Kennel Care Project

12.9.2015

—

Team Members: 3

ISC 329

Introduction

The Kennel Care project aims to create an electronic database and management system to replace a file-based system. An electronic database management system would help create a more organized and orderly system of managing data.

Project Overview

The pet owners are currently making appointments over the phone with the caretakers at Kennel Care. The secretary has to call the pet owners to remind them about their appointment. When they bring their pet in for its stay, they have to fill out paperwork. This paperwork is filed away in a drawer.

Instead of doing everything manually, the employees at Kennel Care would be using a database making everything more beneficial. This would make everything easier because all the data would be in one place and easily accessible. The owner of the pet would be able to provide his/her information online, such as the length of the stay for their animal, email, phone number, etc. So when the day arrives for their animal to stay at the kennel, they would receive an email and a text message to remind them. Also, there will be an online form that provides the information that would normally be done when you walked into the Kennel Care building.

Project Scope

The scope of the project is to create a database that organizes and collects all data for the company. This system will help organize the overall structure of the company and its customers' information. The customers/owners can enter all their contact and pet information on the Kennel Care's website. Staff will have easy access to all information needed to help the overall business run smoothly with any data available at all times. Caretakers can look up their schedules to know when they need to come in to take care of certain dogs or cats. The secretary would view the customer's information to be able to bill them for their service. The manager would have access to all the information to make any appropriate changes/updates.

Document Overview

This document provides an overview of the project, a use-case diagram to show the functionality of the system, a class diagram to illustrate the main data, and a data dictionary that describes the main data.

Project Overview

Client

The clients for this project are Kennels and Dog/Cat Care Facilities. The system will be used by:

1. Employees who will be handling customer service (payment).
2. Employees who deal with the animals directly, such as groomers and caretakers.
3. Managers who deal with payments of employees and scheduling.

Main Functionality

Secretaries will access the Kennel Care system to:

1. Browse owner information: A user would search for the data on the owner and pet to be able to bill the owner for their service.

Owner will access the Kennel Care system to:

1. Provides Contact information: User inserts data, such as address, phone number, email, etc.
2. Provides Pet information: User enters breed, name, age, etc.

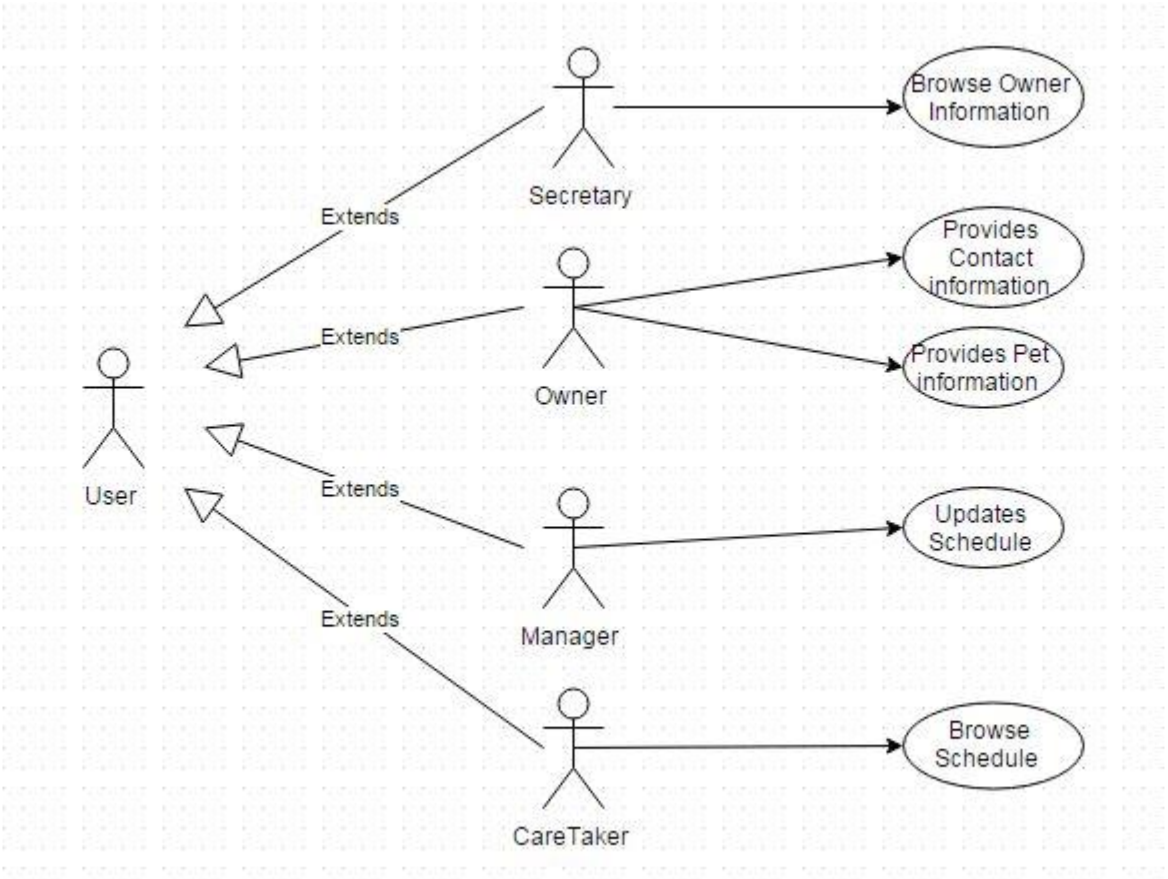
Manager will access the Kennel Care system to:

