National University of Singapore

School of Computing

CS2102

Database Systems

Project Report

PetCare18

Group 18

Chen Penghao (A0122017Y)
Kuang Ming (A0148043L)
Xia Rui (A0148000Y)
Xie Peiyi (A0141123B)

1. Introduction

Pets are often treated as important as a member of the family of their owners. However, sometimes pet owners might be unable to take care of their pets for various reasons. It would be a worrying time if the pet was not properly taken care of, such as not timely fed, or the feces not properly cleaned. It would be helpful if the pets could be taken care of by another caregiver, such that the owner would be less worried of their pets' situations.

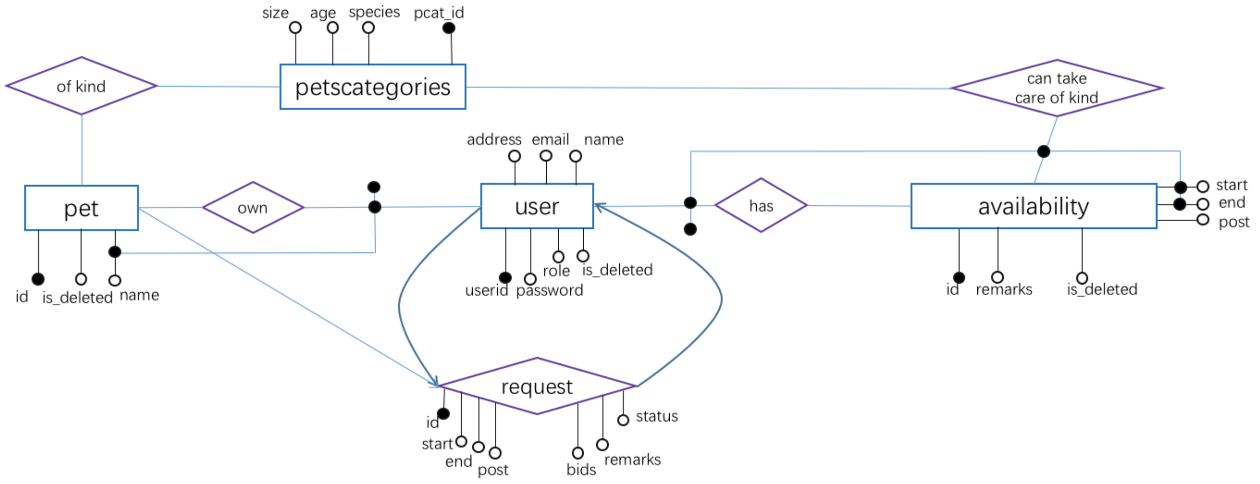
PetCare, our web application, made use of PHP and PostgreSQL to address to this problem, by connecting pet owners and care givers in an interactive and real-time manner.

2. Project Overview

Here are the technical details of our project:

Web server	Bitnami
Server Language	PHP
Database management system used	PostgreSQL
External packages installed	Bootstrap CSS, JQuery

3. Entity Relation Diagram



4. Relational Schema

4.1 Tables

```
CREATE TABLE petcategory(
    pcat_id INT PRIMARY KEY DEFAULT nextval('pcat_seq'),
    age VARCHAR(10),
    size VARCHAR(20),
    species VARCHAR(30)
);

CREATE TABLE pet_user(
    user_id INT PRIMARY KEY DEFAULT nextval('user_id_seq'),
    name VARCHAR(64) NOT NULL,
    password VARCHAR(64) NOT NULL,
    email VARCHAR(64) UNIQUE,
    address VARCHAR(64),
    role VARCHAR(10) DEFAULT 'normal' CONSTRAINT CHK_role CHECK (role in ('admin',
'normal')),
    is_deleted BOOLEAN DEFAULT FALSE
);

CREATE TABLE pet(
    pets_id INT PRIMARY KEY DEFAULT nextval('pets_id_seq'),
    owner_id INT REFERENCES pet_user(user_id) ON DELETE CASCADE,
    pcat_id INT REFERENCES petcategory(pcat_id) ON DELETE CASCADE ON UPDATE CASCADE,
    pet_name VARCHAR(64),
    is_deleted BOOLEAN DEFAULT FALSE,
    UNIQUE (owner_id, pet_name)
);

CREATE TABLE availability(
    avail_id INT PRIMARY KEY DEFAULT nextval('avail_id_seq'),
    post_time timestamp NOT NULL DEFAULT current_timestamp,
    start_time TIMESTAMP NOT NULL,
    end_time TIMESTAMP NOT NULL,
    pcat_id INT REFERENCES petcategory(pcat_id) ON DELETE CASCADE ON UPDATE CASCADE,
    taker_id INT REFERENCES pet_user(user_id) ON DELETE CASCADE,
    remarks VARCHAR(64) DEFAULT 'No',
    is_deleted BOOLEAN DEFAULT FALSE,
    UNIQUE (start_time, end_time, pcat_id, taker_id),
    CONSTRAINT CHK_start_end CHECK (end_time > start_time),
    CONSTRAINT CHK_post CHECK (start_time > post_time)
);

CREATE TABLE request(
    request_id INT PRIMARY KEY DEFAULT nextval('request_id_seq'),
    owner_id INT REFERENCES pet_user(user_id) ON DELETE CASCADE,
    taker_id INT REFERENCES pet_user(user_id) ON DELETE CASCADE,
    post_time TIMESTAMP NOT NULL DEFAULT current_timestamp,
    care_begin TIMESTAMP NOT NULL,
    care_end TIMESTAMP NOT NULL,
    remarks VARCHAR(64) DEFAULT 'No',
    bids NUMERIC NOT NULL,
    pets_id INT REFERENCES pet(pets_id) ON DELETE CASCADE ON UPDATE CASCADE,
    slot VARCHAR(64),
    totaltime DOUBLE PRECISION,
    status VARCHAR(20) CHECK (status IN ('pending', 'failed', 'successful', 'cancelled'))
DEFAULT 'pending',
    CONSTRAINT CHK_start_end CHECK (care_end > care_begin),
    CONSTRAINT CHK_post CHECK (care_begin > post_time)
);

CREATE VIEW requesttime AS
    SELECT SUM(r.bids)/SUM(r.totaltime)*60 AS avgbids, r.taker_id AS taker_id
    FROM request r
    WHERE r.status = 'successful'
    GROUP BY r.taker_id;
```

