Table of Contents:

Log In

Abstract Code

- User enters *UserName* ('$UserName') and *Password* ('$Password') input fields
- If data validation is successful for both *UserName* and *Password* input fields, then:
    - When ***Enter*** button is clicked:

```
SELECT user_name, _password FROM 'User' WHERE user_name = '$UserName';
```

        - If User record is found but User._password != '$Password':
            - Display **Log in Form** along with error message
        - Else
            - Store login information as session variable '$UserName'
            - Determine if a Volunteer has logged in:

```
SELECT user_name FROM Volunteer WHERE user_name = '$UserName';
```

        - If User record is found, store 'Volunteer' as a session variable '$TypeUser'
        - Else
            - Determine if an Admin User has logged in:

```
SELECT user_name FROM AdminUser WHERE user_name = '$UserName';
```

        - If User record is found, store 'AdminUser' as a session variable '$TypeUser'
        - Else
            - Determine if an Employee has logged in:

```
SELECT user_name FROM Employee WHERE user_name = '$UserName';
```

        - If User record is found, store 'Employee' as a session variable '$TypeUser'
        - User is taken to the **Animal Dashboard**
- Else UserName and Password input fields are invalid. Display the **Log in Form** along with error message

Show/Filter/Sort Animals

Abstract Code

- **Animal Dashboard** initially displays all animals currently in the shelter.
- Current Animals is derived.

§ If there has not been an adoption

§ AddAdoption.AdoptionDt is missing for the animal

o Show the following

§ Animal.Name

§ Species.SpeciesName

§ Breed.BreedName

· If there are multiple breeds, concatenate them together in alphabetical order separated by a "/" forward slash

§ Animal.Sex

§ Animal.AlterationStatus

§ Age is calculated by subtracting the current date – the Animal.BirthDt

§ Adoptability Status

· Adoptability status is derived.

o If there is no AddAdoption.AdoptionDt and the Animal.AlterationStatus is "Spayed" or "Neutered" and the required vaccinations have been performed, the adoptability status is "Adoptable"

o Else the animal is "Not adoptable"

o Number of Available Spaces

§ Number of Available spaces is derived.

· Subtract the Species.Capacity from the number of animals without an adoption date for each species

o If Number of Available spaces is greater than 0 and the role of the user is Employee/AdminUser then display Number of Available Spaces and the ***Add Animal*** link

§ If User clicks on ***Add Animal*** link, the **Add Animal Form** is displayed

o ***Add Adoption Application*** link will take any user to the **Add Adoption Application** form

· To perform the Filter task, the User will make their selections and click on the ***Filter*** link:

o Select Adoptability Status from a drop down where the options are: Adoptable / Not Adoptable / View All (View All is Default Selection)

§ If Adoptable is selected, the **Animal Dashboard** will show animals with an Adoptability Status of Adoptable

§ If Not Adoptable is selected, the **Animal Dashboard** will show animals with and Adoptability Status of Not Adoptable

§ If View All is selected, there is no constraint on Adoptability Status

o Select Species from a drop down where the options are: Species.SpeciesName from a drop down populated by Species / View All (View all is Default Selection)

§ If a Species is selected, the **Animal Dashboard** will show animals where the Species is equal to that of the selection

§ If View All is selected, there is no constraint on Species

#Show all current animals in shelter

```
Select a.name, GROUP_CONCAT(DISTINCT a.breed ORDER BY a.breed ASC SEPARATOR '/') as
breed, a.species_name, a.sex, a.alteration_status, DATEDIFF(DAY, a.birth_date, GetDate()) /
365.25 as age, case when a.alteration_status = 'Spayed' or  a.alteration_status = 'Spayed'
then 'Adoptable' else 'Not adoptable' end as Adoptability_Status
from animal a inner join adoption adpt on adpt.pet_id = a.pet
Where adpt.adoption_date = Null
```

#number of spaces available

```
select count(breed)
from animals a
Where adpt.adoption_date = Null
Minus
Select count(b.breed), s.capacity
from breed b inner join species s on s.species_name = b.species_name
Where adpt.adoption_date = Null;
```

- Assume that $adoptobility_status & $species_name are stored variables in the application

```
Select a.name, a.breed, a.species_name, a.sex, a.alteration_status,
DATEDIFF(DAY, a.birth_date, GetDate()) / 365.25 as age, case when
a.alteration_status = 'Spayed' or  a.alteration_status = 'Spayed' then 'Adoptable'
else 'Not adoptable' end as Adoptability_Status
from animal a
where Adoptabiltiy_Status = $Adoptability_Status or species_name = $species_name
```

