

# Project Step 3 Draft

Anneliese Doerfler  
Anphong Nguyen  
Ryan Little  
Rami Hansen

## 1) Overview

Users have virtual catteries in which they own cats. A user can choose to open their cattery up to other users by sharing the unique cattery id. Each cat within the cattery has attributes that are affected by their owner's actions. feed\_stat, groom\_stat, and play\_stat are changed based on how the user interacts with their cats. Users can only interact with cats that they own. The number of cats a cattery can contain is capped at thirty. Users are capped at association with five catteries.

## 2) Rules & Requirements

### Requirements:

- Users login with their username and password
- Logged in users can select which cattery they would like to join during the session. Users can switch to a different cattery at any point.
- User create catteries that have a private "cattery ID" that acts as a password for that cattery
- Users can share cattery ids with other users to share ownership so that multiple people can raise their own cats in a shared cattery
- Users can join other user's catteries if they have the cattery id. After they join, they are given "visitor" status, being able to raise cats, but not delete or rename the cattery
- When a new user joins a cattery, a new cat isn't produced, they need to wait for an existing user to spawn a litter from their own cat. The new user can then claim one of them.
- Users can pull up a list of their owned cats and what cattery they are in
- Users can claim a newborn, unclaimed cat with a cattery
  - A user does not own a cat within a cattery until they claim an unclaimed kitten
- Users can feed, groom, and play with cats by clicking on their respective icons which will increments the stats below the cats image
- Users can choose to remove a cat they own from a cattery, deleting it
- Users can rename a cat that they own
- A cat "ages" by filling its three stats (feed, groom, play)
  - After the three stats are filled the first time, a cat changes from kitten to adult
- Adult cats spawn kittens each time their stats are filled (feed, groom, play)
- Cats stats are reset to zero when they become adults or spawn a kitten
- The color of a kitten and its image is generated when it is born

- After spawning a kitten, the owner can decide to keep one or more of the kittens or leave them unclaimed for other players to claim
- When a new kitten is spawned, its stats are initialized to 0 and name is set to "Kitten"

#### **Business Rules:**

- Cats can live within only one Cattery
- A Cattery can house up to 30 cats
- Each cat can only have one user as its owner
- User can only adopt a kitten that has not yet been claimed
- User's can't interact with the application outside of viewing before logging in
- User's can't adopt cats in a cattery unless they have the cattery ID
- An owner may have access to a cattery, but does not have to own a cat
- User can be associated with up to 5 catteries
- If a user deletes every cat in their cattery, then the cattery is deleted
- Users can only delete cats that they own
- Only the owner of a cat can feed, groom, or play with it
- A cattery must have one "owner" user and may have multiple "visitor" users
- When a cattery owner leaves a cattery they own, they can select another user to become the new owner

### **3) Database Outline, in Words**

#### **Entities and Attributes:**

##### **Cattery**

- Name - varchar
- Cattery\_id - unique

##### **Cats**

- Cat\_id - unique
- Name - varchar
- Feed\_stat - numeric (0-3)
- Groom\_stat - numeric (0-3)
- Play\_stat - numeric (0-3)
- Owner\_id(FK) ->username
- Age (Boolean where T-adult F-kitten)

