**Restaurant Project**

팀원 : 안해민

1. 프로젝트 개요
   1. 음식점에서 손님이 주문하고 사장님이 주문을 확인하고 받는 과정을 웹문서화 시킨것이다.
2. 프로젝트 기능
   1. 사장님 입장에서의 기능
      1. 회원가입
      2. 로그인
      3. 음식메뉴판
         1. 요리 정보보기
            1. 판매량에 따른 순이익과 총매출 확인

매출 초기화 기능

* + - * 1. 조건에 따른 요리정보검색

이름으로 검색하여 정렬

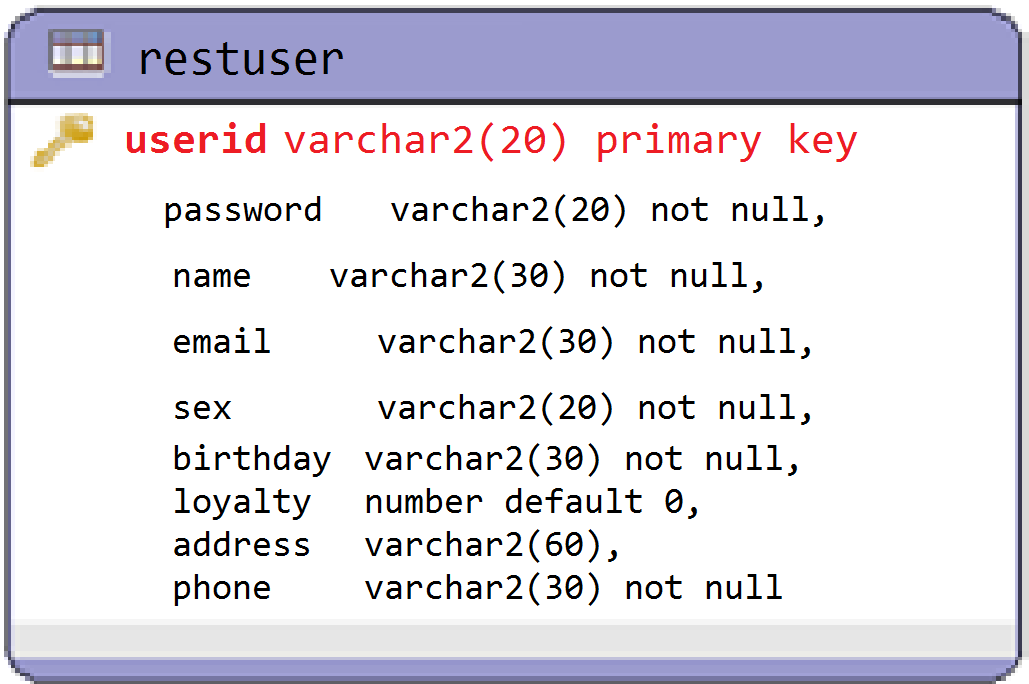
검색가격 이하 정렬

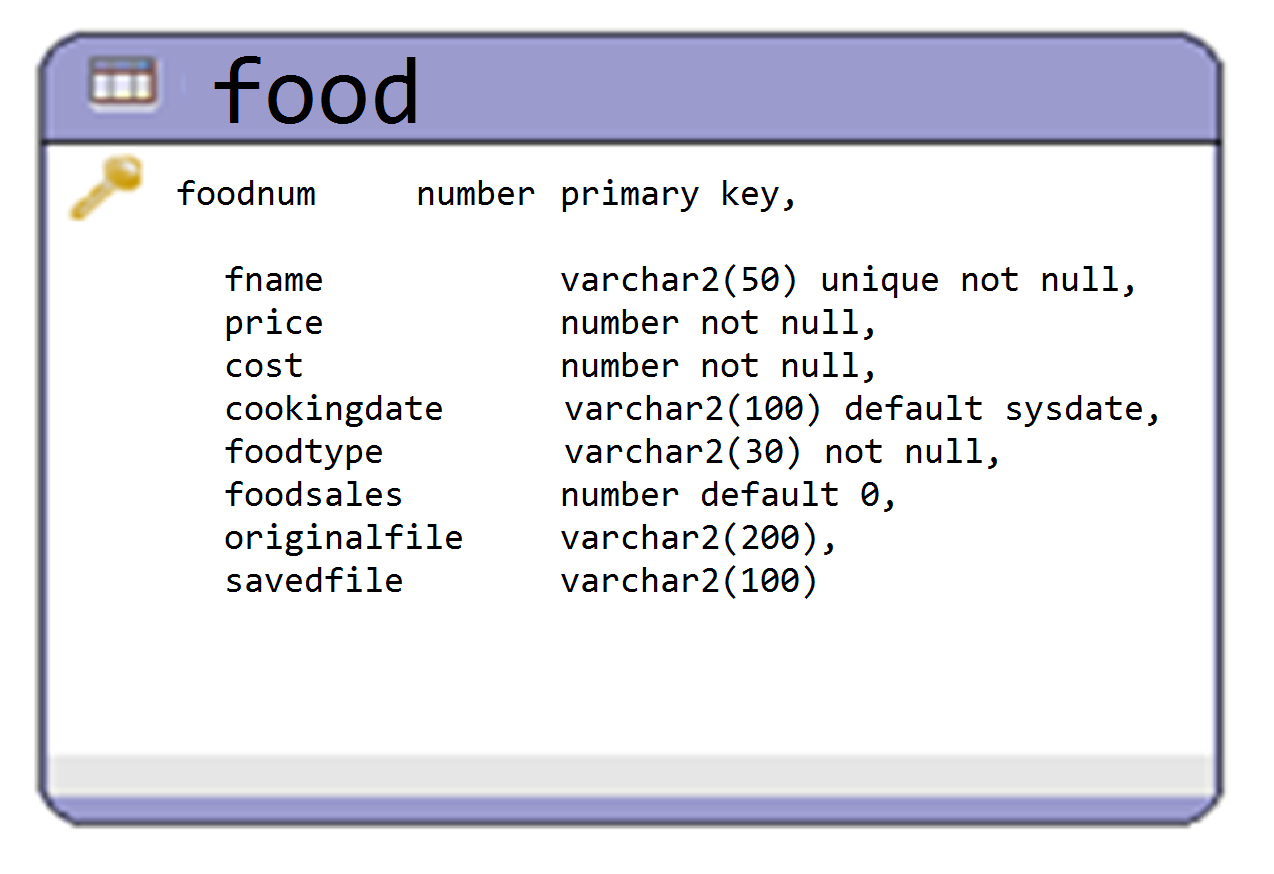