Enter Animal Attributes

Abstract Code

- **Add Animal Form** is displayed after Employee/AdminUser clicks on the ***Add Animal*** link
- A data entry form is displayed where Employee/AdminUser either enters the information or selects a value from a drop down:
    - Animal.Name – User enters $AnimalName'
    - Species.SpeciesName
    - Run the Get Species Name task to query the database to return a list of Species.SpeciesName (s) where the number of available spaces is greater than zero to a drop down

```
SELECT species_name FROM Species;
```

        - Once the user has chosen a species, store species as session variable '$ChosenSpecies'
    - Breed.BreedName
    - Run the Get Breed Names task to query the database to return a list of BreedName.BreedName (s) based on the selected Species

//populate a list of species name to ***choose***

```
SELECT breed_name FROM BreedName WHERE species_name = '$ChosenSpecies';
```

    - If user selects “Unknown” or “Mixed” the user is prevented from selecting multiple breeds
    - Else user can select multiple breeds

§ If multiple breeds are selected, they will be ordered alphabetically and separated by a “/” forward slash

Store breeds in a list $ChosenBreeds'

- Animal.Sex – User select $AnimalSex'
  - § User can select one of three choices: “Male”, “Female”, “Unknown”
- Animal.AlteratonStatus - User select $AnimalAlterationStatus'
  - § User can select one of three choices: “Neutered”, “Spayed”, or “Not Fixed”
- Animal.BirthDt – User enter $AnimalBirthDt'
  - § Age is calculated by subtracting the current date system value from the birth date

  Birth Date is entered by the user using “DD-MM-YYYY” format;
- Animal.Description – User enter $AnimalDescription'
- Animal.MicroChipID – User enter $AnimalMicrochipID'
- Animal.SurrenderDt (User must enter value in a date format. Default to current date) – User enters $AnimalSurrenderDt'
- Animal.SurrenderReason – User enter $AnimalSurrenderReason'
- Animal.SurrenderedByAnimalControl – True or False – User select $AnimalLocalControl'

· When ***Enter*** button is clicked:

- Verify the number of available spaces is greater than zero for the selected species.

```
SELECT capacity FROM Species WHERE species_name = '$ChosenSpecies';
```

If capacity is enough,

```
INSERT INTO Animal (name, surrender_date, local_control, surrender_reason,
alterature_status, sex, birth_date, description, microchip_id, user_name)
VALUES ('$AnimalName', '$AnimalSurrenderDt', '$AnimalLocalControl',
'$AnimalSurrenderReason', '$AnimalAlteratureStatus', '$AnimalSex', '$AnimalBirthDt',
'$AnimalDescription', '$AnimalMicrochipID', '$UserName');
```

- Display the Auto-generated Pet ID to the user for the User just entered with a message: “The animal just entered has been accepted with Pet ID '$PetID'

```
SELECT LAST_INSERT_ID();
```

For each element '$AnimalBreed' in the list of '$ChosenBreeds':

```
INSERT INTO Breed (pet_id, species_name, breed_name)
VALUES ('$PetID', '$ChosenSpecies', '$AnimalBreed');
```

§ If number of Available Spaces is equal to zero, then display message "At Capacity, there is no more room." Cancel submission and close **Add Animal Form** Return to the **Animal Dashboard**

o If a duplicate Animal.MicrochipID is entered into the system, display an error message, "Error: Microchip ID already exists in the system." Cancel submission and return to the **Add Animal Form**.

o All fields except Animal.MicrochipID must contain a value. If any fields contain null values, display an error message "Error: Field must contain a value." Cancel submission and return to the **Add Animal Form**.

Upon Successful submission of the **Add Animal Form**, the user will be taken to the **Animal Detail Form** pass variable '$PetID'.