##### **Users**

- Username - varchar, unique
- Password - hashed
- user\_name

##### **Color**

- description - varchar
- Color\_id - numeric, unique
- File Path - varchar

## Potential Relationships

### Cattery has user - many to many

- Whether they own or were invited to the cattery

### User has Cat - one to many

- Might include when the relationship was created

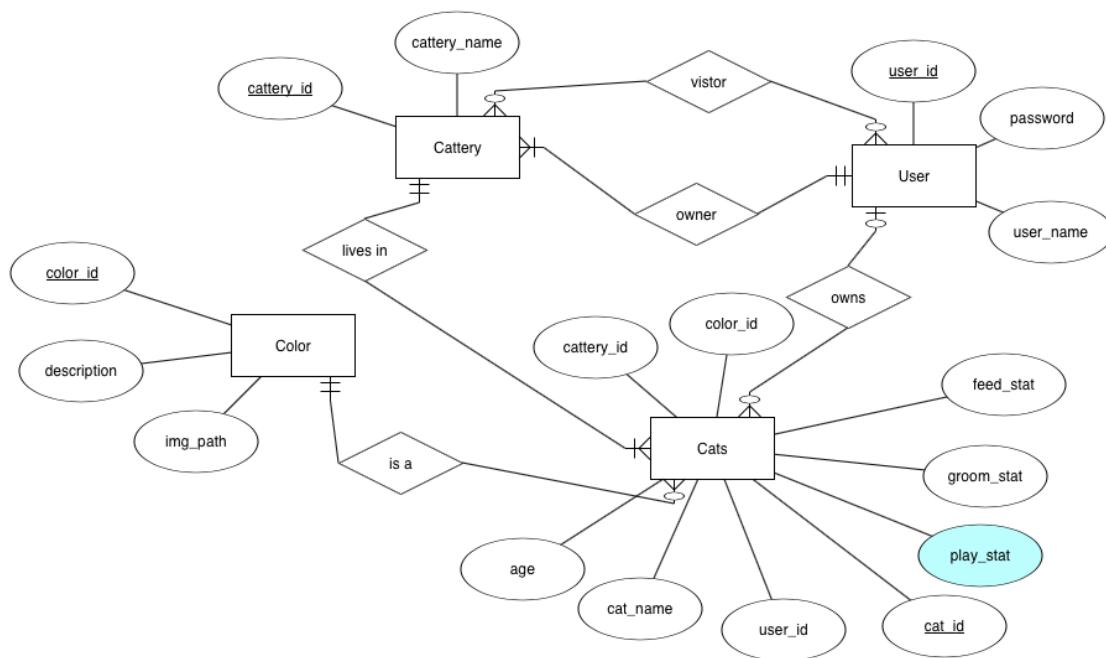
### Cattery has Cat - one to many

- A cat must live with one Cattery, but a Cattery can house many cats
- The cat table has a foreign key catteryID linking it to a specific cattery

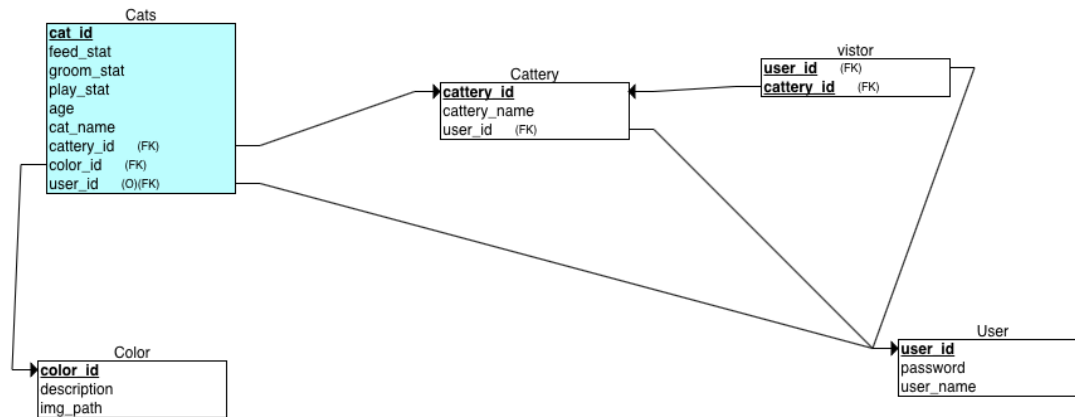
### Cat has Color - One to many

- The cat table contains a foreign key to color name that links it to an entry in the Color Image table

