요구사항 정의서

en·core | PLAYDATA

한화시스템

팀명 : Woof

조원 : 강문혜 강지흔 이창훈 임연진

한화시스템 BEYOND CAMP 2기 - 1st project

목차

1. 프로젝트 개요

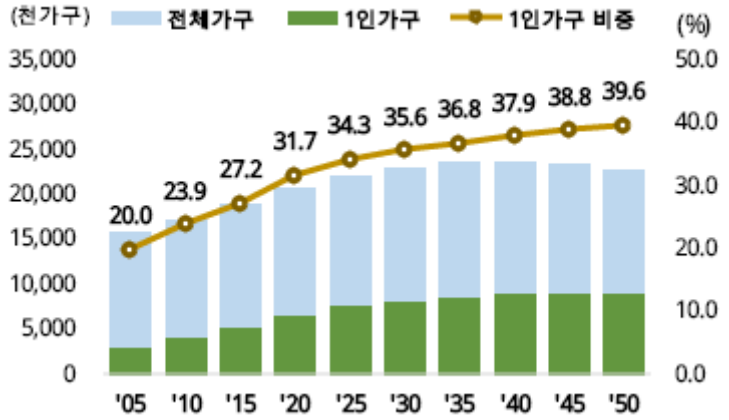
1.1 프로젝트 소개

'Woof' 서비스는 반려동물에게 필요한 업체를 검색하고 방문하는데 어려움을 겪는 사용자들을 위해 만들어진 서비스이다.

최근 꾸준히 반려동물 산업과 반려동물 인구가 증가됨과 동시에 사회적으로 소규모가구 또한 증가하면서 집에 반려동물이 혼자 남아있는 시간이 많거나, 사회성이 부족한 반려동물들에 대한 우려가 생겨나고 있다. 반려동물 유치원이나 병원, 미용실 등에 방문을 해야 하는데 거대해진 산업에 비해 사람이 다니는 학원/병원/미용실보다 훨씬 검색, 예약 등이 어렵고 방문할 시간이 없다는 의견이 많이 존재한다.

이를 해결하기 위하여 'Woof' 에서는 사용자 주변에 있는 반려동물 업체의 정보, 사진, 리뷰를 모아서 보여주고, 바쁜 사회인들을 위하여 전문 자격증을 보유한 매니저가 보호자 대신 반려동물과 업체에 대신 방문해주는 서비스를 제공한다.

1.2 프로젝트 배경



자료 = 통계청 (인구주택총조사, 장래가구추계:2020~2050)

전체 가구 중 1인 가구의 비중은 2005년 20%였으나, 2030년 35.6%, 2050년엔 39.6%에 이를 것으로 전망된다. 반려동물 인구의 증가 역시 늘어나고 있는데 이러면 비교적으로 반려동물 한 마리당 보호자의 수가 줄어들고 있다고 봐야한다.

경제활동이나 개인사정으로 인해 반려동물이 집에서 홀로 보내는 시간이 많다보니 가족과 함께하지 않는 동안에도 안전하게 충분한 자극을 즐길 수 있는 반려동물 유치원의 수요 역시 늘어나고 있다.

'Woof' 서비스는 유치원에 반려동물을 데리고 등원시킬 수 있는 여유가 없거나, 다양한 유치원의 프로그램과 양질의 관리, 가격 등을 비교하는 시간이 부족하다

