**Elaboration Specification**

**Paw Patrol**

**Madison Heil, Lizzie Perkins, Adam Harlow, Patrick Nguyen, Dylan Dombrowski, Binh Nguyen**

**Table of Contents**

**System Requirements**

**Functional Requirements**

About Us

ID: SR001 -The About Us page shall allow for information to be added

ID: SR002 - The About Us page shall allow for information to be edited

ID: SR003 - The About Us page shall allow for information to be deleted

ID: SR004 - The About Us page shall present information about the organization

Adopt

ID: SR005 - The adopt page shall allow for information to be added

ID: SR006 - The adopt page shall allow for information to be edited

ID: SR007 - The adopt page shall allow for information to be deleted

ID: SR008 - The adopt page shall accept applications from users

ID: SR009 - The adopt page shall allow deletion of applications

ID: SR010 - The adopt page shall allow for users to filter through animals

ID: SR011 - The adopt page shall allow for users to view various breeds of available animals

ID: SR012 - The adopt page shall allow users to view descriptions of animals

ID: SR013 - The adopt page shall allow for multiple photos of animals

ID: SR014 - The adopt page shall allow users to click on the photos of the animals to see details

Donate

ID: SR015 - The donation system shall accept payments from donors

ID: SR016 - The donation system shall deny payments from donors

ID: SR017 - The donation system shall allow donors to add information

ID: SR018 - The donation system shall allow donors to edit information

ID: SR019 - The donation system shall allow donors to delete information

ID: SR020 - The donation system shall allow donors to edit donations

ID: SR021 - The donation system shall allow deletion of donations

ID: SR022 - The donation system shall allow anonymous donations

ID: SR023 - The donation system shall allow one-time donations

ID: SR024 - The donation system shall allow recurring donations

Help Us!

ID: SR025 - The Help Us! Page will allow the organization to add information

ID: SR026- The Help Us! Page will allow the organization to delete information

ID: SR027 - The help us page will allow users to create applications

ID: SR028 - The help us page will allow the organization to delete applications.

ID: SR029 - The help us page will allow users to submit applications

ID: SR030 - The help us page will show areas of volunteering

ID: SR031 - The help us application will allow users to insert personal information

ID: SR032 - The help us application will allow users to edit personal information

ID: SR033 - The help us application will allow users to delete personal information

ID: SR034 - The help us application will allow the organization to select availability

ID: SR037 - The help us application will allow the organization to edit availability

ID: SR038 - The help us application will allow the organization to delete availability

ID: SR039 - The help us section will allow the organization to view submitted applications

ID: SR040 - The help us application will allow users to select areas of interest

ID: SR041 - The help us application will allow users to edit areas of interest

ID: SR042 - The help us application will allow users to delete areas of interest

Events

ID: SR043 - The events page will allow for information to be created

ID: SR044 - The events page will allow for information to be edited

ID: SR045 - The events page will allow for information to be deleted

ID: SR046 - The events page will allow for events to be created

ID: SR047 - The events page will allow for events to be edited

ID: SR048 - The events page will allow for events to be deleted

ID: SR049 - The events page will allow users to RSVP to events

ID: SR050 - The events page will allow users to unregister from events

ID: SR051 - The events page will display upcoming events

Contact Us

ID: SR052 - The contact us page will allow information to be created

ID: SR053 - The contact us page will allow information to be edited

ID: SR054 - The contact us page will allow for information to be deleted

ID: SR055 - The contact us page will allow users to insert personal information

ID: SR056 - The contact us page will allow users to edit personal information

ID: SR057 - The contact us page will allow users to delete personal information

ID: SR058 - The contact us page will allow for messages to be created

ID: SR059 - The contact us page will allow for messages to be edited

ID: SR060 - The contact us page will allow for messages to be deleted

ID: SR061 - The contact us page will allow users can change contact preferences

**Non–Functional Requirements**

Performance

ID: SR062 - The website will operate smoothly and load quickly

ID: SR063 - The website will be updated and constantly supported

ID: SR064 - The donation system will run without errors

ID: SR065 - The donation system will run without lost information

ID: SR066 - The website tabs will open without any issues

ID: SR067 - Adopt and volunteer applications will be sent to the organization

Security

ID: SR068 - The system will monitor security through WordPress

ID: SR069 - Website information will be backed-up

ID: SR070 - Donation information will be backed-up and secure

ID: SR071 – System will be SSL Certified

ID: SR072 – System will use firewalls through WordPress

Capacity

ID: SR073 - The donation system will have adequate storage for donor information

ID: SR074 - The volunteer/adopter application will have adequate storage for information

Availability

ID: SR075 - The system will be compatible with most web browsers

ID: SR076 - The website and donation system will allow interaction from many users

ID: SR077 - The website will require minimal clicks for donating

ID: SR078 - The website and donation system will have available support

**Use Case Diagram: About Us Tab**

|  |
| --- |
| **Use Case: Create About Us** |
| ID: 1 |
| Brief Description: Admin can create information in About Us that shows up for users. |
| Primary Actors: Organization/ Admin |
| Secondary Actors: none |
| Preconditions: Must login as an admin to have access to create any content/information |
| Main Flow:   1. The use case starts when the admin click create the information on About Us 2. 2. The admin type content in About Us page. |
| Postconditions: Only admin can create the information for About Us |
| Alternative flows: none |

|  |
| --- |
| **Use Case: Edit About Us** |
| ID: 2 |
| Brief Description: Admin can edit any information on the About Us page. |
| Primary Actors: Organization/ Admin |
| Secondary Actors: none |
| Preconditions: Must have admin access to edit and can only be edited if there’s an exist content |
| Main Flow:  The use case starts when admin chooses which one needs to be edit  Admin clicks edit after choosing which one needs to be changed and will replace the old one. |
| Postconditions: Only admin can edit any information on About Us |
| Alternative flows: none |

|  |
| --- |
| **Use case: Delete About Us** |
| ID: 3 |
| Brief Description: Admin can delete information on About Us |
| Primary Actors: Organization/ Admin |
| Secondary Actors: none |
| Preconditions: Information can only be deleted if they were created and posted on About Us |
| Main Flow:  The use case starts after the information/content is created and admin chooses to delete  The user edits the post and then posts it again, replacing the original post. |
| Postconditions: Only admin can delete content on About Us |
| Alternative flows: |

|  |
| --- |
| **Use case: View About Us** |
| ID: 4 |
| Brief Description: Everyone can view information that admin posted on About Us |
| Primary Actors: Organization/ Admin and users |
| Secondary Actors: none |
| Preconditions: Anyone that has Internet to access the website |
| Main Flow:  The use case starts after users click on the About Us tab.  Users and admin click on About Us tab to view all the information/content on this tab |
| Postconditions: None |
| Alternative flows: none |

|  |
| --- |
| **Use case: Create Links** |
| ID: 5 |
| Brief Description: Admin can create links so the users can click on to navigate to destination that we want |
| Primary Actors: Admin/Organization |
| Secondary Actors: none |
| Preconditions: Must login as an admin account to have access creating the links |
| Main Flow:  The use case starts when admin creates a link.  Admin click create link. |
| Postconditions: Only admin can create links |
| Alternative flows: none |

|  |
| --- |
| **Use case: Edit Links** |
| ID: 6 |
| Brief Description: Admin can edit links |
| Primary Actors: Admin/Organization |
| Secondary Actors: none |
| Preconditions: Admin can on edit links if there’s link already existed |
| Main Flow:  The use case starts after the admin clicks edit the link  Admin chooses to edit the unwanted link |
| Postconditions: Only admin can edit the links |
| Alternative flows: none |

|  |
| --- |
| **Use case: Delete Links** |
| ID: 7 |
| Brief Description: admin user can delete existed links |
| Primary Actors: Admin/Organization |
| Secondary Actors: none |
| Preconditions: Must login as the admin to have access to delete existed links |
| Main Flow:  The use case starts after admin clicks delete unwanted links  Admin deleted unwanted links |
| Postconditions: Only admin can delete links on About Us |
| Alternative flows: none |

|  |
| --- |
| **Use case: Access Links** |
| ID: 8 |
| Brief Description: Everyone can access to the links when in user view |
| Primary Actors: Users |
| Secondary Actors: none |
| Preconditions: The users need to have Internet to access to this website |
| Main Flow:  The use case starts after user click on the links.  Users click on the link that they interested in and be navigated to the destination they want |
| Postconditions: None |
| Alternative flows: none |

|  |
| --- |
| **Use case: Design About Us** |
| ID: 9 |
| Brief Description: Admin can change the design on About Us tab |
| Primary Actors: Admin/Organization |
| Secondary Actors: none |
| Preconditions: Must login as an admin to have access to design the About Us tab |
| Main Flow:  The use case starts after the admin click design About Us tab  Admin can do any design they want for About Us tab |
| Postconditions: Only admin can change the design on About Us tab |
| Alternative flows: none |

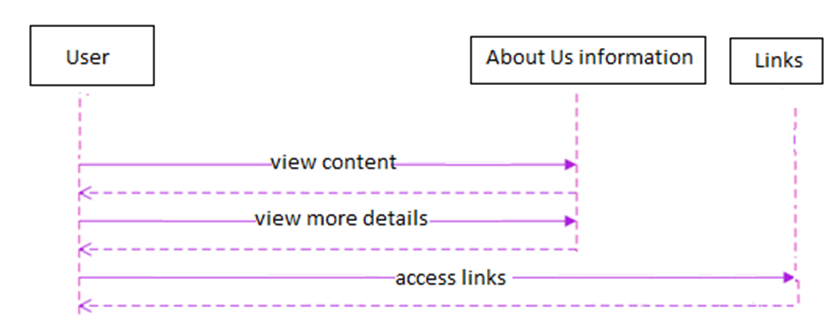
|  |
| --- |
| **Use case: View any details on About Us** |
| ID: 10 |
| Brief Description: Everyone that has Internet can access to the website can view details on About Us tab as user view |
| Primary Actors: User |
| Secondary Actors: none |
| Preconditions: Anyone that has Internet to access the website |
| Main Flow:  The use case starts after users click “*read more…*”  Users can view more details after clicking “*read more…*” |
| Postconditions: None |
| Alternative flows: none |

**Sequence Diagrams**

**Main Flow for users**

1. Users can view any content that admin posted
2. Users can access to links on About Us page to navigate to another page/tan
3. Users can click “read more” to view more details on About Us tab

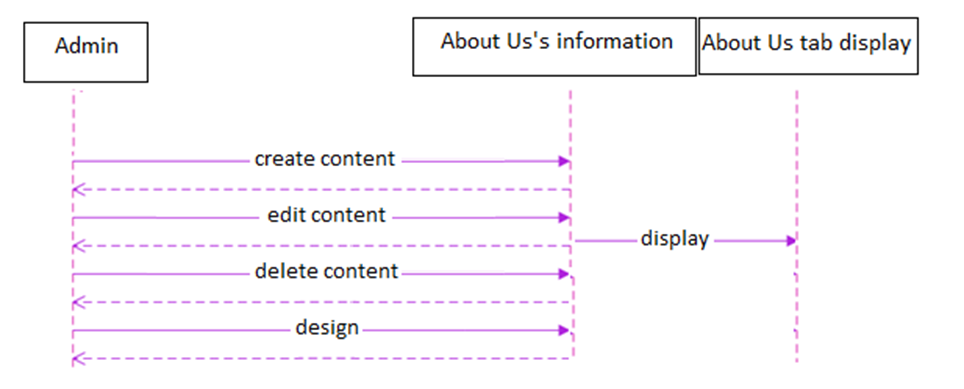




Main Flow admin/organization for About Us content/information

1. Admin creates About Us information/content
2. Admin edits the existed information/content and replaces with the new one
3. Admin deletes unwanted content
4. Admin designs