음식 종류에 따른 정렬

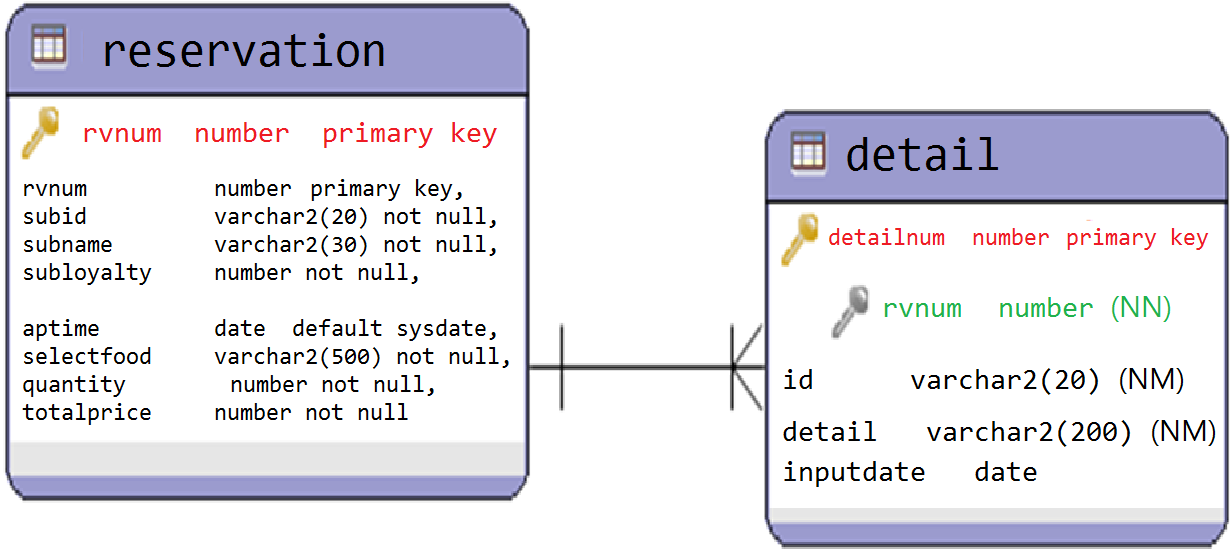
판매량 내림차순 정렬

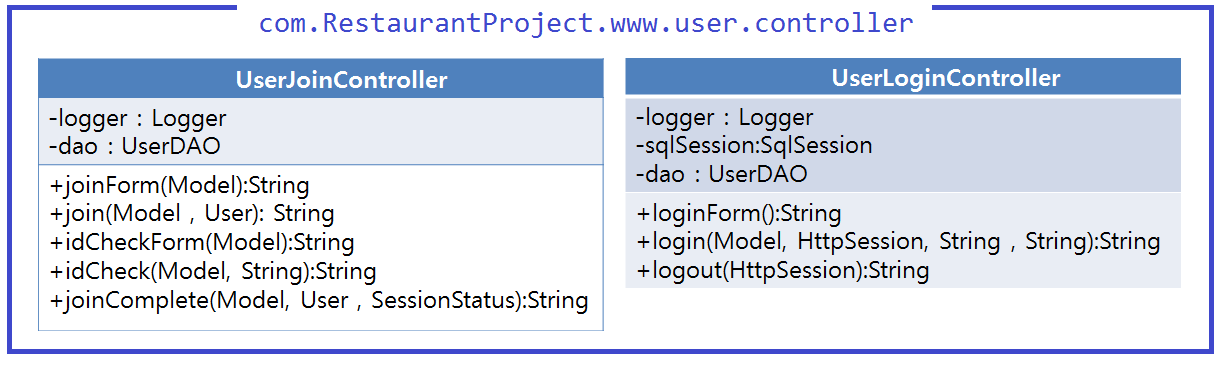
* + - 1. 요리 추가하기
         1. 음식이름 중복확인 기능
         2. 음식물 사진 등록 기능
      2. 요리 수정하기
         1. 음식물 정보 수정 기능
         2. 음식물 사진 수정 기능
      3. 요리 삭제하기
    1. 주문정보 확인하기
       1. 주문정보 검색
          1. 주문자(손님) ID 검색하기 (해당하는 ID의 주문정보 검색)
          2. 주문자(손님)이름 검색하기(해당 이름의 주문정보 검색)
       2. 음식메뉴판 이동하기
       3. 손님의 구체적 요구사항 보기
          1. 손님이 댓글로 작성한 구체적 요구사항 확인가능
       4. 주문 접수하기
          1. 주문접수시 손님의 주문량 만큼 해당 음식의 판매량 증가와 증가에 따른 매출액을 메뉴판에서 확인가능
  1. 손님(고객) 입장에서의 기능
     1. 주문하기
        1. 선택한 음식의 주문수량 정하는 기능
        2. 음식정보 검색기능(이름,가격,음식종류)
        3. 주문시 주문에 특이사항 기재 (댓글)
     2. 주문 취소하기