View Animal Information/Vaccination History

Abstract Code

- Employee/Admin is brought to the **Animal Detail Form** after successfully adding an animal via the **Add Animal Form.**
- A user can also be brought to the **Animal Detail Form** after clicking on the animal's name, Animal.Name from the **Animal Dashboard**
- **View Animal Information/Vaccination History** task is run to display the specific information for the animal just entered or the animal that was clicked on (Passed as '$PetID')
    - Animal.Name
    - Species.SpeciesName
    - Breed.BreedName
    - Animal.Sex
    - Animal.AlterationStatus
    - Animal.Age (Years and Months) Difference between system value of Current date and Animal.BirthDt
    - Animal.Description
    - Animal.MicrochipID
    - Animal.SurrenderDt
    - Animal.SurrenderReason
    - Animal.SurrenderByAnimalControl

```
SELECT  A.pet_id,   A.name,  A.sex,  A.alteration_status,
timestampdatediff(YEAR, A.birth_date, now()) as age_year,
timestampdatediff(MONTH, A.birth_date, now()) % 12 as age_month,
A.description,  A.microchip_id,  A.surrender_date,  A.surrender_reason,
A.local_control,  B.species,B.breed_name
 FROM
  Animal A INNER JOIN (SELECT pet_id, species, GROUP_CONCAT( DISTINCT breed_name
   ORDER BY
   breed_name ASC SEPARATOR '/' )
   FROM
   Breed
     ) B ON A.pet_id = B.pet_id
WHERE  A.pet_id = '$PetId';
```

- Find and display each Vaccination record for the selected animal
    - VaccineType.VaccineName
    - VaccineType.Required
    - Vaccination.AdministerDt
    - Vaccination.ExpDt
    - Vaccination.VaccinationNum

```
SELECT pet_id, vaccine_name, administer_date, expiration_date, vaccination_number
FROM Vaccination
WHERE  pet_id = '$PetID';
```

- Determining Adoptability Status:

If Adoption date is null and the alteration status = TRUE

```
SELECT COUNT(A.pet_id)
FROM Animal A LEFT OUTER JOIN Adoption B ON A.pet_id = B.pet_id
WHERE A.alteration_status = TRUE AND B.adoption_date is NULL
AND A. pet_id = '$PetID';
```

If an animal record exists, perform query to determine if there are required vaccinations which have not yet been administered

```
SELECT  COUNT(vaccine_name)
FROM  VaccinationType A INNER JOIN VaccinationRequired B ON A.vaccine_name =
B.vaccine_name AND B.required = TRUE
LEFT OUTER JOIN Vaccination C ON A.vaccine_name = C.vaccine_name
WHERE  C.pet_id = '$PetID'
AND ( C.administer_date IS NULL OR C.expiration_date < now()  );
```

- If there are null or zero vaccines returned, ie. then the animal is adoptable. If the '$TypeUser' = 'Employee' or 'AdminUser' then display ***Add Adoption*** link
- If user clicks on ***Add Adoption*** link, the user will navigate to the **Adoption form**

- If animal is adoptable certain attributes can be updated. After user makes updates and clicks on ***Update*** button, the following queries will execute. The form will be submitted and the queries for checking adoptability status will execute. Maintaining the current variable '$PetID'

    - If Animal.Sex is “Unknown” open up this field to update

```
UPDATE Animal
SET sex = '$Sex'
WHERE sex = 'Unknown'
AND pet_id = '$PetID';
```

    - Animal.MicrochipID can be updated. However, upon submission if a duplicate MicrochipID is already in the system, an error message will display, “Error: Microchip ID already exists in the system.” Cancel submission and return to the **Animal Detail Form**.

```
UPDATE Animal
SET microchip_id = '$MicrochipID'
WHERE pet_id = '$PetID';
```

    - If Animal.Breed is “Unknown” or “Mixed” open up the field to update. Must follow the rules. Multiple breeds can only be selected if “Unknown” and “Mixed” are not chosen. “Unknown” or “Mixed” can be the only value if selected.

```
UPDATE Breed
SET breed_name = '$Breed'
WHERE breed_name IN ('Unknown ', 'Mixed ')
AND '$Breed' NOT IN ('Unknown ', 'Mixed ')
AND pet_id = '$PetID';
```

For Each subsequent Record Inserted:

```
INSERT INTO Breed (pet_id, species_name, breed_name)
VALUES (
(SELECT pet_id, species_name
FROM Breed
WHERE pet_id = '$PetID'), '$Breed') ;
```

- If Animal.AlterationStatus = FALSE, then the field is updateable

```
UPDATE Animal
SET alteration_status = '$AlterationStatus'
WHERE  pet_id = '$PetID'
AND alteration_status = FALSE;
```

