Phase 2 Report

Xining Li, Yi Zhao, Xiaoyan Liu, Yaguang Chen

GaTech

March 2, 2020

Abstract

This is the EER to Relational Mapping report for Team080

1 Table of Contents

2 Abstract Code

2.1 Login

Abstract Code

- User enters **email**, **password** input fields.
- If data validation is successful for both username and password input fields, then:
- When *Enter* button is clicked
  - If User record is found and user entered password match the username's key associated password in the **User** table: Store login information as session variable '$UserID', and go to the **Animal Dashboard**
  - Else: Go back to the **Login**
- SQL

```
SELECT password FROM LoginUser WHERE email='$enteredEmail';
```

2.2 Animal Dashboard

Abstract Code

- Show the animal's name, species, breed, sex, alteration status, age, and adoptability status by the result of the corresponding SQL query.

```
CREATE VIEW [ANIMALINFO] AS (
With pet_adoption_status AS (
SELECT
    animal.pet_id,
    CASE WHEN adoption.pet_id is NOT NULL THEN 'adopted' ELSE 'not adopted' END AS
        adoptability_status
FROM Animal AS animal
LEFT JOIN AdoptionInformation  AS adoption
    ON animal.pet_id = adoption.pet_id)

 SELECT
    pet_name,
    species,
    sex,
    alteration_status,
    age,
    adoptability_status,
    GROUP_CONCAT(DISTINCT AnimalBreed.breed ORDER BY AnimalBreed.breed SEPARATOR '\')
        as breed
FROM Animal
INNER JOIN AnimalBreed on Animal.pet_id = AnimalBreed.pet_id
 INNER JOIN Breed on Breed.breed = AnimalBreed.breed
 INNER JOIN pet_adoption_status on Animal.pet_id = pet_adoption_status.pet_id
 GROUP BY 1,2,3,4,5,6);
```

- Populate species and adoptability status dropdowns, if no buttons are pushed, do nothing. If clicking on sepcies and/or adopatability status, query corresponding animals and display them on dashboard only.

```
 SELECT
    *
FROM ANIMALINFO
WHERE species = '$species' or adoptability_status = '$adoptability_status'
```

- Upon:
  - Clicking on each column would pop out 2 choices: sort in increasing order, sort in decreasing order.
  - Clicking on the animal's name will go to the **Animal Detail**'s Servlet (implemented using RestFul API GET method).
- Show the number of available spaces.

```
With current_num AS (
SELECT species, count(distinct pet_id) as curr_count
FROM ANIMALINFO
GROUP BY species)

 SELECT
     species,
    (limit_num - IFNULL(curr_count, 0)) AS available_space
FROM Species
LEFT JOIN current_num ON Species.species = current_num.species;
```

- if the user has appropriate permission (fetched by the corresponding SQL query), an *Add Animal* button will show directing to the **Add Animal** screen.
- if the user has appropriate permission (fetch by the corresponding SQL query), an *Add Adoption Application* button will show directing to the **Add Adoption Application** screen.

2.3 Add Animal

Abstract Code

- Process user's post request and convert to a data structure (for example Json or POJO)
- Validate user input, if valid continue, else return user the error message.
- Acquire permit from the semaphore
- TRY:
  - Parse the animal's species from the user's post request
  - If the animal's species from the user's post request is exist in the database and the number of availability associated with the animal's species is greater than 0:
    - Generate the unique petId by accessing the **Animal** table and pass to the setter of the petId field of the upper-mentioned data structure.
    - Submit the data structure to the database.

      ```
      INSERT INTO Animal (pet_name, sex, age, alteration_status,
          descriptions, microchipId)
      VALUES ('$pet_name', '$sex', '$age', '$alteration_status',
          '$descriptions', 'inge');
      ```

    - Take the user to the **Animal Detail** Screen.
  - Elae:
    - Return the user an error message with the corresponding error code.
- Catch Exception:
  - Log the error message and return the error message to the user.
- Finally:
  - Rlease the permit to the semaphore.

2.4 Animal Detail

- Show animal's all details (attributes) by the result of the corresponding SQL query.
- Show a "Vaccinations" section which shows any Vaccination history.
- Show *Add Vaccination* button if the corresponding animal has not been adopted.
- Show *Adopt pet* button if the animal is eligible for adoption and if the session user has the permission.

Abstract Code

AnimalDetailServlet

Background: Animal Dashboard's restful API takes the user to the corresponding Animal Detail's front end, and the Animal Detail's front end call the AnimalDetailServlet using the GET method.

