i-1)*7 + (j+1);

                yearpre = m_year;
                monthpre = m_month;
                CCalenderMemoDlg dlg;
                dlg.DoModal();
                int pday;
                CString memo,memo2,memo3,memo4;
                BOOL check,check2,check3,check4;
                pday = _ttoi(dlg.m_day);
                memo = dlg.m_memo;
                memo2 = dlg.m_memo2;
                memo3 = dlg.m_memo3;
                memo4 = dlg.m_memo4;
                check = dlg.m_check1;
                check2 = dlg.m_check2;
                check3 = dlg.m_check3;
                check4 = dlg.m_check4;

                int year = _ttoi(m_year);
                int month = _ttoi(m_month);

                CDatabase db;                // 데이터베이스 객체 생성
                CRecordset rs (&db);        // 레코드셋 객체 생성

                if(memo!=" " || memo2!=" " ||memo3!=" " || memo4!=" "){

                    try
                    {
                        CString strConnection(L"DRIVER={Microsoft Access Driver
                        (*.mdb)};DBQ=C:\\\\Download\\\\ToyPocketDB.mdb;"); //DB오픈

                        if (db.OpenEx (strConnection) == FALSE)
                            throw _T("데이터베이스 접속실패!");
                        CString strQuery; //데이터베이스에 전송할 쿼리문
                        strQuery.Format(_T("INSERT INTO Calendar VALUES
                        ('%s','%s','%s','%s', %d , %d, %d, %d,%d, %d , %d, %d,
                        %d)"),memo,memo2,memo3,memo4,year ,month,pday,week,day
                        ,check,check2,check3,check4);
                    }
                }
            }
        }
    }
}

```

```

        db.ExecuteSQL(strQuery);           //데이터베이스에 쿼리문 전송
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox (msg);
    }
    MessageBox(_T("성공적으로 추가되었습니다. "));

    dc.SetTextColor( RGB(155,0,155));

if(memo != ""){
    if(check == TRUE){
        if(memo.GetLength() <= 4){
            dc.SetTextColor( RGB(29,219,22));
            dc.TextOut(rect[i][j].left+5,rect[i][j].top+25,_T("●"));
            dc.SetTextColor( RGB(0,51,153));
            dc.TextOut(rect[i][j].left+20,rect[i][j].top+25,memo);
        }else{
            CString lmemo;
            lmemo= memo.Left(4);
            dc.SetTextColor( RGB(29,219,22));
            dc.TextOut(rect[i][j].left+5,rect[i][j].top+25,_T("●"));
            dc.SetTextColor( RGB(0,51,153));
            dc.TextOut(rect[i][j].left+20,rect[i][j].top+25
                ,lmemo+_T("..."));
        }
    }else{
        if(memo.GetLength() <= 4){
            dc.SetTextColor( RGB(140,140,140));
            dc.TextOut(rect[i][j].left+5,rect[i][j].top+25,_T("●"));
            dc.SetTextColor( RGB(0,51,153));
            dc.TextOut(rect[i][j].left+20,rect[i][j].top+25,memo);
        }else{
            CString lmemo;
            lmemo= memo.Left(4);
            dc.SetTextColor( RGB(140,140,140));
            dc.TextOut(rect[i][j].left+5,rect[i][j].top+25,_T("●"));
            dc.SetTextColor( RGB(0,51,153));
            dc.TextOut(rect[i][j].left+20,rect[i][j].top+25
                ,lmemo+_T("..."));
        }
    }
}

if(memo2 != ""){
    if(check2 == TRUE){
        if(memo2.GetLength() <= 4){
            dc.SetTextColor( RGB(29,219,22));
            dc.TextOut(rect[i][j].left+5,rect[i][j].top+45,_T("●"));
            dc.SetTextColor( RGB(0,51,153));
            dc.TextOut(rect[i][j].left+20,rect[i][j].top+45,memo2);
        }else{
            CString lmemo;
            lmemo= memo2.Left(4);
            dc.SetTextColor( RGB(29,219,22));
            dc.TextOut(rect[i][j].left+5,rect[i][j].top+45,_T("●"));
            dc.SetTextColor( RGB(0,51,153));

```