4.2 Functions & Triggers

```
CREATE OR REPLACE FUNCTION timeslot(requestNum INTEGER)
RETURNS VARCHAR(64) AS $$
DECLARE slot VARCHAR(64); hours DOUBLE PRECISION; beginTime timestamp;
BEGIN
SELECT care_begin INTO beginTime FROM request WHERE request_id = requestNum;
hours = extract(HOUR FROM (beginTime));
IF hours BETWEEN 6 AND 11 THEN slot = 'Morning';
ELSE IF hours BETWEEN 12 AND 17 THEN slot = 'Afternoon';
ELSE IF hours BETWEEN 18 AND 23 THEN slot = 'Evening';
ELSE slot = 'Before Dawn';
END IF;
END IF;
END IF;
RETURN slot;
END; $$
LANGUAGE PLPGSQL;

CREATE OR REPLACE FUNCTION calculateTotalTime(requestNum INTEGER)
RETURNS DOUBLE PRECISION AS $$
DECLARE totalmins DOUBLE PRECISION; days DOUBLE PRECISION; hours DOUBLE PRECISION; mins
DOUBLE PRECISION;
startTime timestamp; endTime timestamp;
BEGIN
SELECT care_begin, care_end INTO startTime, endTime FROM request WHERE request_id =
requestNum;
mins = extract(MINUTE FROM (endTime - startTime));
days = extract(DAY FROM (endTime - startTime));
hours = extract(HOUR FROM (endTime - startTime));
totalmins = mins + 60 * (hours + 24 * days);
RETURN totalmins;
END; $$
LANGUAGE PLPGSQL;

CREATE OR REPLACE FUNCTION addRequestInfo()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE request
    SET slot= timeslot(new.request_id), totaltime = calculateTotalTime(new.request_id)
    WHERE request_id = new.request_id;
    RETURN NULL;
END; $$
LANGUAGE PLPGSQL;


CREATE TRIGGER addSlot
AFTER INSERT
ON request
FOR EACH ROW
EXECUTE PROCEDURE addRequestInfo();

CREATE OR REPLACE FUNCTION cleanOutdatedAvail()
RETURNS TRIGGER AS $$
BEGIN
  UPDATE availability
  SET is_deleted = TRUE
  WHERE end_time <= CURRENT_TIMESTAMP
  and is_deleted = FALSE
  RETURN NULL;
END; $$
LANGUAGE PLPGSQL
```

```
CREATE OR REPLACE FUNCTION cleanOutdatedReq()
RETURNS TRIGGER AS $$
BEGIN
  UPDATE request
  SET status = 'cancelled'
  WHERE (end_time <= CURRENT_TIMESTAMP
  AND status = 'pending')
  OR
  (request_id NOT IN (SELECT r.request_id
                      FROM request r INNER JOIN pet p ON r.pets_id = p.pets_id
                                     INNER JOIN availability a ON a.pcat_id = p.pcat_id
                      WHERE r.taker_id = a.taker_id
                      AND a.is_deleted = FALSE
                      AND p.is_deleted = FALSE
                      AND r.care_end <= a.end_time
                      AND r.care_begin >= a.begin_time)
  AND status = 'pending')
  RETURN NULL;
END; $$
LANGUAGE PLPGSQL

CREATE TRIGGER changeAvail
BEFORE INSERT OR UPDATE availability
FOR EACH STATEMENT
EXECUTE PROCEDURE cleanOutdatedAvail();

CREATE TRIGGER changeReq
BEFORE INSERT OR UPDATE request
FOR EACH STATEMENT
EXECUTE PROCEDURE cleanOutdatedReq();
```