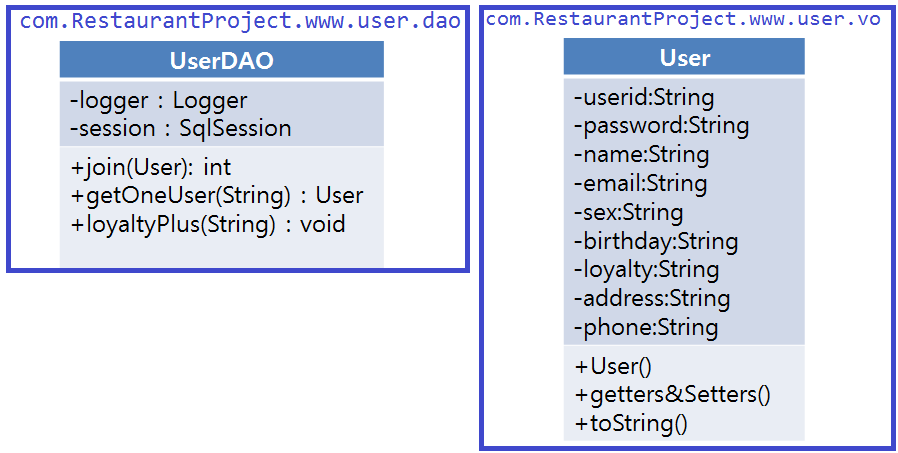
1. ER-Diagram







1. ClassDiagram
2. 



1. DB구조
   1. 사용자 정보 테이블 (테이블명 : restuser)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 컬럼명 | 자료형 | 크기 | 제약조건 | 설명 |
| custid | varchar2 | 20 | primary key | 사용자 아이디 |
| password | varchar2 | 20 | not null | 비밀번호 |
| name | varchar2 | 30 | not null | 사용자 이름 |
| email | varchar2 | 30 | not null | 이메일 |
| sex | varchar2 | 20 | not null | 사용자 성별 |
| birthday | varchar2 | 30 | not null | 사용자 생일 |
| loyalty | number |  | Default 0 | 사용자 주문횟수(고객충성도) |
| Address | varchar2 | 60 |  | 사용자 주소 |
| Phone | varchar2 | 30 | not null | 사용자 휴대전화 번호 |

* 1. 음식물 정보 테이블 (테이블명 : food)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 컬럼명 | 자료형 | 크기 | 제약조건 | 설명 |
| foodnum | number |  | primary key | 음식물 번호 |
| fname | varchar2 | 50 | not null | 음식물 이름 |
| price | number |  | not null | 음식물 가격 |
| cost | number |  | not null | 음식물 원가 |
| cookingdate | date |  | default | 음식물 등록일자 |
| foodtype | varchar2 | 30 | default | 음식물 스타일 |
| originalfile | varchar2 | 200 |  | 첨부파일명 (원래 이름) |
| savedfile | varchar2 | 100 |  | 첨부파일명 (실제 저장된 이름) |

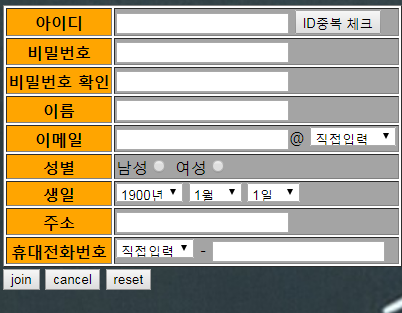
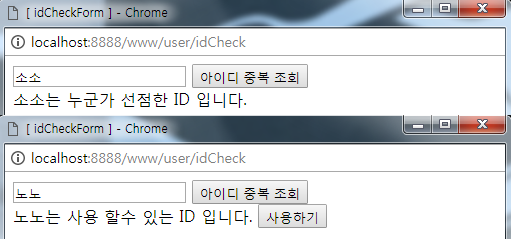
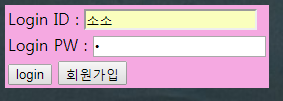
* 1. 음식물 테이블에 사용할 시퀀스 : food\_seq
  2. 주문 정보 테이블 (테이블명 : reservation)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 컬럼명 | 자료형 | 크기 | 제약조건 | 설명 |
| rvnum | number |  | primary key | 주문번호 |
| Subid | varchar2 | 20 | not null | 주문자 아이디 |
| Subname | varchar2 | 30 | not null | 주문자 이름 |
| Subloyalty | number |  |  | 주문자 주문횟수(고객충성도) |
| aptime | date |  | default | 주문한 시간 |
| selectfood | varchar2 | 500 | not null | 주문된 음식 |
| quantity | number |  | not null | 주문 수량 |
| totalprice | number |  | not null | 주문된 음식들의 가격 ( 음식수량 \* 음식가격 ) |

