Phase 1 Report

CS 6400 - Summer 2020

Team 012
Jiatang Dong (jdong86@gatech.edu)
Boyan Lu (blu71@gatech.edu)
Ben Li (bli417@gatech.edu)
Pei Tang (ptang39@gatech.edu)

Table of Contents

Data Types and Constraints

User:

Data Types

Attribute	Data type	Nullable
Email	String	Not Null
Password	String	Not Null
LastName	String	Not Null
FirstName	String	Not Null
CellPhone	String	Not Null
StartedDate	Date	Not Null
IsAdmin	Bool	Not Null

Dog:

Data Types

Attribute	Data type	Nullable
DogID	Integer	Not Null
Name	String	Not Null
Breed	String	Not Null
Sex	ENum	Not Null
Altered	Bool	Not Null
BirthDate	Date	Nullable
Description	Text	Nullable
Surrender	String	Nullable
IsAnimalControlSurrender	Bool	Nullable

SurrenderDate	Date	Nullable
SurrenderReason	Text	Nullable
MicrochipID	String	Nullable
CreatedBy	String	Not Null

Expense:

Data Types

Attribute	**Data type**	**Nullable**
DogID	Integer	Not Null
Vendor	String	Not Null
Date	Date	Not Null
Amount	Float	Not Null
Description	Text	Nullable

Breed:

Data Types

Attribute	**Data type**	**Nullable**
Name	String	Not Null

Adopter:

Data Types

Attribute	**Data type**	**Nullable**
Email	String	Not Null
LastName	String	Not Null
FirstName	String	Not Null
CellPhone	String	Not Null

Street	String	Not Null
City	String	Not Null
State	String	Not Null
ZipCode	String	Not Null

Adoption Application:

Data Types

Attribute	Data type	Nullable
ApplicationNumber	String	Not Null
ApplyDate	String	Not Null
CoAppFirstName	String	Nullable
CoAppLastName	String	Nullable
State	ENum	Not Null
DogID	String	Nullable
AdoptedDate	String	Nullable
Fee	Float	Nullable

Business Logic Constrain

User

- Email has to match the format ‘*@*.[com|net|edu|org]’
- Password before hash has to be longer than 6 characters
- LastName, FirstName cannot be empty string
- CellPhone can only contain number and ‘+’, and is not empty string

Dog

- DogId, name and MicrochipID need to be unique
- Breed values needs to come from the breed table

- If breed type is “Unknown” or “Mixed”, it should be the only breed type value
- If breed type is not “Unknown” or “Mixed”, it’s locked
- Sex can only be “Male”, “Female” and “Unknown”
- If sex is not “Unknown”, it’s locked
- If altered or it’s microchipID is null, it can’t be associated to an application
- BirthDate needs to be smaller than today

Expense

- The combination of DogId, vendor and date needs to be unique
- The amount needs to be positive
- Expense date needs be between the dog’s surrender date and adopted date

Adopter

- Email is unique
- Email has to match the format ‘*@*.[com|net|edu|org]’
- CellPhone can only contain number and ‘+’, and is not empty string
- Zip code must be at least 5 digits
- State should be 2 letters
- Contact information is locked after entered

Adoption Application

- Application number is system-generated
- State can only be “Pending”, “Approved” or “Rejected”
- When first created, the default State is “Pending”
- Only admin can update the State to “Rejected” or “Approved”
- DogID is changeable only if status == approved
- When assigning dogID, adoption date and fee need to be assign at the same time
- Only admin can update DogID
- DogID need to unique (we can’t associate the same dog to more than one application)
- ApplyDate needs to be less or equal to today
- AdoptedDate needs to be greater or equal to ApplyDate
- Fee needs to be positive