```

        dc.TextOut(rect[i][j].left+5,rect[i][j].top+85,_T("●"));
        dc.SetTextColor(RGB(0,51,153));
        dc.TextOut(rect[i][j].left+20,rect[i][j].top+85,lmemo+_T("..."));
    }
}
else{
    if(memo4.GetLength() <= 4){
        dc.SetTextColor(RGB(140,140,140));
        dc.TextOut(rect[i][j].left+5,rect[i][j].top+85,_T("●"));
        dc.SetTextColor(RGB(0,51,153));
        dc.TextOut(rect[i][j].left+20,rect[i][j].top+85,memo4);
    }
    else{
        CString lmemo;
        lmemo= memo4.Left(4);
        dc.SetTextColor(RGB(140,140,140));
        dc.TextOut(rect[i][j].left+5,rect[i][j].top+85,_T("●"));
        dc.SetTextColor(RGB(0,51,153));
        dc.TextOut(rect[i][j].left+20,rect[i][j].top+85
            ,lmemo+_T("..."));
    }
}
}
}
break;
}
}
if(boolpoint == TRUE)
    break;
}

dc.SelectObject(oldFont1);
font16.DeleteObject();

CFormView::OnLButtonDown(nFlags, point);
}

//컨텍스트메뉴 나타내기
void CCalendarManage::OnRButtonDown(UINT nFlags, CPoint point)
{
    // TODO: 여기에 메시지 처리기 코드를 추가 및/또는 기본값을 호출합니다.

    CPoint ptInScreen;

    GetCursorPos(&ptInScreen); //R버튼 눌린좌표얻음

    CMenu muTemp, *pContextMenu;
    muTemp.LoadMenuW( IDR_CALENDER );
    pContextMenu = muTemp.GetSubMenu(0);
    pContextMenu -> TrackPopupMenu(TPM_LEFTALIGN|TPM_RIGHTBUTTON, ptInScreen.x,
        ptInScreen.y, this);
    CFormView::OnRButtonDown(nFlags, point);

    CFormView::OnRButtonDown(nFlags, point);
}

//CalTimeGo 다이얼로그 나타내기
void CCalendarManage::OnDaygo()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    yearpre = m_year;
    monthpre = m_month;

    CCalTimeGo dlg;

```

```

dlg.DoModal();

CString ye = dlg.m_goyear;
CString mo = dlg.m_gomonth;

int m;
UpdateData(TRUE);
m = _ttoi(mo);
if ((m>1) && (m<12)) {
    year = _ttoi(ye);    // 년도는 첫번째 에디트 박스의 값
    month = _ttoi(mo);  // 달은 두번째 에디트 박스의 값
    day_ = 1;
    DrawMode = FALSE;
    Invalidate();
}
UpdateData(FALSE);
}

void CCalendarManage::OnDeletememo() //일정삭제 다이얼로그 띄우기
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.

    yearpre = m_year;
    monthpre = m_month;
    CCalDeleteMemoDlg dlg;
    dlg.DoModal();

    Invalidate();
}

void CCalendarManage::OnInitialUpdate() //달력 배열 초기화
{
    CFormView::OnInitialUpdate();

    // TODO: 여기에 특수화된 코드를 추가 및/또는 기본 클래스를 호출합니다.
    int i, x, y, j, k, r;
    //rect배열 값 넣기
    for (i=0; i<weekcount; i++)
    {
        y = y1 + gap*i;
        for (j=0; j<daycount; j++)
        {
            rect[i][j].left = x1+gap*j;
            rect[i][j].top = y1+gap*i;
            rect[i][j].right = x1+gap*(j+1);
            rect[i][j].bottom = y1+gap*(i+1);
        }
    }

    font.CreateFont(20,15,0,0, FW_NORMAL, FALSE, FALSE, 0,
    DEFAULT_CHARSET, OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS,
    DEFAULT_QUALITY, DEFAULT_PITCH | FF_SWISS, _T("나눔바른고딕"));
    GetDlgItem(IDC_PREYEAR)->SetFont(&font);
    GetDlgItem(IDC_PREMONTH)->SetFont(&font);
    GetDlgItem(IDC_STATIC1)->SetFont(&font);
    GetDlgItem(IDC_STATIC2)->SetFont(&font);

    m_prev.LoadBitmaps(IDB_BMP_PREV,NULL,NULL,NULL);
    m_prev.SizeToContent();
    m_next.LoadBitmaps(IDB_BMP_NEXT,NULL,NULL,NULL);
    m_next.SizeToContent();

```