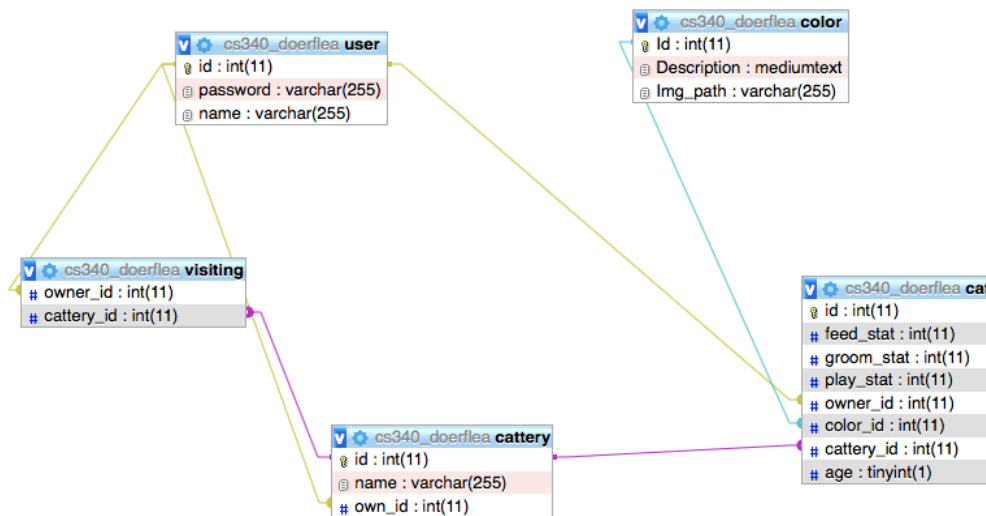
## ER Diagram:



## Relational Schema:



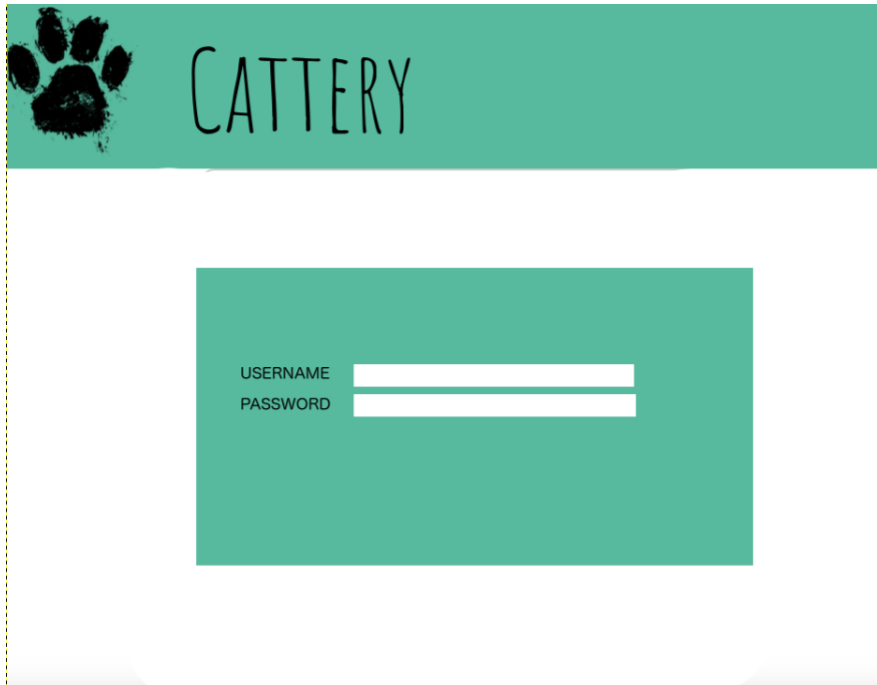
## Table Structure:



## Pages:

### Login Page

---

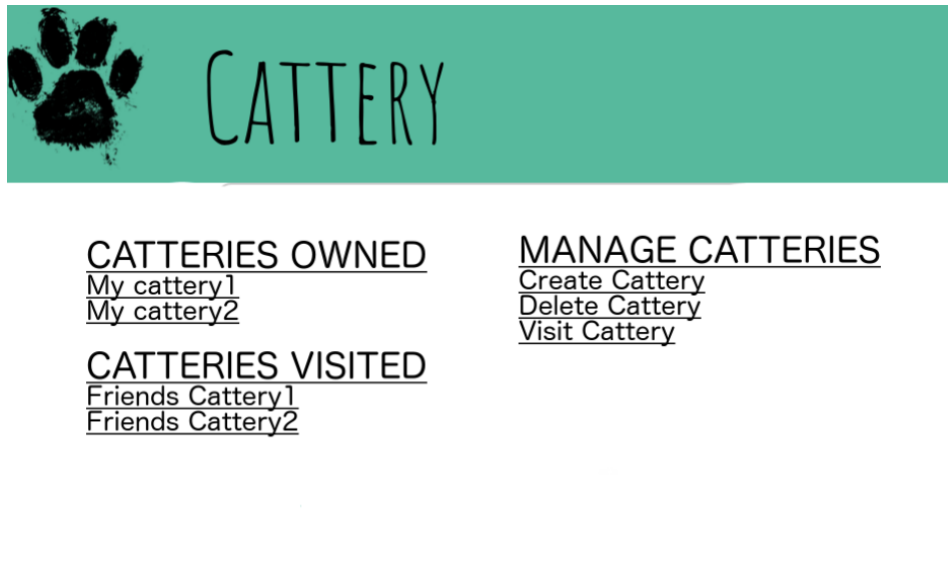
The image shows a web page for a cattery. At the top, there is a teal header bar. On the left side of the header is a black paw print icon, and on the right side, the word "CATTERY" is written in a black, hand-drawn, sans-serif font. Below the header, the main content area is white. In the center of this area is a teal rectangular box. Inside this box, there are two white input fields. The first field is preceded by the label "USERNAME" in a small, black, sans-serif font. The second field is preceded by the label "PASSWORD" in the same font. The input fields are simple white rectangles with thin black borders.

```
SELECT id
FROM user
WHERE name=$name_input AND password=$password_input
```

The login page allows the user to login into their account with their username and password. SQL is used on this page to verify login information. From the application, the user will provide a username and password. The user name is used to query into the database and return the user\_id attribute of any field where it finds a match with username and password

## UserHome Page

---



```
SELECT `name` FROM `cattery` WHERE `owner_id` = $current_user_id
SELECT `name` FROM `cattery` WHERE `owner_id` IN (SELECT owner_id FROM visiting
                                              WHERE owner_id=$current_user_id)
```

The home page allows the user to view catteries they own, catteries in which they are visiting owners, and manage their catteries. This page uses SQL by querying the cattery table to find all catteries owned by the user and then checks the visiting table to find all catteries that the user has a visiting relationship with.

## Create Cattery Page

---

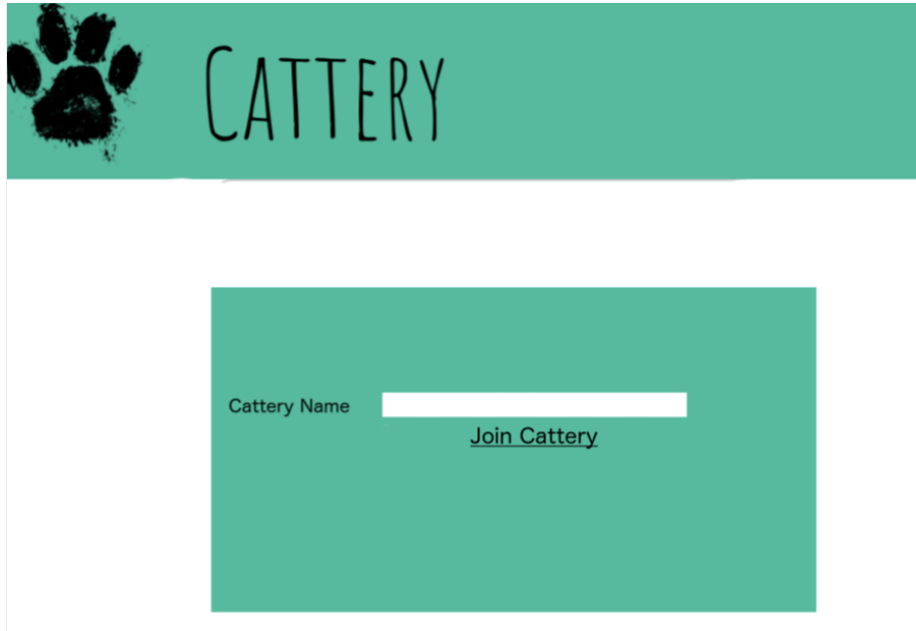
A teal rectangular box containing a form. On the left, the text "Cattery Name" is followed by a white rectangular input field. Below the input field is a button with the text "CREATE CATTERY" in a small, black, underlined font.

