JAVA Oracle 중간 프로젝트

Dis (← cocs): 편의점 상품 및 일일 폐기 조회 처리 시스템

KOSEA 인재개발원 웹프로그래밍 31기 23번 진형래

1차 최종 업데이트 : 2020.06.05 2차 최종 업데이트 : 2020.06.12 3차 최종 업데이트 : 2020.06.22

4차 최종 업데이트 : 2020.06.25

Contents:

1 주차 : 요구사항 확인 및 전체 설계 레이아웃 작성

- 1. 아이디어 구상 동기 및 운용 흐름 요약
- 2. 프로그램 요구 기능 및 데이터
- 3. 개발 계획 및 환경
- 4. 프로세스 Flow
- 5. ERD (By StarUML)

2 주차 : DBMS 모델 세부 조정 및 구현

- 1. 구현 과정 간단 소개
- 2. 2주차 ERD (By DBeaver)
- 3. DB 정의서

Contents:

3 주차 : JAVA 프로그램 설계 세부 조정 및 구현

- 1. 추가 및 수정 내역 소개
- 2. Package 구성 소개
- 3. Class Diagram

4 주차 : 최종 수정 및 시연

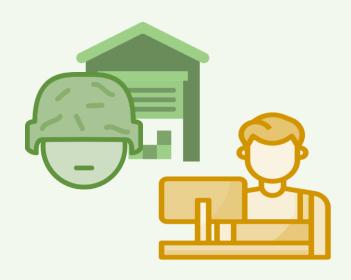
- 1. 프로그램 제작 흐름 및 환경 소개
- 2. 최종 UML
- 3. UI정의서
- 4. 시연
- 5. 중간 프로젝트 소감

1 주차 :

요구사항 확인 및 설계 레이아웃 작성

아이디어 구상 동기

• 관련 근무 경험



- 어쩌다 보니 된 군대 보급병 경험
- 학원 오기 전 마지막 알바가 편의점 (폐기 잘 걸러내는 것으로 유명했…;;)

• 자주 체험한 문제



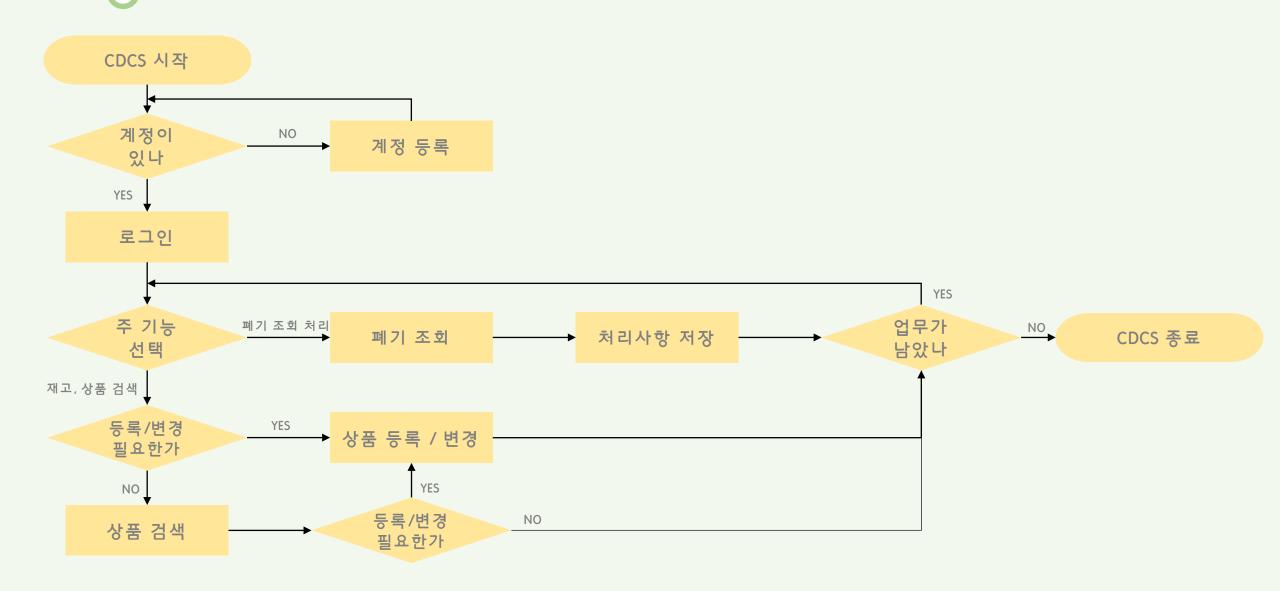
- 폐기 관련 공통 서류 부재(알바)
- 있어도 보기 힘들어서 실수할 경우 생김
- 중요+자주 발생하는 폐기 확인 = 귀찮음

• 추가적인 로컬 DB 필요성



- 어쩔 수 없이 생기는 장부 오차 (가라)
- 관리부대, 본사로 바로 올라가는 부담들
- 비교적 맘편하고 간단한 프로그램 要

기본적인 운용흐름 요약



요구 기본 기능 및 데이터

| 주요 기능 | 세부 기능 | 필요 데이터 |
|---------------------------|--|--|
| 로그인 / 사용자 정보 | 로그인 사용자 등록 ID, PW 찾기 사용자 정보 변경 점내 연락처 확인(부가기능) 탈퇴 | 사용자 : 이름, 아이디, 패스워드, 연락처, 직급, 점포코드 등 점포 : 점포 코드, 점포명 등 |
| 폐기 조회 및 처리 (프로그램 주 목적) | 일일 폐기 조회 월간 폐기 조회 폐기 물품 상태 변경 월간 폐기 상황 조회 | 점포 재고 : 상품코드, 점포코드, 수량, 유통기한, 상품위치, 월간 폐기 횟수, 카운트 기준 월 등 (상품명, 유통기한 타입 등은 상품테이블에서) |
| 재고 조회 및 처리 (부차+필수 기능) | 점내 재고 조회 점내 재고 등록 점내 재고 정보 변경 상품 DB에서 상품 조회 상품 DB에 정보 등록 상품 DB에서 상품 정보 변경 삭제(필요시) | 점포 재고 : 할인율 등의 경우 프로그램 목적상 뺌 상품 : 상품코드, 상품명, 브랜드, 가격, 상품타입, 유통기한 타입 등 |

개발 계획 및 환경

개발 계획 일정

- 개인적인 외부 일정(6월 중순 이사, 병문안 가능성)
- 가능한 학습 범위 내 기능 복습 및 활용
- 기본은 JAVA AWT로 구현 예정, 만약 계획보다 일정의 여유가 생긴다면 Swing학습 및 구현 예정
- 1주차:개발계획및문서작성

2주차: DB작성 (DB는 일정상 구색을 갖추는 정도, 기능구현에 초점)

3주차 ~ 4주차 : Program작성

4주차 : 최종수정 및 발표

사용 툴

- JAVA JDK 14 (IDE : Eclipse WindowBuilder (만약 Swing으로 구현 시))
- Oracle 11g (IDE : DBeaver)
- StarUML
- Google Sheets (유니코드 문제로 MS Excel로 전환)
- MS Power Point

1. Login window



 a

 로그인 실패

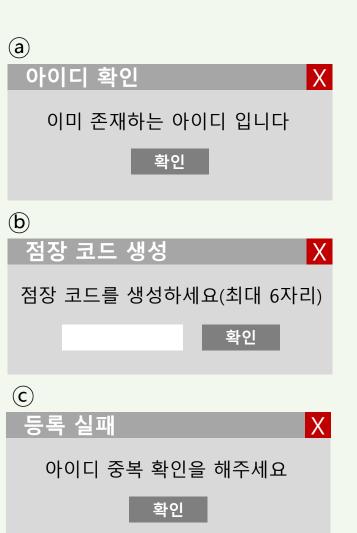
 ID 혹은 PW를 확인해주세요

 확인

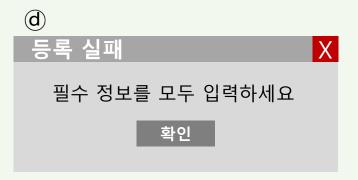
- 로그인 성공 시에만 프로그램의 메인 메뉴를 띄울 수 있도록 함
- ID, PW 입력 후 버튼 클릭 시, 사용자 테이블의 ID, PW 데이터와 대조해 로그인할 수 있게 함
- ID, PW를 입력하지 않고 버튼 클릭 시, 각각의 텍스트 박스에 경고문이 뜨게 함
- ID, PW 오류(DB의 데이터와 일치하지 않음) 발생 시, a창을 띄우게 만듦
- 등록 텍스트 클릭 시, 사용자 등록 창을 띄움
- 아이디 비번 찾기 텍스트 클릭 시, 아이디 비번 창을 띄움
- 로그인 성공 시, 메인 메뉴 창을 띄우고 자동으로 로그인 창 종료

1. Login window





- 로그인 화면에서 등록 텍스트 클릭 시 활성화 (로그인 창은 닫히지 않음)
- ID는 사용자 테이블의 PK이므로 중복확인 기능 필요 (중복 확인 버튼 클릭 시 a or a2창 활성화)
- 직급이 점장일 경우 등록 버튼 클릭 시 b창 활성화
- 등록 시, 상황에 따라 메시지 창 c, d 활성화(b경우도)
- 등록 성공 시, 내용을 DB 사용자 테이블에 저장 후, 창 자동 종료



1. Login window





- 로그인 화면에서 등록 텍스트 클릭 시 활성화 (로그인 창은 닫히지 않음)
- ID는 사용자 테이블의 PK이므로 중복확인 기능 필요 (중복 확인 버튼 클릭 시 a or a2창 활성화)
- 직급이 점장인 경우 등록 버튼 클릭 시 b창 활성화 (6자리 이상 입력이 안되게 함 ← 불가 시 경고창 활성화)
- 등록 시, 상황에 따라 메시지 창 c, d 활성화(b경우도)
- 등록 성공 시, 내용을 DB 사용자 테이블에 저장

등록 실패

 \bigcirc

필수 정보를 모두 입력하세요

확인

1. Login window

검색 실패

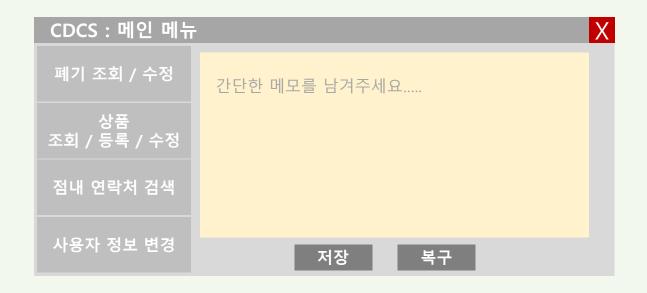


입력 정보를 다시 확인해 주세요

확인

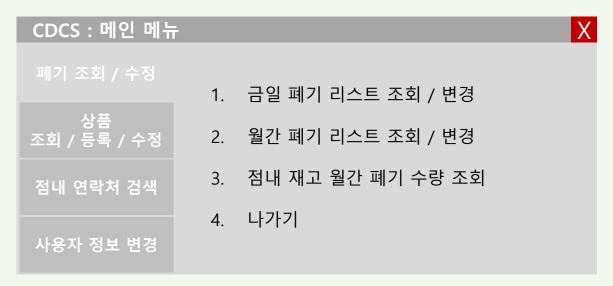
- 로그인 화면에서 아이디 비번 텍스트 클릭 시 활성화 (로그인 창은 닫히지 않음)
- 노란색 텍스트 박스는 입력된 정보에 따라 출력만 가능 (수정 불가)
- 이름, 연락처, 점포코드 입력 후 ID 찾기 클릭 시 해당 유저 테이블의 ID를 노란 ID 텍스트 창에 출력 (입력 정보 부족, 등록된 정보 없을 시 α창 띄움)
- ID 입력 후 힌트확인 버튼 클릭 시,(") 해당 아이디 행의 힌트 데이터를 HINT의 노란 텍스트 박스에 출력
- 힌트 답 입력 후 PW찾기 클릭 시, PW 텍스트 박스에 해당 유저의 PW 데이터 출력 (")

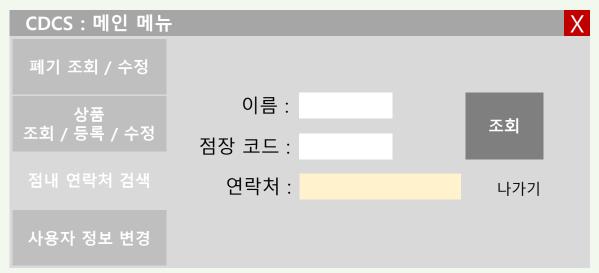
2. Main menu window

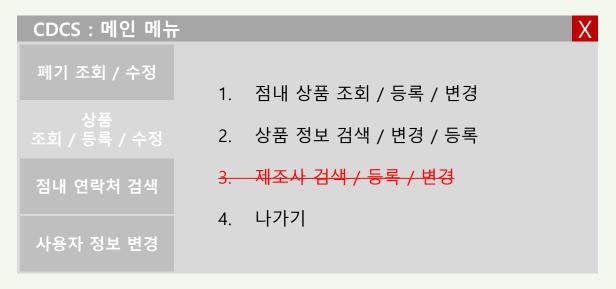


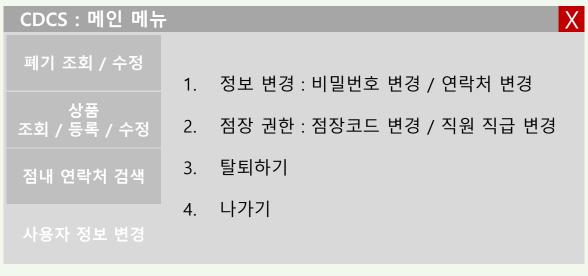
- CDCS 메인 홈 화면
- 텍스트박스에 점내 간단한 공지, 메시지 등을 남길 수 있음
- 텍스트박스의 내용은 자유롭게 수정 가능
- 텍스트 내용은 DB가 아닌 파일로 저장되며, 홈 화면 진입 시 파일에서 텍스트 박스로 출력함
- 수정한 내용은 저장 버튼 클릭 시 파일 내용을 수정시키며,
 복구 버튼을 누르면 가장 최근에 저장된 파일 내용을 불러옴
- 메인 메뉴 왼편의 버튼을 통해 CDCS의 메인 기능창으로 패널을 전환함

