

반려동물 예약 서비스

(PetsCare)

-DB-

Team Project

목차

- (1) Project Organization & User Requirement
- (2) Business Rule
- (3) ER Diagram
- (4) Logical Design
- (5) Physical Design & Data dictionary
- (6) SQL Statement: DDL
- (7) SQL Statement: DML
- (8) System manual

(1) Project Organization & User Requirement

- **웹사이트 소개**

- 이름: PetsCare
- 소개: 반려동물 관련 서비스 예약 사이트
- 목적: 반려동물을 키우는 가정이 늘어나는 시대에 맞춰 오프라인 매장에 직접 방문하지 않고도 온라인 웹 사이트를 통해 병원, 미용, 호텔 등을 예약하는 서비스를 지원

- **User Requirement**

1. 개인 고객

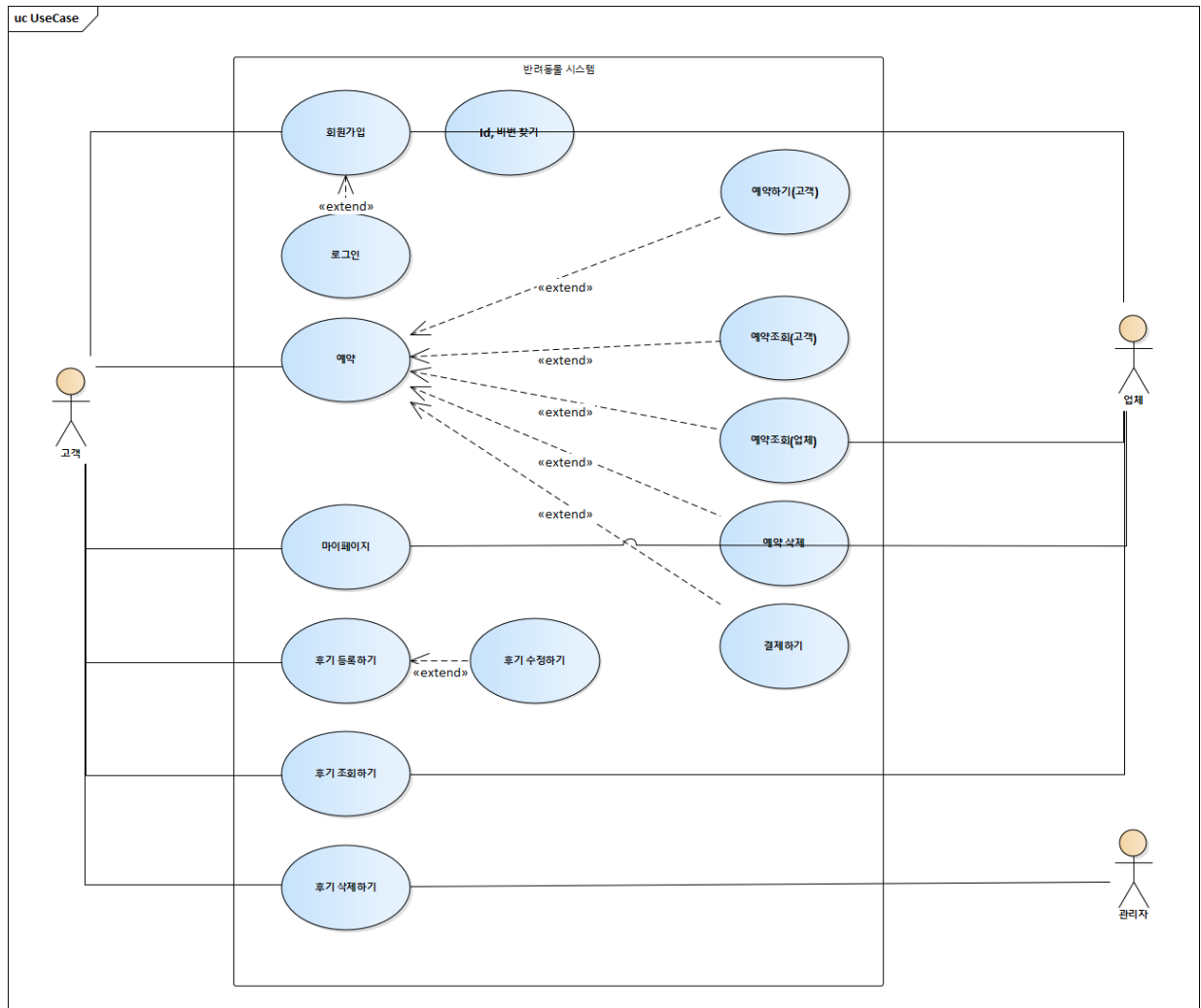
- 회원가입, 로그인을 거쳐 반려동물을 등록한다.
- 마이페이지에서 예약 현황을 조회한다.
- 마이페이지에서 예약, 예약취소를 진행한다.
- 이용한 업체에 대한 후기를 작성한다.
- 후기를 등록, 수정, 조회, 삭제한다.