Task Decomposition and Abstract Code

Login

Lock Types: User table. Read
Number of Locks: Single
Enabling Conditions: None

Frequency: Low
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 data validation is successful for both username and password input fields, then:
        ■ When Enter button is clicked:
                ■ If find a matched User.email
                        ■ If User.password hash matches:
                                ■ Store login information as session variable ‘$UserSession’.
                                ■ Successfully login, jump to Dashboard
                        ■ Else:
                                ■ Go back to Login Form
                                ■ Error message: Wrong password
                ■ Else:
                        ■ Go back to Login Form
                        ■ Error message: Not a registered User
```

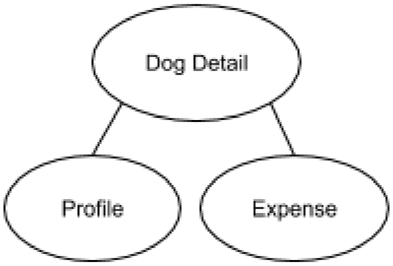
Dashboard

Lock Types: Dog table. Read-only
Number of Locks: Single
Enabling Conditions: Login
Frequency: High
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

```
■ Fetch all the dogs
■ Filter the dogs with selected attribute (default is adoptable) and store it in  ‘$Dogs’
■ Order  ‘$Dogs’ by Dog.SurrenderDate in ascending order
■ For all the ‘$Dogs’:
        ■ Render the profile information including: Dog ID, name, sex etc.
        ■ Render the Detail button
        ■ Jump to Dog Detail View interface when click the dog detail button
■ If the $Dogs.count < shelter capacity:
        ■ Print the Add dog button
                ■ Jump to Dog Form When click the button
■ Render Add Application Button
        ■ Jump to Adoption Application Form when click the Button
■ If User.isAdmin:
    ■ Render Review Applications Button
        ■ Jump to Pending Application View when click the button
■ Render Report Button
    ■ Jump to Reports when click the button
```

Dog Detail View



Lock Types: Dog table. Read; Expense table. Read
Number of Locks: 2
Enabling Conditions: User login
Frequency: Medium
Consistency (ACID): Not critical, order is not critical.
Subtasks: All tasks must be done, but can be done in parallel. Mother task is required to coordinate subtasks. Order is not necessary.

Abstract Code

```
■ User click the Detail Button from each dog item in Dashboard
■ $dogID is passed to here
■ If $dogID doesn't exist:
    ■ Return error: dogID doesn't exist
■ Fetch the information from Dog where Dog.dogID = $dogID
■ Fetch the dog's expense from Expense where Expense.dogID = $dogID
■ Render Edit Button
    ■ Jump to Dog Form when the button is clicked
■ Render Adopt Button
    ■ Jump to Adoption Form when the button is clicked
■ Render the dog information
■ Render Add Expense Button
    ■ Jump to Expense Form when the button is clicked
■ Render expenses in list
```

Dog Form

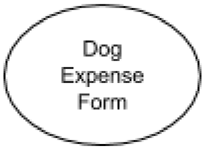
Lock Types: Dog table. Read and Write
Number of Locks: Single
Enabling Conditions: User login
Frequency: Medium
Consistency (ACID): Not critical, order is not critical.

Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

```
■ User click the Add Dog button from Dashboard or Edit Button from Dog Detail View
■ If the request is from Dog Detail View:
        ■ $dogID is passed to here
        ■ If $dogID doesn't exist:
                ■ Return error: dogID doesn't exist
        ■ Grab the $dogID and fetch  information from Dog where Dog.dogID = $dogID
■ Else:
        ■ Create a dummy dog with all fields null
        ■ Toggle on edit mode
■ Render the dog's profile in editable fields
■ Show the Save button
        ■ Trigger the save dog task when the button is clicked
                ■ If name is conflict:
                        ■ Reject
                ■ If Microchip ID is conflict:
                        ■ Reject
                ■ If dog ID is provided:
                        ■ Update the dog in DB
                ■ Else:
                        ■ If the total number of dogs >= shelter capacity:
                                ■ Reject
                        ■ Insert the dog to DB
```

Dog Expense Form



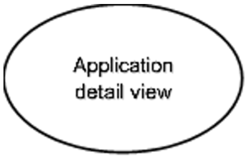
Lock Types: Dog table. Read, Write Expense table. Write
Number of Locks: Single
Enabling Conditions: User Login, dogID exist
Frequency: High
Consistency (ACID): Creating an Expense depends on the existence of the Dog.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