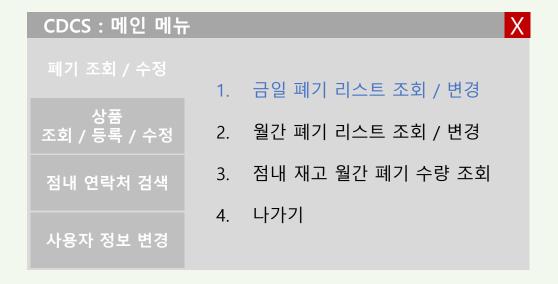
2. Main menu window











- 1, 2, 3번 텍스트 항목을 클릭하면 예시와 같이 대응하는 창이 활성화 됨(메인 메뉴 창은 그대로 유지)
- 4번 나가기 텍스트 클릭 시 홈 화면 이동
- 각 기능 조회 중에는 메인 메뉴 창 비활성화



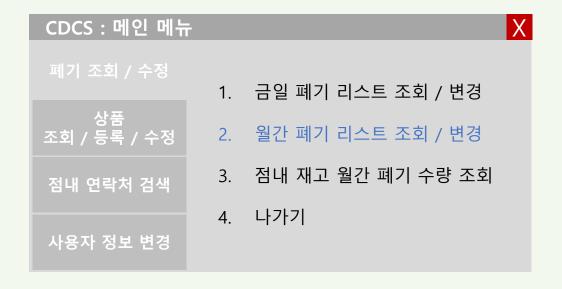
| | 번호 | 상품명 | 위치 | 브랜드 | 재고 | 유통기한 | 폐기타입 |
|---|----|------------|----------|-----|----|------------|------|
| V | 1 | 삼각김밥(전주비빔) | 오픈쇼케이스 A | GS | 2 | 2020-06-25 | 14 |
| | 2 | 인기인가요 샌드위치 | 오픈쇼케이스 A | GS | 1 | 2020-06-25 | 02 |
| V | 3 | 스낵면 | 라면선반 | 농심 | 5 | 2020-06-23 | 00 |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | - |
| | | 재탐색 | 선택 저 | 장 | : | 취소 | |

- FIFO에 따라 조회 된 물품의 재고 및 마지막 유통기한 변경
 (등록 작업과 설비 등 고려 시, 확인 후 변경 사항 입력)
 (설계 목적상 카운터 연동X,
 - 상품코드에 유통기한 정보가 없는 상품이 대부분임 등)
- 조회는 유통기한 속성(DATE)의 데이터와 시스템 날짜를 비교해 놓은 CHAR(1)의 체크1 속성을 참조하여 시스템 효율을 높임
- 변경된 재고의 수량 차이를 재고 테이블의 폐기 카운트 속성에 더해지도록 처리함



- 선택 저장 클릭 시, 확인 창 a가 활성화 됨
 (표 좌측 체크박스에 체크된 행만 DB에 저장이 됨,
 속성 열의 체크박스 클릭 시 전체 선택 가능)
- 재탐색은 최근 저장된 DB정보를 조회함
 (로그인 한 유저의 점포코드, 상품 테이블, 재고 테이블 참조)









- 금일 폐기 리스트 조회 / 변경과 거의 동일한 구조를 지닌다
- 단지 폐기 체크 속성2의 경우 시스템 날짜 데이터와 비교하는 부분이 YYYY-MM부분 까지 같은 경우를 가리키도록 한다



3.1. 폐기 조회 / 수정 window



• 폐기 수량은 폐기 카운트 속성의 값과 카운트 윌의 속성 (시스템 날짜와 비교) 을 이용하여 출력한다

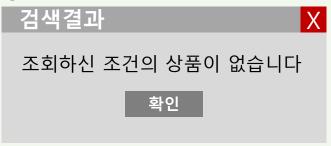
| 번호 | 상품명 | 위치 | 브랜드 | 재고 | 상품 유형 | 폐기수량 |
|----|------------|----------|-----|----|--------|------|
| 1 | 삼각김밥(전주비빔) | 오픈쇼케이스 A | GS | 2 | 즉석식품 | 6 |
| 2 | 인기인가요 샌드위치 | 오픈쇼케이스 A | GS | 1 | 즉석식품 | 4 |
| 3 | 스낵면 | 라면선반 | 농심 | 5 | 라면 | 1 |
| 4 | 알사탕 | 카운터 | 해태 | 3 | 껌 / 사탕 | 1 |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

3.2. 상품 조회 / 등록 / 수정 window





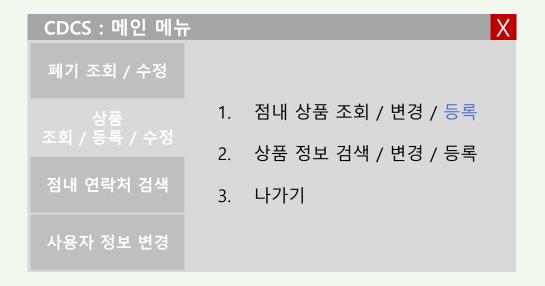
(a)



- 입력된 정보들에 해당하는 값을 표로(다음페이지) 보여줌
- 입력정보에 해당하는 결과가 재고 테이블에 없으면 a창을 활성화
- 상품명과 제조사명의 문자열은 쿼리에서 %문자열%로 검색하게 함
- 카테고리 선택은 대분류를 선택 시, 소분류 선택창이 활성화되게 함

| 번호 | 상품코드 | 상품명 | 가격 | 위치 | 브랜드 | 재고 | 유통기한 | 상품유형1 | 상품유형2 |
|----|---------------|------------|------|----------|-----|----|------------|-------|-------|
| 1 | 1531543680112 | 삼각김밥(전주비빔) | 1000 | 오픈쇼케이스 A | GS | 2 | 2020-06-17 | 즉석식품 | 삼각김밥 |
| 2 | 2315411280112 | 인기인가요 샌드위치 | 2000 | 오픈쇼케이스 A | GS | 1 | 2020-06-18 | 즉석식품 | 샌드위치 |
| 3 | 2415412880112 | 스낵면 | 1200 | 라면선반 | 농심 | 5 | 2020-06-02 | 과자/라면 | 컵라면 |
| 4 | 3212712826612 | 알사탕 | 2000 | 카운터 | 해태 | 3 | 2020-06-23 | 과자/라면 | 사탕 |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |







3.2. 상품 조회 / 등록 / 수정 window



 (a)
 상품코드 검색
 X
 (b)
 검색결과
 X

 상품코드 :
 (c)
 검색결과
 X

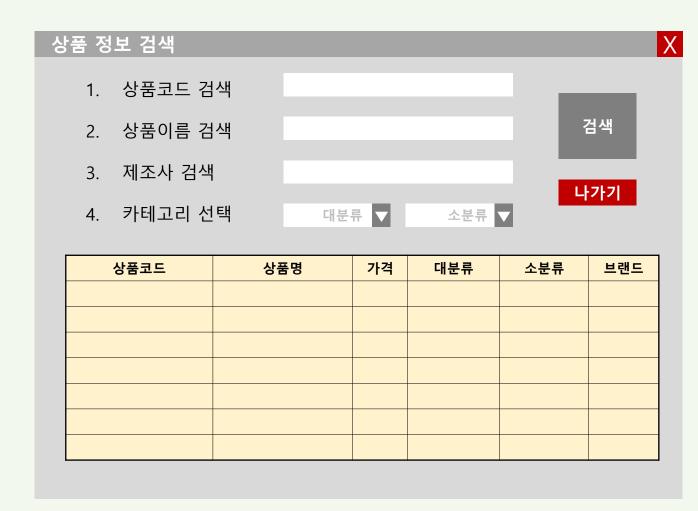
 해당 상품코드의 상품이 없습니다 확인
 확인

- 상품코드는 필수 입력 사항으로 처리함 (후보키의 구성요소) (미입력 후 등록 시, 텍스트 박스에 경고창이 뜨도록 한다)
- 1의 상품코드 검색 열기 텍스트 클릭 시 a창이,
 (상품 이름 입력 후 검색 클릭 시 노란 박스에 결과 출력)
 다중 등록 창 열기 클릭 시 다음 슬라이드의 창이 활성화 됨
 - 중복검사 혹은 상품코드 검색 시, 재고 테이블에 이미 있는 상품일 경우 b창이 활성화 됨 마찬가지로 상품 테이블에도 없는 상품코드 일 경우, c창이 활성화 됨



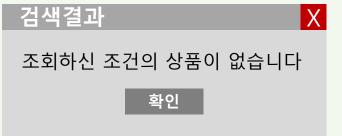
- X 운용원리는 앞 페이지와 유사,단지 표로 최대 10개 리스트까지 입력 가능
 - 상품이름 미 입력 후 중복검사 시,
 중복검사 버튼 아래 텍스트 박스에 중복된
 상품명 나열(줄 바꿈으로 구분)
 - 표의 상품명의 경우, 상품코드 입력 시 바로 해당 상품명을 띄울 수 없는 경우 사양을 변경하기로 함







- 원리는 점내 상품 조회와 유사 (점내 재고가 없는 상품도 조회 가능)
- 검색 결과가 표 범위를 벗어나게 될 경우 자동으로 표의 행 늘림 (만약 불가능 시, 사양 변경 예정)









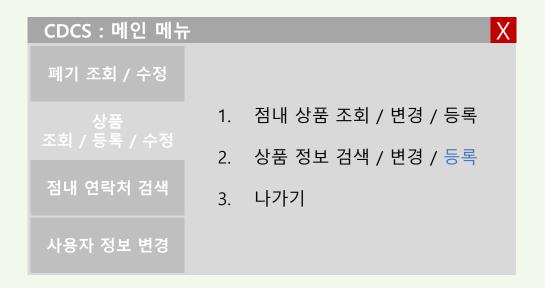
- (a)
 확인
 X

 변경사항을 저장하시겠습니까?
 확인
 취소
- 상제 오류 메세지
 X

 다른 점포에 재고가 남아 삭제가 불가능합니다
 확인
 취소

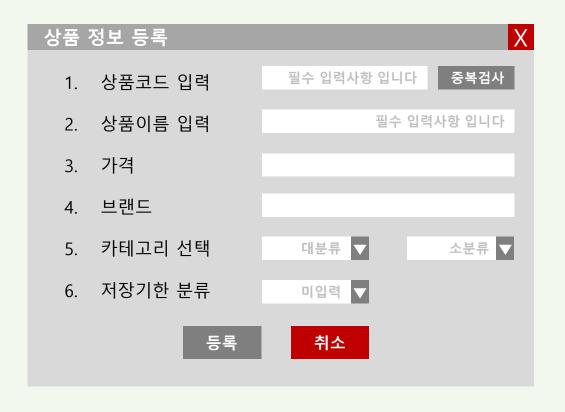
- 프로그램 설계 목적 상 발생 가능한 부주의함과
 여러 점포에서 참고 가능한 상품 테이블의 중요성 상,
 변경 및 삭제는 1상품 씩만 처리하게 하여
 효율성보단 안정성을 택함 (결코 귀찮기 때문이 아님)
- 상품정보 변경 후 변경 혹은 삭제버튼 클릭 시, a버튼 활성화되어 확인 버튼 클릭 시에 만 DB 반영
- 삭제일 경우 해당 상품코드 참조하는 재고 테이블의 모든 행의 재고가 0이 아닐 경우 b 창 활성화
- 안정성을 높이기 위해 이 기능은 아예 삭제하고

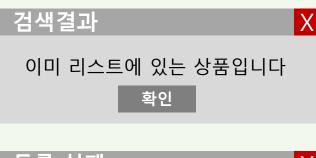
 DB관리자만 편집할 수 있도록 사양을 변경할 수 있음

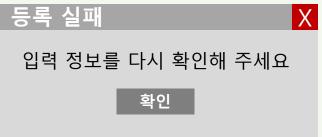




3.2. 상품 조회 / 등록 / 수정 window





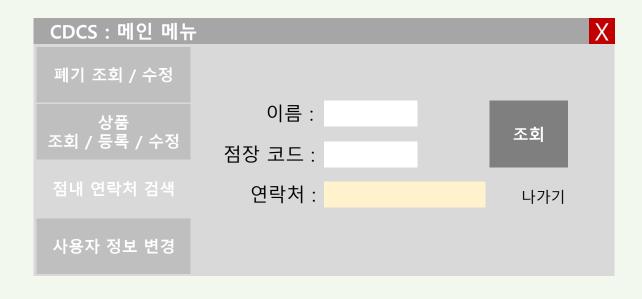




점내 재고 등록과 유사한 흐름
 (참조되는 테이블의 행을 추가
하므로 상품코드는 검색 불가)

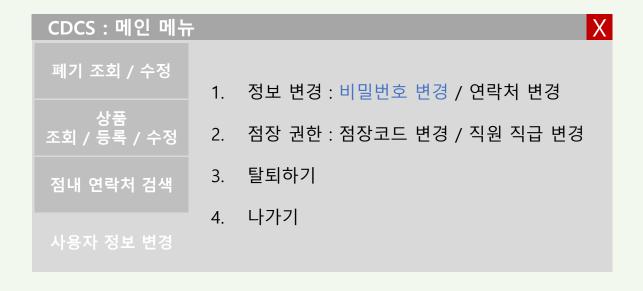
• 브랜드, 제조사, 유통사는 너무 다양하므로 이번 프로그램은 적절히 대표 브랜드만 입력

3.3. 점내 연락처 window



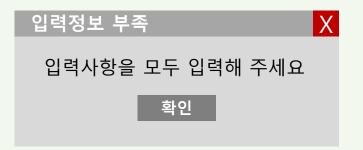
- 가끔 필요할 때 있음 편하겠다 싶은 기능 (카톡방 등이 없는 경우 아르바이트생끼리 연락이 필요할 때)
- 점장코드는 막무가내로 타인의 연락처를 조회할 수 없도록 하는 최소의 보안기능
- 이름, 로그인 된 점포 코드(입력X), 점장코드를 이용하여 조회, 결과는 노란 텍스트 박스에 출력

