**Elaboration Specification Deliverables**

**Team Magyvers**

**Table of Contents**

1. System Request 4

2. Introduction 5

3. Positioning 5

3.1 Problem Statement 5

3.2 Product Position Statement 6

4. Stakeholder and User Descriptions 6

4.1 Stakeholder Summary 7

4.2 User Summary 8

4.3 User Environment 8

4.4 Summary of Key Stakeholder or User Needs 9

4.5 Alternatives and Competition 10

5. Product Overview 11

6.1 Product Perspective 11

6.2 Assumptions and Dependencies 11

7. Product Features 12

8. Other Product Requirements 12

9. System Requirements 13

10. Use Case Diagrams 16

11. Use Case Description 16

12. Trace Matrix 29

13. Use Cases and Sequence Diagrams 30

14. Class Diagram 82

15. Database Design 83

16. User Interface Navigation Diagram 84

17. Screen Layouts 85

17. Physical Architecture Design 87

18. Design Procedures 88

19. Gantt Chart 89

20. Prototype 91

**1. System Request**

**Project Sponsor**

Dr. Zara Hatami

Telephone: 502-852-4679

Email: zahra.hatami@louisville.edu

**Business need**

Helping create a business model and website to help get Blue Ridge Pet Rescues name out to the community to help bring in donations and help get pets adopted. Creating a functioning website with a user-friendly design and options for people to volunteer and newsletters will help bring attention to Blue Ridge non-profit. With more outreach to the community, and more donations to Blue Ridge it will help get them a functioning building to give more animals the opportunity to be taken in and taken care of before they find their forever home.

**Business Requirements**

Provide a functional user-friendly website for easy access to adoptees, volunteers, or others looking to donate. Should give them access to search for their desired animals or filter to better find animals suited for them.

* User-friendly website
* Searchability of animals
* Donate button featured throughout website

**Business Value**

A website will increase donations which will help provide a facility to put more animals that need care in. It will also increase the number of animals able to be adopted. Volunteer numbers will increase helping with care of animals and fundraising events. More volunteers means keeping costs down.

* More donations means more money for animal care and vet bills
* More volunteers helps take pressure off business owners and reduces help cost
* Special facilities for animals

**Special Issues or Constraints**

Needs to be done by the end of semester.

Make sure the website is secure to prevent any leakage of card information or user data

# **2. Introduction**

The purpose of this document is to collect, analyze, and define high-level needs and features of the Blue Ridge Project. It focuses on the capabilities needed by the stakeholders and the target users, and why these needs exist. The details of how the Blue Ridge Project fulfills these needs are detailed in the use-case and supplementary specifications.

# **3. Positioning**

## **3.1 Problem Statement**

| The problem of | Blue Ridge not having a website for users to adopt or support the organization. |
| --- | --- |
| affects | This affects both users and stakeholders of Blue Ridge. |
| the impact of which is | That Blue Ridge is not reaching more people to know about their organization. Since Blue Ridge doesn’t have a website, customers are supporting and adopting from other similar organizations |
| a successful solution would be | For Blue Ridge to have a website that customers/donators can go to and be able to easily navigate. |

## **3.2 Product Position Statement**

| For | Customers who are thinking about adopting an animal. |
| --- | --- |
| Who | Want to adopt dogs or support Blue Ridge with donations. |
| The (product name) | Blue Ridge animal rescue |
| That | Will find the best homes for animals that have special needs. Blue Ridge connects animals to find their forever home. |
| Unlike | Kentucky Humane Society |
| Our product | Checks on animals and customers after the adoption to make sure that home is still the best fit for the animal. |

# 

# **4. Stakeholder and User Descriptions**

In the past, Blue Ridge animal rescue has been relying on word of mouth and Facebook to spread information about animals. Since Blue Ridge is a non-profit organization, they rely heavily on donations

4.1 Stakeholder Summary

| **Name** | **Description** | **Responsibilities** |
| --- | --- | --- |
| Kim Dehler | Owner and founder of Blue Ridge Reduce and Sanctuary. | Cleans the sanctuary, feeds and checks on animals, work with applicants, and posts on Facebook. |
| Name | Description | Responsibilities |
| Volunteers | People who volunteer for Blue Ridge. | Cleans the sanctuary, feeds and checks on animals. |

## 

## **4.2 User Summary**

| **Name** | **Description** | **Responsibilities** | **Stakeholder** |
| --- | --- | --- | --- |
| People who either want to adopt or foster animals. | This is a group of users who we are trying to target our system of this animal rescue organizations based on. | These people will be the ones we try to find the best match for the animal. They will give notice of animals they are interested in. | People who want to adopt or foster animals is represented by who runs the system. The system is designed for users to stay on Blue Ridge’s website to find an animal they want to take care of. |

## **4.3 User Environment**

Currently Blue Ridge consists of about 10 volunteers and Kim herself. The volunteers all have broad skills to help Kim take care of their animals. As Blue Ridge sanctuary gets bigger, more volunteers will be needed.

The adoption process starts with applications being reviewed to ensure that the animal will be in a fit home. Once the review is done, Blue Ridge will work to figure out what kind of animal the family is looking to adopt. Their main priority is safety for everyone. The adoption process usually takes about 2-3 days depending on the animal. Sometimes this process could take longer if the animal needs more work if they came from bad environments.

Blue Ridge is still a new organization, they don’t have a big enough location to store animals. Currently they run the rescue through Kim’s and foster homes. In March 2022, about 25 animals were in Blue Ridge’s sanctuary. 20 of the animals were living in Kim’s home and the other five living in foster homes.

Blue Ridge’s hardware and software is very simple. For hardware, Kim uses her own personal cell phone which can cause a work/life imbalance. She also uses her home computer, printer, and copier.

The software they use is Facebook, QuickBooks, Excel, AOL email, GroupMe, and Constant Contact. Just using Facebook limits their ability to advertise and talk to people about the animals in Blue Ridge. QuickBooks manages their expenses and income that comes from running Blue Ridge.

## **4.4 Summary of Key Stakeholder or User Needs**

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Advertisement | High | Missing out on donators or animal owners who want to support/adopt animals but don’t know about Blue Ridge. | Using Facebook, email, or cellphone to get in contact with people. | | Expanding to other social media platforms such as Twitter. Having a fully functional website. |
| Website | High | The only way people can find out information about Blue Ridge is Facebook. This might cause users to believe that this organization is not real. | Kim had a friend who was going to help her set up a website, but they were not able to. | | Use students and faculty at UofL to help structure and run the organization’s website. |
| Physical Location | High | Blue Ridge is limited on how many animals they can help by how much room they have to operate. | Currently using Kim’s and other volunteer’s homes to house animals. | | Secure a physical location to house animals. |
| Application Process | Medium | The application process is more difficult than they need to be. The current process could cause issues with people wanting animals that have already been adopted. | Currently you must go to Facebook to find the number or email address for Blue Ridge. The application is being done digitally and manually. | | Fully digitalize the applications. This would decrease the time spent emailing back and forth. |
| Staff training | Low | Multiple people are posting on Facebook. This could run the risk of different writing styles, mistakes in communication with users, and mistakes in the post. | Kim and other volunteers update and post on their organizations Facebook page. | | Training on formatting, recording keeping, and communication for volunteers in the organization. |
|  |  |  |  |  |  |

## **4.5 Alternatives and Competition**

Blue Ridge can decide that they don’t agree with out recommendations and decided to keep on running the orgnaization they way they are currently running it. Another option is that the organization can get help from other outside sources with similar CIS experiences.

Blue Ridge is not the first pet rescue organization. There are other local organizations similar to Blue Ridge such as Kentucky Humane Society, Animal Care Society, and Shamrock Pet Foundation.

Kentucky Human Soicety has been around since 1884. They offer more than just adopting animals. They offer spaying and neutering, dog training classes, pet help line, and pet resorts. Kentucky Human Soicety have events where they remove or lower the adoption fee. They have a lot of physical locations that you can walk in and look at animals. Kentucky Human Society has many good customer reviews about their experiences using the organization.

Another alternative to using Blue Ridge Animal Resuce is Animal Care Society. The organization was Louisvlle’s first no kill shelter being founded in 1984. They offer similar services to Blue Ridge.

Shamrock Pet Foundation was founded in 1993 due to pets being killed in shelters or dying on the streets due to overpopulation. Shamrock offers pet adoption and fostering. Shamrock performs pet neutering and surgery themselves. They also offer Financial Aid Program to help low-income, disable, and old owners who can’t afford to care for their animals.

# **5. Product Overview**

## **5.1 Product Perspective**

The product is a system to help users find animals they are interested or wanting to donate to Blue Ridge. The user will create an account to apply to adopt animals. The website will show other animals thats currently in Blue Ridge to keep them on the website.

## **5.2 Assumptions and Dependencies**

The product assumes that website design options for the creation and customization of the website. The product also assumes that the website will show users other listing on the website to keep them on the website. Another assumption is that people will go to the website to view Blue Ridge.

# **6. Product Features**

The products will have the follow tabs on the website for the organization:

Adopt - This tab is where users will look for animals that they are interested to adopt.

Get Involved - This tab is where users will sign up to volunteer for events or to take care of animals.

Donate - The donate tab is where users will be able to provide donations. Donations would include both money and other animal goods.

About - This tab would give users more information about the organizations goals and missions of the organization.

News/Events - This tab will list the organizations upcoming events and other news the organization wants to share.

# **7. Other Product Requirements**

The website needs to be easy to navigate for users. The website also needs to be easy to update and edit since Kim does not have tech skills. The website also needs to be easy searchable, currently it is hard to find Blue Ridge. The website also needs to have social media connections so posting information about animals between their social media is quick and simple. The website also needs to have payment processing so users can set up donations to support Blue Ridge.

# 

# System Requirements

System requirements including functional and non-functional requirements that help outline what it will take for Blue Ridge animal rescue to be successful. Organized in a hierarchy and by unique ID.

**Functional requirements:**

Donation requirements

ID:SR001 - The donation system shall accept donations from donors.

ID:SR002 - The donation system shall allow for recurring donation options.

ID:SR003 - The donation system shall allow for one-time payment options.

ID:SR004 - The donation system shall allow for editing of donor information.

ID: SR005 - The donation system shall allow for registration of new users.

ID: SR006 - The donation system shall allow for deletion of donor information.

ID: SR007 - The donation system shall allow for anonymous donations.

ID: SR008 - The donation system shall accept saving payment options for later.

Adoption requirements

ID: SR009 - The adoption system shall allow for users to complete an application.

ID: SR010 - The adoption system shall allow for users to sort by animal size.

ID: SR011 - The adoption system shall allow for users to register an account.

ID: SR012 - The adoption system shall display all animals available for adoption to users.

ID: SR013 - The adoption system shall remove adopted animals.

ID: SR014 - The adoption system shall remove animals that are pending adoption.

ID: SR015 - The adoption system shall allow users to check application status.

Volunteer requirements

ID: SR016 - The volunteer system shall allow users to register.

ID: SR017 - The volunteer system shall allow for editing of volunteers.

ID: SR018 - The volunteer system shall allow for deletion of volunteers.

ID: SR019 - The volunteer system shall allow users to create a log in.

ID: SR020 - The volunteer system shall allow users to check application status

ID: SR021 - The volunteer system shall show dates and times needed for volunteers.

Social and community requirements

ID: SR022 - The events tab shall display all planned organized events.

ID: SR023 - The social aspect shall allow users to RSVP to events.

ID: SR024 - Shall allow for users to sign up for a newsletter.

ID: SR025 - Shall allow for editing/deleting of newsletter.

ID: SR026 - Shall allow for new users to register and create account

ID: SR027 - Shall allow for updates to be sent for events (Cancellations/Date changes)

**Non-functional requirements:**

Website performance

ID: SR028 - The website shall be updated constantly and quickly

ID: SR029 - The website shall be user friendly and easy to navigate.

ID: SR030 - The website shall be sleek and not clunky.

ID: SR031 - The website shall allow for 1 to 2 click updates to help non-tech owners/volunteers.

Website Security

ID: SR032 - The website shall protect customer data.

ID: SR033 - The website shall be backed up incase of any risks.

ID: SR034 - The website shall allow for safe donations through a third party system.

ID: SR035 - The website shall keep user data protected through a third party system.

# 

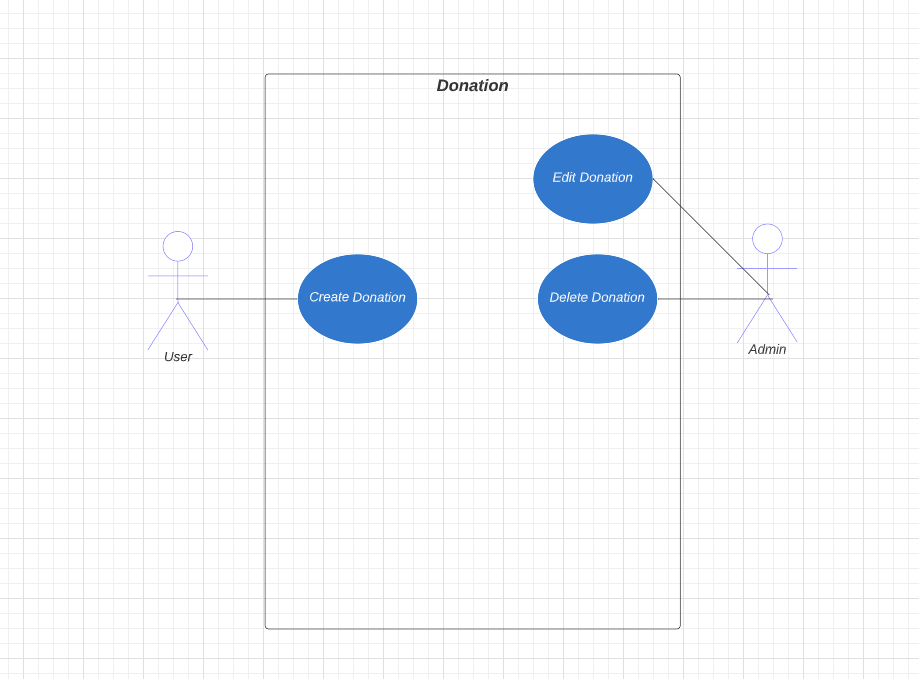
# 

# 

# 

# 

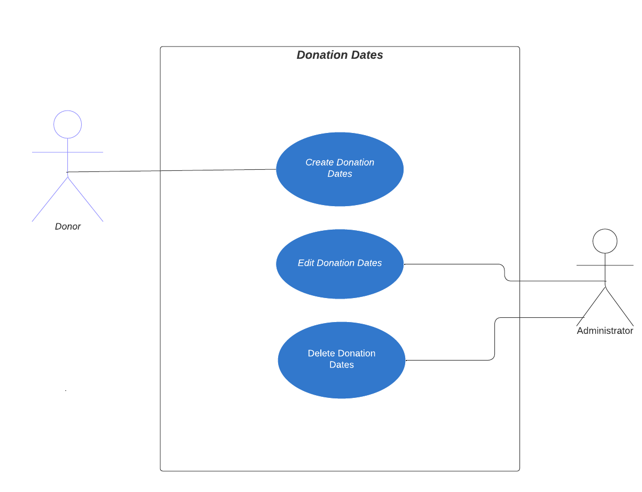
# Use Case Diagram

The diagram below shows the users and system administrator, the actor needs to be able to create donations and the administrator needs to be able to edit and delete donations.

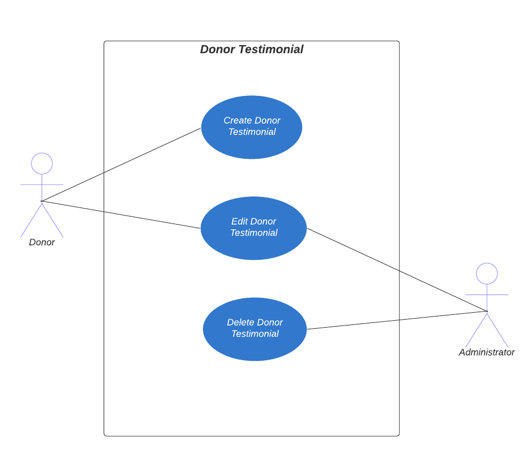
The diagram below shows the users and system administrator, the actor needs to be able to edit donator profile, create donor profile, create existing donor profile, and edit existing profile. The administrator also needs to be able to edit donator profile and edit existing donator profile. The admin also needs to be able to delete both donor profile and existing donor profile.

# 

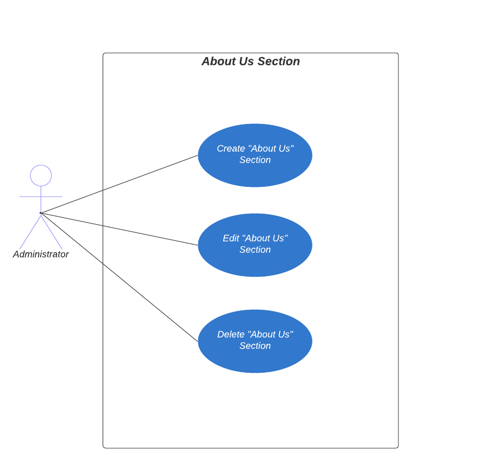
The diagram below shows the donor and system administrator, the donor needs to be able to create donation dates. The administrator needs to be able to edit and delete donation dates.