```
■ User click Add Expense from Dog Detail View
■ $dogID is passed to here
■ Render the expense form
■ Render the Save button
```

- When click the ***save*** button
- If the expense is not valid:
    - Reject
- Trigger the **Add Expense** task

Adoption Application Form

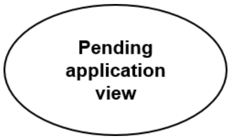


Lock types: Adoption Application table. Write, Adopter table. Read and Write
Number of Locks: Single
Enabling Conditions: User login
Frequency: Medium
Consistency (ACID): Creating an Adoption Application depends on the existence of the Adopter.
Subtasks: None

Abstract Code

- User clicked on ***Add Adoption Application*** button from **Dog Dashboard**
- User enters adopter's email ('$Email') input fields
- If Adopter.email exist:
    - User enters date ('$AdoptedDate') input fields
    - User selects Co-Applicant first name('$CoAppFirstName') select fields and Co-Applicant last name('$CoAppLastName') select fields
    - AdoptionApplication.state == "Pending"
- Else:
    - User enters adopter first name('$FirstName') and last name('$LastName') input fields
    - User enters adopter address, street('$Street'), city('$City'), state('$State'), zip code('$ZipCode') input fields
    - User enters adopter phone number('$CellPhone') input fields
    - User enters date ('$AdoptedDate') input fields
    - User selects Co-Applicant first name('$CoAppFirstName') select fields and Co-Applicant last name('$CoAppLastName') select fields
    - AdoptionApplication.state == "Pending"
- When ***Submit*** button is clicked:
    - Save AdoptionApplication table
    - Return application number('$ApplicationNumber')

Pending Applications View



Lock types: Adoption Application table. Read
Number of Locks: Single
Enabling Conditions: Admin Login
Frequency: Low
Consistency (ACID): Not critical, order is not critical
Subtasks: None

Abstract Code

- User clicked on ***Pending Application*** button from **Dog Dashboard**
- Find AdoptionApplication where AdoptionApplication.state == “Pending”. Save results as (‘$PendingApplication’)
- Sorted (‘$PendingApplication’) by AdoptionApplication.application_number.
- Display the results.

Application Detail View

Lock types: Adoption Application table. Read
Number of Locks: Single
Enabling Conditions: Admin Login
Frequency: Medium
Consistency (ACID): Not critical, order is not critical
Subtasks: None

Abstract Code

- Admin clicked ***Detail*** button from **Pending Application View**
- Find detail by Application Number(‘$ApplicationNumber’)
- Display applicant first name(‘$FirstName’) and last name(‘$LastName’), co-applicant first name(‘$CoAppFirstName’) and last name(‘$CoAppLastName’), apply date(‘$ApplyDate’), phone number(‘$CellPhone’)
- Admin clicked optional button:
    - If admin clicked ***Approve*** optional button:
        - Go to **Approved Applications View**
    - Else if admin clicked ***Reject*** optional button:
        - Go back to **Pending Application View**

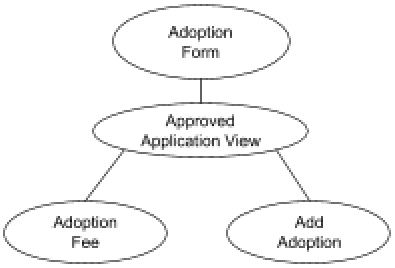
Approved Applications View

Lock types: Adoption Application table. Read
Number of Locks: Single
Enabling Conditions: Admin Login
Frequency: Low
Consistency (ACID): Not critical, order is not critical
Subtasks: None

Abstract Code