1. Updates the schedule: User inserts the start and end date of when the employee will work and the number of hours.

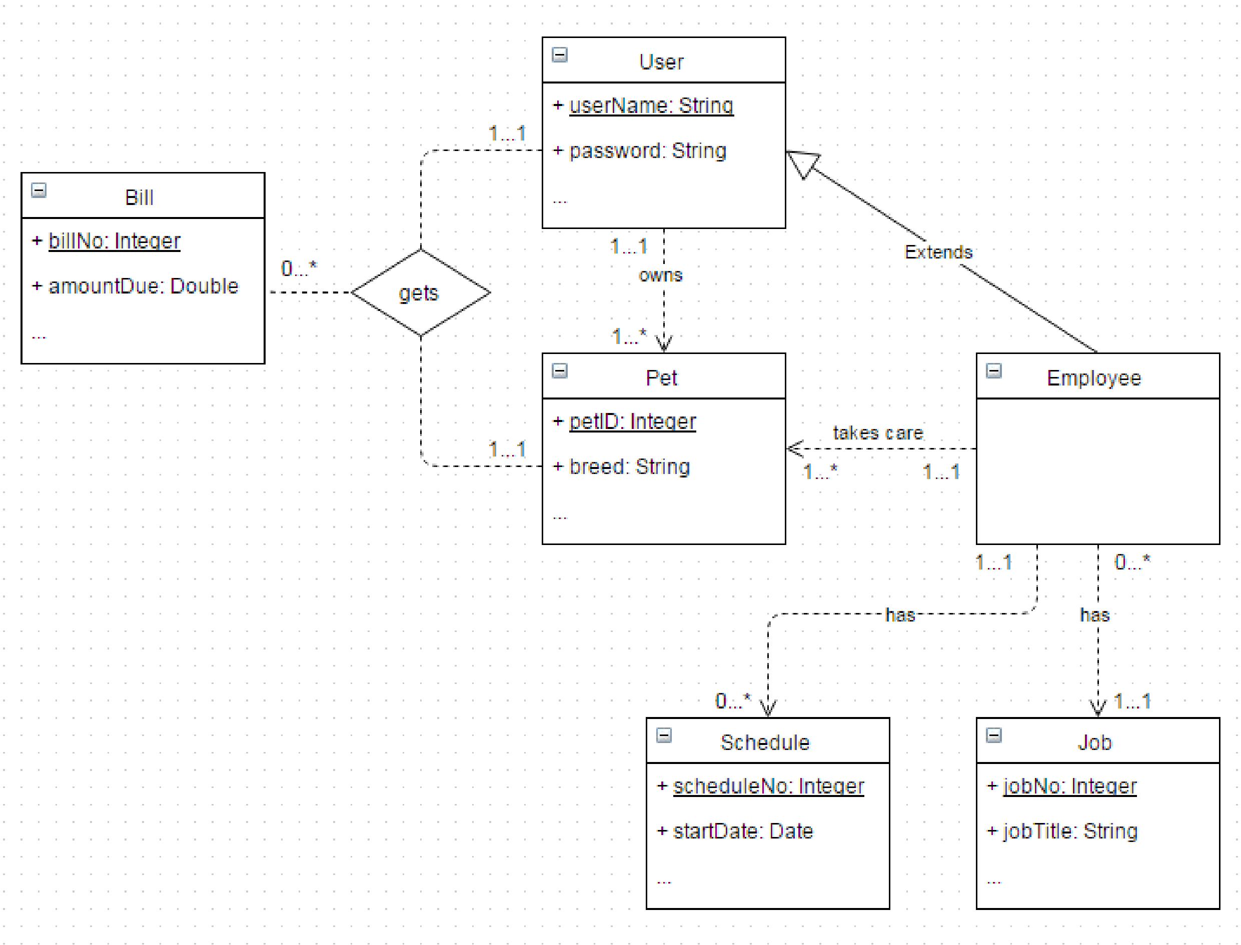
Caretaker will access the Kennel Care system to:

1. Browse schedule: The user will look up what animal and the number of days he/she will have to take care of the pet.

Use-Case Diagram



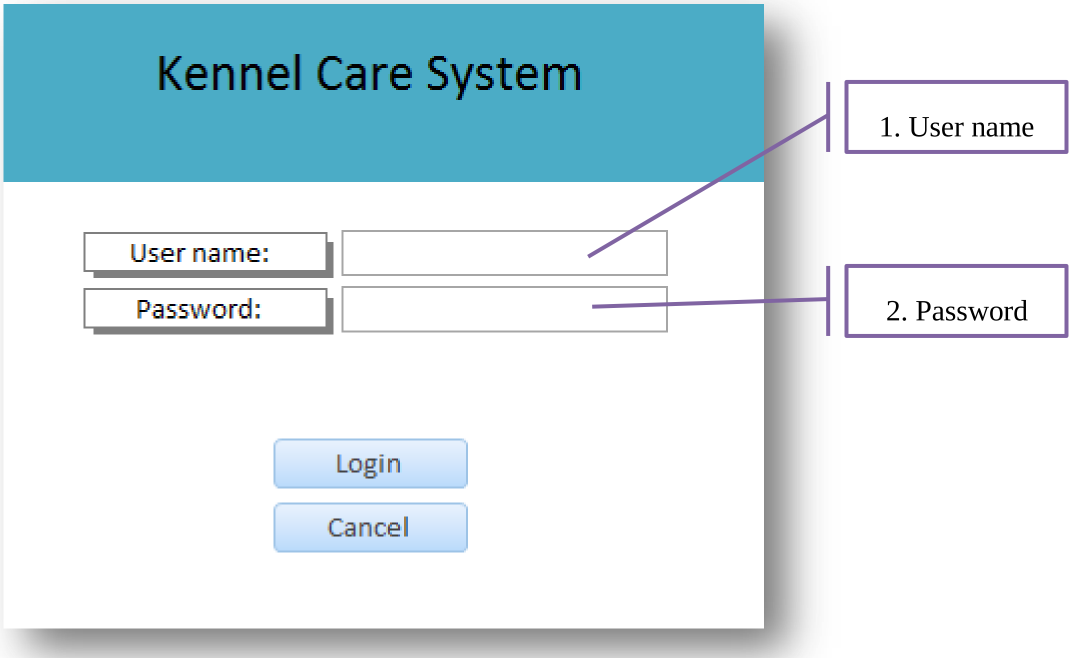
Conceptual Class Diagram



User Views

The following are the user views for the Kennel Care System. Below each user view, there is a set of relations listed.