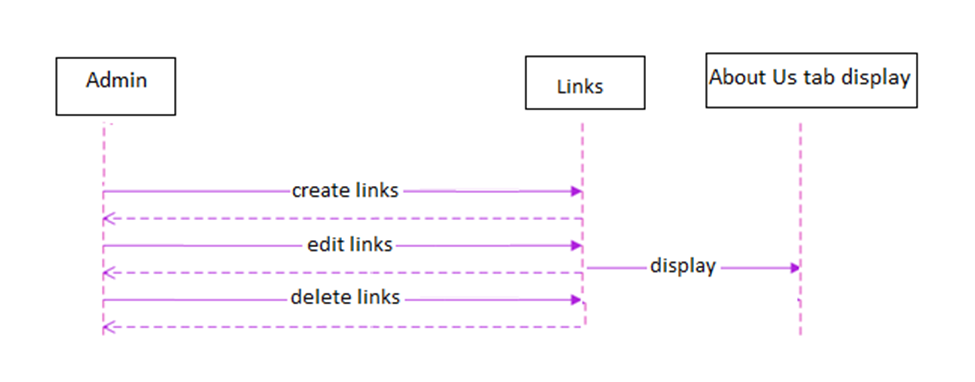


**Main flow admin/organization about links**

Admin creates links

Admin edits that existed

Admin deletes unwanted links

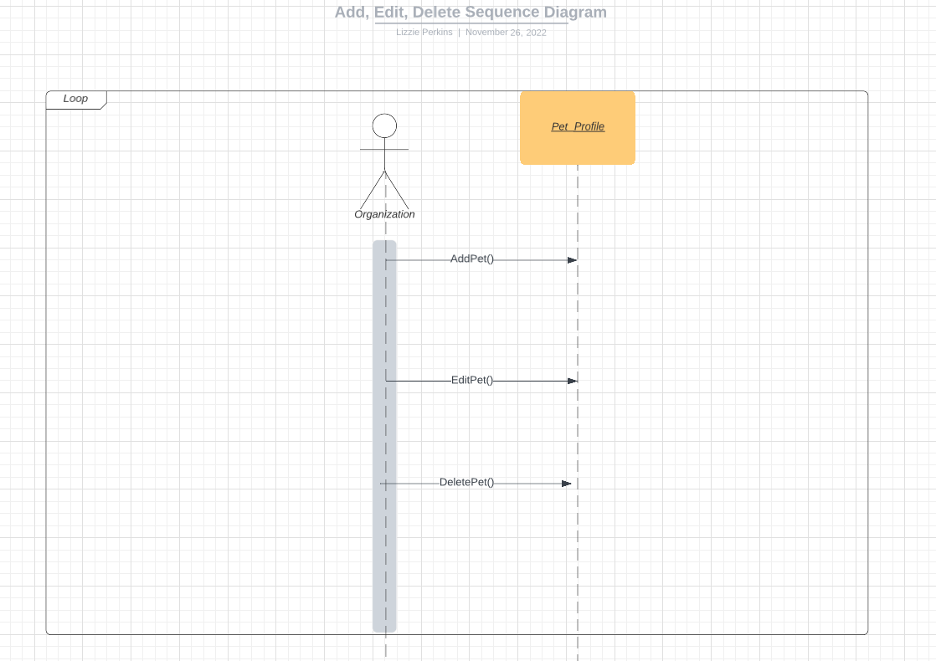


**Use Case: Adopt Tab**

|  |
| --- |
| Use Case: Add Pets |
| ID: 011 |
| Use Case type: detail, essential |
| Primary actor: Organization |
| Brief Description: The organization can add information about the animals including species, breed, age, and other valuable information. |
| Importance level: high |
| Stakeholders: Organization and Users |
| Trigger: Organization receives a new pet that is available for adoption |
| Relationships:  · Association: Edit & Delete Pets  · Include: Show Pet photos, Show Pet details, Show Pet description  · Extend:  · Generalization: Add, Edit, Delete Pets |
| Flow of events:  1. Blue Ridge receives a new pet  2. Blue Ridge fully vets the animal and ensures it is ready for adoption  3. Blue Ridge posts the Pet on the Adoption page |

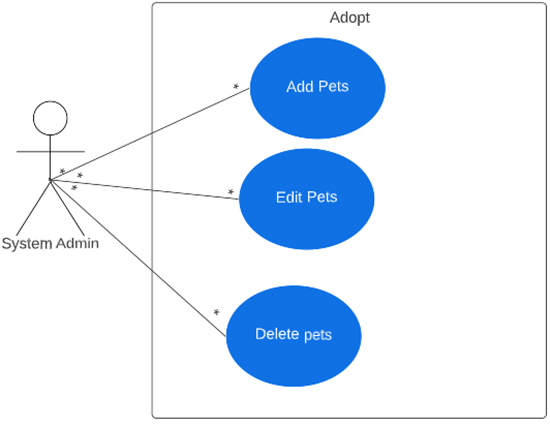
|  |
| --- |
| Use Case: Edit Pets |
| ID: 012 |
| Use Case type: detail, essential |
| Primary actor: Organization |
| Brief Description: The organization can edit animal information as needed about each individual animal. |
| Importance level: high |
| Stakeholders: Organization and Users |
| Trigger: Organization receives new information regarding current pets |
| Relationships:  · Association: Add & Delete Pets  · Include: Show Pet photos, Show Pet details, Show Pet description  · Extend:  · Generalization: Add, Edit, Delete Pets |
| Flow of events:  1. Blue Ridge adds pets to the Adopt tab  2. Information needs to be edited  3. Blue Ridge edits the information about the pet on the Adopt tab |

|  |
| --- |
| Use Case: Delete Pets |
| ID: 013 |
| Use Case type: detail, essential |
| Primary actor: Organization |
| Brief Description: Organization can delete animal information if the animal is adopted |
| Importance level: high |
| Stakeholders: Organization and user |
| Trigger: Animal is adopted and needs to be taken off of the Adopt tab |
| Relationships:  · Association: Add & Edit Pets  · Include: Show Pet photos, Show Pet details, Show Pet description  · Extend:  · Generalization: Add, Edit, Delete Pets |
| Flow of events:  1. Pet is posted on the Adopt tab  2. Pet is adopted by a user  3. Blue Ridge deletes pet from the Adopt tab |



**Use Case 11, 12 13 Main Flow:**

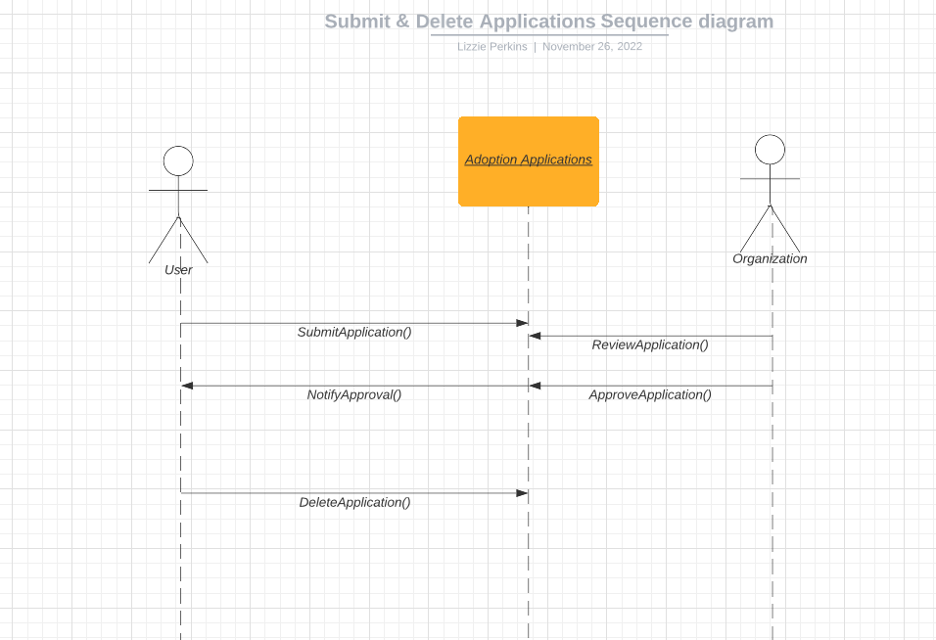
1. Organization obtains information about a new pet that is available for adoption and adds that information to the pet’s profile.
2. Organization needs to make changes to a pet’s profile.
3. Organization may need to delete a pet’s profile once it has been adopted.



Narrative: The organization can add, edit, and delete pets from the Adopt page.

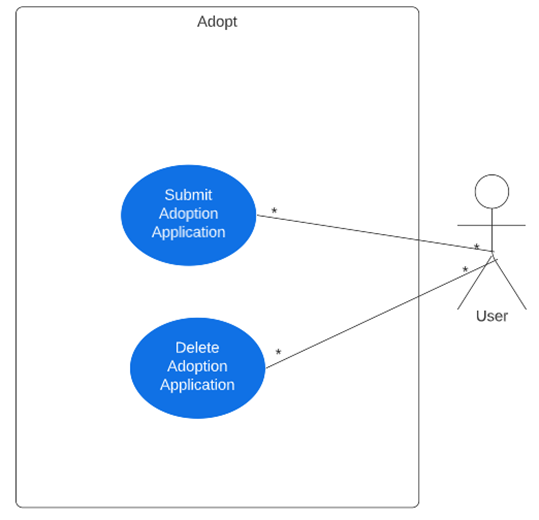
|  |
| --- |
| Use Case: Submit Adoption Application |
| ID: 014 |
| Use Case type: detail, essential |
| Primary actor: User |
| Brief Description: User can apply directly on the Adopt Tab for a specific animal. |
| Importance level: high |
| Stakeholders: User and organization |
| Trigger: User sees an available pet and wants to adopt it |
| Relationships:  · Association: Delete Adoption Application  · Include:  · Extend:  · Generalization: Adoption Application |
| Flow of events:  1. User finds a pet from the Adopt tab that they want to adopt  2. User fills out the application  3. User clicks the submit button |

|  |
| --- |
| Use Case: Delete Adoption Application |
| ID: 015 |
| Use Case type: detail, essential |
| Primary actor: Organization |
| Brief Description: Organization can delete applications submitted if deemed necessary. |
| Importance level: medium |
| Stakeholders: Organization and user |
| Trigger: Incomplete or inaccurate application information |
| Relationships:  · Association: Submit Adoption Application  · Include:  · Extend:  · Generalization: Adoption Application |
| Flow of events:  1. Organization gets notification of incomplete or incorrect application information  2. Organization contacts applicant with questions  3. Organization deletes application |



**Use Case 14 and 15 Main Flow:**

1. User fills out online application and submits.
2. Organization receives notification and reviews the application for adoption.
3. Organization either approves or denies the application.
4. Organization notifies the user of approval or denial.
5. Alternatively, user can delete their application.



Narrative: The user can submit and delete their application for adoption.

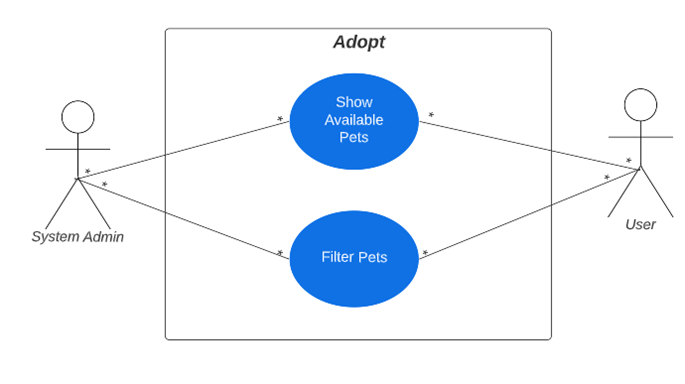
|  |
| --- |
| Use Case: Show Available Pets |
| ID: 016 |
| Use Case type: detail, essential |
| Primary actor: Organization |
| Brief Description: Users can only view available animals for adoption. |
| Importance level: high |
| Stakeholders: Organization and user |
| Trigger: User wants to see pets available for adoption |
| Relationships:  · Association: Filter Pets  · Include:  · Extend:  · Generalization: Filter available pets |
| Flow of events:  1. Organization adopts pets  2. Organization changes the availability of pets  3. User only sees available pets |

|  |
| --- |
| Use Case: Filter Pets |
| ID: 017 |
| Use Case type: detail, essential |
| Primary actor: User |
| Brief Description: User can filter through animals by species and breed. |
| Importance level: Medium |
| Stakeholders: User and organization |
| Trigger: User wants to see available pets by a specific descriptor |
| Relationships:  · Association: Show available pets  · Include:  · Extend:  · Generalization: Filter available pets |
| Flow of events:  1. User wants to filter available animals by a specific descriptor  2. User clicks desired filter in drop down menu |