```

}
void CCalendarManage::OnTodaygo() //오늘날짜로 이동
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.

    CTime timer;
    timer = CTime::GetCurrentTime();
    // 현재 년월일 구하기
    int yea = timer.GetYear();
    int mont = timer.GetMonth();
    year = yea; // 년도는 첫번째 에디트 박스의 값이죠.
    month = mont; // 달은 두번째 에디트 박스의 값입니다.
    day_ = 1;
    DrawMode = FALSE;
    Invalidate();
}

```

### [환경설정 - 관리자페이지 - 권한/사용여부]

```

void CManagerPermitDlg::OnBnClickedOk() //권한 변경
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

    UpdateData(TRUE);

    nIndex = m_manage_Combo.GetCurSel();
    if(nIndex == -1)
        return;
    m_manage_Combo.GetLBText(nIndex, strSort);

    CDatabase db; // 데이터베이스 객체 생성
    CRecordset rs(&db); // 레코드셋 객체 생성
    try
    {
        CString strConnection(L"DRIVER={Microsoft Access Driver (
        *.mdb)};DBQ=C:\\WWWDownload\\WWWToyPocketDB.mdb;"); //DB오픈

        if (db.OpenEx(strConnection) == FALSE)
            throw _T("데이터베이스 접속실패!");

        CString strQuery; //데이터베이스에 전송할 쿼리문
        strQuery.Format(_T("UPDATE UserData SET permission = '%s' ,
        user_YN = '%s' WHERE id='%s'"), strSort, strRadio, m_loginid);

        db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송
    }
    catch (CException * e)
    {
        e->ReportError();
        e->Delete();
    }
    catch (LPCTSTR msg)
    {
        AfxMessageBox(msg);
    }
    MessageBox(_T("성공적으로 수정되었습니다."));
    //OnBnClickedButtonComClear(); //컨트롤 초기화

    rs.Close(); //레코드셋 객체 소멸
    db.Close(); //데이터베이스 객체 소멸
}

```

```

        m_manage_Combo.SetCurSel(-1);
        m_idEdit.SetWindowTextW(_T(""));
        CButton* pButton = (CButton*)GetDlgItem(IDC_RADIO1);
        pButton->SetCheck(false);
        CButton* pButton2 = (CButton*)GetDlgItem(IDC_RADIO2);
        pButton2->SetCheck(false);

        GetDlgItem(IDC_STATIC4)->ShowWindow(SW_HIDE);
        GetDlgItem(IDC_STATIC5)->ShowWindow(SW_HIDE);
        GetDlgItem(IDC_STATIC6)->ShowWindow(SW_HIDE);
        GetDlgItem(IDC_EDIT2)->ShowWindow(SW_HIDE);
        GetDlgItem(IDC_EDIT3)->ShowWindow(SW_HIDE);

        CDialogEx::OnOK();
    }

```

### [환경설정 - 관리자페이지 - 공지사항등록]

```

void CNoticeDlg::OnBnClickedAddBtn() //공지사항 등록
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(TRUE);
    if(m_addnotice!=""){
        CTime tm = CTime::GetCurrentTime();
        CString sdate;
        sdate.Format(_T("%d-%02d-%02d"),tm.GetYear(),
            tm.GetMonth(),tm.GetDay());