And here is the implementation of the AnimalDetailServlet's get method.

- the AnimalDetailServlet query the database with the restful api's parameter (**PetId**) and convert to a POJO
- the AnimalDetailServlet return the Json converted from the POJO to the front end.
- the AnimalDetailServlet call the AddVaccinationServlet using the GET method and return the user the vaccination that can be implemented.
- (frontend code) the user can optionally call the AddVaccinationServlet using the POST method.

AddVaccinationServlet

Background: Animal Dashboard's restful API takes the user to the corresponding Animal Detail's front end, and the Animal Detail's front end call the AddVaccinationServlet using the GET method.

```
public class Vaccination {
   @NotNull Enum VaccinationType;
   @NotNull Date date;
   @NotNull Date nextDoseDate;
   Long vaccineTagNumber;
}

public class AddVaccinationServlet extends HttpServlet {
   Set<VaccinationType> Get(Animal animal){
      synchronized (animal.getSingleton()){
         if (!animal.eligibleForAdoption()){
         return null;
         } else {
            return animal
               .getSpecies()
               .getRequiredVaccinations()
               .setSubtraction(animal
                  .implementedVaccinations
               );
         }
      }
   }
   void Post(Set<Vaccination> userImplementations) {
      synchronized (animal.getSingleton()){
         for (Vaccination v: userImplementations) {
            /*
            implementVaccination talks to
            the database and put the corresponding
            vaccination into the animal's
            corresponding vaccination table.
            */
            DataBaseSingleton.get(animal)
            .implementVaccination(v);
          }
        }
      }
    }
```

Here is the implementVaccination code involk the code below.

```
INSERT INTO AnimalVaccine (vaccine_type, username, date_administered, expiration_date,
    vaccination_number)
VALUES ($vaccine_type, $username, $date_administered, $expiration_date, $vaccination_number);
```

2.5 Vaccinations

Abstract Code

- Populate vaccinations dropdown lists

```
  SELECT distinct vaccine_type FROM Animal INNER JOIN AnimalBreed ON Animal.pet_id =
      AnimalBreed.pet_id INNER JOIN Breed ON AnimalBreed.breed = Breed.breed INNER JOIN
      VaccineSpecies ON VaccineSpecies.species = Breed.species WHERE Animal.pet_id =
      '$pet_id' ;
```

- When submit button is not clicked, nothing happened
- When submit button is clicked:
  - If vaccine is not chosen: catch Exception, log error message and return the error message to the user;
  - If vaccine is chosen, not both vaccination date or next does date are entered: catch Exception, log error message to the user;
  - Else: update AnimalVaccine table

```
     INSERT INTO AnimalVaccine (pet_id,
       vaccine_type ,
       username,
       date_administered,
       expiration_date,
       vaccination_number ) VALUES( '$pet_id', '$vaccine_type', '$username',
           '$date_administered', '$expiration_date', '$vaccination_number' ) ;
```

- If submit successfully, Display **Animal Detail**

2.6 Adoption

Abstract Code

- Show search dialog where user can search both applicant's last name and co-applicant's last name
- Query and display Applicant's contact information

```
  SELECT applicant_first_name, applicant_last_name, coapplicant_first_name,
      coapplicant_last_name, street, city, state, zipcode, phone_number,
      application_date, status FROM ApplicationInformation WHERE status = 'approved' and
      ( applicant_last_name='ApplicantLastName' OR
      coapplicant_last_name='ApplicantLastName');
```

- If select Adopter, show pop up for adoption date and adoption fee

- If enter adoption date and adoption fee, update **Adoption Information** table, display **Animal Detail**

```
INSERT INTO AdoptionInformation (pet_id, application_number, adoption_date,
    adoption_fee) ) VALUES( '$pet_id', '$application_number', '$adoption_date',
    '$adoption_fee' ) ;
```

- If cancellation, display **Animal Detail**

2.7 Add Adoption Application

Abstract Code

- Show screen to let users enter applicant information including Application first name and last name, Address (street, city, state, zip code), phone number, email address, Date of Application, (optional) co-applicant first name and last name
- When click submit button:
  - If at least one contact information is not entered: catch Exception, log error message and return the error message to the user;
  - Else: Add **Contact Information**