Authenticate User UI



1. User name

2. Password

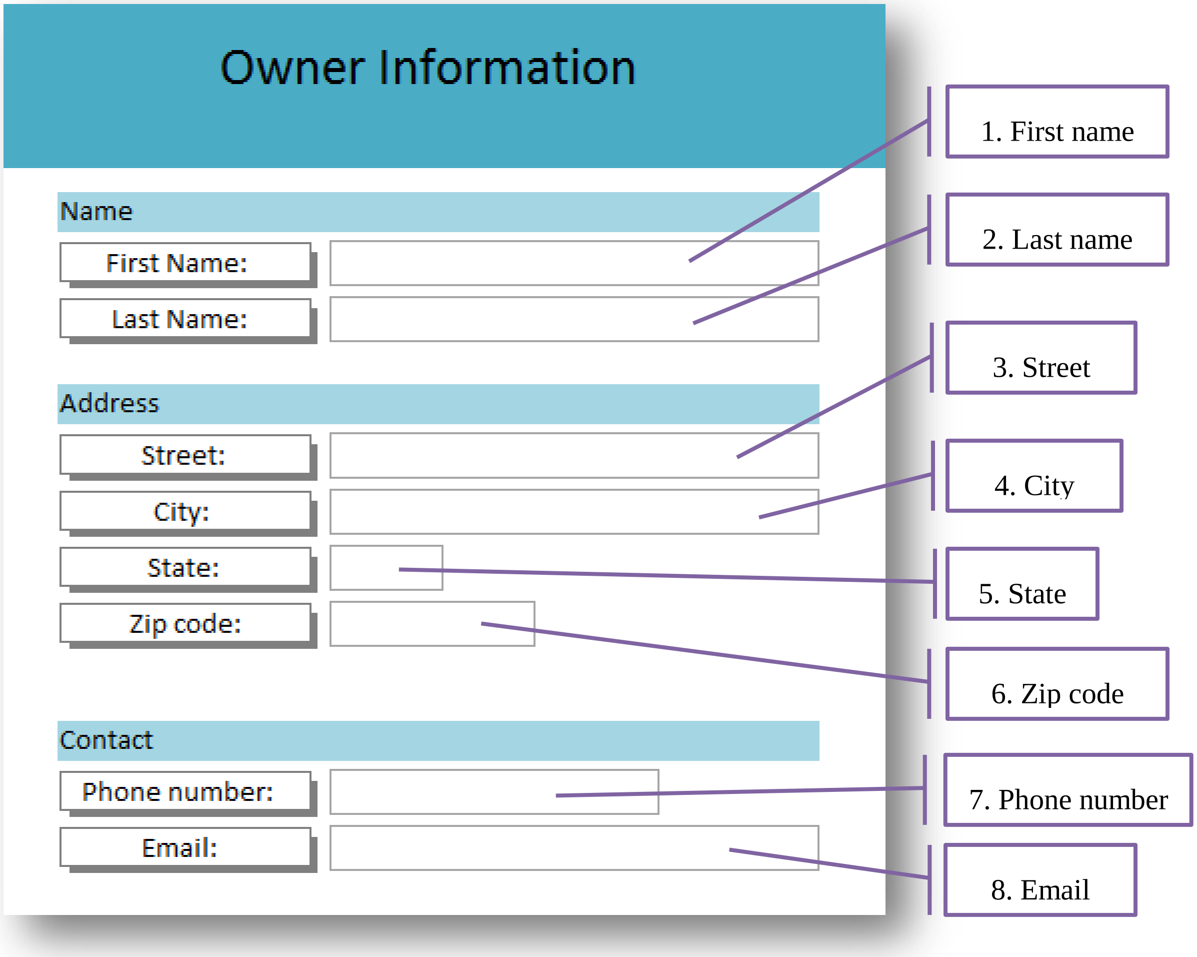
1. User name: user name

2. Password: password

<u>Normalized Relations:</u>

USER(<u>userName</u>, password)

Owner UI



1. First name: first name of the owner

2. Last name: last name of the owner

3. Street: street address of the owner

4. City: city where the owner lives

5. State: state where the owner lives

6. Zip code: zip code of where the owner lives

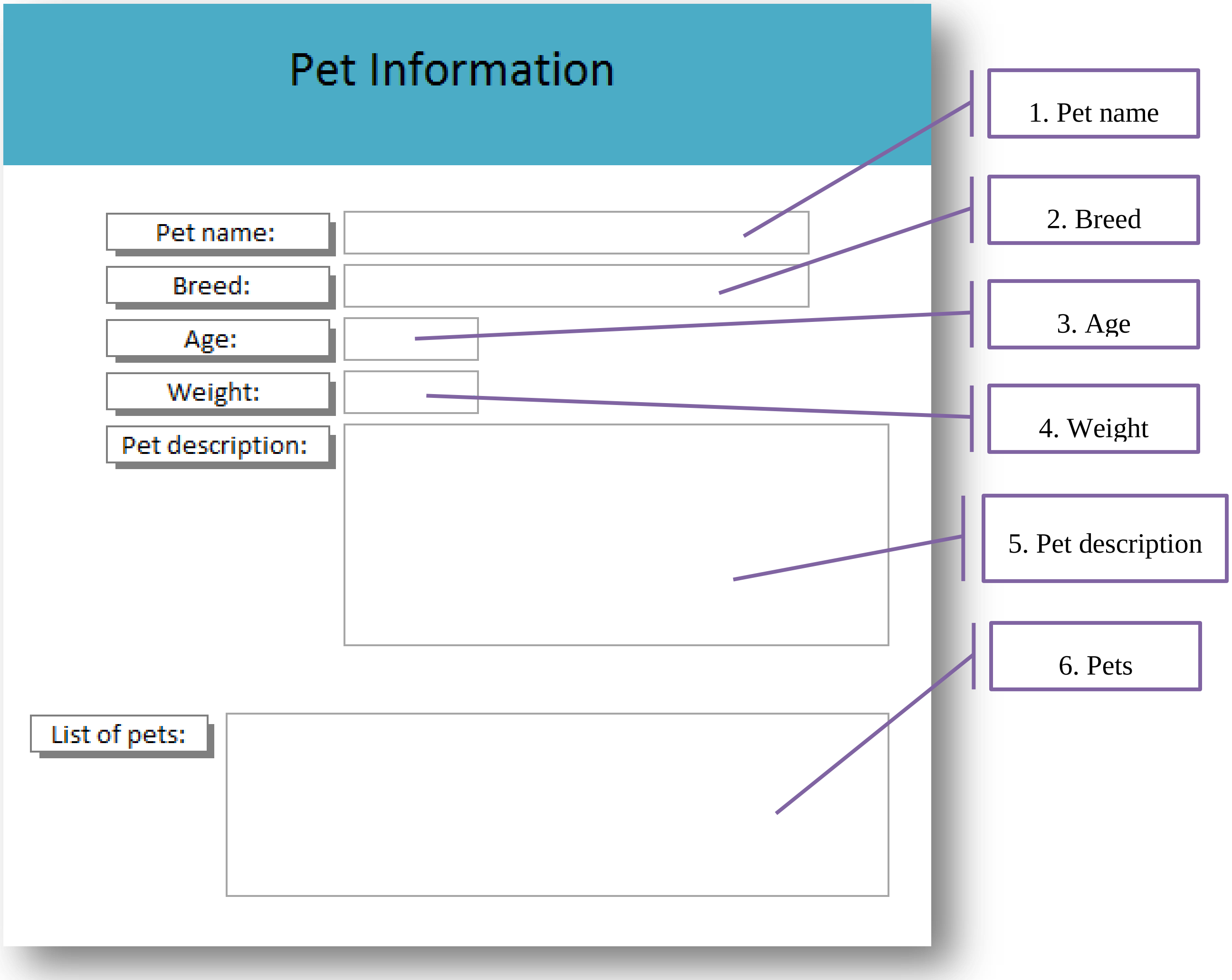
7. Phone number: phone number of owner

8. Email: email of owner

<u>Normalized Relations:</u>

USER(<u>userName</u>, firstName, lastName, street, city, state, zipCode, phoneNumber, email, role)