**Use case 16 and 17 main flow:**

1. Organization can update a pet’s availability.
2. User can select filters to see what pets and breeds are available.

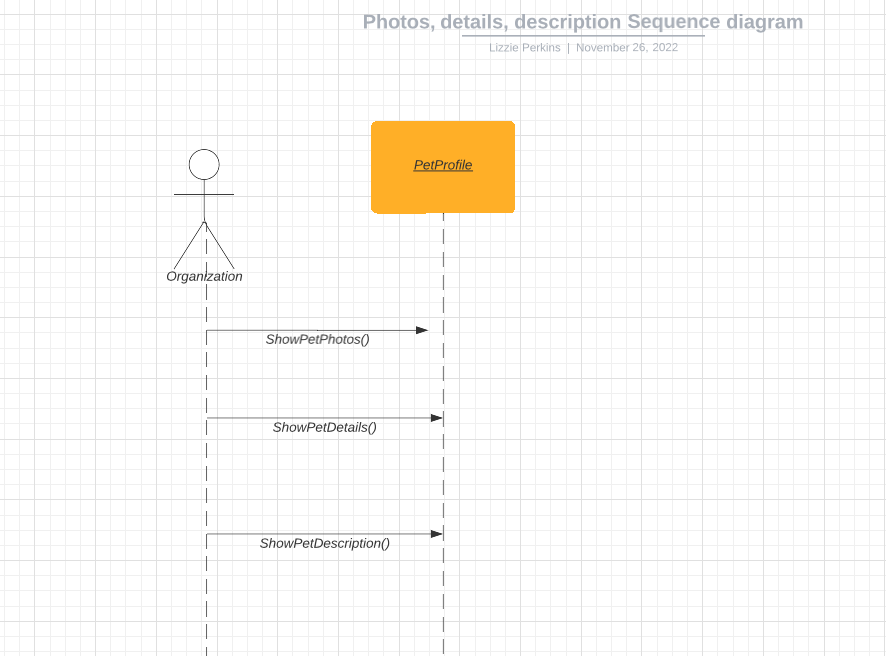


Narrative: The organization can show available pets and the user can filter through by details.

|  |
| --- |
| Use Case: Show Pet Photos |
| ID: 018 |
| Use Case type: detail, essential |
| Primary actor: Organization |
| Brief Description: Users can view multiple photos of each animal. |
| Importance level: high |
| Stakeholders: Organization and User |
| Trigger: Organization takes multiple photos of each animal |
| Relationships:  · Association: Show Pet details, Show Pet description  · Include: Add Pets, Edit Pets, Delete Pets  · Extend:  · Generalization: Pet Details |
| Flow of events:  1. Organization takes multiple photos of animal  2. Organization posts those photos on that animal’s profile  3. Users can see multiple photos of each animal |

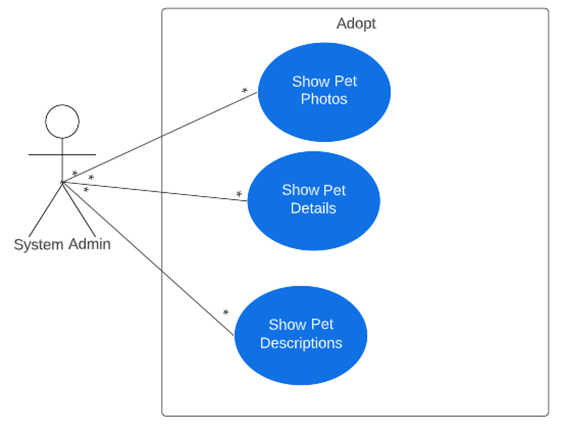
|  |
| --- |
| Use Case: Show Pet Details |
| ID: 019 |
| Use Case type: detail, essential |
| Primary actor: Organization |
| Brief Description: Users can click on the photo of the animal to see more details. |
| Importance level: high |
| Stakeholders: User and organization |
| Trigger: Organization accumulates information about the pet |
| Relationships:  · Association: Show Pet Photos, Show Pet Description  · Include: Add Pets, Edit Pets, Delete Pets  · Extend:  · Generalization: Pet Details |
| Flow of events:  1. Organization accumulates information about each pet  2. Organization puts that information on the animal’s profile  3. Users can see those details on the animal’s profile |

|  |
| --- |
| Use Case: Show Pet Description |
| ID: 020 |
| Use Case type: detail, essential |
| Primary actor: Organization |
| Brief Description: Users can see descriptions of each pet including species, breed, age, and other valuable information. |
| Importance level: high |
| Stakeholders: User and organization |
| Trigger: Organization creates a brief description about the pet |
| Relationships:  · Association: Show Pet Photos, Show Pet Details  · Include: Add Pets, Edit Pets, Delete Pets  · Extend:  · Generalization: Pet Details |
| Flow of events:  1. Organization writes a brief description about each pet  2. Organization puts that description on the animal’s profile  3. Users can see that description on the animal’s profile |



**Use case 18, 19, and 20 Main** **Flow:**

1. Organization can add multiple pet photos to pet profile.
2. Organization can show details about the pet on the pet’s profile.
3. Organization can show a description of the pet on the pet’s profile.



Narrative: The organization can show pictures, details, and a description of each animal.

**Use Case: Donate Tab**

|  |
| --- |
| **Use Case Name:** Create donation |
| **ID:** 021 |
| **Description:** User can create donation. |
| **Primary Actors:** User |
| **Secondary Actors:** None |
| **Preconditions:** User must have their card approved as the donation will be approved. |
| **Main Flow:** User will enter the site, click donate and then state the amount they want to donate. |
| **Post Conditions:** None |
| **Alternative Flows:** None |

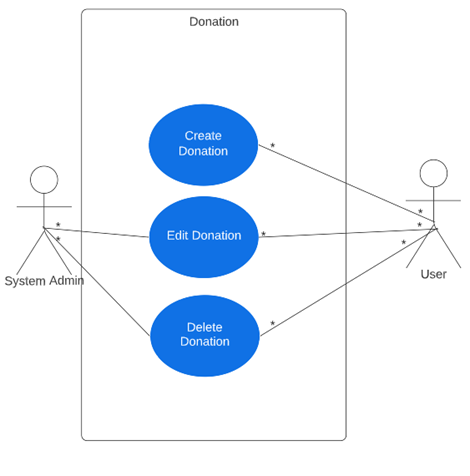
|  |
| --- |
| **Use Case Name:** Edit donation |
| **ID:** 022 |
| **Description:** User can edit their donation amount or timing. |
| **Primary Actors:** User |
| **Secondary Actors:** None |
| **Preconditions:** User must have a donation in the process of being made. |
| **Main Flow:** User will select which donation to edit, then choose if they would like to change the amount, time/date of donation, or to submit donation. |
| **Post Conditions:** Only user that the donation is edited by can change information. |
| **Alternative Flows:** None |

|  |
| --- |
| **Use Case Name:** Delete donation |
| **ID:** 023 |
| **Description:** User can delete donation in the process of it being made. |
| **Primary Actors:** User |
| **Secondary Actors:** None |
| **Preconditions:** User must have a donation in the process of being made. |
| **Main Flow:** User is in the process of donating then decides they do not want to, so they delete the donation. |
| **Post Conditions:** Only the user that was entering the information can delete the donation. |
| **Alternative Flows:** None |

|  |
| --- |
| **Use Case Name:** Edit Donation |
| **ID:** 024 |
| **Description:** Organization can edit donations |
| **Primary Actors:** Organization |
| **Secondary Actors:** None |
| **Preconditions:** Must be donations that have been submitted. |
| **Main Flow:** Organization can spot donations that they want to edit for whatever reason, then choose how to edit them. |
| **Post Conditions:** Only verified controllers can edit donations. |
| **Alternative Flows:** None |

|  |
| --- |
| **Use Case Name:** Delete donation |
| **ID:** 025 |
| **Description:** Organization can delete donations |
| **Primary Actors:** Organization |
| **Secondary Actors:** None |
| **Preconditions:** Must be donations that have been submitted. |
| **Main Flow:** Organization can choose to delete donation for any reason. |
| **Post Conditions:** Only verified controllers can delete donations. |
| **Alternative Flows:** None |

**Use Case 21, Use Case 22, Use Case 23, Use Case 24, Use Case 25 Main Flow:**

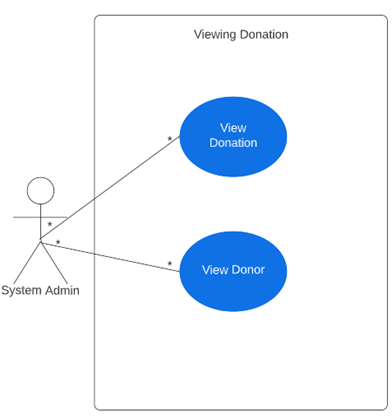


Narrative: The organization can edit or delete donations, and users can create, edit, and delete donations.

|  |
| --- |
| **Use Case Name:** View donation |
| **ID:** 026 |
| **Description:** Organization can view donations after they have been submitted. |
| **Primary Actors:** Organization |
| **Secondary Actors:** None |
| **Preconditions:** Donation must have been submitted to the database. |
| **Main Flow:** Database stores donations, then the organization can look at them later. |
| **Post Conditions:** Only verified personnel can view donations. |
| **Alternative Flows:** None |

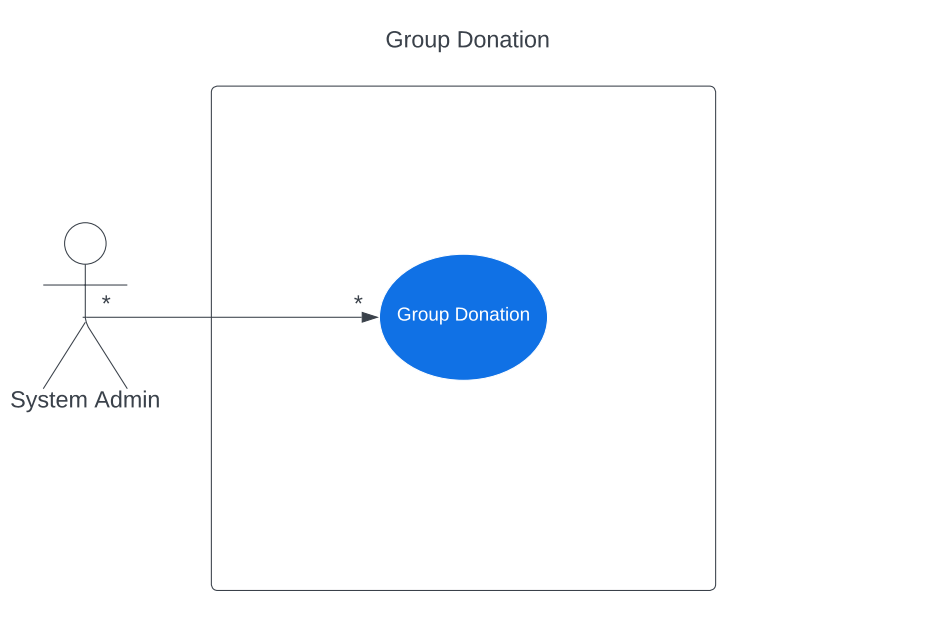
|  |
| --- |
| **Use Case Name:** View donors |
| **ID:** 027 |
| **Description:** Organization can view people that have donated. |
| **Primary Actors:** Organization |
| **Secondary Actors:** None |
| **Preconditions:** Public donations must have been submitted to the database. |
| **Main Flow:** Organization can view people that have donated and the information of the donation. |
| **Post Conditions:** Only verified users can view the people that have donated. |
| **Alternative Flows:** None |

**Use Case 26 & Use Case 27**



Narrative: Organization can view donations and donors.