```
■ Admin clicked Search button from Adoption Form
■ Admin enters name('$Name') input field
■ if name == ('$FirstName') or ('LastName') or ('$CoAppLastName') or ('$CoAppFirstName'):
        ■ Sorted Adoption Application by name
        ■ Return Adoption Application.application number('$ApplicationNumber')
```

Adoption Form



Lock types: Read-only for admin users
Number of Locks: 3
Enabling Conditions: User login, user is admin
Frequency: Low - same frequency for all
Consistency (ACID): Creating an Adoption depends on the existence of the Dog and Adoption Application.
Subtasks: Search in adoption form then view adopter contact information, then adoption fee and add adoption can be parallel

Abstract Code

```
■ Admin clicked on add adoption button from Dog Detail View
■ $dogID is passed to here
■ Admin enters Last Name('$LastName') into the input field
■ If text validation is successful for adopter and co-applicant on last name, then:
        ■ When Search button is clicked:
              ■If adopter.lastName or co-applicant.lastName matches with $LastName and
                Adoption Application.state == "Approved":
                    ■For each adopter:
                        ■Display adopter(s) or co-applicant(s) with contact information and
                         approval state
              ■Else display "No match"
■ Else lastName input field is invalid, display error message
```

```
  ■ Admin can filter and sort by adopter/co-applicant name, address, and phone
  ■ When admin clicked on Adopter Name link:
■Select and display the application with max of application date where
 adopter.application Number=Adoption Application.application number
■Render adoption application, adoption fee and add adoption
```

Adoption Form - Adoption Fee

Lock types: Read and write for admin users
Number of Locks: Single
Enabling Conditions: admin login required
Frequency: Low
Consistency (ACID): Not critical, order is not critical
Subtasks: None

Abstract Code