        CDatabase db; // 데이터베이스 객체 생성
        CRecordset rs(&db); // 레코드셋 객체 생성
        try
        {
            CString strConnection(L"DRIVER={Microsoft Access Driver
            (*.mdb)};DBQ=C:\\WWW\\Download\\WWWToyPocketDB.mdb;"); //DB오픈

            if (db.OpenEx (strConnection) == FALSE)
                throw _T("데이터베이스 접속실패!");

            CString strQuery; //데이터베이스에 전송할 쿼리문
            strQuery.Format(_T("INSERT INTO Notice VALUES ('%s', '%s')"),
                sdate, m_addnotice);

            db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송
        }
        catch (CException * e)
        {
            e->ReportError();
            e->Delete();
        }
        catch (LPCTSTR msg)
        {
            AfxMessageBox (msg);
        }
        MessageBox(_T("성공적으로 추가되었습니다. "));
        //OnBnClickedButtonComClear(); //컨트롤 초기화
        //쿼리문을 이용해 검색한 내용 리스트 컨트롤에 출력
        rs.Close(); //레코드셋 객체 소멸
        db.Close(); //데이터베이스 객체 소멸
    }else MessageBox(_T("입력되지 않은 값이 있습니다.\n모든 값을
        입력(선택)해주세요. "));
}

```

```

        m_addnotice = _T("");
        UpdateData(false);
        Invalidate();
    }

void CNoticeDlg::OnBnClickedDeleteBtn()    //공지사항 삭제
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    int ret;
    ret = AfxMessageBox(_T("삭제하시겠습니까?"), MB_YESNO|MB_ICONSTOP);
    if(ret == IDYES){
        UpdateData(TRUE);

        CDatabase db;                                // 데이터베이스 객체 생성
        CRecordset rs (&db);    // 레코드셋 객체 생성

        try
        {
            CString strConnection(L"DRIVER={Microsoft Access Driver
            (*.mdb)};DBQ=C:\\WWWDownload\\WWWToyPocketDB.mdb;");

            if (db.OpenEx (strConnection) == FALSE)
                throw _T("데이터베이스 접속실패!");

            CString strQuery;    //데이터베이스에 전송할 쿼리문
            strQuery.Format(_T("DELETE * FROM Notice WHERE n_notice = '%s'
            "),nnotice);

            db.ExecuteSQL(strQuery);    //데이터베이스에 쿼리문 전송

        }
        catch (CException * e)
        {
            e->ReportError();
            e->Delete();
        }
        catch (LPCTSTR msg)
        {
            AfxMessageBox (msg);
        }

        rs.Close();    //레코드셋 객체 소멸
        db.Close();    //데이터베이스 객체 소멸
    }
    Invalidate();
}

void CNoticeDlg::OnBnClickedBModiBtn()    //공지사항 수정
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    UpdateData(TRUE);
    if(m_addnotice!=""){

        CTime tm = CTime::GetCurrentTime();
        CString sdate;
        sdate.Format(_T("%d-%02d-%02d"),tm.GetYear(),tm.GetMonth(),tm.GetDay());

        CDatabase db;                                // 데이터베이스 객체 생성
        CRecordset rs (&db);    // 레코드셋 객체 생성
        try
    
```

```

{
    CString strConnection(L"DRIVER={Microsoft Access Driver
(*.mdb)};DBQ=C:\\\\Download\\\\ToyPocketDB.mdb;"); //DB오픈

    if (db.OpenEx (strConnection) == FALSE)
        throw _T("데이터베이스 접속실패!");

    CString strQuery; //데이터베이스에 전송할 쿼리문
    strQuery.Format(_T("UPDATE Notice SET n_date='%s',
n_notice='%s' WHERE n_notice='%s'"),sdate, m_addnotice,
nnotice);

    db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송
}
catch (CException * e)
{
    e->ReportError();
    e->Delete();
}
catch (LPCTSTR msg)
{
    AfxMessageBox (msg);
}
MessageBox(_T("성공적으로 수정되었습니다. "));
//OnBnClickedButtonComClear(); //컨트롤 초기화

rs.Close(); //레코드셋 객체 소멸
db.Close(); //데이터베이스 객체 소멸
}else MessageBox(_T("입력되지 않은 값이 있습니다.\n모든 값을
입력(선택)해주세요. "));

m_addnotice=_T("");
UpdateData(FALSE);
Invalidate();
GetDlgItem(IDC_BUTTON1)->ShowWindow(SW_SHOW);
GetDlgItem(IDC_BUTTON2)->ShowWindow(SW_SHOW);
GetDlgItem(IDC_BUTTON3)->ShowWindow(SW_HIDE);
}

```

## [환경설정 - 비밀번호변경]

