

**DBMS - Mini**  
**Project**  
**Pet**  
**Management**  
**System**

Submitted By:

Vishakha

Hegde

PES1UG20CS506

V Semester Section I

## Short Description and Scope of the Project

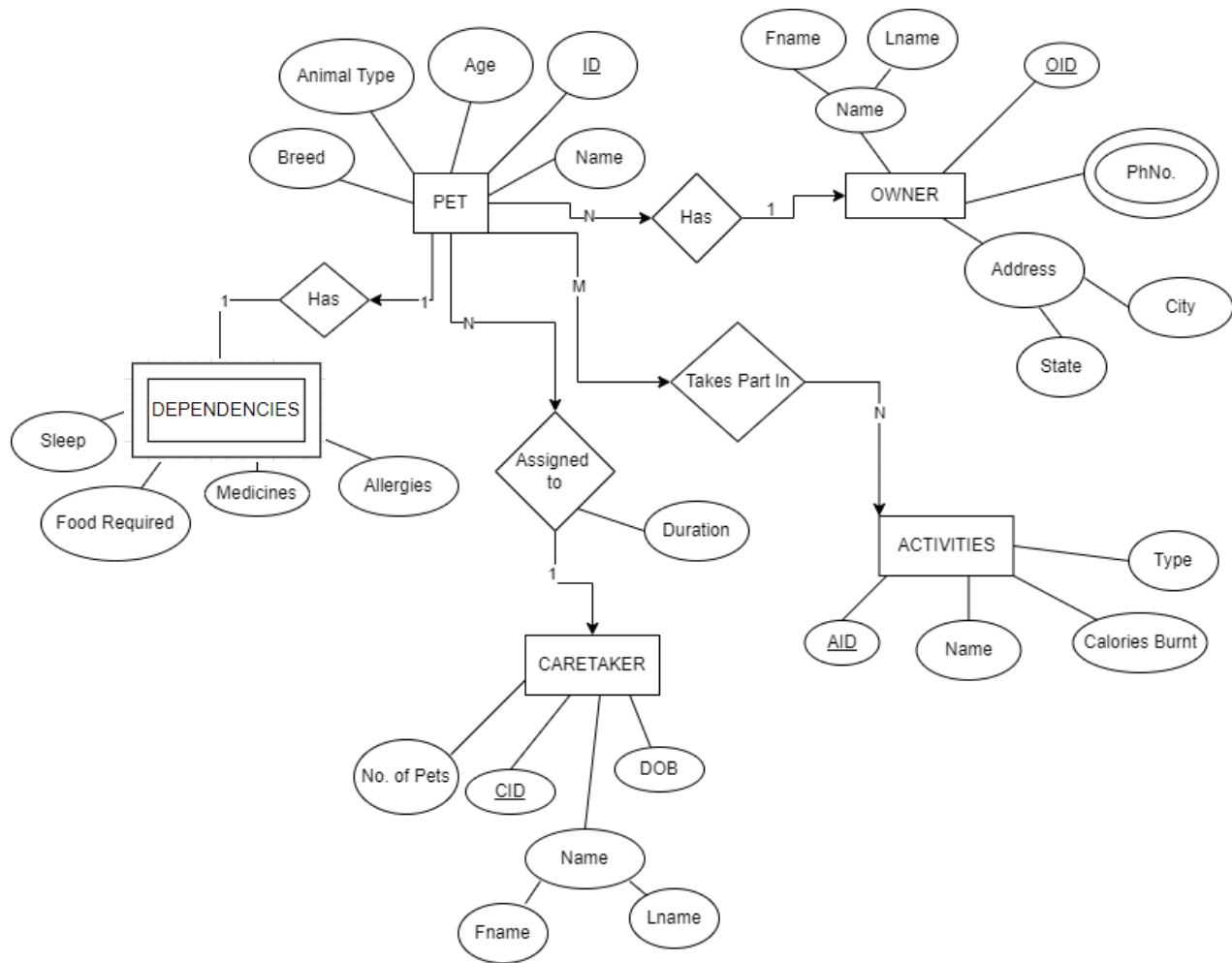
Pet lovers often face the dilemma of having to leave their furry friends home alone or under the care of neighbors/friends when going for a holiday. During this time, pets tend to feel lonely and abandoned if not given proper care. They need timely food, water, medicines in case of allergies and the right amount of exercise and sleep. Hence, the pet management system comes of immense use in these cases.