5. SQL Code Snippets & Screenshot of Webpages

5.1 Pet Owner Dashboard

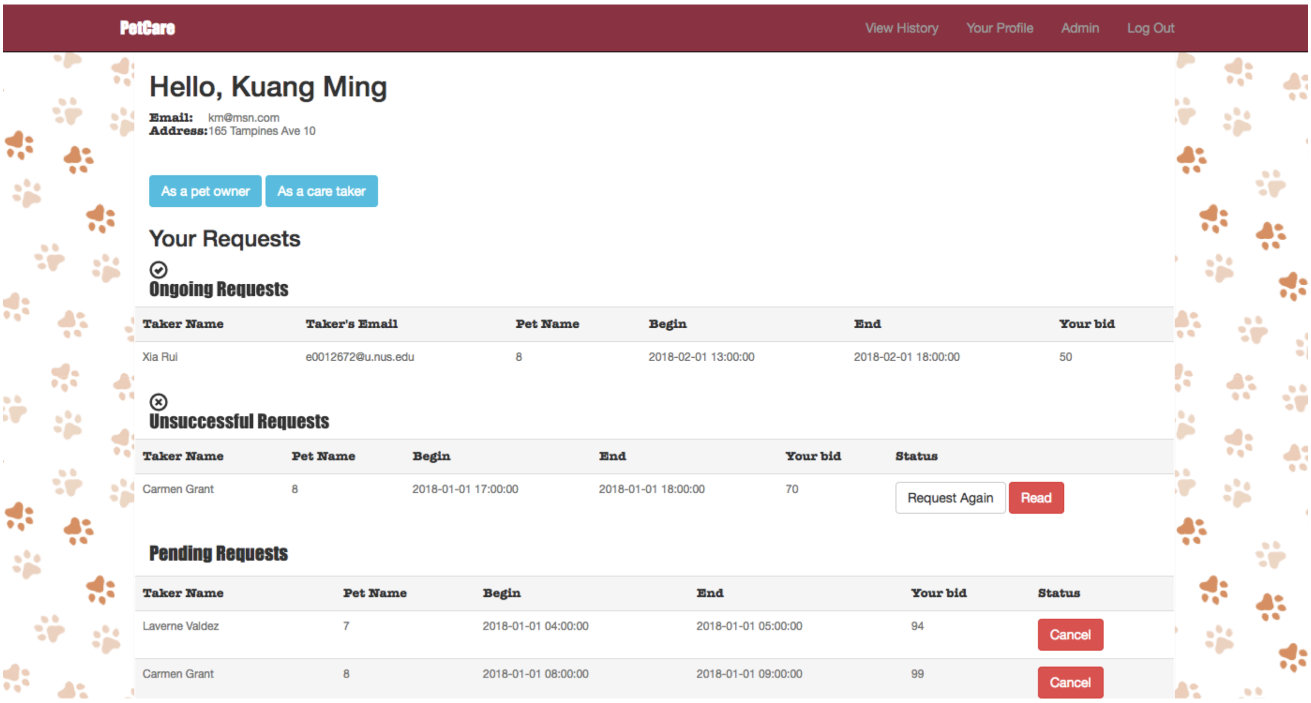
- Show all ongoing/unsuccessful/pending request:

```
SELECT u.name, u.email, r.care_begin, r.care_end, r.bids, p.pet_name
FROM request r, pet_user u, pet p
WHERE r.owner_id = $user_id AND r.status = 'successful'
      AND r.care_end > current_timestamp AND r.taker_id = u.user_id
      AND r.pets_id = p.pets_id AND p.is_deleted = false
ORDER BY care_begin;
```

r.status = ‘successful’ or ‘failed’ or ‘pending’ accordingly in ongoing/unsuccessful/pending requests

- For cancel pending request and read unsuccessful requests:

```
UPDATE request SET status = 'cancelled' WHERE request_id =$request_id;
```



Hello, Kuang Ming

Email: km@msn.com
Address: 165 Tampines Ave 10

Your Requests

Ongoing Requests

Taker Name	Taker's Email	Pet Name	Begin	End	Your bid
Xia Rui	e0012672@u.nus.edu	8	2018-02-01 13:00:00	2018-02-01 18:00:00	50

Unsuccessful Requests

Taker Name	Pet Name	Begin	End	Your bid	Status
Carmen Grant	8	2018-01-01 17:00:00	2018-01-01 18:00:00	70	Request Again Read

Pending Requests

Taker Name	Pet Name	Begin	End	Your bid	Status
Laverne Valdez	7	2018-01-01 04:00:00	2018-01-01 05:00:00	94	Cancel
Carmen Grant	8	2018-01-01 08:00:00	2018-01-01 09:00:00	99	Cancel

5.2 Pet Taker Dashboard

- Show all ongoing/pending request:

```
SELECT r.request_id, u.name, u.email, r.care_begin, r.care_end,
       r.remarks, r.bids, p.pet_name, c.age, c.size, c.species
FROM request r, pet p, petcategory c, pet_user u
WHERE r.taker_id = $user_id AND r.status = 'pending'
      AND r.care_begin > CURRENT_TIMESTAMP AND p.pets_id = r.pets_id
      AND p.pcat_id = c.pcat_id AND u.user_id = r.owner_id
      AND p.is_deleted = false
ORDER BY r.bids DESC;
```

r.status = ‘successful’ or ‘pending’ accordingly in ongoing/pending requests

- Reject:

```
UPDATE request SET status = 'failed' WHERE request_id =$reject_id;
```

- Before Accept check #overlap request:

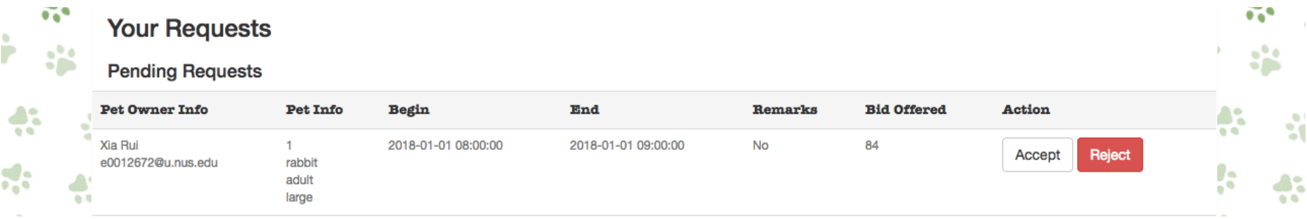
```
SELECT COUNT(*) FROM request r1,request r2
WHERE r1.request_id = $accept_id AND r2.taker_id = $user_id
      AND r2.status = 'successful'
      AND r1.care_begin < r2.care_end AND r1.care_end > r2.care_begin;
```

- Accept and cancel all request with same pet and time overlap:

```
UPDATE request SET status = 'cancelled'
WHERE request_id <> $accept_id AND pets_id = $pets_id
      AND '$start' < care_end AND '$end' > care_begin;
```

- Show all available slot

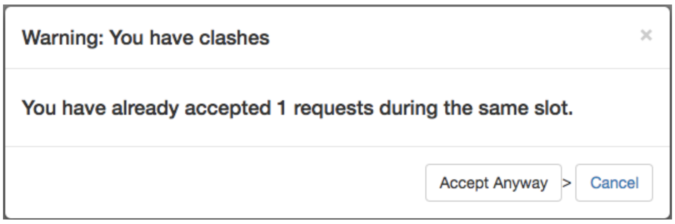
```
SELECT a.avail_id, a.start_time, a.end_time, a.remarks, p.species,
       p.size, p.age
FROM availability a, petcategory p
WHERE a.pcat_id = p.pcat_id AND a.taker_id =$user_id
      AND a.is_deleted = FALSE AND a.start_time > CURRENT_TIMESTAMP
ORDER BY a.start_time;
```



Your Requests

Pending Requests

Pet Owner Info	Pet Info	Begin	End	Remarks	Bid Offered	Action
Xia Rui e0012672@u.nus.edu	1 rabbit adult large	2018-01-01 08:00:00	2018-01-01 09:00:00	No	84	Accept Reject



Warning: You have clashes

You have already accepted 1 requests during the same slot.

5.3 Send Request

- Search without any constraint:

```
SELECT a.avail_id, a.start_time, a.end_time, a.taker_id, p.name,
       (CASE WHEN t.avgbids is NULL THEN 0 ELSE t.avgbids END) AS avgbids,
       a.remarks
FROM (availability a INNER JOIN pet_user p ON p.user_id = a.taker_id
      AND a.is_deleted = FALSE AND p.is_deleted = FALSE)
      LEFT JOIN requesttime AS t ON a.taker_id = t.taker_id
WHERE a.taker_id <> '$user_id'
```

- When pet/start-time/end-time specified:

```
AND a.pcat_id = $pcat_id
AND a.start_time <= '$start_time'
AND a.end_time >= '$end_time'
```

- When all the above three specified:

```
AND a.taker_id NOT IN (SELECT r.taker_id FROM request r WHERE r.care_end >
                       '$start_time' AND r.care_begin < '$end_time' AND r.pets_id
                       = $pet_id AND r.status='pending')
```

- When preferred taker_name specified:

```
AND UPPER(p.name) LIKE UPPER('%$taker_name%')
```

- When average bids/hour range specified:

```
AND (t.avgbids <= $upperbound OR t.avgbids is NULL)
AND (t.avgbids >= $lowerbound OR t.avgbids is NULL)
```

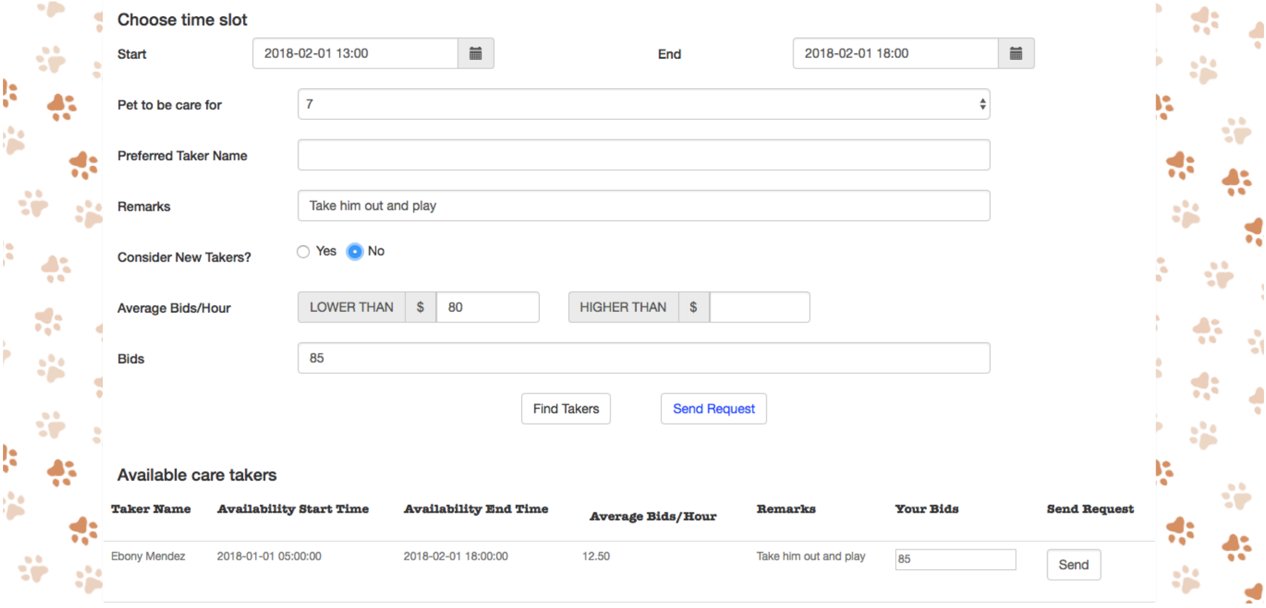
- When owner do not want new takers: `AND t.avgbids is NOT NULL`
- Added in the end: `ORDER BY avgbids ASC;`

