Table of Contents

Data Types

User

Attribute	Data Type	Nullable
Username	String	No
Password	String	No
EmailAddress	String	Yes
Name.FirstName	String	No
Name.LastName	String	No
StartDate	DateTime	No

Volunteer

Attribute	Data Type	Nullable
PhoneNumber	Integer	No

Volunteer Hours

Attribute	Data Type	Nullable
Date	Date	Yes
Hours	Integer	Yes

Animal

Attribute	Data Type	Nullable
PetID	Integer	No
Name	String	No
Description	String	No

Age	Float	No
MicrochipID	String	Yes
Sex	String	No
AlterationStatus	Boolean	No
SurrenderReason	String	No
SurrenderByAnimalControl	Boolean	No
SurrenderDate	DateTime	No

Breed

Attribute	Data Type	Nullable
Name	String	No

Species

Attribute	Data Type	Nullable
Name	String	No
MaxPerShelter	Integer	No

Adopter

Attribute	Data Type	Nullable
EmailAddress	String	No
PhoneNumber	Integer	No
Address.Street	String	No
Address.City	String	No
Address.State	String	No
Address.ZIPCode	String	No

ApplicantName.FirstName	String	No
ApplicantName.LastName	String	No

AdoptionApplication

Attribute	Data Type	Nullable
ApplicationNumber	Integer	No
DateOfApplication	Date	No
CoApplicantName.FirstName	String	Yes
CoApplicantName.LastName	String	Yes
State	String	No

Adoption - Relationship

Attribute	Data Type	Nullable
AdoptionDate	Date	No
AdoptionFee	Float	Yes

Vaccine

Attribute	Data Type	Nullable
VaccineType	String	No
RequiredForAdoption	Boolean	No

VaccinesAdministration

Attribute	Data Type	Nullable
ExpirationDate	Date	No
VaccinationNumber	Float	Yes
DateAdministered	Date	No

Business Logic Constraints

Users

- New Users will be entered by Database Administrator
- Only existing users in the database can login
- Database Administrator will enter new Volunteer Hours

Animals

- If 'Mixed' or 'Unknown' breeds are chosen, then the breed of the animal can be updated in the future
- If Sex is set as "Unknown", then the sex of the animal can be updated in the future. Other available values are "Male" or "Female".
- If no Microchip ID is set for an animal, then the Microchip ID of the animal can be updated in the future
- If the Alteration Status is "Unaltered", then the Alteration Status of the animal can be updated in the future

Adoption Application

- AdoptionApplication.Status can only be updated by Admin User. It may have values of "Pending Approval", "Rejected", or "Approved".
- Adoption Application and Adopter information can only be entered by Employee or Admin users

Adoption

- An Adoption can be performed by either Employee or Admin user

Reports

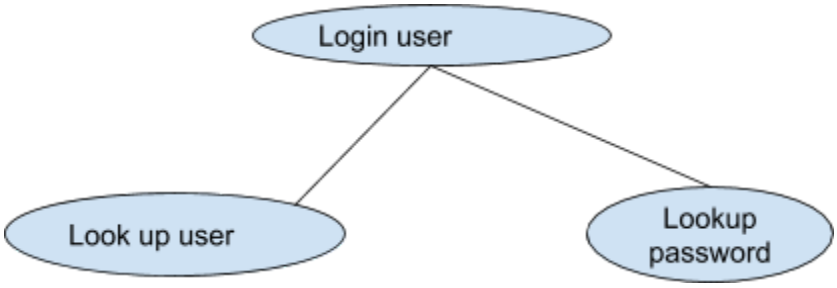
- Only Admin User can view Reports

Vaccination

- System will not allow User to enter a vaccine that is not yet due for an animal or that they have already been given.
- Information about already applied Vaccinations will be entered when an Animal is surrendered.

Task Decomposition / Abstract Code

Logging in



Task Decomp

Lock Types: Read-only on User table and hashed password table
Number of Locks: Single
Enabling Conditions: None
Frequency: Whenever user logs in to page
Consistency(ACID): Not critical. Order is not critical
Subtasks: Mother Task is needed.Must be decomposed into various subtasks

Login form
While ***login*** button not pushed, do nothing
When the ***login*** Button is pushed, do the following:
username and password = get *username* and *password* from form
call login function
login(*username*, *password*):
if forms are valid:
Authenticate the *username* and *password*
if *username* and *password* are correct and authenticated:
Go to the **Animal Dashboard** for user
else:
Display error message with wrong username/password, display **Login** form
else:
Display error message with wrong username/password, **display login form**

Animal Dashboard

Task Decomp

Lock Types: Read-only on Animal Details table
Number of Locks: Single
Enabling Conditions: Login of the user
Frequency: Whenever the user logs in
Consistency(ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed.No decomposition needed.