Pet UI



1. Pet name: name of the pet

2. Breed: pet breed

3. Age: age of the pet

4. Weight: weight of the pet

5. Pet description: description of how the pet looks like

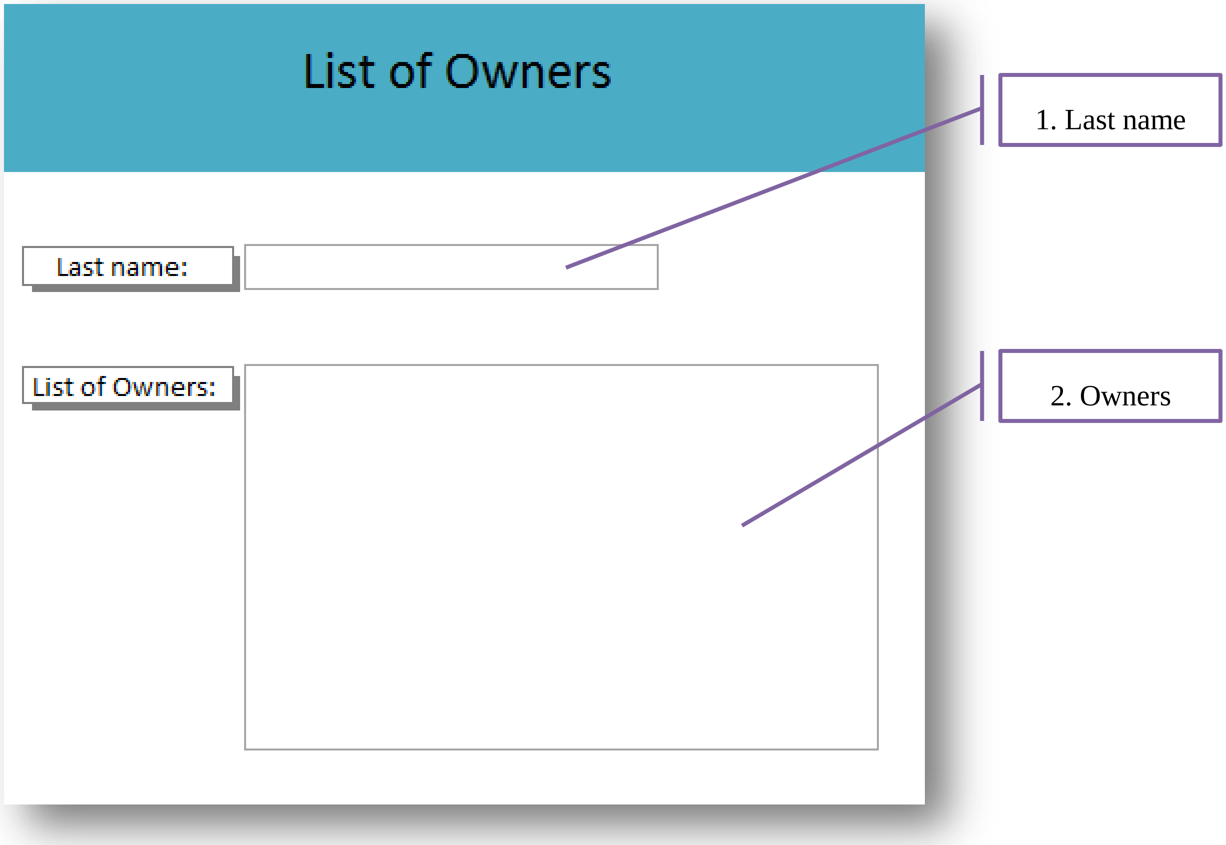
6. Pets: list of the pets for a particular owner

Normalized Relations:

PET(<u>petID</u>, petName, breed, age, weight, petDescription)

USER(userName, role)

Browse Owner UI



1. Last name: last name of the owner

2. Owners: list of all the owners with that last name

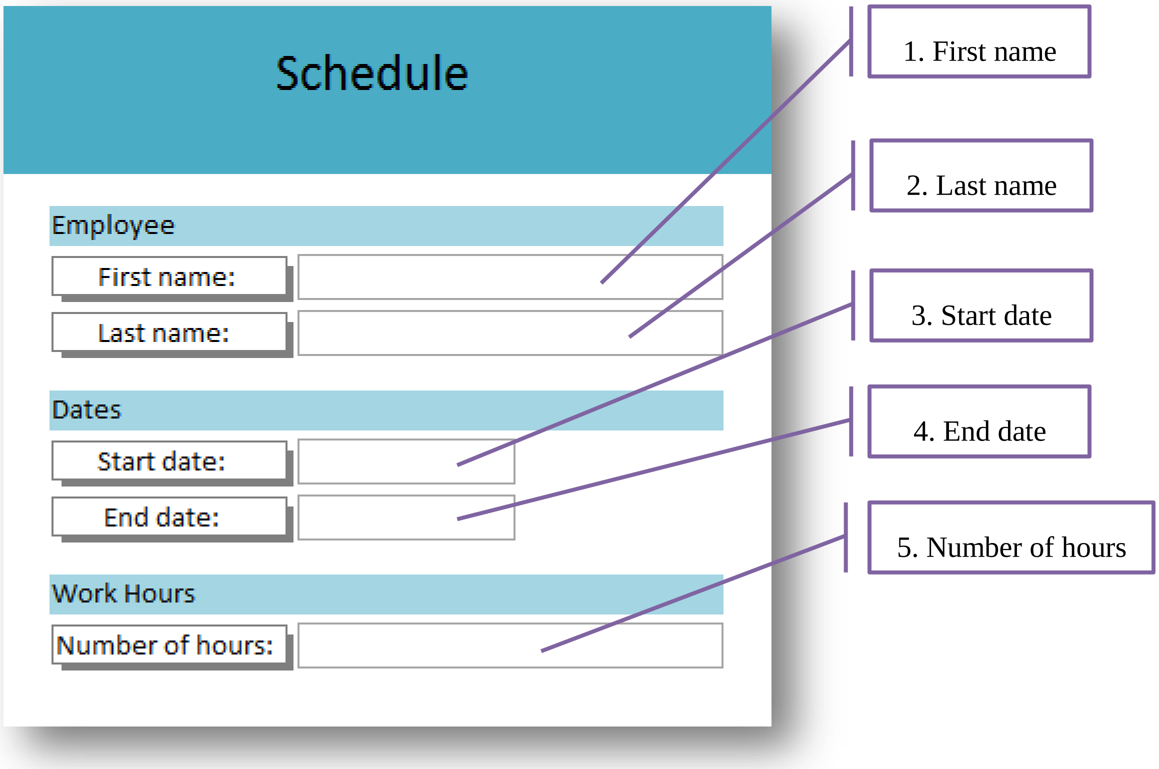
Normalized Relations:

USER(userName, lastName, firstName, role)

EMPLOYEE(userName)

JOB(jobNo, jobTitle)

Schedule UI



1. First name: first name of the employee
2. Last name: last name of the employee
3. Start date: starting date to work
4. End date: ending date of work
5. Number of hours: total number of work hours

Normalized Relations:

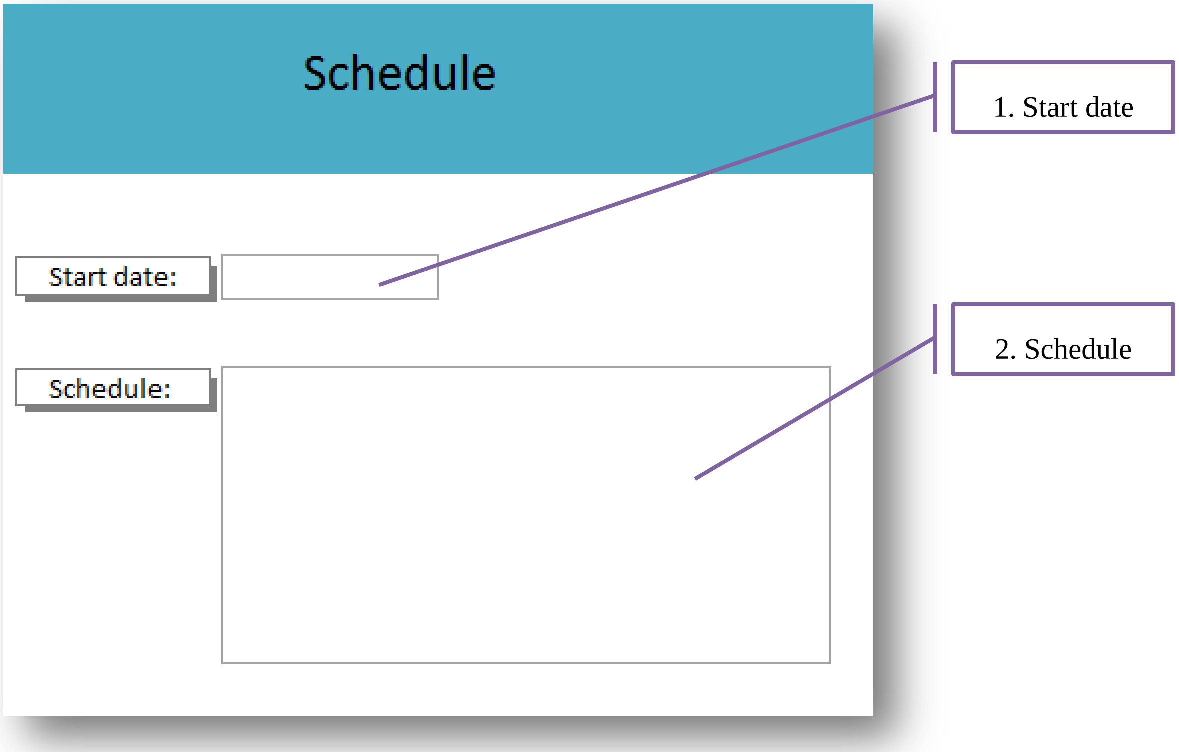
USER(<u>userName</u>, firstName, lastName, role)

SCHEDULE(<u>scheduleNo</u>, startDate, endDate, numberOfHours)

EMPLOYEE(<u>userName</u>)

JOB(<u>jobNo</u>, jobTitle)

Browse Schedule UI



1. Start date: starting date to work

2. Schedule: list of all the pets the employee has to take care of

Normalized Relations:

SCHEDULE(scheduleNo, startDate)

PET(petID, breed, petName)

USER(userName, role)

EMPLOYEE(userName)

JOB(jobNo, jobTitle)

External Data Model

This is a set of normalized relations after merging all the relations from the user views.

Normalized relations:

USER(userName, password, firstName, lastName, street, city, state, zipCode, phoneNumber, email, role)