- Display ***Add Vaccination*** link (Volunteers do not enter Animals and would only navigate to the **Animal Detail Form** from the **Animal Dashboard** which displays animals in the shelter)
- If user clicks on ***Add Vaccination*** link, they navigate to the **Add Vaccination** Form

Add Vaccination

Abstract Code

- If user clicks on the ***Add Vaccination*** link,
  - The following input fields are displayed on the **Vaccination** form
    - Dropdown containing a list of eligible vaccines
- Query for VaccineType.VaccineName by Species
- Once selected, User enters
  - *AdministerDt* in valid date format
  - *ExpDt* in valid date format. (Vaccination.ExpDt must be after the Vaccination.AdministerDt. If not, display error message: "Error: The Expiration Date or Next Dosage Date must fall after the date Administered."
  - *VaccinationNum*
- If user clicks the ***Add Vaccination*** button
  - Validation check to make sure if there is an existing vaccination record for that particular vaccine, the *AdministerDt* must be after the Vaccination.ExpDt of the existing record. If the Administer*Dt*is before the Vaccination.ExpDt display an error message: "Error: This vaccination is not yet due for this animal." Cancel submission
  - If information passes validation checks, send the '$UserName' to populate the database, AddVaccination.VaccAddedBy
  - Upon successful submission user is navigated back to the **Animal Detail Form**

```
select vr.species_name,vt.vaccine_name
from VaccinationType vt inner join VaccinationRequired vr on vr.vaccine_name =
vt.vaccine_name
where species_name = $species_name


Insert into 'Vaccinations'(administer_date, expiration_date,
vaccination_number,user_name)
```

```
values ($administer_date, $expiration_date, $vaccination_number,$user_name);
end for 'Vaccinations'
```

Enter Adoption details

Abstract Code

- Employee/AdminUser clicks on ***Add Adoption*** link from the **Animal Detail Form**.
- Two Input Fields are displayed for the user to enter all or partial
  - *AppLastName: User enters* '$ApplicantLastName'
  - *CoLastName: User enters* '$CoApplicantLastName'
- User enters the information and clicks on ***Search*** button to initiate the Search Adopter task
- Run the Search Adopter task: Query Application where the ApproveApplication.Status is "Approved" and the Application.AppLastName contains $ApplicantLastName' or Application.CoLastName contains '$CoApplicantLastName' and the specific application does not have any related adoption information
  - If *CoLastName* is null, perform Search Adopter Task on $ApplicantLastName',

```
SELECT A.application_number, A.application_date, A.Email, A.coapplicant_first_name,
A.coapplicant_last_name, B.applicant_fist_name, B.applicant_last_name, B.phone_number,
B.street, B.city, B.state, B.zipcode
FROM Application AS A LEFT JOIN
(SELECT Email, applicant_first_name, applicant_last_name, phone_number, street, city, state,
zipcode FROM Adopter WHERE applicant_last_name LIKE $ApplicantLastName') AS B
ON A.Email = B.Email
WHERE A.status = 'Approved';
```

  - If *AppLastName* is null, perform Search Adopter Task on $CoApplicantLastName',

```
SELECT A.application_number, A.application_date, A.Email, A.coapplicant_first_name,
A.coapplicant_last_name, B.applicant_fist_name, B.applicant_last_name, B.phone_number,
B.street, B.city, B.state, B.zipcode
FROM
(SELECT Email, application_number, application_date, email, status, coapplicant_first_name,
coapplicant_last_name FROM Application WHERE coapplicant_last_name LIKE
$CoApplicantLastName') AS A
LEFT JOIN Adopter AS B
ON A.Email = B.Email
WHERE A.status = 'Approved';
```

- If both fields are filled, perform Search Adopter Task on $ApplicantLastName', $CoApplicantLastName',

```
SELECT A.application_number, A.application_date, A.Email, A.coapplicant_first_name,
A.coapplicant_last_name, B.applicant_fist_name, B.applicant_last_name, B.phone_number,
B.street, B.city, B.state, B.zipcode
FROM
(SELECT Email, application_number, application_date, email, status, coapplicant_first_name,
coapplicant_last_name FROM Application WHERE coapplicant_last_name LIKE
$CoApplicantLastName') AS A
INNER JOIN
(SELECT Email, applicant_first_name, applicant_last_name, phone_number, street, city, state,
zipcode FROM Adopter WHERE applicant_last_name LIKE $ApplicantLastName') AS B
ON A.Email = B.Email
WHERE A.status = 'Approved';
```