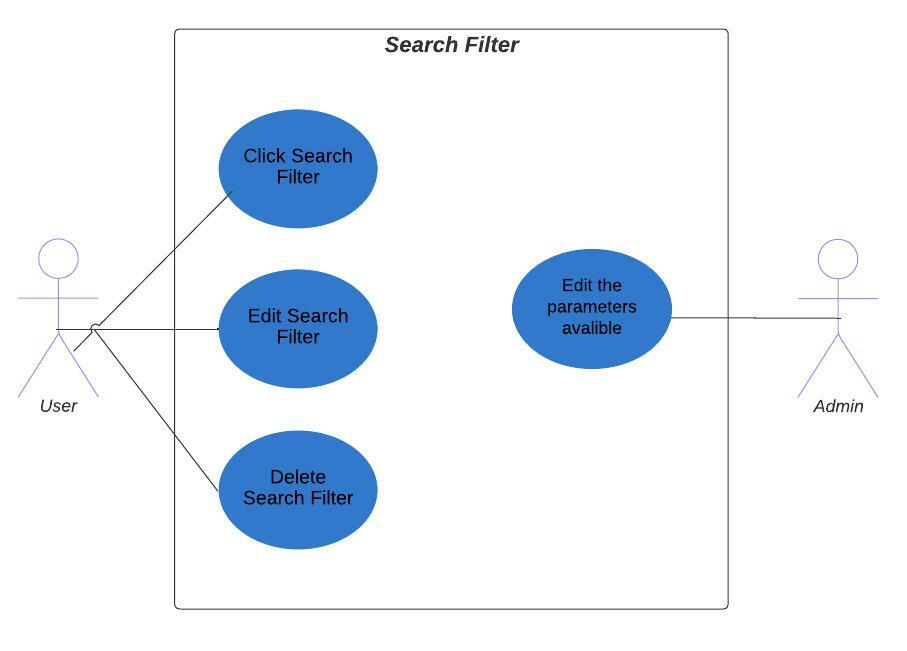
The diagram below shows the donor and system administrator, the donor needs to be able to create donor testimonial and edit donor testimonial. The administrator needs to be able to edit donor testimonial and delete the donor testimonial.



The diagram below shows the system administrator. The administrator needs to be able to create, edit, and delete the “About Us” Section.



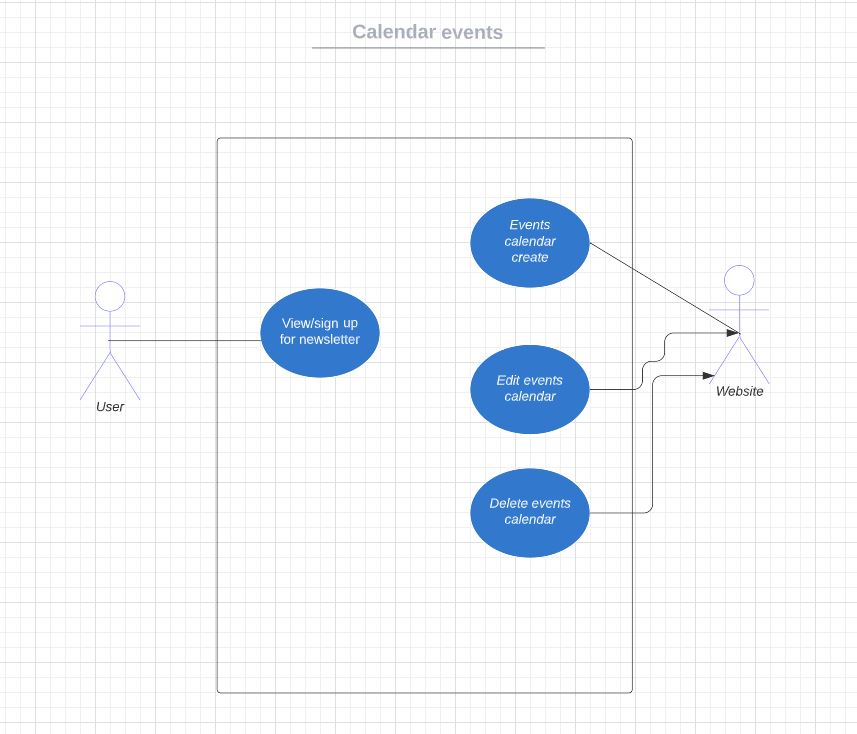
The diagram below shows the user and system administrator, the user needs to be able to click search filter, edit search filter, and delete search filter. The administrator needs to be able to edit the parameters available.



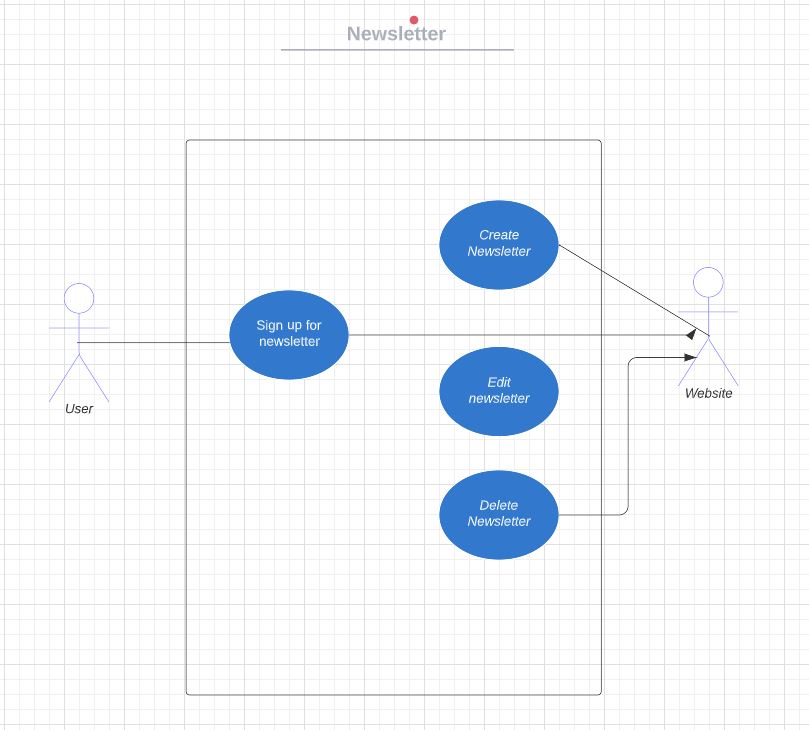
The diagram below shows the user and system administrator, the user needs to be able to use search bar. The administrator needs to be able to create, edit, and delete the pet search box.

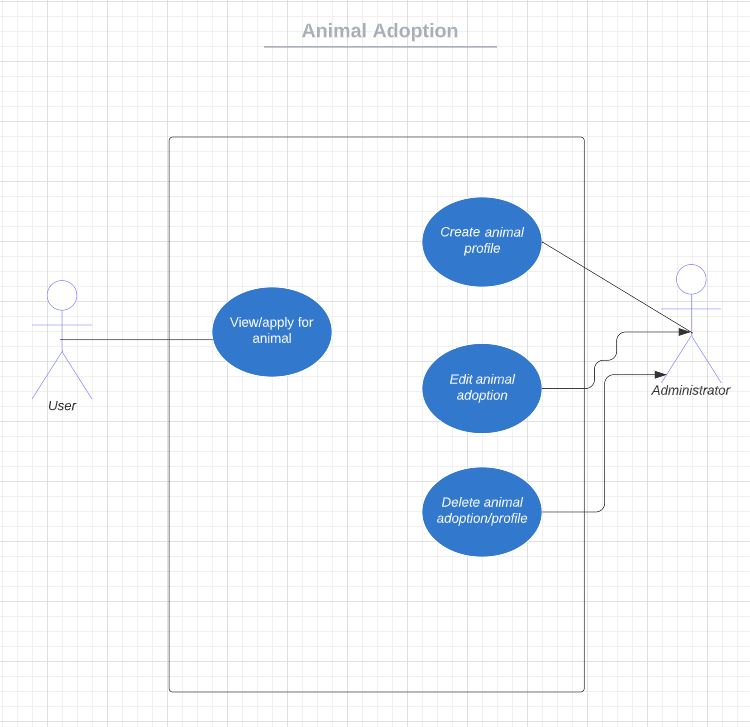
# 

The diagram below shows the actor and system administrator, the actor needs to be able to view and sign up for events. The administrator needs to be able to create, edit, and delete events located on the calendar.

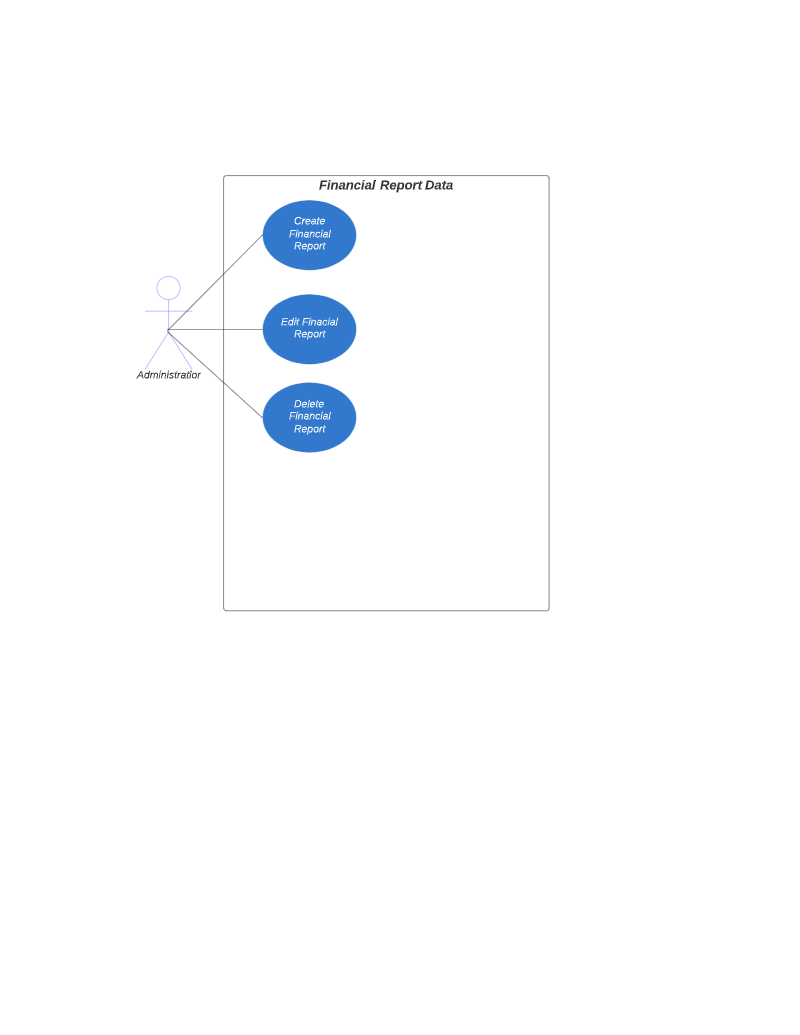


The diagram below shows the actor and system administrator, the actor needs to be able to create newsletter. The administrator needs to be able to create, edit, and delete the newsletter.

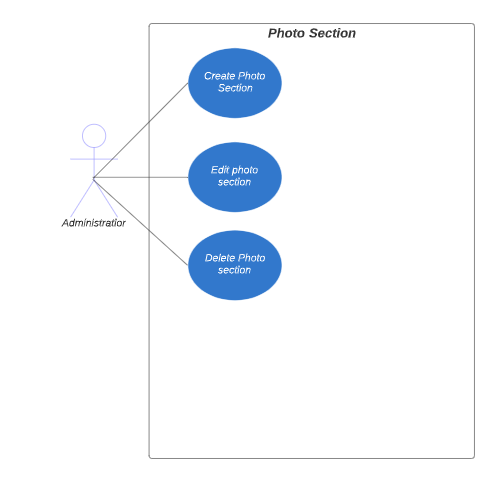




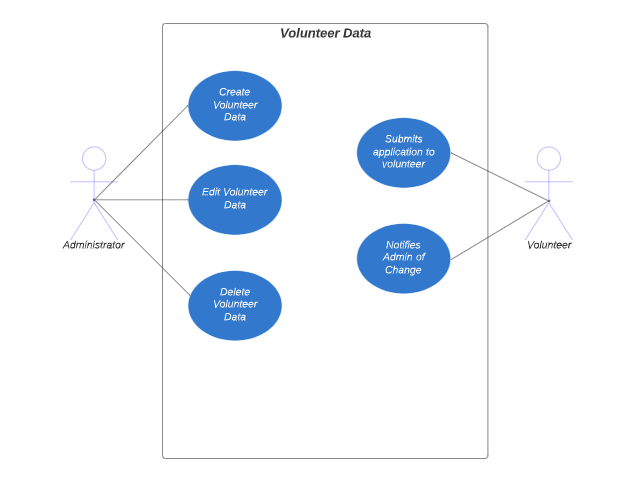
The diagram shows the use case for animal adoption. The administrator creates, edits, and is able to delete the animal adoption profile. While the user is able to view and apply for animals available for adoption.

The diagram below shows the system administrator. The administrator needs to be able to create, edit, and delete the financial reports.

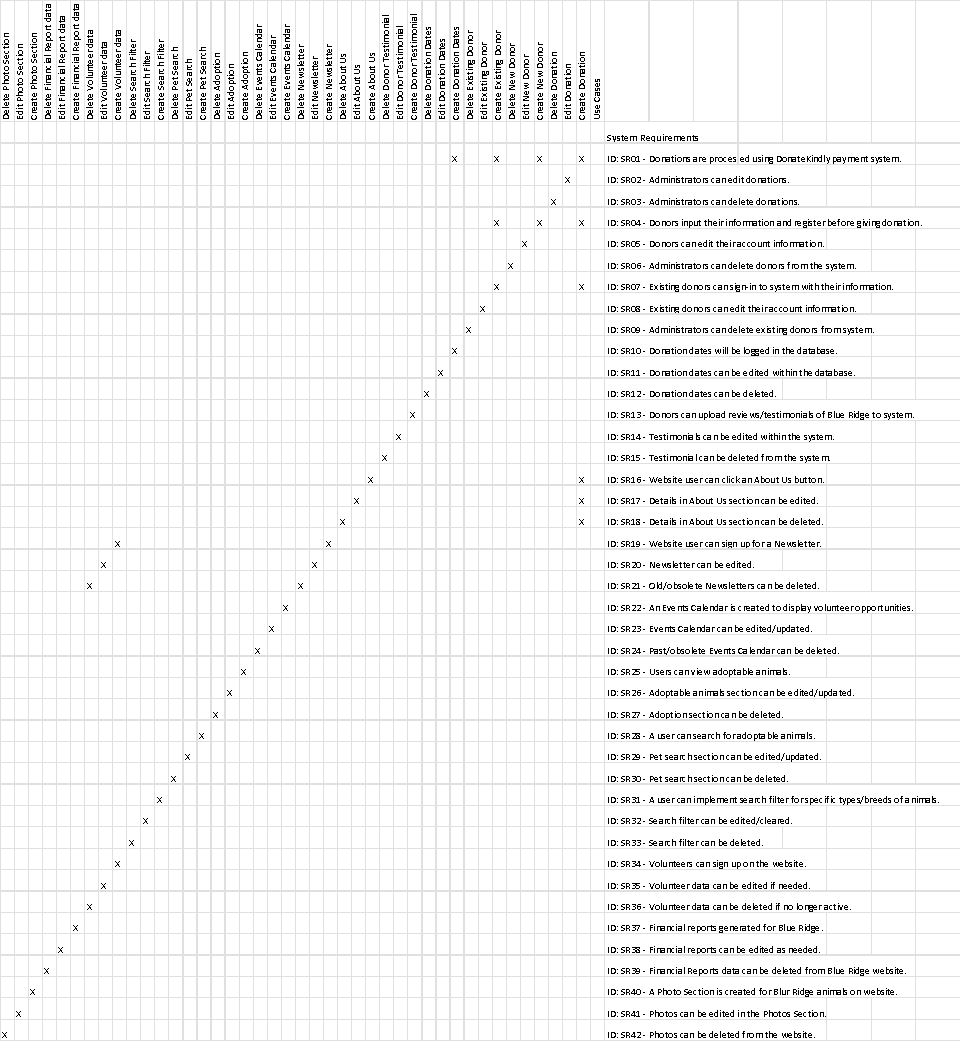
The diagram below shows the system administrator. The administrator needs to be able to create, edit, and delete the photo sections.



The diagram below shows the system administrator and volunteer. The administrator needs to be able to create, edit, and delete volunteer data. The volunteer needs to be able to submit applications to volunteer and notify admin of change.



# Trace Matrix

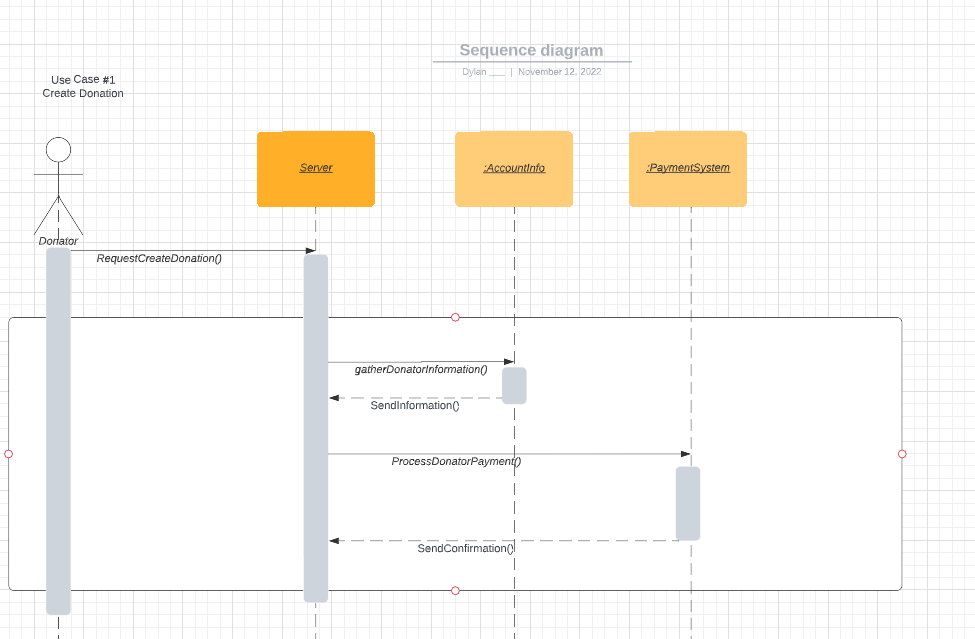


The Trace Matrix portrays the realization of the functional system requirements with the use cases by connecting, or tracing, both to each other. Each X represents the intersection of a system requirement to a use case(s).