Animal Dashboard
View Animal Dashboard, Populate all animals/animal information

While no buttons are pressed, do nothing
When a button is pressed, do the following:

if ***species filter*** is pressed:
  Populate filter with species
  species = *Select species* to filter
  Update dashboard with selected species
  View Animal Dashboard
if ***adoptability status filter*** is pressed:
  Populate filter with adoptability status
  adoptability status = *Select adoptability status* to filter
  Update dashboard with selected adoptability status
  View Animal Dashboard
if ***clear filter*** is pressed:
  Clear filters
  View Animal Dashboard
if ***animal name button*** is pressed:
  Go to **Animal Detail** Screen

Get user information
if user == admin:
  retrieve number of spaces in adoption center
  Display number of spaces in adoption center
  View Animal Dashboard

  if user has adequate permissions:
    display add animal button
    display add adoption button
    View Animal Dashboard

if ***add animal*** button is pressed:
  go to **add animal** form

if ***add adoption*** button is pressed:
  go to **add adoption** form

Add Animal

Task Decomp

Lock Types: Write on Animal details table
Number of Locks: Single
Enabling Conditions: When user clicks add animal button
Frequency: Whenever user wants to add an animal
Consistency(ACID): not critical, order is not critical.
Subtasks: Mother Task is not needed.No decomposition needed.

Add animal form
Display **add animal** form to fill in all information about -
animal(name, species, breed, sex, alteration status, age,and adoptability status)
While ***submit animal*** button is not pushed, do nothing
When ***submit animal*** button is pushed, do the following:
  if forms are and complete:
    Check number of species to see if space available
    if space available:
      Tell user that information is correct
      Go to **Animal Detail** Screen
    else:
      Tell user that space is full
      go to **Add animal** form screen
  else:
    Display error message with invalid form, display **animal form** again

View Animal Details

Task Decomp

Lock Types: Read-only on Animal list table
Number of Locks: Single
Enabling Conditions: When user clicks view animal details

Frequency: Whenever user wants to view animal details
Consistency(ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed.No decomposition needed.

Abstract Code - **View Animal's Detail** Dialog

View animal details upon clicked, will display a list of the animals details including all the fields mentioned.
While Dialog is open:
Display all Animal information included, adoption status, vaccinations and animal details

- if user has appropriate permissions
    - if animal can be adopted:
        - Display button to adopt animal
    - Else if does not have any vaccinations:
        - Add option to add vaccination (displayed below)
        - If nothing in pressed do nothing
        - If button is pressed, redirect to add Vaccination page
- Else:
    - Display Information mentioned above

Add Vaccination

Task Decomp

Lock Types: Write only Animal details table
Number of Locks: Single
Enabling Conditions: When user clicks add vaccination button
Frequency: Whenever user wants to add a vaccination to animal
Consistency(ACID): not critical, order is not critical.
Subtasks: Mother Task is not needed.No decomposition needed.

Abstract Code - **Add Vaccination form**

Located in the Animal Details Page, a button with 'add vaccine' when pushed will display Display Dialogue box with a list of Vaccinations corresponding to that breed of animal coming from the Vaccine Appendix in the Database

While ***add vaccine*** button is not pushed, do nothing
When ***add vaccinel*** button is pushed, do the following:
User will be prompted to enter a vaccine in a text box (optional autocomplete)

- if *vaccine* does not exist:

    - Will through error saying to pick a vaccine from the appendix
- else:
    - if animal already has that *vaccine*:
        - Error will be thrown saying that the animal already has that *vaccine*
        - Prompt user to try adding a Vaccine from the database
    - Else:
        - Database will be updated with animal's new *vaccinations* that the user chose for the animal
        - User will have to enter vaccination date, and next dose date
        - Optional: vaccine/tag number
- If user closes box without entering appropriate vaccine
    - Dialogue box will close

View Vaccine Reminder Report

Task Decomp

Lock Types: Read-only on Animal Details table
Number of Locks: Single
Enabling Conditions: When user clicks view
Frequency: Whenever user wants to add a vaccination to animal
Consistency(ACID): not critical, order is not critical.
Subtasks: Mother Task is not needed.No decomposition needed.