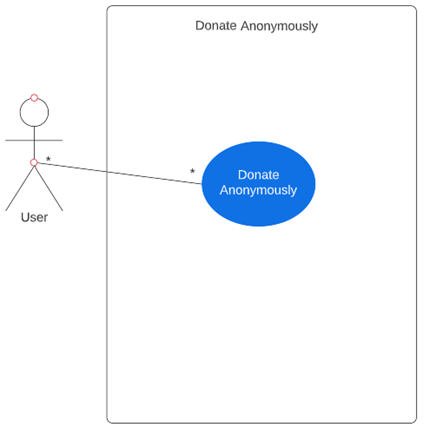
|  |
| --- |
| **Use Case Name:** Group donation |
| **ID:** 028 |
| **Description:** Organization can group donation by amount or when donated. |
| **Primary Actors:** Organization |
| **Secondary Actors:** None |
| **Preconditions:** Donations must be submitted for grouping. |
| **Main Flow:** Organization can make a search for donations based on amount or date/time donated. |
| **Post Conditions:** Only verified users can search and group donations. |
| **Alternative Flows:** None |



Narrative: Organization can group donation by amount and dates.

|  |
| --- |
| **Use Case Name:** Hide donation |
| **ID:** 029 |
| **Description:** User can choose to remain anonymous. |
| **Primary Actors:** User |
| **Secondary Actors:** None |
| **Preconditions:** User must have a donation in the process of being made. |
| **Main Flow:** As the user is donating, they can choose to remain anonymous at the end. |
| **Post Conditions:** None |
| **Alternative Flows:** None |

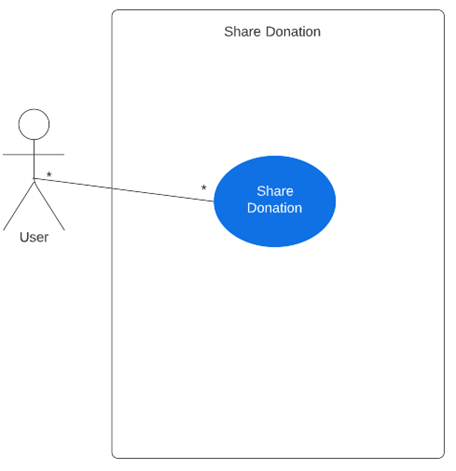
**Use Case 29**



Narrative: User can choose to anonymously donate.

|  |
| --- |
| **Use Case Name:** Share donation |
| **ID:** 030 |
| **Description:** User can choose to share their donation. |
| **Primary Actors:** User |
| **Secondary Actors:** None |
| **Preconditions:** User must have a donation in the process of being made. |
| **Main Flow:** As the user is donating, they can choose to share their donation via different mediums |
| **Post Conditions:** None |
| **Alternative Flows:** None |

**Use Case 30**



Narrative: User can choose to share donation.

|  |
| --- |
| **Use Case Name:** Specify donation |
| **ID:** 031 |
| **Description:** User can specify which area of the organization will benefit. |
| **Primary Actors:** User |
| **Secondary Actors:** None |
| **Preconditions:** User must have a donation in the process of being made. |
| **Main Flow:** As the user is creating the donation, they can choose to pick an area of the organization to send the money too. |
| **Post Conditions:** None |
| **Alternative Flows:** None |

**Use Case 31**

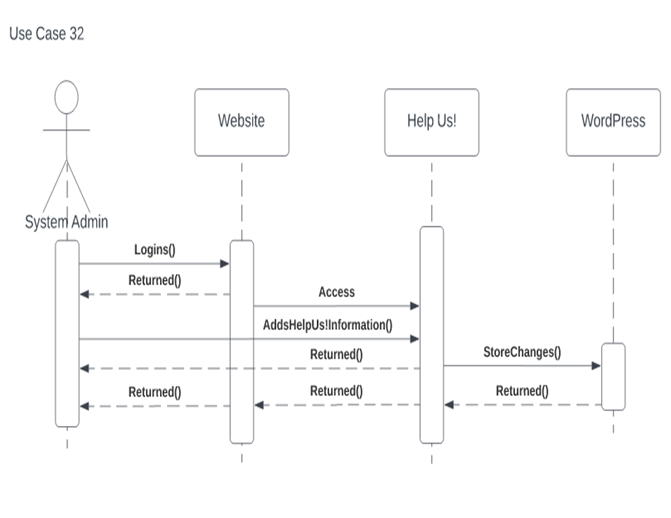


Narrative: User can choose what area they want their donation to go to.

**Use Case Diagram: Help Us! Tab**

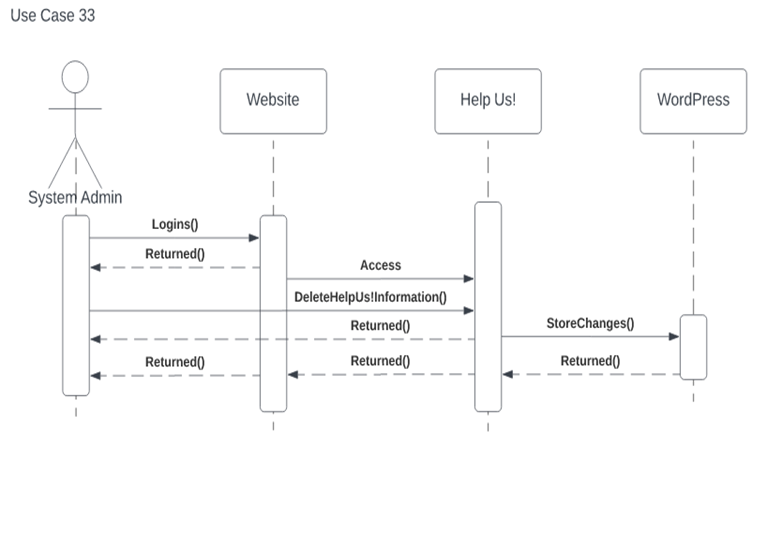
|  |
| --- |
| Use Case Name: Add Help Us! Information |
| ID: 32 |
| Risk Level: Low |
| Description: Through WordPress, the client will be able to add information into the Help Us! Tab. The administrator will access the CMS and use it to add information needed. |
| Primary Actors: System Administrator |
| Normal Flow of Events Flow:   1. System Administrator decides that new information or additional information is needed in the Help Us! Section. 2. Admin logins to WordPress Website    1. Admin will hover over pages    2. Admin clicks Page/Post Editor 3. Admin will then be placed into Page/Post Editor    1. Admin will click on Help Us! 4. Admin will then add information to Help Us! Section |
| Alternative Flows: None |
| Preconditions: The system administrator will need to have access to the content management system to perform this. |
| Postconditions: The system administrator may expect new information to be inserted into the system. This should take immediate effect on the system |
| Special Requirements: None |

**Use Case 32 Main Flow:**

1. System Administrator decides that new information or additional information is needed in the Help Us! Section.
2. Admin logins to WordPress Website
   1. Admin will hover over pages
   2. Admin clicks Page/Post Editor
3. Admin will then be placed into Page/Post Editor
   1. Admin will click on Help Us!
4. Admin will then add information to Help Us! Section

|  |
| --- |
| Use Case Name: Delete Help Us! Information |
| ID: 33 |
| Risk Level: High |
| Description: With the content management system, the administrator will be able to delete information in the Help Us! Tab. Information may not be needed and may be crucial to users. |
| Primary Actors: Organization |
| Normal Flow of Events Flow:   1. System Administrator decides that information is incorrect, or some information isn’t relevant to the organization 2. Admin logins to WordPress Website    1. Admin will hover over pages    2. Admin will then be placed into Page/Post Editor    3. Admin will click on Help Us! 3. Admin will then delete information to Help Us! Section |
| Alternative Flows: None |
| Preconditions: The system administrator must have access to the content management system to perform this action |
| Postconditions: The system will present information in a different manner. |
| Special Requirements: None |

**Use Case 33 Main Flow:**

1. System Administrator decides that information is incorrect, or some information isn’t relevant to the organization
2. Admin logins to WordPress Website
   1. Admin will hover over pages
   2. Admin will then be placed into Page/Post Editor
   3. Admin will click on Help Us!
3. Admin will then delete information to Help Us! Section

**Use Case 32 & 33 Use Case Diagram**



Narrative: The actor, system admin, can add and delete information regarding Help Us! Tab. This may consist of potential volunteer positions and other opportunities to help.

**Use Case 34: Create Application & Use Case 35: Delete Applications**

|  |
| --- |
| Use Case Name: Create Volunteer Application |
| ID: 34 |
| Risk Level: Low |
| Description: This use case describes a user's interest in completing a Help Us! Application. This will occur through the WordPress website and the user will fill out a form. |
| Primary Actors: User |
| Normal Flow of Events Flow:   1. User will access webpage 2. User will login to the webpage 3. User will hover over Help Us! Tab 4. User will click on Volunteer Application 5. Volunteer Application has been created |
| Alternative Flows:   1. User will access webpage 2. User will login to the webpage 3. User will hover over Help Us! Tab 4. User will access Volunteer Information Page 5. User will view volunteer access page 6. User clicks on Volunteer Application Link |
| Preconditions: The system will not necessarily be affected since the user will view it in essentially a read-only mode. |
| Postconditions: The system will shortly receive that an application has been created by a user. |
| Special Requirements: Nonfunctional requirements concerning security will be established. The CMS will need to maintain a secure status to protect user information. |

Diagram

Description automatically generated

**Use Case 34 Main Flow:**

1. User will access webpage
2. User will login to the webpage
3. User will hover over Help Us! Tab
4. User will access Volunteer Information Page
5. User will view volunteer access page
6. User clicks on Volunteer Application Link

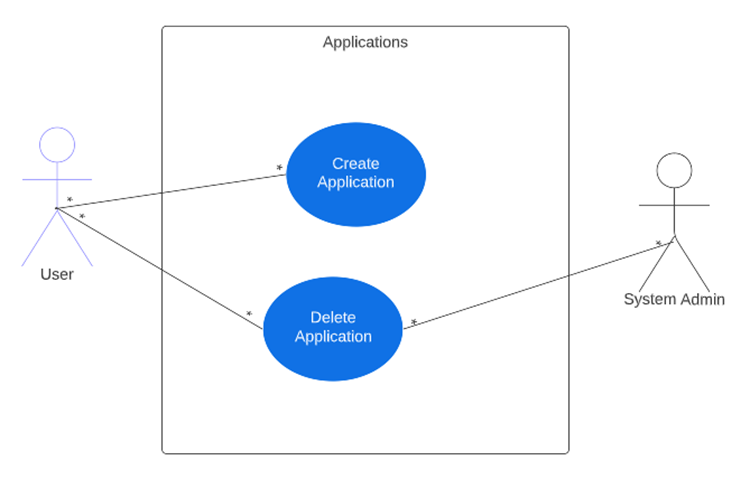
|  |
| --- |
| Use Case Name: Delete Applications |
| ID: 35 |
| Risk Level: High |
| Description: Through WordPress, the system administration can delete applications if deemed necessary. |
| Primary Actors: Organization |
| Normal Flow of Events Flow:   1. System Admin logins to WordPress database 2. System Admin hovers over new volunteer applications 3. System Admin views new volunteer applications 4. System admin determines and deletes application if deemed necessary. |
| Alternative Flows: None |
| Preconditions: An applicant's information in the Help Us! Section will be stored within the system |
| Postconditions: The system will be altered in the sense that the system administrator has deleted a potential Help Us! applicant. |
| Special Requirements: None |

**Use Case 35 Main Flow:**

1. System Admin logins to WordPress database
2. System Admin hovers over new volunteer applications
3. Diagram

   Description automatically generatedSystem Admin views new volunteer applications
4. System admin determines and deletes application if deemed necessary.

**Use Case Diagram: 34 & 35**



Narrative: This use case displays the user and system administrator. The user can create a multitude of applications and delete them if necessary. Deletion applies to System Administrator, if necessary, too.

**Use Case 36: Submit Application**

|  |
| --- |
| Use Case Name: Submit Applications |
| ID: 36 |
| Risk Level: Low |
| Description: After users have completed their applications, they will be able to submit them to the client. This will occur through WordPress and the client will receive them. The purpose of this application is to express interest in volunteering for the client. |
| Primary Actors: User |
| Normal Flow of Events Flow:   1. User will access webpage 2. User will login to the webpage 3. User will hover over Help Us! Tab 4. User will click on Volunteer Application 5. Volunteer Application has been created 6. User will fill out information 7. User can submit application |
| Alternative Flows: None |
| Preconditions: WordPress is a precondition to this use case. The user will have an application that displays their interest in volunteering. |
| Postconditions: After submission, the user will have completed their part and the system will be altered in the sense that a new application has been received. |
| Special Requirements: Nonfunctional requirements concerning security will be established. The CMS will need to maintain a secure status to protect user information. |

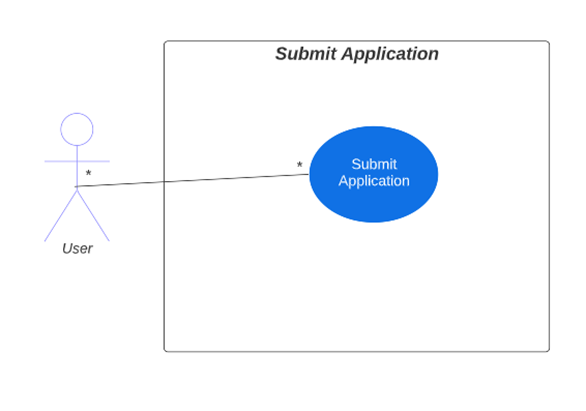
**Diagram

Description automatically generated**

**Use Case 36 Main Flow:**

1. User will access webpage
2. User will login to the webpage
3. User will hover over Help Us! Tab
4. User will click on Volunteer Application
5. Volunteer Application has been created
6. User will fill out information
7. User can submit application

**Use Case 36 Diagram:**



Narrative: Users will be able to submit their Help Us! Applications after completion.