```

void CPasswordChangeDlg::OnBnClickedOk() //비밀번호 변경
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

    UpdateData(TRUE);
    if(m_pwd == g_pwd){
        if(m_changepwd == m_changepwd2){
            CDatabase db; // 데이터베이스 객체 생성
            CRecordset rs (&db); // 레코드셋 객체 생성
            try
            {
                CString strConnection(L"DRIVER={Microsoft Access Driver
                (*.mdb)};DBQ=C:\\WWDownload\\WWToyPocketDB.mdb;"); //DB오픈

                if (db.OpenEx (strConnection) == FALSE)
                    throw _T("데이터베이스 접속실패!");

                CString strQuery; //데이터베이스에 전송할 쿼리문
                strQuery.Format(_T("UPDATE UserData SET password = '%s' WHERE
                id='%s' AND password = '%s'"), m_changepwd, g_setid, m_pwd);

                db.ExecuteSQL(strQuery); //데이터베이스에 쿼리문 전송
            }
            catch (CException * e)
            {
                e->ReportError();
                e->Delete();
            }
            catch (LPCTSTR msg)
            {
                AfxMessageBox (msg);
            }
            MessageBox(_T("성공적으로 수정되었습니다. "));
            m_pwdEdit.SetWindowTextW(_T(""));
            m_chpwdEdit.SetWindowTextW(_T(""));
            m_chpwd2Edit.SetWindowTextW(_T(""));
            //OnBnClickedButtonComClear(); //컨트롤 초기화

            rs.Close(); //레코드셋 객체 소멸
            db.Close(); //데이터베이스 객체 소멸
        }

    }

    else{
        AfxMessageBox(_T("비밀번호가 서로 다릅니다. "));
        m_chpwdEdit.SetWindowTextW(_T(""));
        m_chpwd2Edit.SetWindowTextW(_T(""));
    }
}

else{
    AfxMessageBox(_T("현재 비밀번호와 다릅니다. "));
    m_pwdEdit.SetWindowTextW(_T(""));
    m_chpwdEdit.SetWindowTextW(_T(""));
    m_chpwd2Edit.SetWindowTextW(_T(""));
}