    ```
    INSERT INTO ApplicationInformation (applicant_first_name, applicant_last_name,
        coapplicant_first_name, coapplicant_last_name, street, city, state,
        zipcode, phone_number, application_date, status)
        VALUES('$application_first_name', '$applicant_last_name',
        '$coapplicant_first_name', '$coapplicant_last_name', '$street', '$city',
        '$state', '$zipcode', '$phone_number', '$application_date', 'pending
        approval'
    );
    ```

  - display application ID generated by system

    ```
    SELECT MAX(application_number) FROM ApplicationInformation ;
    ```

- When submit button is not clicked, nothing happen

2.8 Adoption Application Review

Abstract Code

- Find applicationID with pending approval, display applicationID and status

```
SELECT application_number, status FROM ApplicationInformation
ORDER BY application_number ASC;
```

- Update status with dropdown, approved or rejected, Save application status to Contact Information table

```
UPDATE ApplicantInformation
SET status=$status
```

2.9 Animal Control Report

Abstract Code

- SuperUser clicked on *Animal Control Report* button from **Animal Dashboard**:
- Run the **Animal Control Report** task: query for animals' information.

  - Select month and find the animals that was surrendered by Animal Control in selected month;
  - Sort by Pet ID ascending;
  - Display animal's information;

```
SELECT pet_id, surrender_id, animalcontrol, surrender_reason,
 MONTH(surrender_date) FROM Surrender
WHERE MONTH(surrender_date) = selected_month AND animalcontrol = $animalcontrol
ORDER BY pet_id ASC;
```

  - Find the animals that was adopted in selected month;
  - Find these adopted animal's adopted date and surrendered date;
  - Display animal's information if sheltered dates are greater or equal to 60 days in selected month:

```
SELECT pet_name, adoption_date, surrender_date FROM ((Animal
INNER JOIN Surrender ON Animal.pet_id =Surrender.pet_id)
INNER JOIN AdoptionInformation ON Animal.pet_id = AdoptionInformation.pet_id)
WHERE ((adoption_date - surrender_date) > =60) AND
MONTH(surrender_date) = $month;
```

2.10 Volunteer of the Month

Abstract Code

- SuperUser clicked on *Volunteer of the Month* button from **Animal Dashboard**:
- Run the **Volunteer of the Month** task: query for information and volunteered hours about the volunteers.
  - Find all volunteers that has volunteered for selected month;
  - Calculate the total volunteered hours for each of these volunteers;
  - Sort the list of volunteers by descending total hours volunteered for selected month and year;
  - Display first 5 volunteers' first name, last name, email address and total volunteered hours in that month

```
SELECT TOP 5 SUM(hoursWorked)(first_name, last_name, email_address,
SUM(hoursWorked)
FROM LoginUser INNER JOIN VolunteerWorkHours
ON LoginUser.username = VolunteerWorkHours.username
GROUP BY username
ORDER BY SUM(hoursWorked) DESC)
WHERE MONTH(dateWorked) = $selected_ month;
```

2.11 Monthly Adoption Report

Abstract Code

- **find the adoption** info for this month
- **find the surrender** info for this month
- **find the animal** info according to adoption and surrender
- group by species, breed and count total
  - order by time and alphabetical order
  - combine breed for mixed animal
- Data Validation