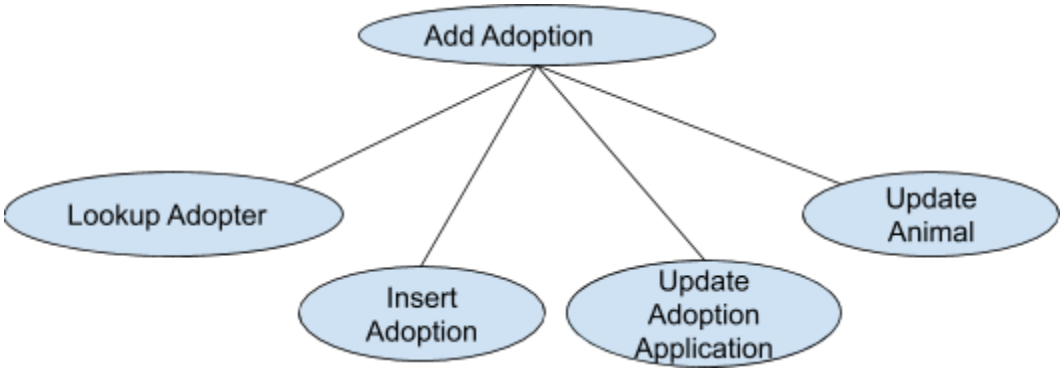
Abstract Code - **View Vaccine Reminder Report**

Seperate page with all the animals in a table with their corresponding vaccines expiration date.

- *User clicks the **View Report** Button*
- If Vaccination is due in the current and next three months:
    - Display vaccination type
    - Display vaccination due date
    - Display pet ID, species
    - DIsplay the breed - array format just in case there are multiple
    - Display rest of fields such as: sex, alteration status, microID, surrender date
    - First and last name of person who recorded the last vaccination of this type for this animal.
- Data will be displayed in ascending order of the vaccination due date && ny pet ID ascending

Add Adoption

Task Decomp



Lock Types: 2 Read-only lookups for Adopter and AdoptionApplication; 1 Update for Animal and 1 Update for AdoptionApplication with new Adoption relationship
Number of Locks: 2 Read-Only, 2 Updates
Enabling Conditions: Triggered by clicking ***Add Adoption*** after an Animal Lookup in **Animal Details**
Frequency: Once per animal and application. Dozens per month.
Consistency(ACID): Critical as adoption must occur once per animal and AdoptionApplication
Subtasks: Mother Task is needed because it can be decomposed into subtasks and need to keep an order.

Abstract Code - **Add Adoption** Form

- User clicks on ***Add Adoption*** button from **Animal Detail**:
- Prompt **Search Dialog** to look up an approved adopter. User can input *Applicant Last Name* and *Co-Applicant Last Name*
- While ***Search*** button is not pressed, do nothing
- When ***Search*** button is pressed, then:
    - Find each Adopter where Adopter.LastName contains *Applicant Last Name* input; and Application.Coapplicant.LastName contains *Co-Applicant Last Name* input.
    - From previous results, find each Adopter related to an Application where Application.State is 'Approved' and not related to any Adoption

    - Display from Adopter: Applicant Name, Address, Phone Number, Email Address. And from AdopterApplication: Application Number, Co-Applicant Name, and Date of Application
- Upon selection of Adopter
    - Prompt User to enter *Adoption Date* and *Adoption Fee*
- When ***Submit*** button is clicked
    - Update Animal with Adoption relationship and *Adoption Date and Adoption Fee*
    - Update Application with an Adoption Relationship and *Adoption Date* and *Adoption Fee*

Add Adoption Application

Task Decomp

Lock Types: Insert new Adoption Application tuple
Number of Locks: 1 Insert
Enabling Conditions: When user clicks on ***Add Adoption Application*** button from the **Animal Dashboard** form.
Frequency: Whenever user wants to add an adoption application
Consistency(ACID): Not critical, order is not critical
Subtasks: Mother task is not needed. No decomposition needed.