- Before sending requests, check the pet has not been taken care of in the specific period:

```
SELECT * FROM request r WHERE r.care_begin <'$end_time' AND r.care_end >
'$start_time' AND r.pets_id = $pet_id AND r.status = 'successful';
```

- Finally send the request:

```
INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids,
pets_id) VALUES ($user_id, $taker_id, '$start_time', '$end_time', '$remarks',
$bids, $pet_id);
```



Choose time slot

Start	2018-02-01 13:00	End	2018-02-01 18:00

Pet to be care for	7
Preferred Taker Name	
Remarks	Take him out and play
Consider New Takers?	○ Yes ● No
Average Bids/Hour	LOWER THAN $ 80 · HIGHER THAN $
Bids	85

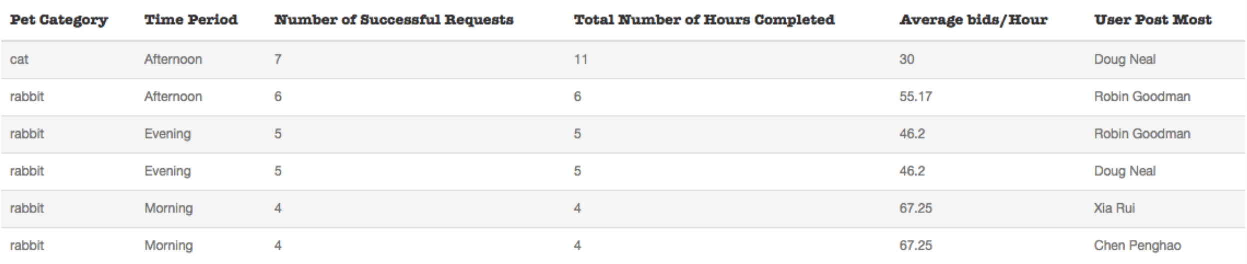
Available care takers

Taker Name	Availability Start Time	Availability End Time	Average Bids/Hour	Remarks	Your Bids	Send Request
Ebony Mendez	2018-01-01 05:00:00	2018-02-01 18:00:00	12.50	Take him out and play	85	Send

5.4 Admin

- Group all the successful request by timeslots(morning/afternoon/evening) and pet species
- For each group, find total #successful request, average bids/hour, and users that posts most

```
SELECT k.species, k.timeslot, k.RequestNum, k.average, r1.owner_id, k.totaltime
FROM (SELECT c.species AS species, r.slot AS timeslot,
             COUNT(r.request_id) AS RequestNum,
             (SUM(r.totaltime)/60) AS totaltime,
             (SUM(r.bids)/SUM(r.totaltime)*60) AS average
      FROM petcategory c, pet p, request r
      WHERE r.pets_id = p.pets_id AND c.pcat_id = p.pcat_id
            AND r.status = 'successful'
      GROUP BY r.slot, c.species) AS k, request r1, petcategory c1, pet p1
WHERE r1.pets_id = p1.pets_id AND c1.pcat_id = p1.pcat_id
      AND r1.status = 'successful' AND c1.species = k.species
      AND r1.slot = k.timeslot
GROUP BY r1.owner_id, k.species, k.timeslot, k.RequestNum, k.average, k.totaltime
HAVING COUNT(*) >= ALL( SELECT COUNT(*)
                        FROM request r2, petcategory c2, pet p2
                        WHERE r2.pets_id = p2.pets_id AND c2.pcat_id = p2.pcat_id
                              AND r2.status = 'successful'
                              AND c2.species = k.species AND r2.slot = k.timeslot
                        GROUP BY r2.owner_id)
ORDER BY k.RequestNum DESC;
```



Pet Category	Time Period	Number of Successful Requests	Total Number of Hours Completed	Average bids/Hour	User Post Most
cat	Afternoon	7	11	30	Doug Neal
rabbit	Afternoon	6	6	55.17	Robin Goodman
rabbit	Evening	5	5	46.2	Robin Goodman
rabbit	Evening	5	5	46.2	Doug Neal
rabbit	Morning	4	4	67.25	Xia Rui
rabbit	Morning	4	4	67.25	Chen Penghao