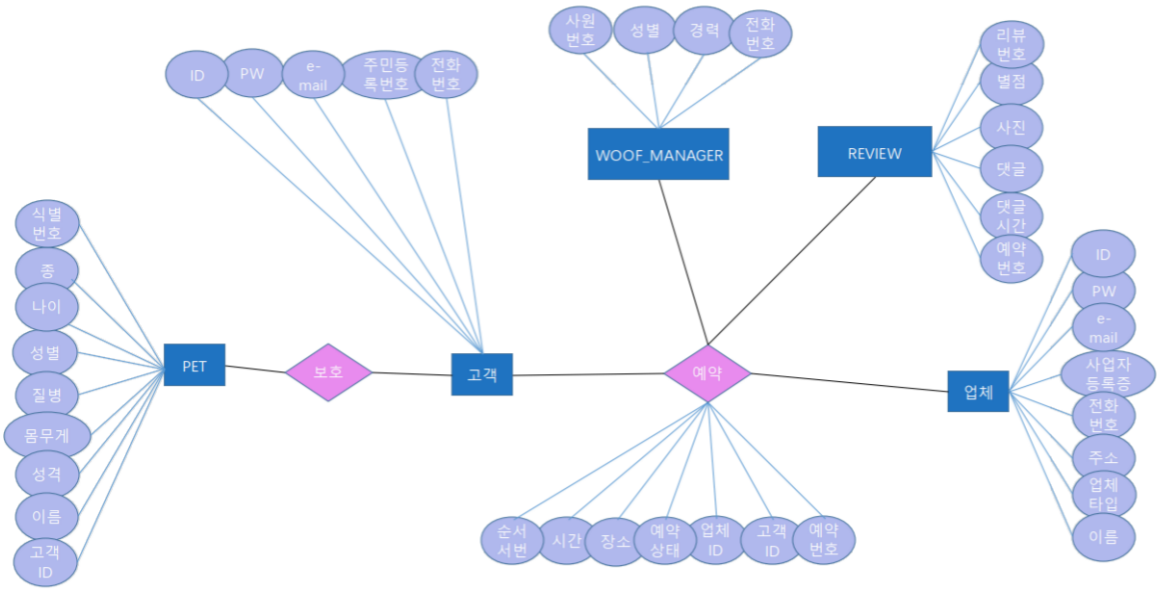
던가 하는 등 바쁜 사회인들을 위해 해결방안을 주기 위해 만들어졌다.

현재는 사용자 주변의 유치원, 병원, 미용실에 대한 정보를 제공하고, Woof 매니저를 통해 보호자 대신 원하는 업체로 반려동물을 픽업해주는 서비스를 제공하고 있고, 반려동물 사업이 확대됨에 따라 'Woof'의 서비스 역시 다양해질 것으로 기대되고 있다.

1.3업무 배경도(Context Diagram)