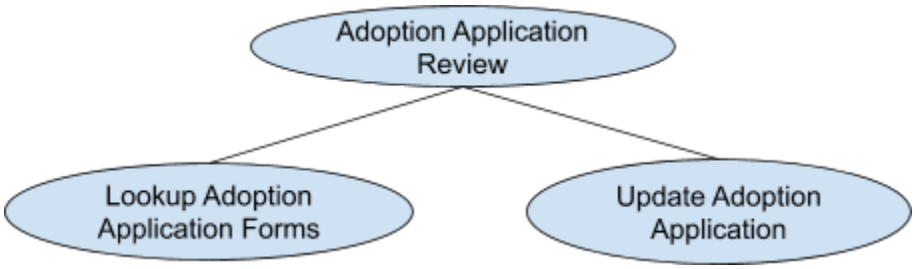
Abstract Code - **Add Adoption Application** Form

- Display **Add Adoption Application** form, allowing user to fill in the following fields:
    - *Applicant's First Name* ($ApplicantFirstName)
    - *Applicant's Last Name* ($ApplicantLastName)
    - *Co-Applicant first name* and *Co-Applicant last name* (Optional) ($CoApplicantFirstName & $CoApplicantLastName)
    - Address: *Street*, *city*, *state*, *zip code* ($Street, $City, $State, $ZIPCode)
    - *Phone number* ($PhoneNumber)
    - *Email address* ($EmailAddress)
    - *Date of Application* ($DateOfApplication)
- While ***Submit Adoption Application*** button is not pushed, do nothing
- When ***Submit Adoption Application*** button is pushed, do the following:

- Verify all required fields are filled in and match their data type. *Co-Applicant first name* and *Co-Applicant last name* fields need to be both filled in or both empty for application to be valid.
- If all fields are valid, then:
  - Insert a new tuple in Adoption Application table
    - Assign a new AdoptionApplication.ApplicationNumber
    - AdoptionApplication.DateOfApplication = '$DateOfApplication'
    - AdoptionApplication.CoApplicantName.FirstName = '$CoApplicantFirstName'
    - AdoptionApplication.CoApplicantName.LastName = '$CoApplicantLastName'
    - AdoptionApplication.State = "Pending Approval"
  - If Adopter record does not exist: Adopter.EmailAddress != '$EmailAddress'; then insert a new tuple in Adopter table.
    - Adopter.EmailAddress = '$EmailAddress'
    - Adopter.ApplicantFirstName = '$ApplicantFirstName'
    - Adopter.ApplicantLastName = '$ApplicantLastName'
    - Adopter.PhoneNumber = '$PhoneNumber'
    - Adopter.Address.Street = '$Street'
    - Adopter.Address.City = '$City'
    - Adopter.Address.State = '$State'
    - Adopter.Address.ZIPCode = '$ZIPCode'
  - Display a Success message; display AdoptionApplication.ApplicationNumber
- Else
  - Display error message with invalid form, keep displaying **Add Adoption Application** form

Adoption Application Review

Task Decomp



Lock Types: 1 read-only lookup of Adoption Applications; 1 update on a tuple of Adoption Application table
Number of Locks: 1 read-only; 1 update
Enabling Conditions: By clicking on ***Adoption Application Review*** button from the **Animal Dashboard**
Frequency: When admin user decides to review the existing adoption applications
Consistency(ACID): Not critical as only admin user is reviewing and updating adoption applications
Subtasks: Mother task is required to coordinate subtasks.

Abstract Code - **Adoption Application Review Form**

- ***Adoption Application Review*** button is visible in **Animal Dashboard** only for Admin users
- User clicks ***Adoption Application Review*** button from **Animal Dashboard**
    - Find each AdoptionApplication where AdoptionApplication.State == "Pending Approval"
        - Display a list of AdoptionApplication ApplicationNumber, Co-Applicant.FirstName, Co-Applicant.LastName, DateOfApplication
        - For each AdoptionApplication displayed, display linked Adopter ApplicantName.FirstName, ApplicantName.LastName, Address (Street, City, State, ZIP Code), PhoneNumber, and EmailAddress
        - Display an ***Approve*** and ***Reject*** buttons for each set of results: Adoption Application + Adopter
    - Upon click of ***Approve*** button
        - Set AdoptionApplication.State = "Approved' for that tuple
        - Remove that AdoptionApplication tuple from **Adoption Application Review**
    - Upon click of ***Reject*** button
        - Set AdoptionApplication.State = "Rejected' for that tuple
        - Remove that AdoptionApplication tuple from **Adoption Application Review**
    - Upon click of ***Close*** button on the **Adoption Application Review** form, form is closed and user is back in **Animal Dashboard**

Lookup Volunteer

Task Decomp

Lock Types: Read-only on Volunteer table
Number of Locks: Single
Enabling Conditions:None
Frequency: Around 10 lookups per day
Consistency(ACID): not critical, order is not critical.
Subtasks: Mother Task is not needed.No decomposition needed.