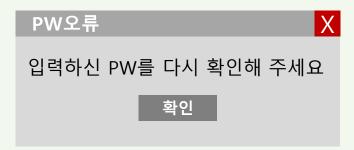
3.4. 사용자 정보 변경 window



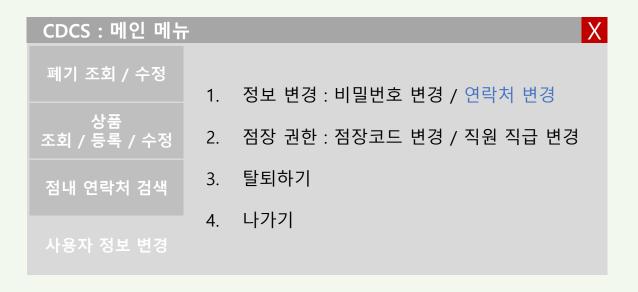






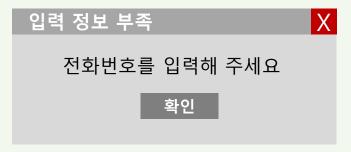


3.4. 사용자 정보 변경 window

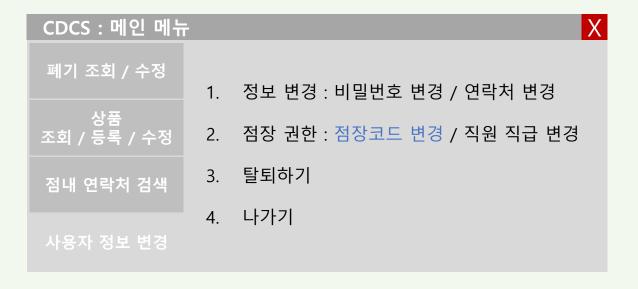




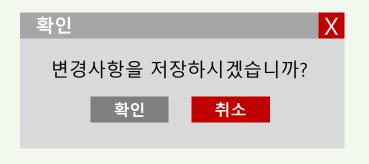


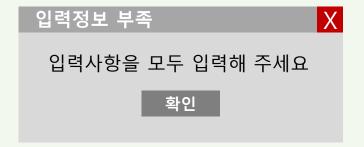


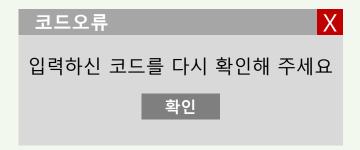
3.4. 사용자 정보 변경 window



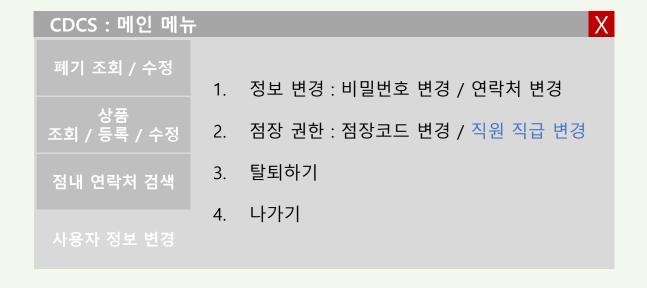


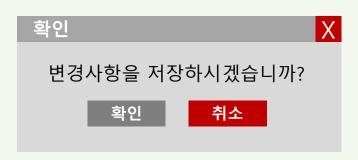






3.4. 사용자 정보 변경 window



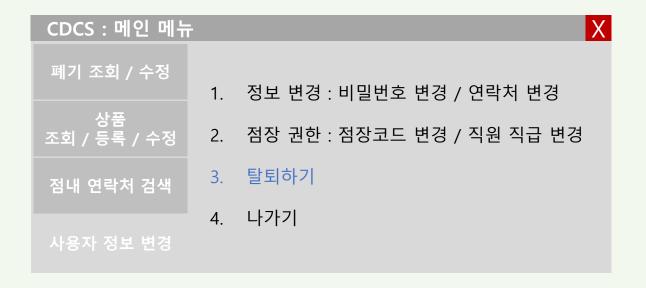


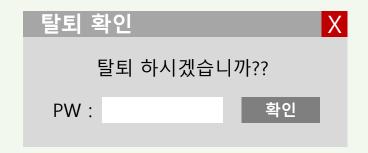


• 직원 아이디 입력 후 확인 버튼 클릭 시, 노란 박스에 현 직급이 뜨도록 구현

Process Flow

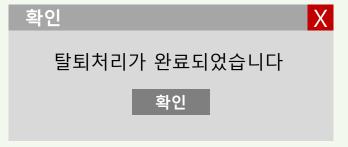
3.4. 사용자 정보 변경 window

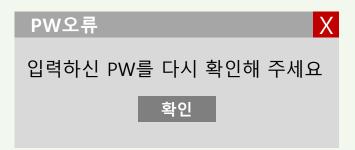




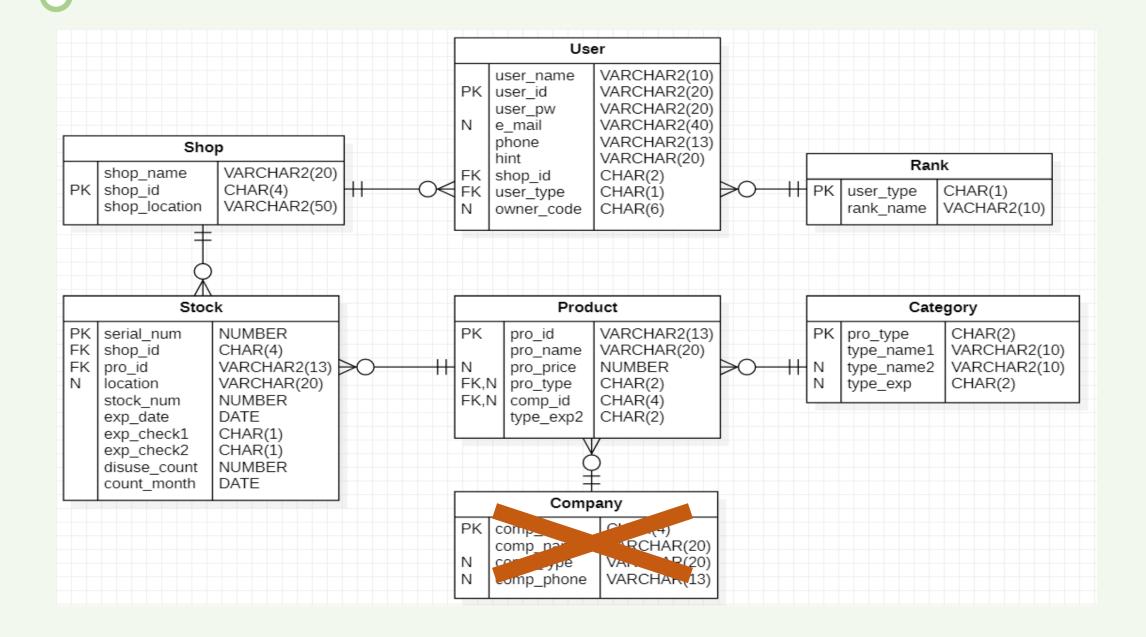
• 탈퇴처리 완료 창(a)의 확인 버튼 클릭 시,
CDCS 메인 메뉴창을 종료 하고 로그인 창 활성화



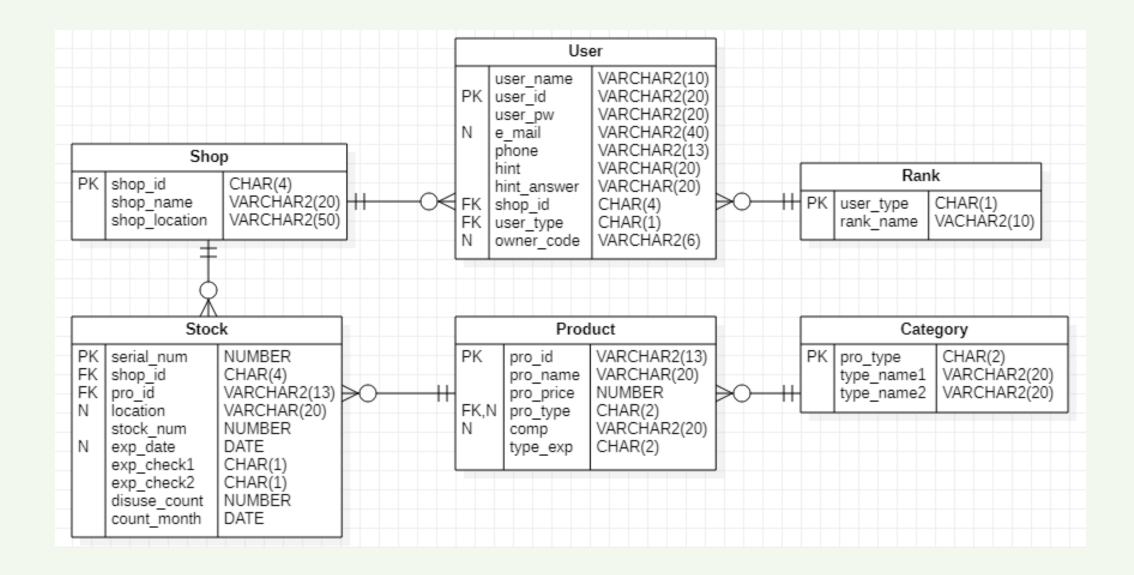




ERD



ERD

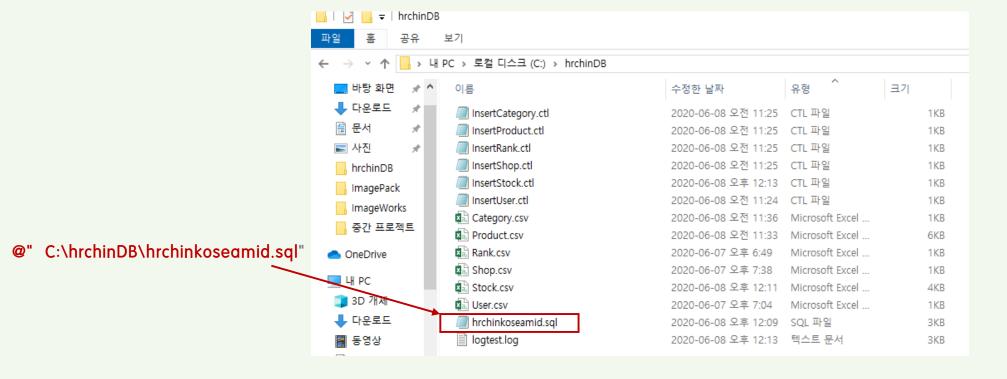


2 주차 :

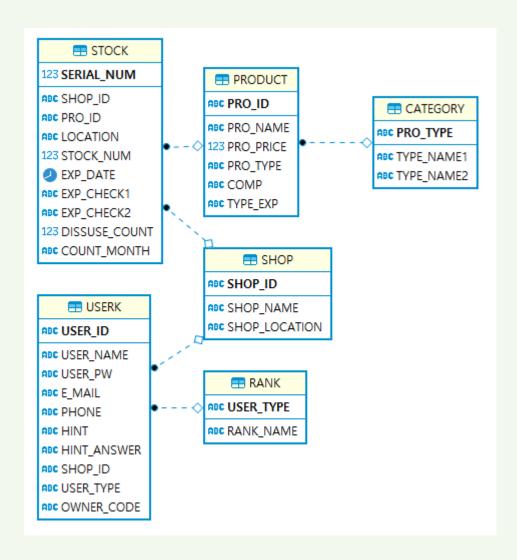
DBMS 모델 세부 조정 및 구현

DB 구현과정

- MS 오피스를 통해 각 테이블에 입력될 데이터 생성 (상품정보는 롯데프레쉬 사이트 참조) 및 *.csv 파일로 저장
- 계정 및 테이블 생성 Query는 SQL plus를 통해 *.sql 파일로 한번에 입력될 수 있도록 작성
- 데이터 입력의 경우 *.ctl 파일과 *.log 파일 생성을 통해 *.csv 파일을 sqlldr로 바로 입력될 수 있도록 Query 작성(위 파일에 저장)
- 로딩 경로의 경우 충돌 문제가 없도록 Root Dir (C:\)에 넣을 hrchinDB폴더로 지정 (C:\hrchinDB\)



ERD (by Dbeaver)



• 데이터 삽입 시, VARCHAR2타입 범위 초과로 인해, 앞 UML에서 정해 놓은 설정 값보다 더 넉넉하게 지정

- COUNT_MONTH는 작업 月만 넣어 비교할 수 있도록 CHAR(2)로 변경 (작업상황에 따라 다시 DATE타입으로 재작성 가능성 있음)
- EXP_CHECK1, 2의 경우 JOB기능을 통해 매일 0시에 자동 업데이트를 하여, 폐기 조회 시점에 검색 효율을 높이려고 계획 했으나,

구현 계획 일정이 촉박할 경우, 위 컬럼 삭제 후 EXP_DATE를 직접 비교하는 방향으로 재작성

3.1. Rank, Shop Table

| 테이블 명 | Rank | | 사용처 | | 계정 등 | 등록, 직 | 급 변경 | 테이블 생성 Query |
|--------|-----------|----------|-------|---------------------|------|-------|------|--|
| 테이블 용도 | 접근 | 권한 레벨에 영 | 향을 주는 | CREATE TABLE Rank (| | | | |
| 컬럼명 | 컬럼 ID | 타입 | 길이 | NOT NULL | PK | FK | 비고 | user_type CHAR(1), rank_name VARCHAR2(10)not null, |
| 직급 번호 | user_tye | CHAR | 1 | Υ | Υ | Υ | | CONSTRAINT PK_Rank PRIMARYKEY(user_type) |
| 직급 명 | rank_name | VARCHAR2 | 10 | Υ | N | N | |); |

| 테이블 명 | Shop | | 사용처 | | 7 | 계정 등 | 록 | 테이블 생성 Query |
|--------|---------------|----------|---------|----------------------|----|------|----|--|
| 테이블 용도 | 로그인한 | 유저가 참조하 | 는 점포 코! | CDEATE TABLE Chara / | | | | |
| 컬럼명 | 컬럼 ID | 타입 | 길이 | NOT NULL | PK | FK | 비고 | CREATE TABLE Shop (shop_id CHAR(4), |
| 점포 코드 | shop_id | CHAR | 4 | Υ | Υ | Υ | | shop_name VARCHAR2(50) not null, shop_location VARCHAR2(50), |
| 점포명 | shop_name | VARCHAR2 | 50 | Υ | N | N | | CONSTRAINT PK_Shop PRIMARY KEY(shop_id) |
| 점포 위치 | shop_location | VARCHAR2 | 50 | N | N | N | |); |