2. 기업 고객

- 회원가입, 로그인을 거쳐 업체를 등록한다.
- 마이페이지에서 예약 현황을 조회한다.
- 고객의 후기를 조회하고, 답글을 작성한다.

3. 관리자

- 형식에 맞지 않거나 부적절한 후기를 삭제한다.



• User Requirement에 따른 DB 설계 목적

1. 개인 고객 관리

- 회원 정보 등록
- 반려동물 정보 등록
- 후기등록

2. 기업 고객 관리

- 기업 정보 등록
- 후기 조회 및 답글 등록

3. 예약 관리

- 예약 등록, 조회, 수정, 삭제 진행

(2) Business Rule

PetsCare 웹사이트는 반려동물을 키우는 가정이 늘어나는 시대에 맞춰 오프라인 매장에 직접 방문하지 않고도 사이트를 통해 병원, 미용, 호텔 등을 예약하는 서비스를 지원하는 온라인 웹사이트로 반려동물 관련 시설 예약을 편리하게 이용하는 것을 목적으로 한다.

개인 회원 테이블은 고유 번호, ID, PW, 이름, 주민등록번호, 전화번호, 주소, 이메일, 프로필 이미지를 가지고 있다.

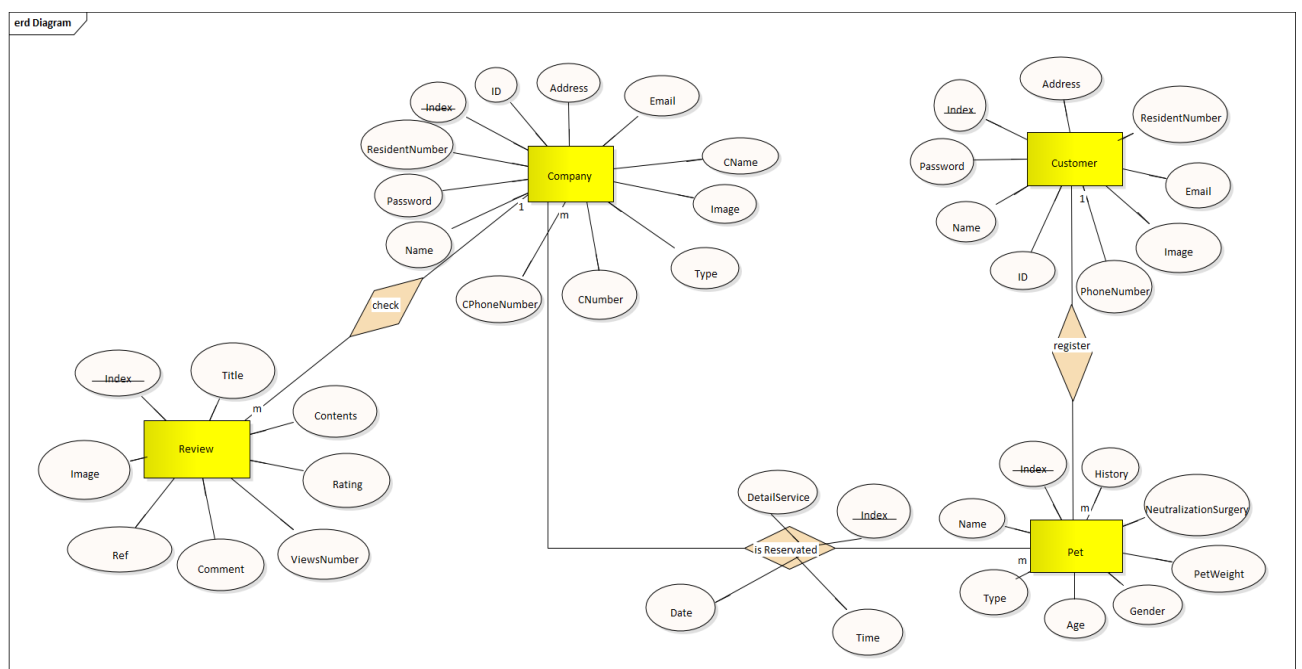
반려동물 테이블은 개인 회원에 의해 등록된다. 한 마리의 반려동물은 고유번호, 이름, 종, 나이, 성별, 몸무게, 중성화 수술 여부, 병력을 가지고 있으며 한 명의 고객은 여러 마리의 반려동물을 등록할 수 있지만 한 마리의 반려동물은 한 명의 고객에 의해 등록된다.