- If there are no results, display message “No applicants meet the search criteria.”
- Else display the results from the Search Adopter Task query
  - Application.AppNum
  - Application.AppDt
  - Adopter.AppFirstName
  - Adopter.AppLastName
  - Application.CoFirstName
  - Application.CoLastName
  - Adopter.Email
  - Adopter.PhoneNumber
  - Adopter.Street
  - Adopter.City
  - Adopter.State
  - Adopter.ZipCode

- User confirm the right adopter and application number, user ***click*** the correct application number $AdoptApplicationNumber'

- Run the Enter adoption subtask, user enter into three data entry fields
    - *PetID* - Data must be entered in a valid date format else display error message, this can be referenced by the paper copy of the user
    - *AdoptionDt* - Data must be entered in a valid date format else display error message
    - *AdoptionFee* – Data must be entered in a valid float format else display error message

    Store the information into $AdoptPetID', $AdoptionDt', $AdoptionFee'

- User clicks ***enter***:

```
INSERT INTO Adoption (application_number, pet_id, adoption_date, adoption_fee)
VALUES ($AdoptApplicationNumber', '$AdoptPetID', $AdoptionDt', $AdoptionFee');
```

- User will find the correct adopter and and complete entering the Input fields and click on ***Enter Adoption*** button. Upon successful submission, $AdoptionDt and $AdoptionFee will be sent to the database for the selected approved adopter

Add New Adoption Application

Abstract Code

- The user navigates to the **Add Adoption Application** form after clicking the ***Add Adoption Application*** link on the **Animal Dashboard**
- To perform the **Add New Adoption Application** task, the user enters the information in the following input fields:
    - *AppFirstName* '$AppFirstName'
    - *AppLastName* '$AppLastName'
    - *CoFirstName* '$CoFirstName'
    - *CoLastName* '$CoLastName'
    - *Email* '$Email'
    - *AppDt (must be in Date format e*lse display error message*)* '$AppDt'
    - *PhoneNumber* '$PhoneNumber'
    - Address
        - *Street* '$Street'
        - *City* '$City'
        - *State* '$State'
        - *ZipCode* '$ZipCode'
- User clicks on ***Add Application*** button to submit information to the database.
    - Upon submission, only *CoFirstName* and *CoLastName* can have missing values otherwise display error message “Error: Field must contain a value.” Cancel submission and return to the **Add Adoption Application** form
    - Upon successful submission, the following fields are updated in the database
        - Adopter.Email
        - Adopter.AppFirstName
        - Adopter.AppLastName
        - Adopter.PhoneNumber
        - Adopter.Address.Street
        - Adopter.Address.City
        - Adopter.Address.State
        - Adopter.Address.ZipCode

```
INSERT INTO Adopter (email, primary_first_name, primary_last_name, phone_number,
street, city, state, zip_code)
VALUES ('$Email', '$AppFirstName', '$AppLastName', '$PhoneNumber', '$Street', '$Street',
'$City', '$State', '$ZipCode');
```

- Application.AppNum
- Application.AppDt
- Application.Email
- Application.Status
- Application.CoFirstName
- Application.CoLastName

```
INSERT INTO Application (AppDt, Email, Status, CoFirstName, CoLastName)
VALUES ('$AppDt', '$Email', 'pending approval', '$CoFirstName', '$CoLastName');
```

- Display the Auto-generated Application Number to the user for the Email/Adopter just entered with a message: “The adoption application for email '$Email' has been accepted with Application Number '$AppNum' “

```
SELECT AppNum, Email
FROM Application
WHERE Email = '$Email';
```

- User is returned to the **Animal Dashboard**

List Pending Applications/Mark Application

Abstract Code

- AdminUser/Inge will click on the ***Approve Applicaton*** link found on the **Animal Dashboard**
- **List Pending Applications/Mark Application** task is run to query the database where the ApproveApplication.Status is "Pending Approval" and return Application information
- Application.AppFirstName
- Application.AppLastName
- Application.CoFirstName
- Application.CoLastName
- Application.Email
- Application.AppDt
- Application.PhoneNumber
- Application.Address.Street
- Application.Address.City
- Application.Address.State
- Application.Address.ZipCode
- Once AdminUser selects a specific application, two buttons will be displayed:
- ***Approve***
  - Clicking on the ***Approve*** button will update the database, ApproveApplication.Status, to "Approved" for the selected application
- ***Reject***
  - Clicking on the ***Reject*** button will update the database, ApproveApplication.Status, to "Rejected" for the selected application
- Once a decision has been made on an application and the data is submitted to the database, the **List Pending Applications/Mark Application** task is run to query the database once again and remove the decisioned application from the list

- Once Application review is completed, AdminUser/Inge clicks on the ***Animal Dashboard*** link to return to the **Animal Dashboard** form.