# 

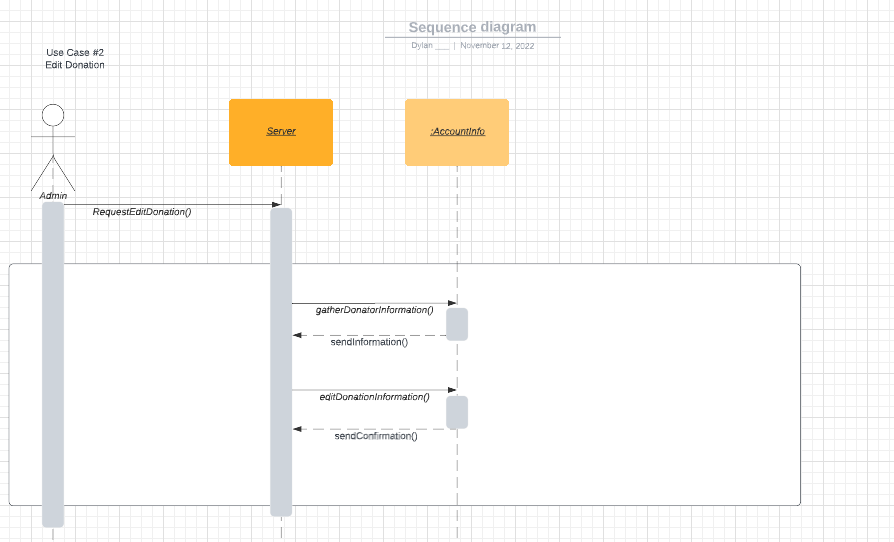
# Use Cases

| Use Case: Create Donation |
| --- |
| ID: 1 |
| Brief Description: Donations are created and processed using Kindful payment system |
| Primary Actors: User |
| Secondary Actors: none |
| Preconditions: User must be able to create donation on website |
| Main Flow:  1. The user clicks on donate button on the home page that prompts the users to enter an amount.  2. The user submits the amount with a confirmation button. |
| Postconditions: The donation will be confirmed and payment will start to begin processing. |
| Alternative flows: The user clicks on the donate tab to display different donation options. |



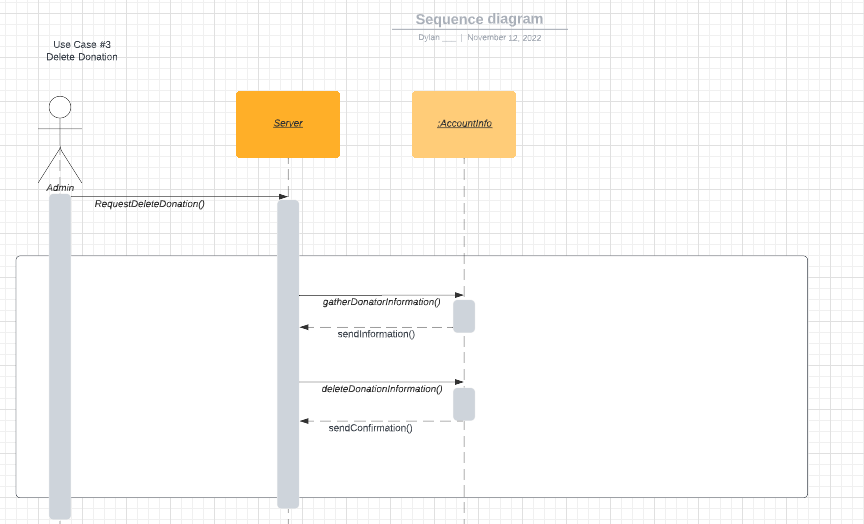
1. Donator request to create donation
2. The server gather donator information
3. The server processes payment to the payment system
4. The payment system sends confirmation back to the server

| Use Case: Edit Donation |
| --- |
| ID: 2 |
| Brief Description: Administrators can edit donation. |
| Primary Actors: Administrators |
| Secondary Actors: None |
| Preconditions: The admin must have privileges to make edits to donations. |
| Main Flow:   1. The administrators goes to confirmed donations tab. 2. Admin goes to donation that needs to be edited and makes edit. |
| Postconditions: Only the administrator can edit the donation. |
| Alternative flows: none |



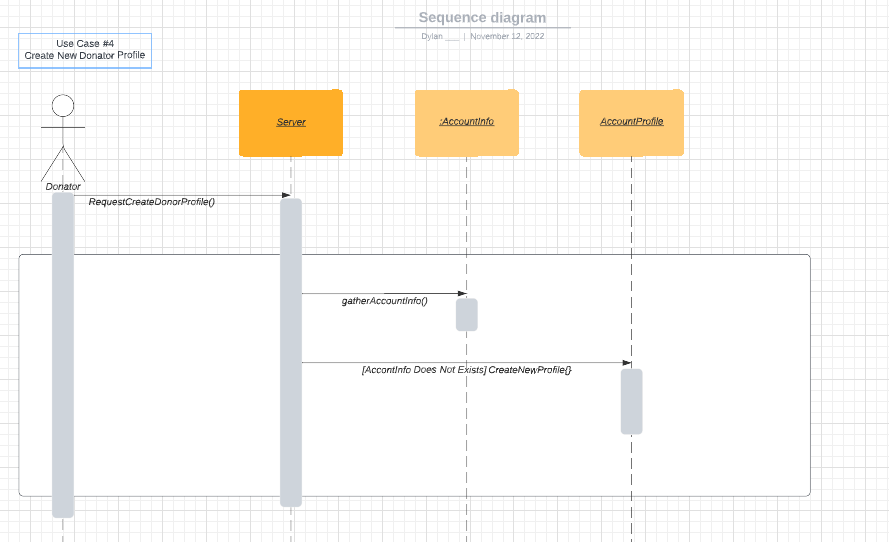
1. Donator request to edit donation
2. The server gather donator information
3. The server edits to the account information
4. The account system sends confirmation back to the server

| Use Case: Delete Donation |
| --- |
| ID: 3 |
| Brief Description: Administrators can delete donation. |
| Primary Actors: Administrators |
| Secondary Actors: None |
| Preconditions: The admin must have privileges to make deletions to donations. |
| Main Flow:   1. The administrators goes to confirmed donations tab. 2. Admin goes to donation that needs to be edited and makes delete. |
| Postconditions: Only the administrator can delete the donation. |



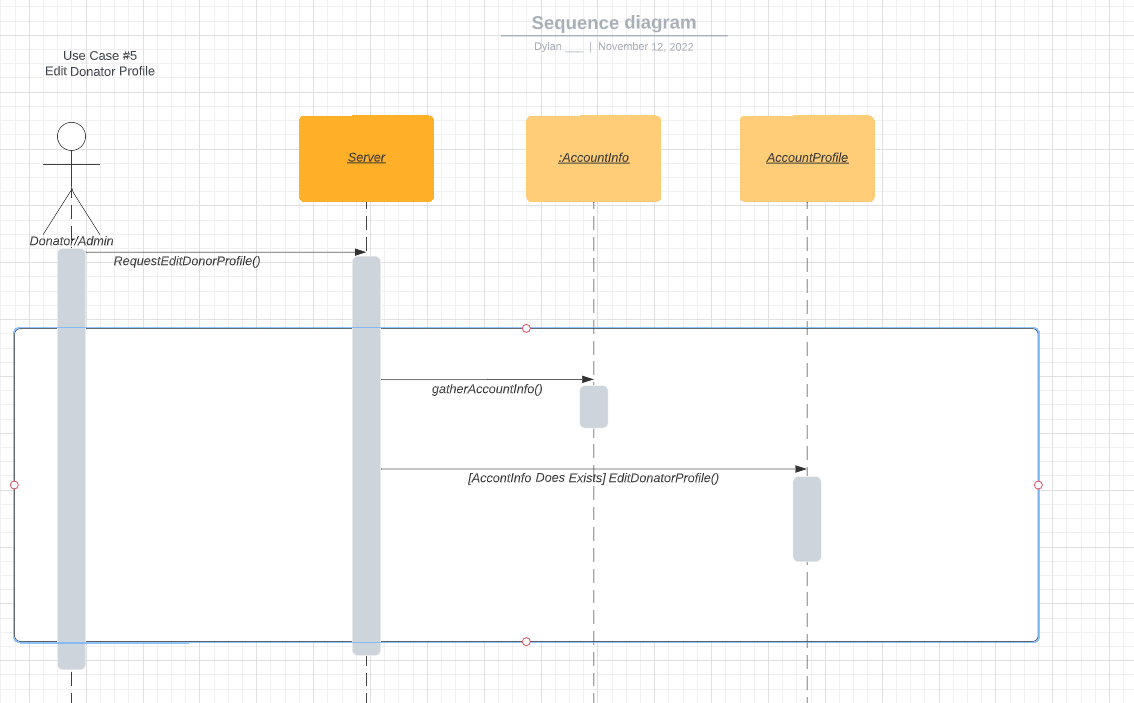
1. Donator request to delete donation
2. The server gather donator information
3. The server deletes to the account information
4. The account system sends confirmation back to the server

| Use Case: Create New Donor Profile |
| --- |
| ID: 4 |
| Brief Description: Donators input their information to create an account before giving donation. |
| Primary Actors: Donor |
| Secondary Actors: None |
| Preconditions: The user must have privileges to create donator profile. |
| Main Flow:   1. Donor goes to donation tab to create profile. 2. Once profile is create, the donator is able to donate. |
| Postconditions: None |



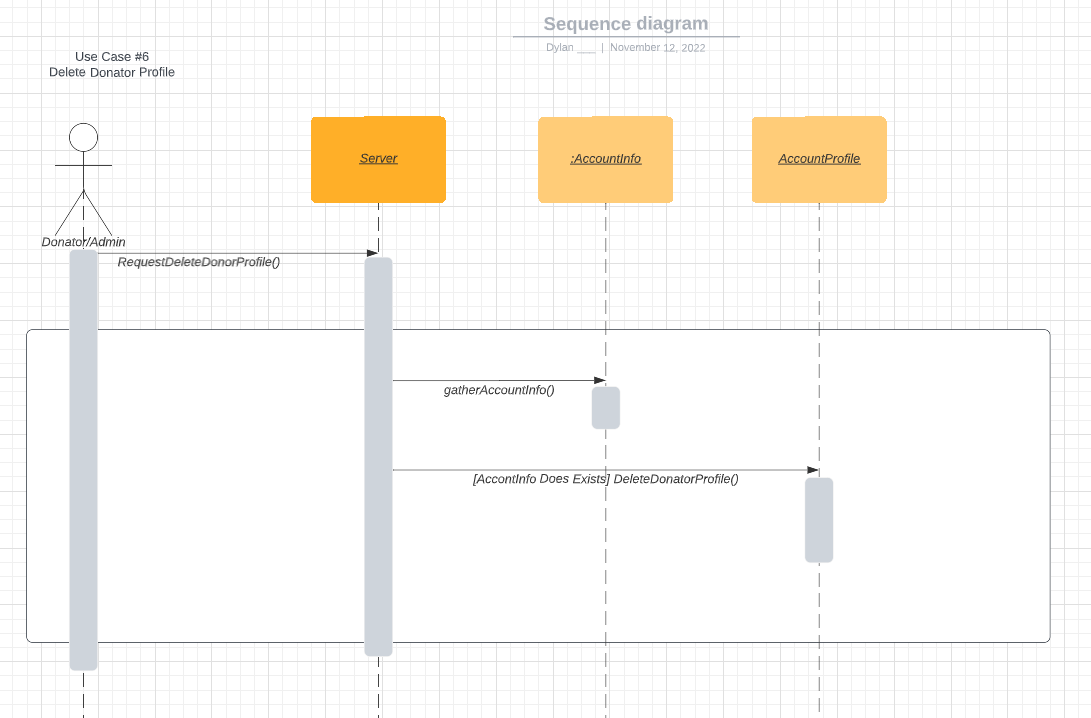
1. Donator request to create donor profile
2. The server gathers donator information
3. Account profile is created if no account already exist.

| Use Case: Edit Donator Profile |
| --- |
| ID: 5 |
| Brief Description: Donators edit their information to update their profile. |
| Primary Actors: Donor |
| Secondary Actors: None |
| Preconditions: The user must have privileges to edit donator profile. |
| Main Flow:   1. User goes to donator profile. 2. User edits their information. 3. User saves their updated information. |
| Postconditions: None |



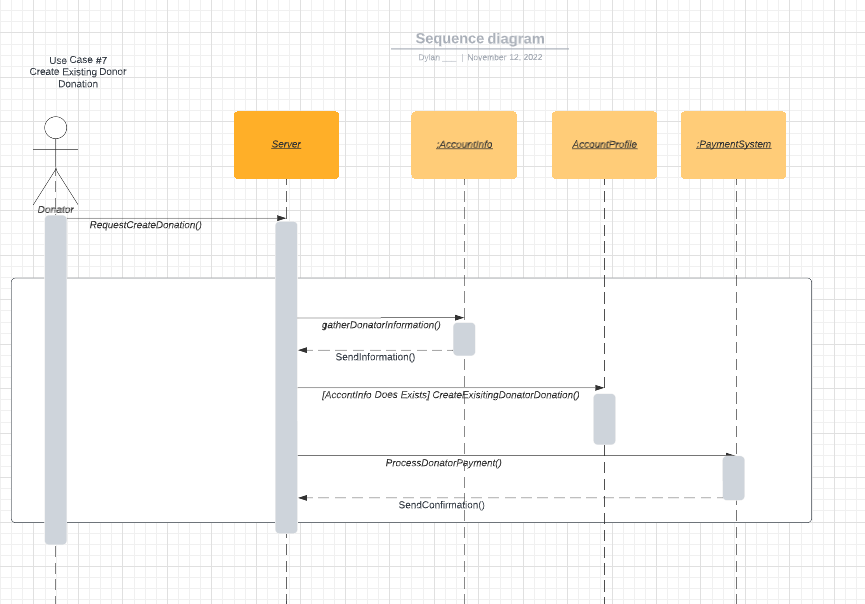
1. Donator request to edit donor profile
2. The server gathers donator information
3. Account profile if edit if account does exist.

| Use Case: Delete Donator Profile |
| --- |
| ID: 6 |
| Brief Description: Donators can request to delete their information from the system. |
| Primary Actors: Donor |
| Secondary Actors: None |
| Preconditions: The user must have privileges to delete their donator profile. |
| Main Flow:   1. User goes to donator profile. 2. User deletes their information. 3. User confirms their deletion.. |
| Postconditions: None |



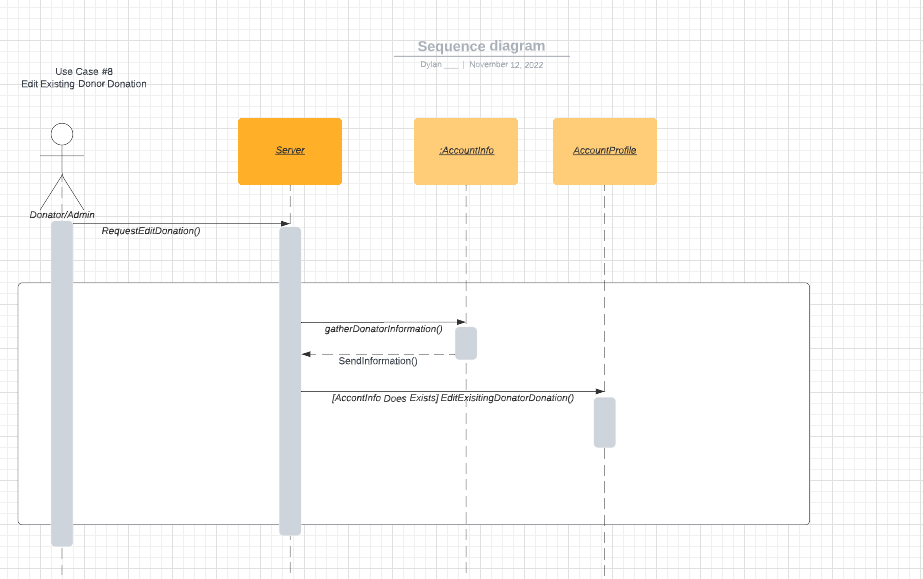
1. Donator request to delete donor profile.
2. The server gathers donator information.
3. If Account profile exists, delete profile.