The database consists of the following tables:

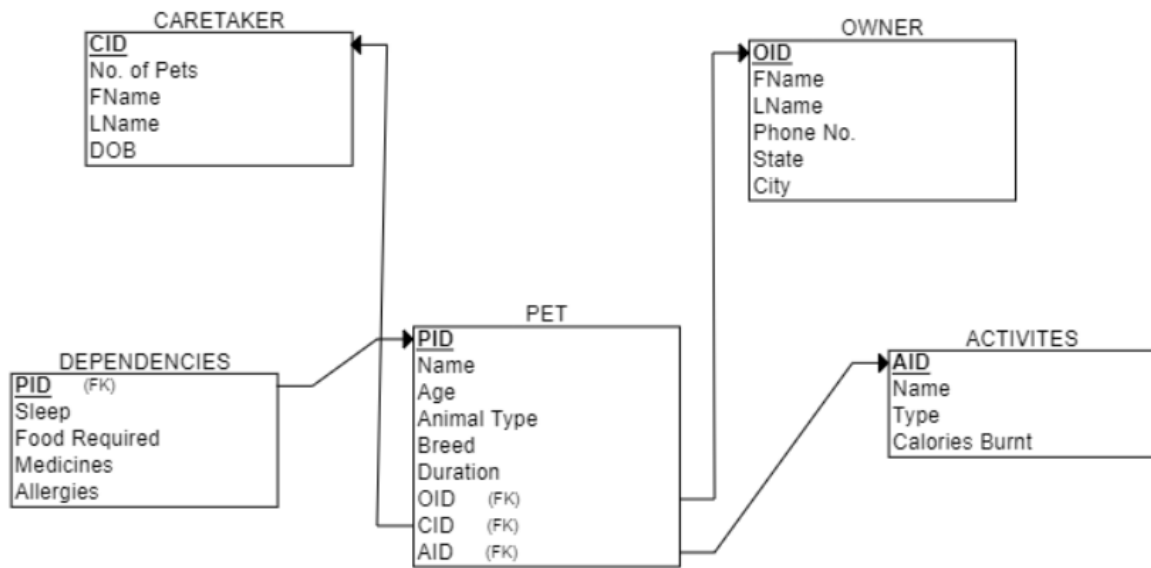
- Owner Table: Contains owner contact info. Each owner can have multiple pets.
- Caretaker Table: Contains information of the designated caretakers in the pet daycare center. Each caretaker can handle more than one pet.
- Pet Table: Contains the animal type, breed, age, caretaker and owner information, along with the duration of stay of the pet at the daycare center.
- Activities Table: Each pet can take part in numerous activities that the care center has to offer. Each activity has a certain number of calories associated with it. This guarantees that the pet gets adequate amounts of playtime and exercise.
- Dependencies: Lists out the pet's specific food, sleep and medicine requirements. Caretakers have to ensure that the needs of each pet are taken care of.

Thus, this comprehensive database makes sure that pet owners can go on holidays stress-free with their pets in safe hands. After all, a man's best friend deserves only the best!