**Use Case 37: Show Volunteer Positions**

|  |
| --- |
| Use Case Name: Show Volunteer Positions |
| ID: 37 |
| Risk Level: Low |
| Description: The organization will use the content management system to display volunteer positions within the Help Us! Section of the website. The purpose of this is to allow users to explore the volunteer opportunities that the client has to offer. |
| Primary Actors: Organization |
| Normal Flow of Events Flow:   1. System Administrator decides that information is incorrect, or some information isn’t relevant to the organization 2. Admin logins to WordPress Website 3. Admin will hover over pages 4. Admin will then be placed into Page/Post Editor 5. Admin will click on Help Us! 6. Admin will click on Volunteer Information page 7. Admin will then add volunteer positions |
| Alternative Flows: None |
| Preconditions: The preceding system will show a website without information regarding volunteering opportunities. The system admin will also need to use WordPress as a precondition to complete this matter. |
| Postconditions: After showing positions, the system will present an altered page to the users. |
| Special Requirements: None |

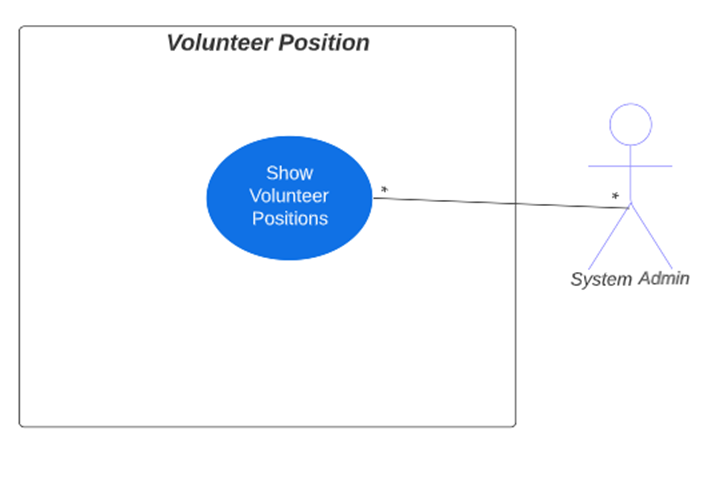
**Diagram

Description automatically generated**

**Use Case 37 Main Flow**

1. System Administrator decides that information is incorrect, or some information isn’t relevant to the organization
2. Admin logins to WordPress Website
3. Admin will hover over pages
4. Admin will then be placed into Page/Post Editor
5. Admin will click on Help Us!
6. Admin will click on Volunteer Information page
7. Admin will then add volunteer positions

**Use Case 37 Diagram:**



Narrative: System admin will be able to display volunteer positions within the Help Us! section.

**Use Case 38:** Add Volunteer Information **Use Case 39:** Edit Volunteer Information, **Use Case 40:** Delete Volunteer Information

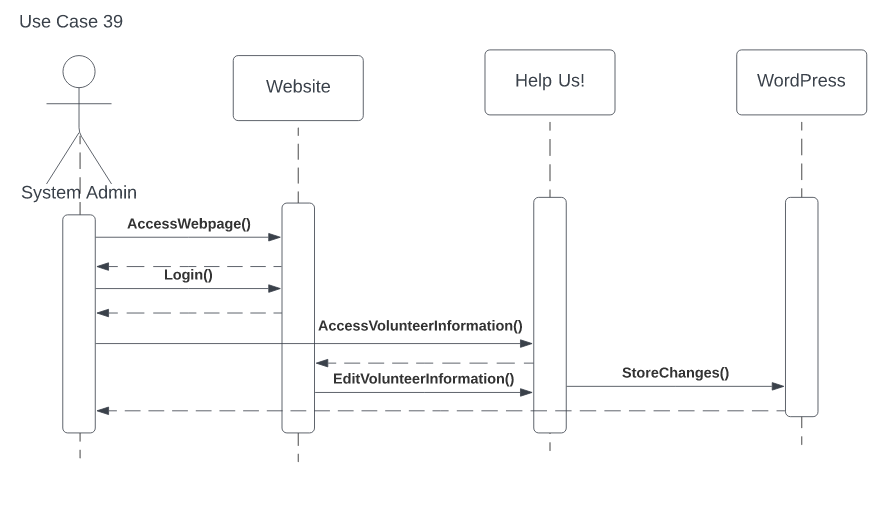
|  |
| --- |
| Use Case Name: Add Volunteer Information |
| ID: 38 |
| Risk Level: Low |
| Description: Within the application process, users can volunteer information. This information may concern their name, phone number, and areas of interest. The CMS will be responsible for taking in this information through the Help Us! Section of the system |
| Primary Actors: User |
| Normal Flow of Events Flow:   1. User will express interest in volunteering for organization 2. User accesses WordPress Webpage 3. User logins or signs up for webpage 4. User hovers over Help Us! Tab 5. User clicks on Volunteer Application 6. User views Volunteer Application 7. User adds information to volunteer application. |
| Alternative Flows:   1. User will express interest in volunteering for organization 2. User accesses WordPress Webpage 3. User logins or signs up for webpage 4. User hovers over Help Us! Tab 5. User clicks on Volunteer Information 6. User views Volunteer Information 7. User clicks on Volunteer Application Link 8. User adds information to volunteer application. |
| Preconditions: Information regarding the applicant will be nonexistent within the system. |
| Postconditions: The system will be altered due to the added information. Information will be stored based on the user's input. |
| Special Requirements: None |

**Use Case 38 Main Flow:**

1. Diagram

   Description automatically generatedUser will express interest in volunteering for organization
2. User accesses WordPress Webpage
3. User logins or signs up for webpage
4. User hovers over Help Us! Tab
5. User clicks on Volunteer Application
6. User views Volunteer Application
7. User adds information to volunteer application.

|  |
| --- |
| Use Case Name: Edit Volunteer Information |
| ID: 39 |
| Risk Level: Low |
| Description: Once the Help Us! Application is created, users will have the opportunity to edit their information. Information from the user may be incorrect giving the user a need to correct it for accuracy. |
| Primary Actors: User |
| Normal Flow of Events Flow:   1. User is currently reviewing volunteer application 2. User determines that some information is incorrect 3. User edits volunteer information |
| Alternative Flows: None |
| Preconditions: Information will be presented within the application and the user may have information that needs changed. |
| Postconditions: The system will be altered within the user’s application |
| Special Requirements: None |



**Use Case 39 Main Flow:**

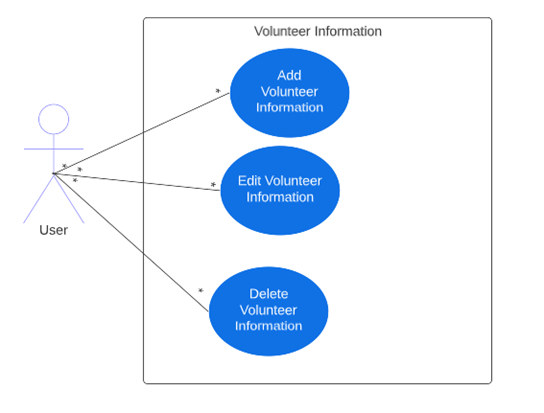
1. User is currently reviewing volunteer application
2. User determines that some information is incorrect
3. User edits volunteer information

|  |
| --- |
| Use Case Name: Delete Volunteer Information |
| ID: 40 |
| Risk Level: High |
| Description: With this use case, personal information within the Help Us! Application may need to be deleted. Information regarding the user could be incorrect, causing the user to delete the information and potentially replace it. For example, the user may have decided to delete areas of interest. |
| Primary Actors: User |
| Normal Flow of Events Flow:   1. User is currently viewing the volunteer application 2. User determines that some information isn’t needed 3. User deletes information. |
| Alternative Flows: None |
| Preconditions: Current information on the system’s user application will remain unaltered. |
| Postconditions: The system will present altered information within the user’s application. |
| Special Requirements: None |

**Use Case 40 Main Flow:**

1. User is currently viewing the volunteer application
2. User determines that some information isn’t needed
3. Diagram

   Description automatically generatedUser deletes information.



Narrative: The user will be able to add, alter, and delete volunteering information inside of the Help Us! Application.

**Use Case 41: Add Availability, Use Case 42: Edit Availability, & Use Case 43: Delete Availability**

|  |
| --- |
| Use Case Name: Add Availability |
| ID: 41 |
| Risk Level: Low |
| Description: Through WordPress, the system administrator will be able to display volunteer availability. The purpose of this is to inform users about availability for users to volunteer. |
| Primary Actors: System Administrator |
| Normal Flow of Events Flow:   1. System Administrator decides that new information or additional information is needed in the Help Us! Section. 2. Admin logins to WordPress Website 3. Admin will hover over pages 4. Admin clicks Page/Post Editor 5. Admin will then be placed into Page/Post Editor 6. Admin will click on Help Us! 7. Admin clicks on Volunteer Information 8. Admin will then add availability information to Volunteer Information |
| Alternative Flows: None |
| Preconditions: The system will need access to the content management system. Our client’s system will need to present a state where adding availability times is needed. |
| Postconditions: The system will be altered and display availability times. |
| Special Requirements: None |

**Use Case 41 Main Flow:**

1. System Administrator decides that new information or additional information is needed in the Help Us! Section.
2. Diagram

   Description automatically generatedAdmin logins to WordPress Website
3. Admin will hover over pages
4. Admin clicks Page/Post Editor
5. Admin will then be placed into Page/Post Editor
6. Admin will click on Help Us!
7. Admin clicks on Volunteer Information
8. Admin will then add availability information to Volunteer Information

|  |
| --- |
| Use Case Name: Edit Availability |
| ID: 42 |
| Risk Level: Low |
| Description: Through the content management system, WordPress, the system administrator will be able to edit availability times within the Help Us! Section. The purpose of this is to edit current availability times if deemed necessary by the client. |
| Primary Actors: System Administrator |
| Normal Flow of Events Flow:   1. System Administrator decides that information is not available or relevant to the Help Us! Section. 2. Admin logins to WordPress Website 3. Admin will hover over pages 4. Admin clicks Page/Post Editor 5. Admin will then be placed into Page/Post Editor 6. Admin will click on Help Us! 7. Admin clicks on Volunteer Information 8. Admin will then edit availability information to Volunteer Information |
| Alternative Flows: None |
| Preconditions: The client will need access to WordPress to perform this function. The current, precondition system will display old availability times. |
| Postconditions: The system will present altered availability times. |
| Special Requirements: None |

**Use Case 42 Main Flow:**

1. System Administrator decides that information is not available or relevant to the Help Us! Section.
2. Admin logins to WordPress Website
3. Admin will hover over pages
4. Diagram

   Description automatically generatedAdmin clicks Page/Post Editor
5. Admin will then be placed into Page/Post Editor
6. Admin will click on Help Us!
7. Admin clicks on Volunteer Information
8. Admin will then edit availability information to Volunteer Information

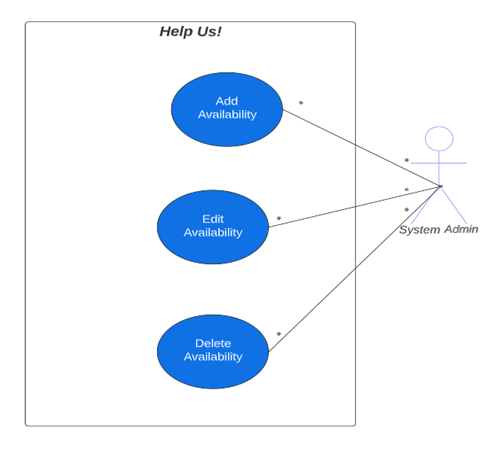
|  |
| --- |
| Use Case Name: Delete Availability |
| ID: 43 |
| Risk Level: High |
| Description: Using the content management system, WordPress, the system administrator will be able to delete availability times in the volunteer section of Help Us!. The purpose of this is to create an updated version of availability times if current availability is not possible to the client. |
| Primary Actors: System Administrator |
| Normal Flow of Events Flow:   1. System Administrator decides that new information or additional information is needed in the Help Us! Section. 2. Admin logins to WordPress Website 3. Admin will hover over pages 4. Admin clicks Page/Post Editor 5. Admin will then be placed into Page/Post Editor 6. Admin will click on Help Us! 7. Admin clicks on Volunteer Information 8. Admin will then delete availability information to Volunteer Information |
| Alternative Flows: None |
| Preconditions: The system may present inaccurate availability times. Access to this will need to be through WordPress. |
| Postconditions: After changes have been made, the system will present an altered version within availability times. |
| Special Requirements: None. |

**Use Case 43 Main Flow:**

1. Diagram

   Description automatically generatedSystem Administrator decides that information is not available or relevant to the Help Us! Section.
2. Admin logins to WordPress Website
3. Admin will hover over pages
4. Admin clicks Page/Post Editor
5. Admin will then be placed into Page/Post Editor
6. Admin will click on Help Us!
7. Admin clicks on Volunteer Information
8. Admin will then edit availability information to Volunteer Information

**Use Case 41, 42 & 43 Diagram:**



Narrative: The actor, System administrator, can add, edit, and delete availability times for volunteering opportunities.