The create cattery page allows the user to create a new cattery (users are allowed to create up to five).

```
INSERT INTO `cattery`  
  (`name`, `own_id`)  
VALUES  
  ($creator_user_name, $cattery_name_input);
```

## Join Cattery Page

---

The image shows a web page for joining a cattery. At the top, there is a teal banner with a black paw print icon on the left and the word "CATTERY" in a large, black, hand-drawn font on the right. Below the banner, there is a teal rectangular box containing a form. The form has a label "Cattery Name" followed by a white text input field. Below the input field, there is a teal button with the text "Join Cattery" in a black, hand-drawn font.

```
INSERT INTO `visiting`  
  (`owner_id`, `cattery_id`)  
VALUES  
  ($current_user_id, $cattery_id);
```

The join cattery page allows the user to join another user's cattery as long as they have the cattery's id.



## Cattery Cat List

---



My cattery1

Felix

Lucy

Cheddar

Toast

Butter

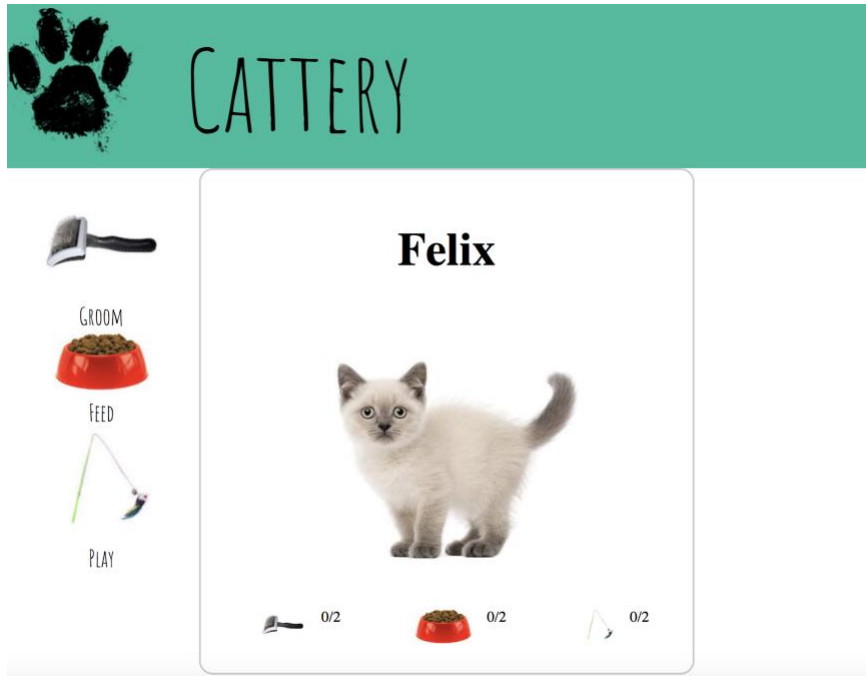
Ralph



```
SELECT * FROM `cat` WHERE `cattery_id` = $current_cattery;
```

The cattery cat list page allows the user to view a list of all the cats contained within a cattery. When a user selects a cattery, all cats whose cattery id matches that cattery will be pulled up as a list.

## Cat Interaction Page



The cat interaction page allows the user to groom, feed, and play with the cats within a specific cattery.

The SQL queries as the user interacts the cat and increments that cat will be similar to:

```
UPDATE Cat
SET play-stat = play-stat + 1
WHERE id=current-cat-id;
```

with an update query for each stat (feed, groom, play)

When a kitten is generated and a cat is created, the SQL query is similar to:

```
INSERT INTO Cat VALUES (
Owner-id, Color-id, Cattery-id);
```