  - Not Needed as there's no input

- SQL

```
WITH adoption AS
(
  SELECT bn.breed,
         bn.species,
         MONTH(a.adoption_date) AS MONTH
  FROM AdoptionInformation a
    INNER JOIN (SELECT a.petId,
                       GROUP_CONCAT(breed ORDER BY breed DESC SEPARATOR '/') AS breed,
                       b.species
                FROM AdoptionInformation a
                  LEFT JOIN AnimalBreed ab ON a.petId = ab.petId
                  LEFT JOIN Breed b ON b.breed = ab.breed
                GROUP BY a.petId,
                         b.species) breed_name bn ON bn.petId = a.pedId
  WHERE a.adoption_date > DATE_SUB(DATE_FORMAT(NOW(),'%Y-%m-01'),INTERVAL 12 MONTH)
),
adoption_breed_count AS
(
  SELECT MONTH,
         species,
         breed,
         COUNT(1) AS counts
  FROM adoption
  GROUP BY MONTH,
           speicies,
           breed
) adoption_species_count
AS
(SELECT month,
       species,
       'Total',
       COUNT(1) AS counts
FROM adoption
GROUP BY month,
         species),
         Surrenders AS (SELECT bn.breed,
                               bn.species,
                               MONTH(s.surrender_date) AS MONTH
                        FROM surrender s
                          INNER JOIN (SELECT s.petId,
                    GROUP_CONCAT(breed ORDER BY breed DESC SEPARATOR '/') AS breed,
                    b.species
                    FROM Surrender s
                    LEFT JOIN AnimalBreed ab ON s.petId = ab.petId
                    LEFT JOIN Breed b ON b.breed = ab.breed
                    GROUP BY s.petId,
                    b.species) breed_name bn ON bn.petId = s.pedId
        WHERE s.surrender_date > DATE_SUB(DATE_FORMAT(NOW(),'%Y-%m-01'),INTERVAL 12 MONTH)),
        surrender_breed_count
AS
(SELECT MONTH,
       species,
       breed,
       COUNT(1) AS counts
FROM surrenders
GROUP BY MONTH,
         speicies,
         breed),
         surrender_species_count AS (SELECT MONTH,
                                            species,
                                            'Total',
                                            COUNT(1) AS counts
                                     FROM surrenders
                                     GROUP BY MONTH,
                                              species),adoption_all as
        (SELECT * FROM adoption_breed_count UNION adoption_species_count),
        surrender_all AS (SELECT *FROM surrender_breed_count
        UNION
        surrender_species_count)
SELECT CASE
         WHEN aa.month IS NULL THEN sa.month
         ELSE aa.month
       END AS month,
       CASE
         WHEN aa.species IS NULL THEN sa.species
         ELSE aa.species
       END AS species,
       CASE
         WHEN aa.breed IS NULL THEN sa.breed
         ELSE aa.breed
       END AS breed,
       CASE
         WHEN aa.counts IS NULL THEN 0
         ELSE aa.counts
       END AS Adoption_Number,
       CASE
         WHEN sa.counts IS NULL THEN 0
         ELSE sa.counts
       END AS Surrender_number
       FROM adoption_all aa outer join surrender_alls sa ON aa.month = sa.month
       AND aa.species = sa.species AND aa.breed = sa.breed
       order by aa.month,aa.species,aa.breed
```

2.12 Volunteer Lookup

Abstract Code

- **find people** by their first name matching the search item (jon means jonson, jonstone etc.)
- **find users** by first and last name and email
- **find phone number** from volunteer table
- Data Validation
  - - if user type in non-string type, it will be transform to string type
  - - if user type noting in either first name or last name, transform into ”
- SQL

```
SELECT l.first_name,
   l.last_name,
   v.phone_number,
   l.email_address
FROM Volunteer v
LEFT JOIN LoginUser l ON v.username = l.username
WHERE l.first_name LIKE CONCAT('$First_Name','%')
AND   l.last_name LIKE CONCAT('$Last_Name','%')
ORDER BY l.last_name,
     l.first_name;
```

2.13 Vaccine Reminder Report

Abstract Code

- User expiration date to **find all vaccine** that are expiring
- **Find animals** by vaccine activity id
- **Find person** who did the vaccine
- Data Validation
  - Not Needed as there's no input
- SQL

```
WITH most_recent_expiring_vaccination AS (
select
   animalvaccine_id,
        pet_id,
        vaccine_type,
        first_name,
        last_name,
        max(expiration_date) as expiration_date
    FROM AnimalVaccine
    INNER JOIN LoginUser on AnimalVaccine.username = LoginUser.username
    WHERE PERIOD_DIFF(expiration_date, CURDATE()) >= 0 and
    PERIOD_DIFF(expiration_date, CURDATE()) <=3
    GROUP BY 1,2,3,4)
   ,pets AS (
    select
        pet_id,
   species,
   sex,
   alteration_status,
   microchip_id,
   surrender_date,
   GROUP_CONCAT(DISTINCT AnimalBreed.breed ORDER BY AnimalBreed.breed SEPARATOR '\') as breed
FROM Animal
LEFT JOIN Surrender on Animal.pet_id = Surrender.pet_id
INNER JOIN AnimalBreed on Animal.pet_id = AnimalBreed.pet_id
INNER JOIN Breed on Breed.breed = AnimalBreed.breed
GROUP BY 1,2,3,4,5,6)
SELECT
   vaccination_type,
   expiration_date,
   m.pet_id,
   species,
   breed,
   sex,
   alteration_status,
   microchip_id,
   surrender_date,
   first_name,
   last_name
FROM most_recent_expiring_vaccination AS m
INNER JOIN pets ON m.pet_id = pets.pet_id
ORDER BY expiration_date, pet_id;
```