## ER Diagram



## Relational Schema



## DDL statements - Building the database

```
--OWNER
CREATE TABLE owner
(
    oid VARCHAR(10) NOT NULL,
    fname VARCHAR(20),
    lname VARCHAR(20),
    ph_no INT,
    state VARCHAR(20),
    city VARCHAR(20),
    PRIMARY KEY (oid)
);

--ACTIVITIES
CREATE TABLE activities
(
    aid VARCHAR(10) NOT NULL,
    name VARCHAR(20),
    type VARCHAR(20),
    calories_burnt FLOAT,
    PRIMARY KEY (aid)
);

--CARETAKER
CREATE TABLE caretaker
(
    cid VARCHAR(10) NOT NULL,
    fname VARCHAR(20),
    lname VARCHAR(20),
    ph_no INT,
    age INT,
    PRIMARY KEY (cid)
);

--PET
CREATE TABLE pet
(
    pid VARCHAR(10) NOT NULL,
    name VARCHAR(20),
    animal_type VARCHAR(10),
    age INT,
    breed VARCHAR(10),
    duration INT,
    oid VARCHAR(10) NOT NULL,
    cid VARCHAR(10) NOT NULL,
    aid VARCHAR(10) NOT NULL,
```

```

PRIMARY KEY (pid),
FOREIGN KEY (oid) REFERENCES owner(oid),
FOREIGN KEY (cid) REFERENCES caretaker(cid),
FOREIGN KEY (aid) REFERENCES activities(aid)
);

--DEPENDENCIES
CREATE TABLE dependencies
(
  pid VARCHAR(10) NOT NULL,
  sleep FLOAT,
  food_required VARCHAR(20),
  medicines VARCHAR(20),
  allergies VARCHAR(20),
  PRIMARY KEY (pid),
  FOREIGN KEY (pid) REFERENCES pet(pid)
);

```

```

MariaDB [pet_management]> show tables;
+-----+
| Tables_in_pet_management |
+-----+
| activities                |
| caretaker                 |
| dependencies              |
| owner                     |
| pet                       |
+-----+

```

## Populating the Database

```
--OWNER
LOAD DATA LOCAL INFILE 'D:/Data/Mini Projects/SEM 5/DBMS/owner.csv'
INTO TABLE pet_management.owner
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

--ACTIVITES
LOAD DATA LOCAL INFILE 'D:/Data/Mini Projects/SEM 5/DBMS/activities.csv'
INTO TABLE pet_management.activities
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

--CARETAKER
LOAD DATA LOCAL INFILE 'D:/Data/Mini Projects/SEM 5/DBMS/caretaker.csv'
INTO TABLE pet_management.caretaker
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

--PET
LOAD DATA LOCAL INFILE 'D:/Data/Mini Projects/SEM 5/DBMS/pet.csv'
INTO TABLE pet_management.pet
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

--DEPENDENCIES
LOAD DATA LOCAL INFILE 'D:/Data/Mini Projects/SEM 5/DBMS/dependencies.csv'
INTO TABLE pet_management.dependencies
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

## Join Queries

Join of Pet table and Owner table based on OID (Owner ID).

```
select  
owner.oid, owner.fname, owner.lname, pet.pid, pet.name, pet.animal_type, pet.breed from  
pet inner join owner on pet.oid=owner.oid;
```

### Join of Pet and Owner

Join Pet and Owner

View Join

	oid	fname	lname	pid	name	animal_type	breed
0	10	Port	Tattoo	12	time	Turtle	African Si
1	8	Roxanne	Fawdry	13	Roberto	Hamster	Persian
2	9	Lamont	Sandilands	14	Mable	Cat	Maine Coon
3	7	Ashly	Rollings	15	Magdalene	Bird	Duck
4	6	Tedra	Bauer	16	Stanton	Dog	Bulldog
5	5	Madelena	Fetters	17		Bird	Duck
6	4	Dani	Wylde	18	Kale	Rabbit	Rex
7	2	Mead	Bretherick	20	Martguerita	Cat	Siamese
8	2	Mead	Bretherick	3	Danni	Dog	Husky
9	2	Mead	Bretherick	40	hello	Dog	Poodle



## Aggregate Functions

Number of pets that are under each caretaker:

```
SELECT caretaker.cid, caretaker.fname, caretaker.lname, COUNT(caretaker.cid) AS  
NumberOfAnimals FROM pet  
LEFT JOIN caretaker ON pet.cid = caretaker.cid  
GROUP BY cid;
```

### Aggregate of number of pets that are under each caretaker:

Aggregate

View Aggregate

	cid	fname	lname	NumberOfAnimals
0	1	Reinhard	Measor	3
1	10	Tracey	Chidler	1
2	2	Arlena	Fearey	3
3	3	Crissy	Hayne	2
4	5	Jethro	Dubery	1
5	7	Dwayne	Lingley	1
6	8	Juliet	Risborough	2
7	9	Frankie	Allnutt	1

## Set Operations

Union of Pet and Caretaker

