# Table of Contents

Table of Contents

## Data Types

### User

| Attribute | Data Type | Nullable |
|---|---|---|
| Email | String | Not Null |
| Password | String | Not Null |
| First Name | String | Not Null |
| Last Name | String | Not Null |
| Start Volunteer Date | Date | Not Null |
| Cell Phone Number | int | Not Null |

### Volunteer

| Attribute | Data Type | Nullable |
|---|---|---|
| Email | String | Not Null |

### Admin

| Attribute | Data Type | Nullable |
|---|---|---|
| Email | String | Not Null |

### Dog

| Attribute | Data Type | Nullable |
|---|---|---|
| Name | String | Not Null |
| Sex | String | Not Null |

Table of Contents

| | | |
|---|---|---|
| Alteration Status | String | Not Null |
| Age | int | Not Null |
| Description | String | Not Null |
| Microchip ID | String | Nullable |
| Dog ID | int | Not Null |

### Breed

| Attribute | Data Type | Nullable |
|---|---|---|
| Breed Name | String | Not Null |

### Dog Breeds

| Attribute | Data Type | Nullable |
|---|---|---|
| DogID | int | Not Null |
| Breed Name | String | Not Null |

### Surrender

| Attribute | Data Type | Nullable |
|---|---|---|
| Date | Date | Not Null |
| Reason | String | Not Null |
| Animal Control | String | Not Null |

### Applicant

| Attribute | Data Type | Nullable |
|---|---|---|
| First Name | String | Not Null |

Table of Contents

| | | |
|---|---|---|
| Last Name | String | Not Null |
| Street | String | Not Null |
| City | String | Not Null |
| Zip Code | String | Not Null |
| State | String | Not Null |
| Phone Number | Int | Not Null |
| Email Address | String | Not Null |

### Application

| Attribute | Data Type | Nullable |
|---|---|---|
| Date | Date | Not Null |
| Co-Applicant First Name | String | Nullable |
| Co-Applicant Last Name | String | Nullable |
| Application Number | Int | Not Null |

### Approve

| Attribute | Data Type | Nullable |
|---|---|---|
| Status | String | Not Null |

### Adoption

| Attribute | Data Type | Nullable |
|---|---|---|
| Date | Date | Not Null |
| Fee | Float | Not Null |

Table of Contents

**Expense**

| Attribute | Data Type | Nullable |
|---|---|---|
| Date | Date | Not Null |
| Description | String | Nullable |
| Amount | Float | Not Null |

**Vendor**

| Attribute | Data Type | Nullable |
|---|---|---|
| Name | String | Not Null |

Table of Contents

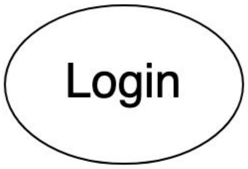
## Business Logic Constraints

1. Breeds is a predefined list maintained by DBA.
2. When adding a dog, all features must be editable. Values cannot be changed for adopted dogs.
3. Users can select multiple breeds for a dog, but if “unknown” or “mixed” is selected, then multi-selection will not be allowed.
4. If and only if “unknown” or “mixed” are chosen for a dog’s breed, the breed of the dog can be updated.
5. Sex can have 3 values: “male”, “female”, or “unknown”. If a dog’s sex is “Unknown”, this can be updated, but “Male” and “Female” values cannot.
6. Alteration Status can only have 2 values: “unaltered” or “altered”. A dog’s alteration status can only be updated after surrender if the dog was “unaltered” when it is surrendered.
7. A dog cannot be adopted until it has been spayed or neutered.
8. A dog must have a microchip implanted before it can be adopted.
9. The value for “Animal Control” has to be “yes” or “no”. Cannot be null.
10. The value of microchip can be updated only if “not currently present” was selected when a dog was first added into the system.
11. A dog’s expenses will only be tracked before it was adopted and after the dog has been surrendered.
12. For existing adopters, no additional adopter contact information needs to be recorded.
13. All adoption applications have a default state of “pending approval”. Application status can have 3 values: “pending approval”, “approved” or “rejected”.
14. Mo will personally review paper applications and will approve the application if she likes what she sees, otherwise reject. At this moment, Mo is the only person who can mark adoption applications as approved, enter adoption info and view reports.
15. An adopter may apply to adopt more than one dog. But each adoption application can only be matched to a single dog.
16. The shelter has a limitation of 15 dogs at a given time, limit can only adjust by DB amdmin with SQL.System only allows adding dog(s) when space is available (under 15 dogs).
17. There is only one expense per dog per vendor per day.
18. Contact info for adopters cannot be updated once entered in the system.
19. Users will get a prompt asking users for different names if User enters Uga as a dog name.
20. Adoption fee is calculated by taking all veterinary expenses and multiplying them by 115% - in other words, the adoption fee is the sum of all veterinary expenses plus 15%. If a dog was brought in by animal control, then the adoption fee is 15% of veterinary expenses
21. The age of a dog must be recorded in months.
22. There will be no concurrent use of the system, that is, we expect only one user to be logged in at a time. Mo might access reports while operations occur, but as this will be read-only, no transactional locking will be necessary.