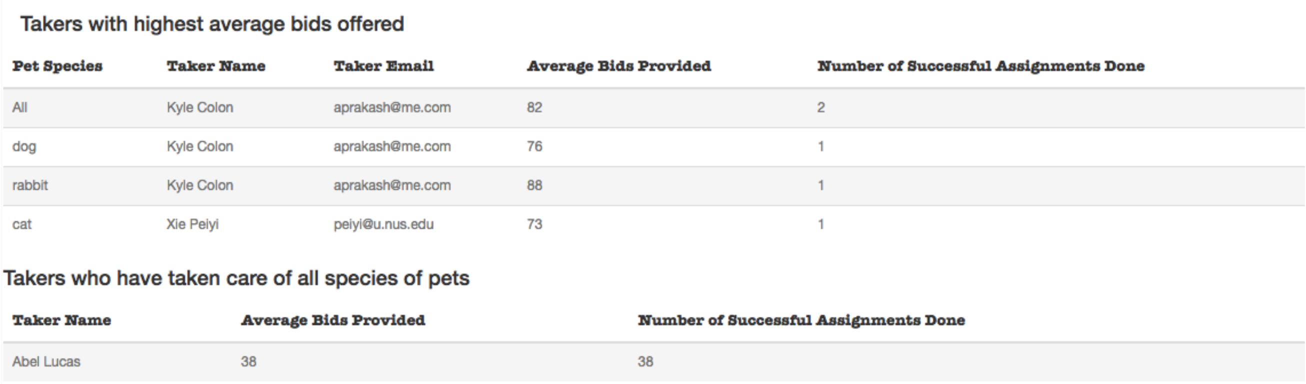
- Find the average bids/hour… of takers who have taken care of all pet species

```
SELECT u.name, (SUM(r1.bids)/SUM(r1.totaltime)*60) AS average, SUM(r1.totaltime)
FROM request r1, pet_user u
WHERE r1.taker_id = u.user_id AND r1.status = 'successful'
      AND NOT EXISTS (SELECT c1.species
                      FROM petcategory c1
                      WHERE NOT EXISTS (SELECT *
                                        FROM request r2, pet p, petcategory c2
                                        WHERE r2.taker_id = r1.taker_id
                                              AND r2.pets_id = p.pets_id
                                              AND p.pcat_id = c2.pcat_id
                                              AND c2.species = c1.species
                                              AND r2.status = 'successful'))
GROUP BY r1.taker_id, u.name
ORDER BY average DESC;
```

- For all takers, find the takers with highest average bids/hour
- For every species, find the takers with highest average bids/hour

```
SELECT u.name, u.email, k.average, k.num
FROM (SELECT r.taker_id AS id, (SUM(r.bids)/SUM(r.totaltime)*60) AS average,
             (SUM(r.totaltime)/60) AS num
     FROM request r WHERE r.status = 'successful'
     GROUP BY r.taker_id) AS k, pet_user u
WHERE u.user_id = k.id AND NOT EXISTS(SELECT *
                                      FROM (SELECT
                                           (SUM(r1.bids)/SUM(r1.totaltime)*60) AS
                                            avg FROM request r1
                                      GROUP BY r1.taker_id) AS k1
                                      WHERE k.average < k1.avg);

SELECT k.species, u.name, u.email, k.average, k.num
FROM (SELECT r.taker_id AS id, (SUM(r.bids)/SUM(r.totaltime)*60) AS average,
             (SUM(r.totaltime)/60) AS num, c.species AS species
      FROM request r, pet p, petcategory c
      WHERE r.pets_id = p.pets_id AND p.pcat_id = c.pcat_id
            AND r.status = 'successful'
      GROUP BY c.species, r.taker_id) AS k, pet_user u
WHERE u.user_id = k.id AND NOT EXISTS(
                               SELECT * FROM (SELECT
                              (SUM(r1.bids)/SUM(r1.totaltime)*60) AS avg
                               FROM request r1, pet p1, petcategory c1
                               WHERE r1.pets_id = p1.pets_id
                                     AND p1.pcat_id = c1.pcat_id
                                     AND c1.species = k.species
                                     AND r1.status = 'successful'
                               GROUP BY r1.taker_id) AS k1
                               WHERE k.average < k1.avg);
```



Takers with highest average bids offered

Pet Species	Taker Name	Taker Email	Average Bids Provided	Number of Successful Assignments Done
All	Kyle Colon	aprakash@me.com	82	2
dog	Kyle Colon	aprakash@me.com	76	1
rabbit	Kyle Colon	aprakash@me.com	88	1
cat	Xie Peiyi	peiyi@u.nus.edu	73	1

Takers who have taken care of all species of pets

Taker Name	Average Bids Provided	Number of Successful Assignments Done
Abel Lucas	38	38