**Use Case 44. View Applications**

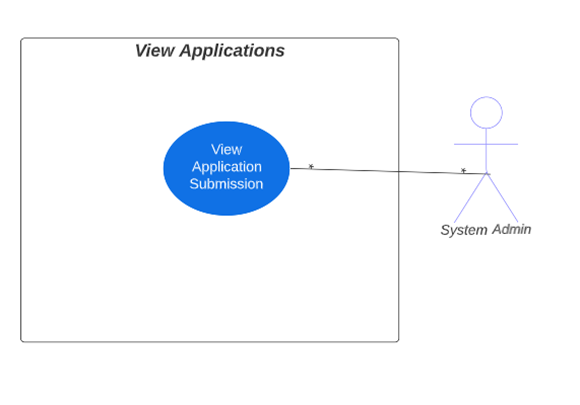
|  |
| --- |
| Use Case Name: View Application Submissions |
| ID: 44 |
| Risk Level: Low |
| Description: After applications have been submitted through WordPress, the organization will be able to view submitted applications. |
| Primary Actors: Organization |
| Normal Flow of Events Flow:   1. System Admin receives new volunteer applications 2. System admin accesses WordPress Webpage 3. System admin logins to Webpage 4. System Admin accesses database 5. System admin views submitted applications |
| Alternative Flows: Applications may take a different approach depending on the user. The user could have directly contacted the client as a means for expressing interest and took an alternative route for submitting an application. |
| Preconditions: Submitted applications will be stored within the system. The system admin will need to access WordPress to view these applications. |
| Postconditions: Postconditions with this use case will alter the system. |
| Special Requirements: None |

**Use Case 44 Main Flow:**

1. System Admin receives new volunteer applications
2. System admin accesses WordPress Webpage
3. Diagram

   Description automatically generatedSystem admin logins to Webpage System Admin accesses database
4. System admin views submitted applications

**Use Case 44 Diagram:**



Narrative: The System Admin will be able to view submitted applications from users.

**Use Case 45:** Add Areas of Interest, **Use Case 46:** Edit Areas of Interest, **Use Case 47:** Delete Areas of Interest

|  |
| --- |
| Use Case Name: Add Areas of Interest |
| ID: 45 |
| Risk Level: Low |
| Description: Adding areas of interest allows the user to choose volunteering positions that they may be interested in. This would take place during the application process in the Help Us! section. |
| Primary Actors: Users |
|  |
| Normal Flow of Events Flow:   1. User accesses webpage 2. User logs on or signs up the web page 3. User creates volunteer application 4. User fills contact information 5. User adds area of interest |
| Alternative Flows: None |
| Preconditions: Preconditions require that the system follow the WordPress website and that the user accesses the web page. The preceding system may display no information regarding the user’s interest for positions. |
| Postconditions: After completing this section, the system will present a completed section for areas of interest within the Help Us! application. |
| Special Requirements: None |

Diagram

Description automatically generated **Use Case 45 Main Flow:**

1. User accesses webpage
2. User logs on or signs up the web page
3. User creates volunteer application
4. User fills contact information
5. User adds area of interest

|  |
| --- |
| Use Case Name: Edit Areas of Interest |
| ID: 46 |
| Risk Level: Low |
| Description: Adding areas of interest allow for the user to choose volunteering positions that they may be interested in. This would take place during the application process in Help Us! Section. Editing areas of interest serves the purpose of allowing the user to change volunteering positions in the application. Maybe a certain position does not fit their volunteering criteria and needs changed. |
| Primary Actors: User |
| Normal Flow of Events Flow:   1. User accesses webpage 2. User logs on or signs up for the web page 3. User creates volunteer application 4. User fills contact information 5. User adds area of interest 6. User determines that they’re not interested in a position 7. User edits areas of interest. |
| Alternative Flows: None |
| Preconditions: Criteria for interest may not be what the user wants or maybe the user wants to add more to their interest. The current system displays their current area of interest. |
| Postconditions: After editing, the user may have added or deleted areas of interest, thus presenting an altered system. |
| Special Requirements: None. |

**Use Case 46 Main Flow:**

1. User accesses webpage
2. **Diagram

   Description automatically generated**User logs on or signs up for the web page
3. User creates volunteer application
4. User fills contact information
5. User adds area of interest
6. User determines that they’re not interested in a position
7. User edits areas of interest.

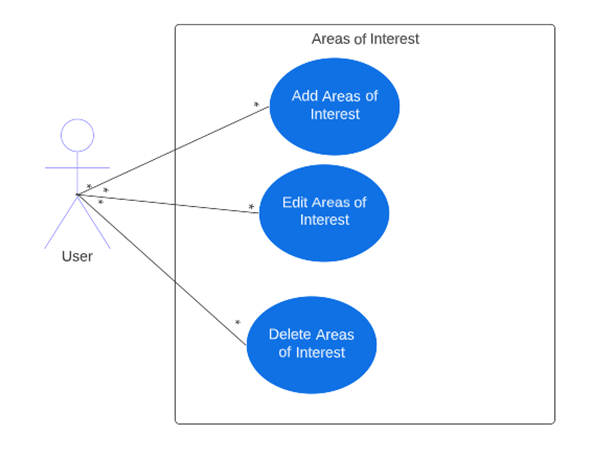
|  |
| --- |
| Use Case Name: Delete Areas of Interest |
| ID: 47 |
| Risk Level: Low |
| Description:  Adding areas of interest allow for the user to choose volunteering positions that they may be interested in. This would take place during the application process in the Help Us! Section. Deleting an area of interest allows the user to completely remove a volunteering position from their application. |
| Primary Actors: User |
| Normal Flow of Events Flow   1. User accesses webpage 2. User logs on or signs up for the web page 3. User creates volunteer application 4. User fills contact information 5. User adds area of interest 6. User determines that they’re not interested in a position 7. User edits areas of interest. |
| Alternative Flows: None |
| Preconditions: May show areas of interest that do not fit the criteria of the applicant and the user will need to delete them. |
| Postconditions: System will be altered within the application. |
| Special Requirements: None |

**Use Case 47 Main Flow:**

1. User accesses webpage
2. User logs on or signs up for the web page
3. Diagram

   Description automatically generatedUser creates volunteer application
4. User fills contact information
5. User adds area of interest
6. User determines that they’re not interested in a position
7. User edits areas of interest.

**Use Case 45, 46, & 47 Diagram**



Narrative: Users will be able to select areas of interest within the Help Us! Application.

**Use Case: Event Tab**

|  |
| --- |
| Use Case Name: Create Event |
| ID: 43 |
| Risk Level: Low |
| Description: The organization can create events to display on the events tab |
| Primary Actors: Admin |
| Normal Flow of Events Flow   1. The use case begins when the admin is logged into the system 2. The admin creates event content |
| Alternative Flows: None |
| Preconditions: The admin must be verified and logged into the system to create an event. |
| Postconditions: The system will show event on calendar. |
| Special Requirements: None |

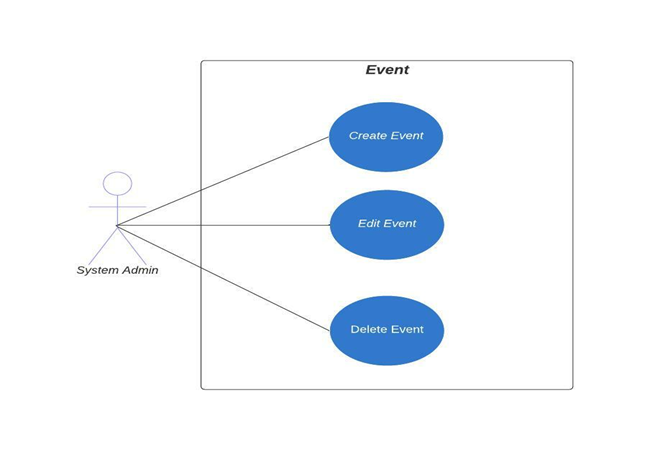
|  |
| --- |
| Use Case Name: Edit Event |
| ID: 44 |
| Risk Level: Low |
| Description: The organization can edit events as needed. |
| Primary Actors: Admin |
| Normal Flow of Events Flow   1. The use case begins when the admin logs into the system 2. The admin accesses all current events and edits them as needed |
| Alternative Flows: None |
| Preconditions: The admin must be verified and logged into the system to access event details. |
| Postconditions: The system will confirm edits. |
| Special Requirements: None |

|  |
| --- |
| Use Case Name: Delete Event |
| ID: 45 |
| Risk Level: High |
| Description: The organization can delete events as needed |
| Primary Actors: Admin |
| Normal Flow of Events Flow   1. The use case begins when the admin is logged into the system 2. The admin accesses all current events and deletes them as needed |
| Alternative Flows: None |
| Preconditions: The admin must be verified and logged into the system to delete events. |
| Postconditions: **:** The system will confirm the deletion. |
| Special Requirements: None |

**Box and whisker chart

Description automatically generated with medium confidenceUse Case 48-50 Main Flow:**

1. Admin enters log in to system administrator account.
2. Admin Navigates to the event calendar.
3. Selects create event/edit event/delete event/edit attendees.
4. Selects submit.



|  |
| --- |
| Use Case Name: View Event Calendar |
| ID: 46 |
| Risk Level: Low |
| Description:  The user can view the event calendar. |
| Primary Actors: User |
| Normal Flow of Events Flow   1. The use case begins when the user visits the website 2. The user views the event calendar |
| Alternative Flows: None |
| Preconditions: The user must be on the event tab to view the calendar. |
| Postconditions: The calendar will remain on page. |
| Special Requirements: None |

|  |
| --- |
| Use Case Name: Add Attendee |
| ID: 47 |
| Risk Level: Low |
| Description: The user can sign up to attend events. |
| Primary Actors: User |
| Normal Flow of Events Flow   1. The use case begins when the user visits the event tab on the website 2. The user clicks RSVP on the calendar 3. The user gives contact information and registers for an account |
| Alternative Flows: None |
| Preconditions: The user must RSVP to the event, create an account, and give demographic background. |
| Postconditions: The user receives a confirmation for attendee information. |
| Special Requirements: None |

|  |
| --- |
| Use Case Name: Edit Attendee |
| ID: 48 |
| Risk Level: Low |
| Description: The user can edit their attendee status as needed. |
| Primary Actors: User |
| Normal Flow of Events Flow   1. The use case begins when the user visits the event tab on the website 2. The user must log in to registered information before making edits 3. The user edits attendee information |
| Alternative Flows: None |
| Preconditions: The user must be logged into their user account. |
| Postconditions: The user receives a confirmation for edits. |
| Special Requirements: None |

|  |
| --- |
| Use Case Name: Cancel Attendee |
| ID: 49 |
| Risk Level: Low |
| Description:  The user can cancel their attendee status. |
| Primary Actors: User |
| Normal Flow of Events Flow   1. The use case begins when the user visits the event tab on the website 2. The user must log in to registered information before canceling their attendee status 3. The user cancels their attendee status |
| Alternative Flows: None |
| Preconditions: The user must be logged into their account. |
| Postconditions: The user receives a confirmation for cancelation. |
| Special Requirements: None |

|  |
| --- |
| Use Case Name: Share Event |
| ID: 50 |
| Risk Level: Low |
| Description: The user can share the event with others. |
| Primary Actors: User |
| Normal Flow of Events Flow   1. The use case begins when the user accesses the event calendar 2. The user can then access the event to access the link to share the event |
| Alternative Flows: None |
| Preconditions: The user must be on the event tab |
| Postconditions: The user can copy/paste a link to the event details. |
| Special Requirements: None |

|  |
| --- |
| Use Case Name: Delete Attendee |
| ID: 51 |
| Risk Level: Low |
| Description: The organization can cancel their attendee status. |
| Primary Actors: Admin |
| Normal Flow of Events Flow   1. The use case begins when the admin is logged into the system 2. The admin can then access all attendee information 3. The admin deletes attendees as needed |
| Alternative Flows: None |
| Preconditions: The admin must be verified and logged into the system to create an event. |
| Postconditions: The admin will receive confirmation of the deletion. |
| Special Requirements: None |
| Use Case Name: Get Directions to Event |
| ID: 52 |
| Risk Level: Low |
| Description: The user can access the location of the event. |
| Primary Actors: User |
| Normal Flow of Events Flow   1. The use case begins when the user accesses the event calendar 2. The user can then access the details of the event by clicking on the event on the calendar 3. The user accesses details such as event time/date and location |
| Alternative Flows: None |
| Preconditions: The user must be on event details |
| Postconditions: The user will view event details. |