3.2. UserK Table

| 테이블 명 | Userk | (| 사용처 | 로그인, | 계정등· | 록, 계정 | l찾기, 정보변경 | 테이블 생성 Query |
|--------|-------------|-----------|------|-----------------------|------|-------|-----------|--|
| 테이블 용도 | <u> </u> | 로그인 및 프로그 | 램에 접 | CDEATE TABLE Hearly (| | | | |
| 컬럼명 | 컬럼 ID | 타입 | 길이 | NOT NULL | PK | FK | 비고 | CREATE TABLE UserK (user_name VARCHAR2(30)not null, |
| 사용자 이름 | user_name | VARCHAR2 | 30 | Υ | N | N | | user_id VARCHAR2(30), user_pw VARCHAR2(30)not null, |
| 아이디 | user_id | VARCHAR2 | 30 | Υ | Υ | N | | e_mail VARCHAR2(50), |
| 패스워드 | user_pw | VARCHAR2 | 30 | Υ | N | N | | phone VARCHAR2(13)not null, hint VARCHAR2(50)not null, |
| 사용자 메일 | e_mail | VARCHAR2 | 50 | N | N | N | | hint_answer VARCHAR2(50)not null, shop_id CHAR(4), |
| 연락처 | phone | VARCHAR2 | 13 | Υ | N | N | | user_type CHAR(1), |
| 비번 힌트 | hint | VARCHAR2 | 50 | Υ | N | N | | owner_code VARCHAR2(6), CONSTRAINT PK_UserK PRIMARY KEY(user_id), |
| 힌트 답변 | hint_answer | VARCHAR2 | 50 | Υ | N | N | | CONSTRAINT FK_UserK1 FOREIGN KEY(shop_id) |
| 점포 코드 | shop_id | CHAR | 4 | Υ | N | Υ | | REFERENCES Shop(shop_id) ON DELETE CASCADE, CONSTRAINT FK_UserK2 FOREIGN KEY(user_type) |
| 직급 코드 | user_type | CHAR | 1 | Υ | N | Y | | REFERENCES Rank(user_type) ON DELETE SET NULL |
| 점장 코드 | owner_code | VARCHAR2 | 6 | N | N | N | |); |

3.3. Category Table

| 테이블 명 | Catego | 폐기, | 재고, 성 | 상품 검 | 색 등록 삭제 | 테이블 생성 Query | | |
|--------|------------|----------|---------|-------------|----------------------------|--------------|----|---|
| 테이블 용도 | 상품 및 재고 | 검색에서 사용될 | 릴 (주로 7 | 검색 범위 | CDEATE TABLE Cotto warms (| | | |
| 컬럼명 | 컬럼 ID | 타입 | 길이 | NOT NULL | PK | FK | 비고 | CREATE TABLE Category (pro_type CHAR(2), |
| 타입 코드 | pro_type | CHAR | 2 | Υ | Υ | Υ | | type_name1 VARCHAR2(30)not null, type_name2 VARCHAR2(30)not null, |
| 대분류 | type_name1 | VARCHAR2 | 30 | Υ | N | N | | CONSTRAINT PK_Category PRIMARY KEY(pro_type) |
| 소분류 | type_name2 | VARCHAR2 | 30 | Υ | N | N | |); |

3.4. Product Table

| 테이블 명 | Produc | ct | 사용처 | 재고 | 등록, 성 | 상품 검색 | 색 등록 삭제 | 테이블 생성 Query |
|--------|-----------|--------------|------|-------------|--------------------------|-------|------------------------|---|
| 테이블 용도 | 폐기 | 'l, 재고, 상품 조 | 회 등에 | 필요한 상 | CDEATE TABLE Due do de (| | | |
| 컬럼명 | 컬럼 ID | 타입 | 길이 | NOT NULL | PK | FK | 비고 | CREATE TABLE Product (pro_id VARCHAR2(50), |
| 상품 코드 | pro_id | VARCHAR2 | 50 | Υ | Υ | Υ | | pro_name VARCHAR2(50)not null, pro_price NUMBER DEFAULT 0 CHECK(pro_price >= 0), |
| 상품명 | pro_name | VARCHAR2 | 50 | Υ | N | N | | pro_type CHAR(2), |
| 상품 가격 | pro_price | NUMBER | | Υ | N | N | Default 0 Check >=0 | comp VARCHAR2(50), type_exp CHAR(10) not null, |
| 상품 타입 | pro_type | CHAR | 2 | Υ | N | Υ | | CONSTRAINT PK_Product PRIMARY KEY(pro_id), CONSTRAINT FK_Product FOREIGN KEY(pro_type) |
| 브랜드 | comp | VARCHAR2 | 50 | Ν | N | N | 제조사는 X | REFERENCES Category(pro_type) ON DELETE SET NULL |
| 저장성 타입 | type_exp | CHAR | 10 | Υ | N | N | |); |

3.5. Stock Table

| 테이블 명 | Stock | | 사용처 | 폐 | 기, 재고 | 1 검색 | 수정 등록 | 테이블 생성 Query |
|--------|--------------|-----------|-------|-------------|-------|------|------------------------|---|
| 테이블 용도 | 주기능인 | 폐기 조회 및 지 | ㅐ고 검색 | , 수정, 등 | 록에 필 | 요한 정 | 보 저장 | CREATE TABLE Stock (|
| 컬럼명 | 컬럼 ID | 타입 | 길이 | NOT NULL | PK | FK | 비고 | serial_num NUMBER, shop_id CHAR(4), |
| 재고번호 | serial_num | NUMBER | | Υ | Υ | N | 대리키 | pro_id VARCHAR2(50), location VARCHAR2(50), |
| 점포 코드 | shop_id | CHAR | 4 | Υ | N | Y | 로그인 유저 정보 참조 | stock_num NUMBER DEFAULT CHECK(stock_num > = 0), exp_date DATE, |
| 상품 코드 | pro_id | VARCHAR2 | 50 | Υ | N | Y | | exp_check1 CHAR(1), exp_check2 CHAR(1), |
| 점내 위치 | location | VARCHAR2 | 50 | N | N | N | | dissuse_count NUMBER DEFAULT 0 |
| 재고 수량 | stock_num | NUMBER | | Υ | N | N | Default 0 Check >=0 | CHECK(dissuse_count >= 0), count_month CHAR(2), CONSTRAINT PK_Stock PRIMARY KEY(serial_num), |
| 유통 기한 | exp_date | DATE | | Υ | N | N | | CONSTRAINT FK_Stock1 FOREIGN KEY(shop_id) REFERENCES Shop(shop_id) ON DELETE CASCADE, |
| 폐기일 검사 | exp_check1 | CHAR | 1 | Υ | N | N | 진행상황에 | CONSTRAINT FK_Stock2 FOREIGN KEY(pro_id) REFERENCES Product(pro_id) ON DELETE CASCADE, |
| 폐기월 검사 | exp_check2 | CHAR | 1 | Υ | N | N | 따라 삭제 가능 | CONSTRAINT CK_Stock_exp_check1 |
| 폐기 횟수 | disuse_count | NUMBER | | Υ | N | N | Default 0 Check >=0 | CHECK(exp_check1 = '0' OR exp_check1 = '1'), CONSTRAINT CK_Stock_exp_check2 |
| 카운트 월 | count_month | CHAR | 2 | Υ | N | N | | CHECK(exp_check2 = '0' OR exp_check2 = '1')); |

3 주차 : 저 미 그혀

JAVA 프로그램 설계 세부 조정 및 구현

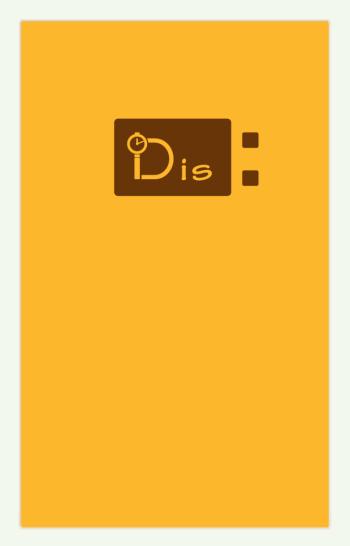
추가 및 수정 내역

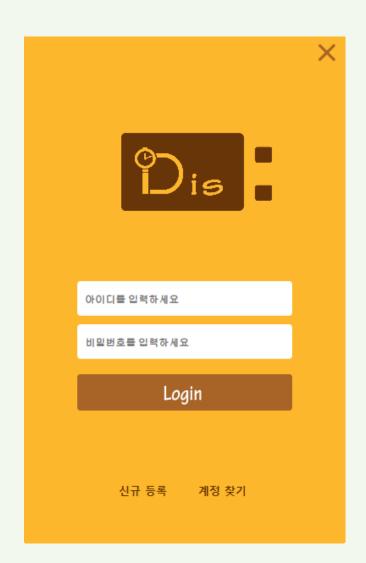
- AWT를 통한 구현에서 Swing으로 변경 확정
- 이미지 삽입을 통해 구현 레이아웃 디자인 대폭 수정 디자인에 따라 필요시 일부 기능 구현 방법 변경
- 삽입되는 이미지(패널, 버튼, 아이콘) 디자인을 위해 개발환경 추가 (Adobe Photoshop, Adobe illustrator, MS PowerPoint)



https://speedwagwon.tistory.com/m/880

패널 이미지ㅠㅠ





• 디자인 시간은 그렇다치고… 깨지고 그림자는 안되고 투명한 부분도 안되고

〈이상〉 〈현실〉

Package 구성1

loginPack

로그인화면, 계정등록 및 찾기 화면 구현 클래스 모음 (프로그램 시작점)

mainPack

메인 메뉴 및 홈 화면을 구현한 클래스 모음

mainResultPanel

메인 메뉴의 세부 기능 패널들을 구현한 클래스 모음

connPack

DB 연결 및 쿼리문 실행, 읽은 DATA값을 임시로 저장하는 클래스들의 모음

Package 구성2

popUpPack

경고 및 확인 등 팝업 창 구현을 위한 클래스 모음

mainPopUp

메인메뉴 세부 기능 중 재고 및 상품 리스트를 띄우는 팝업 창 구현을 위한 클래스 모음

toolPack

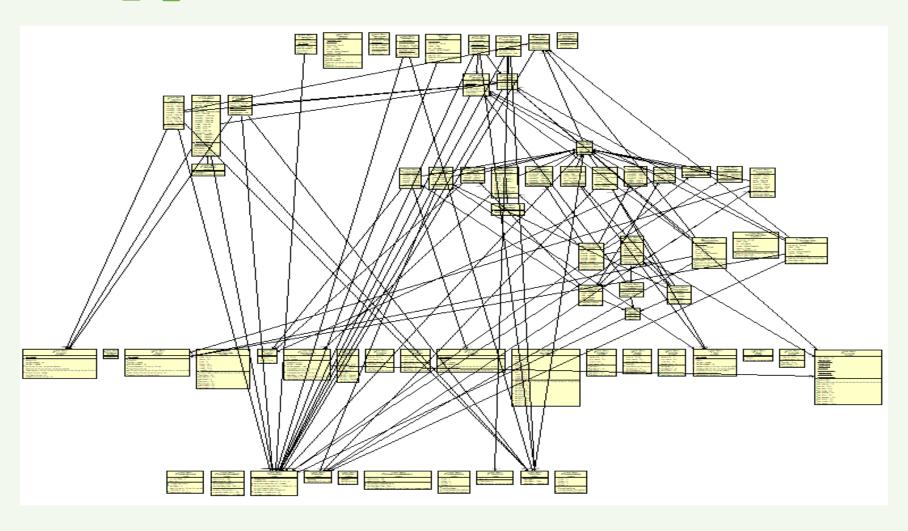
이미지 수정, 마우스 이동 등 자주 사용하는 기능을 구현한 메소드들의 클래스 모음

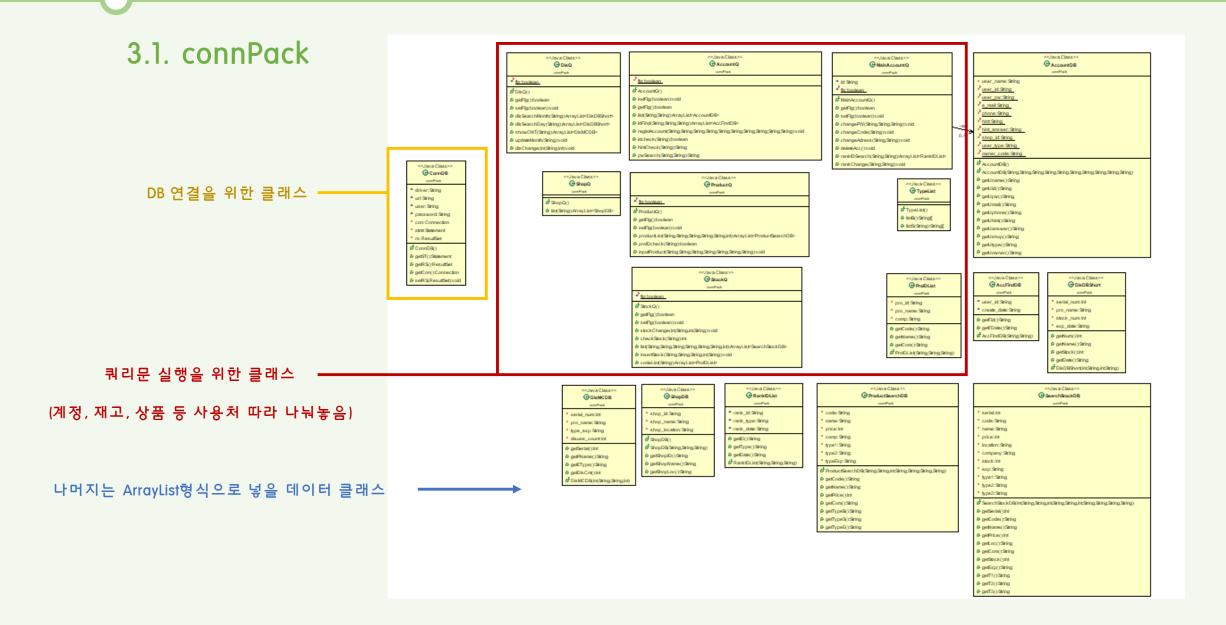
• 기타 (loginImage, mainImage, warningImage 등)