6. Sample Data

6.1. Pet Users

```
INSERT INTO pet_user(name, password, email, address, role) VALUES ('Xia
Rui',12345,'e0012672@u.nus.edu','30 Ang Mo Kio Ave 8', 'admin');
INSERT INTO pet_user(name, password, email, address, role) VALUES ('Chen
Penghao',12345,'e0004801@u.nus.edu','33 Lorong 2 Toa Payoh', 'admin');
INSERT INTO pet_user(name, password, email, address, role) VALUES ('Xie
Peiyi',12345,'peiyi@u.nus.edu','55 Hougang Ave 10', 'admin');
INSERT INTO pet_user(name, password, email, address, role) VALUES ('Kuang
Ming',12345,'km@msn.com','', 'admin');

INSERT INTO pet_user(name, password, email, address) VALUES ('Patti
Dennis',12345,'empathy@msn.com','157 Foxrun Street Newnan, GA 30263');
INSERT INTO pet_user(name, password, email, address) VALUES ('Carmen
Grant',23456,'presoff@hotmail.com','9 South Surrey Street Rockford, MI 49341');
INSERT INTO pet_user(name, password, email, address) VALUES ('Abel
Lucas',34567,'keijser@optonline.net','930 Storm Court Washington, PA 15301');
INSERT INTO pet_user(name, password, email, address) VALUES ('Marguerite
Jennings',45678,'curly@gmail.com','508 E. Longfellow Rd. Revere, MA 02151');
INSERT INTO pet_user(name, password, email, address) VALUES ('Samuel
Lawrence',56789,'squirrel@aol.com','8807 Aurora Road Ogden, UT 84404');
INSERT INTO pet_user(name, password, email, address) VALUES ('Lydia
Turner',67900,'cantu@verizon.net','29 Paradise Court Moorhead, MN 56560');
INSERT INTO pet_user(name, password, email, address) VALUES ('Eloise
Cooper',79011,'pajas@msn.com','9267 1st St. Wenatchee, WA 98801');
INSERT INTO pet_user(name, password, email, address) VALUES ('Maxine
Ramos',90122,'vertigo@aol.com','671 Liberty Dr. Ankeny, IA 50023');
INSERT INTO pet_user(name, password, email, address) VALUES ('Kyle
Colon',12334,'aprakash@me.com','49 Walt Whitman Street Apopka, FL 32703');
INSERT INTO pet_user(name, password, email, address) VALUES ('Laverne
Valdez',12344,'lishoy@verizon.net','12 Bald Hill Street Norfolk, VA 23503');
INSERT INTO pet_user(name, password, email, address) VALUES ('David
Reynolds',23455,'marnanel@hotmail.com','224 Second Drive Cocoa, FL 32927');
INSERT INTO pet_user(name, password, email, address) VALUES ('Clyde
Mack',34566,'smartfart@verizon.net','870 Addison Court Dacula, GA 30019');
INSERT INTO pet_user(name, password, email, address) VALUES ('Cameron
Huff',45677,'petersko@yahoo.ca','7834 Ann Street Quincy, MA 02169');
INSERT INTO pet_user(name, password, email, address) VALUES ('Ebony
Mendez',56788,'avalon@att.net','8789 Hart St. Ballston Spa, NY 12020');
INSERT INTO pet_user(name, password, email, address) VALUES ('Joe
Munoz',67899,'ournews@live.com','94 Meadowbrook St.Apt 36 Florence, SC 29501');
INSERT INTO pet_user(name, password, email, address) VALUES ('Travis
Pearson',79010,'chaffar@mac.com','436 E. Second Avenue Missoula, MT 59801');
INSERT INTO pet_user(name, password, email, address) VALUES ('Robin
Goodman',90121,'mdielmann@hotmail.com','11 Brewer Road Chardon, OH 44024');
INSERT INTO pet_user(name, password, email, address) VALUES ('Marcus
Gilbert',81232,'weazelman@yahoo.com','12 Summerhouse St. Hoboken, NJ 07030');
INSERT INTO pet_user(name, password, email, address) VALUES ('Doug
Neal',12343,'msloan@me.com','5 East Proctor Street Missoula, MT 59801');
INSERT INTO pet_user(name, password, email, address) VALUES ('Josephine
Erickson',23454,'goresky@msn.com','7943 East Lakeshore Street Rockford, MI 49341');
```

6.2. Pet Categories

```
INSERT INTO petcategory (age, size, species) VALUES ('puppy','small','cat');
INSERT INTO petcategory (age, size, species) VALUES ('puppy','small','dog');
INSERT INTO petcategory (age, size, species) VALUES ('puppy','small','rabbit');
INSERT INTO petcategory (age, size, species) VALUES ('puppy','medium','cat');
INSERT INTO petcategory (age, size, species) VALUES ('puppy','medium','dog');
INSERT INTO petcategory (age, size, species) VALUES ('puppy','medium','rabbit');
INSERT INTO petcategory (age, size, species) VALUES ('puppy','large','cat');
INSERT INTO petcategory (age, size, species) VALUES ('puppy','large','dog');
INSERT INTO petcategory (age, size, species) VALUES ('puppy','large','rabbit');
INSERT INTO petcategory (age, size, species) VALUES ('puppy','giant','cat');
INSERT INTO petcategory (age, size, species) VALUES ('puppy','giant','dog');
INSERT INTO petcategory (age, size, species) VALUES ('puppy','giant','rabbit');
INSERT INTO petcategory (age, size, species) VALUES ('adult','small','cat');
INSERT INTO petcategory (age, size, species) VALUES ('adult','small','dog');
INSERT INTO petcategory (age, size, species) VALUES ('adult','small','rabbit');
INSERT INTO petcategory (age, size, species) VALUES ('adult','medium','cat');
INSERT INTO petcategory (age, size, species) VALUES ('adult','medium','dog');
INSERT INTO petcategory (age, size, species) VALUES ('adult','medium','rabbit');
INSERT INTO petcategory (age, size, species) VALUES ('adult','large','cat');
INSERT INTO petcategory (age, size, species) VALUES ('adult','large','dog');
INSERT INTO petcategory (age, size, species) VALUES ('adult','large','rabbit');
INSERT INTO petcategory (age, size, species) VALUES ('adult','giant','cat');
INSERT INTO petcategory (age, size, species) VALUES ('adult','giant','dog');
INSERT INTO petcategory (age, size, species) VALUES ('adult','giant','rabbit');
```