**A picture containing box and whisker chart

Description automatically generated**

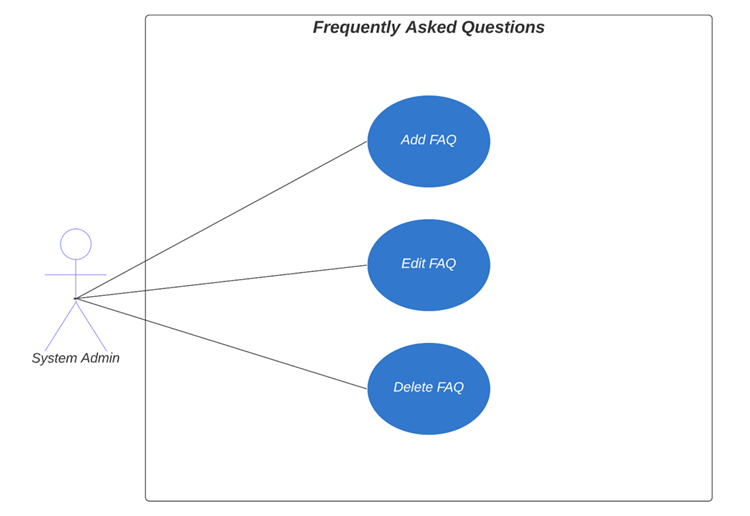
**Use Case 51-57 Main Flow:**

1. The user visits the website.
2. The user enters login.
3. The user navigates to the event calendar.
4. The user can add/edit/cancel their attendee status.
5. The user can share the event.
6. The user can access the event address for directions.



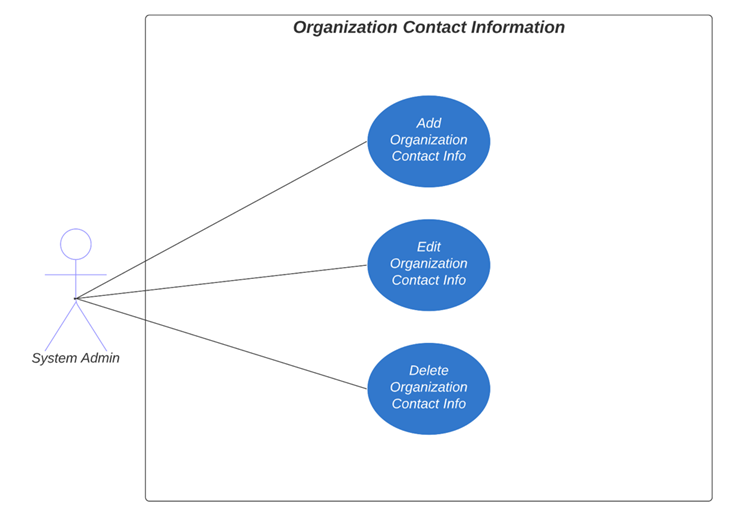
**Use Case: Contact Us! Tab**

**Use Case 58 - 60: Add, Edit, Delete Frequently Asked Questions (FAQ)**



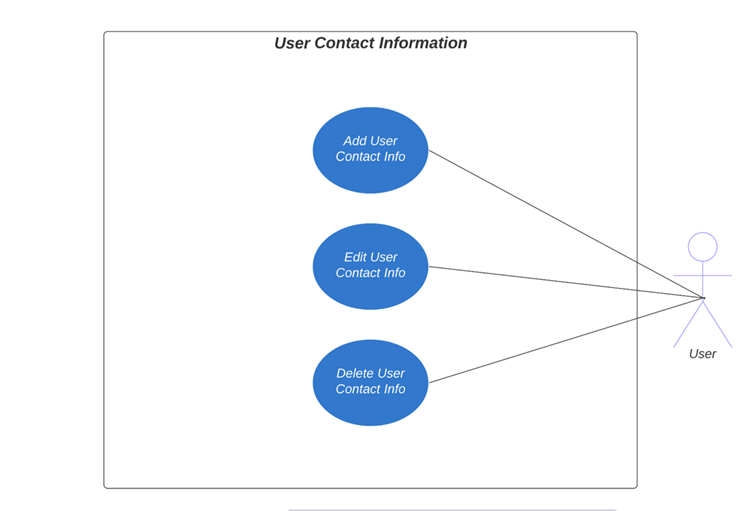
Narrative: The system admin will be able to add, edit and delete Frequently Asked Questions.

**Use Case 61 - 63: Add, Edit, Delete Organization Contact information**



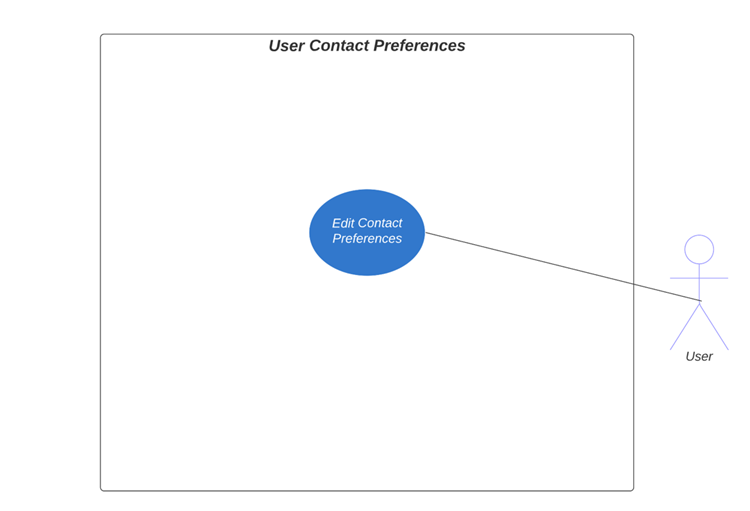
Narrative: The System admin will be able to create, edit and delete Contact Information.

**Use Case 64-66: Add, Edit, Delete User Contact Information**



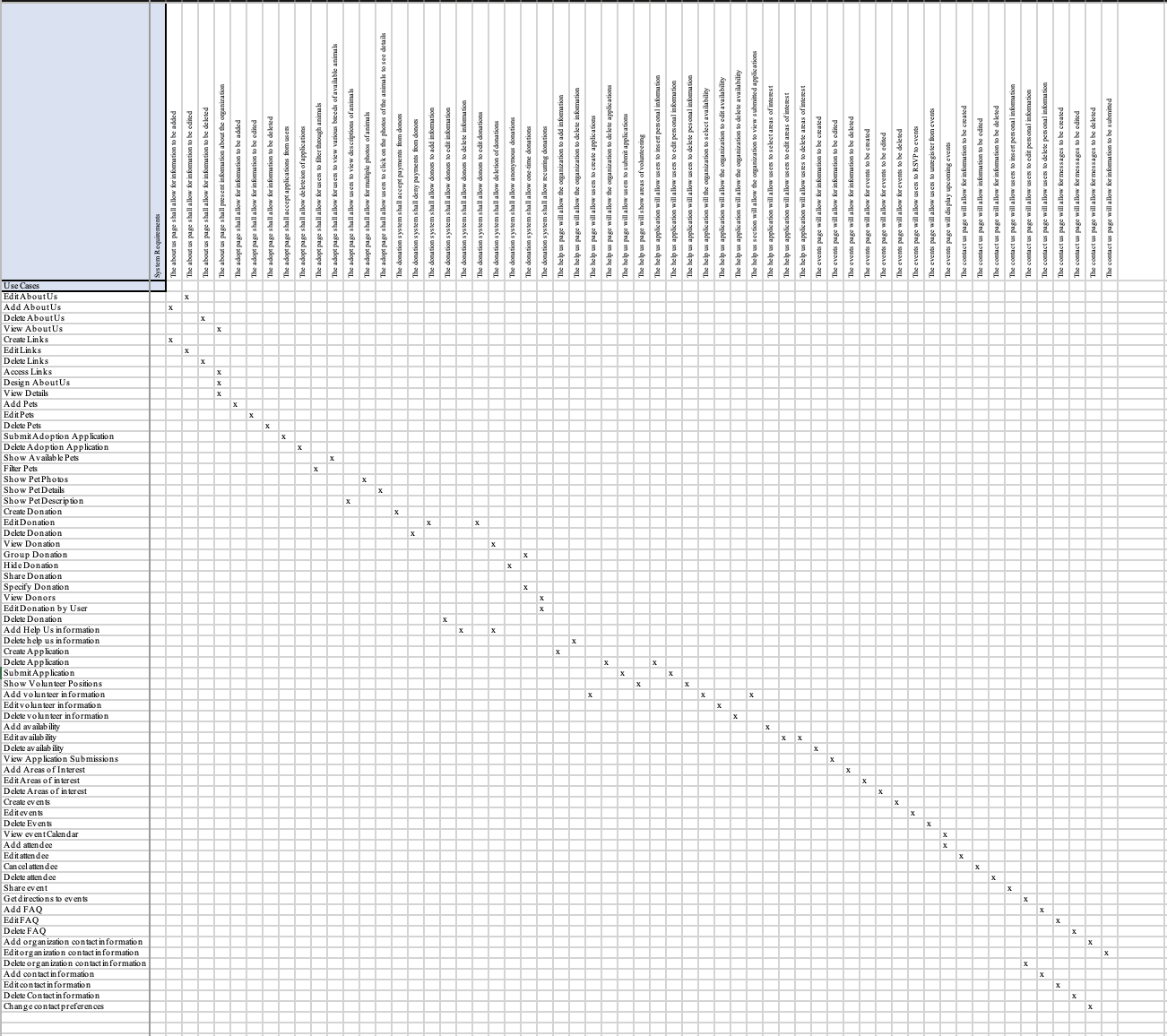
Narrative: The User will be able to add, edit and delete their information using a contact us form.

**Use Case 67: Change Contact Preferences**



Narrative: The User will be able to change the method of contact from the Organization.

**Trace Matrix**



**Narrative:** This diagram is to show the correlation between system requirements and use cases.

**Class Diagram**

Diagram, schematic

Description automatically generated

**Narrative:**  Classes were identified through a series of CRC cards. These cards stand for Class, Responsibility, Collaborator and are used to brainstorm series of classes/objects within a system. In this instance, we took the approach of both the system administrator and a user through roleplay. How does the user interact with a website. Some areas of the system are composed of other classes. When a user visits the Events tab, they will be greeted with class Event Calendar which is part of the Events Tab. The same goes for the Help  Us! Tab.  Classes are composed within it and the user will interact with all.