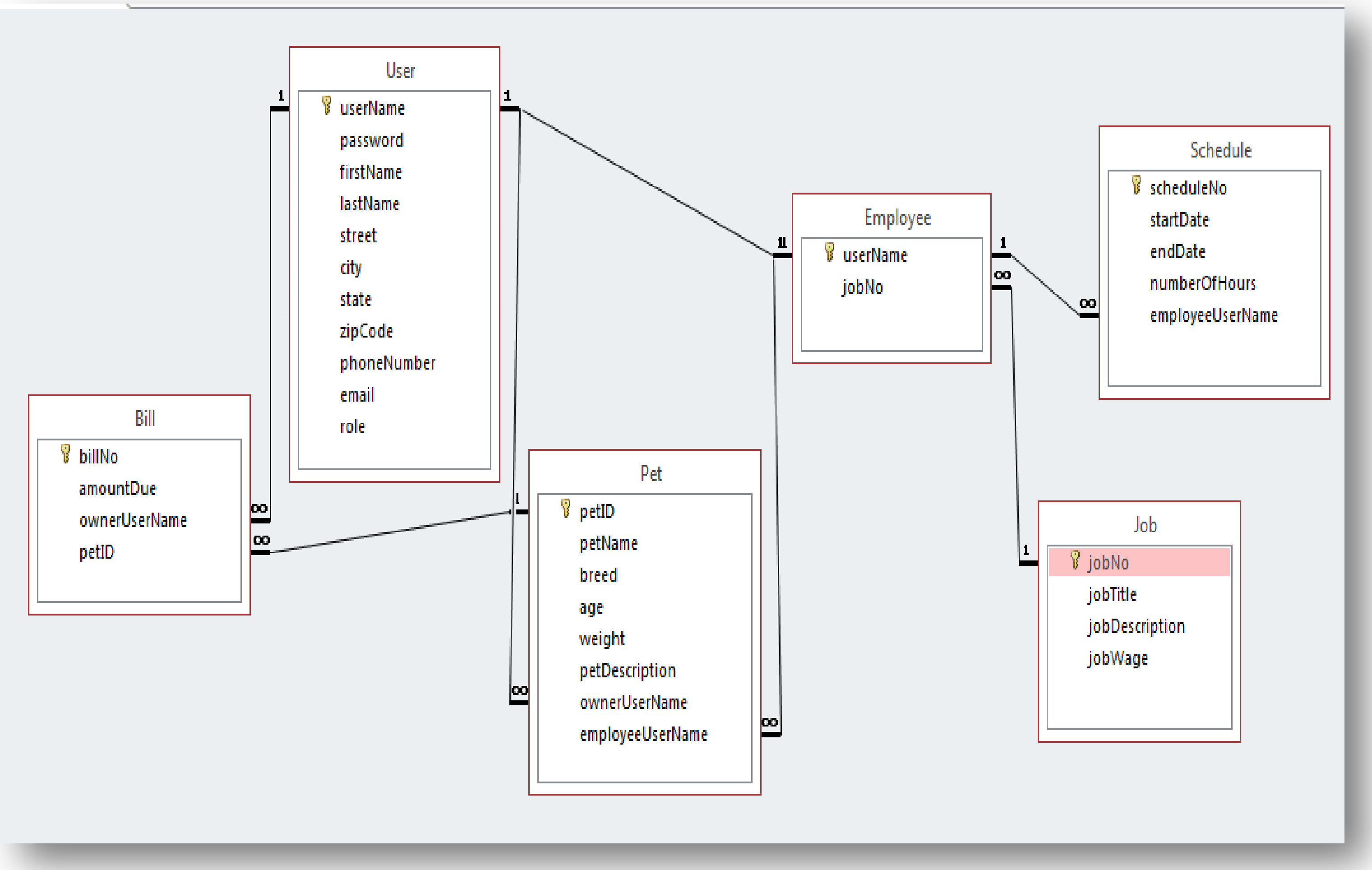
PET(petID, petName, breed, age, weight, petDescription)

SCHEDULE(scheduleNo, startDate, endDate, numberOfHours)

EMPLOYEE(userName)

JOB(jobNo, jobTitle)

Internal Data Model



*Note: After comparing the external and internal data models, it should be noted that not all the attributes in the JOB table are used in the external data model and the BILL table is not used at all as well. Even though these attributes and table are not used, they may be used later on.

Data Model

The following SQL code is to create the Pet table…

```
CREATE TABLE `Pet` (
  `petID` INTEGER NOT NULL AUTO_INCREMENT,
  `petName` VARCHAR(50) NOT NULL,
  `breed` VARCHAR(50) NOT NULL,
  `age` INTEGER NOT NULL DEFAULT 0,
  `weight` DOUBLE NULL NOT NULL DEFAULT 0,
  `petDescription` VARCHAR(255) NOT NULL,
  `ownerUserName` VARCHAR(50) NOT NULL,
  `employeeUserName` VARCHAR(50) NOT NULL,
  INDEX (`petID`),
  PRIMARY KEY (`petID`),
  FOREIGN KEY (`ownerUserName`) REFERENCES User (`userName`),
  FOREIGN KEY (`employeeUserName`) REFERENCES Employee (`userName`)
) ENGINE=myisam DEFAULT CHARSET=utf8;
```

Note: The rest of the tables and data are located in the Appendix.

Data Dictionary

Class	Attribute	Data Type	Description	Key
User	firstName	String	First name of the user.	
	lastName	String	Last name of the user.	
	street	String	Street address of the user.	
	city	String	City where user lives.	
	state	String	State where user lives.	
	zipCode	String	Zip code of the user.	
	phoneNumber	String	Phone number of the user.	
	email	String	Email address of the user.	
	role	String	Role of the user (Owner/Employee).	
	password	String	Password for the user to access the system.	
	userName	String	Unique user name.	PK
Pet	petName	String	Name of the pet.	
	breed	String	Breed type of the pet.	
	age	Integer	Age of the pet.	
	weight	Double	Weight of the pet.	
	petDescription	String	Description of how the pet looks.	
	petID	Integer	Pet ID number.	PK
	ownerUserName	String	Owner user name.	FK

	employeeUserName	String	Employee user name.	FK
Employee	userName	String	Employee user name.	PK(FK)
	jobNo	Integer	Employee job number.	FK
Job	jobDescription	String	Description of the job.	
	jobTitle	String	Title of the job.	
	jobWage	Double	Wage obtained from the job.	
	jobNo	Integer	Unique job number.	PK
Schedule	startDate	Date	Start date of taking care of the pet.	
	endDate	Date	End date of taking care of the pet.	
	numberOfHours	Double	Total number of hours an employee has to work.	
	scheduleNo	Integer	Unique schedule number.	PK
	employeeUserName	String	Employee user name.	FK
Bill	amountDue	Double	Amount that the pet owner owes.	
	billNo	Integer	Unique bill number.	PK
	ownerUserName	String	Owner user name.	FK
	petID	Integer	Pet ID number.	FK

Conclusion

While working on this project, our group learned a lot of things, such as MS Access all the way to learning how to create PHP files. Before this class, no members of our group had ever taken a class that had worked with MYSQL and PHP. So we would frequently get stuck or confused while working on the project. This led us to explore these applications to get a better understanding. We found the project extremely helpful with turning the material we learn in class into a way we would actually use it in the real world. We could potentially sit here and list every little thing we learned, but truthfully, this entire project was new to us. We all definitely learned a lot.