```
select cid from pet
UNION
select cid from caretaker;
```

Table:

caretaker	
Union of Pet and caretaker	

**Union of Pet and caretaker**

View Union

	id
0	1
1	10
2	2
3	3
4	5
5	7
6	8
7	9
8	4
9	6

Union of Pet and Owner

```
select oid from pet
union
select oid from owner;
```

Table:

owner

Union of Pet and owner

### Union of Pet and owner

View Union

	id
0	10
1	13
2	14
3	15
4	2
5	4
6	5
7	6
8	7
9	8

Union of Pet and Activities

```
select aid from pet
UNION
select aid from activities;
```

Table:

activities

Union of Pet and activities

### Union of Pet and activities

View Union

	id
0	1
1	10
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

## Functions and Procedures

Function: Display medicines required for sick pets.

```
DELIMITER $$
CREATE FUNCTION sick_pet (pid INT)
RETURNS VARCHAR(20)
BEGIN
DECLARE med VARCHAR(20);
SELECT medicines INTO med from dependencies where dependencies.pid = pid;
RETURN med;
END; $$
DELIMITER ;
```

**A pet is sick! Choose Pet ID to see which medicine to give:**

Select pet

7

Click

**This pet requires:**

**Metformin**

Procedure: If any pet has 2 or less than 2 weeks left, alert the owner that pet has to leave soon.

```
DELIMITER $$
--DROP PROCEDURE IF EXISTS alert_owner$$
CREATE procedure alert_owner(OUT msg VARCHAR(30))
BEGIN
select pid,name,duration,oid from pet where duration<=2;
END;$$
DELIMITER ;
```

**Alert! These pets have less than 2 weeks left to stay!**

View Pets

	Pet ID	Pet Name	Duration	Owner ID
0	16	Stanton	1	6
1	8	Michelina	1	14
2	9	Valerye	2	13

## Triggers and Cursors

Trigger: While adding new pet entry check if duration is within range of 1 to 24 weeks.