6.3. Pets

```
INSERT INTO pet(pcat_id, owner_id, pet_name) VALUES (2,1,'Ah Beng');
INSERT INTO pet(pcat_id, owner_id, pet_name) VALUES (3,5,'Ah Lian');
INSERT INTO pet(pcat_id, owner_id, pet_name) VALUES (5,4,'Ah Hong');
INSERT INTO pet(pcat_id, owner_id, pet_name) VALUES (1,2,'Ah Ben');
INSERT INTO pet(pcat_id, owner_id, pet_name) VALUES (4,8,'Ah Wong');
INSERT INTO pet(pcat_id, owner_id, pet_name) VALUES (7,3,'Ah Kay');
INSERT INTO pet(pcat_id, owner_id, pet_name) VALUES (2,9,'Ah Seng');
INSERT INTO pet(pcat_id, owner_id, pet_name) VALUES (5,6,'Ah Leong');
INSERT INTO pet(pcat_id, owner_id, pet_name) VALUES (2,10,'Ah Mah');
INSERT INTO pet(pcat_id, owner_id, pet_name) VALUES (8,7,'Ah Wai');
```

6.4. Availabilities

```
7. INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   00:00:00','2018-02-01 01:00:00',3,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   18:00:00','2018-01-21 20:00:00',3,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   23:00:00','2018-02-01 01:00:00',3,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   17:00:00','2018-02-01 19:00:00',3,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   22:00:00','2018-02-01 23:00:00',3,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   00:00:00','2018-02-01 01:00:00',9,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   20:00:00','2018-02-01 01:00:00',9,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   22:00:00','2018-02-01 00:00:00',9,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   12:00:00','2018-02-01 13:00:00',9,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   19:00:00','2018-02-01 21:00:00',9,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   00:00:00','2018-02-01 10:00:00',13,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   17:00:00','2018-02-01 23:00:00',13,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   17:00:00','2018-02-01 19:00:00',13,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   09:00:00','2018-02-01 11:00:00',13,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   10:00:00','2018-02-01 12:00:00',13,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   06:00:00','2018-02-01 17:00:00',19,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   14:00:00','2018-02-01 20:00:00',19,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   23:00:00','2018-02-01 00:00:00',19,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   06:00:00','2018-02-01 09:00:00',19,1);
   INSERT INTO availability(start_time, end_time, pcat_id, taker_id) VALUES ('2018-01-01
   22:00:00','2018-02-01 23:00:00',19,1);
```

7.1. Requests

```
8.  INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id)
    VALUES (19,9,'2018-01-01 09:00:00','2018-01-01 10:00:00','No',3,38);
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id)
    VALUES (19,22,'2018-01-01 03:00:00','2018-01-01 04:00:00','No',26,38);
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id)
    VALUES (19,12,'2018-01-01 09:00:00','2018-01-01 10:00:00','No',63,38);
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id)
    VALUES (20,6,'2018-01-01 06:00:00','2018-01-01 07:00:00','No',59,39);
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id)
    VALUES (20,22,'2018-01-01 20:00:00','2018-01-01 21:00:00','No',13,39);
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id)
    VALUES (20,18,'2018-01-01 14:00:00','2018-01-01 15:00:00','No',30,39);
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id)
    VALUES (20,22,'2018-01-01 05:00:00','2018-01-01 06:00:00','No',1,40);
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id)
    VALUES (20,24,'2018-01-01 01:00:00','2018-01-01 02:00:00','No',7,40);
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id,
    status) VALUES (20,22,'2018-01-01 16:00:00','2018-01-01
    17:00:00','No',36,40,'successful');
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id,
    status) VALUES (21,20,'2018-01-01 17:00:00','2018-01-01
    18:00:00','No',64,41,'successful');
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id,
    status) VALUES (21,21,'2018-01-01 12:00:00','2018-01-01
    13:00:00','No',62,41,'successful');
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id,
    status) VALUES (21,21,'2018-01-01 06:00:00','2018-01-01
    07:00:00','No',41,41,'successful');
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id,
    status) VALUES (21,13,'2018-01-01 23:00:00','2018-01-02
    00:00:00','No',88,42,'successful');
    INSERT INTO request(owner_id, taker_id, care_begin, care_end, remarks, bids, pets_id,
    status) VALUES (21,19,'2018-01-01 21:00:00','2018-01-01
    22:00:00','No',33,42,'successful');
```