Appendix

Database Creation DDL and Populated Data (SQL File)

```
#
# DUMP FILE
#
# Database is ported from MS Access
#----------------------------------------------------------------
# Created using "MS Access to MySQL" form http://www.bullzip.com
# Program Version 5.3.259
#
# OPTIONS:
#   sourcefilename=C:\Users\Public\Desktop\KennelCare.accdb
#   sourceusername=
#   sourcepassword=
#   sourcesystemdatabase=
#   destinationdatabase=2015-kpingels
#   storageengine=MyISAM
#   dropdatabase=0
#   createtables=1
#   unicode=1
#   autocommit=1
#   transferdefaultvalues=1
#   transferindexes=1
#   transferautonumbers=1
#   transferrecords=1
#   columnlist=1
#   tableprefix=
#   negativeboolean=0
#   ignorelargeblobs=0
#   memotype=LONGTEXT
#

CREATE DATABASE IF NOT EXISTS `2015-kpingels`;
USE `2015-kpingels`;

#
# Table structure for table 'Bill'
#
```

```
DROP TABLE IF EXISTS `Bill`;

CREATE TABLE `Bill` (
  `billNo` INTEGER NOT NULL AUTO_INCREMENT,
  `amountDue` DOUBLE NULL NOT NULL DEFAULT 0,
  `ownerUserName` VARCHAR(50) NOT NULL,
  `petID` INTEGER NOT NULL DEFAULT 0,
  INDEX (`petID`),
  PRIMARY KEY (`billNo`),
  FOREIGN KEY (`ownerUserName`) REFERENCES User (`userName`),
  FOREIGN KEY (`petID`) REFERENCES Pet (`petID`)
) ENGINE=myisam DEFAULT CHARSET=utf8;

SET autocommit=1;

#
# Dumping data for table 'Bill'
#

INSERT INTO `Bill` (`billNo`, `amountDue`, `ownerUserName`, `petID`) VALUES (1, 350, 'jmarble', 6);
INSERT INTO `Bill` (`billNo`, `amountDue`, `ownerUserName`, `petID`) VALUES (2, 500, 'jgibson', 4);
INSERT INTO `Bill` (`billNo`, `amountDue`, `ownerUserName`, `petID`) VALUES (3, 300, 'mtrudeau', 2);
INSERT INTO `Bill` (`billNo`, `amountDue`, `ownerUserName`, `petID`) VALUES (4, 200, 'cpingelski', 7);
INSERT INTO `Bill` (`billNo`, `amountDue`, `ownerUserName`, `petID`) VALUES (5, 850, 'atylock', 5);
INSERT INTO `Bill` (`billNo`, `amountDue`, `ownerUserName`, `petID`) VALUES (6, 150, 'lgibson', 3);
# 6 records

#
# Table structure for table 'Employee'
#

DROP TABLE IF EXISTS `Employee`;

CREATE TABLE `Employee` (
  `userName` VARCHAR(50) NOT NULL,
  `jobNo` INTEGER DEFAULT 0,
  PRIMARY KEY (`userName`),
  FOREIGN KEY (`jobNo`) REFERENCES Job (`jobNo`)
) ENGINE=myisam DEFAULT CHARSET=utf8;

SET autocommit=1;

#
# Dumping data for table 'Employee'
#

INSERT INTO `Employee` (`userName`, `jobNo`) VALUES ('kkenyon', 3);
INSERT INTO `Employee` (`userName`, `jobNo`) VALUES ('mdenney', 1);
INSERT INTO `Employee` (`userName`, `jobNo`) VALUES ('kpingelski', 1);
INSERT INTO `Employee` (`userName`, `jobNo`) VALUES ('pgibson', 2);
INSERT INTO `Employee` (`userName`, `jobNo`) VALUES ('mgowett', 3);
INSERT INTO `Employee` (`userName`, `jobNo`) VALUES ('meagan', 3);
# 6 records

#
# Table structure for table 'Job'
#

DROP TABLE IF EXISTS `Job`;

CREATE TABLE `Job` (
  `jobNo` INTEGER NOT NULL AUTO_INCREMENT,
  `jobTitle` VARCHAR(50) NOT NULL,
  `jobDescription` VARCHAR(255) NOT NULL,
  `jobWage` DOUBLE NULL NOT NULL DEFAULT 0,
  PRIMARY KEY (`jobNo`)
) ENGINE=myisam DEFAULT CHARSET=utf8;

SET autocommit=1;

#
# Dumping data for table 'Job'
#
```

```
INSERT INTO `Job` (`jobNo`, `jobTitle`, `jobDescription`, `jobWage`) VALUES (1, 'Manager', 'in charge of company', 40);
INSERT INTO `Job` (`jobNo`, `jobTitle`, `jobDescription`, `jobWage`) VALUES (2, 'Secretary', 'keeps records and answers phone', 20);
INSERT INTO `Job` (`jobNo`, `jobTitle`, `jobDescription`, `jobWage`) VALUES (3, 'Caretaker', 'takes care of animals', 25);
# 3 records

#
# Table structure for table 'Pet'
#

DROP TABLE IF EXISTS `Pet`;

CREATE TABLE `Pet` (
  `petID` INTEGER NOT NULL AUTO_INCREMENT,
  `petName` VARCHAR(50) NOT NULL,
  `breed` VARCHAR(50) NOT NULL,
  `age` INTEGER NOT NULL DEFAULT 0,
  `weight` DOUBLE NULL NOT NULL DEFAULT 0,
  `petDescription` VARCHAR(255) NOT NULL,
  `ownerUserName` VARCHAR(50) NOT NULL,
  `employeeUserName` VARCHAR(50) NOT NULL,
  INDEX (`petID`),
  PRIMARY KEY (`petID`),
  FOREIGN KEY (`ownerUserName`) REFERENCES User (`userName`),
  FOREIGN KEY (`employeeUserName`) REFERENCES Employee (`userName`)
) ENGINE=myisam DEFAULT CHARSET=utf8;

SET autocommit=1;

#
# Dumping data for table 'Pet'
#

INSERT INTO `Pet` (`petID`, `petName`, `breed`, `age`, `weight`, `petDescription`, `ownerUserName`, `employeeUserName`) VALUES (2, 'Molly', 'Bengal', 5, 30, 'brown', 'mtrudeau', 'mgowett');
INSERT INTO `Pet` (`petID`, `petName`, `breed`, `age`, `weight`, `petDescription`, `ownerUserName`, `employeeUserName`) VALUES (3, 'Shay', 'Tortoise Shell Care', 1, 10, 'black', 'lgibson', 'mgowett');
INSERT INTO `Pet` (`petID`, `petName`, `breed`, `age`, `weight`, `petDescription`, `ownerUserName`,
`employeeUserName`) VALUES (4, 'Zues', 'Dashhound', 7, 25, 'long ears', 'jgibson', 'kkenyon');
INSERT INTO `Pet` (`petID`, `petName`, `breed`, `age`, `weight`, `petDescription`, `ownerUserName`,
`employeeUserName`) VALUES (5, 'Sam', 'Great Dane', 7, 150, 'giant brown spot', 'atylock', 'kkenyon');