기업 테이블은 고유번호, ID, PW, 대표 이름, 대표 주민번호, 사업자 번호, 기업 이름, 기업 전화번호, 기업 주소, 기업 종류, 이메일, 기업 프로필 이미지를 가진다. 한 개의 기업은 여러 마리의 반려동물을 예약 받을 수 있고 한 마리의 반려 동물도 여러 개의 기업의 서비스를 예약할 수 있다.

후기 테이블은 고유번호, 제목, 내용, 평점, 조회수, 답글, 게시물 번호, 후기 이미지를 가지고 있으며 한 개의 후기는 하나의 기업에게만 확인되어 지지만 한 개의 기업은 여러 개의 후기를 확인할 수 있다.

예약 테이블은 고유번호, 날짜, 세부 서비스 내용, 날짜를 가지고 있다.

(3) ER Diagram



(4) Logical Design

1. Customer (customer_Index, customer_Id, customer_Password, customer_Name, customer_ResidentNumber, customer_PhoneNumber, customer_Address, customer_Email, customer_Image)

2. Pet (pet_Index, customer_Index, pet_Name, pet_Type, pet_Gender, pet_Age, pet_IsNeutralized, pet_Weight, pet_History)

Foreign key(customer_Index) references Customer(customer_Index)

3. Company (company_Index, company_Id, company_Password, company_UserName, company_ResidentNumber, company_Address, company_Email, company_UserPhoneNumber, company_Name, company_Number, company_Type, company_Image)

4. Reservation (reservation_Index, pet_Index, company_Index, reservation_Date, reservation_Time, reservation_DetailService)

Foreign key(pet_Index) references Pet(pet_Index)

Foreign key(company_Index) references Company(company_Index)

5. Review (review_Index, company_Index, review_Title, review_Content, review_Rating, review_ViewsNumber, review_Image, review_Comment, review_Ref)

Foreign key (company_Index) references Company (company_Index)

(5) Physical Design & Data dictionary

☞ Customer Table

Attribute	Description	Data Type	Constraint
<u>customer_Index</u>	개인 회원 고유번호	INT	NOT NULL Primary key , Auto increment
customer_Id	개인 회원 아이디	VARCHAR (12)	NOT NULL
customer_Password,	개인 회원 비밀번호	VARCHAR (12)	NOT NULL Unique
customer_Name	개인 회원 이름	VARCHAR (10)	NOT NULL
customer_ResidentNumber	개인 회원 주민번호	VARCHAR (13)	NOT NULL Unique
customer_PhoneNumber	개인 회원 전화번호	VARCHAR (13)	NOT NULL Unique
customer_Address	개인 회원 주소	VARCHAR (50)	NOT NULL
customer_Email	개인 회원 이메일	VARCHAR (80)	NOT NULL
customer_Image	개인 회원 프로필 사진	Medium blob	

☞ Company Table

Attribute	Description	Data Type	Constraint
<u>company_Index</u>	기업 고유번호	INT	NOT NULL Primary key , Auto increment
company_Id	기업 아이디	VARCHAR (12)	NOT NULL
company_Password	기업 비밀번호	VARCHAR (12)	NOT NULL
company_UserName	기업 회원 이름	VARCHAR (10)	NOT NULL
company_ResidentNumber	기업 회원 주민번호	VARCHAR (3)	NOT NULL Unique
company_Address	기업 주소	VARCHAR (13)	NOT NULL
company_Email	기업 이메일	VARCHAR (50)	NOT NULL
company_UserPhoneNumber	기업 회원 전화번호	VARCHAR (80)	NOT NULL
company_Name	기업 이름	VARCHAR (20)	NOT NULL
company_Number	사업자 번호	INT (15)	NOT NULL Unique
company_Type	기업 종류	VARCHAR (20)	NOT NULL
company_Image	기업 프로필 이미지	Medium Blob	