#mark application

```
Select app.primary_first_name, app.primary_last_name, app.co_first_name,
app.co_last_name, app.email, app.application_date, app.phone_number, app.street,
app.city, app.state, app.zip_code
from application
Where app.status = 'Pending Approval'

Insert into 'Application'(*) values ($app.primary_first_name, $app.primary_last_name,
$app.co_first_name, $app.co_last_name, $app.email, $app.application_date,
$app.phone_number, $app.street, $app.city, $app.state, $app.zip_code) where application =
'approved';
end for 'Application'
```

Reports

Abstract Code

- From the **Animal Dashboard** AdminUser/Inge will have a ***Reports*** button displayed
- The **Reports task** is initiated when the ***Reports*** button is clicked, a Menu will display the following reports as links. AdminUser will click on the link to initiate the task to run the report
- Show " ***Animal Control Report*** ", " ***Volunteer of the Month***", "***Monthly Adoption Report*** " ", "***Volunteer Lookup*** " ", "***Vaccine Reminder Report*** " tabs in the menu.
- ***Animal Control Report*** link – Jump to the **Animal Control Report** task
- ***Volunteer of the Month*** link – Jump to the **Volunteer of the Month** task
- ***Monthly Adoption Report*** link – Jump to the **Monthly Adoption Report** task
- ***Volunteer Lookup*** link – Jump to the **Volunteer Lookup** task
- ***Vaccine Reminder Report*** link – Jump to the **Vaccine Reminder Report** task

Animal Control Report

Abstract Code

- The **Animal Control Report task** is initiated when the ***Animal Control*** link is clicked
- Animal is queried and returns the following counts and displays on the screen
- Count of Surrenders and Adoptions Task:
    - Return Count of animals and month of Animal.SurrenderDt where Animal.LocalControl is True and Animal.SurrenderDt is in current month and last 6 months
    - Count of Adoptions for animals in rescue more than 60 days

```
SELECT 'Animals Surrendered' AS metric, MONTH(surrender_date) AS _month, count(pet_id)
AS animal_count
FROM Animal
WHERE surrender_date >= DATE_SUB(now(), INTERVAL 6 MONTH)
AND local_control = TRUE
GROUP BY MONTH(surrender_date)
UNION
SELECT 'Animals Adopted' AS metric, MONTH(adoption_date) AS _month, count(pet_id) AS
Animal_count
FROM Animal A INNER JOIN Adoption B ON A.pet_id = B.pet_id
WHERE surrender_date >= DATE_SUB(now(), INTERVAL 60 DAY)
GROUP BY MONTH(adoption_date);
```

- Display the Counts as links on the screen with the month, '$_month' and '$metric' as variables to pass
- If user clicks on ***Count*** link for Surrender or Adoption, the Animal Control Drill Down Report task is initiated.
    - If '$metric' = 'Animals Surrendered'
    - The data retrieved from the Animal Control Drill Down Report task is displayed on a new screen and shows the following sorted by Animal.PetID:
        - Animal.PetId
        - Species.SpeciesName
        - Breed.BreedName (If multiple, in alphabetical order separated by forward slash)
        - Animal.Sex
        - Animal.AlterationStatus
        - Animal.MicroChipID
        - Animal.SurrenderDt

```
SELECT  A.pet_id,   C.species_name, C. breed_name
 , A.sex,  A.alteration_status,
A.microchip_id,  A.surrender_date
FROM
  Animal
  INNER JOIN (SELECT pet_id, species_name, GROUP_CONCAT( DISTINCT breed_name
   ORDER BY
   breed_name ASC SEPARATOR '/' ) AS breed_name
   FROM
   Breed) as C ON A.pet_id = C.pet_id
 WHERE surrender_date >= DATE_SUB(now(), INTERVAL 6 MONTH)
AND MONTH(surrender_date) = '$_month'
AND local_control = TRUE
ORDER BY A.pet_id asc;
```