```
DELIMITER $$
CREATE TRIGGER add_pet
BEFORE INSERT
ON pet FOR EACH ROW
BEGIN
IF NEW.duration>24 OR NEW.duration<1 THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT="Duration must be between 1 and 24 weeks!";
END IF;
END;$$
DELIMITER ;
```

### Enter pet details

Pet ID:	Age:	Owner Id:
56 - +	5 - +	3 - +
Pet Name:	Breed:	Caretaker Id:
dobby	Poodle	4 - +
Animal Type:	Duration in weeks:	Activity Id:
Dog ▼	25 - +	8 - +

Add new pet

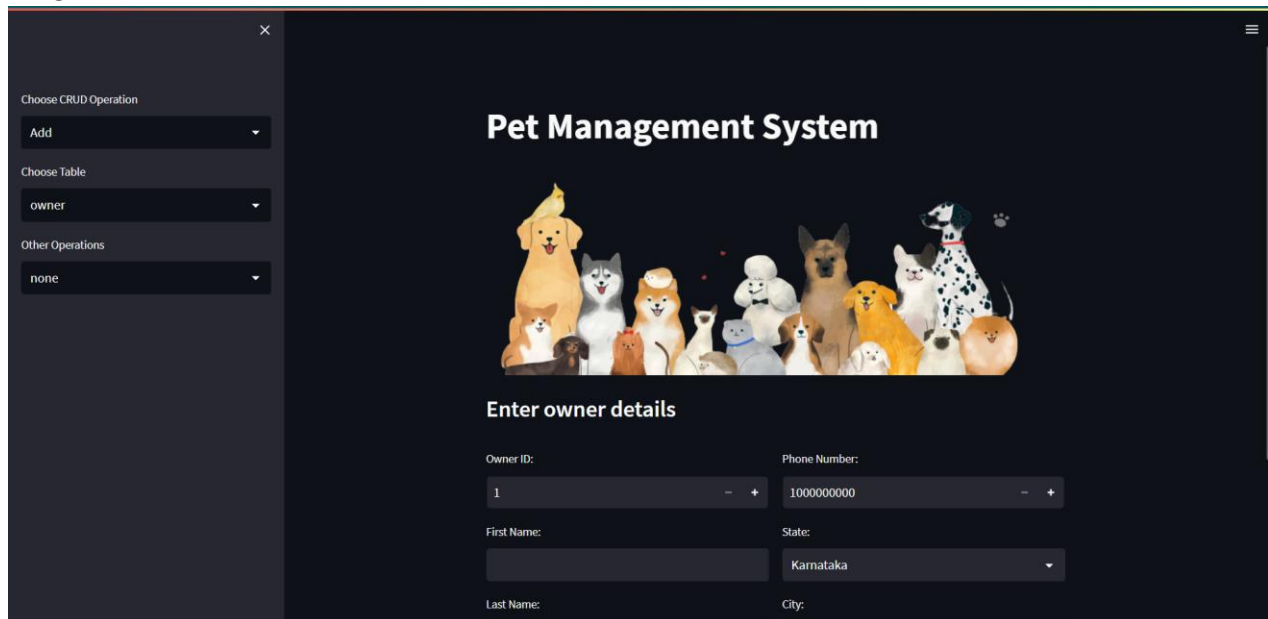
DatabaseError: 1644 (45000): Duration must be between 1 and 24 weeks!

## Developing a Frontend

The frontend should support

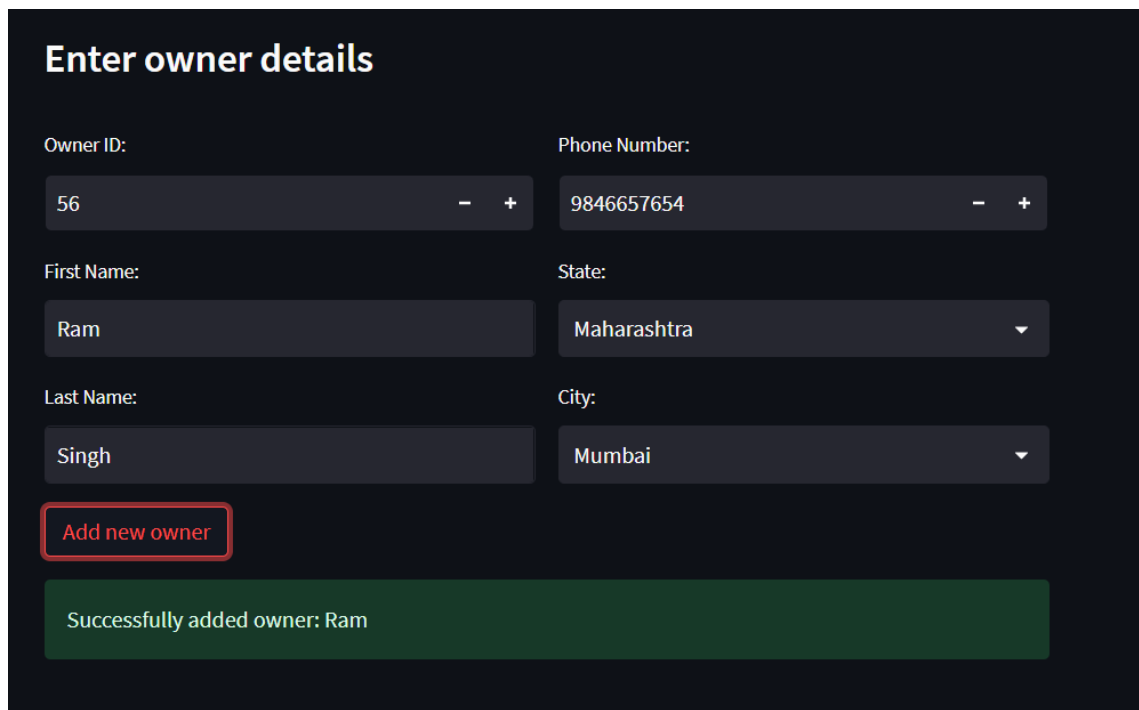
1. Addition, Modification and Deletion of records from any chosen table
2. There should be a window to accept and run any SQL statement and display the result

FRONTEND:



The screenshot shows a web application titled "Pet Management System" with a sidebar on the left. The sidebar contains three dropdown menus: "Choose CRUD Operation" (set to "Add"), "Choose Table" (set to "owner"), and "Other Operations" (set to "none"). The main content area features a header with the title "Pet Management System" and a group of cartoon animals. Below this is a form titled "Enter owner details" with the following fields: "Owner ID" (input with value 1), "Phone Number" (input with value 1000000000), "First Name" (empty input), "State" (dropdown menu with "Karnataka" selected), "Last Name" (empty input), and "City" (empty input).

Insert new owner:



This screenshot shows the "Enter owner details" form with the following values: "Owner ID" is 56, "Phone Number" is 9846657654, "First Name" is Ram, "State" is Maharashtra, "Last Name" is Singh, and "City" is Mumbai. A red rectangular box highlights the "Add new owner" button. Below the form, a green banner displays the message "Successfully added owner: Ram".

View Pet details:

Read pet details

View all Pets

	pid	name	animal_type	age	breed	duration	oid	cid	aid
0	12	time	Turtle	1	African Si	3	10	1	1
1	13	Roberto	Hamster	13	Persian	4	8	1	1
2	14	Mable	Cat	11	Maine Coon	3	9	2	3
3	15	Magdalene	Bird	1	Duck	9	7	3	5
4	16	Stanton	Dog	2	Bulldog	1	6	5	2
5	17		Bird	1	Duck	3	5	1	1