Pet Table

Attribute	Description	Data Type	Constraint
pet_Index	반려동물 고유번호	INT	NOT NULL Primary key , Auto increment
customer_Index	개인 회원 아이디	INT	NOT NULL Foreign key
pet_Name	반려동물 이름	VARCHAR (12)	NOT NULL
pet_Type	반려동물 종류	VARCHAR (15)	NOT NULL
pet_Gender	반려동물 성별	VARCHAR (3)	NOT NULL
pet_Age	반려동물 나이	INT (4)	NOT NULL
pet_IsNeutralized	반려동물 중성화 여부	BOOLEAN	NOT NULL
pet_Weight	반려동물 몸무게	INT	NOT NULL
pet_History	반려동물 병력	TEXT	

Reservation Table

Attribute	Description	Data Type	Constraint
<u>reservation_Index</u>	예약 고유번호	INT	NOT NULL Primary key , Auto increment
pet_Index	반려동물 고유번호	INT	NOT NULL Foreign key
company_Index	기업 고유번호	VARCHAR (12)	NOT NULL Foreign key
reservation_Date	예약 날짜	VARCHAR (12)	NOT NULL
reservation_Time	예약 시간	VARCHAR (10)	NOT NULL
reservation_DetailService	세부 서비스 정보	VARCHAR (13)	NOT NULL

Reservation Table

Attribute	Description	Data Type	Constraint
<u>review_Index</u>	후기 고유번호	INT	NOT NULL Primary key , Auto increment
company_Index	회사 고유번호	INT	NOT NULL Foreign key
review_Title	후기 제목	VARCHAR (60)	NOT NULL
review_Content	후기 내용	TEXT	NOT NULL
review_Rating	후기 평점	VARCHAR (15)	NOT NULL

review_ViewsNumber	후기 조회수	INT	NOT NULL
review_Image	후기 이미지	Medium Blob	
review_Comment	후기 답글	Text	
review_Ref	후기 게시물 번호	INT	

(6) SQL Statement: DDL

Customer Table

```
create table Customer (  
  
customer_Index int not null auto_increment,  
  
customer_Id varchar(12) not null,  
  
customer_Password varchar(12) not null,  
  
customer_Name varchar(10) not null,  
  
customer_ResidentNumber varchar(13)not null,  
  
customer_PhoneNumber varchar(13) not null,  
  
customer_Address varchar(50) not null,  
  
customer_Email varchar(80) not null,  
  
customer_Image mediumblob,  
  
primary key(customer_Index),  
  
unique key(customer_ResidentNumber),  
  
unique key(customer_PhoneNumber));
```

Company Table

```
create table Company(  
  
company_Index int not null auto_increment,  
  
company_Id varchar(12) not null,  
  
company_Password varchar(12) not null,  
  
company_UserName varchar(10) not null,  
  
company_ResidentNumber varchar(13)not null,  
  
company_Address varchar(50) not null,
```

```
company_Email varchar(80) not null,  
company_UserPhoneNumber varchar(13) not null,  
company_Name varchar(20) not null,  
company_Number int(15) not null,  
company_Type varchar(20)not null,  
company_Image mediumblob ,  
primary key(company_Index),  
unique key(Company_Number),  
unique key(company_ResidentNumber));
```

Pet Table

```
create table Pet(  
pet_Index int not null auto_increment,  
customer_Index int not null ,  
pet_Name varchar(12) not null,  
pet_Type varchar(15) not null,  
pet_Gender varchar(3) not null,  
pet_Age int(4) not null,  
pet_IsNeutralized boolean default null,  
pet_Weight int not null,  
pet_History text,  
primary key(pet_Index),  
constraint pet_fk foreign key(customer_Index)  
references Customer(customer_Index));
```