- If '$metric' = ‘Animals Adopted’
- The data retrieved from the Animal Control Drill Down Report task is displayed on a new screen and shows the following sorted by days in rescue descending:
    - Animal.PetId
    - Species.SpeciesName
    - Breed.BreedName (If multiple, in alphabetical order separated by forward slash)
    - Animal.Sex
    - Animal.AlterationStatus
    - Animal.MicroChipID
    - Animal.SurrenderDt
    - Adoption.DaysinSurrender

```
SELECT  A.pet_id,   D.species_name,  D.breed_name
, A.sex,  A.alteration_status,
A.microchip_id,  A.surrender_date , DATEDIFF(C.adoption_date, A.surrender_date) AS
days_in_surrender
FROM
  Animal A
  INNER JOIN Adoption C ON A.pet_id = C.pet_id
  INNER JOIN (SELECT pet_id, species_name, GROUP_CONCAT( DISTINCT breed_name
   ORDER BY
   breed_name ASC SEPARATOR '/' ) AS breed_name
   FROM
   Breed) as D ON A.pet_id = D.pet_id
WHERE surrender_date >= DATE_SUB(now(), INTERVAL 60 DAY)
AND MONTH(surrender_date) = '$_month'
ORDER BY DATEDIFF(C.adoption_date, A.surrender_date)  desc;
```

- Closing screen returns user to the **Animal Dashboard**

Volunteer of the Month

Abstract Code

- User clicks on the ***Volunteer of the Month*** link to initiate the task for Date Selection
- The database is queried and returns the year and month combinations found within Volunteer.WorkDt
- The Year/Month results are displayed in a drop down selection box for the AdminUser to select a Year/Month combination
  - $Year
  - $Month
- Once Admin/User makes a selection, they click on the ***Run Report*** button
- The database is queried and returns the top 5 volunteers by Volunteer.HoursWorked where the year of Volunteer.WorkDt = $Year and he month of Volunteer.WorkDt = $Month
- The following information is displayed on the report ordered by Volunteer.HoursWorked descending, followed by User.Name.LastName
  - User.Name.FirstName
  - User.Name.LastName
  - User.Email
  - Volunteer.HoursWorked
- UserAdmin can close screen to return to the **Animal Dashboard**

```
Select u.first_name,u.last_name, u.email, dw.hours_worked
from user u inner join volunteer vol on vol.user_name = u.user_name
inner join daywork dw on dw.user_name = vol.user_name
order by vol.hours_worked
where year(work_date) = $year and month(work_date) = $month
```

Monthly Adoption Report

Abstract Code

- AdminUser clicks on the ***Monthly Adoption Report*** link to initiate the **Monthly Adoption Report** task
- The database is queried to return data where the Animal.SurrenderDt or the AddAdoption.AdoptionDt is within the last 12 months beginning with last month grouped by Month sorted by month in ascending order, followed by Species.SpeciesName, and finally Breed.BreedName
    - Monthly Surrenders and Adoptions report Task:
        - Return Adoption.PetID, Breed.BreedName, Species.SpeciesName and Adoption.AdoptionDt is within the last 12 months beginning with last month
        - Return Animal.PetID, Breed.BreedName, Species.SpeciesName and Animal.SurrenderDt is within the last 12 months beginning with last month
        - Count the number of pet_id for both surrender animals and adopted animals
        - Group the number by month, species and breed
        - The table should be sorted by month in ascending order (earliest to latest), and species name alphabetically (A-Z) followed by breed name alphabetically (A-Z, in the case of an animal with multiple breeds, breeds should be combined into a single value in alphabetical order.

```
SELECT month, species_name, breed_name, num_surrender, num_adoption
FROM

(SELECT MONTH(SurDate) as month, species_name, breed_name, COUNT(pet_id) AS
num_surrender
FROM
(SELECT surrender_info.pet_id, Breed.species_name, Breed.breed_name,
surrender_info.surrender_date as SurDate
FROM
(SELECT pet_id, surrender_date
FROM Animal
WHERE surrender_date >= DATE_SUB(now(), INTERVAL 12 MONTH) ) AS surrender_info
JOIN Breed on Breed.pet_id = surrender_info.pet_id
ORDER BY surrender_info.surrender_date) AS surrender_count
GROUP BY MONTH(SurDate) , species_name, breed_name
ORDER BY species_name, breed_name) AS surrender_all

OUTER JOIN

(SELECT MONTH(AdopDate) as month, species_name, breed_name, COUNT(pet_id) AS
num_adoption
FROM
(SELECT adoption_info.pet_id, Breed.species_name, Breed.breed_name,
adoption_info.adoption_date as AdopDate
FROM
(SELECT pet_id, adoption_date
FROM Adoption
WHERE adoption_date >= DATE_SUB(now(), INTERVAL 12 MONTH) ) AS adoption_info
JOIN Breed on Breed.pet_id = adoption_info.pet_id
ORDER BY adoption_info.adoption_date) AS adoption_count
GROUP BY MONTH(SurDate) , species_name, breed_name
ORDER BY species_name, breed_name) AS adoption_all

ON surrender_all.month = adoption_all.month, , surrender_all.species_name =
adoption_all.species_name, surrender_all.breed_name = adoption_all.breed_name
```