```
■ After admin clicked on Adopter Name link:
          ■Render Adoption Fee with $0.00 on the adoption form
■Select and sum all $expenses in the expense form where dog.DogID = adoption.DogID
           ■If dog.animalControl is True:
                     ■Total expenses*0.15
            ■Else Total expenses*1.15
■ Insert and render total application fee to the adoption form
```

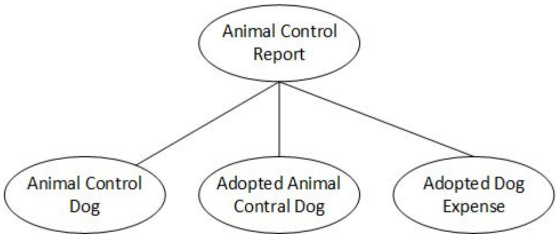
Adoption Form - Add Adoption

Lock types: Write for admin users
Number of Locks: Single
Enabling Conditions: Admin login required
Frequency: Low
Consistency (ACID): Inherit from parent
Subtasks: None

Abstract Code

- After admin clicked on **Adopter Name** link:
    - Render **date input field** on the adoption form
- Admin **enter** adoption date
- If text validation is successful for date, then:
    - If **Submit** button is clicked:
        - Insert adoption.*DogID* and adoption.*adoptionDate* into **adoption application**
- Else *applicationDate* input field is invalid
    - Display error message

Animal Control Report



Lock Types: Dog, Breed and Expense tables. All are Read-only.

Number of Locks: 3

Enabling Conditions: User login, user is admin

Frequency: All have the same frequency.

Consistency (ACID): Not critical, order is not critical.

Subtasks: All tasks must be done. Task 'Animal Control Dog' should be done before 'Adopted Animal Control Dog' and 'Adopted Dog Expense'. Mother task is required to coordinate subtasks.

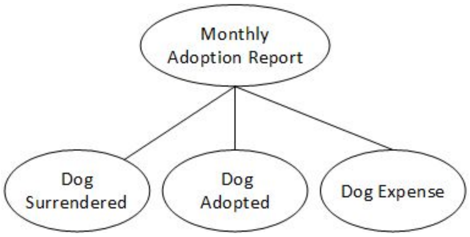
Abstract Code

- Admin clicked on **Animal Control Report** link on **Dog Dashboard**
- FInd Dog brought in by animal controls in current and last 6 months for each month ('$DogSetA).
    - Count the number of dogs in ('$DogSetA) as ('$DogCountA) by month.
    - Display ('$DogCountA) by month in the **Animal Control Dog Count** section.
- Among dogs from ('$DogSetA), find Dog adopted during each month who had spent in the rescue 60 days or more - ('$DogSetB).
    - Count the number of dogs in ('$DogSetB) as ('$DogCountB) by month.
    - Display the number by month in the **Adopted Dog Count** section.
- Calculate total expense for Dog adopted for each month among ('$DogSetA). Group result by month.
    - Display ('$DogExpense) by month in the **Dog Expense** section.
- When a particular month ('$Month_A) in **Animal Control Dog Count** section is clicked:
    - Find Dog details for dogs in ('$DogSetA') of ('$Month_A').
        - Display Dog details for each dog (dog ID, breed, sex, alteration status, microchip ID, and surrender date). Sort data by dog ID ascending.
- When a particular month ('$Month_B) in the **Adopted Dog Count** section is clicked:
    - Find Dog details for dogs in ('$DogSetB') of ('$Month_B').

- ■ For dogs in ('$DogSetB'), calculate the number of days they were in rescue ('$DaysInRescue').
  - ■ Display Dog details for each dog (dog ID, breed, sex, alteration status, microchip ID, and surrender date). Display ('$DaysInRescue'). Sort data by ('$DaysInRescue') descending, then by dog ID descending.

Monthly Adoption Report

Task Decomp



Lock Types: Dog, Breed and Expense tables. All are Read-only.
Number of Locks: 3
Enabling Conditions: User login, user is admin
Frequency: All have the same frequency.
Consistency (ACID): Not critical, order is not critical.
Subtasks: All tasks must be done. Task 'Dog Surrendered' and task 'Dog Adopted' should be done before task 'Dog Expense'. Mother task is required to coordinate subtasks.

Abstract Code

- ■ Admin click on **Monthly Adoption Report** link on **Dog Dashboard**
- ■ Find Dog surrendered in the last 12 months for each month ('$DogSet_Sur).
  - ■ Count the number of dogs from ('$DogSet_Sur) as ('$DogCount_Sur), grouping results by month, then by breed.
- ■ Find Dog adopted in the last 12 months for each month ('$DogSet_Ado).
  - ■ Count the number of dogs from ('$DogSet_Ado) as ('$DogCount_Ado), grouping results by month, then by breed.
- ■ Calculate expense, adoption fees, and net profit for Dog in ('$DogSet_Sur) ∪ ('$DogSet_Ado), save results in ('$DogMoney)
  - ■ Calculate total expense, adoption fees, and net profit from ('$DogMoney) as ('$DogMoney_Sum), grouping results by month, then by breed.
- ■ Join data from ('$DogCount_Sur), ('$DogCount_Ado) and ('$DogMoney_Sum) on month and breed, save results as ('$Mon_Ado_Rep)**.**
  - ■ Sort ('$Mon_Ado_Rep) by month in ascending order, and by breed alphabetically
  - ■ Display ('$Mon_Ado_Rep).

Volunteer Lookup

Task Decomp

Lock Types: User table. Read-only.
Number of Locks: Single
Enabling Conditions: User login, user is admin
Frequency: Low
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Admin click on **Volunteer Lookup** link on **Dog Dashboard**
- Admin enters *search_text* ('$SearchText').
    - When ***Enter*** button is clicked:
        - Find any User (Volunteer) with FirstName or LastName contains ('$SearchText'), case insensitive, sorting by LastName ascending and FirstName ascending. Save results as ('$MatchedUser')
        - Display User's first name, last name, email address, and phone number of ('$MatchedUser').

Expense Analysis

Lock Types: Expense table. Read-only.
Number of Locks: Single
Enabling Conditions: User login, user is admin
Frequency: Low
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Admin click on **Expense Analysis** link on **Dog Dashboard**
- Sum up expenses with grouping by vendor name.
- Sort results by total expenses descending.
- Display the results.