| Use Case: Create Existing Donor Donation |
| --- |
| ID: 7 |
| Brief Description: Donators can donate again using existing information from their profile. |
| Primary Actors: Donor |
| Secondary Actors: None |
| Preconditions: The user must have privileges to create another donation with their donator profile. |
| Main Flow:   1. User signs to donator profile. 2. User goes to donation tab. 3. The user submits the amount with a confirmation button. |
| Postconditions: None |

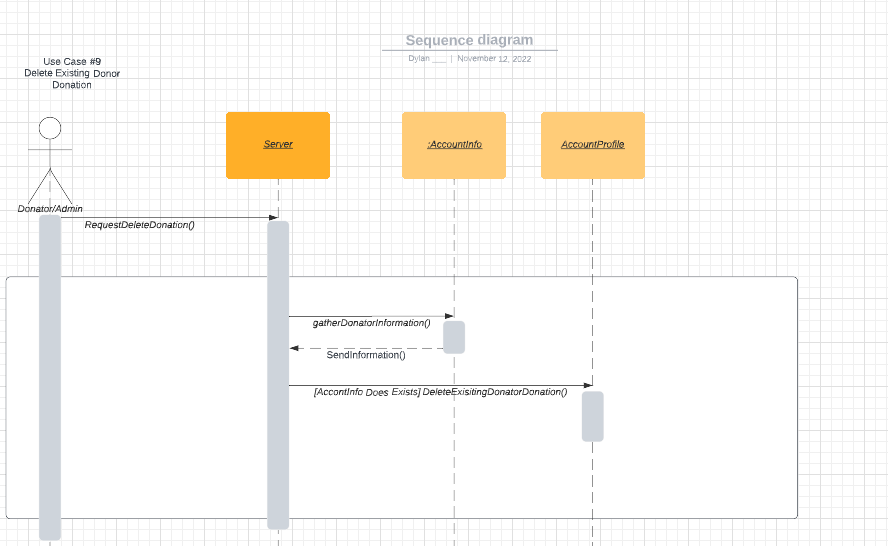


1. Donator request to create donation.
2. The server gathers information.
3. If account already exist, create existing donator donation.
4. Process payment.
5. Send confirmation.

| Use Case: Edit Existing Donor Donation |
| --- |
| ID: 8 |
| Brief Description: Existing donators can edit information from their profile. |
| Primary Actors: Donor |
| Secondary Actors: None |
| Preconditions: The user must have privileges to edit their existing profile. |
| Main Flow:   1. User signs to donator profile. 2. User goes to profile tab to edit profile. 3. The user edits their information with a confirmation button. |
| Postconditions: None |

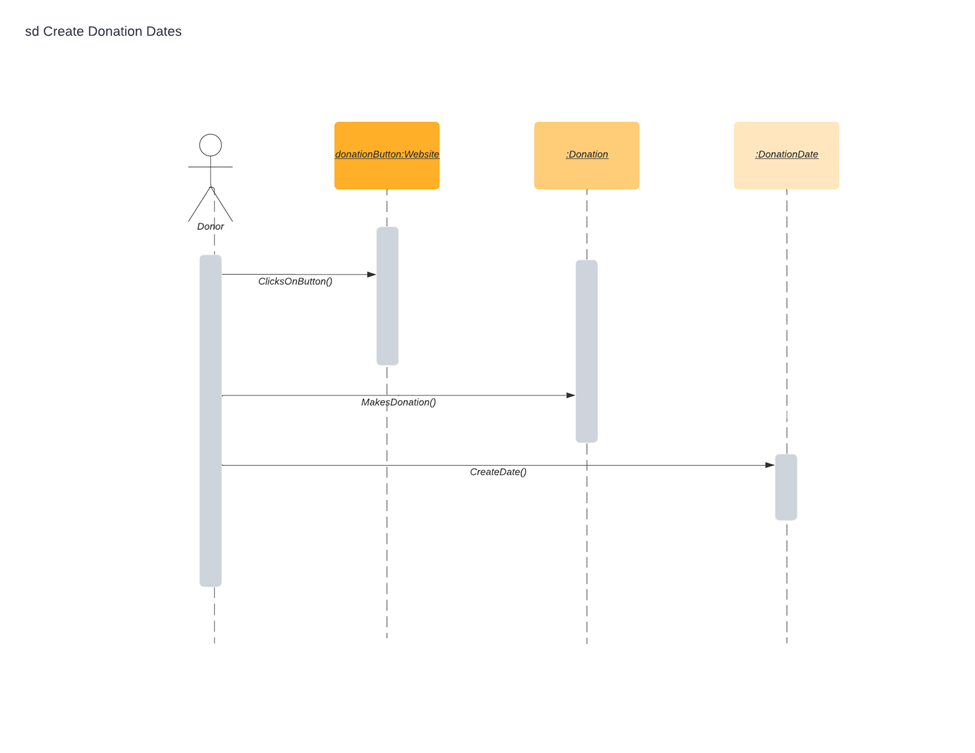
1. Donator request to edit donation.
2. Server gather information.
3. If account does exist, edit existing donator donation.

| Use Case: Delete Existing Donor Donation |
| --- |
| ID: 9 |
| Brief Description: Donators information can be deleted from their profile. |
| Primary Actors: Administrator |
| Secondary Actors: None |
| Preconditions: The admin must have privileges to delete their existing profile. |
| Main Flow:   1. Admin goes to profile tab to delete profile. 2. The admin deletes user information with a confirmation button. |
| Postconditions: None |



1. Donator request to delete donation.
2. Server gather information.
3. If account does exist, delete existing donator donation.

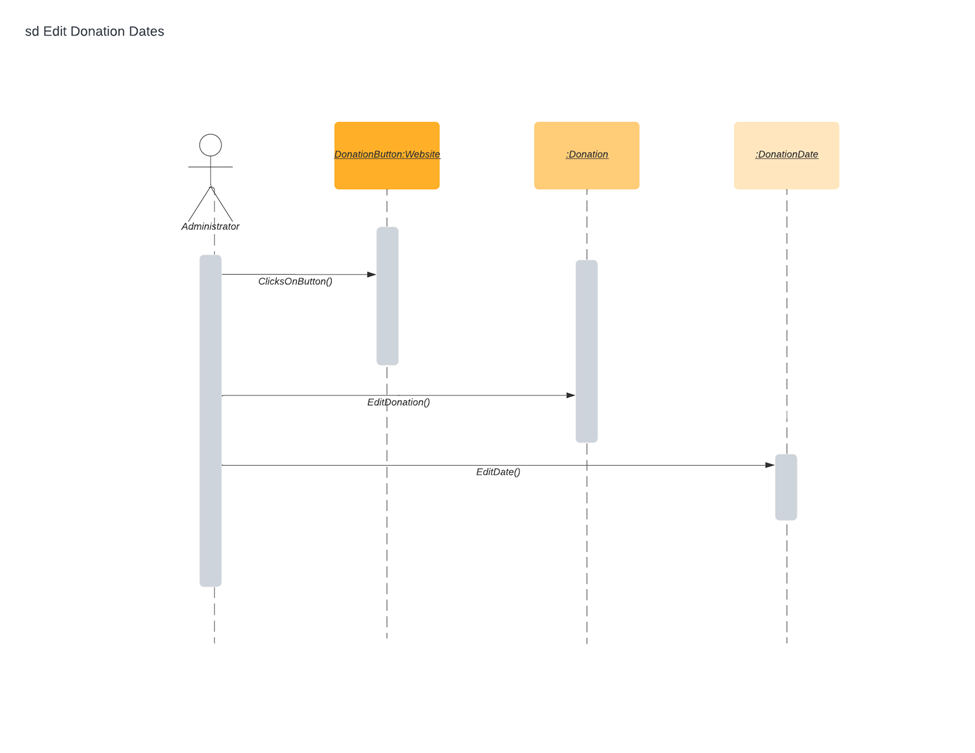
| Use Case: Create Donation Dates |
| --- |
| ID: 10 |
| Brief Description: This use case describes how donation dates are created. |
| Primary Actors: Donor |
| Secondary Actors: none |
| Preconditions: The user must donate to create a date of donation. |
| Main Flow:  1. The use case starts when a user clicks the donation button on the website and gives a donation.  2. A date is assigned to that donation.  3. A donation date is created. |
| Postconditions: Only the user who donated can create a donation date. |
| Alternative flows: none |

1. The use case starts when a user clicks the donation button on the website and gives a donation.

2. A date is assigned to that donation.

3. A donation date is created.

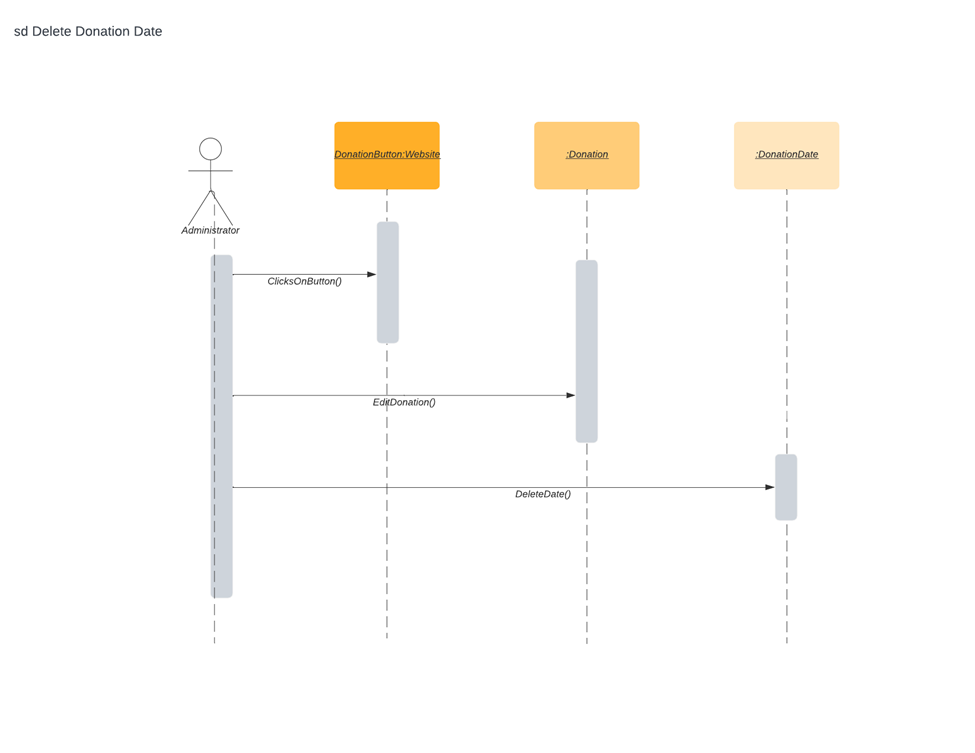
| Use Case: Edit Donation Dates |
| --- |
| ID: 11 |
| Brief Description: This use case describes how donations dates can be edited within the database. |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: A donation must be previously made to create a donation date to edit. |
| Main Flow:  1. The use case starts when the administrator clicks on a donation the donor made.  2. The administrator edits donation, replacing the original donation. |
| Postconditions: Only an administrator can edit a donation. |
| Alternative flows: none |



1. The use case starts when the administrator clicks on a donation the donor made.

2. The administrator edits donation, replacing the original donation.

| Use Case: Delete Donation Dates |
| --- |
| ID: 12 |
| Brief Description: This use case describes how donations dates can be deleted. |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: User must have first donated to create a donation date to delete. |
| Main Flow:  1. This use case starts when administrator clicks on donation made button.  2. The administrator deletes donation and date created when user donated. |
| Postconditions: Only administrators can delete donation dates. |
| Alternative flows: none |

****

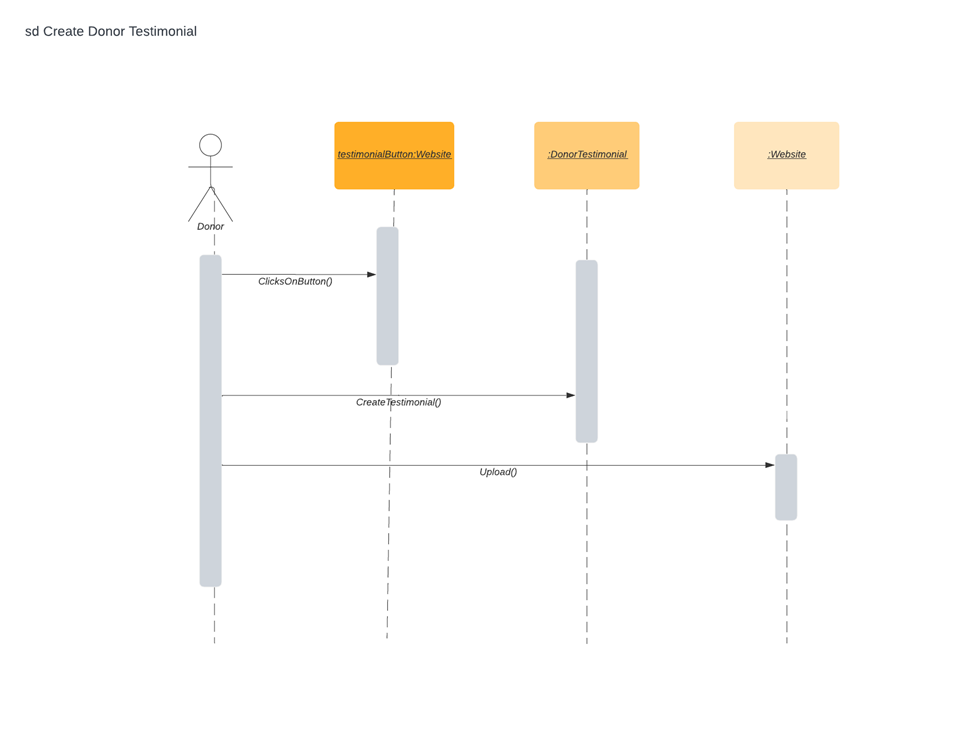
1. This use case starts when administrator clicks on donation made button.

2. The administrator deletes donation and date created when user donated.

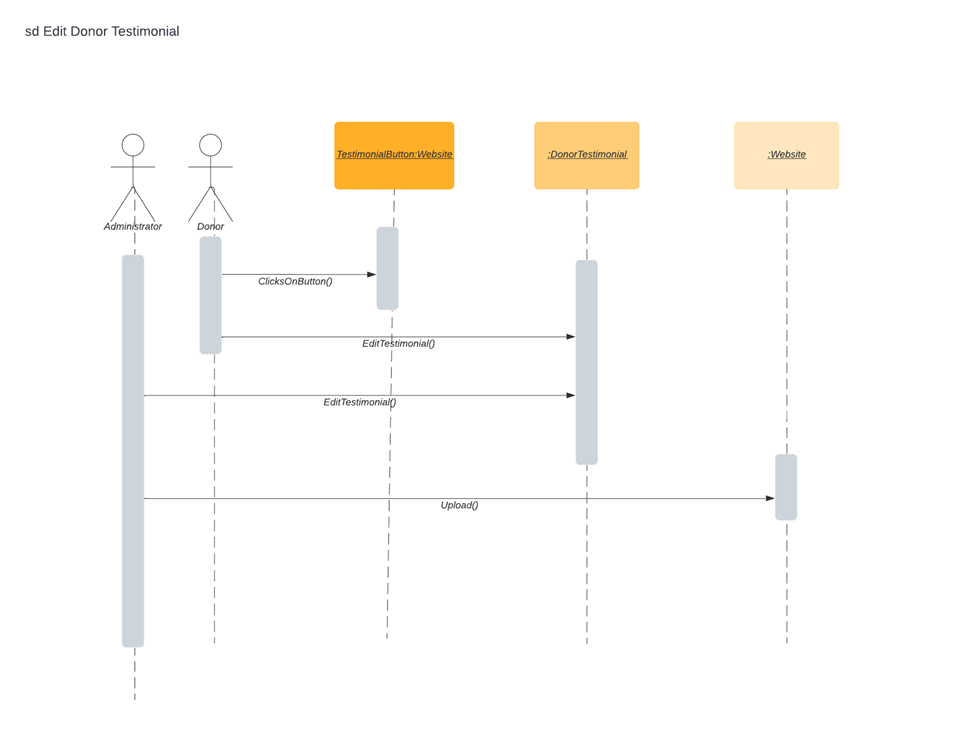
| Use Case: Create Donor Testimonial |
| --- |
| ID: 13 |
| Brief Description: This use case describes how donors can create and upload reviews/testimonials of Blue Ridge. |
| Primary Actors: Donor |
| Secondary Actors: none |
| Preconditions: User must donate to Blue Ridge before uploading a testimonial. |
| Main Flow:  1. The use case starts when a donor clicks the testimonial button.  2. The donor writes a review/testimonial of Blue Ridge.  3. The donor uploads their review/testimonial to the website. |
| Postconditions: Only users that donated can upload a testimonial. |
| Alternative flows: none |

1. The use case starts when a donor clicks the testimonial button.

2. The donor writes a review/testimonial of Blue Ridge.

3. The donor uploads their review/testimonial to the website.****

| Use Case: Edit Donor Testimonial |
| --- |
| ID: 14 |
| Brief Description: This use case describes how reviews/testimonials can be edited on the website. |
| Primary Actors: Donor |
| Secondary Actors: Administrator |
| Preconditions: Donor must create testimonial and upload it before it can be edited. |
| Main Flow:  1. The use case starts when a testimonial is created.  2. The testimonial is uploaded to the website.  3. The donor or administrator can then edit and upload the updated testimonial to website. |
| Postconditions: Only users who created testimonials can make edits. |
| Alternative flows: none |