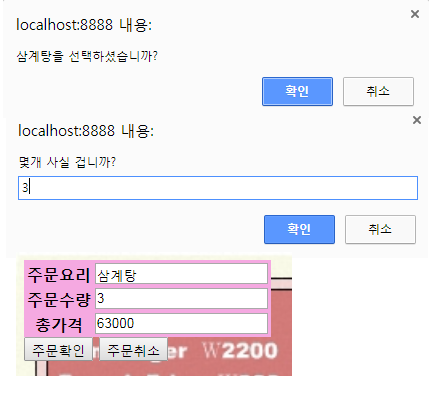
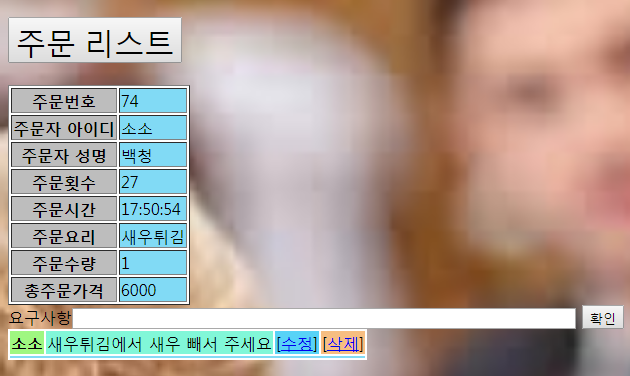
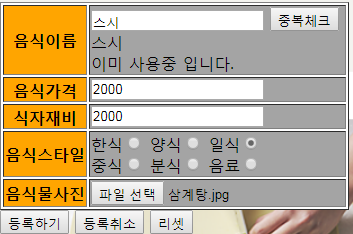
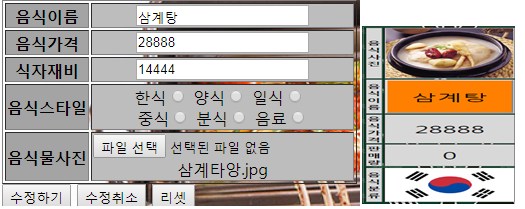
* 1. 주문 테이블에 사용할 시퀀스 : rvnum\_seq
  2. 고객의 요구 테이블 (테이블명 : detail)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 컬럼명 | 자료형 | 크기 | 제약조건 | 설명 |
| Detailnum | number |  | primary key | 요구글 번호 |
| Rvnum | number |  | foreign key  not null | 주문글 번호 |
| Id | varchar2 | 20 | not null | 요구글 작성자 Id |
| Detail | varchar2 | 200 | not null | 요구내용 |
| inputdate | date |  | default | 작성날짜 |

* 1. 고객 요구에 사용할 시퀀스 : detail\_seq

1. 동작 화면
   1. 회원가입 
   2. 아이디 중복 체크 
   3. 로그인 기능 
   4. 로그인 후 화면



* 1. 고객 음식물 주문 화면 
  2. 주문 버튼 클릭후 화면 
  3. 주문후 주문리스트 확인 
  4. 주문자 id로 주문내역 검색 
  5. 주문시 특이사항 입력하기 (댓글형태 입력) 
  6. 사장님 음식메뉴 관리 화면 
  7. 사장님 음식메뉴 추가하기 
  8. 사장님 음식물 정보 상세보기 & 정보 수정 & 정보 삭제 
  9. 음식물 정보 수정화면 / 수정후 
  10. 음식물 정보 검색(판매량 내림차순) 
  11. 음식 스타일 별 검색 ( 이미지 클릭 혹은 검색 ) 
  12. 총 매출 과 순이익 확인 그리고 매출액 초기화
  13. 손님의 주문을 접수하는 기능(접수 시 음식 판매량증가) 
  14. 주문접수후 매출액 변화 
  15. 장사 끝나면 매출 초기화 

1. 느낀점
   1. Ajax 자유자재로 쓸 수 있는 능력의 부족
   2. 데이터의 이동을 다루는 능력의 부족
   3. 페이징과 자료 업로드의 이해 부족
   4. 사용자가 사장이냐 손님이냐에 따라 실제로 사용하는 user가 누구냐에 따른 사용자 편의성을 제공하는 것을 생각 해야한다. 실제로 만들고 실행하면서 불편함을 느끼는 경우가 있다.