}

CDialogEx::OnOK();
}

```

### 3.7 전체 프로그램의 사용법(메뉴, 컨트롤 등)

#### [사용자 권한]

- super 가장 최상위 관리자로서 한 사람만이 가질 수 있는 권한이다.  
이 권한은 모든 메뉴에 접근이 가능하다.
- admin 상위 관리자로서, super만이 접근 가능한 관리자페이지를 제외한 모든 메뉴를 이용가능하다.
- manager 하위 관리자로서, 회사의 이윤에 직접적인 영향을 끼치는 메뉴를 제외한 거래처 관리, 상품관리, 일정관리, 비밀번호 변경 메뉴만 이용가능하다.

#### [초기화면]

##### 로그인

- 회원가입이 되어 있는 아이디와 비밀번호를 입력하여 '로그인' 버튼을 누르게 되면 홈 화면으로 이동하여 프로그램을 정상적으로 이용가능하다.

##### 회원가입

- 개인 정보를 입력 후 '회원가입' 버튼을 누르면 사용자 정보를 저장한다.  
처음 가입하는 사용자는 권한이 'manager' 이다.

#### [홈 화면]

##### - 간단한 일정

홈 화면에서 일정을 등록하려면 원하는 날짜를 클릭하고 일정을 등록할 수 있는 다이얼로 그가 뜨게 된다. 그 다음 일정 등록 후 '확인' 버튼을 누르면 일정을 등록한 날짜에 색이 채워지게 되고 동시에 DB에 저장된다.

##### - 여섯개의 퀵버튼(quick button)

- |   |       |                  |
|---|-------|------------------|
|  | 거래처등록 | 거래처 등록 창으로 이동한다. |
|  | 상품등록  | 상품 등록 창으로 이동한다.  |
|  | 매출조회  | 판매 조회 창으로 이동한다.  |
|  | 교환조회  | 교환 조회 창으로 이동한다.  |
|  | 반품조회  | 반품 조회 창으로 이동한다.  |
|  | 환경설정  | 비밀번호 변경 창이 뜬다.   |

#### [거래처/상품/판매/교환/반품 관리]

##### 확인

- 등록 탭에 있는 모든 컨트롤에 정보를 입력하고 '확인' 버튼을 누르면 정보를 DB에 저장한다.

##### 취소

- 모든 컨트롤을 초기 상태로 초기화 시킨다.

##### 조회

- 조건을 지정하여 키워드를 입력 후 '조회' 버튼을 누르면 리스트에 결과에 맞는 목록을 볼 수 있다.

##### 초기화

- 모든 컨트롤을 초기 상태로 초기화 시킨다.

##### 엑셀 저장

- 조건 검색 후 리스트 목록들을 엑셀로 저장할 수 있다.

### 적용

- 수정 탭에 있는 컨트롤 중에 수정하고 싶은 정보를 입력하고 '적용' 버튼을 누르면 DB에 있는 정보를 수정한다.

### 이미지 등록

- 이미지를 불러온다.

### 이미지 삭제

- 선택한 이미지를 취소한다.
- 조회 탭에서 수정하고 싶은 정보를 더블클릭하게 되면 더욱 손쉽게 수정이 가능하다.
- 삭제하고 싶은 정보는 조회 탭의 리스트에 삭제하고 싶은 항목을 선택 후 마우스 오른쪽 버튼을 누르면 삭제 메뉴가 뜨게 되는데 이를 통해 삭제가 가능하다.
- 조회 탭에서 오름차순/내림차순을 지정하고 싶은 열의 헤더를 클릭하게 되면 리스트 목록이 클릭에 따라 나열되게 된다.(가장 왼쪽에 위치한 열의 헤더에 있는 삼각형의 모양으로 오름차순/내림차순을 판별가능하다.) 또한 열의 이동이 가능하여 이동하고 싶은 열을 드래그하여 원하는 위치에 이동할 수 있다.

### [일정관리]

(※여기서 등록한 일정은 홈 화면에도 갱신되어 나타난다.)



이전 달로 이동가능하다.



다음 달로 이동가능하다.

- 원하는 날짜를 클릭하면 일정을 등록할 수 있는 다이얼로그가 뜨고 일정 등록 후 '확인' 버튼을 누르면 달력에 일정이 표시되고 동시에 DB에 저장된다.