2. 설계

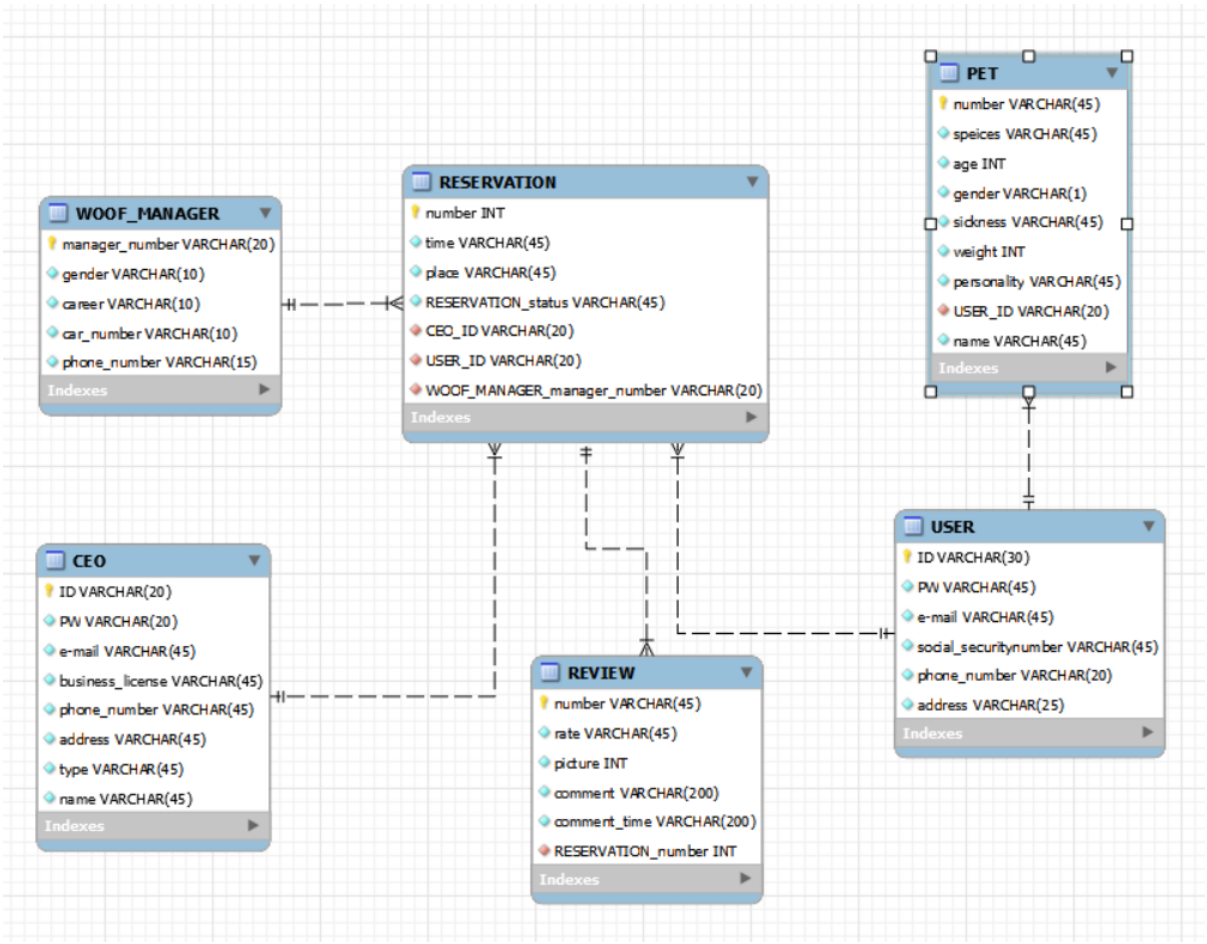
2.1 ERD



상세보기

테이블	속성	관계
업체	ID/ 비밀번호/ E-mail/ 사업자등록증/전화번호/ 주소/업체타입/이름	
REVIEW	리뷰 번호/ 별점/ 사진/ 댓글/ 댓글 시간/ 예약 번호	
WOOF_MANAGER	사원번호/ 성별/ 경력/ 전화 번호	
고객	고객ID/ 비밀번호/ E-mail/ 주민등록번호/ 전화번호	보호, 예약
반려동물	식별번호/ 종/ 나이/ 성별/ 질병/ 몸무게/ 성격/ 고객 ID	
	예약 번호/ 시간/ 장소/ 예약 상태/ 업체 ID/ 고객 ID / 예약번호	예약

2.2 릴레이션 스키마


WOOF_MANAGER
manager_number VARCHAR(20)
gender VARCHAR(10)
career VARCHAR(10)
car_number VARCHAR(10)
phone_number VARCHAR(15)
Indexes
RESERVATION
number INT
time VARCHAR(45)
place VARCHAR(45)
RESERVATION_status VARCHAR(45)
CEO_ID VARCHAR(20)
USER_ID VARCHAR(20)
WOOF_MANAGER_manager_number VARCHAR(20)
Indexes
PET
number VARCHAR(45)
speices VARCHAR(45)
age INT
gender VARCHAR(1)
sickness VARCHAR(45)
weight INT
personality VARCHAR(45)
USER_ID VARCHAR(20)
name VARCHAR(45)
Indexes
CEO
ID VARCHAR(20)
PW VARCHAR(20)
e-mail VARCHAR(45)
business_license VARCHAR(45)
phone_number VARCHAR(45)
address VARCHAR(45)
type VARCHAR(45)
name VARCHAR(45)
Indexes
REVIEW
number VARCHAR(45)
rate VARCHAR(45)
picture INT
comment VARCHAR(200)
comment_time VARCHAR(200)
RESERVATION_number INT
Indexes
USER
ID VARCHAR(30)
PW VARCHAR(45)
e-mail VARCHAR(45)
social_securitynumber VARCHAR(45)
phone_number VARCHAR(20)
address VARCHAR(25)
Indexes