Table of Contents

## Task Decomposition and Abstract Code

# Login



## Task Decomposition

**Lock Types:** Read-only on User table.
**Number of Locks:** Single.
**Enabling Conditions:**None.
**Frequency:** Medium.
**Consistency(ACID):** Not critical, order is not critical.
**Subtasks:** Mother Task is not needed. No decomposition needed.

## Abstract Code

- User enters *email* (‘$Email’), *password* (‘$Password’) input fields
- If both *username* and *password* input fields are filled:
    - When ***Enter*** button is clicked:
        - If User record is NOT found
            - Go back to **Login** form with error message
        - If User record is found but User.Password != ‘$Password’:
            - Go back to **Login** form with error message
        - Else:
            - Store login information as session variable '$Email'.
            - Go to **Dog Dashboard** Page
- Else *Email* and *password* input fields are NOT filled, display **Login** form with an error message

# Dog Dashboard

## Task Decomposition

**Lock Types:** Read-only on Dog, Breed, Dog Breeds and Surrender tables.
**Number of Locks:** 4.
**Enabling Conditions:** Trigger by successful login.
**Frequency:** High. More than the number of successful logins.
**Consistency(ACID):** Not critical, order is not critical.
**Subtasks:** Mother Task is not needed. No decomposition needed.

Table of Contents

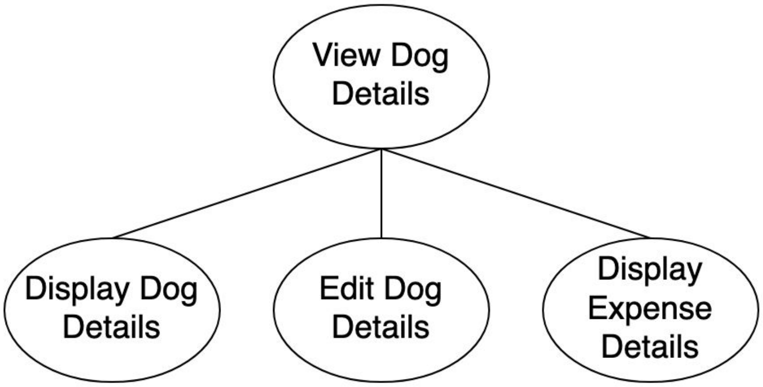
# Abstract Code

- Upon successful login:
    - Display all dogs currently in the shelter. For each dog, show the following information:
        - Dog.Name
        - Dog.Age
        - Dog.Sex
        - Dog.AlterationStatus
        - DogBreeds.BreedName
        - Dog.AdoptabilityStatus. This is derived from Dog.AlterationStatus.
            - If Dog.AlterationStatus is "altered", then the system automatically populates "adoptable".
            - If Dog.AlterationStatus is "unaltered", then the system automatically populates "not adoptable".
    - Display the **number of available spaces**. The information is derived from counting the number of dogs currently in the shelter.
        - The number of currently available spaces in the shelter = 15 - the number of dogs currently in the shelter
            - At any time, 0 <= the number of available spaces <=15
            - If the number of available spaces = 0, the "***Add Dog***" button will be disabled.
            - If the number of available spaces > 0, the "***Add Dog***" button will be clickable.
- Show ***"Filter Dogs", "Sort Dogs", "Add Dog", "Add Adoption Application", "Adoption Application Review (Admin Only)"*** and ***"Reports (Admin Only)"*** tabs.
- Upon:
    - Click on a Dog in the dog list - Jump to **View Dog Details** task.
    - Click ***Filter Dogs*** button -User select an option of ***Adoption Status*** from dropdown menu: Adoptable / Not Adoptable
        - If '$Adoptable' is selected, the **Dog Dashboard** will show the dogs whose Adoptability Status is "adoptable".
        - If '$Not Adoptable' is selected, the **Dog Dashboard** will show the dogs whose Adoptability Status is "not adoptable".
        - Run the **Filter Dogs** task: query for information about dogs based on the selected Adoptability Status.
    - Click ***Sort Dogs*** button - User can select a sorting criterion.
        - By default, sort all dogs by Surrender.Date.
        - Display sorted information based on the selected attribute in ascending order.
        - Only can be sorted by one attribute at a time.
    - Click ***Add Dog*** button - Jump to the **Add Dog** form.