- 일정 추가/수정 다이얼로그에는 수행여부를 나타내는 체크박스들이 있는데 체크박스를 체크하게 되면 초록색의 동그라미로 체크하지 않은 일정은 회색의 동그라미로 나타낸다.
- 일정 이동/삭제/오늘 날짜로 이동을 하고 싶은 경우, 일정관리 페이지에서 마우스 오른쪽 버튼을 클릭하면 이용가능하다.

### [환경설정 - 관리자페이지 - 권한/사용여부]

- 권한/사용여부를 변경하길 원하는 사용자를 '검색' 버튼을 이용하여 정보를 불러온 후 '확인' 버튼을 누르면 정보를 수정 가능하다.

### [환경설정 - 관리자페이지 - 공지사항등록]

(※여기서 등록한 공지사항은 홈 화면에도 갱신되어 나타난다.)



컨트롤에 입력된 내용을 공지사항에 등록할 수 있다.



삭제하고 싶은 공지사항을 클릭 후 이 버튼을 누르면 삭제가 가능하다.

### 수정

수정하고 싶은 공지사항을 더블클릭하거나 마우스 오른쪽 버튼을 이용하여 내용을 변경한 후 이 버튼을 누르면 수정가능하다.

### [환경설정 - 비밀번호변경]

- 현재비밀번호를 입력하고 변경하고 싶은 비밀번호를 입력한 후 '확인' 버튼을 누르면 변경된 비밀번호로 이용가능하다.

### 3.8 업그레이드 요구 사항

- 현재 미해결 기능

- 일정 수정 기능

일정을 등록한 후 수정하고 싶은 일정에 대해서는 아직 수정이 불가하다.

현재까지는 수정한 경우 DB에 새로 저장되도록 되어 있다. 그렇게 되면 같은 날짜에 일정DB의 정보가 두 개나 생기게 된다. 하여 수정한 부분만 업데이트하려고 시도했지만 코딩이 더욱 복잡해지고 어려웠다. 그래서 전에 입력되어 있던 일정을 delete 쿼리문을 이용해 지우고 새로 insert하는 방법을 시도해 볼 것이다.

- 리스트 목록의 깜빡임 현상으로 인한 문제

리스트가 자신의 스크롤 영역보다 아이템들이 더 많아지게 되면 깜빡임 현상을 나타냈다. 이는 스크롤을 한번 클릭하여 주면 이러한 현상은 사라지지만 재고 관리 프로그램 특성 상 뷰가 갱신되는 경우가 많은데 갱신될 때마다 깜빡임 현상을 나타냈다. 알고보니 onPaint() 함수에서는 '그리기 메시지에 대해서는 CPropertyPage::OnPaint()을(를) 호출하지 마십시오.' 이러한 문구가 쓰여있었다. 이를 해결하기 위해 OnInitDialog() 함수에 코딩하여 초기에 그려주고 등록할 때마다 함수를 호출하여 갱신하도록 하였다. 그러나 같은 PropertySheet에 묶여있는 PropertyPage 클래스 사이에서는 함수를 호출하여 갱신이 가능했지만 아예 다른 클래스들은 갱신이 되지 않고 처음 프로그램을 켰을 때 초기 값을 유지하고 있다.

- 교환/반품관리의 기본이미지 사라짐 현상

초기화면에서는 기본이미지가 잘 유지되어 있지만 다른 메뉴를 이용하고 왔을 시 기본이미지가 사라지는 현상이 나타난다.

- 향후 개선이 요구되는 기능

- 권한에 따른 메뉴 세팅

현재는 활성화/비활성화를 이용하여 권한마다 차이를 두었다. 허나 이는 불필요한 공간/이동이 생길 수 있기 때문에 권한마다 메뉴를 다르게 두게 되면 자신의 권한에 알맞은 기능들만 이용가능하다. 그러나 기존의 메뉴를 지우는 것 까지는 성공했으나 메뉴를 교체하는데 어려움이 있어 이후 개선하고 싶은 사항이다.

- 일정관리에서 오늘날짜 표시/ 마우스를 올려 났을 때 포커스 두기

현재 일정관리에는 오늘 날짜에 해당하는 일에 표시가 되지 않아 오늘이 정확히 며칠인지 알 수 없다. 또한 마우스 커서가 어느 날짜에 있는지 알 수 없어 정확히 표현해주지 않는다. 이는 처음에 초기화된 rect배열 값과 오늘 날짜를 비교하여 색을 채워 넣는데 어려움이 많고 부족한 시간으로 인해 하지 못했지만 향후 개선 할 것이다.