6	18	Kale	Rabbit	9	Rex	3	4	2	7
7	20	Martguerita	Cat	7	Siamese	10	2	8	8
8	3	Danni	Dog	7	Husky	6	2	3	4
9	40	hello	Dog	2	Poodle	67	2	2	2



Update pet:

Pet to Update

Roberto

New Name:

Roberto

New Caretaker ID:

3

New Age:

2

New Activity ID:

4

New Duration:

3

Update Pet

Successfully updated: Roberto

View updated Pets

	pid	name	animal_type	age	breed	duration	oid	cid	aid
0	12	time	Turtle	1	African Si	3	10	1	1
1	13	Roberto	Hamster	2	Persian	3	8	3	4
2	14	Mable	Cat	11	Maine Coon	3	9	2	3
3	15	Magdalene	Bird	1	Duck	9	7	3	5
4	16	Stanton	Dog	2	Bulldog	1	6	5	2



Delete Pet:

## Delete pet details

View all Pets ▾

Pet to Delete

time ▾

Do you want to delete :time

Delete Pet

Pet and its dependencies have been deleted successfully

User defined SQL Statement:

## Enter your SQL query here:

Enter SQL query here

select fname, cid from caretaker;

Execute!

Successfully executed query!

	0	1
0	Reinhard	1
1	Tracey	10
2	Arlena	2
3	Crissy	3
4	Rozelle	4
5	Jethro	5
6	Darby	6
7	Dwayne	7
8	Juliet	8

## Making Modifications

Modification to be made:

Add a new column 'Sickness' in the dependencies table and display Pet Name, Caretaker Name and Medicine if that pet is sick.

Adding new column 'sickness' in Dependencies table:

```
MariaDB [pet_management]> desc dependencies;
```

Field	Type	Null	Key	Default
pid	varchar(10)	NO	PRI	NULL
sleep	float	YES		NULL
food_required	varchar(20)	YES		NULL
medicines	varchar(20)	YES		NULL
allergies	varchar(20)	YES		NULL
sickness	int(11)	YES		NULL

PROCEDURE:

```
delimiter //
DROP PROCEDURE IF EXISTS sick_animal//
create procedure sick_animal()
begin
truncate table sick_table;
insert into sick_table (name,medicines,fname)
select pet.name, dependencies.medicines, caretaker.fname
from pet, dependencies,caretaker
where pet.pid=dependencies.pid and pet.cid=caretaker.cid and
dependencies.sickness=1;
end //
delimiter ;
```

FRONTEND:

## Sick Pets

Click!

View Sick Pets and their corresponding caretakers and medicines: ^

	Pet Name	Medicines	Caretaker Name
0	Mable	Metformin	Arlena
1	Stanton	Antibiotics	Jethro
2	Kale	Metformin	Arlena
3	Martguerita	Antibiotics	Juliet
4	Arvy	Hydrocodone	Tracey
5	Michelina	Antibiotics	Juliet