- Click ***Add Adoption Application*** button - Jump to **<u>Add Adoption Application</u>** form.
- Click ***Adoption Application Review (Admin Only)*** button - Jump to **Adoption Application Review (Admin Only)** task.
- Click ***Reports (Admin Only)*** button - Jump to **Reports (Admin Only)** task.
- Click ***Log Out*** button - Invalidate login session and go back to the **<u>Login</u>** form.

# View Dog Details

## Task Decomposition



**Lock Types:** Read and write to Dog and Dog Breeds Table. Read from Breed, Expense and Vendor tables.
**Number of Locks:** Several different schema constructs are needed.
**Enabling Conditions:** Enable by clicking a Dog in the dog list on **<u>Dog Dashboard</u>**.
**Frequency:** Medium.
**Consistency(ACID):** Not critical, order is critical.
**Subtasks**: Display Dog Details and Display Expense Details can be done in parallel. Edit Dog Details must be done after Display Dog Details. Mother task is required to coordinate subtasks.

## Abstract Code

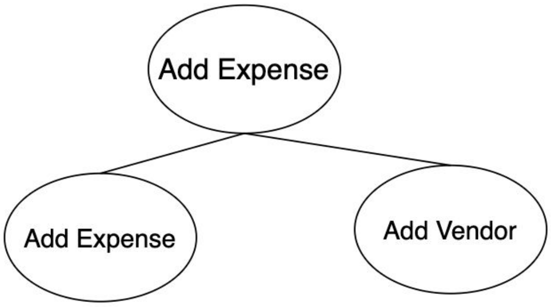
- User clicks on ***View Dog Details*** button from **<u>Dog Dashboard</u>.** The **<u>View Dog Details</u>** screen is divided into 3 sections: **Dog Details**, **Expense Details** and ***Add Adoption****.*
- Run the **Dog Details** task: query for information about the Dog that was clicked on. Display the following information for the selected Dog:
  - Dog.Id
  - Dog.MicroChipId
  - Dog.Description
  - Dog.Name

  - Dog.Age
  - Dog.Sex
  - Dog.AlterationStatus
  - DogBreeds.BreedName
  - Dog.AdoptabilityStatus. This is derived from Dog.AlterationStatus.
    - If Dog.AlterationStatus is "altered", then the system automatically populates "adoptable".
    - If Dog.AlterationStatus is "unaltered", then the system automatically populates "not adoptable".
- ***Edit*** button is shown in the **Dog Details** section that allows a user to edit the following attributes of the selected Dog:
  - If Dog.Sex is "unknown", the Dog.Sex fied is editable.
  - If Dog.MicroChipId is null, the Dog.MicroChipId field is editable.
  - If DogBreeds.BreedName is "unknown" or "mixed", the DogBreeds.BreedName field is editable.
  - If Dog.AlterationStatus is "unaltered", the Dog.AlterationStatus field is editable.
- ***Save*** button is shown in the **Dog Details** section that allows a user to save the changes for the selected Dog.
- Run the **Expense Details** task: query for information about the expenses associated with the Dog that was clicked on. Under the **Expense Details** section, for each expense, show the following information:
  - Expense.Date
  - Expense.Description
  - Expense.Amount
  - Vendor.Name
- Display ***Add Expense*** button that allows a user to add an Expense under the **Expense Details** section.
  - Click on the ***Add Expense*** button - jump to **Add Expense** form.
- ***Add Adoption*** button is shown at the button of the screen.
  - If Adoptability Status is "not adoptable", the ***Add Adoption*** button is disabled.
  - If Adoptability Status is "adoptable", clicking on the ***Add Adoption*** button will open up the **Add Adoption** form.