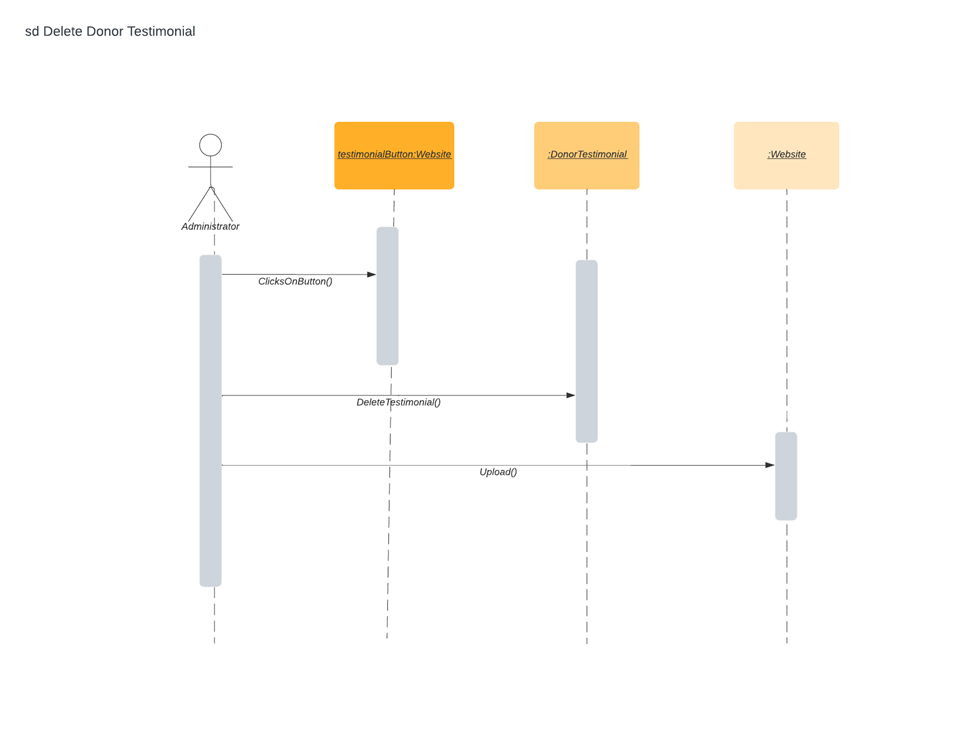
****

1. The use case starts when a testimonial is created.

2. The testimonial is uploaded to the website.

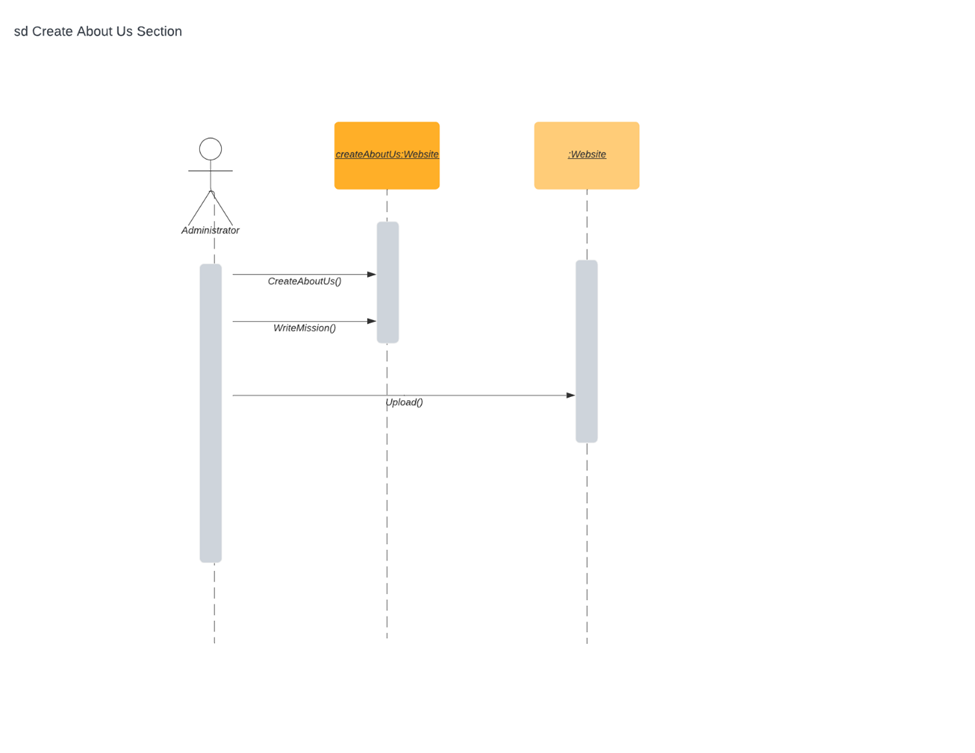
3. The donor or administrator can then edit and upload the updated testimonial to website.

| Use Case: Delete Donor Testimonial |
| --- |
| ID: 15 |
| Brief Description: This use case describes how testimonials can be deleted. |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: Donors must create a donation on the website before testimonials can be deleted. |
| Main Flow:  1. The use case starts when an administrator clicks on the testimonial button.  2. The administrator can delete donor testimonials from the website. |
| Postconditions: none |
| Alternative flows: none |

1. The use case starts when an administrator clicks on the testimonial button.

2. The administrator can delete donor testimonials from the website.

| Use Case: Create “About Us” Section |
| --- |
| ID: 16 |
| Brief Description: This use case describes how an About Us section is created on the website. |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: none |
| Main Flow:  1. The use case starts when an administrator creates an About Us section on website.  2. The administrator writes about the mission and vision of Blue Ridge.  3. The administrator uploads the About Us section. |
| Postconditions: Only an administrator can create the About Us section. |
| Alternative flows: none |

****1. The use case starts when an administrator creates an About Us section on website.

2. The administrator writes about the mission and vision of Blue Ridge.

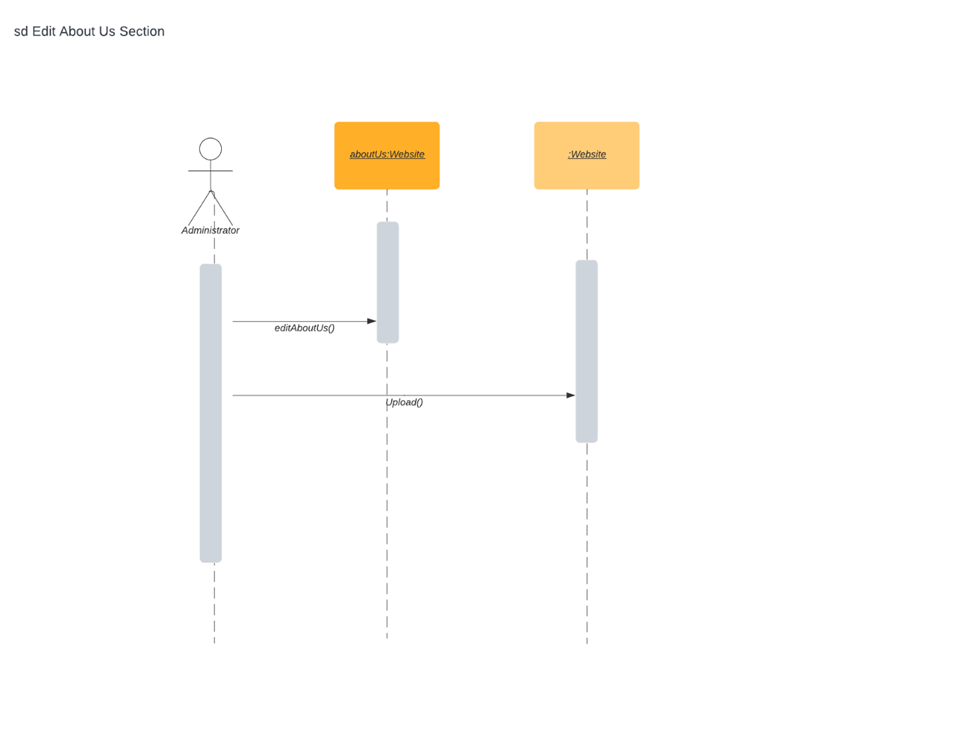
3. The administrator uploads the About Us section.

| Use Case: Edit “About Us” Section |
| --- |
| ID: 17 |
| Brief Description: This use case describes how the About Us section on website can be edited. |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: Administrator must create an About Us section in order to edit it. |
| Main Flow:  1. The use case starts when an administrator creates an About Us section on the website.  2. The administrator can edit the About Us section.  3. The administrator uploads the changes made to the About Us section. |
| Postconditions: Administrator must have previously created About Us section to edit. |
| Alternative flows: none |

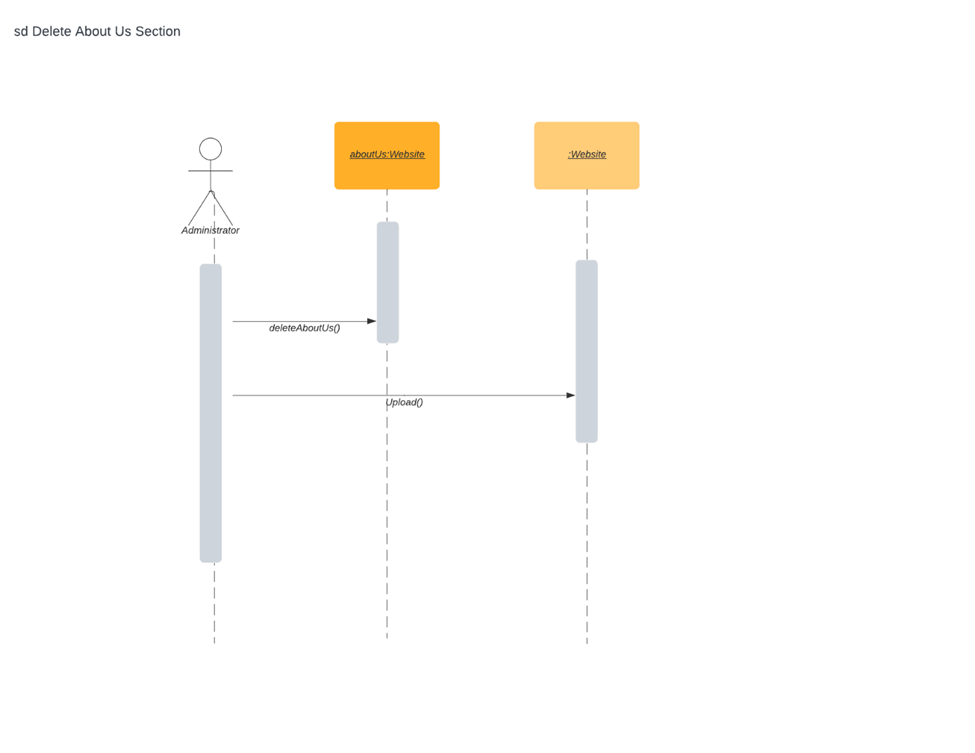
1. The use case starts when an administrator creates an About Us section on the website.

2. The administrator can edit the About Us section.

3. The administrator uploads the changes made to the About Us section.

****

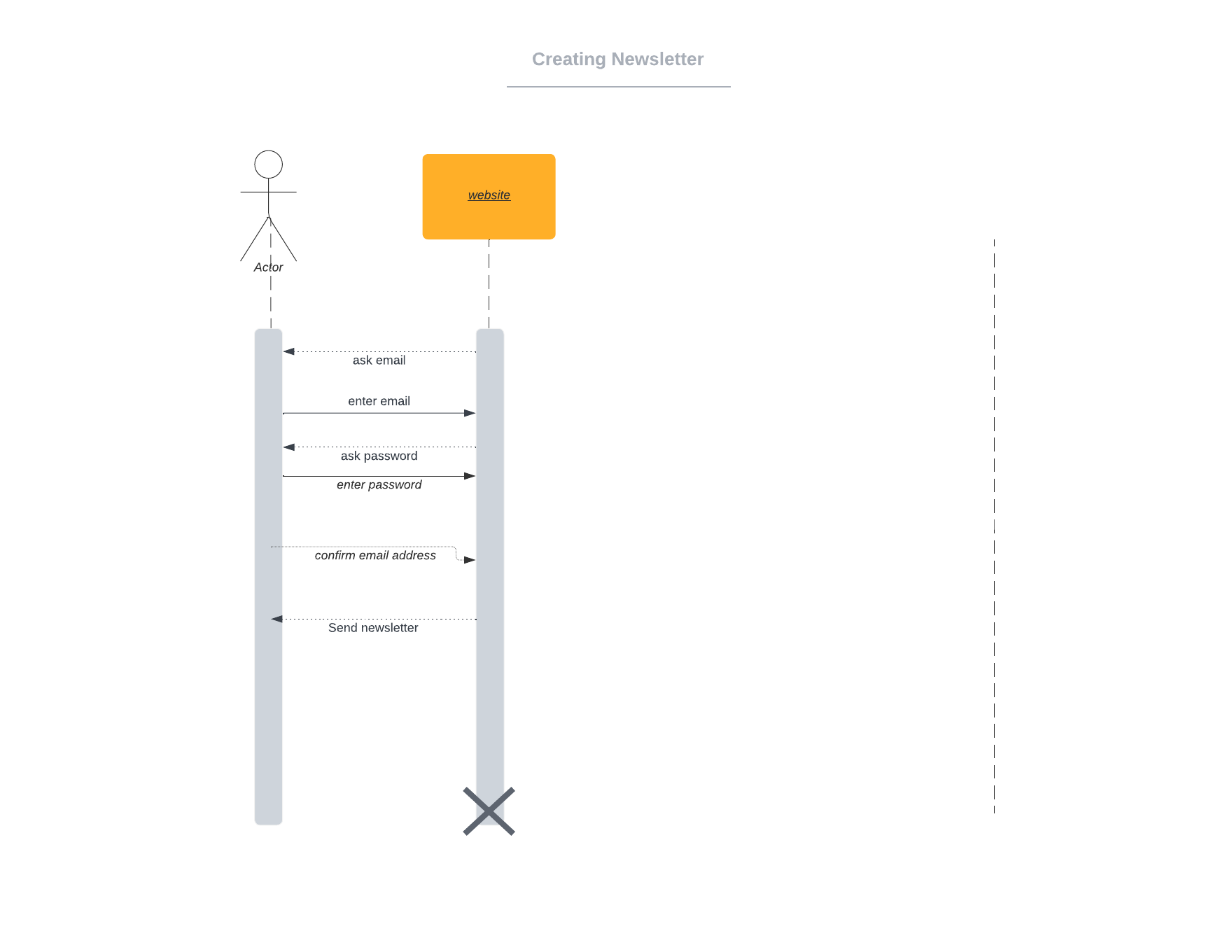
| Use Case: Delete “About Us” Section |
| --- |
| ID: 18 |
| Brief Description: This use case describes how the content in the About Us section on the website can be deleted. |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: Administrator must create an About Us section. |
| Main Flow:  1. The use case starts when Administrator creates an About Us section on the website.  2. The Administrator can then delete contents of the About Us section. |
| Postconditions: Only Administrators can delete the About Us section on website. |
| Alternative flows: none |

****

1. The use case starts when Administrator creates an About Us section on the website.

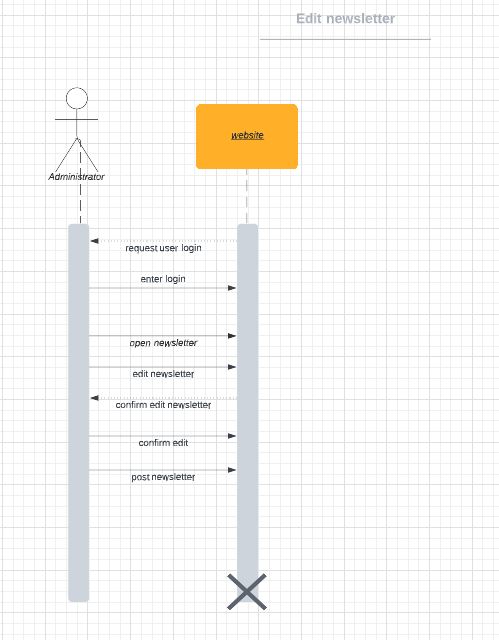
2. The Administrator can then delete contents of the About Us section.

| Use Case: Create Newsletter |
| --- |
| ID: 19 |
| Brief Description: Website users can sign up for a newsletter |
|  |
| Primary Actors: User |
| Secondary Actors: Doner |
| Preconditions: The user must have created a login and profile first |
| Main Flow:   1. Users signs up for newsletter with email 2. User receives a weekly/monthly newsletter in their inbox |
| Postconditions: Only users who signed up for a newsletter will receive emailed newsletter |

****

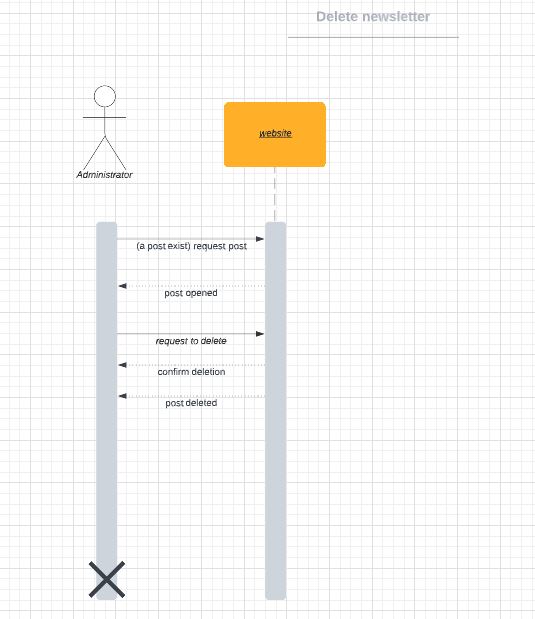
1. User signs up for newsletter
2. Users receives a weekly/monthly newsletter

| Use Case: Edit Newsletter |
| --- |
| ID: 20 |
| Brief Description: Newsletter can be edited |
| Primary Actors: Administrators |
| Secondary Actors: None |
| Preconditions: Administrators must have privileges access to the newsletter. |
| Main Flow:   1. Newsletter is posted to the website. 2. Newsletter has errors/date issues and is edited. 3. Newsletter is reposted with corrected information. |
| Postconditions: None |