INSERT INTO `Pet` (`petID`, `petName`, `breed`, `age`, `weight`, `petDescription`, `ownerUserName`,
`employeeUserName`) VALUES (6, 'Maggie', 'Border Collie', 3, 60, 'black and white', 'jmarble',
'meagan');
INSERT INTO `Pet` (`petID`, `petName`, `breed`, `age`, `weight`, `petDescription`, `ownerUserName`,
`employeeUserName`) VALUES (7, 'Cassie', 'Bombay', 2, 15, 'white paws', 'cpingelski', 'meagan');
# 6 records

#
# Table structure for table 'Schedule'
#

DROP TABLE IF EXISTS `Schedule`;

CREATE TABLE `Schedule` (
  `scheduleNo` INTEGER NOT NULL AUTO_INCREMENT,
  `startDate` DATETIME NOT NULL,
  `endDate` DATETIME NOT NULL,
  `numberOfHours` DOUBLE NULL NOT NULL DEFAULT 0,
  `employeeUserName` VARCHAR(50),
  INDEX (`numberOfHours`),
  PRIMARY KEY (`scheduleNo`),
  FOREIGN KEY (`employeeUserName`) REFERENCES Employee (`userName`)
) ENGINE=myisam DEFAULT CHARSET=utf8;

SET autocommit=1;

#
# Dumping data for table 'Schedule'
#

INSERT INTO `Schedule` (`scheduleNo`, `startDate`, `endDate`, `numberOfHours`,
`employeeUserName`) VALUES (1, '2015-11-01 00:00:00', '2015-11-07 00:00:00', 45, 'kpingelski');
INSERT INTO `Schedule` (`scheduleNo`, `startDate`, `endDate`, `numberOfHours`,
`employeeUserName`) VALUES (2, '2015-11-01 00:00:00', '2015-11-01 00:00:00', 45, 'mdenney');
INSERT INTO `Schedule` (`scheduleNo`, `startDate`, `endDate`, `numberOfHours`,
`employeeUserName`) VALUES (3, '2015-11-01 00:00:00', '2015-11-07 00:00:00', 40, 'pgibson');
```

```
INSERT INTO `Schedule` (`scheduleNo`, `startDate`, `endDate`, `numberOfHours`,
`employeeUserName`) VALUES (4, '2015-11-01 00:00:00', '2015-11-07 00:00:00', 20, 'meagan');
INSERT INTO `Schedule` (`scheduleNo`, `startDate`, `endDate`, `numberOfHours`,
`employeeUserName`) VALUES (5, '2015-11-01 00:00:00', '2015-11-07 00:00:00', 35, 'mgowett');
INSERT INTO `Schedule` (`scheduleNo`, `startDate`, `endDate`, `numberOfHours`,
`employeeUserName`) VALUES (6, '2015-11-01 00:00:00', '2015-11-07 00:00:00', 35, 'kkenyon');
# 6 records

#
# Table structure for table 'User'
#

DROP TABLE IF EXISTS `User`;

CREATE TABLE `User` (
  `userName` VARCHAR(50) NOT NULL,
  `password` VARCHAR(50) NOT NULL,
  `firstName` VARCHAR(50) NOT NULL,
  `lastName` VARCHAR(50) NOT NULL,
  `street` VARCHAR(50) NOT NULL,
  `city` VARCHAR(50) NOT NULL,
  `state` VARCHAR(2) NOT NULL,
  `zipCode` VARCHAR(10) NOT NULL,
  `phoneNumber` VARCHAR(20) NOT NULL,
  `email` VARCHAR(40) NOT NULL,
  `role` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`userName`),
  INDEX (`zipCode`)
) ENGINE=myisam DEFAULT CHARSET=utf8;

SET autocommit=1;

#
# Dumping data for table 'User'
#

INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('mgowett', 'Elmira11', 'Mikayla', 'Gowett', '140
Ridge Rd', 'Oswego', 'NY', '13126', '3155550265', 'mgowett@aol.com', 'Employee');
```

```
INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('mtrudeau', 'Mech11', 'Mackenzie', 'Trudeau', '145
Wayne St', 'Oswego', 'NY', '13126', '3155201259', 'mtrudeau@aol.com', 'Owner');
INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('lgibson', 'Shay5', 'Lindsey', 'Gibson', '15 Main St',
'Oswego', 'NY', '13126', '3158456525', 'lgibson@nycap.rr.com', 'Owner');
INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('jmarble', 'Love3', 'John', 'Marble', '121 Cayuga St',
'Oswego', 'NY', '13126', '3156542565', 'jmarble@nycap.rr.com', 'Owner');
INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('meagan', 'Dog21', 'Morgan', 'Eagan', '150 Cayuga
St', 'Oswego', 'NY', '13126', '3155552002', 'meagan@aol.com', 'Employee');
INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('pgibson', 'House25', 'Patty', 'Gibson', '10 Erie St',
'Oswego', 'NY', '13126', '3156523656', 'pgibson@gmail.com', 'Employee');
INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('kpingelski', 'Soccer17', 'Katelyn', 'Pingelski', '17
Hillside Ave', 'Oswego', 'NY', '13126', '3156526526', 'kpingelski@nycap.rr.com', 'Employee');
INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('mdenney', 'Madison11', 'Matthew', 'Denney', '15
Hillside Ave', 'Oswego', 'NY', '13126', '3156526654', 'mdenney@aol.com', 'Employee');
INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('atylock', 'Bella34', 'Ange', 'Tylock', '234 Main St',
'Oswego', 'NY', '13126', '3152527812', 'atylock@gmail.com', 'Owner');
INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('cpingelski', 'Basketball32', 'Chris', 'Ping', '345
Wayne St', 'Oswego', 'NY', '13126', '3156201212', 'cping@gmail.com', 'Owner');
INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('jgibson', 'Mont17', 'John', 'Gibson', '887 Ridge
Rd', 'Oswego', 'NY', '13126', '3159634455', 'jgibson@gmail.com', 'Owner');
INSERT INTO `User` (`userName`, `password`, `firstName`, `lastName`, `street`, `city`, `state`,
`zipCode`, `phoneNumber`, `email`, `role`) VALUES ('kkenyon', 'Allie7', 'Kaile', 'Kenyon', '126 Main St',
'Oswego', 'NY', '13126', '3152601133', 'kkenyon@gmail.com', 'Employee');
# 12 records
```