# Add Expense

## Task Decomposition

Table of Contents



**Lock Types:** Read from Dog, Expense and Vendor tables. Write to Expense and Vendor tables.
**Number of Locks:** Several different schema constructs are needed.
**Enabling Conditions:** Enable by clicking the ***Add Expense*** button in **View Dog Details** screen.
**Frequency:** Different frequency.
**Consistency(ACID):** Critical, order is critical.
**Subtasks**: View Dog task needs to be done first. Add Expense and Add Vendor Tasks can be done in parallel. Mother task is required to coordinate subtasks.

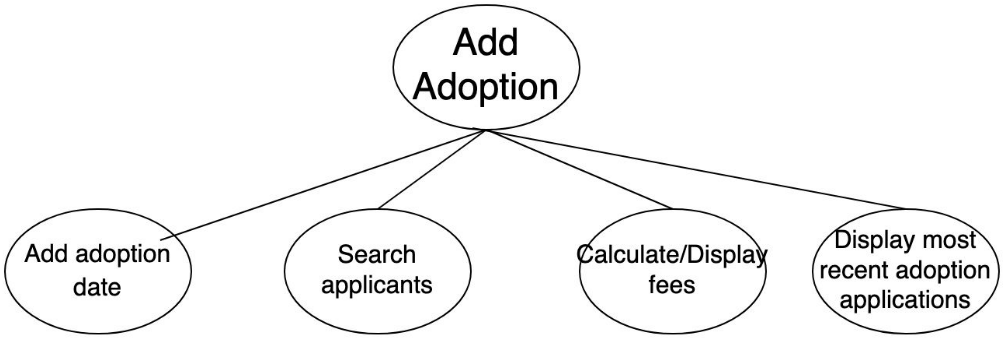
# Abstract Code

- User clicks on ***Add Expense*** button from **View Dog Details:**
- User enters the following information on the form:
  - Automatically populates ('$Dog.Id') input field with the selected dog's ID.
  - Enter ('$Expense.date') input field.
  - Enter ('$Expense.Description') input field.
  - Enter ('$Expense.Amount') input field.
  - Enter ('$Vendor.Name') input field.
- Show ***Save*** button on the form.
  - If '$Dog.Id', '$Expense.date' and '$Vendor.Name' match with one of the expenses that has been recorded in the database, clicking on the ***Save*** button will display an error message "Duplicated expense - a dog can incur only one expense associated with one vendor at a given date."
  - Else, clicking on the ***Save*** button will add this new expense into the database.
    - If '$Vendor.Name' doesn't match with any Vendor.name in the Vendor table, clicking on the ***Save*** button will add '$Vendor.Name' to the Vendor table with a message "A new vendor has been added."

Table of Contents

# Add Adoption

## Task Decomposition



**Lock Types:** Read from Dog, Applicant, Surrender and Application. Write to Add Adoption.
**Number of Locks:** Several different schema constructs are needed.
**Enabling Conditions:** Enable by clicking ***Add Adoption*** button on **View Dog Details** form.
**Frequency:** Low.
**Consistency(ACID):** Critical, order is critical.
**Subtasks**: Search applicants and display the most recent adoption applications that need to be done first. Then calculate fees need to be done before display fees. Add adoption date can be done in parallel to calculate fees /display fees. Mother task is required to coordinate subtasks.

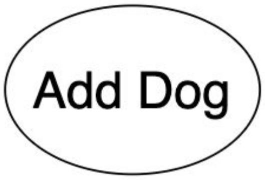
## Abstract Code

- User clicks on ***Add Adoption*** button from **View Dog Details:**
- Run the **Add Adoption** task: query for information about the application and the associated applicant information.
    - Find eligible adoption application by searching applican's last name and the co-applicant's last name (case insensitive, partial match allowed):
        - Display the following information for each applicant shown in the search result:
            - Applicant.LastName
            - Applicant.FirstName
            - Applicant.PhoneNumber
            - Applicant.Email

        - Applicant.Street
        - Applicant.City
        - Applicant.ZipCode
        - Applicant.State
    - Display co-applicant’s Last Name if match is made by co-applicant last name.
- Choose an adopter; display that adopter’s most recent Adoption Application.
- Calculate adoption fee for Applicant; display Adoption Fees.
    - If Surrender.Animal Control is “yes”, then Adoption.Fee equals sum of Expense for this dog  multiplied by 1.15.
    - If Surrender.Animal Control is “no”, then Adoption.Fee equals sum of Expense for this dog multiplied by 0.15.
- Add Adoption Date
- ***Submit Adoption*** Button will save the entered information to the Adoption table.