화면에 구현될 배경, 아이콘, 버튼 이미지 등을 저장 및 불러오기 위한 패키지

Class Diagram

3.0. 아.... 이건 좀;;





3.2. toolPack

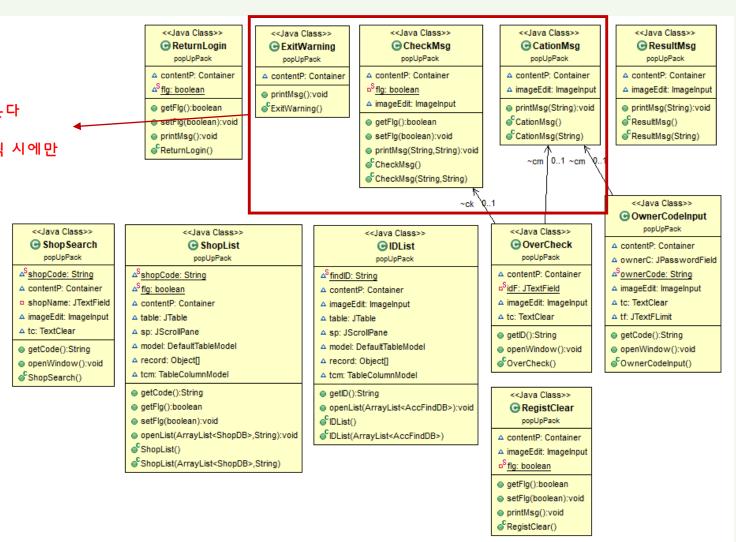
관련 메소드들을 유기적으로 모아 놓음 <<Java Class>> <<Java Class>> <<Java Class>> <<Java Class>> OragMoveDialog ImageInput ● TextClear PersonalComboBoxUI toolPack toolPack toolPack toolPack △ xDrag: int △ background: Color TextClear() o^ClmageInput() △ yDrag: int △ selectground: Color imageSetsize(ImageIcon,int,int):ImageIcon textClean():JTextField △ xPress: int PersonalComboBoxUI() passClean():JPasswordField setOnelmage(JButton,lmagelcon):void △ yPress: int setButtonImage(JToggleButton,Imagelcon,Imagelcon):void setColorBack(Color,Color):void <<Java Class>> DragMoveDialog() setButtonImage(JButton,ImageIcon,ImageIcon):void mouseDragged(MouseEvent):void buttonOpaque(JToggleButton):void installUl(JComponent):void toolPack mouseMoved(MouseEvent):void buttonOpaque(JButton):void o createArrowButton():JButton △ currentValue: int. setPaintLabel(JLabel,Imagelcon):void createRenderer():ListCellRenderer <<Java Class>> □ limit: int setHint(JLabel):void MouseDragMove <<Java Class>> IntegerDocument(int) panelPaint(Imagelcon):JPanel toolPack PersonalComboBoxRenderer textPaint(ImageIcon):JTextArea qetValue():int toolPack △ xDrag: int tableSet(JTable,JScrollPane):void insertString(int,String,AttributeSet):void △ background: Color △ yDrag: int remove(int,int):void △ selectground: Color <<Java Class>> △ xPress: int <<Java Class>> checkInput(String,int):int SizeTool △ yPress: int DateTool PersonalComboBoxRenderer() toolPack toolPack <<Java Class>> MouseDragMove() setColorBack(Color,Color):void DateTool() getListCellRendererComponent(JLi... SizeTool() mouseDragged(MouseEvent):void toolPack mouseMoved(MouseEvent):void vildationDate(String):boo.. setCenterX(int):int □ limit: int setCenterY(int):int JTextFLimit(int) insertString(int,String,AttributeSet):void

만들어 놓고 가장 자주 인스턴스를 생성한 클래스, 이미지 사이즈 조정, 버튼 및 라벨 종류에 따른 이미지 처리

3.3. popPack

가장 자주 사용된 팝업(Dialog) 클래스, 경고, 처리 결과 문구를 문자열 파라미터로 받아 띄운다

CheckMsg 팝업의 경우 위 기능에 더불어 확인 버튼을 클릭 시에만 기능을 수행할 수 있게 하는 확인 팝업 창

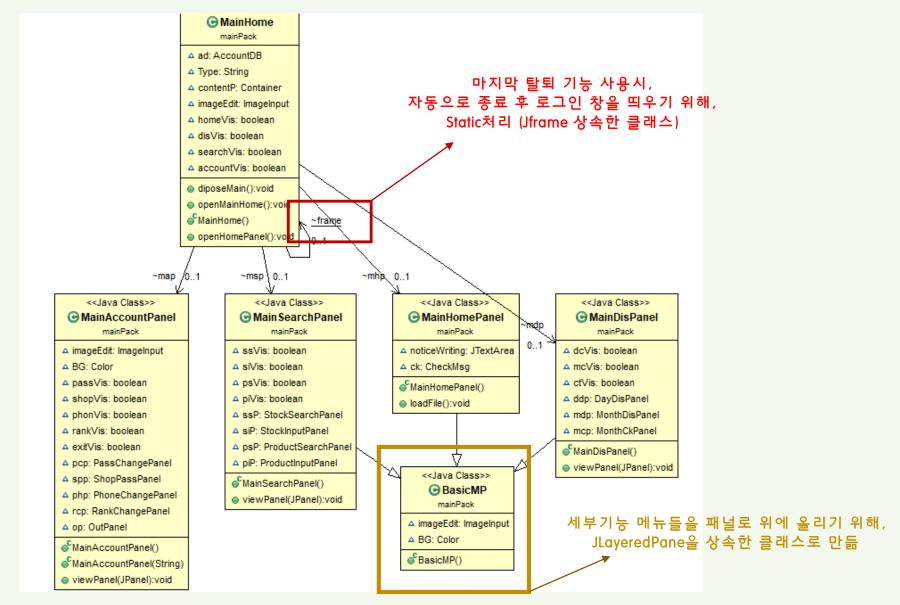


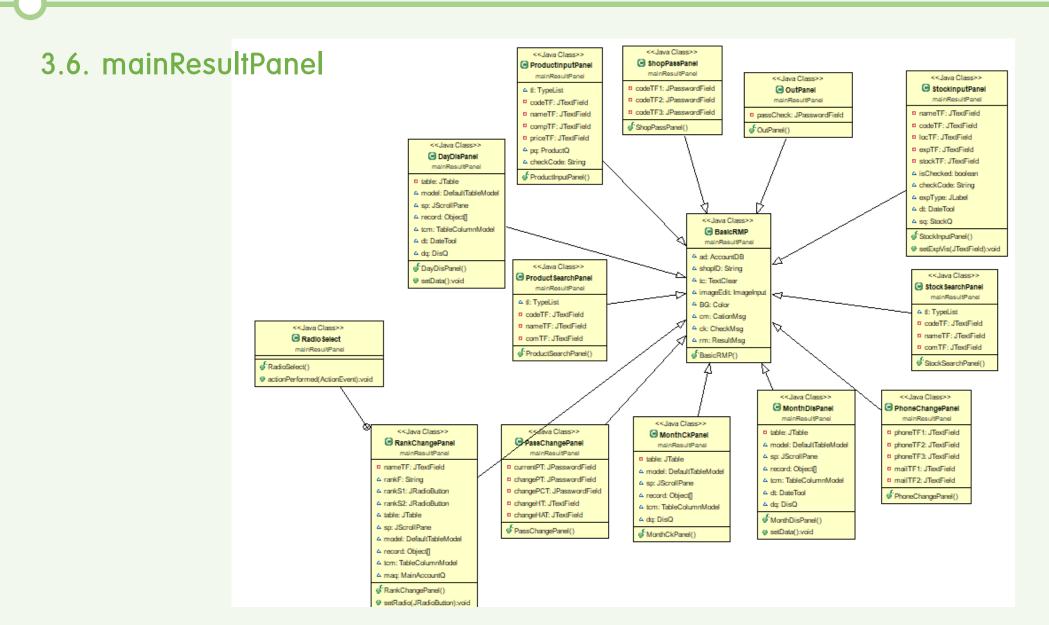


<<Java Class>> ⊕ LoginFind loginPack △ contentP: Container △ imageEdit: ImageInput △ tc: TextClear △ ec: AccountQ △ cm: CationMsg △ rm: ResultMsg nameF: JTextField phoneF1: JTextField phoneF2: JTextField phoneF3: JTextField shopF: JTextField idF: JTextField hintF: JTextField hintAF: JTextField textField: JTextField openWindow():void CLoginFind() <<Java Class>> loginPack RadioSelect() actionPerformed(ActionEvent):void

tfsetting(JTextField):JTextField

3.5. mainPack





3.7. mainPopUp



- △ScodeP: String
- △ contentP: Container
- △ table: JTable
- △ sp: JScrollPane
- △ model: DefaultTableModel
- △ record: Object[]
- △ tcm: TableColumnModel
- △ sud: StockQ
- getCodeP():String
- openList(ArrayList<ProIDList>):void
- StockInputPop()
- StockInputPop(ArrayList<ProIDList>)

<<Java Class>> DisLongPop mainPopUp

- △ contentP: Container
- △ table: JTable
- △ sp: JScrollPane
- △ model: DefaultTableModel
- △ record: Object[]
- △ tcm: TableColumnModel
- △ dt: DateTool
- △ cm: CationMsg
- △ ck: CheckMsg
- △ rm: ResultMsg
- △ sud: StockQ
- △ imageEdit: ImageInput
- openList(ArrayList<DisDB>):void
- ©DisLongPop()

<<Java Class>> ProductSearchPop mainPopUp

- △ contentP: Container
- △ table: JTable
- △ sp: JScrollPane
- △ model: DefaultTableModel
- △ record: Object[]
- △ tcm: TableColumnModel
- openList(ArrayList<ProductSearchDB>):void
- ProductSearchPop()

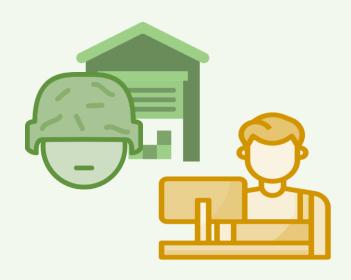
<<Java Class>> Stock SearchPop mainPopUp

- △ contentP: Container
- △ table: JTable
- △ sp: JScrollPane
- △ model: DefaultTableModel
- △ record: Object[]
- △ tcm: TableColumnModel
- △ dt: DateTool
- △ cm: CationMsg
- △ ck: CheckMsg
- △ sud: StockQ
- △ imageEdit: ImageInput
- openList(ArrayList<SearchStockDB>):void
- StockSearchPop()
- StockSearchPop(ArrayList<SearchStockDB>)

4 주차 : 최종 발표 및 시연

아이디어 구상 동기

• 관련 근무 경험



- 어쩌다 보니 된 군대 보급병 경험
- 학원 오기 전 마지막 알바가 편의점 (폐기 잘 걸러내는 것으로 유명했…;;)

• 자주 체험한 문제



- 폐기 관련 공통 서류 부재(알바)
- 있어도 보기 힘들어서 실수할 경우 생김
- 중요+자주 발생하는 폐기 확인 = 귀찮음

• 추가적인 로컬 DB 필요성



- 어쩔 수 없이 생기는 장부 오차 (가라)
- 관리부대, 본사로 바로 올라가는 부담들
- 비교적 맘편하고 간단한 프로그램 要

프로그램 개발 목적 및 구현 테마

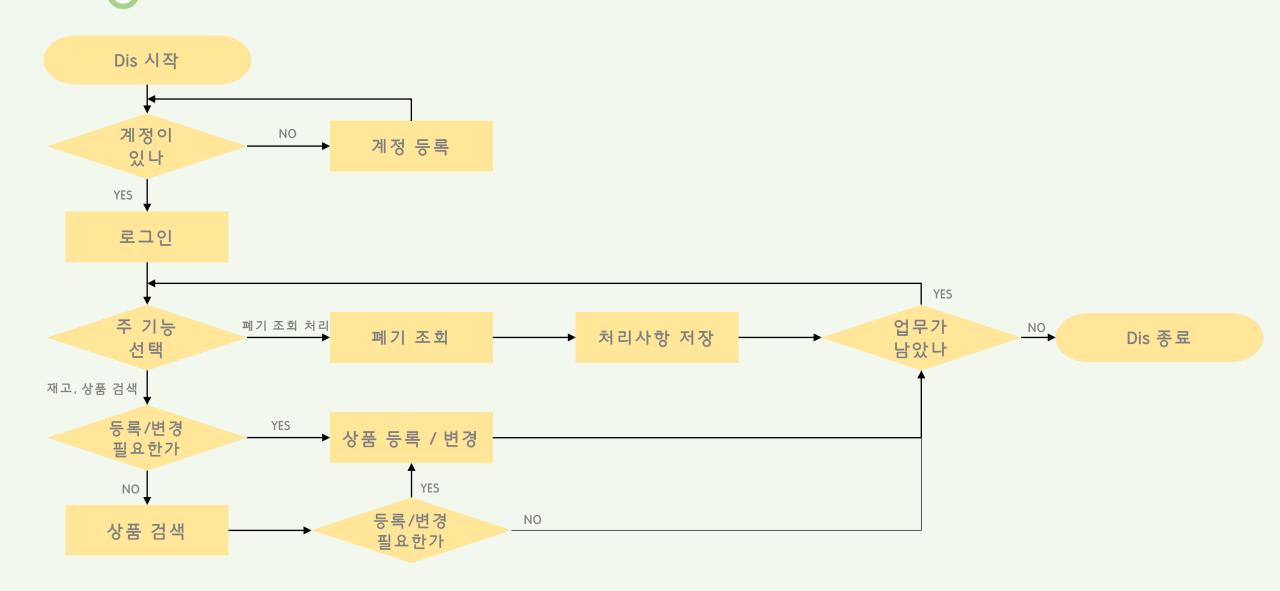
개발 목적

로컬환경에서 비교적 자유롭고 효율적으로 사용할, 폐기 및 재고 조회 관리 프로그램 작성

구현 테마

사용자에게 가능한 세심한 사용 경험을 제공한다

기본적인 운용흐름 요약



개발 환경 및 흐름

OS Window 10 Home Edition JAVA JDK 14 (IDE : Eclipse) JAVA GUI WindowBuilder (in Eclipse) Oracle Oracle 11g (IDE: DBeaver) 문서, DB MS office 2016 (PowerPoint, Excel) Adobe Photoshop, Illustrator (cs6) 디자인