3. 구현

3.1 릴레이션

테이블	릴레이션
CEO 업체	CREATE TABLE CEO (`ID` VARCHAR(20) NOT NULL, `PW` VARCHAR(20) NOT NULL, `e-mail` VARCHAR(45) NOT NULL, `business_license` VARCHAR(45) NOT NULL, `phone_number` VARCHAR(45) NOT NULL, `address` VARCHAR(45) NOT NULL, `type` VARCHAR(45) NOT NULL, `name` VARCHAR(45) NOT NULL, PRIMARY KEY (`ID`)) ENGINE = InnoDB;
USER 사용자	CREATE TABLE USER (`ID` VARCHAR(30) NOT NULL, `PW` VARCHAR(45) NOT NULL, `e-mail` VARCHAR(45) NOT NULL, `social_securitynumber` VARCHAR(45) NOT NULL, `phone_number` VARCHAR(20) NOT NULL, `address` VARCHAR(25) NOT NULL, PRIMARY KEY (`ID`)) ENGINE = InnoDB;
WOOF_MANAGER WOOF 매니저	CREATE TABLE WOOF_MANAGER (`manager_number` VARCHAR(20) NOT NULL, `gender` VARCHAR(10) NOT NULL, `career` VARCHAR(10) NOT NULL,

	`car_number` VARCHAR(10) NOT NULL, `phone_number` VARCHAR(15) NOT NULL, PRIMARY KEY (`manager_number`)) ENGINE = InnoDB;
RESERVATION 예약	CREATE TABLE RESERVATION (`number` INT NOT NULL, `time` VARCHAR(45) NOT NULL, `place` VARCHAR(45) NOT NULL, `RESERVATION_status` VARCHAR(45) NOT NULL, `CEO_ID` VARCHAR(20) NOT NULL, `USER_ID` VARCHAR(20) NOT NULL, `WOOF_MANAGER_manager_number` VARCHAR(20) NOT NULL, PRIMARY KEY (`number`), INDEX `fk_RESERVATION_CEO_idx` (`CEO_ID` ASC) VISIBLE, INDEX `fk_RESERVATION_USER_idx` (`USER_ID` ASC) VISIBLE, INDEX `fk_RESERVATION_WOOF_MANAGER_idx` (`WOOF_MANAGER_manager_number` ASC) VISIBLE, CONSTRAINT `fk_RESERVATION_CEO` FOREIGN KEY (`CEO_ID`) REFERENCES `USER1`.`CEO` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `fk_RESERVATION_USER` FOREIGN KEY (`USER_ID`)

	REFERENCES `USER1`.`USER` (`ID`)
	ON DELETE NO ACTION
	ON UPDATE NO ACTION,
	CONSTRAINT
	`fk_RESERVATION_WOOF_MANAGER`
	FOREIGN KEY
	(`WOOF_MANAGER_manager_number`)
	REFERENCES
	`USER1`.`WOOF_MANAGER`
	(`manager_number`)
	ON DELETE NO ACTION
	ON UPDATE NO ACTION)
	ENGINE = InnoDB;

REVIEW 리뷰	CREATE TABLE REVIEW (`number` VARCHAR(45) NOT NULL, `rate` VARCHAR(45) NOT NULL, `picture` INT NOT NULL, `comment` VARCHAR(200) NOT NULL, `comment_time` VARCHAR(200) NOT NULL, `RESERVATION_number` INT NOT NULL, PRIMARY KEY (`number`), INDEX `fk_REVIEW_RESERVATION1_idx` (`RESERVATION_number`ASC)VISIBLE, CONSTRAINT `fk_REVIEW_RESERVATION1` FOREIGN KEY (`RESERVATION_number`) REFERENCES`USER1`.`RESERVATION`(`numb er`) ON DELETE NO ACTION ON UPDATE NO ACTION) ENGINE = InnoDB;