Diagram

Description automatically generated**Database Design:**

**Database Definitions**

**User Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **UserID** | Unique Identifier for user | **4 bytes** | UInt | **Primary** |
| **LastName** | User’s last name | **Max 255 bytes** | String | N/A |
| **FirstName** | User’s First Name | **Max 255 bytes** | String | N/A |

**Volunteer Application Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **ApplicationID** | Unique Identifier for Applications | **4 bytes** | UInt | **Primary** |
| **Applicant**  **Information** | Information regarding applicant | **Max 255 bytes** | String | N/A |

**Volunteer Page Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **Volunteer Positions** | Positions available for volunteers | **Max 255 bytes** | String | **Primary** |
| **Volunteer**  **Application**  **Link** | Link to Volunteer Application | **Max 255 bytes** | String | N/A |
| **Availability**  **Information** | Availability Information on Volunteer Position | **Max 255 bytes** | String | N/A |
| **Contact**  **Information** | Volunteer’s Contact Information | **Max 255 bytes** | String | N/A |

**Help Us! Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **Volunteer**  **Information** | Volunteer’s Information | **Max 255 bytes** | String | **Primary** |
| **Volunteer**  **Application** | Uses Volunteer Application information | **Max 255 bytes** | String | **Foreign** |

**Event Calendar Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **EventName** | Name of Event | **Max 255 bytes** | String | **Primary** |
| **Event**  **Description** | Description of the event | **Max 255 bytes** | String | N/A |
| **EventDate** | The Date of the Event | **4 bytes** | Date | N/A |
| **EventTime** | The Time of the Event | **4 bytes** | Int | N/A |

**Events Tab Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **Event Information** | Information about upcoming events | **Max 255 bytes** | String | **Primary** |

**Adoption Application Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **Adopter Information** | An Adopter’s information | **Max 255 bytes** | String | **Primary** |
| **Address** | Adopter’s Street Address | **Max 255 bytes** | String | N/A |
| **Phone Number** | Adopter’s Phone Number | **8 bytes** | Unsigned Long Int | N/A |
| **Email** | Adopter’s Email | **Max 255 bytes** | String | N/A |
| **State** | Adopter’s State | **Max 255 bytes** | String | N/A |
| **Zip code** | Adopter’s Zip code | **2 bytes** | Int | N/A |

**Adopt Page Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **Adopt Information** | Information about current adoptions | **Max 255 bytes** | String | **Primary** |
| **Adoption Application Link** | **Uses Adop**tion Application information and links to it | **Max 255 bytes** | String | **Foreign** |

**Donation Application Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **Donor FirstName** | **Donor’s First Name** | **Max 255 bytes** | String | **Primary** |
| **Donor LastName** | **Donor’s Last Name** | **Max 255 bytes** | String | N/A |
| **Donation Amount** | **Amount** D**onated by User** | **8 bytes** | Double | N/A |

**Donate Tab Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **Donation Information** | Information from the Donor and amount donated | **Max 255 bytes** | String | **Primary** |
| **Donation Form** | Donation Form using PayPal | **Max 255 bytes** | String | **Foreign** |

**Paypal Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **Donor Information** | Donor’s information | **Max 255 bytes** | String | **Primary** |
| **Donation Amount** | The amount donated saved as a decimal (double) | **Max 255 bytes** | String | N/A |

**Contact Us Page Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **Organization Information** | General information and contact methods | **Max 255 bytes** | String | **Primary** |
| **Social Media Information** | Information and social media handles | **Max 255 bytes** | String | N/A |
| **User FirstName** | User’s First Name | **Max 255 bytes** | String | N/A |
| **User LastName** | User’s Last Name | **Max 255 bytes** | String | N/A |
| **User Phone Number** | User’s Phone Number | **8 bytes** | Unsigned Long Int | N/A |
| **User Email** | User’s Email | **Max 255 bytes** | String | N/A |
| **User Message** | User created message to administrator | **Max 255 bytes** | String | N/A |

**System Admin Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **AdminID** | Unique Identifier for Administrator | **4 bytes** | UInt | **Primary** |
| **LastName** | System Admin Last Name | **Max 255 bytes** | String | N/A |
| **FirstName** | System Admin First Name | **Max 255 bytes** | String | N/A |

**Animal Profile Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **Name** | Name of the associated animal | **Max 255 bytes** | String | **Primary** |
| **Breed** | The Breed of associated animal | **Max 255 bytes** | String | N/A |
| **Age** | Age of associated animal | **4 bytes** | Int | N/A |
| **Biography** | A brief Biography of animal | **Max 255 bytes** | String | N/A |
| **Animal Image** | An Image of the animal | **10 kilobytes** | Binary Large Object (BLOB) | N/A |

**Home Page Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **News Feed Information** | The information displayed on the news feed | **Max 255 bytes** | String | **Primary** |

**About Us Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary Or Foreign Key?** |
| **About Us Information** | Information displayed in About Us | **Max 255 bytes** | String | **Primary** |
| **Mission Statement** | Mission statement that is displayed | **Max 255 bytes** | String | N/A |

**Navigation Diagram:**

**Diagram, schematic

Description automatically generated**

**Screen Layouts:**

Graphical user interface

Description automatically generated

This would be the screen layout for a user entering information to RSVP for an event that would be on the calendar. This layout supports use cases 51-57. From here a user will be able to manage attendees, share events and access directions to the event.

Graphical user interface, text, application

Description automatically generated   
This would be the screen layout for a user entering information for applying for adopting after navigating through the adopt tab. This layout supports use cases 14 and 15.

Graphical user interface, application

Description automatically generated  
This would be the screen layout for someone entering information for the company to reach back out to them after navigating through the contact us tab. This layout supports use cases 64-67. These use cases consist of adding contact information, editing the information, deleting information, and changing methods of contact.

Graphical user interface

Description automatically generated  
This would be the screen layout for a user trying to send a donation when navigating through the donate tab. This layout supports use cases 21-23 and 26-28. These use cases consists of managing donations from the front-end and back-end and giving users the options for anonymous donations, sharing donations and specifying donation amounts.

**Physical Architecture Design**

**Design Procedures for Security Concerns and Non-functioninuirements:**

The procedures needed for the nonfunctional requirements include:

* Acquiring an SSL Certification – Domain host
* Signing up for a WordPress
* Controlling site access

Securing the client’s website is a crucial step within implementing the system. It allows our users to feel comfortable sharing personal information such as their names, phone numbers, and addresses. Vulnerability is always a concern with webpages, and we can prevent this with the use of WordPress. WordPress has various methods for protecting user websites. Some include firewalls, monitoring suspicious activity, security testing, through encryption, and data backup and recovery.

Encryption is an important aspect to securing a website. Strong encryption helps ensure privacy and security. WordPress encrypts all sites, including ones that have custom domain hosts. Encryption is through Secure Socket Layer (SSL) Certificates, and it is a standard with WordPress. As a result, WordPress does not offer the option to disable as it would compromise the security of the WordPress site. WordPress automatically installs an SSL certificate, but this is also provided with the DreamHost domain host.

Firewalls are another procedure taken with WordPress. Like encryption, WordPress provides these and the processes alert the company about unauthorized attempts to access a WordPress account. The same can be said for monitoring suspicious activity and security testing. WordPress constantly watches web traffic and monitors suspicious activities. Some security measures are in place to help protected against distributed denial of service (DDoS) attacks. These attacks can make the site temporarily unavailable, which could hurt the organization.

Security testing is regularly checked with WordPress. Potential vulnerabilities within the site can cause for the client’s system to be compromised and WordPress ensures that this will not happen. A bug bounty program is also operated which improves security by allowing people to discover the vulnerabilities within the system.

**Gantt Chart**

**A picture containing chart

Description automatically generated**

**Task Dependencies**

**A picture containing chart

Description automatically generated**

**Text

Description automatically generated  
Chart, treemap chart

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence  
Chart, treemap chart

Description automatically generated**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Chart, treemap chart

Description automatically generated**

**Elaboration Phase Prototype**

**Use Case 1 – 10: About Us Prototype**

The following image represents the About Us tab. From this page, users can view all the information about the organization on this page and have access/click to the links in Quick Links section to jump to the items of the organization want to focus on which are donation, volunteer, and adoption form. From the backend, the client can add, edit, create, design, and delete all the information on this tab and create, add, and delete navigate links in the Quick Links section.

Graphical user interface

Description automatically generated 

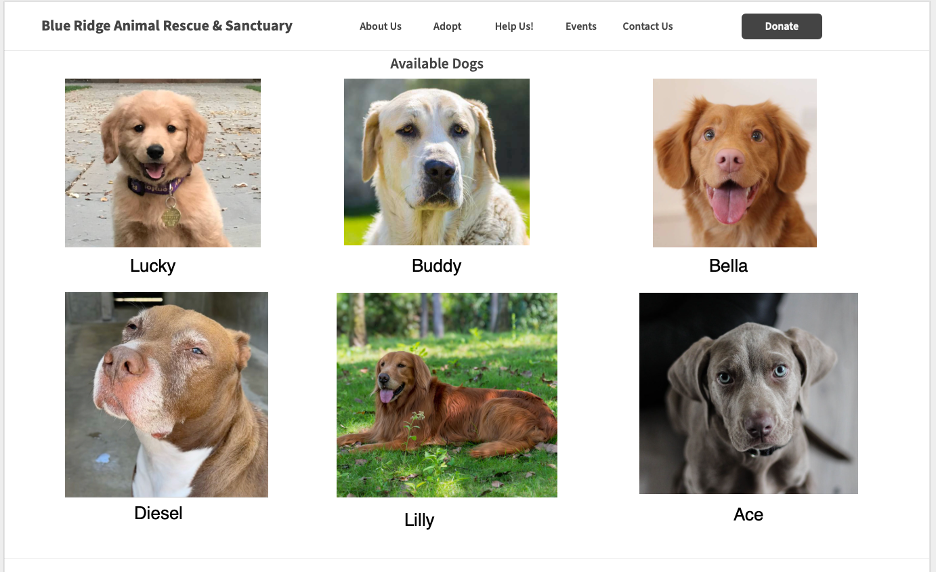
**Use Case 14 & Use Case 15: Adoption Application**

This is a prototype of the adoption application. A user will be able to access this through Blue Ridge’s webpage and will be able to create and apply. They will be required to input information for the organization, and, and the discretion of the organization,  they can delete it.

Graphical user interface, text, application

Description automatically generated

**Use Case 11-13 & Use Case 16-17**

****The Prototype shown will involve adding pets, editing pets, and deleting pets. These are from the administrative side, and the admin can do so as pleased. This is the user view and what we expect the user will see when viewing Blue Ridge’s webpage. This will more than likely involve the adopt tab.

**Use Case 18-20: Animal Biography**

The use cases involved in this prototype display the user view of pet information. When the user visits the web page, we expect that they will view animal information as they look to adopt. Information here is added through the admin and can be changed as needed.

   
**Graphical user interface

Description automatically generated**

**Use Case 21-29: Donation Form**

The prototype below shows the user view of the donation form. Users can create and submit donations to the organization, and, if needed, they can remain anonymous. The organization will be able to view these donations on their end and manage them as necessary.

Graphical user interface, chart

Description automatically generated

**Use Case 32, 33, 37, & 41-43: Volunteer Information**

The first prototype displayed involves use cases 32, 33, 37, and 41-43. It displays the user view of the website, and the information that the client may provide within the Help Us! Page. From here, the user will be able to see information regarding volunteering within the organization, positions and potential availability for volunteering. From the administrative point of view, the client will be able to alter information on this portion of the website. Once altered, changes will be made to the user's end.

Graphical user interface, text, application

Description automatically generated

**Use Cases 34-35, 36, 38, 40, 41-43, 44, & 45-47: Volunteer Application**

This prototype displays the user view of the volunteer application. Accessing this will be through the first prototype displayed or through Help Us! Tab. Once the user accesses this tab, they will be able to fill out the form and submit when finished. This prototype involves use cases 34-35, 36, 38, 40, 41-43, 44, and 45-47. From the backend, the client should be able to receive these submitted applications and view/delete them as necessary.

**Graphical user interface

Description automatically generated**

**Use Case 48-50: Events**

Table

Description automatically generatedThe following image represents the Event tab prototype. This prototype includes a 7-day calendar with time slots on each day to allow the system administrator to add, edit, and delete events as needed. It also allows users to access important event information and RSVP if necessary.

**Use Case 58-67**

The final prototype below shows a mockup of the contact form on the user end. On the back end, the admin will be able to manage their contact information displayed and managed frequently asked questions. Users can manage their information as necessary and change their contact preferences if needed.

Graphical user interface, application

Description automatically generated