개발 환경 및 흐름



2.1. Entity Relationship Model

AccountD 테이블

로그인 및 사용권한 설정 등을 위한 데이터를 저장

• Shop 테이블

사용자, 특히 재고 구분을 위한 Shop_id와 관련된 데이터 저장

Stock

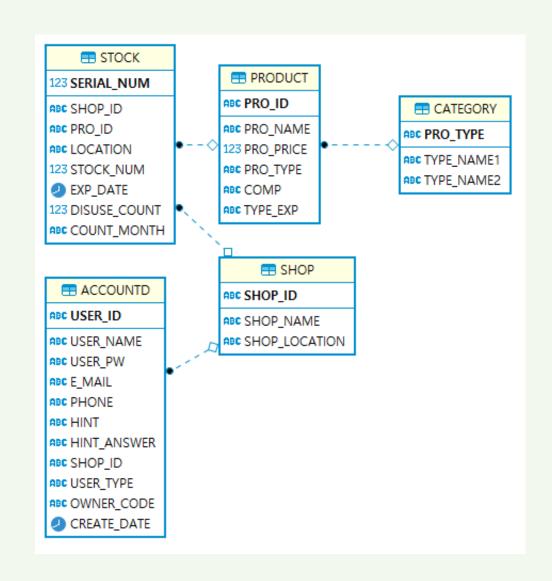
폐기, 사용자 점내 재고 관리를 위한 데이터들을 저장

Product

재고 데이터의 바탕이 되는 상품 데이터들을 저장

Category

효율적 상품, 재고 조회를 위한 상품 분류 데이터들을 저장



2.2. AccountD Table

| 테이블 명 | Accoun | 로그인, | 계정등 | 록, 계정 | 찾기, 정보변경 | 테이블 생성 Query | | |
|--------|-------------|-----------|-------|-------------|----------|-----------------|---------------------------------|---|
| 테이블 용도 | <u>i</u> | 로그인 및 프로그 | 1램에 접 | | | | | |
| 컬럼명 | 컬럼 ID | 타입 | 길이 | NOT NULL | PK | FK | 비고 | CREATE TABLE AccountD (|
| 사용자 이름 | user_name | VARCHAR2 | 30 | Υ | N | N | | user_name VARCHAR2(30)not null, user_id VARCHAR2(30), |
| 아이디 | user_id | VARCHAR2 | 30 | Υ | Υ | N | | user_pw VARCHAR2(30)not null, |
| 패스워드 | user_pw | VARCHAR2 | 30 | Υ | N | N | | e_mail VARCHAR2(50), phone VARCHAR2(13)not null, |
| 사용자 메일 | e_mail | VARCHAR2 | 50 | N | N | N | | hint VARCHAR2(50) not null, |
| 연락처 | phone | VARCHAR2 | 13 | Υ | N | N | | hint_answer VARCHAR2(50)not null, shop_id CHAR(4), |
| 비번 힌트 | hint | VARCHAR2 | 50 | Y | N | N | える テルロ | user_type VARCHAR2(30), |
| 힌트 답변 | hint_answer | VARCHAR2 | 50 | Y | N | N접 . | 추후 추가될 -권한 관련 기능 권한 양도 관련 | owner_code VARCHAR2(6), create_date DATE, |
| 점포 코드 | shop_id | CHAR | 4 | Y | N | Y | 현인 장도 편인 | CONSTRAINT PK_AccountD PRIMARY KEY(user_id), |
| 직급 코드 | user_type | VARCHAR2 | 30 | Y | N | γ 아 | 이디 중복 생성/ 아이디 검색 구분 | CONSTRAINT FK_UserK1 FOREIGN KEY(shop_id) REFERENCES Shop(shop_id) ON DELETE CASCADE |
| 점장 코드 | owner_code | VARCHAR2 | 6 | N | N | N | |); |
| 계정생성일 | create_date | DATE | | Y | | | YYYY/MM/DD | |

2.2. Shop, Category Table

| 테이블 명 | Shop | | 사용처 | | 7 | 계정 등 | 록 | 테이블 생성 Query |
|--------|---------------|----------|--------|----------------------|----|------|----|--|
| 테이블 용도 | 로그인한 | 유저가 참조하는 | - 점포 코 | CDEATE TABLE Chair (| | | | |
| 컬럼명 | 컬럼 ID | 타입 | 길이 | NOT NULL | PK | FK | 비고 | CREATE TABLE Shop (shop_id CHAR(4), |
| 점포 코드 | shop_id | CHAR | 4 | Υ | Υ | Υ | | shop_name VARCHAR2(50) not null, shop_location VARCHAR2(50), |
| 점포명 | shop_name | VARCHAR2 | 50 | Υ | N | N | | CONSTRAINT PK_Shop PRIMARY KEY(shop_id) |
| 점포 위치 | shop_location | VARCHAR2 | 50 | N | N | N | |), |

| 테이블 명 | Catego | Category | | | 재고, 성 | 상품 검 | 색 등록 삭제 | 테이블 생성 Query |
|--------|------------|----------|-------|-------------|-------|------|---------|---|
| 테이블 용도 | 상품 및 재고 | 검색에서 사용될 | 릴 (주로 | | | | | |
| 컬럼명 | 컬럼 ID | 타입 | 길이 | NOT NULL | PK | FK | 비고 | CREATE TABLE Category (pro_type CHAR(2), |
| 타입 코드 | pro_type | CHAR | 2 | Υ | Υ | Υ | | type_name1 VARCHAR2(30)not null, type_name2 VARCHAR2(30)not null, |
| 대분류 | type_name1 | VARCHAR2 | 30 | Υ | N | N | | CONSTRAINT PK_Category PRIMARY KEY(pro_type) |
| 소분류 | type_name2 | VARCHAR2 | 30 | Υ | N | N | |); |

2.2. Product Table

| 테이블 명 | Produc | ct | 사용처 | 재고 | 등록, 성 | 색 등록 삭제 | 테이블 생성 Query | |
|--------|-----------|----------------------|-------------------|-------------|------------------------|---------|------------------------|---|
| 테이블 용도 | 폐2 | 'l, 재고 <i>,</i> 상품 조 | 회 등에 [:] | 필요한 상 | CDEATE TABLE Due don't | | | |
| 컬럼명 | 컬럼 ID | 타입 | 길이 | NOT NULL | PK | FK | 비고 | CREATE TABLE Product (pro_id VARCHAR2(50), |
| 상품 코드 | pro_id | VARCHAR2 | 50 | Υ | Υ | Υ | | pro_name VARCHAR2(50)not null, pro_price NUMBER DEFAULT 0 CHECK(pro_price >= 0), |
| 상품명 | pro_name | VARCHAR2 | 50 | Υ | N | N | | pro_type CHAR(2), |
| 상품 가격 | pro_price | NUMBER | | Υ | N | N | Default 0 Check >=0 | comp VARCHAR2(50), type_exp CHAR(10) not null, |
| 상품 타입 | pro_type | CHAR | 2 | Υ | N | Υ | | CONSTRAINT PK_Product PRIMARY KEY(pro_id), CONSTRAINT FK_Product FOREIGN KEY(pro_type) |
| 브랜드 | comp | VARCHAR2 | 50 | N | N | N | 제조사는 X | REFERENCES Category(pro_type) ON DELETE SET NULL |
| 저장성 타입 | type_exp | CHAR | 10 | Υ | N | N | |); |

2.2. Stock Table

| 테이블 명 | Stock | Stock 사용처 | | | | 그 검색 | 수정 등록 | 테이블 생성 Query |
|---------|--------------|--------------------|-------|----------------------|----|------|------------------------|--|
| 테이블 용도 | 주기능인 | <u>!</u> 폐기 조회 및 제 | 대고 검색 | CREATE TABLE Stock (| | | | |
| 컬럼명 | 컬럼 ID | 타입 | 길이 | NOT NULL | PK | FK | 비고 | serial_num NUMBER, shop_id CHAR(4), |
| 재고번호 | serial_num | NUMBER | | Υ | Υ | N | 대리키 | pro_id VARCHAR2(50), location VARCHAR2(50), |
| 점포 코드 | shop_id | CHAR | 4 | Υ | N | Υ | 로그인 유저 정보 참조 | stock_num NUMBER DEFAULT 0 CHECK(stock_num >= 0), |
| 상품 코드 | pro_id | VARCHAR2 | 50 | Υ | N | Υ | | exp_date DATE, |
| 점내 위치 | location | VARCHAR2 | 50 | N | N | N | | dissuse_count NUMBER DEFAULT 0 CHECK(dissuse_count >= 0), |
| 재고 수량 | stock_num | NUMBER | | Υ | N | N | Default 0 Check >=0 | count_month CHAR(7), CONSTRAINT PK_Stock PRIMARY KEY(serial_num), |
| 유통 기한 | exp_date | DATE | | Υ | N | N | YYYY/MM/DD | CONSTRAINT FK_Stock1 FOREIGN KEY(shop_id) REFERENCES Shop(shop_id) ON DELETE CASCADE, |
| 월 폐기 빈도 | disuse_count | NUMBER | | Υ | N | N | Default 0 Check >=0 | CONSTRAINT FK_Stock2 FOREIGN KEY(pro_id) REFERENCES Product(pro_id) ON DELETE CASCADE |
| 카운트 월 | count_month | CHAR | 7 | Υ | N | N | YYYY/MM 형식 |); |

프로그램 특성 및 현장 업무 처리 특성상 폐기 수량 보다 빈도가 그나마 유용한 데이터로 판단

2.3. Class Diagram.... 패키지 요약 설명1

loginPack

로그인화면, 계정등록 및 찾기 화면 구현 클래스 모음 (프로그램 시작점 main)

mainPack

메인 메뉴 및 홈 화면을 구현한 클래스 모음

mainResultPanel

메인 메뉴의 세부 기능 패널들을 구현한 클래스 모음

connPack

DB 연결 및 쿼리문 실행, 읽은 DATA값을 임시로 저장하는 클래스들의 모음

2.3. Class Diagram.... 패키지 요약 설명2

popUpPack

경고 및 확인 등 팝업 창 구현을 위한 클래스 모음

mainPopUp

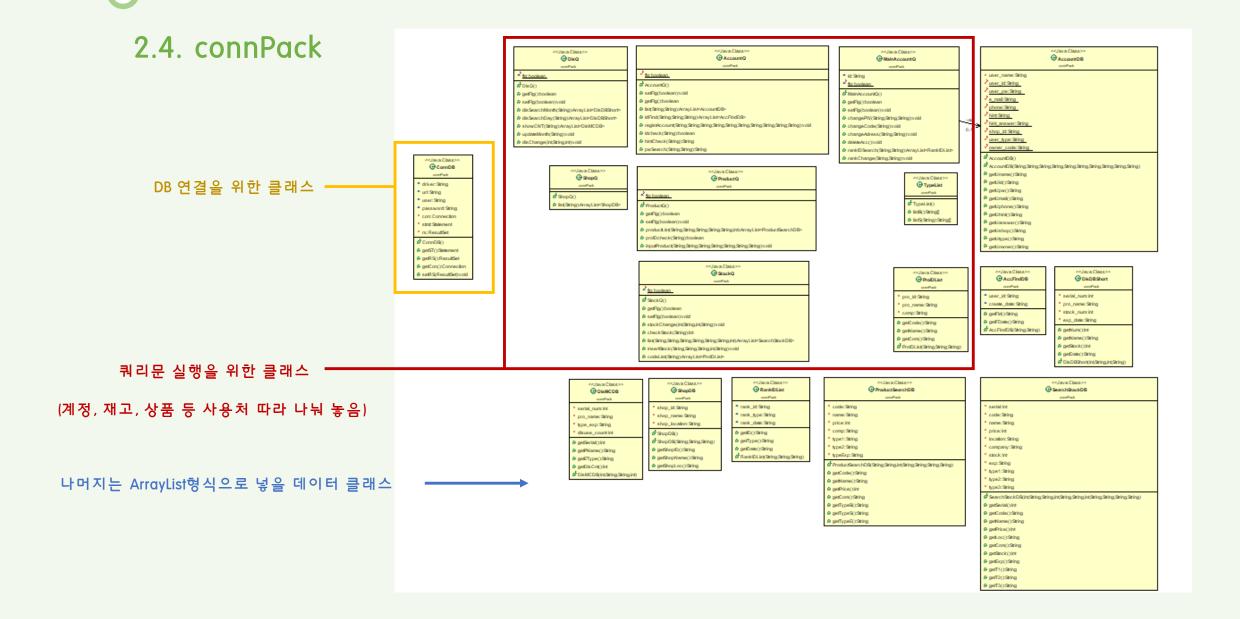
메인메뉴 세부 기능 중 재고 및 상품 리스트를 띄우는 팝업 창 구현을 위한 클래스 모음

toolPack

이미지 수정, 마우스 이동 등 자주 사용하는 기능을 구현한 메소드들의 클래스 모음

• 기타 (loginImage, mainImage, warningImage 등)