****

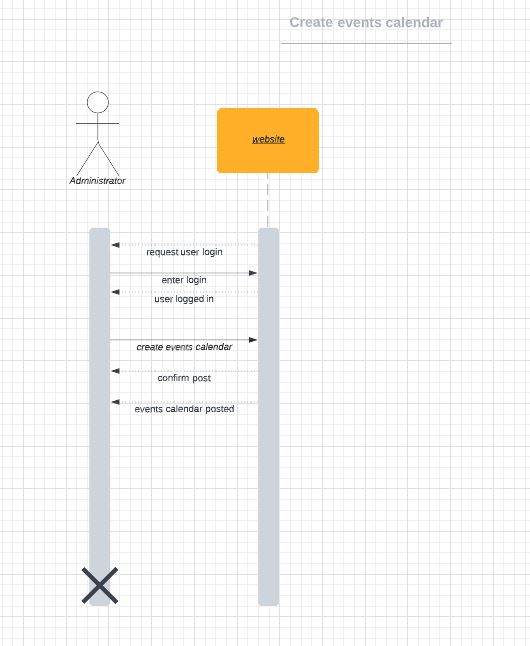
1. Newsletter is posted to the website
2. Newsletter has errors/date issues and is edited
3. Newsletter is republished with correction

| Use Case: Delete Newsletter |
| --- |
| ID: 21 |
| Brief Description: Old/obsolete Newsletters can be deleted. |
| Primary Actors: Administrator |
| Secondary Actors: None |
| Preconditions: Newsletter is already posted to the website. |
| Main Flow:   1. Newsletter is posted to the website. 2. Newsletter is old/outdated and needs to be deleted. 3. Newsletter is deleted. |
| Postconditions: None |

****

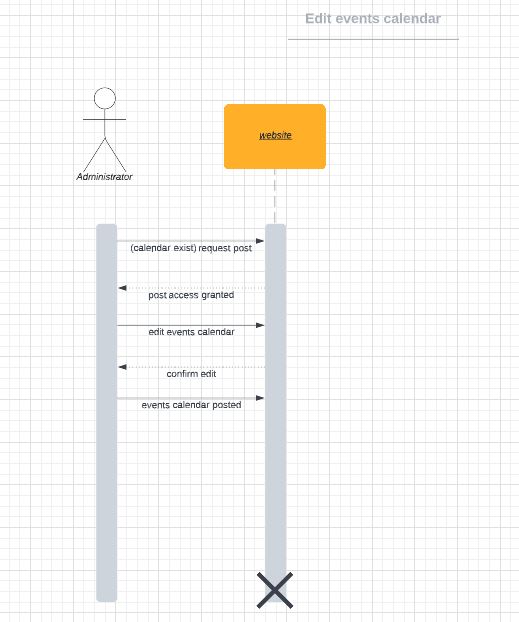
1. Newsletter is posted to website
2. Newsletter is old/outdated and needed deleted
3. Newsletter is deleted

| Use Case: Create events calendar |
| --- |
| ID: 22 |
| Brief Description: An Events Calendar is created to display volunteer opportunities. |
| Primary Actors: Administrator |
| Secondary Actors: |
| Preconditions: Administrator must have access to website to create calendar |
| Main Flow:   1. Calendar is formatted 2. Dates and times of events are input along as RSVP 3. Calendar is posted for users to view |
| Postconditions: None |

****

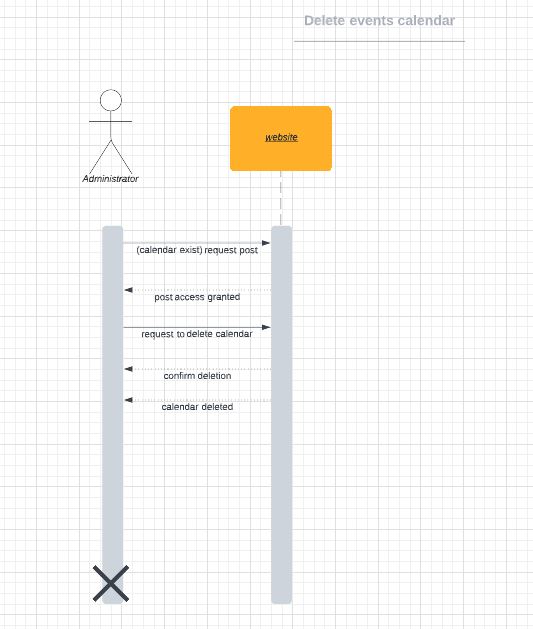
1. Calendar is formatted
2. Dates and times of events are input
3. Calendar is posted for users to view

| Use Case: Edit events calendar |
| --- |
| ID: 23 |
| Brief Description: Events Calendar can be edited/updated. |
| Primary Actors: Administrators |
| Secondary Actors: None |
| Preconditions: Administrator has privileges to access events calendar. |
| Main Flow:   1. Events calendar is posted. 2. Events calendar has date/time issues, needs to be updated. 3. Events calendar is fixed and reposted with corrections. |
| Postconditions: None |

****

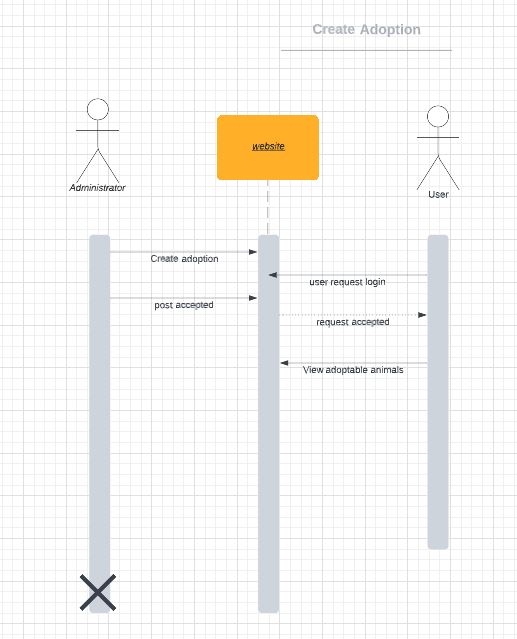
1. Events calendar is posted
2. Events calendar has date/time issues and needs to be updated
3. Events calendar is fixed and reposted

| Use Case: Delete events calendar |
| --- |
| ID: 24 |
| Brief Description: Past/obsolete Events Calendar can be deleted. |
| Primary Actors: Administrators |
| Secondary Actors: None |
| Preconditions: Calendar needs to be posted and in need of changes |
| Main Flow:   1. Calendar is posted and outdated 2. Calendar events that are old/outdated are deleted 3. Calendar is reposted with updated events and times. |
| Postconditions: None |

****

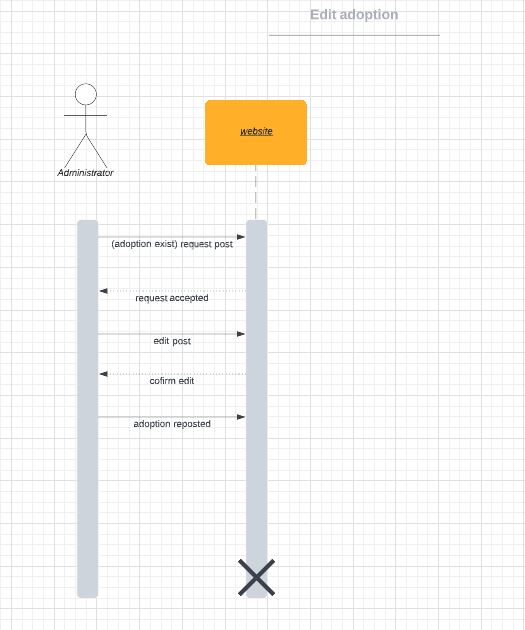
1. Calendar is posted and outdated
2. Old calendar events deleted

| Use Case: Create adoption |
| --- |
| ID: 25 |
| Brief Description: Users can view adoptable animals. |
| Primary Actors: Users |
| Secondary Actors: None |
| Preconditions: User must be logged into created account |
| Main Flow:   1. Animals are posted for users to view 2. Users view all animals available for adoption. |
| Postconditions: None |

****

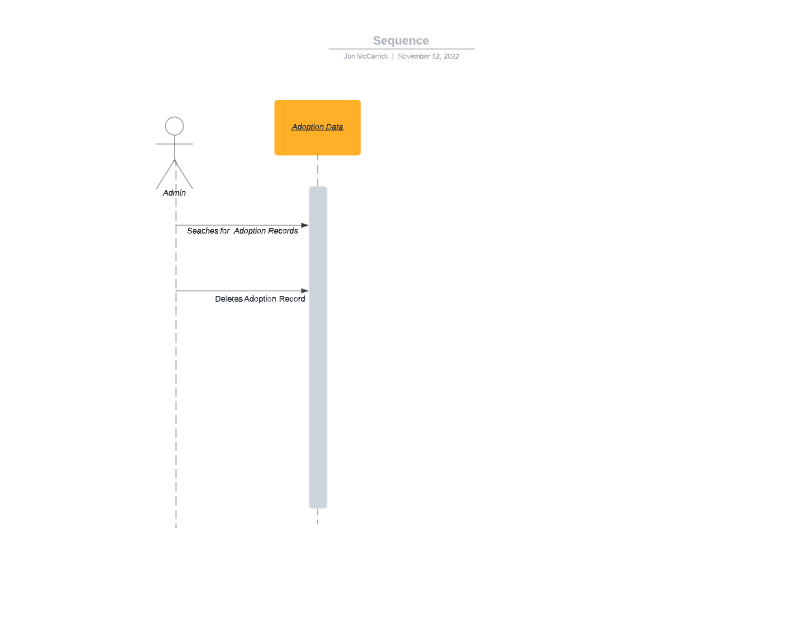
1. Animals are posted for users to view
2. Users view all animals available for adoption

| Use Case: Edit adoption |
| --- |
| ID: 26 |
| Brief Description: Adoptable animals section can be edited/updated. |
| Primary Actors: Administrators |
| Secondary Actors: None |
| Preconditions: Administrators has privileges to access and adoption has already been posted |
| Main Flow:   1. Animal adoption has been posted. 2. New animal information has been available, more pictures, updated shot charts. 3. Animal information changed and updated to the website. |
| Postconditions: None |

****

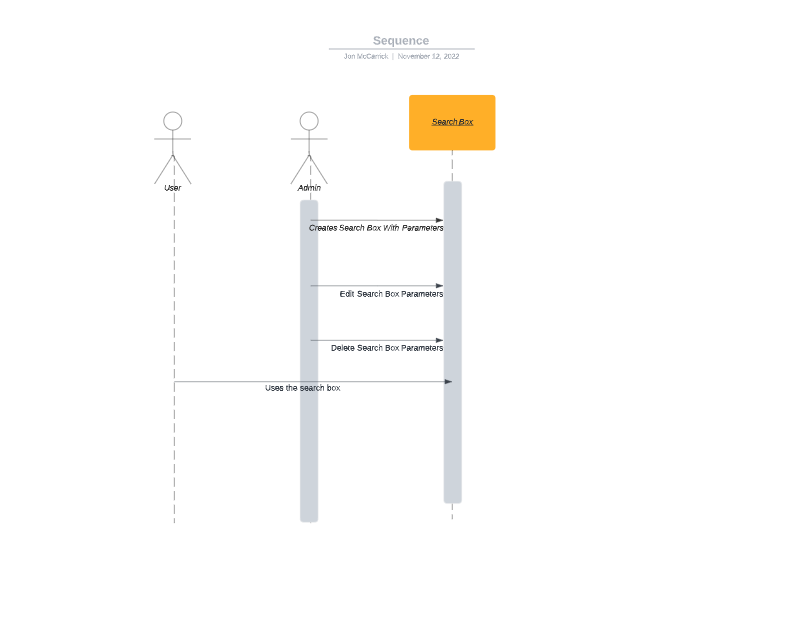
1. Animal adoption has been posted
2. New animal information becomes available, new pics, updated vaccine records
3. Animal info is changed and updated and reposted.

| Use Case: Delete Adoption |
| --- |
| ID: 27 |
| Brief Description: Adoption section can be deleted. |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: The user has to be an administrative user |
| Main Flow:  1. The use case starts when the admin deletes the adoption page or form. |
| Postconditions: none |
| Alternative flows: none |

****

1. Search for adoption
2. Deletion of adoption records

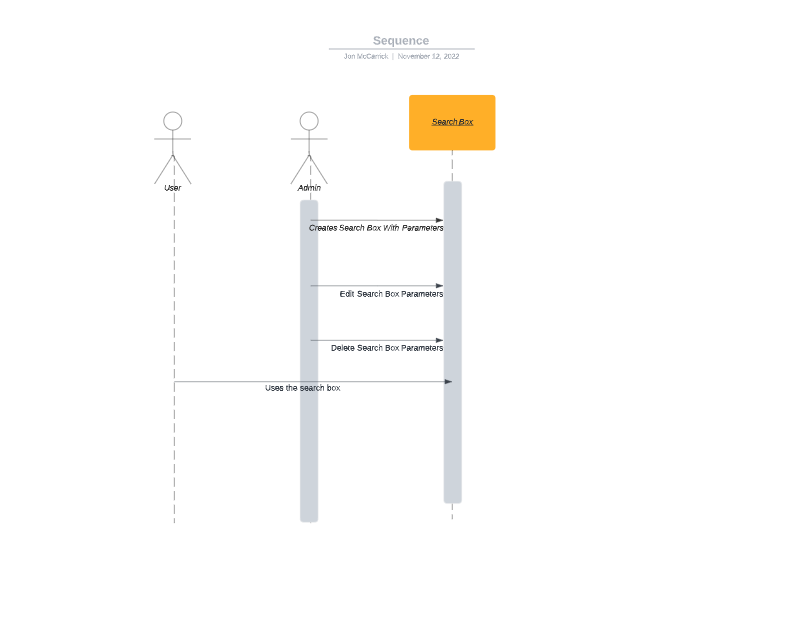
| Use Case: Create Pet Search box |
| --- |
| ID: 28 |
| Brief Description: A user can search for adoptable animals. |
| Primary Actors: User |
| Secondary Actors: none |
| Preconditions: none |
| Main Flow:  1. The use case starts when a user clicks the search box  2. The user types in the name of the dog, or the characteristic of the pet |
| Postconditions: none |
| Alternative flows: none |



1. The use case starts when a user clicks the search box

2. The user types in the name of the dog, or the characteristic of the pet

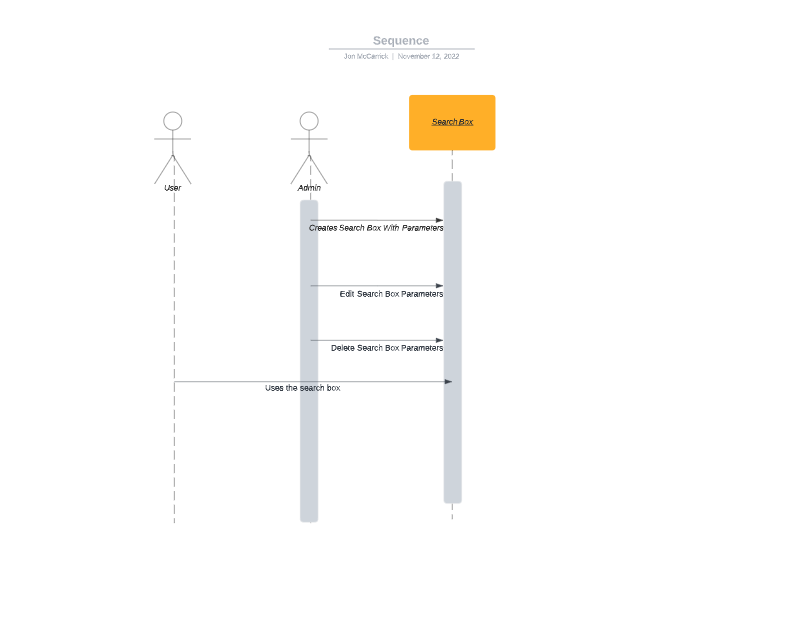
| Use Case: Edit Pet Search |
| --- |
| ID: 29 |
| Brief Description: Pet criteria can be edited/updated when we get new pets with different descriptions |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: The user has to be an administrative user |
| Main Flow:  1. The use case starts when the rescue gets a new pet with different descriptors  2. The admin changes/ updates search model |
| Postconditions: none |
| Alternative flows: none |