# Add Dog

## Task Decomposition



**Lock Types**: Write access to Dog, DogBreeds and Surrender table. Read access to the Breed and User table.
**Number of Locks:** Several different schema constructs are needed
**Enabling Conditions**: Enable by successful login and click the ***Add Dog*** button in **Dog Dashboard** when the shelter is not full.
**Frequency:** Low.
**Consistency(ACID):** Critical, order is not critical.
**Subtasks**: Mother Task is not needed. No decomposition needed.

## Abstract Code

- User clicks on ***Add Dog*** button from **Dog Dashboard:**

Table of Contents

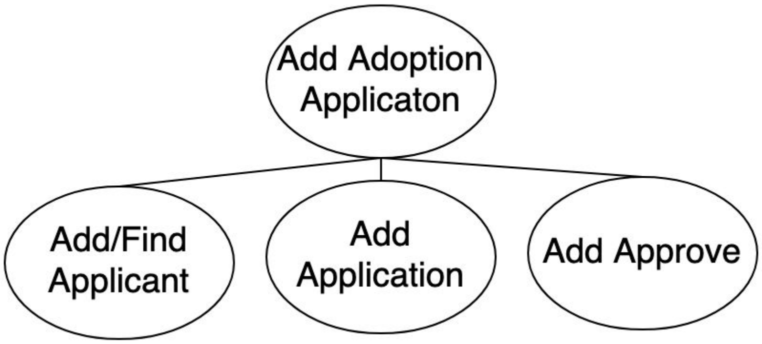
- The following information of a dog need to be entered:
    - System automatically populates ('$Dog_Id') in the input field.
        - The assigned '$Dog_Id' equals the last Dog_Id saved in the Dog table plus 1. The value will be saved as the new Dog.DogId.
    - Enter ('$Dog.MicroChipId') input field which will be saved as Dog.MicroChipId.
        - Can be null. If it is not null, it must be unique.
    - Enter ('$Dog.Description') input field which will be saved as Dog.Description.
        - String limited to 200 characters.
    - Enter ('$Dog.Name') input field which will be saved as Dog.Name.
        - String limited to 20 characters.
    - Enter ('$DogBreeds.BreedName') input field where multiple breed names can be selected from the Breed table. They will be saved as DogBreeds.BreedName.
        - The system populates available options to choose from.
            - If "unknown" or "mixed" is selected, none of the other options can be selected.
            - If "unknown" or "mixed" is not selected, one or multiple values from the predefined Breed table can be selected.
            - If multiple breeds are selected, they should be concatenated into a single value, delimited with a forward slash (/). The combination of breed names should be in alphabetical order.
    - Enter ('$Dog.Sex') input field which will be saved as Dog.Sex.
        - User must choose from one of the 3 options: "male", "female", or "unknown".
    - Enter ('$Dog.AlterationStatus') input field which will be saved as Dog.AlterationStatus.
        - User must choose from one of the 2 options: "unaltered" or "altered".
    - Enter ('$Dog.Age') input field which will be saved as Dog.Age.
        - The age must be recorded in months.
    - In ('$Dog.AdoptabilityStatus') input field, the system automatically populates a value.
        - If '$Dog.AlterationStatus' is "altered", then the system automatically populates "adoptable" in ('$Dog.AdoptabilityStatus') field.
        - If '$Dog.AlterationStatus' is "unaltered", then the system automatically populates "not adoptable" in ('$Dog.AdoptabilityStatus') field.
    - Enter ('$Surrender.Reason') input field which will be saved as Surrender.Reason.
        - String limited to 200 characters.
    - Enter ('$Surrender.Date') input field which will be saved as Surrender.Date.
        - Enter in date time format.
    - Enter ('$Surrender.AnimalControl') input field which will be saved as Surrender.AnimalControl.
        - User must choose from one of the 2 options: "yes" or "no".
    - System automatically populates ('User.email') in the input field based on the User Login information.
    - For auditing purposes, the system needs to keep track of which user originally entered the dog's information.