화면에 구현될 배경, 아이콘, 버튼 이미지 등을 저장 및 불러오기 위한 패키지



2.4. toolPack

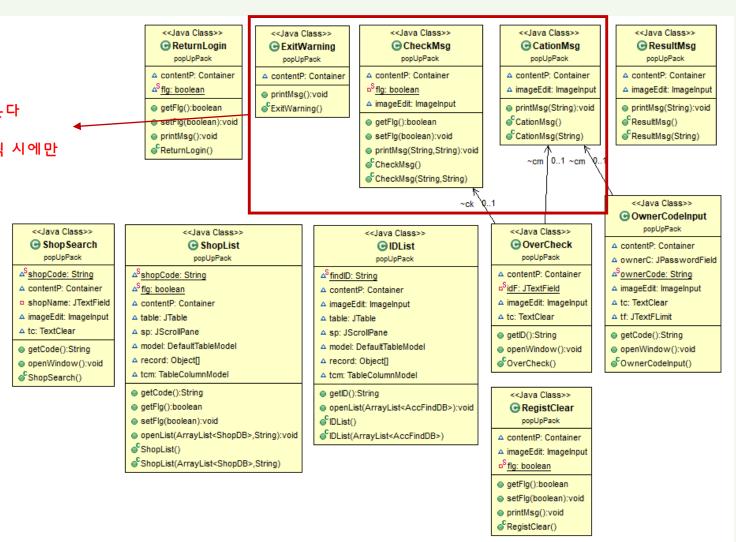
관련 메소드들을 유기적으로 모아 놓음 <<Java Class>> <<Java Class>> <<Java Class>> <<Java Class>> ● DragMoveDialog ImageInput PersonalComboBoxUI toolPack toolPack toolPack toolPack △ xDrag: int △ background: Color TextClear() o^ClmageInput() △ yDrag: int △ selectground: Color imageSetsize(ImageIcon,int,int):ImageIcon textClean():JTextField △ xPress: int passClean():JPasswordField PersonalComboBoxUI() setOnelmage(JButton,lmagelcon):void △ yPress: int setButtonImage(JToggleButton,Imagelcon,Imagelcon):void setColorBack(Color,Color):void <<Java Class>> DragMoveDialog() setButtonImage(JButton,Imagelcon,Imagelcon):void ScreateUI(JComponent):ComponentUI mouseDragged(MouseEvent):void buttonOpaque(JToggleButton):void installUI(JComponent):void toolPack mouseMoved(MouseEvent):void buttonOpaque(JButton):void o createArrowButton():JButton △ currentValue: int. setPaintLabel(JLabel,Imagelcon):void createRenderer():ListCellRenderer <<Java Class>> □ limit: int setHint(JLabel):void <<Java Class>> IntegerDocument(int) panelPaint(Imagelcon):JPanel toolPack PersonalComboBoxRenderer textPaint(ImageIcon):JTextArea qetValue():int toolPack △ xDrag: int tableSet(JTable,JScrollPane):void insertString(int,String,AttributeSet):void △ background: Color △ yDrag: int remove(int,int):void △ selectground: Color <<Java Class>> △ xPress: int <<Java Class>> checkInput(String,int):int O Date Tool SizeTool △ yPress: int PersonalComboBoxRenderer() toolPack toolPack <<Java Class>> MouseDragMove() setColorBack(Color,Color):void **⊙** JTextFLimit getListCellRendererComponent(JLi... DateTool() SizeTool() mouseDragged(MouseEvent):void toolPack mouseMoved(MouseEvent):void vildationDate(String):boo.. setCenterX(int):int □ limit: int setCenterY(int):int JTextFLimit(int) insertString(int,String,AttributeSet):void

만들어 놓고 가장 자주 인스턴스를 생성한 클래스, 이미지 사이즈 조정, 버튼 및 라벨 종류에 따른 이미지 처리

2.4. popPack

가장 자주 사용된 팝업(Dialog) 클래스, 경고, 처리 결과 문구를 문자열 파라미터로 받아 띄운다

CheckMsg 팝업의 경우 위 기능에 더불어 확인 버튼을 클릭 시에만 기능을 수행할 수 있게 하는 확인 팝업 창



2.4. loginPack

프로그램 시작점 (main 메소드 지점)

- △ contentP: Container
- □ fieldID: JTextField
- □ fieldPW: JPasswordField
- △ imageEdit: ImageInput
- △ tc: TextClear
- △ cm: CationMsg
- △ loginC: AccountQ
- Smain(String[]):void
- openWindow():void
- CLoginHome()

<<Java Class>>

- △ contentP: Container
- nameF: JTextField
- □ idF: JTextField
- pwF: JPasswordField
- checkPwF: JPasswordField
- hintF: JTextField
- hAnswerF: JTextField
- phoneF1: JTextField
- phoneF2: JTextField
- phoneF3: JTextField
- mailF1: JTextField
-
- mailF2: JTextField
- rankF: String
- a shopF: JTextField
- △ checkinfo2: JLabel
- △ rankS1: JRadioButton
- △ rankS2: JRadioButton
- △ rankS3: JRadioButton
- △ passflg: boolean
- △ tc: TextClear
- △ imageEdit: ImageInput
- △ cm: CationMsg
- △ ck: CheckMsq
- △ regAcc: AccountQ
- openWindow():void
- settextQ(JLabel):void
- setRadio(JRadioButton):void
- tfsetting(JTextField):JTextField

loginPack

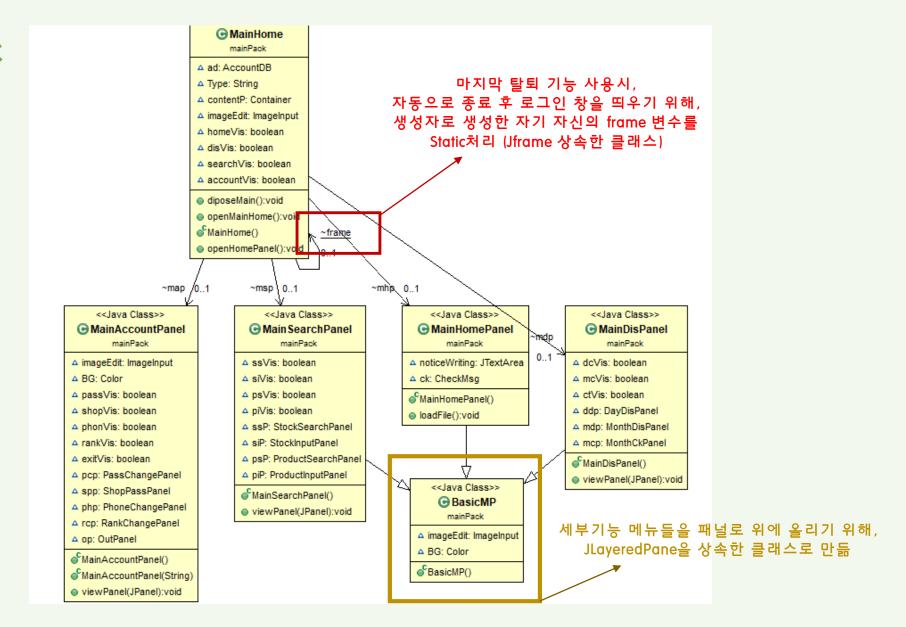
- △ contentP: Container
- △ imageEdit: ImageInput
- △ tc: TextClear
- △ ec: AccountQ
- △ cm: CationMsg
- △ rm: ResultMsg
- nameF: JTextField
- phoneF1: JTextField
- phoneF2: JTextField
- phoneF3: JTextField
- shopF: JTextField
- idF: JTextField
- hintF: JTextField
- hintAF: JTextField
- textField: JTextField
- openWindow():void
- CLoginFind()

<<Java Class>> Radio Select

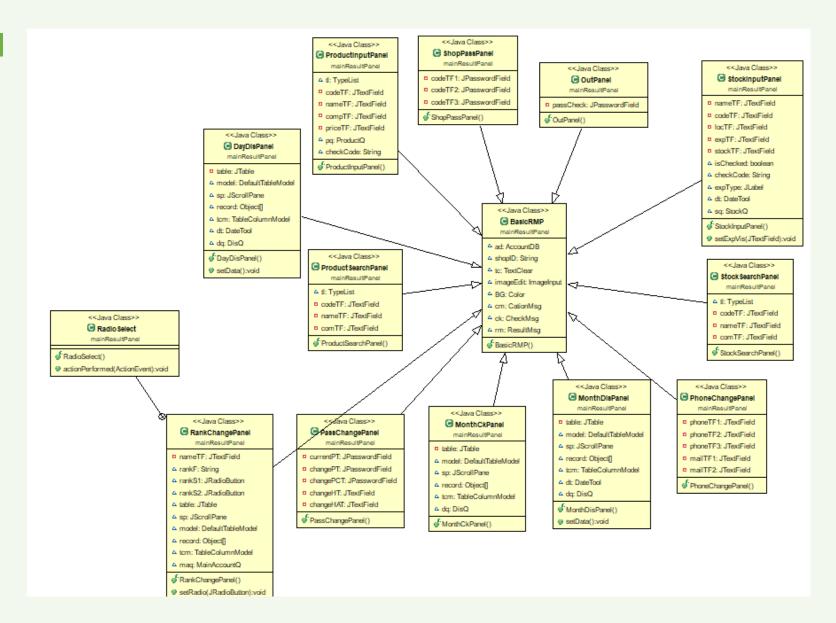
loginPack

- ©RadioSelect()
- actionPerformed(ActionEvent):void

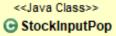
2.4. mainPack



2.4. mainResultPanel



2.4. mainPopUp



mainPopUp

- △ScodeP: String
- △ contentP: Container
- △ table: JTable
- △ sp: JScrollPane
- △ model: DefaultTableModel
- △ record: Object[]
- △ tcm: TableColumnModel
- △ sud: StockQ
- getCodeP():String
- openList(ArrayList<ProIDList>):void
- StockInputPop()
- StockInputPop(ArrayList<ProIDList>)

<<Java Class>>

⊕ DisLongPop

mainPopUp

- △ contentP: Container
- △ table: JTable
- △ sp: JScrollPane
- △ model: DefaultTableModel
- △ record: Object[]
- △ tcm: TableColumnModel
- △ dt: DateTool
- △ cm: CationMsg
- △ ck: CheckMsg
- △ rm: ResultMsg
- △ sud: StockQ
- △ imageEdit: ImageInput
- openList(ArrayList<DisDB>):void
- o^CDisLongPop()

<<Java Class>> ProductSearchPop

mainPopUp

- △ contentP: Container
- △ table: JTable
- △ sp: JScrollPane
- △ model: DefaultTableModel
- △ record: Object[]
- △ tcm: TableColumnModel
- openList(ArrayList<ProductSearchDB>):void
- ProductSearchPop()
- FroductSearchPop(ArrayList<ProductSearchDB>)

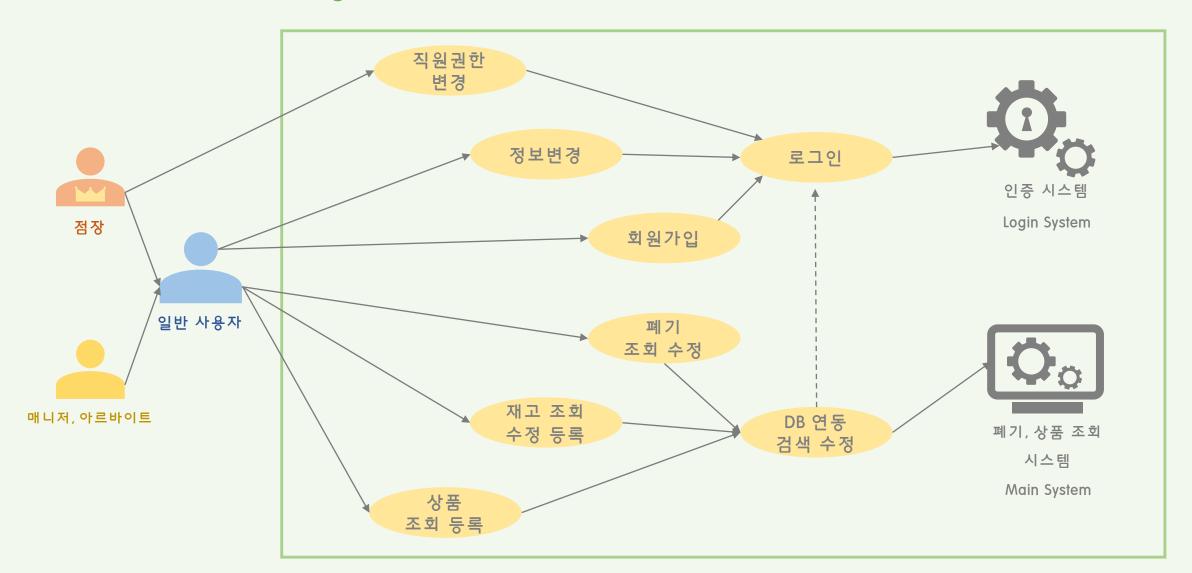
<<Java Class>>

Stock SearchPop

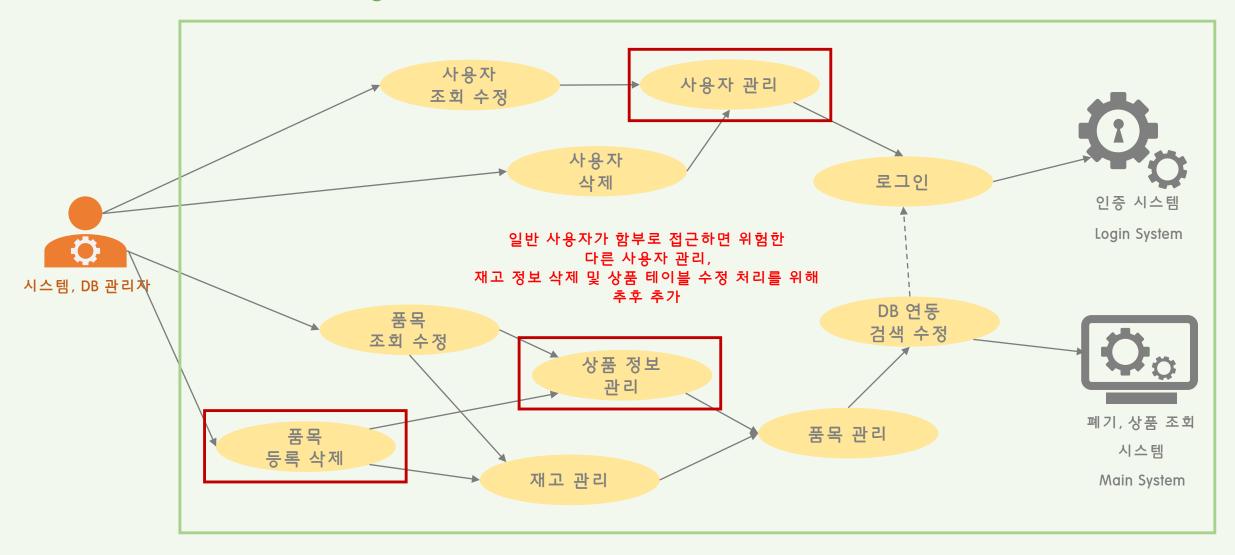
mainPopUp

- △ contentP: Container
- △ table: JTable
- △ sp: JScrollPane
- △ model: DefaultTableModel
- △ record: Object
- △ tcm: TableColumnModel
- △ dt: DateTool
- △ cm: CationMsg
- △ ck: CheckMsg
- △ sud: StockQ
- △ imageEdit: ImageInput
- openList(ArrayList<SearchStockDB>):void
- StockSearchPop()
- ©StockSearchPop(ArrayList<SearchStockDB>)