1. The use case starts when the rescue gets a new pet with different descriptors

2. The admin changes/ updates search model

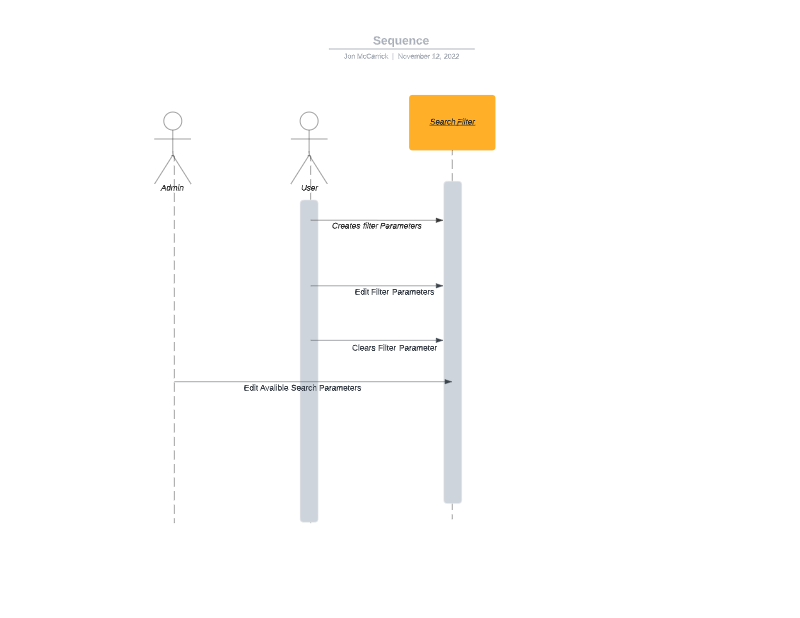
| Use case: Delete Pet Search |
| --- |
| ID: 30 |
| Brief Description: Pet can be deleted from the database for the pet search |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: The user has to be an administrative user |
| Main Flow:  1. The use case starts after a pet has been adopted or is no longer with the rescue.  2. The admin deletes pet from database and from the ability to search for. |
| Postconditions: none |
| Alternative flows: |



1. The use case starts after a pet has been adopted or is no longer with the rescue.

2. The admin deletes pet from database and from the ability to search for.

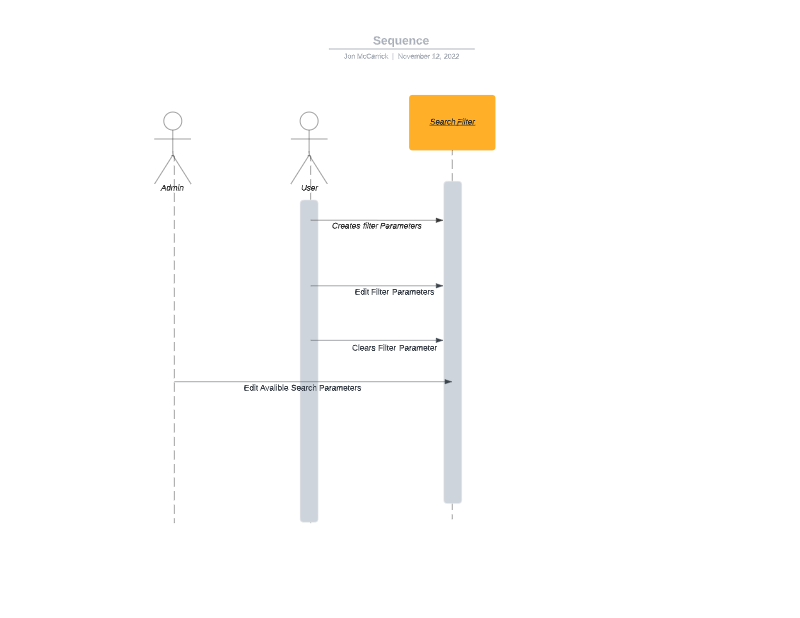
| Use case: Create Search Filter |
| --- |
| ID: 31 |
| Brief Description: A user can implement a search filter for specific types/breeds of animals. |
| Primary Actors: user |
| Secondary Actors: none |
| Preconditions: None |
| Main Flow:  1. The use case starts when the user is looking for a pet.  2. The user uses the search/filter box to find pets. |
| Postconditions: Only user that started the search can make edits |
| Alternative flows: none |



1. The use case starts when the user is looking for a pet.

2. The user uses the search/filter box to find pets.

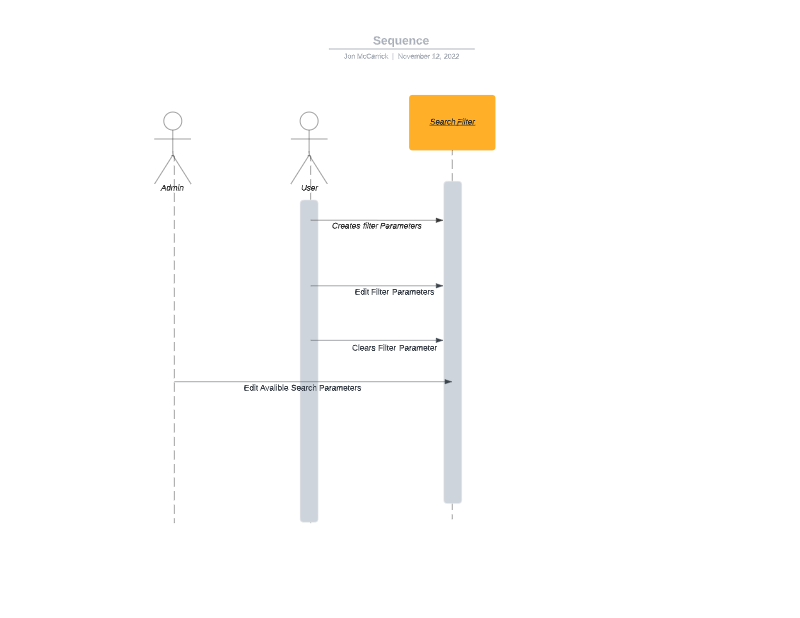
| Use case: Edit Search Filter |
| --- |
| ID: 32 |
| Brief Description: Search filter can be edited if user doesn’t find what they are looking for. |
| Primary Actors: User |
| Secondary Actors: none |
| Preconditions: none. |
| Main Flow:  1. The use case starts when a user needs to edit search.  2. The users clears or changes search parameters. |
| Postconditions: none |
| Alternative flows: none |



1. The use case starts when a user needs to edit search.

2. The users clears or changes search parameters

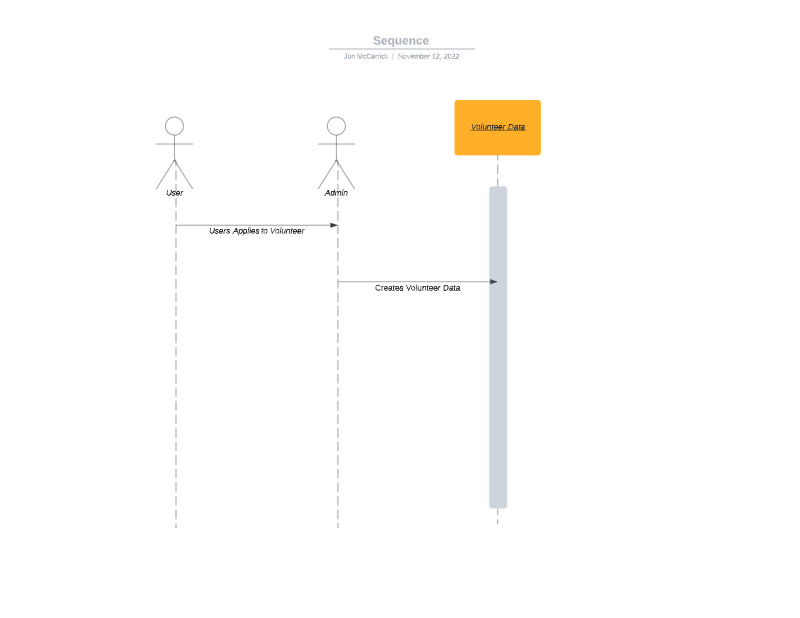
| Use case: Delete Search Filter |
| --- |
| ID: 33 |
| Brief Description: Search filter can be deleted or cleared if user doesn’t find what they are looking for. |
| Primary Actors: user |
| Secondary Actors: none |
| Preconditions: none |
| Main Flow:  1. The use case starts after a user needs to delete or clear the search filter  2. The users clear the search parameters. |
| Postconditions: none |
| Alternative flows: none |



1. The use case starts after a user needs to delete or clear the search filter

2. The users clear the search parameters.

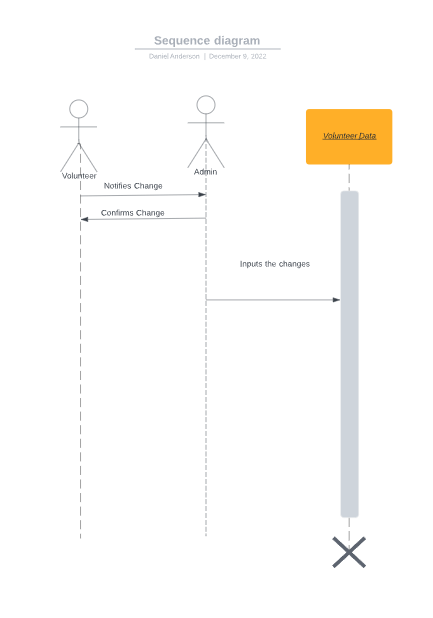
| Use case: Create Volunteer data |
| --- |
| ID: 34 |
| Brief Description: Volunteers can sign up on the website. |
| Primary Actors: Volunteers |
| Secondary Actors: none |
| Preconditions: none |
| Main Flow:  1. The use case starts after when user decides to apply to be volunteer  2. User fills out volunteer form and submits to admin |
| Postconditions: none |
| Alternative flows: none |

****

1. The use case starts after when user decides to apply to be volunteer

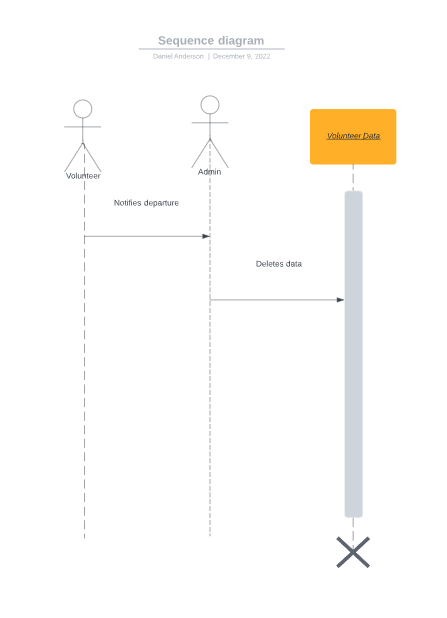
2. User fills out volunteer form and submits to admin

| Use Case: Edit Volunteer Data |
| --- |
| ID: 35 |
| Brief Description: Volunteer data can be edited if needed. |
| Primary Actors: Administrator |
| Secondary Actors: Volunteer |
| Preconditions: none |
| Main Flow  1.Volunteer has a change of information  2. The Administrator sees a change in data for a volunteer  3. Administrator changes the needed information  4. Applies it to the database |
| Postconditions: none |
| Alternative flows: none |



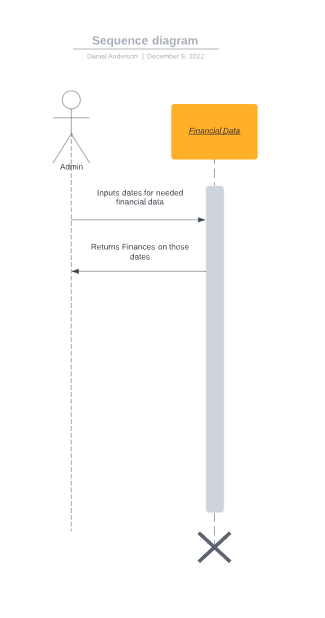
1. Volunteer notifies admin of a change to information.
2. Admin Confirms the change.
3. Admin inputs the changes to the information to the volunteer data.

| Use Case: Delete Volunteer data |
| --- |
| ID: 36 |
| Brief Description: Volunteer data can be deleted if no longer active. |
| Primary Actors: Administrator |
| Secondary Actors: Volunteer |
| Preconditions: none |
| Main Flow:  1. Admin selects data regarding a volunteer that no longer works there.  2. Admin deletes data regarding a volunteer that no longer works there.  3. Applies changes to the database |
| Postconditions: none |
| Alternative flows: none |



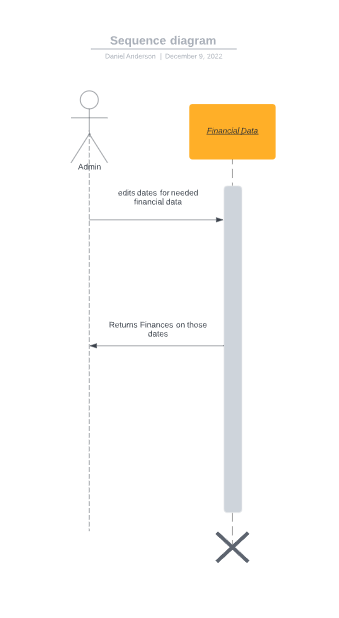
1. Volunteer inputs resignation/ notifies departure
2. Admin deletes data regarding volunteer..

| Use Case: Create Financial Report Data |
| --- |
| ID: 37 |
| Brief Description: Financial Reports generated for Blue Ridge |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: none |
| Main Flow:  1. Admin selects data regarding a period of financial history.  2. Admin uses the database to compile it into a report. |
| Postconditions: none |
| Alternative flows: none |



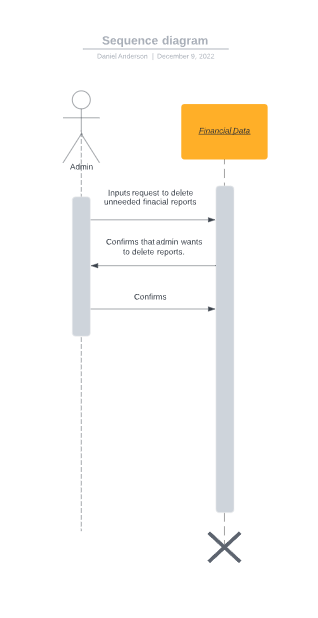
1. Admin selects dates of finances that they want.
2. System returns all financial data during those dates

| Use Case: Edit Financial Report Data |
| --- |
| ID: 38 |
| Brief Description: Financial Reports Edited as needed |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: none |
| Main Flow:  1. Admin selects the report and realizes its over the incorrect dates.  2. Admin selects the correct dates.  3. Admin uses the database to compile it into a report. |
| Postconditions: none |
| Alternative flows: none |

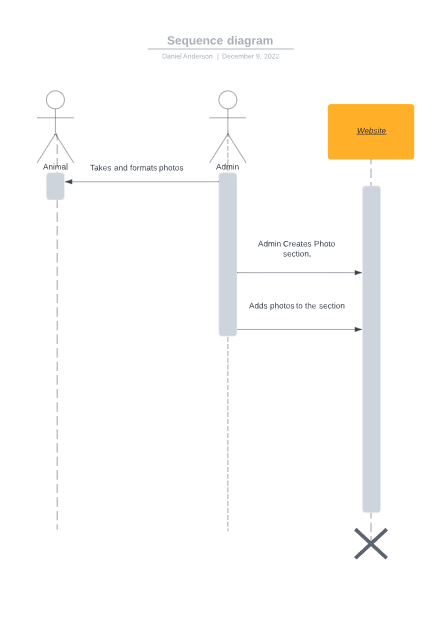


1. Admin Edits dates of needed information.
2. System returns the data on the altered dates.

| Use Case: Delete Financial Report data |
| --- |
| ID: 39 |
| Brief Description: Financial Reports can be deleted from Blue Ridge Website. |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: none |
| Main Flow:  1. Admin selects the report.  2. Admin deletes the report.  3. Database contains information. |
| Postconditions: none |
| Alternative flows: none |

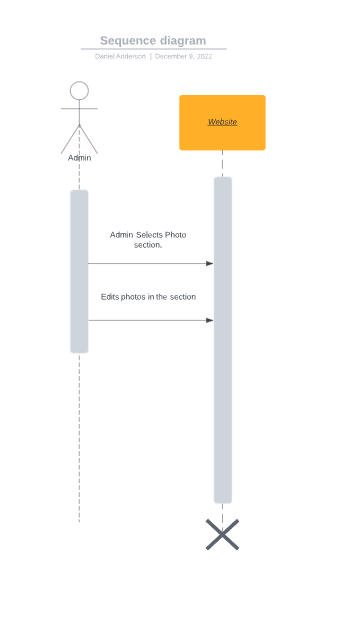
1. Admin inputs request to delete unneeded reports.
2. System confirms the admin wants to delete reports
3. Admin confirms.

| Use Case: Create Photo Section |
| --- |
| ID: 40 |
| Brief Description: A photo Section is created for Blue Ridge animals on the website |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: none |
| Main Flow:  1. Admin takes photos of animals and formats it.  2. Admin selects formatted photos and links it to a profile of the animal.  3. Admin applies the changes to the website. |
| Postconditions: none |
| Alternative flows: none |



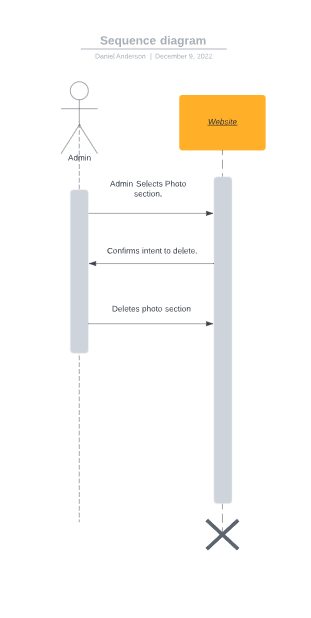
1. Admin takes and formats photos.
2. Admin Creates photo section on website.
3. Admin adds formatted photos to the section.