Table of Contents

- Show ***Save*** button on the form.
    - If '$DogBreeds.BreedName' is "bulldog" and '$Dog.Name' is "Uga", clicking on the ***Save*** button will display an error message "Please enter a different name."
    - If any fields other than '$Dog.MicroChipId' is null, clicking on the ***Save*** button will display an error message "Please enter the required information."
    - Else, clicking on the ***Save*** button will add this new dog into the database.
- Show ***View Dog Details*** button on the form.
    - Upon click, jump to the **View Dog Details** screen.
- Show ***Dog dashboard*** button on the form.
    - Upon click, jump to the **Dog Dashboard** screen.

# Add Adoption Application

## Task Decomposition



**Lock Types:** read Applicant and write on Applicant , Application and Approve tables.
**Number of Locks:** 4
**Enabling Conditions**: Enable by successful login and clicking ***Add Adoption Application*** button in **Dog Dashboard.**
**Frequency:** Medium.
**Consistency(ACID):** Not critical, order is critical.
**Subtasks: Add Applicant** task must be done before adding an application if the applicant is not in the database. Mother task is required to coordinate subtasks.

## Abstract Code

- User clicks on ***Add Adoption Application*** button from **Dog Dashboard:**
- Display **Applicant Form**;

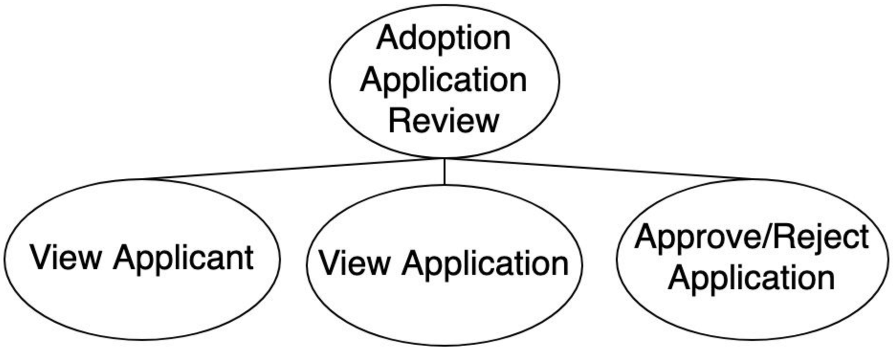
Table of Contents

- Enter ('$Email') input field in **Applicant Form**;
- Search the applicant by matching the '$Email' with the Applicant.email in Applicant table.
  - If '$Email' matches one of the Applicant.email in Applicant table:
    - Autofill **Applicant Form** and display the following information for the existing applicant**:**
      - Applicant**.**FirstName
      - Applicant**.**LastName
      - Applicant**.**Street
      - Applicant**.**City
      - Applicant**.**State
      - Applicant**.**ZipCode
      - Applicant**.**PhoneNumber
  - If '$Email' does not match with any Applicant.email in Applicant table:
    - User needs to create a new applicant by entering the following information on the **Applicant Form**:
      - Enter ('$Applicant.Email') input field.
      - Enter ('$Applicant.FirstName') input field.
      - Enter ('$Applicant.LastName') input field.
      - Enter ('$Applicant.Street') input field.
      - Enter ('$Applicant.City') input field.
      - Enter ('$Applicant.State') input field.
      - Enter ('$Applicant.ZipCode') input field.
      - Enter ('$Applicant.PhoneNumber') input field.
    - Show ***Save*** button on the form.
      - Click on ***Save*** button to add the new applicant's information into the Applicant table.
- Display **Application Form** after the applicant's information is found or entered.
- User enters the following information on the **Application Form**:
  - Enter ('$Application.Date') input field.
  - Enter ('$Application.CoApplicantFirstName') input field. Can be null.
  - Enter ('$Application.CoApplicantLastName') input field. Can be null.
  - System automatically populates "pending approval" in the ('$Approve.Status') input field;
- **Show *Submit*** button on the form.
  - Click on ***Submit*** button will add the new application into the Application table.
- System automatically populates ('$Application.ApplicationNumber') in the input field and displays it on screen. The value will be saved as Application.ApplicationNumber in the Application table.