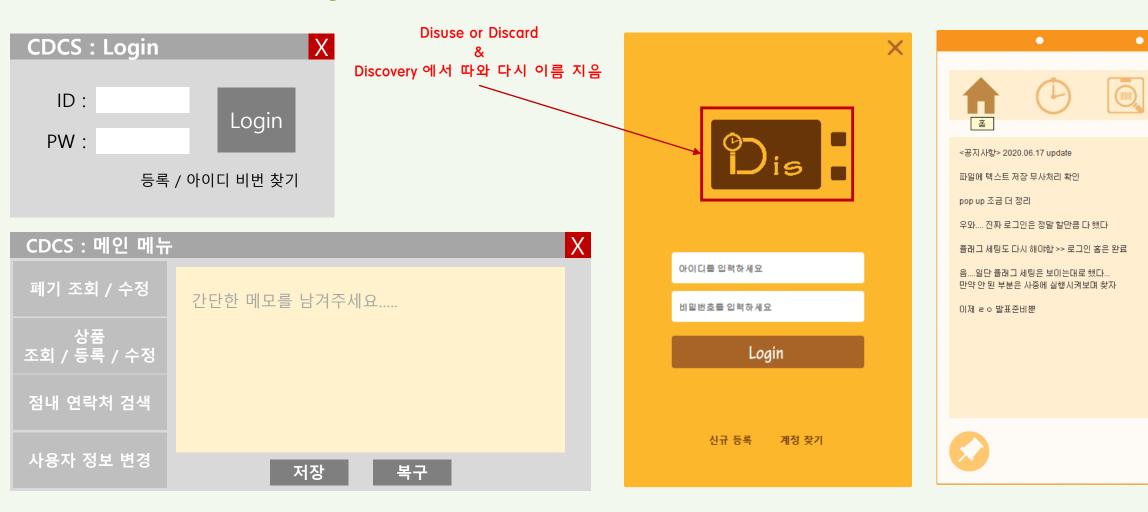
2.5. Use Case Diagram



2.5. Use Case Diagram (시스템 관리자 계정 : 미구현)



3.0. 디자인 변경



⟨before⟩

<after>

3.1. 로그인화면



최종 주요 UI정의서

3.2. 계정 등록 & 찾기





ID 텍스트 박스는 직접 입력 불가, 중복체크 팝업 테스트를 통과해 사용 결정을 하면, 자동으로 입력된다





3.2. 계정 등록 & 찾기



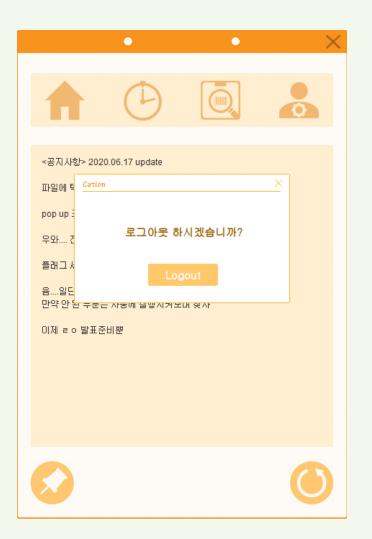


같은 사람이 같은 연락처와 점포코드로 여러 계정을 만들 경우, 필요한 아이디가 어떤 것인지 구분하기 위해 기준 추가

최종 주요 UI정의서

3.3. 메인 홈





3.4. 메인 주요 기능1 - 폐기 관리

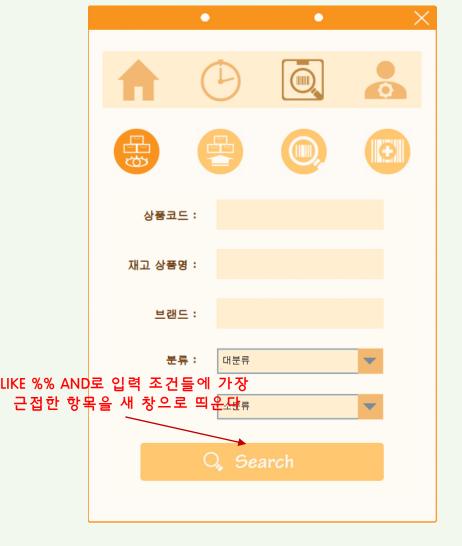


이 프로그램에서 거의 유일하게 직접 수정이 가능한 테이블 컬럼 종류 4가지 저장 아이콘을 누르면 1열에서 체크된 행의 값들만 DB에 업데이트 된다



최종 주요 UI정의서

3.4. 메인 주요 기능 2 - 재고, 상품 검색 및 등록







3.4. 메인 기능2 - 팝업 예시

| Ť | No. | 상품코드 | 상품명 | 가격 | 진열위치 | 재고 | 유통기한 | 브랜드 | 대분류 | 소분류 | 저장 |
|---|-----|---------------|------------------|------|-------|----|------------|-------------|-------------|-----------|----|
| ľ | 58 | 8801019306495 | 초코홈런볼46g | 1300 | 과자선반 | 3 | 2020/10/12 | 하타제과 | 과자/시리얼/안주/빵 | 콘/초코 | 장기 |
| t | 1 | 8801043036597 | 양파링84q | 1300 | | 4 | 2020/07/28 | (주)농심 | 과자/시리얼/안주/빵 | 일반스낵 | 장기 |
| t | 2 | 8801043036719 | 조청유과96g | 1300 | | 5 | 2020/08/22 | (주)농심 | 과자/시리얼/안주/빵 | 일반스낵 | 장기 |
| t | 3 | 8801117760106 | 포카칩소금맛66g | 1300 | | 3 | 2020/06/23 | (주)오리온특수영업부 | 과자/시리얼/안주/빵 | 감자스낵 | 장기 |
| t | 4 | 8801117760304 | 포카칩양파맛66g | 1300 | 과자선반B | 3 | 2020/06/25 | (주)오리온특수영업부 | 과자/시리얼/안주/빵 | 감자스낵 | 장기 |
| t | 5 | 8801117775001 | 스윙칩볶음고추장60g | 1300 | 과자선반B | 4 | 2020/12/11 | (주)오리온특수영업부 | 과자/시리얼/안주/빵 | 감자스낵 | 장기 |
| t | 6 | 8801019605000 | 자가비짭짤한맛90g | 3200 | 과자선반B | 2 | 2020/12/11 | 해태제과 | 과자/시리얼/안주/빵 | 감자스낵 | 장기 |
| t | 7 | 8801062012701 | 초코빼빼로54g | 1200 | 과자선반B | 1 | 2020/06/19 | 롯데제과(주) | 과자/시리얼/안주/빵 | 쿈 / 초코 | 장기 |
| Ť | 8 | 8801062012725 | 아몬드빼빼로37g | 1200 | 과자선반A | 3 | 2020/10/11 | 롯데제과(주) | 과자/시리얼/안주/빵 | 콘/초코 | 장기 |
| Ť | 9 | 8801068086003 | 정통크림빵70g | 1300 | 빵 선반 | 3 | 2020/06/25 | 삼립식품 | 과자/시리얼/안주/빵 | 빵/제과 | 00 |
| Ī | 10 | 8801068385328 | 펭수터질듯한땅콩미니샌드100G | 1500 | 빵 선반 | 1 | 2020/06/19 | 삼립식품 | 과자/시리얼/안주/빵 | 빵/제과 | 00 |
| T | 12 | 8801068385304 | 펭수러블리우유롤롤170g | 1500 | 빵 선반 | 3 | 2020/06/27 | 삼립식품 | 과자/시리얼/안주/빵 | 빵/제과 | 00 |
| Ī | 13 | 8801073210363 | 큰컵붉닭볶음면105g | 1300 | 라면선반A | 4 | 2021/01/11 | 삼양식품 | 라면/면류 | 컵라면 | 장기 |
| Ī | 14 | 8801128503211 | 왕뚜껑용기115g | 1100 | 라면선반A | 4 | 2020/10/22 | (주)팔도 | 라면/면류 | 컵라면 | 장기 |
| I | 15 | 8801043016070 | 튀김우동큰사발111g | 1150 | 라면선반A | 5 | 2020/06/28 | (주)농심 | 라면/면류 | 컵라면 | 장기 |
| Ι | 16 | 8801043016049 | 짜파게티큰사발123g | 1150 | 라면선반A | 3 | 2020/12/12 | (주)농심 | 라면/면류 | 컵라면 | 장기 |
| I | 17 | 8801043015967 | 새우탕큰사발115g | 1150 | 라면선반B | 2 | 2021/01/28 | (주)농심 | 라면/면류 | 컵라면 | 장기 |
| I | 18 | 8801043015745 | 오징머짬뽕컵67g | 900 | 라면선반B | 5 | 2021/01/04 | (주)농심 | 라면/면류 | 컵라면 | 장기 |
| П | 19 | 8801062278053 | 가나초코바아몬드45g | 1000 | 초콜릿/사 | 10 | 2020/10/11 | 롯데제과(주) | 과자/시리얼/안주/빵 | 초콜릿 / 초코바 | 장기 |



재고가 아닌 상품 정보 리스트는 일반 사용자가 수정하지 못한다



3.4. 메인 기능 - 사용자 계정 관리

| • | • | X | | • | • | X |
|------------------|-------|------|---|------------------|----|------|
| A (| | 0 | | A (| | 0 |
| | | Exit | | | | Exit |
| 기존 PW : | | | | 기존 PW : | | |
| 변경 할 PW : | | | 일반사용자의 직급에 따라 추가되는 기능이 있음 (추후 더 추가예정) | 변경 할 PW : | | |
| 변경할 PW 확인 : | | | | 변경랄 PW 확인 : | | |
| 변경칼 Hint : | 좋아하는것 | | | 변경활 Hint : | 숫자 | |
| 변경할 Hint 답: | 잠 | | | 변경할 Hint 답: | 순서 | |
| | | | | | | |

최종 주요 UI정의서

3.4. 메인 기능 - 사용자 계정 관리

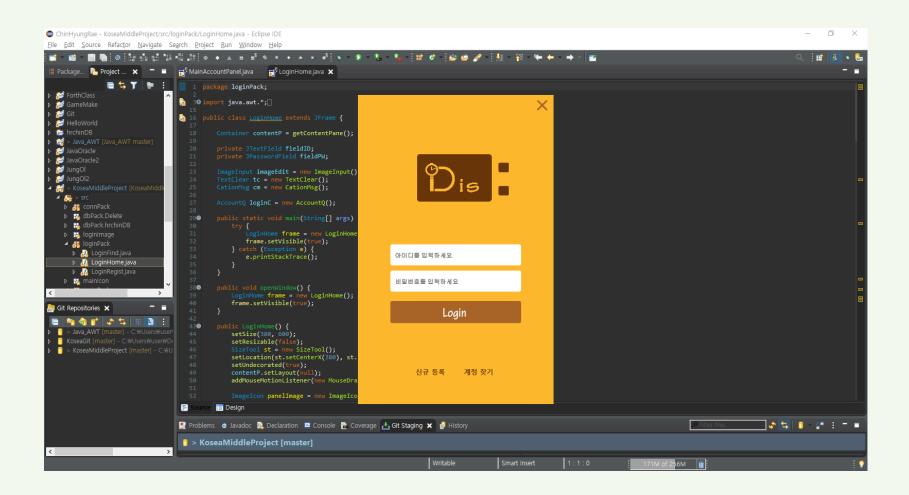


3.4. 메인 기능 - 사용자 계정(점장 계정)





프로그램 시연



Go!! →

개선할 점과 프로젝트 진행 소감

개선점



- 1. 관리자 계정 시스템 도입 (입장은 ID PW 필드에 커맨드 입력 후, 특정 아이콘 입력 시, 뜨는 회색 로그인 창을 통해)
- 2. 계정 생성 및 정보 변경 시, 아이디 비밀번호 메일 등 입력에 있어 제약조건 강화
- 3. 바코드 입력 시스템 도입??
- 4. 앞으로는 주석도 잘 챙길 수 있는 습관을 점점 들이자…… 등

개선할 점과 프로젝트 진행 소감

소감

- 1. 개인적인 일정이 겹치고 꼬이다 보니, 흔히 이야기 나오는 고통의 개발 일정의 편린을 살짝 느낄 수 있었음
- 2. 아직 실력이 부족해, 점점 갈수록 정신없이 코드를 짜다 보니,
 - 시스템 효율을 고려하거나 검색한 코드들을 다시 공부할 타이밍 부족해져서 아쉬웠음
- 3. GUI환경을 설계하다 보니 결과물이 만들어지는 과정이 눈에 보여 즐겁게 작업을 할 수 있었음
- 4. 이론적으로 이해해도 체감이 잘 안되던 부분들이 실질적으로 다가오는 경우가 있었음 (특히 static)
- 5. 작지만 스스로의 성장 피드백이 나름대로 되는 느낌이 있었음 (저번 주의 나OO는 왜 이리 짜 놨냐…. 하아….)

Git Hub 주소

https://github.com/chinmichael/KoseaMiddleProject.git

Thank you for Listening!!



네이버 웹툰 천리마 마트 그렌절