| Use Case: Edit Photo Section |
| --- |
| ID: 41 |
| Brief Description: Photos can be edited in the photo section |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: none |
| Main Flow:  1. Admin takes updated photos of the animal and formats them.  2. Admin selects formatted photos and links it to a profile of the animal.  3. Admin applies the changes to the website. |
| Postconditions: none |
| Alternative flows: none |



1. Admin selects photo section
2. Admin makes edits to the photos (either adds new ones or deletes non-applicable ones)

| Use Case: Delete Photo Section |
| --- |
| ID: 42 |
| Brief Description: Photos can be Deleted from the website |
| Primary Actors: Administrator |
| Secondary Actors: none |
| Preconditions: none |
| Main Flow:  1. Admin selects the dated photos.  2. Admin deletes dated photos from the particular animal's profile.  3. Admin applies the changes to the website. |
| Postconditions: none |
| Alternative flows: none |

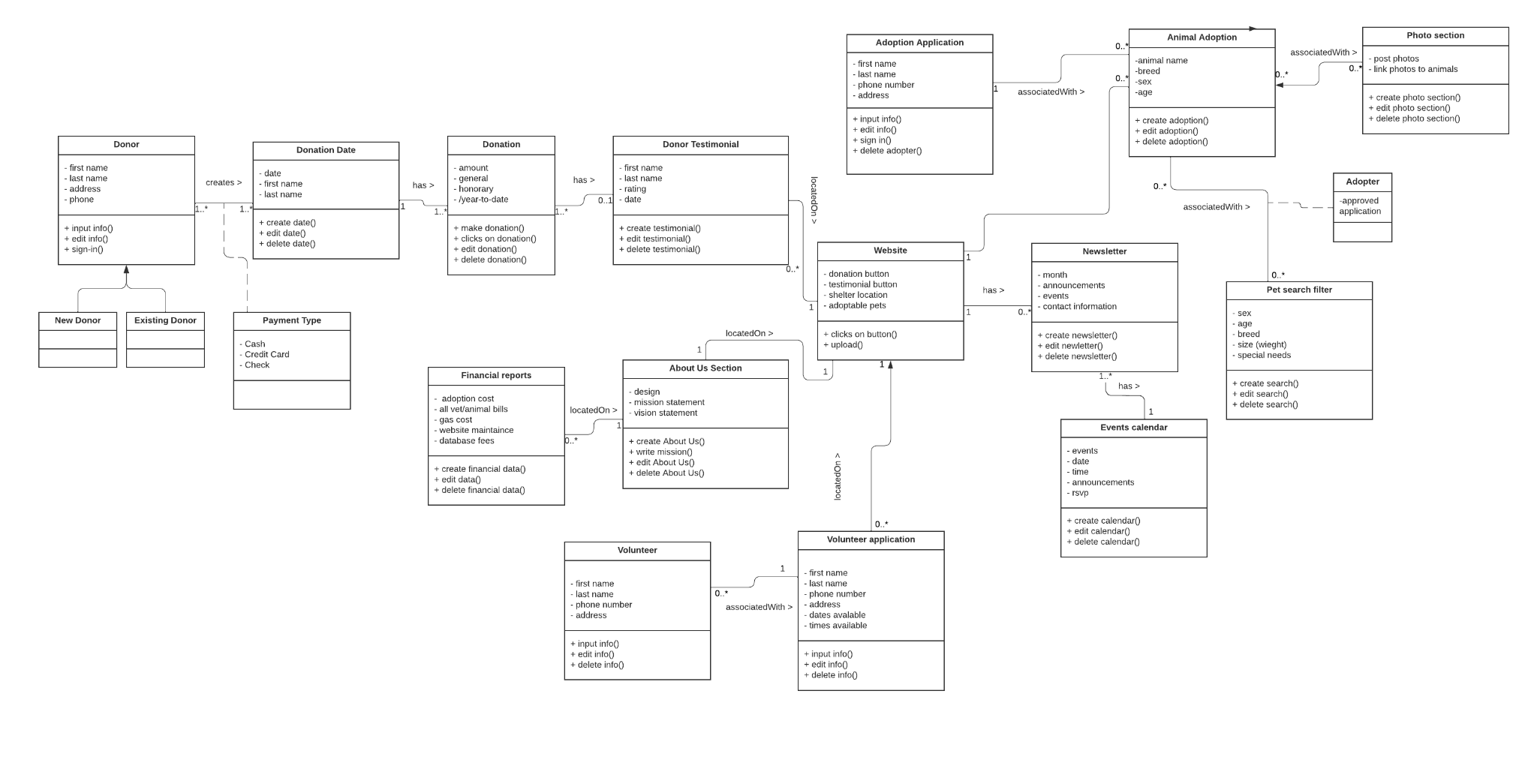


1. Admin selects photo section that they want to delete.
2. Website confirms the intent to delete a section.
3. Admin confirms and deletes the photo section.

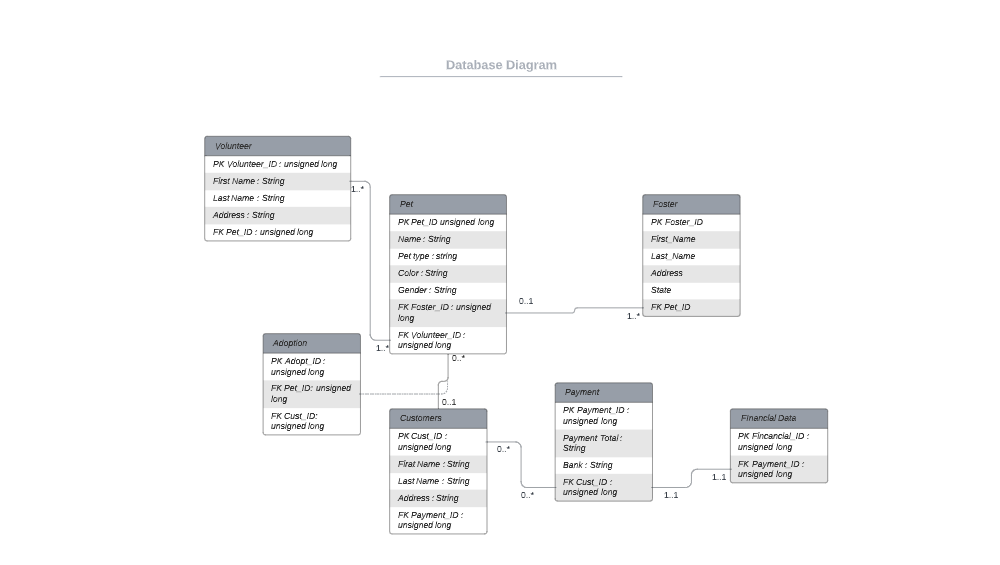
# 

# Class Diagram

A donor, either new or existing, makes a donation using their preferred payment type. This donation creates a donation date, which the donor can then create a testimonial/review describing their experience with Blue Ridge on the website. The website contains an About Us section, which explains the mission and vision of Blue Ridge. There is a Newsletter located on the website describing announcements. The user can also view an Events Calendar showing upcoming events hosted/affiliated with Blue Ridge. They can view adoptable pets on the website using the Photo section and, using a pet search filter, they can narrow to specific criteria such as breed, age, sex, weight, etc. The user can also fill out and submit an adoption application, as well as a volunteer application. Finally, there is a Financial Report section that contains financials such as adoption cost and veterinarian bills.



# Database Design



# 

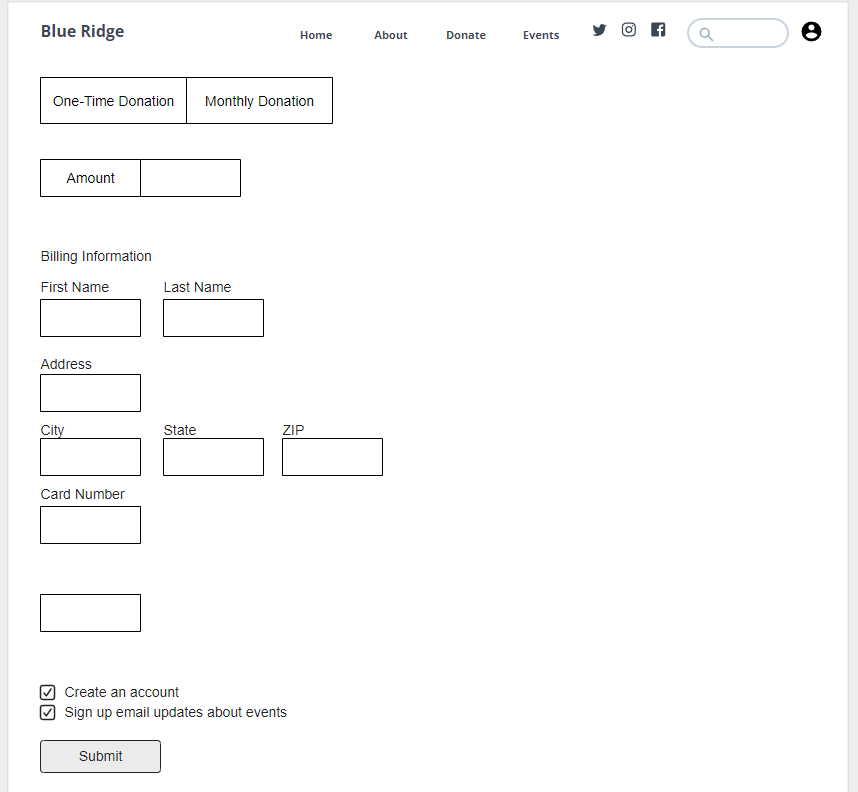
# User Interface Navigation Diagram

The user interface navigation diagram defines all the basic structures of the diagram. The diagram displays all the interfaces throughout the diagram and how they are connect to each other.

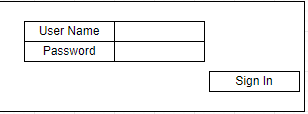


Screen Layouts

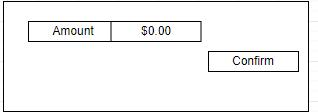
Create/Edit Account - This enables users to create or edit their account information when changes are needed.



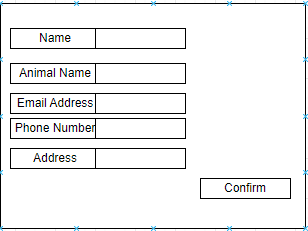
Log in - This form enables users to skip repeating their information when they want to donate by signing in.



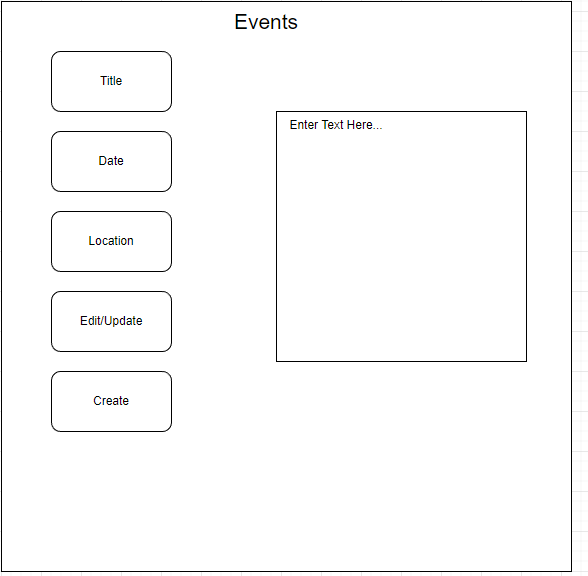
Donate - This form enables users to enter their donation amount information into the website.



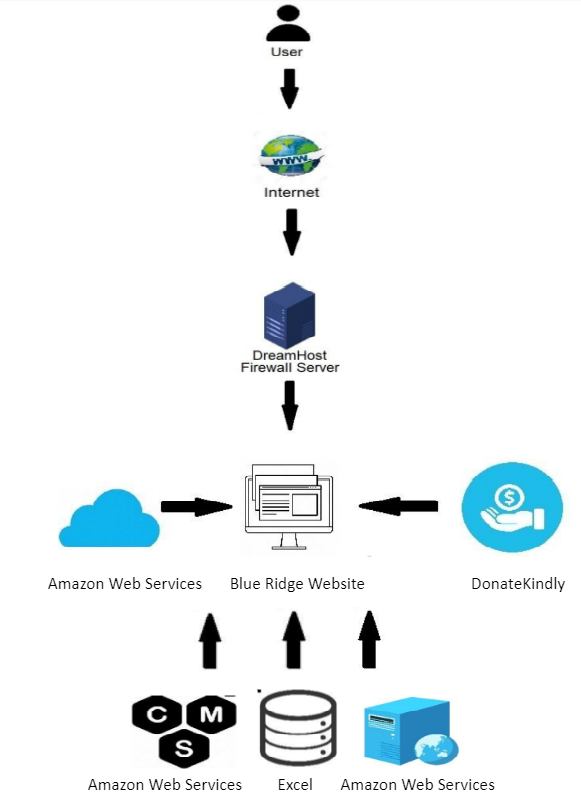
Adoption - This form enables user who dont have an account to apply for animals they are interested in.



Events - This form enables the admin to create and update events they plan on having.



# Physical Architecture Design

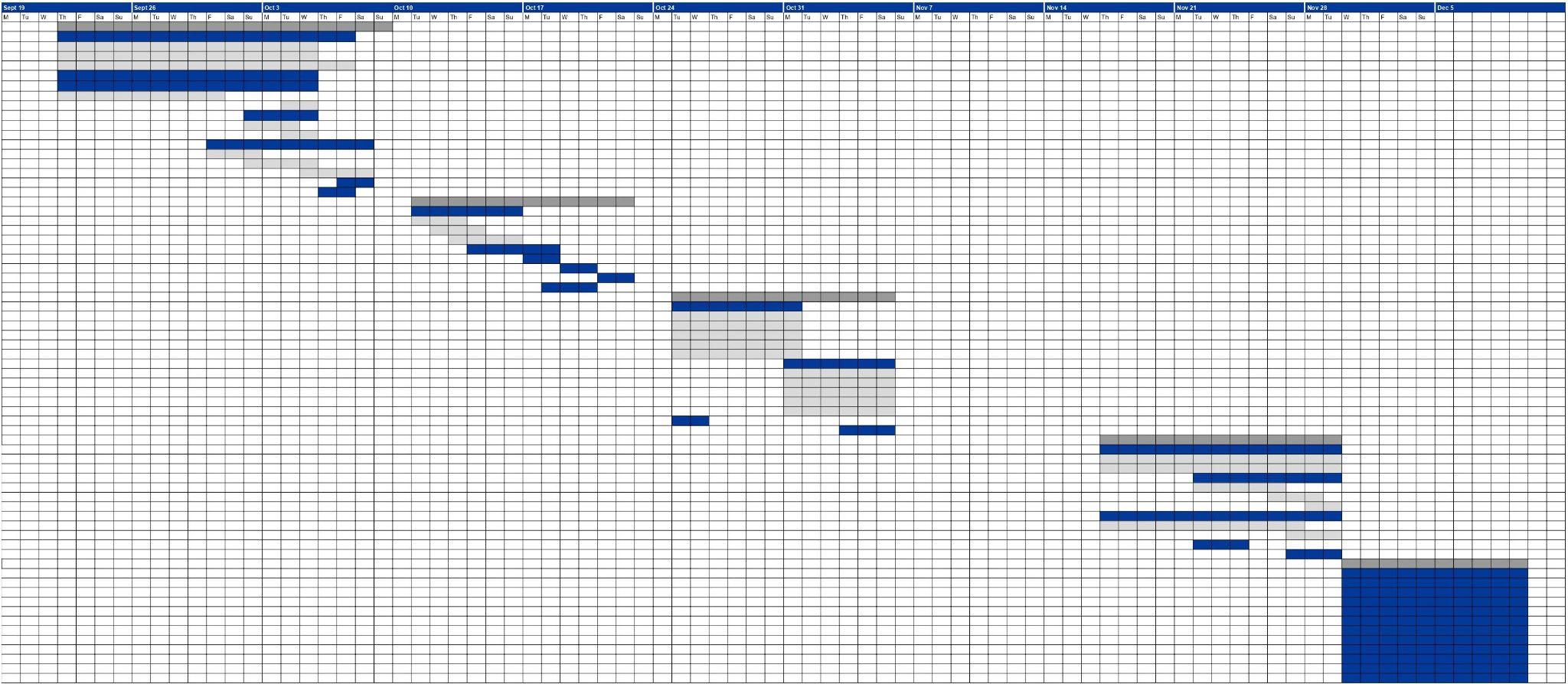


# Design Procedures

The design for the website was built to be functional and secure. We decided to use Amazon web services for a few reasons. Some of the Non-functional requirements that helped us make that decision were the Scalability, performance, security, maintainability, and availability. These are things that amazon web services is able to do better than other competitors. With AWS you can access you site anywhere, and even access it on your smartphone. Amazon makes it easy to maintain and secure. Its utilzed up in the cloud so you do not need any special software to use it. The drawbacks in our opinion do not out way the security and scalability of AWS. But it doesn’t have an easy to use drag and drop template designer. You have to upload your premade HTML pages to the server. Which makes it not as easy to use, but not impossible. From this we determined for the growth of the business and the security that is provided by AWS that it was a good choice.

# Gantt Chart





# 

# Prototype

<http://blueridgeanimalrescue.s3-website.us-east-2.amazonaws.com/Home.html>