Table of Contents

# Adoption Application Review (Admin Only)

## Task Decomposition



**Lock Types:** Read Applicant and Application table. Read and write access in the Approve table for User.Admin.
**Number of Locks:** 3
**Enabling Conditions:** Enable by Admin User clicking ***Adoption Application Review (Admin Only)*** button from **Dog Dashboard**.
**Frequency:** Medium.
**Consistency(ACID):** Critical, order is critical.
**Subtasks**: View Applicant and Application Tasks can be done in parallel and need to be done before Approve/Reject Application Task. Mother task is required to coordinate subtasks.

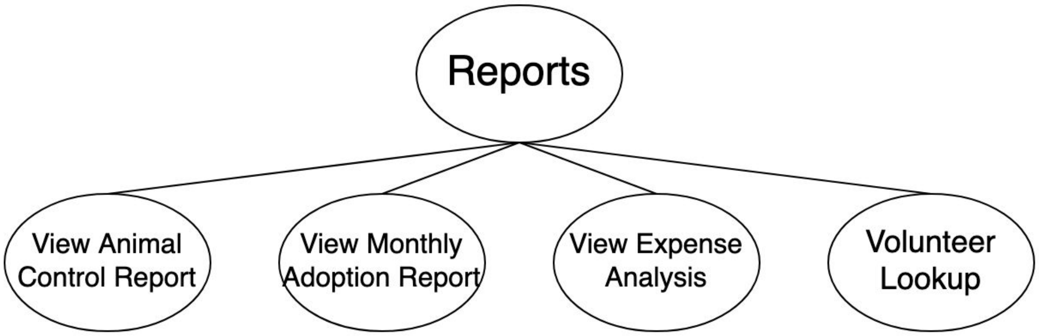
## Abstract Code

- User clicks on ***Adoption Application Review (Admin Only)*** button from **Dog Dashboard:**
- Display all **Application** with Approve**.**Status == “pending approval” with following information:
    - Application.Date
    - Application.CoApplicantFirstName
    - Application.CoApplicantLastName
    - Applicant**.**Email
    - Applicant**.**FirstName
    - Applicant**.**LastName
    - Applicant**.**Street
    - Applicant**.**City
    - Applicant**.**State
    - Applicant**.**ZipCode
    - Applicant**.**Phone Number

- Enter ('$Approve.Status') input field which will be saved as Approve.Status.
    - User must choose from one of the 2 options: "approved" or "rejected".
- Show ***Save*** button on the form.
    - Click on ***Save*** button to save the updated Approve.Status.
- Remove the applications whose Approve.Status have been changed to "approved" or "rejected" from the **Adoption Application Review** Form***.***

# Reports (Admin Only)

## Task Decomposition



**Lock Types:** Read access in database for User, Dog, Breed, Dog breeds, Surrender, Adoption, Vendor and Expense tables.
**Number of Locks:** Several different schema constructs are needed.
**Enabling Conditions:** Enable by Admin User clicking ***Reports (Admin Only)*** button from **Dog Dashboard.**
**Frequency:** High.
**Consistency(ACID):** Consistency is critical.
**Subtasks:** View Animal Control Report, View Monthly Adoption Report, View Expense Analysis, Volunteer Lookup. Mother task is required to coordinate subtasks. All tasks can be done in parallel.

## Abstract Code

- User clicks on ***Reports (Admin Only)*** button from **Dog Dashboard:**
- Show the following tabs: ***View Animal Control Report, View Monthly Adoption Report, Expense Analysis,*** and ***volunteer Lookup.***
    - When click on ***View Animal Control Report***, run the following queries:
        - Filter Surrender**.**AnimalControl with value equals to true