PET 반려동물	CREATE TABLE PET (`number` VARCHAR(45) NOT NULL, `speices` VARCHAR(45) NOT NULL, `age` INT NOT NULL, `gender` VARCHAR(1) NOT NULL, `sickness` VARCHAR(45) NOT NULL, `weight` INT NOT NULL, `personality` VARCHAR(45) NOT NULL, `USER_ID` VARCHAR(20) NOT NULL, `name` VARCHAR(45) NOT NULL, PRIMARY KEY (`number`), CONSTRAINT `USERfk_pet_USER` FOREIGN KEY (`USER_ID`) REFERENCES `USER1`.`USER` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION) ENGINE = InnoDB;

3.2기능별 쿼리

예약하는 화면 - 어떤 가게에 갈 것이냐	select CEO.type, CEO.phone_number, CEO.address, CEO.name, RESERVATION.time, RESERVATION.place, REVIEW.rate, REVIEW.picture, REVIEW.comment, REVIEW.comment_time, RESERVATION.time, RESERVATION.place, WOOF_MANAGER.manager_number, WOOF_MANAGER.gender, WOOF_MANAGER.career,

	WOOF_MANAGER.car_number FROM RESERVATION JOIN CEO ON RESERVATION.CEO_ID = CEO.ID JOIN REVIEW ON RESERVATION.number = REVIEW.RESERVATION_number JOIN USER ON RESERVATION.USER_ID = USER.ID JOIN WOOF_MANAGER ON RESERVATION.manager_number = WOOF_MANAGER.manager_number;
예약하는 화면 - 고객 인적사항 화면	SELECT USER.ID, USER.phone_number, PET.name, PET.speices, PET.age, PET.gender, PET.sickness, PET.weight, PET.personality FROM PET JOIN USER ON PET.USER_ID = USER.ID WHERE ID = 'id2';
예약하는 화면 - 예약완료 후 확인페이지	SELECT CEO.type, CEO.phone_number, CEO.address, CEO.name, RESERVATION.time, RESERVATION.place, RESERVATION.number, WOOF_MANAGER.manager_number, PET.name FROM CEO

	JOIN RESERVATION ON CEO.ID= RESERVATION.CEO_ID JOIN USER ON RESERVATION.USER_ID = USER.ID JOIN PET ON PET.USER_ID = USER.ID JOIN WOOF_MANAGER ON RESERVATION.manager_number = WOOF_MANAGER.manager_number WHERE CEO.phone_number = '010-1111-1111';
매니저 화면	select RESERVATION.time, USER.ID, USER.phone_number, USER.addr, CEO.name, CEO.phone_number, CEO.address, PET.name, PET.speices, PET.age, PET.gender, PET.sickness, PET.weight, PET.personality from RESERVATION JOIN CEO ON RESERVATION.CEO_ID = CEO.ID JOIN USER ON RESERVATION.USER_ID = USER.ID JOIN PET ON USER.ID = PET.USER_ID ORDER BY time ASC;
CEO 화면	select RESERVATION.time, USER.phone_number, USER.ID, USER.addr, PET.speices, PET.age, PET.gender, PET.sickness, PET.weight, PET.personality, WOOF_MANAGER.phone_number,

	WOOF_MANAGER.manager_number from USER join PET on USER.ID=PET.USER_ID JOIN RESERVATION ON USER.ID=RESERVATION.USER_ID JOIN WOOF_MANAGER ON WOOF_MANAGER.manager_number=RESERVATION.manager_number ORDER BY RESERVATION.time ASC;select * from USER; UPDATE USER SET phone_number = '010-5000-5000' WHERE ID = 'id5';

3.3시스템 아키텍처