Abstract Code

- Display form with input fields of *first_name ($FirstName)* and *last_name ($LastName)*
- Wait for user to press ***Lookup*** button
- If ***Lookup*** button is pressed:
- If *first_name* and *last_name* form fields are empty:
    - Display message: “Please enter first name and/or last name. You may enter just the beginning of the name as well.”
- else:
    - Collect data from form fields into variables $FirstName and $LastName
    - Find volunteer in database
      where
        first name attribute starts with $FirstName value and
        last name attribute starts with $LastName value
      order by
        sorted by last name ascending and first name ascending
    - Display a table header with following columns of First name, Last name, Email Address and Phone Number
    - For each $row in rows returned from database, display $row.FirstName, $row.LastName, $row.EmailAddress, $row.PhoneNumber

Display Monthly Adoption Report

Task Decomp

Lock Types: Read-only on *Adoption and Surrender* tables
Number of Locks: Two
Enabling Conditions:None
Frequency: Low - a few reports per month
Consistency(ACID): not critical, order is not critical.
Subtasks: Mother Task is not needed.No decomposition needed.

Abstract Code

- *$CurrentDate* = System Current Date
- *$StartDate* = Date-Add(*$CurrentDate*, -12 Months)
- *$StartDate* = *$StartDate.Year* + *$StartDate.Month* + 1st day of month
- *$EndDate* = *$CurrentDate.Year* + *$CurrentDate.Month* + 1st day of month
- Summarize *Adoptions* in Database
  Count Surrenders
  Count Adoptions
  Grouped by Month, Species, Breed
  Ordered by Month, Species, Breed
- Display table header with following columns of Month, Species, Breed, Number of surrenders, and Number of adoptions
- For each $row in rows returned from database, display $row.Month, $row.Species, $row.Breed, $row.CountOfSurrenders, and Count of adoptions

View Animal Control Report

Task Decomp

Lock Types: Read-only by Animal Control Report and Surrenders tables
Number of Locks: Two
Enabling Conditions: When user clicks view
Frequency: Whenever user wants to view animal control Report
Consistency(ACID): not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code - **View Animal Control Report**

Will display one of two things, the number of animals surrounded by them in a given month, and also any animals adopted in that month that were in the rescue for more than 60 days.
Goal is to meet auditors qualifications
When Report Active:

- Include: Current Month and Previous 6 months
- If animal was turned in by Animal Control**:**
  - Display count of Animals
- Display count of animals turned over by animal control
- If rescued for more than 60 days && animals adopted for corresponding month
  - Display count of animals
- For current Month:
  - Display current information up to the date the report was run

- The count for each month will be a button that displays a drill down report for that month
    - Will include the animals from each category separately
- Information that will be presented will include
    - PetID, species, breed (alphabetically listed array), sex, alterattion status, microchip ID, surrender date.
    - If animal has been in rescue for more than 60 days or more before adoption:
        - Display number of days they have been in rescue
- Information will be sorted based off the following criteria
    - If “animal control surrenders”:
        - sort by pet ID ascending
    - If “listing of animals over 60 days before being adopted”:
        - Sort by number of days in rescue descending

Display Volunteer of the Month

Task Decomp

Lock Types: Read-only on *Volunteer* and *VolunteerHours* tables
Number of Locks: Two
Enabling Conditions:None
Frequency: Low - a few reports per month
Consistency(ACID): not critical, order is not critical.
Subtasks: Mother Task is needed to coordinate subtasks.

Abstract Code

- Find distinct month and year combinations with volunteer hours in database
- For each row returned from the database, populate Drop Down List.
- Make most recent month the current selection in Drop Down List.
- When ***Run*** button is pressed,
- Find *Volunteers* in database
  Sum VolunteerHours
  Orderby Sum of VolunteerHours descending
- Display table header with following columns of First Name, Last Name, Email, and Number of Hours Volunteered
- For each $row of the rows returned from database, display $row.FirstName, $row.LsstName, $row.EmailAddress, and $NumberOfHours volunteered

- If sequential rows have the different $NumberOfHours, $TopCount ++
- If $TopCount == 5 then exit