- The following is displayed on the screen:
- Month

- Species.SpeciesName
- Breed.BreedName
- Number of Surrenders
- Number of Adoptions
    - Closing screen returns user to the **Animal Dashboard**

Volunteer Lookup

Abstract Code

- The **Volunteer Lookup** task is initiated when AdminUser/Inge clicks on the ***Volunteer Lookup*** link
- An input form is displayed containing two input fields:
  - *FirstName*
  - *LastName*
- The AdminUser enters full or part of a Volunteers first or last name
- The AdminUser clicks on the **Search** button to initiate the Volunteer Lookup link
- The database is queried to return records where User.FirstName contains '$FirstName' and/or User.LastName contains '$LastName'

- If '$FirstName' is not Null and ' $LastName' is Null

```
SELECT FirstName, LastName, Email, VolPhoneNum
FROM 'User' INNER JOIN Volunteer ON 'User'.UserName = Volunteer.UserName
WHERE FirstName like '$FirstName%'
ORDER BY LastName, FirstName
```

- ElseIf ' $FirstName ' is Null and ' $LastName ' is not Null

```
SELECT FirstName, LastName, Email, VolPhoneNum
FROM 'User' INNER JOIN Volunteer ON 'User'.UserName = Volunteer.UserName
WHERE LastName like '$LastName%'
ORDER BY LastName, FirstName
```

- ElseIf '$FirstName ' is not Null and '$LastName' is not Null

```
SELECT FirstName, LastName, Email, VolPhoneNum
FROM 'User' INNER JOIN Volunteer ON 'User'.UserName = Volunteer.UserName
WHERE FirstName like '$FirstName ' and LastName like '$LastName%'
ORDER BY LastName, FirstName
```

- Else If nothing is entered for either box and **Search** button is clicked, the user will remain on the Volunteer Lookup Screen without querying the database

The results are displayed on the screen for all matching records sorted by User.LastName ascending and User.FirstName ascending:

- User.FirstName
- User.LastName
- User.Email
- Volunteer.VolPhoneNum

- Closing screen returns user to the **Animal Dashboard**

Vaccine Reminder Report

Abstract Code

- The **Vaccine Reminder Report task** is initiated when the ***Vaccine Reminder Report*** link is clicked
- Database is queried and returns records where the Vaccination.ExpDt is in the current month and the next three months
- The following will be displayed
- VaccineType.VaccineName
- Vaccination.ExpDt
- Species.SpeciesName
- Breed.BreedName (If multiple values have been selected, order alphabetically and use a forward "/" as the separator)
- Animal.Sex
- Animal.AlterationStatus
- Animal.MicrochipID
- Animal.SurrenderDt
- AddAnimal.AddedBy

- Closing screen returns user to the **Animal Dashboard**

#vaccine reminder report

```
select vacc.vaccine_name, vacc.expiration_date, vr.species_name, bn.breed_name, a.sex,
a.alteration_status, a.microchip_id, a.surrender_date, a.user_name
from vaccination vacc
inner_join vaccinationRequired vr on vr.species_name = vacc.species_name
inner join breed_name br on br.species_name = vacc.species_name
inner join animal a on a.species_name = a.species_name
Where month(vacc.expiration_date) = month(today()) and  month(vacc.expiration_date) <=
monthd(today()+3))
```