#### 4. 프로젝트 소감(팀원별로 작성)

**조민재:** 맨 처음에는 할 수 있을까하는 막막함 뿐이었는데 조원들과 함께 해나가다 보니 순조롭게 진행되고 협력의 중요성을 알게 된 계기였습니다. 조원들에게 고맙고 고마울 뿐입니다.

**정수연:** 프로그램을 개발해야 한다는 생각에 처음 시작할 때에는 막막하고 걱정이 많았었지만 조원들과 함께 상의하고 많은 문제들을 해결하며 최종적으로 프로그램을 완성할 수 있었던 것 같다. 개발자로서의 첫 결과물로 학교에서 산업체상을 받게 되어서 너무 기쁘고 앞으로도 이번 경험을 토대로 더 좋은 프로그램을 만들 수 있게 되었으면 좋겠다.

**배인정:** 바로 1년 전인가, 그때 시간이 꽤 걸리는 과제가 있었기 때문에 강의실을 찾았습니다. 강의실의 진풍경은 선배님께서 철야를 하신다고 간식거리를 잔뜩 사오시고, 담요와 텀블러까지 준비하신 모습이었습니다. 그리고 지금, 기시감인 듯, 기시감 아닌 기시감 같은 소위 데자뷰를 겪고 있습니다. 하지만 선배님께 들었던 그 어떠한 말보다도 몸소 부딪히면서 조원들과 함께 졸업 작품을 진행하다보니 생각했던 것 보다는 훨씬 좋았던 경험이었습니다. 저희 조원들은 서로 보듬어 줄 수 있는 따뜻한 마음으로 이해해주고, 의견을 나누었기 때문에 잘 지낼 수 있었습니다. 그리고 저 또한, 코딩하면서 느꼈던 어려움과 그 어려움을 헤쳐 나가는 방법 속에서 인내심을 배울 수 있게 되어서 너무나도 소중한 경험이었습니다. 저와 우리 조원들에게 감사하고, 또 감사합니다.

**김원선:** 많은 걸 배운 시간인 것 같습니다. 매우 뜻 깊었던 시간 같습니다. 수고 많으셨습니다.

**지연실:** 조원들끼리 서로 여름 방학때도 자주 만나고, 개학 후엔 학교에 자주 남아서 졸업 작품을 하는 건 힘들었지만 조원들과 친해져서 좋았고, 같이 하나의 프로그램을 만들어서 산업체상을 타서 너무 기뻐다.

**김문선: "Difficulty, But Interest"**

작년 선배들의 졸업 작품 전시회를 보면서 이 전시회가 끝나 가면 이번에 우리 차례가 오는 구나라고 생각했습니다. 그때는 나름 자신 있던 개발언어가 JSP, HTML 등 홈페이지 개발언어들이었습니다. 2학년 말, 프로그램 개발언어를 반마다 다르게 해서 정해졌는데, MFC인 C언어로 결정이 났습니다. 제일 자신 없는 MFC로 선정되면서 3학년 올라오자마자 졸업 작품에 대한 긴장감과 조원들에 대해 민폐가 될까 걱정도 했었습니다. 하지만, 조원선정이 끝나고 조원구성에 대해 만족감으로 잠시 긴장감을 줄일 수 있었습니다. 프로그램 개발 분야 및 주제부터 해서 많은 의견들의 시행착오가 있었지만, 조원들의 빠른 두뇌회전과 화목함이 시행착오를 이겨낼 수 있었습니다. 졸업 작품을 하면서 조원들 각각의 매력(모습)들을 느끼고 밤늦게까지 졸업 작품과의 씨름에서 이겨내어 점점 프로그램의 완성도에 가까워지는 모습에 뿌듯했습니다. 스킨을 입힌 프로그래밍을 분석하고 적용을 하려고 시도를 많이 해봤지만. 종종 실패했었지만, 졸업 작품을 통해 끈기, 분석능력을 배워서인지 여러 번의 시도 끝에 프로그램에 스킨을 입혔습니다. 그때의 감정은 프로그래머들이 개발을 할 때의 느끼는 기쁨 및 희열이 가슴을 두근두근 거리게 하고 조원들에게 조금이나마 스킨 프로그래밍이라도 도울 수 있다는 기쁨에 손가락이 덜덜덜 떨렸습니다. 마침내 졸업 작품 전시회 때, 조원들 덕분에 산업체상을 수상하였고, MFC 개발에 대한 두려움과 어려움이 아직은 쉬움이 아닌 관심으로 바뀌게 되었습니다. 마지막으로 우리 조원들에게 감사하고 사랑하며 앞으로의 취업부터 해서 모든 것이 성공적으로 풀리기를 기원합니다. 우리 조 짱!