Reservation Table

```
create table Reservation(  
  
reservation_Index int not null auto_increment,  
  
pet_Index int not null,  
  
company_Index int not null,  
  
reservation_Date Date not null,  
  
reservation_Time time not null,  
  
reservation_DetailService text,  
  
primary key (reservation_Index),  
  
constraint reservation_fk foreign key(pet_Index)  
references Pet(pet_Index),  
  
constraint reservation_fk2 foreign key(company_Index)  
references Company(company_Index));
```

Review Table

```
create table Review (  
  
review_Index int not null auto_increment,  
  
company_Index int not null,  
  
review_Title varchar(60) not null,  
  
review_Content text not null,  
  
review_Rating varchar(15) not null,  
  
review_ViewsNumber int  
  
review_Image mediumblob,  
  
review_Comment text,  
  
review_Ref int,
```

primary key (review_Index),

constraint review_fk foreign key(company_Index)

references Company(company_Index));

(7) SQL Statement: DML

Insert records in Customer Table

```
Insert into Customer VALUES(1, 'cust', '113', '홍길동', '9609241345912', '01090195201', '서울성북구  
돈암동', 'kkamikoontest@naver.com', null);
```

Insert records in Company Table

```
Insert into Company VALUES(1, 'com', '113', '김철수', '9609241345912', '서울성북구돈암동',  
'kkamikoontest@naver.com', '01090195201', '행복병원', 32314122, '병원', null);
```

insert records in Pet Table

```
Insert into Pet VALUES(1, 1, '루미', '고양이', '수컷', 1, true, 3, null);
```

insert records in Reservation Table

```
Insert into Reservation VALUES(1,1,1, '2019-11-02','11:00:00', null);
```

insert records in Review Table

```
Insert into Review values(1,1,'안녕하세요','여기 병원 너무 좋아요','5점',3, null, null, null, null);
```

(8) System manual

Company Table

<List-기업회원(ID, PW 일치하는)>

```
select * from Company where company_Id = #{ company_Id } and company_Password =  
#{ company_Password }
```

<List-기업회원>

```
select * from Company where company_Index = #{company_Index}
```

<List-기업유형별>

```
select * from Company where company_Type = #{ company_Type }
```

<Delete>

```
delete from Company where company_Index = #{ company_Index }
```

Customer Table

<List-개인 회원(ID, PW 일치하는)>

```
select * from Customer where customer_Id = #{customer_Id} and customer_Password =  
#{customer_Password}
```

<List-개인 회원>

```
select * from Customer where customer_Index = #{ customer _Index}
```

<Delete>

```
delete from Custoemr where customer_Index = #{ customer _Index }
```

<Update>

```
update Customer set customer_Password = #{customer_Password}, customer_PhoneNumber =  
#{customer_PhoneNumber}, customer_Email = #{customer_Email}, customer_Address =  
#{customer_Address} where customer_Id = #{customer_Id} and customer_Name =  
#{customer_Name}
```

Pet Table

<List-반려동물>

```
select * from Pet where customer_Index = #{ customer_Index }
```

<Delete>

```
delete from Pet where pet_Index = #{ pet_Index }
```

<Update>

```
update Pet set pet_Name =#{pet_Name}, pet_Type=#{pet_Type}, pet_Gender=#{pet_Gender},  
pet_Age=#{pet_Age}, pet_IsNeutralized=#{pet_IsNeutralized}, pet_Weight=#{pet_Weight},  
pet_History=#{pet_History} where customer_Index=#{customer_Index} and pet_Index =  
#{pet_Index}
```

Reservation Table

<List-날짜 지난 예약>

```
select * from Reservation <![CDATA[ where #{todayDate} > reservation_Date
```

<List - 개인회원 예약>

```
Select * from reservation reservation left join pet pet on reservation.pet_Index = pet.pet_Index  
left join company company on reservation.company_Index = company.company_Index where  
reservation.pet_Index = #{ pet_Index }
```

<Delete>

```
delete from Reservation where reservation_Index = #{reservation_Index }
```

<Cancel>

```
update Reservation set reservation_Check = 'canceled' where reservation_Index =#  
{reservation_Index}
```

<Finish>

```
update Reservation set reservation_Check = 'finished' where reservation_Index =  
#{reservation_Index}
```


Review Table

<List>

Select * from Review where company_Index =#{company_Index}

<Delete>

delete from Review where review_Index = #{review_Index}

<Update>

update Review set review_Comment = #{review_Comment} where review_Index = #{review_Index}