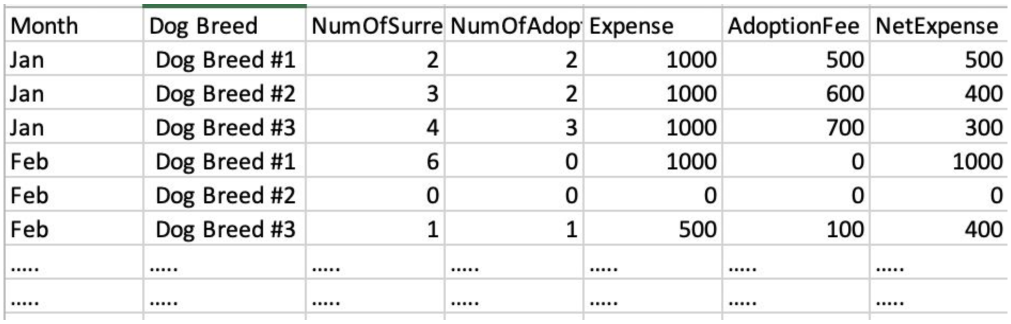
- ▪ Filter Surrender**.**Date to include current month (inclusive report date) and the previous six months
- ▪ Groupby number of dogs surrendered by each month
- ▪ For each month, report will separate into 2 categories:
    - ● Surrendered (sort by Dog ID ascending):
        - o Dog.DogID
        - ○ DogBreeds.BreedName (If a dog has multiple breeds, the breeds should be concatenated into a single value delimited with a forward slash in alphabetical order.)
        - ○ Dog.Sex
        - ○ Dog.AlterationStatus
        - ○ Dog.MicrochipId
        - ○ Surrender.Date
    - ● Adopted (listed by number of days in rescue descending, followed by dog ID descending)
        - o Dog.DogID
        - ○ DogBreeds.BreedName (If a dog has multiple breeds, the breeds should be concatenated into a single value delimited with a forward slash in alphabetical order.)
        - ○ Dog.Sex
        - ○ Dog.AlterationStatus
        - ○ Dog.MicrochipId
        - ○ Surrender.Date
        - ○ If dog's Surrender.Date equals or greater than 60 from Adoption.Date, display the difference in Dates
        - ○ Total Expense.Amount for the Expense.Date with adoption month

○ When click on ***View Monthly Adoption Report***:

- ▪ Display Monthly Adoption Report
    - ● Filter DogBreeds.BreedName by Surrender.Date and Adoption.Date equals last 12 months. This will generate a list of breeds that needs to be included for each month's report (sorted alphabetically).
    - ● Create a table with the following columns:
        - o Month: derived from Surrender.Date and Adoption.Date.
        - o Dog Breed: derived from filtered dog breed list.
        - o Number of surrenders: derived from counting the number of surrenders of each filtered dog breed per month.

Table of Contents

        - Number of adoptions: derived from count number of adoptions of each filtered dog breed per month.
        - Expenses: derived from sum of expenses of each filtered dog breed per month.
        - Adoption fees: derived from sum of adoption fees of each filtered dog breed per month.
        - Net expenses: difference of expenses and adoption fees of each filtered dog breed per month.
    - Each row would be an unique representation of Month and one breed from the filtered list. (starting with January)
    - In case a breed from the previous filtered result has no record for adoption or surrederation, should be default to 0 in the cell.
    - Visual Representation of the Table for 3 filtered dog breeds:



| Month | Dog Breed | NumOfSurre | NumOfAdop | Expense | AdoptionFee | NetExpense |
|---|---|---|---|---|---|---|
| Jan | Dog Breed #1 | 2 | 2 | 1000 | 500 | 500 |
| Jan | Dog Breed #2 | 3 | 2 | 1000 | 600 | 400 |
| Jan | Dog Breed #3 | 4 | 3 | 1000 | 700 | 300 |
| Feb | Dog Breed #1 | 6 | 0 | 1000 | 0 | 1000 |
| Feb | Dog Breed #2 | 0 | 0 | 0 | 0 | 0 |
| Feb | Dog Breed #3 | 1 | 1 | 500 | 100 | 400 |
| ..... | ..... | ..... | ..... | ..... | ..... | ..... |
| ..... | ..... | ..... | ..... | ..... | ..... | ..... |

- When click on ***Expense Analysis***:
    - List all the vendors in the Vendor table. For each vendor, calculate the total spent regardless of date and display the total.
    - Sort the report by total in descending order.
- When click on ***Volunteer Lookup***:
    - Search volunteers by Volunteer.FirstName and/or Volunteer.LastName (case insensitive and allow partial matching)
    - Display the following information for the matched volunteers:
        - Volunteer.FirstName
        - Volunteer.LastName
        - Volunteer.email
        - Volunteer.phone
    - Sort the results by Volunteer.LastName in ascending order and Volunteer.First name in ascending order.